



ソフトウェアシステムのマニ ュアル

v1.7.4

目次

1. ようこそ	1
2. 更新説明	3
2.1. アップグレードに関する注意事項	3
2.2. Mech-Visionの更新説明	7
2.3. Mech-Vizの更新説明	19
2.4. Mech-Centerの更新説明	34
2.5. プロジェクトを1.7.0バージョンへアップグレードする方法	38
3. ソフトウェアのインストールガイド	44
4. Mech-Vision	52
4.1. Mech-Visionの概要	52
4.1.1. 専門用語・概念	52
4.1.2. ユーザーインターフェイス	54
4.1.2.1. メニューバー	57
4.1.2.2. ツールバー	61
4.1.2.3. プロジェクトリスト	61
4.1.2.4. ステップライブラリ	64
4.1.2.5. プロジェクトツールバー	67
4.1.2.6. プロジェクト編集エリア	68
4.1.2.7. プロジェクト設定エリア	71
4.1.2.8. ログバー	77
4.1.3. ソリューションの構築手順	79
4.2. 基本操作	80
4.2.1. ソリューションライブラリに関する操作	81
4.2.1.1. ソリューションライブラリを開く	81
4.2.1.2. ソリューションやプロジェクトをフィルター/検索	83
4.2.1.3. ソリューションやプロジェクト情報を表示	84
4.2.1.4. ソリューションやプロジェクトのヘルプマニュアルを閲覧	84
4.2.1.5. ソリューションライブラリからソリューションを新規作成	85
4.2.2. ソリューションに関する操作	86
4.2.2.1. ソリューションを新規作成して保存	87
4.2.2.2. ソリューションを開く	88
4.2.2.3. ソリューションの自動読み込み	89
4.2.2.4. ソリューションの名前を変更	90
4.2.2.5. ソリューションを閉じる	90
4.2.2.6. ソリューションのファイル構造	91
4.2.3. プロジェクトに関する操作	92
4.2.3.1. プロジェクトを新規作成して保存	93
4.2.3.2. プロジェクトの名前を変更	95
4.2.3.3. プロジェクトを開く	96
4.2.3.4. プロジェクトを実行してデバッグ	97
4.2.3.5. プロジェクトを閉じる	98
4.2.3.6. プロジェクトの自動読み込み	99
4.2.3.7. プロジェクト番号を調整	99

4.2.3.8. 既存のプロジェクトをソリューションに変換	100
4.2.3.9. プロジェクトのファイル構造	101
4.2.4. ロボット通信設定に関する操作	102
4.2.4.1. Adapter通信設定	102
4.2.4.2. EtherNet/IP	104
4.2.4.3. Modbus TCP	105
4.2.4.4. PROFINET	106
4.2.4.5. ロボット型番を選択	107
4.2.4.6. Siemens PLC Snap 7	108
4.2.4.7. TCP/IP	110
4.2.4.8. Mitsubishi MC	113
4.2.4.9. UDP	115
4.2.4.10. 読み込みファイルを取得	116
4.2.4.11. ロボットの選択	116
4.2.4.12. インターフェースサービスを起動	119
4.2.4.13. ロボットライブラリツールに関する操作	119
4.2.5. ステップに関する操作	122
4.2.5.1. ステップを検索・追加・削除	123
4.2.5.2. ステップ間のつながりを作成・削除	124
4.2.5.3. ステップの入力・出力を表示	126
4.2.5.4. ステップの異なる状態を表示	127
4.2.5.5. ステップを実行して可視化出力結果を表示	127
4.2.5.6. ステップのコメントを追加・変更・表示	133
4.2.5.7. ステップの組合せの基本操作	135
4.2.5.8. ステップの組合せのパラメータを変更	137
4.2.5.9. ステップの組合せをカスタマイズ	142
4.3. ステップの参照情報	151
4.3.1. 2Dマッチング	164
4.3.2. 2D位置姿勢を3D位置姿勢に変換（正投影に基づく）	169
4.3.3. 2D形状マッピング	170
4.3.4. 3D位置姿勢低精度推定（マルチモデル）	171
4.3.5. 3D位置姿勢低精度推定（V2）	181
4.3.6. 3D位置姿勢低精度推定	190
4.3.7. 3D位置姿勢高精度推定（簡易版）	199
4.3.8. 3D位置姿勢高精度推定（マルチモデル）	203
4.3.9. 3D位置姿勢高精度推定	219
4.3.10. ワーク認識	233
4.3.10.1. ワーク認識の可視化設定ツール	234
4.3.11. 全てのパラメータを受け入れる	239
4.3.12. 位置姿勢にラベルを付ける	240
4.3.13. カメラの歪みによる不正確な位置姿勢を調整	241
4.3.14. オフセットによって位置姿勢を調整	242
4.3.15. 傾きによって位置姿勢を調整	242
4.3.16. 正確な経路を得るまで目標点を調整	243
4.3.17. 位置姿勢を点群の表面に調整	243
4.3.18. 経路の時計回り方向を調整	244
4.3.19. 中継	244

4.3.20. マスク内の対応する画像を抽出	246
4.3.21. マスク内の対応する点群を抽出	247
4.3.22. 2D経路を抽出	250
4.3.23. 背景除去	250
4.3.24. 画像二値化処理	251
4.3.25. ブロブ解析	255
4.3.26. ブールリストの論理演算	256
4.3.27. プロジェクト間にデータを送信	257
4.3.28. 絶対値を計算	257
4.3.29. 両3Dベクトル間の角度を計算	258
4.3.30. マスク面積を計算	259
4.3.31. 箱の寸法を計算	259
4.3.32. 3Dベクトルの外積を計算	260
4.3.33. 対角線の長さを計算	260
4.3.34. 直径と厚さを計算	261
4.3.35. 視差画像を計算	261
4.3.36. 指定方向に沿って両位置姿勢の距離を計算	262
4.3.37. 位置姿勢間の距離を計算	263
4.3.38. 3Dベクトルの内積を計算	264
4.3.39. エッジ点群の法線ベクトルを計算	264
4.3.40. 平面度を計算	264
4.3.41. ヒストグラムを計算	266
4.3.42. 二つの位置姿勢の角度を計算	267
4.3.43. 2つの線分の交点を計算	268
4.3.44. 線分と円との交点を計算	270
4.3.45. 指定軸方向の点群長さを計算	272
4.3.46. マスク間の距離を計算	273
4.3.47. 指定直線上のマスクの長さを計算	274
4.3.48. 平均諸調値を計算	274
4.3.49. 長方形の指定辺の中点を計算	275
4.3.50. マスクの最小外接長方形を計算	275
4.3.51. 3Dベクトルの単位ベクトルを計算	275
4.3.52. 点群の法線ベクトルを計算してエッジを推定	276
4.3.53. 点群の法線ベクトルを計算してフィルタリング	281
4.3.54. 平行度を計算	283
4.3.55. 指定高度のピクセルサイズを計算	285
4.3.56. 平面の幅を計算	286
4.3.57. 点群の曲率を計算	286
4.3.58. 平面点群の位置姿勢とサイズを計算	287
4.3.59. 長方形の寸法と位置姿勢を計算	290
4.3.60. 輪郭度の平面度を計算 (ダウンサンプリング)	293
4.3.61. 輪郭度の平面度を計算	293
4.3.62. 基準方向に沿った投影距離を計算	294
4.3.63. 長方形の2D位置姿勢を計算	295
4.3.64. Pythonを使用して結果を計算	295
4.3.65. 指定した点群のプロパティを計算	305
4.3.66. 構造化光センサーのキャリブレーション行列を計算	305

4.3.67. ゼロでない領域の中心点を計算	305
4.3.68. 円柱の把持位置姿勢を計算	306
4.3.69. 3Dベクトルの長さを計算	306
4.3.70. キャリブレーションボードの位置姿勢を計算	307
4.3.71. キャリパスツール	307
4.3.72. 2Dカメラ	312
4.3.73. カメラから画像を取得	313
4.3.74. 点群のサイズによって分類	323
4.3.75. 点群の歪み補正	327
4.3.76. モデルフィッティングからの位置姿勢によって点群をフィルタリング	327
4.3.77. 点群リストの要素を削除	328
4.3.78. 点群処理 (GPU)	328
4.3.79. 点群スケーリング	331
4.3.80. 点群平滑化と法線ベクトル計算	331
4.3.81. 3D ボックス内の点群を抽出	332
4.3.82. 点群における法線ベクトルを追加	332
4.3.83. 点群をクラスタリングして要件を満たす点群を出力	333
4.3.84. 連続した線を描画	334
4.3.85. 3D ROI内の位置姿勢を取得	334
4.3.86. 二つの深度画像を比較	335
4.3.87. 数値比較	336
4.3.88. 各入力位置姿勢を一部抽出して新しい位置姿勢を合成	337
4.3.89. 四元数と並進ベクトルから位置姿勢を合成	338
4.3.90. 軸と角度から回転ベクトルを合成	339
4.3.91. 両軸から回転ベクトルを合成 (右手系)	339
4.3.92. 数値から3Dベクトルを合成	340
4.3.93. 制御光源	341
4.3.94. 2D位置姿勢を3D位置姿勢に変換	341
4.3.95. データ型を変換	341
4.3.96. ピクセルを物理的な長さに変換	344
4.3.97. 2D点を3D点に変換	348
4.3.98. 円を2D位置姿勢に変換	348
4.3.99. 画像の色空間を変換	349
4.3.100. 画像をコピー	353
4.3.101. 箱を数える	354
4.3.102. データリスト内の指定された次元の要素の数をカウント	354
4.3.103. 3D点の数を計算	355
4.3.104. 色情報をカウント	356
4.3.105. 対象物のモデルを作成	358
4.3.106. 画像を切り抜く	358
4.3.107. 対象物の寸法を分解	359
4.3.108. 位置姿勢を四元数と並進ベクトルに分解	360
4.3.109. 回転ベクトルをX-Y-Z軸に分解	361
4.3.110. 3Dベクトルを数値に分解	362
4.3.111. ディープラーニングモデルを推論 (Mech-DLK 2.1.0/2.0.0)	362
4.3.112. ディープラーニングモデルパッケージを推論	363
4.3.113. ディープラーニング結果を解析	373

4.3.114. スキャンラインに沿ったディープクラスタリング	376
4.3.115. 深度画像のエンコード	377
4.3.116. 円を検出して測定	377
4.3.117. 直線を検出して測定	378
4.3.118. 長穴を検出して測定	378
4.3.119. 箱検出 (4側面)	382
4.3.120. 箱検出 (最大内接長方形)	382
4.3.121. 箱検出 (最大内接長方形・V2)	383
4.3.122. 箱検出 (2側面)	384
4.3.123. 円心検出	384
4.3.124. コーナー検出	384
4.3.125. エッジ検出	385
4.3.126. マーカーを検出	385
4.3.127. 内接円検出	386
4.3.128. 線分を検出	386
4.3.129. マスク内で最大面積の最初のN個の長方形を検出	389
4.3.130. 遮蔽された対象物を検出	389
4.3.131. 領域の形状特徴を検出	390
4.3.132. 頂点を検出	390
4.3.133. ピクセルサイズを決定	391
4.3.134. しきい値によって数値を二項分類	391
4.3.135. 点群を均等に分割	392
4.3.136. 点群をダウンサンプリング	393
4.3.137. マスクの最小外接長方形を描画	394
4.3.138. 座標系をクイック変換	395
4.3.139. 対象物のインデックスリストをクイック作成	395
4.3.140. 数値リストをクイック作成	396
4.3.141. 位置姿勢をクイック作成	396
4.3.142. 四元数をクイック作成	397
4.3.143. ラベルリストをクイック作成	397
4.3.144. 3Dベクトルをクイック作成	398
4.3.145. 位置姿勢を基準点にクイック指向	398
4.3.146. 2D方法による点群エッジを推定	399
4.3.147. 3D方法による点群エッジを推定	401
4.3.148. 画像の鮮明さを評価	402
4.3.149. HDevEngineによる結果を評価	404
4.3.150. JavaScriptエンジンによる結果を評価	405
4.3.151. 深度画像の変動を評価	405
4.3.152. 2D位置姿勢を評価	406
4.3.153. 外部インターフェース	406
4.3.154. 直方体内の点群を抽出	407
4.3.155. 円柱以内の点群を抽出	409
4.3.156. 3D ROI内の点群を抽出	412
4.3.157. 深度画像の3D ROIにゼロ深度値の領域を抽出	416
4.3.158. 穴埋め処理	416
4.3.159. ラベルによってフィルタリング	418
4.3.160. 位置姿勢によってマスクをフィルタリング	419

4.3.161. 箱外の位置姿勢を除去.....	420
4.3.162. フィルタリング.....	420
4.3.163. 2Dコーナーを取得.....	423
4.3.164. 位置姿勢とオフセットの対応関係を求める.....	423
4.3.165. 2D平面で丸穴を検出.....	424
4.3.166. 指定した内外層で2D輪郭を検出.....	425
4.3.167. 円のフィッティング.....	426
4.3.168. 直線のフィッティング.....	428
4.3.169. 位置姿勢の座標軸方向を反転.....	429
4.3.170. 実際のサイズをピクセルサイズに変換.....	431
4.3.171. Cloud (XYZ-Normal)をCloud (XYZ-RGB)に変換.....	432
4.3.172. Cloud (XYZ-RGB) をカラー画像に変換.....	434
4.3.173. 深度画像を点群に変換.....	435
4.3.174. 視差画像によって深度画像を生成.....	439
4.3.175. NumberListをSize3DListに変換.....	439
4.3.176. NumberListをVariantListに変換.....	440
4.3.177. PoseListをMatrix4Dに変換.....	440
4.3.178. PoseListをPoseListsに変換.....	440
4.3.179. 四元数位置姿勢をオイラー角に変換.....	441
4.3.180. Shape2DListをPose2DListに変換.....	442
4.3.181. VariantをVariantListに変換.....	444
4.3.182. 任意の変数からラベルに変換.....	445
4.3.183. VariantListをNumberListに変換.....	445
4.3.184. VariantListをVariantに変換.....	446
4.3.185. 3D位置姿勢を2Dに変換.....	446
4.3.186. 点群で仮想側壁を生成.....	446
4.3.187. 円柱形の点群モデルを生成.....	447
4.3.188. 基準位置姿勢を中心に回転した離散位置姿勢を生成.....	447
4.3.189. 把持位置姿勢を生成.....	448
4.3.190. 点群モデルを生成.....	448
4.3.191. リング形状の点群を生成.....	449
4.3.192. 位置姿勢のオフセットを生成.....	449
4.3.193. 長方形の経路を生成.....	450
4.3.194. 長方形候補を生成.....	450
4.3.195. 指定したサイズの長方形のエッジテンプレートを生成.....	451
4.3.196. 螺旋状経路を生成.....	452
4.3.197. テスト画像を生成.....	452
4.3.198. テスト点群を生成.....	453
4.3.199. 輪郭から経路を生成.....	453
4.3.200. 深度画像から経路を生成.....	459
4.3.201. ジグザグ経路を生成.....	460
4.3.202. 最初の画像を取得.....	460
4.3.203. フランジ位置姿勢を取得.....	461
4.3.204. 深度画像で最高領域を取得.....	462
4.3.205. 最高層の点群を取得.....	464
4.3.206. 最高層の点を取得.....	467
4.3.207. 最高層の位置姿勢を取得.....	468

4.3.208. 最高スコアの結果を取得	469
4.3.209. リング形状の点群リストをフィルタリング	470
4.3.210. データをグループ化	470
4.3.211. 2D位置姿勢をグループ化	471
4.3.212. ヒストグラムマッチング	471
4.3.213. 画像の明るさと色バランサー	472
4.3.214. 画像フィルタリング	477
4.3.215. 画像調整	480
4.3.216. 終了点を挿入して移動パラメータを送信	481
4.3.217. 深度画像の3D ROI外の領域を無効に設定	481
4.3.218. 位置姿勢を逆変換	485
4.3.219. 四元数を逆変換	486
4.3.220. 位置姿勢のZ値をしきい値と比較	487
4.3.221. ラベル接続	487
4.3.222. 正ポリゴンに沿った分布する位置姿勢を保持	488
4.3.223. ラベルマッピング	488
4.3.224. ラインスキャンレーザカメラ	490
4.3.225. 2Dテンプレートを読み込む	497
4.3.226. 経路点の読み込みと変換	498
4.3.227. 2D経路を読み込む	498
4.3.228. 校正球を位置決めする	499
4.3.229. 位置姿勢の軸の方向を基準点に指向	500
4.3.230. 2Dテンプレートを作成	501
4.3.231. 深度画像をカラー画像にマッピング	503
4.3.232. 複数の把持位置姿勢にマッピング	503
4.3.233. マスクによって画像領域を抽出	505
4.3.234. マスククラスタリング	507
4.3.235. マスクフィルタリング	507
4.3.236. マスクをメッシュ化	507
4.3.237. マスクの論理演算	508
4.3.238. 線分間の角度を測定	510
4.3.239. 円の測定	513
4.3.240. 円から円までの距離を測定	514
4.3.241. 円から線分までの距離を測定	517
4.3.242. 点から円までの距離を測定	520
4.3.243. 点から点までの距離を測定	523
4.3.244. 点から線分までの距離を測定	526
4.3.245. 線分間の距離を測定	529
4.3.246. ギャップ幅を測定	532
4.3.247. 点から点までの高さの差を測定	533
4.3.248. 点から基準線までの高さの差を測定	536
4.3.249. 点から平面までの高さの差を測定	539
4.3.250. 最長の線分を測定	541
4.3.251. 軸に平行な方向に沿った平面の高さの差を測定	542
4.3.252. 測定結果を計算して表示	543
4.3.253. 点群リストをマージ	543
4.3.254. データをマージ	544

4.3.255. 深度画像をマージ.....	545
4.3.256. ラベルリストをマージ.....	545
4.3.257. 線分リストをマージ.....	546
4.3.258. 近似高度の点群をマージ.....	546
4.3.259. 点群をマージ.....	547
4.3.260. 位置姿勢リストをマージ.....	548
4.3.261. マスク画像をマージ.....	549
4.3.262. 画像のモフォロジー処理.....	550
4.3.263. 点群を指定方向に沿って移動.....	556
4.3.264. 位置姿勢をZ軸に沿って点群表面に移動.....	556
4.3.265. 法線ベクトルの計算.....	558
4.3.266. 通知.....	559
4.3.267. 数値スケーリング.....	560
4.3.268. 数値演算.....	560
4.3.269. 光学文字認識.....	561
4.3.270. 円柱の位置姿勢オフセット.....	562
4.3.271. 正投影の逆変換.....	562
4.3.272. 正投影.....	564
4.3.273. パック.....	568
4.3.274. パレット情報認識.....	569
4.3.275. 経路計画.....	569
4.3.275.1. 経路計画設定ツール.....	571
4.3.275.2. 画面.....	575
4.3.275.3. ロボットハンド.....	582
4.3.275.4. シーンの物体.....	586
4.3.275.5. 対象物.....	588
4.3.275.6. ステップパラメータ.....	590
4.3.276. サイクルトリガー.....	591
4.3.277. 透視変換.....	592
4.3.278. 対象物の把持範囲.....	592
4.3.279. 配置する対象物の位置姿勢を計画.....	594
4.3.280. 位置姿勢のある軸を指定された方向に向ける.....	595
4.3.281. 点群クラスタリング.....	597
4.3.282. 点群フィルタリング.....	600
4.3.283. 点群形状検出器.....	604
4.3.284. 点のフィルタリング.....	608
4.3.285. 位置姿勢を一括調整.....	611
4.3.285.1. 位置姿勢編集ツール.....	613
4.3.286. 位置姿勢フィルタリング.....	618
4.3.287. 対象物座標系で四元数ベクトルによって位置姿勢を回転.....	619
4.3.288. 対象物座標系で行列によって位置姿勢を変換.....	619
4.3.289. 統計データで位置姿勢の繰返し精度をチェック.....	620
4.3.290. 把持位置姿勢を予測（複数タイプ）.....	627
4.3.291. 把持位置姿勢を予測（V2）.....	640
4.3.291.1. 物流業界（セマンティックセグメンテーション）に適応可能なパラメータ.....	643
4.3.291.2. 物流業界（対象物検出）に適応可能なパラメータ.....	647
4.3.291.3. スーパーマーケットに適応可能なパラメータ.....	653

4.3.291.4. ケーブルに適用可能なパラメータ	657
4.3.291.5. 薬の外箱に適用可能なパラメータ	659
4.3.292. 出力	661
4.3.293. ステップの組合せ	663
4.3.294. 2D形状を処理	670
4.3.295. 点を平面に投影	674
4.3.296. 3D点群を2D画像に投影	675
4.3.297. バーコードを読み取る	678
4.3.298. 画像を読み取る (V2)	678
4.3.299. 画像を読み取る	678
4.3.300. 対象物の寸法を読み込む	679
4.3.301. 点群を読み取る (V2)	682
4.3.302. 点群を読み取る	683
4.3.303. ファイルから位置姿勢を読み取る	683
4.3.304. QRコードを読み取る	684
4.3.305. STLファイルを読み取る	684
4.3.306. 3D ROIの中心を読み取る	685
4.3.307. 基準位置姿勢を記録して変換を計算	685
4.3.308. 2D ROI画像のスケール復元	687
4.3.309. 画像の座標系を補正	689
4.3.310. リング形状の対象物の位置姿勢を補正	691
4.3.311. 近すぎる位置姿勢を除去	692
4.3.312. 点群の点を一部除去	692
4.3.313. マスクのノイズを除去	693
4.3.314. 重複対象物を除去 (V2)	693
4.3.315. 重複対象物を除去	699
4.3.316. 重複ポリゴンを除去	705
4.3.317. 点群の点を一部除去	708
4.3.318. マスク以外のポリゴンを除去	708
4.3.319. 指定した下付き文字に従って並べ替える	709
4.3.320. インデックスによって要素を並べ替える	710
4.3.321. データを繰り返して連結	711
4.3.322. リスト内の要素を置き換える	712
4.3.323. リストを逆順に並べる	713
4.3.324. 指定軸が基準方向と最小角度になるように位置姿勢を調整	714
4.3.325. 指定した位置姿勢によって画像を回転	717
4.3.326. 指定した軸と角度を中心に位置姿勢を回転	717
4.3.327. 指定された軸を中心に位置姿勢を回転	718
4.3.328. ローカル座標系で位置姿勢を回転	719
4.3.329. 位置姿勢を目標方向に回転 (制限なし)	720
4.3.330. 対称性制約で位置姿勢を指定方向に回転	720
4.3.331. 位置姿勢の軸を指定方向に回転	721
4.3.332. 画像保存	723
4.3.333. 位置姿勢の近傍領域を3D ROIとして保存	726
4.3.334. 結果をXMLファイルに保存	726
4.3.335. 結果をファイルに保存	727
4.3.336. 経路点を保存	728

4.3.337. ステップパラメータをファイルに保存	728
4.3.338. 2D ROI内の画像をスケーリング	728
4.3.339. 深度画像分割	735
4.3.340. 点群をMech-Vizに送信	738
4.3.341. 位置姿勢の四元数を設定	741
4.3.342. 位置姿勢の並進ベクトルを設定	742
4.3.343. 画像表示	743
4.3.344. 点群と位置姿勢を表示	745
4.3.345. 深度画像を平滑化処理	748
4.3.346. 経路を平滑化	749
4.3.347. ソートしてインデックスリストを出力	749
4.3.348. ソートと階層化	750
4.3.349. 点群をソート	751
4.3.350. 入力したスコアで位置姿勢をソート	752
4.3.351. 位置姿勢のXYZ値で位置姿勢をソート	754
4.3.352. 2D位置姿勢をソート	755
4.3.353. 3D位置姿勢をソート	756
4.3.354. リング形状の画像領域をまっすぐにする	766
4.3.355. しきい値によって数値を分類	767
4.3.356. 実数減算	770
4.3.357. 一部領域で低いスコアの位置姿勢を除去(NMS)	770
4.3.358. 対象物の回転対称性	772
4.3.359. 2Dテンプレートマッチング	774
4.3.360. テスト	779
4.3.361. 経路点のマッチング	779
4.3.362. 画像変換	780
4.3.363. 平面の点群を指定平面に合わせる	781
4.3.364. 点群変換 (直行ロボット)	783
4.3.365. 点群変換	785
4.3.366. 位置姿勢を変換 (直行ロボット)	790
4.3.367. 位置姿勢を変換	792
4.3.368. 基準方向に沿って位置姿勢を並進	798
4.3.369. 位置姿勢を指定方向に移動	799
4.3.370. 指定した方向と距離によって位置姿勢を並進	803
4.3.371. 3Dベクトルで位置姿勢を並進	804
4.3.372. ローカル座標系で位置姿勢を並進	804
4.3.373. ブール値によるトリガー制御	805
4.3.374. トリガー	806
4.3.375. 出力の数を制限	806
4.3.376. 位置姿勢の数を制限	808
4.3.377. データをアンパックしてマージ	809
4.3.378. データをアンパック	810
4.3.379. マスクによって2D位置姿勢を検証	813
4.3.380. カメラのパラメータ補正を検証して計算	813
4.3.381. 箱の寸法が有効であるかどうかを検証	814
4.3.382. 箱形状の対象物のマスクを検証	815
4.3.383. 位置姿勢が3D ROI内にあるかどうかを検証	816

4.3.384. 必要なラベルであるかどうかを検証	817
4.3.385. マスクに2D位置姿勢があるかどうかを検証	819
4.3.386. 点群が要求を満たすかどうかを検証	820
4.3.387. 基準方向との角度によって位置姿勢が有効かどうかを検証	821
4.3.388. 位置姿勢が遮蔽されているかどうかを検証	822
4.3.389. 法線ベクトルの偏差が大きい領域を抽出	823
4.3.390. 3Dベクトル演算	824
4.3.391. 把持位置姿勢を検証	825
4.3.392. 範囲内の深度を可視化	825
4.3.393. 画像で情報を可視化	826
4.3.394. 大型部品の測定	826
4.3.394.1. 大型部品の測定プロジェクトの構築手順	827
4.3.394.2. 丸穴の中心位置姿勢と直径を計算	829
4.3.394.3. 長穴の中心位置姿勢と長軸・短軸を計算	832
4.3.394.4. 平面点群を抽出	835
4.3.394.5. 深度画像から点群を分割	837
4.3.394.6. 位置姿勢をカスタマイズされた座標系に変換	838
4.3.395. ソート	840
4.3.395.1. 2Dソート	840
4.3.395.2. 3Dソート	842
4.3.395.3. 把持位置姿勢をソート	844
4.3.396. 3D位置姿勢推定	847
4.3.397. マスクが対応するカラー画像を取得	853
4.3.398. 最高層のカラー画像を取得	855
4.3.399. 最高層のマスクを取得	863
4.3.400. 点の数が制限を超えた点群を除外	868
4.3.401. ROI以外の位置姿勢を除去	869
4.3.402. 点群前処理	871
4.3.403. 画像のデータとパラメータを保存	874
4.4. ハンド・アイ・キャリブレーションの使用ガイド	874
4.4.1. ロボットハンド・アイ・キャリブレーションの実行手順を選択	875
4.4.2. 自動キャリブレーション (Eye to Hand)	877
4.4.3. 自動キャリブレーション (Eye in Hand)	900
4.4.4. 標準キャリブレーション法を使用する手動キャリブレーション (Eye to Hand)	924
4.4.5. 標準キャリブレーション法を使用する手動キャリブレーション (Eye in Hand)	941
4.4.6. TCPタッチ法を使用する手動キャリブレーション (Eye to Hand)	956
4.4.7. TCPタッチ法を使用する手動キャリブレーション (Eye in Hand)	974
4.4.8. 直行ロボットのハンド・アイ・キャリブレーション	991
4.4.9. ハンド・アイ・キャリブレーション (Eye to Eye)	1008
4.4.10. 参考情報	1022
4.4.10.1. キャリブレーションの関連概念	1022
4.4.10.2. キャリブレーション原理	1024
4.4.10.3. 直行ロボットのキャリブレーション説明	1031
4.4.10.4. キャリブレーション結果の確認と分析	1034
4.4.10.5. よくある問題と解決策	1045
4.5. 測定モードの使用ガイド	1052
4.5.1. 測定モードを使ってみる	1054

4.5.2. 主要なアルゴリズムの紹介	1057
4.6. 補助ツールの使用ガイド	1058
4.6.1. カメラビューア	1058
4.6.2. 静的背景を取得して干渉を除去	1060
4.6.3. ROI 設定	1063
4.6.4. シーンの点群を設定	1064
4.6.5. マッチングモデル・把持位置姿勢エディタ	1066
4.6.6. ディープラーニングサーバー	1067
4.6.7. ディープラーニングモデルパッケージ管理ツール	1068
4.6.8. パラメータレシピ	1075
4.6.9. データ保存	1090
4.6.10. ログ	1094
5. Mech-Viz	1097
5.1. Mech-Viz の概要	1097
5.1.1. 基本概念	1097
5.1.2. ユーザーインターフェイス	1100
5.1.2.1. メニューバー	1102
5.1.2.2. ツールバー	1105
5.1.2.3. リソース	1105
5.1.2.4. 仮想空間	1109
5.1.2.5. ワークフロー	1112
5.1.2.6. ロボット	1115
5.1.2.7. 衝突検出	1116
5.1.2.8. 計画履歴	1117
5.1.2.9. その他	1119
5.1.2.10. ログ	1121
5.1.3. プロジェクトの構築手順	1121
5.2. リソース設定	1122
5.2.1. ロボット	1123
5.2.2. ロボットハンド	1126
5.2.3. 対象物	1131
5.2.3.1. 対象物の回転対称性	1132
5.2.3.2. 対象物の把持範囲	1135
5.2.4. 地面	1137
5.2.5. シーンの物体	1137
5.3. ワークフローの構築	1140
5.3.1. ワークフローの構築	1141
5.3.2. 仮想ロボットを動かしてみよう	1142
5.3.3. ロボット実機を動かしてみよう	1143
5.4. ステップライブラリ	1143
5.4.1. 移動ステップ	1145
5.4.1.1. 動的移動	1145
5.4.1.2. 移動	1151
5.4.1.3. グリッドによる移動	1157
5.4.1.4. リストによる移動	1164
5.4.1.5. 外部移動	1170
5.4.1.6. 相対移動	1173

5.4.2. DI DO	1181
5.4.2.1. DI をチェック	1181
5.4.2.2. DI リストをチェック	1183
5.4.2.3. DI を待つ	1184
5.4.2.4. DO を設定	1186
5.4.2.5. DO リストを設定	1187
5.4.3. 論理トポロジ	1188
5.4.3.1. 道標によって異なる分岐を実行	1188
5.4.3.2. メッセージによって異なる分岐を実行	1190
5.4.3.3. ステップの組合せ	1192
5.4.3.4. ステップの組合せの出口	1192
5.4.3.5. 道標を設定	1194
5.4.4. パレタイジング	1196
5.4.4.1. カスタマイズのパレットパターン	1196
5.4.4.2. 混載パレットパターン	1204
5.4.4.3. 複数把持のパレタイジング	1218
5.4.4.4. 事前計画パレットパターン	1227
5.4.4.5. ビジョン処理による継続パレタイジング	1233
5.4.4.6. パレットパターンの基本設定	1238
5.4.5. ロボットハンド	1242
5.4.5.1. ロボットハンドをチェック	1242
5.4.5.2. 関節角度を取得	1243
5.4.5.3. 最大可搬質量設定	1245
5.4.5.4. 把持状態を設定	1246
5.4.5.5. ロボットハンドを切り替え	1247
5.4.5.6. 制御を移転	1248
5.4.6. サービス	1249
5.4.6.1. 通知	1249
5.4.7. ツール	1251
5.4.7.1. 分類	1251
5.4.7.2. カウンター	1254
5.4.7.3. 完了確認	1256
5.4.7.4. インデックスを変更	1257
5.4.7.5. ステップをリセット	1259
5.4.7.6. 待つ	1260
5.4.8. 経路	1261
5.4.8.1. 経路ステップの組合せ	1261
5.4.9. ビジョン	1261
5.4.9.1. ビジョン処理の結果をチェック	1261
5.4.9.2. パレットの位置姿勢を更新	1263
5.4.9.3. 把持済み対象物を更新	1265
5.4.9.4. シーンの物体を更新	1269
5.4.9.5. ビジョン処理の結果を使い切る	1273
5.4.9.6. ビジョン処理による認識	1276
5.4.9.7. ビジョン処理による移動	1277
5.4.10. その他	1302
5.4.10.1. 実行を中止	1302

5.4.11. 共通パラメータ	1302
5.4.11.1. 移動ステップの共通パラメータ	1303
5.4.11.2. 把持された対象物との衝突検出モード	1304
5.4.11.3. 非移動ステップの共通パラメータ	1305
5.4.11.4. 基本的な移動設定	1306
5.4.11.5. 目標点タイプ	1307
5.4.11.6. 関節角度の制約条件	1308
5.5. プロジェクトのシミュレーションと最適化	1309
5.5.1. 衝突検出	1310
5.5.2. 計画履歴	1313
5.5.3. ロボット実機の実行	1315
5.5.4. 計算設定	1316
5.6. ツールの使用	1316
5.6.1. モデルエディタ	1316
5.6.1.1. インターフェイス	1324
5.6.1.2. STEP/STP ファイル有効性の判断	1327
5.6.2. デパレタイズ用吸盤コンフィギュレータ	1328
5.6.3. 配列タイプグリッパコンフィギュレータ	1329
5.6.4. パレットパターンエディタ	1329
5.7. 付録	1330
5.7.1. ロボットモデルの作成とインポート	1330
5.7.1.1. ロボット構造のタイプ	1338
5.7.1.2. [robot]_algo.json ファイル属性	1340
5.7.1.3. [robot]_algo.json ファイルのパラメータ説明	1342
5.7.2. その他	1343
5.7.2.1. 設定-オプション	1343
5.7.2.2. ディスプレイ-ディスプレイ設定	1343
5.7.2.3. ツール-ビジョン処理の記録を設定	1344
6. Mech-Center	1347
6.1. Mech-Centerの概要	1347
6.1.1. メニューバー	1348
6.1.2. ツールバー	1350
6.1.3. サービスステータスバー	1352
6.1.4. プロジェクトのステータスバー	1352
6.1.5. ログバー	1353
6.2. Mech-Centerを使ってみる	1353
6.3. Mech-Interface	1355
6.4. 標準インターフェース	1359
6.5. Adapter (アダプター)	1361
6.5.1. Adapterの概要	1362
6.5.2. Adapterジェネレーター	1364
6.5.3. Adapterプログラミングのガイド	1371
6.5.3.1. Adapterプログラミングスタイル規則	1372
6.5.3.2. 抽象親クラスインターフェース	1374
6.5.3.3. Adapter utilパッケージ	1392
6.5.3.4. インターフェースの取得	1394
6.5.3.5. Mech-Visionインターフェース	1394

6.5.3.6. Mech-Vizインターフェース.....	1398
6.5.3.7. RobotServiceインターフェース	1407
6.5.3.8. その他のインターフェース.....	1411
6.5.4. Adapterプログラミングの例	1413
6.5.4.1. Mech-Visionのみを使用してビジョンポイントを送信.....	1413

1. ようこそ

Mech-Mindビジョン関連ソフトウェアのユーザーズマニュアルへようこそ！

Mech-Mindソフトウェアシステムでは、ビジョン処理、経路計画、ロボット制御は、主にMech-Vision、Mech-Viz、Mech-Centerのビジョン関連ソフトウェアで実現されます。

更新説明

Mech-Visionの更新説明

Mech-Vision 最新バージョンの新機能や機能最適化、問題修復について説明します。

[詳細はこちら](#)

Mech-Vizの更新説明

Mech-Viz 最新バージョンの新機能や機能最適化、問題修復について説明します。

[詳細はこちら](#)

Mech-Centerの更新説明

Mech-Center 最新バージョンの新機能や機能最適化、問題修復について説明します。

[詳細はこちら](#)

アップグレードに関する注意事項

Mech-Vision、Mech-Viz、Mech-Centerのアップグレードに関する注意事項について説明します。

[詳細はこちら](#)

ソフトウェアのインストールガイド

ソフトウェアのダウンロード、インストール、アップグレード、アンインストール、メンテナンス、変更などについて説明します。

[詳細はこちら](#)

プロジェクトのアップグレード方法

ソフトウェアのアップグレードに必要なプロジェクト変更について説明します。

[詳細はこちら](#)

ソフトウェアのユーザーズマニュアル

Mech-Vision

画像処理ソフトウェア

[詳細はこちら](#)

Mech-Viz

ロボット制御ソフトウェア

[詳細はこちら](#)

Mech-Center

通信中枢・制御センター

[詳細はこちら](#)

2. 更新説明

本章では、1.7バージョンMech-Vision、Mech-VizおよびMech-Centerの新機能や機能最適化、問題修復について説明します。



Mech-Vision、Mech-VizおよびMech-Centerをアップグレードする前に、以下の内容をよくお読みください。

ソフトウェアやプロジェクトのアップグレードに関する注意事項については、[アップグレードに関する注意事項](#)と[プロジェクトを1.7.0バージョンへアップグレードする方法](#)をお読みください。

Mech-Vision 1.7バージョンの新機能や機能最適化、問題修復については、[Mech-Vision 1.7バージョンの更新説明](#)をお読みください。

Mech-Viz 1.7バージョンの新機能や機能最適化、問題修復については、[Mech-Viz 1.7バージョンの更新説明](#)をお読みください。

Mech-Center 1.7バージョンの新機能や機能最適化、問題修復については、[Mech-Center 1.7バージョンの更新説明](#)をお読みください。

2.1. アップグレードに関する注意事項

1.7.2 バージョンへのアップグレード

1.7.2バージョンのMech-Visionでは、[ディープラーニングモデルパッケージを推論](#) ステップが追加され、「ディープラーニングモデルパッケージを推論（CPU）」および「ディープラーニングモデルパッケージを推論（Mech-DLK 2.2.0+）」ステップが削除されました。

Mech-Visionが1.7.2バージョンにアップグレードされた後、過去バージョンのプロジェクトの「ディープラーニングモデルパッケージを推論（CPU）」および「ディープラーニングモデルパッケージを推論（Mech-DLK 2.2.0+）」ステップは、自動的に「ディープラーニングモデルパッケージを推論」に置き換えられます。「ディープラーニングモデルパッケージを推論」ステップは削除された「ディープラーニングモデルパッケージを推論（CPU）」および「ディープラーニングモデルパッケージを推論（Mech-DLK 2.2.0+）」ステップと同じ機能を提供するため、この変更はプロジェクトの通常の使用には影響しません。

1.7.2バージョンのMech-Visionでは、過去バージョンで最適化されたモデルパッケージを使用す

る場合、「ディープラーニングモデルパッケージを推論」ステップでモデルパッケージを初めて実行する際に時間がかかる場合があります。

1.7.0 バージョンへのアップグレード

以下では、Mech-Vision、Mech-VizおよびMech-Centerを1.7.0バージョンにアップグレードする際の注意事項について説明します。

Mech-Vision

「ソリューション」について

Mech-Vision 1.7.0バージョンでは、ロボット選択、通信設定、ビジョンプロジェクトの構築、経路計画をまとめて行うことができる「ソリューション」機能を追加しました。使用中に以下の事項に注意してください。

- 標準インターフェース通信またはAdapter通信を使用する場合、ロボットプログラムが呼び出すMech-Visionプロジェクト番号はMech-Centerからではなく、Mech-Visionプロジェクトリストから取得するようにしました。
- 過去バージョンのMech-Visionプロジェクトについては、アップグレード後にプロジェクトの変換を完了させる必要があります。詳細については、[プロジェクトを1.7.0バージョンへアップグレードする方法](#)をご参照ください。

「Mech-Centerの通信サービス」の統合について

Mech-Vision 1.7.0バージョンでは、インターフェース通信サービスを内蔵しているため、以下の事項に注意が必要です。

- Mech-Visionソリューションを使用し、通信方式が標準インターフェースまたはAdapter通信の場合、Mech-Centerソフトウェアは必要ありません。
 - Mech-Visionのツールバーで[**ロボット通信設定**]をクリックし、ロボットを選択して通信方式を設定し、[**適用**]をクリックすると、インターフェースサービスが自動的に起動されます。
 - ロボットのメーカーがABB、FANUC、YASKAWA、KAWASAKI、KUKA、UR、TM、ELITE、JAKAの場合、デフォルトで選択される通信設定は、Mech-Mindが提供するロボット側の標準インターフェースのサンプルプログラムで使用されている通信設定と一致しています。その他のメーカーのロボットの場合、デフォルトで選択されている通信設定はTCP/IPです。
 - インターフェース設定を行う際に、「ソリューションを開くとインターフェースサービスを自動的に起動」にチェックを入れることが可能です。これにより、次回ソリューションを開くと、プロジェクトがロボット側と直接通信ようになります。
 - 通信の関連ログは、Mech-Visionのログバーの「Console」タブで確認できます。
 - Adapterのインターフェース設定を行う際に、Adapterプロジェクトフォルダを選択すると、選択したフォルダは自動的に現在のソリューションフォルダに移動されます。

- ソリューション内のAdapterプログラムを変更して有効にするには、Adapterプログラムを再起動し、インターフェースサービスを再起動する必要があります。
- Mech-Centerで「PC起動時にMech-Centerを自動的に実行」のチェックを外す必要があります。
- 通信方式がVizティーチングの場合や、Mech-Vizを使用して複雑な作業を実施する場合は、Mech-Centerソフトウェアが必要です。
 - まずMech-Centerを起動します。次にMech-Visionを起動し、過去バージョンと同様に通信設定を行う必要があります。
 - Mech-Centerで「PC起動時にMech-Centerを自動的に実行」にチェックを入れること、Mech-CenterからMech-Visionを起動することを推奨します。

キャリブレーションについて

Mech-Vision 1.7.0バージョンでは、キャリブレーションの設定手順内で、ロボットの選択、プログラムの読み込み、ロボットの接続ができるようになりました。プロジェクトがソリューションに割り当てられていない場合、キャリブレーション機能は使用できません。

ステップについて

Mech-Vision 1.7.0バージョンでは、ステップに関する注意事項は以下の通りです。

一部のステップを削除

Mech-Vision 1.7.0バージョンでは、下記のいくつかのステップを削除しました。

親グループ-子グループ	削除したステップ
2D特徴抽出-ほか	指定コーナーおよび寸法によって長方形を検出
2D汎用処理-ほか	ポリゴンを配置
ディープラーニング-未知対象物の把持	把持のヒートマップから位置姿勢を計算、把持位置姿勢を予測（同じタイプ）、各画素の把持可能な確率を予測
測定-3D長さ/距離	点から二つの平面の交線までの距離を計算、点から平面までの距離を計算
古いバージョンのステップ	画像で情報を可視化、障害物に応じて目標点を調整
位置姿勢-並進/方向を調整	障害物に応じて目標点を調整（V2）
その他	把持可能な長方形を検出

過去バージョンのディープラーニング関連ステップについて

欠陥検出、エッジ検出、画像分類、インスタンスセグメンテーション、対象物検出、ディープラーニングモデルを推論（Mech-DLK 2.1.0/2.0.0）ステップはメンテナンスが終了しており、以

降のリリースで削除される予定です。

- ハードウェアタイプがGPUで、欠陥検出、画像分類、インスタンスセグメンテーション、対象物検出、ディープラーニングモデルを推論（Mech-DLK 2.1.0/2.0.0）ステップを使用する必要がある場合は、新しいバージョンのステップ「ディープラーニングモデルパッケージを推論（Mech-DLK 2.2.0+）」を使用してください。
- ハードウェアタイプがCPUで、画像分類、インスタンスセグメンテーション、対象物検出、ディープラーニングモデルを推論（Mech-DLK 2.1.0/2.0.0）ステップを使用する必要がある場合は、新しいバージョンのステップ「ディープラーニングモデルパッケージを推論（CPU）」を使用してください。

Mech-Viz

Mech-Viz 1.7.0バージョンでは、ロボットの命名が標準化され、多数のロボットモデル名が変更されたため、ロボットモデルライブラリでロボットを再選択する必要があります。

Mech-Viz 1.7.0バージョンでは、一部のロボットのDOポート制限が変更されたため、一部の過去バージョンのプロジェクトを開くとポップアップが表示されることがあります。このポップアップを消すには、robot_abilityで該当するロボットのdigital_out_rangeを手動で変更する必要があります。

Mech-Viz 1.7.0バージョンでは、箱の位置姿勢の有効範囲を手動で設定できるようになりました。箱を使用する過去バージョンのプロジェクトでは、箱の位置姿勢の有効範囲は、箱の底面から上への無限拡張から、箱の上面から1メートル以内まで調整されます。

「ビジョン処理による移動」ステップの「デパレタイズ用吸盤」の実行モードが完全に再構築され、元のソフトウェアの機能の一部が分割され、新しいバージョンの機能に直接対応しないため、箱の複数把持に関連するパラメータを再設定する必要があります。

「ビジョン処理による移動」ステップの「デパレタイズ用吸盤-デパレタイズ用吸盤（複数把持）」の実行モードでは、箱の組み合わせ後のモデルは、過去バージョンの1つの大きな箱から複数の小さな箱に変更されました。モデルはより実際の状況に近いですが、モデルの変更により、ロボットの経路計画で追加の衝突が発生することがあります。

吸盤設定ウィンドウは、リソースパネルで開くことができるロボットハンド設定ウィンドウに統合されています。吸盤設定を使用するプロジェクトについては、吸盤を再設定する必要があります。

ロボットハンドの衝突モデルはobj形式のみに対応し、stl形式には非対応になりました。衝突検出の設定画面から、関連する衝突設定パラメータが削除されました。

ロボットハンドの衝突モデルが3Dモデルと全く同じファイル名とファイルタイプを持っている場合、ソフトウェアは3Dモデルを衝突モデルに置き換えます。

Mech-Center

Mech-Center 1.7.0バージョンは、Mech-VisionおよびMech-Viz1.7.0バージョン以降と併用する

必要があります。

ソリューションのプロジェクトに「経路計画」ステップを使用する場合、ロボット側でインターフェイスプログラムを再読み込み、Mech-Visionの「ロボット通信設定」で設定する必要があります。

2.2. Mech-Visionの更新説明

以下では、Mech-Vision 1.7バージョンの新機能や機能最適化、問題修復について説明します。

Mech-Vision 1.7.4バージョンの更新説明

以下では、Mech-Vision 1.7.4バージョンの機能最適化および問題修復について説明します。

機能最適化

ドングルライセンスにソフトウェアの使用地域制限を追加

ソフトウェアの販売地域に対し、ドングルライセンスにソフトウェアの使用地域制限を追加しました。

Mech-Vision 1.7.4では、**ヘルプ**・**ソフトウェアについて**をクリックして、ソフトウェアライセンスと使用地域などを確認できます。

「システム言語」オプションを削除

Mech-Vision 1.7.4バージョンから、ソフトウェアの表示言語設定（**設定**・**オプション**・**基本設定**・**言語**）から「システム言語」オプションを削除しました。PCのオペレーティングシステムの言語が、ソフトウェアでサポートされていない言語（英語、簡体中国語、日本語、韓国語以外の言語）である場合、ソフトウェアの初回インストール時のデフォルト言語は英語になります。

この最適化により、以下の2つの問題が解決されました。

- 過去のバージョンでは、ソフトウェアがPCのオペレーティングシステムの言語をサポートしていない場合、ソフトウェアでリンクが開かない問題と、ステップ名が正しく表示されない問題を修正しました。
- ソフトウェアがPCの言語設定に基づいて言語パックを正しく設定できない問題を修正しました。

シーンの点群の設定はリスト形式の点群に対応

1.7.4バージョンから、**プロジェクトアシスタント**でリスト形式の点群を背景として使用する点群に設定できるようになりました。

一部ソリューションやプロジェクトの取下げ

ソリューションライブラリの品質とユーザーエクスペリエンスを保証するために、1.7.4では一部のソリューションやプロジェクトを一時的に取り下げました。最適化が完了した後、後続のバージョンで再度公開される予定です。詳細は下表の通りです。

種類	名前（太字で表示されているのはソフトウェアに組み込まれているソリューションやプロジェクト）
部品供給	クランクシャフト（大型） 、 単一のパーテーション 、浅型ボックス、アルミニウムインゴット、ルーフレール、クランクシャフト（小型）、コンデンサ、等速ジョイント、銅線、コネクティングロッド、トラックリンク、圧縮機、4面壁の鉄製ボックス（点群品質に対して高い要求）、ガスケット（小型）、ギアシャフト、ハブ、鉄球、鉄製ボックス（4本柱）、全貌が撮影できなかったボックス、鉄製ボックス（2面壁）、板バネ、大型リング、複数のパーテーション、整列して並べられた小型部品、PVCパネル、ローター、リングギア、回転軸、バラ積みの小型部品（高反射性）、バラ積みの小型部品（ボルト）、バラ積みの小型部品（低反射性）、スリーブ、スペーサー、棒鋼、正方形レンガ、トレイ、バルブコネクタ
パレタイジング・デパレタイジング	ドラム
位置決め・組立	ホイールハブの位置決め 、自動充電の位置決め、ボルトの位置決め、車のフレーム位置決め、車のドア枠の位置決め、ネジ穴の位置決め、ホイールハブのバルブコアの位置決め
商品仕分け	ケーブル、葉の外箱



- 上記のソリューションやプロジェクトを既に使用しているユーザーにとって、引き続き使用に影響はありません。
- 過去バージョンのソフトウェアで上記のソリューションやプロジェクトを取得した場合、1.7.4バージョンにアップグレード後、ソリューションやプロジェクトのカードがまだ存在していますが、使用できない可能性があります。

問題修復

Mech-Vision 1.7.4バージョンでは、以下の問題を修復しました。

- 「点群をクラスタリングして要件を満たす点群を出力」ステップで、ごくわずかな確率で誤った数の点群が出力される問題を修正しました。
- ニューラルネットワークに入力する画像がシングルチャンネル画像の場合、「ディープラーニングモデルパッケージを推論」ステップで「画像で欠陥のマスクを描画」機能を有効にし

てプロジェクトを実行すると、OpenCV関連のエラーが発生する問題を修正しました。

- Mech-DLK 2.4.1バージョンからエクスポートした対象物検出モデルパッケージを「ディープラーニングモデルパッケージを推論」ステップで読み込む際、モデルパッケージのエクスポート時に「インスタンスの最大数」が1で、モデルパッケージの推論の「ハードウェアタイプ」が「CPU」の場合、モデルパッケージの推論時間が長すぎる問題を修正しました。
- 「ディープラーニングモデルパッケージを推論」ステップを使用して、一部の入力画像のインスタンスセグメンテーションを行う際に、ごくわずかな確率で発生するOpenCV関連のエラーを修正しました。
- 直行ロボットのキャリブレーション済みの外部パラメータファイルは、「位置姿勢を変換（直行ロボット）」および「点群変換（直行ロボット）」で使用できない問題を修正しました。
- 「経路計画」ステップでは、「移動」と「相対移動」ステップの「ロボットに送信するか」パラメータを「計画するが送信しない」に設定した場合でも、上記ステップによって経路点がロボットに送信される問題を修正しました。

Mech-Vision 1.7.2バージョンの更新説明

以下では、Mech-Vision 1.7.2バージョンの新機能や機能最適化について説明します。

新機能

「把持位置姿勢を予測（V2）」で薬の外箱のシーンに対応

把持位置姿勢を予測（V2） ステップでは、薬の外箱のシーンに対応し、バラ積みされた薬の外箱を仕分けすることができます。

また、薬の外箱に対応可能なソリューション（ディープラーニングモデル搭載）は、ソリューションライブラリから取得できます。

「ディープラーニングモデルパッケージ管理ツール」について

ディープラーニングモデルパッケージ管理ツール では、モデルパッケージの推論効率を「バッチサイズ」と「精度」の両方で設定できます（Mech-DLK 2.4.1以降バージョンによってエクスポートされたモデルパッケージのみ、推論効率を設定可能です）。



Mech-Vision1.7.2バージョン以降とMech-DLK 2.4.1バージョン以降によってエクスポートされたディープラーニングモデルパッケージとの併用が推奨されています。

Mech-Vision 1.7.1バージョンでは、Mech-DLK 2.4.1バージョンによってエクスポートされたディープラーニングモデルパッケージも使用できますが、いくつかの [互換性問題](#) もあります。

ステップ「ディープラーニングモデルパッケージを推論」を追加

Mech-Vision1.7.2バージョン以降、「ディープラーニングモデルパッケージを推論（CPU）」と「ディープラーニングモデルパッケージを推論（Mech-DLK 2.2.0+）」のステップは **ディープ**

ラーニングモデルパッケージを推論 ステップに統合されました。

Mech-Vision 1.7.2バージョンで過去バージョンのプロジェクトを開くと、「ディープラーニングモデルパッケージを推論（CPU）」と「ディープラーニングモデルパッケージを推論（Mech-DLK 2.2.0+）」ステップは自動的に「ディープラーニングモデルパッケージを推論」に置き換わります。

このステップはMech-DLKによってエクスポートされた単体モデルまたは直列モデルに対して推論を行い、推論結果を出力することができます。Mech-DLK 2.2.0以降バージョンによってエクスポートされたモデルパッケージのみ対応しています。



Mech-DLK 2.4.1以降、モデルパッケージには、単体モデルと直列モデルの2種類があります。

- 単体モデルパッケージ：1つのディープラーニングアルゴリズムモジュールのみを搭載したモデルパッケージのことを指します。例：「インスタンスセグメンテーション」モデル。
- 直列モデルパッケージ：ディープラーニングアルゴリズムモジュールの複数のモデルを、前のモデルの出力が次のモデルの入力となる直列形式で搭載したモデルパッケージのことを指します。例えば、モデルパッケージに「対象物検出」と「インスタンスセグメンテーション」の2つのモデルがある場合、モデルの推論順序は**対象物検出・インスタンスセグメンテーション**となり、「対象物検出」の出力は「インスタンスセグメンテーション」の出力になります。「対象物検出」モデルの出力は、「インスタンスセグメンテーション」モデルの入力として使用されます。

このステップが直列モデルの推論を行う場合、直列モデルの推論結果は「ディープラーニング結果を解析」ステップで解析することができます。

ステップ「ディープラーニング結果を解析」を追加

Mech-Vision 1.7.2バージョンでは、**ディープラーニング結果を解析** ステップが追加され、「ディープラーニングモデルパッケージを推論」ステップから出力される直列モデルの推論結果を解析できるようになりました。

ソリューションライブラリについて

Mech-Vision 1.7.2バージョンでは、以下のソリューションとプロジェクトをソリューションライブラリに追加しました。

- ソリューション：ブレーキディスク（単一セル）。
- プロジェクト：薬の外箱、一般的な部品認識、バラ積みの小型部品（ボルト）。



プリインストールされているソリューションやプロジェクトを使用する場合は、最新バージョンのソフトウェアを使用してください。

機能の最適化

アルゴリズムの改善

Mech-Vision 1.7.2バージョンでは、ステップの処理速度を向上させるために3Dマッチング関連ステップ（「3D位置姿勢低精度推定」、「3D位置姿勢高精度推定」など）を最適化しました。

ステップとパラメータ名の最適化

Mech-Vision 1.7.2バージョンでは、以下のステップとパラメータ名の最適化を行いました。

	Mech-Vision 1.7.2より前のバージョン	Mech-Vision 1.7.2
ステップ名	2D位置姿勢で3D位置姿勢を調整	2D位置姿勢を3D位置姿勢に変換
パラメータ名	指定軸の角度による候補位置姿勢をフィルタリング（「3D位置姿勢低精度推定（V2）」、「3D位置姿勢高精度推定」などのステップ）	モデルの回転角度による位置姿勢をフィルタリング

Mech-Vision 1.7.1バージョンの更新説明

以下では、Mech-Vision 1.7.1バージョンでの問題修正について説明します。

問題修復

Mech-Vision 1.7.1バージョンでは、以下の問題を修正しました。

- Mech-Visionを起動するために「.vis」プロジェクトファイルをダブルクリックすると、「インターフェースサービス」が起動しない問題を修正しました。
- ネットワークの遅延により、Mech-Visionがオンラインドキュメントのジャンプに失敗する場合がある問題を修正しました。
- LSR、DEEPシリーズのカメラを「カメラから画像を取得」ステップで接続し、「深度画像の座標系に補正」パラメータのチェックが外れている場合に、カラー点群の表示でエラーが発生する問題を修正しました。
- 「カメラから画像を取得」ステップの名前を変更した後、仮想モードで「データパス」を選択しても反映しないことを修正しました。
- 「ワーク認識の可視化設定ツール」のモデルライブラリでモデルにチェックを入れた後にチェックを外すと、再度モデルライブラリに入ったときにチェックが入ったままになる問題を修正しました。
- 「ワーク認識の可視化設定ツール」を開いた後、プロジェクト実行中にそれを直接終了するとソフトウェアがクラッシュする問題を修正しました。

「3D位置姿勢高精度推定（簡易版）」ステップのサーフェスマッチング結果でエラーが発生

- する可能性がある問題を修正しました。
- 「マッチングモデル・把持位置姿勢エディタ」で位置姿勢のマニピュレータのタイプを切り替えた後、マニピュレータの軸が太くなる問題を修正しました。
- 3D ROI編集画面を開いたままMech-Visionを終了すると、3D ROI編集画面が正しく閉じない問題を修正しました。
- キャリブレーション中の画像取得中にカメラがフレーム落ちしたり、カメラの接続が切れたりするとソフトウェアがクラッシュする問題を修正しました。
- ロボットが接続されている状態でカメラの外部パラメータのキャリブレーションを行うために、ロボットの接続を解除した後でないと実行モードとロボットの制御方法を選択できない問題を修正しました。

Mech-Vision 1.7.0バージョンの更新説明

以下では、Mech-Vision 1.7.0バージョンの新機能や機能最適化、問題修復について説明します。

新機能

Mech-Visionによるビジョンアプリケーションのワンストップ展開をサポートする新しい「ソリューション」

Mech-Vision 1.7.0バージョンでは、「ソリューション」を追加しました。ソリューションとは、ビジョンアプリケーションを実現するために必要なロボットと通信、ビジョン処理、経路計画などの機能構成やデータの集りのことです。

ソリューションは1つまたは複数のプロジェクトで構成されています。プロジェクトを単独で使用することは推奨されず、ソリューションに割り当てて使用する必要があります。

ソリューション機能のサポートにより、ロボットの選択、ロボット通信方式の設定、ビジョンプロジェクトの構築、ロボットの動作経路計画などを1つのソリューションで行うことができ、ビジョンアプリケーションへの実装を実現しています。



- ビジョン結果と簡単な経路計画が必要なプロジェクトでは、Mech-Visionという1つのソフトウェアだけでアプリケーション全体を展開することが可能です。
- 標準インターフェース通信またはAdapter通信を使用する場合、ロボットプログラムが呼び出すMech-Visionプロジェクト番号はMech-Centerからではなく、Mech-Visionプロジェクトリストから取得するようにしました。
- 過去バージョンのMech-Visionプロジェクトについては、アップグレード後にプロジェクトの変換を完了させる必要があります。詳細については、[プロジェクトを1.7.0バージョンへアップグレードする方法](#)をご参照ください。

Mech-Visionに統合したMech-Centerの通信サービス機能

Mech-Vision 1.7.0バージョンでは、「ロボット通信設定」機能を追加しました。この機能により、Mech-Centerソフトウェアを起動せずに、Mech-Visionソフトウェア内で通信設定を行った後に、ロボットのインポートや選択、ロボットなどの外部機器との通信を行えるようになります。

した。

統合した通信サービスを使用する際に、次のことに注意してください。

- Mech-Visionソリューションを使用し、通信方式が標準インターフェースまたはAdapter通信の場合、Mech-Centerソフトウェアは必要ありません。
 - Mech-Visionのツールバーで[**ロボット通信設定**]をクリックし、ロボットを選択して通信方式を設定し、「適用」をクリックすると、インターフェースサービスが自動的に起動されます。
 - ロボットのメーカーがABB、FANUC、YASKAWA、KAWASAKI、KUKA、UR、TM、ELITE、JAKAの場合、デフォルトで選択される通信設定は、Mech-Mindが提供するロボット側の標準インターフェースのサンプルプログラムで使用されている通信設定と一致しています。その他のメーカーのロボットの場合、デフォルトで選択されている通信設定はTCP/IPです。
 - インターフェース設定を行う際に、「ソリューションを開くとインターフェースサービスを自動的に起動」にチェックを入れることが可能です。これにより、次回ソリューションを開くと、プロジェクトがロボット側と直接通信するようになります。
 - 通信の関連ログは、Mech-Visionのログバーの「Console」タブで確認できます。
 - Adapterのインターフェース設定を行う際に、Adapterプロジェクトフォルダを選択すると、選択したフォルダは自動的に現在のソリューションフォルダに移動されます。
 - ソリューション内のAdapterプログラムを変更して有効にするには、Adapterプログラムを再起動し、インターフェースサービスを再起動する必要があります。
 - Mech-Centerで「PC起動時にMech-Centerを自動的に実行」のチェックを外す必要があります。
- Mech-Visionソリューションを使用しない場合、または通信方式がVizティーチングの場合、Mech-Centerソフトウェアが必要です。
 - まずMech-Centerを起動します。次にMech-Visionを起動し、過去バージョンと同様に通信設定を行う必要があります。
 - Mech-Centerで「PC起動時にMech-Centerを自動的に実行」にチェックを入れること、Mech-CenterからMech-Visionを起動することを推奨します。

複合ステップ「経路計画」

Mech-Vision 1.7.0バージョンでは、「経路計画」機能を追加しました。この機能により、入力されたビジョンポイントを使用してロボットの動作経路を計画し、衝突のない経路を出力することができます。金属部品供給に適しており、簡単な経路計画に対応しています。

複合ステップ「ワーク認識」

Mech-Vision 1.7.0バージョンでは、「ワーク認識」ステップを追加しました。このステップでは、点群前処理、3Dマッチング、積み重ねられた対象物除去などのビジョン処理機能が統合されています。これにより、ワークの高速認識を実現しています。金属部品のピック&プレース

に適しており、様々な形状と配置状態のワークを認識可能です。

「ようこそページ」

Mech-Vision 1.7.0バージョンでは、「ようこそページ」を追加しました。この画面では、ソフトウェアのバージョン情報だけでなく、ユーザーマニュアルや関連操作へのクイックアクセスを提供しています。

「ソリューションライブラリ」

Mech-Vision 1.7.0バージョンでは、「ソリューションライブラリ」を追加しました。ソリューションライブラリには、部品供給、パレタイジング・デパレタイジング、位置決め・組立、商品仕分け、品質検査の5つの業界に適用可能なソリューションやプロジェクトに加え、サンプルデータも提供しています。初心者には、ソリューションやプロジェクトの添付画像と説明をもとに、適切なソリューションやプロジェクトを選択し、簡単な修正を加えた後、そのまま使用することができます。

Mech-Mindは今後も、最新のソリューションライブラリをダウンロードして使用できるオンラインソリューションライブラリを拡充していく予定です。

「一般的な部品ピッキング」のソリューションテンプレート

Mech-Vision 1.7.0バージョンでは、ソリューションライブラリにソリューションテンプレート「一般的な部品ピッキング」を内蔵しています。様々な形状と配置状態のワーク認識をサポートし、ロボットが「衝突ゼロ」の部品ピッキングを完了することを実現します。4つの手順だけで3Dビジョンソリューションを構築することが可能です。工作機械の加工及び輸送、部品搬送などに適しています。

アルゴリズムの強化

Mech-Vision 1.7.0バージョンでは、下記のいくつかの新しいステップを追加しました。

新しいステップ	説明
3D位置姿勢高精度推定（簡易版）	このステップは、「3D位置姿勢高精度推定」を使いやすくしたもので、より直感的なパラメータ調整により、点群モデルとシーンの点群を正確にマッチングさせて対象物の位置姿勢を出力することが可能です。
重複対象物を除去（V2）	このステップは指定されたルールに従って積み重ねられた対象物のビジョン認識結果を除去できます。「重複対象物を除去」と比較して、このステップは投影方法の処理速度を向上させます。
大型部品測定に関するステップの組合せ	大型部品測定の業界向けに、一般的な機能を搭載したステップの組合せを提供します。現場で簡単な測定プロジェクトを構築する時に役立ちます。

新しいステップ	説明
把持位置姿勢を予測 (V2)	このステップは2D画像と深度画像から把持対象物を認識し、把持位置姿勢を出力できます。
点群の法線ベクトルを計算してエッジを推定	このステップは、法線ベクトルを計算し、点群内の対象物エッジを推定することでエッジ点群を出力することができます。
点群変換 (直行ロボット)	このステップは、入力点群をカメラ座標系または直行ロボットの座標系に変換して出力できます。
位置姿勢を変換 (直行ロボット)	このステップは、入力位置姿勢をカメラ座標系または直行ロボットの座標系に変換して出力できます。
円のフィッティング	このステップは入力した2D画像内の点を円にフィッティングすることができます。通常、計測のシーンに使用されます。
直線のフィッティング	このステップは入力した2D画像内の点を直線にフィッティングすることができます。通常、計測のシーンに使用されます。
2つの線分の交点を計算	このステップは、2つの線分の交点のピクセル座標を計算できます。通常、計測シーンに対象物の特徴点を見つけるために使用されます。
線分と円との交点を計算	このステップは、入力線分または線分と円の双方向延長線の交点のピクセル座標を計算できます。通常、計測シーンに対象物の特徴点を見つけるために使用されます。
2D形状を処理	このステップは、指定方法に従って入力された2値画像の形状を処理できます。通常、計測シーンでの様々な計算を容易にするために対象物の輪郭を処理するために使用されます。
穴埋め処理	このステップは入力された2値画像の穴、すなわち非ゼロピクセルで完全に囲まれた領域を埋めることができます。通常、対象領域の完全な画像を取得し、穴領域の画像欠落による干渉を避けるために、画像分割に使用されます。
画像の鮮明さを評価	このステップは、様々な計算方法を指定することにより、入力画像の鮮明さを定量的に評価することができます。通常、計測シーンでカメラのパラメータや位置調整に役立ちます。
色情報をカウント	このステップは、指定された色空間のカラー画像を入力し、指定されたチャンネル内のピクセル値（平均値、標準偏差、最大値、最小値を含む）をカウントすることができます。通常、計測シーンで画像の色を評価するために使用されます。

新しいステップ	説明
キャリパスツール	このステップは、狭いROIの垂直方向に沿ってエッジポイントまたはエッジポイントペアを検出し、エッジポイントの座標とエッジポイントペア間の距離をピクセル単位で出力することができます。
Shape2DListをPose2DListに変換	このステップでは、入力された3つの2D形状情報リストから、X値リスト、Y値リスト、Theta値リスト（Thetaは傾斜角度）を取り出し、組み合わせて新しい2D位置姿勢リストを作成することができます。

LNXカメラに対応

「ラインスキャンレーザーカメラ」ステップでは、LNXカメラに対応可能なLNXカメラタイプを追加しました。

直行ロボットのハンド・アイ・キャリブレーションに対応

Mech-Vision 1.7.0バージョンでは、キャリブレーションツールを最適化し、直行ロボットのハンド・アイ・キャリブレーションに対応します。直行ロボット専用のキャリブレーション手順を設計し、複雑な設定をせずに直行ロボットをキャリブレーションすることができるようになりました。

また、直行ロボット専用の「点群変換（直行ロボット）」と「位置姿勢を変換（直行ロボット）」ステップを追加しました。Mech-Visionプロジェクトでは、動的な外部パラメータを計算するためにこれらのステップを使用する必要があります。

「ディープラーニングモデルパッケージ管理ツール」を追加

Mech-Vision 1.7.0バージョンでは、「ディープラーニングモデルパッケージ管理ツール」を追加しました。これは、「ディープラーニングモデルパッケージを推論（Mech-DLK 2.2.0+）」と「ディープラーニングモデルパッケージを推論（CPU）」ステップに使用するディープラーニングモデルパッケージを最適化し、実行モードやハードウェアタイプ、モデルパッケージ状態などを管理することが可能です。さらに、このツールはIPCのGPU使用率監視に対応します。

機能の最適化

アルゴリズムの改善

Mech-Vision 1.7.0バージョンでは、下記のいくつかのステップを最適化しました。

最適化したステップ	説明
ディープラーニングモデルパッケージを推論（Mech-DLK 2.2.0+）	モデルパッケージの設定と欠陥判定ルールの設定 （欠陥検出向け）のパラメータグループを追加しました。

最適化したステップ	説明
ディープラーニングモデル パッケージを推論（CPU）	モデルパッケージの設定 パラメータグループを追加しました。
統計データで位置姿勢の繰 返し精度をチェック	異常値処理 パラメータを追加しました。数値の偏差が設定されたしきい値を超えた場合、 異常値を発見したら即エラー報告 または 異常値を記録してマーク のいずれかを選択することができるようになりました。 異常値を記録してマーク を選択した場合、異常値は出力テキストファイルに赤色でマーカーが付けられます。
ピクセルを物理的な長さを 変換	キャリブレーションによる計算 パラメータを追加しました。入力画像を使用して自動的にキャリブレーションを行い、キャリブレーションで得られたカメラパラメータに基づいて、実際の距離を計算します。キャリブレーションを行わない場合よりも、より正確な計算が行われます。
画像二値化処理	DualThreshold、DynamicThreshold の分割操作のタイプを追加しました。
ソートと階層化	各層の開始位置 パラメータを追加しました。層間隔とソートする位置姿勢配列の入力により、ソート後の配列とインデックスを出力することができます。

一部のステップを削除

Mech-Vision 1.7.0バージョンでは、下記のいくつかのステップを削除しました。

親グループ-子グループ	削除したステップ
2D特徴抽出-ほか	指定コーナーおよび寸法によって長方形を検出
2D汎用処理-ほか	ポリゴンを配置
ディープラーニング-未知 対象物の把持	把持のヒートマップから位置姿勢を計算、把持位置姿勢を予測（同じタイプ）、各画素の把持可能な確率を予測
測定-3D長さ/距離	点から二つの平面の交線までの距離を計算、点から平面までの距離を計算
古いバージョンのステップ	画像で情報を可視化、障害物に応じて目標点を調整
位置姿勢-並進/方向を調整	障害物に応じて目標点を調整（V2）
その他	把持可能な長方形を検出

「マッチングモデル・把持位置姿勢エディタ」について

Mech-Vision 1.7.0バージョンでは、マッチングモデル・把持位置姿勢エディタを次のように最

適化しました。

- メイン画面を最適化し、主要な機能を強調し、使いやすさを改善しました。
- ツールバーのレイアウトを最適化し、操作説明を表示するための動画を追加しました。
- カメラを使用した点群取得の流れを最適化し、エッジ点群の取得に対応する機能を追加しました。
- CADファイルをインポートする時の単位選択機能を最適化しました。

「モデルエディタ（古いバージョン）」を削除

Mech-Vision 1.7.0バージョンでは、「モデルエディタ（古いバージョン）」を削除しました。点群モデルの作成や把持位置姿勢の生成が必要な場合、マッチングモデル・把持位置姿勢エディタを使用してください。

6軸ロボットのキャリブレーション手順について

Mech-Vision 1.7.0バージョンでは、6軸ロボットのキャリブレーション手順を次のように最適化しました。

- キャリブレーションの設定手順内では、ロボット選択やロボットプログラムの読み込み、ロボット接続などの機能を追加しました。ロボットへの接続は、キャリブレーションの設定手順内で完了できます。
- 新しいロボットシミュレーション画面（シーンビュー）により、Mech-Vizソフトウェアがなくても、設定したキャリブレーション経路とロボット位置をキャリブレーション中にリアルタイムに確認することができます。
- インターフェイスのレイアウトを最適化し、キャリブレーションのためのガイダンスもより明確になりました。

「パラメータレシピ」について

Mech-Vision 1.7.0バージョンでは、「パラメータレシピ」を最適化しました。[パラメータを更新]をクリックし、ワンクリックでプロジェクト内のパラメータを現在のレシピに同期することができるようになりました。

「プロジェクト編集エリア」について

Mech-Vision 1.7.0バージョンでは、プロジェクト編集エリアのインタラクションデザインを最適化し、グリッドと位置合わせ機能を追加し、必要に応じて位置合わせグリッドの表示有無を設定できます。また、基準線、ステップドラッグ&ドロップ吸着を調整できるようにしました。

コメントしたカメラ名の表示に対応

Mech-Visionでカメラを接続した後、カメラIDにカーソルを合わせると、コメントしたカメラ名とIPアドレスが表示されるようになりました。

長さと角度の単位の初期設定を変更

長さと角度の単位の初期設定を、「ステップに組み込み単位を使用」から「mm」と「°」に変更しました。

日本語と韓国語の言語パックのオンラインおよびオフライン更新をサポート

Mech-Vision 1.7.0バージョンでは、日本語と韓国語のUI言語パックのオンラインおよびオフライン更新に対応しています。さらに、Mech-Mind株式会社からオフラインの日本語の言語パックの取得をサポートしています。言語パックの更新のためにソフトウェアに言語パックファイルをドラッグすることで更新が可能です。

問題修復

ステップ「カメラから画像を取得」について

Mech-Vision 1.7.0バージョンでは、「カメラから画像を取得」ステップでHikonの2Dカメラに接続できない問題を修正しました。

ステップ「Pythonを使用して結果を計算」について

Mech-Vision 1.7.0バージョンでは、ステップ「Pythonを使用して結果を計算」の問題を修正しました。詳細は以下の通りです。

- プロジェクト実行中にこのステップがスムーズに実行できないこと。
- PostListデータ型のマージ時に、出力された位置姿勢の四元数の順序が正しくないこと。

ステップ「位置姿勢を一括調整」について

Mech-Vision 1.7.0バージョンでは、プロジェクト実行中に「位置姿勢を一括調整」によって引き起こされたソフトウェアクラッシュを修正しました。

過去バージョンの更新説明

[Mech-Vision 1.6.2バージョンの更新説明](#)

[Mech-Vision 1.6.1バージョンの更新説明](#)

[Mech-Vision 1.6.0バージョンの更新説明](#)

2.3. Mech-Vizの更新説明

以下では、Mech-Viz 1.7バージョンの新機能や機能最適化、問題修復について説明します。

Mech-Viz 1.7.4バージョンの更新説明

以下では、Mech-Viz 1.7.4バージョンの問題修正について説明します。

機能最適化

ドングル認証のためのソフトウェア地域制限を追加

ソフトウェアの販売地域に対し、ドングル認証のためのソフトウェア地域制限を追加しました。

Mech-Viz 1.7.4では、**ヘルプ**・**ソフトウェアについて**をクリックするとライセンスと地域制限を確認できます。

「言語」のオプションから「システム言語」オプションを削除

Mech-Viz 1.7.4以降は、ソフトウェアの言語オプション（**設定**・**オプション**・**基本設定**・**言語**）から「システム言語」をなくします。コンピュータのオペレーティングシステムの言語がソフトウェアにサポートされていない場合(システム言語は英語、簡体字中国語、日本語、韓国語以外の言語である場合)、初めてソフトウェアをインストールするときに英語をデフォルト言語とします。

この最適化により、以下の2つの問題が解決されました。

- 過去バージョンでは、ソフトウェアがコンピュータのオペレーティングシステムの言語をサポートしていない場合、ユーザーズマニュアルにアクセスできません。
- ソフトウェアがPCの言語設定に基づいて言語パックを正しく設定できません。

ほか

- 「パレット」ステップのパラメータ「パレット」を「パレット位置」に変更しました。
- 「ビジョン処理による移動」ステップに関するログと plan_result に箱の組合せの行数 (box_group_row_count) を追加しました。

問題修復

Mech-Viz 1.7.4バージョンでは、以下の問題を修復しました。

- 仮想空間の右クリックメニューの「削除」機能がプロジェクトのシミュレーションと実行中に無効にならない問題を修復しました。
- 対象物の回転対称性とロボットハンド対称性を同時に使用する時に対象物回転対称性試行が失敗する問題を修復しました。
- 「メッセージによって異なる分岐を実行」ステップの前に移動ステップが配置されていない場合にスキップする時に対応する出口から実行しない問題を修復しました。
- ビジョン処理の結果を再使用したら結果のラベルがなくなる問題を修復しました。

- ログのアイテムをドラッグしたらソフトウェアがクラッシュする問題を修復しました。
- 「ビジョン処理による移動」ステップの「把持総計数制限」パラメータを設定したら実行可能な組み合わせ結果が全部提供されない問題を修復しました。
- 「ビジョン処理による移動」ステップによって設定した吸盤の DO が間違っている問題を修復しました。
- 「ビジョン処理による継続パレタイジング」ステップによってパレットパターンをマッチングする時に方向が間違っている問題を修復しました。
- 対象物モデルに外部モデルを使用する時に衝突検出に検出漏れが発生する問題を修復しました。

Mech-Viz 1.7.2バージョンの更新説明

以下では、Mech-Viz 1.7.2バージョンの機能最適化および問題修復について説明します。

機能最適化

全ての経路点をシミュレート

過去のバージョンでは、移動ステップの「移動目標点を送信」パラメータをチェックしないとその目標点は計画にのみ使用され、ロボット移動のシミュレーションに関与しません。

Mech-Viz 1.7.2バージョンでは、計画した経路点を全部シミュレートするようになりました。

過去のバージョンの設定を使いたい場合は、設定 > オプション > 基本設定で設定することができます。

STL・DAE ソースファイルが削除されずにロボットをエクスポート可能

過去のバージョンでは、ロボットをエクスポートする時にこのロボットの STL、DAE ソースファイルが削除されてしまいます。

Mech-Viz 1.7.2バージョンでは、ロボットをエクスポートしても STL・DAE ソースファイルが保存されます。

「側吸吸盤」を削除

過去のバージョンでは、「ビジョン処理による移動」ステップのパラメータと「リソース」の「ロボットハンド設定」で「側吸吸盤」を選択して設定することができます。

Mech-Viz 1.7.0バージョン以降、「デパレタイズ用吸盤」には「側吸吸盤」のすべての機能が含まれています。

Mech-Viz 1.7.2バージョンでは「側吸吸盤」オプションをなくしました。「側吸吸盤」を設定した場合は、「デパレタイズ用吸盤」を選択して再度設定してください。

オイラー角タイプを18種に補充

過去のバージョンでは、よく使う5種のオイラー角しか対応できません。Mech-Viz 1.7.2バージョンでは18種に補充しました。

「メッセージによって異なる分岐を実行」ステップのパラメータ「外部コマンドを一時的に保管」を削除

Mech-Viz 1.7.2バージョンでは、「メッセージによって異なる分岐を実行」ステップのパラメータ「外部コマンドを一時的に保管」を削除しました。

SCARAロボットのソフトリミット修復

過去のバージョンでは、SCARA ロボットの J3 ソフトリミット設定を保存できません。Mech-Viz 1.7.2バージョンではアルゴリズムを修正し保存できるようになりました。ただし、ロボットの J3 ソフトリミットが変化するため手動で調整する必要があります。

問題修復

Mech-Viz 1.7.2バージョンでは、以下の問題を修復しました。

- 仮想空間におけるロボットハンドの表示（透明度）は、リソースパレットで設定した透明度と一致しない問題を修復しました。
- シーンの箱を設定する時、「シーンの物体設定」ウィンドウで「位置姿勢の有効範囲を設定」が機能しない問題を修復しました。
- 「自動障害物回避」機能を使用するとソフトウェアがクラッシュする問題を修復しました。
- 「リストによる移動」ステップのパラメータ「一度に全ての目標点」をチェックすると経路点のロスが発生する問題を修復しました。
- 「事前計画パレットパターン」ステップで箱の寸法を調整する時にパレタイズする順序がパレットパターンの変化とともに自動調整しない問題を修復しました。
- 「混載パレタイジング」ステップ実行時にすでに箱が配置された位置に箱を配置する問題を修復しました。
- 「ビジョン処理による継続パレタイジング」ステップではパレットパターンを生成するステップを指定してもプロンプトメッセージが表示される問題を修復しました。
- デパレタイジングアルゴリズムは吸盤の隙間を吸着機能がある部分として処理する問題を修復しました。
- 箱を組み合わせた後に出力する結果に箱のラベル情報がない問題を修復しました。

Mech-Viz 1.7.1バージョンの更新説明

以下では、Mech-Viz 1.7.1バージョンの機能最適化および問題修復について説明します。

機能最適化

「ビジョン処理による移動」ステップの「許容干渉」パラメータの初期値を40mmに修正

過去のバージョンでは、「許容干渉」パラメータの初期値が20mmであり、デパレタイジングプロジェクトでは、ビジョン位置姿勢が対象物の中心点から外れたり、対象物の寸法が実際より大きくなったりし、ソフトウェアが受信した箱モデル同士の積み重ねなどの干渉が20mmを超えることがあるため、実際に実行可能な把持が放棄されます。

この問題を解決するためにMech-Viz 1.7.1バージョンでは「ビジョン処理による移動」ステップの「許容干渉」パラメータの初期値を40mmに修正しました。

問題修復

Mech-Viz 1.7.1バージョンでは、以下の問題を修復しました。

リソース - ロボット

- ロボットのソフトリミットを修正するとソフトウェアがクラッシュする問題を修復しました。
- FANUC R-2000iC/270F ロボットの3Dモデルの間違いを修復しました。
- ソフトウェア実行中にロボットパネルで「TCP」と「関節角度」が切り替えられない問題を修復しました。

リソース - ロボットハンド

- ロボットハンド設定ウィンドウを開いているときにツールバーをフリーズします。
- ロボットハンド設定の「TCP キャリブレーション」ウィンドウで、「位置姿勢を追加」ボタンをクリックして「ロボットフランジ位置姿勢を入力」ウィンドウが正確に表示されない問題を修復しました。
- ロボットハンド設定ウィンドウで「キャンセル」ボタンをクリックしても「ロボットハンドの種類」の設定が保存される問題を修復しました。

リソース - 対象物

- 対象物の回転対称設定に「X 軸を中心に」と「Y 軸を中心に」が同時にチェックできる問題を修復しました。
- 対象物設定ウィンドウを開いているときにツールバーをフリーズします。
- デフォルトの対象物の設定が削除される問題を修復しました。

リソース - シーンの物体

- シーンの物体を設定する時にソフトウェアがクラッシュする問題を修復しました。
- シーンの物体設定ウィンドウを開いているときにツールバーをフリーズします。

- ボックスは衝突検出に使用されない場合、ビジョン位置姿勢がボックス外にあると判断される問題を修復しました。

ステップ

- 2つの「相対移動」を互いに依存するように設定すると、ワークフローの実行が無限ループにはまる問題を修復しました。
- 「ビジョン処理による継続パレタイジング」がクラッシュする問題を修復しました。
- 「複数把持のパレタイジング」をリセットしても配置済みの箱がクリアされない問題を修復しました。
- ロボットが移動しないときに「把持済み対象物を更新」ステップは把持済み対象物のモデルを生成できない問題を修復しました。
- 「把持済み対象物を更新」ステップを実行するときにソフトウェアが動かなくなる問題を修復しました。
- 「分類」ステップを「ビジョン処理による移動」ステップの前に配置し、かつ「ビジョン処理による移動」の「同じ対象物/把持点の把持を回避」機能をオンにした時にワークフローの実行が無限ループにはまる問題を修復しました。
- 「ビジョン処理による移動」ステップの「失敗」出口にその他の移動ステップを繋いでもエラーメッセージが表示される問題を修復しました。

把持・配置計画

- 「ビジョン処理による移動」ステップのパラメータ設定では、「Y 軸に沿う」を選択したとき「行ごと把持」が機能しない問題を修復しました。
- 「ビジョン処理による移動」ステップのパラメータ設定では、「組み合わせ方式」を「Y 軸に沿う」に設定すると吸盤のオフセットが間違う問題を修復しました。
- 浮動小数点演算中のエラーのため、「複数把持のパレタイジング」ステップではパレットパターンマッチングができない問題を修復しました。
- 「複数把持のパレタイジング」ステップでは箱の把持数が多い解が優先的に試行されない問題を修復しました。

組み込みツール

- 「パレットパターンエディタ」使用中にクラッシュする問題を修復しました。
- 「配列タイプグリッパ設定」の設定が保存できない問題を修復しました。
- 「ビジョン処理の記録を設定」ウィンドウでビジョン処理の記録が正確に読み込めない問題を修復しました。
- 「パレットパターンエディタ」で既存のパレットパターンの変更が保存できない問題を修復しました。

計画履歴

- トップレベルの移動ステップが全部表示されない問題を修復しました。
- ロボットが指定位置に到達できないために計画が失敗したとき、ロボットハンドモデルが計画履歴に正確に表示されない問題を修復しました。

その他

- ロボットを緊急停止したあと、エラーメッセージが繰り返し表示される問題を修復しました。

Mech-Viz 1.7.0バージョンの更新説明

以下では、Mech-Viz 1.7.0バージョンの新機能や機能最適化、問題修復について説明します。

新機能

「ようこそページ」

ソフトウェアを起動・終了するとき、「ようこそページ」が表示されます。「ようこそページ」でプロジェクトを新規作成し、または最近使ったプロジェクトを開くことができます。

「リソース」パネル

Mech-Viz 1.7.0バージョンでは、「シーンの物体」と「ロボットハンドと対象物」パネルをなくし、プロジェクト構築に必要なリソースを設定・管理する機能はリソースパネルによって実現されます。

プロジェクトとロボット、座標系、ロボットハンド、対象物、地面、シーンの物体、モデルライブラリなどは、「リソース」パネルで設定できます。

- プロジェクト：プロジェクト名、変更の有無、自動的に読み込むに設定されたかを表示します。
- ロボット：現在のプロジェクトに使用するロボットの型番を表示します。
- 座標系：3D シミュレーションエリアに使用するすべての座標系を表示します。
- ロボットハンド：ロボットハンドを作成、削除します。また、追加されたロボットハンドを設定します。
- 対象物：対象物を作成、削除します。また、追加された対象物を設定します。
- 地面：地面の高さを調整します。
- シーンの物体：シーンの物体を作成、削除します。また、追加されたシーンの対象物を設定します。
- モデルライブラリ：プロジェクトに使用するすべてのモデルを管理します。モデルを使用するために、まずモデルライブラリにインポートする必要があります。

「現在のロボットハンドに設定」機能

新バージョンのソフトウェアには「現在のロボットハンド」概念を使用します。「現在のロボットハンド」に設定されたロボットハンドは次のシミュレーション/実行に最初のロボットハンドとして使用されます。

- 非シミュレーション/実行状態で、リソースパネルでロボットハンド名を右クリックしてドロップダウンメニューで「現在のロボットハンドに設定」をクリックすると今使用するロボットハンドを変更します。
- シミュレーション/実行状態で、「ロボットハンドを切り替え」ステップを使用して「現在のロボットハンド」を切り替えます。

「事前計画したパレットパターン」にパレットパターンの自動的生成機能

「事前計画したパレットパターン」ステップにより、パレットの寸法、箱の寸法、箱同士の間隔、箱の山の高さの制限に基づいて箱の数が最も多くて奇・偶数段の積み方が最も合理的で、段数が最も多いパレットパターンが自動的に生成されます。

その他

- 「ファイル」に「プロジェクトを新規作成」オプションを追加しました。
- ROKAE SR3、SR4 ロボットモデルを追加しました。
- 複数のステップを選択してから、ショートカット Ctrl + G を押してそれらを「ステップの組み合わせ」に合成することが可能になりました。Ctrl + Shift + G を押して「ステップの組み合わせ」を分割することも可能になりました。
- 「メッセージによって異なる分岐を実行」に「待機タイムアウト」出口を追加しました。待機タイムアウト時間を設定すると、待機時間がタイムアウトになったら「待機タイムアウト」出口を実行します。
- 「混載パレットパターン」オフラインモードでは事前計算したパレットパターンの保存とインポートが可能になりました。
- 「混載パレットパターン」に箱の方向の指定が可能になりました。
- 「事前計画パレットパターン」に「風車形」積み付けを指定することが可能になりました。
- 「事前計画パレットパターン」に対象物の方向を指定することが可能になりました。

機能最適化

ロボットモデルライブラリの拡充

オンラインロボットモデルライブラリにロボットモデルの数を 214 から 600 以上に増加しました。ABB、DENSO、FANUC、Kawasaki、KUKA、Mitsubishi、Nachi、UR、STAUBLI、YASKAWA など多数のロボットモデルが利用できます。



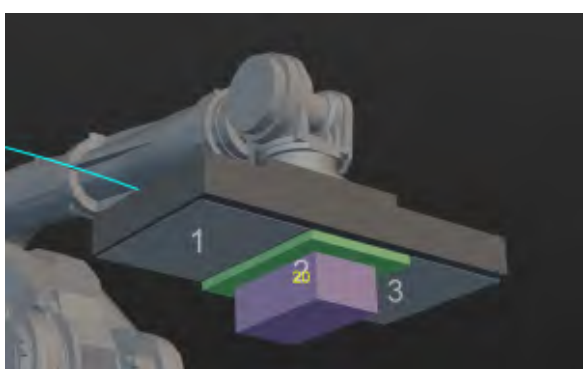
また、ABB、FANUC、Kawasaki、KUKA、Nachi、UR、YASKAWA などのロボットモデルは、パラメータ校正が行われました。

STL 形式のロボットハンド衝突モデル非対応化

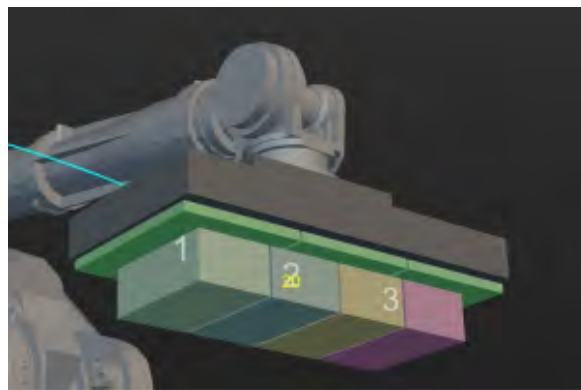
ロボットハンドと点群との衝突検出実行を改善するために、Mech-Viz 1.7.0バージョンでは STL 形式のロボットハンド衝突モデルに対応しなくなります。新バージョンのソフトウェアがサポートするモデル形式については、「リソースのモデル」をお読みください。

「ビジョン処理による移動」デパレタイジングのアルゴリズムの最適化

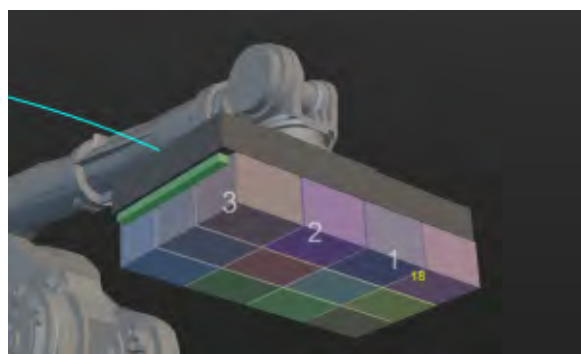
Mech-Viz 1.7.0 バージョンでは、「ビジョン処理による移動」のデパレタイジングのアルゴリズムを最適化し、以下の機能が可能になりました。

<p>単一ブロック吸盤</p>	
<p>複数ブロック吸盤</p>	
<p>単一把持</p>	

行ごと把持



複数行把持

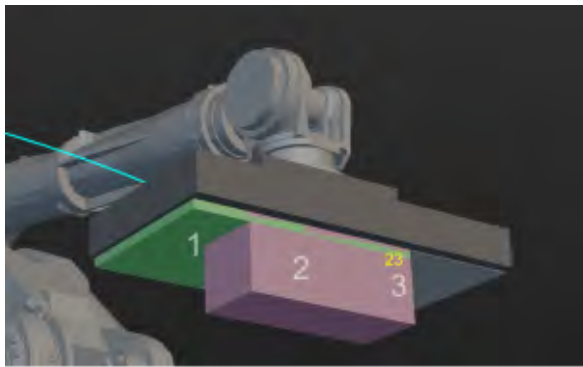


中心合わせ



辺の中心合わせ



<p>コーナー合わせ</p>	
<p>吸盤の長辺が箱の組み合わせ長辺と平行</p>	
<p>吸盤の長辺が箱の組み合わせ長辺と垂直</p>	

さらに、TCP 座標系の X 軸が吸盤の長辺に平行し、対象物把持点座標系の X 軸が箱の長辺に平行しなければならないという制限をなくしました。以上の機能は互いに干渉せず、パラメータを設定することで様々な把持戦略を作成することができます。

モデルエディタの最適化

- モデルエディタに「座標系設定」機能を追加しました。「座標系設定」機能により、OBJ 形式モデルがロボットフランジに正確に取り付けられるように基準モデルの座標系の原点と各軸の向きを設定することができます。
- モデルエディタのプロジェクトは .m3d ファイルとして保存され、フォルダを作成して保存する必要はありません。
- モデルエディタによって生成された OBJ 形式モデルは、Mech-Viz 1.6.0 以前のソフトウェアでも使用可能です。

オイラー角の表示形式の最適化

ブランドによって、ロボットのオイラー角はティーチングペンダントで異なる符号で表示されます。例えば ABB は EX、EY、EZ で、KUKA は A、B、C で、Kawasaki は O、A、T を使用します。

Mech-Viz 1.7.0バージョンでは、一部のブランドのロボットにオイラー角タイプの対応関係が表示され、ロボットのオイラー角の符号も確認できます。ユーザーがロボットを選択すると、ソフトウェアではプロジェクトに使用するオイラー角のタイプが自動的にロボットに対応するタイプに変換されます。

ロボットハンド設定の最適化

Mech-Viz 1.7.0バージョンでは、ロボットハンド設定の画面と機能を最適化しました。ロボットハンドの種類、制御ロジックは設定可能になり、対称性設定のパラメータも最適化しました。

対象物設定の最適化

Mech-Viz 1.7.0バージョンでは対象物設定ウィンドウで対称物に関する設定を統合しています。対象物対称性、把持範囲、解選択戦略などの設定を実行できます。

シーンの物体設定の最適化

Mech-Viz 1.7.0バージョンではシーンの物体の作成と設定を「シーンの物体設定」ウィンドウに統合しました。また箱に対し、位置姿勢の有効範囲設定パラメータを最適化しました。

ステップ名の変更

Mech-Viz 1.7.0以降のバージョンでは、「タスク」や「タスクの組み合わせ」はそれぞれ「ステップ」と「ステップの組み合わせ」に変更されます。また、以下のステップの名前を変更しました。

変更前	変更後
タスクの組み合わせ	ステップの組み合わせ
メッセージによって異なるブランチを実行	メッセージによって異なる分岐を実行
道標によって異なるブランチを実行	道標によって異なる分岐を実行
タスクの組み合わせを終了	ステップの組合せの出口
把持対象物を更新	把持済み対象物を更新
視覚処理による認識	ビジョン処理による認識
視覚処理による移動	ビジョン処理による移動
視覚結果をチェック	ビジョン結果をチェック
視覚結果が使用済み	ビジョン結果が使い切れ
ロボットを呼び出す	ロボットの関数を呼び出す
ツールをチェック	ロボットハンドをチェック
ツールを設定	ロボットハンドを設定
視覚処理による継続パレタイジング	ビジョン処理による継続パレタイジング

その他

メニューバー

- ツールバーの「ロボットに送信するメッセージを出力しない」ボタンをなくしました。ロボットに送信するメッセージを常に出力するようにしました。
- ツールバーの「デバッグファイル (.dmp) を生成」ボタンをなくしました。新バージョンのソフトウェアではこの機能は使用できません。
- 設定→オプションでは「長さの単位」と「角度の単位」の初期値はそれぞれ「mm」と「°」に変更しました。

リソース

- ツールバーの「自動的に読み込む」ボタンをなくし、新バージョンのソフトウェアでは**リソース**→**プロジェクト**→**ドロップダウンメニュー**→**自動的に読み込む**をクリックするとこの機能を実現します。
- 吸盤の設定とロボットハンドの設定を「ロボットハンド設定」ウィンドウに統合しました。
- ロボットハンド設定、対象物設定、シーンの物体設定ウィンドウを開いたまま3Dシミュレー

ションエリアで視角調整や拡大・縮小の操作を実行できます。

ステップ

- 移動ステップのクイック設定、位置姿勢変換、位置姿勢校正、位置姿勢編集ボタンは再配置されました。
- 「DOリストを設定」のポート番号の上限を30000に引き上げました。
- 「ビジョン結果をチェック」の「未完成」出口を「待ち時間タイムアウト」に変更しました。
- 「ビジョン処理による移動」の箱の複数把持機能のログを最適化しました。箱の組み合わせ、オフセットの計画履歴が記録されます。
- 「シーンの物体を更新」に「失敗」出口を追加しました。
- 「ビジョン処理による移動」ステップを実行して複数の箱を把持する場合、組み合わせられた箱は一つの大きな箱でなく、元のまま複数の箱として扱います。
- 「相対移動」ステップの「移動基準」のオプションに、「選択」を削除しました。
- 「RobotIQ設定」と「ロボット関数を呼び出す」ステップを削除しました。
- 「ビジョン処理による移動」の「実際に把持を実行した位置姿勢を保存」パラメータと関連機能をなくしました。

ロボットパネル

- ロボット関節角度のスライダーにより動的リミットが表示されるようになりました。
- Yaskawa PL80、MPL80II、MPL100IIロボットの関節角度表示を最適化しました。
- ロボットを切り替えてからロボットハンドリストがクリアされません。
- プロジェクトに必要なロボットが見つからない場合、ロボットライブラリからロボットを再選択することが可能になりました。

衝突検出

- 点群衝突検出の実行にかかわらず、「ビジョン処理による移動」により把持対象物の点群を除去できるようになりました。
- 点群衝突検出の実行にかかわらず、「ビジョン処理による認識」をリセットすると点群もリセットできるようになりました。

通信

- 「その他」パネルに、「隣接する経路点をスキップ」パラメータを追加し、これを設定することで隣接する経路点の二番目の経路点を送信するかを指定することが可能になりました。
- 「その他」パネルの「TCP位置姿勢を送信」ボタンをなくしました。新バージョンのソフトウェアでは常に送信するようになりました。

問題修復

Mech-Viz 1.7.0バージョンでは、以下の問題を修復しました。

- 過去バージョンのプロジェクトを開けない時に一部のワークフローがインポートされる問題を修復しました。
- プロジェクトをロックした状態でパスワード無しで新規ユーザー登録できる問題を修復しました。
- プロジェクトを閉じるときに未保存の変更があるとのメッセージが表示されない問題を修復しました。
- シミュレート/実行中に座標軸をドラッグするとシーンの物体の位置が変更できる問題を修復しました。
- シミュレート/実行中にロボット TCP 座標軸をドラッグすると仮想ロボットが移動できる問題を修復しました。
- ロボットインストールパッケージファイルの大文字が原因でインストールが失敗する問題を修正しました。
- ロボットハンドを作成する時に、3DシミュレーションエリアにおけるTCP座標系は設定通りに変化しない問題を修復しました。
- ワークフロー自動レイアウトしたあと、ステップのアイコンが重なる問題を修復しました。
- ワークフローで検索機能で検索するステップが変化する問題を修復しました。
- ビジョンサービスが「ビジョン処理による認識」にトリガーされていない時、「ビジョン処理による移動」のエラーメッセージが間違える問題を修復しました。
- 「ビジョン処理による移動」がビジョン結果を再使用して経路計画が失敗した時、エラーメッセージが不明確な問題を修復しました。
- 「ビジョン処理による移動」の把持数が設定した値に達していない時に表示されるメッセージが間違える問題を修復しました。
- 「把持済み対象物を更新」がビジョン結果を使用する時に対象物の寸法が使用される問題を修復しました。
- 外部モデルを使用した対象物を把持する時に、対象物モデルが把持されない問題を修復しました。

過去バージョンの更新説明

[Mech-Viz 1.6.2バージョンの更新説明](#)

[Mech-Viz 1.6.1バージョンの更新説明](#)

[Mech-Viz 1.6.0バージョンの更新説明](#)

2.4. Mech-Centerの更新説明

以下では、Mech-Center 1.7バージョンの新機能や機能最適化、問題修復について説明します。

Mech-Center 1.7.4バージョンの更新説明

以下では、Mech-Center 1.7.4バージョンの問題修正について説明します。

問題修復

1.7.4バージョンでは、210コマンドで返される位置姿勢データからロボット架台の高さが削除されない問題を修正しました。

Mech-Center 1.7.2バージョンの更新説明

以下では、Mech-Center 1.7.2バージョンの新機能や機能最適化、問題修復について説明します。

新機能

標準インターフェースは三菱MCプロトコルに対応

標準インターフェースでは、ビジョンシステムをクライアント、三菱PLC機器をサーバーとして、三菱PLCとのMC通信に対応しています。PLCのIPアドレスとポート、通信フレーム、Dレジスタのベースアドレスは、Mech-Visionの「ロボット通信設定」で設定できます。

標準インターフェースはUDPプロトコルに対応

標準インターフェースはUDP通信をサポートしており、ビジョンシステムはサーバーとして動作し、TCPプロトコルと同様の方法とコマンドを使用します。適応可能なロボットの中で、ビジョンシステムとの通信にUDPを使用しているのは、Hyundaiロボットだけです。

標準インターフェースはROKAE（産業用）、Hyundai（産業用）、Nachi（産業用）ロボットに対応

標準インターフェースはROKAE（産業用）、Hyundai（産業用）、Nachi（産業用）ロボットに対応しています。1.7.2バージョンでは、新たにロボット標準インターフェースプログラムを追加しました。

URCap 1.5.0はPolyscope 6システムに対応

Mech-Center 1.7.2バージョンでは、URCap 1.5.0プラグイン（1.5.0.urcapおよび1.5.0.urcapx）を追加しました。



UR Polyscope 6システムでは、URCapプラグインの拡張子はurcapxです。Polyscope 6システムは、過去の.urcapファイルとの互換性がなくなりました。

URCap 1.5.0プラグインに対応可能なPolyscopeバージョンとMech-Mind統合パッケージのバー

ジョンは、下表の通りです。お使いのロボットのPolyscopeバージョンに応じて、対応するURCapプラグインファイルを選択してください。

URCapプラグインファイル	Polyscopeのバージョン要件	Mech-Mind統合パッケージのバージョン要件
1.5.0.urcap	Eシリーズ：5.9以降（6.0以下） CBシリーズ：3.14以降	1.7.2
1.6.2	Eシリーズ：6.0以降	1.7.2
1.7.0	Eシリーズ：5.3以降（6.0以下） CBシリーズ：3.9以降	1.6.1以降

URCap 1.4.6プラグインと比較して、URCap 1.5.0プラグインでは、以下の機能が追加されています。

- Mech-Visionの「経路計画」ステップを使用して、計画された経路を取得します。
- プロジェクト番号、レシピ番号、分岐番号、分岐出口番号の入力に変数を使用することをサポートします。
- Mech-Visionから複数の出力を受信することをサポートします：Basic（ビジョンポイントとラベルを受信）、Custom（ビジョンポイント、ラベル、カスタマイズされたデータを受信）、Planned path（「経路計画」ステップから出力された経路点、ラベルを受信）。

AB PLCとのEtherNet/IP通信に対応

標準インターフェースでは、ビジョンシステムをスレーブ、Allen-Bradley PLC（AB PLC）をマスターとして、AB PLCとのEtherNet/IP通信をサポートしています。EtherNet/IP通信は、Mech-Visionの「ロボット通信設定」で設定できます。

機能最適化

撮影完了時に101コマンドを返すことに対応

過去のEye In Handシーンでは、カメラ撮影後にMech-Visionプロジェクトの実行終了を待たずにロボットを動かすには、Mech-Visionプロジェクトの「カメラから画像を取得」ステップの後に「通知」ステップを追加する必要がありました。

今回の新バージョンでは、ユーザーが「通知」ステップを追加する必要はなく、Mech-Visionの「ロボット通信設定」の「詳細設定」をクリックし、**撮影完了後、「1102：トリガー成功」を返す**にチェックを入れる必要があります。ロボットがMech-Visionプロジェクトをトリガーして「1102：トリガー成功」を受信したら、動作ができるので、タクトタイムの向上を実現しています。

今回の新バージョンでは、PROFINETプロトコルの標準インターフェースを使用する場合、以下のいずれかの方法で、撮影完了後にロボットを動作することが可能です。



- 「カメラから画像を取得」ステップの後に、Trigger Acknowledge信号の値を判定条件とする「通知」ステップを追加します。
- 1102のステータスコードを判定条件として、**撮影完了後、「1102：トリガー成功」を返す**にチェックを入れます。

問題修復

Mech-Center 1.7.2バージョンでは、以下の問題を修正しました。

- 標準インターフェースを使用して自動キャリブレーションを行う場合、Mech-Visionのキャリブレーションパラメータを設定すると通信タイムアウトする問題を修正しました。
- Mech-Vizプロジェクトコマンドをトリガーした直後にMech-Viz分岐を設定すると、失敗する問題を修正しました。

Mech-Center 1.7.1バージョンの更新説明

以下では、Mech-Center 1.7.1バージョンでの問題修正について説明します。

問題修復

Mech-Center 1.7.1バージョンでは、以下の問題を修正しました。

- コマンド210により、ワークの向きに関する情報が正しく出力されない問題を修正しました。
- 特殊な状況下で、新規IPCで「Mech-Center -> 設定 -> Mech-Interface -> ロボット一覧」を選択できない問題を修正しました。
- 「Adapterジェネレーター -> ロボットの設定 Eye in Hand -> ロボット名」パラメータのパスが正しくないことを修正しました。
- Mech-Centerが誤ったロボット型番を取得することを修正しました。修正後は、Mech-Visionで選択したロボット型番とMech-Vizで選択したロボット型番が一致しない場合に、Mech-Visionからインターフェースサービスを起動すると、Mech-CenterはMech-Visionで選択したロボット型番を取得します。
- キャリブレーション中にMech-Centerを終了できない問題を修正しました。
- Siemens PLCとの接続に失敗したときに、Mech-Centerがインターフェースサービスを直接終了する問題を修正しました。修正後、Mech-Centerはインターフェースサービスを有効にしたまま、自動的に再接続します。
- STEPロボットのオイラー角パラメータが欠落している問題を修正しました。

Mech-Center 1.7.0バージョンの更新説明

以下では、Mech-Center 1.7.0バージョンの新機能や機能最適化、問題修復について説明しま

す。

新機能

標準インターフェースとAdapter通信のMech-Centerでの設定が不要

Mech-Vision 1.7.0バージョンでは、従来のMech-Centerのインターフェース通信サービスの機能を統合しました。Mech-Centerソフトウェアを使用することなく、標準インターフェースとAdapter通信設定をMech-Visionで直接行うことができます。



- 標準インターフェース通信方式を使用するソリューションの場合、標準インターフェースプログラムの再読み込みが必要です。
- Vizティーチング通信の設定は、Mech-Centerで行う必要があります。Mech-Centerを起動し、Mech-Visionを起動し、過去バージョンと同様に通信設定を行う必要があります。

コマンド105を追加

1.7.0バージョンでは、標準インターフェースの通信プロトコル

(TCP/IP、PROFINET、EtherNet/IP、Modbus TCP、Siemens PLC Snap 7) にコマンド105を追加しました。このコマンドは、Mech-Visionの新しいステップ「経路計画」から出力されたロボット動作経路を受信するために使用されます。

標準インターフェースを使用した把持サンプルプログラムを追加

1.7.0バージョンでは、ABB、FANUC、YASKAWA、KAWASAKI、KUKA、ELITE、JAKAのロボットの標準インターフェースのサンプルプログラム「Mech-Visionの「経路計画」ステップの結果を取得」を提供します。

Mech-Visionが計画した経路を必要とするソリューションを使用する場合、標準インターフェースプログラムとサンプルプログラムを再読み込む必要があります。

JAKAとELITEロボットの標準インターフェースプログラムを追加

1.7.0バージョンでは、JAKAとELITEロボットの標準インターフェースプログラムを追加しました。

左手系を使用するロボットに対応

1.7.0バージョンでは、左手系を使用するロボットに対応しています。

Siemens PLCがコマンド110に対応

1.7.0バージョンでは、Siemens PLCにコマンド110を追加しました。このコマンドは、Siemens PLCがMech-Visionのカスタマイズされたポートからのデータを受信するために使用されます。

機能最適化

コマンド101と201について

1.7.0バージョンでは、コマンド101と201の **ロボット位置姿勢とタイプ** パラメータを変更し、ロボット側でカスタマイズした関節角度を送信できるようになりました。カメラの取り付け方式がETHの場合、ロボットはこのカスタマイズした関節角度から経路の計画を開始します。

ESTUNとAUBOロボットのVizとの通信プログラムに再適用

1.7.0バージョンでは、ESTUNとAUBOロボットのVizとの通信プログラムに再適用しました。

Adapterプロジェクトについて

1.7.0バージョンでは、Adapterプロジェクトのソースパスを直接読み込むことができます。Adapterプロジェクトファイルを変更した後は、Mech-Centerソフトウェアを再起動する必要はなく、インターフェースサービスを再起動するだけでAdapterプロジェクトの読み込みを有効にすることができます。

問題修復

EPSON SCARA伸縮アームの計算不具合を修正

1.7.0バージョンでは、EPSON SCARAロボットの伸縮アームの計算不具合を修正しました。

FANUC SCARAロボットのベース座標系の不具合を修正

1.7.0バージョンでは、FANUC SCARAロボットのベース座標系の不具合を修正しました。

過去バージョンの更新説明

[Mech-Center 1.6.1バージョンの更新説明](#)

[Mech-Center 1.6.0バージョンの更新説明](#)

2.5. プロジェクトを1.7.0バージョンへアップグレードする方法

Mech-Vision 1.7.0バージョンでは **ソリューション** 機能が追加され、新しいソリューションファイル構造が生成されたため、過去バージョンのMech-Visionプロジェクトは使用する前に変換を行う必要があります。

1.7.0バージョンにアップグレード後、本節を参照してプロジェクトの変換を完了してください。

アップグレード概要

プロジェクト	変換方法	アップグレード後の操作
標準インターフェースまたはAdapter通信を使用し、Mech-Visionプロジェクトのみで、ビジョンポイントを取得	Mech-Visionプロジェクトをソリューションに変換し、Mech-Visionでロボット通信を再設定します。 Mech-Visionを単独で起動する必要があります。	Mech-Visionプロジェクトをソリューションに変換
		ロボット通信設定
		PC起動時のソフトウェア自動起動を変更
Mech-Vizを使用して簡単な経路計画の実行と取得	Mech-Visionソリューションが要件を満たすことができるかどうかを評価するために、Mech-Mind株式会社にお問い合わせください。 要件を満たすことができれば、「経路計画」ステップを使用して、衝突のない計画経路を出力するソリューションを構築します。	Mech-Visionプロジェクトをソリューションに変換
		ロボット通信設定
		PC起動時のソフトウェア自動起動を変更
		ステップ「経路計画」をMech-Visionプロジェクトに追加
		ロボットのインターフェースプログラムの読み込み
Vizティーチングを使用 Mech-Vizを使用して複雑なタスク（混載パレタイジング、複数把持・配置、複雑な経路生成など）を実行	従来のソフトウェア使用方法（Mech-Center + Mech-Vision + Mech-Viz）や操作方法と変わりません。 Mech-Centerを起動し、それを通じてMech-Vision、Mech-Vizを起動する必要があります。	ロボットプログラムの変更
		Mech-Visionプロジェクトをソリューションに変換

アップグレード後の操作

Mech-Visionプロジェクトをソリューションに変換

割り当てられていないMech-Visionプロジェクトを番号でソリューションに割り当てます。



プロジェクトがソリューションに割り当てられていない場合、カメラキャリブレーション機能は使用できません。

1	<p>Mech-Visionでプロジェクトを元のプロジェクト番号順に開きます。</p> <p>この状態では、プロジェクトは 未割り当てプロジェクト にあります。</p>	
2	<p>ファイル > ソリューションを新規作成 をクリックします。</p> <p>ファイル > ソリューションを保存 をクリックし、ソリューションを保存するためのパスを選択します。</p>	
3	<p>ソリューション名を右クリックし、[ソリューションを自動的に読み込む]にチェックを入れます。</p>	
4	<p>プロジェクト1を右クリックし、[プロジェクトを削除]をクリックします。</p>	

5	<p>未割り当てのプロジェクトを右クリックし、[現在のソリューションに移動]をクリックします。</p> <p>割り当てられていないすべてのプロジェクトに対して、この操作を順次に行います。</p>	
6	<p>プロジェクト番号が元のプロジェクト番号と一致していることを確認します。</p> <p>ファイル > ソリューションを新規作成 をクリックします。</p>	

ロボット通信設定

Mech-Visionでロボット通信設定を行います。以下の操作を始める前に、Mech-Centerが起動していないことと、Mech-Visionが単独で起動していることを確認してください。

1. Mech-Visionのツールバーで[**ロボット通信設定**]をクリックします。
2. **ロボットを選択**のドロップダウンリストで、「適応可能なロボット」を選択します。
3. [**ロボット型番を選択**]をクリックし、ロボットライブラリから現在のプロジェクトに使用したロボットを選択します。
4. [**次へ**]をクリックし、通信設定の画面に入ります。
5. 現在のプロジェクトの通信設定に応じて、インターフェースタイプ、通信プロトコル、IPアドレスなどを設定します。
6. [**適用**]をクリックすると、インターフェースサービスが起動されます。また、対応するログがログバーに出力され、ツールバーの「インターフェースサービス」がオンになります。

PC起動時のソフトウェア自動起動を変更

Mech-Centerで「PC起動時にMech-Centerを自動的に実行」のチェックを外し、Mech-Visionで「PC起動時にMech-Visionを起動」にチェックを入れます。

1. Mech-Centerを起動し、**設定** > **外観と動作** をクリックします。
2. [**PC起動時にMech-Centerを自動的に実行**]のチェックを外します。
3. [**保存**]をクリックし、Mech-Centerを終了します。

「PC起動時にMech-Visionを起動」を有効にします。

1. Mech-Visionを起動し、**設定**、**オプション**、**詳細設定** をクリックします。
2. [PC起動時にMech-Visionを起動] にチェックを入れ、[OK] をクリックします。

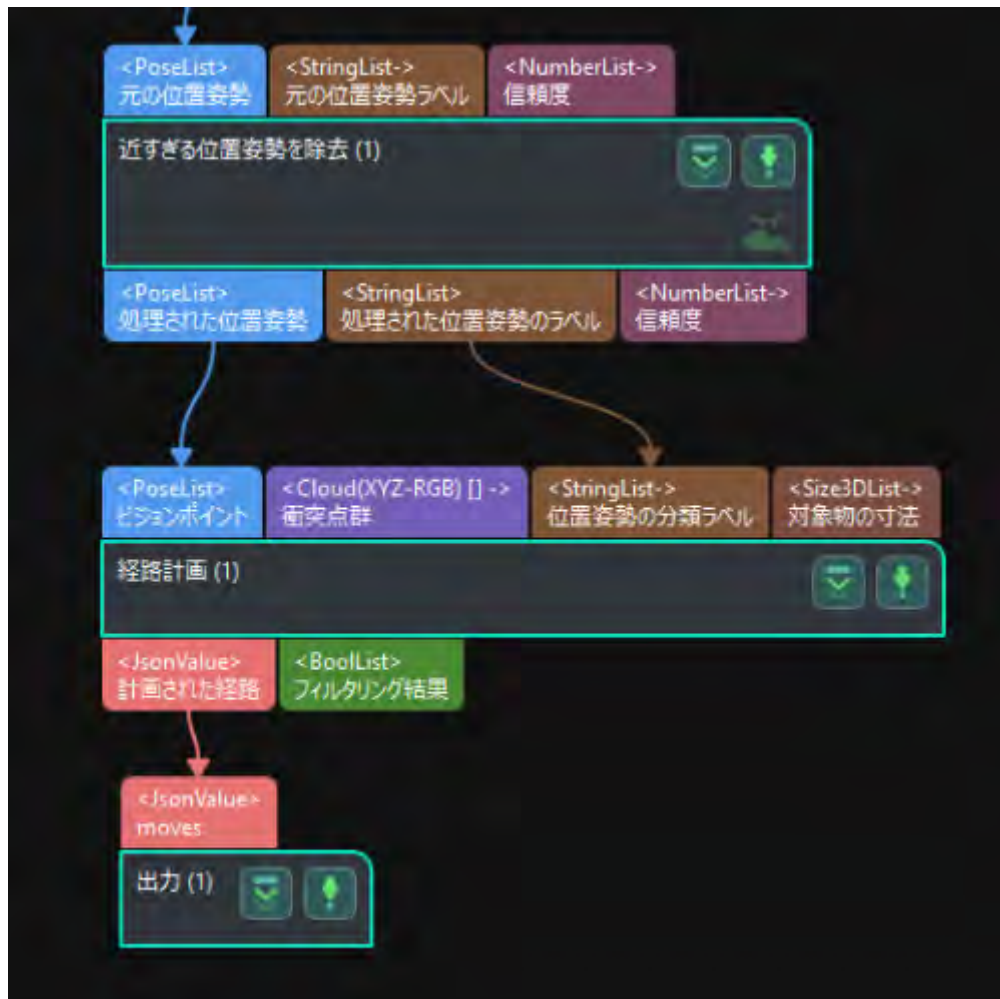
ステップ「経路計画」をMech-Visionプロジェクトに追加

Mech-Visionプロジェクトにステップ「経路計画」を追加し、ロボットの動作経路を計画します。

下図のようなプロジェクトを例として説明します。



1. ステップライブラリから「経路計画」を検索し、それをプロジェクト編集エリアにドラッグします。
2. 「出力」のステップパラメータで「ポートタイプ」を「事前定義済み（ロボット経路）」に設定します。
3. 下図を参照してポートをつなぎます。



4. 「経路計画」ステップパラメータで [経路計画設定ツールを開く] をクリックします。
5. 「経路計画設定ツール」を参照して、ロボットの動作経路計画を完了します。

ロボットのインターフェースプログラムの読み込み

ロボットのインターフェースプログラムを読み込みます。

1. ロボット通信設定で対応するロボットブランドを選択した後、通信設定画面で [プログラムフォルダを開く] をクリックします。表示されるフォルダには、標準インターフェースのプログラムファイルと把持のサンプルプログラムが格納されています。
2. [標準インターフェース通信](#) を参照してプログラムの読み込みを行います。

ロボットプログラムの変更

ロボット側のプログラムを修正し、把持経路を取得するためにMech-VizプロジェクトのトリガーからMech-Visionプロジェクトのトリガーに変更します。

- call mm_start_vizをcall mm_start_visに置き換えます。
- call mm_get_vizdataをcall MM_Get_VisPathに置き換えます。

3. ソフトウェアのインストールガイド

本節では、Mech-Mindビジョン関連ソフトウェアのダウンロード、インストール、アップグレード、アンインストール、メンテナンスと変更の操作について説明します。

Mech-Mindビジョン関連ソフトウェアは、Mech-Mind株式会社が独自研究開発したMech-Vision、Mech-Viz、Mech-Centerのソフトウェアの総称です。Mech-Mindビジョン関連ソフトウェアは、使いやすいセットアップウィザードを提供し、標準化されたインストール、アップグレードおよびアンインストールの機能とプロセスを実現します。また、中国語、英語、日本語、韓国語に対応し、マルチ解像度の拡大縮小率にも対応しています。

Mech-Mindビジョン関連ソフトウェアを初めてご利用になる場合、[ソフトウェアをインストール](#)を参照してソフトウェアをインストールしてください。

Mech-Mindビジョン関連ソフトウェアをインストールした場合は、[ソフトウェアをアップグレード](#)を参照してソフトウェアを最新バージョンにアップグレードすることができます。

システム要件

Mech-Mindビジョン関連ソフトウェアを使用するには、まずIPCまたはPCにインストールする必要があります。ソフトウェアがインストールされているIPCまたはPCは、下記の最小システム要件を満たす必要があります。

- **オペレーティングシステム (OS)** : Windows 10以降
- **CPU** : AVX2命令セットをサポートし、以下のいずれかの条件を満たす必要があります：
 - ディスクリットGPU非搭載の場合：Intel i5-12400以上。
 - ディスクリットGPU搭載の場合：Intel i7-6700以上、GTX 1050 Ti以上（GPU）。
- **メモリ** : 8GB以上
- **GPU** : NVIDIA GTX 1050Ti以上（ディスクリットGPU搭載の場合）
- **ハードドライブ** : 128GB SSD以上



- Mech-Mindビジョン関連ソフトウェアは、他のブランドのディスクリットGPUをサポートしていません。
- ソフトウェアがIPCまたはPCにインストールされているディスクに10GB以上の空き容量があることを確認してください。そうでなければ、インストールが失敗する可能性があります。
- ドングルドライバーがIPCまたはPCにインストールされていない場合、セットアップウィザードは、ソフトウェアのインストールプロセスに dongle ドライバーを C ドライブに自動的にインストールします。ライセンス認証が失敗し、Mech-Mindソフトウェアが使用できなくなるので、ドライバーをアンインストールしたり、インストールファイルを移動したりしないでください。

ソフトウェアのインストールパッケージをダウンロード

Mech-Mindは、ビジョンシステムに必要なソフトウェアを簡単、かつすばやくインストールするために、すべてのビジョン関連ソフトウェアを統合するパッケージ（Mech-Mind Software Suite、以下は「統合パッケージ」）を提供しています。また、Mech-Mindは各ビジョン関連ソフトウェアの単体のインストールパッケージ（以下は「単体パッケージ」）も提供しています。



- 統合パッケージには、Mech-Vision、Mech-Viz、Mech-Centerの3つの製品が含まれています。
- 統合パッケージと単体パッケージの両方にドングルドライバーが統合されています。ドングルドライバーは、ドングルを適合させ、ソフトウェアのライセンス認証を完了するために使用されます。
- 統合パッケージとMech-Centerソフトウェアの単体パッケージはPython 3.6.5環境を統合し、環境チェックツールは含まれていません。
- Mech-Visionソフトウェアの単体パッケージはPython 3.6.8環境を統合しています。

統合パッケージまたは単体パッケージの取得は、[Mech-Mindオンラインコミュニティ](#)からダウンロードするか、Mech-Mindのプリセールスエンジニアまたは営業担当者にお問い合わせください。

ソフトウェアをインストール

インストールパッケージの整合性を検査

ソフトウェアのインストールパッケージは転送中またはダウンロード中に破損する可能性があるため、ソフトウェアをインストールする前に、その整合性を検査する必要があります。インストールパッケージの整合性は、ダウンロードページに記載されているCRC32値で確認することができます。

インストールパッケージの整合性を検査するには、下記の手順に従って操作してください。

1. ダウンロードしたソフトウェアのインストールパッケージを、IPCまたはPCの指定したディレクトリ（D:¥ など）にコピーします。
2. ソフトウェアインストールパッケージを右クリックし、**CRC SHA** ▶ **CRC32** を選択して、ソフトウェアインストールパッケージのCRC32値を計算します。



ソフトウェアインストールパッケージが圧縮パッケージの場合、圧縮パッケージ全体、解凍されたインストールファイルに対してCRC32値を計算し、Contentフォルダに対してデータのCRC32値を計算する必要があります。

3. 計算されたCRC32値が、ダウンロードページに記載されているCRC32値と同じであることを確認します。



ソフトウェアインストールパッケージのCRC32値が異なる場合は、ソフトウェアイン

ストールパッケージを再度ダウンロードしてください。解凍したインストールファイルまたはContentフォルダのCRC32値が異なる場合は、ソフトウェアインストールパッケージを再度解凍してください。

複数のビジョン関連ソフトウェアをすばやくインストール

統合パッケージを使用して複数のビジョン関連ソフトウェアをすばやくインストールするには、下記の手順に従って操作してください。

1. ダウンロードした統合パッケージを、IPCまたはPCの指定したディレクトリ（**D:/Mech-Mind** など）にコピーし、解凍します。



解凍後、ソフトウェアのインストールファイル（Mech-Mind Software Suite Installer x.x.x.exe）および「content」フォルダを移動しないでください。

2. ソフトウェアのインストールファイルをダブルクリックして **Mech-Mind Software Suite セットアップウィザード** を起動します。すると、**ようこそ** 画面が表示されます。
3. **ようこそ** 画面で製品の紹介を閲覧し、**[次へ]** をクリックします。
4. **使用許諾契約** 画面で使用許諾契約を注意深く読み、**[使用許諾契約の条項に同意します]** にチェックを入れて、**[次へ]** をクリックします。
5. **製品の選択** 画面でインストールする製品を選択し、必要に応じて**[デスクトップにショートカットを作成]** にチェックを入れてから、**[次へ]** をクリックします。
6. **パスの設定** 画面でインストールパスを設定します。インストールパス（D:\%projectなど）を変更したい場合は、「変更」をクリックしてパスを選択し、**[次へ]** をクリックします。

デフォルトでソフトウェアが以下のパスにインストールされます。



- Mech-Vision : **C:/Mech-Mind/Mech-Mind Software Suite-x.x.x/Mech-Vision**
- Mech-Viz : **C:/Mech-Mind/Mech-Mind Software Suite-x.x.x/Mech-Viz**
- Mech-Center : **C:/Mech-Mind/Mech-Mind Software Suite-x.x.x/Mech-Center**

7. **インストールする前の確認** 画面で、インストールする製品を確認したら **[インストール]** をクリックします。
8. **インストール** 画面でインストールが完了するまで待ちます。
9. インストールが完了したら、**[完了]** 画面で **[完了]** をクリックします。

特定のビジョンソフトウェアをインストール

特定のビジョンソフトウェアをインストールするには、下記の手順に従って操作してください。

1. ダウンロードした単体パッケージを、IPCまたはPCの指定したディレクトリ（**D:/Mech-Mind** など）にコピーし、（必要があれば）解凍します。



解凍後、ソフトウェアのインストールファイル（Xxxx-xxx Installer x.x.x.exe）およ

び「content」フォルダを移動しないでください。

2. ソフトウェアのインストールファイルをダブルクリックして **Xxxx-xxx セットアップウィザード** を起動します。すると、**ようこそ** 画面が表示されます。
3. **ようこそ** 画面で製品の紹介を閲覧し、**[次へ]** をクリックします。
4. **使用許諾契約** 画面で使用許諾契約を注意深く読み、**[使用許諾契約の条項に同意します]** にチェックを入れて、**次へ** をクリックします。
5. **製品の選択** 画面でインストールする製品を選択し、必要に応じて **[デスクトップにショートカットを作成]** にチェックを入れてから、**[次へ]** をクリックします。
6. **パスの設定** 画面でインストールパスを設定します。インストールパス (D:\¥projectなど) を変更したい場合は、「変更」をクリックしてパスを選択し、**[次へ]** をクリックします。

デフォルトでソフトウェアが以下のパスにインストールされます。



- Mech-Vision : **C:/Mech-Mind/Mech-Vision-x.x.x**
- Mech-Viz : **C:/Mech-Mind/Mech-Viz-x.x.x**
- Mech-Center : **C:/Mech-Mind/Mech-Center-x.x.x**

7. **インストールする前の確認** 画面で、インストールする製品を確認したら **[インストール]** をクリックします。
8. **インストール** 画面でインストールが完了するまで待ちます。
9. インストールが完了したら、**完了** 画面で **[完了]** をクリックします。



- バージョン1.6.0以降、Mech-Vizソフトウェアは主要メーカーのロボットインストールパッケージのみを統合します。
- 他のブランドまたは型番のロボットインストールパッケージが必要な場合は、[オンラインロボットライブラリ](#) からダウンロードしてインポートしてください。

ソフトウェアをアップグレード



- この機能は、1.6.0バージョン以降のソフトウェアでのみサポートされています。
- 1.6.0以前のバージョンから1.6.0以降への直接アップグレードをサポートしていません。以前バージョンのソフトウェアがインストールされていた場合、1.6.0以降にアップグレードする前に、以前バージョンのソフトウェアをアンインストールしてから、[新しいバージョンのソフトウェアをインストールする](#) 必要があります。アップグレードする前に、[アップグレードに関する注意事項](#) をよくお読みください。
- ソフトウェアが統合パッケージを使用してインストールされていた場合、新しいバージョンの統合パッケージを取得してから、アップグレードしてください。ソフトウェアが単体パッケージを使用してインストールされていた場合、新しいバージョンの単体パッケージを取得してから、アップグレードしてください。

ソフトウェアをアップグレードするには、下記の手順に従って操作してください。

1. ダウンロードした新しいバージョンの統合パッケージまたは単体パッケージを、IPCまた

はPCの指定したディレクトリ（D:/Mech-Mindなど）にコピーし、（必要があれば）解凍します。

2. ソフトウェアのインストールファイルをダブルクリックして **Xxxx-xxx セットアップウィザード** を起動します。すると、**アップグレード** 画面が表示されます。
3. **アップグレード** 画面で **[アップグレードして過去バージョンを保持]**（統合パッケージまだ対応しません）または **[アップグレードして過去バージョンを削除]** を選択します。
4. アップグレードが完了するまで待ちます。

ソフトウェアをアンインストール

特定のビジョンソフトウェアを使用する必要がなくなった場合は、以下の2つの方法でアンインストールできます。

- セットアップウィザードを使用してアンインストール（推奨）
- コントロールパネルを使用してアンインストール

セットアップウィザードを使用してアンインストール



ソフトウェアが統合パッケージを使用してインストールされていた場合、統合パッケージを使用してアンインストールしてください。ソフトウェアが単体パッケージを使用してインストールされていた場合、単体パッケージを使用してアンインストールしてください。

セットアップウィザードを使用してソフトウェアをアンインストールには、下記の手順に従って操作してください。

1. ソフトウェアのインストールファイルをダブルクリックして **Xxxx-xxx セットアップウィザード** を起動します。すると、**メンテナンス** 画面が表示されます。
2. **メンテナンス** 画面で **[アンインストール]** をクリックします。
3. **アンインストール** 画面で **[ユーザー構成ファイルを保持]** または **[ユーザー構成ファイルを放棄]** を選択します。
4. アンインストールが完了するまで待ちます。

コントロールパネルを使用してアンインストール

コントロールパネルを使用してソフトウェアをアンインストールするには、下記の手順に従って操作してください。

1. IPCまたはPCで **コントロールパネル** を開きます。
2. **プログラム > プログラムと機能** を選択します。
3. アンインストールするソフトウェアを右クリックして、**[アンインストール]** を選択します。



ソフトウェアが統合パッケージを使用してインストールされていた場合、インストー

ルされたすべてのソフトウェアは、プログラム一覧で「Mech-Mind Software Suite x.x.x」として表示されます。

4. アンインストールが完了するまで待ちます。

ソフトウェアを修復

Mech-Mindビジョン関連ソフトウェアに異常が発生して正常に使用できない場合は、修復機能を使用してソフトウェアを再インストールしてください。



- この機能は、1.6.0バージョン以降のソフトウェアでのみサポートされています。
- ソフトウェアが統合パッケージを使用してインストールされていた場合、統合パッケージを使用して修復してください。ソフトウェアが単体パッケージを使用してインストールされていた場合、単体パッケージを使用して修復してください。

ソフトウェアを修復するには、下記の手順に従って操作してください。

1. ソフトウェアのインストールファイルをダブルクリックして **Xxxx-xxx セットアップウィザード** を起動します。すると、**メンテナンス** 画面が表示されます。
2. **メンテナンス** 画面で **[修復]** をクリックします。
3. ソフトウェアの修復が完了するまで待ちます。

インストールするソフトウェアを変更



この機能は、1.6.0バージョン以降の統合パッケージでのみサポートされています。

インストールされた後、インストールするソフトウェアを変更するには、下記の手順に従って操作してください。

1. ソフトウェアのインストールファイルをダブルクリックして **Mech-Mind Software Suite セットアップウィザード** を起動します。すると、**メンテナンス** 画面が表示されます。
2. **メンテナンス** 画面で **[変更]** をクリックします。
3. **製品の選択** 画面で新しくインストールする必要がある製品またはコンポーネントにチェックを入れるか、アンインストールする製品またはコンポーネントのチェックを外します。
4. その後のインストールが完了するまで待ちます。

使用許諾契約

Mech-Mindビジョン関連ソフトウェアの使用許諾契約については、[エンドユーザーライセンス契約](#) をご参照ください。

インストールに関するよくある問題

インストールパッケージが正常に起動できない

問題のディスクリプション：

インストールパッケージを実行した後、それが正常に起動しないか、起動画面が一瞬消えます。

考えられる理由：

システムドライブの空き容量が不足しています。

問題解決の手順：

システムドライブの空き容量がインストールパッケージのサイズよりも大きいかどうかを確認します。

- 空き容量がインストールパッケージのサイズよりも少ない場合は、不要なファイルを削除してソフトウェアのインストールに十分な容量を増やしてから、再度インストールしてください。それでも解決しない場合は、Mech-Mind株式会社にお問い合わせください。
- 使用可能な空き容量がインストールパッケージのサイズよりも大きい場合は、Mech-Mind株式会社にお問い合わせください。

インストールに失敗

問題のディスクリプション：

ソフトウェアのインストール中に、「インストールに失敗しました」というエラーメッセージが表示されます。

考えられる理由：

- インストールパッケージが破損しているか、ファイルが見つかりません。
- ユーザーは管理者権限がありません。
- 他のアプリケーションのインストールが実行されているか、Windowsシステムが自動的に更新されています。
- その他の原因。

問題解決の手順：

1. インストールパッケージを再度取得し、[その整合性を検査](#)してから、インストールを再試行します。
 - 解決した場合、トラブル対処は終了します。

- それでも解決しない場合は、手順2に進みます。

2. C:\ProgramData\Package Cacheに格納されている{180DA015-BC2A-4262-928A-D312B20D9977}\v7.40.4997.501 フォルダを見つけて削除します。



このフォルダが見つからない場合は、リソースマネージャーの **表示オプション** で **隠しファイル** にチェックを入れてください。

- 解決した場合、トラブル対処は終了します。
- それでも解決しない場合は、手順3に進みます。

3. インストールパッケージを右クリックで「管理者として実行」を選択します。

- 解決した場合、トラブル対処は終了します。
- それでも解決しない場合は、手順4に進みます。

4. 他のアプリケーションのインストールまたはWindowsシステムの自動更新が完了するのを待ってから、インストールを再試行します。

- 解決した場合、トラブル対処は終了します。
- それでも解決しない場合は、手順5に進みます。

5. IPCまたはPCを再起動し、ソフトウェアを再度インストールします。

- 解決した場合、トラブル対処は終了します。
- 問題が解決されない場合は、**完了** 画面の「インストールログ」リンクをクリックします。インストールログを取得した後、それを当社のサポートチームに提供してください。

4. Mech-Vision

Mech-Visionのユーザーズマニュアルへようこそ！

Mech-Visionは、グラフィカルなインターフェイスにより、プログラミング不要でバラ積みピッキング / 高精度な位置決め / パレタイジング / デパレタイジングなどの実装ができる高度な **画像処理ソフトウェア** です。

主に以下の内容について説明します。

番号	内容	意味
1	Mech-Visionの概要	ソフトウェア機能やビジョン関連の専門用語・概念、ユーザーインターフェイス、Mech-Visionでソリューションを構築する一般的な手順
2	Mech-Visionの基本操作	Mech-Visionの基本操作と使用方法
3	[vision-steps:steps:::steps]	ステップの基本紹介と使用方法
4	[vision-calibration:calibration:::calib-guide]	ロボットハンド・アイ・キャリブレーションについての概念と実行手順
5	[vision-measure-mode:measure-mode:::measure-mode]	測定モードの概要と主要アルゴリズム
6	[vision-tools:tools:::tools-guide]	よく使うツールの基本紹介と使用方法

4.1. Mech-Visionの概要

4.1.1. 専門用語・概念

本節では、マシンビジョンに関する専門用語や概念について説明します。

ソリューション・プロジェクト

ソリューションライブラリ

様々な業界に適応可能なソリューションやプロジェクトに加え、サンプルデータも提供しています。

ソリューション

ソリューションとは、ビジョンソリューションを実現するために必要なロボットと通信、ビジョン処理、経路計画などの機能設定とデータの集りのことです。

プロジェクト

プロジェクトとは、Mech-Vision プロジェクトのことです。ソリューションを形成するには、1つまたは複数のプロジェクトが必要です。プロジェクトは単独で使用することはできず、ソリューションに割り当てて使用する必要があります。

ステップ

ステップはプロジェクト構築の基礎となります。ステップはアルゴリズム処理ユニットであり、異なるステップを組み合わせることで異なるアルゴリズム処理流れが形成されます。

ステップの組合せ

ステップの組合せは、複数ステップの集りです。異なるプロジェクトでは、アルゴリズム処理が一致していたり、類似していることが多く、これらの固定的なアルゴリズム処理ステップを組み合わせ、パッケージ化することで、簡単に再利用することが可能です。

パラメータレシピ

パラメータレシピとは、様々な状況でプロジェクトを実行するための調整が必要なパラメータ設定の集りです。実際の状況に応じてパラメータレシピを設定し、複数のプロジェクトを構築せずに異なるパラメータレシピを切り替えることで様々な適用シーンに対応可能です。それにより、プロジェクトの柔軟性を高め、生産効率を向上させます。

ロボットハンド・アイ・キャリブレーション

ロボットハンド・アイ・キャリブレーション

カメラ座標系とロボット座標系の対応関係を確立することを指します。これにより、ビジョンシステムによって確立された対象物の位置姿勢をロボット座標系での位置姿勢を変換し、ロボットの作業をガイドすることが可能です。

内部パラメータ

カメラ内部パラメータにのみ関係するものです。

外部パラメータ

カメラの座標系とワールド座標系との間の座標変換関係を指します。

オイラー角

3次元空間における物体の姿勢を記述する方法で、3次元空間における物体の回転を3つの角度パラメータ（ロール・ピッチ・ヨー）で表現するものです。

TCP

TCP（ツールセンターポイント）、ロボットハンドの先端にある点を指します。ワークの把持などの作業のためにロボットを空間上のある点に移動させる場合、実際はその点にツ

ルセンターポイントを移動させることです。対象物把持などを実行する場合、ロボットをある点に移動するのは、TCP をこの点に移動することです。

ビジョン処理の流れ

点群

データの表面特徴を含む点の集まりです。

位置姿勢

ロボットの把持をガイドする位置・方向情報です。

マスク

選択した画像、図形、または対象物を用いて、処理する画像 (すべてまたは一部) を遮り、画像処理の領域または処理流れを制御します。画像を遮るための画像または対象物はマスクと呼びます。

ROI

マシンビジョンや画像処理では、処理する画像から処理が必要な領域を枠で囲み、関心領域と呼びます。

4.1.2. ユーザーインターフェイス

本節では、Mech-Vision のようこそ画面とメイン画面について説明します。

ようこそ画面

自動的に読み込まれたソリューションやプロジェクトがない場合、Mech-Vision を起動すると、下図のようなようこそ画面が表示されます。



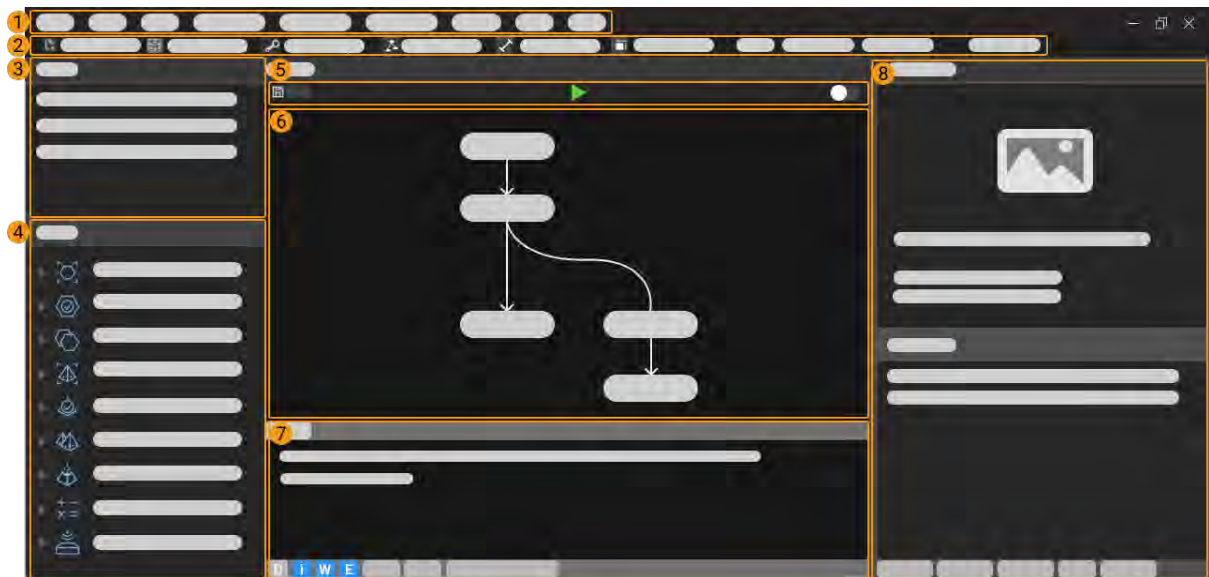
ようこそ画面で、マウスポをカードの右下隅の「？」に合わせると、その説明が表示されます。

Mech-Vision のようこそ画面は、以下の部分で構成されています。

番号	項目	説明
1	バージョン情報	バージョン情報、ソフトウェアアップグレードの注意事項などを確認できます。
2	クイックアクセス	ソリューションやプロジェクトを開く、ソリューションライブラリからソリューションやプロジェクトを新規作成する、空白ソリューションを新規作成するなどの操作を素早く行うことができます。
3	ユーザーマニュアル	該当するボタンをクリックすることでMech-Visionのユーザーマニュアルを閲覧できます。
4	最近使ったもの	最近使ったソリューションやプロジェクトを素早く開くことができます。

メイン画面

ようこそ画面を終了したら、下図のようなMech-Visionのメイン画面に入ります。



メイン画面は、以下の部分で構成されています。

番号	項目	説明
1	メニューバー	ファイル、編集、ビュー、ロボットと通信、カメラヘルパー、ディープラーニング、ツールキット、設定、ヘルプのタブが含まれています。
2	ツールバー	各ツールや機能のショートカットボタンがあります。
3	プロジェクトリスト	開いているソリューションやプロジェクト名、状態（通信状態や自動読み込み状態を含む）を確認できます。
4	ステップライブラリ	プロジェクトを構築するためのすべてのステップが含まれています。
5	プロジェクトツールバー	プロジェクト保存や実行、デバッグ結果出力などの操作を行うことができます。
6	プロジェクト編集エリア	ビジョンプロジェクト処理流れや論理を編集できます。
7	プロジェクト設定エリア	デバッグ結果出力、ステップパラメータ、プロジェクトアシスタント、ステップの説明、履歴およびステップのコメントが含まれています。
8	ログバー	プロジェクト実行中にログ情報をリアルタイムで表示できます。

4.1.2.1. メニューバー

ファイル(F) 編集(E) ビュー(V) ロボットと通信(R) カメラヘルパー(C) ディープラーニング(D) ツールキット(T) 設定(S) ヘルプ(H)

ソリューションやプロジェクト操作に関連する基本機能と、カメラやディープラーニングなどの補助ツールを提供します。

ファイル

ソリューションやプロジェクトの処理に使用されます。

オプション	説明	ショートカット
ソリューションを新規作成	空白ソリューションを作成します。	Ctrl+Shift+N
プロジェクトを新規作成	空白プロジェクトを新規作成	Ctrl+N
ソリューションライブラリから新規作成	ソリューションライブラリからソリューションを選択して新規作成します。	なし
ソリューションを開く	既存のソリューションを開きます。	なし
プロジェクトを開く	既存のプロジェクトを開きます。	Ctrl+O
最近使ったファイルを開く	最近使用したソリューションやプロジェクト名が表示され、クリックして開きます。	なし
実行可能ファイルの場所を開く	ソフトウェアのインストールディレクトリを開きます。	なし
ソリューションを保存	現在ソリューションへの変更を保存します。	Ctrl+Shift+S
プロジェクトを保存	現在プロジェクトへの変更を保存します。	Ctrl+S
プロジェクトをJSON形式として保存	プロジェクトを保存します。また、プロジェクトフォルダの.visファイルを.jsonファイルとして保存します。	なし
プロジェクトの名前を付けて保存	プロジェクトを指定場所に保存します。	なし
ソリューションを閉じる	選択したソリューションを閉じます。	なし
終了	Mech-Visionを終了します。	Ctrl+Q

編集

オプション	説明	ショートカット
取り消し	前の操作を取り消します。	Ctrl+Z
やり直し	前の操作をやり直します。	Ctrl+Y

ビュー

画面の表示設定を変更します。チェックを入れると、対応するタブが表示されます。

オプション	説明
プロジェクトリスト	プロジェクトリストを表示するかどうかを設定します。デフォルトでは、チェックが入っています。
ステップライブラリ	ステップライブラリを表示するかどうかを設定します。デフォルトでは、チェックが入っています。
デバッグ結果出力	「デバッグ結果出力」ウィンドウを表示するかどうかを設定します。デフォルトでチェックが入っています。また、チェックを外すことに対応しません。
履歴	「履歴」タブを表示するかどうかを設定します。デフォルトでは、チェックが入っています。
プロジェクトアシスタント	「プロジェクトアシスタント」タブを表示するかどうかを設定します。デフォルトでは、チェックが入っています。
ステップの説明	「ステップの説明」タブを表示するかどうかを設定します。デフォルトでは、チェックが入っています。
ステップのコメント	「ステップのコメント」タブを表示するかどうかを設定します。デフォルトでは、チェックが入っています。
ステップパラメータ	「ステップパラメータ」タブを表示するかどうかを設定します。デフォルトでは、チェックが入っています。
ステップの入力ポート選択	ステップの入力・出力ポートの選択のタブを表示するかどうかを設定します。デフォルトでは、チェックが入っていません。
測定結果の出力	測定結果の出力のタブを表示するかどうかを設定します。デフォルトでは、チェックが入っていません。
結果のビュー	結果ビュー（測定モードのみ使用可能）を表示するかどうかを設定します。デフォルトでは、チェックが入っていません。

オプション	説明
ログ	「ログ」タブを表示するかどうかを設定します。デフォルトでは、チェックが入っています。
ツールバー	ツールバーのタブを表示するかどうかを設定します。デフォルトでは、チェックが入っています。
レイアウトを初期状態に戻す	Mech-Vision 画面のレイアウトをデフォルトの状態に戻します。
位置合わせ	位置合わせグリッドと基準線を有効にするか無効にするかを設定できます。デフォルトでは、チェックが入っていません。
基準線のスタイル	アンカーポイント、水平方向の線、垂直方向の線を選択可能です。

ロボットと通信

ロボットと通信に関連する設定を行います。

オプション	説明
ロボット通信設定	ロボットとインターフェース通信を設定します。
ロボットライブラリツール	ロボットをインポート/エクスポートします。

カメラヘルパー

画像取得、カメラキャリブレーション、パラメータ分析に対応します。

オプション	説明	ショートカット
カメラビューア	画像をキャプチャして保存します。	Ctrl+Shift+V
カメラキャリブレーション	標準モード：通常のキャリブレーション手順です。 クイックモード：既存のパラメータに基づいてクイック計算します。	なし

ディープラーニング

オプション	説明	ショートカット
ディープラーニングモデルパッケージ管理ツール	ディープラーニングモデルパッケージ管理ツールを開きます。	なし
ディープラーニングサーバー	ディープラーニングサーバーを起動するかどうかを設定します。	Ctrl+Alt+D

ツールキット

補助ツールを開きます。

オプション	説明
マッチングモデル・把持位置姿勢エディタ	対象物の点群モデルの生成や編集、把持位置姿勢の追加に対応します。

設定

ソフトウェアの通常設定を変更できます。

オプション	説明	ショートカット
Mech-Centerアドレスを設定	Mech-Center ソフトウェアのIPアドレスを設定します。	なし
プロジェクトをロック	パラメータを入力して、現在開かれているプロジェクトをロックします。	なし
オプション	ソフトウェアの通用設定（言語、単位など）を変更できます。	Ctrl + Shift + O
ログレベル	保存するログメッセージのレベル（デバッグ情報、一般情報、ワーニング、エラーの4種類）を選択します。	なし

ヘルプ

ソフトウェアのバージョン情報、更新説明およびマニュアルを確認できます。

オプション	説明
Mech-Visionについて	現在のバージョン番号を確認できます。

オプション	説明
更新説明	更新説明のWebページに入ります。
マニュアル	ソフトウェアのマニュアルです。
プラグインについて	表示するプラグインにチェックを入れることができます。また、各プラグインの名前、バージョン、説明、ソフトウェア内の場所を確認できます。

4.1.2.2. ツールバー

ソリューションを新規作成したり、組み込みツールを使用したり、プロジェクトモードを切り替えたりすることができます。

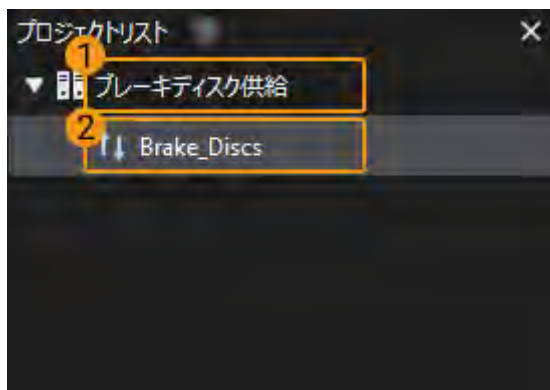




オプション	説明
ソリューションライブラリから新規作成	ソリューションをクイック作成できます。
ロボット通信設定	ロボットとインターフェース通信を設定します。
カメラキャリブレーション（標準モード）	カメラキャリブレーションの標準流れとなります。
マッチングモデル・把持位置姿勢エディタ	対象物の点群モデルの生成や編集、把持位置姿勢の追加に対応します。
標準編集モード	デフォルトモードとなります。
測定モード	計測、検出、位置決め、認識などのシーンで使用されます。
オペレーター・インターフェイス（カスタム）	カスタマイズされたオペレーター・インターフェイスとして、実際にニーズに応じて様々なデータを表示できます。
インターフェースサービス	インターフェースサービスを起動するために使用されます。

4.1.2.3. プロジェクトリスト

開かれたソリューションやプロジェクト、その状態（通信状態、自動読み込み状態を含む）を表示します。

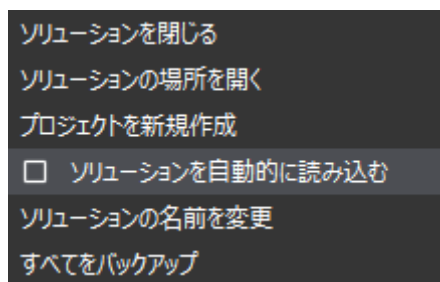
下図に示すように、1にソリューション名、2にプロジェクト名が表示されます。



- マウスを  に合わせると、ソリューションの説明が表示されます。
-  をクリックすると、ソリューションの関連マニュアルが表示されます。

ソリューションに関する操作

ソリューションを選択して右クリックすると、表示される画面でソリューションを閉じる、ソリューションの場所を開く、ソリューションを自動的に読み込む、ソリューションの名前を変更などが選択可能です。

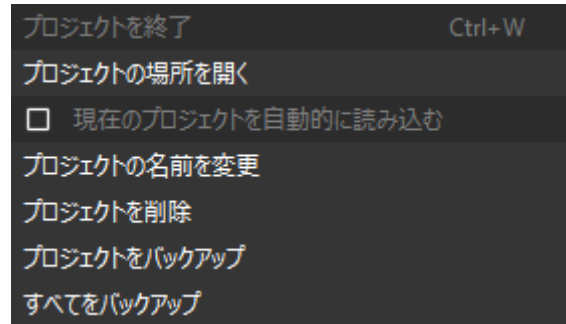


オプションとその説明は下表のとおりです。

オプション	説明
ソリューションを閉じる	現在のソリューションを閉じます。
ソリューションの場所を開く	ソリューションが格納しているフォルダを開きます。
ソリューションを自動的に読み込む	ソリューション内のすべてのプロジェクトを自動読み込みに設定します。
ソリューションの名前を変更	ソリューションの名前を変更します。
すべてをバックアップ	ソリューションをバックアップします。

プロジェクトに関する操作

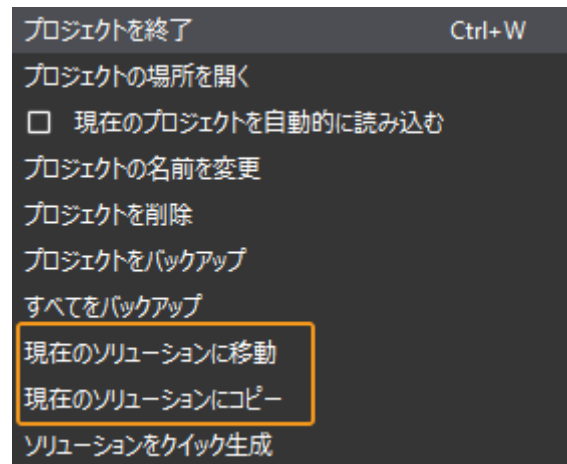
プロジェクトを選択して右クリックすると、表示される画面でプロジェクトを終了、プロジェクトの場所を開く、現在のプロジェクトを自動的に読み込む、プロジェクトの名前を変更などが選択可能です。



オプションとその説明は下表のとおりです。

オプション	説明	ショートカット
プロジェクトを閉じる	現在のプロジェクトを閉じます。	Ctrl+W
プロジェクトの場所を開く	プロジェクトが格納しているフォルダを開きます。	なし
プロジェクトの自動読み込み	選択したプロジェクトを自動読み込みに設定します。	なし
プロジェクトの名前を変更	選択したプロジェクトの名前を変更します。	なし
プロジェクトを削除	選択したプロジェクトを削除します。	なし
プロジェクトをバックアップ	選択したプロジェクトをバックアップします。	なし
すべてをバックアップ	すべてのプロジェクトのバックアップを作成します。	なし
プロジェクトIDをリセット	プロジェクトIDをリセットします。	なし

ソリューションが開かれ、プロジェクトが**未割り当て**の状態である場合、このプロジェクトは**現在のソリューションに移動・コピー**することができます。



オプションとその説明は下表のとおりです。

オプション	説明
現在のソリューションに移動	選択したプロジェクトを開かれているソリューションに移動します。
現在のソリューションにコピー	選択したプロジェクトを開かれているソリューションにコピーします。

4.1.2.4. ステップライブラリ

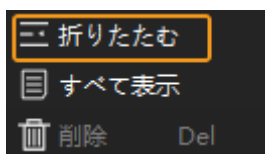
下図に示すように、ステップライブラリにはプロジェクト構築に使用できるすべてのステップが含まれます。



ステップの表示

ステップを展開/折りたたむ




ステップライブラリツリーが完全に開いている状態で、ステップライブラリを右クリックし、**折りたたむ**をクリックするとステップツリー全体が折りたたまれます。



ステップライブラリツリーが完全に折りたたまれている状態で、ステップライブラリを右クリックし、**展開**をクリックするとステップツリー全体が展開されます。



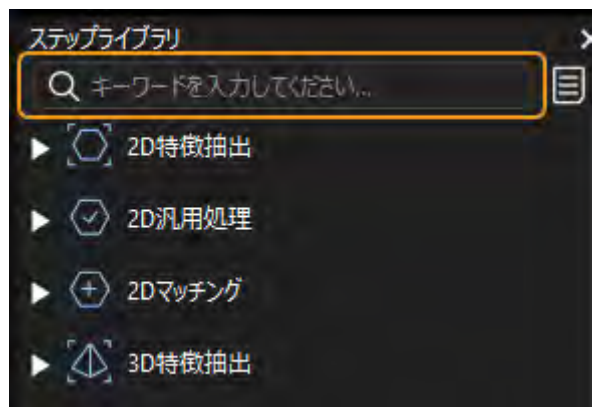
よく使うステップのみを表示/すべてのステップを表示

検索ボックスの右側にあるアイコンが  の状態になっている場合、よく使うステップのみが表示されます。すべてのステップを表示するには、 をクリックして  の状態になると、すべてのステップが表示されます。または、ステップライブラリの任意場所を右クリックし、[すべて表示]をクリックします。

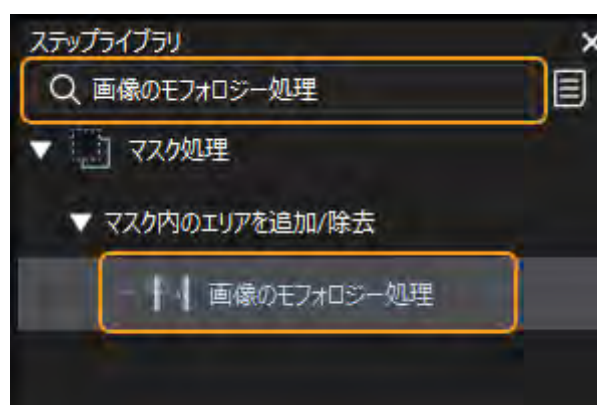


ステップの検索

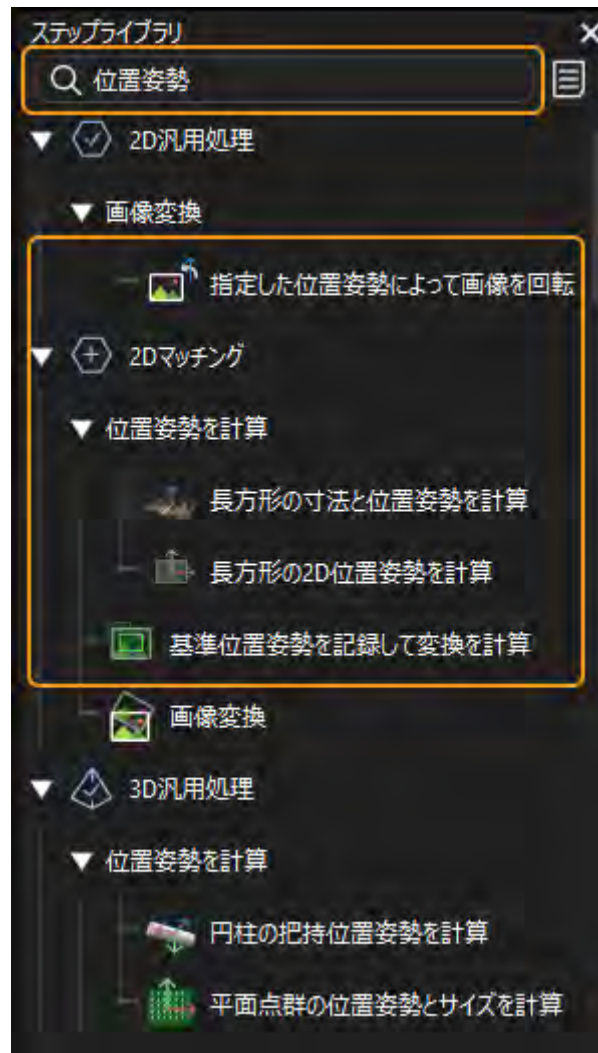
検索はステップライブラリの一番上にある検索ボックスから検索することができます。ステップの完全な名前またはキーワードで検索をするとステップが見つかります。



ステップの完全な名前で検索をすると、該当ステップのみが表示されます。下図に示すように、「画像のモフォロジー処理」を例とします。



キーワードで検索をすると、キーワードに関連するすべてのステップが見つかります。下図に示すように、「位置姿勢」を例とします。



4.1.2.5. プロジェクトツールバー

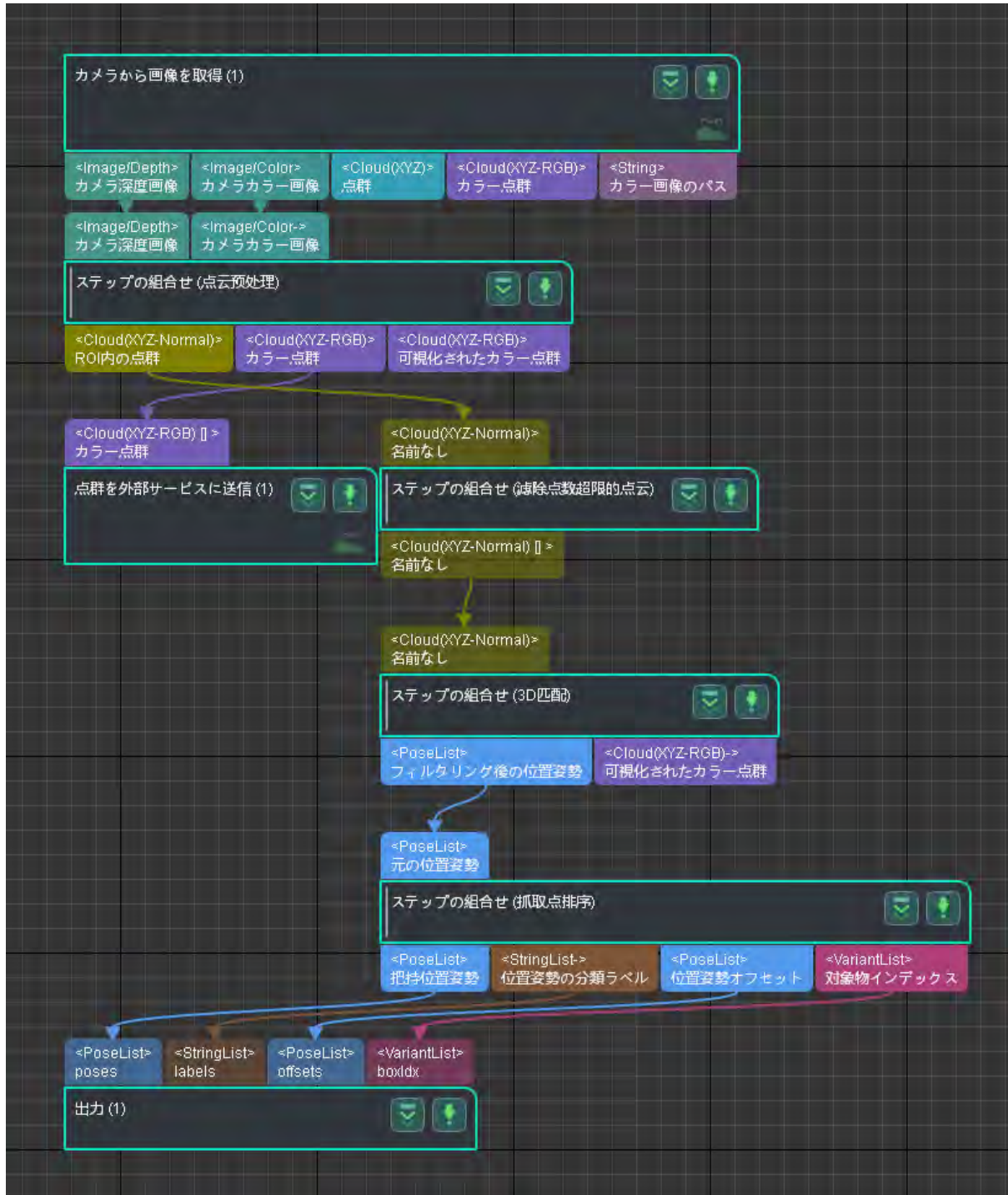
プロジェクトの保存、実行、デバッグ結果出力などに対応します。



オプション	説明
保存	プロジェクトを保存
取り消し	操作を取り消します。
やり直し	操作をやり直します。
実行	プロジェクトを実行します。
デバッグ結果出力	「デバッグ結果出力」を有効にすると、追加のデバッグ情報が表示されます。

4.1.2.6. プロジェクト編集エリア

プロジェクト編集エリアは、ソフトウェアのデバッグ作業において重要なエリアであり、ビジョンプロジェクトの処理流れや論理を編集するために使用されます。ステップのつながりは、標準編集モードでデータ流れを示し、測定モードで処理流れを示しています。



プロジェクト編集エリアの任意場所で右クリックして、ポップアップメニューが表示されます。実際のニーズに応じて選択してください。



各オプションの説明は次の表に示します。

オプション	説明	ショートカット
一覧表示に戻す	ステップの組合せの一覧表示に戻します	PgUp
詳細を表示	ステップの組合せの詳細を表示します（ステップの組合せを右クリックで有効にする）	PgDown
簡易モードに切り替える	プロジェクト内の全てのステップのポートが1つのポートで表示します	なし
コピー	ステップをコピーします	Ctrl+C

オプション	説明	ショートカット
切り取り	ステップを切り取ります	Ctrl+X
貼り付け	ステップを貼り付けます	Ctrl+V
削除	ステップまたはつながきを削除します	Del
ステップを検索	プロジェクトにステップを検索します	Ctrl+F
自動レイアウト	プロジェクト内のステップを自動的に配置します	Alt+Shift+F
すべて選択	プロジェクト内のステップおよびデータフローをすべて選択します	Ctrl+A
プロジェクトのスクリーンショットを保存	プロジェクトのスクリーンショットを指定場所に保存します	なし
データフローを表示	デバッグ結果出力の画面に 入力・出力データを表示します (データフローを右クリックで有効にする)	Shift+V
ステップの組合せのパラメータを編集	ステップの組合せの表示パラメータを編集します (ステップの組合せを右クリックで有効にする)	なし
ステップをステップの組合せに合成	複数のステップを選択した場合、複数のステップをステップの組合せとして作成することができます	なし
ステップの組合せをステップに分割	ステップの組み合わせを選択して、複数のステップに分割することができます	なし
ファイルからステップの組合せを読み込む	既存の.jsonファイルからステップの組合せを読み込みます	なし
ステップの組合せをファイルに保存	ステップの組合せを.jsonファイルとして保存します	なし

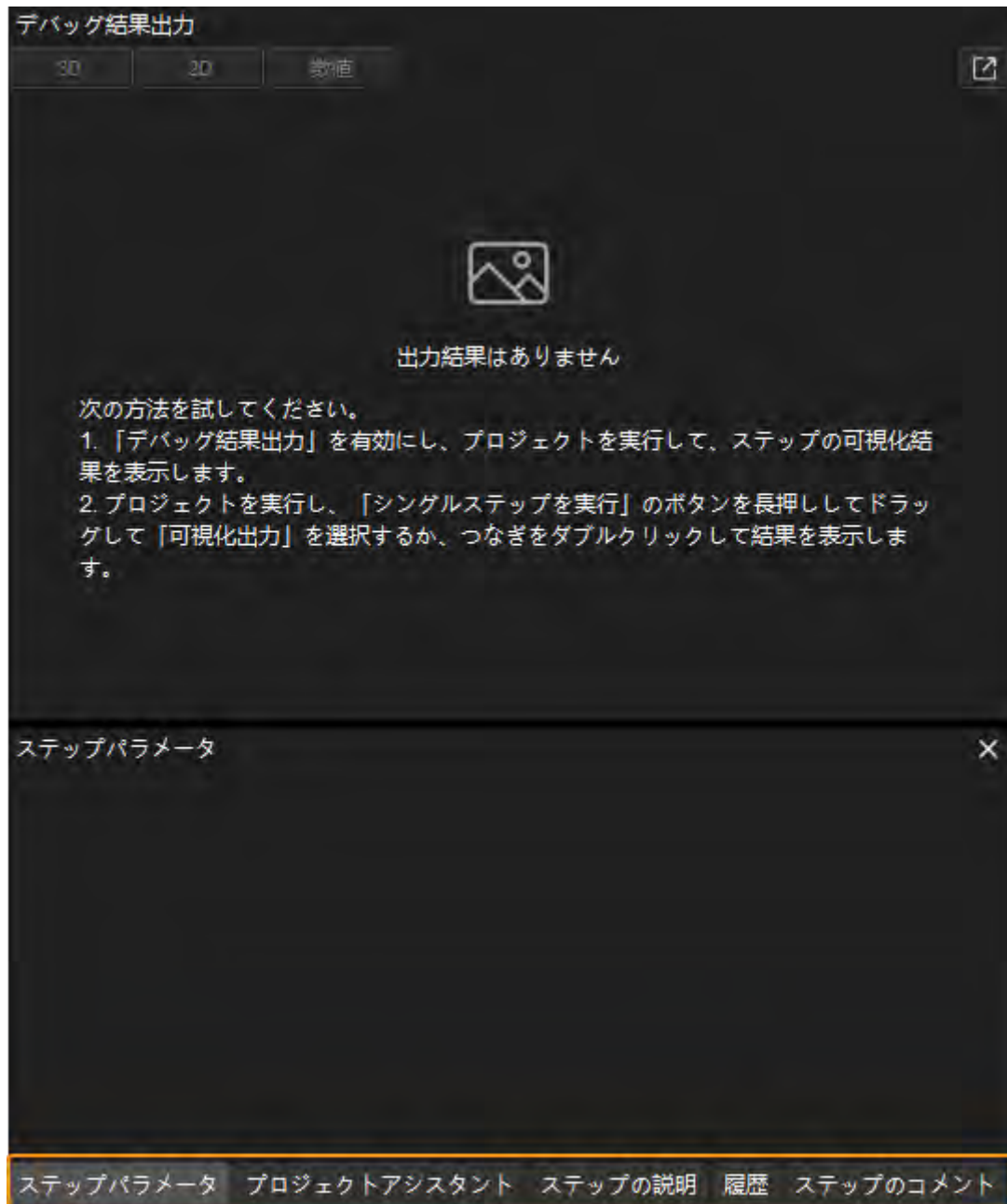
オプション	説明	ショートカット
ステップの組合せをテンプレートとして保存	選択したステップの組合せをステップライブラリの「カスタマイズ」グループに保存します	なし
現在選択されているステップのコメントを追加	現在選択されているステップにコメントを追加します	なし
すべてのステップの実行フラグを設定	テキスト出力をキャンセル	なし
	ファイルの再読み込みを設定/キャンセル	なし
	出力なしで実行継続を設定/キャンセル	なし
ステップポートの表示を設定	タイプと名前をすべて表示	Ctrl+Alt+A
	名前のみを表示	Ctrl+Alt+N
	タイプのみを表示	Ctrl+Alt+T

4.1.2.7. プロジェクト設定エリア

プロジェクト設定エリアには、デバッグ結果出力、ステップパラメータ、プロジェクトアシスタント、ステップの説明、履歴、およびステップのコメントなどが含まれます。

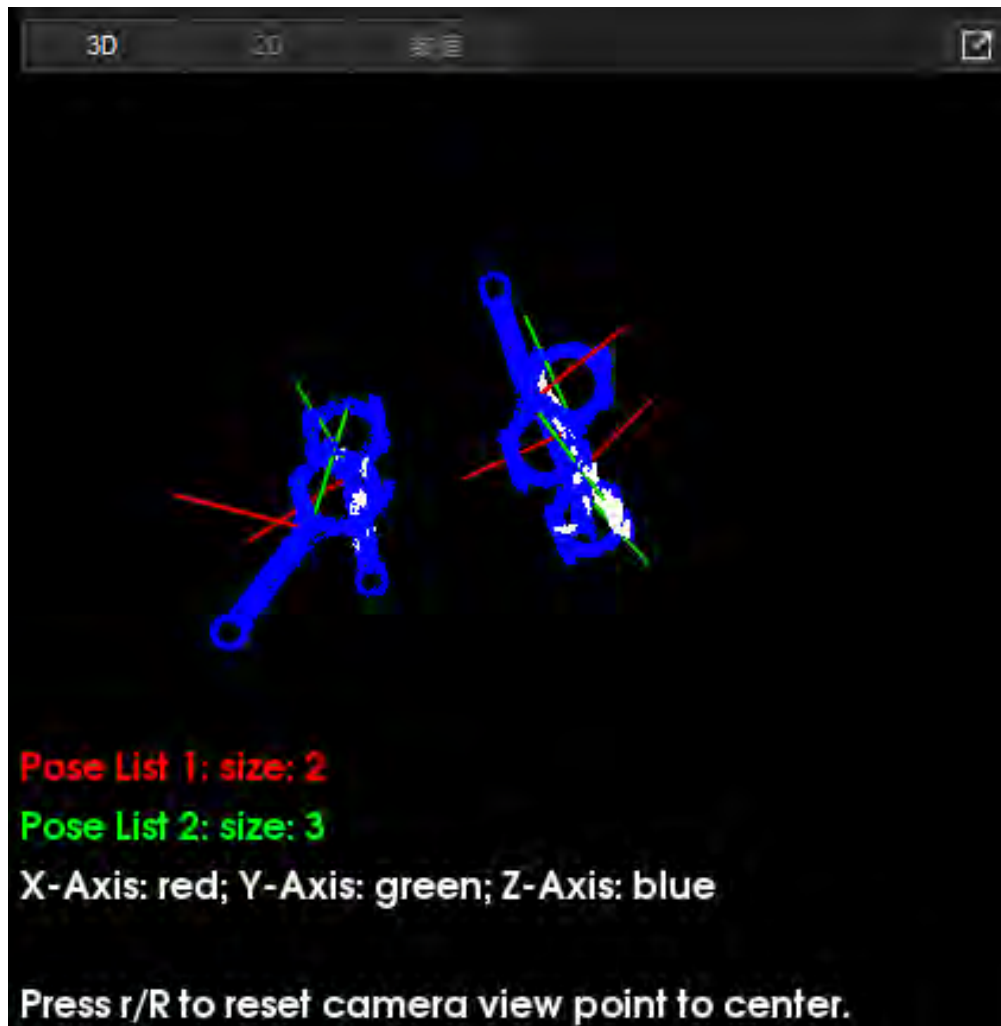


プロジェクト設定エリアにタブが表示されていない場合、**メニューバー**の**ビュー**でチェックが入っているかどうかを確認することができます。



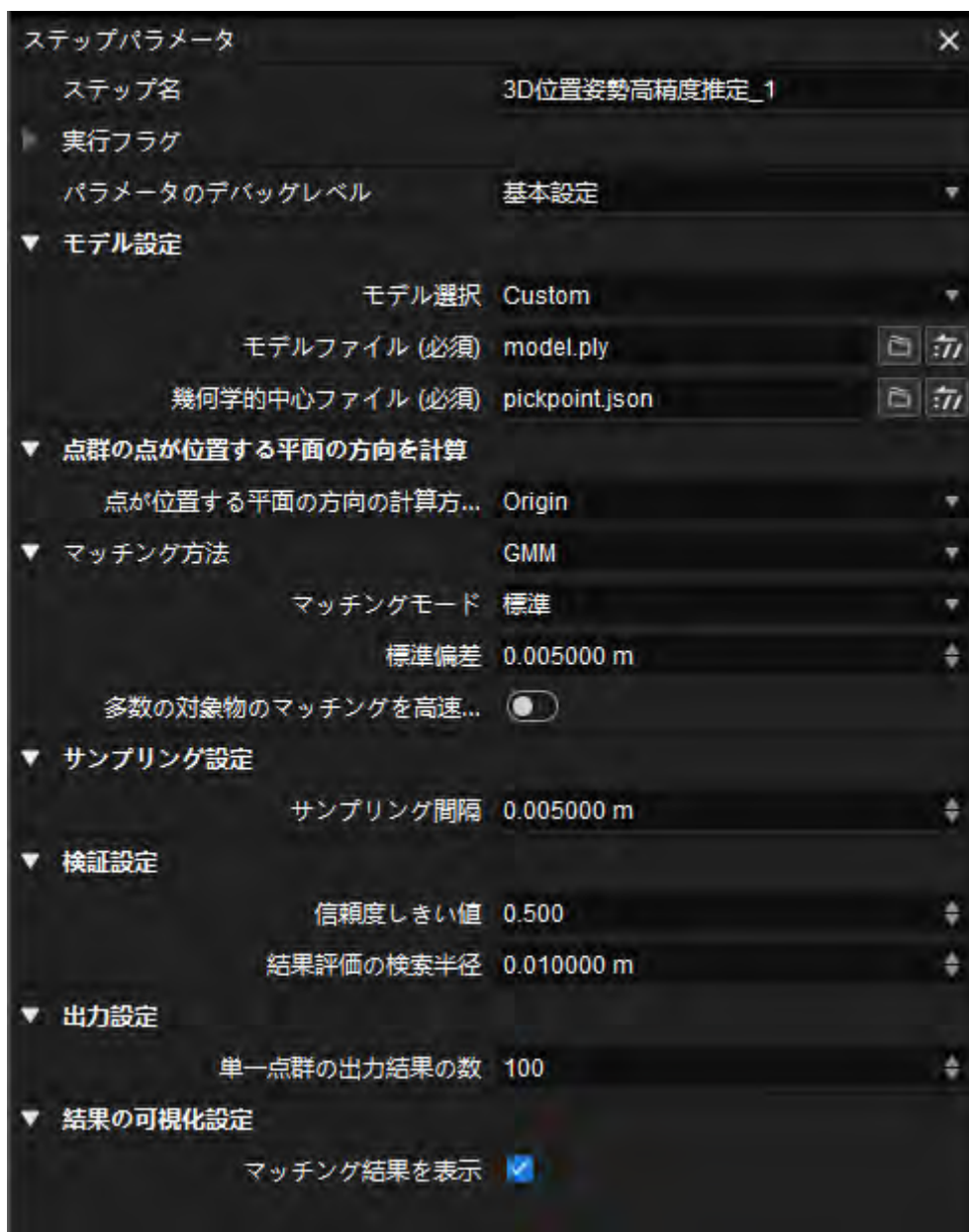
デバッグ結果出力

「デバッグ結果出力」を有効にすれば、実行中に必要なステップの出力結果が表示されます。詳細については、[ステップを実行して可視化出力結果を表示](#)をご参照ください。

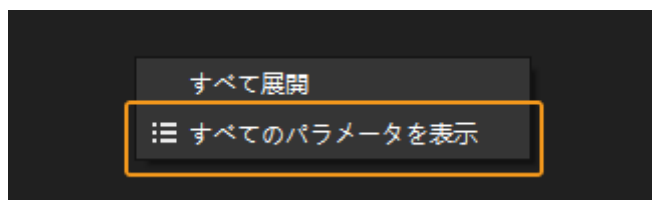


ステップパラメータ

ステップを選択して、ここでこのステップのパラメータを調整できます。

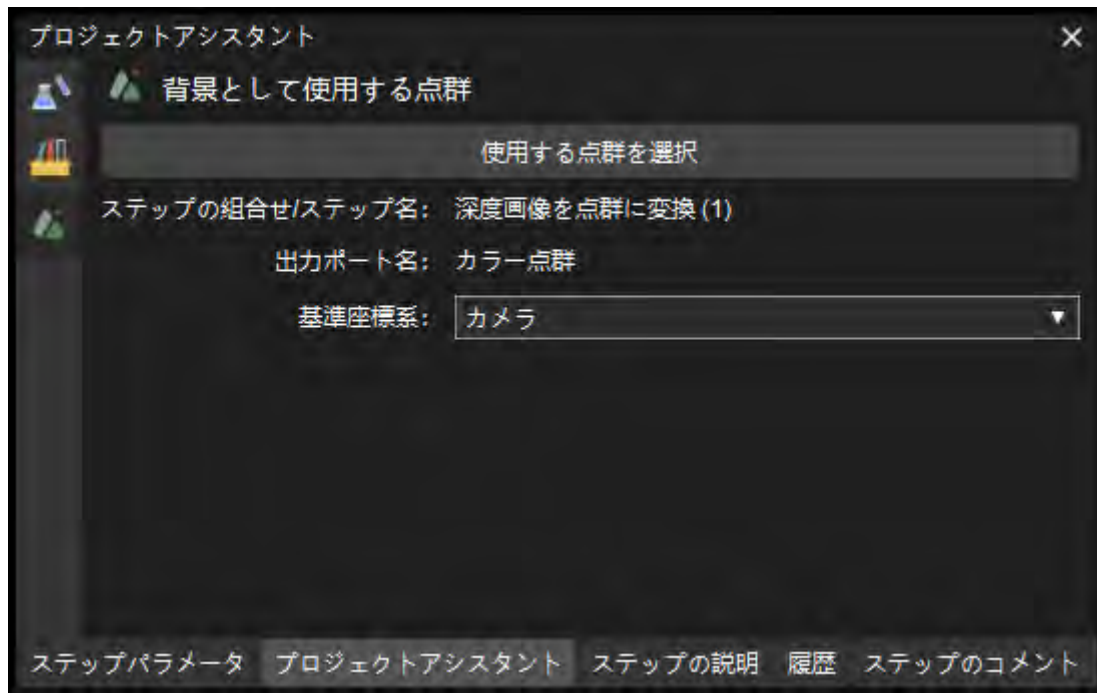


ステップパラメータタブには、デフォルトでよく使うパラメータのみが表示されます。すべてのパラメータを表示するには、このタブの空白スペースを右クリックし、**すべてのパラメータを表示**をクリックします。



プロジェクトアシスタント

プロジェクトに関連する設定を表示・設定します。

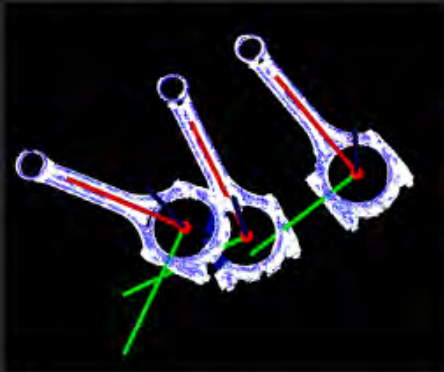


ステップの説明

下図に示すように、ステップを選択して、ここでこのステップの機能、使用シーンおよび入出力の説明が表示されます。

ステップの説明
×

機能のディスクリプション:
点群モデルを使用して元の点群に対して精確にマッチングし、対象物の更に正確な位置姿勢を出力します。



使用シーン:
このステップは、「3D位置姿勢低精度推定」に出力された大まかな候補位置姿勢に基づいてもっと精確なマッチングを実行し、より正確な対象物の位置姿勢（把持位置姿勢にすることはできる）を出力します。

[詳細情報 \(F1\)](#)

入力:

- このポートに入力される点群リストはマッチングされます（通常は、「3D位置姿勢低精度推定」ステップで使用される点群と同じです）。
- 候補位置姿勢リスト（通常は、「3D位置姿勢低精度推定」ステップから取得された大まかな位置姿勢です）。
- 位置姿勢の分類ラベルリスト（通常は、ディープラーニングの関連ステップによって出力された分類ラベルのリストです）。
- 位置姿勢の信頼度リスト（通常は、ディープラーニングの関連ステップによって出力された信頼度のリストです）。

出力:

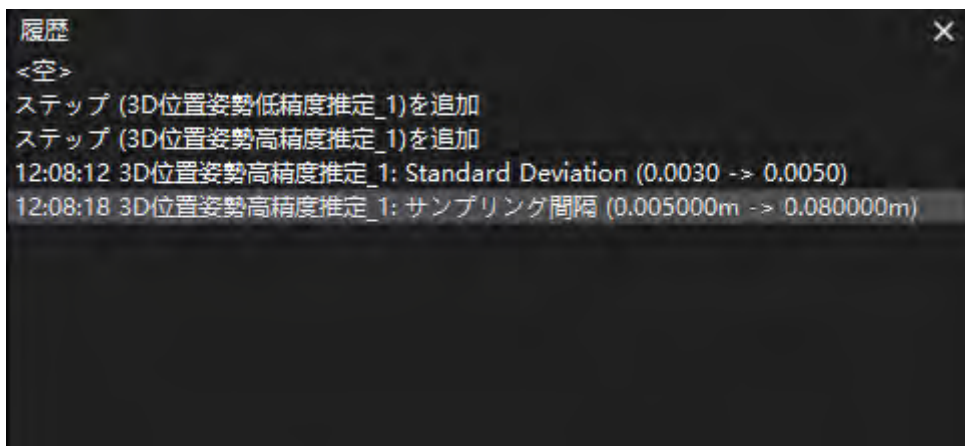
- 把持位置姿勢リスト。
- 位置姿勢推定によって生成されたモデル点群リスト。
- モデルからシーンへの変換リスト。
- 把持位置姿勢のラベルリスト。
- 把持位置姿勢の信頼度リスト。
- 各位置姿勢の推定スコア。



F1を押して、対応するドキュメントページにジャンプします。

履歴

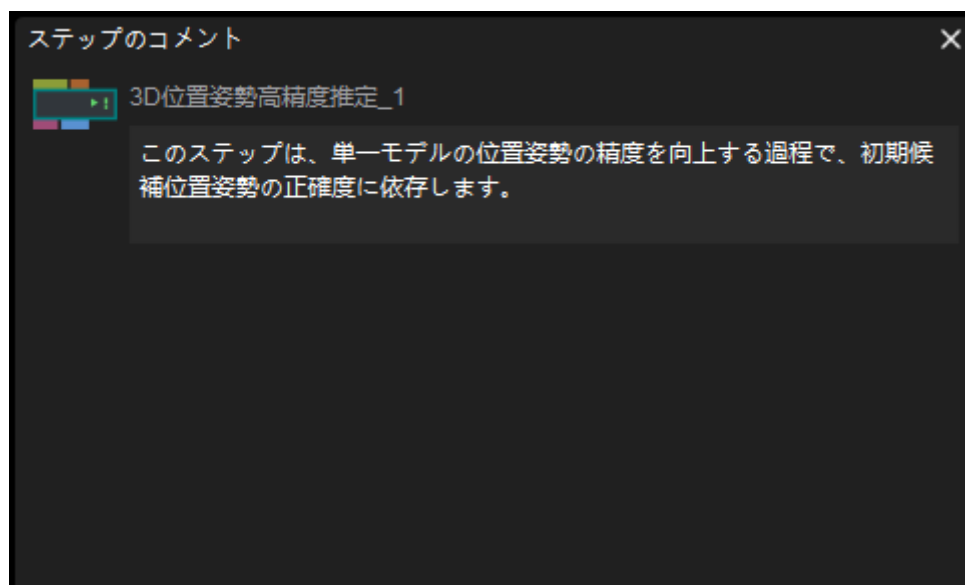
操作履歴を記録して表示し、項目をクリックすると、項目の後のすべての操作ステップをすばやく取り消すことができます。



「測定モード入力」または「測定結果出力」ビューを使用する必要がある場合は、「ビュー」タブから対応するオプションをチェックします。

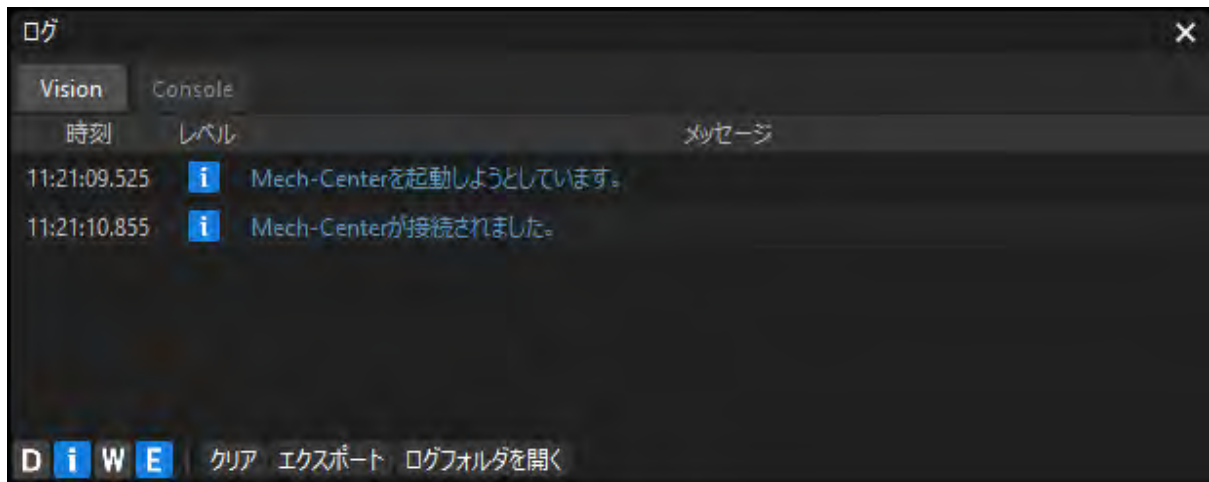
ステップのコメント

ステップを選択して、コメントを追加・表示することができます。詳細については、[ステップのコメントを追加・変更・表示](#) をご参照ください。



4.1.2.8. ログバー

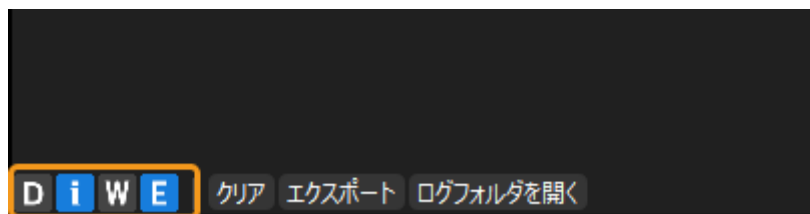
ソリューション/プロジェクトの実行中にログ情報をリアルタイムで表示でき、特定時間に実行中のレコードを確認するのに便利です。



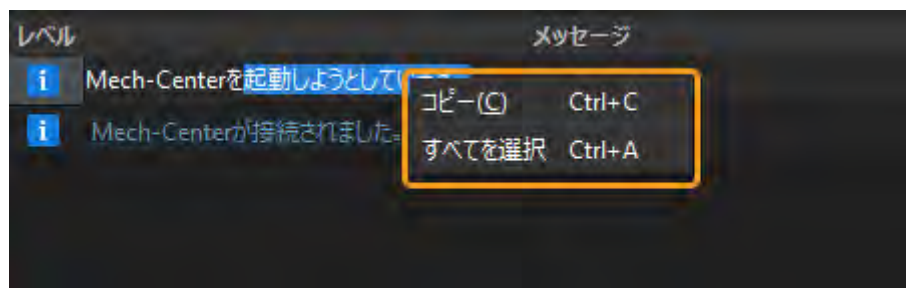
ログレベルの説明は次の表に示します。

ログレベル	説明
D	デバッグ情報
i	一般情報
W	ワーニング
E	エラー

ログバーの左下隅にあるログレベルをクリックして、表示するレベル情報を選択します。ログレベルボタンの色が青色の場合は表示され、ログレベルボタンの色が灰色の場合は表示されません。



ある行のメッセージをダブルクリックし、メッセージを選択し、マウスを右クリックしてコピーを選択して選択したコンテンツをコピーします。

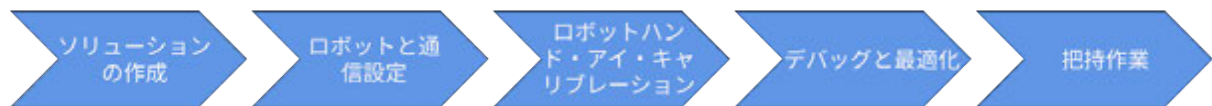


4.1.3. ソリューションの構築手順

本節では、ソリューションの構築方法について説明します。

ソリューションとは、ビジョンソリューションを実現するために必要なロボットと通信、ビジョン処理、経路計画などの機能設定とデータの集りのことです。

下図に示すように、ビジョンソリューションの実装は、一般的に5つの段階に分けられます。



ソリューションの作成

ソリューションとは、ビジョンソリューションを実現するために必要なロボットと通信、ビジョン処理、経路計画などの機能設定とデータの集りのことです。ソリューションを実装する前に、ソリューションを作成する必要があります。

1. [\[vision-operation-guide:create-and-save-solution:::create-and-save-solution\]](#)
2. [\[vision-operation-guide:auto-load-solution:::autoload-solution\]](#)
3. [プロジェクトの構築](#)

ロボット通信設定

ソリューションを新規作成したら、ロボットと通信の設定を行うことが必要です。

通信とは、ビジョン側がロボット側と情報を交換・転送することです。正常に通信できるように、**ビジョン側**と**ロボット側**で通信設定を完了させる必要があります。

Mech-Mindビジョンシステムでは、インターフェース通信とVizティーチング通信を利用可能です。その中、インターフェース通信が標準インターフェース通信とAdapter通信に分けられます。

インターフェース通信

インターフェース通信を使用する場合、ビジョン側とロボット側の通信設定を別々に行った後に、ロボット側とビジョン側が正常に通信できることをテストする必要があります。

ロボット側とビジョン側が正常に通信できるかどうかをテストする方法は、ロボットブランドによって異なります。

1. [ビジョン側の通信設定](#)
2. [ロボット側で通信設定を行い、ロボット側とビジョン側が正常に通信できるかどうかをテスト](#)

Vizティーチング通信

Vizティーチング通信を使用する場合、ロボット側で通信設定を行った後に、ロボット側とビジョン側が正常に通信できることをテストする必要があります。

1. ロボット側の通信設定
2. ロボット側とビジョン側が正常に通信できることをテスト

ロボットのハンド・アイ・キャリブレーション

ロボットと通信設定が完了後、[ロボットのハンド・アイ・キャリブレーション](#)を実行します。

ハンド・アイ・キャリブレーションとは、カメラ座標系とロボット座標系との対応関係を求めることです。ビジョンシステムにより取得した対象物の位置姿勢をロボット座標系に変換し、ロボットが対象物の把持を完了させるように制御します。

キャリブレーション結果の精度や安定性は、ロボットの把持作業の精度に影響します。

デバッグと最適化

上記のことを完了したら、プロジェクトを実行・デバッグすることができます。

1. [\[vision-operation-guide:run-project-and-debug:::run-project-and-debug\]](#)
2. [\[vision-operation-guide:run-and-view-outputs:::run-and-view-outputs\]](#)

把持作業

プロジェクトの実行やデバッグが完了後、ロボット実機を使用した把持作業を実行できます。

- インターフェース通信を使用する場合、ロボットの把持プログラムを作成する必要があります。これにより、標準インターフェース通信で Mech-Vision プロジェクトを実行し、Mech-Vision から出力されたビジョン位置姿勢や計画された経路点を取得します。
- Vizティーチング通信を使用する場合、Mech-Viz プロジェクトを構築することで経路点を計画します。

4.2. 基本操作

本節では、Mech-Visionの操作方法について説明します。

[ソリューションライブラリの操作方法](#)については、以下の内容をお読みください。

[ソリューションライブラリに関する操作](#)

[ソリューションやプロジェクトの操作方法](#)については、以下の内容をお読みください。

[ソリューションに関する操作](#)

[プロジェクトに関する操作](#)

通信関連の操作方法については、以下の内容をお読みください。

[ロボット通信設定に関する操作](#)

ステップ関連の操作方法については、以下の内容をお読みください。

[ステップに関する操作](#)

4.2.1. ソリューションライブラリに関する操作

Mech-Visionのソリューションライブラリでは、様々な業界のソリューションやプロジェクト事例に加え、サンプルデータも提供しています。

本節では、ソリューションライブラリの基本操作について説明します。

[ソリューションライブラリを開く](#)

[ソリューションやプロジェクトをフィルター/検索](#)

[ソリューションやプロジェクト情報を表示](#)

[ソリューションやプロジェクトのヘルプマニュアルを閲覧](#)

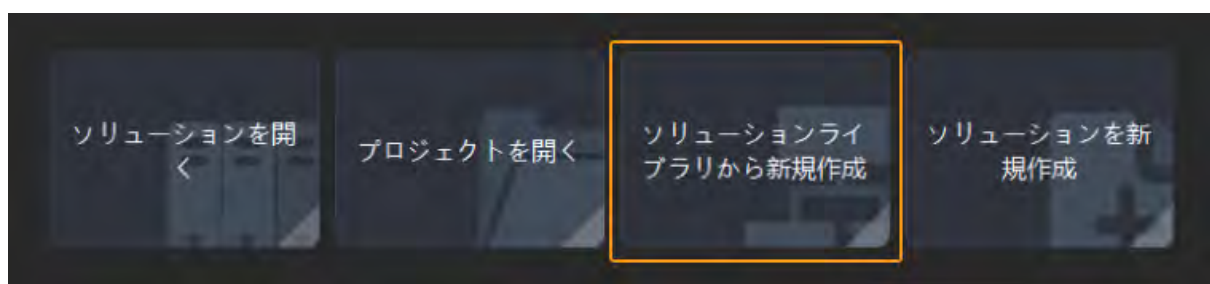
[ソリューションライブラリからソリューションを新規作成](#)

4.2.1.1. ソリューションライブラリを開く

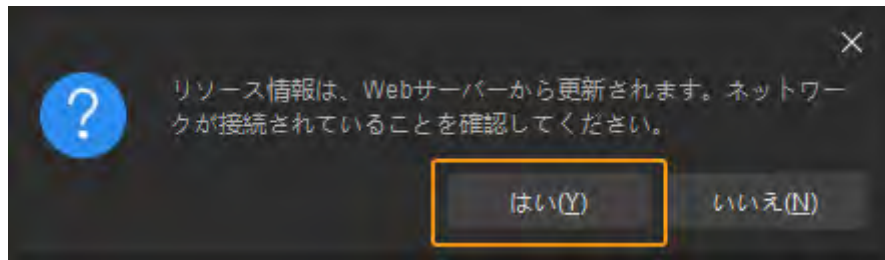
本節では、ソリューションライブラリを開く方法について説明します。

ようこそ画面から開く

Mech-Visionを起動し、ようこそ画面で[**ソリューションライブラリから新規作成**]をクリックします。すると、ソリューションライブラリが開かれます。



ソリューションライブラリには、プリインストールされたリソースがあります。そのほかのリソースについては、下部にある[もっと詳しく]をクリックし、表示される画面で[はい (Y)]をクリックするとソリューションライブラリのリソースを取得することができます。



下図に示すように、リソースの読み込みが完了すると、他のリソースが表示されます。



- **ソリューション**：ビジョンアプリケーションを実現するために必要なロボットと通信、ビジョン処理、経路計画などの機能設定とデータの集りのことです。
- **特集**：プロジェクトは現場で何度も検証されており、幅広いシーンに適応可能です。
- **組込み**：プロジェクトは、ソフトウェアのインストールパッケージと一緒にローカルディレクトリにインストールされており、ダウンロードせずに使用することが可能です。

ソフトウェアのメイン画面から開く

ソフトウェアのメイン画面に入った場合、ソリューションライブラリを開くには、次のいずれの方法を使用します。

- ツールバーで[ソリューションライブラリから新規作成]をクリックします。
- メニューバーで**ファイル**、**ソリューションライブラリから新規作成**をクリックします。

4.2.1.2. ソリューションやプロジェクトをフィルター/検索

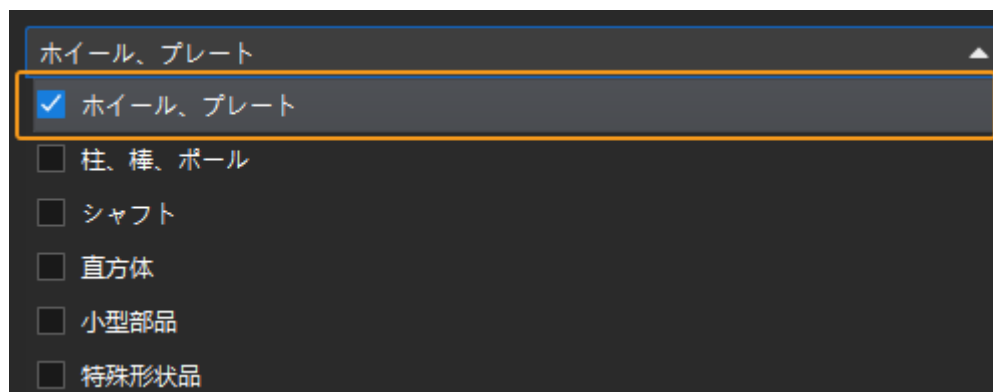
ソリューションライブラリでは、業界に従ってソリューションやプロジェクトを検索することに対応します。また、部品供給における、部品外観とプロジェクトにおける難点に従ってソリューションやプロジェクトをフィルタリングすることが可能です。

部品供給業界の混在している対象物という難点があるブレーキディスクのプロジェクトを例として説明します。

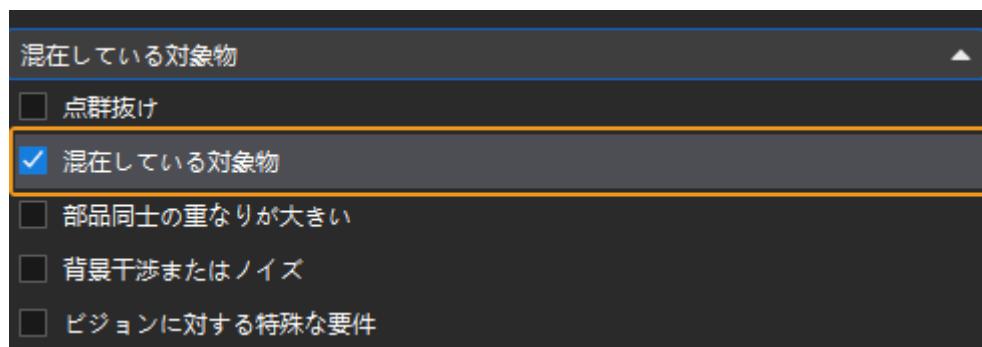
1. 業界に従って検索します。画面の上部にある[部品供給]をクリックすると、関連するソリューションやプロジェクトが表示されます。



2. 部品外観に従って検索します。ドロップダウンリストでホイール、プレートにチェックを入れると、関連するソリューションやプロジェクトが表示されます。



3. プロジェクトにおける難点に従って検索します。ドロップダウンリストで混在している対象物にチェックを入れると、関連するソリューションやプロジェクトが表示されます。



上記が完了すると、ブレーキディスクのプロジェクトを見つけることができます。

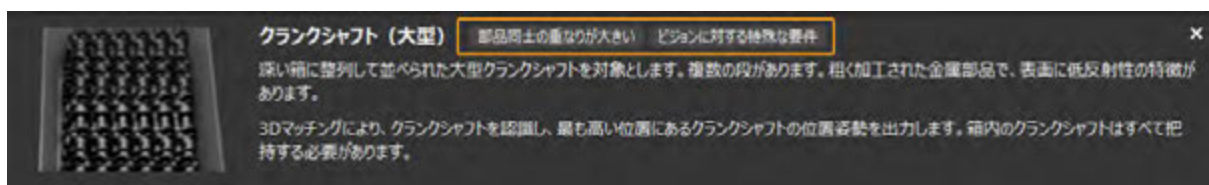


4.2.1.3. ソリューションやプロジェクト情報を表示

ソリューションやプロジェクトの情報を表示するには、ソリューションやプロジェクトをクリックすると、関連情報が画面の下部に表示されます。



下図に示すように、プロジェクト名の右側にあるのはプロジェクトにおける難点のラベルです。



4.2.1.4. ソリューションやプロジェクトのヘルプマニュアルを閲覧

ソリューションまたはプロジェクトのヘルプマニュアルには、ソリューションまたはプロジェクトの適用シーン、技術仕様、プロジェクトにおける難点、ビジョンソリューション、パラメ

ータ調整のアドバイスなどが含まれます。

ソリューションやプロジェクトにヘルプマニュアルが備えられている場合、ソリューションやプロジェクトをクリックすると、画面の下部に[詳細はこちら]ボタンが表示されます。

ソリューションやプロジェクトのヘルプマニュアルを表示するには、ソリューションやプロジェクトをクリックし、下部にある[詳細はこちら]をクリックすると、表示される画面でそのソリューションやプロジェクトのヘルプマニュアルを確認できます。

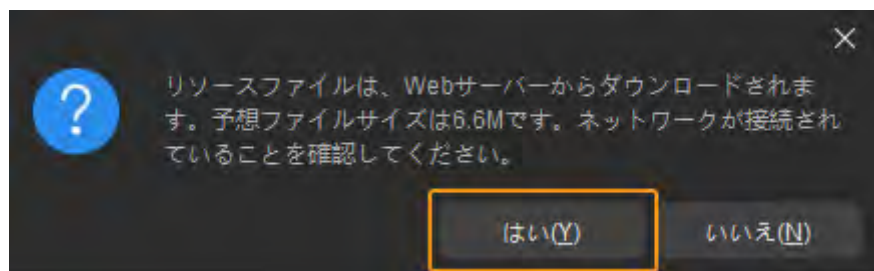


4.2.1.5. ソリューションライブラリからソリューションを新規作成

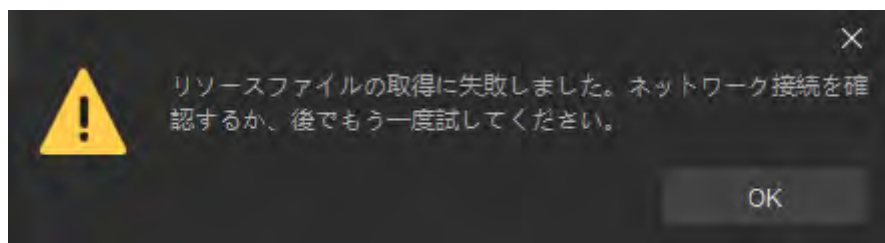
ソリューションライブラリからソリューションを新規作成するには、[ソリューションライブラリを開き](#)、クリックしてソリューションを選択し、画面の下部にある**ソリューション名**と**保存先**を記入して[作成]をクリックします。



表示される画面で[はい (Y)]をクリックすると、ソリューションを指定したパスに作成します。



下図のような画面が表示される場合は、バックグラウンドに関連するソリューションがないことを意味します。まず、ネットワーク接続を確認し、正常に動作していることを確認してから再試行してください。それでも問題が解決しない場合は、Mech-Mind株式会社にお問い合わせください。



4.2.2. ソリューションに関する操作

ソリューションとは、ビジョンソリューションを実現するために必要なロボットと通信、ビジョン処理、経路計画などの機能設定とデータの集りのことです。

本節では、ソリューション関連の操作方法について説明します。

ソリューションを新規作成して保存

ソリューションを開く

ソリューションの自動読み込み

ソリューションの名前を変更

ソリューションを閉じる

ソリューションのファイル構造

4.2.2.1. ソリューションを新規作成して保存

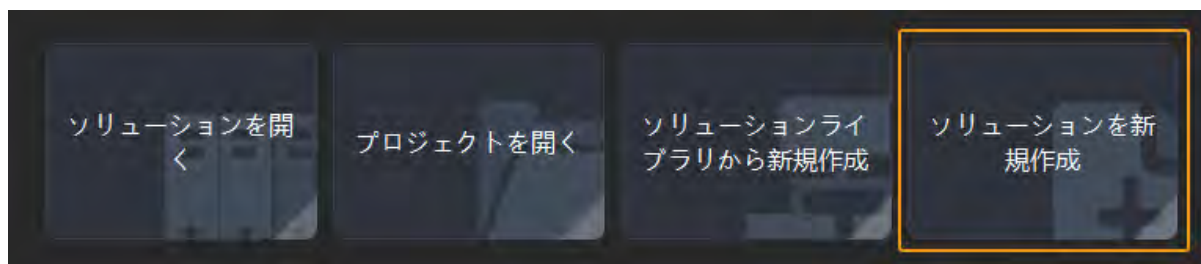
本節では、ソリューションの新規作成や保存方法について説明します。

ソリューションを新規作成

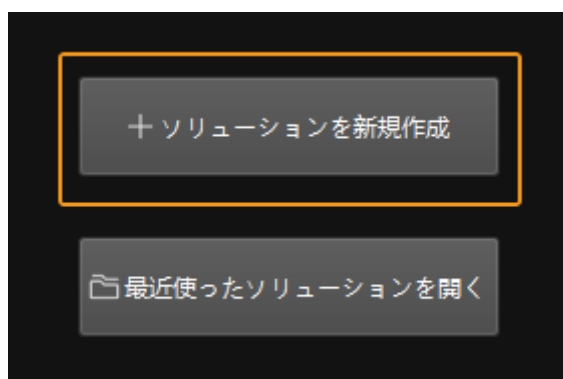
空白ソリューションを作成

空白ソリューションを新規作成するには、次のいずれの方法を使用します。

- ようこそ画面で[**ソリューションを新規作成**]をクリックして作成します。



- イン画面のメニューバーで**ファイル**・**ソリューションを新規作成**をクリックして作成します。
- ソフトウェアでソリューションやプロジェクトが開かれていない場合、プロジェクト編集エリアの[**ソリューションを新規作成**]をクリックして作成します。



ソリューションライブラリから新規作成

詳細については、[ソリューションライブラリからソリューションやプロジェクトを新規作成](#)をご参照ください。

ソリューションを保存

ソリューションを作成したら、保存パスを設定し、適時保存することをお勧めします。

ソリューションを保存するには、次のいずれの方法を使用します。

- メニューバーで**ファイル**、**ソリューションを保存**をクリックして保存します。
- ショートカットキー **Ctrl** + **Shift** + **S** で保存します。



- ソリューションの名前を変更するには、[ソリューションの名前を変更](#)をご参照ください。
- 次回ソフトウェアを起動したときにソリューションを自動的に開くには、[ソリューションの自動読み込み](#)をご参照ください。

4.2.2.2. ソリューションを開く

本節では、ソリューションを開く方法について説明します。

既存のソリューションを開く

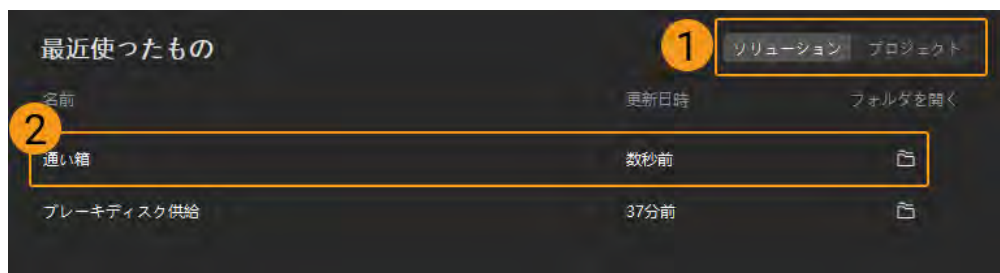
既存のソリューションを開くには、次のいずれの方法を使用します。

- ようこそ画面で[**ソリューションを開く**]をクリックし、表示されるファイル選択画面でソリューションフォルダを選択して開きます。
- メニューバーで**ファイル**、**ソリューションを開く**を順番に選択し、表示されるファイル選択画面でソリューションフォルダを選択して開きます。

最近使ったソリューションを開く


最近使ったソリューションを開くには、次のいずれの方法を使用します。

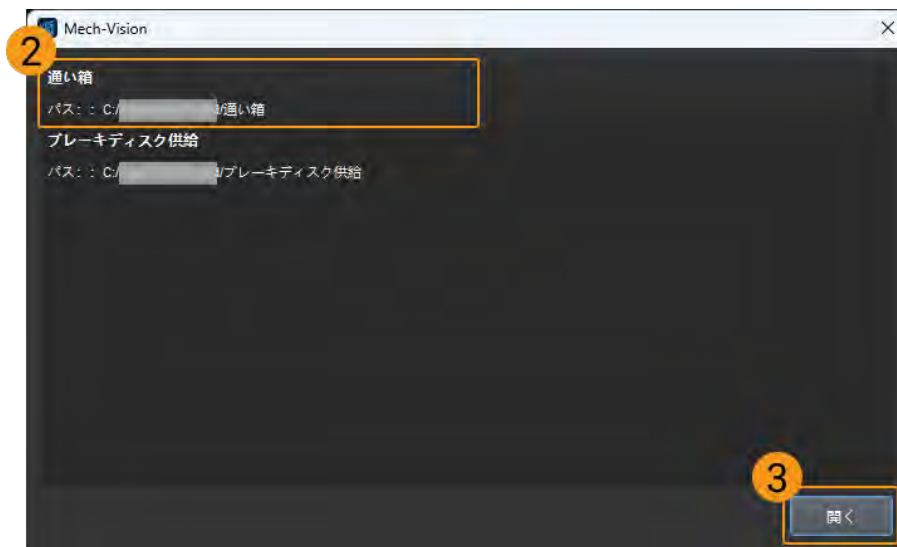
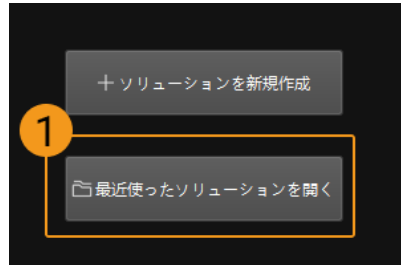
- ようこそ画面の**最近使ったもの**で右側のソリューション/プロジェクト切り替えボタンを**ソリューション**に切り替え、ソリューションをダブルクリックします。



- メニューバーで**ファイル**、**最近使ったファイルを開く**を順番に選択し、最近使ったソリュー

ションをクリックします。

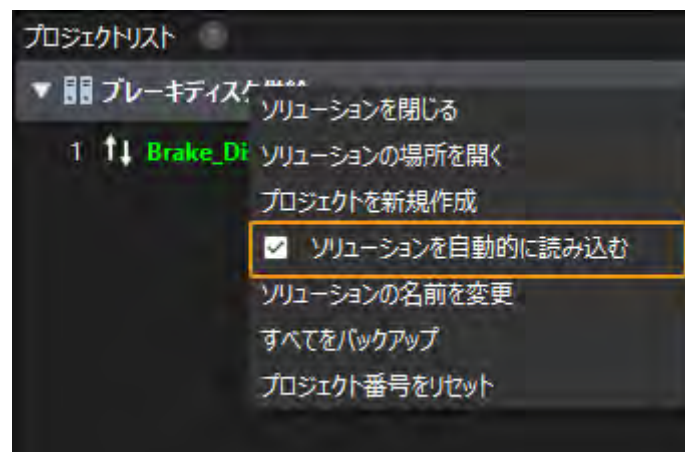
- ソリューションが開けない場合は、初期画面のプロジェクト編集エリアにある[最近使ったソリューションを開く]をクリックします。その後、表示される画面でソリューションを選択し、をクリックします。



4.2.2.3. ソリューションの自動読み込み

次回から Mech-Vision を開く時にソリューションと一緒に開かれるには、次の操作を実行します。

下図に示すように、プロジェクトリストからソリューションを選択し、右クリックして**ソリューションを自動的に読み込む**にチェックを入れます。



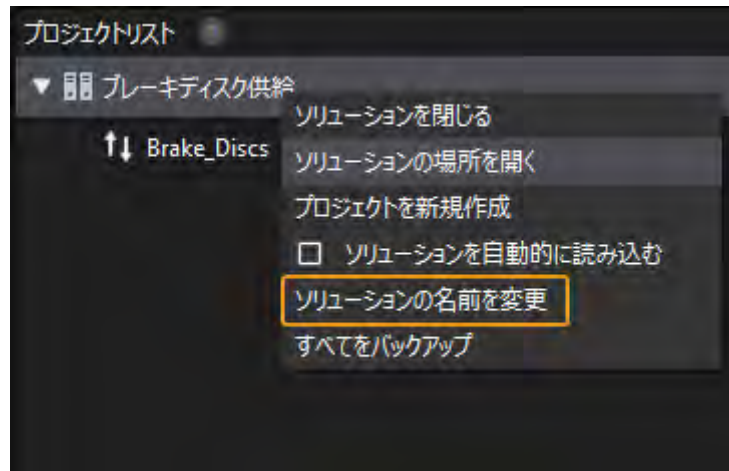


ソリューションを自動的に読み込むにチェックを入れると、ソリューション内のプロジェクトも自動的に「自動読み込み」モードに設定されます。次回ソフトウェアを起動すると、ようこそ画面が表示されることなく、ソフトウェアのメイン画面に入ります。

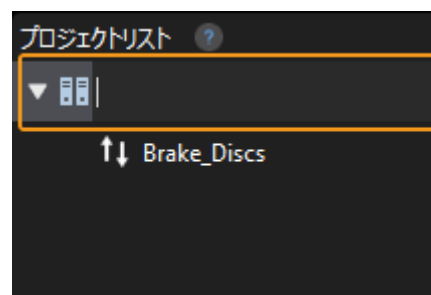
4.2.2.4. ソリューションの名前を変更

本節では、ソリューションの名前を変更する方法について説明します。

ソリューションの名前を変更するには、ソリューション名を右クリックして**ソリューションの名前を変更**をクリックします。



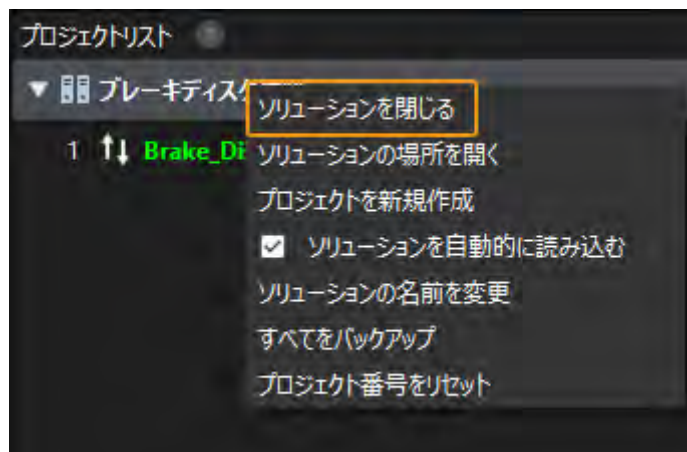
その後、新しい名前を入力して **Enter** キーを押します。



4.2.2.5. ソリューションを閉じる

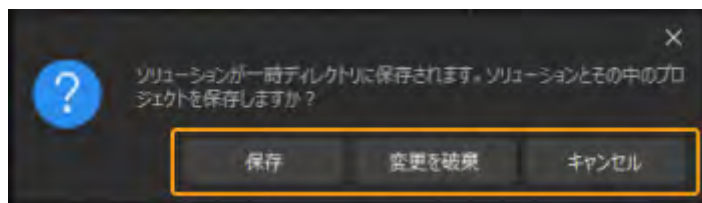
ソリューションを閉じるには、以下のいずれの方法を使用します。

- 下図に示すように、ソリューションを右クリックして**ソリューションを閉じる**をクリックすると閉じられます。



- メニューバーで**ファイル**、**ソリューションを閉じる**を順番に選択します。

ソリューション内のプロジェクトに保存されていない変更がある場合は、下図のようなウィンドウがポップアップするので、実際の状況に応じて選択できます。



ソリューションを閉じると、ソリューション内のすべてのプロジェクトも閉じられます。

4.2.2.6. ソリューションのファイル構造

本節では、ソリューションの構造について説明します。ソリューションには、ビジョンアプリケーションに必要なロボットと通信、ビジョン処理、経路計画などの機能構成やデータのファイルが含まれています。

ファイル構造を下図に示します。



4.2.3. プロジェクトに関する操作

プロジェクトはソリューションを構成するためのMech-Visionビジョンプロジェクトを指します。ソリューションを形成するには、1つまたは複数のプロジェクトが必要です。プロジェクトは単独では使用できず、ソリューションに割り当てて使用する必要があります。

本節では、プロジェクトの基本操作について説明します。

[プロジェクトを新規作成して保存](#)

[プロジェクトの名前を変更](#)

[プロジェクトを開く](#)

[プロジェクトを実行してデバッグ](#)

[プロジェクトを閉じる](#)

[プロジェクトの自動読み込み](#)

[プロジェクト番号を調整](#)

[既存のプロジェクトをソリューションに変換](#)

[プロジェクトのファイル構造](#)

4.2.3.1. プロジェクトを新規作成して保存

本節では、Mech-Vision プロジェクトの新規作成や保存方法、プロジェクトの命名規則について説明します。

プロジェクトを新規作成

ソリューションライブラリから新規作成

詳細については、[ソリューションライブラリからソリューションやプロジェクトを新規作成](#)をご参照ください。

空白プロジェクトを新規作成

空白プロジェクトを作成するには、次のいずれの方法を使用します。

- [ソリューションを新規作成](#) した後、ソリューション内に新しい空白のプロジェクトが自動的に作成されます。
- ソリューションで、メニューバーの **ファイル** > **プロジェクトを新規作成** をクリックして作成します。
- ソリューションで、ショートカットキー **Ctrl** + **N** を押して作成します。




プロジェクトの新規作成は、ソリューションを新規作成した後にのみ可能です。

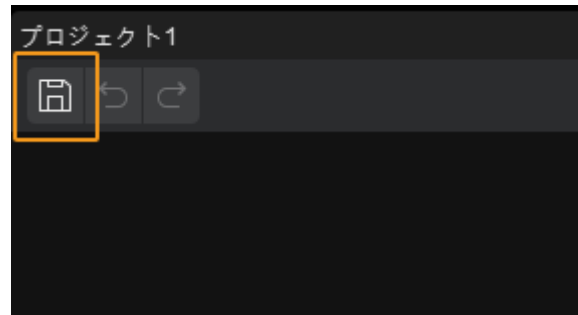
プロジェクトを保存

プロジェクトを保存する際、ソリューション内の単一のプロジェクトのみを保存するか、ソリューション内のすべてのプロジェクトを保存するかを選択することができます。

ソリューション内の単体プロジェクトを保存

ソリューション内の単体プロジェクトを保存するには、次のいずれかの方法を使用します。

- メニューバーで**ファイル**、**プロジェクトを保存**をクリックして保存します。
- プロジェクトツールバーの左側にある  をクリックするか、ショートカットキー **Ctrl** + **S** を押して保存します。



ソリューション内のすべてのプロジェクトを保存

ソリューション内のすべてのプロジェクトを保存するには、メニューバーで**ファイル**、**ソリューションを保存**をクリックします。

プロジェクトの命名規則

詳細な内容を次の表に示します。

名前の構造	説明
Vis	Mech-VisionプロジェクトとMech-Vizプロジェクトを区別するために使用されます（必須）
ワークの名前と使用シーンの組合せ	例：段ボール箱のデパレタイジング・パレタイジング/ブレーキディスクの供給/モーター組立/スポンジ接着剤塗布/ドアロックピッキング
セル情報番号/方位情報	セル情報番号で命名する例：1#探傷1#熱処理1# 方位情報で命名する例：左側のセル/左側のカメラ （このような情報がない場合は入力がなくとも構いません）
異なる機能/特別な機能	異なる機能で命名する例：タイプ1のワーク認識/反射部品の認識/ボックス認識/モデル作成 特別な機能で命名する例：点群結合/両カメラ融合/カメラ補正 （このような情報がない場合は入力がなくとも構いません）

命名の例を次の表に示します。

シーン	例
単一セルと単一カメラの組合せ	Vis-段ボール箱のデパレタイジング・パレタイジング
	Vis-ブレーキディスクの供給
	Vis-モーター組立
	Vis-スポンジ接着剤塗布
	Vis-ドアロックピッキング
単一セルと複数カメラの組合せ	Vis-金属部品の給料-左側のカメラ
複数セルと複数カメラの組合せ	Vis-クランクシャフト供給-探傷1#左側のカメラ
複数セル、複数カメラと異なる機能の組合せ	Vis-クランクシャフト供給-探傷1#左側のカメラ-点群結合
単一セルのプロジェクトの組合せ	Vis-クランクシャフト供給-5#左側のカメラ-タイプ1のワーク認識
	Vis-クランクシャフト供給-5#左側のカメラ-タイプ2のワーク認識
	Vis-クランクシャフト供給-5#左側のカメラ-ボックス認識
	Vis-クランクシャフト供給-5#左側のカメラ-モデル作成
	Vis-クランクシャフト供給-5#右側のカメラ-タイプ1のワーク認識
	Vis-クランクシャフト供給-5#右側のカメラ-タイプ2のワーク認識



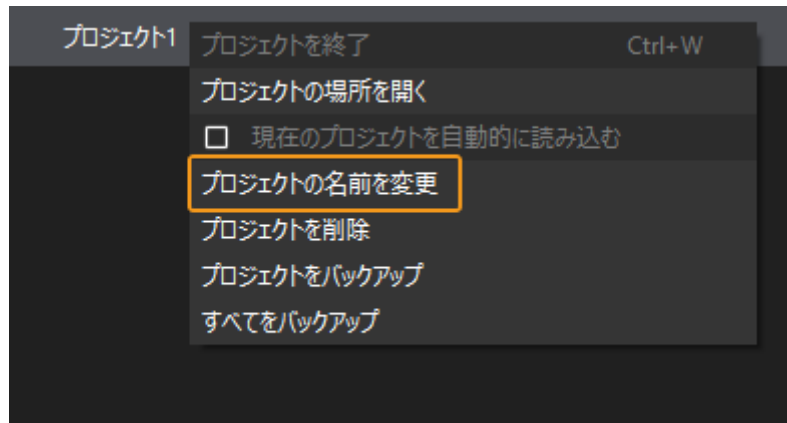
英語でプロジェクトに名前を付けるときは、以下の点にご注意ください。

- 英語の略語を使用することを推奨します。
- ローマ字の使用は禁止されています。
- 複数の英単語を一緒に使用する場合は、各単語をキャメル ケースで区切る必要があります。

4.2.3.2. プロジェクトの名前を変更

プロジェクトの名前を変更するには、以下の操作を実行します。

プロジェクトリストでプロジェクトを右クリックし、[プロジェクトの名前を変更]をクリックし、新しいプロジェクト名を入力して **Enter** を押すとプロジェクトの名前変更が完了します。



上記の操作が完了したら、プロジェクトフォルダとプロジェクトフォルダ内の.visファイルの名前が変更されます。

4.2.3.3. プロジェクトを開く

本節では、Mech-Visionプロジェクトを開く方法について説明します。

既存のプロジェクトを開く

既存のプロジェクトを開くには、次のいずれの方法を使用します。

- ようこそ画面で[プロジェクトを開く]をクリックし、表示されるファイル選択画面でプロジェクトフォルダを選択して開きます。対応するプロジェクトフォルダを選択するだけで、.visファイルに選択する必要はありません。
- メニューバーで**ファイル**、**プロジェクトを開く**をクリックするか、またはショートカットキー **Ctrl + O** を押します。対応するプロジェクトフォルダを選択するだけで、.visファイルに選択する必要はありません。
- プロジェクトフォルダ内の .vis ファイルをダブルクリックすると、Mech-Visionソフトウェアを起動して、プロジェクトを自動的に読み込みます。

最近使ったプロジェクトを開く

最近使ったプロジェクトを開くには、次のいずれの方法を使用します。

- ようこそ画面で右側のソリューション/プロジェクト切り替えボタンを **プロジェクト** に切り替え、プロジェクトをダブルクリックします。



- メニューバーで**ファイル**、**最近使ったファイルを開く**を順番に選択し、プロジェクトをクリックして開きます。



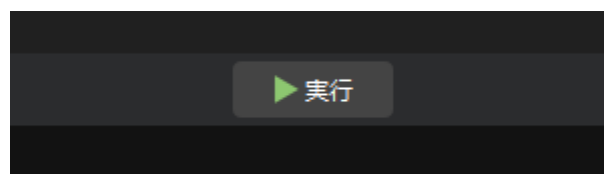
- 1.4.0バージョン以降のMech-Visionで過去バージョンのプロジェクトを開くと、保存後に元の.jsonファイルは自動的に.visファイルに書き換えられ、元のファイルは.bakファイルとしてバックアップされます。
- メニューバーで**ファイル**、**プロジェクトをJSON形式として保存**をクリックして、プロジェクトを.json形式として保存することができます。

4.2.3.4. プロジェクトを実行してデバッグ

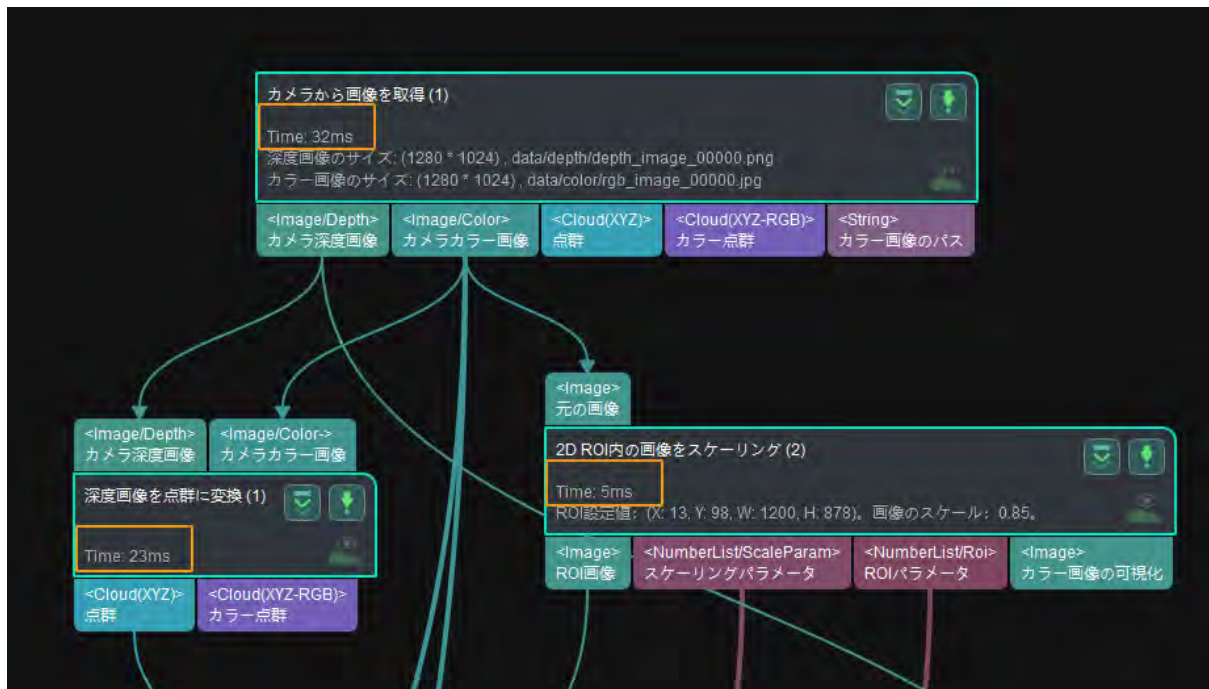


- Mech-Vision v1.4.0でプロジェクトの実行時にバージョン互換性チェックが実行されます。1.4.0バージョン以降のMech-Viz、Mech-Centerと併用することをお薦めします。
- Mech-Viz または Mech-Centerのバージョンが1.4.0よりも低い場合、プロジェクトの実行中にリスクプロンプトウィンドウが表示されます。

下図に示すように、プロジェクトツールバーの「実行」をクリックするか、ショートカット **Ctrl** + **R**を押すと、Mech-Visionプロジェクトが実行されます。



プロジェクトの実行結果が期待結果と一致しない場合、このプロジェクトをデバッグする必要があります。まずはステップを1つずつチェックして異常のステップを見つけます。期待結果と一致しないステップのパラメータを調整することで最適化できます。下図に示すように、実行済みの各ステップの実行時間は画面の左下に表示されます。

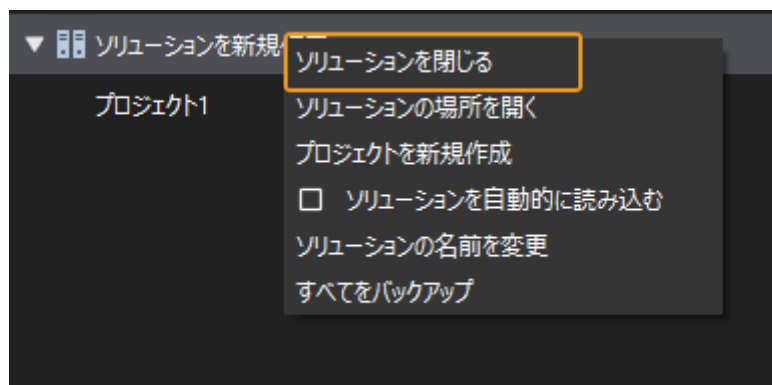


プロジェクトのデバッグについては、[\[vision-operation-guide:run-and-view-outputs::run-and-view-outputs\]](#) をご参照ください。

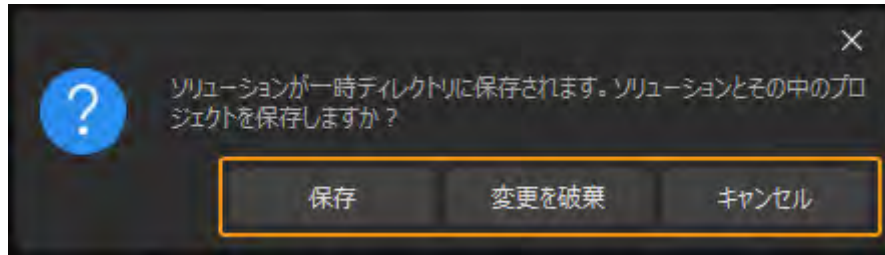
4.2.3.5. プロジェクトを閉じる

ソリューション内の個々のプロジェクトを閉じることはできず、ソリューションを閉じることでその中のすべてのプロジェクトを閉じることができます。詳細は以下の通りです。

下図に示すように、プロジェクトの対応するソリューションを右クリックし、[**ソリューションを閉じる**]をクリックすると、ソリューションとその中のプロジェクトがすべて閉じられます。



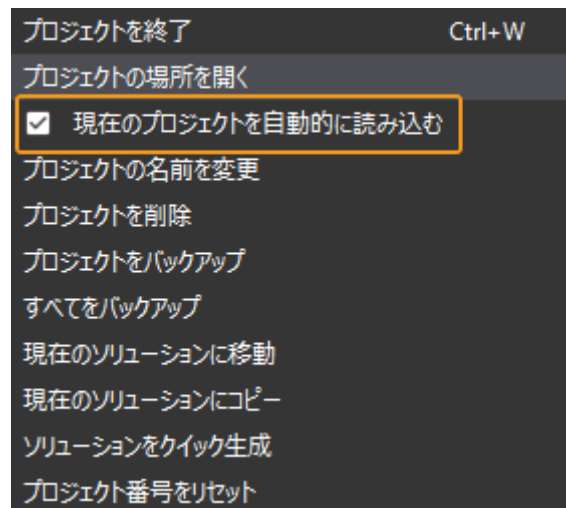
ソリューション内のプロジェクトに保存されていない変更がある場合は、下図のようなウィンドウがポップアップするので、実際の状況に応じて選択できます。



4.2.3.6. プロジェクトの自動読み込み

次回から Mech-Vision を開く時にプロジェクトと一緒に開かれるには、次の操作を実行します。

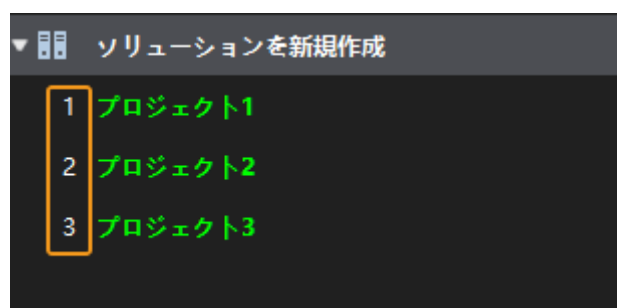
- 下図に示すように、プロジェクトが割り当てられていない場合、プロジェクトリストからプロジェクトを右クリックして、**現在のプロジェクトを自動的に読み込む**にチェックを入れます。



- プロジェクトがソリューションに割り当てられている場合、**ソリューションを自動的に読み込む**にチェックを入れると、ソリューション内のすべてのプロジェクトが「自動読み込みモード」に設定されます。

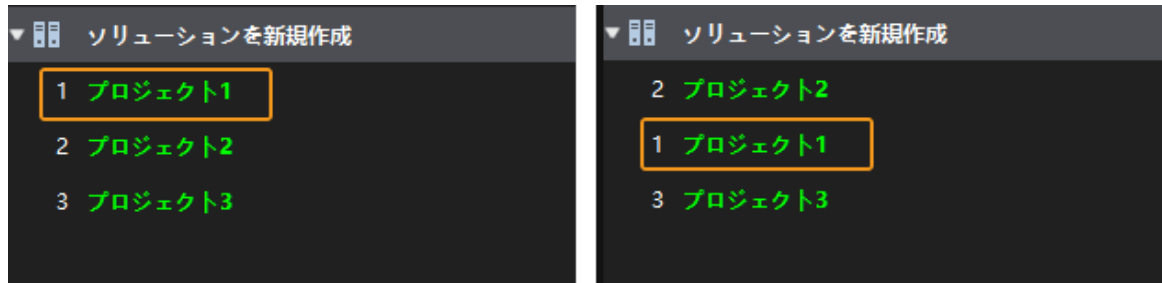
4.2.3.7. プロジェクト番号を調整

下図に示すように、ソリューションが自動読み込みに設定された場合、ソリューション内のプロジェクトも自動読み込みに設定されます。また、各プロジェクトにプロジェクト番号が割り当てられます。



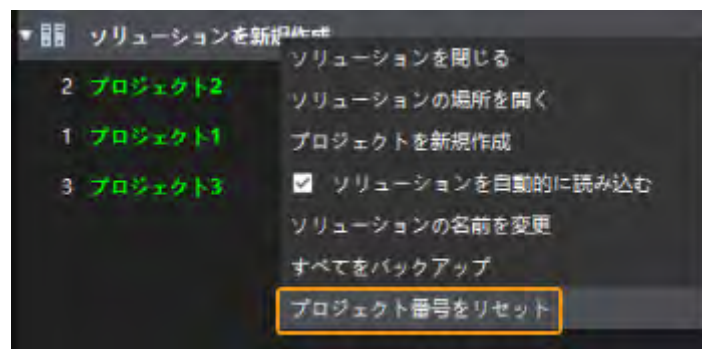
プロジェクト番号を調整するには、次のいずれの方法を使用します。

1. プロジェクトをドラッグして調整します。プロジェクトを選択し、上下にドラッグしてプロジェクトの順番を調整します。下図に示すように、左側が調整前、右側が調整後の効果です。

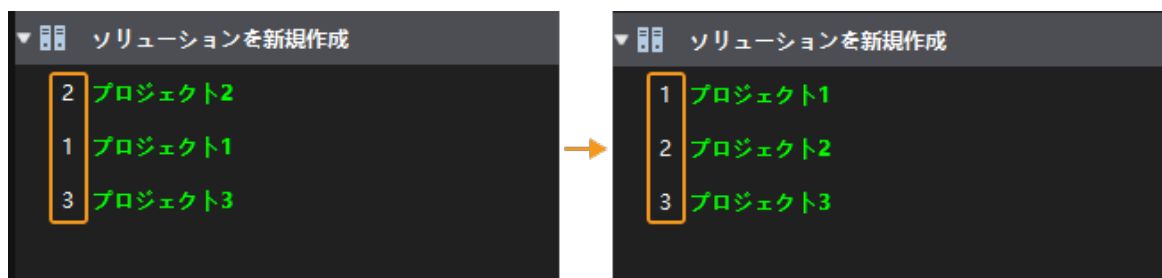


この場合では、プロジェクト番号は変更されておらず、プロジェクトリストにおけるプロジェクトの位置だけが調整されています。そのため、「プロジェクト番号をリセット」する必要があります。

2. プロジェクト番号をリセットします。ソリューションまたはプロジェクトを右クリックして、[プロジェクト番号をリセット]をクリックします。



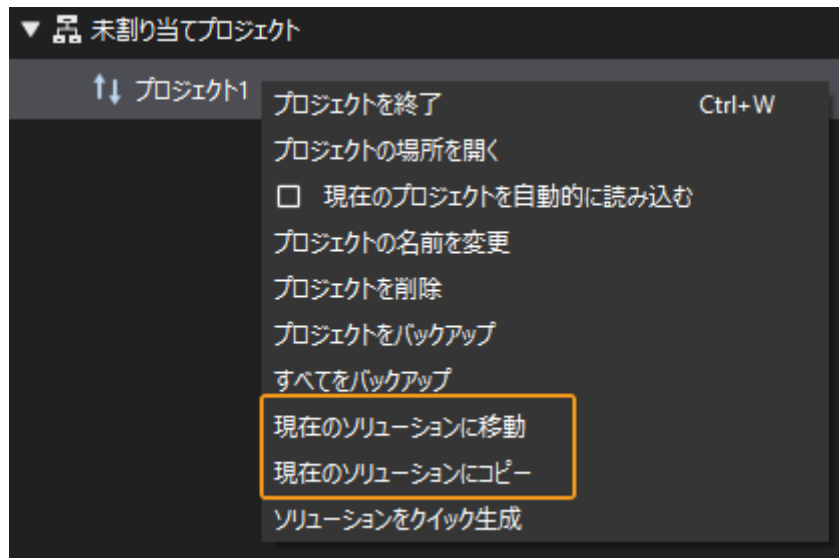
下図に示すように、左がリセット前、右がリセット後の効果です。



4.2.3.8. 既存のプロジェクトをソリューションに変換

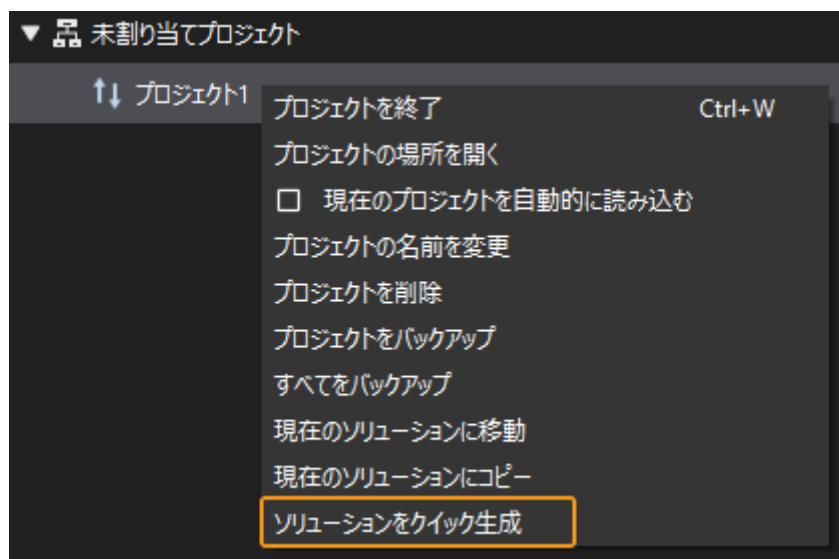
既存のプロジェクトをソリューションに変換するには、次の方法を使用します。

- 下図に示すように、ソリューションを新規作成して保存した後、プロジェクトリストで未割り当てのプロジェクトを選択し、右クリックして[現在のソリューションに移動]または[現在のソリューションにコピー]にチェックを入れます。



変換が完了したプロジェクトで、ステップパラメータに絶対パスや相対パスを使用している場合は、変換後にプロジェクトを検証し、変換によって発生した不正なパスを修正する必要があります。

- プロジェクトリストから未割り当てのプロジェクトを選択し、右クリックして[**ソリューションをクイック生成**]を選択し、ポップアップウィンドウの[**はい**]をクリックするとプロジェクトがソリューションに変換され、プロジェクトの親フォルダがソリューションフォルダとして使用されるようになります。



プロジェクトを新しいソリューションに変換する場合、プロジェクトの親ディレクトリをディスクのルートディレクトリにすることはできません。

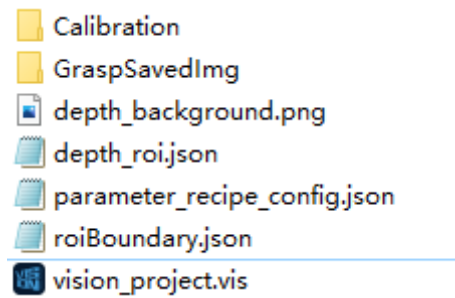
4.2.3.9. プロジェクトのファイル構造

Mech-Visionプロジェクトのファイル構造は下図に示します。これは、主に以下のファイルで構成されています。

アルゴリズムフローファイル (vision_project.vis) : アルゴリズム処理フローを保存する

- ために使用されます。
- カメラパラメータファイル（Calibration）：カメラの内部・外部パラメータやキャリブレーションデータを保存するために使用されます。
- プロジェクトのコンフィグファイル
（depth_background.pn、depth_roi.json、roiBoundary.jsonなど）：プロジェクト設定ファイルには作業環境の前提条件が記載され、プロジェクトに制限をかけるために使用されます。

Mech-Visionプロジェクトファイルは、対応する操作を実行した後、対応するプロジェクトのディレクトリに自動的に生成されるので、手動で作成する必要はありません。



4.2.4. ロボット通信設定に関する操作

本節では、ロボット通信設定の操作方法について説明します。


設定を行う前に、[通信方式の概要](#)を参照して通信方式と通信プロトコルを確認してください。

- 標準インターフェース通信またはAdapter通信の場合、本節をお読みください。
 - [ロボットの選択](#)
 - 通信設定
 - [TCP/IP](#)
 - [Siemens PLC Snap 7](#)
 - [PROFINET](#)
 - [EtherNet/IP](#)
 - [Modbus TCP](#)
 - [Adapter通信設定](#)
 - [インターフェースサービスを起動](#)
- Vizティーチングの場合、[Vizティーチング通信](#)をお読みください。

4.2.4.1. Adapter通信設定

本節では、Adapterの通信設定について説明します。その前に、[ロボットを選択](#)し、通信設定画面に入っていることを確認します。



1. 通信方式で、インターフェースタイプを**Adapter**に設定します。
2.  をクリックし、Adapterプロジェクトフォルダを選択します。



ユーザは事前にAdapterプログラムを作成する必要があります。詳細については、[Adapterプログラミングガイド](#) をご参照ください。

3. IPアドレスでホストIPアドレスとポート番号を設定します。



- ホストIPは、ロボットIPと同じネットワークセグメントにある必要があります。
- ホストポート番号を50000以上に設定することを推奨します。

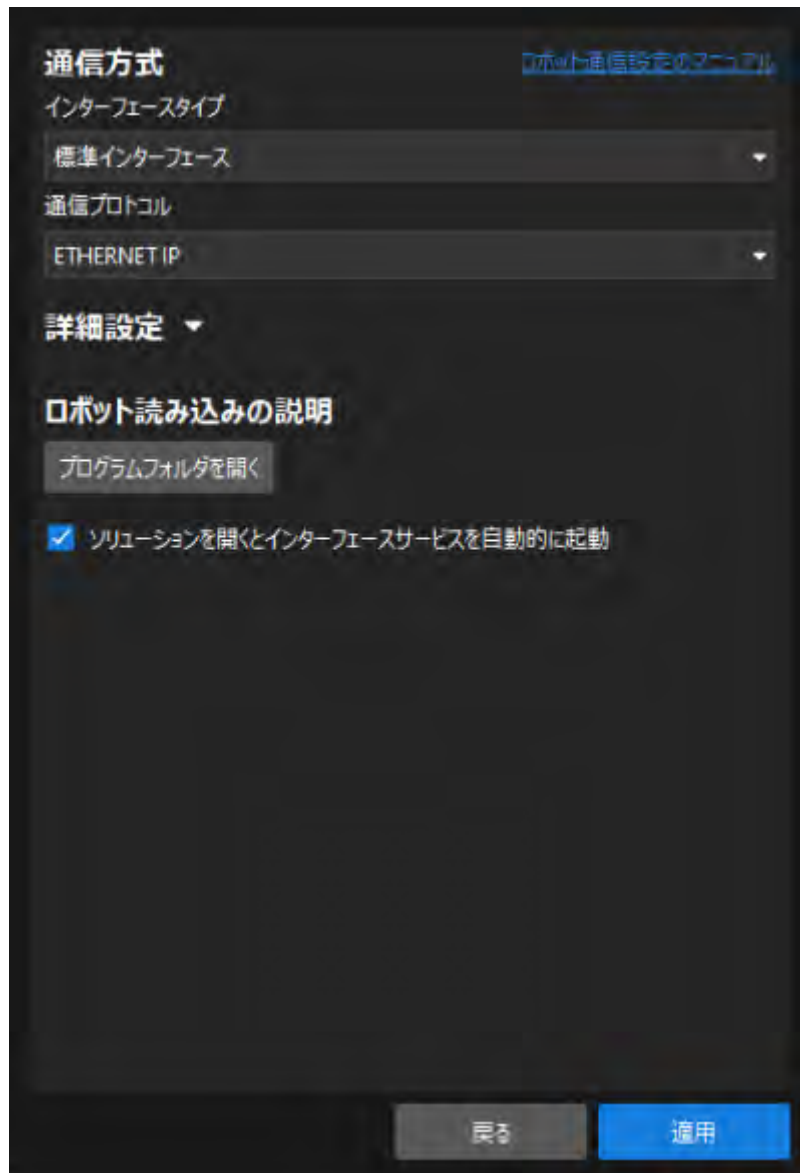
4. ソリューションを開くとインターフェースサービスを自動的に起動オプションを選択可能です。チェックを入れると、次回ソリューションを開く時に [\[vision-operation-guide:start-communication:::start-communication\]](#) する必要はありません。
5. [適用]をクリックします。

6. [\[vision-operation-guide:start-communication:::start-communication\]](#) します。

その後、選択されたロボットに応じてロボット側のプログラムを作成し、プログラムの読み込みを実行します。詳細については、[標準インターフェース開発者向けマニュアル](#)をご参照ください。

4.2.4.2. EtherNet/IP

本節では、EtherNet/IPの通信設定について説明します。その前に、[ロボットを選択](#)し、通信設定画面に入っていることを確認します。



1. **通信方式**で、インターフェースタイプを**標準インターフェース**に、通信プロトコルを**ETHERNET IP**に設定します。
2. **ソリューションを開くとインターフェースサービスを自動的に起動**オプションを選択可能です。チェックを入れると、次回ソリューションを開く時に [\[vision-operation-guide:start-communication:::start-communication\]](#) する必要はありません。

3. [適用]をクリックします。

4. [vision-operation-guide:start-communication::start-communication] します。

選択した通信先に応じて、標準インターフェースの設定を行います。

4.2.4.3. Modbus TCP

本節では、Modbus TCPの通信設定について説明します。その前に、ロボットを選択し、通信設定画面に入っていることを確認します。



1. 通信方式で、インターフェースタイプを標準インターフェースに、通信プロトコルをMODBUS TCP SLAVEに設定します。

2. IPアドレスで、次の設定を行います。

- a. ホスト（スレーブ）ポートを記入します。ポート番号を50000以上に設定することを推奨します。

- b. バイトオーダーを選択します。マスターの浮動小数点のバイトオーダーに基づき、標準的なビッグエンディアンデータの場合はDCBA、標準的なリトルエンディアンデータの場合はABCDを選択してください。
 - c. スレーブのアドレスを選択します。
3. ソリューションを開くとインターフェースサービスを自動的に起動オプションを選択可能です。チェックを入れると、次回ソリューションを開く時に [\[vision-operation-guide:start-communication:::start-communication\]](#) する必要はありません。
4. [適用]をクリックします。
5. [\[vision-operation-guide:start-communication:::start-communication\]](#) します。

選択した通信先に応じて、[標準インターフェースの設定](#)を行います。

4.2.4.4. PROFINET

本節では、PROFINETの通信設定について説明します。その前に、[ロボットを選択](#)し、通信設定画面に入っていることを確認します。



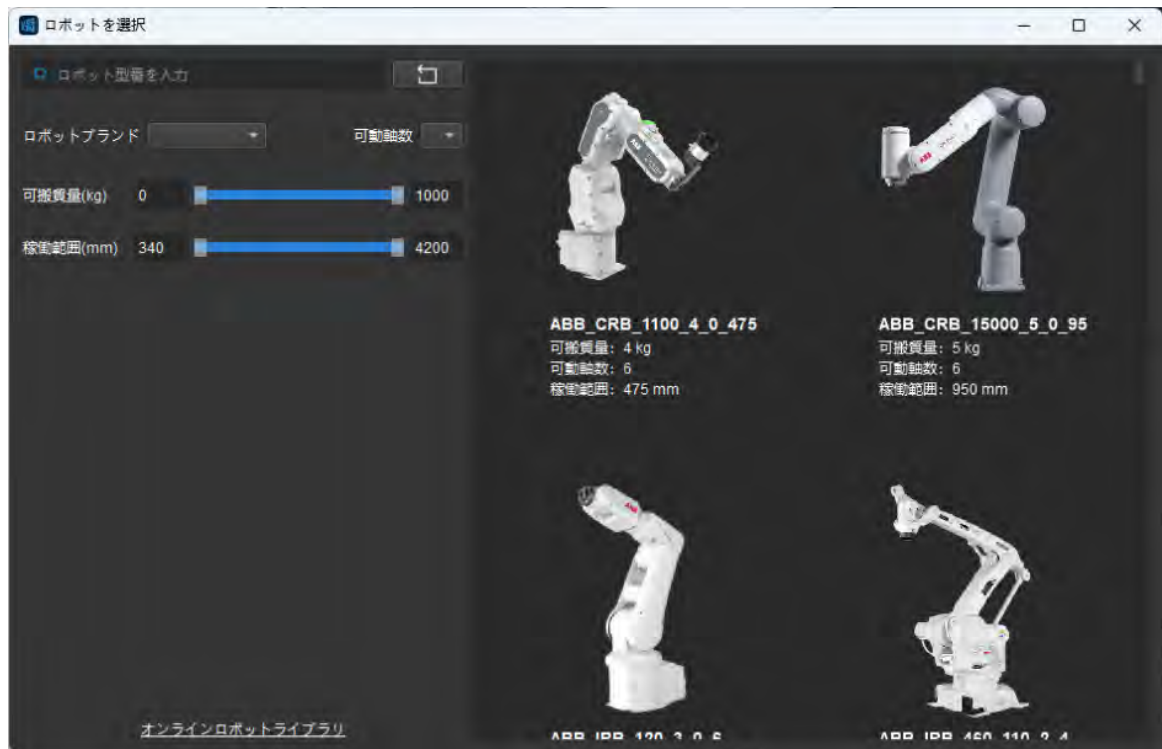
1. 通信方式で、インターフェースタイプを標準インターフェースに、通信プロトコルをPROFINET-IRTに設定します。
2. ソリューションを開くとインターフェースサービスを自動的に起動オプションを選択可能です。チェックを入れると、次回ソリューションを開く時に [\[vision-operation-guide:start-communication:::start-communication\]](#) する必要はありません。
3. [適用]をクリックします。
4. [\[vision-operation-guide:start-communication:::start-communication\]](#) します。

選択した通信先に応じて、 [標準インターフェースの設定](#) を行います。

4.2.4.5. ロボット型番を選択

本節では、ロボット型番の選択方法について説明します。

1. [ロボットを選択](#) の画面を下図に示します。



2. **ロボットブランド** のドロップダウンボックスで、ロボットブランドを選択すると、そのブランドのすべてのロボットが右側に表示されます。



- **可動軸数、可搬質量、稼働範囲**を選択してロボットを検索できます。
- **検索ボックス**にロボットブランドと型番（英語は大文字小文字を区別しません）を入力し、ロボットも検索することができます。🔍をクリックすると、入力ボックスがクリアされます。



ロボットライブラリに必要なロボットがない場合があります。この時、[ロボットをインポート](#)してから、ここで操作を行います。

3. 右側の選択するロボットにカーソルを合わせると、オフホワイト色の枠と[**選択**]ボタンが表示されますので、それをクリックします。

4.2.4.6. Siemens PLC Snap 7

本節では、Siemens PLC Snap 7の通信設定について説明します。その前に、[ロボットを選択](#)し、通信設定画面に入っていることを確認します。



1. 通信方式で、インターフェースタイプを標準インターフェースに設定し、通信プロトコルをSiemens PLC Clientに設定します。
2. IPアドレスで、次の設定を行います。
 - a. ホストIPアドレスとポート番号を設定します。PLCのIPは、IPCのIPアドレスと同じネットワークセグメントにある必要があります。
 - b. 下表を参照してスロット番号を選択します。

Siemens S7シリーズ機種	スロット番号
S7-300	2
S7-1200	0または1
S7-1500	0または1

Siemens S7シリーズ機種	スロット番号
その他	0

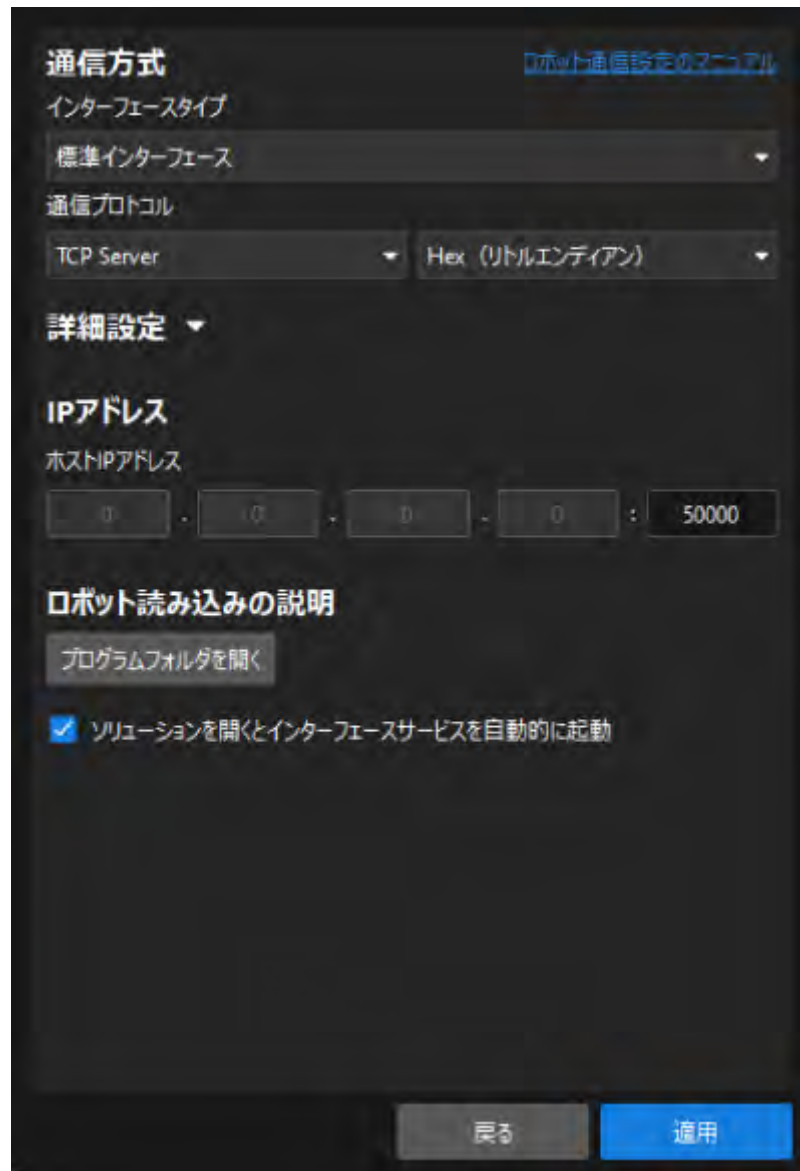
a. DBブロック番号を選択します。

1. ソリューションを開くとインターフェースサービスを自動的に起動オプションを選択可能です。チェックを入れると、次回ソリューションを開く時に [\[vision-operation-guide:start-communication::start-communication\]](#) する必要はありません。
2. [適用]をクリックします。
3. [\[vision-operation-guide:start-communication::start-communication\]](#) します。

選択した通信先に応じて、[標準インターフェースの設定](#)を行います。

4.2.4.7. TCP/IP

本節では、TCP/IPの通信設定について説明します。その前に、[ロボットを選択](#)し、通信設定画面に入っていることを確認します。



1. 通信方式で、インターフェースタイプを標準インターフェースに、通信プロトコルをTCP Serverに設定します。通信プロトコルの形式は、下表に従って選択します。

ロボット	通信プロトコル
ABB	HEX (リトルエンディアン)
FANUC	HEX (ビッグエンディアン)
KUKA	HEX (リトルエンディアン)
Yaskawa	ASCII
Kawasaki	ASCII
UR	ASCII
TM	ASCII

ロボット	通信プロトコル
ELITE	ASCII
JAKA	ASCII
ROKAE	ASCII
NACHI	ASCII
その他	ユーザーはロボット側の通信プログラムを作成する必要があるので、ロボットが対応しているHEX、ASCIIに応じてプロトコル形式を決定することができます。

2. IPアドレスでホストポート番号を設定します。



ホストポート番号を50000以上に設定することを推奨します。

3. ソリューションを開くとインターフェースサービスを自動的に起動オプションを選択可能です。チェックを入れると、次回ソリューションを開く時に [\[vision-operation-guide:start-communication:::start-communication\]](#) する必要はありません。

4. [適用]をクリックします。

5. [\[vision-operation-guide:start-communication:::start-communication\]](#) します。

その後、選択したロボットに応じて以下の操作を実行します。

- 選択されたロボットが以下のブランドの場合、対応するマニュアルを参照して操作してください。

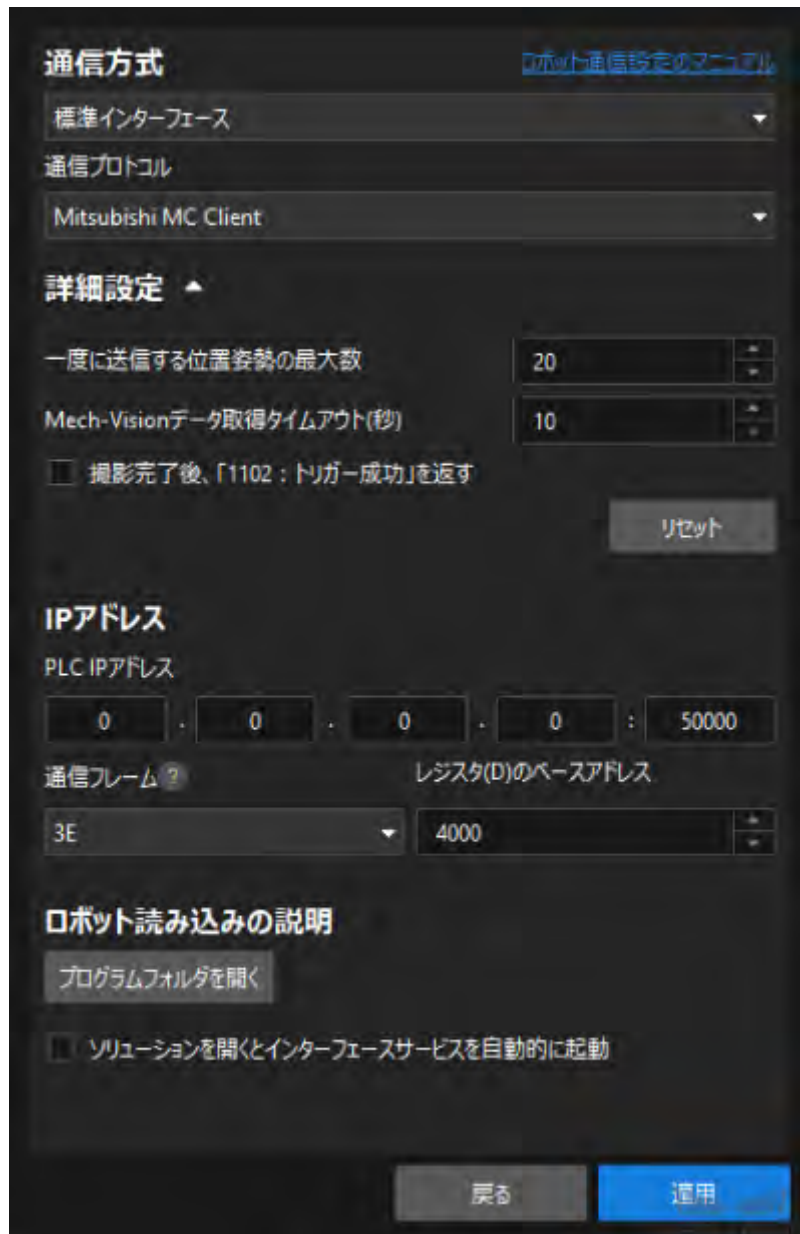
ロボット	マニュアル
ABB	ABB 標準インターフェースの通信設定
FANUC	FANUC 標準インターフェースの通信設定
FANUC CRX	英語版をご参照ください
Kawasaki	Kawasaki 標準インターフェースの通信設定
KUKA	KUKA 標準インターフェースの通信設定
YASKAWA	YASKAWA 標準インターフェースの通信設定
UR	英語版をご参照ください
TM	英語版をご参照ください
ELITE	英語版をご参照ください

ロボット	マニュアル
JAKA	英語版をご参照ください

- 選択したロボットが上記以外のブランドの場合、[標準インターフェース開発者向けマニュアル](#)を参照してロボット側の通信プログラムを作成します。キャリブレーションについては、対応するロボットの[Vizとの通信プログラムの読み込み](#)を参照し、[自動キャリブレーション](#)を完成させてください。
- 選択したロボットがそのほかのロボットである場合、[標準インターフェース開発者向けマニュアル](#)を参照してロボット側の通信プログラムを作成します。キャリブレーションについては、[手動キャリブレーション](#)を参照して実行してください。

4.2.4.8. Mitsubishi MC

本節では、Mitsubishi MCの通信設定について説明します。その前に、[ロボットを選択](#)し、通信設定画面に入っていることを確認します。



1. **通信方式** で、インターフェースタイプを **標準インターフェース** に設定し、通信プロトコルを **Mitsubishi MC Client** に設定します。
2. **IPアドレス** で、次の設定を行います。
 - a. **PLC IPアドレス** には、PLCのIPアドレスを記入し、ポート番号は50000以上に設定することを推奨します。
 - b. **通信フレーム** では、通常は3Eを選択します。
 - c. レジスタ(D)のベースアドレスを設定します。



通信に使用される構造タイプは、合計728本のDレジスタを占有します。

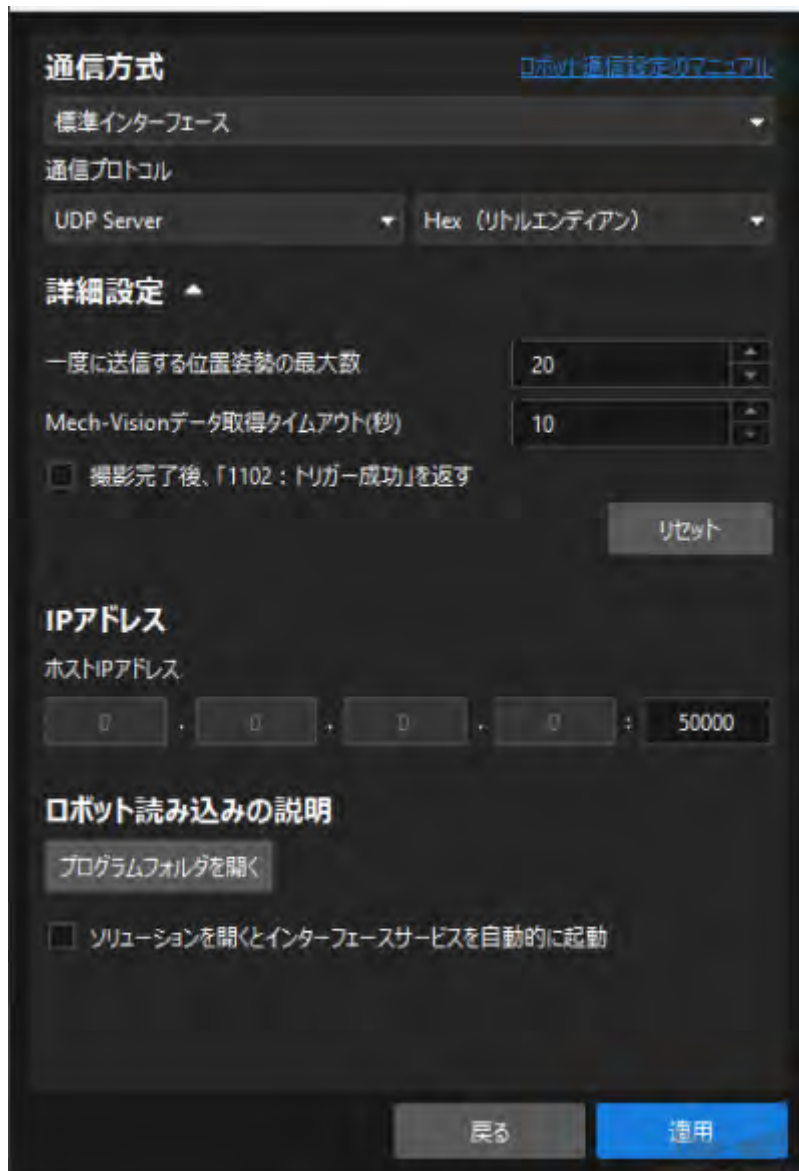
3. **ソリューションを開くとインターフェースサービスを自動的に起動** オプションを選択可能です。チェックを入れると、次回ソリューションを開く時に **インターフェースサービスを起動** する必要はありません。

4. [適用]をクリックします。
5. インターフェースサービスを起動 します。

選択した通信先に応じて、標準インターフェースの設定を行います。

4.2.4.9. UDP

本節では、UDPの通信設定について説明します。その前に、ロボットを選択 し、通信設定画面に入っていることを確認します。



1. 通信方式 で、インターフェースタイプを 標準インターフェース に設定し、通信プロトコルを UDP Server に設定します。通信プロトコル形式は、ロボットのHEX、ASCIIの対応状況に応じて選択できます。
2. IPアドレスでホストポート番号を設定します。



ホストポート番号を50000以上に設定することを推奨します。

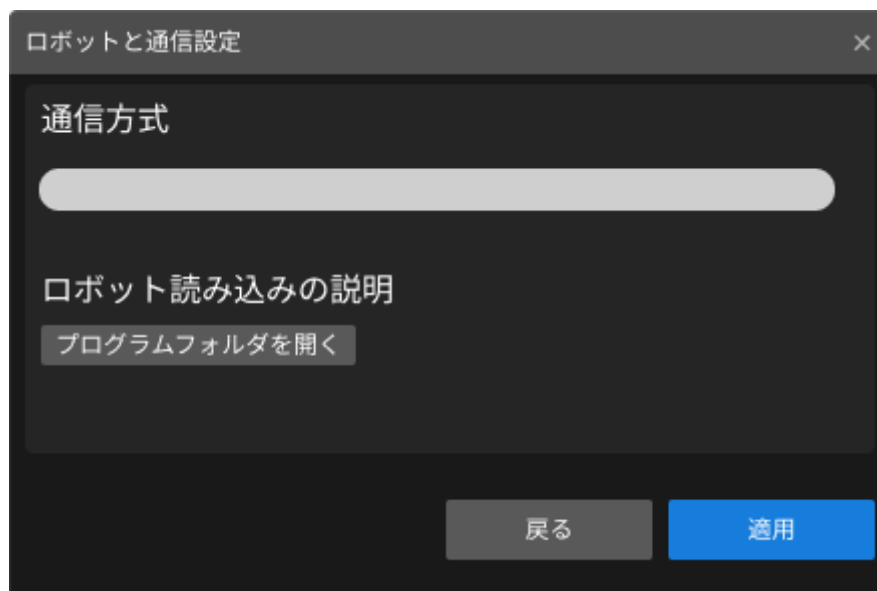
3. ソリューションを開くとインターフェースサービスを自動的に起動オプションを選択可能です。チェックを入れると、次回ソリューションを開く時に **インターフェースサービスを起動** する必要はありません。
4. [適用] をクリックします。
5. **インターフェースサービスを起動** します。

選択した通信先に応じて、**標準インターフェースの設定** を行います。

4.2.4.10. 読み込みファイルを取得

本節では、読み込みファイルの取得方法について説明します。

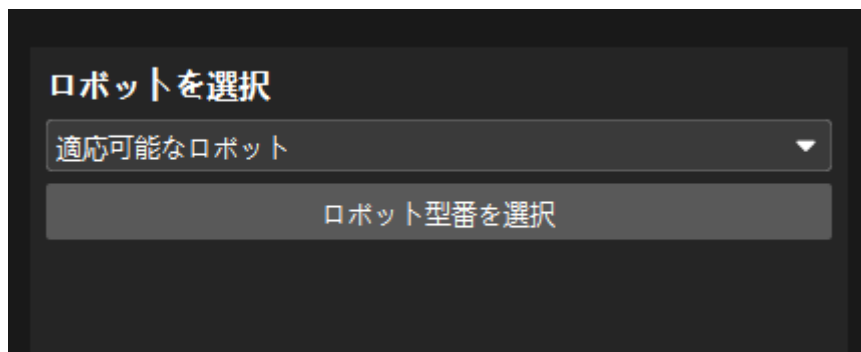
ロボットを選択 して通信画面に入った後、[**プログラムフォルダを開く**] をクリックすると、対応するロボットの通信プログラムとサンプルプロジェクトが格納されているフォルダが表示されます。



4.2.4.11. ロボットの選択

本節では、ロボットの選択方法について説明します。

1. Mech-Visionのツールバーで[**ロボット通信設定**] をクリックします。
2. ロボットを選択します。
 - 適応可能なロボットを選択
 - **ロボットを選択** のドロップダウンボックスをクリックし、**適応可能なロボット** を選択して、[**ロボット型番を選択**] をクリックします。



- **ロボットブランド**のドロップダウンボックスで、ロボットブランドを選択すると、そのブランドのすべてのロボットが右側に表示されます。

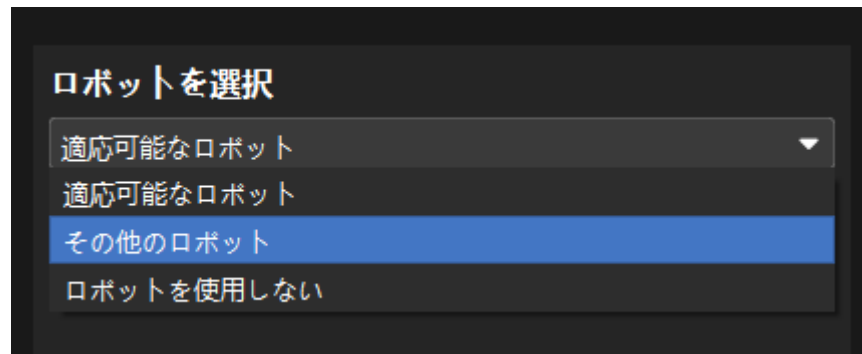


- **可動軸数、可搬質量、稼働範囲**を選択してロボットを検索できます。
- **検索ボックス**にロボットブランドと型番（英語は大文字小文字を区別しません）を入力し、ロボットも検索することができます。🔍をクリックすると、入力ボックスがクリアされます。

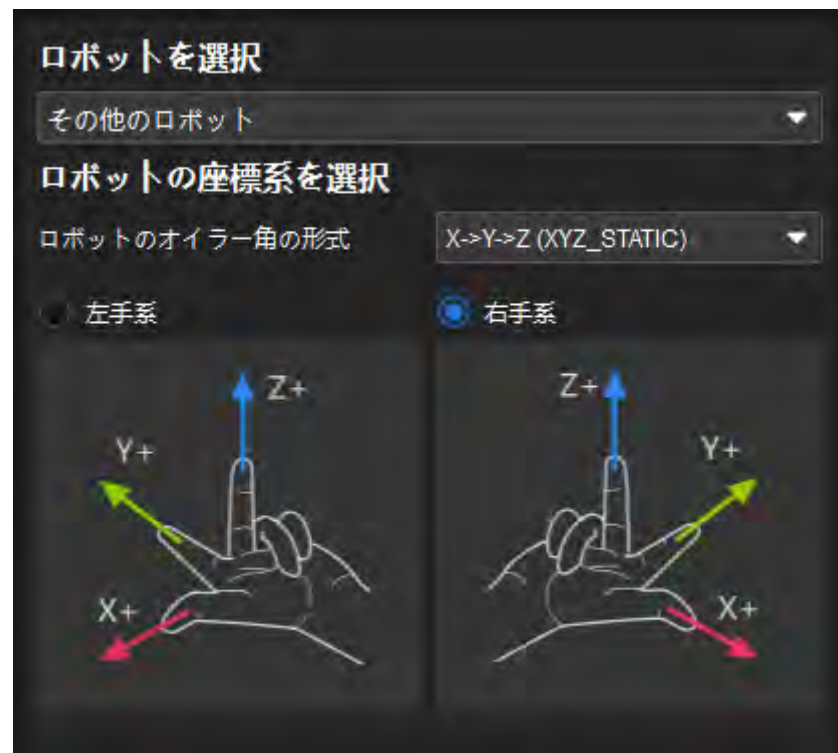


ロボットライブラリに必要なロボットがない場合があります。この時、**ロボットをインポート**してから、ここで操作を行います。

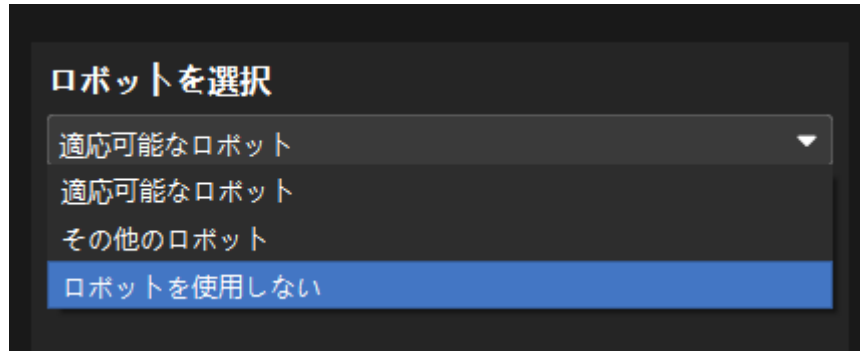
- 右側の選択するロボットにカーソルを合わせると、オフホワイト色の枠と[**選択**]ボタンが表示されますので、それをクリックします。
- [**次へ**]をクリックし、「通信方式の設定」画面に入ります。
- その他のロボットを選択（直行ロボットなど）
 - **ロボットを選択**のドロップダウンボックスをクリックし、**その他のロボット**を選択します。



- ロボットのオイラー角の形式を選択します。
- ロボットの座標系を選択します。



- [次へ]をクリックし、「通信方式の設定」画面に入ります。
- ロボットを使用しない（計測など、ロボットが必要としない場合に対応）
 - **ロボットを選択**のドロップダウンボックスをクリックし、**ロボットを使用しない**を選択します。



- [次へ]をクリックし、「Adapter通信設定」の画面に入ります。

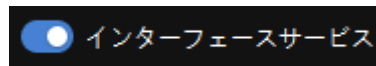


ロボットを使用しないを選択した場合、通信方式はAdapter通信のみ選択可能です。

4.2.4.12. インターフェースサービスを起動

本節では、インターフェースサービスの起動方法について説明します。

Mech-Visionでツールバーの右側にある「インターフェースサービス」のスイッチをクリックし、下図のような状態であればインターフェースサービスが起動されます。



正常に接続すると、Mech-VisionのログのConsoleタブに「XXXXインターフェースサービスを起動します」が表示されます。

起動に失敗した場合、ロボットと通信設定が正しいかどうか確認してください。

4.2.4.13. ロボットライブラリツールに関する操作

本節では、ロボットライブラリツールの操作方法について説明します。

ロボットをインポート

ロボットをインポートする前に、[オンラインロボットライブラリ](#) から必要なロボットモデルファイル（拡張子は.mrob）をダウンロードする必要があります。

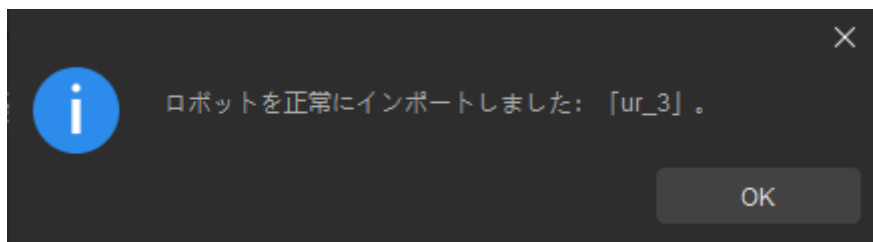


オンラインロボットライブラリで必要なロボットがない場合は、ロボットモデルを作成してMech-Visionにインポートする必要があります。

ロボットをインポートするには、次のいずれの方法を使用します。

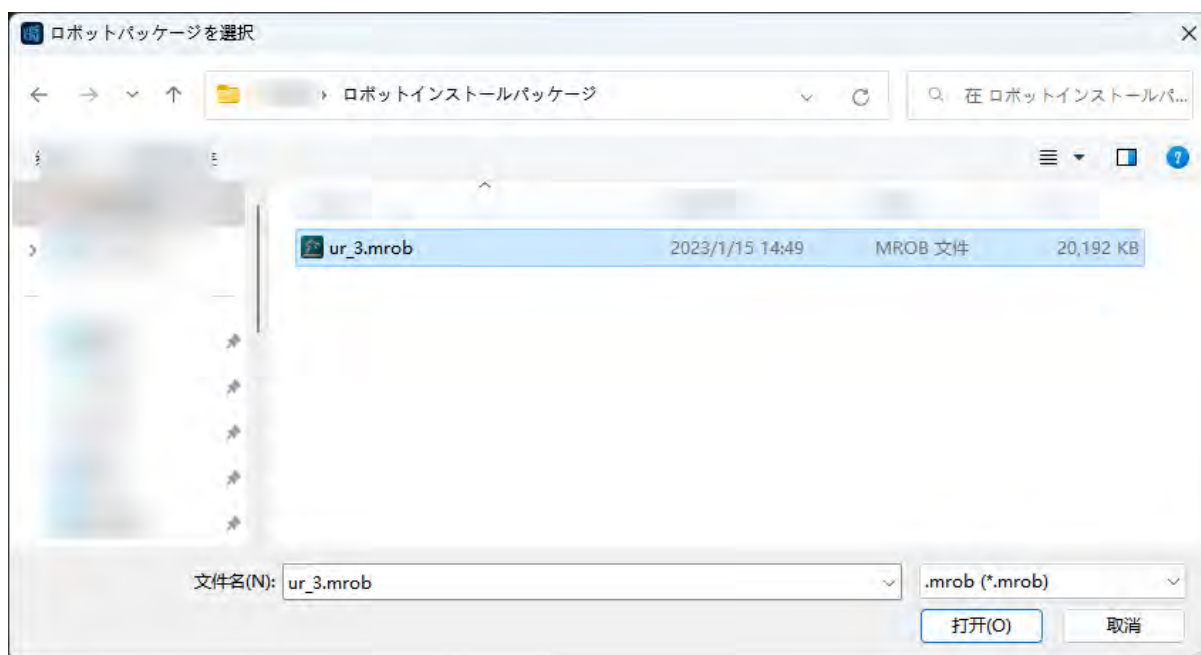
● 方法1

- ダウンロードした.mrobファイルをMech-Visionソフトウェアウィンドウにドラッグし、マウスの左ボタンを離します。
- すると、「ロボットを正常にインポートしました」というメッセージが表示され、[OK]をクリックします。

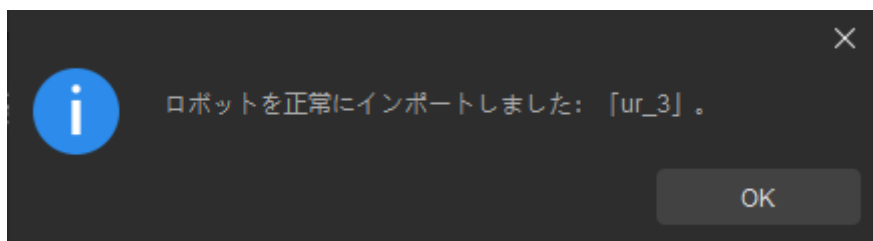


● 方法2

- Mech-Visionのメニューバーで**ロボットと通信**、**ロボットライブラリツール**、**ロボットをインポート**を順番にクリックします。
- すると、ファイル選択画面が表示され、ダウンロードした.mrobファイルを選択して[開く]をクリックします。



- すると、「ロボットを正常にインポートしました」というメッセージが表示され、[OK]をクリックします。

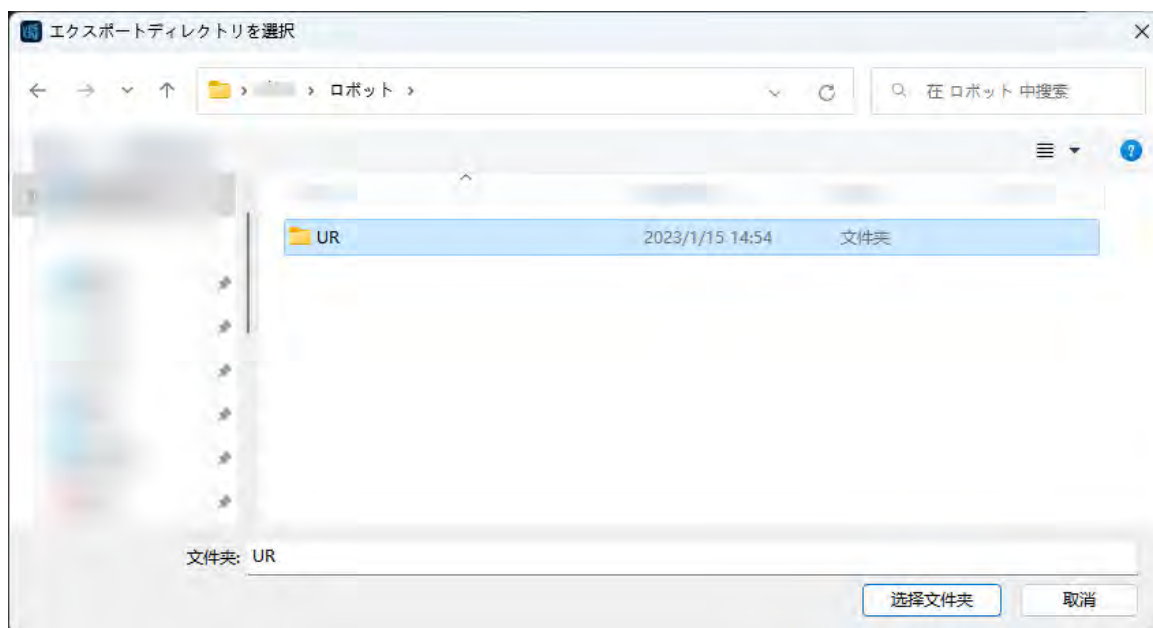


現在のロボットをエクスポート

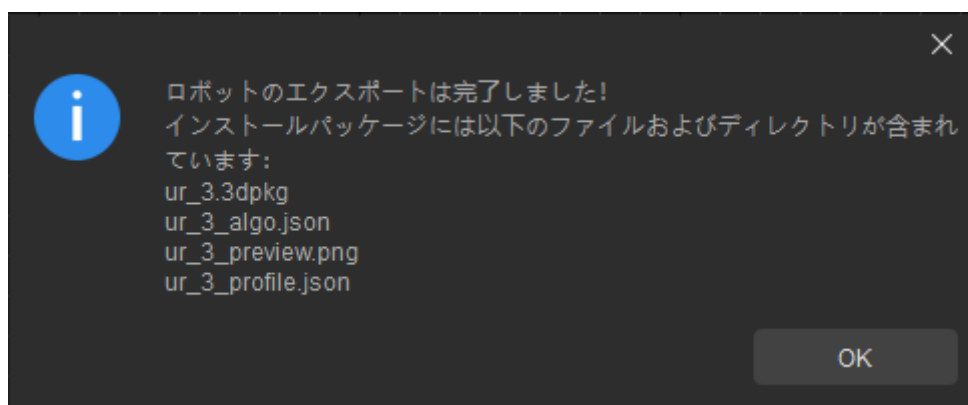
現在のロボットをエクスポートするには、次の方法を使用します。

1. Mech-Visionのメニューバーで**ロボットと通信**、**ロボットライブラリツール**、**現在のロボットをエクスポート**を順番にクリックします。

- 表示される画面で、エクスポートするフォルダを選択して[OK]をクリックします。



- すると、「ロボットのエクスポートは完了しました」というメッセージが表示され、[OK]をクリックします。

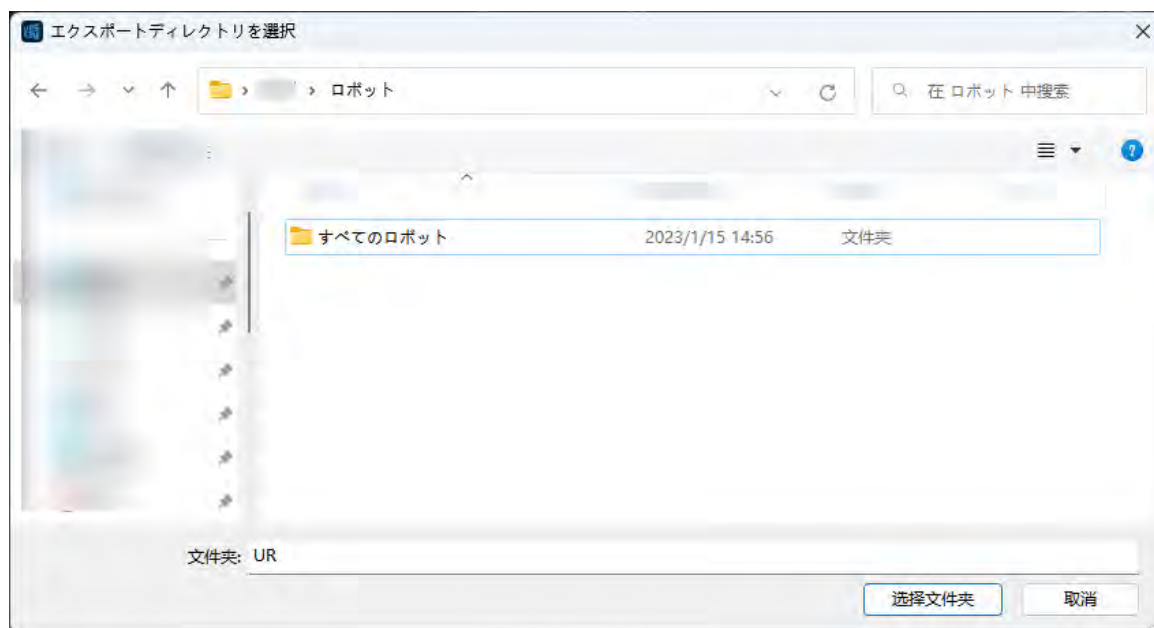


エクスポートが完了すると、エクスポートしたフォルダに.mrobファイルが表示され、これがエクスポートされたロボットモデルです。

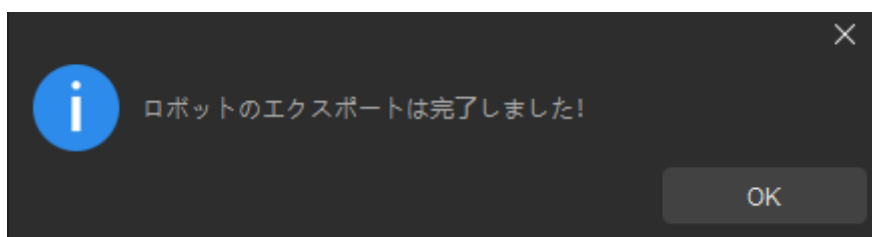
すべてのロボットをエクスポート

Mech-Visionですべてのロボットをエクスポートするには、次の方法を使用します。

- Mech-Visionのメニューバーで**ロボットと通信 > ロボットライブラリツール > すべてのロボットをエクスポート**を順番にクリックします。
- 表示される画面で、エクスポートするフォルダを選択して[OK]をクリックします。



- すると、「ロボットのエクスポートは完了しました」というメッセージが表示され、[OK]をクリックします。



エクスポートが完了すると、エクスポートしたフォルダに複数の.mrobファイルが表示され、これらがエクスポートされたロボットモデルです。

4.2.5. ステップに関する操作

本節では、ステップのよく使われる操作について説明します。

[ステップを検索・追加・削除](#)

[ステップ間のつながぎを作成・削除](#)

[ステップの入力・出力を表示](#)

[ステップの異なる状態を表示](#)

[ステップを実行して可視化出力結果を表示](#)

[ステップのコメントを追加・変更・表示](#)

ステップの組合せは、複数ステップの集りです。異なる項目やプロジェクトでは、アルゴリズム

ム処理が一致していたり、類似していることが多く、これらの固定的なアルゴリズム処理ステップを組み合わせ、パッケージ化することで、簡単に再利用することが可能です。

ステップの組合せに関する操作は、以下の内容をお読みください。

[ステップの組合せの基本操作](#)

[ステップの組合せのパラメータを変更](#)

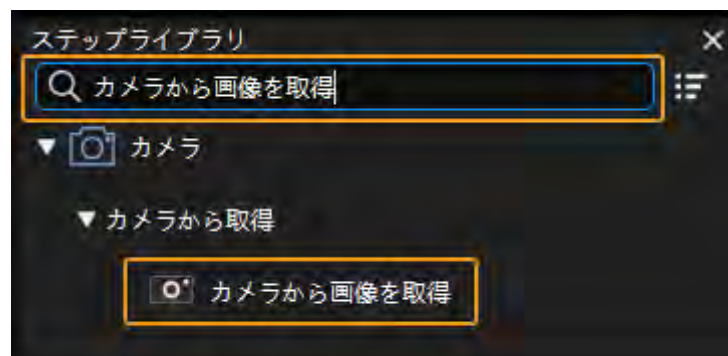
[ステップの組合せをカスタマイズ](#)

4.2.5.1. ステップを検索・追加・削除

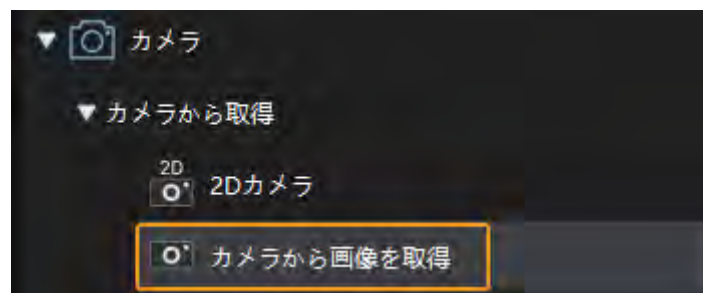
以下は、ステップ「カメラから画像を取得」を例として説明します。

ステップの検索方法

- ステップライブラリの検索ボックスにステップ名（キーワードに対応）を入力して検索します。



- またはステップのタイプで検索します。



ステップの追加方法

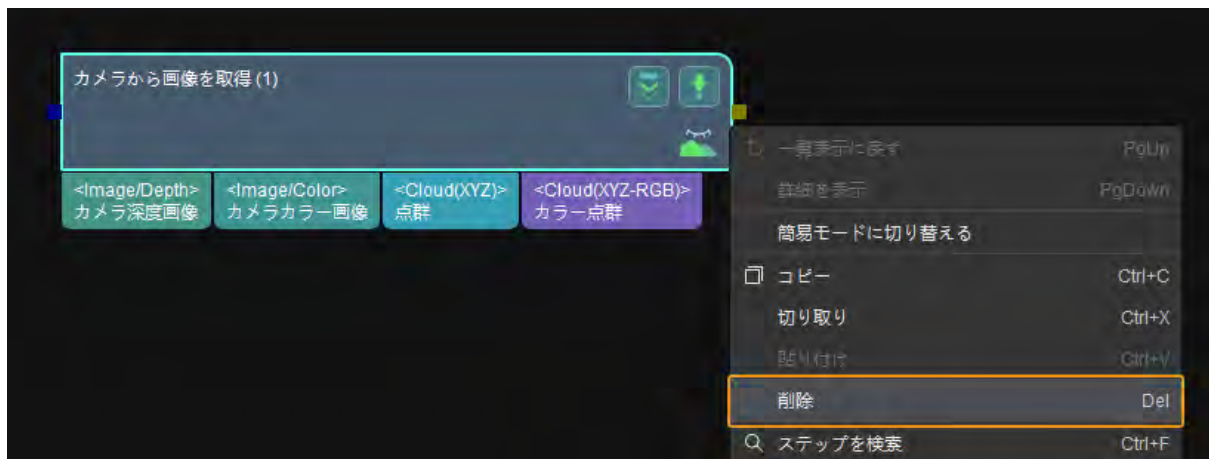
必要なステップを選択してマウスの左ボタンを押したまま、プロジェクト編集エリアの任意場所にドラッグして放します。



プロジェクトが新規作成された場合、または既存のプロジェクトが開かれた場合のみにステップを追加することができます。

ステップの削除方法

- マウスの左ボタンで選択したステップをクリックしたら、**Delete** キーを押して削除します。
- マウスの右ボタンを押し、ポップアップメニューの[削除]をクリックします。

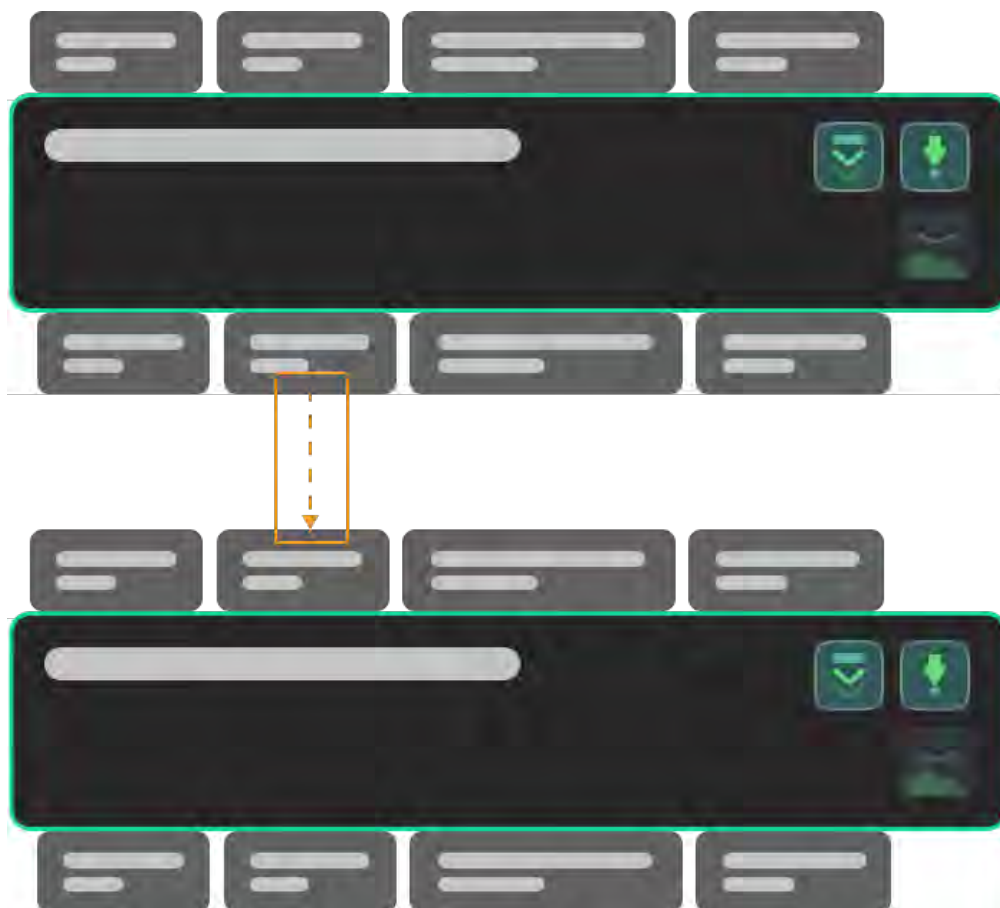


4.2.5.2. ステップ間のつながりを作成・削除

つながりの作成方法

以下は、ステップ「カメラから画像を取得」と「深度画像を点群に変換」を例として説明します。ステップ「カメラから画像を取得」の出力ポートにマウスカーソルを合わせ、マウスの左ボタンを押したまま、ステップ「深度画像を点群に変換」の入力ポートにマウスをドラッグし、左ボタンを離してつながりを作成します。

一つの出力ポートを複数の入力ポートにつなぐことはできますが、一つの入力ポートは一つの出力ポートにしかつなぐことはできません。



つながりは、同じデータタイプのポート間でのみ作成できます。たとえば、**Cloud(XYZ-Normal)**の出力ポートは、**Cloud(XYZ-Normal)**の入力ポートにのみつながります。異なるデータタイプのポートの間にはつながりを作成することはできません。

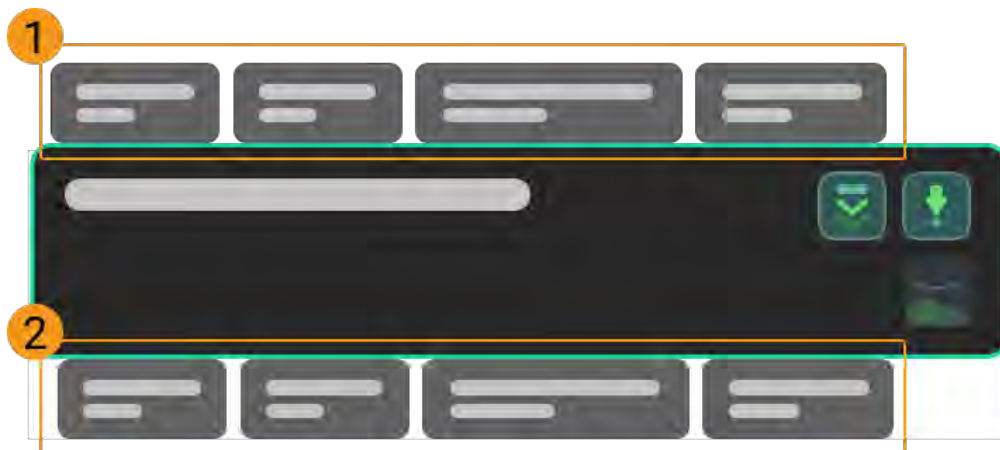
つながりの削除方法

- つなぎを左クリックするとつながりがハイライト表示され、**Delete** キーを押して削除します。
- つなぎを右クリックするとつながりがハイライト表示され、ポップアップメニューの[削除]をクリックして削除します。

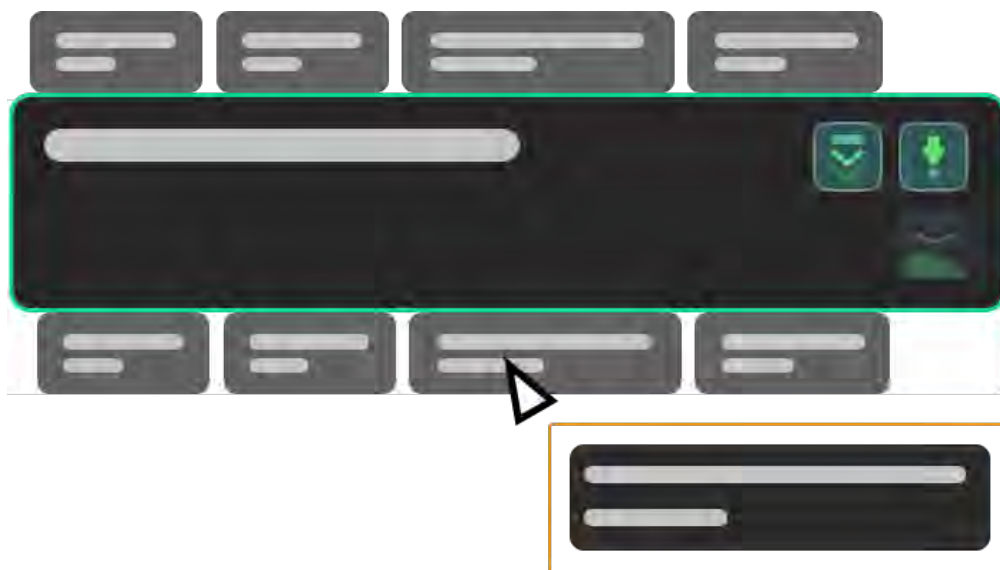


4.2.5.3. ステップの入力・出力を表示

ステップの下側に位置しているものが出力、上側に位置しているものが入力になります。下図に示すように、1は入力ポートで、2は出力ポートです。



マウスをポートに移動すると、下図に示すようなデータタイプの詳細説明が表示されます。



4.2.5.4. ステップの異なる状態を表示

ステップを実行した後、ステップ枠の色で異なる状態を確認できます。

ステータス	サンプル図
正常	
選択中	
ワーニング	
エラー	

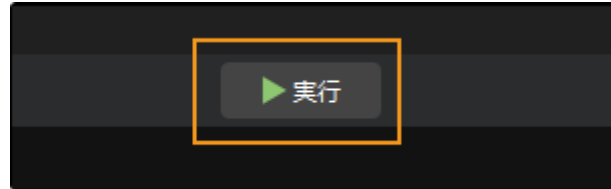
4.2.5.5. ステップを実行して可視化出力結果を表示


プロジェクト全体を正常に実行することは、個々のステップの実行に依存します。本節では、

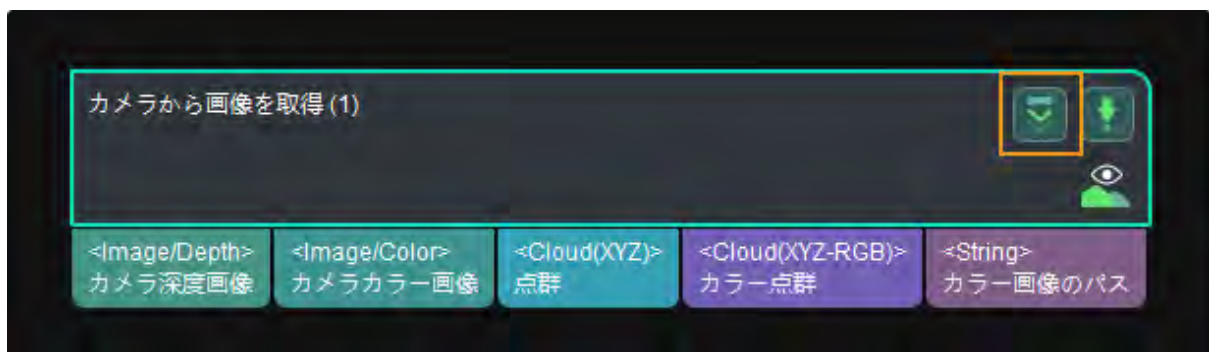
ステップを実行し、デバッグ結果出力の機能でステップの実行効果を表示する方法について紹介していきます。


ステップを実行

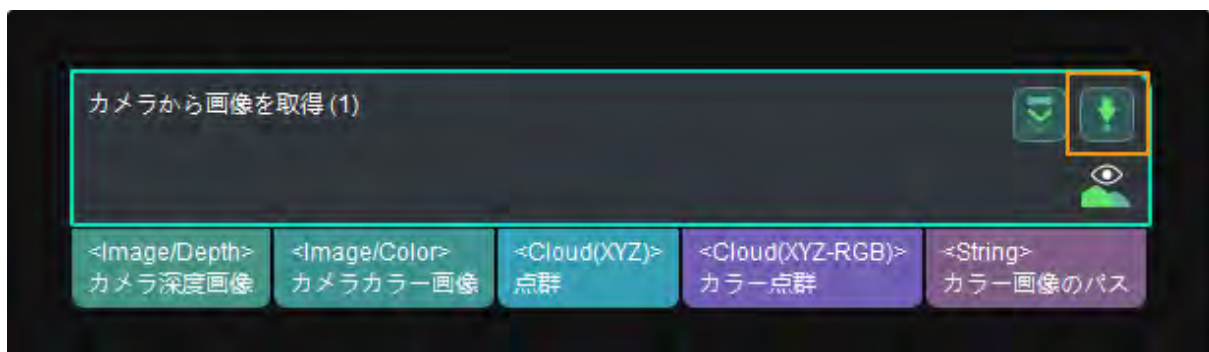
- プロジェクトツールバーの[実行]をクリックすると、プロジェクト全体が実行されます。



-  をクリックすると、該当ステップからプロジェクトが実行されます。



-  をクリックすると、ステップがシングル実行されます。



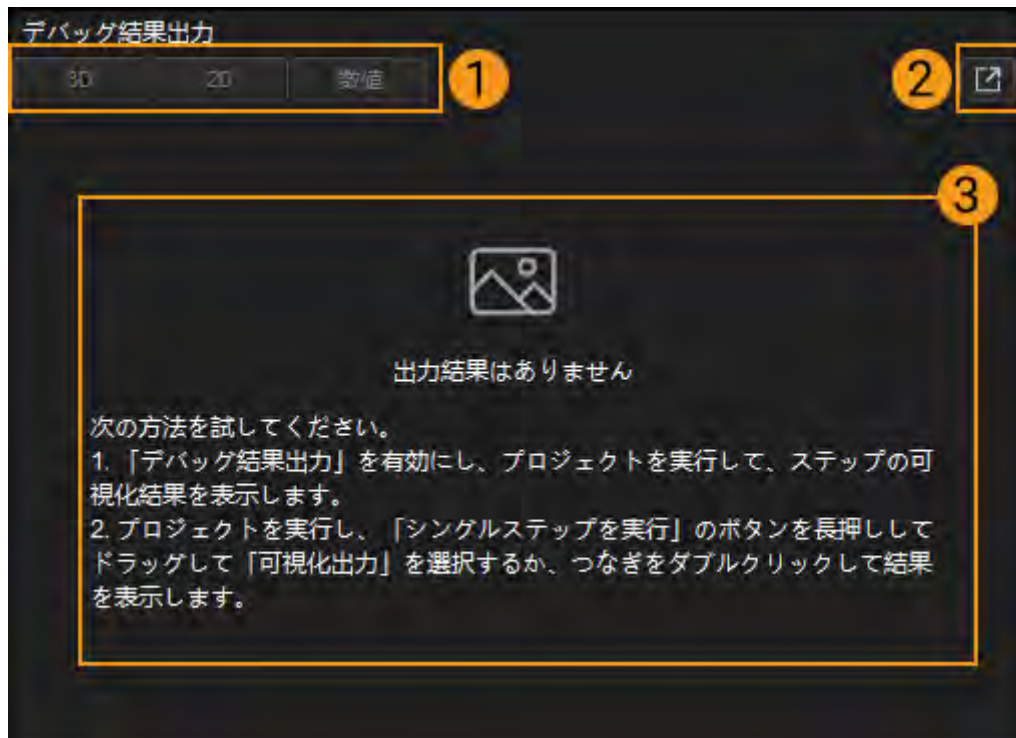
デバッグ結果出力

デバッグ結果出力の可視化ウィンドウは、プロジェクト結果の統合表示画面であり、Mech-Visionの重要な部分でもあります。

プロジェクトのデバッグ中に、デバッグ結果出力の機能を使用して、現在のステップの可視化出力効果を直感的に確認できます。

画面紹介

下図に示すように、デバッグ結果出力の画面はMech-Visionのメインインターフェースの右上にあります。





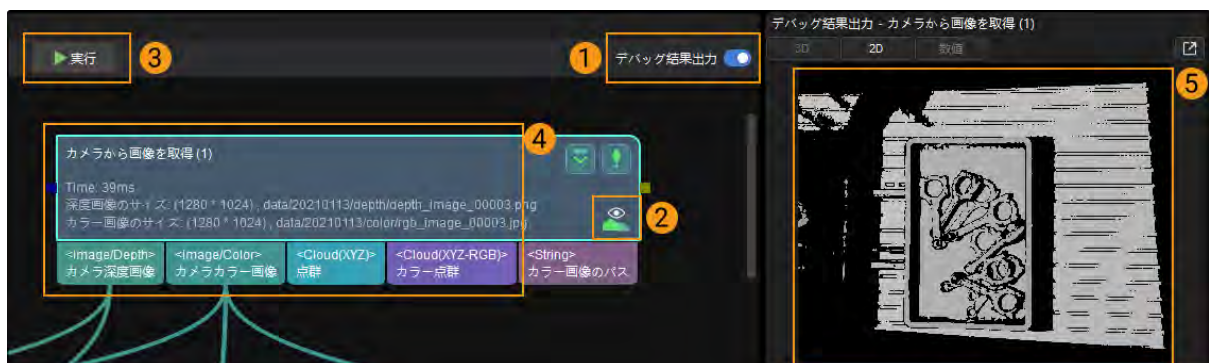
この画面の説明は以下の通りです。

- ①は出力結果のデータタイプです。
- ②をクリックするとウィンドウが表示され、独立したウィンドウとして表示されます。
- ③から可視化出力結果を表示することができます。


使用方法

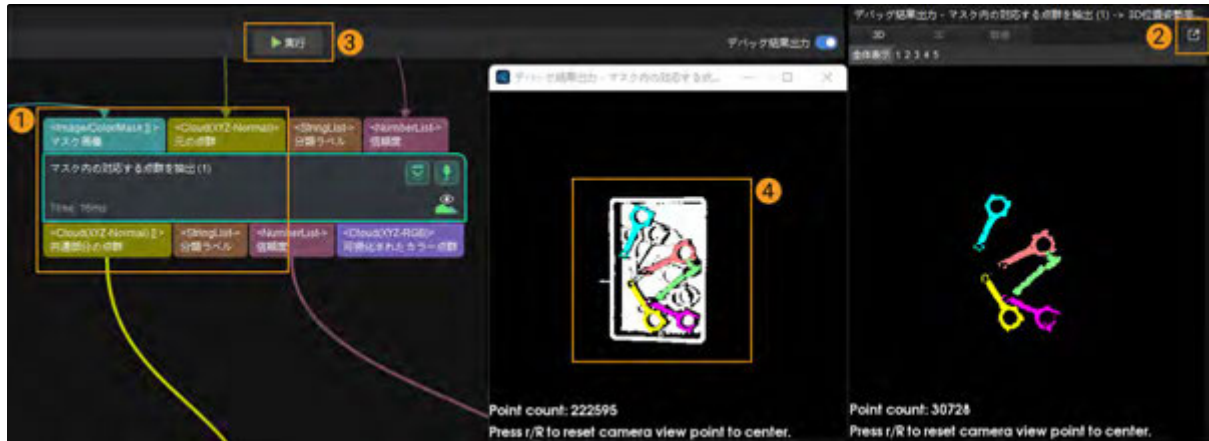
デバッグモードで単独ステップの可視化出力結果を表示

プロジェクトツールバーの**デバッグ結果出力** (①) をオンにして、結果を可視化するステップの可視化出力のスイッチ (②) を有効にします。すると、右側にある  が  になります。次に、プロジェクトツールバーの[実行]をクリックし、可視化出力が有効になっているステップをクリックして、これらのステップの結果をデバッグ結果出力の可視化ウィンドウにすばやく確認することができます。



デバッグモードで単独ステップを複数回実行した可視化出力結果を表示

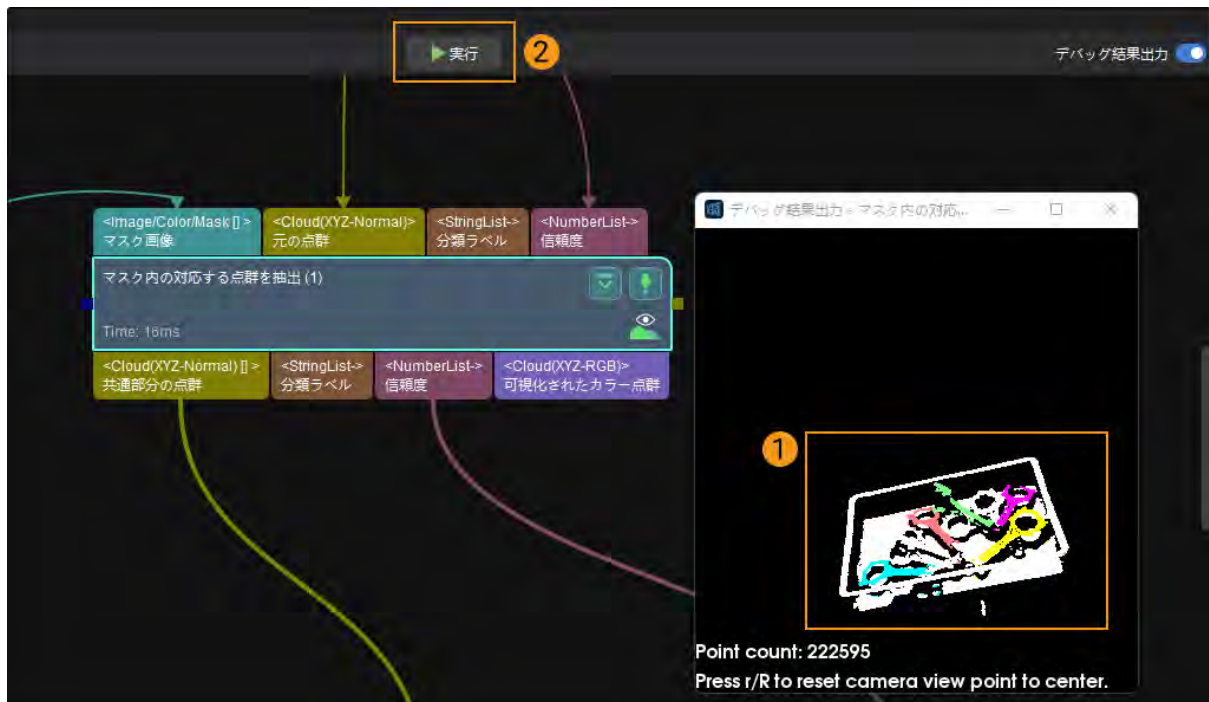
特定ステップを複数回実行した結果を表示する場合は、最初にこのステップを選択し、をクリックしてデバッグ結果出力のウィンドウを別々に表示します。次にプロジェクトツールバーの[実行]をクリックして、独立したデバッグ結果出力のウィンドウ（④）でこのステップの複数回の実行結果を表示できます。




複数ステップを複数回実行した結果を表示する場合は、上記の方法で複数のデバッグ結果出力のウィンドウを別々に表示してください。

デバッグモードで他の視点から単独ステップを複数回の可視化出力結果を表示

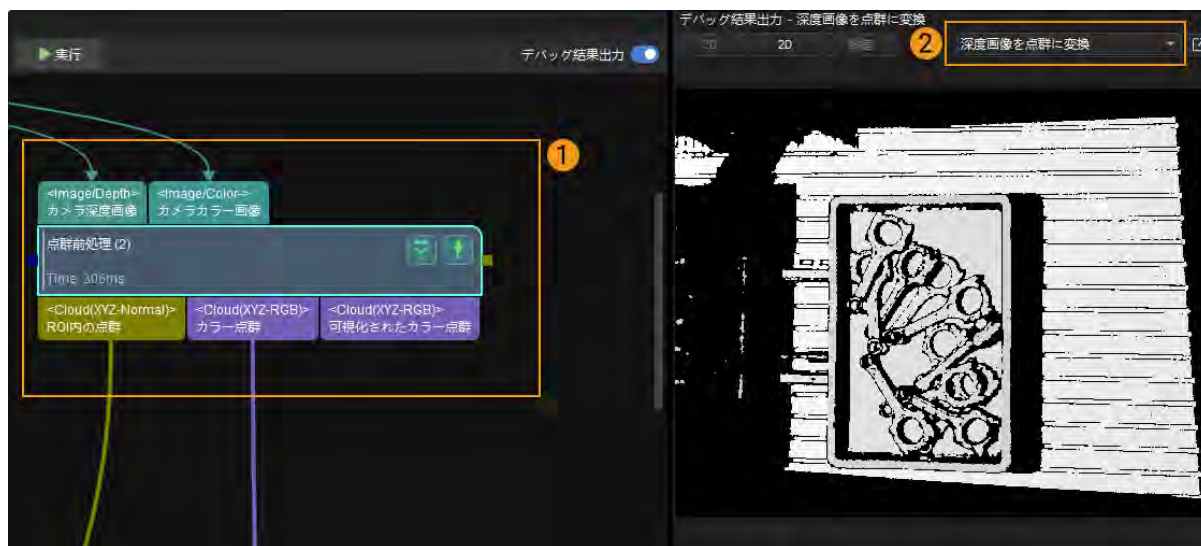
ステップの出力結果を他の視点から表示する場合は、独立したデバイス結果出力のウィンドウでマウスをドラッグして視点を調整し、プロジェクトツールバーの[実行]をクリックして、この視点でステップを複数回実行した結果を表示できます。



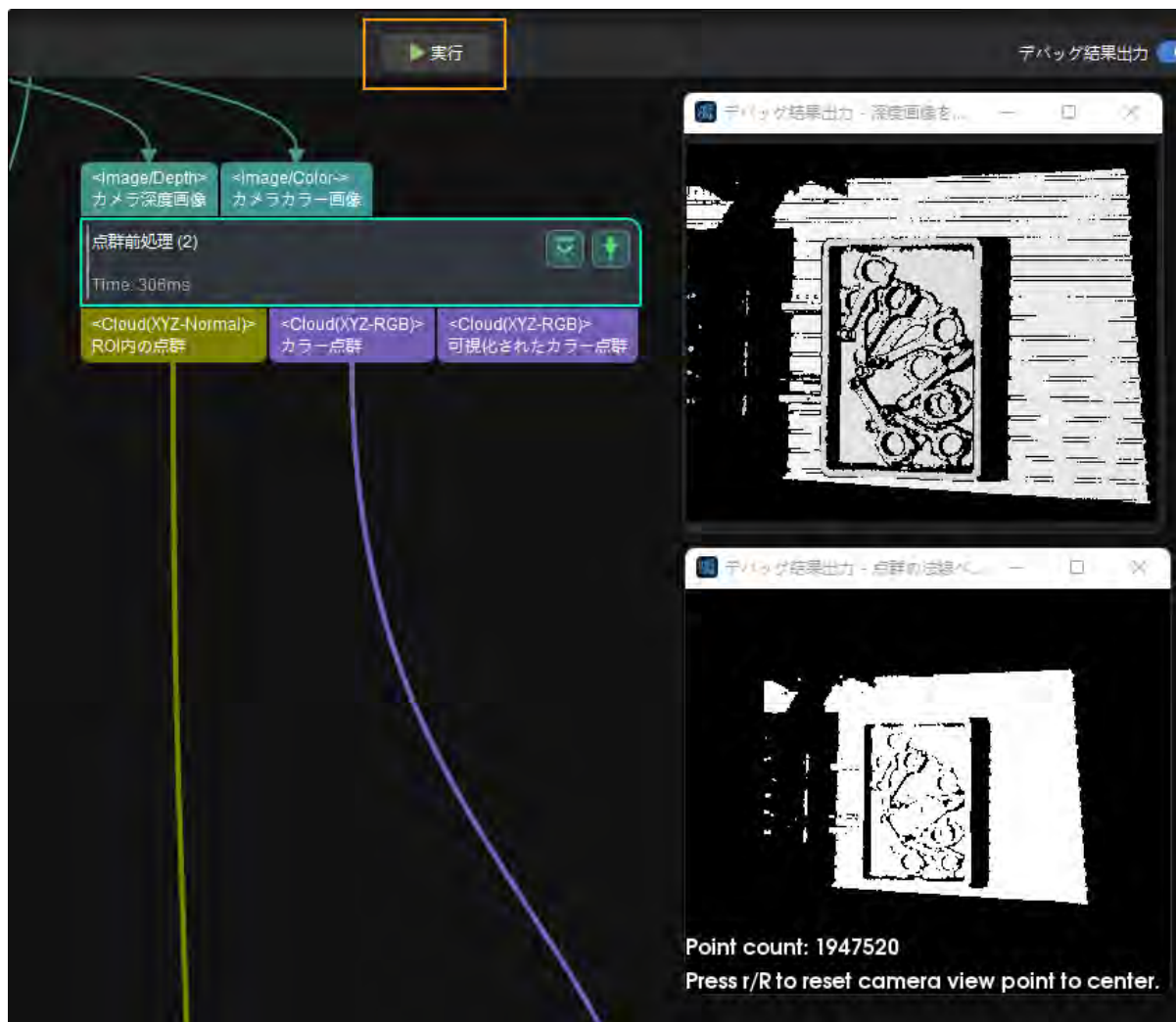
ステップの組合せ内のステップの可視化出力結果を切り替えて表示

プロジェクトにステップの組合せがあり、ステップの組合せに出力結果を可視化できるステップが含まれており、 がオンになっている場合、デバッグ結果出力の機能を使用して、ステップの組合せ内のステップの結果を表示できます。

最初にステップの組合せを選択し、デバッグ結果出力の可視化ウィンドウのドロップダウンバー（下図の②）で可視化出力結果を独立して表示したいステップを選択します。




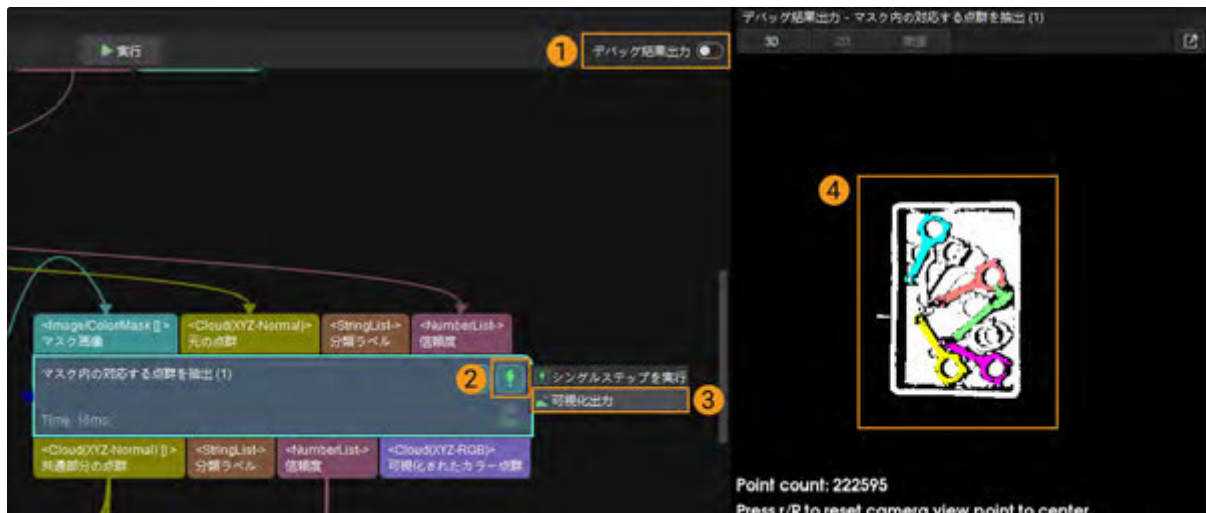
次に、プロジェクトツールバーの[実行]をクリックして、ステップの組合せ内のステップの複数回実行した可視化出力結果を表示できます。



非デバッグモードでステップの可視化出力結果を表示

プロジェクトツールバーの**デバッグ結果出力**がオフになっているときに、ステップの可視化出力結果を表示する方法は2つあります。

-  でマウスの左ボタンを押したままドラッグし、[可視化出力]をクリックして、デバッグ結果出力のウィンドウからステップの可視化出力結果を表示できます。



- ステップ間のつながりをダブルクリックして、デバッグ結果出力のウィンドウからステップの可視化出力結果を表示できます。



ステップの可視化出力結果を表示するには、プロジェクトまたはその中のステップが既に実行されている必要があります。

4.2.5.6. ステップのコメントを追加・変更・表示

ステップコメントを使用して、プロジェクト内のステップまたはステップの組合せにディスクリプションを追加できます。

コメントの追加方法

1. コメントを追加する必要があるステップまたはステップの組合せを右クリックし、**現在選択されているステップのコメントを追加**を選択します。




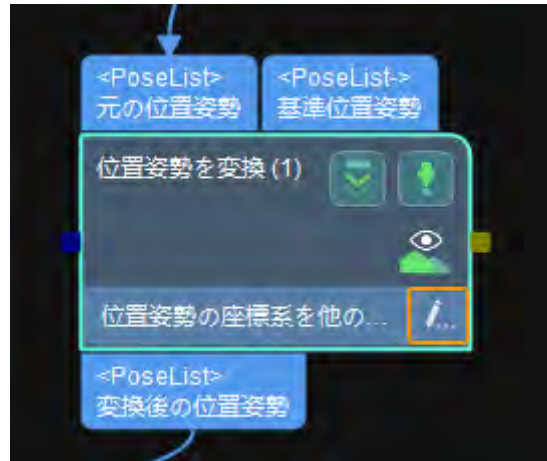
2. ポップアップウィンドウでコメントを追加し、空白スペースをクリックしてウィンドウを閉じます。

以下にディスクリプションを入力してください（入力と出力のディスクリプション、アルゴリズムの目的など）。

位置姿勢の座標系を他の座標系に変換します。

コメントの変更方法

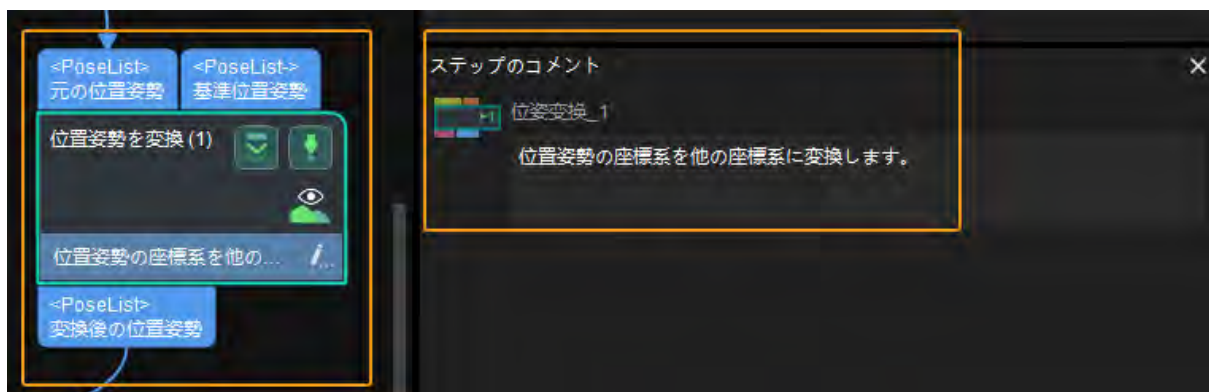
 をクリックしてコメントを変更することができます。



コメントの表示方法

ステップのコメント画面を開きます。右側にあるプロジェクト設定エリアにこのタブがない場合、**ビュー・ステップのコメント** にチェックを入れたかどうかを確認してください。

この画面で追加されたコメントが表示されます。コメントをクリックすると、プロジェクト編集エリアで対応するステップまたはステップの組合せにジャンプし、点滅表示します。



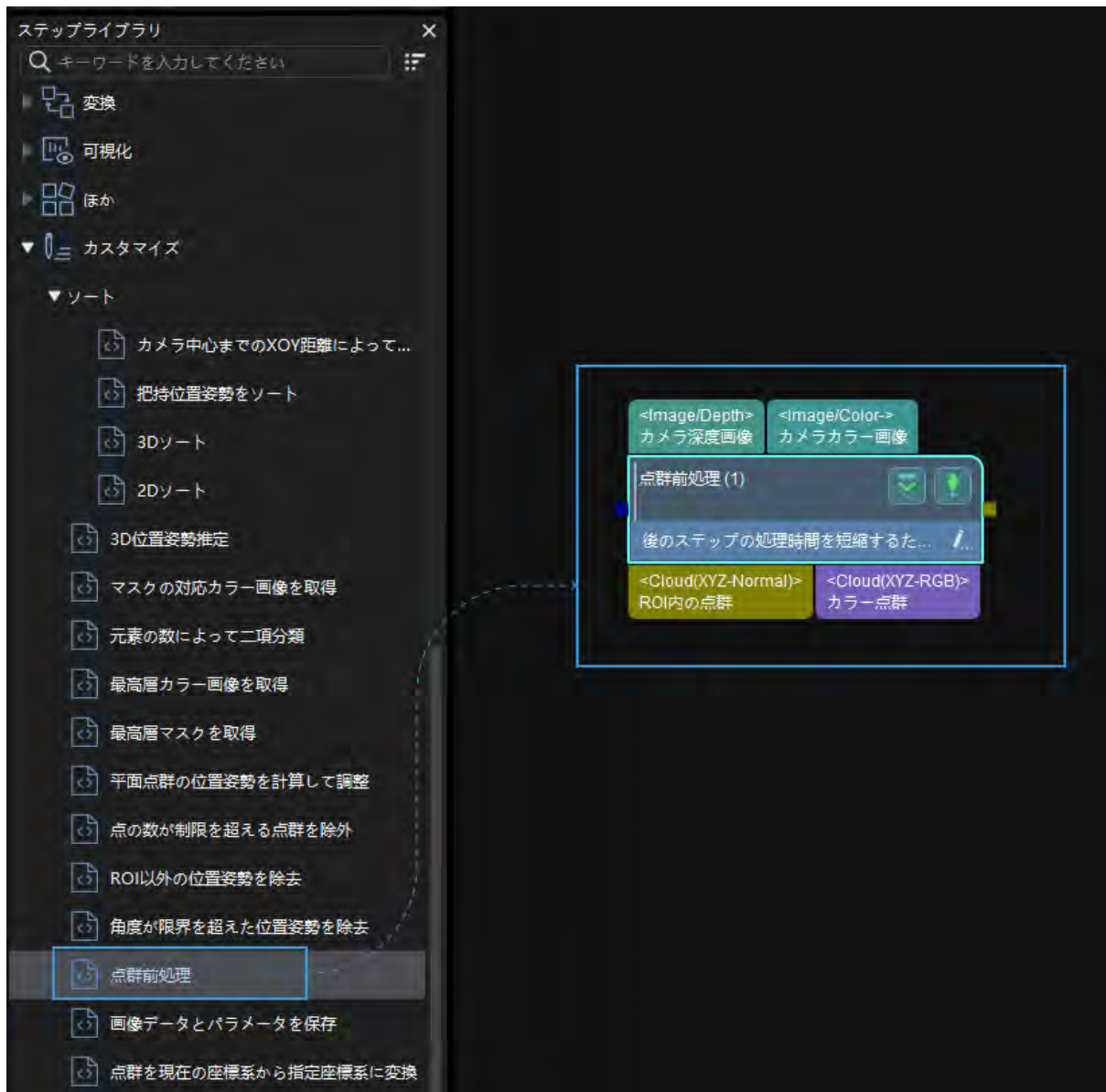
4.2.5.7. ステップの組合せの基本操作

ステップの組合せを使用する前に、**ステップの組合せの基本操作**をお読みください。

本節を開始する前に、新しいプロジェクトを作成する必要があります。**点群前処理**のステップの組合せを使用して説明します。

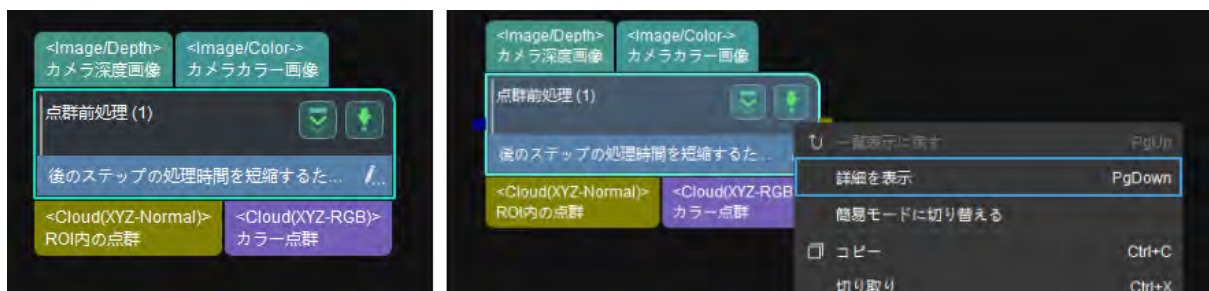
- ステップの組合せをプロジェクトに追加する方法

ステップライブラリから**カスタマイズ・点群前処理**を見つけて、それをプロジェクト編集エリアにドラッグします。



● ステップの組合せの詳細を表示する方法

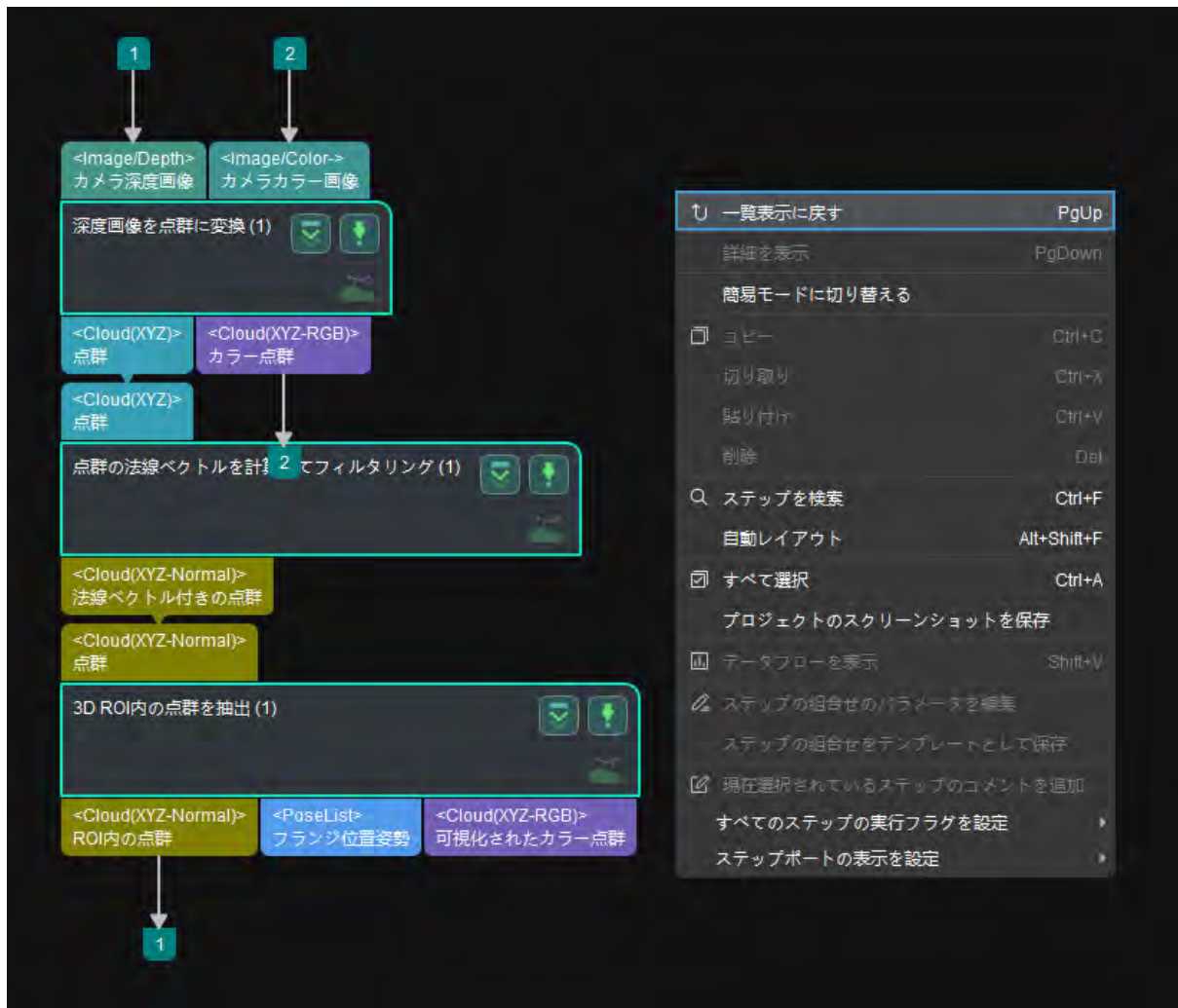
ステップの組合せをダブルクリックするか、右クリックで表示されるメニューから[詳細を表示] (ショートカット **PgDown**) をクリックします。



● ステップの組合せを終了する方法

プロジェクト編集エリアの任意場所で右クリックして、表示されるメニューから[一覧表示に戻]

す] (ショートカット **PgUp**) をクリックします。

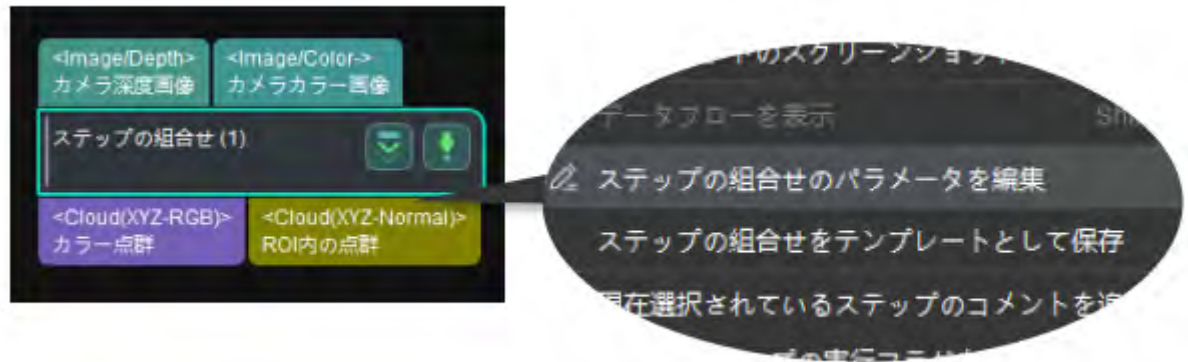


4.2.5.8. ステップの組合せのパラメータを変更

使用シーンによってステップの組合せのパラメータを変更する必要があります。本節では、**点群前処理**を例としてステップの組合せのパラメータ変更について説明します。主に以下の2つの部分に分けられています。

- **パラメータの追加と削除**
- **パラメータ表示情報のカスタマイズ**

点群前処理を右クリックし、表示されるメニューが[**ステップの組合せのパラメータを編集**]をクリックします。




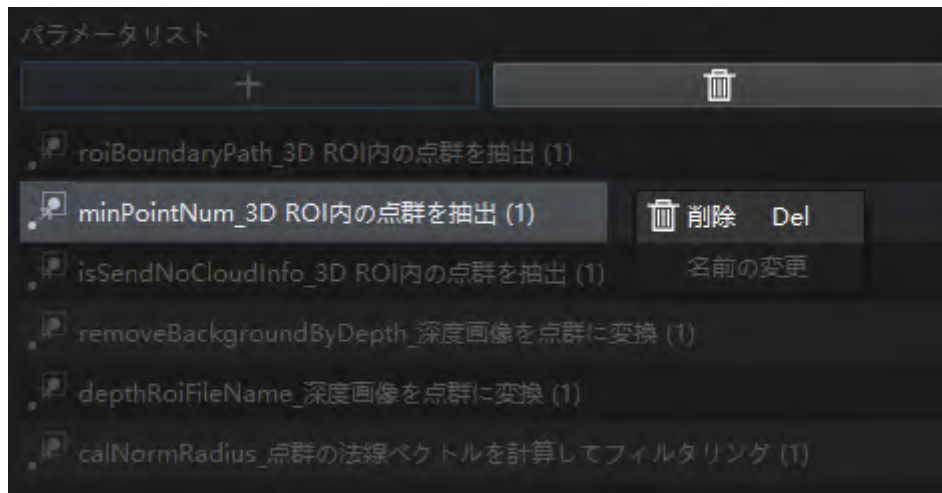
パラメータの追加と削除

下図に示すように、パラメータリストのパラメータは、ステップパラメータバーのパラメータに1つずつ対応しています。



削除

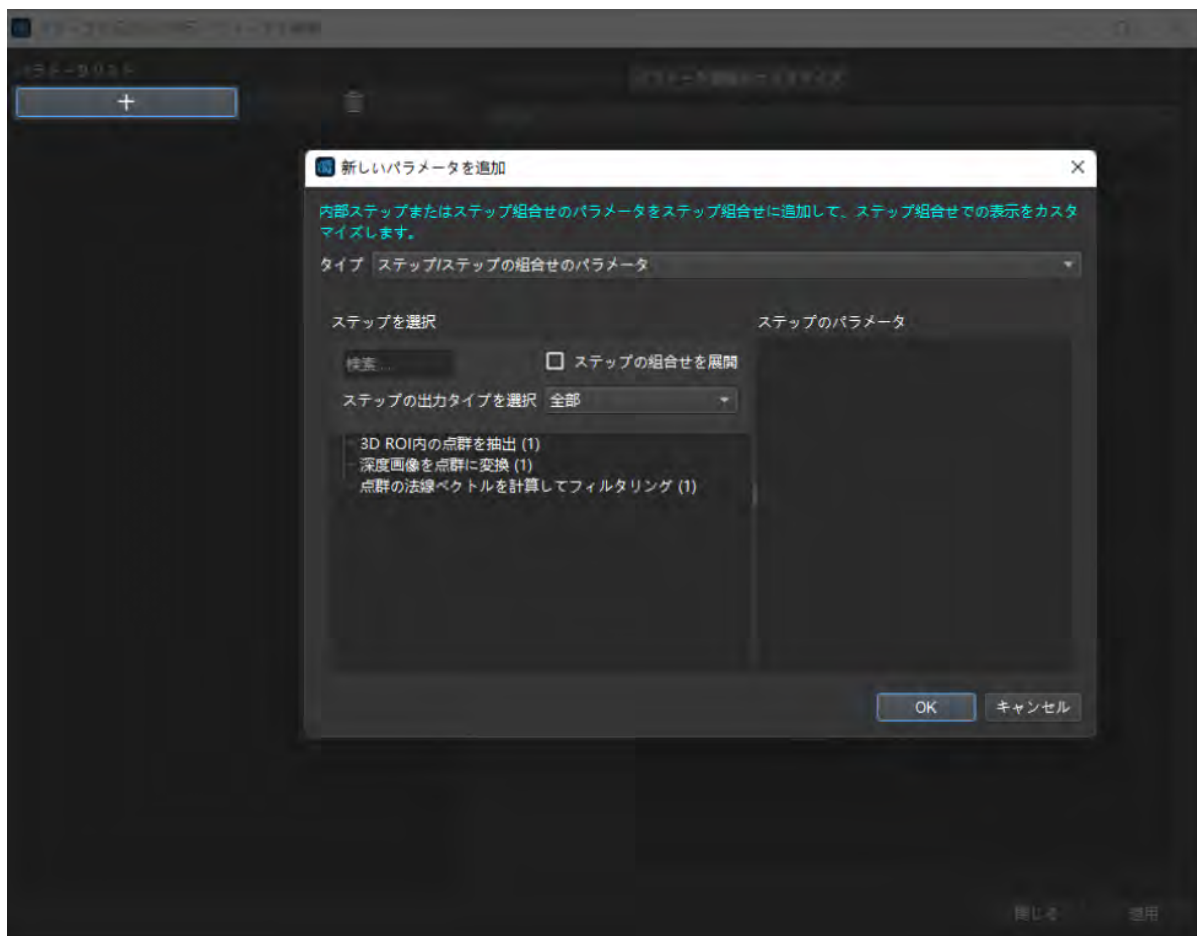
削除するパラメータを選択して  をクリックするか、右クリックして表示されるメニューから **[削除]** をクリックしてから、ポップアップウィンドウで **[Yes]** をクリックします。



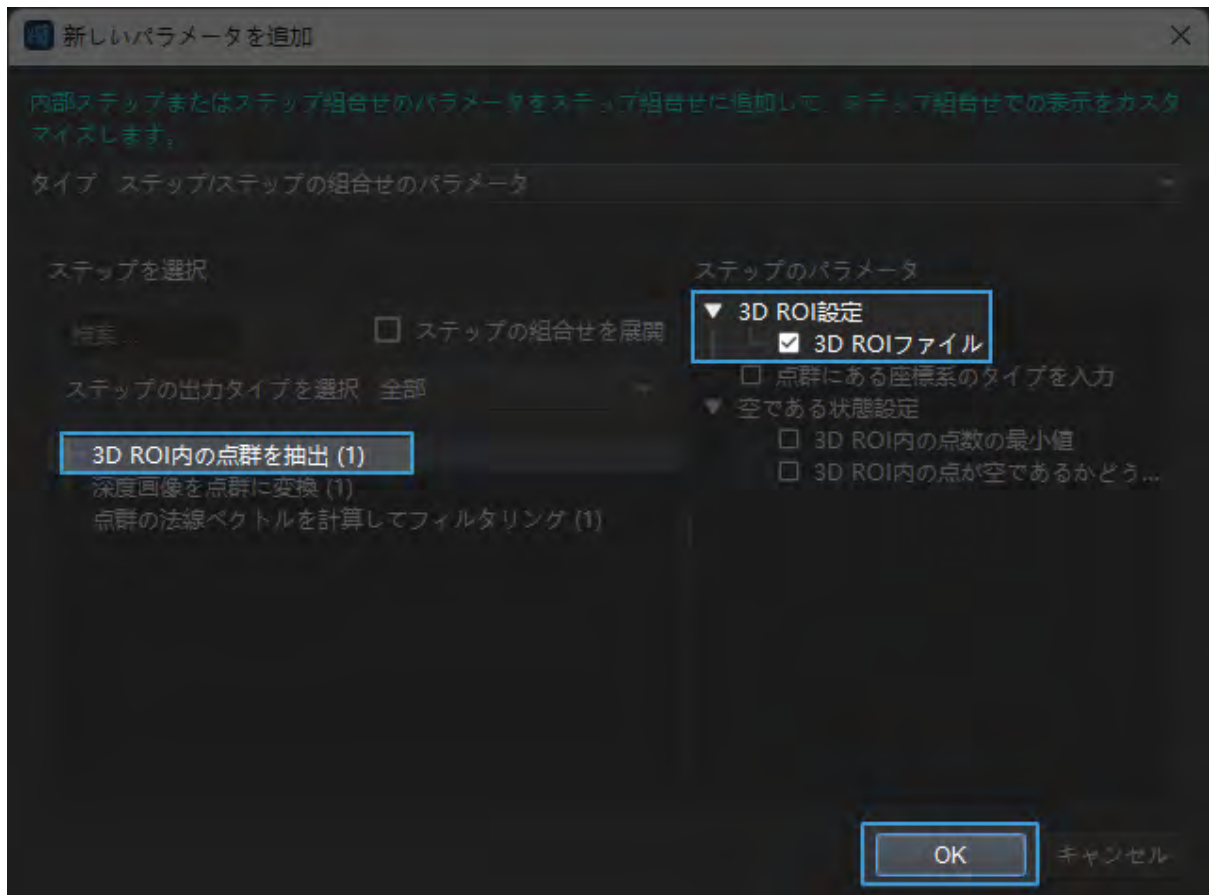
[適用]をクリックした後、**ステップパラメータ**で対応するパラメータが削除されます。

追加

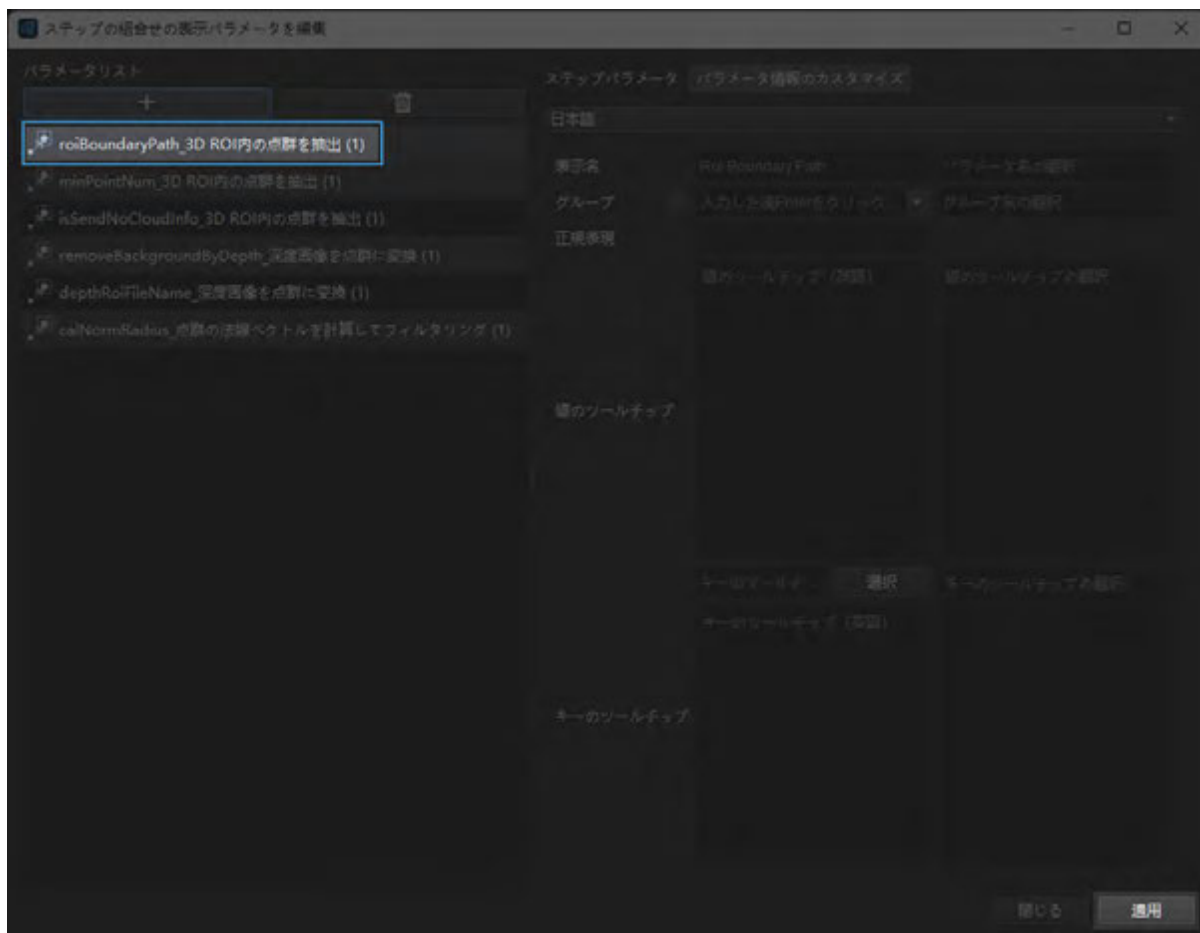
ステップの組合せの表示パラメータを編集画面で[+]をクリックして新しいパラメータを追加画面に入ります。



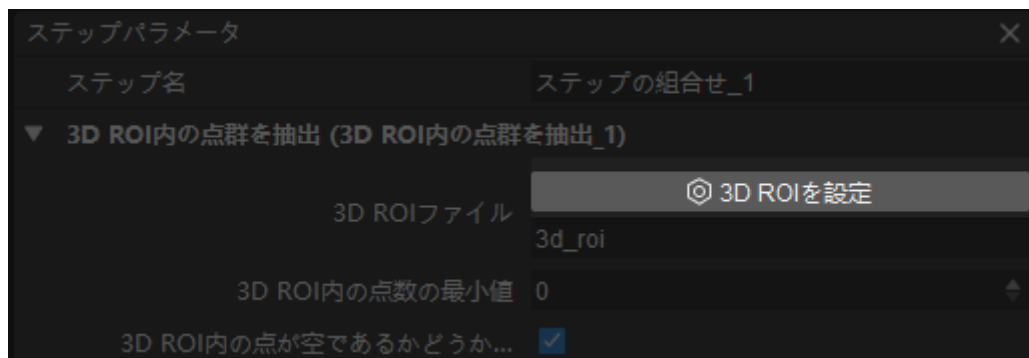
ステップの組合せ内のステップのパラメータを選択します。下図に示すように、**3D ROI内の点群を抽出**の**3D ROI**パラメータを選択し、[OK]をクリックして設定を保存します。



これで、前のステップで選択したステップがパラメータリストに表示されます。[適用]をクリックして、パラメータの追加を有効にします。



ステップパラメータで新しく追加されたパラメータが表示されます。



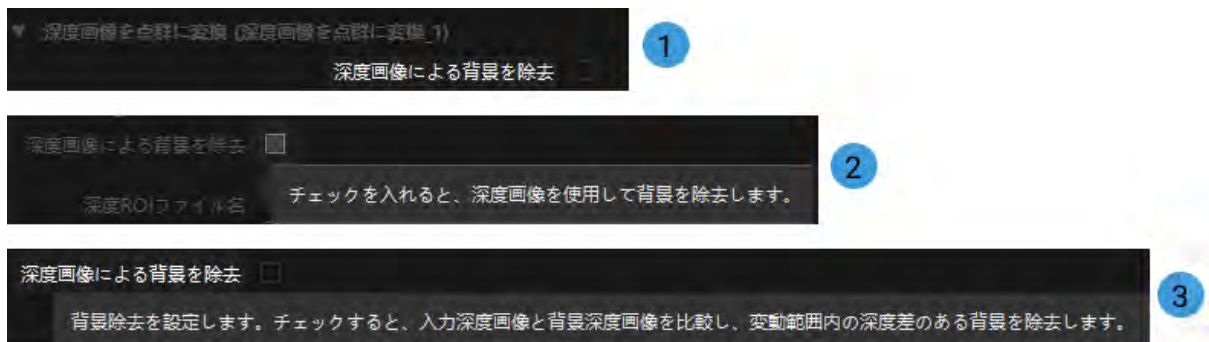
パラメータ表示情報のカスタマイズ

ステップの組合せの表示パラメータを編集画面で編集するパラメータをクリックしてパラメータ情報のカスタマイズ画面に入ります。

下図に示すように、表示名、値のツールチップ、キーのツールチップなどを編集したら、[適用]をクリックします。



変更後のパラメータ表示情報を下図に示します。



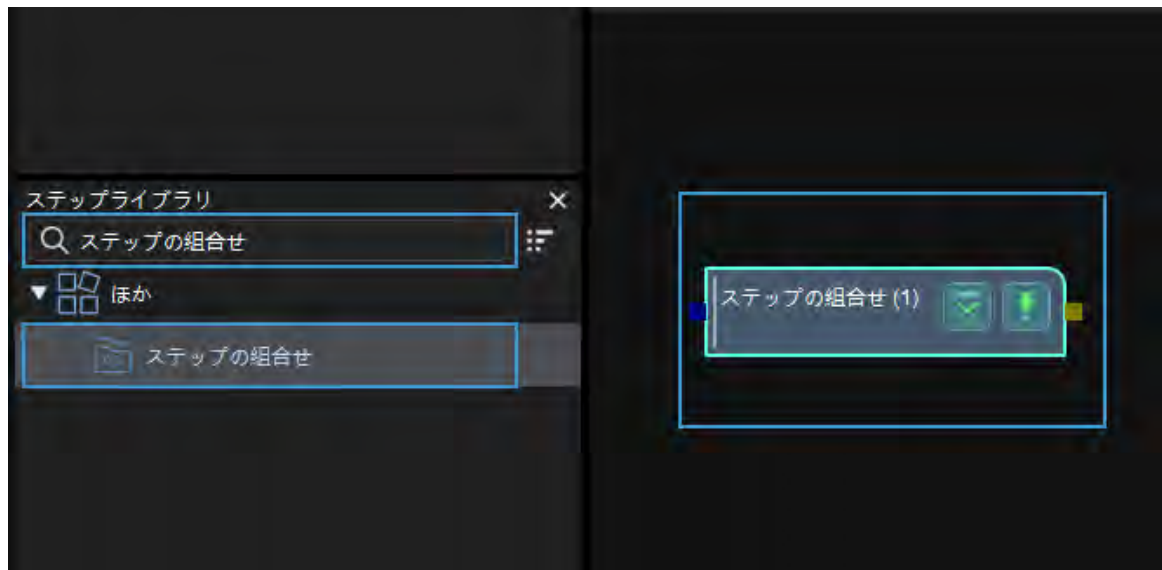
4.2.5.9. ステップの組合せをカスタマイズ

ソフトウェアに組み込まれているよく使うステップの組合せに加え、ステップの組合せをカスタマイズすることも可能です。

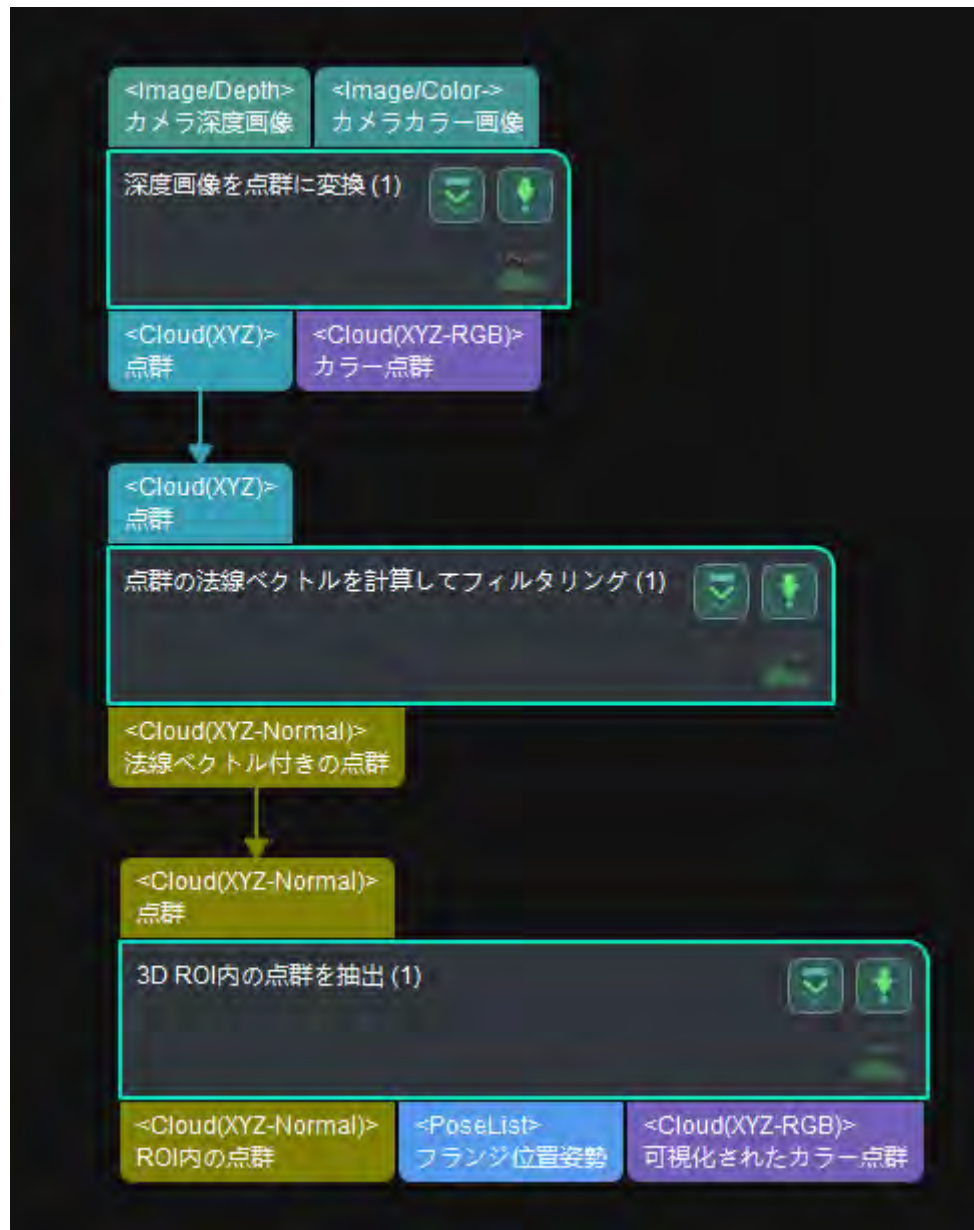
本節では、点群処理のステップの組合せをパッケージ化することを例として説明しています。使用するステップは **深度画像を点群に変換**、**点群の法線ベクトルを計算してフィルタリング**、**3D ROI内の点群を抽出** です。これらのステップをパッケージ化するには、**ステップの組合せ** が必要です。

ステップを追加

1. 下図に示すように、ステップライブラリから [\[vision-steps:procedure:::vision-procedure\]](#) をプロジェクト編集エリアにドラッグします。



2. クリックして [\[vision-steps:procedure:::vision-procedure\]](#) の中に入り、下図を参照してステップを追加してつなぎます。



ステップの組合せの入力と出力ポートを設定

- 入力ポート：

[[vision-steps:from-depth-map-to-point-cloud:::from-depth-map-to-point-cloud](#)] の入力ポートをダブルクリックし、番号線付きのポートはステップの組合せの入力ポートとして使用されます。



入力ポートのレイアウト順序は、ポートをダブルクリックする順序によって決まります。

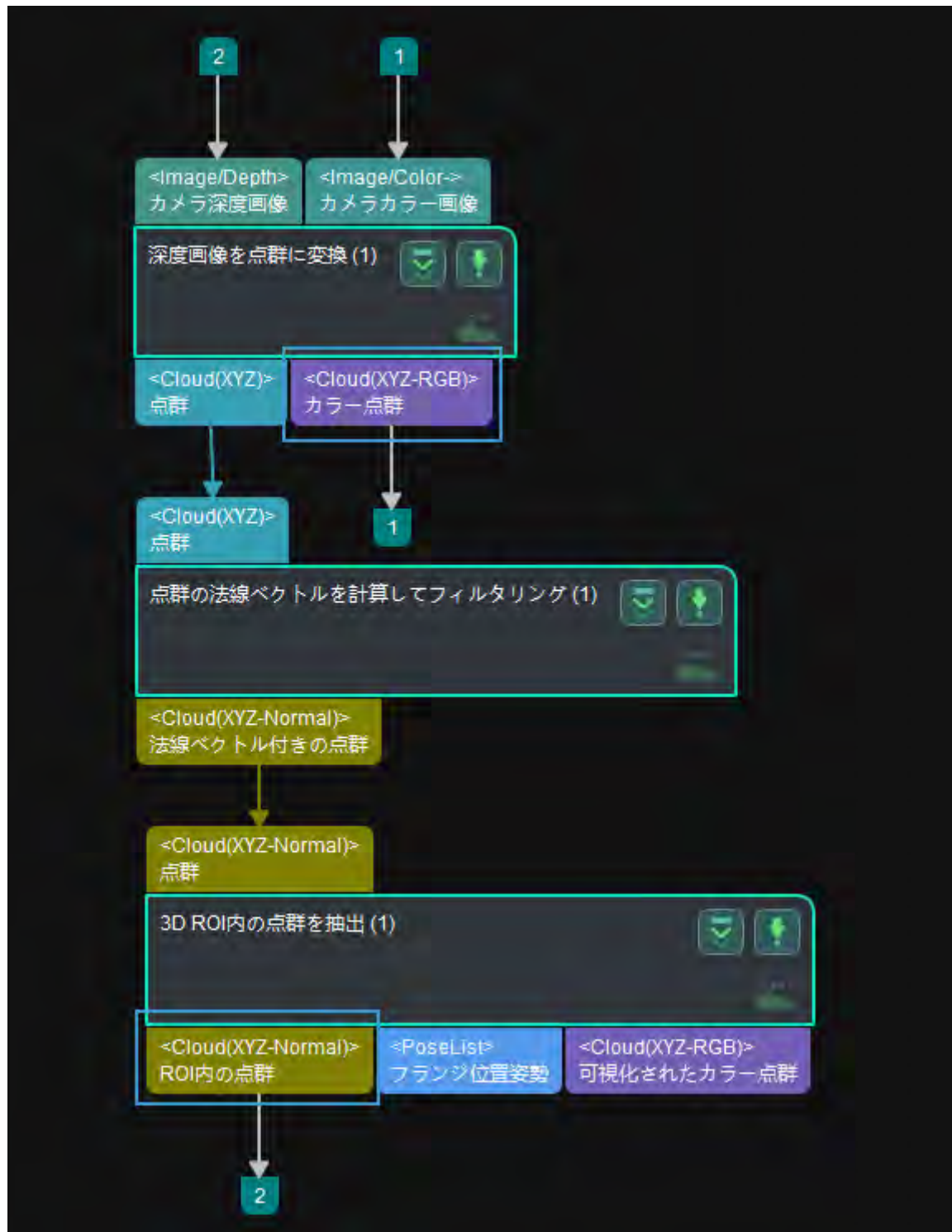
カメラ深度画像ポートをクリックしてからカメラカラー画像ポートをクリックする場合の効果は上図に示し、カメラカラー画像ポートをクリックしてからカメラ深度画像ポートをクリックする場合の効果は下図に示します。



ポート上部の数字を選択して **Delete** キーを押すか、右クリックで表示されるメニューから[削除]をクリックしてポートを削除します。

● 出力ポート：

設定方法は上記の入力ポートの設定と同じです。



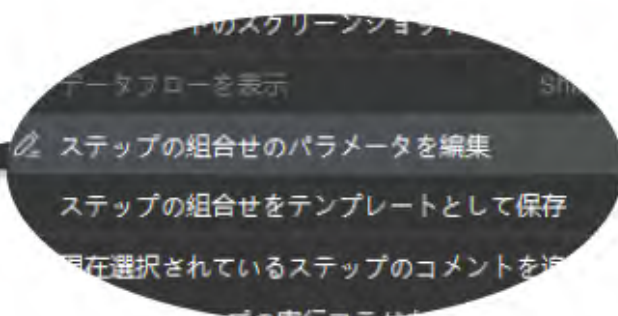
ステップの組合せのパラメータを設定

下図に示すように、ステップの組合せを作成した後、名前のみを変更できます。

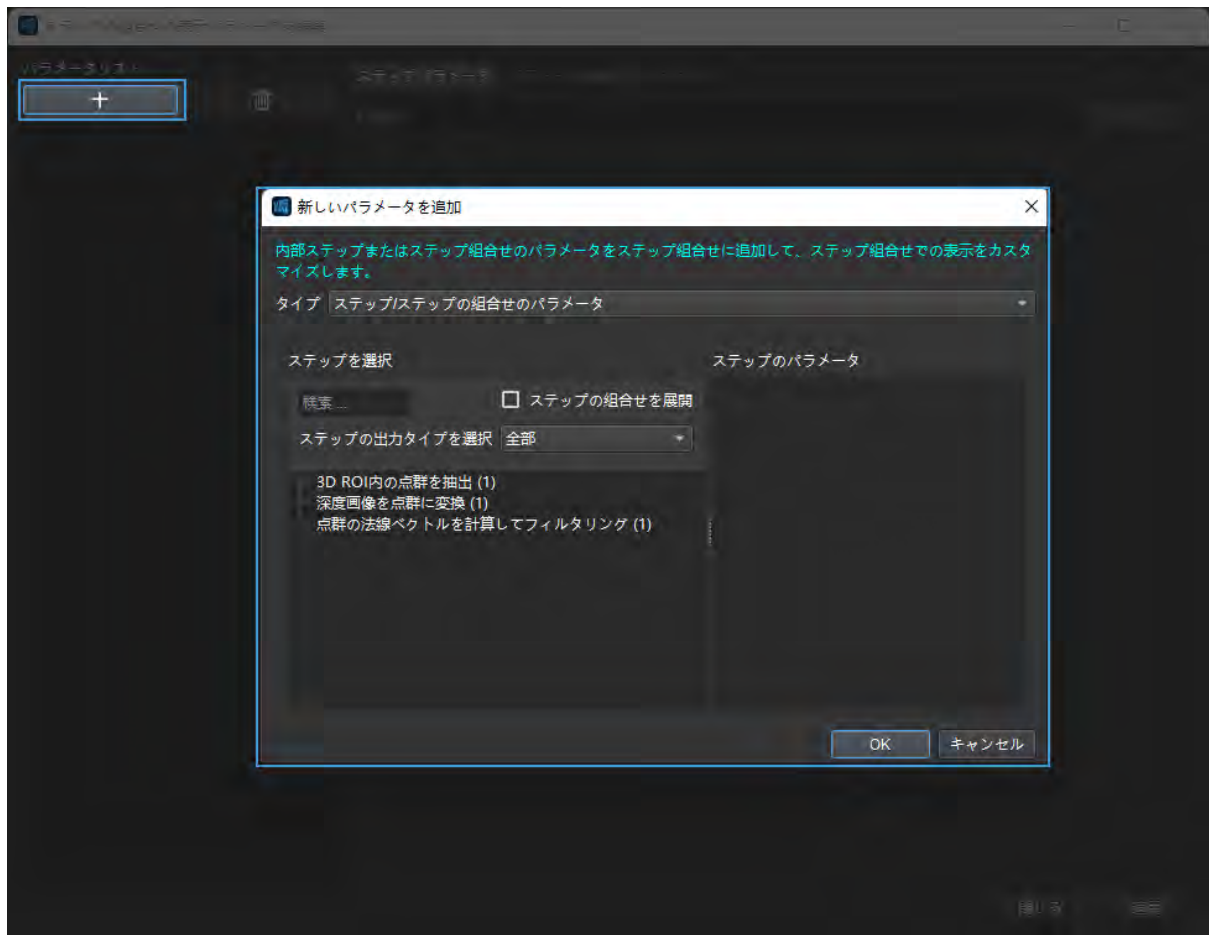


ステップの組合せのパラメータを調整しやすくするために、その中のステップの主要パラメータをステップの組合せのパラメータとして設定することができます。

1. ステップの組合せを右クリックし、表示されるメニューから[**ステップの組合せのパラメータを編集**]をクリックします。



2. ステップの組合せの表示パラメータを編集画面で[+]をクリックして新しいパラメータを追加画面に入ります。

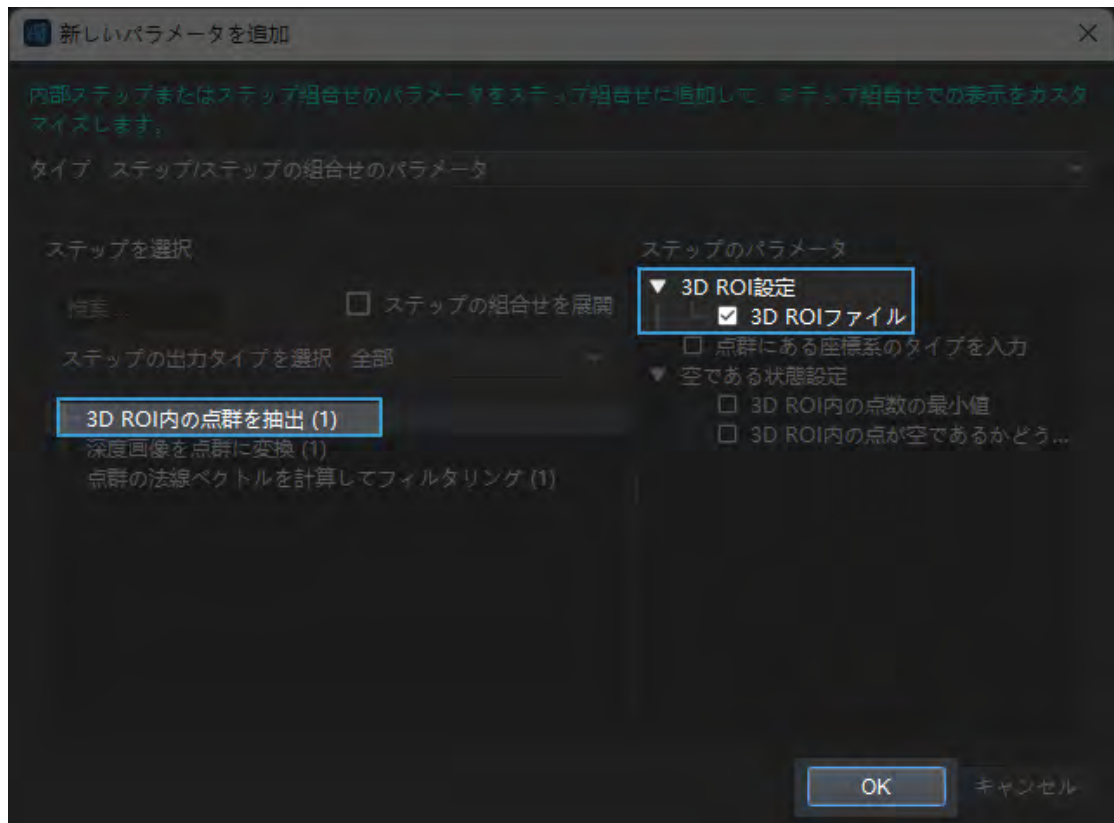


設定可能なパラメータタイプは以下の3つあります。

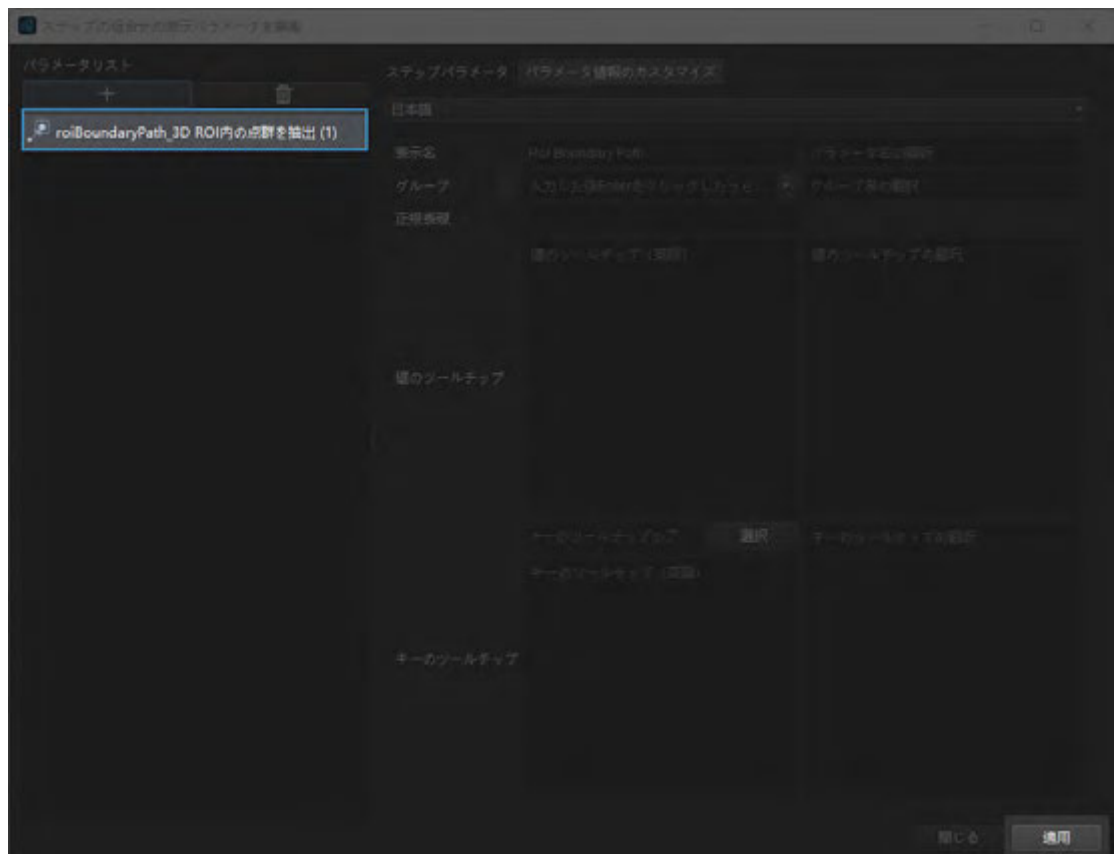
- ステップ/ステップの組合せのパラメータ（本節で紹介）
 - マッピングパラメータをカスタマイズ
 - レシピパラメータをカスタマイズ
1. ステップの組合せ内のステップのパラメータを選択します。下図に示すように、[\[vision-steps:extract-3d-points-in-3d-roi:::extract-3d-points-in-3d-roi\]](#) の**3D ROI**パラメータを選択し、**[OK]**をクリックして設定を保存します。



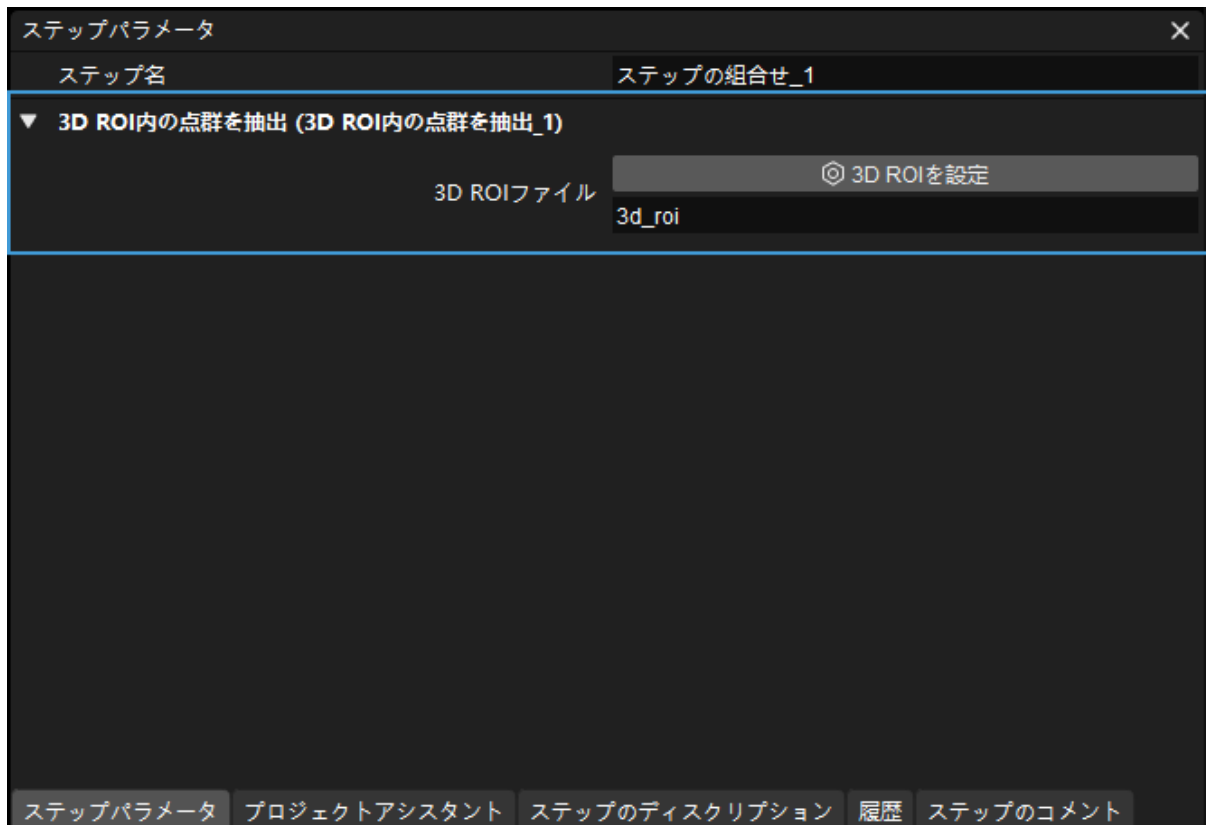
実際の状況に応じて複数のパラメータにチェックを入れることができます。



- これで、前のステップで選択したステップがパラメータリストに表示されます。[適用] をクリックして、パラメータの追加を有効にします。



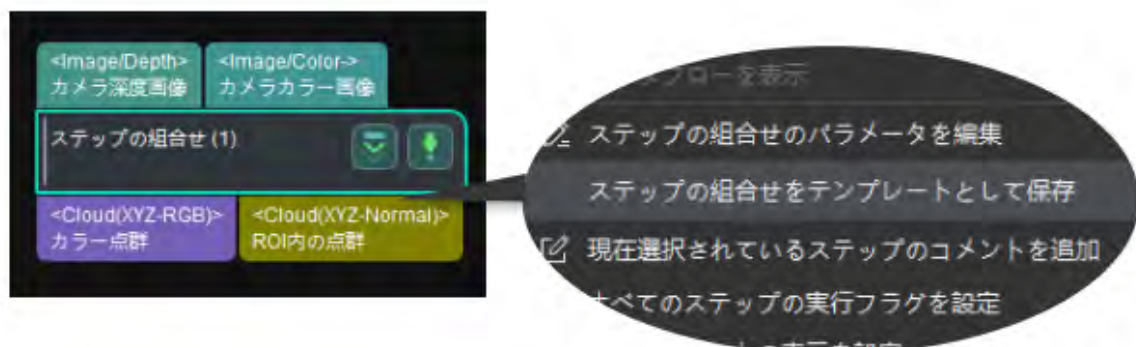
3. ステップの組合せの**ステップパラメータ**で1つのパラメータが正常に設定されます。



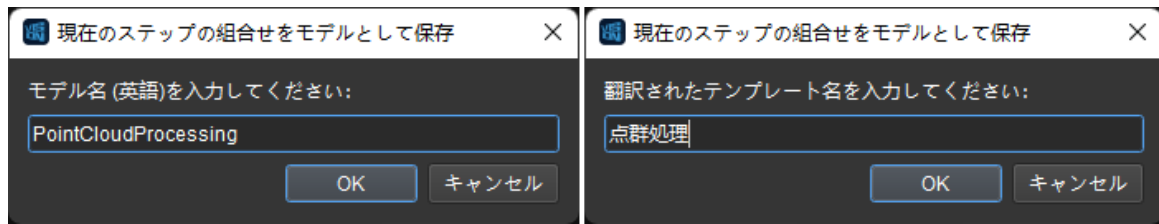
ステップの組合せを保存

設定後のステップの組合せをステップライブラリの「カスタマイズ」グループに追加することができます。

1. ステップを右クリックし、表示されるメニューで[**ステップの組合せをテンプレートとして保存**]をクリックします。このオプションは開発者モードでのみ表示されます。

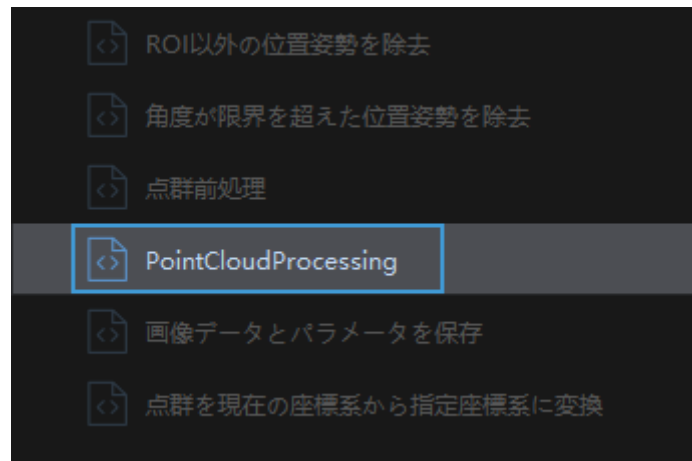


2. ソフトウェアのプロンプトに従ってテンプレート名を入力します。

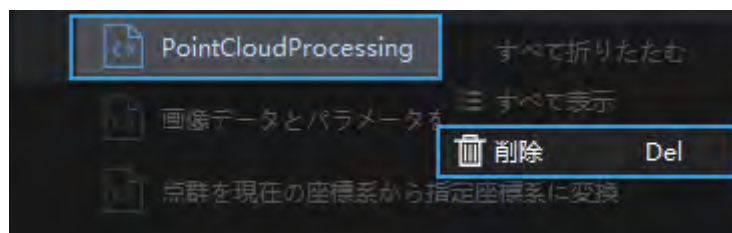


テンプレート名はステップライブラリに既にあるステップ名を入力しないでください。

3. 入力後、[OK]をクリックして保存すると、ステップライブラリの「カスタマイズ」グループに前で追加されたステップの組合せが表示されます。



カスタマイズしたステップの組合せをクリックして **Delete** キーを押すか、右クリックして表示されるメニューから **[削除]** をクリックして、追加されたステップの組合せを削除することができます。



- [custom-mapped-parameter.pdf](#)
- [custom-recipe-parameter.pdf](#)

4.3. ステップの参照情報

Mech-Visionステップとステップ組合せの概要や使用方法について説明します。

ステップ

- [2Dマッチング](#)
- [2D位置姿勢を3D位置姿勢に変換（正投影に基づく）](#)

- 2D形状マッピング
- 3D位置姿勢低精度推定（マルチモデル）
- 3D位置姿勢低精度推定（V2）
- 3D位置姿勢低精度推定
- 3D位置姿勢高精度推定（簡易版）
- 3D位置姿勢高精度推定（マルチモデル）
- 3D位置姿勢高精度推定
- ワーク認識
- 全てのパラメータを受け入れる
- 位置姿勢にラベルを付ける
- カメラの歪みによる不正確な位置姿勢を調整
- オフセットによって位置姿勢を調整
- 傾きによって位置姿勢を調整
- 正確な経路を得るまで目標点を調整
- 位置姿勢を点群の表面に調整
- 経路の時計回り方向を調整
- 中継
- マスク内の対応する画像を抽出
- マスク内の対応する点群を抽出
- 2D経路を抽出
- 背景除去
- 画像二値化処理
- ブロブ解析
- ブールリストの論理演算
- プロジェクト間にデータを送信
- 絶対値を計算
- 両3Dベクトル間の角度を計算
- マスク面積を計算
- 箱の寸法を計算
- 3Dベクトルの外積を計算
- 対角線の長さを計算
- 直径と厚さを計算

- 視差画像を計算
- 指定方向に沿って両位置姿勢の距離を計算
- 位置姿勢間の距離を計算
- 3Dベクトルの内積を計算
- エッジ点群の法線ベクトルを計算
- 平面度を計算
- ヒストグラムを計算
- 二つの位置姿勢の角度を計算
- 2つの線分の交点を計算
- 線分と円との交点を計算
- 指定軸方向の点群長さを計算
- マスク間の距離を計算
- 指定直線上のマスクの長さを計算
- 平均諧調値を計算
- 長方形の指定辺の中点を計算
- マスクの最小外接長方形を計算
- 3Dベクトルの単位ベクトルを計算
- 点群の法線ベクトルを計算してエッジを推定
- 点群の法線ベクトルを計算してフィルタリング
- 平行度を計算
- 指定高度のピクセルサイズを計算
- 平面の幅を計算
- 点群の曲率を計算
- 平面点群の位置姿勢とサイズを計算
- 長方形の寸法と位置姿勢を計算
- 輪郭度の平面度を計算（ダウンサンプリング）
- 輪郭度の平面度を計算
- 基準方向に沿った投影距離を計算
- 長方形の2D位置姿勢を計算
- Pythonを使用して結果を計算
- 指定した点群のプロパティを計算
- 構造化光センサーのキャリブレーション行列を計算

- ゼロでない領域の中心点を計算
- 円柱の把持位置姿勢を計算
- 3Dベクトルの長さを計算
- キャリブレーションボードの位置姿勢を計算
- キャリパスツール
- 2Dカメラ
- カメラから画像を取得
- 点群のサイズによって分類
- 点群の歪み補正
- モデルフィッティングからの位置姿勢によって点群をフィルタリング
- 点群リストの要素を削除
- 点群処理（GPU）
- 点群スケーリング
- 点群平滑化と法線ベクトル計算
- 3D ボックス内の点群を抽出
- 点群における法線ベクトルを追加
- 点群をクラスタリングして要件を満たす点群を出力
- 連続した線を描画
- 3D ROI内の位置姿勢を取得
- 二つの深度画像を比較
- 数値比較
- 各入力位置姿勢を一部抽出して新しい位置姿勢を合成
- 四元数と並進ベクトルから位置姿勢を合成
- 軸と角度から回転ベクトルを合成
- 両軸から回転ベクトルを合成（右手系）
- 数値から3Dベクトルを合成
- 制御光源
- 2D位置姿勢を3D位置姿勢に変換
- データ型を変換
- ピクセルを物理的な長さに変換
- 2D点を3D点に変換
- 円を2D位置姿勢に変換

- 画像の色空間を変換
- 画像をコピー
- 箱を数える
- データリスト内の指定された次元の要素の数をカウント
- 3D点の数を計算
- 色情報をカウント
- 対象物のモデルを作成
- 画像を切り抜く
- 対象物の寸法を分解
- 位置姿勢を四元数と並進ベクトルに分解
- 回転ベクトルをX-Y-Z軸に分解
- 3Dベクトルを数値に分解
- ディープラーニングモデルを推論 (Mech-DLK 2.1.0/2.0.0)
- ディープラーニングモデルパッケージを推論
- ディープラーニング結果を解析
- スキャンラインに沿ったディープクラスタリング
- 深度画像のエンコード
- 円を検出して測定
- 直線を検出して測定
- 長穴を検出して測定
- 箱検出 (4側面)
- 箱検出 (最大内接長方形)
- 箱検出 (最大内接長方形・V2)
- 箱検出 (2側面)
- 円心検出
- コーナー検出
- エッジ検出
- マーカーを検出
- 内接円検出
- 線分を検出
- マスク内で最大面積の最初のN個の長方形を検出
- 遮蔽された対象物を検出

- 領域の形状特徴を検出
- 頂点を検出
- ピクセルサイズを決定
- しきい値によって数値を二項分類
- 点群を均等に分割
- 点群をダウンサンプリング
- マスクの最小外接長方形を描画
- [draw-polygon-vertices.pdf](#)
- 座標系をクイック変換
- 対象物のインデックスリストをクイック作成
- 数値リストをクイック作成
- 位置姿勢をクイック作成
- 四元数をクイック作成
- ラベルリストをクイック作成
- 3Dベクトルをクイック作成
- 位置姿勢を基準点にクイック指向
- 2D方法による点群エッジを推定
- 3D方法による点群エッジを推定
- 画像の鮮明さを評価
- HDevEngineによる結果を評価
- JavaScriptエンジンによる結果を評価
- 深度画像の変動を評価
- 2D位置姿勢を評価
- 外部インターフェース
- 直方体内の点群を抽出
- 円柱以内の点群を抽出
- 3D ROI内の点群を抽出
- 深度画像の3D ROIにゼロ深度値の領域を抽出
- 穴埋め処理
- ラベルによってフィルタリング
- 位置姿勢によってマスクをフィルタリング
- 箱外の位置姿勢を除去

- フィルタリング
- 2Dコーナーを取得
- 位置姿勢とオフセットの対応関係を求める
- 2D平面で丸穴を検出
- 指定した内外層で2D輪郭を検出
- 円のフィッティング
- 直線のフィッティング
- 位置姿勢の座標軸方向を反転
- 実際のサイズをピクセルサイズに変換
- Cloud (XYZ-Normal)をCloud (XYZ-RGB)に変換
- Cloud (XYZ-RGB) をカラー画像に変換
- 深度画像を点群に変換
- 視差画像によって深度画像を生成
- NumberListをSize3DListに変換
- NumberListをVariantListに変換
- PoseListをMatrix4Dに変換
- PoseListをPoseListsに変換
- 四元数位置姿勢をオイラー角に変換
- Shape2DListをPose2DListに変換
- VariantをVariantListに変換
- 任意の変数からラベルに変換
- VariantListをNumberListに変換
- VariantListをVariantに変換
- 3D位置姿勢を2Dに変換
- 点群で仮想側壁を生成
- 円柱形の点群モデルを生成
- 基準位置姿勢を中心に回転した離散位置姿勢を生成
- 把持位置姿勢を生成
- 点群モデルを生成
- リング形状の点群を生成
- 位置姿勢のオフセットを生成
- 長方形の経路を生成

- 長方形候補を生成
- 指定したサイズの長方形のエッジテンプレートを生成
- 螺旋状経路を生成
- テスト画像を生成
- テスト点群を生成
- 輪郭から経路を生成
- 深度画像から経路を生成
- ジグザグ経路を生成
- 最初の画像を取得
- フランジ位置姿勢を取得
- 深度画像で最高領域を取得
- 最高層の点群を取得
- 最高層の点を取得
- 最高層の位置姿勢を取得
- 最高スコアの結果を取得
- リング形状の点群リストをフィルタリング
- データをグループ化
- 2D位置姿勢をグループ化
- ヒストグラムマッチング
- 画像の明るさと色バランサー
- 画像フィルタリング
- 画像調整
- 終了点を挿入して移動パラメータを送信
- 深度画像の3D ROI外の領域を無効に設定
- 位置姿勢を逆変換
- 四元数を逆変換
- 位置姿勢のZ値をしきい値と比較
- ラベル接続
- 正ポリゴンに沿った分布する位置姿勢を保持
- ラベルマッピング
- ラインスキャンレーザーカメラ
- 2Dテンプレートを読み込む

- 経路点の読み込みと変換
- 2D経路を読み込む
- 校正球を位置決めする
- 位置姿勢の軸の方向を基準点に指向
- 2Dテンプレートを作成
- 深度画像をカラー画像にマッピング
- 複数の把持位置姿勢にマッピング
- マスクによって画像領域を抽出
- マスククラスタリング
- マスクフィルタリング
- マスクをメッシュ化
- マスクの論理演算
- 線分間の角度を測定
- 円の測定
- 円から円までの距離を測定
- 円から線分までの距離を測定
- 点から円までの距離を測定
- 点から点までの距離を測定
- 点から線分までの距離を測定
- 線分間の距離を測定
- ギャップ幅を測定
- 点から点までの高さの差を測定
- 点から基準線までの高さの差を測定
- 点から平面までの高さの差を測定
- 最長の線分を測定
- 軸に平行な方向に沿った平面の高さの差を測定
- 測定結果を計算して表示
- 点群リストをマージ
- データをマージ
- 深度画像をマージ
- ラベルリストをマージ
- 線分リストをマージ

- 近似高度の点群をマージ
- 点群をマージ
- 位置姿勢リストをマージ
- マスク画像をマージ
- 画像のモフォロジー処理
- 点群を指定方向に沿って移動
- 位置姿勢をZ軸に沿って点群表面に移動
- 法線ベクトルの計算
- 通知
- 数値スケーリング
- 数値演算
- 光学文字認識
- 円柱の位置姿勢オフセット
- 正投影の逆変換
- 正投影
- パック
- パレット情報認識
- 経路計画
- サイクルトリガー
- 透視変換
- 対象物の把持範囲
- 配置する対象物の位置姿勢を計画
- 位置姿勢のある軸を指定された方向に向ける
- 点群クラスタリング
- 点群フィルタリング
- 点群形状検出器
- 点のフィルタリング
- 位置姿勢を一括調整
- 位置姿勢フィルタリング
- 対象物座標系で四元数ベクトルによって位置姿勢を回転
- 対象物座標系で行列によって位置姿勢を変換
- 統計データで位置姿勢の繰返し精度をチェック

- 把持位置姿勢を予測（複数タイプ）
- 把持位置姿勢を予測（V2）
- 出力
- ステップの組合せ
- 2D形状を処理
- 点を平面に投影
- 3D点群を2D画像に投影
- バーコードを読み取る
- 画像を読み取る（V2）
- 画像を読み取る
- 対象物の寸法を読み込む
- 点群を読み取る（V2）
- 点群を読み取る
- ファイルから位置姿勢を読み取る
- QRコードを読み取る
- STLファイルを読み取る
- 3D ROIの中心を読み取る
- 基準位置姿勢を記録して変換を計算
- 2D ROI画像のスケール復元
- 画像の座標系を補正
- リング形状の対象物の位置姿勢を補正
- 近すぎる位置姿勢を除去
- 点群の点を一部除去
- マスクのノイズを除去
- 重複対象物を除去（V2）
- 重複対象物を除去
- 重複ポリゴンを除去
- 点群の点を一部除去
- マスク以外のポリゴンを除去
- 指定した下付き文字に従って並べ替える
- インデックスによって要素を並べ替える
- データを繰り返して連結

- リスト内の要素を置き換える
- リストを逆順に並べる
- 指定軸が基準方向と最小角度になるように位置姿勢を調整
- 指定した位置姿勢によって画像を回転
- 指定した軸と角度を中心に位置姿勢を回転
- 指定された軸を中心に位置姿勢を回転
- ローカル座標系で位置姿勢を回転
- 位置姿勢を目標方向に回転（制限なし）
- 対称性制約で位置姿勢を指定方向に回転
- 位置姿勢の軸を指定方向に回転
- 画像保存
- 位置姿勢の近傍領域を3D ROIとして保存
- 結果をXMLファイルに保存
- 結果をファイルに保存
- 経路点を保存
- ステップパラメータをファイルに保存
- 2D ROI内の画像をスケーリング
- 深度画像分割
- 点群をMech-Vizに送信
- 位置姿勢の四元数を設定
- 位置姿勢の並進ベクトルを設定
- 画像表示
- 点群と位置姿勢を表示
- 深度画像を平滑化処理
- 経路を平滑化
- ソートしてインデックスリストを出力
- ソートと階層化
- 点群をソート
- 入力したスコアで位置姿勢をソート
- 位置姿勢のXYZ値で位置姿勢をソート
- 2D位置姿勢をソート
- 3D位置姿勢をソート

- リング形状の画像領域をまっすぐにする
- しきい値によって数値を分類
- 実数減算
- 一部領域で低いスコアの位置姿勢を除去(NMS)
- 対象物の回転対称性
- 2Dテンプレートマッチング
- テスト
- 経路点のマッチング
- 画像変換
- 平面の点群を指定平面に合わせる
- 点群変換（直行ロボット）
- 点群変換
- 位置姿勢を変換（直行ロボット）
- 位置姿勢を変換
- 基準方向に沿って位置姿勢を並進
- 位置姿勢を指定方向に移動
- 指定した方向と距離によって位置姿勢を並進
- 3Dベクトルで位置姿勢を並進
- ローカル座標系で位置姿勢を並進
- ブール値によるトリガー制御
- トリガー
- 出力の数を制限
- 位置姿勢の数を制限
- データをアンパックしてマージ
- データをアンパック
- マスクによって2D位置姿勢を検証
- カメラのパラメータ補正を検証して計算
- 箱の寸法が有効であるかどうかを検証
- 箱形状の対象物のマスクを検証
- 位置姿勢が3D ROI内にあるかどうかを検証
- 必要なラベルであるかどうかを検証
- マスクに2D位置姿勢があるかどうかを検証

- 点群が要求を満たすかどうかを検証
- 基準方向との角度によって位置姿勢が有効かどうかを検証
- 位置姿勢が遮蔽されているかを検証
- 法線ベクトルの偏差が大きい領域を抽出
- 3Dベクトル演算
- 把持位置姿勢を検証
- 範囲内の深度を可視化
- 画像で情報を可視化

ステップ組合せ

Mech-Visionでは、ステップ組合せはステップライブラリの「カスタマイズ」グループにあります。クリックして **ステップ組合せの詳細な説明** をお読みください。

- 大型部品の測定
- ソート
- 3D位置姿勢推定
- マスクが対応するカラー画像を取得
- 最高層のカラー画像を取得
- 点の数が制限を超えた点群を除外
- ROI以外の位置姿勢を除去
- 点群前処理
- 画像のデータとパラメータを保存

4.3.1. 2Dマッチング

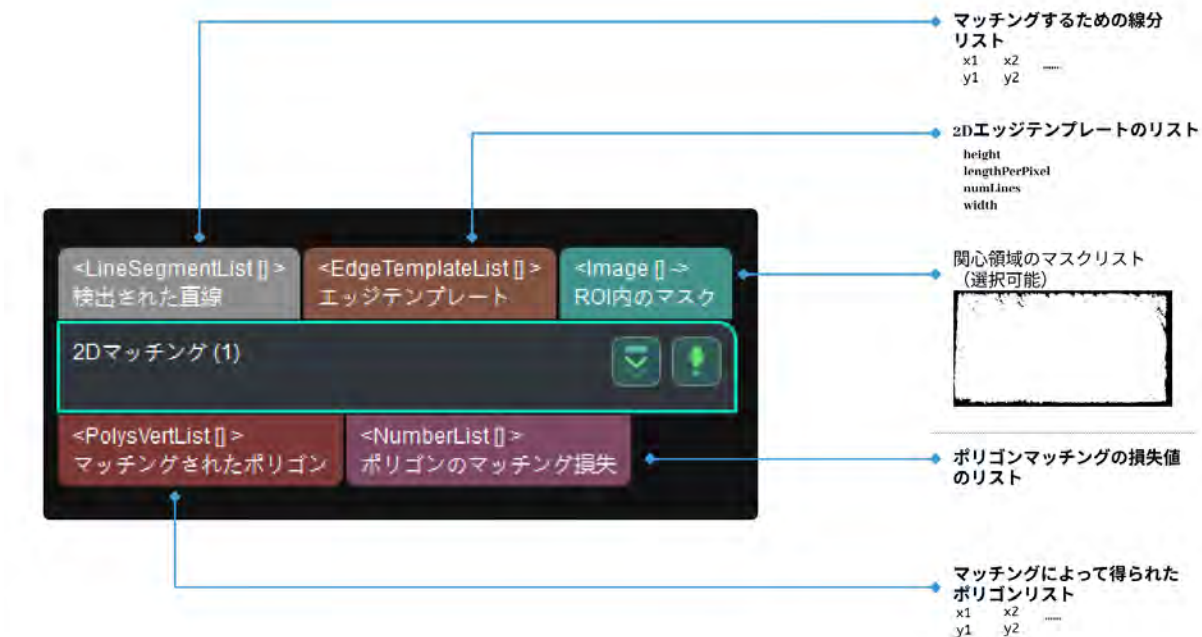
機能

2Dマッチングアルゴリズムによって、モデルを使用して指定形状の点群をマッチングします。

使用シーン

2D画像でポリゴン対象物を位置決めし、検出されたポリゴン特徴をその後のポリゴンの3D位置姿勢を計算するために提供します。通常、ステップ **マスク以外のポリゴンを除去** と併用されます。

入力と出力



パラメータの説明

テンプレート設定

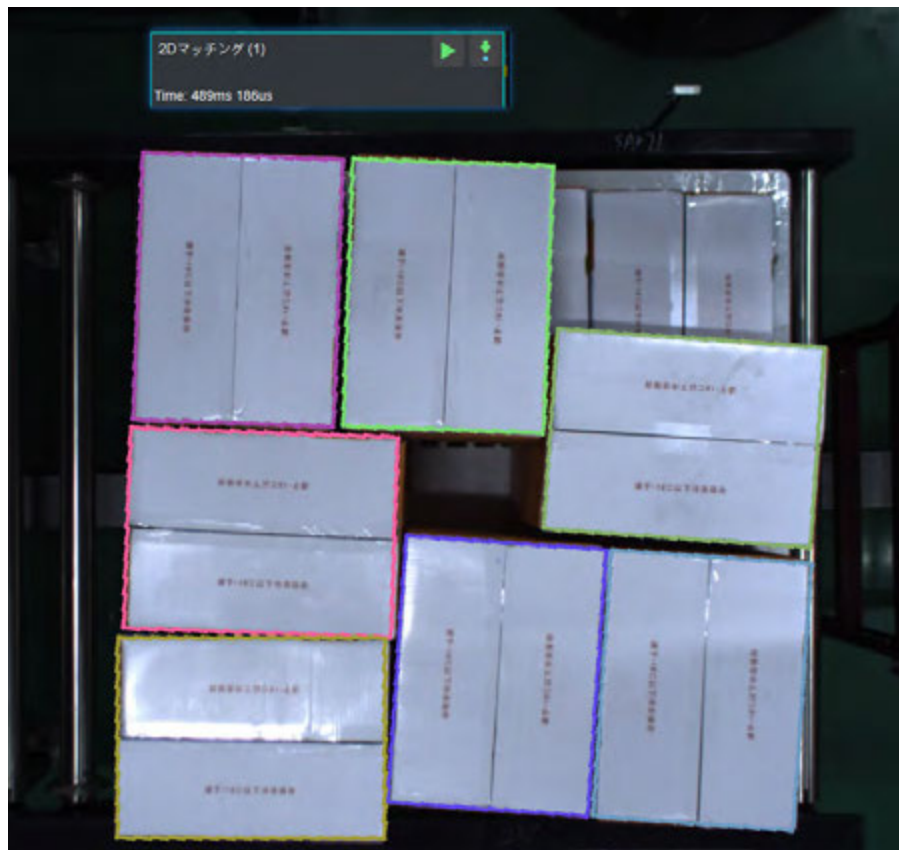
輪郭線角度定量化 (1-360)

初期値：60

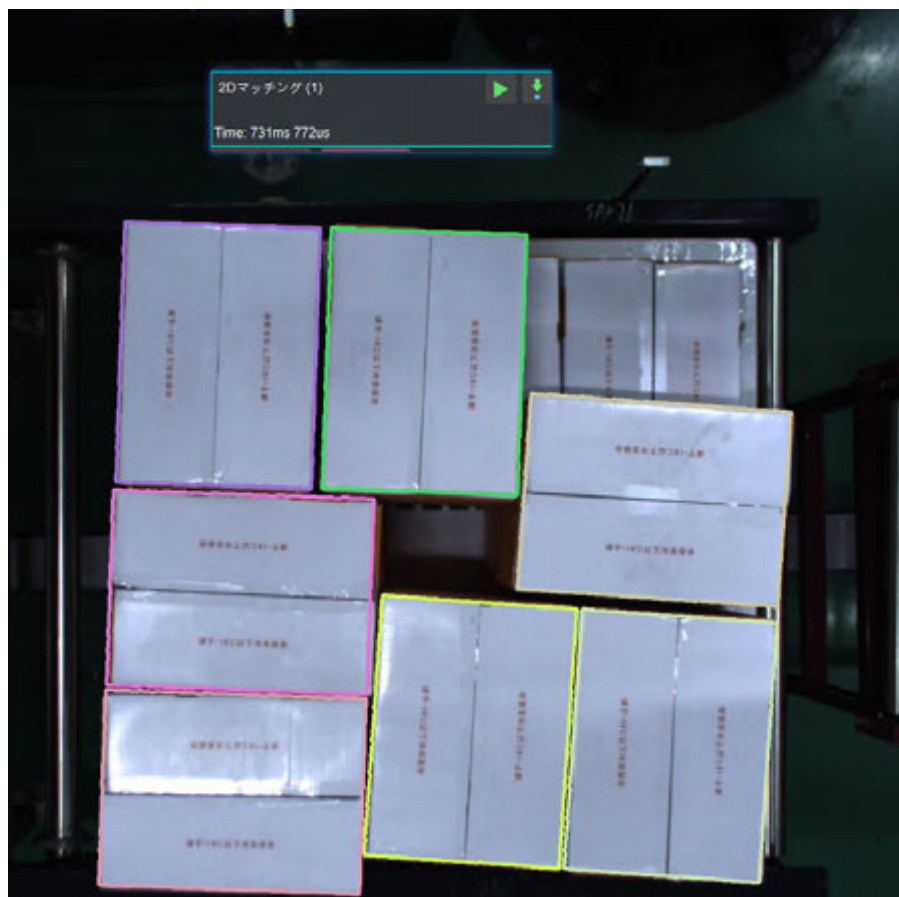
調整説明：輪郭線の方向を定量化します。例えば、長方形は180度対称なので、180度が定量化され、入力されたパラメータはコピーの数です。パラメータが90に設定されている場合、「 $180^\circ/90=2$ 」によって得られたコピー数は2です。定量化されたコピーが多いほど、エッジは細くなりますが、時間がかかります。

調整の例：

パラメータ値が30に設定した場合、ステップの実行時間と定量化の結果を下図に示します。



パラメータ値が90に設定した場合、ステップの実行時間と定量化の結果を下図に示します。



角度損失の重み

初期値：0.1

調整説明：値が大きいほど結果は精確になりますが、時間がかかります。

ベーススケール

初期値：1.1

調整説明：テンプレートは、「基本アスペクト比」、「スケール指数」によって変更されます。「スケール指数」は、「最小スケール指数」から「最大スケール指数」までの整数です。値が大きいほど、正確にマッチングする可能性が高くなりますが、時間がかかります。

最小スケール指数

初期値：0

調整説明：-3、-2、-1、0は参考値となります。値が小さいほど対象物に正しくマッチングする可能性が高くなりますが、時間がかかります。

最大スケール指数

初期値：0

調整説明：0、1、2、3は参考値となります。値が大きいほど対象物に正しくマッチングする可能性が高くなりますが、時間がかかります。

基本アスペクト比

初期値：1.1

調整説明：テンプレートは、「基本アスペクト比」、「アスペクト指数」によって変更されます。「アスペクト指数」は、「最小アスペクト比」から「最大アスペクト比」までの整数です。値が大きいほど、正確にマッチングする可能性が高くなりますが、時間がかかります。

最小アスペクト指数

初期値：0

調整説明：-3、-2、-1、0は参考値となります。値が小さいほど対象物に正しくマッチングする可能性が高くなりますが、時間がかかります。

最大アスペクト指数

初期値：0

調整説明：0、1、2、3は参考値となります。値が大きいほど対象物に正しくマッチングする可能性が高くなりますが、時間がかかります。

候補検索設定

対象物の大まかな姿勢 (0-360)

初期値：0°

調整説明：シーンの物体の大まかな方向（単位は°）です。長方形の場合、横方向を0°方向、縦方向を90°方向と定義します。

対象物の姿勢の変化範囲 (0-180)

初期値：90°

調整説明：シーンの物体の位置姿勢変化範囲（単位は°）です。

テンプレート検索の並進ステップサイズ

初期値：5.000 mm

調整説明：値が小さいほど結果は精確になりますが、時間がかかります。単位はミリメートル（mm）です。

テンプレート検索の回転ステップサイズ (0-180)

初期値：1.5000°

調整説明：値が小さいほど結果は精確になりますが、時間がかかります。単位は度（°）です。

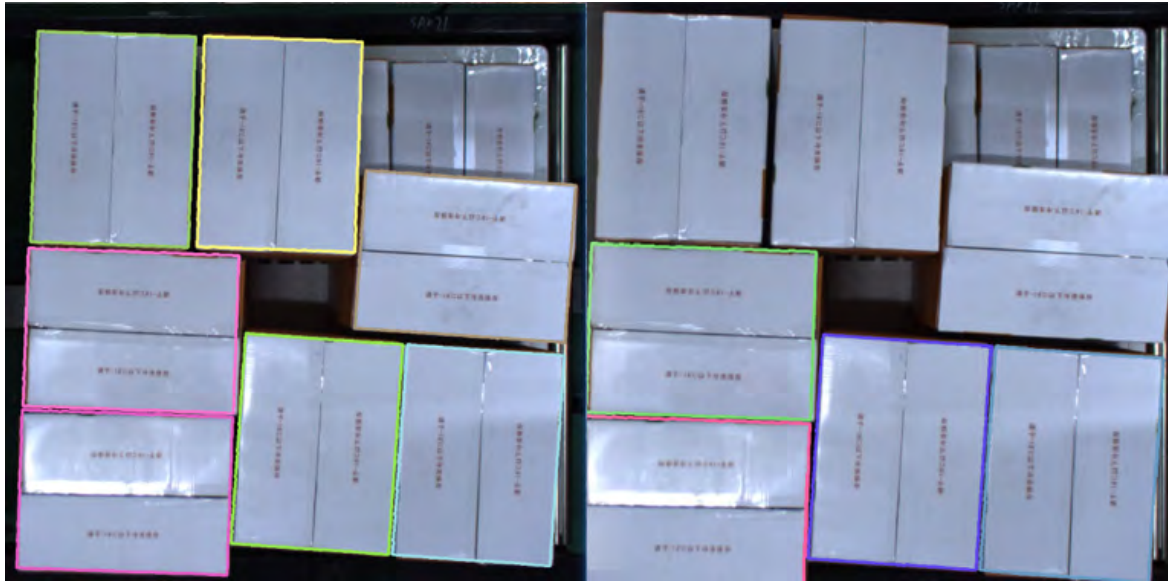
しきい値設定

平均距離誤差の最大しきい値

初期値：8.000 mm

調整説明：各テンプレートマッチングの平均距離の最大誤差です。単位はミリメートル（mm）です。値が高いほど、結果はより正確になります。

調整の例：下図に示すように、左側が初期値での出力結果、右側がパラメータを0.002に設定した場合の出力結果です。

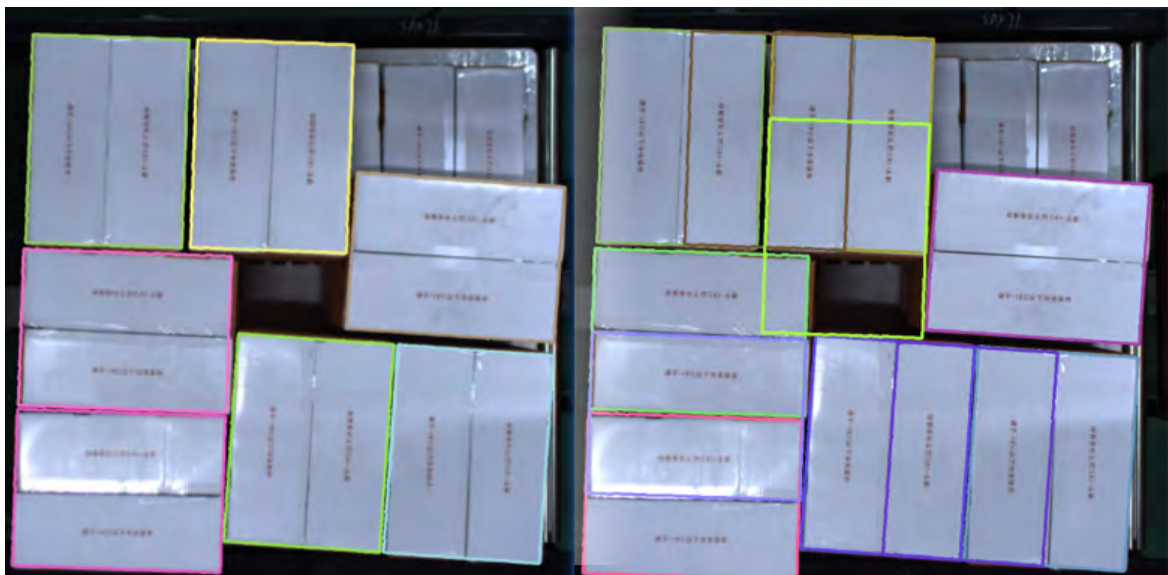


重複しきい値 (0–1.0)

初期値 : 0.5

調整説明：結果の重複度がこのしきい値より大きい場合は削除されます。値が大きいくほど、より多くの重複結果が出力される傾向があります。

調整の例：下図に示すように、左側が0.2に設定した場合の出力結果であり、右側が0.5に設定した場合の出力結果です。



4.3.2. 2D位置姿勢を3D位置姿勢に変換 (正投影に基づく)



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社 (docs@mech-mind.net) までお問い合わせください。

機能

正投影に基づいて生成された2D位置姿勢を、3D位置姿勢に変換します。

使用シーン

通常は計測シーンで、2D画像の位置姿勢を3D位置姿勢に変換するために使用されます。その中、2D画像はステップ「正投影」からのものです。

入力と出力

● 入力ポート：

1. 正投影に基づいて生成された2D位置姿勢。
2. 3D正投影から2Dに投影したマスク。
3. 3次元座標軸 (X, Y, Z) 上の点群内のすべての点の最小値。
4. 正投影のスケーリング係数。
5. 正投影によって生成されたマスク境界の幅。
6. 正投影後のマスク内の対象物が対応する深度画像。

● 出力ポート：

1. 2D位置姿勢から変換された3D位置姿勢。

4.3.3. 2D形状マッピング



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社 (docs@mech-mind.net) までお問い合わせください。

機能

2D形状マッチングの方法により、画像内の対象物を位置決めします。

使用シーン

汎用の2D形状マッチングのステップです。特定の使用シーンはありません。

これは古いバージョンのステップであり、メンテナンスが停止されていますので、新しいバージョンのステップ [\[vision-steps:2d-matching:::2d-matching\]](#) を使用してください。

入力と出力

● 入力ポート：

1. シーンの画像（その中の対象物はテンプレートによってマッチングされます）。
2. 対象物のテンプレート画像。

- 出力ポート：

1. マッチング結果のマスク。
2. マッチング結果のスコア。

4.3.4. 3D位置姿勢低精度推定（マルチモデル）

機能

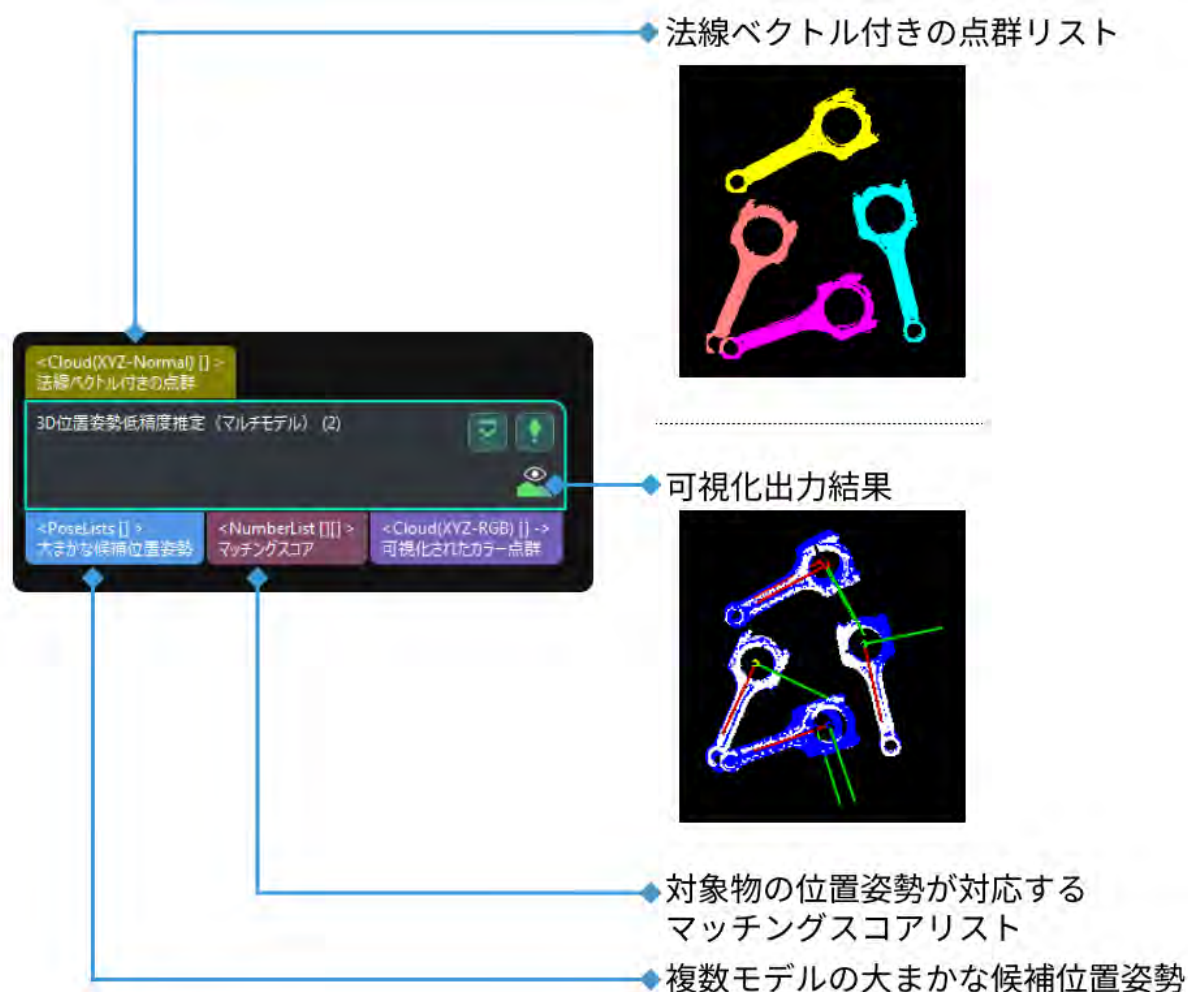
複数のモデルを使用してシーン内の対象物に対して大まかなマッチングを行い、対象物の大まかな候補位置姿勢を出力します。

使用シーン

複数のモデルを使用してシーン内の対象物の大まかな位置姿勢を計算します。これは [3D位置姿勢低精度推定](#) を拡張したステップで、パラメータの調整方法はほぼ同じです。

複数のモデル点群を入力してマッチングを実行できます。マルチモデルのシーンでは、部品の種類を確認または分類するために使用されます。通常、より正確な位置姿勢を取得するために、このステップの後に [3D位置姿勢高精度推定（マルチモデル）](#) を繋ぎます。

入力と出力



パラメータ説明

モデルと把持位置姿勢の設定

モデルファイルと把持位置ファイルのパスを設定します。

モデルファイル (必須)

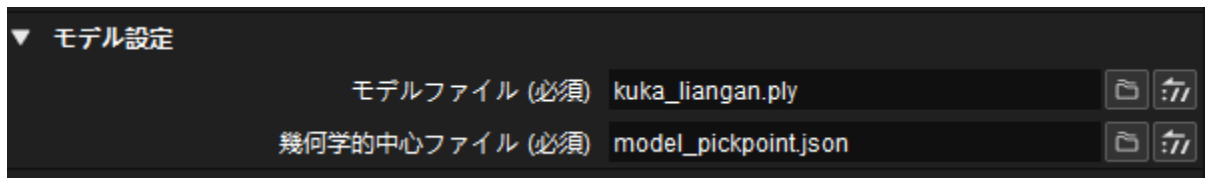
初期値：model.ply

調整説明：ply形式のモデル点群です。複数のファイルパスを入力することができます。ファイル名の上にセミコロンを入力してください。

幾何学的中心ファイル (必須)

調整説明：json形式の幾何学的中心ファイルです。複数のファイルパスを入力することができます。ファイル名の上にセミコロンを入力してください。

調整の例：各パラメータにファイルを同じ順序で入力してください。 **モデルファイル** と **幾何学的中心** に同じファイルパスを入力してください。下図に示すように、ファイル名の上に `` ; `` を入力してください。



点群の点が位置する平面の方向を計算

点が位置する平面の方向の計算方法

初期値：入力点群の法線を使用（Origin）

オプション	調整説明
入力点群の法線を使用（Origin）	入力された点群の元の法線ベクトルを使用します。
StandardMode	CPUを使用して入力された点群の法線ベクトルを再計算します。モデルに法線ベクトルがない場合に使用することをお勧めします。目標点の付近に目標点にもっとも近いk個の点を検索して、PCA（主成分分析）を使用して最小の固有ベクトルを取得してこの点の法線ベクトルとします。
EdgeTangent	入力されたエッジ点群の接線を計算し、点群の接線の方向を法線ベクトルとします。輪郭が鏡像となる異なる物体を識別でき、平面形状の物体のエッジ点群をマッチングする場合に使用することをお勧めします。
EdgeNormal	入力されたエッジ点群の法線ベクトルを計算し、この点の接線の方向を法線ベクトルとします。平面形状の物体のエッジ点群をマッチングする場合に使用することをお勧めします。



EdgeTangent または **EdgeNormal** を使用するとき、各エッジ点群に複数の物体がないこと、つまり各物体の点群がすでに分割されたことを確認してください。

選択された隣接点の数

初期値：10

調整説明：このパラメータは、計算する点の方向に沿う隣接点の数を計算するために使用されます。つまり、**StandardMode** のkの値を計算します。

アルゴリズムタイプ

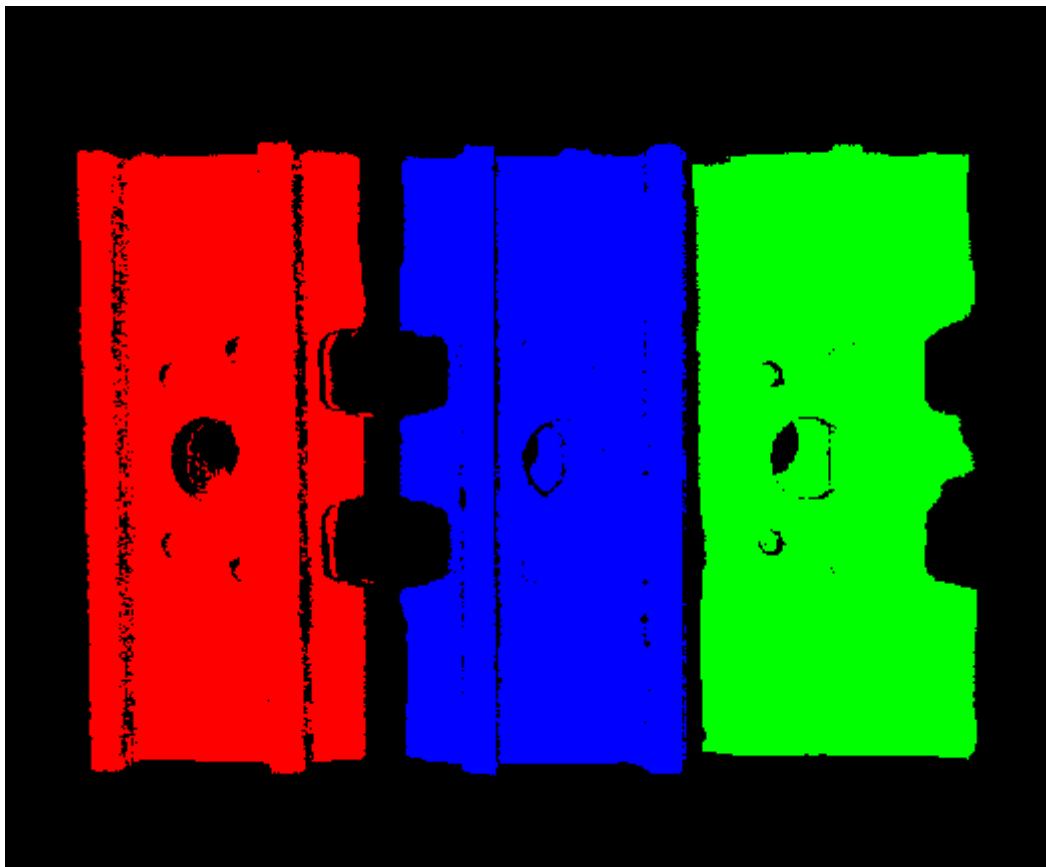
初期値：SurfaceMatchingEasyMode

オプション：SurfaceMatchingEasyMode、SurfaceMatching

調整説明：二つのアルゴリズムのタイプがあります。結果可視化の設定を、どちらのタイプでも調整可能なパラメータに設定します。アルゴリズムのタイプの調整は下図に示すように、正面・裏面モデルを使用してマッチングします。まず、**SurfaceMatchingEasyMode** アルゴリズムの調整可能なパラメータについて説明します。

SurfaceMatchingEasyMode アルゴリズム：調整可能なパラメータは速度制御と出力設定です。

SurfaceMatching アルゴリズム：調整可能なパラメータはサンプリング設定、投票設定、クラスタリング設定、位置姿勢検証設定です。



SurfaceMatchingEasyModeパラメータの説明

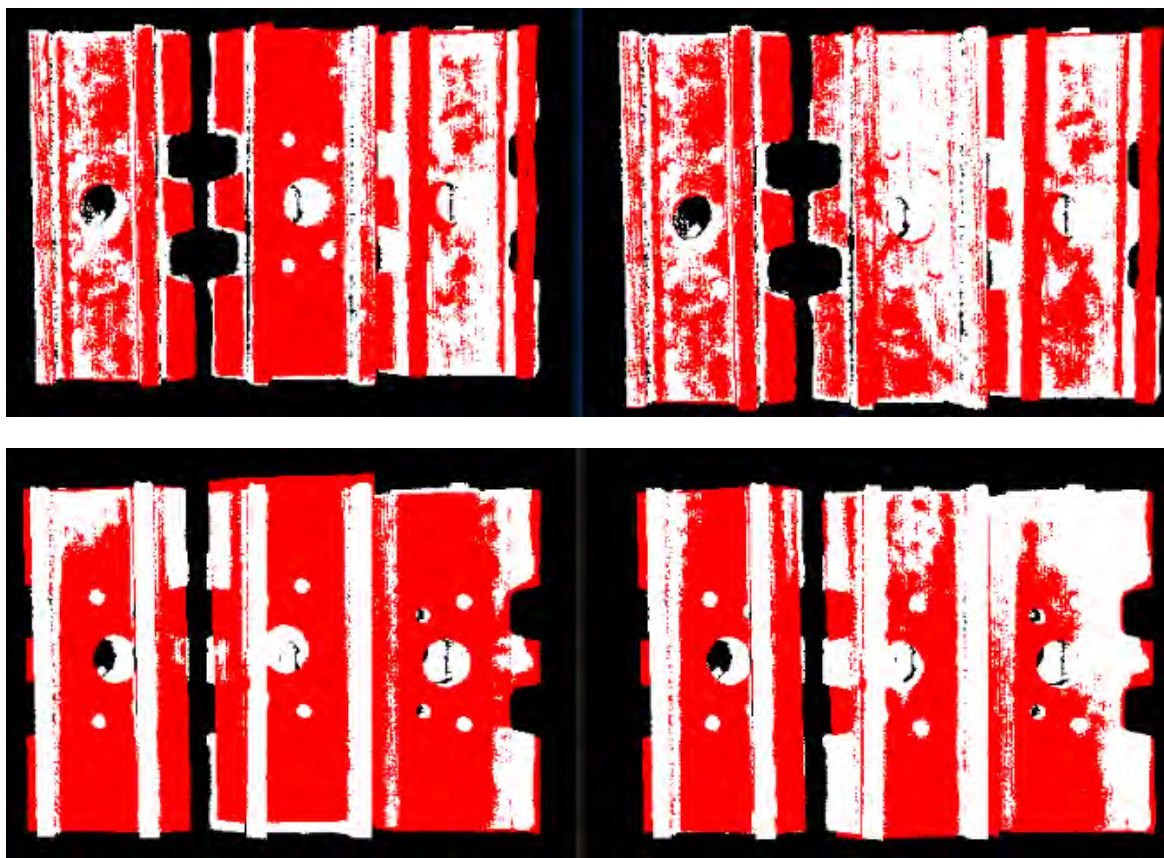
速度制御

メイン速度制御パラメータ

初期値：2

調整説明：このパラメータは、アルゴリズムの速度を調整します。大きくすればアルゴリズムの速度は速くなりますが、マッチングの精度は低下します。この効果は **サブ速度制御パラメータ** より強いです。有効な範囲は1~6です。

調整の例：下図のように示します。左側は2（初期値）にした結果で、右側は6にした結果です。このパラメータを大きくすると、マッチング精度が低下します。

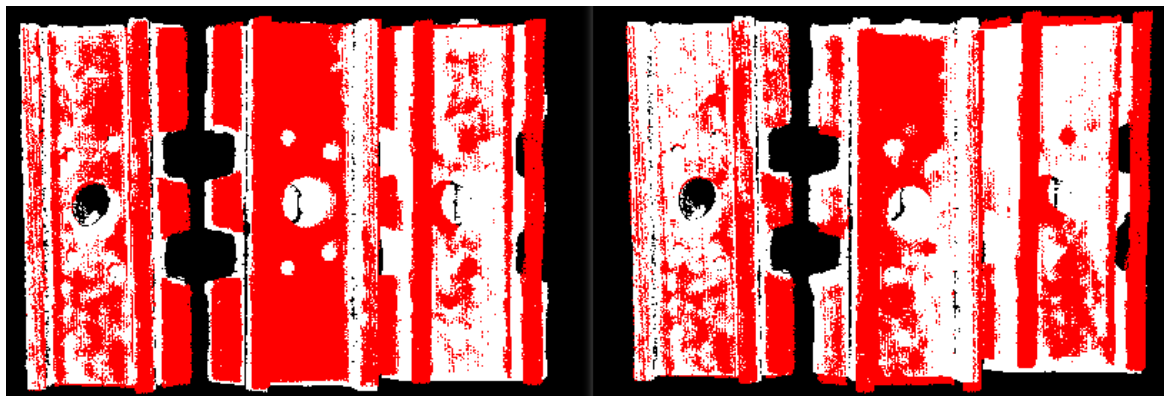


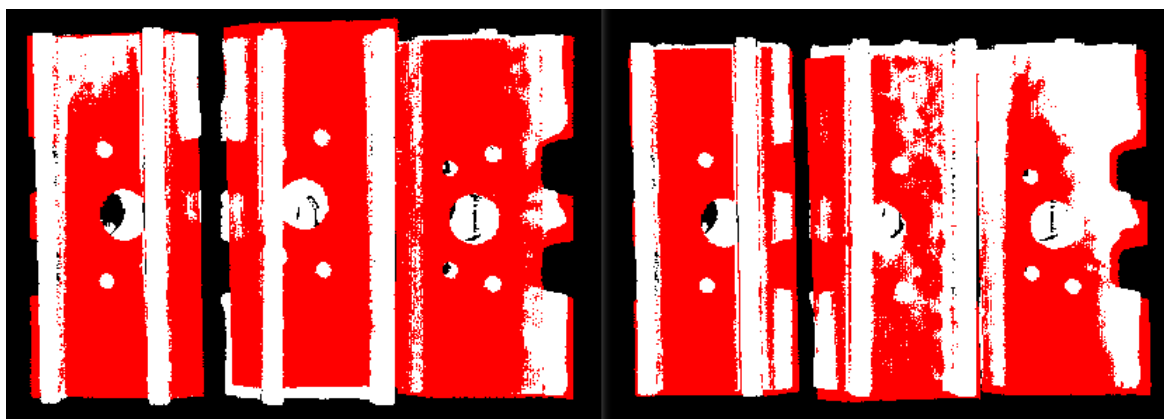
サブ速度制御パラメータ

初期値：10

調整説明：このパラメータは、アルゴリズムの速度を調整します。大きくすればアルゴリズムの速度は速くなりますが、マッチングの精度は低下します。この効果は **メイン速度制御パラメータ** より弱いです。有効な範囲は1~20です。

調整の例：下図のように示します。左側は10（初期値）にした結果で、右側は15にした結果です。値を大きくしたらマッチングの精度が低下しましたが、その効果はメイン速度制御パラメータより弱いです。





出力設定

単一点群の出力結果の数

初期値：3

調整説明：各点群のマッチングに出力される結果の数を予測します。値が大きいほど出力される結果は多くなります。

調整の例：対比の結果は下図に示します。左側はパラメータを1にした結果で、右側は3にした結果です。



SurfaceMatchingパラメータの説明

サンプリング設定

自動ダウンサンプリング

初期値：チェックを入れる

調整説明：このパラメータは、自動ダウンサンプリングを使用するかどうかを選択するために使用されます。チェックを入れれば、サンプリングされたモデルの期待点数に基づいてモデル点群のダウンサンプリングの間隔を自動的に調整します。

サンプリングされたモデルの期待点数

初期値：1000

調整説明：このパラメータは、サンプリングする点群の点数を調整するために使用されます。**自動ダウンサンプリング** にチェックを入れた場合に有効になり、取得する点群の点数はこの値に近いです。この値が小さいほど、サンプリングする点群の点数は少なくなり、位置姿勢推定の精度も低下します。

モデルサンプリングの最大点数

初期値：4000

調整説明：ダウンサンプリングされたモデル点群の点数の最大値を調整します。すなわち、ダウンサンプリングされたモデル点群の点の数の上限を設定します。マッチングの効果は良くない場合にこの値を大きくしてください。マッチングの速度を上げたい場合に小さくしてください。

シーンサンプリング最大点数

初期値：3000

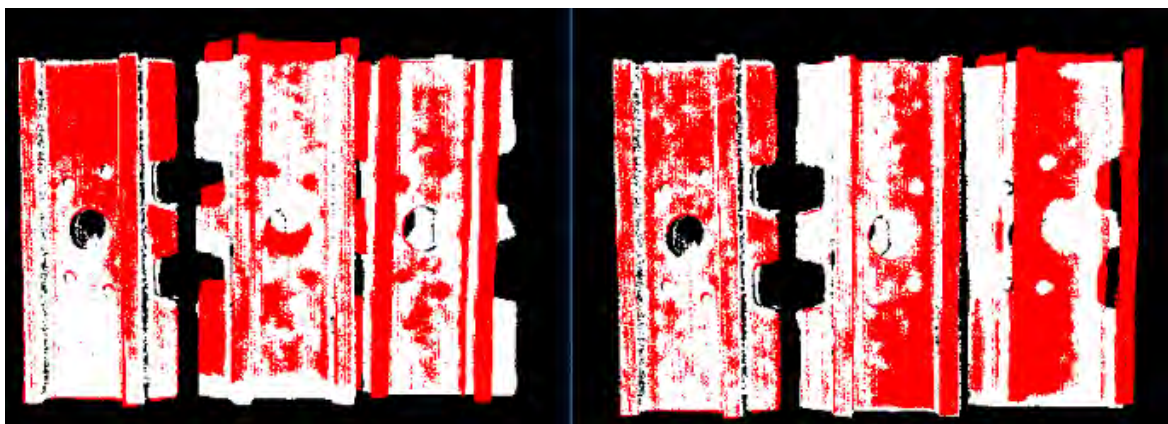
調整説明：ダウンサンプリングされたシーン点群の点の最大値を設定します。すなわち、ダウンサンプリングされたシーン点群の点の数の上限を設定します。マッチングの効果は良くない場合にこの値を大きくしてください。マッチングの速度を上げたい場合に小さくしてください。

サンプリング間隔

初期値：0.01

調整説明：ダウンサンプリングする点の間の最大距離を調整します。mを単位とします。モデル点群のダウンサンプリング間隔が最小サンプリング間隔より小さい場合に、最小サンプリング間隔を実際のサンプリング間隔とします。この値が大きいほど、サンプリング後に計算に使用する点群の数は少なくなり、マッチングの精度は低下し、アルゴリズムの実行時間は短くなります。

調整の例：下図のように示します。左側は0.01（初期値）にした結果で、右側は0.02にした結果です。



最小サンプリング間隔

初期値：0.003

調整説明：このパラメータは、サンプリング間隔（メートル単位）を計算するために使用されます。**自動ダウンサンプリング**にチェックを入れた場合に有効になります。計算したサンプリング間隔がこの値より小さい場合に、この値を実際のサンプリング間隔とします。

投票設定

距離の定量化

初期値：1

調整説明：点と点との間の距離を定量化します。**距離間隔 = 距離定量化 × サンプリング間隔**。このパラメータを大きくしたら間隔も大きくなり、マッチングの精度も低下します。

角度の定量化

初期値：60

調整説明：点の法線の角度を定量化します。**角度の間隔 = 2×3.14 / 角度の定量化**。このパラメータを大きくしたら、角度の間隔は大きくなり、マッチングの精度は低下します。

最大投票率

初期値：0.8

調整説明：投票の数が最大投票数に占める比率のしきい値を設定します。前のステップで各位置姿勢が対応する投票数を取得し、最大投票数にこのパラメータの値をかけるとしきい値を取得します。投票数がこのしきい値より大きい位置姿勢が保持されてクラスタリングされます。この値が小さいほど、正確なマッチング結果を得る確率も高くなりますが、実行時間も長くなります。このパラメータの有効範囲は0~1です。

参考点のサンプリングステップサイズ

初期値：5

調整説明：参考点のサンプリングステップサイズを調整します。点群では、このステップサイズを間隔としてサンプリングします。この値が大きいほどサンプリングする点は少なくなり、実行の速度は早くなりますが、マッチングの精度は低下します。

参考される点のサンプリングステップサイズ

初期値：1

調整説明：参考される点のサンプリングステップサイズを調整します。点群では、このステップサイズを間隔としてサンプリングします。この値が大きいほどサンプリングする点は少なくなり、実行の速度は早くなりますが、マッチングの精度は低下します。

- **参考点**と**参考される点**とはペアポイントになります。サンプリング間隔が大きいほど参考点と参考された点は少なくなり、ペアポイントも少なくなり、実行速度は速くなります。

す。

- **参考点** とは、マッチングモデルにあるサンプリングする点です。**参考される点** とは、マッチングモデルにないサンプリングする点です。

クラスタリング設定

クラスター比

初期値：0.1

調整説明：クラスタリングに使用する位置姿勢が計算によって取得した全ての位置姿勢に占める比率を調整します。いかなる位置姿勢も計算によってスコアが得られ、スコアによって位置姿勢をソートします。このパラメータは、クラスタリングに使用する位置姿勢の比率を決めます。初期値は0.1で、つまり最初の10%の位置姿勢をクラスタリングに使用します。この値が大きいほど、正確なマッチングの結果を得る確率は高くなりますが、実行時間も長くなります。

角度差しきい値

初期値：15

調整説明：このパラメータはクラスタリングのプロセスの角度増量を調整します。最終的な計算結果には、一つの対象物に複数の位置姿勢が計算されることがあります。この場合、パラメータが非常に近い位置姿勢をマージするとき、このパラメータは角度の増量を決めます。このパラメータが大きいほど、角度の差が大きい位置姿勢は最終的な結果にマージされ、マッチング精度も低下します。

距離差しきい値

初期値：0.02

調整説明：このパラメータはクラスタリングのプロセスの距離増量を調整します。最終的な計算結果には、一つの対象物に複数の位置姿勢が計算されることがあります。この場合、パラメータが非常に近い位置姿勢をマージするとき、このパラメータは距離の増量を決めます。このパラメータが大きいほど、距離の差が大きい位置姿勢は最終的な結果にマージされ、マッチング精度も低下します。

スコアが最も高いN個の結果を出力

初期値：5

調整説明：このパラメータはクラスタリングして取得した複数のマッチング結果にスコアが最も高いN個の結果を最終的な結果として出力します。

位置姿勢検証の設定

位置姿勢検証を使用

初期値：チェックを入れる

調整説明：このパラメータは位置姿勢検証を使用するかを選択するために使用されます。こ

のパラメータにチェックを入れれば、クラスタリングのパラメータは全部無効になります。姿勢検証とクラスタリングは、最終的なマッチング結果の検証とフィルタリングのための異なる方法であり、同時に使用することはできません。

隣接点の検索半径

初期値：1

調整説明：このパラメータは位置姿勢検証のプロセスに検証するエリアのサイズを調整します。ボクセルを単位とします。この値を大きくしたら、位置姿勢検証のエリアは広くなり、最終的な結果を検証するために使用する点は多くなるのでマッチングの精度は低下します。

ボクセルの長さ

初期値：3

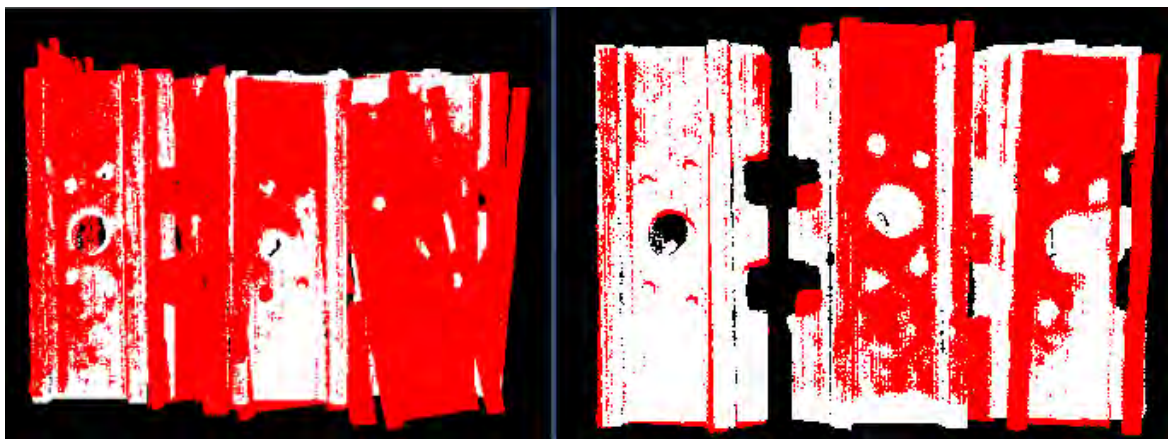
調整説明：点群がある空間を3Dグリッドにします。このパラメータは3Dグリッドの最小単位（mm）になります。この値を大きくしたら、選択する範囲は広くなり、アルゴリズムの実行速度も速くなりますが、マッチングの精度は低下します。

単一点群の出力結果の数

初期値：3

調整説明：SurfaceMatchingアルゴリズムを実行した場合、このパラメータの機能はSurfaceMatchingEasyModeと同じです。

調整の例：下図に示すように、左側は3（初期値）にした結果で、右側は初期値1にした結果です。



結果可視化の設定

ダウンサンプリングされたモデルの点群を表示

初期値：チェックを入れない

調整説明：このパラメータはダウンサンプリングされたモデルの点群を表示します。

サンプリングされたシーンの点群を表示

初期値：チェックを入れない

調整説明：このパラメータはダウンサンプリングされたシーンの点群を表示します。

マッチング結果を表示

初期値：チェックを入れる

調整説明：このパラメータはマッチングされたモデルとシーンの点群を表示します。

4.3.5. 3D位置姿勢低精度推定（V2）

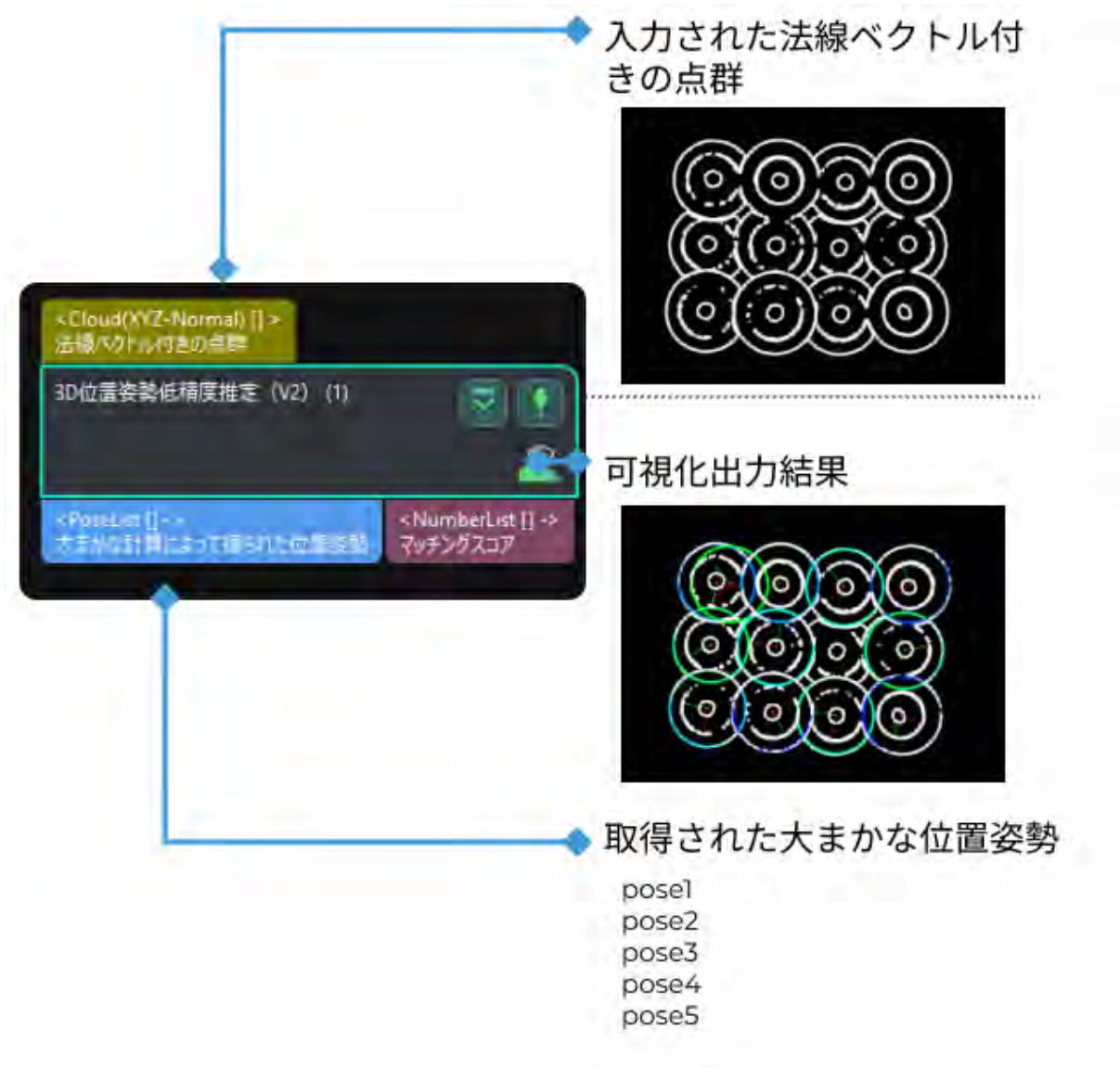
機能

点群モデルは元の点群と大まかなマッチングを行い、対象物の大まかな候補位置姿勢が出力されます。

使用シーン

通常、シーンの点群に対象物を見つけてその大まかな候補位置姿勢を取得するために使用されます。このステップは通常、ステップ [3D位置姿勢高精度推定](#) と併用されます。

入力と出力



パラメータの説明

パラメータのデバッグレベル

パラメータ説明：このパラメータは、ステップパラメータの調整レベルを選択するために使用されます。

オプション： [基本設定](#)、[詳細設定](#)。

初期値：基本設定

基本設定

モデル設定

モデル選択

パラメータ説明：このパラメータは、モデルライブラリ（[プロジェクトフォル](#)

ダ/resource/3d_matching) から点群モデルファイルを選択するために使用されます。

調整説明： [マッチングモデル・把持位置姿勢エディタ](#) で点群モデルを作成したら、このパラメータの右側にあるドロップダウンボタンをクリックすると、点群モデルファイルを素早く選択・切り替えできます。

マッチングモード

パラメータ説明：このパラメータは、マッチングモードを選択するために使用されます。

オプション：エッジマッチング、サーフェスマッチング。

初期値：エッジマッチング。

調整説明：点群モデルに応じて選択する必要があります。点群モデルがワークのエッジモデルとして作成する場合、 **エッジマッチング** を選択してください。点群モデルがワークのサーフェスモデルとして作成する場合、 **サーフェスマッチング** を選択してください。

前処理設定

サンプリングされたモデルの期待点数

パラメータ説明：このパラメータは、ダウンサンプリング後の点群の点数を設定するために使用されます。値が小さいほど、サンプリングされた点群内の点が少なくなり、マッチングの精度が低くなります。

初期値：200

調整アドバイス：実際の状況に応じて設定してください。調整効果を [調整の例](#) に示します。

サンプリング後の入力点群の点の最大数

パラメータ説明：このパラメータは、ダウンサンプリング後のシーンにおける点群の最大点数を設定するために使用されます。ダウンサンプリング後のシーン点群の点数がこの値より多い場合、点群は無視され、空のデータが出力されます。

初期値：100000

調整アドバイス：実際の状況に応じて設定してください。調整効果を [調整の例](#) に示します。

位置姿勢検証設定

単一点群の結果の期待数

パラメータ説明：このパラメータは、各入力点群のマッチング位置姿勢の期待数を示します。

初期値：3

調整アドバイス：実際の状況に応じて設定してください。調整効果を [調整の例](#) に示します。

詳細設定

モデル設定


モデル選択

パラメータ説明：このパラメータは、モデルライブラリ（**プロジェクトフォルダ/resource/3d_matching**）から点群モデルファイルを選択するために使用されます。

調整説明：**マッチングモデル・把持位置姿勢エディタ**で点群モデルを作成したら、このパラメータの右側にあるドロップダウンボタンをクリックすると、点群モデルファイルを素早く選択・切り替えできます。

モデルファイル

パラメータ説明：このパラメータは、点群モデルファイルを選択するために使用されます。

調整説明：をクリックして、ply形式のモデル点群ファイルを選択します。

幾何学的中心ファイル

パラメータ説明：このパラメータは、幾何学的中心ファイルを選択するために使用されます。

調整説明：をクリックして、json形式の幾何学的中心ファイルを選択します。

マッチングモード

パラメータ説明：このパラメータは、マッチングモードを選択するために使用されます。

オプション：エッジマッチング、サーフェスマッチング。

初期値：エッジマッチング。

調整説明：点群モデルに応じて選択する必要があります。点群モデルがワークのエッジモデルとして作成する場合、**エッジマッチング**を選択してください。点群モデルがワークのサーフェスモデルとして作成する場合、**サーフェスマッチング**を選択してください。

前処理設定

サンプリングされたモデルの期待点数

パラメータ説明：このパラメータは、ダウンサンプリング後の点群の点数を設定するために使用されます。値が小さいほど、サンプリングされた点群内の点が少なくなり、マッチングの精度が低くなります。

設定可能な範囲：20~3000

初期値：200

調整アドバイス：実際の状況に応じて設定してください。調整効果を **調整の例** に示します。

サンプリング後の入力点群の点の最大数

パラメータ説明：このパラメータは、ダウンサンプリング後のシーンにおける点群の最大点数を設定するために使用されます。ダウンサンプリング後のシーン点群の点数がこの値より多い場合、点群は無視され、空のデータが出力されます。

設定可能な範囲：100~100000

初期値：100000

調整アドバイス：実際の状況に応じて設定してください。調整効果を [調整の例](#) に示します。

近傍点の数

パラメータ説明：このパラメータは、点の方向における近傍点の数を設定するために使用されます。

設定可能な範囲：1~100

初期値：30

投票設定

単一特徴に含むポイントペアの最大数

パラメータ説明：このパラメータは、モデル分析に各特徴に含むポイントペアの最大数を示します。値が小さいほど実行は速くなりますが、精度は低くなります。

初期値：50

距離の定量化

パラメータ説明：このパラメータは、点間の距離を定量化するために使用されます（距離間隔 = 距離定量化 × サンプリング間隔）。値が大きいほど距離間隔が大きくなり、マッチング精度が低下します。

初期値：1.0000

角度の定量化

パラメータ説明：このパラメータは、点の特徴法線ベクトル間の角度を定量化するために使用されます（角度間隔 = $360^\circ / \text{角度定量化}$ ）。値が大きいほど角度間隔が小さくなり、マッチング精度が高くなりますが、より高精度の点群が必要になります。

初期値：60

最大投票率

パラメータ説明：スコアは「投票での最高スコア × 最大投票率」より高い位置姿勢は位置姿勢の検証に使用されます。値が低いほど、位置姿勢の検証に使用される位置姿勢が多くなり、正確なマッチングを見つける可能性が高くなりますが、実行時間は長くなります。

初期値：0.80

参考点のサンプリングステップサイズ

パラメータ説明：このパラメータは、参考点のステップサイズを調整するために使用されます。点群では、このステップサイズを間隔としてサンプリングします。値が大きいほどサンプリングされた点の数が少なくなり、実行速度は速くなりますが、マッチング精度は低下します。

初期値：5

参考される点のサンプリングステップサイズ

パラメータ説明：このパラメータは、参考される点のサンプリングステップサイズを調整するために使用されます。点群では、このステップサイズを間隔としてサンプリングします。値が大きいほどサンプリングされた点の数が少なくなり、実行速度は速くなりますが、マッチング精度は低下します。

初期値：1



- **参考点** とは、点群モデルにおけるサンプリングする点を指します。
- **参考される点** とは、点群モデルにないサンプリングする点を指します。

参考点と参考される点は、ポイントペアを形成します。サンプリング間隔が大きいほど、参考点と参考される点は少なくなり、ポイントペアも少なくなり、実行の速度は速くなります。

位置姿勢フィルタリング設定

位置姿勢距離に基づいた非最大抑制を使用

パラメータ説明：このパラメータをオンにすると、選択した位置姿勢からの距離が対象物の直径の0.1倍未満である候補位置姿勢が除外されます。

初期値：オンの状態。

調整説明：調整効果を [調整の例](#) に示します。

モデルの回転角度による位置姿勢をフィルタリング

パラメータ説明：エッジマッチングモードでは、シーンの点群にマッチングするために点群モデルをある角度だけ回転させる必要がある場合があり、点群モデルの回転角度で位置姿勢が除去されます。モデルの回転角度が「角度差の上限しきい値」を超えると、その位置姿勢は除去されます。

初期値：オンの状態。

角度差の上限しきい値

詳細については、「モデルの回転角度による位置姿勢をフィルタリング」パラメータの説明をご参照ください。

初期値：135.000°

位置姿勢検証設定

単一点群の結果の期待数

パラメータ説明：このパラメータは、各入力点群のマッチング位置姿勢の期待数を示します。

初期値：3

調整アドバイス：実際の状況に応じて設定してください。調整効果を [調整の例](#) に示します。

ボクセル長さの生成戦略

パラメータ説明：このパラメータは、ボクセル長さの生成戦略を選択するために使用されます。

オプション：自動生成、手動設定。

初期値：自動生成

調整アドバイス：このステップを始めて使用する場合、**自動生成** を使用することをお勧めします。**手動設定** を選択すると、**ボクセル長さ** のパラメータを設定できます。

最小ボクセル長さ

パラメータ説明：このパラメータは、ボクセル長さの最小値（ミリメートル単位）を設定するために使用されます。初期値：1.000mm

最大ボクセル長さ

パラメータ説明：このパラメータは、ボクセル長さの最大値（ミリメートル単位）を設定するために使用されます。初期値：15.000mm



点群における空間は3Dグリッドに分割され、**ボクセル長さ** は3Dグリッドの最小単位のスサイズです。

結果可視化の設定

可視化オプション

パラメータ説明：このパラメータは、可視化する項目を選択するために使用されます。

オプション：サンプリング後のモデル、サンプリング後のシーン、マッチング結果、位置姿勢検証中に点の使用状況。

初期値：マッチング結果。

調整説明：実際のニーズに応じて設定してください。調整効果を [調整の例](#) に示します。

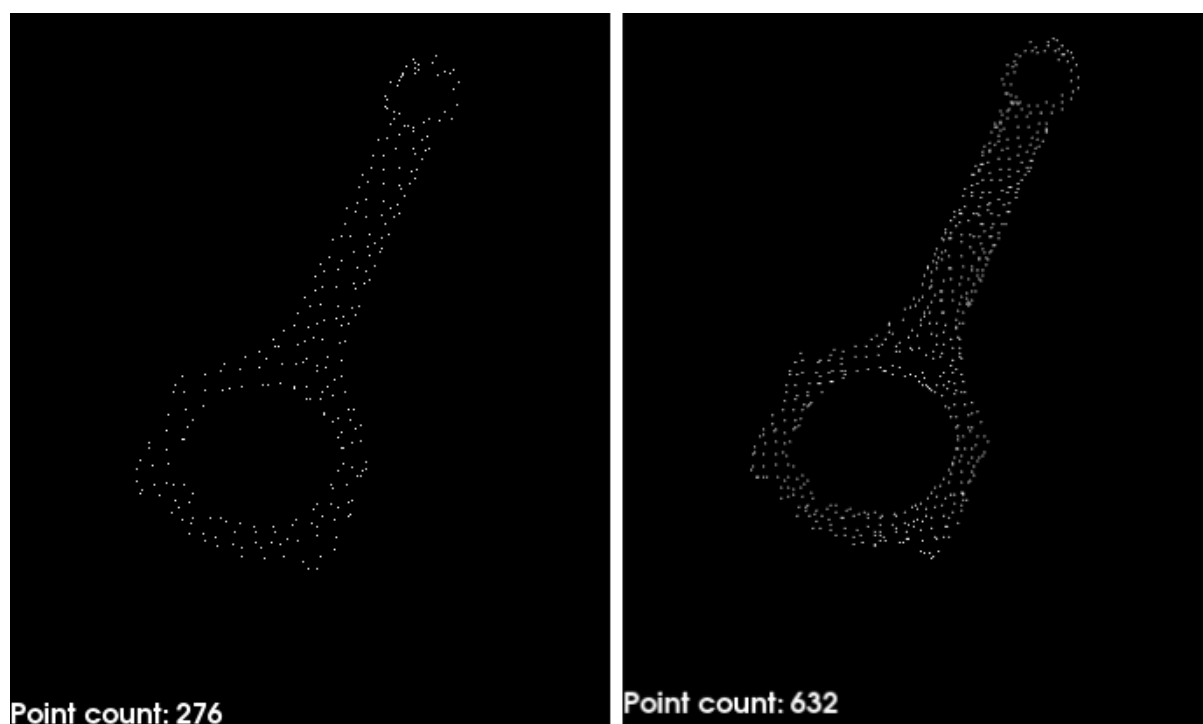
調整の例

「基本設定」のパラメータ調整

サンプリングされたモデルの期待点数

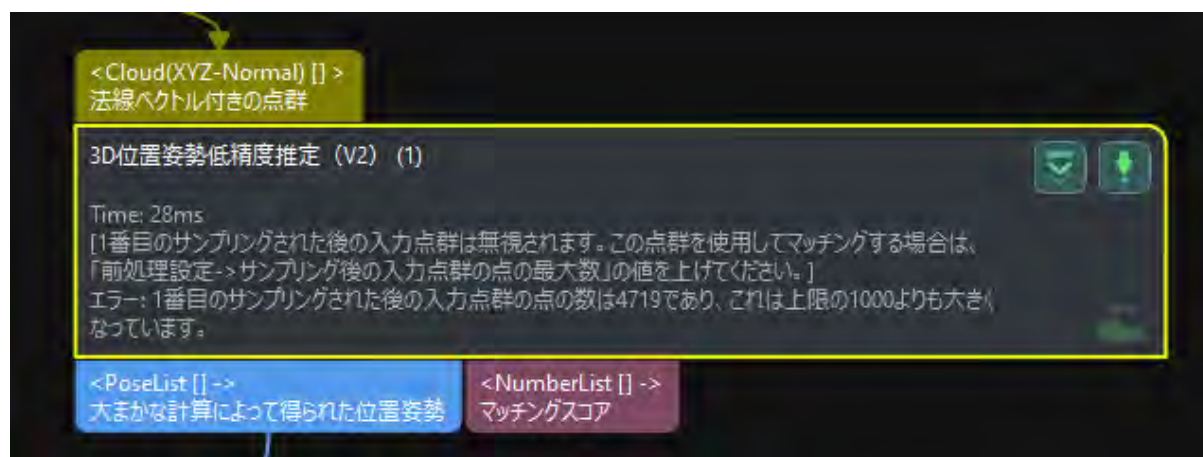
コネクティングロッドを例として説明します。 **サンプリングされたモデルの期待点数** をそれぞれ300と600に設定した場合、サンプリング後のワーク点群の点数を下図の左側、右側に示します。

左図の点群はまばらで、右図の点群は密集しています。



サンプリング後の入力点群の点の最大数

入力点群の点数がこのパラメータで設定された値を超えると、以下のようなメッセージが表示されます。



単一点群の結果の期待数

単一点群の結果の期待数 がそれぞれ6と12の場合、このステップが出力された位置姿勢結果の数を下図の左側、右側に示します。

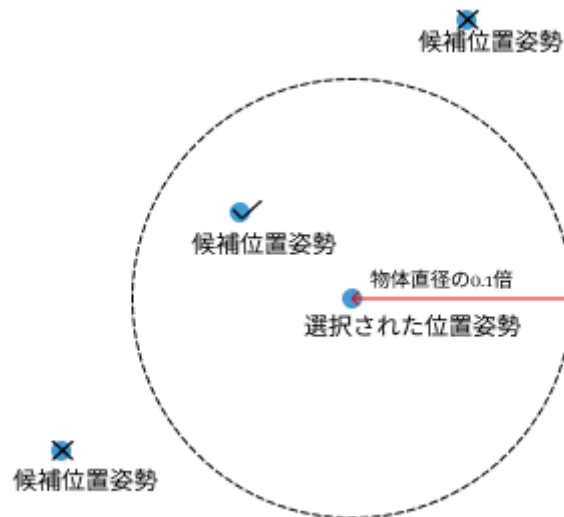
左図の位置姿勢結果の数は6で、右図の位置姿勢結果の数は12です。



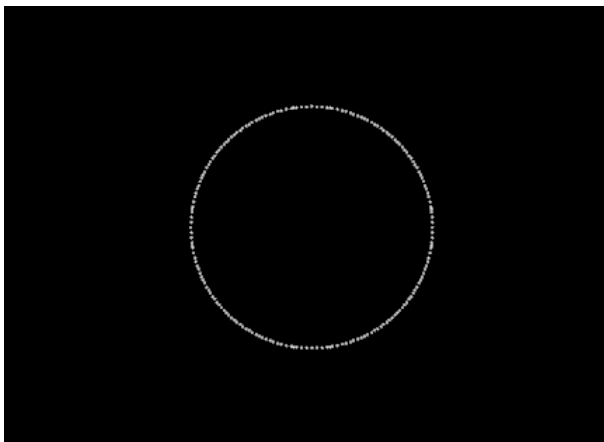
「詳細設定」のパラメータ調整

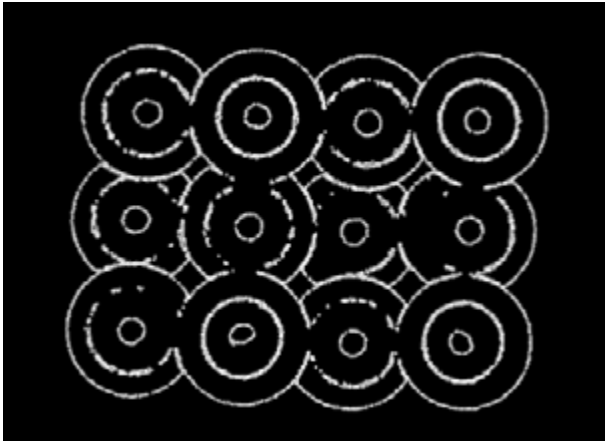
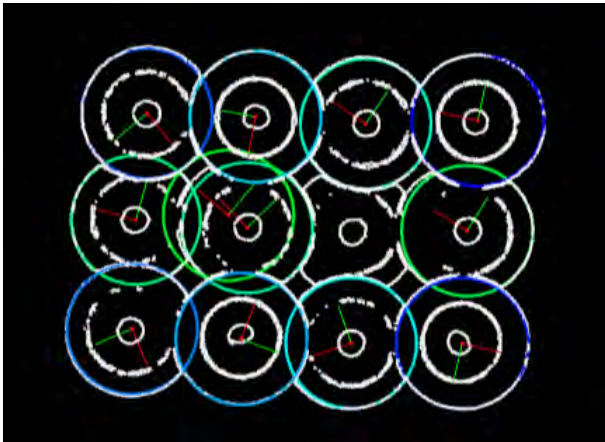
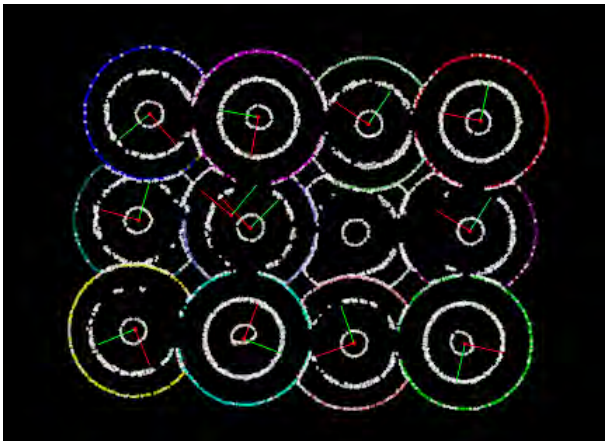
位置姿勢距離に基づいた非最大抑制を使用

下図に示すように、このパラメータをオンにすると、選択した位置姿勢からの距離が対象物の直径の0.1倍未満である候補位置姿勢が除外されます。



結果可視化の設定

可視化オプション	サンプル図
サンプリング後のモデル	

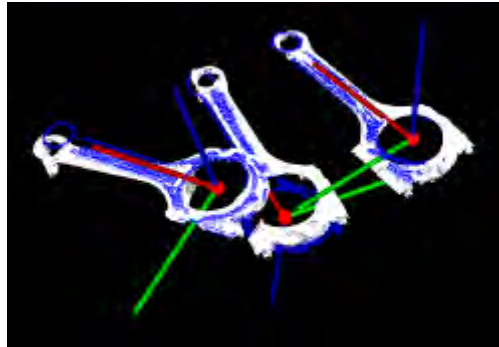
可視化オプション	サンプル図
サンプリング後のシーン	
マッチング結果	
位置姿勢検証中に点の使用状況	

4.3.6. 3D位置姿勢低精度推定

機能

点群モデルを使用して元の点群に対して大まかにマッチングし、対象物の大まかな候補位置姿勢を出力します。

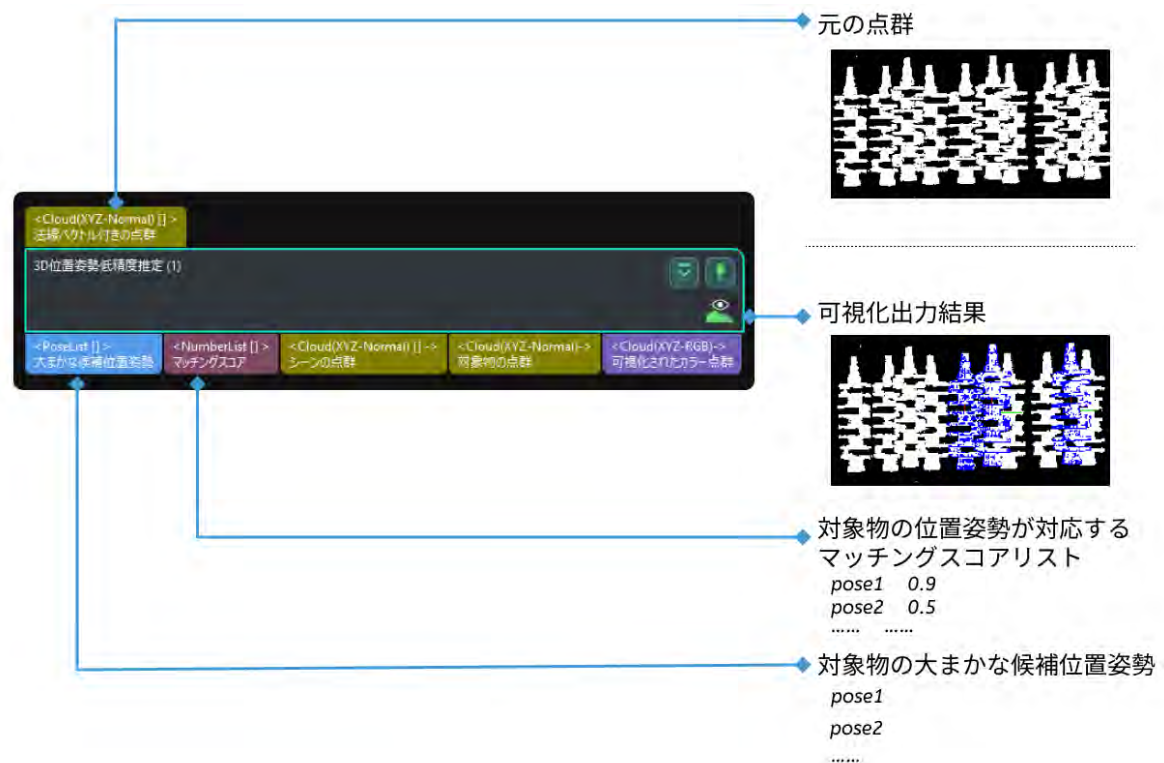
(点群モデルの作成は [点群モデルを生成](#) をご参照ください。)



使用シーン

通常、シーンの点群で対象物を検出し、その大まかな位置姿勢を取得するために使用されます。このステップは一般的に [3D位置姿勢高精度推定](#) と併用されます。

入力と出力



パラメータの説明

モデル設定

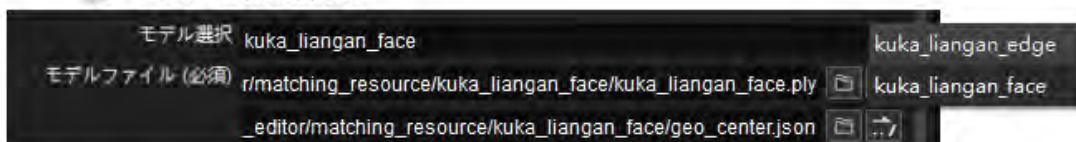
モデル選択

調整説明：右側のドロップダウンボタンをクリックして、モデルファイルと対応する幾何学的中心ファイルをすばやく選択して切り替えることができます。

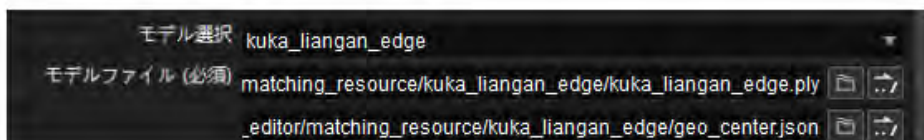
1 ▼をクリック




2 モデルを切り替える




3 自動変更



モデルファイル

調整説明：  をクリックして、ply形式のモデル点群ファイルを選択します。モデルファイルの作成は点群モデルの生成に関係します。詳しくは、[マッチングモデル・把持位置姿勢エディタ](#)をご参照ください。

幾何学的中心ファイル

調整説明：  をクリックして、json形式の幾何学的中心ファイルを選択します。幾何学的中心ファイルの作成は把持位置姿勢の追加に関係します。詳しくは、[マッチングモデル・把持位置姿勢エディタ](#)をご参照ください。

点群の点が位置する平面の方向を計算

点が位置する平面の方向の計算方法

初期値：入力点群の法線を使用（Origin）

オプション	調整説明
入力点群の法線を使用（Origin）	入力された点群の元の法線ベクトルを使用します。

StandardMode	CPUを使用して入力点群の法線ベクトルを再計算します。モデルに法線ベクトルがない場合に使用することをお勧めします。目標点の付近にそれに最も近い点をk個を見つけ、PCA（主成分分析）を使用して最小特徴ベクトルを取得してこの点の法線ベクトルとします。
EdgeTangent	入力されたエッジ点群の接線を計算し、それを法線ベクトルとします。輪郭が鏡像となる異なる物体を判別できます。平面物体のエッジ点群をマッチングする時に使用することをお勧めします。
EdgeNormal	入力されたエッジ点群の法線ベクトルを計算し、その点の接線方向を法線ベクトルとします。平面物体のエッジ点群をマッチングする時に使用することをお勧めします。

選択された隣接点の数

初期値：10 調整説明：このパラメータは点が位置する平面の方向にある隣接点の数、すなわち **StandardMode** にした時のk値を計算します。

アルゴリズムタイプ

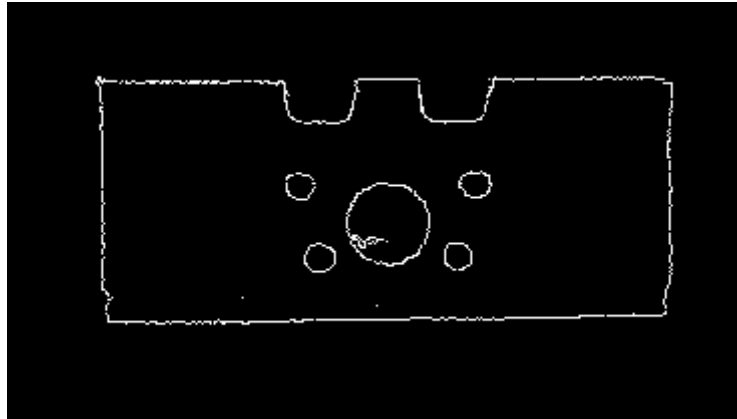
初期値：SurfaceMatchingEasyMode

オプション：SurfaceMatchingEasyMode、SurfaceMatching

調整説明：このアルゴリズムは2つのタイプがあります。アルゴリズムのタイプの調整は下図を例として、**SurfaceMatchingEasyMode** アルゴリズムの調整可能なパラメータを紹介します。

SurfaceMatchingEasyMode アルゴリズム：調整可能なパラメータは速度制御と出力設定です。

SurfaceMatching アルゴリズム：調整可能なパラメータはサンプリング設定、投票設定、クラスタリング設定、位置姿勢検証設定です。



SurfaceMatchingEasyModeパラメータの説明

速度制御

メイン速度制御パラメータ

初期値：2

調整説明：このパラメータは、アルゴリズムの速度を調整します。大きくすればアルゴリズムの速度は速くなりますが、マッチングの精度は低下します。この効果は **サブ速度制御パラメータ** より強いです。有効な範囲は1~6です。

調整の例：下図に示すように、左側は2（初期値）にした結果で、右側は6にした結果です。このパラメータを大きくすると、マッチング精度が低下します。

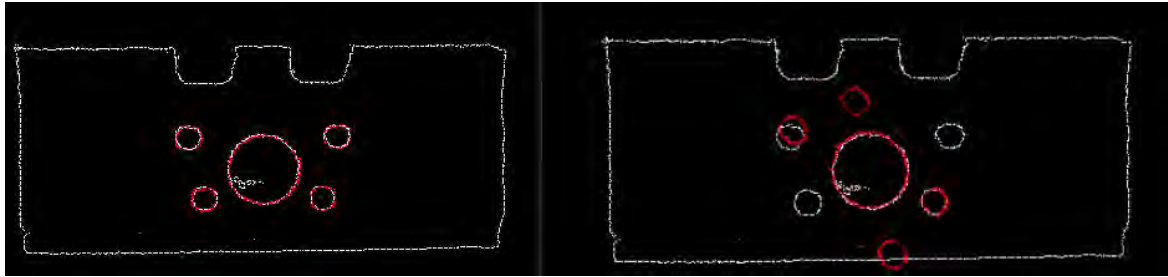


サブ速度制御パラメータ

初期値：10

調整説明：このパラメータは、アルゴリズムの速度を調整します。大きくすればアルゴリズムの速度は速くなりますが、マッチングの精度は低下します。この効果は **メイン速度制御パラメータ** より弱いです。有効な範囲は1~20です。

調整の例：下図に示すように、左側は10（初期値）にした結果で、右側は18にした結果です。このパラメータを大きくしたら精度も低下しますが、影響はメイン速度制御パラメータより弱いです。



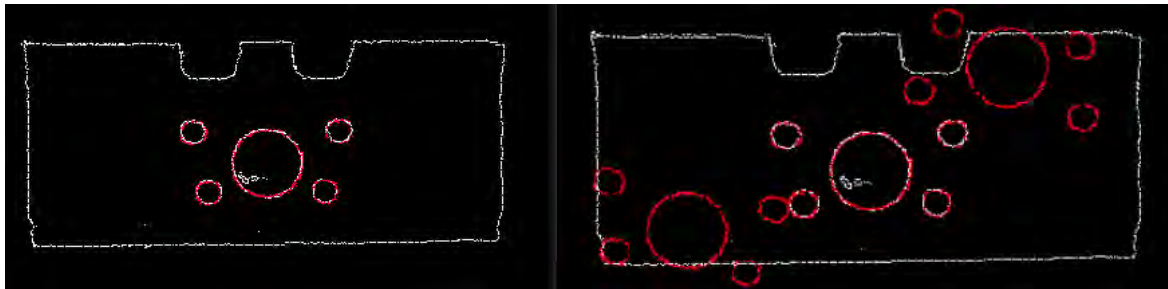
出力設定

単一点群の出力結果の数

初期値：3

調整説明：各点群のマッチングに出力される結果の数を予測します。値が大きいほど出力される結果は多くなります。有効な範囲は1~∞です。

調整の例：下図に示すように、左側は1にした結果で、右側は3（初期値）にした結果です。



SurfaceMatchingパラメータの説明

サンプリング設定

自動ダウンサンプリング

初期値：チェックを入れる

調整説明：このパラメータは、自動ダウンサンプリングを使用するかどうかを選択するために使用されます。チェックを入れれば、サンプリングされたモデルの期待点数に基づいてモデル点群のダウンサンプリングの間隔を自動的に調整します。

サンプリングされたモデルの期待点数

初期値：1000

調整説明：このパラメータは、サンプリングする点群の点数を調整するために使用されます。**自動ダウンサンプリング** にチェックを入れた場合に有効になり、取得する点群の点数はこの値に近いです。この値が小さいほど、サンプリングする点群の点数は少なくなり、位置姿勢推定の精度も低下します。

モデルサンプリングの最大点数

初期値：4000

調整説明：このパラメータはモデル点群をダウンサンプリングした後の点群の点数の最大値を設定します。すなわち、ダウンサンプリング後のモデル点群の点数に上限を設定することです。マッチングの効果は良くない場合にこの値を大きくしてください。マッチングの速度を上げたい場合に小さくしてください。

シーンサンプリングの最大点数

初期値：30000

調整説明：このパラメータはシーンの点群をダウンサンプリングした後の点群の点数の最大値を設定します。すなわち、ダウンサンプリング後のシーンの点群の点数に上限を設定することで最長実行時間を制限します。このパラメータの値を実際のサンプリング点数より小さくしてはなりません。普通、調整する必要はありませんが、最長実行時間を制限したい場合にこの値を小さくするようお勧めします。

モデル点群サンプリング間隔

初期値：0.01

調整説明：ダウンサンプリングする点の間の最大距離を調整します。mを単位とします。モデル点群のダウンサンプリング間隔が最小サンプリング間隔より小さい場合に、最小サンプリング間隔を実際のサンプリング間隔とします。この値が大きいほど、サンプリング後に計算に使用する点群の数は少なくなり、マッチングの精度は低下し、アルゴリズムの実行時間は短くなります。

調整の例：下図のように示します。左側は0.01（初期値）にした結果で、右側は0.04にした結果です。



最小サンプリング間隔

初期値：0.003

調整説明：このパラメータは、サンプリング間隔（メートル単位）を計算するために使用されます。**自動ダウンサンプリング**にチェックを入れたら有効になります。計算したサンプリング間隔がこの値より小さい場合に、この値を実際のサンプリング間隔とします。

投票設定

距離の定量化

初期値：1

調整説明：点と点との間の距離を定量化します。距離間隔 = 距離定量化 × サンプル間隔。このパラメータを大きくしたら間隔も大きくなり、マッチングの精度も低下します。

角度の定量化

初期値：60

調整説明：このパラメータは点と点の間のベクトル間の角度を定量化し、単位は°です。角度間隔 = 2×3.14 / 角度の定量化。この値が大きいほど、マッチングの精度は向上します。

最大投票率

初期値：0.8

調整説明：このパラメータは投票数が最大投票数に占める比率のしきい値を設定します。前のステップで各位置姿勢の投票数を取得し、その中の最大値にこのパラメータを掛けてしきい値を得ます。投票数がこのしきい値を超えた位置姿勢は保持されてクラスタリングします。この値が小さいほど、正確なマッチングの結果を取得する確率は高くなりますが、実行時間は長くなります。このパラメータの有効範囲は0~1です。

参考点のサンプリングステップサイズ

初期値：5

調整説明：参考点のサンプリングステップサイズを調整します。点群では、このステップサイズを間隔としてサンプリングします。この値が大きいほどサンプリングする点は少なくなり、実行の速度は早くなりますが、マッチングの精度は低下します。

参考される点のサンプリングステップサイズ

初期値：1

調整説明：参考される点のサンプリングステップサイズを調整します。点群では、このステップサイズを間隔としてサンプリングします。この値が大きいほどサンプリングする点は少なくなり、実行の速度は早くなりますが、マッチングの精度は低下します。

- **参考点** と **参考される点** とはペアポイントになります。サンプリング間隔が大きいほど参考点と参考された点は少なくなり、ペアポイントも少なくなり、実行速度は速くなります。
- **参考点** とはマッチングモデルにあるサンプリング点を指します。**参考される点** とはマッチングモデルにないサンプリング点を指します。

クラスタリング設定

クラスター比

初期値：0.1

調整説明：このパラメータはクラスタリングに使用する位置姿勢が計算して取得した位置姿勢に占める比率を調整します。あらゆる位置姿勢は計算中にスコアが付けられます。そのスコアによって位置姿勢をソートします。このパラメータはクラスタリングに使用する位置姿勢の比率を決めます。0.1（初期値）にしたら、最初の10%の位置姿勢をクラスタリングします。この値が大きいくほど、正確なマッチング結果を取得する確率は高くなりますが、実行時間も長くなります。

角度差しきい値

初期値：15

調整説明：このパラメータはクラスタリングのプロセスの角度増量を調整します。最終的な計算結果には、一つの対象物に複数の位置姿勢が計算されることがあります。この場合、パラメータが非常に近い位置姿勢をマージするとき、このパラメータは角度の増量を決めます。このパラメータが大きいくほど、角度の差が大きいく位置姿勢は最終的な結果にマージされ、マッチング精度も低下します。

距離差しきい値

初期値：0.02

調整説明：このパラメータはクラスタリングのプロセスの距離増量を調整します。最終的な計算結果には、一つの対象物に複数の位置姿勢が計算されることがあります。この場合、パラメータが非常に近い位置姿勢をマージするとき、このパラメータは距離の増量を決めます。このパラメータが大きいくほど、距離の差が大きいく位置姿勢は最終的な結果にマージされ、マッチング精度も低下します。

スコアが最も高いN個の結果を出力

初期値：5

調整説明：このパラメータはクラスタリングして取得した複数のマッチング結果にスコアが最も高いN個の結果を最終的な結果として出力します。

位置姿勢検証の設定

位置姿勢検証を使用

初期値：チェックを入れる

調整説明：このパラメータは位置姿勢検証を使用するかを選択するために使用されます。このパラメータにチェックを入れれば、クラスタリングのパラメータは全部無効になります。姿勢検証とクラスタリングは、最終的なマッチング結果の検証とフィルタリングのための異なる方法であり、同時に使用することはできません。

隣接点の検索半径

初期値：1

調整説明：このパラメータは位置姿勢検証のプロセスに検証するエリアのサイズを調整します。ボクセルを単位とします。この値を大きくしたら、位置姿勢検証のエリアは広くなり、

最終的な結果を検証するために使用する点は多くなるのでマッチングの精度は低下します。

ボクセルの長さ

初期値：3

調整説明：点群がある空間を3Dグリッドにします。このパラメータは3Dグリッドの最小単位（mm）になります。この値を大きくしたら、選択する範囲は広くなり、アルゴリズムの実行速度も速くなりますが、マッチングの精度は低下します。

単一点群の出力結果の数

初期値：3

調整説明：**SurfaceMatching** アルゴリズムを実行した場合、このパラメータの機能は**SurfaceMatchingEasyMode** と同じです。

調整の例：下図に示すように、左側は1にした結果で、右は3（初期値）にした結果です。



結果可視化の設定

サンプリングされたモデルの点群を表示

初期値：チェックを入れない

調整説明：このパラメータはダウンサンプリングされたモデルの点群を表示します。

サンプリングされたシーンの点群を表示

初期値：チェックを入れない

調整説明：このパラメータはダウンサンプリングされたシーンの点群を表示します。

マッチング結果を表示

初期値：チェックを入れる

調整説明：このパラメータはマッチングされたモデルとシーンの点群を表示します。

4.3.7. 3D位置姿勢高精度推定（簡易版）

機能

ステップ [\[vision-steps:3d-fine-matching:::3d-fine-matching\]](#) の使いやすいバージョンで、より

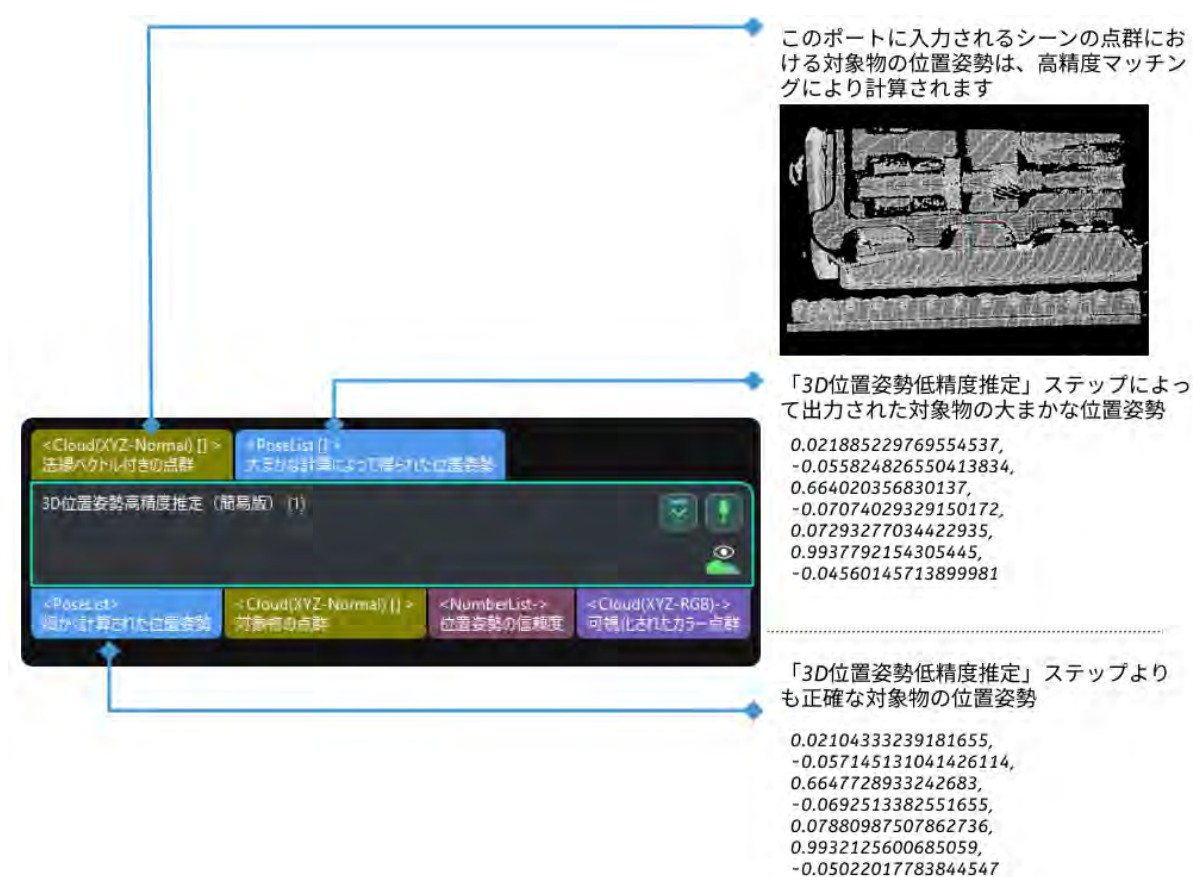
直感的にパラメータを調整することができます。

このステップにより、点群モデルとシーンの点群を正確にマッチングさせ、対象物の位置姿勢を出力することができます。

使用シーン

[vision-steps:3d-coarse-matching::3d-coarse-matching] または [vision-steps:3d-coarse-matching-v2::3d-coarse-matching-v2] ステップで出力された大まかな候補位置姿勢に基づいて、より正確なマッチングを行い、より正確な対象物の位置姿勢を出力するために使用されます。

入力と出力



パラメータの説明

パラメータのデバッグレベル

パラメータ説明：このパラメータは、パラメータのデバッグレベルを選択するために使用されます。レベルによって調整可能なパラメータは異なります。

オプション：基本設定、詳細設定

- **基本設定**：よく使われるパラメータを表示します。

- **詳細設定**：より多くのパラメータを調整可能です。

初期値：基本設定


調整アドバイス：実際のニーズに応じて選択してください。

基本設定

モデル設定

モデル選択

パラメータ説明：このパラメータは、作成された点群モデルファイルを選択するために使用されます。

調整説明：ステップパラメータで  をクリックし、ドロップダウンバーから作成された点群モデルファイルを選択します。点群モデルファイルの作成については、[マッチングモデル・把持位置姿勢エディタ](#) をご参照ください。

マッチング方法

マッチングモード

パラメータ説明：このパラメータは、エッジマッチングとサーフェスマッチングのどちらを使用するかを選択するために使用されます。

オプション：サーフェスマッチング（初期値）、エッジマッチング。

サーフェスマッチング：対象物表面の点群モデルを使用して点群モデルのマッチングを行います。エッジマッチング：対象物エッジの点群モデルを使用して点群モデルのマッチングを行います。

調整アドバイス：対象物の表面に明らかな認識可能な特徴がある場合（クランクシャフト、ローター、鉄筋など）、サーフェスマッチングを推奨しています。対象物の表面の特徴を反映した点群モデルを作成する必要があります。対象物が平らで、カメラの下に明確な固定エッジの特徴がある場合（パネル、トラックリンク、コネクティングロッド、ブレーキディスクなど）、エッジマッチングを推奨しています。対象物のエッジ特徴を反映した点群モデルを作成する必要があります。



マッチングモードは、**モデル選択** パラメータで指定されたモデルタイプ（サーフェスモデルまたはエッジモデル）と一致する必要があります。例えば、マッチングモードがサーフェスマッチングの場合、**モデル選択** パラメータでサーフェスモデルを設定する必要があります。

偏差補正能力

パラメータ説明：このパラメータは、[\[vision-steps:3d-coarse-matching:::3d-coarse-matching\]](#) または [\[vision-steps:3d-coarse-matching-v2:::3d-coarse-matching-v2\]](#) によって出力された結果の偏差を補正するために使用されます。この能力が高ければ高いほど、位置姿勢高精度推定において、大まかな位置姿勢を正確な位置姿勢に補正することができます。

偏差補能力 を大きくしすぎると、マッチング精度が低下することがありますので、適切に調整する必要があります。

オプション：高レベル、中レベル、低レベル。

初期値：低レベル（ほとんどの場合に適用可能）。

調整アドバイス：マッチング結果に大きな偏差がある場合は、まずパラメータ **偏差補正能力** を確認してください。マッチング偏差が大きすぎて、補正能力の範囲を超えている場合は、[\[vision-steps:3d-coarse-matching:::3d-coarse-matching\]](#) または [\[vision-steps:3d-coarse-matching-v2:::3d-coarse-matching-v2\]](#) のステップパラメータを調整することをお勧めします。

実行モード

パラメータ説明：このパラメータは、マッチングの実行モードを選択するために使用されます。このパラメータは、マッチング精度とマッチング速度の関係を表しています。精度が高いほど、時間がかかります。

オプション：高速、標準、高精度、超高精度。

初期値：標準。

調整アドバイス：実際のニーズに応じて選択してください。

検証設定

信頼度しきい値

パラメータ説明：このパラメータは、マッチング結果が有効であるかどうかを判定するために使用されます。マッチング結果の検証スコアがこのしきい値よりも大きい場合、結果は有効であると見なされます。この値が高いほど、結果はより正確になる傾向があります。

初期値：0.000

モデルの可視表面のみを考慮

パラメータ説明：このパラメータは、マッチングにおいてモデルの可視部分のみを考慮するかどうかを選択するために使用されます。マッチングモードを **サーフェスマッチング** に設定する場合、このパラメータを設定する必要があります。チェックを入れると、モデルの可視部分のみを考慮してマッチングを行うため、計算が容易になり、マッチングの信頼度が高くなります。

初期値：チェックを入れない。

調整説明：同じ視点にカメラの見えない部分がある対象物のモデル（完全な円柱の対象物モデルなど）には、このオプションにチェックを入れることをお勧めします。同じ視点から完全に見える対象物のモデル（プレーキディスク表面のマッチングモデルなど）の場合、このオプションにチェックを入れる必要はありません。

出力設定

単一点群の出力結果の数

パラメータ説明：1つの点群に対して出力される結果数の上限値です。

初期値：1

調整説明：1つの点群に対して複数のマッチング結果があった場合、スコアの降順にソートされ、上限値に従ってスコアの高い結果が出力されます。

詳細設定

詳細については、ステップ [詳細設定](#) をご参照ください。

4.3.8. 3D位置姿勢高精度推定（マルチモデル）

機能

複数のモデルを使用してシーン内の物体に対して位置姿勢の高精度推定を実行し、対象物のより正確な把持位置姿勢を出力します。

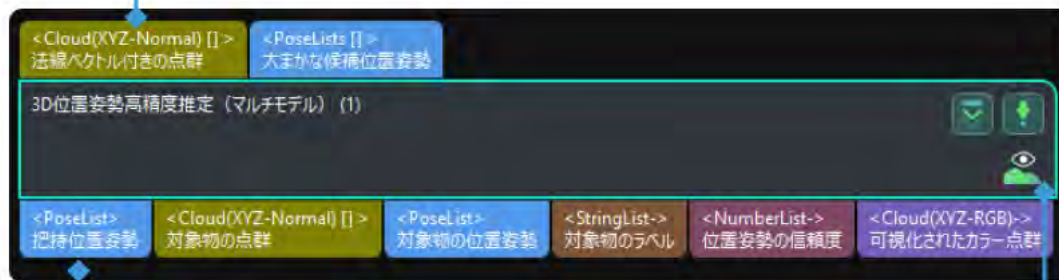
使用シーン

このステップは [3D位置姿勢低精度推定（マルチモデル）](#) に出力された候補位置姿勢に基づいて実行し、さらに正確なマッチングをし、さらに正確な位置姿勢を出力します（把持位置姿勢として使用可能）。

入力と出力



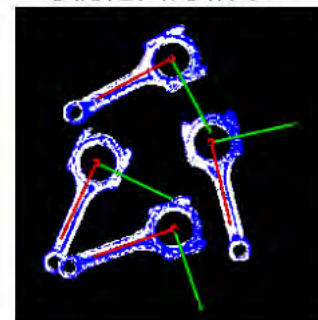
マッチングする点群リスト



得られた把持位置姿勢

```
pose1
pose2
pose3
pose4
pose5
```

可視化出力結果



パラメータの説明

パラメータのデバッグレベル

初期値：基本設定

オプション：基本設定、詳細設定

調整説明：二つの異なるパラメータ調整レベルを設定できます。

- **Basic** は基本設定で、一部のパラメータを調整できます。
- **Advanced** は詳細設定で、より多くのパラメータを調整できます。

基本設定

ここで、基本設定（Basic）モードで調整可能なパラメータについて説明します。これらのパラメータは [詳細設定](#) でも調整できます。

モデルと把持位置姿勢の設定

モデルファイル（必須）

初期値：model.ply

調整説明：ply形式のモデル点群です。複数のファイルパスを入力することができます。ファイル名の間にセミコロンを入力してください。

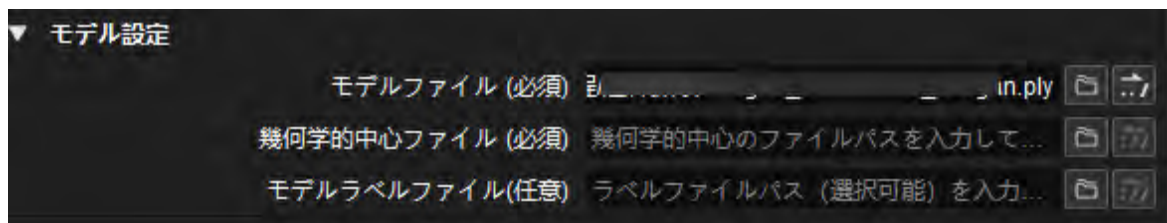
幾何学的中心ファイル（必須）

調整説明：json形式の幾何学的中心ファイルです。複数のファイルパスを入力することができます。ファイル名の間にセミコロンを入力してください。

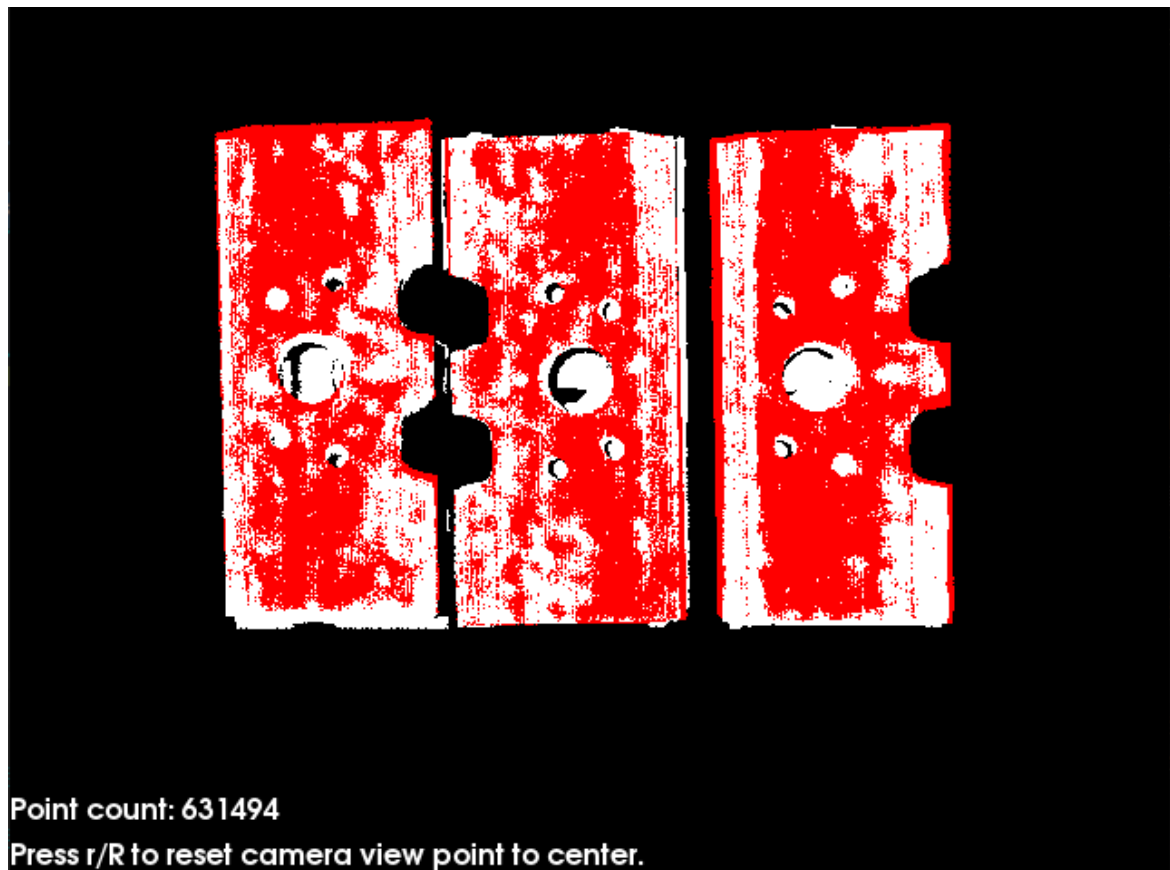
モデルラベルファイル（選択可能）

調整説明：json形式のモデルラベルファイルです。複数のファイルパスを入力できます。ファイル名の間にセミコロンを入力してください。

調整の例：各パラメータにファイルを同じ順序で入力してください。下図に示すように、**モデルラベルファイル** に先に正面モデルのファイルパスを入力したら、**モデルファイル** と **幾何学的中心ファイル** にも先に正面モデルのファイルパス（positive）を入力してください。



設定後に出力された点群は下図に示すように、物体の正面モデルも裏面モデルもシーンの点群に正確にマッチングできます。



点群の点が位置する平面の方向を計算

点が位置する平面の方向の計算方法

初期値：入力点群の法線を使用（Origin）

調整説明：点群の点の方向を計算する方法は四つあります。実際のニーズに応じて選択します。

オプション	調整説明
入力点群の法線を使用（Origin）	入力された点群の元の法線ベクトルを使用します。
StandardMode	CPUを使用して入力された点群の法線ベクトルを再計算します。モデルに法線ベクトルがない場合に使用することをお勧めします。目標点の付近に目標点にもっとも近いk個の点を検索して、PCA（主成分分析）を使用して最小の固有ベクトルを取得してこの点の法線ベクトルとします。

オプション	調整説明
EdgeTangent	入力されたエッジ点群の接線を計算し、それを法線ベクトルとします。輪郭が鏡像となる異なる物体を判別できます。平面物体のエッジ点群をマッチングする時に使用することをお勧めします。
正面・裏面を判別します。 3D位置姿勢高精度推定（マルチモデル） ステップを使用できます。モデル点群は物体の正面・裏面モデルです。	正面・裏面を判別しません。 3D位置姿勢高精度推定 ステップは十分です。
EdgeNormal	入力されたエッジ点群の法線ベクトルを計算します。平面形状の物体のエッジ点群をマッチングする場合に使用することをお勧めします。



EdgeTangent または **EdgeNormal** を使用するとき、各エッジ点群に複数の物体がないこと、つまり各物体の点群がすでに分割されたことを確認してください。

選択された隣接点の数

初期値：10

調整説明：点が位置する平面の方向を計算するために使用する隣接点の数です。最小値は1です。

対応の設定

初期値：[GMM](#)

オプション：[GMM](#)、[nearest-neighbor](#)

調整説明：このパラメータは局所マッチング方法を選択し、モデル点群とシーンの点群をマッチングします。モデル点群は前のステップで取得したシーンの点群とマッチングするために、設定した範囲によって移動・回転します。

調整の例：ほとんどのシーンでは、[GMM](#) はより優れている外乱耐性とより速いマッチング速度を実現できるのでこのモードをお勧めします。ごく一部の場合に最近点（[nearest-neighbor](#)）モードを使用してもいいです。

GMMモードパラメータ設定

マッチングモード

初期値：Standard

オプション：高速、標準、高精度

調整説明：マッチングモードは3つあり、実際の状況に応じて適切なものを選択してください

い。

高速：速度が最も速いですが、精度は低いです。 **標準**：安定的に実行できるモードです。 **高**

精度：高精度を実現できますが、速度が遅いです。

反復回数

初期値：30

調整説明：反復とは、条件を満たす結果が出るまである指定を繰り返し実行することです。反復回数とはこの過程に指令を実行する回数です。ここで設定するのは反復回数の上限で、大きく設定すればマッチングの実行回数も多くなり、かかる時間も長くなりますが、マッチングの精度が向上します。

標準偏差

初期値：0.005

調整説明：前のステップ（[3D位置姿勢低精度推定](#)）で処理された大まかな位置姿勢の偏差が大きい、即ち、シーンの点群とモデルの点群の偏差が大きい場合、**標準偏差**を大きくしてください。そうすると実行の速度も速くなります。

調整の例：大まかな位置姿勢の偏差に応じて設定します。大まかな位置姿勢の偏差よりも大きく設定したら、誤差が極めて大きい結果を出力することがあります。したがって普通は0.01m以下に設定してください。

標準偏差減衰の更新ステップサイズ

初期値：3

調整説明：マッチング処理中、**標準偏差**は**最小標準偏差**で設定された値に達するまでどんどん小さくなります。その過程で標準偏差が減衰する回数がこのパラメータの値です。標準誤差が大きい場合に、減衰の回数が多いほど、減衰が安定的に実行できるようになります。普通、このパラメータを調整する必要はありません。

最小標準偏差

初期値：0.001

調整説明：**標準偏差**の最小値、つまり減衰後に達した最小値です。

反復に重みを使用

初期値：チェックを入れない

調整説明：チェックを入れれば、モデルの重みがアルゴリズムの反復に使用されるので、重みモデルをより正確にマッチングすることができます。

最近点モードのパラメータ設定

反復設定

反復回数

初期値：30

調整説明：反復とは、条件を満たす結果が出るまである指定を繰り返し実行することです。

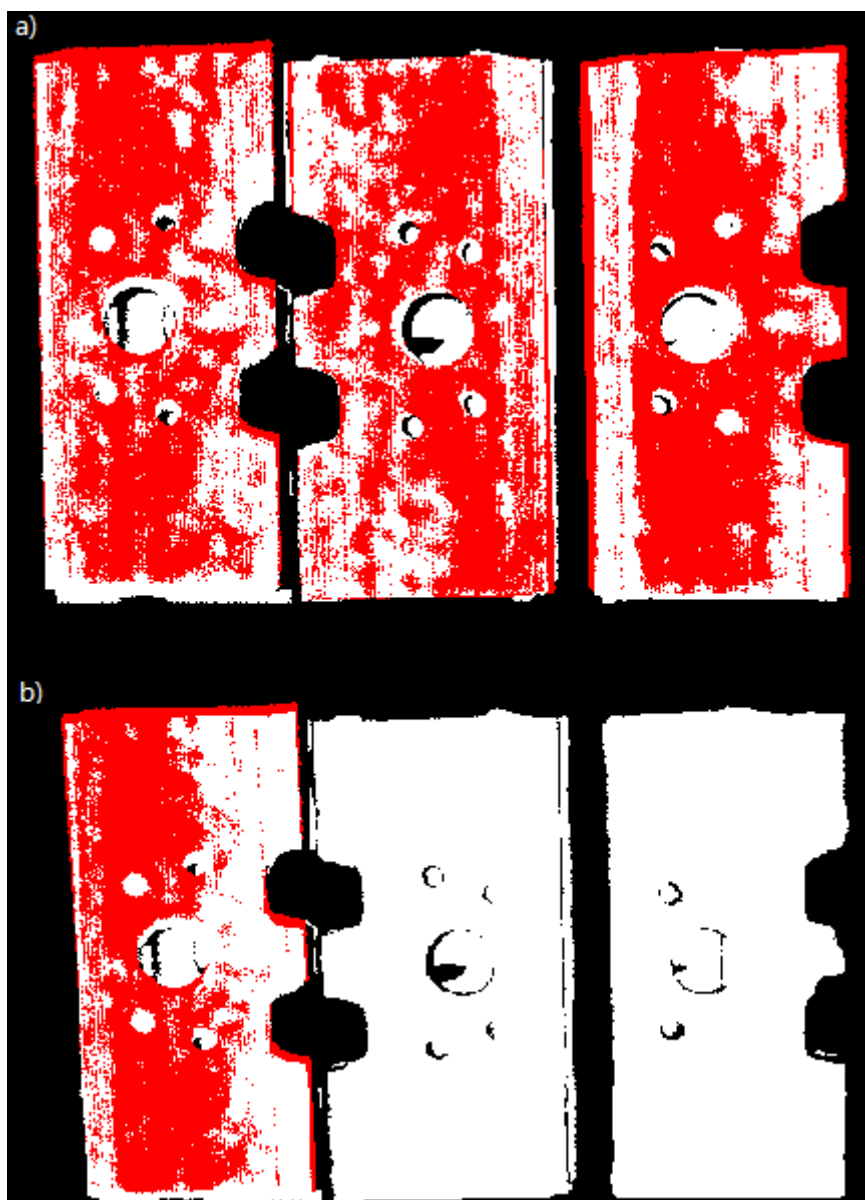
反復回数とはこの過程に指令を実行する回数です。最近点モードでは、計算がアルゴリズムの精度要件を満たす時、反復は中止します。ここで設定するのは反復回数の上限で、大きくすればマッチングの実行回数も多くなり、かかる時間も長くなりますが、マッチングの精度が向上します。

最近点の検索半径

初期値：0.01

調整説明：最近点の検索半径を調整します。単位はmです。大まかな位置姿勢の偏差に応じて設定します。入力された大まかな位置姿勢の偏差が大きい場合に、つまりシーンの点群とモデル点群の偏差が大きい場合に、この値を大きくして最近点の検索範囲を広くします。これは、各点にそれぞれ最近点を検索するためです。大まかな位置姿勢の偏差よが小さい場合、この値を小さくしてください。設定可能な最小値は0.001です。この下限値より小さいと、一部の疎らな点に最近点を検索できなく、出力に結果に影響が出ることがあります。

調整の例：下図のbに示すように、大まかな位置姿勢の偏差が大きくて、このパラメータの最小値が0.001である場合に、不完全なマッチングの結果が出ます。下図のaに示すように、このパラメータを0.01に設定したら、検索範囲が広くなり、より完全なマッチング結果を出力することになります。



平均二乗誤差しきい値

初期値：0.001

調整説明：誤差の二乗和の平均値を調整します。反復するたびにこの値が記録され、その後の反復ステップの結果を比較するために使用されます。設定した値より小さい誤差の二乗和の平均値が出たら、期待された結果を取得したことになり、反復を中止します。

ウィンドウサイズ

初期値：10

調整説明：反復の過程に誤差の変動が小さい連続反復の回数で、最小値は3です。この値が小さすぎると一部だけの最適解が出てきて最終的なマッチング精度に影響が出る可能性があります。例えば、ウィンドウには3回連続の反復の結果しかない場合、この3回だけの反復の誤差を比較して誤差変動のトレンドを把握します。こうすれば全体でなく、一部だけの最適解を取得するかもしれません。

対応する点を表示

初期値：チェックを入れない

調整説明：チェックを入れたら、毎回反復実行中に対応する点を表示します。

複雑形状の対象物であるか

初期値：チェックを入れない

調整説明：対象物が複雑形状の物体（シンプルな幾何学的形状ではなく）で、かつ **反復ポイントペアの重みを自動的に計算** を使用すれば、このパラメータにチェックを入れてください。

反復ポイントペアの重みを自動的に計算

初期値：チェックを入れない

調整説明：チェックを入れれば、反復ポイントペアの重みを自動的に計算します。マッチングの結果が良くない対象物にこれを使用することをお勧めします。

ポイントペアの除去設定

異常ポイントペア除去

初期値：チェックを入れない

調整説明：最近点を検索する過程に複数のポイントペアが生じるので、この機能を使用してポイントペアをフィルタリングします。チェックを入れれば、この機能を使用して下記の **ポイントペア除去シグマ複数しきい値** と **ポイントペア角度差のしきい値** パラメータを調整して異常ポイントペアを除去します。チェックを外すと、この二つのパラメータは機能しません。

ポイントペア除去シグマ複数しきい値

初期値：1.0000

調整説明：シグマ複数しきい値とは、ポイントペアの距離の分散の倍数です。ポイントペアの距離によってフィルタリングを行います。距離がこの値を超えたポイントペアは除去されます。

ポイントペア角度差のしきい値

初期値：45°

調整説明：ペアポイントの法線ベクトルの角度がこの値より大きい場合、このペアポイントは除去されます。

反復で重みを使用

初期値：チェックを入れない

調整説明：チェックを入れれば、モデルの重みがアルゴリズムの反復に使用されるので、重みモデルをより正確にマッチングすることができます。

重複ポイントペア除去

初期値：チェックを入れない

調整説明：チェックを入れれば、重複ポイントペアを除去して **重複点群タイプ除去** は有効になります。モデル点群の点数がシーンの単一物体の点数より多い場合に使用することをお勧めします。

重複点群タイプ除去

初期値：MinDis

オプション：MinDis、GlobalMin

調整説明：このパラメータはペアポイントの重複対応を削除する方法を選択します。二つの方法があります。

MinDis：対応が重複するポイントペアの間の距離を計算し、距離が最も小さいポイントペアを保持して残りのポイントペアを除去します。**GlobalMin**：対応が重複するポイントペアの間の距離の平均値を基準として、この平均値との差が最も小さいポイントペアを保持して、残りのポイントペアを除去します。

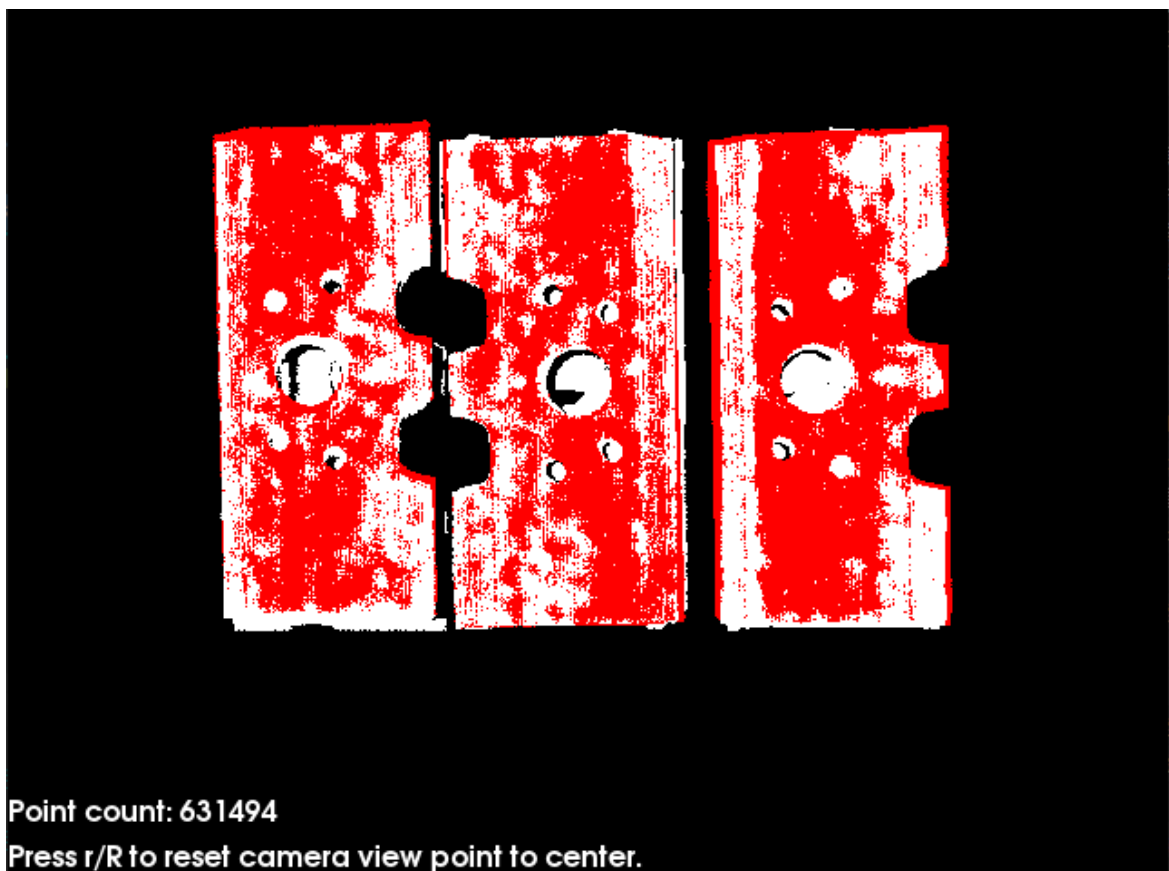
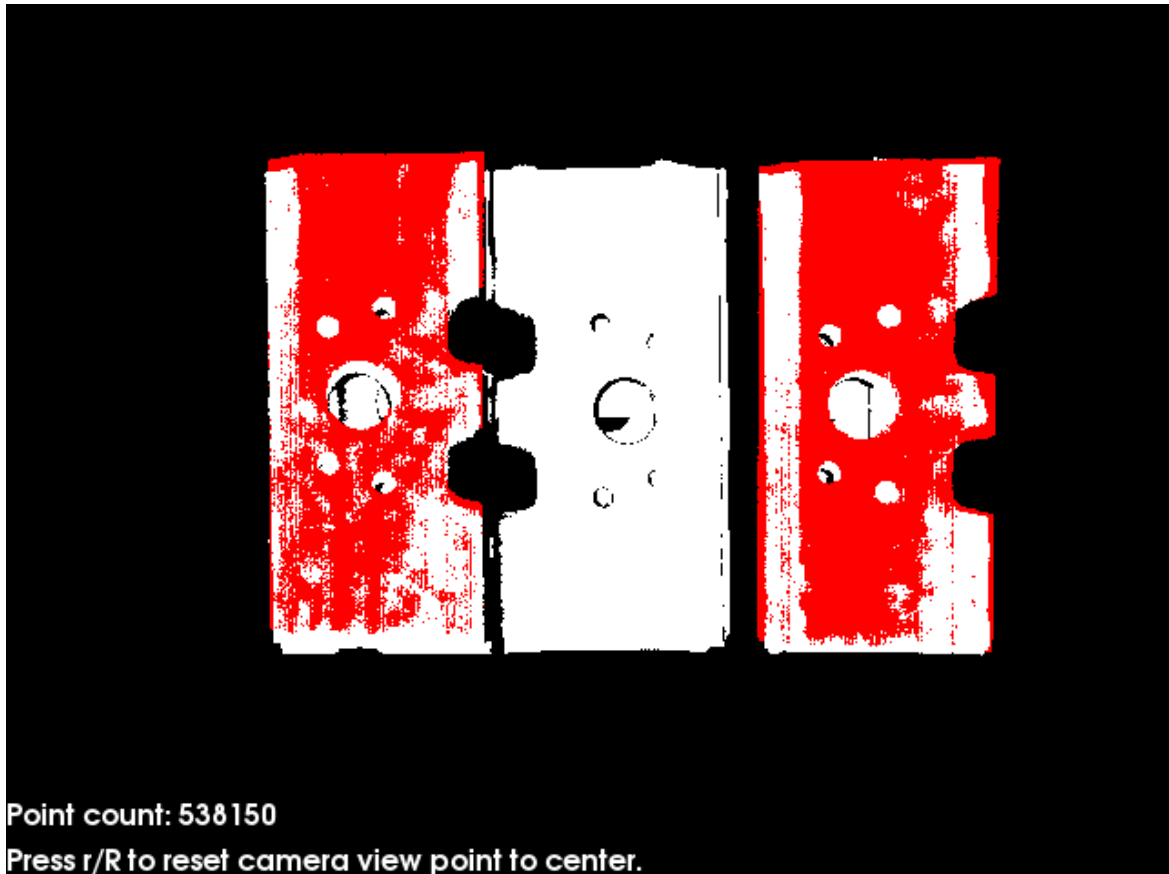
サンプリング設定

サンプリング間隔

初期値：0.005

調整説明：モデルとシーンの点群ダウンサンプリングのためのサンプリング間隔です。この値を高くすれば、サンプリングによって取得する点群は少なくなり、モデルマッチングの精度も低下します。

調整の例：下記の2つの画像はそれぞれ0.0128と0.01に設定したときに取得した点群を示します。白いのは入力されたシーンの点群です。二枚の画像を比較して、サンプリング間隔を小さく設定すれば（上の画像）、より正確な結果が出力されることが分かります。



検索点数

初期値：20

調整説明：パラメータの調整レベルを「詳細設定」にしたときにこのパラメータを調整できます。互換性のためにこのパラメータを保持し、サンプリングの過程に検索する点数を調整するために使用します。

位置姿勢フィルタリング設定

モデルの回転角度による位置姿勢をフィルタリング

パラメータ説明：エッジマッチングモードでは、シーンの点群にマッチングするために点群モデルをある角度だけ回転させる必要がある場合があり、点群モデルの回転角度で位置姿勢が除去されます。モデルの回転角度が「角度差の上限しきい値」を超えると、その位置姿勢は除去されます。

初期値：オフの状態。

角度差の上限しきい値

詳細については、「モデルの回転角度による位置姿勢をフィルタリング」パラメータの説明をご参照ください。

初期値：135.000°

検証設定

信頼度しきい値

初期値：0.5

有効範囲： $[0, +\infty)$

調整説明：結果の検証スコアがこのしきい値よりも大きい場合、結果は有効であると見なされます。この値が高いほど、結果はより正確になる傾向があります。

結果評価の検索半径

初期値：0.01 m

有効範囲： $[0, +\infty)$

調整説明：信頼度を計算するときの検索半径です。物体の点群によって設定します。疎らな点群に対して大きく設定してください。

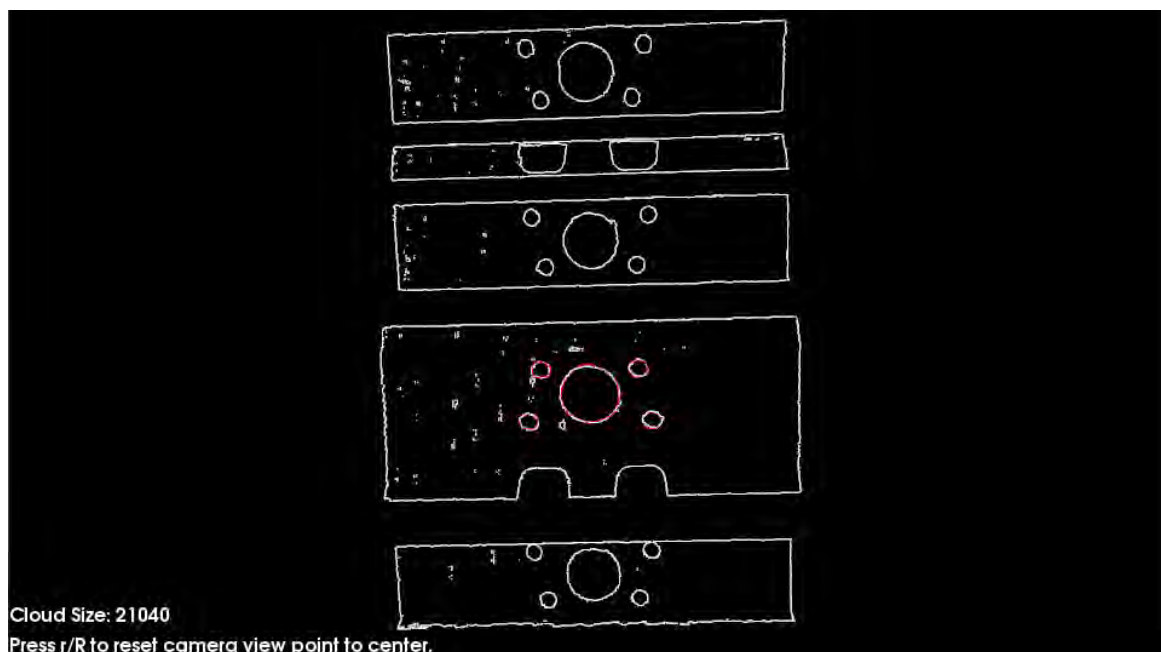
出力設定

単一点群の出力結果の数

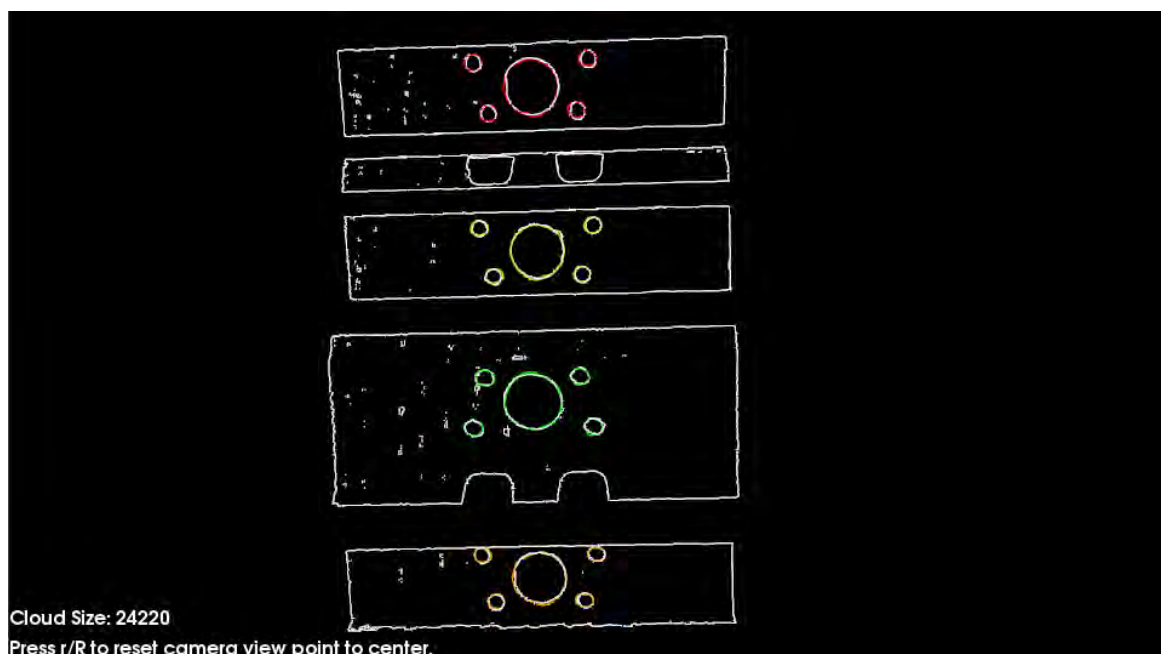
初期値：1

調整説明：この値が大きいほど、マッチング結果の出力は多くなります。

調整の例：下図は初期値で、1つの点群から1つの結果を出力しています。



このパラメータを4に設定すると、下図のように、より多くのマッチング結果が出力されるようになります。



結果可視化の設定

マッチング結果を表示

初期値：チェックを入れる

調整説明：チェックを入れれば、出力されたモデルとシーンの点群が表示されます。

詳細設定

ここでは、詳細設定（Advanced）に追加された調整可能なパラメータについて説明します。基本設定（Basic）と重なる内容は、[基本設定](#)をご参照ください。

対称性設定

認識する部品が局所対称性を持つ場合、局所最適解を回避するために、対称性を設定する必要があります。幾何学的中心点の軸を回転軸として指定することにより、モデルは設定された軸を中心に回転してマッチングし、最適なマッチング結果を取得します。

対称回転軸

初期値：ROTATE_BY_Z

オプション：ROTATE_BY_X、ROTATE_BY_Y、ROTATE_BY_Z

調整説明：このパラメータは、幾何学的中心の回転軸を選択するために使用されます。

対称角度ステップサイズ

初期値：360°

調整説明：対称角度を調整します。**最小回転角**を**最大回転角**に変更する角度ステップサイズです。

調整の例：対象物が対称軸を中心に60°回転する前後で同じように見える場合、その対称角度ステップサイズは60°です。

最小回転角度

初期値：-180°

調整説明：このパラメータは最小回転角度を調整するために使用されます。

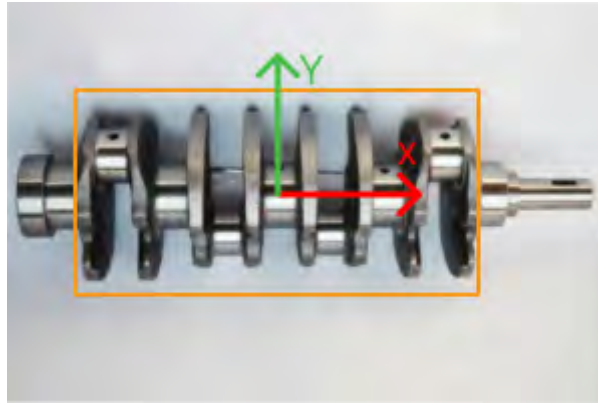
最大回転角度

初期値：180°

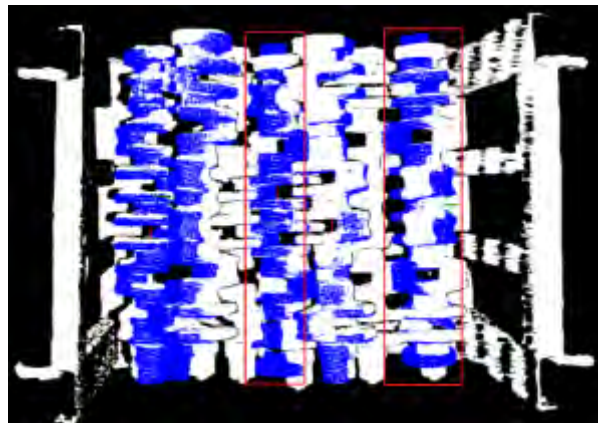
調整説明：このパラメータは最大回転角度を調整するために使用されます。

調整の例

下図に示すクランクシャフトの場合、その主要部分、つまりボックス内の部分は、Y軸に沿って180°対称です。

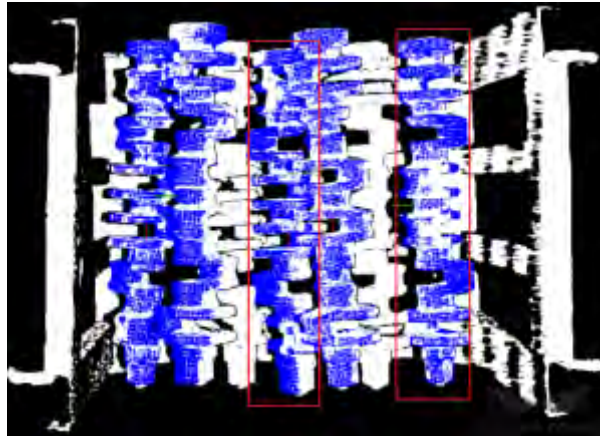


対称性が設定されていない場合、下図に示すように、局所的な形状が類似しているため、マッチング結果が局所最適になり、マッチングエラーが発生する可能性があります。



エラー結果を分析すると、モデルはクランクシャフトの中間部分のみにマッチングします。モデルがマッチング中に180°回転した後にマッチングを試行すると、クランクシャフト全体がマッチングします。また、そのマッチングスコアは、回転前のマッチングスコアよりも高くなければなりません。そうすれば、正しい結果を得ることができます。

下図に示すように、対称回転軸をROTATE_BY_Y（幾何学的中心点のY軸）に設定し、対称角度ステップサイズを180°に設定すると、正しいマッチング結果が得られます。



検証時のモデルの重み

認識する部品の「主体」が類似して「局所」だけが異なった場合、モデルの「局所」の重みを大きくして「局所」のマッチングの結果を出力します。

モデルの重みファイル

調整説明：このファイル（ply形式）は元の点群モデルファイルにある一部の点群です。この一部の点群は高い重みがあり、**重み値**を使用して設定します。現場では扱う部品に応じて選択します。

重み値

初期値：2.0

調整説明：点群に重み値を設定します。単位はありません。大まかな点群の重み係数がいずれも1とすれば、目標点群にこのパラメータを設定したら、目標点群の重み係数は**1 X このパラメータ**になり、後続のマッチングステップでそれを強調することになります。

重み設定の検索半径

初期値：0.0030

調整説明：点群の重みを増加する過程に検索半径を設定します。。重みの設定は元のモデル点群に対する処理で、モデル点群をマッチングに使用する前にダウンサンプリングするため、重みを設定する点は位置変動、またはロスすることがあります。ロスした点の付近の点に対して重みを設定する過程に検索半径を設定する必要があり、この検索半径はこのパラメータを使用して設定します。

調整の例

この例は、対称性設定の例に続いて説明します。

クランクシャフトの「主体」が類似して「局所」だけが異なった場合、モデルの「局所」の重みを大きくして「局所」のマッチングの結果を出力します。

検証設定

結果評価時の点から法線ベクトルへの角度偏差を考慮

初期値：チェックを入れない

調整説明：チェックを入れれば、位置姿勢を検証する過程に対応するポイントペアの法線ベクトルの角度差を考慮しますので、出力の数が少なくなりますが精度が高くなります。

スコアに対応する点がシーンの点群に占める比例をかける

初期値：チェックを入れない

調整説明：チェックを入れれば、マッチングのスコアに「対応する点がシーンの点群に占める比例」をかけます。1つの点群に複数の物体位置姿勢がある場合に適していません。

結果可視化の設定

サンプリングされたモデル点群を表示

初期値：チェックを入れない

調整説明：チェックを入れれば、出力の結果にサンプリングされたモデル点群が表示されます。現場の状況に応じて設定します。

サンプリングされたシーンの点群を表示

初期値：チェックを入れない

調整説明：チェックを入れれば、出力の結果にサンプリングされたシーンの点群が表示されます。現場の状況に応じて設定します。

モデルとシーンの点群の対応関係を表示

初期値：チェックを入れない

調整説明：チェックを入れれば、出力の結果にモデルとシーンの点群の対応関係が表示されます。現場の状況に応じて設定します。

4.3.9. 3D位置姿勢高精度推定

機能

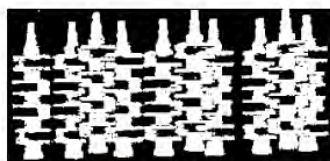
このステップにより、点群モデルを使用して元の点群に対して精確にマッチングし、対象物の更に正確な位置姿勢を出力します。（点群モデルの作成は [点群モデルを生成](#) をご参照ください。）



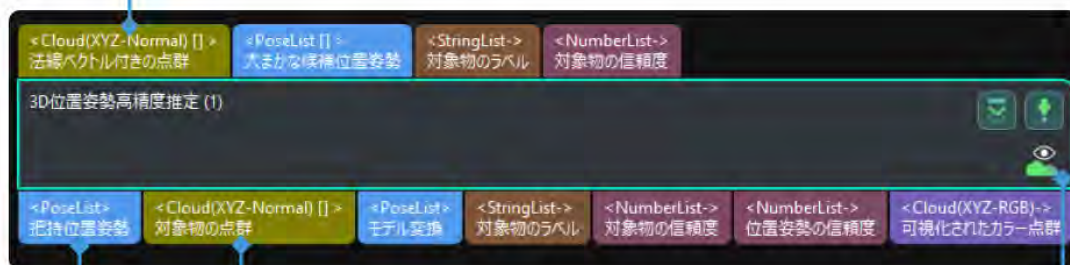
使用シーン

このステップは [3D位置姿勢低精度推定](#) に出力された大まかな候補位置姿勢に基づいて実行し、さらに正確なマッチングをし、さらに正確な位置姿勢を出力します（把持位置姿勢として使用可能）。

入力と出力



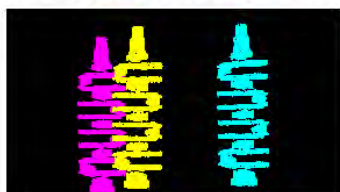
このポートに入力される点群リストはマッチングされます



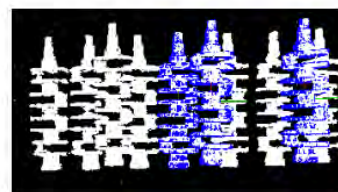
得られた把持位置姿勢

```
pose1
pose2
pose3
pose4
pose5
```

現在のマッチング結果によって生成されたモデル点群リスト



可視化出力結果



パラメータの説明

パラメータのデバッグレベル

初期値：基本設定

オプション：基本設定、詳細設定

調整説明：二つの異なるパラメータ調整レベルを設定できます。

- **Basic** は基本設定で、一部のパラメータを調整できます。
- **Advanced** は詳細設定で、より多くのパラメータを調整できます。

基本設定

ここで、基本設定（Basic）モードで調整可能なパラメータについて説明します。これらのパラメータは [詳細設定](#) でも調整できます。

モデル設定

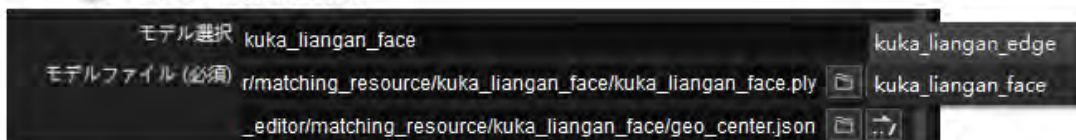
モデル選択

調整説明：右側のドロップダウンボタンをクリックして、モデルファイルと対応する幾何学的中心ファイルをすばやく選択して切り替えることができます。

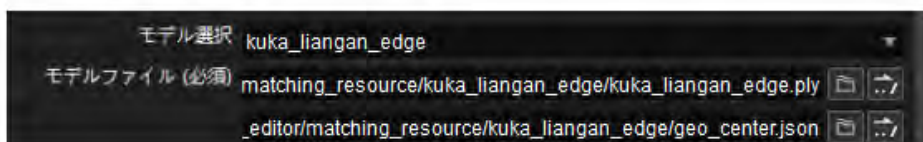
1 ▼をクリック




2 モデルを切り替える




3 自動変更



モデルファイル

調整説明：  クリックして、ply形式のモデル点群ファイルを選択します。モデルファイルの作成は点群モデルの生成に関係します。詳しくは、 [マッチングモデル・把持位置姿勢エディタ](#) をご参照ください。

幾何学的中心ファイル

調整説明：  をクリックして、json形式の幾何学的中心ファイルを選択します。幾何学的中心ファイルの作成は把持位置姿勢の追加に関係します。詳しくは、 [マッチングモデル・把持位置姿勢エディタ](#) をご参照ください。

点群の点が位置する平面の方向を計算

点が位置する平面の方向の計算方法

初期値：入力点群の法線を使用（Origin）

調整説明：点群の点の方向を計算する方法は4つあります。実際の状況に応じて選択します。

オプション	調整説明
入力点群の法線を使用（Origin）	入力点群の元の法線ベクトルを使用します。
StandardMode	CPUを使用して入力点群の法線ベクトルを再計算します。モデルに法線ベクトルがない場合に使用することをお勧めします。目標点の付近にそれに最も近い点をk個を見つけ、PCA（主成分分析）を使用して最小特徴ベクトルを取得してこの点の法線ベクトルとします。
EdgeTangent	入力されたエッジ点群の接線を計算し、それを法線ベクトルとします。輪郭が鏡像となる異なる物体を判別できます。平面物体のエッジ点群をマッチングする時に使用することをお勧めします。以下の2つの場合に使用されます。
	1：表・裏を判別します。 3D位置姿勢高精度推定（マルチモデル） ステップを使用でき、モデル点群は物体の表・裏モデルです。
	2：表・裏を判別しません。 3D位置姿勢高精度推定 ステップは十分です。
EdgeNormal	入力されたエッジ点群の法線ベクトルを計算します。平面形状の物体のエッジ点群をマッチングする場合に使用することを推奨します。



EdgeTangent または **EdgeNormal** を使用する時、各エッジ点群に複数の物体がないことを確認してください。つまり、各物体の点群が分割されたことを確認してください。

選択された隣接点の数

初期値：10

調整説明：このパラメータは点の方向にある隣接点の数を計算します。最小値は1です。

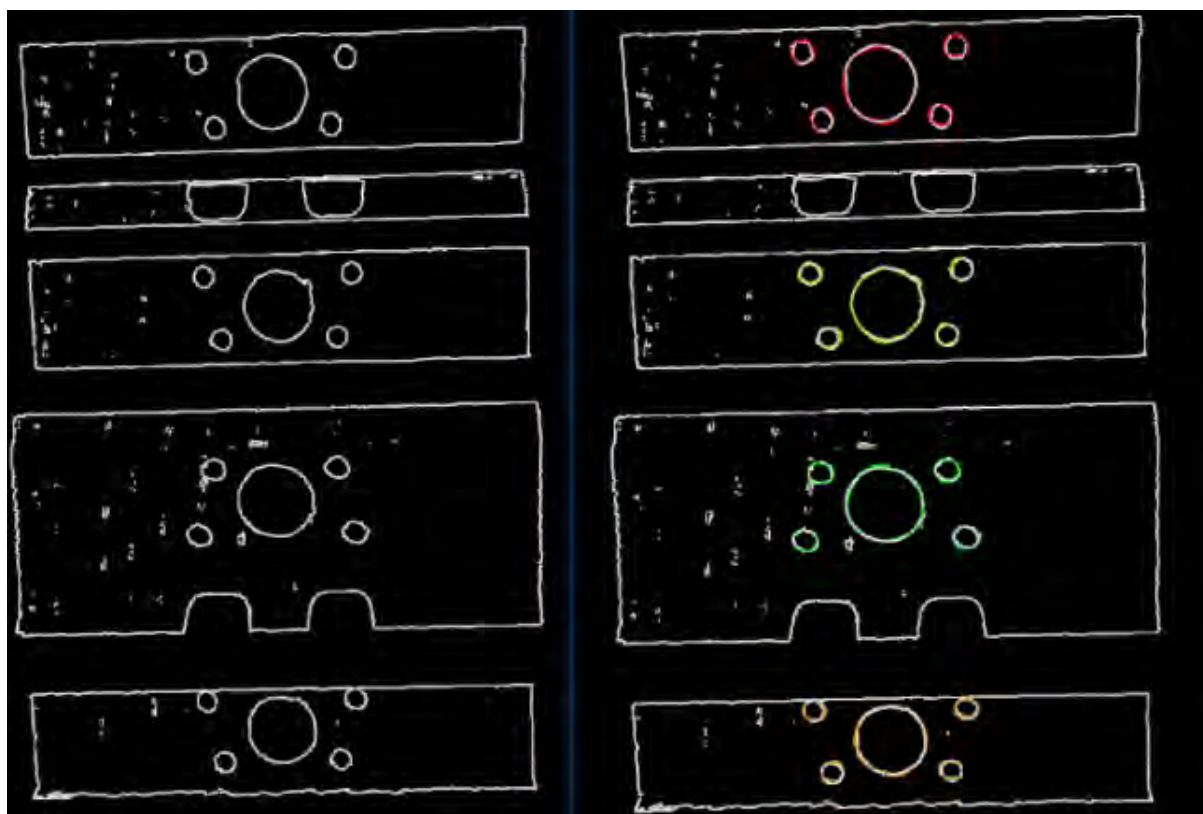
対応の設定

初期値： **GMM**

オプション： **GMM**、 **nearest-neighbor**

調整説明：このパラメータは局所マッチング方法を選択し、モデル点群とシーンの点群をマッチングします。モデル点群は前のステップで取得したシーンの点群とマッチングするために、設定した範囲によって移動・回転します。

調整の例：ほとんどのシーンでは、**GMM** はより優れている外乱耐性とより速いマッチング速度を実現できるのでこのモードを推奨します。特別な場合（GMMでは要求を満たさない時）、最近点（**nearest-neighbor**）モードを試行してください。



GMMモードパラメータ設定

マッチングモード

初期値：標準

オプション：高速、標準、高精度

調整説明：マッチングモードは3つあり、実際の状況に応じて適切なものを選択してください。

高速：速度が最も速いですが、精度は低いです。 **標準**：安定的に実行できるモードです。 **高**

精度：高精度を実現できますが、速度が遅いです。

反復回数

初期値：30

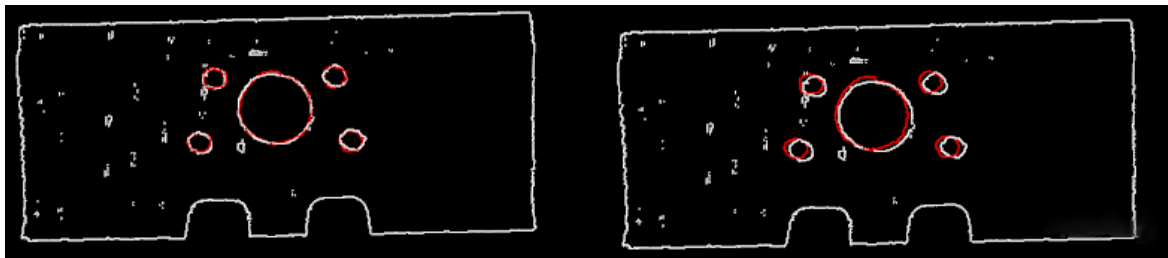
調整説明：反復とは、条件が満たされるまで、特定の命令に対してループを実行することです。反復回数はこのプロセスにコマンドを繰り返す回数を指し、ここで設定するパラメータは反復回数の上限です。この値が大きいほど、マッチング計算する回数は多くなり、実行時間も長くなりますが、マッチング精度は向上します。

標準偏差

初期値：3.000 mm

調整説明：このパラメータは元位置姿勢の偏差と対応します。この前のステップ（[3D位置姿勢低精度推定](#)）で処理された元位置姿勢の偏差が大きい、即ちシーンの点群とモデルの点群の偏差が大きいほど、**標準偏差** を大きくしなければなりません。この時、実行速度は速くなります。

調整の例：設定された値が元の位置姿勢より大きい場合に、出力結果に大きな誤差が生じることがあります。下図に示すように、右側の **標準偏差** は0.04mで、その元の位置姿勢よりずっと大きいので、初期値（下図の左側）と比較すれば、出力された結果の精度は大幅に低下します。通常、このパラメータの値は0.01m以下です。



小さいワークのマッチングでは、より良いマッチング結果を得るために、標準偏差を低く調整することを推奨します。

標準偏差減衰の更新ステップサイズ

初期値：3

調整説明：マッチング処理中、**標準偏差** は**最小標準偏差** で設定された値に達するまでどんどん小さくなります。その過程で標準偏差が減衰する回数がこのパラメータの値です。標準誤差が大きい場合に、減衰の回数が多いほど、減衰が安定的に実行できるようになります。普通、このパラメータを調整する必要はありません。

多数の対象物のマッチングを高速化

初期値：チェックを入れない。

調整説明：シーンに多数の対象物がある場合は、このパラメータにチェックを入れることを推奨します。チェックを入れると、多数の対象物をマッチングする場合はこのステップが高速になりますが、少数の対象物をマッチングする場合は遅くなります。

最近点モードのパラメータ設定

反復設定

最近点の検索半径

初期値：10.000 mm

有効範囲：[0.0010,+∞)

調整説明：このパラメータは最近点の検索半径を調するために使用されます。その値の設定は入力された元位置姿勢の偏差と対応します。入力された元位置姿勢の偏差が大きい時、即ちシーンの点群とモデルの点群との偏差が大きい時に、このパラメータの値を大きくして最近点の検索半径範囲を広めます。こうしてあらゆる点は最近点を見つけられます。元位置姿勢の偏差が小さい時にこの値を小さくします。このパラメータの最小値は0.001mです。この値より小さい場合、一部分の疎らな点は最近点を見つけられなくて出力の結果に影響が出ます。

平均二乗誤差しきい値

初期値：0.001

調整説明：誤差の二乗和の平均値を調整します。反復するたびにこの値が記録され、その後の反復ステップの結果を比較するために使用されます。設定した値より小さい誤差の二乗和の平均値が出たら、期待された結果を取得したことになり、反復を中止します。

ウィンドウサイズ

初期値：10

有効範囲：[3,+∞)

調整説明：このパラメータは反復プロセスに誤差変動が小さい連続反復回数です。この値が小さい場合に、局所最適解が起こり得るので最終的なマッチング精度に影響がでます。例えば、ウィンドウには三回の連続反復の結果の誤差が含まれるので、この三回の反復の誤差を比較して誤差変動のトレンドを取得するしかないので全部の最適解（反復回数推奨値を使用して全体は30回の反復があり、後にさらに明らか誤差が小さくなった反復が起こり得る）でなく、一部だけの最適解を取得（二回目の反復の誤差は一回目と三回目より小さい）するかもしれません。

対応する点を表示

初期値：チェックを入れない

調整説明：チェックを入れれば、毎回の反復に対応する点を表示します。

複雑形状の対象物であるか

初期値：チェックを入れない

調整説明：対象物が複雑形状の対象物（単純な幾何学的形状ではない）であり、**反復ポイントペアの重みを自動的に計算**が必要な場合は、このパラメータにチェックを入れる必要があ

ります。

反復ポイントペアの重みを自動的に計算

初期値：チェックを入れない

調整説明：チェックを入れれば、反復ポイントペアの重みを自動的に計算します。マッピングの結果が良くない場合にこれを使用してください。

ポイントペアの除去設定

異常ポイントペア除去

初期値：チェックを入れない

調整説明：最近点を検索する過程に複数のポイントペアが生じるので、この機能を使用してポイントペアをフィルタリングします。チェックを入れれば、この機能を使用し、**ポイントペア除去シグマ複数しきい値**と**ポイントペア角度差のしきい値**パラメータを使用して異常ポイントペアを除去します。チェックを外すと、その二つのパラメータは無効になります。

ポイントペア除去シグマ複数しきい値

初期値：1.0000

調整説明：シグマ複数しきい値とはペアポイント間の距離分散の倍数の値です。このパラメータはペアポイント間の距離によってフィルタリングします。ペアポイント間の距離がこの範囲を超えたら、このペアポイントは除去されます。

ポイントペア角度差のしきい値

初期値：45°

調整説明：ペアポイントの法線ベクトルの角度がこの値より大きい場合、このペアポイントは除去されます。

反復で重みを使用

初期値：チェックを入れない

調整説明：チェックを入れれば、モデルの重みがアルゴリズムの反復に使用されるので、重みモデルをより正確にマッチングすることができます。

重複ポイントペア除去

初期値：チェックを入れない

調整説明：チェックを入れれば、重複ポイントペアを除去して**重複点群タイプ除去**は有効になります。モデル点群の点数がシーンの単一物体の点数より多い場合に使用することを推奨します。

重複点群タイプ除去

初期値：MinDis

オプション：MinDis、GlobalMin

調整説明：このパラメータはペアポイントの重複対応を削除する方法を選択します。二つの方法があります。

MinDis：対応が重複するポイントペアの間の距離を計算し、距離が最も小さいポイントペアを保持して残りのポイントペアを除去します。**GlobalMin**：対応が重複するポイントペアの間の距離の平均値を基準として、この平均値との差が最も小さいポイントペアを保持して、残りのポイントペアを除去します。

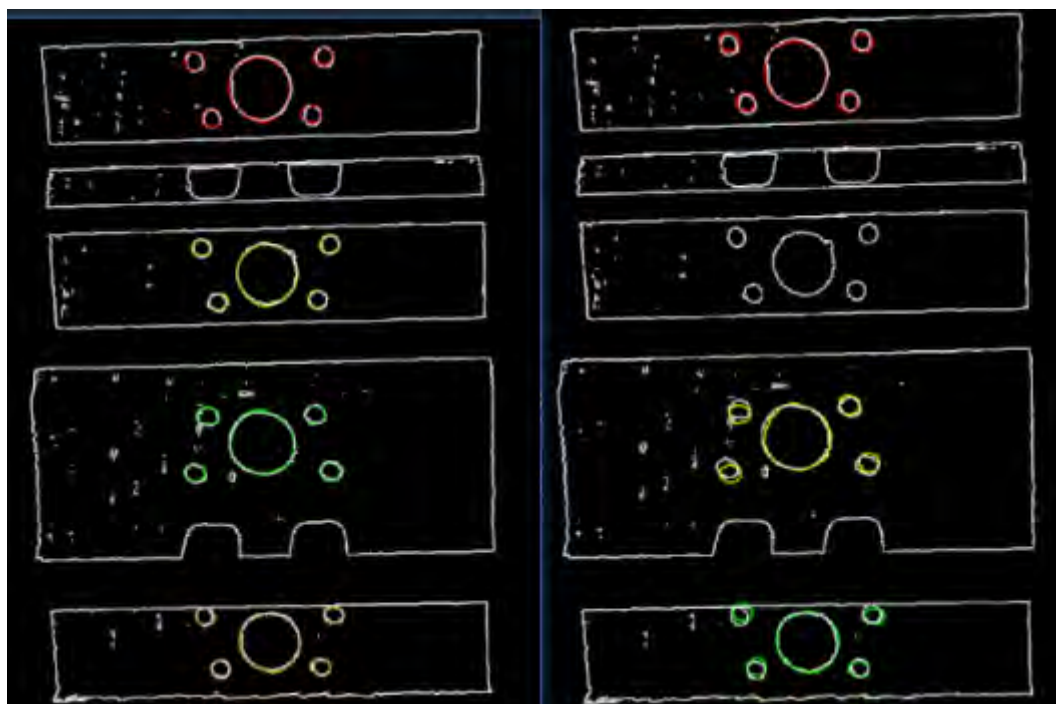
サンプリング設定

サンプリング間隔

初期値：5.000 mm

調整説明：モデルとシーンの点群ダウンサンプリングのためのサンプリング間隔です。この値を高くすれば、サンプリングによって取得する点群は少なくなり、モデルマッチングの精度も低下します。

調整の例：下図は0.005（左）と0.025（右）にしたときに取得した点群の結果です。白いのは入力されたシーンの点群です。それらを比較すれば、サンプリング間隔が小さいほど、取得する出力結果は精確になります。



結果検証設定

信頼度しきい値

初期値：0.5

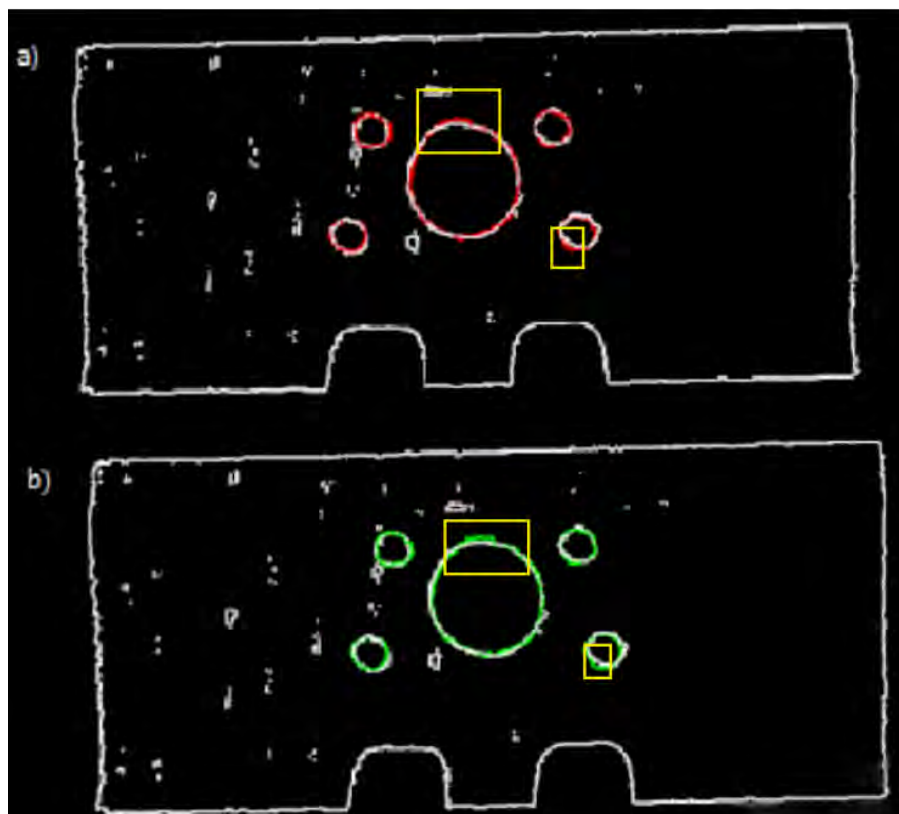
有効範囲：[0, +∞)

調整説明：結果の検証スコアがこのしきい値よりも大きい場合、結果は有効であると見なされます。高いしきい値有効なマッチングが少なくなることを意味しますが出力の精度は高くなります。低いしきい値は有効なマッチングが多くなることを意味しますが出力の精度は低くなります。

調整の例：下図では、aの **信頼度しきい値** は0.500で、bの値は0.92です。bの点群マッチングの結果はaより精度が高いです。そのため、 **信頼度しきい値** が大きいほど、精度の高いマッチングの結果が保持される可能性は高くなります。



しきい値を高くしたら、特に対象物の異なる部分に同じ形状がある場合にマッチングを出力しないこともあります。



結果評価の検索半径

初期値：10.000 mm

有効範囲：[0, +∞)

調整説明：このパラメータの値は、位置姿勢のマッチングスコアに影響します。この値が大きいほど位置姿勢のマッチング結果の評価が甘くなり、位置姿勢のマッチング精度が出なくても位置姿勢のマッチングスコアが高くなります。この値が小さいほど位置姿勢のマッチング結果の評価が厳しくなります。対象物の点群に応じて設定する必要があります。点群がまばらな場合は、このパラメータを適切に上げることができます。



まずは初期の **結果評価の検索半径** を設定し、ステップを実行して可視化出力ウィンドウで

位置姿勢のマッチング効果を確認し、位置姿勢のマッチングスコアと対比します。マッチング効果とマッチングスコアが一致していない場合、実際の状況に応じて **結果評価の検索半径** の値をマッチング効果とマッチングスコアが一致するまで調整することができます。

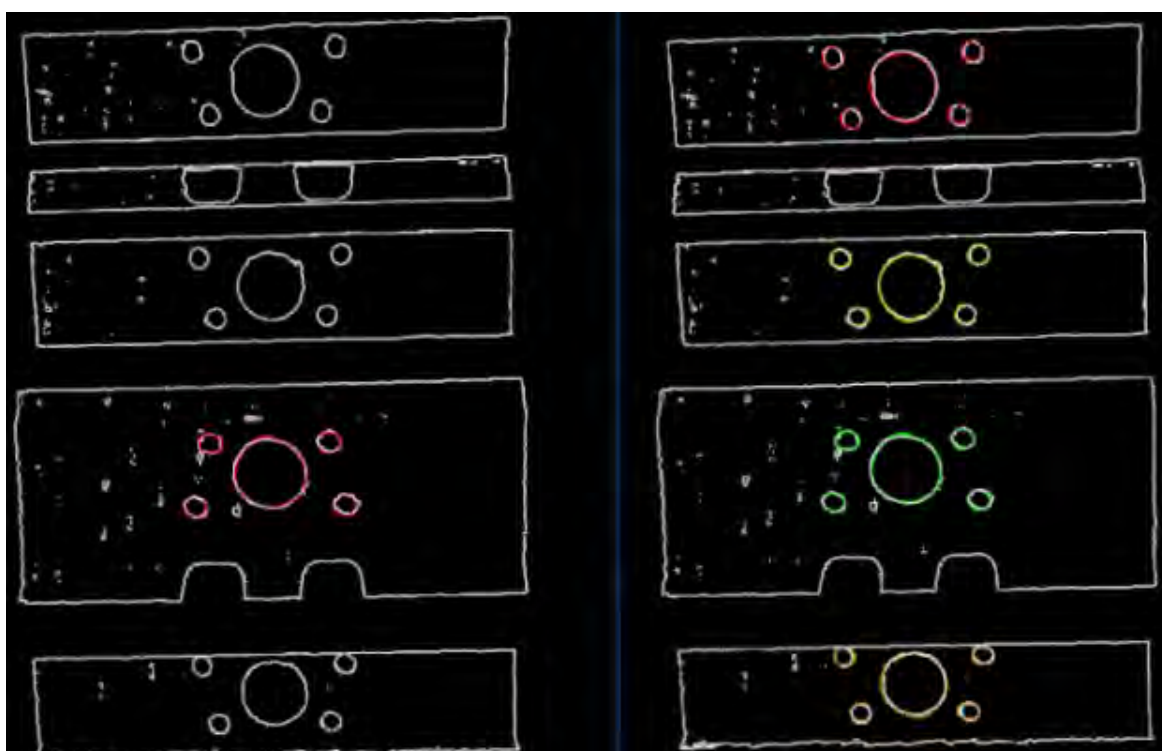
出力設定

単一点群の出力結果の数

初期値：1

調整説明：この値が大きいほど、マッチング結果の出力は多くなります。

調整の例：このパラメータを調整して異なるマッチング出力結果を取得します。下図の左側はこのパラメータを1にした結果で、右側は4にした結果です。



結果可視化の設定

マッチング結果を表示

初期値：チェックを入れる

調整説明：チェックを入れれば、出力されたモデルとシーンの点群が表示されます。

詳細設定

詳細設定に新たに追加した調整可能なパラメータについて説明します。基本設定と同じパラメータがあれば、[基本設定](#) をご参照ください。

対称性設定

認識する部品が局所対称性を持つ場合、局所最適解を回避するために、対称性を設定する必要があります。幾何学的中心点の軸を回転軸として指定することにより、モデルは設定された軸を中心に回転してマッチングし、最適なマッチング結果を取得します。

対称回転軸

初期値：ROTATE_BY_Z

オプション：ROTATE_BY_X、ROTATE_BY_Y、ROTATE_BY_Z

調整説明：このパラメータは、幾何学的中心の回転軸を選択するために使用されます。

対称角度ステップサイズ

初期値：360°

調整説明：このパラメータは対称角度を調整します。

調整の例：対象物が対称軸を中心に60°回転する前後で同じように見える場合、その対称角度ステップサイズは60°です。

最小回転角度

初期値：-180°

調整説明：このパラメータは最小回転角度を調整するために使用されます。

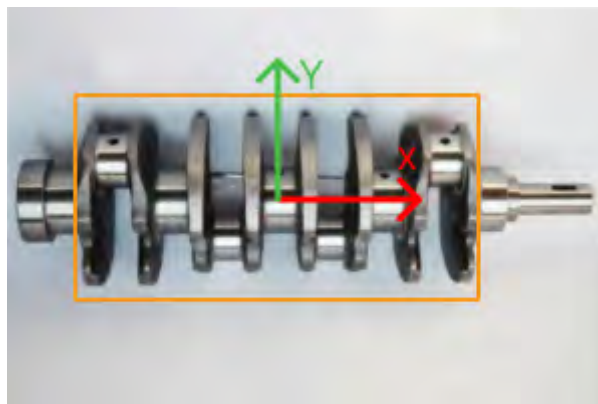
最大回転角度

初期値：180°

調整説明：このパラメータは最大回転角度を調整するために使用されます。

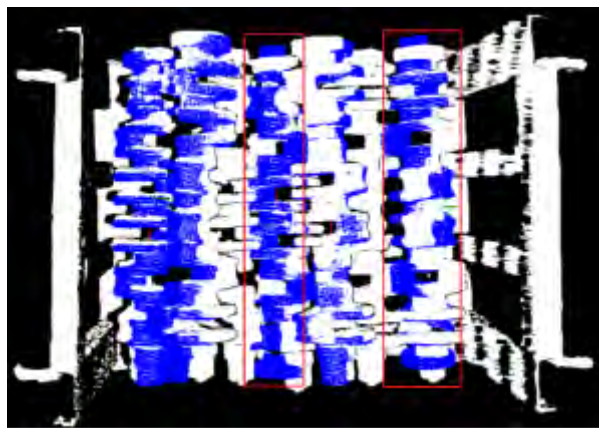
調整の例

下図に示すクランクシャフトの場合、その主要部分、つまりボックス内の部分は、Y軸に沿って180°対称です。



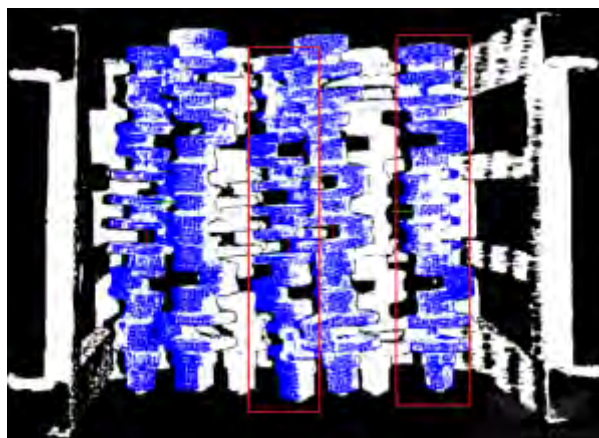


対称性が設定されていない場合、下図に示すように、局所的な形状が類似しているため、マッチング結果が局所最適になり、マッチングエラーが発生する可能性があります。



エラー結果を分析すると、モデルはクランクシャフトの中間部分のみにマッチングします。モデルがマッチング中に180°回転した後にマッチングを試行すると、クランクシャフト全体がマッチングします。また、そのマッチングスコアは、回転前のマッチングスコアよりも高くなるはずです。そうすれば、正しい結果を得ることができます。

下図に示すように、対称回転軸をROTATE_BY_Y（幾何学的中心点のY軸）に設定し、対称角度ステップサイズを180°に設定すると、正しいマッチング結果が得られます。



位置姿勢フィルタリング設定

モデルの回転角度による位置姿勢をフィルタリング

パラメータ説明：エッジマッチングモードでは、シーンの点群にマッチングするために点群モデルをある角度だけ回転させる必要がある場合があり、点群モデルの回転角度で位置姿勢が除去されます。モデルの回転角度が「角度差の上限しきい値」を超えると、その位置姿勢は除去されます。

初期値：オフの状態。

角度差の上限しきい値

詳細については、「モデルの回転角度による位置姿勢をフィルタリング」パラメータの説明をご参照ください。

初期値：135.000°

検証時のモデルの重み

認識する部品の「主体」が類似して「局所」だけが異なった場合、モデルの「局所」の重みを大きくして「局所」のマッチングの結果を出力します。

モデルの重みファイル

調整説明：このファイル（ply形式）は元の点群モデルファイルの一部分の点群で、点群のこの部分は重みが高く、パラメータ **重み値** によって設定されます。

重み値

初期値：2.0

調整説明：このパラメータは点群に重みを設定します。すべての元点群の重み係数が1とすれば、目標点群にこのパラメータを設定したら、目標点群の重み係数は **1 X このパラメータ** になるので、後続のマッチングステップでは強調を実現できます。

重み設定の検索半径

初期値：0.0030

調整説明：このパラメータは点群に重みを増加する過程に検索半径を設定します。重みの設定は元のモデル点群に基づいて実行し、モデル点群はマッチングに使用される前にダウンサンプリングされるので重み設定が必要な点が移動したり損失したりすることはあります。そのような点の付近の点に重み設定するプロセスに、検索半径を設定する必要があります。この検索半径はこのパラメータによって設定します。

結果検証設定

結果評価時の点から法線ベクトルへの角度偏差を考慮

初期値：チェックを入れない

調整説明：チェックを入れれば、位置姿勢を検証する過程に対応するポイントペアの法線ベクトルの角度差を考慮するので、出力の数が少なくなりますが精度が高くなります。

スコアに対応する点がシーンの点群に占める比例をかける

初期値：チェックを入れない

調整説明：チェックを入れれば、マッチングのスコアに「対応する点がシーンの点群に占める比例」をかけます。1つの点群に複数の物体位置姿勢がある場合に適していません。

結果可視化の設定

サンプリングされたモデル点群を表示

初期値：チェックを入れない

調整説明：チェックを入れれば、出力の結果にサンプリングされたモデル点群が表示されます。現場の状況に応じて設定します。

サンプリングされたシーンの点群を表示

初期値：チェックを入れない

調整説明：チェックを入れれば、出力の結果にサンプリングされたシーンの点群が表示されます。現場の状況に応じて設定します。

モデルとシーンの点群の対応関係を表示

初期値：チェックを入れない

調整説明：チェックを入れれば、出力の結果にモデルとシーンの点群の対応関係が表示されます。現場の状況に応じて設定します。

4.3.10. ワーク認識

機能

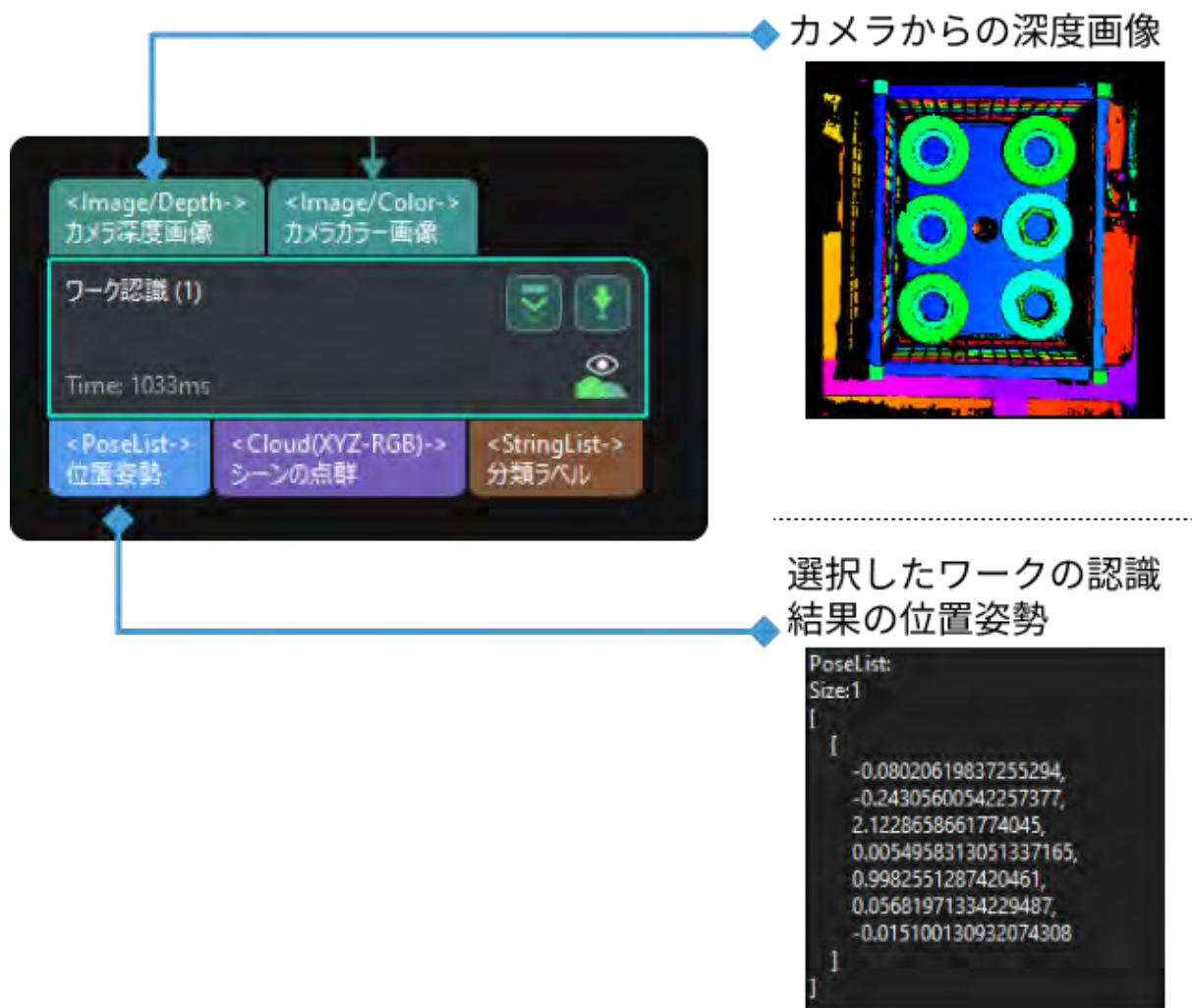
このステップには、点群前処理、3Dマッチング、積み重ねられた対象物の除去に対応可能なビジョン処理機能が内蔵されています。これにより、ワークを迅速に認識することを実現します。

使用シーン

このステップは、金属部品供給シーンで様々な形状のワークや配置状態に対応します。単独に配置されるワーク、単段に整列されるワーク、複数段に整列されるワーク、バラ積みのワークなどを認識するために使用されます。通常、[\[vision-steps:capture-images-from-camera:::capture-images-from-camera\]](#) ステップと併用されます。

通信方式は標準インターフェースまたはAdapter通信となります。

入力と出力



パラメータの説明

エディタを開く

パラメータ説明：クリックしてワーク認識の可視化設定ツールを開きます。

調整説明：使用方法については、[\[vision-steps:3d-workpiece-recognition-visual-configuration-tool:::3d-workpiece-recognition-visual-configuration-tool\]](#) をご参照ください。

ワーク選択

パラメータ説明：このパラメータは、ドロップダウンリストから認識するワークを選択するために使用されます。ワークは事前に追加しておく必要があります。

4.3.10.1. ワーク認識の可視化設定ツール

本節では、ワーク認識の可視化ツールの使用方法について説明します。

機能

ワーク認識の可視化設定ツールは、いくつかのビジョン処理機能が組み込まれています。次の3

つの手順を実行するだけで、ワーク認識の関連設定を迅速に行うことができます。



- 1. ワーク選択：** [マッチングモデル・把持位置姿勢エディタ](#) でワークの点群モデルを作成し、認識するワークを選択します。
- 2. 前処理：** 有効な認識領域の設定により、領域外の干渉要因を除去し、認識領域内の点群に対して基本処理を行い、認識効率を向上させます。
- 3. ワーク認識：** 精度とタクトタイムの要件を満たすために、3Dマッチングのパラメータを可視化方式で調整します。必要に応じてディープラーニングと組み合わせて使用することができます。

使用手順

使用手順は以下の通りです。

ワーク選択

- 1. [ワーク選択]** をクリックします。すると、モデルライブラリが表示されます。
- 2. [モデルエディタを開く]** をクリックしてマッチングモデル・把持位置姿勢エディタを開きます。[マッチングモデル・把持位置姿勢エディタ](#) を参照して把持するワークのマッチングモデルを作成して、このツールを終了します。
- モデルライブラリで把持するワークの点群モデルにチェックを入れ、マッチングモードを選択してから、**[次へ]** をクリックして前処理を行います。



モデルライブラリには、要件を満たしているワークの点群モデルのみが表示されます。

ワークの点群モデルには、少なくとも1つの把持位置姿勢（幾何学的中心点としても使用可能）が設定されていることが必要です。

前処理

- 1. [設定]** をクリックし、表示されるROI設定画面で [3D ROIを設定](#) します。



- 3D ROIが設定後、**[保存して適用]** をクリックします。



[ステップを実行] をクリックすると、前処理結果が画面の左側に表示されます。



- エッジ検出とノイズを除去のパラメータを設定し、ノイズを除去します。この2つのパラメータは、エッジマッチングモードでのみ調整可能です。



前処理後の2D画像と3D画像を表示するには、画面の上部に対応するオプションを切り替えます。



- 前処理後、[次へ] をクリックしてワークの認識を実行します。

ワーク認識

詳細な手順は以下の通りです。

- 3Dマッチングと結果判定に関連するパラメータを設定します。詳細については、[パラメータ説明](#) をご参照ください。

2. (オプション) ディープラーニングの使用が必要な場合、**ディープラーニングの使用** のドロップダウンリストからCPUまたはGPUを選択してから、モデルパッケージ管理ツールのボタンをクリックします。
3. ディープラーニングモデルパッケージのツール画面で、使用するディープラーニングモデルパッケージをインポートしたら終了します。その後、ドロップダウンリストからインポートしたモデルパッケージを選択し、信頼度しきい値を調整します。
4. ボタン **[ステップを実行]** をクリックすると、3Dマッチングの認識結果が画面の左側に表示されます。
5. 認識精度とタクトタイムが要件を満たしている場合、**[完了]** をクリックしてワーク認識の可視化設定ツールが自動的に終了します。または、**[次のワーク]** をクリックして次のワーク認識を実行します。

パラメータ説明

3Dマッチング

実行モード

パラメータ説明：このパラメータは、ワークの認識過程において認識の精度とパフォーマンスを設定するために使用されます。

オプション：高速、標準、高精度、超高精度。

- 高速：単独に配置されるワーク、または少量の整列されるワークに対応可能です。
- 標準：一般的なワーク供給に適しており、速度と精度のバランスを考慮した戦略です。
- 高精度：大量のワーク、またはバラ積みのワークに対応可能です。
- 超高精度：高精度位置決めが必要なシーンにおすすめです。

調整アドバイス：認識精度とパフォーマンスの要件に応じてこのパラメータを調整する必要があります。

偏差補正能力

パラメータ説明：このパラメータは、偏差補正能力のレベルを設定するために使用されます。レベルが大きいほど、ずれた位置姿勢を正確なものに補正することができますが、大きすぎると精度が落ちます。

オプション：高レベル、中レベル、低レベル。

- 低レベル：ほとんどの場合に適用可能です。
- 中レベル：実際の状況に応じて選択する必要があります。
- 高レベル：実際の状況に応じて選択する必要があります。

調整アドバイス：マッチング結果が実際のワークと大きな偏差がある場合、このパラメータ

を調整する必要があります。

対称性設定

パラメータ説明：認識するワークが局所的な対称性を持っている場合は、マッチング結果が局所最適にならないように対称性の設定を大きくし、より良いマッチング結果を得ることができます。

調整説明：詳細は以下の通りです。

- 回転対称軸：ワークの幾何学的中心のある軸を回転軸として選択します。
- 対称角度のステップサイズ：回転軸を中心に60°回転する前後でワークが同じに見える場合、このパラメータを60°に設定することができます。
- 最小回転角度：ワークを対称軸を中心に回転させることができる最小の角度のことです。
- 最大回転角度：ワークを対称軸を中心に回転させることができる最大の角度のことです。

位置姿勢フィルタリング設定

パラメータ説明：認識結果に異常な向きの位置姿勢がある場合、異常な認識結果を除去するために位置姿勢のフィルタリング設定を行います。エッジマッチングの場合、これを使用することを推奨します。

モデルの回転角度による位置姿勢をフィルタリング：エッジマッチングモードでは、シーンの点群にマッチングするために点群モデルをある角度だけ回転させる必要がある場合があり、点群モデルの回転角度で位置姿勢が除去されます。モデルの回転角度が「角度差の上限しきい値」を超えると、その位置姿勢は除去されます。

結果判定

3Dマッチングの信頼度しきい値

パラメータ説明：このパラメータは、3Dマッチングの認識結果の信頼度しきい値を設定するために使用されます。3Dマッチングの認識結果の信頼度がこの値より大きい場合、有効な結果と見なされます。値が高いほど、より正確な結果を得ることができます。

初期値：0.300

調整アドバイス：実際の状況に応じて設定する必要があります。

最大出力数

パラメータ説明：このパラメータは、3Dマッチングの認識結果の最大数を設定するために使用されます。複数の結果がある場合、信頼度で降順にソートし、設定された **最大出力数** に従って信頼度の高い結果を出力します。

初期値：1

調整アドバイス：認識結果が多い場合、または結果の数に対して要件がある場合、このパラ

メータを調整する必要があります。



- 取得されたワークの把持点には、幾何学的中心であり把持点でもある点のみを表示します。
- 実際に出力される3Dマッチングの認識結果の数は、設定された **最大出力数** と一致しない場合があります。例えば、設定された **最大出力数** を5に設定しても3Dマッチングの認識結果が合計で3つしかない場合、実際に出力される結果の数は3になります。

4.3.11. 全てのパラメータを受け入れる

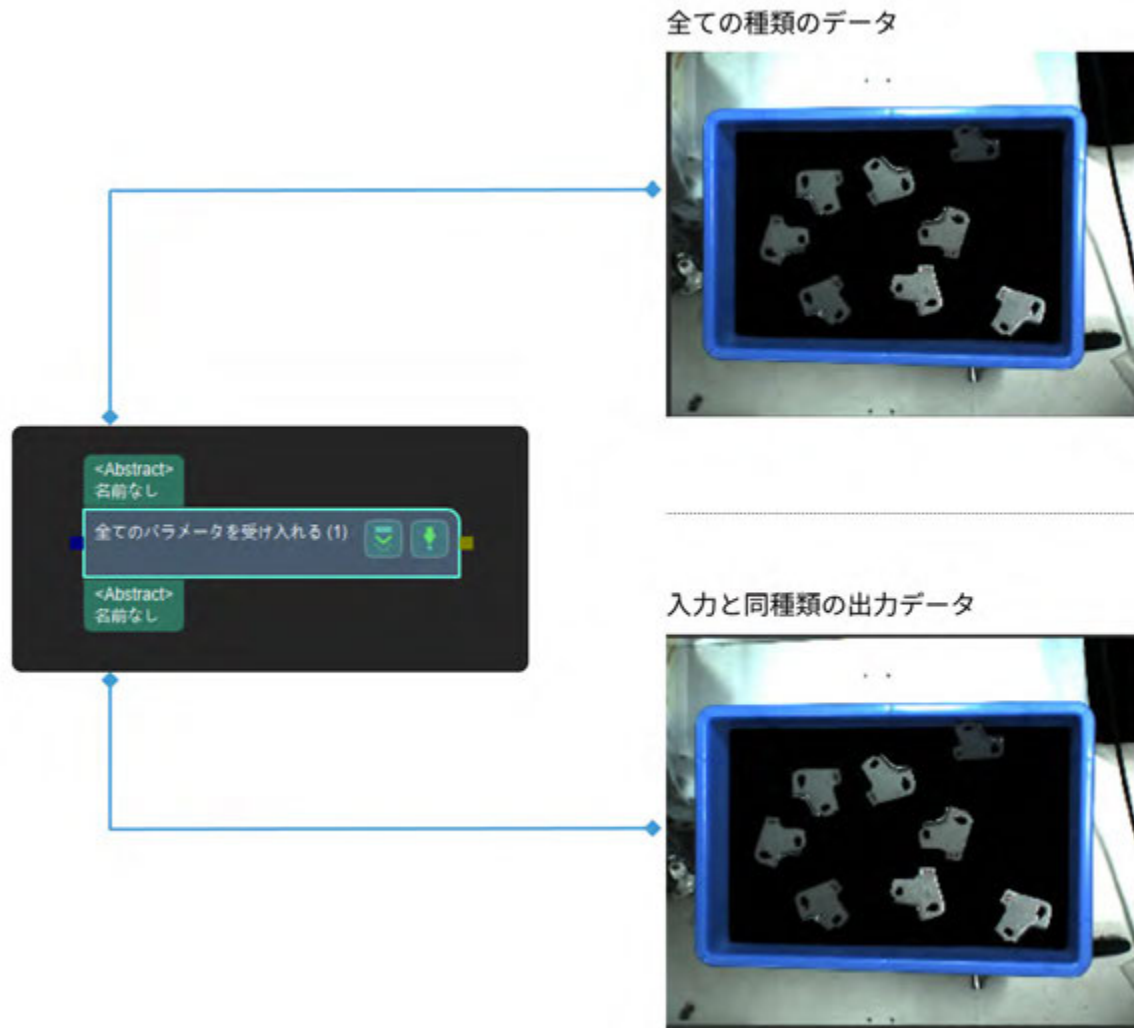
機能

このステップは、その他のステップの出力を全部受け入れるために使用されます。

使用シーン

デバッグ時にあるポートのデータを確認するために使用されます。また、デバッグ時間を短縮するために実行時間が長いステップのポート情報の一時的な保存にも使用されます。

入力と出力



パラメータの説明

パラメータはありません。

4.3.12. 位置姿勢にラベルを付ける

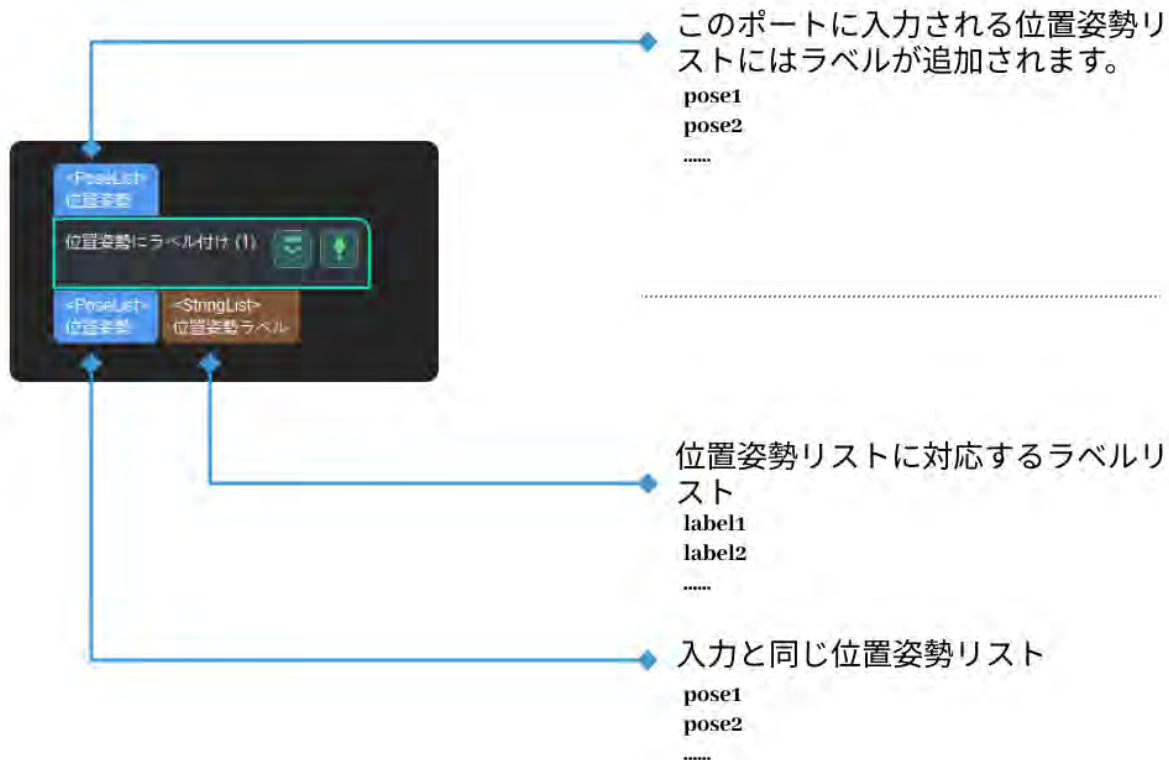
機能

各入力位置姿勢に同じラベルを追加します。

使用シーン

汎用位置姿勢処理ステップであり、特定の使用シーンはありません。

入力と出力



パラメータの説明

ファイル設定

ラベルファイル

初期値：addLabels.json

調整説明：ラベルファイルを保存する絶対パスまたは相対パスです。ファイルには、必要なラベル名が含まれています。

4.3.13. カメラの歪みによる不正確な位置姿勢を調整



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

カメラの歪み係数（内部パラメータに含まれている）を使用して位置姿勢を補正します。

使用シーン

特定のシーン向けのカスタマイズのステップで、位置姿勢に対してカメラの歪み補正に使用されます。

入力と出力

- **入力：** このポートに入力される位置姿勢は調整されます。
- **出力：** 調整された位置姿勢。

4.3.14. オフセットによって位置姿勢を調整



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

円柱の位置姿勢と正確なロボットハンドの先端位置姿勢（把持位置姿勢）の間の位置姿勢オフセットを計算することで、把持位置姿勢を調整します。円柱の実際の位置姿勢および位置姿勢から把持位置姿勢までのオフセットを出力します。

使用シーン

通常、端が密集している円柱形状のバーの把持シーンで、2つのバーを同時に把持するのを防ぐために使用されます。

入力と出力

- **入力：**
 1. 端が密集している円柱形状のバーを含む点群。
 2. 補正するロボットハンドの先端位置姿勢。
 3. 1つの円柱形状のバーの長さ。
 4. 円柱形状の情報。
- **出力：**
 1. 円柱形状のバーの位置姿勢。
 2. 計算された位置姿勢に対する円柱形状のバーの把持位置姿勢オフセット。

4.3.15. 傾きによって位置姿勢を調整



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

指定されたベクトルに対する傾きに基づいて、位置姿勢を一定の距離だけ並進します。並進距離は、位置姿勢に垂直な方向へ指定されたベクトルが投影する長さです。

使用シーン

特定のシーン向けのカスタマイズのステップです。

入力と出力

- **入力：** このポートに入力される位置姿勢は調整されます。
- **出力：** 調整された位置姿勢。

4.3.16. 正確な経路を得るまで目標点を調整



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

正確な経路を得るために、目標点を回転させてZ軸に沿って並進します。

使用シーン

通常、ロボットハンドが経路に沿って運動するときの回転角度を調整するために使用されます。

入力と出力

- **入力：** このポートに入力される目標点は、回転され、Z軸に沿って並進されます。
- **出力：** 回転され、Z軸に沿って並進された目標点。

4.3.17. 位置姿勢を点群の表面に調整



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

入力位置姿勢を指定方向に沿って並進し、その中心が入力点群の表面に位置するようにします。

使用シーン

通常、把持位置姿勢を点群表面に補正するために使用されます。

入力と出力

- **入力：**
 1. このポートに入力される点群は、位置姿勢調整の基準として使用されます。
 2. このポートに入力される位置姿勢は点群表面に調整されます。

● 出力：

1. 点群表面に調整された位置姿勢。
2. 可視化されたカラー点群。

4.3.18. 経路の時計回り方向を調整



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

経路の時計回り方向を調整します。

使用シーン

通常、ドア/ガラス接着関連の応用シーンで使用されます。移動中にロボットがソフトリミットの状態になると、始点と終点が切り替わり、ロボットが新しく設定された位置姿勢に従って移動します。

入力と出力

● 入力：

1. このポートに入力される元の経路は、調整されます。
2. 元の経路でロボットのツールの位置姿勢リスト。
3. フランジ座標系における、元の経路でロボットのツールの位置姿勢リスト。

● 出力：

1. 調整された位置姿勢（これらの位置姿勢は、時計回り方向が調整された経路を形成します）。

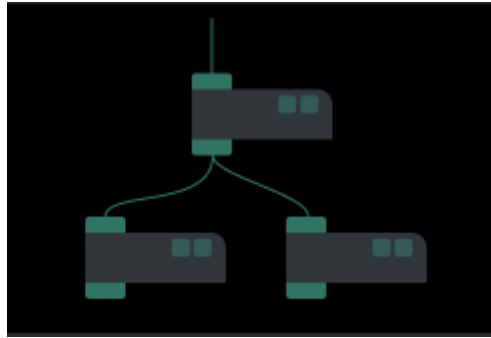
4.3.19. 中継

機能

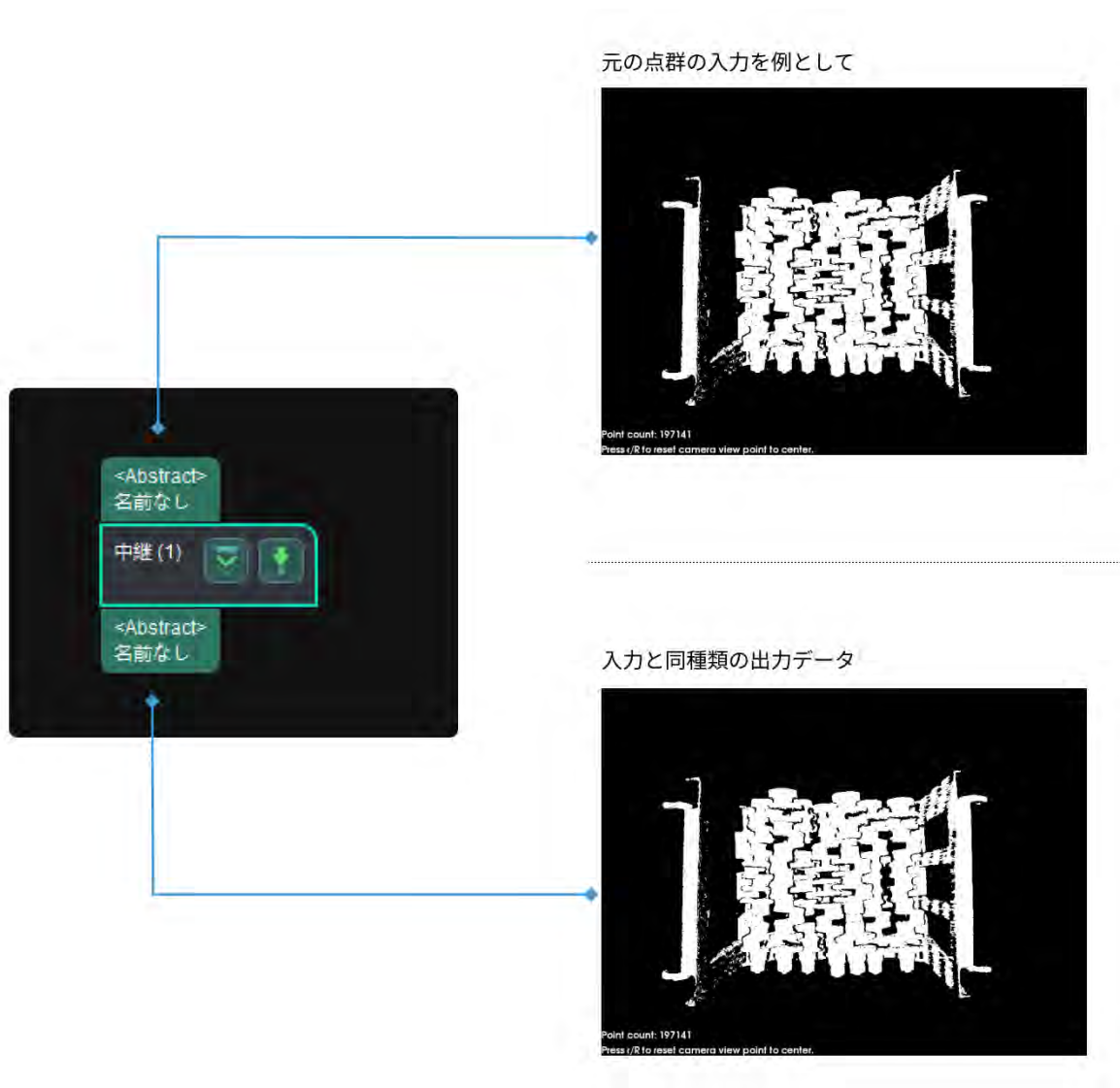
ステップの組合せで入力ポートを分配するために使用されます。

使用シーン

通常、ステップの組合せの入力ポートの数を減少するために使用されます。ステップの組合せの内部にデータを送信し、その内部でポートを分配します。



入力と出力



パラメータの説明

パラメータはありません。

4.3.20. マスク内の対応する画像を抽出

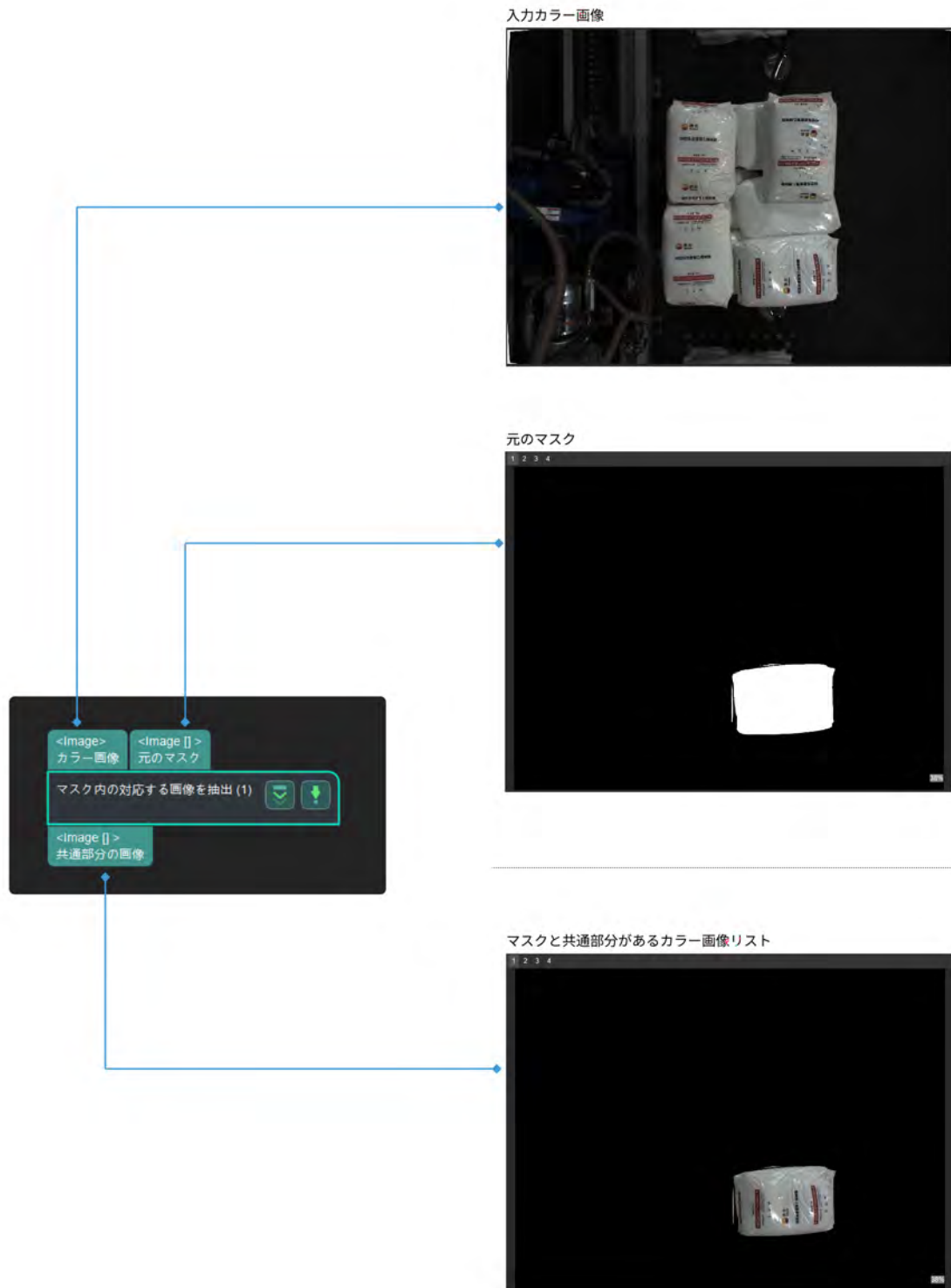
機能

マスクをカラー画像に使用します。マスクで覆われた領域は抽出され、残りの部分は黒で塗りつぶされます。

使用シーン

通常はステップ **最高層マスクを取得** と併用され、シーン内の関係ない部分を無視して干渉を回避するために使用されます。後続のディープラーニングの関連ステップに良好な入力を提供します。

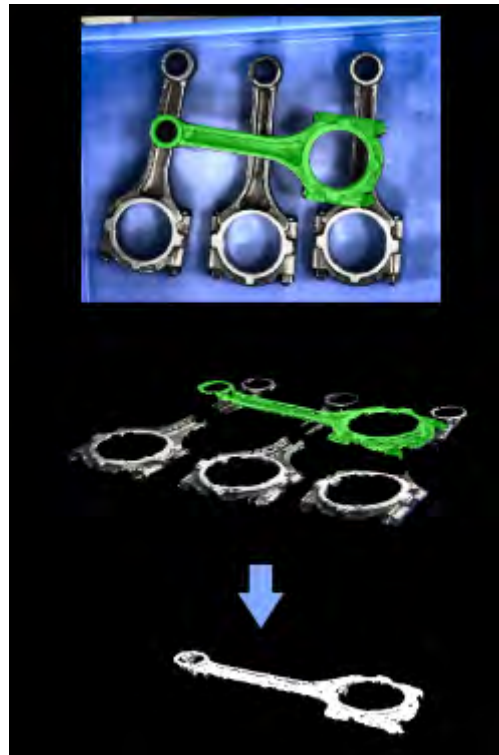
入力と出力



4.3.21. マスク内の対応する点群を抽出

機能

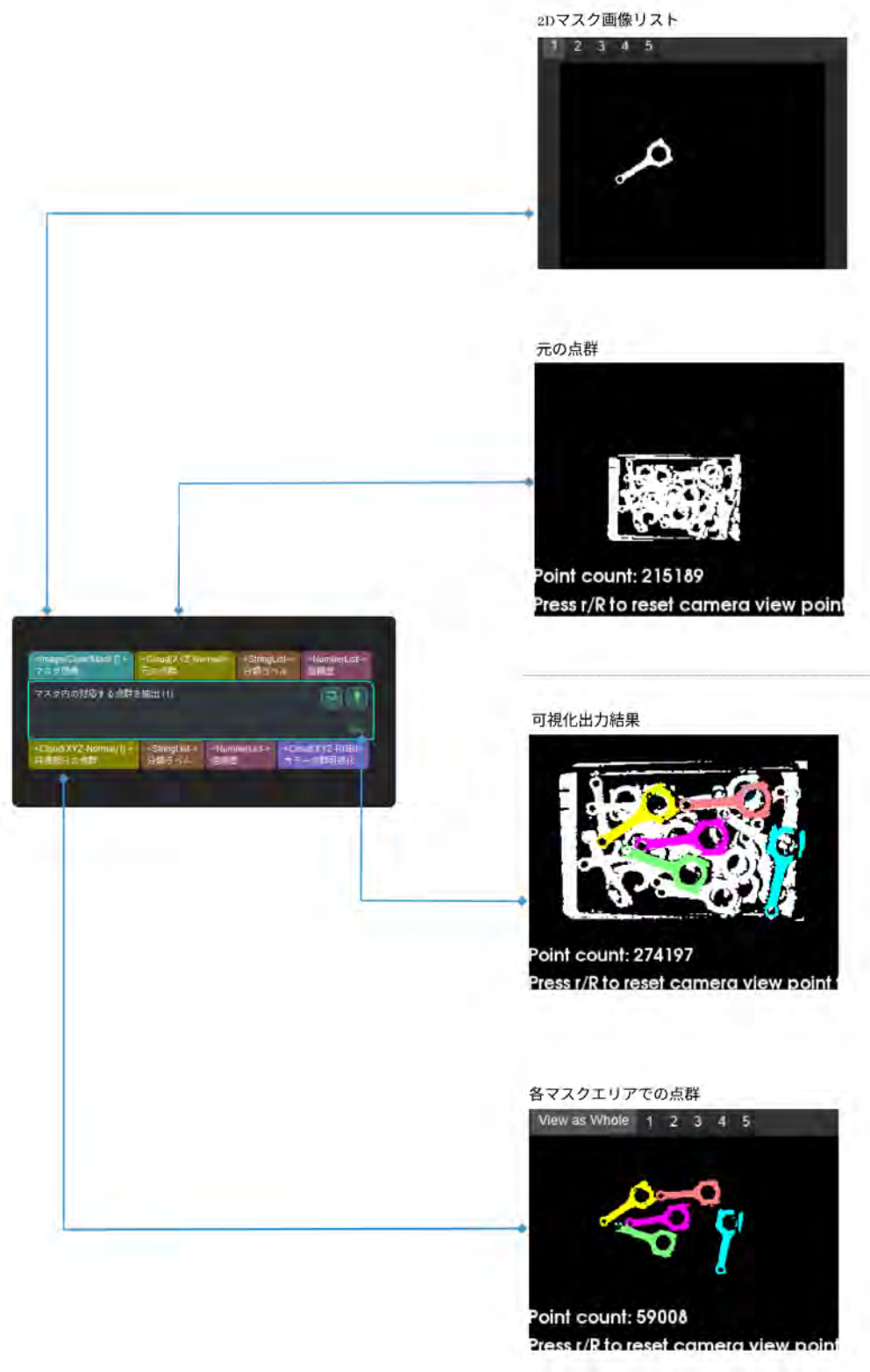
マスクを点群に使用します。マスクで覆われた点群は抽出され、残りの点群は除去されます。



使用シーン

マスクが対応する3D点群を抽出するために使用されます。

入力と出力



パラメータの説明

パラメータはありません。

4.3.22. 2D経路を抽出



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

入力された元の経路の画像と対象物の点群によって、2D経路点を抽出します。

使用シーン

特定のシーン向けのカスタマイズのステップです。

入力と出力

- 入力：
 1. 元の経路の画像。
 2. フィルタリング後の経路の画像（元の画像から減算して画像内の経路を取得するために使用されます）。
 3. 対象物の点群。
 4. 対象物のカラー画像。
- 出力：
 1. 経路の正投影2D画像。
 2. 生成された経路を表示するための2D位置姿勢。

4.3.23. 背景除去



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

2D画像の背景を除去し、前景を保持します。

使用シーン

2D画像の背景によって引き起こされる後続の画像処理への干渉を排除するために使用されます。

入力と出力

- 入力：
 1. このポートに入力される画像は、背景が除去されます。
 2. 入力される画像が対応する深度画像。

● 出力：

1. 背景が除去された画像。

4.3.24. 画像二値化処理

機能

設定されたしきい値によって画像のピクセルをフィルタリングし、設定されたルールに従ってしきい値以上・以下のピクセルを二値化します。



画像のピクセルのグレイ値が最大値と最小値の2つの値、つまり「黒または白のいずれか」しかない場合、このような画像は2値画像と呼ばれます。

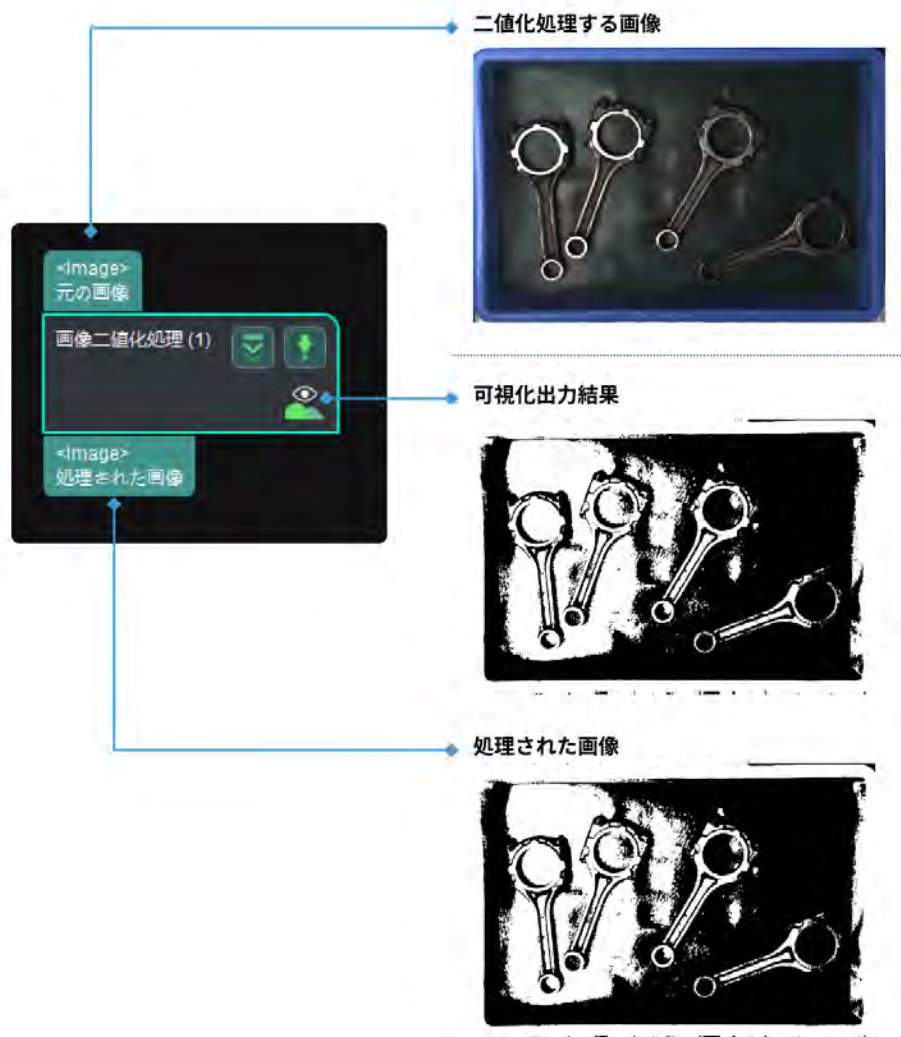
非二値画像を二値画像に演算することを画像の二値化といいます。



使用シーン

汎用の画像処理ステップです。通常、2D画像でしきい値の条件を満たすピクセルを分割するために使用されます。

入力と出力



パラメータの説明

このステップでは、特徴対象物と背景を分割するために、4種類の方法が用意されています。

- **AdaptiveThreshold** : グローバル適応しきい値。THRESH_BINARY、THRESH_BINARY_INVの2つの分割操作タイプを選択可能です。
- **DualThreshold** : 2つのしきい値。この方法は、高しきい値または低しきい値の設定により、しきい値の条件を満たすピクセルを分割できます。
- **DynamicThreshold** : 動的しきい値。4種類のしきい値分割タイプ (Light、Dark、InRange、OutOfRange) と3種類の画像フィルタリング方法 (平均化フィルタ、ガウシアンフィルタ、メディアンフィルタ) を選択可能です。また、いくつかの補助パラメータ (ピクセルのオフセット、フィルタのカーネルサイズ) も設定できます。
- **Threshold** : グローバル固定しきい値。この方法では、固定なしきい値を設定し、7種類の分割操作タイプ (THRESH_BINARY、THRESH_BINARY_INV、THRESH_TRUNC、THRESH_TOZERO、THRESH_TOZERO_INV、THRESH_OTSU、THRESH_TRIANGLE) を用いて、しきい値の条件を

満たすピクセルを分割できます。

AdaptiveThreshold

分割タイプ

パラメータ説明：このパラメータは、特徴対象物と背景を分割する操作タイプを選択するために使用されます。

オプション：THRESH_BINARY、THRESH_BINARY_INV

- THRESH_BINARY：しきい値を超える強度を持つピクセルのマスク画像を生成します。
- THRESH_BINARY：しきい値を下回る強度を持つピクセルのマスク画像を生成します。

初期値：THRESH_BINARY

調整アドバイス：実際のニーズに応じて選択してください。

DualThreshold

低しきい値

パラメータ説明：低しきい値が高しきい値より低く設定された場合、2つのしきい値の間の値を持つピクセルは255に設定され、その他のピクセルは0に設定されます。低しきい値が高しきい値よりも高く設定されている場合、2つのしきい値で設定された間隔を超えた値を持つピクセルは255に設定され、その他のピクセルは0に設定されます。

初期値：0

調整アドバイス：実際のニーズに応じて設定してください。

高しきい値

パラメータ説明：「低しきい値」と同じです。

初期値：100

調整アドバイス：実際のニーズに応じて設定してください。

DynamicThreshold

しきい値分割タイプ

パラメータ説明：このパラメータは、どのピクセルを255に設定し、どのピクセルを0に設定するかを設定します。

オプション：明るい領域、暗い領域、範囲内、範囲外。 P_o を元の画像のピクセルにし、 P_f をフィルタリング後の画像内の P_o が対応するピクセルにし、 **offset** を **ピクセル値のオフセット** パラメータにした場合、各分割タイプの説明は下記ようになります。

- 明るい領域： $P_o \geq P_f + \text{offset}$ の場合、 P_o を255に設定し、それ以外の場合は0に設定します。

- 暗い領域： $P_o \leq P_f - \text{offset}$ の場合、 P_o を255に設定し、それ以外の場合は0に設定します。
- 範囲内： $P_f - \text{offset} \leq P_o \leq P_f + \text{offset}$ の場合、 P_o を255に設定し、それ以外の場合は0に設定します。
- 範囲外： $P_o < P_f - \text{offset}$ または $P_o > P_f + \text{offset}$ の場合、 P_o を255に設定し、それ以外の場合は0に設定します。

画像のフィルタリング方法

パラメータ説明：このパラメータは、画像フィルタリングの方法を選択するために使用されます。

オプション：平均化フィルタ、ガウシアンフィルタ、メディアンフィルタ。

平均化フィルタ：スライディング ウィンドウの中心値をウィンドウ内の全ピクセル値の平均に置き換えることで、画像を滑らかにします。

- ガウシアンフィルタ：画像を滑らかにし、ディテールとノイズを除去します。
- メディアンフィルタ：各ピクセルをウィンドウ内の隣接ピクセルの中央値に置き換えます。

初期値：平均化フィルタ。

調整説明：実際のニーズに応じて選択してください。

フィルタリングのカーネルサイズ

パラメータ説明：このパラメータは、画像フィルタリングのためのスライディングウィンドウの辺の長さ（ピクセル単位）を設定するために使用されます。

初期値：3

調整説明：ウィンドウの中心には常にピクセルが存在する必要があるため、奇数で入力してください。偶数で入力した場合は、1つずつ増加します。

ピクセル値のオフセット

パラメータ説明：フィルタリング後の画像のすべてのピクセルに適用されるオフセットのピクセル値を設定するために使用されます。

初期値：15.00

Threshold

しきい値 (0-255)

パラメータ説明：このパラメータは、画像ピクセルをフィルタリングするためのしきい値を設定するために使用されます。

初期値：128

調整アドバイス：実際のニーズに応じて設定してください。

分割タイプ

パラメータ説明：このパラメータは、特徴対象物と背景を分割する操作タイプを選択するために使用されます。

オプション

：THRESH_BINARY、THRESH_BINARY_INV、THRESH_TRUNC、THRESH_TOZERO、THRESH_TOZERO_INV、THRESH_OTSU、THRESH_TRIANGLE

- THRESH_BINARY：画像のピクセルのグレイ値が設定されたしきい値より大きい場合は、255に設定されます。それ以外の場合は0に設定されます。
- THRESH_BINARY_INV：画像のピクセルのグレイ値が設定されたしきい値より大きい場合は、0に設定されます。それ以外の場合は255に設定されます。
- THRESH_TRUNC：画像のピクセルのグレイ値が設定されたしきい値より大きい場合は、その値が設定されたしきい値に設定されます。それ以外の場合は、元のピクセルのグレイ値を保持します。
- THRESH_TOZERO：画像のピクセルのグレイ値が設定されたしきい値より大きい場合は、元のピクセルのグレイ値を保持します。それ以外の場合は0に設定されます。
- THRESH_TOZERO_INV：画像のピクセルのグレイ値が設定されたしきい値より大きい場合は、0に設定されます。それ以外の場合は元のピクセルのグレイ値を保持します。
- THRESH_OTSU：「Otsu法（大津法）」を使ってグローバルなしきい値を求めます。
- THRESH_TRIANGLE：「三角法」を使ってグローバルなしきい値を求めます。

初期値：THRESH_BINARY

調整アドバイス：THRESH_BINARY、THRESH_BINARY_INVを推奨します。

4.3.25. ブロブ解析



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

入力画像を二値化処理し、画像内のブロブを検出し、その輪郭、ピクセル領域、真円度、および長方形性を計算し、設定された基準に従ってブロブを1つずつ確認します。

使用シーン

通常、計測シーンで2D画像内の対象物を検出するために使用されます。

入力と出力

- 入力：

1. このポートに入力される画像は二値化され、blobが検出されます。
2. このポートに入力される2D変換は、以前設定したROIを適切な位置に変換するために使用されます。

● 出力：

1. 可視化するための画像。
2. blob輪郭の頂点のピクセル座標。
3. blob領域の最小外接長方形の頂点のピクセル座標。
4. 出力されたblobの数。
5. blobのピクセル領域。
6. 各blobの領域をblobのまわりの長さで囲まれた円の面積で割って得られたblobの真円度の値。
7. 各blobの領域をblobの最小外接長方形面積で割って得られたblobの長方形性の値。
8. 出力されたblobの検出結果のbool値のリスト。検出基準は、「検出設定」グループのパラメータによって設定されます。
9. blobのマスク画像（画像の非ゼロピクセル領域はマスクとなります）。

4.3.26. ブールリストの論理演算



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

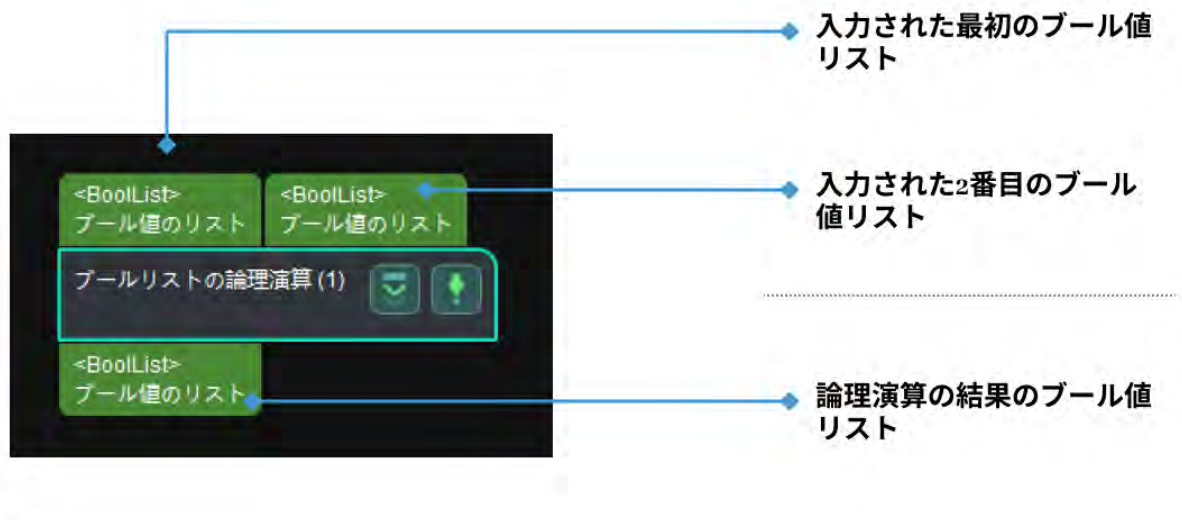
機能

設定された論理演算子に従って、入力bool値（True/False）リストに対して論理演算を実行します。

使用シーン

汎用の論理演算ステップで、特定の使用シーンはありません。

入力と出力



4.3.27. プロジェクト間にデータを送信



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

データをほかのプロジェクトに送信します。

使用シーン

データの保存と読み込みに時間がかかり、プロジェクト間でデータを転送する必要がある場合に使用されます。

入力と出力

- 入力：なし。
- 出力：なし。

4.3.28. 絶対値を計算



これは古いバージョンのステップであり、メンテナンスが停止されていますので、新しいバージョンのステップ [\[vision-steps:numeric-operation:::numeric-operation\]](#) を使用してください。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

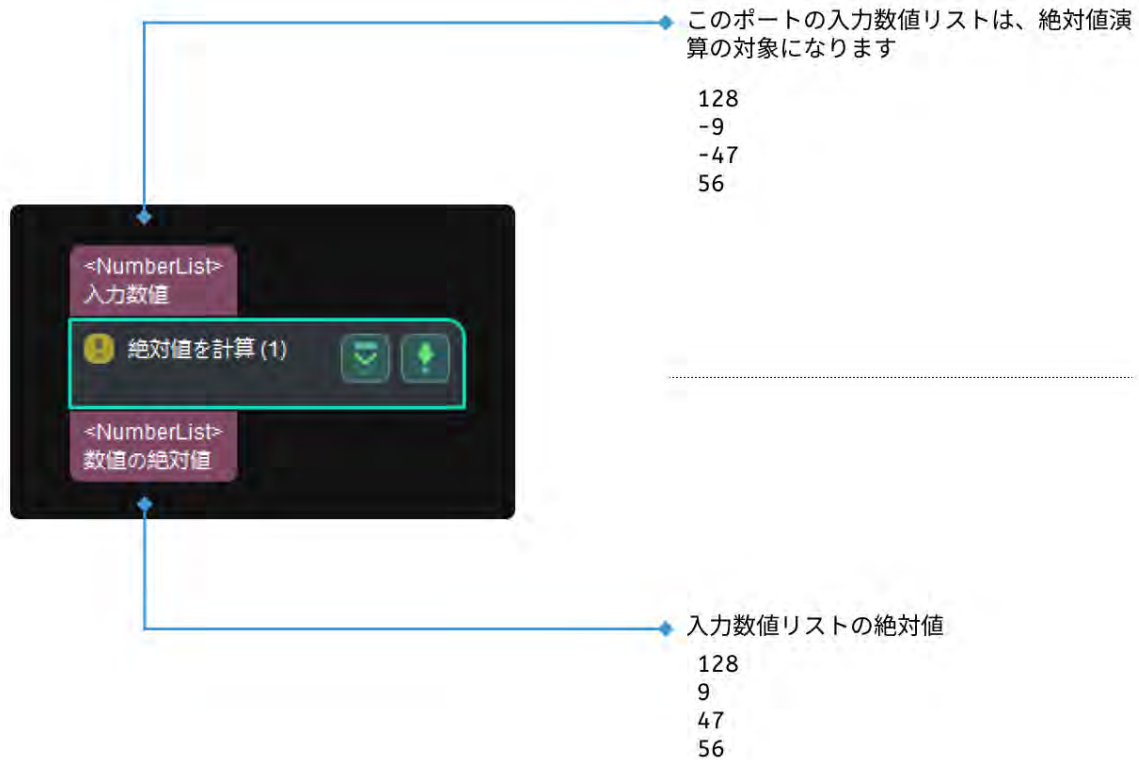
機能

入力数値リストの絶対値を計算します。

使用シーン

これは古いバージョンのステップで、より完全な演算機能を備えた新しいバージョンのステップ [\[vision-steps:numeric-operation:::numeric-operation\]](#) を使用することを推奨します。

入力と出力



4.3.29. 両3Dベクトル間の角度を計算



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社 (docs@mech-mind.net) までお問い合わせください。

機能

2つセットのVector3D型のベクトル間の角度を計算します。

使用シーン

汎用のVector3D型のベクトル間の角度計算ステップです。特定の使用シーンはありません。

入力と出力

● 入力：

1. このポートのVector3D型のベクトルともう一つのポートのVector3D型が対応するベクトル間の角度が計算され、出力されます。

2. このポートのVector3D型のベクトルともう一つのポートのVector3D型が対応するベクトル間の角度が計算され、出力されます。

● 出力：

1. 計算によって得られたベクトル間の角度。

4.3.30. マスク面積を計算



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

複数のマスク画像の面積を計算します。

使用シーン

計算によって得られた面積数値を使用してその後のソート、フィルタリング処理などを行います。

入力と出力



4.3.31. 箱の寸法を計算



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

深度画像および箱のマスクで、箱の上表面の実際の寸法を計算し、箱の寸法と上表面の点群を出力します。

使用シーン

特定のシーン向けのカスタマイズのステップです。

入力と出力

● 入力：

1. このポートに入力される深度画像には、箱の寸法が計算されます。
2. 深度画像の箱のマスク。

● 出力：

1. 計算された箱の上表面の寸法。
2. 箱の上表面の点群。

4.3.32. 3Dベクトルの外積を計算



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

2つセットのVector3Dベクトルの1対1の外積を計算します。

使用シーン

汎用のベクトル計算ステップです。特定の使用シーンはありません。

入力と出力

● 入力：

1. 外積を計算するための最初セットのVector3Dベクトル。
2. 外積を計算するための2番目セットのVector3Dベクトル。

● 出力：

1. 計算によって得られた外積。

4.3.33. 対角線の長さを計算

機能

箱のような対象物をXY平面に投影した後、その投影する対角線の長さを計算します。

使用シーン

特定のシーン向けのカスタマイズのステップです。

入力と出力

- **入力：** このポートに入力される対象物の寸法は、対角線の長さを計算するために使用されます。
- **出力：** 計算された対角線の長さ。

4.3.34. 直径と厚さを計算



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

入力点群の直径と厚さを計算します。

使用シーン

特定のシーン向けのカスタマイズのステップです。

入力と出力

- **入力：**
 1. このポートに入力される点群には、直径が計算されます。
 2. このポートに入力される点群には、厚さが計算されます。
 3. 点群の位置姿勢。
- **出力：**
 1. 計算された直径。
 2. 計算された厚さ。
 3. 対応する計算結果が使用可能かどうかを表示するラベル。

4.3.35. 視差画像を計算



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

入力された補正後のステレオカメラ画像を使用して視差画像を計算します。

使用シーン

カメラのパッシブステレオモードをデバッグするための開発者機能です。普通のユーザーは、一般的な使用シーンでこのステップを使用する必要はありません。

入力と出力

● 入力：

1. ステレオカメラモードで出力された補正後の左側面図。
2. ステレオカメラモードで出力された補正後の右側面図。
3. カメラの視差値から深度値へのマッピング行列（4 x 4行列である必要があります）。
4. 入力サイズを検証するためのスケーリング係数のパラメータリスト（このステップでは、スケーリングされていない元カラー画像が必要であるため、スケーリング係数パラメータのいずれかが1でなければ、ステップは実行されません）。
5. エッジ拡張処理が行われた左側面図（選択可能）。
6. 基準用の深度画像（オプション）。パッシブステレオカメラは、アクティブに投影されたステレオカメラの計算結果を基準として使用して、欠落している点群を融合または補正させることができます。ユーザーは、アクティブなステレオカメラの深度画像をこの入力ポートからこのステップに入力できます。

● 出力：

1. 視差画像の結果。注意：この視差画像は他のステップの深度画像の入力として使用できず、データタイプの変換が必要です。
2. 視差画像が対応する信頼度マスク。深度の不連続性が大きい領域は、低い信頼度で表示されます。

4.3.36. 指定方向に沿って両位置姿勢の距離を計算



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

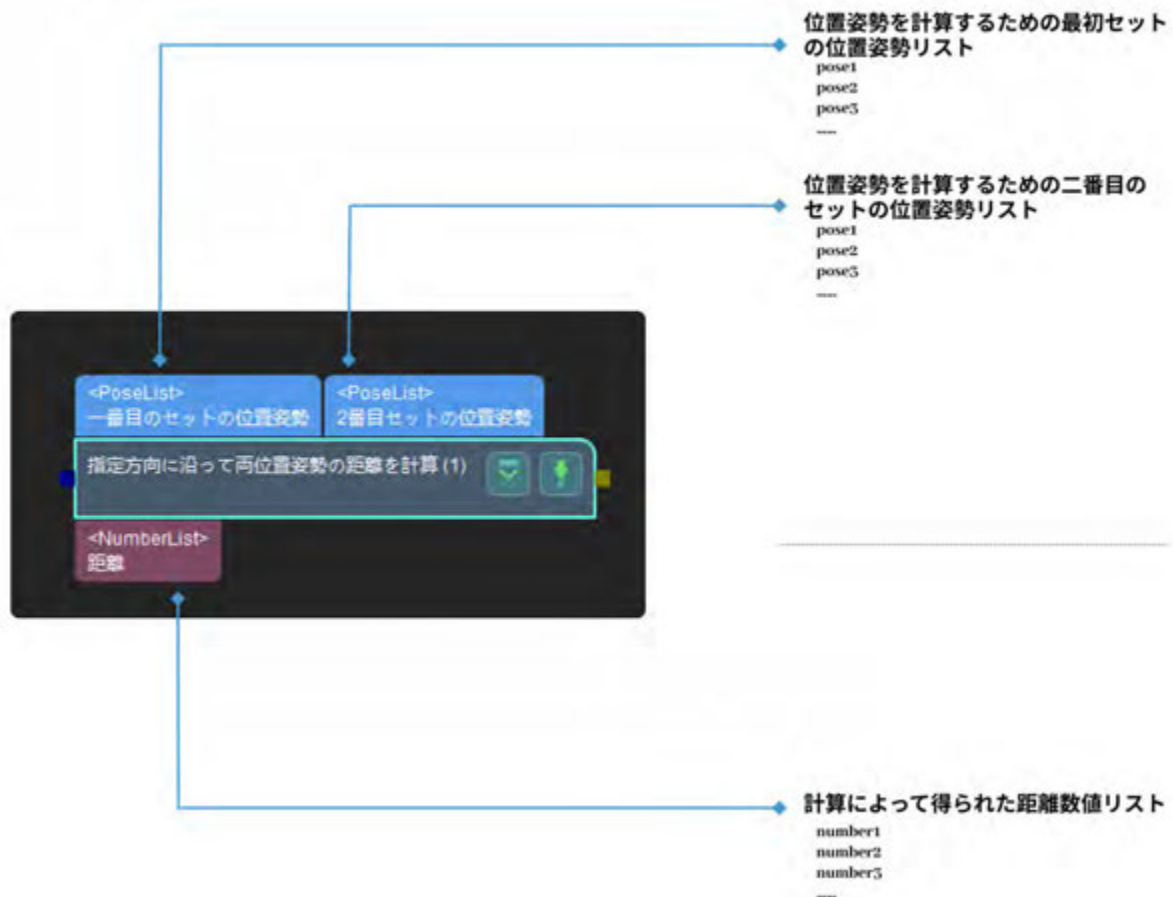
機能

二つの位置姿勢の中心によって形成されたベクトルの長さ、もしくはこのベクトルが指定した軸または平面に沿う投影の長さを計算します。

使用シーン

汎用の位置姿勢距離計算のステップで、特定の使用シーンはありません。

入力と出力



4.3.37. 位置姿勢間の距離を計算



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

2つの位置姿勢の中心を繋いでベクトルを作り、このベクトルの長さ、またはXY平面へ投影する長さを計算します。

使用シーン

汎用の位置姿勢距離の計算ステップで、特定の使用シーンはありません。

入力と出力

● 入力：

1. このポートに入力された位置姿勢は距離ベクトルの開始点として使用されます。
2. このポートに入力された位置姿勢は距離ベクトルの終了点として使用されます。

● 出力：

1. 距離ベクトルの長さ、またはXY平面へ投影する長さ。

4.3.38. 3Dベクトルの内積を計算



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

Vector3Dベクトルの2つのセットの内積を1対1で計算します。

使用シーン

汎用のベクトル計算ステップです。特定の使用シーンはありません。

入力と出力

● 入力：

1. 内積を計算するための最初セットのVector3Dベクトル。
2. 内積を計算するための2番目セットのVector3Dベクトル。

● 出力：

1. 計算によって得られた内積。

4.3.39. エッジ点群の法線ベクトルを計算



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

エッジ点群の法線ベクトルを計算します。

使用シーン

汎用の点群計算ステップです。特定の使用シーンはありません。

入力と出力

- 入力： このポートに入力される対象物のエッジ点群には、法線ベクトルが計算されます。
- 出力： 法線ベクトル付きの対象物のエッジ点群。

4.3.40. 平面度を計算

このステップは、平面点群の平面度誤差を計算します。



- 平面度とは、実際の平面の誤差を測定するために使用される指標です。
- 平面度とは、測定する対象となる平面と理想的に正しい平面と比較して、両者の線の値の距離が誤差値です。

このステップを実行する前に、[\[vision-measure-mode:getting-started::start-use-measure-mode\]](#) をお読みください。

例



ステップをつなぐ

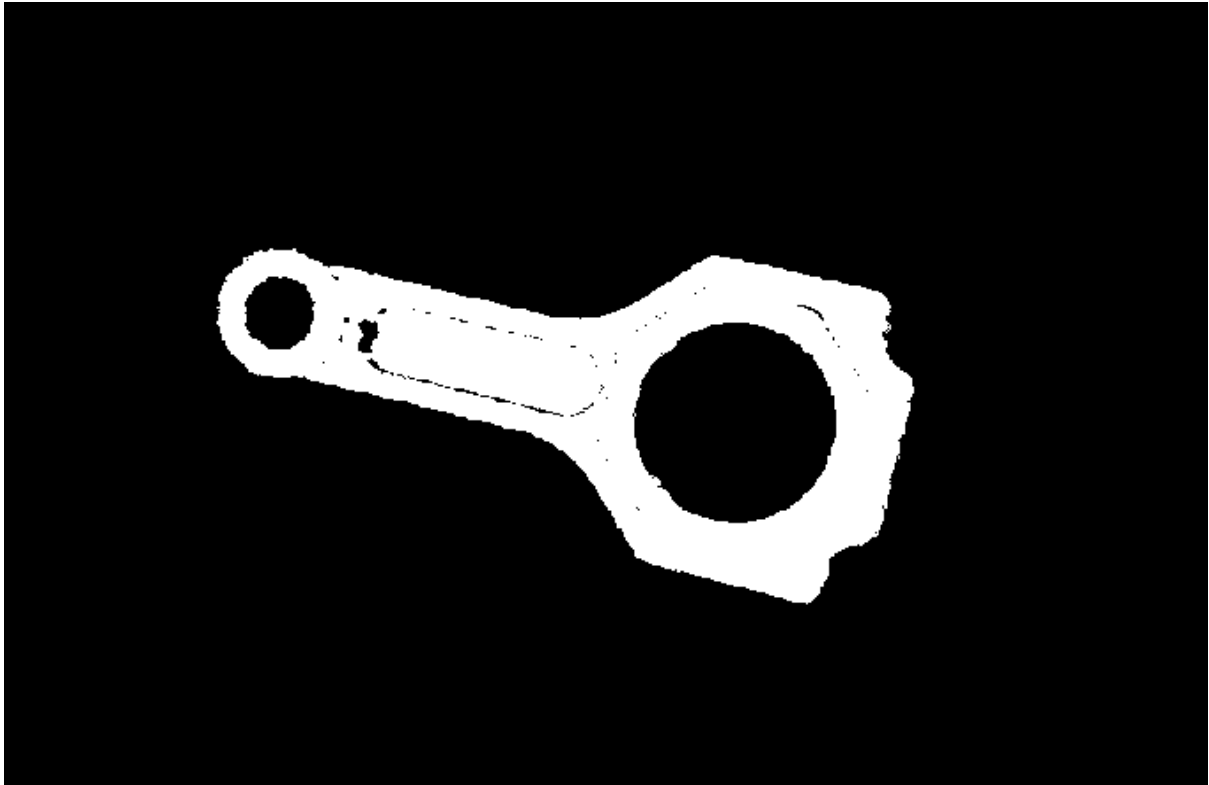
[平面度を計算](#) をクリックして、[ステップ入力ポート選択](#) 画面で「入力1（法線ベクトル付きの点群）」のドロップダウンバーで「点群を読み取る_1_法線ベクトル付きの点群」に設定します。

パラメータ設定

[\[vision-steps:read-point-cloud-v2::read-point-cloud-v2\]](#) 設定：

このステップをクリックし、[ステップのパラメータ](#) 画面で測定する対象物の点群ファイルパスなどを設定します（ユーザーは測定する対象物の点群を用意する必要があります）。設定を完了したら、**[実行]** をクリックしてプロジェクトを実行します。

本例では、使用する点群を下図に示します。



結果の確認

プロジェクトを実行したら、[平面度を計算](#) をクリックして **測定結果出力** 画面で平面度の測定結果を確認できます。

測定結果の出力					📄 ×
名前	値	下限規格値	上限規格値	オフセット	
平面度を計算 ...	0.058	0.000	0.000	0.000	

ユーザーは **測定結果の出力** または **結果ビュー** 画面で上限・下限を設定することで測定の値が合格したかどうかを確認できます。

4.3.41. ヒストグラムを計算



これは古いバージョンのステップであり、メンテナンスが停止されています。このステップに関する詳細が必要な場合は、当社 (docs@mech-mind.net) までお問い合わせください。

機能

2D画像のヒストグラムをピクセル単位で計算します。

使用シーン

特定のシーン向けのカスタマイズのステップです。

入力と出力

● 入力：

1. このポートに入力される2D画像には、ヒストグラムがピクセル単位で計算されます。
2. このポートに入力されるマスクは、ヒストグラムの計算範囲を指定するために使用されます。

● 出力：

1. 計算によって得られたカラー画像。

4.3.42. 二つの位置姿勢の角度を計算



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

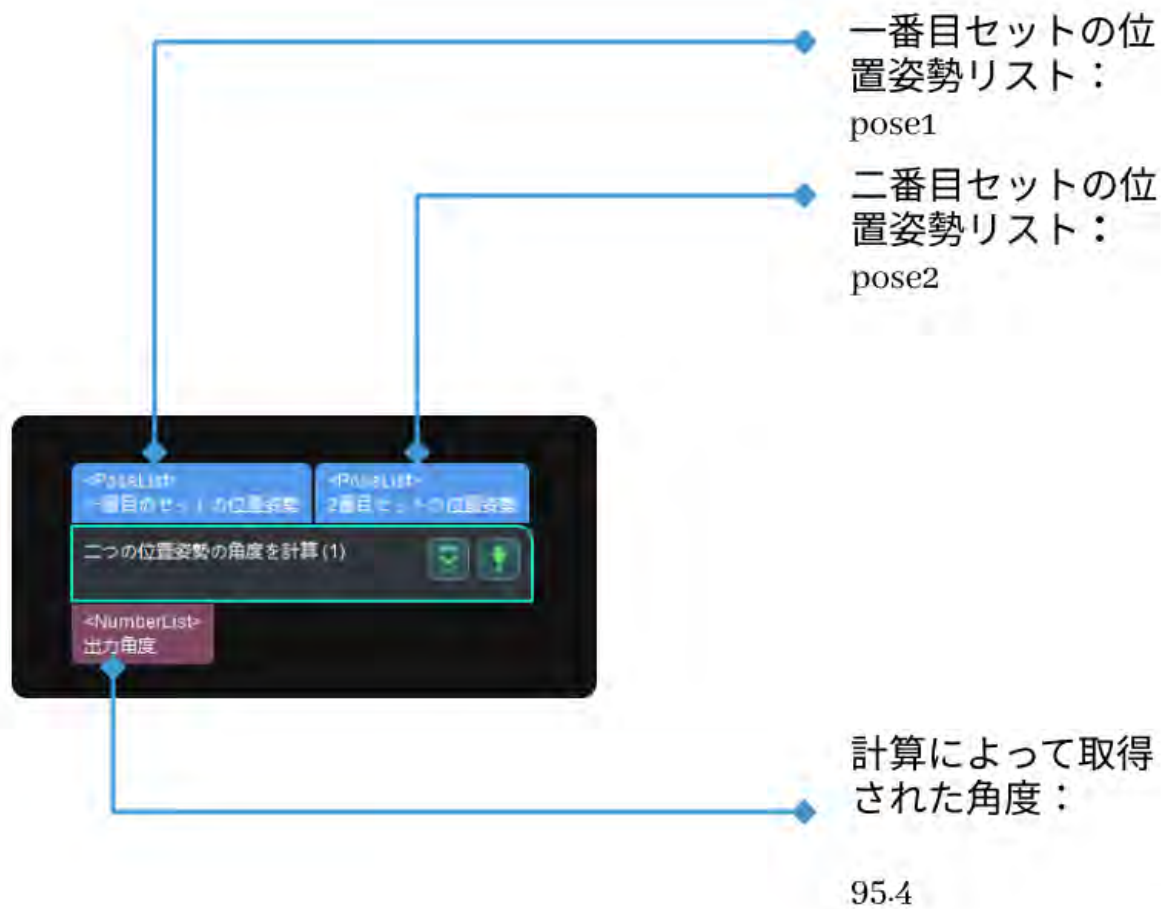
機能

入力された二つの位置姿勢の指定された軸の角度を計算します。

使用シーン

汎用位置姿勢角度計算ステップで、特定使用シーンはありません。

入力と出力



4.3.43. 2つの線分の交点を計算

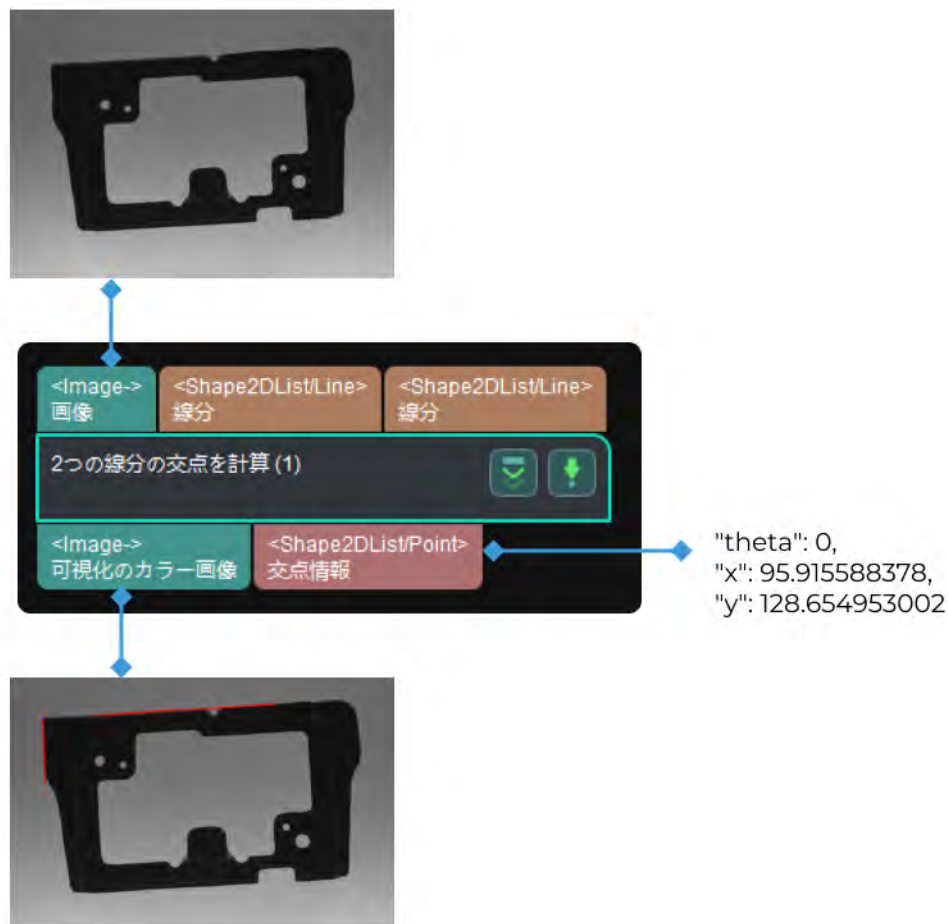
機能

2つの線分の交点のピクセル座標を計算します。

使用シーン

通常、計測シーンで対象物の特徴点を位置決めするために使用されます。

入力と出力



パラメータの説明

交点タイプ

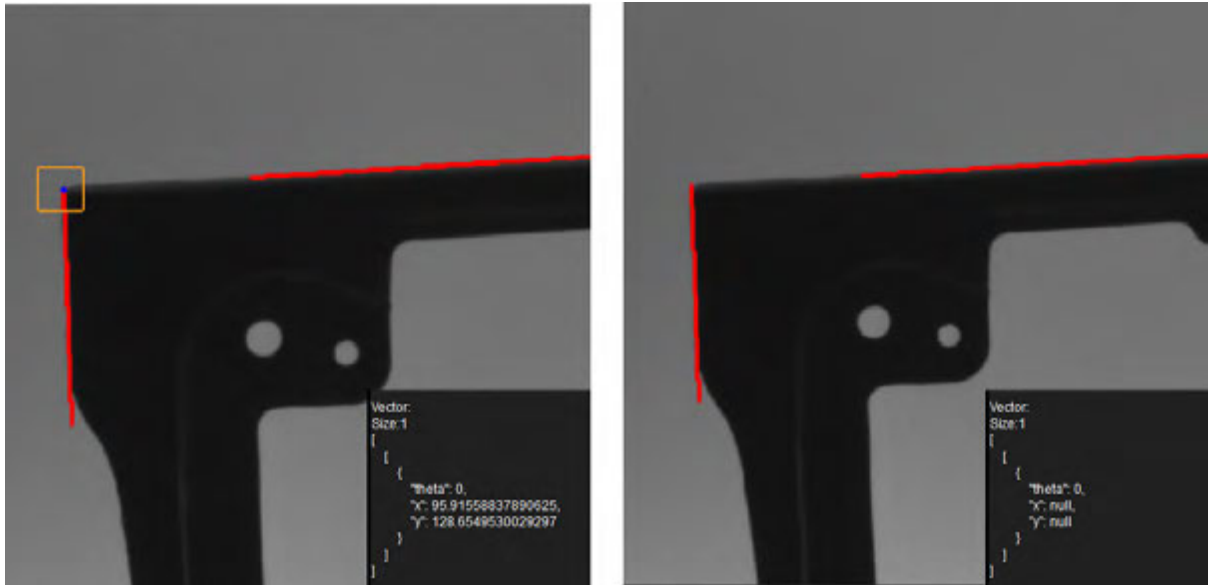
パラメータ説明：計算する交点のタイプを指定するために使用されます。

オプション：端点内、延長線上

初期値：端点内

- 端点内：線分内にある交点を計算します。2つの線分が交差していない場合、出力される交点情報は空です。
- 延長線上：線分の延長線上にある交点を計算します。2つの線分が重なるか平行である場合、出力される交点情報は空です。

下図に示すように、左側は交差点タイプを **延長線** に設定した場合の計算結果で、右側は交差点タイプを **端点内** に設定した場合の計算結果です。交差点は青色で表示されています。



4.3.44. 線分と円との交点を計算

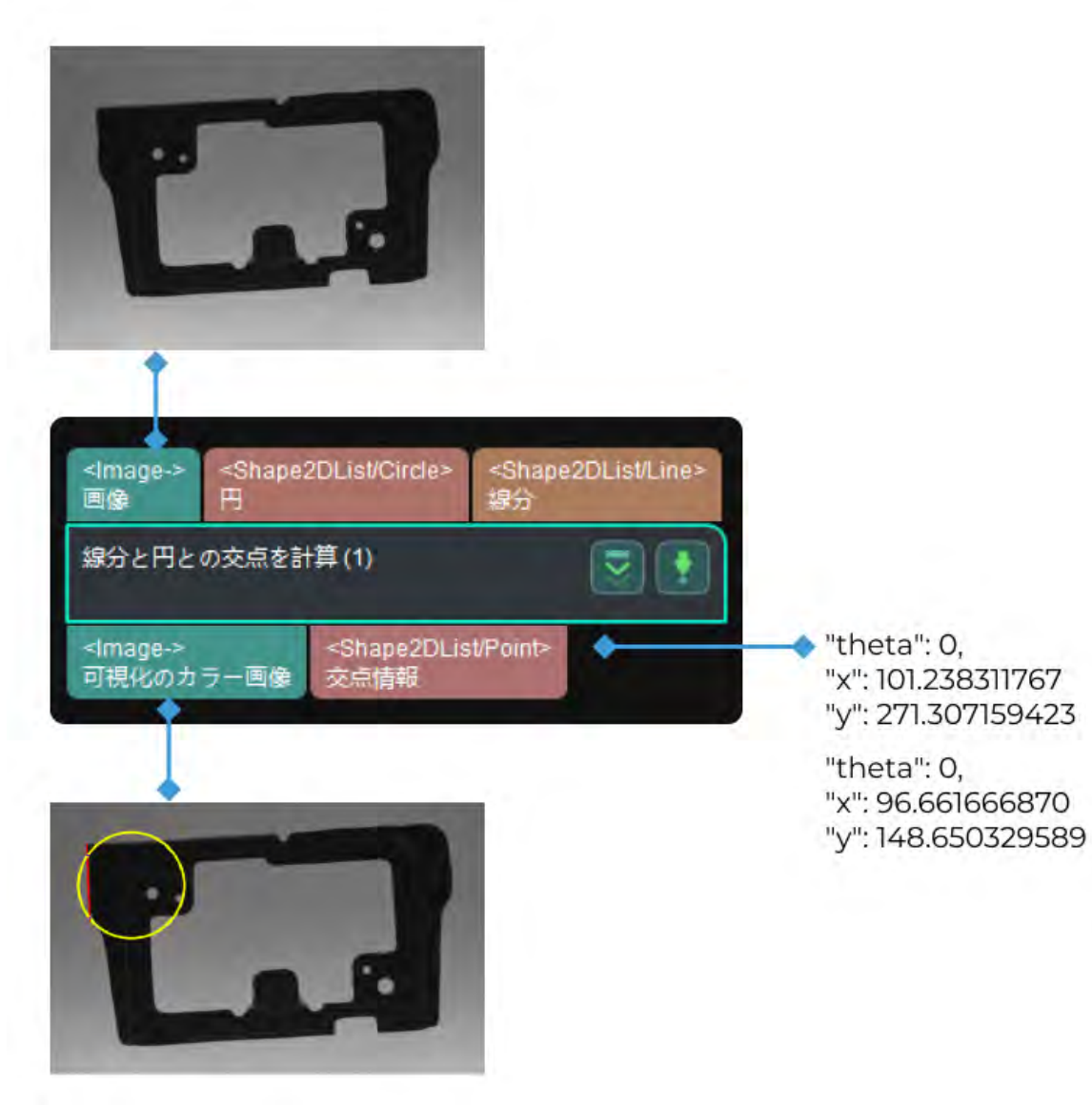
機能

入力線分または線分の双方向延長線と円との交点のピクセル座標を計算するために使用されます。

使用シーン

通常、計測シーンで対象物の特徴点を位置決めするために使用されます。

入力と出力



パラメータの説明

交点タイプ

パラメータ説明：計算する交点のタイプを指定するために使用されます。

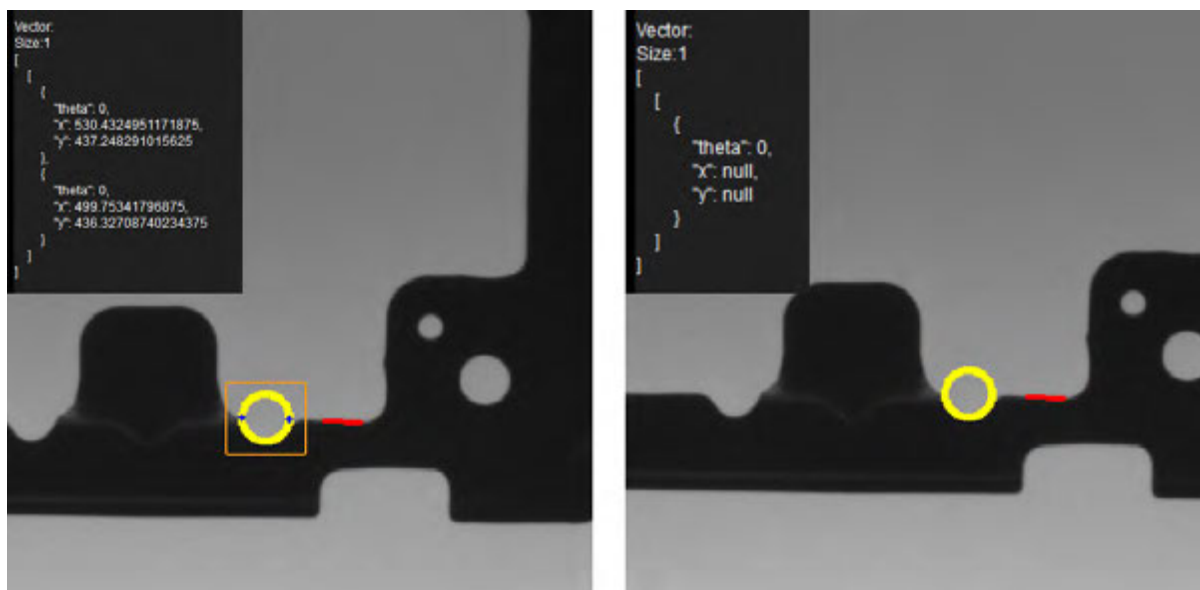
オプション：端点内、延長線上

初期値：端点内

- 端点内：線分内にある円との交点を計算します。線分と円が交差しない場合、出力される交点情報は空です。
- 延長線上：延長線にある交点を計算します。延長線と円が交差しない場合、出力される交点情報は空です。

下図に示すように、左側は交差点タイプを **延長線** に設定した場合の計算結果で、右側は交差点

タイプを **端点内** に設定した場合の計算結果です。交差点は青色で表示されています。



4.3.45. 指定軸方向の点群長さを計算



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

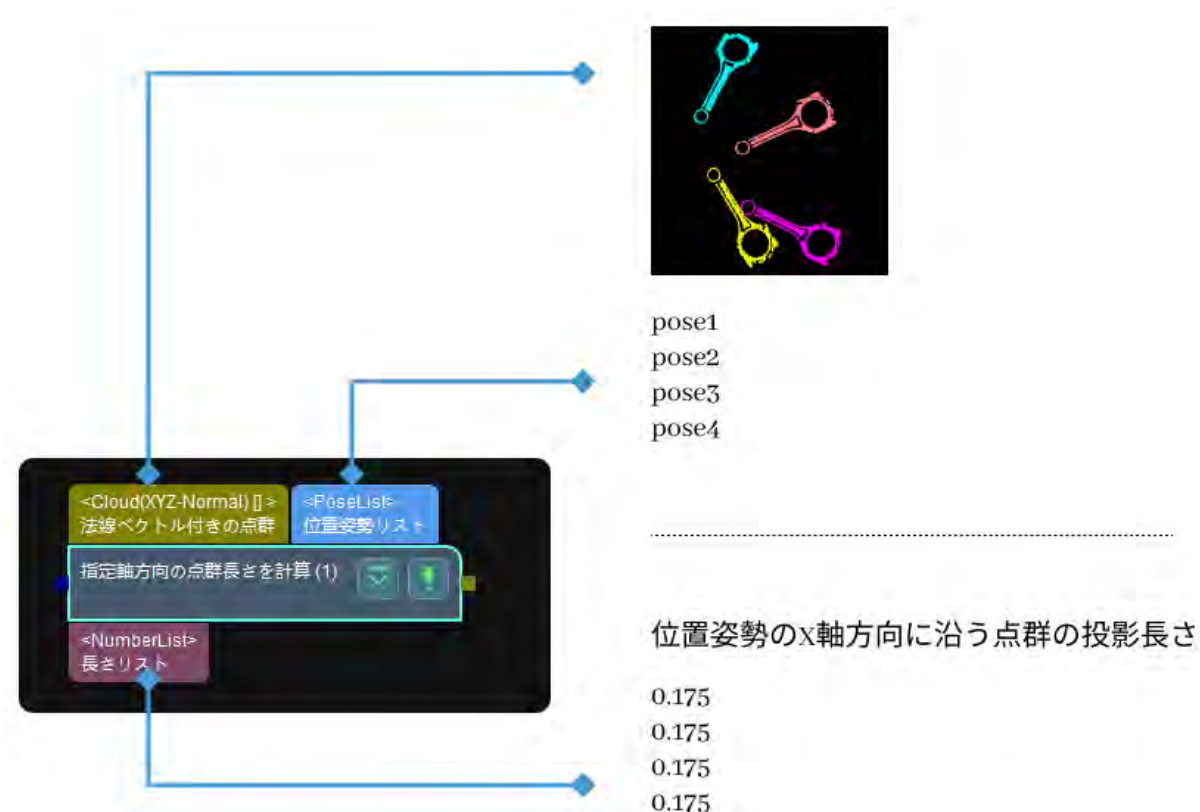
機能

点群が位置姿勢の指定軸へ投影する長さを計算します。

使用シーン

汎用の3D点群計算ステップです。特定の使用シーンはありません。

入力と出力



4.3.46. マスク間の距離を計算



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

2つリストの入力マスク間の距離を計算します。距離の計算方法はパラメータで設定できます。

使用シーン

特定のシーン向けのカスタマイズのステップです。

入力と出力

● 入力：

1. このポートに入力されるマスクは、距離を計算するために使用されます。
2. このポートに入力されるマスクは、距離を計算するために使用されます。
3. このポートに入力されたマスクは、回転の中心を表示します（計算方法が「Rot180SubMassCenter」に設定されている場合に入力されます）。

● 出力：

1. 計算によって得られたマスク間の距離。

4.3.47. 指定直線上のマスクの長さを計算



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

3D位置姿勢の情報に基づいて2D直線を作成します。位置姿勢の投影点に対応するマスク内にある場合、線の2つの交点とマスクの最も遠いエッジの間の距離が計算されます。それ以外の場合、計算された距離は無効と見なされ、0に設定されます。

使用シーン

マスクの幅を計算するためによく使われます。

入力と出力

- 入力：
 1. このポートに入力される位置姿勢は2Dマスクに投影されます。
 2. このポートに入力されるマスクには、指定直線における長さが計算されます。
- 出力：
 1. 計算された指定直線におけるマスクの長さ。

4.3.48. 平均諧調値を計算



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

入力された2D画像のピクセルの平均諧調値を計算します。

使用シーン

汎用の2D画像処理ステップです。特定の使用シーンはありません。

入力と出力

- 入力：
 1. このポートに入力される2D画像のマスクで覆われている領域には、平均諧調値が計算されます。
 2. 平均諧調値を計算する必要がある領域のマスク。

● 出力：

1. 計算された平均諸調値。

4.3.49. 長方形の指定辺の midpoint を計算



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

マスク画像の非ゼロピクセルから計算された外接長方形の指定辺の midpoint を計算します。

使用シーン

これは通常、ハッチ/ガラスの接着剤塗布などのシーンで、接着剤塗布軌道の開始点を計算するために使用されます。

入力と出力

- **入力：** このポートに入力されるマスク画像には、ゼロ以外のピクセル領域の外接長方形が計算されます。
- **出力：** 外接長方形の指定辺の midpoint。

4.3.50. マスクの最小外接長方形を計算



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

入力されたマスク画像に、非ゼロのピクセル領域の最小外接長方形を計算します。

使用シーン

汎用のマスク画像処理ステップです。特定の使用シーンはありません。

入力と出力

- **入力：** このポートに入力されるマスクは、最小外接長方形が計算されます。
- **出力：** 入力されたマスクの非ゼロピクセル領域の最小外接長方形。

4.3.51. 3Dベクトルの単位ベクトルを計算

機能

入力Vector3Dの正規化された単位ベクトルを計算します。

使用シーン

汎用のベクトル計算ステップです。特定の使用シーンはありません。

入力と出力

- **入力：** このポートに入力されるベクトルには、単位ベクトルが計算されます。
- **出力：** 計算された単位ベクトル。

4.3.52. 点群の法線ベクトルを計算してエッジを推定

機能

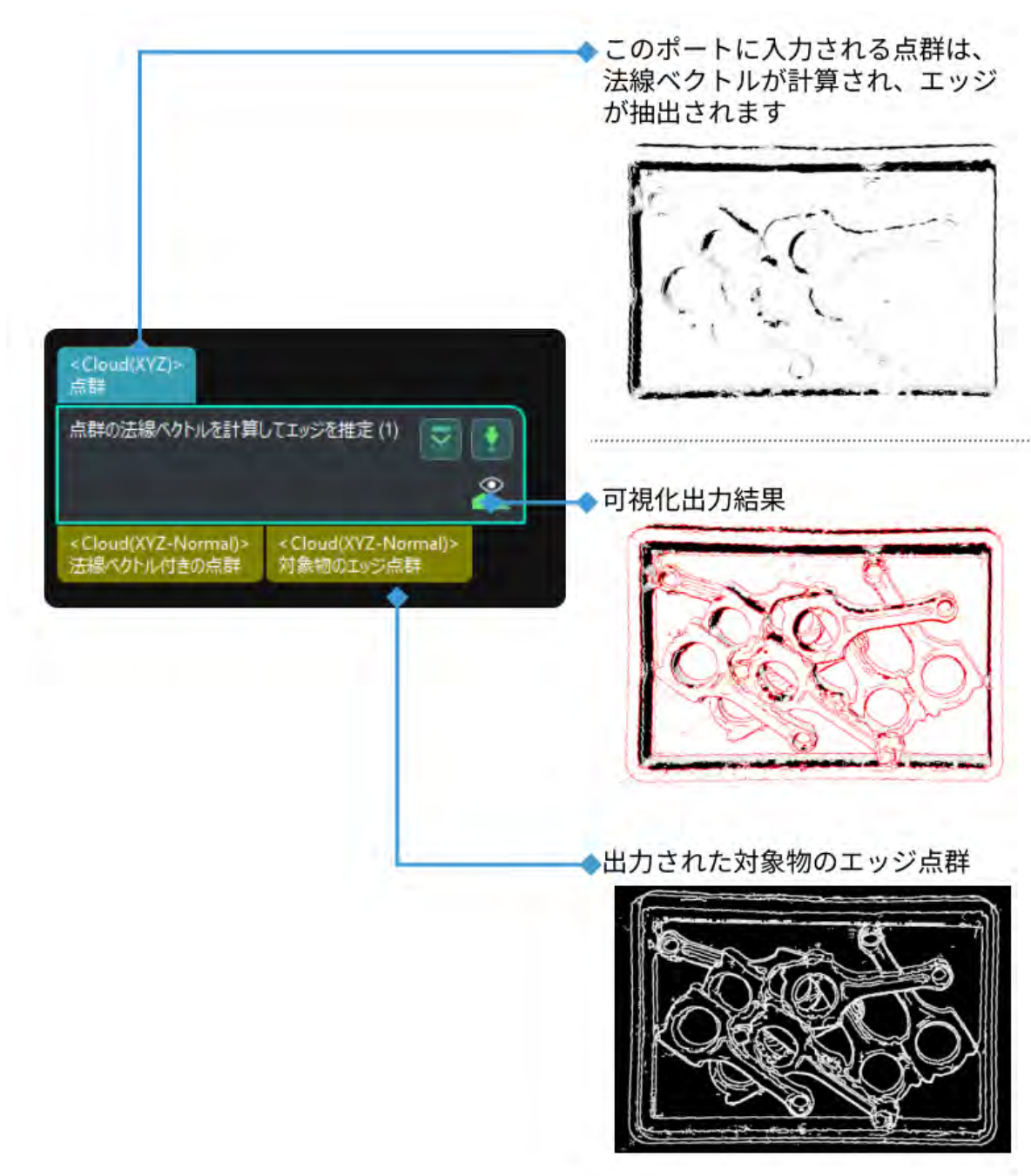
このステップは、法線ベクトルを計算し、点群内の対象物エッジを推定することでエッジ点群を出力できます。

主な流れは、入力点群を深度画像に変換し、点群の法線ベクトルと深度情報を用いて、どの点がエッジ点ポイントであるかを判定します。

使用シーン

通常、計測シーンで対象物のエッジ特徴を見つけるために使用されます。

入力と出力



パラメータの説明

このステップは、実行中に法線ベクトルの計算とエッジ抽出の2つの処理からなり、それぞれの処理流れとパラメータは以下の通りです。

法線ベクトルの計算

法線ベクトルの計算方法

パラメータ説明：このパラメータは、法線ベクトルの計算を選択するために使用されます。

オプション：CloudPreprocessingGPU、CloudXYZToCloudNormal、NormalEstimation

- CloudPreprocessingGPU：この方法は、法線を計算して入力を変換します。GPUが存在し、法線ベクトルの情報を必要とする場合に推奨されます。
- CloudXYZToCloudNormal：この方法は、実際の法線ベクトルを計算することなく、簡単かつ高速に型変換を行うことができます。この方法を選択した場合、パラメータの設定は必要ありません。
- NormalEstimation：通常の方法となります。法線ベクトルの情報を必要としているが、GPUが存在しない場合に選択することができます。この方法を選択した場合、パラメータの設定は必要ありません。

調整アドバイス：実際のニーズに応じて選択してください。

計算設定

バッチ処理

このパラメータは、点群のバッチ処理をするかどうかを設定するために使用されます。

初期値：チェックを入れない。

調整アドバイス：点群内の点数が多く、メモリが小さい場合にこのパラメータにチェックを入れる必要があります。

法線ベクトルを計算するときの半径

パラメータ説明：このパラメータは、法線ベクトルを計算するときの検索半径を設定するために使用されます。検索半径内の点は法線ベクトルの計算に使用されます。単位はミリメートル（mm）です。

初期値：15.000mm

調整アドバイス：実際の状況に応じて設定してください。

凹凸点除去設定（すべてのパラメータを表示した場合にのみ設定可能）

凹凸点を除去

パラメータ説明：このパラメータは、凹凸点を除去するかどうかを設定するために使用されます。チェックを入れた後、設定された **凹凸点を除去するための検索半径**、**凹点除去しきい値**、**凸点除去しきい値** に従って凹凸点を除去します。

初期値：チェックを入れない。

調整アドバイス：実際の状況に応じて設定してください。

凹凸点を除去するための検索半径

パラメータ説明：このパラメータは、凹凸点を除去するための検索半径を設定するために使用されます。このステップでは、この検索半径内の凹凸点を除去します。単位はミリメートル（mm）です。

初期値：20.000mm

調整アドバイス：実際の状況に応じて設定してください。

凹点除去しきい値

パラメータ説明：このパラメータは、凹点を除去する場合、検索半径内の点の凹み値の最大値を設定するために使用されます。現在の点の検索半径内にある点の凹み値がこのしきい値より小さい場合、その点は破棄されます。

初期値：1.0000

調整アドバイス：実際の状況に応じて設定してください。

凸点除去しきい

パラメータ説明：このパラメータは、凸点を除去する場合、検索半径内の点の凸度の最大値を設定するために使用されます。現在の点の検索半径内にある点の凸度がこのしきい値より小さい場合、その点は破棄されます。

初期値：1.0000

調整アドバイス：実際の状況に応じて設定してください。

外れ値の除去設定

外れ値を除去

パラメータ説明：このパラメータは、外れ値を除去するかどうかを設定するために使用されます。チェックを入れ後、設定された **外れ値を除去するための検索半径** と **外れ値のしきい値** に従って外れ値を除去します。

初期値：チェックを入れない。

調整アドバイス：実際の状況に応じて設定してください。

外れ値を除去するための検索半径

パラメータ説明：このパラメータは、外れ値を除去するための検索半径を設定するために使用されます。検索半径内の点は、外れ値の除去に使用されます。単位はミリメートル（mm）です。

初期値：10.000mm

調整アドバイス：実際の状況に応じて設定してください。

外れ値のしきい値

パラメータ説明：このパラメータは、外れ値を除去する場合、検索半径内の点数の最小値を設定するために使用されます。検索半径内の点の数がこのしきい値より少ない場合、その点は外れ値であるとみなされます。

初期値：0

調整アドバイス：実際の状況に応じて設定してください。

エッジ抽出

このステップでは、エッジポイントを判定するための3つの方法が用意されています。

- 法線ベクトルの変動しきい値に基づいてエッジポイントであるかどうかを判定します。
- 深度差のしきい値に基づいてエッジポイントであるかどうかを判定します。
- 点のZ軸と法線ベクトルとの最大角度に基づいてエッジポイント以外の点を判定します。

エッジポイントの判定では、まず、**法線ベクトルとZ軸との最大角度** パラメータを使用してエッジポイント以外の点を判定し、**法線ベクトルの変動しきい値** と **深度差のしきい値** パラメータを使用して残りの点のうちどれがエッジポイントであるかを判定します。ある点がエッジポイントと判定される条件は、**法線ベクトルの変動しきい値** と **深度差のしきい値** 条件の両方を満たしていることです。



パラメータ **深度差のしきい値**、**ギャップ幅の許容範囲**、**エッジポイントの最小数** 和 **法線ベクトルとZ軸との最大角度** は、**すべてのパラメータを表示** が設定された場合にのみ表示されます。

非構造化点群を処理

パラメータ説明：このパラメータは、非構造化点群を処理するかどうかを選択するために使用されます。

初期値：チェックを入れない（ほとんどの場合に適用可能）。

調整アドバイス：入力点群が非構造化点群の場合、このオプションにチェックを入れてください。また、チェックを入れた場合は **法線ベクトルの変動しきい値** などのパラメータに加え、**深度画像の解像度** の設定も必要です。

法線ベクトルの変動しきい値

パラメータ説明：このパラメータは、エッジポイントを判定するための法線ベクトルの変動しきい値を設定するために使用されます。単位は度（°）です。隣接する点の法線ベクトルの変動がこのしきい値より大きい場合、その点はエッジポイントと見なされます。法線ベクトルの変動とは、隣接点の法線ベクトル変動を総合的に考慮した値をいいます。

初期値：10°

深度画像の解像度

パラメータ説明：入力点群が非構造化点群の場合、点群から深度画像への変換には、深度画像の解像度を指定する必要があります。解像度とは、単位長さあたりのピクセル数のことです。単位長さは1mです。非構造化点群が入力された場合、このパラメータの設定が必要です。

初期値：1000（ほとんどの場合に適用可能）。

調整アドバイス：入力点群の点密度が大きい場合は、このパラメータを上げ、点密度が小さい場合は、このパラメータを下げてください。

深度差のしきい値

パラメータ説明：このパラメータは、エッジポイントを判定するの深度差の上限を設定するために使用されます。単位はミリメートル（mm）です。隣接する8点との深度差の最大値がこのしきい値より大きい場合、エッジポイントと見なされます。

初期値：5mm

ギャップ幅の許容範囲

パラメータ説明：点群にギャップがある場合、このパラメータを使用してギャップ両側にエッジがあるかどうかを判定します。単位はピクセル（px）です。ギャップの水平方向の幅がこのパラメータの値より小さく、ギャップ両端の高さの差が **深度差のしきい値** より小さい場合、ギャップ両側はエッジとみなされません。

初期値：2px

エッジポイントの最小数

パラメータ説明：このパラメータは、このステップで出力されるエッジ点群の最小点数を設定するために使用されます。このパラメータより少ない点数のエッジ点群は無視されます。

初期値：10

法線ベクトルとZ軸との最大角度

パラメータ説明：このパラメータは、Z軸と点の法線ベクトルとの角度の最大値を設定するために使用されます。単位は度（°）です。Z軸と点の法線ベクトルとの角度がこの値より大きい場合、その点はエッジポイントと見なされません。

初期値：70°（ほとんどの場合に適用可能）。

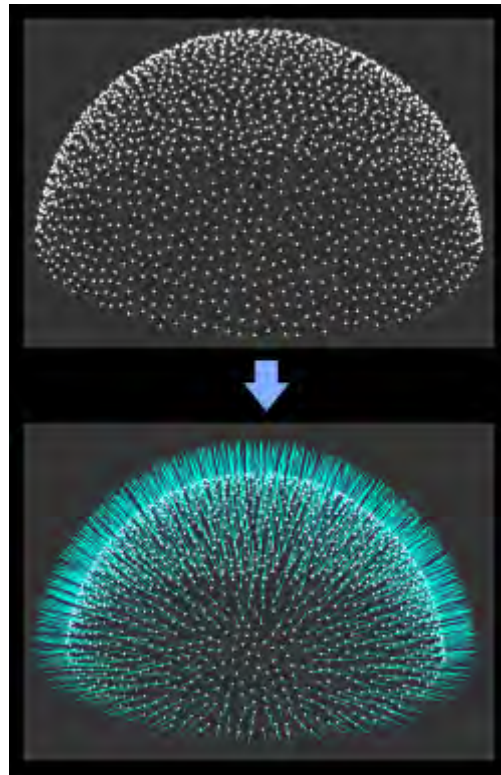
4.3.53. 点群の法線ベクトルを計算してフィルタリング



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

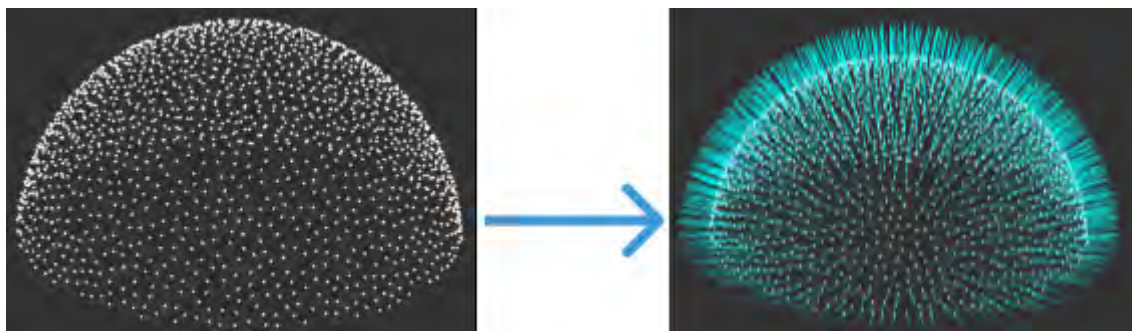
機能

点群の法線ベクトルを計算し、外れ値を除去します。

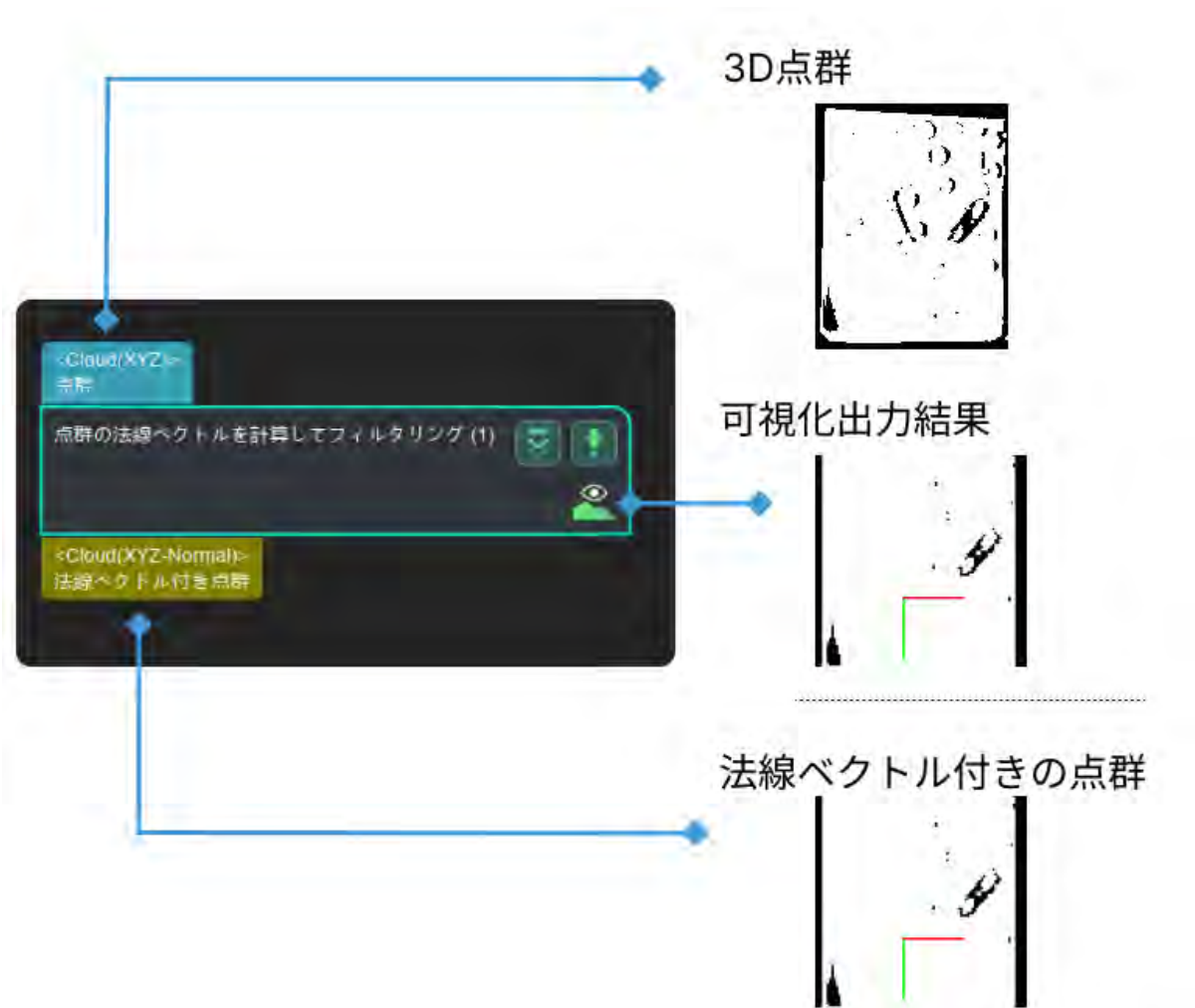


使用シーン

元の点群に法線ベクトルを追加し、外れ値を除去します。通常、[\[vision-steps:capture-images-from-camera::capture-images-from-camera\]](#)、深度画像を点群に変換 ステップの後で使用されます。



入力と出力



4.3.54. 平行度を計算

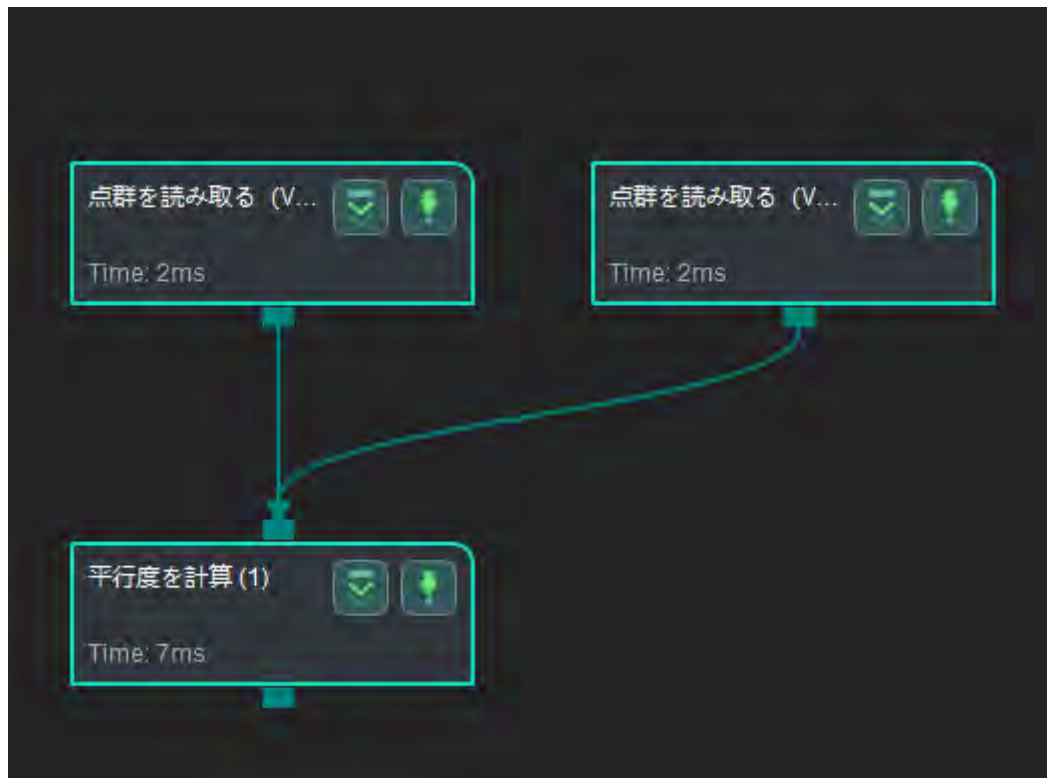
このステップは、基準平面の点群を基準とする対象平面の点群の平行度を計算します。



- 平行度とは、2つの平面または2つの直線が平行である程度を指し、一方の平面（辺）がもう一方の平面（辺）と平行であるという誤差の最大許容値を指します。平行度は、線間、平面間、または線と平面間の平行度の状態を評価します。
- 平行度誤差は、データに平行な理想的な要素に対する測定された実際の要素の変動量です。

このステップを実行する前に、[\[vision-measure-mode:getting-started:::start-use-measure-mode\]](#) をお読みください。

例



ステップをつなぐ

平行度を計算 をクリックして、**ステップ入力ポート選択** 画面で下記のように設定します。

- 「入力1 (対象平面点群)」のドロップダウンバーで「点群を読み取る_1_法線ベクトル付きの点群」に設定します。
- 「入力2 (対象平面点群)」のドロップダウンバーで「点群を読み取る_2_法線ベクトル付きの点群」に設定します。

パラメータ設定

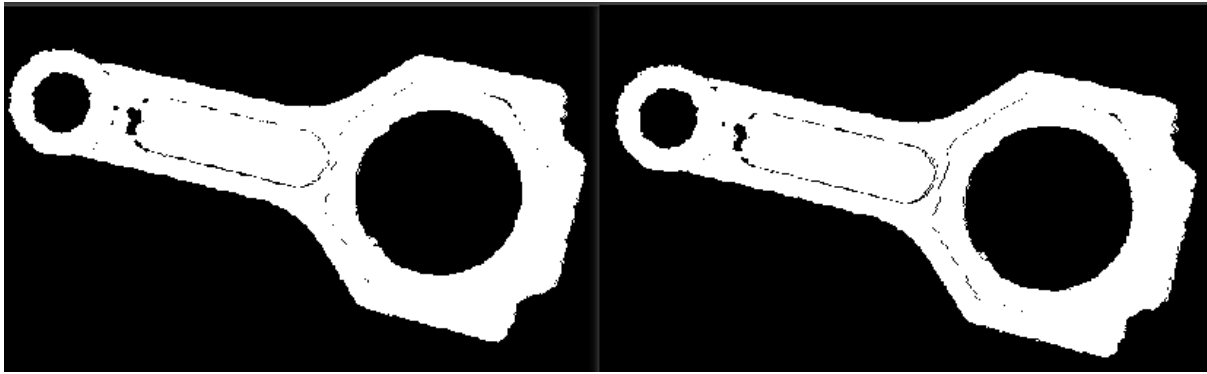
- **[vision-steps:read-point-cloud-v2:::read-point-cloud-v2]** (1) 設定：

このステップをクリックし、**ステップのパラメータ** 画面で測定する対象物の点群ファイルパスなどを設定します（ユーザーは測定する対象物の点群、つまり対象平面の点群を用意する必要があります）。

- **[vision-steps:read-point-cloud-v2:::read-point-cloud-v2]** (2) 設定：

このステップをクリックし、**ステップのパラメータ** 画面で測定する対象物の点群ファイルパスなどを設定します（ユーザーは対象平面の点群を用意する必要があります）。

本例では、使用する点群は下図に示します（左は対象点群で、右は基準点群です）。



設定を完了したら、[実行]をクリックしてプロジェクトを実行します。

結果の確認

プロジェクトを実行したら、**平行度を計算**をクリックして**測定結果出力**画面で平面度の測定結果を確認します。

測定結果の出力					🗑️ ✕
名前	値	下限規格値	上限規格値	オフセット	
平行度を計算 ...	0.058	0.000	0.000	0.000	

ユーザーは**測定結果の出力**または**結果ビュー**画面で上限・下限を設定することで測定の値が合格したかどうかを確認できます。

4.3.55. 指定高度のピクセルサイズを計算



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

あるピクセルの対応する3D点の高さを入力し、ピクセルの実際のサイズを計算します。

使用シーン

特定のシーン向けのカスタマイズのステップで、デパレタイジングの応用専用です。

入力と出力

- **入力**：このポートに入力される3D点の高さは、ピクセルサイズを計算するために使用されます。
- **出力**：計算によって得られたピクセルが対応する実際のサイズ。

4.3.56. 平面の幅を計算



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

スキャンラインの方向に沿って深度値をクラスター化し、同じセットに属するスキャンラインの長さを計算し、長さの中央値を平面幅とします。

使用シーン

特定のシーン向けのカスタマイズのステップです。

入力と出力

● 入力：

1. このポートに入力される深度画像には、幅を計算する必要のある平面が含まれています。
2. 正投影のスケーリング係数。

● 出力：

1. 計算された平面の幅。

4.3.57. 点群の曲率を計算



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

入力点群リストにすべての点群の点の曲率を計算します。

使用シーン

汎用の点群計算ステップです。特定の使用シーンはありません。

入力と出力

● 入力：

1. このポートに入力される点群には、点の曲率が計算されます。

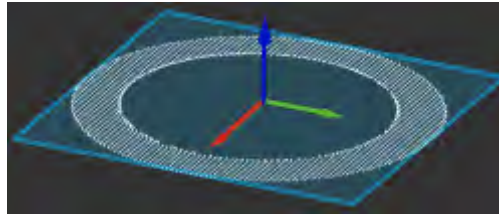
● 出力：

1. 計算された曲率の値。
2. 点の曲率の値が再割り当てされた点群。

4.3.58. 平面点群の位置姿勢とサイズを計算

機能

平面点群の位置姿勢と点群境界ボックスのサイズを計算します。



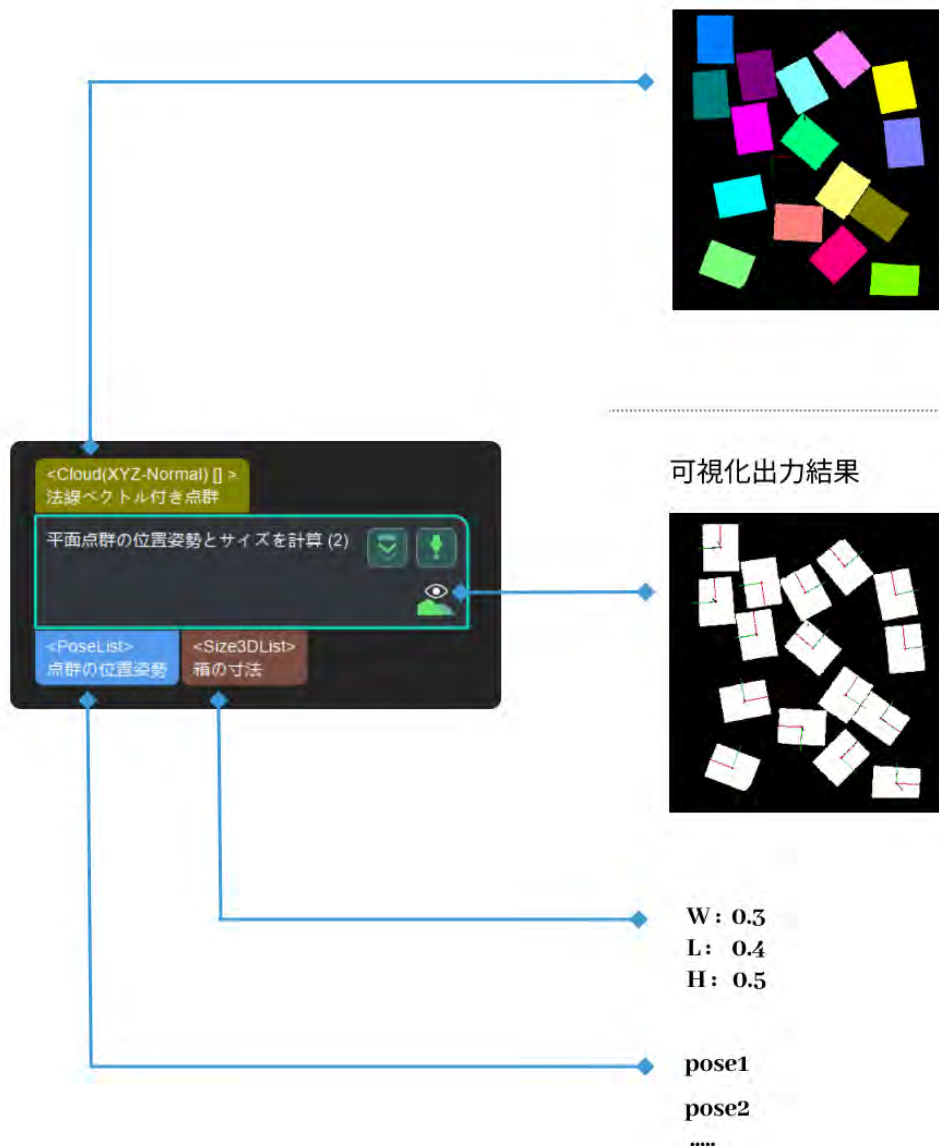
使用シーン

通常、段ボール箱や麻袋などの平面点群の把持位置姿勢とサイズを計算するために使用されます。



このステップを使用する前に、結果に対するノイズの影響を減らすために、点群をフィルタリングしてクラスター化する必要があります。

入力と出力



よく使うパラメータの調整説明

位置姿勢計算の設定



位置姿勢中心の計算方法 と 位置姿勢の姿勢の計算方法 は対応させる必要があります。

- BoundingRectCenterはBoundingRectOrientationに対応します。
- CloudCenterはCloudOrientationに対応します。

位置姿勢中心の計算方法

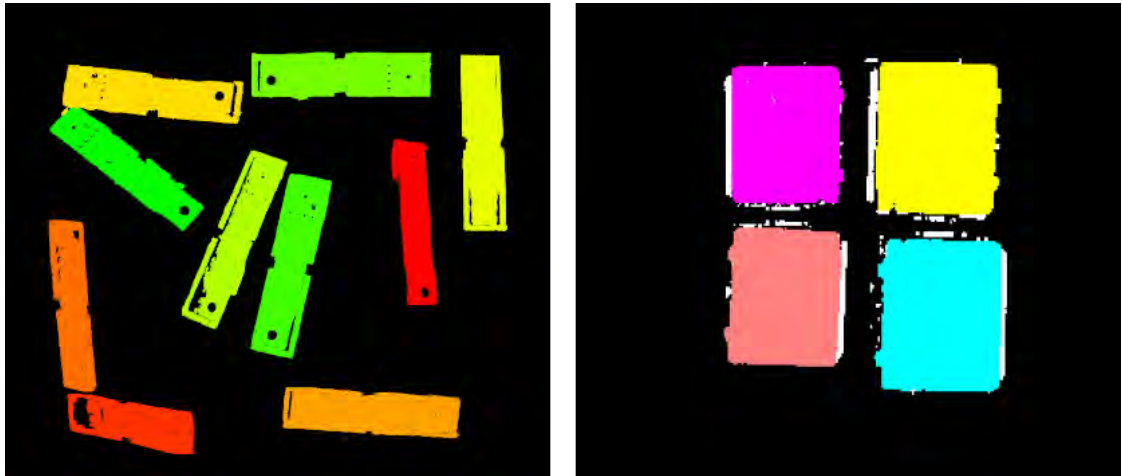
パラメータ説明：このパラメータは、位置姿勢の位置中心を指定するために使用されます。

オプション：

- BoundingRectCenter（点群境界ボックスの中心）：初期値となります。下図の右側に示

すように、対象物の点群が均等に分布している場合に使用します。

- CloudCenter（点群中心）：下図の左側に示すように、対象物の点群が均等に分布していない場合に使用します。



調整アドバイス：実際の状況に応じて設定してください。

位置姿勢方向の計算方法

パラメータ説明：このパラメータは、位置姿勢の向きを指定するために使用されます。

オプション：

- BoundingRectOrientation（点群境界ボックスの姿勢）：上図の右側に示すように、対象物の点群が均等に分布している場合に使用します。
- CloudOrientation（点群の中心姿勢）：上図の左側に示すように、対象物の点群が均等に分布していない場合に使用します。

調整アドバイス：実際の状況に応じて設定してください。

エッジ揃えの外接長方形を使用

パラメータ説明：このパラメータは、外接長方形の辺を対象物の輪郭とできるだけ平行にするかどうかを決定します。チェックを外すと、最小の外接長方形が使用されます。

初期値：チェックを入れない。

調整アドバイス：実際の状況に応じて設定してください。

エッジを揃えた検索半径

パラメータ説明：このパラメータは、外接長方形が対象物の輪郭点の位置を合わせるかどうかを判断するために使用されます。単位はピクセルです。輪郭点と外接長方形との距離がこの値より小さい場合、輪郭点が外接長方形の位置を合わせるとみなされます。

初期値：3px



エッジ揃えの外接長方形にチェックを入れると、このパラメータを設定可能です。

箱の位置姿勢計算の設定



以下のパラメータは箱のみに適用され、麻袋の計算には使用できません。

箱の高さを計算

パラメータ説明：このパラメータは、Z方向における箱の長さを計算するために使用されます。

初期値：チェックを入れない。

調整アドバイス：箱のZ方向の長さを計算するときのみチェックを入れます。

箱の底部の固定高さ

パラメータ説明：このパラメータは、箱のZ方向の高さの計算を容易にするために、箱を載せる台の高さを設定するために使用されます。一般的には、ロボット座標系で箱が搬送されるコンベアや積み上げ台の高さを値とします。

初期値：0.000 mm

調整アドバイス：実際の状況に応じて設定してください。



箱の高さを計算 にチェックを入れると、このパラメータを設定可能です。

4.3.59. 長方形の寸法と位置姿勢を計算

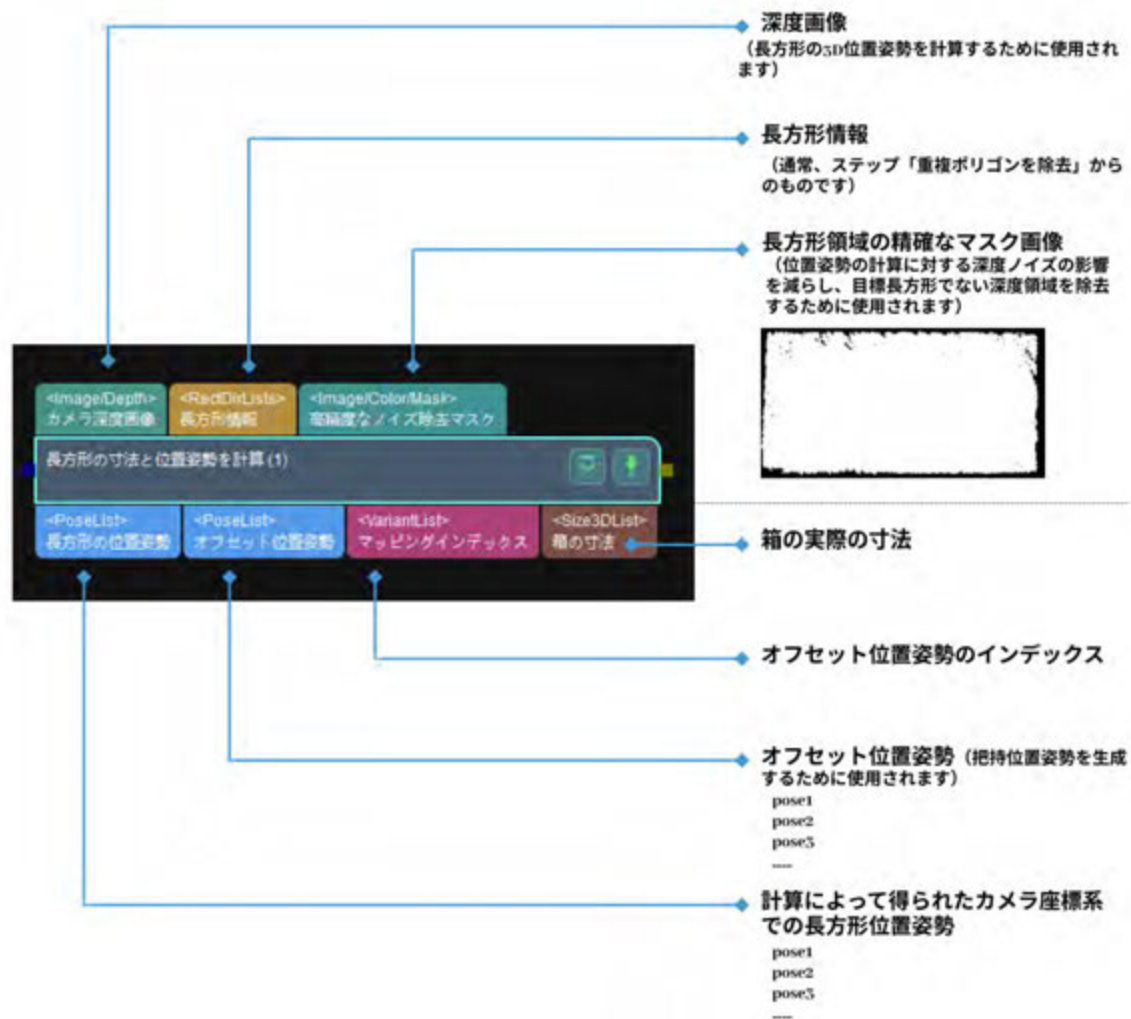
機能

深度画像と長方形情報を通じて、3D空間における長方形の対応する位置姿勢を計算します。

使用シーン

ステップ [重複ポリゴンを除去](#) と併用することになっています。 [2Dマッチング](#) によって得られたカメラ座標系における長方形の物体の位置姿勢を計算することができます。

入力と出力



パラメータの説明

入力タイプ

パラメータ説明：このパラメータは、入力タイプを選択するために使用されます。

初期値：Depth

オプション：Depth、VertexAndSize

調整アドバイス：実際の状況に応じて設定してください。

位置姿勢計算の設定

位置姿勢のZ値の検索半径を計算

調整説明：このパラメータは、画像内の指定位置の深度の検索半径を調整するために使用されます。単位はピクセルです。「入力タイプ」が **Depth** の場合にのみ有効です。

初期値：20

推奨値：20

有効範囲： $[0, +\infty)$

吸盤の設定

X方向における吸盤のサイズ

調整説明：このパラメータは、吸盤被覆領域のX方向のサイズを調整するために使用されます。単位はミリメートルです。吸盤が対象箱よりも大きい場合は、近傍位置の箱を吸着しないように小さく調整されます。

初期値：0

推奨値：吸盤が対象箱よりも小さい場合は0に設定することをお勧めします。吸盤が対象箱よりも大きい場合は、実際の吸盤のサイズに合わせて設定してください。

有効範囲： $[0, +\infty)$

Y方向における吸盤のサイズ

調整説明：このパラメータは、吸盤被覆領域のY方向のサイズを調整するために使用されます。単位はミリメートルです。吸盤が対象箱よりも大きい場合は、近傍位置の箱を吸着しないように小さく調整されます。

初期値：0

推奨値：吸盤が対象箱よりも小さい場合は0に設定することをお勧めします。吸盤が対象箱よりも大きい場合は、実際の吸盤のサイズに合わせて設定してください。

有効範囲： $[0, +\infty)$

吸盤が他の箱との重複比のしきい値

調整説明：作業中に吸盤が近傍位置の箱やボックスを吸着するのを防ぐために、このパラメータは現在の吸着が実行されているかどうかを判断するための基準として使用されます。非対象箱との重複比がこの値よりも小さい場合は吸着を行い、この値よりも大きい場合は吸着を行いません。

初期値：0.3

推奨値：実際のニーズに応じて設定してください。

出力戦略

位置姿勢の数

パラメータ説明：このパラメータは、出力される箱の位置姿勢の数を設定するために使用されます。

初期値：One

オプション：One、Multiple

調整アドバイス：実際の状況に応じて設定してください。

4.3.60. 輪郭度の平面度を計算（ダウンサンプリング）



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

取得された対象物の表面点群とモデル点群をダウンサンプリングした後、両者の差を計算して、対象物の点群とモデル点群の密度の不一致による計算への影響を排除します。

使用シーン

特定のシーン向けのカスタマイズのステップです。

入力と出力

- 入力：
 1. 輪郭度を計算する対象物の点群。
 2. 輪郭度を計算するためのモデル点群。
- 出力：
 1. 対象物とモデル点群の間の最大の高さの差。
 2. 対象物と中心点でのモデル点群との高さの差。
 3. 最大の高さの差の位置をラベリングするための線分。
 4. 中心点での高さの差をラベリングするための線分。
 5. 関心領域内のモデル点群。
 6. 関心領域内の対象物の点群。
 7. 対象物の点群とモデル点群の高さの差。

4.3.61. 輪郭度の平面度を計算



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

取得された表面点群とモデルの間の高さの差（輪郭度）を計算します。通常、同時に2つを使用します。1つは測定対象物の点群を入力し、もう1つは標準モデルの点群を入力します。それぞれフィッティング平面との差（平面度）を計算し、平面度の差を出力して輪郭度を取得しま

す。通常、平面度または輪郭度は、点群の高さの差の最大値です。

使用シーン

特定のシーン向けのカスタマイズのステップです。

入力と出力

● 入力：

1. このポートに入力される点群は、輪郭度の平面度を計算するために使用されます。測定対象物または標準モデルの点群です。
2. 計算された平面の幅。

● 出力：

1. 入力点群の関心領域内の指定された点とフィッティング平面の間の高さの差。
2. 関心領域を選択した後の対象物またはモデル点群。
3. 指定された2D座標に移動した後の対象物またはモデル点群（後で異なる点群をフィッティングするために使用され、計算を容易にします）。
4. 指定された2D座標に移動した後の対象物またはモデル点群、およびフィッティング平面の点群。
5. 断面点群。

4.3.62. 基準方向に沿った投影距離を計算



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

入力位置姿勢の中心を座標系の原点につなぎ、ベクトルを生成して、このベクトルが基準方向における投影の長さを計算します。

使用シーン

汎用の位置姿勢の投影距離計算ステップです。特定の使用シーンはありません。

入力と出力

- **入力：** このポートに入力される位置姿勢には、指定方向へ入力位置姿勢の投影する長さが計算されます。
- **出力：** 指定方向へ入力位置姿勢の投影する長さ。

4.3.63. 長方形の2D位置姿勢を計算



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

長方形の頂点情報を通じて、長方形の中心点の位置姿勢を計算します。

使用シーン

特定のシーン向けのカスタマイズのステップで、デパレタイジングの応用専用です。

入力と出力

- **入力：**
 1. 長方形の頂点。
- **出力：**
 1. 計算によって得られた長方形の中心位置姿勢。
 2. 計算によって得られた長方形の寸法。

4.3.64. Pythonを使用して結果を計算

機能

Pythonを介してカスタマイズのスクリプトを実行し、計算結果をMech-Visionに出力します。

このステップの特徴は以下の通りです。

- マルチスレッドでの使用をサポートします。
- Pythonスクリプトをリアルタイムに読み込むことができます。
- C++側とPython側でデータを転送する際に、複数のデータ型変換をサポートします。
- Python側からのログをMech-Visionにリダイレクトすることに対応します。

使用シーン

カスタマイズの計算が必要な場合は、このステップを使用してビジョンソリューションを簡素化するPythonスクリプトを実行することができます。

入力と出力

- **入力：** パラメータ **入力ポート** に入力されたデータ型によって決定されます。
- **出力：** パラメータ **出力ポート** に入力されたデータ型によって決定されます。



このステップの入出力ポートのデータ型は、前後のステップの入出力ポートのデータ型に基づいて決定することができます。

インストールと使用

インストール方法

Python 3.6.8はMech-Visionに組み込まれており、このステップではソフトウェアの組み込み環境を使用します。使用中にPythonライブラリが不足した場合は、Mech-Visionの組み込みPython環境に不足したPythonライブラリをインストールする必要があります。インストール方法は以下の通りです。

1. コマンドプロンプトを開きます。
2. コマンドラインから「cd」コマンドを使用して、Mech-VisionのPythonディレクトリへのパスを切り替えます。
3. `python -m pip install` Pythonライブラリのコマンドを実行し、対応するPythonライブラリをダウンロードしてインストールします。



Pythonの一般的なライブラリであるNumPyとOpenCVは、Mech-Visionに組み込まれています。

使用方法

詳細は以下の通りです。

1. 入出力ポートのデータ型を設定します。**入力ポート**と**出力ポート**のデータ型は、前後のステップの入出力ポートのデータ型に従って、または必要に応じて入力してください。
2. Pythonスクリプトのパスを設定します。読み込むスクリプトのパスを**スクリプトファイルのパス**で選択します。
3. 呼び出される関数の名前を設定します。スクリプトパスを選択すると、このステップではスクリプト内の関数名が自動的に取得されます。次に、**呼び出される関数名**のドロップダウンリストで必要なスクリプト関数名を選択する必要があります。
4. ステップを実行します。



このステップで、最新のPythonスクリプトが変更されたときにリアルタイムで実行させたい場合は、ステップパラメータの**実行フラグ**の下にある**必要なファイルを再読み込み**にチェックを入れてください。



各パラメータの詳細については、[パラメータの説明](#)をご参照ください。

使用上の注意事項

このステップでPythonスクリプトを作成し、実行する場合、以下の点に注意する必要があります。

サードパーティライブラリの使用を推奨

Mech-VisionでPythonスクリプトを実行することと、Python環境でPythonスクリプトを実行することは異なるため、一部のPythonライブラリがインストールされていない、またはインストールしても動作しない場合がありますので、サードパーティ製ライブラリの使用を推奨します。

NumPyライブラリの使用に注意

Pythonがサポートするデータ型のうち、より複雑な型はNumPyライブラリを介して中間フォーマットとして使用されるデータ型です。NumPyの配列型であるパラメータ型があるにもかかわらず、NumPyにインポートされていない場合、エラーが発生します。したがって、スクリプトの先頭に **import numpy** を追加する必要があります。

スクリプト作成際にデータ型の次元に注意

Pythonスクリプトを作成には、ステップポートの基本データ型のデータ次元に注意を払う必要があります。

- デフォルトのデータ次元は0： Image; Cloud(XYZ); Cloud(XYZ-Normal)；
- デフォルトのデータ次元は1： NumberList; BoolList; IndexList; StringList；
- デフォルトのデータ次元は2： PoseList; Pose2DList; Size3DList。



データ型の後に[align="center"]記号を追加して、データ次元の追加を示します。

例えば、NumberListは1次元の値のリスト、NumberList[align="center"]は2次元の値のリストを表します。

パラメータの説明

入力ポート

パラメータ説明：このパラメータは、ステップの入力ポートのデータ型を設定するために使用されます。入力データ型は、対応する順序で呼び出された関数にパラメータとして渡されます。

初期値：なし。

出力ポート

パラメータ説明：このパラメータは、ステップの出力ポートのデータ型を設定するために使用されます。関数が返すデータは、対応する順序でステップに返され、対応するデータ型に従って解析されることになります。

初期値：なし。

現在対応しているデータ型は以下の通りです。

ポートタイプ	例
PoseList	[[10, 20, 30, 0.951, 0.255, 0.168, 0.045]], [10, 20, 30, 0.951, 0.255, 0.168, 0.045]] (各データグループの最初の3つの値は座標値を表し、後ろの4つの値は四元数を表します。)
Pose2DList	[[0, 0, 0]], [2, 0, 120]] (各データグループの最初の2つの値はXとY値を表し、3番目の値は角度を表します。)
NumberList	[1.1, 2, 999.9, -22]
StringList	['string_1', 'string_2', 'string_3']
Image	画像
Cloud(XYZ)	点群
Cloud(XYZ-Normal)	法線ベクトル付きの点群
Cloud(XYZ-RGB)	カラー点群
Size3DList	[[2.5, 5, 0.001]], [6, 5, 0.02]] (各データグループの最初の2つの値は幅と高さを表し、3番目の値は各ピクセルの長さを表します。)
IndexList	[45, 10, 90]
BoolList	[True, False, True]

スクリプトファイルのパス

パラメータ説明：このパラメータは、読み込むスクリプトのファイルパスを選択するために使用されます。

初期値：なし。

呼び出される関数名

パラメータ説明：このパラメータは、呼び出されるスクリプト関数の名前を設定するために使用されます。

初期値：なし。

Pythonのサンプルコード

このステップのポートタイプを理解しやすく、Pythonスクリプトを使用してこのステップを実行するために、以下ではPythonのサンプルコードを紹介します。

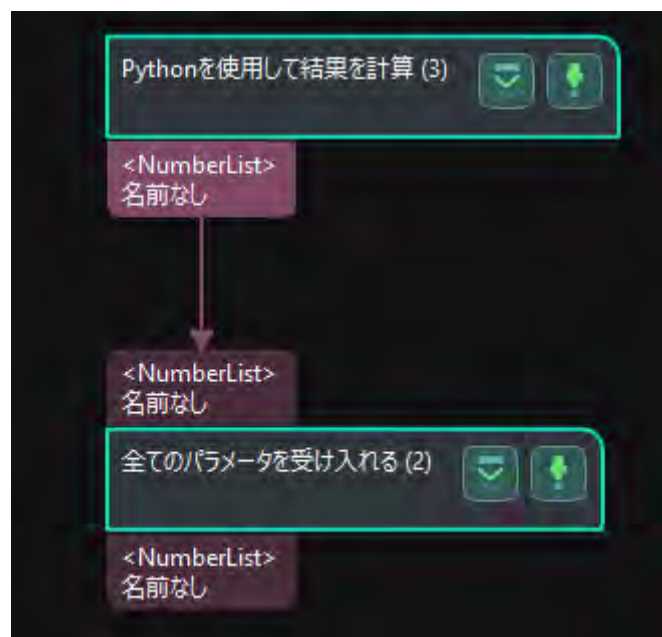
クイックスタート

Pythonスクリプトによる同じ型データを出力

以下のサンプルでは、**同じ型データを出力** する関数を定義しています。 [使用方法](#) ステップの使用方法を参照して、この関数を呼び出すことができます。

```
def get_doublelist():
    return [1.1,22,3.3]
```

このサンプルに対応するMech-Visionプロジェクトを下図に示します。出力ポートのデータフローをダブルクリックして、出力結果を確認できます。

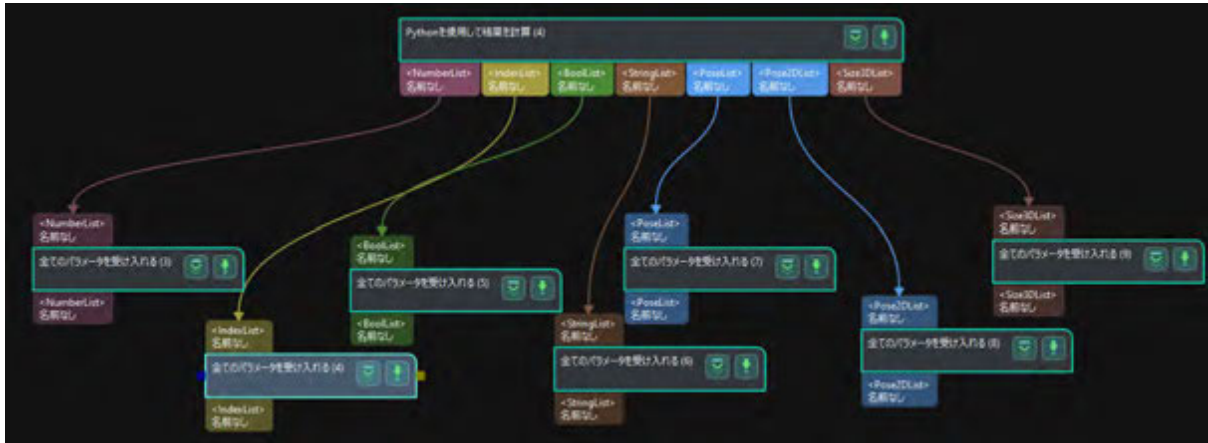


Pythonスクリプトによる複数型のデータを出力

以下のサンプルでは、**複数型のデータを出力** する関数を定義しています。 [使用方法](#) ステップの使用方法を参照して、この関数を呼び出すことができます。

```
def example_of_basic_portTypes():
    numList = [1.1, 2, 999.9, -22]
    indexList = [45, 10, 90]
    boolList = [True, False, True]
    strList = ['string_1', 'string_2', 'string_3']
    poseList = [[10, 20, 30, 0.951, 0.255, 0.168, 0.045], [10, 20, 30, 0.951, 0.255, 0.168, 0.045]]
    pose2dList = [[0, 0, 0], [2, 0, 0.785]]
    size3dList = [[2.5, 5, 0.001], [6, 5, 0.02]]
    return numList, indexList, boolList, strList, poseList, pose2dList, size3dList
```

このサンプルに対応するMech-Visionプロジェクトを下図に示します。各出力ポートのデータフローをダブルクリックして、出力結果を確認できます。



2つの「Pythonを使用して結果を計算」ステップによるデータ処理

サンプル1は数値リストを出力する関数を定義し、サンプル2は数値リストの各リスト内の値に対して合計演算を実行する関数を定義します。

使用方法 ステップの使用方法を参照して、このステップで以下のサンプルコードを呼び出し、ステップをつないで入力の数値リストに対して合計演算を実行することができます。

#例程1

```
def get_numberlist_1():
    return [[1, 2, 3],[4,5,6]]
```

#例程2

```
def cal_number_list(nums_list):
    sums = [align="center"]
    for nums in nums_list:
        sum = 0
        for num in nums:
            sum += num
        sums.append(sum)
    return [sums]
```

このサンプルに対応するMech-Visionプロジェクトを下図に示します。各出力ポートのデータフローをダブルクリックして、出力結果を確認できます。



複数の「Pythonを使用して結果を計算」ステップによるデータ処理

サンプル1は数値リストを出力する関数を定義し、サンプル2はブール値リストを出力する関数を定義します。サンプル3は数値リストに「9999」を追加し、ブール値リストに「False」を追加し、新しい数値リストおよびブール値リストを出力するために使用されます。

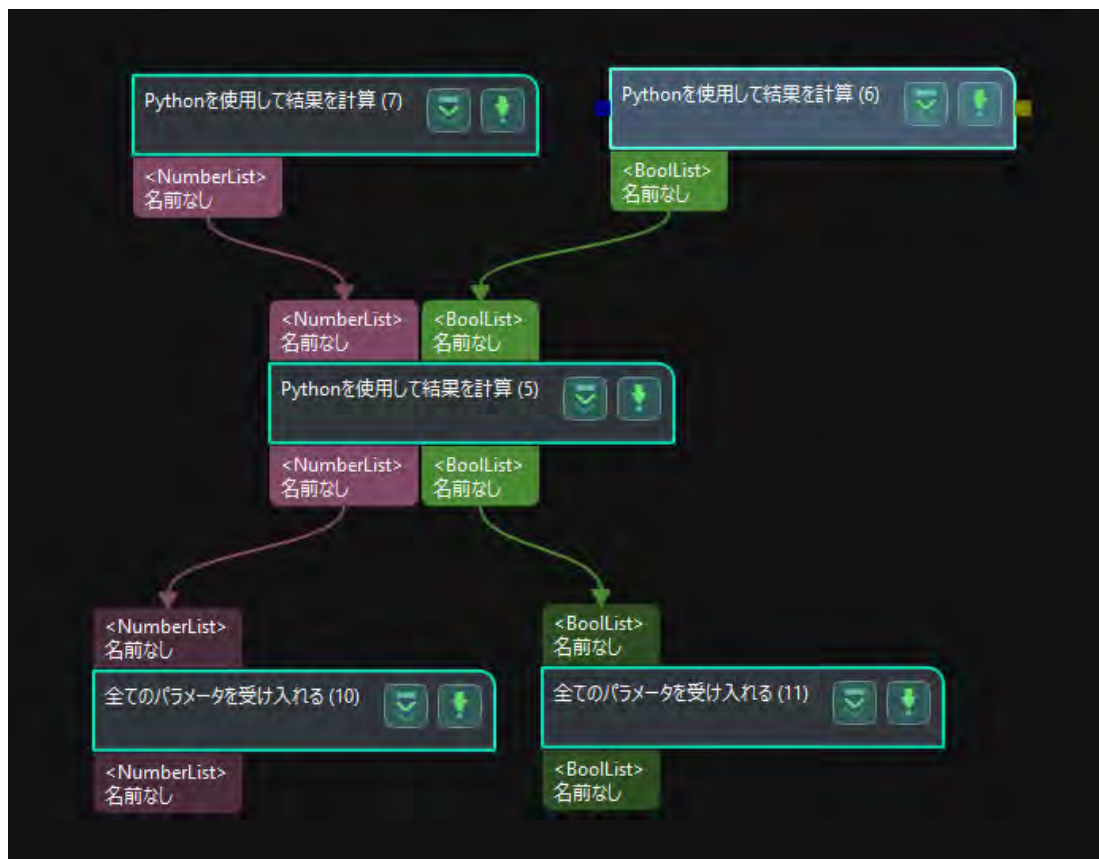
使用方法 ステップの使用方法を参照して、このステップで以下のサンプルを呼び出して、複数の入力データに対して処理を行うことができます。

```
#例程1
def get_doublelist():
    return [1.1,22,3.3]
```

```
#例程2
def get_bool_list():
    return [[True,False],[True,True],[False,False]]
```

```
#例程3
def print_multi_values(numberList, boolList):
    numberList.append(9999)
    boolList.append(False)
    return numberList, boolList
```

このサンプルに対応するMech-Visionプロジェクトを下図に示します。各出力ポートのデータフローをダブルクリックして、出力結果を確認できます。



実際応用に適応可能なサンプルコード

プロジェクト実行中の異常データを保存

実際の応用シーンでは、プロジェクト実行中に様々な問題が発生することがあります。Mech-Visionプロジェクト内のデータ保存機能とカスタマイズされたPythonスクリプトにより、異常データを保存し、問題のトラブルシューティングやビジョンシステムの安定性向上を実現しています。

詳細は以下の通りです。

1. データ保存 機能を有効にします。

Mech-Visionのプロジェクトアシスタントで **データとパラメータを保存** を有効にすると、プロジェクトでエラーポップアップが発生した場合、生データが **プロジェクトフォルダ/data/error_data** パスに保存されます。

2. Pythonスクリプトを作成します。

「3D位置姿勢推定」の出力位置姿勢が空かどうかを判定するPythonサンプルコードを例に説明します。詳細は以下の通りです。

```

def abnormal_output_detection(outputs):
    if len(outputs) == 0:

```



```
raise Exception("NO RESULT!!!")
else:
    pass
```

3. **使用方法** ステップの使用方法を参照して、「Pythonを使用して結果を計算」ステップで上記のサンプルを呼び出して、このステップを「3D位置姿勢推定」の後につなぎます。

プロジェクト実行後、「3D位置姿勢推定」の出力位置姿勢が空の場合、ポップアップウィンドウが表示され、異常データが **プロジェクトフォルダ/data/error_data** パスに保存されます。

入力点群のZ値に基づいて疑似カラー点群を生成

プロジェクトの出力点群がカラー情報を持たない場合、出力点群をより可視化表示するために、カスタマイズされたPythonスクリプトを使って、入力点群のZ値に基づいて疑似カラー点群を生成することができます。

主な処理流れは次の通りです。Z値を0～255の範囲にマッピングしてグレースケール値を取得します。その後、OpenCVを使用してグレースケール値をカラー値にマッピングして、疑似カラー点群を取得します。

Pythonのサンプルコードは以下の通りです。

```
import math

import cv2
import numpy as np

def generate_color_point_cloud_by_depth(point_cloud):
    x = point_cloud[:, 0]
    y = point_cloud[:, 1]
    z = point_cloud[:, 2]

    z_min = np.min(z)
    z_max = np.max(z)

    if math.isclose(z_max - z_min, 0):
        return np.column_stack((x, y, z, np.full(z.shape, 0xFFFFFFFF, np.uint32).view(np.float32)))

    color = 255 * (z - z_min) / (z_max - z_min)
    color = cv2.applyColorMap(color.astype(np.uint8), cv2.COLORMAP_JET)

    color = np.squeeze(color).astype(np.uint32)
    color = (0xFF000000 | (color[:, 0] << 16) | (color[:, 1] << 8) | color[:, 2]).view(np.float32)
```

```
return np.column_stack((x, y, z, color))
```

4.3.65. 指定した点群のプロパティを計算



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

ステップパラメータで点群の表す点を設定し、表す点の関連するプロパティ値を出力します。

使用シーン

汎用の点群プロパティ計算ステップです。特定の使用シーンはありません。

入力と出力

- **入力：** このポートに入力される点群には、プロパティが計算されます。
- **出力：** 点群をソートするための点群の指定したプロパティ値。

4.3.66. 構造化光センサーのキャリブレーション行列を計算



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

ラインレーザーのスキャン方向（レーザー平面に対して）をキャリブレーションします。

使用シーン

レーザーラインスキャンカメラのスキャン方向とレーザー平面の関係をキャリブレーションして、点群の変形を補正します。

入力と出力

- **入力：** キャリブレーションブロックの点群。
- **出力：** 計算によって得られたキャリブレーション行列。

4.3.67. ゼロでない領域の中心点を計算



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

2D画像のゼロでないピクセルの領域の中心点を計算します。

使用シーン

汎用の2D位置姿勢計算ステップです。特定の使用シーンはありません。

入力と出力

- **入力：** このポートに入力されるマスク画像には、中心点が計算されます。
- **出力：** 計算されたマスクの中心点。

4.3.68. 円柱の把持位置姿勢を計算



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

円柱軸の中心にある位置姿勢を計算します。

使用シーン

認識された円柱の中心の位置姿勢が軸の中心にない場合は、通常、ステップ [\[vision-steps:point-cloud-shape-detector:::point-cloud-shape-detector\]](#) の後で、このステップを使用して補正できます。

入力と出力

- **入力**
 1. このポートに入力される円柱の点群には、中心位置姿勢を計算して出力されます。
 2. 形状情報（通常はステップ「点群形状検出器」の出力からのものです）。
- **出力**
 1. 計算された入力点群の中心位置姿勢。

4.3.69. 3Dベクトルの長さを計算

機能

Vector3Dベクトルの長さを計算します。

使用シーン

汎用のベクトル長さの計算ステップです。特定の使用シーンはありません。

入力と出力

- **入力：** このポートに入力されるベクトルには、長さが計算されます。
- **出力：** 計算されたベクトル長さ。

4.3.70. キャリブレーションボードの位置姿勢を計算



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

カメラからの深度画像とカラー画像を入力し、カメラ座標系における固定なキャリブレーションボードの位置姿勢を計算します。

使用シーン

開発者のテスト用のステップです。普通のユーザーは、一般的な使用シーンでこのステップを使用する必要はありません。

入力と出力

- **入力：**
 1. カメラからの深度画像。
 2. カメラからのカラー画像。
- **出力：**
 1. 計算されたキャリブレーションボードの位置姿勢。
 2. キャリブレーションボードのドットを可視化するための画像。

4.3.71. キャリパスツール

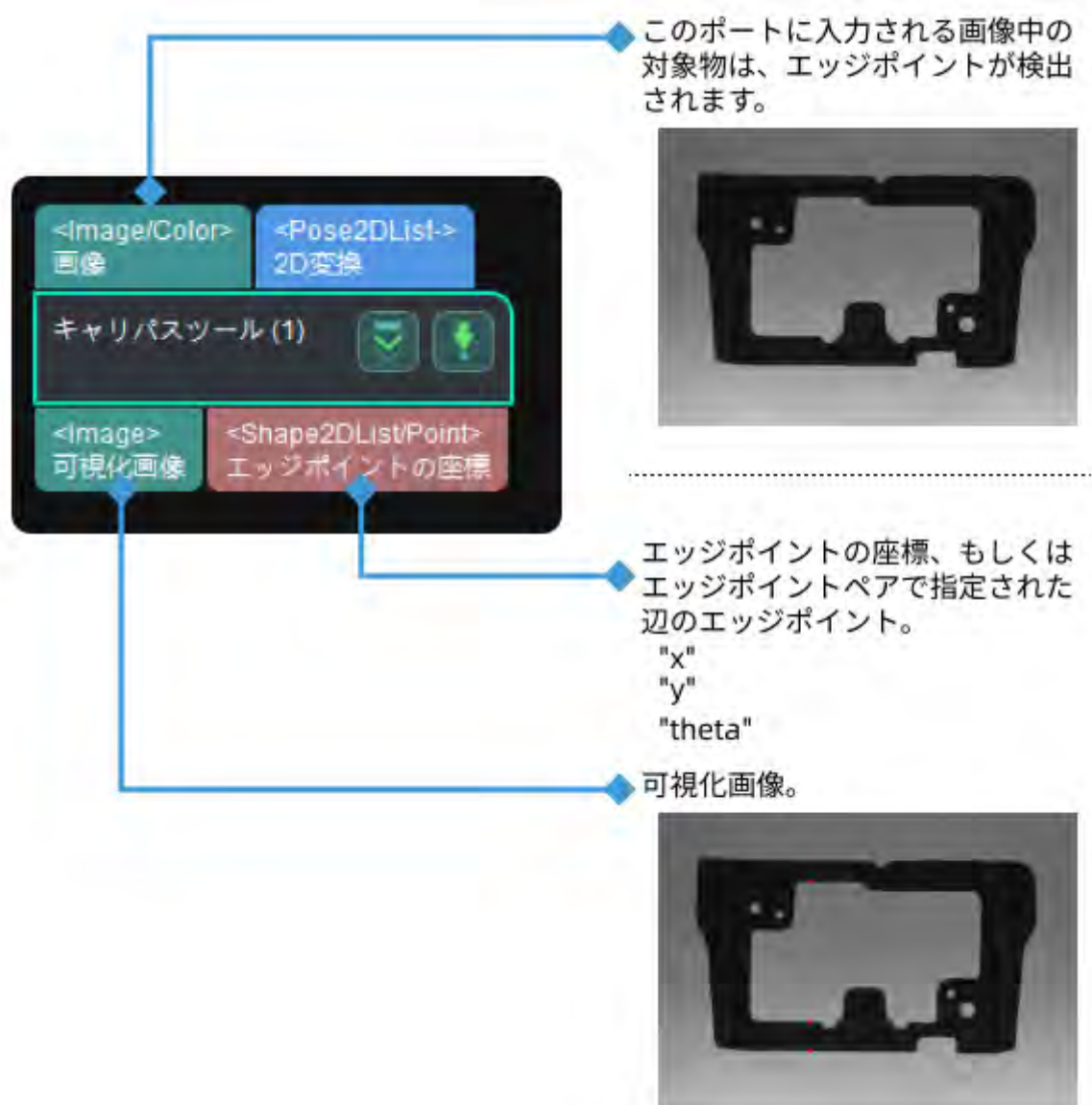
機能

狭いROIの垂直方向に沿ってエッジポイントまたはエッジポイントペアを検出し、エッジポイントの座標とペア間の距離（ポイントペアの検出の場合）をピクセル単位で出力するために使用されます。

使用シーン

対象物の特定部分の幅を測定したり、直線上に配置されたエッジポイントを見つけるために使用されます。このステップは通常、 [ピクセルを物理的な長さに変換](#) ステップと併用して、実際の幅の値を求めるために使用されます。

入力と出力



パラメータの説明

ROI

パラメータ説明：このパラメータは、エッジポイントまたはエッジポイントペアを検出する領域を指定し、その領域内に細くて長いエッジ検出フレームを生成するために使用されます。

調整アドバイス：ROIは **測定モード** で設定する必要があります。詳細な設定方法は [円の測定](#) をご参照ください。画像の品質が良い場合は、ROIの **幅** を1に設定し、**高さ** を対象物の検出範囲をカバーすることができます。

位置と向きの補正

パラメータ説明：このパラメータは、ROIを適切な位置に変換するために使用されます。

調整説明：このパラメータにチェックを入れると、以前実行中に設定されたROIが入力され

た「2D位置姿勢」に従って適切な位置に変換されます。

グレースケール変化の下限しきい値

パラメータ説明：このパラメータは、エッジ検出フレーム内のポイントのグレースケール変化の下限しきい値を設定するために使用されます。このしきい値を超えるグレースケール変化を持つポイントは、エッジポイントと見なされます。

初期値：20

ガウシアンフィルタ係数

パラメータ説明：ガウシアンフィルタリングの目的は、画像ノイズを除去することです。このパラメータは、エッジ検出フレームが垂直方向にフィルタリングするための係数を設定するために使用されます。

初期値：1.0

推奨値：1.0

調整アドバイス：画像の品質が良い場合、1の初期値を使用します。画像の品質が良くない場合、このパラメータを適切に上げます。

最大結果数

パラメータ説明：このパラメータは、出力するエッジポイントまたはエッジポイントペアの最大数を設定するために使用されます。

エッジポイントの場合、グレーの勾配が大きいものほど出力優先度が高くなります。

エッジポイントペアの場合、距離値が **エッジポイントペアの期待距離** に近いものほど、出力優先度が高くなります。

初期値：1

調整アドバイス：実際の状況に応じて設定してください。

出力順序

パラメータ説明：このパラメータは、選択した結果をどのような方式でソートして出力するかを設定するために使用されます。

オプション：上から下へ、下から上へ、中心から両側へ。

上から下へ：ROIの上部に近いほど、出力リストの中で結果が上位になります。

下から上へ：ROIの下部に近いほど、出力リストの中で結果が上位になります。

中心から両側へ：ROIの中心に近いほど、出力リストの中で結果が上位になります。

初期値：上から下へ。

エッジタイプ

パラメータ説明：このパラメータは、どのようなエッジを検出するかを設定するために使用されます。

オプション：シングルエッジ、エッジペア。

シングルエッジ：片側のみのエッジを検出し、個々のエッジポイントのリストを出力します。

エッジペア：両側のエッジを検出し、個々のポイントペアを表す2つのエッジポイントのリストを出力します。

初期値：シングルエッジ。

調整説明：調整効果を [調整の例](#) に示します。



エッジタイプを **エッジペア** に設定する場合、**エッジ極性0**、**エッジ極性1** および **エッジポイントペアの期待距離** を設定する必要があります。

エッジ極性0

パラメータ説明：エッジ極性は、エッジのどのようなグレースケール変化を対象物のエッジとみなすかを指定するために使用されます。このパラメータは、エッジポイントペアの単一エッジポイントまたは単一側のポイントを検出するために使用されます。

オプション：白から黒へ、黒から白へ、両方。

白から黒へ：ROI内の白から黒へのグレースケールの変化を対象物のエッジポイントと見なされます。

黒から白へ：ROI内の黒から白へのグレースケールの変化を対象物のエッジポイントと見なされます。

両方：ROI内の白から黒へのグレースケールの変化も黒から白へのグレースケールの変化も対象物のエッジポイントと見なされます。



グレースケール変化とは、ROI内の上から下へのグレースケールの変化をいいます。

初期値：白から黒へ。

調整説明：調整効果を [調整の例](#) に示します。

エッジ極性1

パラメータ説明：エッジ極性は、エッジのどのようなグレースケール変化を対象物のエッジとみなすかを指定するために使用されます。このパラメータは、エッジポイントペアの反対側にあるポイントを検出するために使用されます。エッジタイプを **エッジペア** に設定する場合、このパラメータを設定する必要があります。

オプション：パラメータ **エッジ極性0** と同じです。

エッジポイントペアの期待距離

パラメータ説明：このパラメータは、エッジポイントペア間の期待距離（ピクセル単位）を設定するために使用されます。

初期値：10

調整説明：距離値がこのパラメータに近いほど、エッジポイントペアの出力優先度は高くなります。

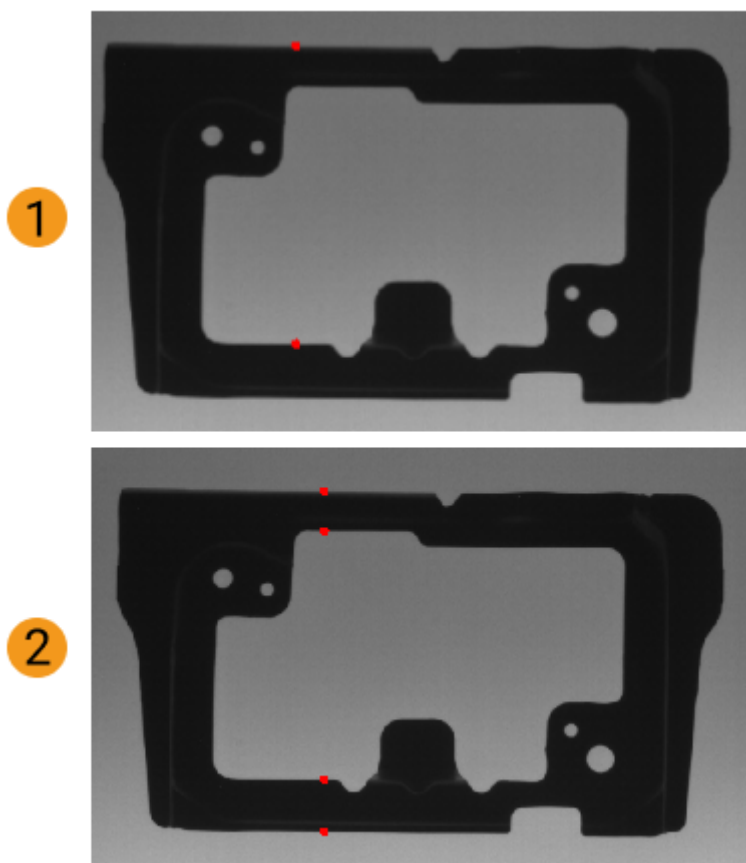


最大結果数を設定した後、優先度の低いポイントペアは無視されることがあります。

調整の例

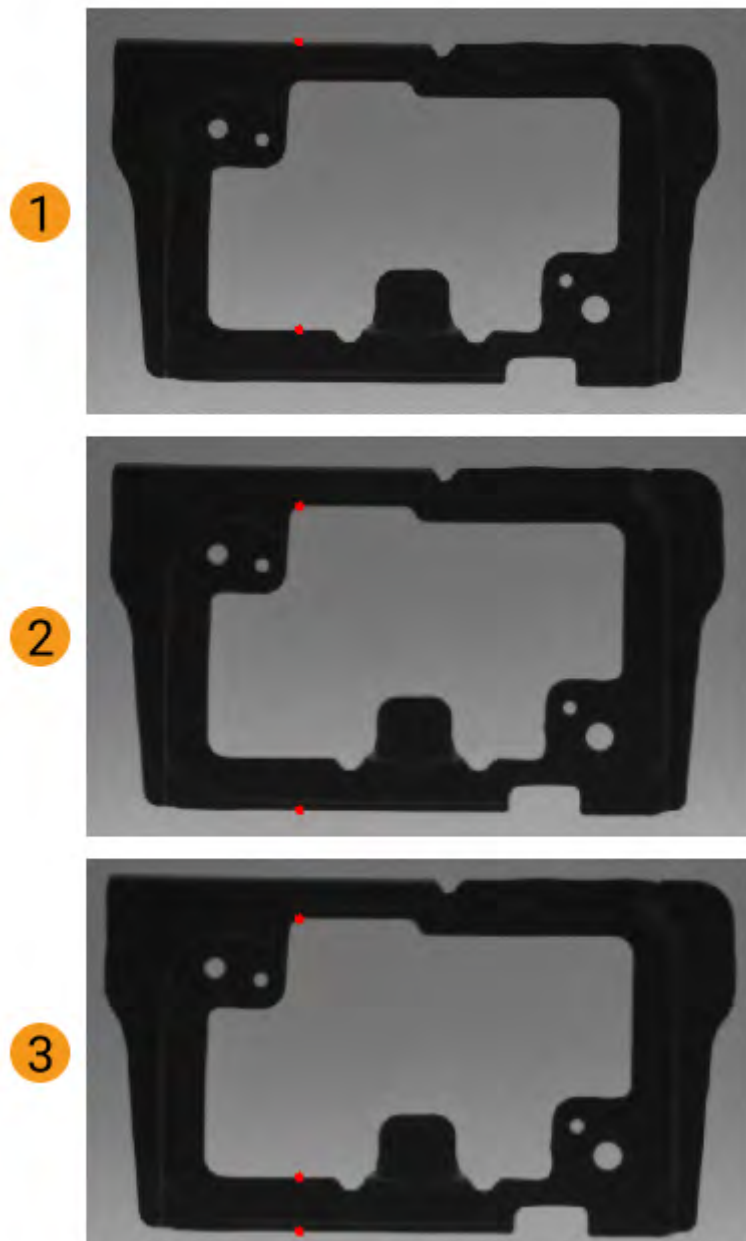
エッジタイプ

エッジタイプが**シングルエッジ**と**エッジペア**の場合、出力結果を以下の図1、図2のようになります。



エッジ極性

エッジ極性が**白から黒へ**、**黒から白へ**、**両方**の場合、出力結果をそれぞれ以下の図1、図2、図3のようになります。



4.3.72. 2Dカメラ



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

サードパーティ製の2Dカメラで画像を取得します。

使用シーン

汎用の2Dカメラステップです。

入力と出力

● 入力：

1. なし

● 出力：

1. カメラから取得したカラー画像。
2. カメラから取得したカラー画像が対応するタイムスタンプ文字列。これは、画像の名前付けプレフィックスを保存するために使用できます。

4.3.73. カメラから画像を取得

機能

このステップは、実際のカメラやローカルデータから、カラー画像、深度画像、点群データを取得します。

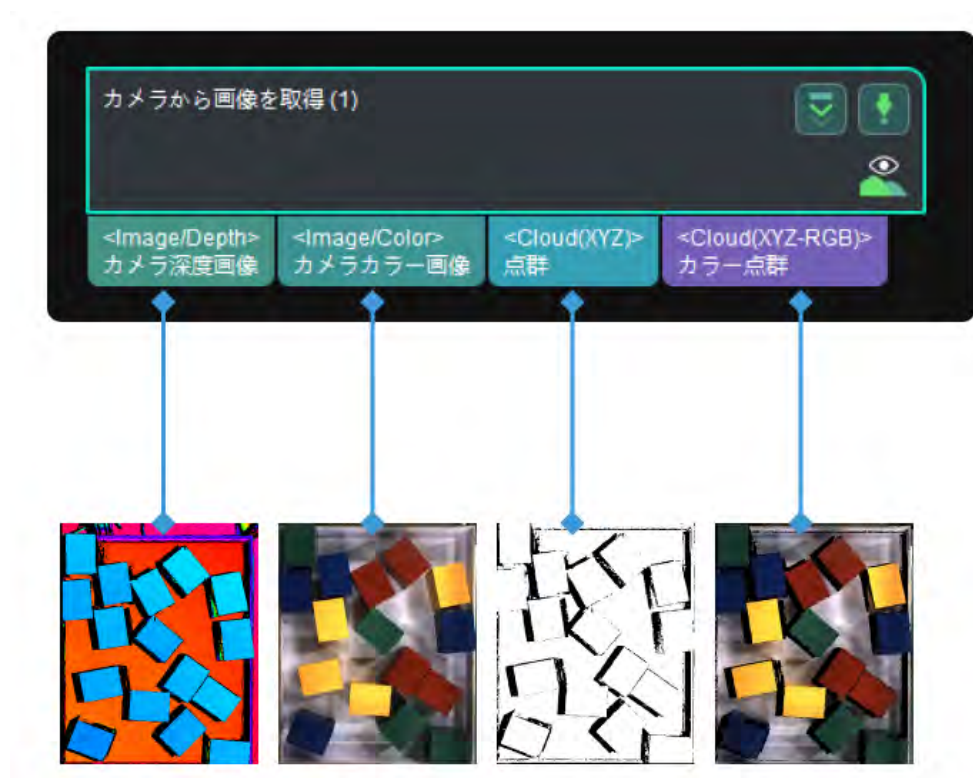


このステップでDEEP（V4）とLSR（V4）シリーズのカメラを接続する場合、出力されるカラー画像は、実際には2D画像（深度ソース）です。2D画像（深度ソース）を調整するには、Mech-Eye Viewerで **2D画像（深度ソース）の露光モード** と **2D画像（深度ソース）の露光時間** を調整してください。

使用シーン

通常、Mech-Visionプロジェクトの開始入力として使用されます。Mech-Eye産業用3Dカメラやサードパーティ製のカメラからデータを取得し、または仮想モードで保存された画像データを読み取るために使用されます。

入力と出力



パラメータの調整説明

カメラ設定

カメラのタイプ

パラメータ説明：このパラメータは、使用するカメラのタイプを選択するために使用されます。

オプション：

- External2D：外部2Dカメラ。
- LMI：LMIカメラ。
- Mech-Eye：初期値、Mech-Eye産業用3Dカメラ。
- Mech-EyeTOF：Mech-Eye産業用3Dカスタムカメラ。



- Mech-Eye産業用3Dカメラのファームウェアバージョンを2.0.0以降にアップグレードした後は、1.6.2以降バージョンのMech-Visionを使用することをお勧めします。1.6.1以前バージョンのMech-Visionを使用する場合、**External2D EBUSCamera** でMech-Eye産業用3Dカメラに接続するとソフトウェアがクラッシュする可能性があります。
- Mech-Eyeのみ、**仮想モード** に対応しています。

調整説明：調整可能なパラメータは、カメラタイプによって異なります。実際の状況に応じて設定してください。

Mech-Eye


仮想モード を無効にする場合、実際のカメラを使用します。**仮想モード** を有効にする場合、ローカルデータを使用します。デフォルトでは無効になっています。

実際カメラを使用



カメラID

パラメータ説明：このパラメータは、カメラを接続し、キャリブレーションパラメータグループを選択するために使用されます。

設定方法：

1. **[カメラを選択]** をクリックして **カメラとパラメータの選択画面** が表示されます。
2. 接続するカメラを選択し、 をクリックします。



カメラが正常に接続されると、 ボタンが  に変わります。

3. 接続後、**パラメータグループを選択** のドロップダウンリストからキャリブレーションパラメータグループを選択し、**[OK]** をクリックします。



キャリブレーションパラメータグループが選択後、カメラID、**キャリブレーションパラメータグループ**、**IP** と **ポート** が自動的に記入されます。

キャリブレーションパラメータグループ

パラメータ説明：このパラメータは、カメラのキャリブレーションパラメータグループを選択するために使用されます。

IP

パラメータ説明：このパラメータは、カメラのIPアドレスを表示するために使用されます。変更しないでください。

ポート

パラメータ説明：このパラメータは、カメラのポート番号を表示するために使用されます。変更しないでください。



IP と **ポート** パラメータは、表示のみに使用されます。この2つのパラメータを変更すると、カメラは切断されます。

タイムアウト

パラメータ説明：このパラメータは、カメラがメッセージを受信してからクライアントにデータを返すまでの最大時間（秒単位）を設定するために使用されます。次の2つの場合はタイムアウトとなります。

- 最大時間内、カメラが正常に接続されなかった
- 最大時間内、カメラが正常にデータを取得できなかった

初期値：20s

再接続の最大回数

パラメータ説明：このパラメータは、カメラ接続タイムアウトによりカメラ接続に失敗した場合に、カメラ接続を再試行する最大回数を指定するために使用されます。

初期値：3

コンフィグパラメータファイル

パラメータ説明：このパラメータは、カメラのパラメータグループを選択するために使用されます。カメラは、現在のパラメータグループ内のパラメータに従ってデータを取得します。



また、Mech-Eye Viewerで [パラメータグループを削除/追加](#) することができます。

カメラ型番

パラメータ説明：読み取り専用パラメータ。このパラメータは、現在接続されているカメラの型番を表示するために使用されます。

2D画像のソース

パラメータ説明：このパラメータは、カメラが出力する2D画像のタイプを設定するために使用されます。

オプション：

- 同じソースのテクスチャ画像：初期値。2D画像（テクスチャ）、深度画像と同じ座標系、補正は不要。
- 外部カラー画像：2D画像（テクスチャ）で、使用する前に補正が必要です。このパラメータを使用するには、**深度画像の座標系に補正**にチェックを入れる必要があります。



このパラメータは、LSR (V4)とDEEP (V4)シリーズカメラにのみ適用可能です。

深度画像の座標系に補正

パラメータ説明：このパラメータは、外部のカラー画像を深度画像の座標系に補正するために使用されます。

初期値：チェックを入れる。



このパラメータは、**2D画像のソース**が**外部カラー画像**の場合にのみ設定できます。

静的背景除去設定

深度画像による背景を除去

パラメータ説明：このパラメータは、深度画像内の背景情報を除去するために使用されます。

初期値：オフ。

深度背景画像

パラメータ説明：このパラメータは、深度背景画像の取得または読み取りに使用されます。


初期値：depth_background.png

設定方法：

1. **[深度背景画像ファイルを選択]** をクリックし、背景設定の画面に入ります。

適切な背景画像がローカルにある場合は、手順2に進みます。

。適切な背景画像がローカルにない場合は、手順3に進みます。

2. **[...]** をクリックして画像を選択し、**[確認]** をクリックすると深度背景画像の選択は完了です。
3. 画面の右側にある **検出されたカメラ** でカメラを選択し、 をクリックしてカメラを接続します。



カメラが正常に接続されると、左側の **画像ビュー** でカメラから取得した深度画像が表示されます。

4. 深度画像を更新する場合、**[一回キャプチャ]** または **[連続キャプチャ]** をクリックします。
5. 背景画像のファイル名を変更する場合、**画像名** にファイル名を入力します。
6. **[確認]** をクリックし、深度背景画像の選択は完了です。

背景深度の変動

パラメータ説明：このパラメータは、対象物内の点の深度と背景画像の深度を比較するために使用されます。単位はミリメートル（mm）です。点の深度と背景画像の深度がこの値より小さい場合、その点は背景点として除去されます。

初期値：10mm

撮影回数

パラメータ説明：このパラメータは、カメラ取得タイムアウトによりカメラがデータ取得に失敗した場合に、データを取得する総回数を指定するために使用されます。

初期値：3

推奨値：3

Mech-Centerにあるロボットの名前

パラメータ説明：このパラメータは、ロボットサービスの名前を入力するために使用されます。Mech-Centerに接続されているロボットの名前および型番と一致する必要があります。

初期値：なし。

仮想カメラを使用


仮想モード を有効にした場合、設定が必要なパラメータは以下の通りです。

キャリブレーションパラメータグループ

調整説明： **画像のデータパス** を選択したら、 **キャリブレーションパラメータグループ** が自動的に自動的に入力されます。 **データパス** に複数の キャリブレーションパラメータグループがある場合、このオプションのドロップダウンリストから使用するキャリブレーションパラメータグループを選択できます。

データパス

パラメータ説明：このパラメータは、画像と内部・外部パラメータが格納されているフォルダを選択するために使用されます。

設定方法：右側にある  をクリックして、画像と内部・外部パラメータが格納されているフォルダを選択します。




仮想モードでは、画像ファイル名とファイル形式は以下の要件に従う必要があります。

- カラー画像は、rgb_image_xxxx.jpgのような名前を付ける必要があります。
- 深度画像は、depth_image_xxxx.pngのような名前を付ける必要があります。


カメラがEye-in-Hand方式で取り付けられている場合、画像番号とフランジ位置姿勢が対応していることに注意してください。

仮想カメラアシスタント

- 以下の場合で、画像データを選択するための **仮想カメラアシスタント** がトリガーされます。
 - **データパス** にパスを入力していない場合。
 - 選択されたフォルダには複数グループのデータがある場合。
 - 選択されフォルダにはカラー画像や深度画像、内部・外部パラメータデータが欠落している場合。
- 仮想カメラアシスタントを使用する場合、データの読み取り手順は以下の通りです。
 - **仮想カメラアシスタント** で  をクリックし、データパスを選択します。



パスを選択した後、 **カメラID** と **パラメータグループ** が自動的に記入されます。

- 次に **[検証]** をクリックすると、 **カメラパラメータグループが更新されました。** というメッセージが表示されます。
-  をクリックし、カラー画像、深度画像およびフランジ位置姿勢が格納されているフォルダを選択します。選択したら、 **[確認]** をクリックします。



カメラがEye in Hand方式で取り付けられた場合にのみ、フランジ位置姿勢の追加が必要です。

- 最後に、表示される **設定済み** の画面で **[OK]** をクリックします。



仮想モードでは、データベース下の画像データが変更された場合、データベースの再選択が必要です。そうしないと、変更された画像データを読み取ることはできません。

プレイモード

パラメータ説明：このパラメータは、画像の読み取り順序を指定するために使用されます。

オプション

- 順次読み込み：初期値。フォルダ内の画像順序で画像を読み取ります。
- 1つ繰り返し：現在の画像を繰り返し読み取ります。
- すべて繰り返し：すべての画像を順番に読み取り、最後の1枚の読み取りが終わった時点で最初から読み取ります。
- ランダム読み込み：画像をランダムに読み取ります。

調整説明：実際のニーズに応じて選択してください。

現在のフレーム

パラメータ説明：このパラメータは、現在読み込んでいる画像の番号や時刻を表示するために使用されます。

画像名タイプ

パラメータ説明：このパラメータは、**カラー画像のパス** の出力ポートから出力された画像名のタイプを選択するために使用されます。

オプション：フルパス、ベース名、ファイル名。

初期値：フルパス。

静的背景除去設定

深度画像による背景を除去

パラメータ説明：このパラメータは、深度画像内の背景情報を除去するために使用されます。

初期値：オフ。


深度背景画像

パラメータ説明：このパラメータは、深度背景画像の取得または読み取りに使用されます。

初期値：depth_background.png

設定方法：

1. **[深度背景画像ファイルを選択]** をクリックし、背景設定の画面に入ります。

- 適切な背景画像がローカルにある場合は、手順2に進みます。
 - 適切な背景画像がローカルにない場合は、手順3に進みます。
2. [...] をクリックして画像を選択し、[確認] をクリックすると深度背景画像の選択は完了です。
 3. 画面の右側にある **検出されたカメラ** でカメラを選択し、 をクリックしてカメラを接続します。



カメラが正常に接続されると、左側の **画像ビュー** でカメラから取得した深度画像が表示されます。

4. 深度画像を更新する場合、[一回キャプチャ] または [連続キャプチャ] をクリックします。
5. 背景画像のファイル名を変更する場合、**画像名** にファイル名を入力します。
6. [確認] をクリックし、深度背景画像の選択は完了です。

背景深度の変動

パラメータ説明：このパラメータは、対象物内の点の深度と背景画像の深度を比較するために使用されます。単位はミリメートル（mm）です。点の深度と背景画像の深度がこの値より小さい場合、その点は背景点として除去されます。


初期値：10mm

External2D

カメラID

パラメータ説明：このパラメータは、カメラを接続し、キャリブレーションパラメータグループを選択するために使用されます。

設定方法：

1. [カメラを選択] をクリックして **カメラとパラメータの選択画面** が表示されます。
2. 接続するカメラを選択し、 をクリックします。



カメラが正常に接続されると、 は  に変わります。

3. 接続後、**パラメータグループを選択** のドロップダウンリストからキャリブレーションパラメータグループを選択し、[OK] をクリックします。



キャリブレーションパラメータグループが選択後、カメラID、**キャリブレーションパラメータグループ**、IP と **ポート** が自動的に記入されます。

キャリブレーションパラメータグループ

パラメータ説明：このパラメータは、カメラのキャリブレーションパラメータグループを選択するために使用されます。

IP

パラメータ説明：このパラメータは、カメラのIPアドレスを表示するために使用されます。変更しないでください。

ポート

パラメータ説明：このパラメータは、カメラのポート番号を読み取るために使用されます。



IP と ポート パラメータは、読み取りのみに使用されます。この2つのパラメータを変更すると、カメラは切断されます。

タイムアウト

パラメータ説明：このパラメータは、カメラがメッセージを受信してからクライアントにデータを返すまでの最大時間（秒単位）を設定するために使用されます。次の2つの場合はタイムアウトとなります。

- 最大時間内、カメラが正常に接続されなかった
- 最大時間内、カメラが正常にデータを取得できなかった

初期値：20s

再接続の最大回数

パラメータ説明：このパラメータは、カメラ接続タイムアウトによりカメラ接続に失敗した場合に、カメラ接続を再試行する最大回数を指定するために使用されます。

初期値：3

Auto

パラメータ説明：このパラメータは、自動露光を使用するかどうかを設定するために使用されます。

初期値：チェックを入れる

平均諸調値

パラメータ説明：このパラメータは、期待される画像グレースケール値を設定するために使用されます。値が高いほど画像は明るくなり、値が低いほど画像は暗くなります。

初期値：100

露光時間

パラメータ説明：このパラメータは画像の明るさに影響します。値が高いほど画像は明るくなり、値が低いほど画像は暗くなります。

初期値：10

撮影回数

パラメータ説明：このパラメータは、カメラ取得タイムアウトによりカメラがデータ取得に

失敗した場合に、データを取得する総回数を指定するために使用されます。

初期値：3

推奨値：3

Mech-Centerにあるロボットの名前

パラメータ説明：このパラメータは、ロボットサービスの名前を入力するために使用されます。Mech-Centerに接続されているロボットの名前および型番と一致する必要があります。


初期値：なし。

LMI



カメラID

パラメータ説明：このパラメータは、カメラを接続し、キャリブレーションパラメータグループを選択するために使用されます。

設定方法：

1. **[カメラを選択]** をクリックして **カメラとパラメータの選択画面** が表示されます。
2. 接続するカメラを選択し、 をクリックします。



カメラが正常に接続されると、 ボタンが  に変わります。

3. 接続後、**パラメータグループを選択** のドロップダウンリストからキャリブレーションパラメータグループを選択し、**[OK]** をクリックします。



キャリブレーションパラメータグループが選択後、カメラID、**キャリブレーションパラメータグループ**、**IP** と **ポート** が自動的に記入されます。

キャリブレーションパラメータグループ

パラメータ説明：このパラメータは、カメラのキャリブレーションパラメータグループを選択するために使用されます。

IP

パラメータ説明：このパラメータは、カメラのIPアドレスを表示するために使用されます。変更しないでください。

撮影回数

パラメータ説明：このパラメータは、カメラ取得タイムアウトによりカメラがデータ取得に失敗した場合に、データを取得する総回数を指定するために使用されます。

初期値：3

推奨値：3

Mech-Centerにあるロボットの名前

パラメータ説明：このパラメータは、ロボットサービスの名前を入力するために使用されます。Mech-Centerに接続されているロボットの名前および型番と一致する必要があります。

初期値：なし。

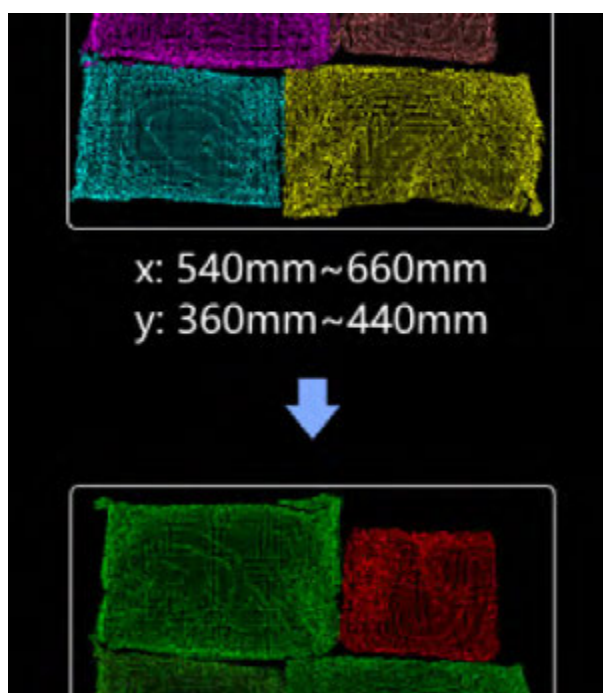
Mech-EyeTOF

必要な場合、Mech-Mind株式会社にお問い合わせください。

4.3.74. 点群のサイズによって分類

機能

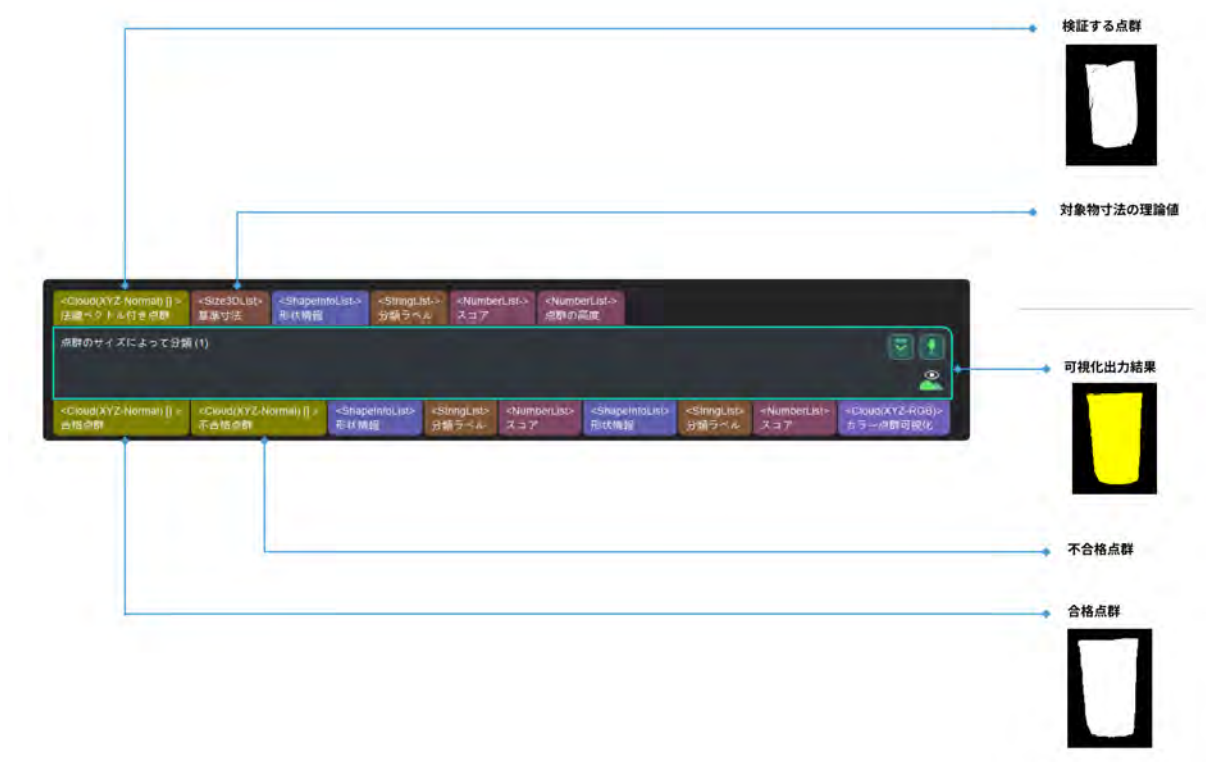
基準サイズに基づいて、入力点群が合格かどうかを判断します。入力点群のサイズと基準サイズの偏差がしきい値の範囲内にある場合、点群は合格と見なされます。それ以外の場合は、点群は不合格と見なされます。



使用シーン

通常、麻袋のデパレタイジングシーンに使用されます。ディープラーニングによる麻袋の認識偏差を防ぐために、このステップを使用して3Dアルゴリズムのエラー防止メカニズムを強化し、寸法に従って前のステップで認識した麻袋の点群が有効かどうかを判断します。[\[vision-steps:apply-masks-to-point-cloud:::apply-masks-to-point-cloud\]](#)、[最高層の点群を取得](#)などのステップと併用されます。

入力と出力



パラメータの説明

等長さの差のしきい値

長さの差 = 点群サイズの長さ - 基準サイズの長さ

比率の下限しきい値

パラメータ説明：このパラメータは、長さの差と基準サイズの長さの間の最小比率のしきい値を設定するために使用されます。基準サイズの長さに対する長さの差の比率がこのしきい値より大きい場合、点群は合格となり、基準サイズの長さに対する長さの差の比率がこのしきい値より小さい場合、点群は不合格となります。

初期値：-0.2500

値の範囲：-1~0

調整アドバイス：実際の状況に応じて設定してください。

比率の上限しきい値

パラメータ説明：このパラメータは、長さの差と基準サイズの長さの間の最大比率のしきい値を設定するために使用されます。基準サイズの長さに対する長さの差の比率がこのしきい値より小さい場合、点群は合格となり、基準サイズの長さに対する長さの差の比率がこのしきい値より大きい場合、点群は不合格となります。

初期値：0.2500

値の範囲：0~1

調整アドバイス：実際の状況に応じて設定してください。



- 比率の下限しきい値<長さの差/基準サイズの長さ<比率の上限しきい値の場合、点群は合格となります。
- 長さの差/基準サイズの長さ>比率の上限しきい値の場合、点群は不合格となります。
- 長さの差/基準サイズの長さ<比率の下限しきい値の場合、点群は不合格となります。

幅の差のしきい値設定

幅の差 = 点群サイズの幅 - 基準サイズの幅

比率の下限しきい値

パラメータ説明：このパラメータは、幅の差と基準サイズの幅の間の最小比率のしきい値を設定するために使用されます。基準サイズの幅に対する幅差の比率がこのしきい値より大きい場合、点群は合格となり、基準サイズの幅に対する幅差の比率がこのしきい値より小さい場合、点群は不合格となります。

初期値：-0.2500

値の範囲：-1~0

調整アドバイス：実際の状況に応じて設定してください。

比率の上限しきい値

パラメータ説明：このパラメータは、幅の差と基準サイズの幅の間の最大比率のしきい値を設定するために使用されます。基準サイズの幅に対する幅差の比率がこのしきい値より小さい場合、点群は合格となり、基準サイズの幅に対する幅差の比率がこのしきい値より大きい場合、点群は不合格となります。

初期値：0.2500

値の範囲：0~1

調整アドバイス：実際の状況に応じて設定してください。



- 比率の下限しきい値<幅の差/基準サイズの幅<比率の上限しきい値の場合、点群は合格となります。
- 幅の差/基準サイズの幅>比率の上限しきい値の場合、点群は不合格となります。
- 幅の差/基準サイズの幅<比率の下限しきい値の場合、点群は不合格となります。

高さの差のしきい値設定

高さの差 = 点群サイズの高さ - 基準サイズの高さ

比率の下限しきい値

パラメータ説明：このパラメータは、基準サイズの高さに対する高さの差の最小比率のしきい値を設定するために使用されます。基準サイズの高さに対する高さの差の比率がこのしきい値より大きい場合、点群は合格となり、基準サイズの高さに対する高さの差の比率がこのしきい値より小さい場合、点群は不合格となります。

初期値：-100.0000

調整アドバイス：実際の状況に応じて設定してください。

比率の上限しきい値

パラメータ説明：このパラメータは、基準サイズの高さに対する高さの差の最大比率のしきい値を設定するために使用されます。基準サイズの高さに対する高さの差の比率がこのしきい値より小さい場合、点群は合格となり、基準サイズの高さに対する高さの差の比率がこのしきい値より大きい場合、点群は不合格となります。

初期値：100.0000

調整アドバイス：実際の状況に応じて設定してください。

基準高さ

パラメータ説明：このパラメータは、基準高さ（ミリメートル単位）を設定するために使用されます。

初期値：1000.000mm

調整アドバイス：実際の状況に応じて設定してください。



- 比率の下限しきい値 < 高さの差/基準サイズの高さ < 比率の上限しきい値の場合、点群は合格となります。
- 高さの差/基準サイズの高さ > 比率の上限しきい値の場合、点群は不合格となります。
- 高さの差/基準サイズの高さ < 比率の下限しきい値の場合、点群は不合格となります。

アスペクト比の設定

アスペクト比の差 = 点群のアスペクト比 - 基準アスペクト比

アスペクト比のしきい値

パラメータ説明：このパラメータは、基準アスペクト比に対するアスペクト比の差の比率のしきい値を設定するために使用されます。基準アスペクト比に対するアスペクト比の差の比率が（1±しきい値）以内であれば点群が合格となり、（1±しきい値）以外であれば点群が不合格となります。

初期値：1000000.0000

調整アドバイス：実際の状況に応じて設定してください。



- $-(\text{アスペクト比のしきい値}) < \text{アスペクト比の差/基準アスペクト比} < \text{アスペクト比のしきい値}$ の場合、点群が合格となります。
- $\text{アスペクト比の差/基準アスペクト比} > \text{アスペクト比のしきい値}$ の場合、点群が不合格となります。
- $\text{アスペクト比の差/基準アスペクト比} < -(\text{アスペクト比のしきい値})$ の場合、点群が不合格となります。

4.3.75. 点群の歪み補正



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

入力点群は、カメラの歪み係数で歪み補正を実行します。

使用シーン

特定のシーン向けのカスタマイズのステップです。

入力と出力

- **入力：** このポートに入力される点群は補正されます。
- **出力：** 補正された点群。

4.3.76. モデルフィッティングからの位置姿勢によって点群をフィルタリング



これは古いバージョンのステップであり、メンテナンスが停止されますので、新しいバージョンのステップ [\[vision-steps:remove-cloud-points-from-point-cloud::remove-cloud-points-from-point-cloud\]](#) を使用してください。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

入力点群とパラメータで読み込まれたモデルを対比し、重複する点が除去されます。

使用シーン

通常は、シーンから点群を除去する、もしくは、モデルの対応する点群を取得するために使用されます。

入力と出力

● 入力：

1. このポートに入力される点群はフィルタリングされます。
2. モデルをシーンに変換するための位置姿勢。

● 出力：

1. 残りの点群。
2. 除去された点群。

4.3.77. 点群リストの要素を削除



これは古いバージョンのステップであり、メンテナンスが停止されています。このステップに関する詳細が必要な場合は、当社 (docs@mech-mind.net) までお問い合わせください。

機能

点群リストの点群の順序に従って、最初のN個の点群を保持します（または、最後のN個の点群を除去します）。N値はステップのパラメータで設定できます。

使用シーン

点群リストの要素を削除するために使用されます。

これは古いバージョンのステップであり、メンテナンスが停止されますので、新しいバージョンのステップの組合せを使用してください。

入力と出力

● 入力：

1. このポートに入力された点群リストの要素は一部削除されます。
2. 各入力点群のラベル。
3. 各点群の信頼度。

● 出力：

1. 残りの点群リスト。
2. 残りの点群ラベル。
3. 残りの点群の信頼度。

4.3.78. 点群処理（GPU）



これは古いバージョンのステップであり、メンテナンスが停止されますので、新しいバージョンのステップ [点群の法線ベクトルを計算してフィルタリング](#) を使用してください。こ

のステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

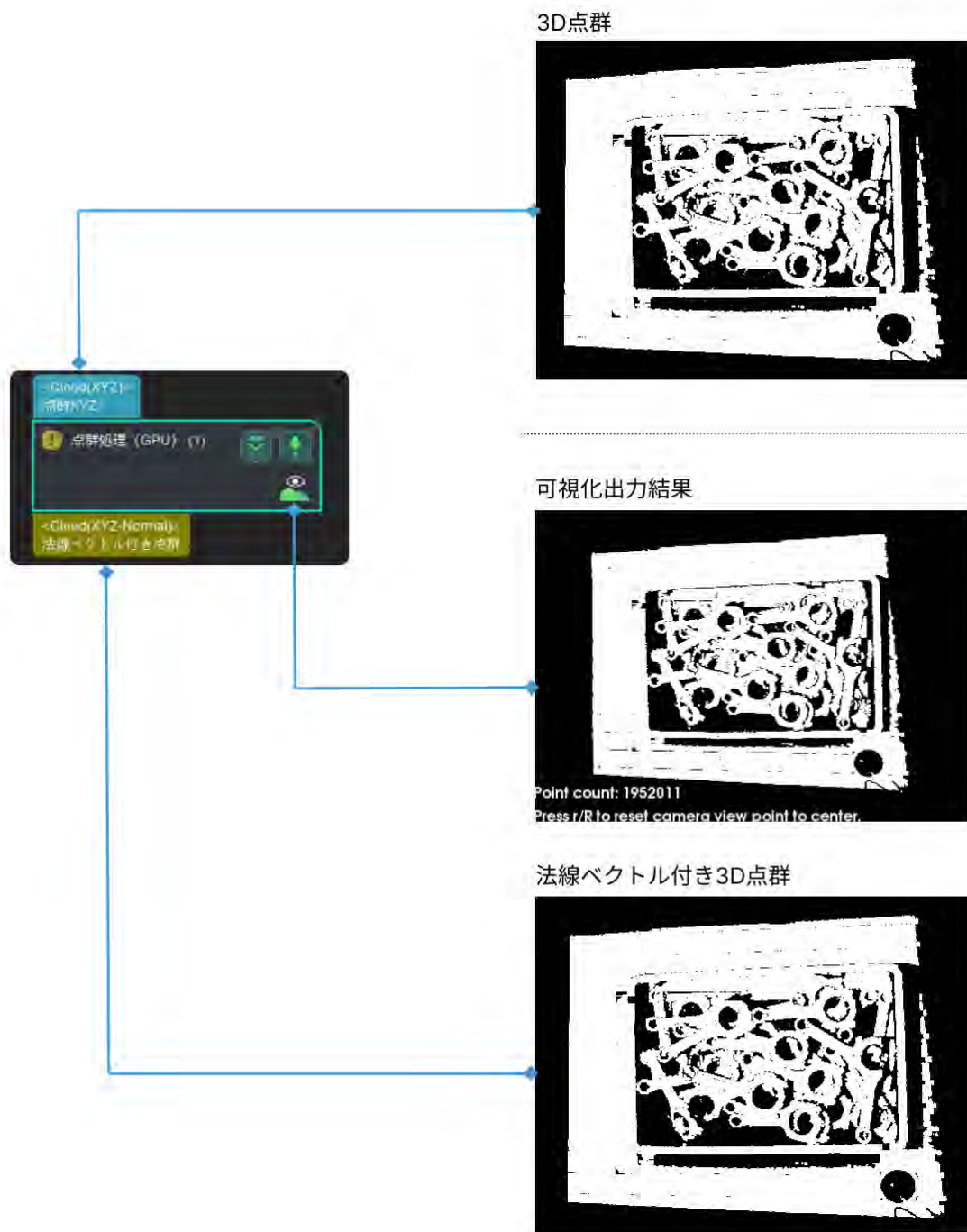
機能

点群の法線ベクトルを計算し、外れ値を除去します。

使用シーン

元の点群に法線ベクトルを追加し、外れ値を除去します。通常、[カメラから画像を取得](#)、[深度画像を点群に変換](#) ステップの後で使用されます。

入力と出力



3D点群

可視化出力結果

法線ベクトル付き3D点群

パラメータの説明

計算設定

バッチ処理するか

パラメータ説明：このパラメータは、点群のバッチ処理を実行するかどうかを設定するために使用されます。

初期値：チェックを入れない。

調整アドバイス：点群の点数が多く、PCのメモリが小さい場合、このパラメータにチェックを入れることをお勧めします。

法線ベクトルを計算するときの半径

パラメータ説明：このパラメータは、法線ベクトルを計算する時の検索半径を設定するために使用されます。

初期値：5.000 mm

外れ値の除去

外れ値を除去

パラメータ説明：このパラメータは、点群の外れ値を除去するかどうかを設定するために使用されます。

初期値：チェックを入れない。

調整説明：チェックを入れると、設定された外れ値のしきい値に従って外れ値を除去することができます。

4.3.79. 点群スケーリング



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

入力点群の各ポイントの座標にスケーリング係数を掛けます。

使用シーン

汎用の点群処理ステップです。特定の使用シーンはありません。

入力と出力

- **入力**：このポートに入力される点群は、スケーリングされます。
- **出力**：スケーリング後の点群。

4.3.80. 点群平滑化と法線ベクトル計算



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

点群平滑化および法線ベクトルの計算を行います。

使用シーン

通常は点群変動が激しいシーンで点群を平滑化するために使用されます。

入力と出力

- **入力：** このポートに入力される点群は、平滑化処理および法線ベクトルの計算が行われます。
- **出力：** 平滑化処理および法線ベクトルの計算が行われた点群。

4.3.81. 3D ボックス内の点群を抽出



これは古いバージョンのステップであり、メンテナンスが停止されていますので、新しいバージョンのステップ [3D ROI内の点群を抽出](#) を使用してください。このステップに関する詳細が必要な場合は、当社 (docs@mech-mind.net) までお問い合わせください。

機能

3D ROI内の点群を抽出します。

使用シーン

汎用の点群処理ステップです。特定の使用シーンはありません。

入力と出力

- **入力：** このポートに入力される点群には、3D ROIが抽出されます。
- **出力：** 3D ROI内の点群。

4.3.82. 点群における法線ベクトルを追加



これは古いバージョンのステップであり、メンテナンスが停止されますので、新しいバージョンのステップ [\[vision-steps:calc-normals-of-point-cloud-and-filter-it:::calc-normals-of-point-cloud-and-filter-it\]](#) を使用して、法線ベクトルの計算方法を **CloudXYZToCloudNormal** に設定してください。このステップに関する詳細が必要な場合は、当社 (docs@mech-mind.net) までお問い合わせください。

機能

カメラ座標系の法線ベクトルを点群に追加します。

使用シーン

汎用の点群処理ステップです。

入力と出力

入力： 3D点群。

- 出力：法線ベクトル付きの3D点群。

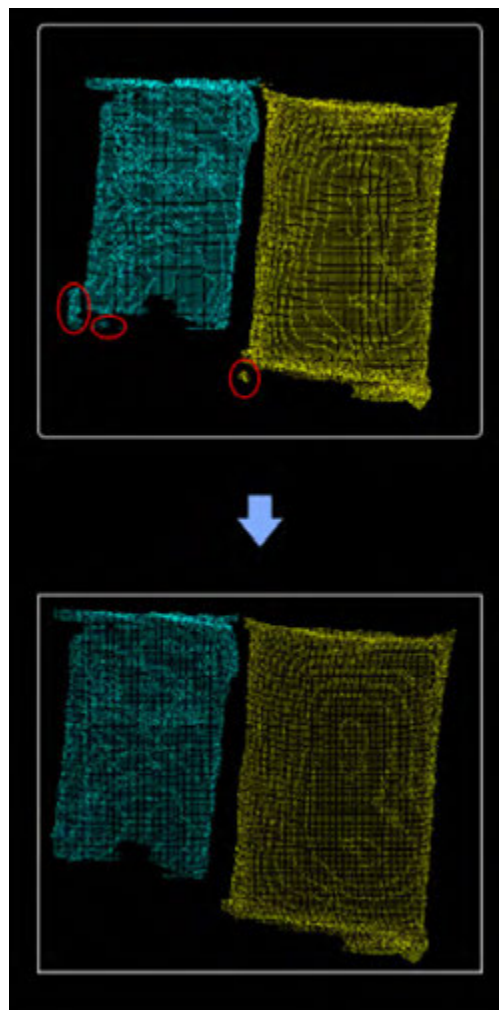
4.3.83. 点群をクラスタリングして要件を満たす点群を出力



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

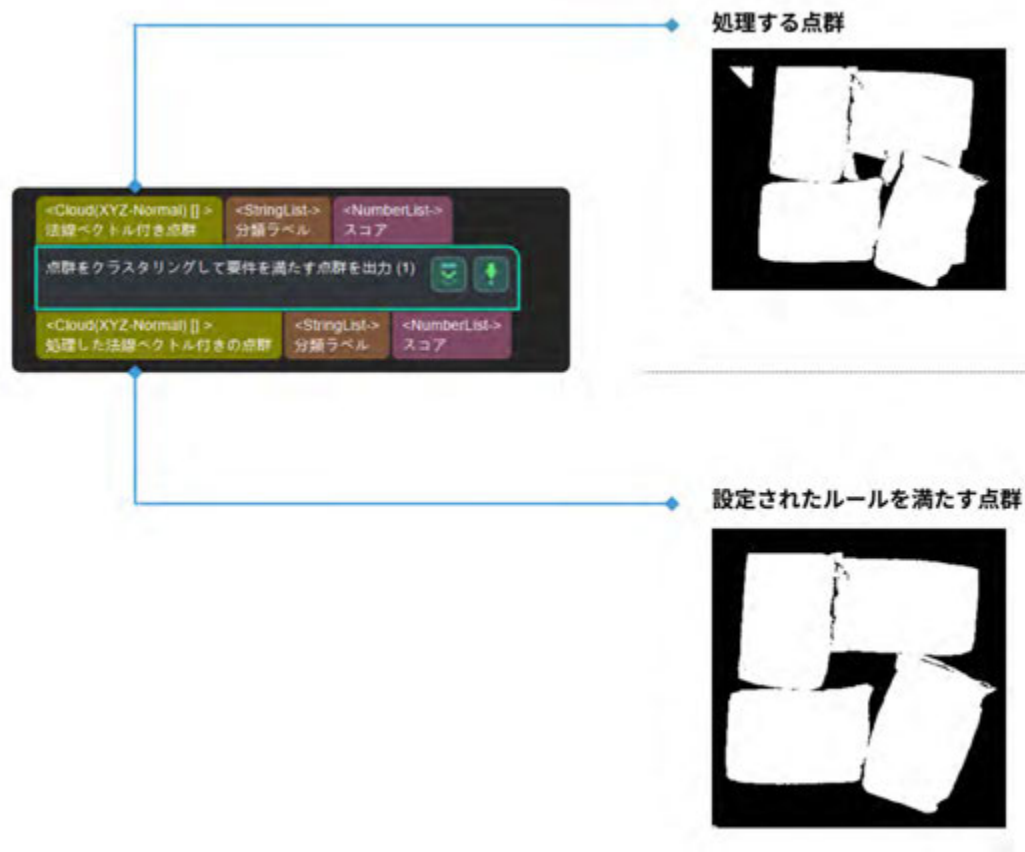
点群をクラスタリングし、クラスタリング結果に設定されたルールに準拠した点群を出力します。



使用シーン

通常、個々の対象物の点群のノイズを除去するために使用されます。

入力と出力



4.3.84. 連続した線を描画



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

入力された2D画像内の対象物の輪郭線分を検出し、線分を連続な線として描画します。

使用シーン

汎用の2D画像エッジ接続ステップです。特定の使用シーンはありません。

入力と出力

- **入力：** このポートに入力される画像を元に、連続な輪郭が描かれます。
- **出力：** 連続した線のマスク。

4.3.85. 3D ROI内の位置姿勢を取得



これは古いバージョンのステップであり、メンテナンスが停止されますので、新しいバージョンのステップ [位置姿勢が3D ROI内にあるかどうかを検証](#) を使用してください。このス

トップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

入力位置姿勢リストを、設定された3D ROIによってフィルタリングします。ROI内の位置姿勢は保持され、それ以外の位置姿勢は破棄されます。

使用シーン

汎用の位置姿勢フィルタリングのステップです。

入力と出力



パラメータの説明

3D ROI設定

3D ROIを設定

パラメータ説明：このパラメータは、3D ROIを設定するために使用されます。

調整説明：右側にある [3D ROIを設定] をクリックし、表示される画面で [ROI設定](#) を行います。

入力位置姿勢の座標系

パラメータ説明：このパラメータは、位置姿勢の座標系を選択するために使用されます。

オプション：ロボット座標系、カメラ座標系。

初期値：ロボット座標系。

4.3.86. 二つの深度画像を比較



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

パッシブモードとアクティブモードのイメージングアルゴリズムをデバッグするために、両深度画像の各ピクセルの深度値の差を比較し、深度値の差を2D画像として表示します。

使用シーン

カメラのパッシブステレオモードをデバッグするための開発者機能です。普通のユーザーは、一般的な使用シーンでこのステップを使用する必要はありません。

入力と出力

● 入力：

1. 深度値の差を計算するための最初の深度画像。
2. 深度値の差を計算するための2番目の深度画像。

● 出力：

1. 深度値の差を表示する画像（つまり、入力された最初の深度画像と2番目の深度画像の各ピクセルの深度値の差）。

4.3.87. 数値比較



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

設定された操作タイプに従って入力された二つの数値リストを比較して、比較結果のブール値（True/False）リストを出力します。

使用シーン

汎用の数値比較ステップで、特定の使用シーンはありません。

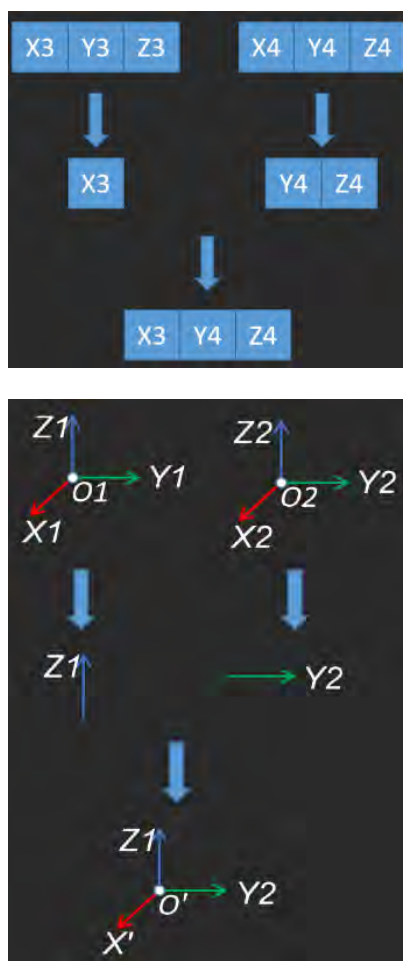
入力と出力



4.3.88. 各入力位置姿勢を一部抽出して新しい位置姿勢を合成

機能

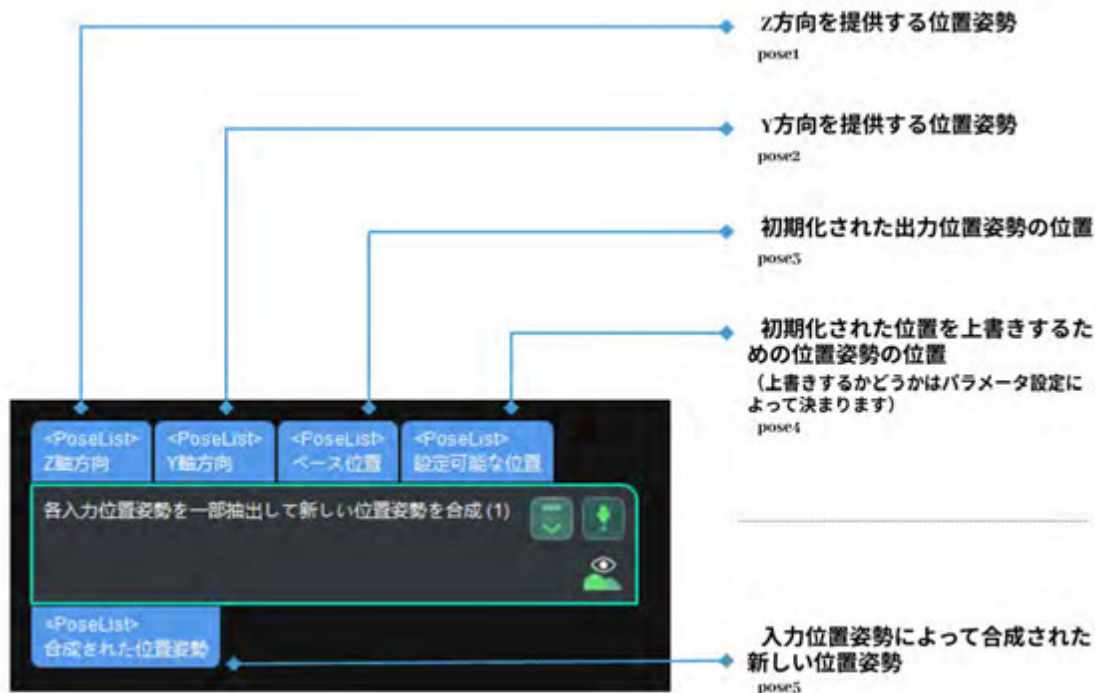
入力位置姿勢の指定部分を新しい位置姿勢に合成します。



使用シーン

汎用の位置姿勢計算ステップで、特定の使用シーンはありません。

入力と出力



パラメータの説明

可視化設定

位置姿勢表示の設定

初期値：出力位置姿勢のみを表示。

値のリスト：入力位置姿勢のみを表示、出力位置姿勢のみを表示、入力と出力位置姿勢をすべて表示。

調整説明：処理後の位置姿勢、処理前の位置姿勢、処理前後の全ての位置姿勢をそれぞれ表示します。

4番目の入力の使用軸

初期値：X

オプション：X、Y、Z、XY、XZ、YZ、XYZ、NONE

調整説明：並進のパラメータを選択します。例えば、Xを選択する場合、対応する操作は、3番目の入力にXを選択し、4番目の入力にYZを選択します。

4.3.89. 四元数と並進ベクトルから位置姿勢を合成



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場

合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

回転ベクトル（四元数）と並進ベクトルを組み合わせる位置姿勢に合成します。

使用シーン

汎用の位置姿勢作成ステップです。特定の使用シーンはありません。

入力と出力

● 入力：

1. このポートに入力される回転ベクトル（四元数）は位置姿勢を合成するために使用されます。
2. このポートに入力される位置姿勢の中心（並進ベクトル）は位置姿勢を合成するために使用されます。

● 出力：

1. 入力された回転ベクトルと位置姿勢の中心で合成された位置姿勢。

4.3.90. 軸と角度から回転ベクトルを合成



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

Vector3D型のベクトルおよび回転角度を入力し、回転ベクトルを合成します。

使用シーン

汎用の回転ベクトルの合成ステップです。特定の使用シーンはありません。

入力と出力

● 入力：

1. このポートに入力される軸は、回転ベクトルを作成するために使用されます。
2. このポートに入力される回転角度は、回転ベクトルを作成するために使用されます。

● 出力：

1. 作成された回転ベクトル（四元数で表示される）。

4.3.91. 両軸から回転ベクトルを合成（右手系）



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場

合は、当社 (docs@mech-mind.net) までお問い合わせください。

機能

入力された2つのベクトルを指定軸とし、右手系によって四元数で表示される回転ベクトルを合成します。入力された2つの軸が垂直でない場合、右手系によって3番目の軸が生成された後、2番目のポートの軸を保持し、再び右手系を使用して2番目のポートの軸と生成された3番目の軸から回転ベクトルを生成します。

使用シーン

汎用のデータ型変換・合成ステップです。特定の使用シーンはありません。

入力と出力

● 入力：

1. このポートに入力される3Dベクトルは、回転の開始軸として使用されます。
2. このポートに入力される3Dベクトルは、回転の終了軸として使用されます。

● 出力：

1. 回転ベクトルを表示する四元数。

4.3.92. 数値から3Dベクトルを合成

機能

3つの数値（X、Y、Z）をVector3D型のベクトルに合成します。同時に複数の（X、Y、Z）を入力して複数のベクトルを合成することができます。

使用シーン

汎用のデータ型の変換ステップです。特定の使用シーンはありません。

入力と出力

● 入力：

1. ベクトルを合成するためのX値。
2. ベクトルを合成するためのY値。
3. ベクトルを合成するためのZ値。

● 出力：

1. 合成されたベクトル。

4.3.93. 制御光源



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

このステップは、Mech-Visionにより、光源制御を実現します。

4.3.94. 2D位置姿勢を3D位置姿勢に変換



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

対象物の2D位置姿勢と基準3D位置姿勢から、新しい3D位置姿勢を変換します。新しい3D位置姿勢のZ軸方向とZ値は基準3D位置姿勢のZ軸方向とZ値と一致し、X軸方向、X値、Y値は入力された2D位置姿勢のX軸方向、X値、Y値によって決まります。

使用シーン

特定のシーン向けのカスタマイズのステップです。

入力と出力

● 入力：

1. このポートに入力される2D位置姿勢は、新しい3D位置姿勢のX、Y値およびX軸方向を変換するために使用されます。
2. このポートに入力される3D位置姿勢のZ軸方向とZ値が、新しい3D位置姿勢のZ軸方向とZ値として使用されます。

● 出力：

1. 変換後の3D位置姿勢。

4.3.95. データ型を変換

機能

入力ポートと出力ポートのデータ型をカスタマイズします。このステップでは、入力データ型を指定された出力データ型に変換できます。

使用シーン

汎用のデータ変換ステップで、特定の使用シーンはありません。

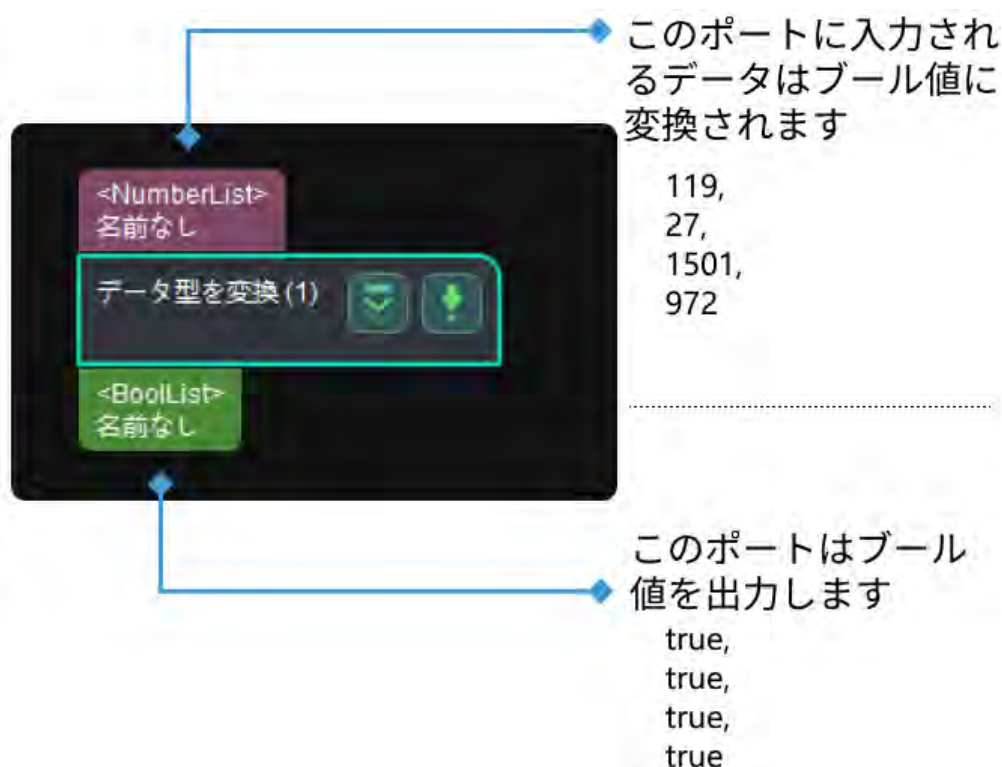
特定の組み合わせがありません。実際のニーズに応じてほかのステップと併用します。

入力と出力

入力のデータ型と出力のデータ型の変換関係を次の表に示します。実際のニーズに応じて選択してください。

入力のデータ型	出力のデータ型
Pose2DList	StringList
BoolList	StringList、String、String
StringList	BoolList、String、NumberList
String	BoolList、StringList、NumberList
NumberList	BoolList、StringList、String、Variant、VariantList
LineSegmentList	StringList
Variant	BoolList、StringList、String、NumberList、VariantList
VariantList	BoolList、StringList、String、NumberList、Variant
Shape2DList	StringList

NumberListデータ型からBoolListデータ型への変換を例として、入力と出力を下图に示します。



パラメータの説明

1. このステップをプロジェクト編集エリアにドラッグすると、初期状態は下図のようになります。

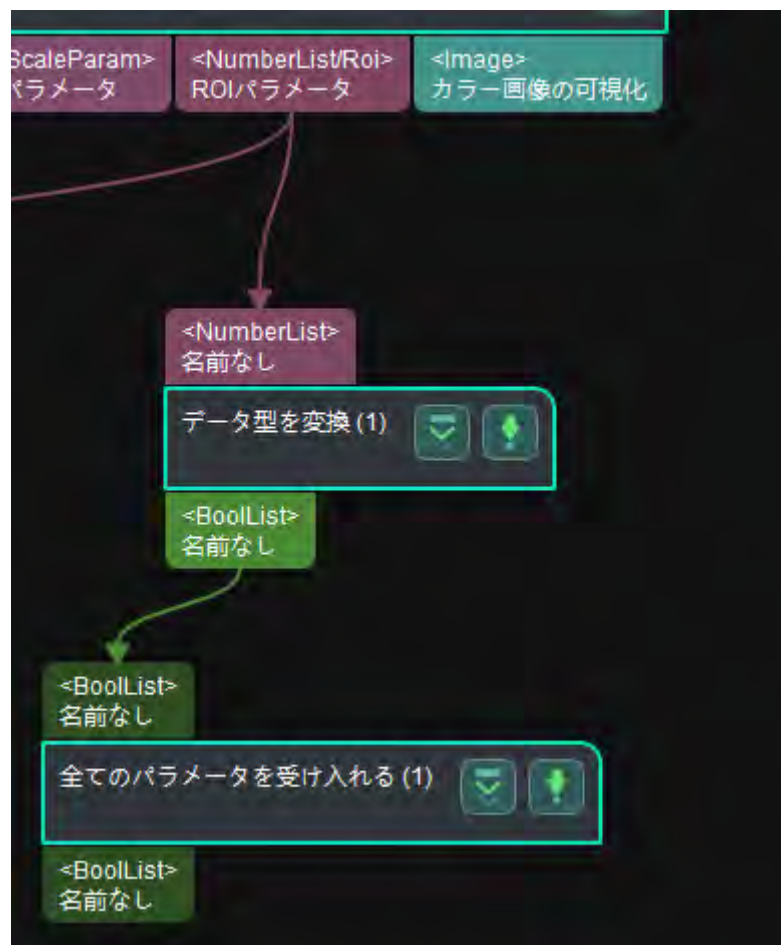


2. パラメータバーで入力の変データ型と出力の変データ型を選択します。対応するポートは選択によって異なります。



3. ステップをつなぎます。

ステップのつながぎを下図に示します。





このステップを使用するときは、入力ポートと出力ポート間の依存関係に注意してください。

- 一方のポートのデータ型が変更されたときに、他方のポートと対応しない変換関係がある場合はリセットされます。
- このステップがつながれている場合、ポートのデータ型が変更されると、ポートのつながりがリセットされます。つまり、ポートのつながりが切断されます。

4.3.96. ピクセルを物理的な長さに変換

機能

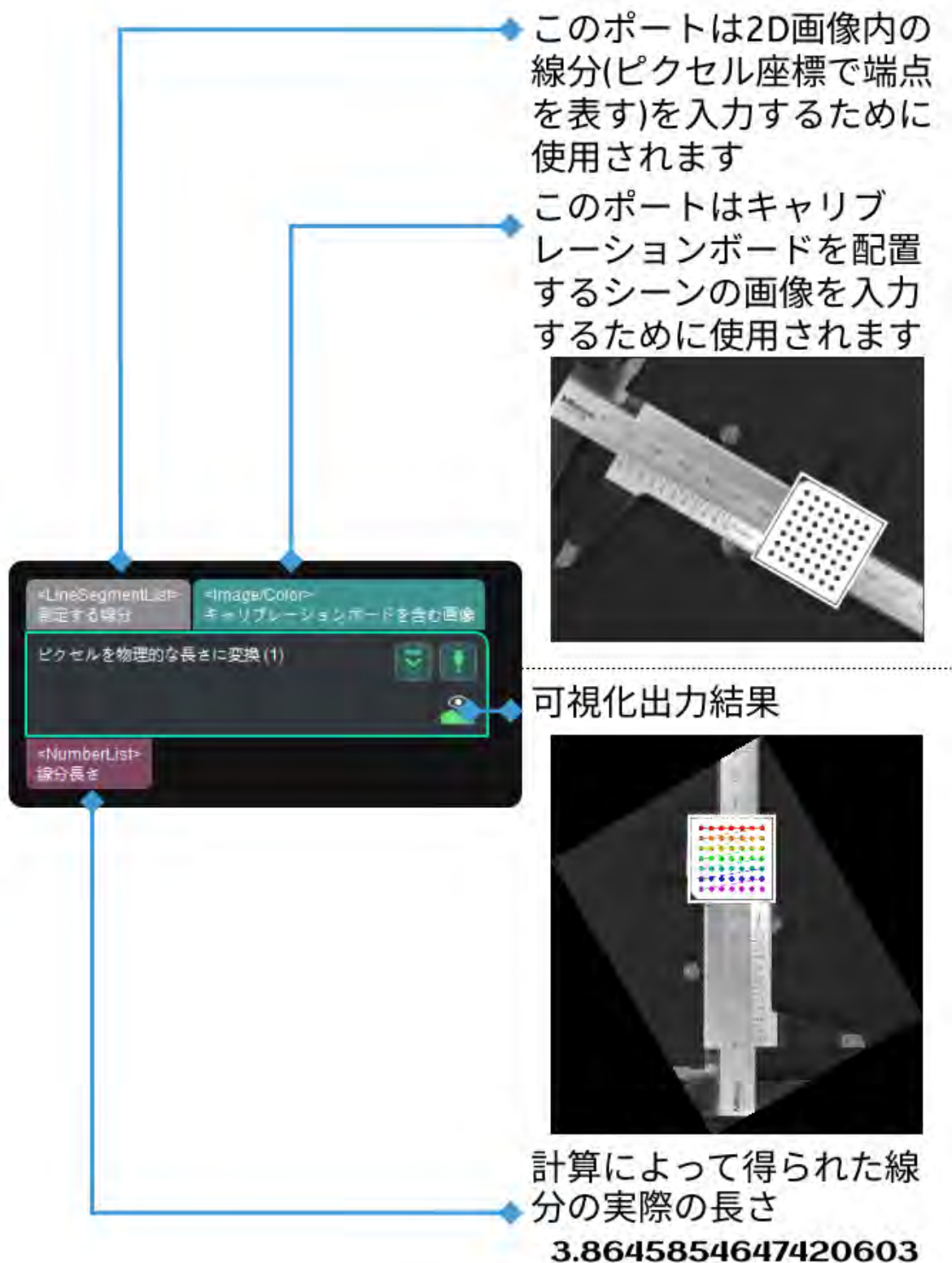
2D画像で指定された線分の実際の長さを計算します。このステップを使用するにはキャリブレーションボードが必要です。また、キャリブレーションボードは測定する線分と同じ平面上にある必要があります。

使用シーン

2D画像から平らな物体のサイズを測定するために使用されます。

測定する線分を入力する必要があるため、このステップは通常、[\[vision-steps:measure-distances-segments-to-segments:::measure-distances-segments-to-segments\]](#)、[点から円までの距離を測定](#)、[円から線分までの距離を測定](#)などのLineSegmentList出力ポートを持つ測定クラスのステップの後に使用されます。

入力と出力



パラメータの説明

変換方法

パラメータ説明：このパラメータは、ピクセルを物理的な長さに変換する方法を設定するため

に使用されます。

オプション：変換率、ホモグラフィー。

- 変換率：線分の実際の長さは、ピクセル単位の線分の長さに変換率を掛けた値に等しくなります。
- ホモグラフィー：線分の実際の長さは、画像平面を実際の平面にマッピングすることによって計算されます。

初期値：変換率。

変換率

変換方法を **変換率** に設定する場合、線分の実際の長さはピクセル単位の線分の長さに変換率を掛けた値に等しくなります。

変換率

パラメータ説明：ピクセル距離から実際の距離への変換スケールです。

初期値：1.0000

ホモグラフィー

変換方法を **ホモグラフィー** に設定する場合、画像平面から実際の平面へのマッピングを通じて線分の実際の長さを計算します。

このステップは、2つのホモグラフィー方法で画像平面から実際の平面へのマッピングを計算して線分の実際の長さを計算することができます。

- **キャリブレーションを使用しない方法。** パラメータ **キャリブレーションボードの白い円の中心距離**、**キャリブレーションボードの白い円の行数**、**キャリブレーションボードの白い円の列数** で画像平面から実際の平面へのマッピングを計算して線分の実際の長さを計算します。この方法で計算処理にかかる時間は短くて済みます。
- **キャリブレーションの方法。** パラメータ **キャリブレーションボードの白い円の中心距離**、**キャリブレーションボードの白い円の行数**、**キャリブレーションボードの白い円の列数** で確認したキャリブレーションボードを使用してキャリブレーションを行い、カメラの内部パラメータと外部パラメータを取得し、画像平面から実際の平面へのマッピングを計算して、パラメータ **キャリブレーションボードの平面高さ**、**測定する平面距離** と組み合わせて線分の実際の長さを計算します。この方法の結果は、キャリブレーションされていない方法よりも正確です。

ホモグラフィー に関連するパラメータおよび説明は以下の通りです。

キャリブレーションボードの白い円の中心距離

初期値：1

調整説明：隣接する2つのキャリブレーションボードの白い円の中心間の実際の距離を設定するために使用されます。キャリブレーションボードの白い円のグリッドの水平方向と垂直方向の隣接する円の中心間の距離は等しくなければなりません。実際の状況に応じて設定してください。

キャリブレーションボードの白い円の行数

初期値：7

調整説明：このパラメータは、キャリブレーションボード上の白い円の行数を設定するために使用されます。実際の状況に応じて設定してください。

キャリブレーションボードの白い円の列数

初期値：7

調整説明：このパラメータ、キャリブレーションボード上の白い円の列数を設定するために使用されます。実際の状況に応じて設定してください。

キャリブレーションによる計算

初期値：チェックを入れない。

調整説明：このパラメータは、キャリブレーションによる実際の長さを計算するかどうかを設定するために使用されます。チェックを入れると、このステップに入力される画像を使用して自動的にキャリブレーションが行われ、キャリブレーションで得られたカメラパラメータに基づいて長さが計算されます。

画像番号

パラメータ説明：複数枚の画像を入力する場合、このパラメータを通じてどの画像で線分の実際の長さを計算するかを選択することができます。

初期値：1

調整説明：画像番号を選択した後、「デバッグ結果出力」ウィンドウでこの番号が対応する可視化出力結果を表示できます。

キャリブレーションボードの平面高さ

パラメータ説明：キャリブレーションボードの平面と基準面との距離です。基準平面は、通常、キャリブレーションボードと測定する対象物が置かれている面を指します。

初期値：0

調整説明：キャリブレーションボードをテーブルの上に置き、ボードの厚さが約2cmの場合、キャリブレーションボードの平面高さは2cmとなります。

測定する平面距離

パラメータ説明：キャリブレーションボードの平面と基準面との距離です。基準平面は、通常、キャリブレーションボードと測定する対象物が置かれている面を指します。

初期値：0

調整説明：測定する対象物をテーブルの上に置き、測定する対象物の平面がテーブルから約2cmの場合、測定する平面距離は2cmとなります。

4.3.97. 2D点を3D点に変換



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

2D画像のピクセルを3D点群に変換します。

使用シーン

このステップは通常、2D形状のマッチングに使用され、マッチングのために2Dエッジを3D点群に変換します。

入力と出力

- **入力：**
 1. このポートに入力される画像は点群に変換されます。
 2. このポートに入力される位置姿勢は、画像を点群に変換するために使用されます。
- **出力：**
 1. 入力される2D画像および位置姿勢の変換によって得られた3D点群。

4.3.98. 円を2D位置姿勢に変換

機能

円形情報のデータを2D位置姿勢データに変換します。

使用シーン

汎用のデータ変換ステップで、特定の使用シーンはありません。

入力と出力

- **入力：** このポートに入力される円形情報の形式のデータ項目（radius、theta、X、Y）は、2D位置姿勢のデータ項目（theta、X、Y）に変換されます。円のthetaは通常0です。
- **出力：** 入力データから変換された2D位置姿勢のデータ項目。theta、XおよびYの値を保持し、radius値を破棄します。

4.3.99. 画像の色空間を変換

機能

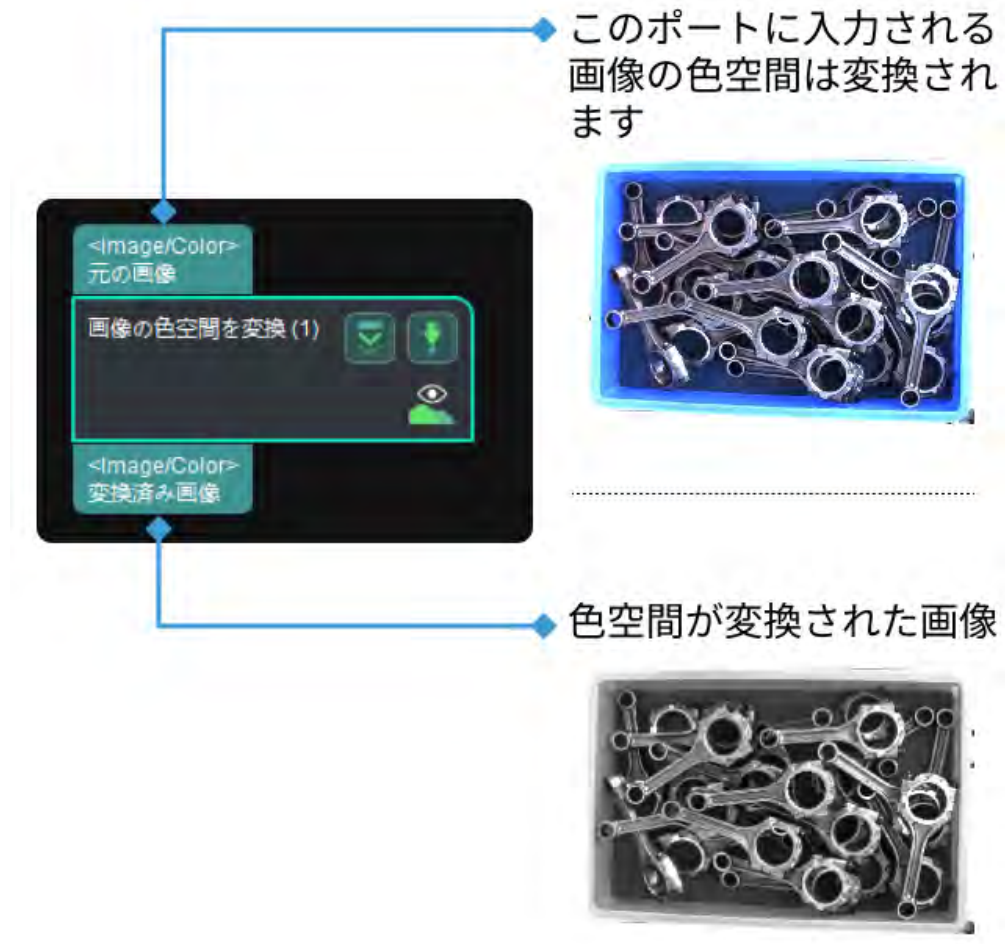
RGBカラー画像からグレースケール画像への変換、BGRカラー画像からグレースケール画像への変換など、画像の色空間を変換します。



使用シーン

汎用の2D画像の色空間の変換ステップです。特定の使用シーンはありません。

入力と出力



パラメータの調整説明

変換タイプ

BGR画像からグレースケール画像へ

調整説明：BGR画像をグレースケール画像に変換します。

変換効果を下図に示します。左側のはBGR画像で、右側のは変換後のグレースケール画像です。



RGB画像からグレースケール画像へ

調整説明：RGB画像をグレースケール画像に変換します。選択後に変換方法を設定する必要

があります。通常の場合は「色彩心理学の公式」を選択できます。実際のニーズに応じて選択してください。変換方法の詳細な説明は次の通りです。

変換方法：

- 色彩心理学の公式：公式 $\text{Gray} = 0.299 \cdot R + 0.587 \cdot G + 0.114 \cdot B$ を使用して、RGBの3チャンネルの加重平均を取得します。これは、人間の目の色感に最適です。
- 平均値：R、G、Bの3チャンネルの平均値を計算して出力グレースケール画像のチャンネル値として使用します。変換公式は $\text{Gray} = (R + G + B) / 3$ です。
- 最大値：グレースケール画像の各ピクセルに対して、R、G、Bの3つのチャンネルのピクセルの中で最大値を取得します。
- 最小値：グレースケール画像の各ピクセルに対して、R、G、Bの3つのチャンネルのピクセルの中で最小値を取得します。
- Rチャンネル値：Rチャンネル値を出力グレースケール画像の値として使用し、他のチャンネルの情報を無視します。
- Gチャンネル値：Gチャンネル値を出力グレースケール画像の値として使用し、他のチャンネルの情報を無視します。
- Bチャンネル値：Bチャンネル値を出力グレースケール画像の値として使用し、他のチャンネルの情報を無視します。
- カスタマイズの重み：R、G、Bチャンネルの重みをそれぞれカスタマイズします。

調整の例：変換方法をそれぞれ「Rチャンネル値」、「Gチャンネル値」、「Bチャンネル値」に設定した場合、変換効果を下図に示します。下図では、左からRチャンネル値、Gチャンネル値、Bチャンネル値の変換結果に対応します。



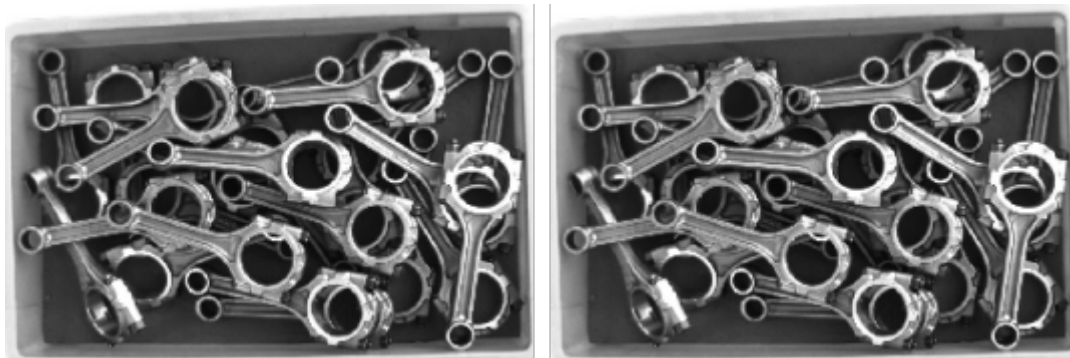
グレースケール画像からBGR画像へ

調整説明：グレースケール画像をBGR画像に変換します。

変換効果を下図に示します。左側のはグレースケール画像で、右側のは変換後のBGR画像です。



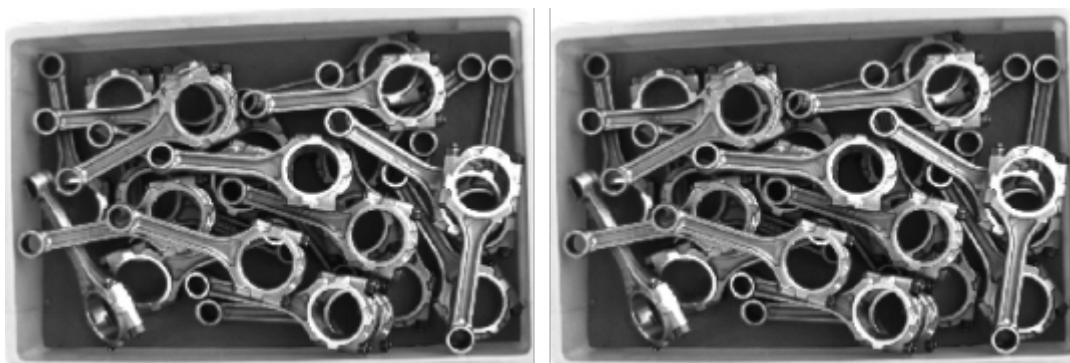
グレースケール画像をBGR画像に変換する過程で、B、G、Rチャンネルにそれぞれグレースケール値のみが割り当てられるため、変換後のカラー画像のBGRの3つのチャンネルの値は同じです。それにより、変換後のBGR画像は白黒ですが、実際に出力されるのはカラー画像です。



グレースケール画像からRGB画像へ

調整説明：グレースケール画像をRGB画像に変換します。

変換効果を下図に示します。左側のはグレースケール画像で、右側のは変換後のRGB画像です。



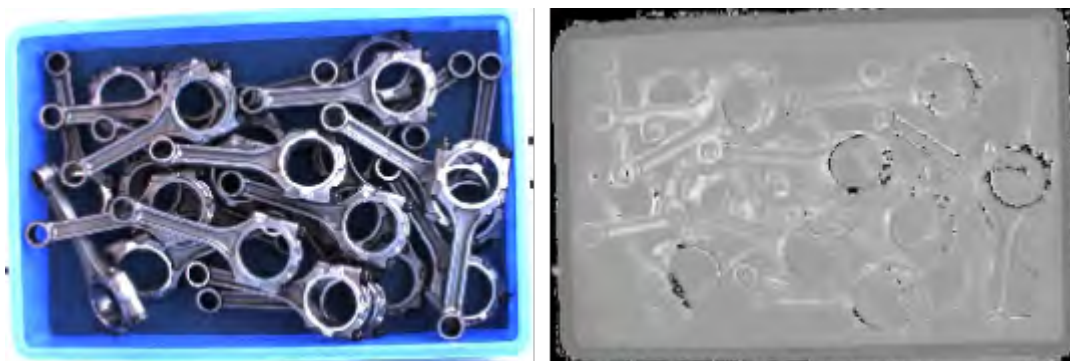
ヒートマップからBGR画像へ

調整説明：ヒートマップをBGR画像に変換します。

RGB画像からHSIへ

調整説明：RGB画像をHSIに変換します。選択後に出力チャンネルのタイプを選択する必要があります。

調整の例：「出力チャンネルのタイプ」を「最初のチャンネル」に設定した場合、変換効果を下図に示します。左側のはRGB画像で、右側のは変換後のHSIです。



RGB画像からHSVへ

調整説明：RGB画像をHSVに変換します。選択後に出力チャンネルのタイプを選択する必要があります。

調整の例：「出力チャンネルのタイプ」を「最初のチャンネル」に設定した場合、変換効果を下図に示します。左側のはRGB画像で、右側のは変換後のHSVです。



RGB画像からYUVへ

調整説明：RGB画像をYUVに変換します。選択後に出力チャンネルのタイプを選択する必要があります。

調整の例：「出力チャンネルのタイプ」を「最初のチャンネル」に設定した場合、変換効果を下図に示します。左側のはRGB画像で、右側のは変換後のYUVです。



4.3.100. 画像をコピー



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

読み取った画像の名前を変更し、指定したパスに保存します。

使用シーン

特定のシーン向けのカスタマイズのステップです。

入力と出力

- 入力：なし。
- 出力：コピー過程のステータスを表示するための文字列。

4.3.101. 箱を数える



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

最高層の箱の数を計算し、パレット高さおよびパレットパターンによってパレットに配置されている箱の合計数を計算します。

使用シーン

特定のシーン向けのカスタマイズのステップで、デパレタイジングのプロジェクトで箱の数をチェックするために使用されます。

入力と出力

- 入力：
 1. ロボット座標系における認識された箱の位置姿勢。
 2. 最高層の箱のマスク。
 3. 箱のマスクのサイズ（ピクセル単位）。
 4. 最高層の認識された箱のマスク。
- 出力：
 1. なし。

4.3.102. データリスト内の指定された次元の要素の数をカウント



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

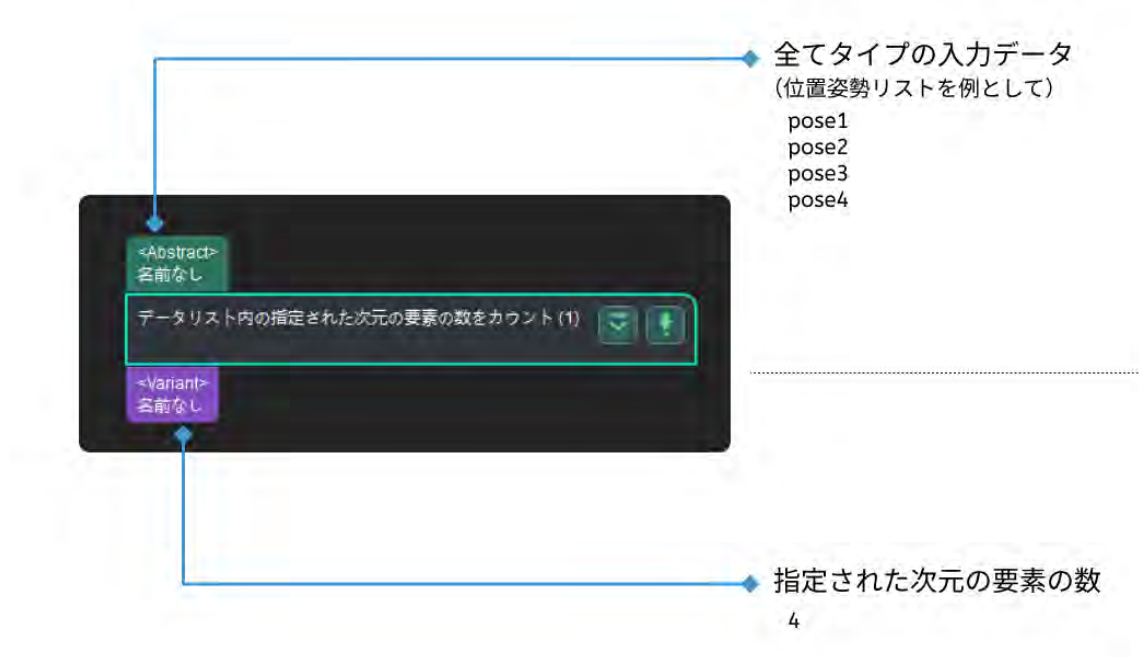
機能

入力データが指定された次元内にある要素の数をカウントします。

使用シーン

汎用の要素数を統計するステップです。これをステップ「データを繰り返して連結」と併用することでその後のステップの各入力ポートのデータ長さを等しくすることができます。

入力と出力



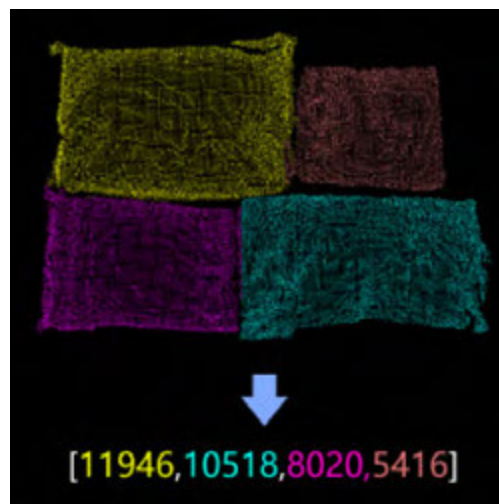
4.3.103. 3D点の数を計算



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社 (docs@mech-mind.net) までお問い合わせください。

機能

入力点群の点を数えます。

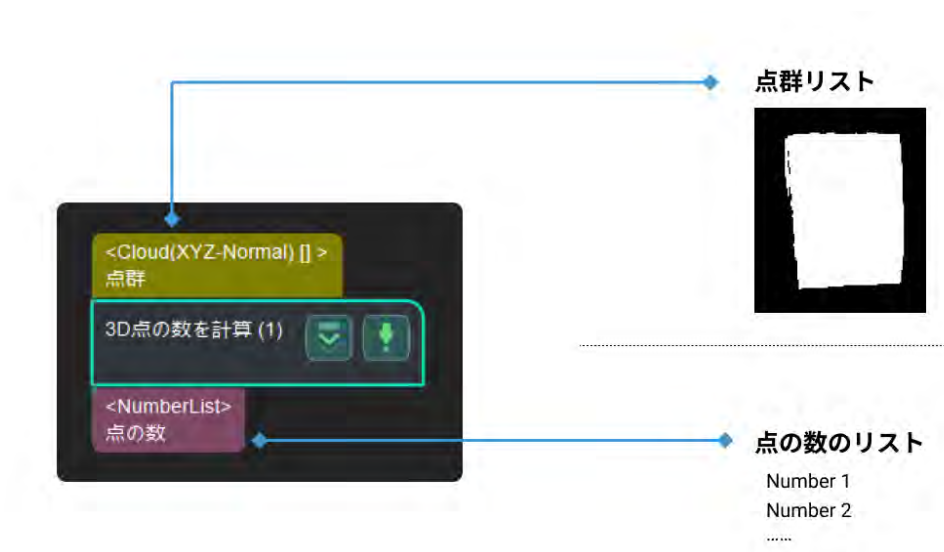


使用シーン

複数の点群をソートまたは分類するシーンに使用されます。通常、「ソート」クラス、または [\[vision-steps:dichotomize-values-by-threshold:::dichotomize-values-by-threshold\]](#) などのス

テップと併用されます。

入力と出力



4.3.104. 色情報をカウント

機能

カラー画像を入力し、色空間を選択して指定したチャンネル内のピクセル値（平均値、標準偏差、最大値、最小値を含む）を統計します。

使用シーン

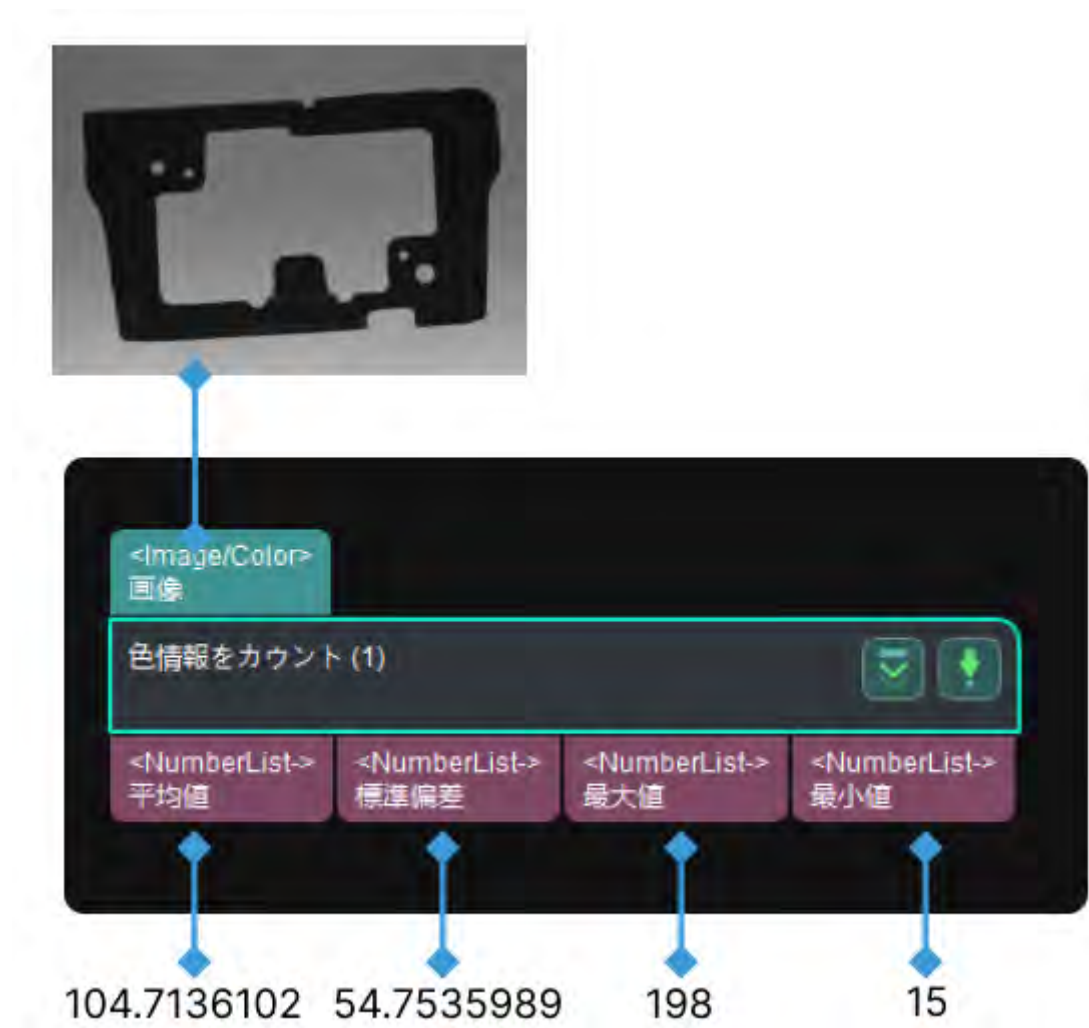
通常、計測シーンで画像の色を評価するために使用されます。

入力と出力

入力画像のRGB色空間における最初チャンネルの色情報をカウントした場合のステップの出力を下図に示します。

- RGB色空間の最初のチャンネルに含まれる全ピクセルの平均値。
- RGB色空間の最初のチャンネルに含まれる全ピクセルの標準偏差。
- RGB色空間の最初のチャンネルに含まれる全ピクセルの最大値。
- RGB色空間の最初のチャンネルに含まれる全ピクセルの最小値。

このステップの入出力効果を下図に示します。



パラメータの説明

色空間

パラメータ説明：このパラメータは、入力画像の色空間を選択するために使用されます。

オプション：RGB、HSV、HSI、Gray

- RGB：一般的に使用される色空間です。この色空間は、赤、緑、青の3色を基づいており、さまざまな度合いで重ね合わせて豊かな色を生成します。一般に3元色モードと呼ばれています。
- HSV：人間の目の視覚特性に合わせた色空間です。この色空間は、色度、彩度、輝度を表現するために使用することができます。
- HSI：人間の色の記述や解釈に準拠した色空間です。この色空間は、色相、彩度、強度を表現するために使用することができます。
- Gray：最も基本的な色空間の1つです。画像に輝度情報だけで色情報がない場合、グレー色空間が使われることが多いです。

初期値：RGB

チャンネル

パラメータ説明：このパラメータは、画像色空間のどのチャンネルで色情報をカウントするかを選択するために使用されます。

オプション：最初チャンネル、2番目のチャンネル、3番目のチャンネル。

初期値：最初チャンネル。

調整説明：チャンネル順序は、各色空間のチャンネルのアルファベット順です。例えば、RGB色空間が赤チャンネル（R）、緑チャンネル（G）、青チャンネル（B）で構成されている場合、最初、2番目、3番目チャンネルはそれぞれ赤チャンネル（R）、緑チャンネル（G）、青チャンネル（B）です。



色空間が **Gray** の場合、チャンネルは1つだけです。そのため、**チャンネル** パラメータを設定する必要はありません。

4.3.105. 対象物のモデルを作成



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

円柱の端面（またはリング）形状の点群を生成します。それらの寸法はステップパラメータで設定できます。

使用シーン

指定した簡単形状の点群を自動的に生成するために使用され、その後のモデルとして使用されます。

入力と出力

- **入力**：なし。
- **出力**：作成された対象物の点群モデル。

4.3.106. 画像を切り抜く



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

ステップパラメータでROIの位置とサイズを設定し、切り抜かれた画像とROIが対応するマスクを出力します。

汎用の2D画像の切り抜きステップです。特定の使用シーンはありません。

入力と出力

- 入力：

1. このポートに入力される画像には、2D ROIが切り抜かれます。

- 出力：

1. 2D ROI内の画像。
2. 2D ROIのマスク。

4.3.107. 対象物の寸法を分解



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

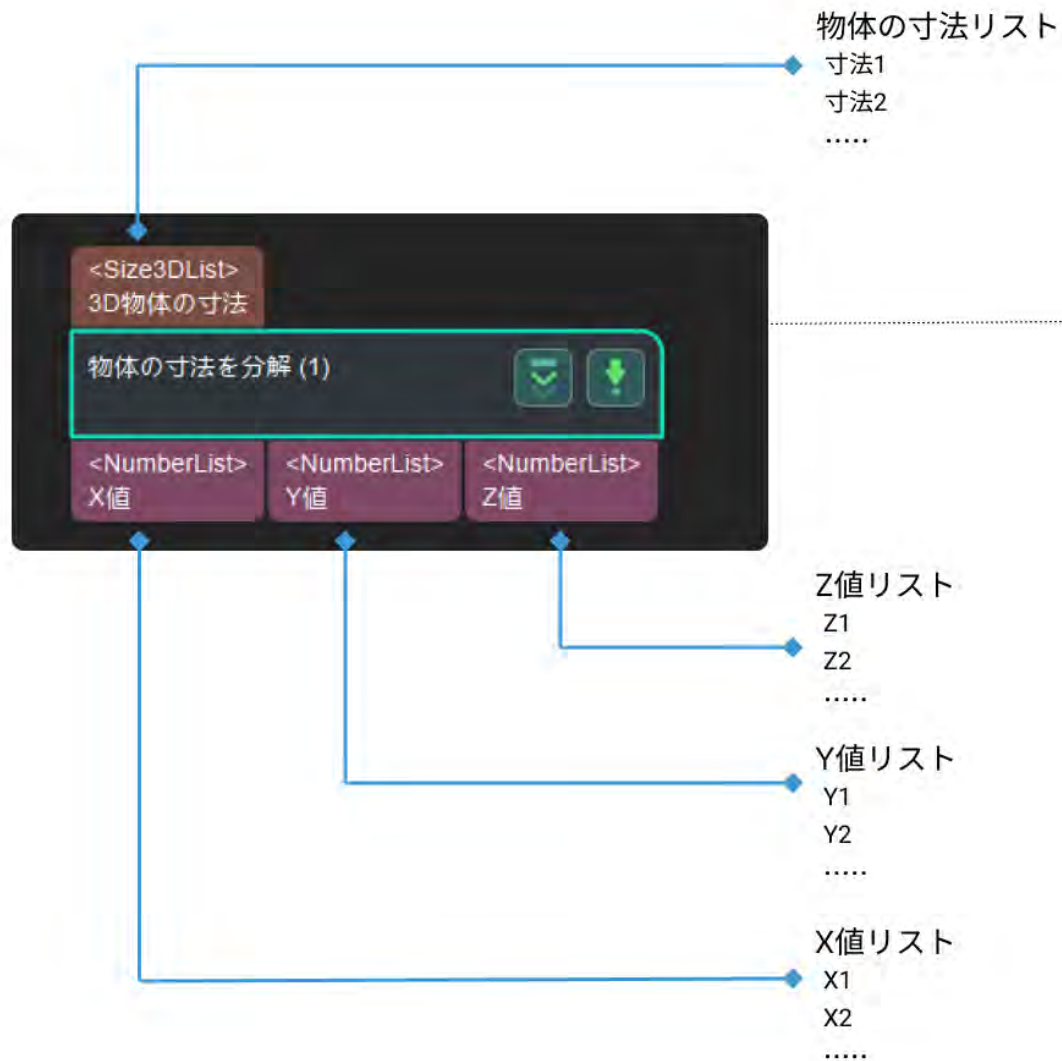
機能

リストにある物体の長さ、幅、高さ（X、Y、Z）を三つの数値リストに分解します。

使用シーン

物体の長さ、幅、高さを抽出した後それぞれに対して後処理を実行するために使用されます。

入力と出力



4.3.108. 位置姿勢を四元数と並進ベクトルに分解



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

位置姿勢を、四元数と並進ベクトルに分解します。

使用シーン

汎用の位置姿勢分解ステップです。その後の **回転ベクトル** と **位置姿勢中心** のデータタイプを必要とするステップにデータソースを提供します。

入力と出力



4.3.109. 回転ベクトルをX-Y-Z軸に分解

機能

四元数形式の回転ベクトルを、X軸ベクトル、Y軸ベクトル、Z軸ベクトルの3つのベクトルに分解します。

使用シーン

汎用の位置姿勢分解ステップです。分解された部分は、他のステップの入力として使用できます。

入力と出力

● 入力：

1. このポートに入力される四元数が対応するX、Y、Z軸のベクトルは分解されます。

● 出力：

1. 入力される四元数のX軸のベクトル。
2. 入力される四元数のY軸のベクトル。
3. 入力される四元数のZ軸のベクトル。

4.3.110. 3Dベクトルを数値に分解

機能

Vector3D型のベクトルを、数値リスト型の3つの数値に分解します。

使用シーン

汎用のデータ型の変換・分解ステップです。特定の使用シーンはありません。

入力と出力

- **入力：**
 1. このポートに入力されたVector3D型のベクトルは、分解されます。
- **出力：**
 1. 入力ベクトルのX値。
 2. 入力ベクトルのY値。
 3. 入力ベクトルのZ値。

4.3.111. ディープラーニングモデルを推論（Mech-DLK 2.1.0/2.0.0）



このステップはもうメンテナンスされていませんので、以降のリリースで削除される予定です。新しいステップ [ディープラーニングモデルパッケージを推論](#) を使用してください。

機能

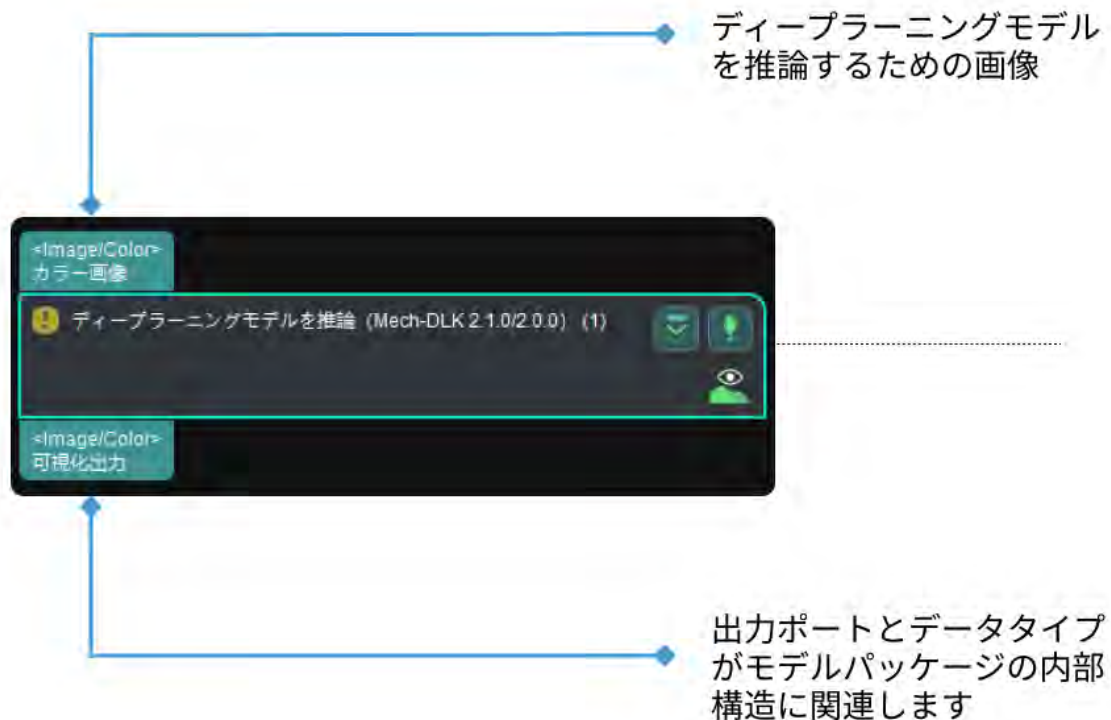
Mech-DLKにエクスポートされたモデルパッケージを使用して推論し、結果を出力します。

このステップは、**開発者モード** でのみ使用できます。

使用シーン

分類、対象物検出、欠陥検出などのシーンに使用されます。プロジェクト実行現場から取得した画像に対し、ディープラーニングソフトウェアにエクスポートされたモデルパッケージを使用して推論します。

入力と出力



4.3.112. ディープラーニングモデルパッケージを推論



Mech-Vision 1.7.2バージョン以降、「ディープラーニングモデルパッケージを推論（CPU）」と「ディープラーニングモデルパッケージを推論（Mech-DLK2.2.0+）」ステップは、「ディープラーニングモデルパッケージを推論」に統合されます。「ディープラーニングモデルパッケージを推論」ステップは、.dlkpackCモデルと.dlkpackモデルの両方をサポートします。

Mech-Vision 1.7.2バージョンで過去バージョンのプロジェクトを開くと、「ディープラーニングモデルパッケージを推論（CPU）」と「ディープラーニングモデルパッケージを推論（Mech-DLK 2.2.0+）」ステップは自動的に「ディープラーニングモデルパッケージを推論」に置き換わります。

機能

このステップはMech-DLKによってエクスポートされた単体モデルまたは直列モデルに対して推論を行い、推論結果を出力することができます。Mech-DLK 2.2.0以降バージョンによってエクスポートされたモデルパッケージのみ対応しています。



Mech-DLK 2.4.1以降、モデルパッケージには、単体モデルと直列モデルの2種類があります。

- 単体モデルパッケージ：1つのディープラーニングアルゴリズムモジュールのみを搭載したモデルパッケージのことを指します（例：「インスタンスセグメンテーション」モデル）。
- 直列モデルパッケージ：ディープラーニングアルゴリズムモジュールの複数のモデルを、前のモデルの出力が次のモデルの入力となる直列形式で搭載したモデルパッケー

ジのことを指します。例えば、モデルパッケージに「対象物検出」と「インスタンスセグメンテーション」の2つのモデルがある場合、モデルの推論順序は**対象物検出・インスタンスセグメンテーション**となり、「対象物検出」の出力は「インスタンスセグメンテーション」の出力になります。「対象物検出」モデルの出力は、「インスタンスセグメンテーション」モデルの入力として使用されます。

このステップが直列モデルの推論を行う場合、直列モデルの推論結果は **ディープラーニング結果を解析** ステップで解析することができます。

使用シーン

このステップは通常、画像分類、対象物検出、欠陥セグメンテーションなどのシーンに使用されます。このステップの互換性については、**ディープラーニングステップの互換性に関する説明** をご参照ください。

入力と出力

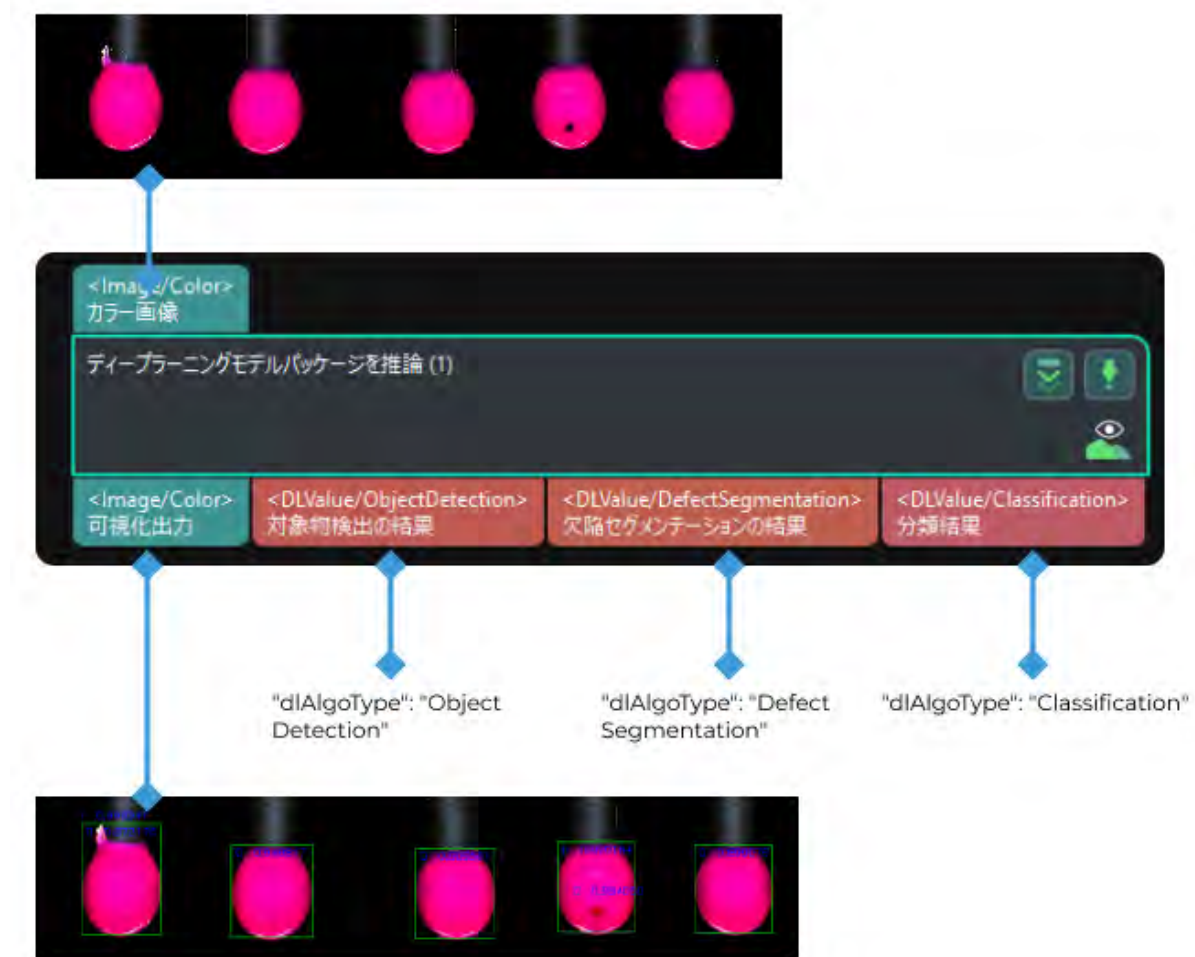
対象物検出

単体モデルパッケージをインポートする場合、このステップの入力と出力は、「対象物検出」を例にすると、下図のようになります。



対象物検出+欠陥セグメンテーション+画像分類

直列モデルをインポートする場合、このステップの入力と出力は、「対象物検出+欠陥セグメンテーション+画像分類」を例にすると、下図のようになります。



パラメータ説明



このステップで直列モデルの推論を行う場合、その後の「ディープラーニング結果を解析」ステップでパラメータを調整することができます。

共通パラメータ

モデルパッケージの設定

モデルパッケージ管理ツールを開く

パラメータ説明：ディープラーニングモデルパッケージ管理ツールを開き、ディープラーニングモデルパッケージをインポートするために使用されます。モデルパッケージファイルとは、Mech-DLKによってエクスポートされた「.dlkpack」または「.dlkpackC」を指します。

調整説明：ディープラーニングモデルパッケージ管理ツールの使用方法については、[ディープラーニングモデルパッケージ管理ツール](#)をご参照ください。

モデルパッケージ名

パラメータ説明：このパラメータは、ディープラーニングモデルパッケージをインポートした後、インポートしたモデルパッケージを選択するために使用されます。

調整説明：モデルパッケージ管理ツールを使用してディープラーニングモデルをインポートした後、ここでドロップダウンリストから対応するモデルパッケージ名を選択します。

モデルパッケージのタイプ

パラメータ説明： **モデルパッケージ名** を選択すると、 **モデルパッケージのタイプ** が自動的に記入されます。

デバイスID

パラメータ説明：このパラメータは、モデルパッケージ推論に使用するGPUのデバイスIDを設定するために使用されます。

調整説明：モデルパッケージ名を選択した後、ここでドロップダウンリストからモデルパッケージ推論に使用するGPUのデバイスIDを選択する必要があります。

ROI設定

ROIファイル

パラメータ説明：このパラメータは、ROIの設定や変更のために使用されます。

調整説明：

1. 初期状態では、デフォルトのROI設定が既にあります。ROI設定を変更するには、[**ROI ファイルを設定**]をクリックします。
2. 次に、表示された画面で **ROI 設定** を行います。



- ROI設定後、下図に示すように、ROIファイル名を設定する必要があります。ROI名が設定されていない場合は、ポップアッププロンプトが表示されます。



- ROI設定後、初期のROIを使用するには、「ROIファイルを設定」ボタンの下のROIファイル名を削除します。

可視化設定



本ステップを使用して欠陥セグメンテーションを行う場合は、このパラメータは使用できません。

カスタマイズのフォントサイズを使用

パラメータ説明：このパラメータは、可視化出力結果のフォントサイズをカスタマイズするかどうかを設定するために使用されます。チェックを入れた後、**フォントサイズ (0-10)**を設定する必要があります。

初期値：チェックを入れない。

調整アドバイス：実際の状況に応じて設定してください。

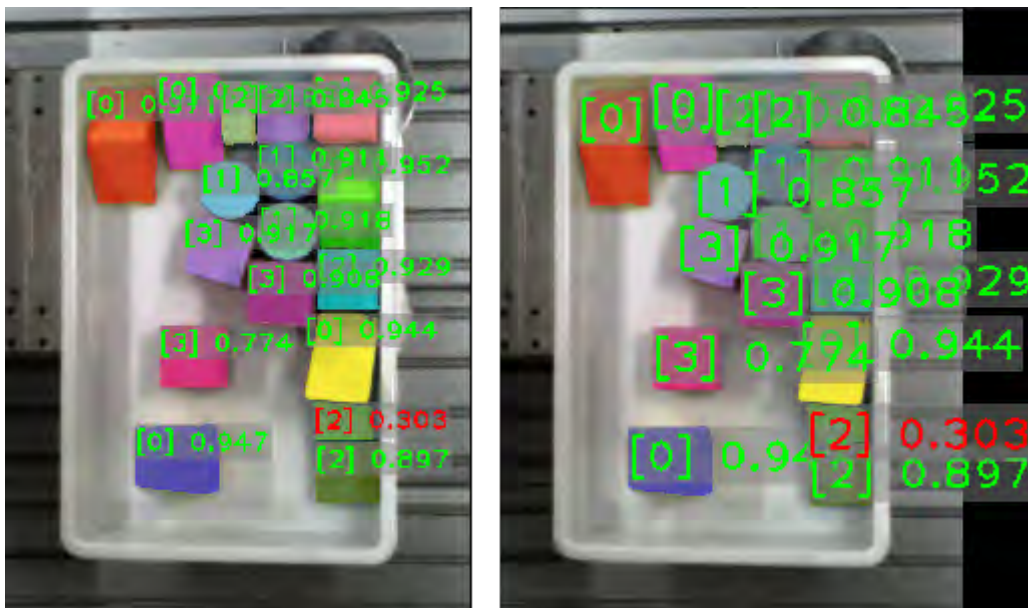
フォントサイズ (0-10)

パラメータ説明：このパラメータは、可視化出力結果のフォントサイズを設定するために使用されます。

初期値：3.0

調整アドバイス：実際の状況に応じて設定してください。

調整の例：インスタンスセグメンテーションを例として説明します。この値をそれぞれ3.0と5.0に設定すると、可視化出力結果のフォントサイズは下図のようになります。左側のフォントサイズは3で、右側のフォントサイズは5です。



すべての結果を表示

パラメータ説明：このパラメータは、直列モデルのすべての推論結果を可視化するために使用されます。このパラメータは、「ディープラーニングモデルパッケージを推論」を使用して直列モデルの推論を行う場合にのみ設定可能です。

調整アドバイス：実際の状況に応じて設定してください。



本ステップを使用して欠陥セグメンテーションを行う場合は、このパラメータは表示されません。

ディープラーニングモデルパッケージの固定パラメータ

画像分類

画像分類の信頼度しきい値 (0.0-1.0)

パラメータ説明：このパラメータは、画像分類に使用する信頼度しきい値を設定するために使用されます。信頼しきい値より大きい結果は緑色で表示され、信頼しきい値より小さい結果は赤色で表示されます。

初期値：0.7000

調整アドバイス：実際の状況に応じて設定してください。

CAMを表示

パラメータ説明：画像分類を行う場合、このパラメータは画像のどの部分のピクセルが分類結果に大きく寄与しているかを確認するために使用されます。青は貢献度が小さく、赤は貢献度が大きいことを意味します。

調整説明：このパラメータを設定するには、「すべてのパラメータを表示」を設定する必要があります。



Mech-Vision1.7.2バージョンでは、**CAMを表示** 機能を有効にすると、モデルパッケージの推論が遅くなります。

インスタンスセグメンテーション

可視化設定

画像で検出対象物を絵画

調整説明：このパラメータは、インスタンスセグメンテーションのマスクと境界ボックスを画像に表示するかどうかを設定するために使用されます。

初期値：チェックを入れない。

調整アドバイス：実際の状況に応じて設定してください。

対象物の可視化方式

パラメータ説明：このパラメータは、可視化出力結果の対象物の可視化方式を選択するために使用されます。

初期値：Instances

オプション：Threshold、Instances、Classes、CentralPoint

対象物の可視化方式	説明	サンプル図
Threshold	信頼度によります。計算されたしきい値がしきい値より大きい場合、対応するインスタンスは緑色に表示されます。それ以外の場合は赤色で表示されます。	
Instances	各インスタンスは固有の色を持っています。	
Classes	同じラベルを持つインスタンスは同じ色になります。	
CentralPoint	対象物自体の色を表示します。	

インスタンスセグメンテーションの信頼度しきい値 (0.0-1.0)

パラメータ説明：このパラメータは、インスタンスセグメンテーションに使用する信頼度しきい値を設定するために使用されます。信頼しきい値より大きい結果は緑色で表示され、信頼し

きい値より小さい結果は赤色で表示されます。

初期値：0.7000

調整アドバイス：実際の状況に応じて設定してください。

対象物検出

可視化設定

画像で検出結果を描画

パラメータ説明：このパラメータは、画像にマスクと境界ボックスを表示するかどうかを設定するために使用されます。

初期値：チェックを入れない。

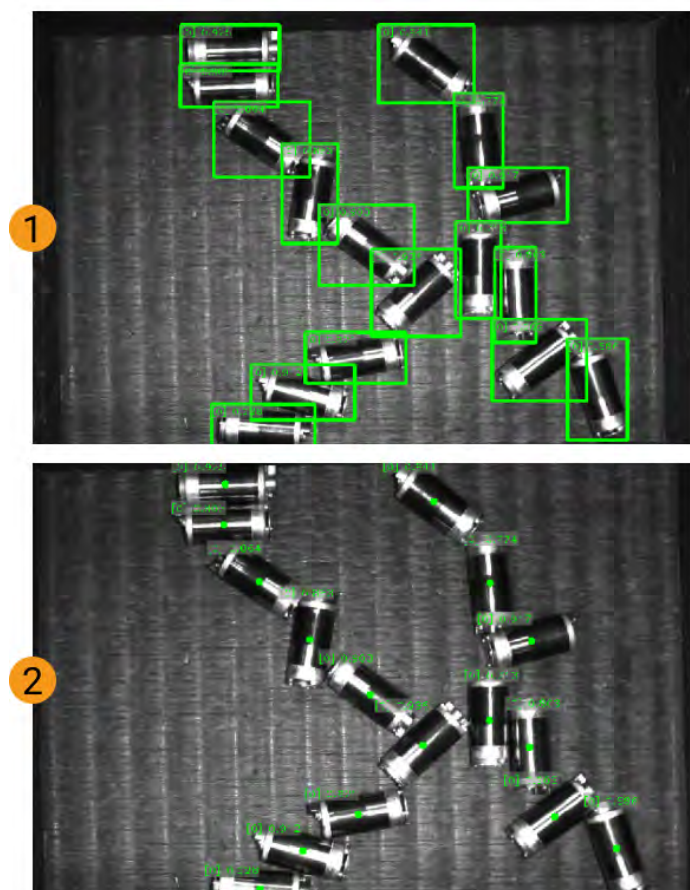
調整アドバイス：実際の状況に応じて設定してください。

結果の可視化方式

初期値：CentralPoint

オプション：BoundingBox、CentralPoint

- BoundingBox：対象物の境界ボックスで結果を表示します。下図の1に示します。
- CentralPoint：対象物の中心点で結果を表示します。下図の2に示します。



調整アドバイス：実際の状況に応じて設定してください。

対象物検出の信頼度しきい値（0.0-1.0）

初期値：0.7000

調整説明：このしきい値を超える結果は保持されます。

欠陥セグメンテーション

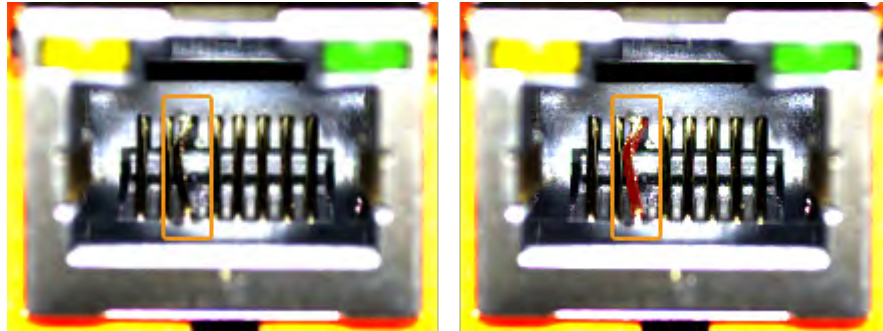
可視化設定

画像で欠陥マスクを描画

パラメータ説明：このパラメータは、画像に欠陥のマスクを描画するかどうかを設定するために使用されます。チェックを入れた後、入力画像で欠陥マスクを描画します。

初期値：チェックを入れない。

調整の例：下図に示すように、左側のはチェックを入れる前の効果で、右側はチェックを入れた後の効果です。



- Mech-Vision 1.7.2バージョン以降、「ディープラーニングモデルパッケージを推論」ステップでは、欠陥判定ルール、欠陥数のしきい値、欠陥面積範囲パラメータが表示されなくなりました。必要に応じてMech-DLKで設定することができます。
- Mech-Vision 1.7.2バージョンでは、Mech-DLK 2.2.0バージョン以前でエクスポートされたモデルパッケージに欠陥判定ルールを設定した状態で、「ディープラーニングモデルパッケージを推論」ステップを使用すると、欠陥判定ルールが無効となり、Mech-DLK 2.4.1バージョン以降のソフトウェアでモデルパッケージの欠陥判定ルールを再設定してエクスポートし、「ディープラーニングモデルパッケージを推論」ステップでモデルパッケージの推論を行う必要があります。

4.3.113. ディープラーニング結果を解析

機能

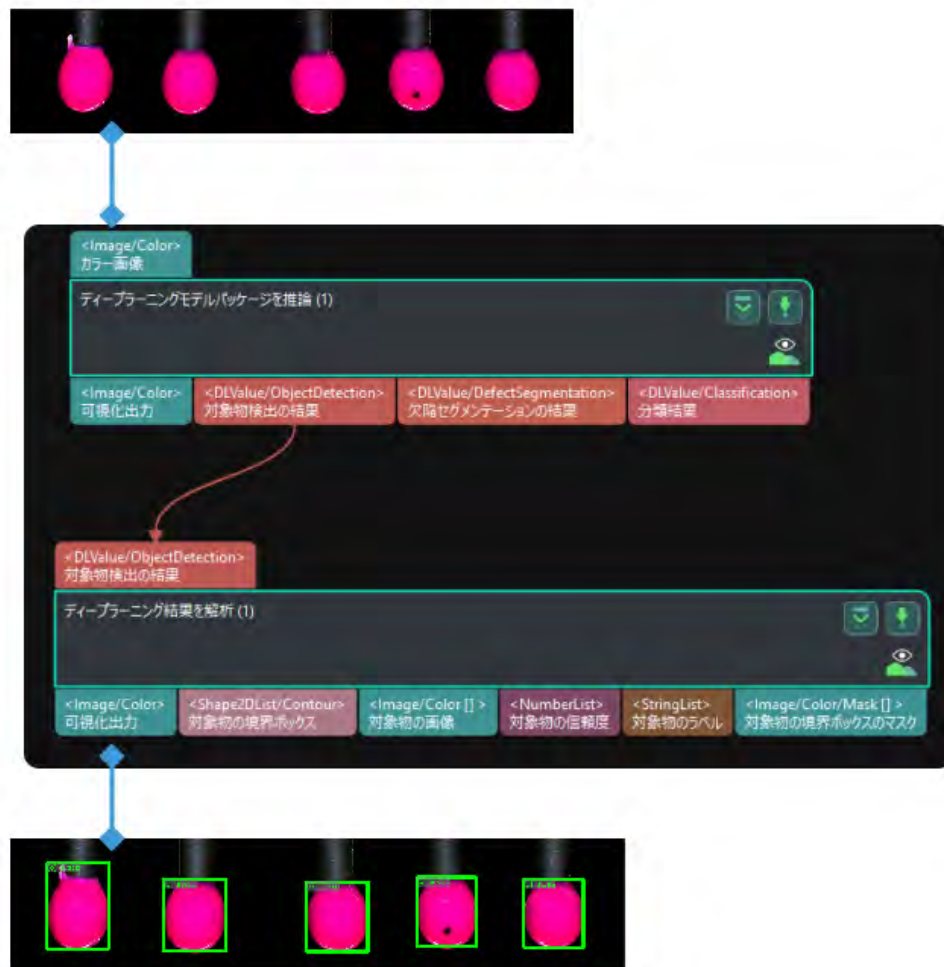
ディープラーニングモデルパッケージを推論 ステップから出力された直列モデルの推論結果を解析することができます。

使用シーン

「ディープラーニングモデルパッケージを推論」ステップを使用して直列モデルを推論する場合、このステップを「ディープラーニングモデルパッケージを推論」の後につなぎます。

複数の画像を同時に入力して推論する場合、各画像の解析結果を確認するには、「ディープラーニングモデルパッケージを推論」と「ディープラーニング結果を解析」の間に **データをアンパック** を追加することを推奨します。

入力と出力

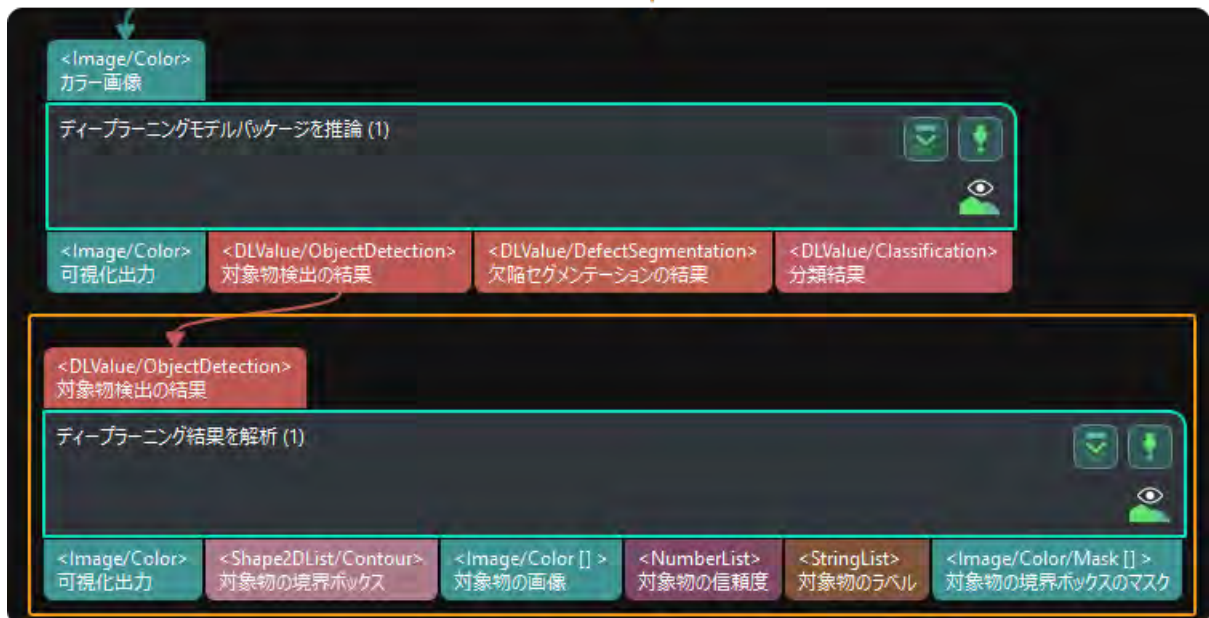
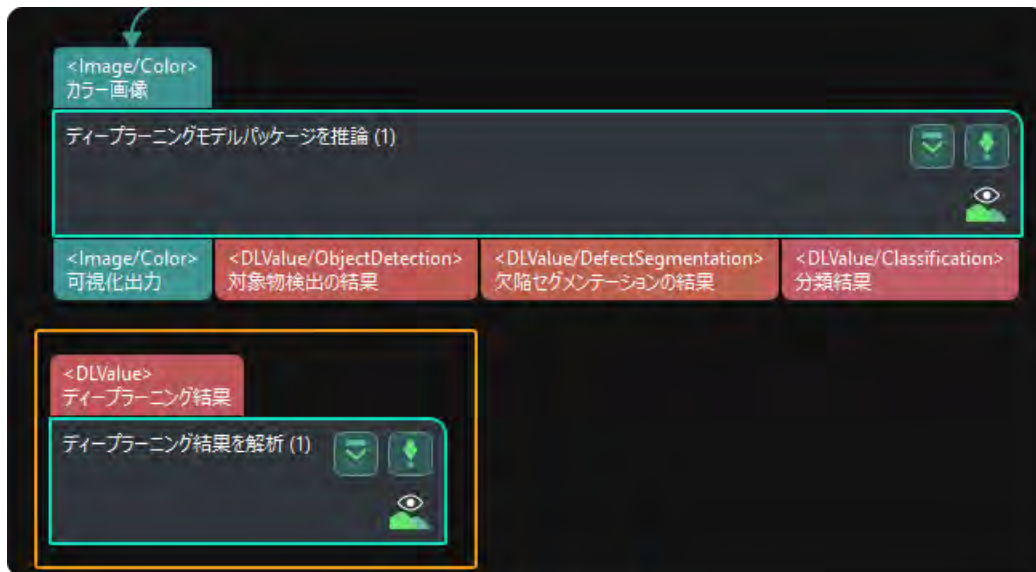


使用方法

「ディープラーニングモデルパッケージを推論」ステップの後につなぐ場合

「ディープラーニングモデルパッケージを推論」ステップの後にこのステップをつなぐと、ディープラーニングデータのタイプに応じて異なるパラメータが表示されます。

例えば、本ステップが「ディープラーニングモデルパッケージを推論」の「対象物検出の結果」出力ポートにつながれている場合、本ステップは対象物検出に関連するパラメータを表示します。パラメータ調整については、[ディープラーニングモデルパッケージを推論](#)をご参照ください。



「データをアンパック」を使用する場合

「ディープラーニングモデルパッケージを推論」と「ディープラーニング結果を解析」の間に「データをアンパック」を追加した場合、下図のようになります。



- 「データをアンパック」の **出力ポートの数** は、入力画像の数と一致させる必要があります。
- データをアンパックした後、サブタイプのデータがないため、「ディープラーニング結果を解析」ステップでモデルに対応するシーンを自動判定できず、「ディープラーニング結果を解析」ステップの出力ポートが生成できない場合があります。この場合、「ディープラーニング結果を解析」ステップの **ディープラーニングデータのタイプ** パラメータを手動で設定する必要があります。



ディープラーニングデータのタイプの選択は変更できませんので、ディープラーニングデータのタイプを変更する場合は、「ディープラーニング結果を解析」ステップを削除して再度追加してください。

4.3.114. スキャンラインに沿ったディープクラスタリング



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社 (docs@mech-mind.net) までお問い合わせください。

機能

ノイズ除去のために、レーザースキャンラインの方向に沿って深度値をクラスター化します。

使用シーン

特定のシーン向けのカスタマイズのステップです。

入力と出力

- **入力：** このポートに入力される深度画像はクラスター化されます。
- **出力：** クラスター化された深度画像。

4.3.115. 深度画像のエンコード



これは古いバージョンのステップであり、メンテナンスが停止されています。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

設定された深度範囲では、深度画像をエンコードし、深度画像をカラー画像に変換します。

使用シーン

特定のシーン向けのカスタマイズのステップです。

入力と出力

- **入力：** このポートに入力される深度画像は、カラー画像に変換されます。
- **出力：** 変換によって得られたカラー画像。

4.3.116. 円を検出して測定



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

検出する円を含む関心領域を手動で設定します。このステップでは、関心領域にある円を検出して円の情報を出力します。手動で選択された関心領域に、円周方向に配置された大まかなエッジ検出フレームが生成され、対象物のエッジポイントはエッジ検出フレームのグレースケール変化によって検出されます。最後に、正確な円線はエッジポイントによってフィッティングされます。上記の円周は、ROIの中心を中心とし、ROIの半分の高さで半分の幅の小さい方を半径とします。

使用シーン

このステップは、計測シーンで画像内の円の中心位置と半径（ピクセル単位）を認識して、その後の物理サイズの計算に備えるために使用されます。

入力と出力

- **入力：**

1. このポートに入力される画像の関心領域内で円が検出されます。
2. このポートに入力される2D変換は、以前設定したROIを適切な位置に変換するために使用されます。

● 出力：

1. 可視化するための画像。
2. 検出された円の中心座標と半径（ピクセル単位）。

4.3.117. 直線を検出して測定



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

検出する線分を含む関心領域を手動で設定します。このステップでは、線分を検出して線分情報を出力します。手動で選択した関心領域に、関心領域の横方向に平行に配置されたエッジ検出フレームが生成され、対象物のエッジポイントはエッジ検出フレームのグレースケール変化によって検出されます。最後に、正確な直線はエッジポイントによってフィッティングされます。

使用シーン

このステップは、計測シーンで画像内の対象物のエッジ線分の位置（ピクセル単位）を認識して、その後の物理サイズの計算に備えるために使用されます。

入力と出力

● 入力：

1. このポートに入力される画像の関心領域内で線分が検出されます。
2. このポートに入力される2D変換は、以前設定したROIを適切な位置に変換するために使用されます。

● 出力：

1. 可視化するための画像。
2. 検出された線分の2つ端点のピクセル座標。
3. 検出された線分の midpoint のピクセル座標。
4. 検出された直線と画像の水平方向との角度。範囲は-180°～180°です。画像の水平方向は右に0°、時計回りが正です。

4.3.118. 長穴を検出して測定

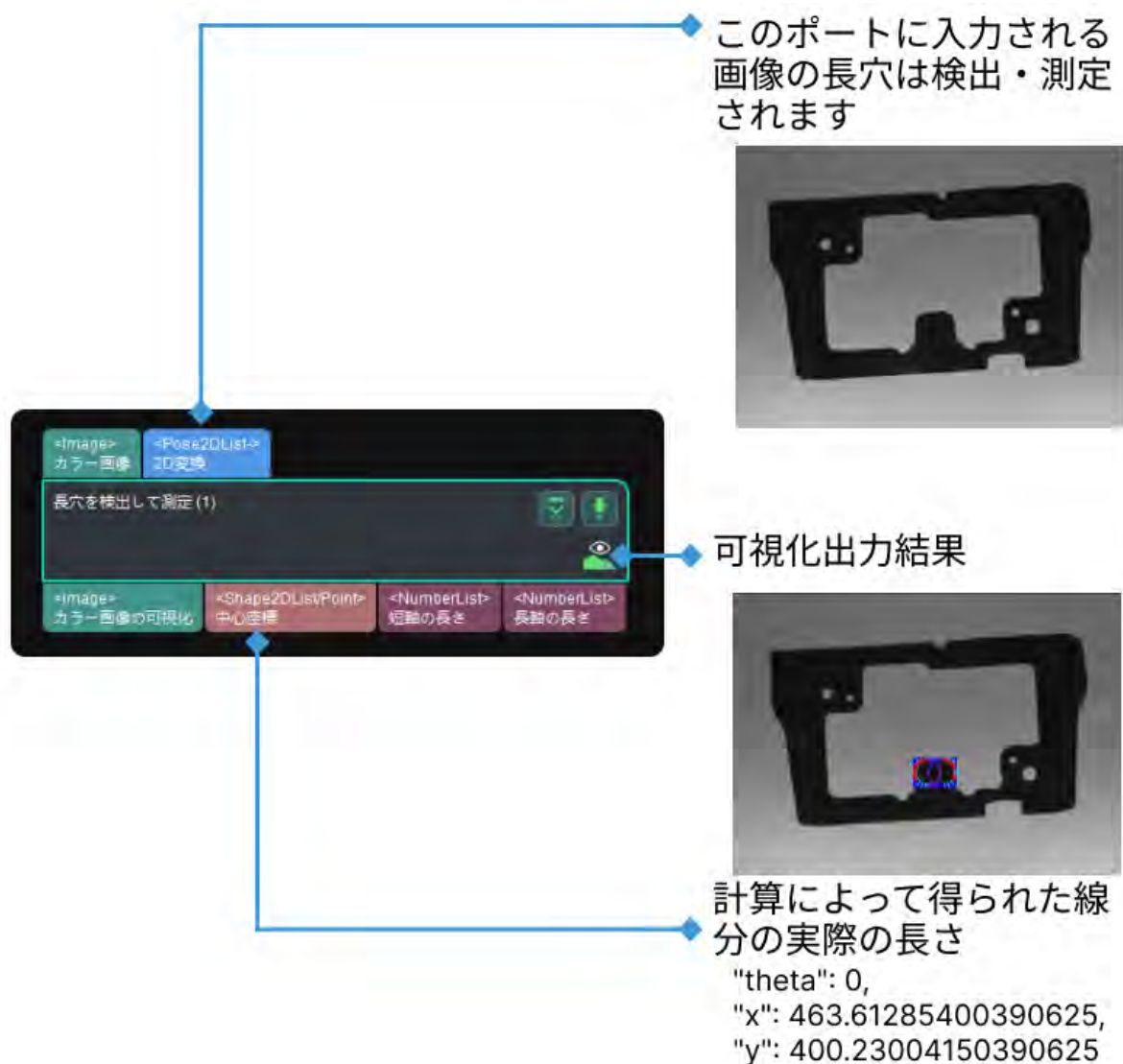
機能

入力画像の長穴を検出して測定します。

使用シーン

このステップは、計測のシーンで画像内の長穴の位置とサイズ（ピクセル単位）を検出して、その後の物理サイズの計算に備えるために使用されます。

入力と出力



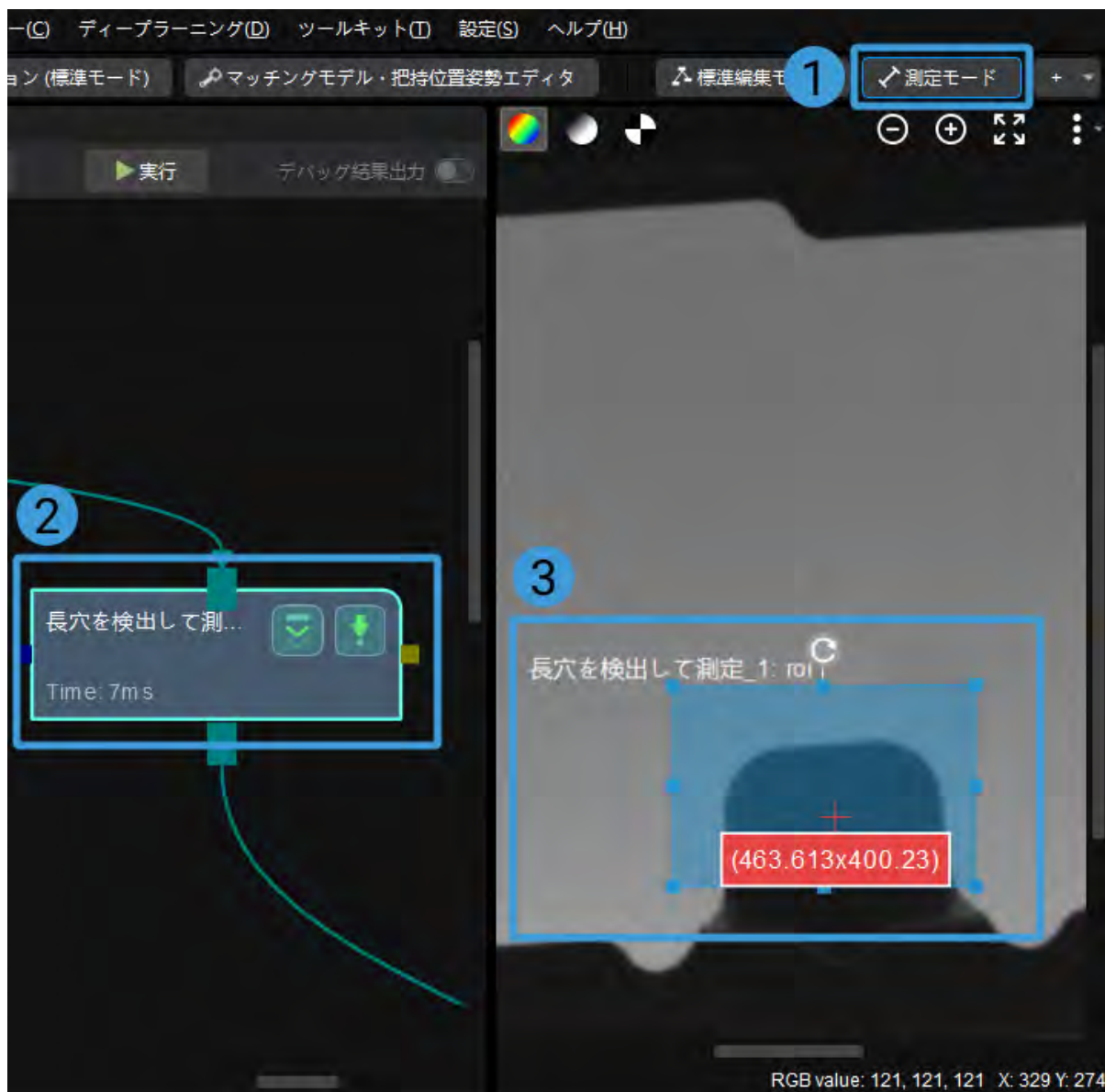
パラメータの説明

ROI

初期値：0

調整説明：最初にツールバーの[測定モード]をクリックして、測定モードに切り替えます。次にこのステップを選択し、検出する領域を選択します。選択枠は画像の左上隅にあり

ます。マウスを選択枠に移動します（移動する前に画像をスケーリングすることをお勧めします）。矢印が触手に変わると、マウスを長押しして選択枠を移動してサイズを設定します。



下図に示すように、選択したら、各パラメータの値がROIパラメータに表示されます。

▼ ROI	[(463.61, 400.23), 126.74 x 92.92, 0.00]
中心X	463.61
中心Y	400.23
幅	126.74
高さ	92.92
角度	0.00

位置と向きの補正

初期値：チェックを入れない。

調整説明：チェックを入れると、入力された2D変換に従って画像内の対象物の位置と向きに合わせるようにROIを変換します。

Cannyエッジ検出の低しきい値

初期値：40

調整説明：勾配が高しきい値よりも大きい場合、それは実際のエッジであると見なされます。勾配が高しきい値と低しきい値の間にあり、勾配が高しきい値よりも大きい実際のエッジに接続されている場合も、実際のエッジと見なされます。勾配が低いしきい値にある場合、それは無視されます。勾配の計算にはSoble導関数演算子が使用され、勾配の最大値は1440です。

Cannyエッジ検出の高しきい値

初期値：100

調整説明：勾配が高しきい値よりも大きい場合、それは実際のエッジであると見なされます。勾配が高しきい値と低しきい値の間にあり、勾配が高しきい値よりも大きい実際のエッジに接続されている場合も、実際のエッジと見なされます。勾配が低いしきい値にある場合、それは無視されます。勾配の計算にはSoble導関数演算子が使用され、勾配の最大値は1440です。

ガウシアンフィルタ係数

初期値：1

調整説明：このパラメータは、エッジ検出フレーム内で垂直方向にフィルタリングするために使用されます。推奨値は1です。

エッジ極性

初期値：両方。

オプション：白から黒へ、黒から白へ、両方。

調整説明：このパラメータは、物体のエッジと見なされるグレースケールの変化を指定するために使用されます。グレースケールの変化は、エッジ検出フレームの上から下へのグレースケールの変化を指します。

- 白から黒へ：フレーム内の白から黒への変化は、物体のエッジポイントと見なされます。
- 黒から白へ：フレーム内の黒から白への変化は、物体のエッジポイントと見なされます。
- 両方：フレーム内の白から黒への変化も黒から白への変化も、物体のエッジポイントと見なされます。

エッジ選択

初期値：最初。

オプション：最初、2番目、最後、すべて。

調整説明：このパラメータは、完全なエッジをフィッティングするために各エッジ検出フレーム内で検出されたエッジポイントを選択します。

最初：エッジ検出フレームの上から下への最初のポイントがフィッティングに使用されます。

- 2番目：エッジ検出フレームの上から下への2番目のポイントがフィッティングに使用されます。
- 最後：エッジ検出フレームの上から下への最後のポイントがフィッティングに使用されます。
- すべて：エッジ検出フレームのすべてのポイントがフィッティングに使用されます。

4.3.119. 箱検出（4側面）



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

箱とその4つ側面の位置姿勢および寸法を計算します。

使用シーン

規則な形状を持つ4つ側面の箱を認識し、認識結果を外部サービスに送信して衝突検出を行うために使用されます。

入力と出力

- 入力：
 1. 箱の上表面の点群。
 2. 箱の上表面のマスク。
 3. カメラから直接取得した箱を含む深度画像。
- 出力：
 1. 箱の3D中心位置姿勢。
 2. 箱の4つ側面の位置姿勢。
 3. 箱および4つ側面の3D寸法。

4.3.120. 箱検出（最大内接長方形）



これは古いバージョンのステップであり、メンテナンスが停止されますので、新しいバージョンのステップ [\[vision-steps:detect-bin-largest-inscribed-rect:::detect-bin-largest-inscribed-rect\]](#) を使用してください。このステップに関する詳細が必要な場合は、当社

(docs@mech-mind.net) までお問い合わせください。

機能

箱の上表面のマスキの最大内接長方形によって、箱全体およびその4つの側面の位置姿勢を計算します。

使用シーン

箱を認識し、認識結果を外部サービスに送信して衝突検出を行うために使用されます。

入力と出力

- 入力：
 1. 箱の上表面のマスキ。
 2. カメラから直接取得した箱を含む深度画像。
- 出力：
 1. 箱全体および4つの側面の位置姿勢。
 2. 箱全体および4つの側面の3D寸法。

4.3.121. 箱検出（最大内接長方形・V2）



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

箱の上表面の点群および位置姿勢を入力し、箱の位置姿勢および寸法を推定します。出力寸法は箱の外部寸法であることに注意してください。

使用シーン

不規則な4つの側面の箱を認識し、認識結果を外部サービスに送信して衝突検出を行うために使用されます。

入力と出力

- 入力：
 1. 箱の上表面の点群。
 2. 箱の上表面の位置姿勢。
- 出力：
 1. 検出された箱の位置姿勢。
 2. 検出された箱の外部輪郭の寸法。

4.3.122. 箱検出（2側面）



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

箱の二つの側面の位置姿勢および衝突モデルの設定を入力し、各側面の衝突モデルの位置姿勢およびサイズを計算します。

使用シーン

箱を認識し、認識結果を外部サービスに送信して衝突検出を行うために使用されます。

入力と出力

- **入力：**
 1. 箱の二つの側面の位置姿勢。
- **出力：**
 1. 外部サービスに送信される衝突モデルの位置姿勢。
 2. 二つの側面の位置姿勢。
 3. 衝突モデルのサイズ。

4.3.123. 円心検出



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

このステップは、画像の円心を検出するために使用されます。

使用シーン

汎用の円心検出ステップです。特定の使用シーンはありません。

入力と出力

- **入力：** このポートに入力される画像内の円の円心は検出されます。
- **出力：** 検出された円の情報（円心および半径を含む）。

4.3.124. コーナー検出



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

このステップにより、入力画像からコーナーを検出して出力できます。

使用シーン

汎用の2D特徴抽出ステップです。特定の使用シーンはありません。

入力と出力

● 入力：

1. このポートに入力される画像のコーナーは検出されます。
2. このポートに入力される画像は、検出されたコーナーを描画するために使用されます。

● 出力：

1. 検出された方向性を持つコーナー。
2. コーナーを持つおよびコーナーによって形成された対象物の輪郭のカラー画像。

4.3.125. エッジ検出



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

このステップは、2D画像の対象物のエッジを検出するために使用されます。

使用シーン

汎用の2D画像のエッジ検出ステップです。特定の使用シーンはありません。

入力と出力

- **入力：** このポートに入力される2D画像内の対象物のエッジが検出されます。
- **出力：** 検出された対象物のエッジマスク。

4.3.126. マーカーを検出



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

このステップにより、シーン内のマーカーの位置姿勢を検出できます。

使用シーン

マーカーを検出します。マーカーによって位置姿勢を推定し、把持位置姿勢を動的に候補します。

入力と出力

- 入力：
 1. シーンの深度画像。
 2. シーンのカラー画像。
 3. シーンの点群。
- 出力：
 1. 検出されたマーカーの位置姿勢。
 2. 検出されたマーカーの識別子。
 3. マーカーのキーポイントの検索範囲を表す点群。

4.3.127. 内接円検出



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

このステップにより、入力の2D画像から輪郭を抽出し、抽出した輪郭で内接円を検出できます。

使用シーン

汎用の2D特徴抽出ステップです。特定の使用シーンはありません。

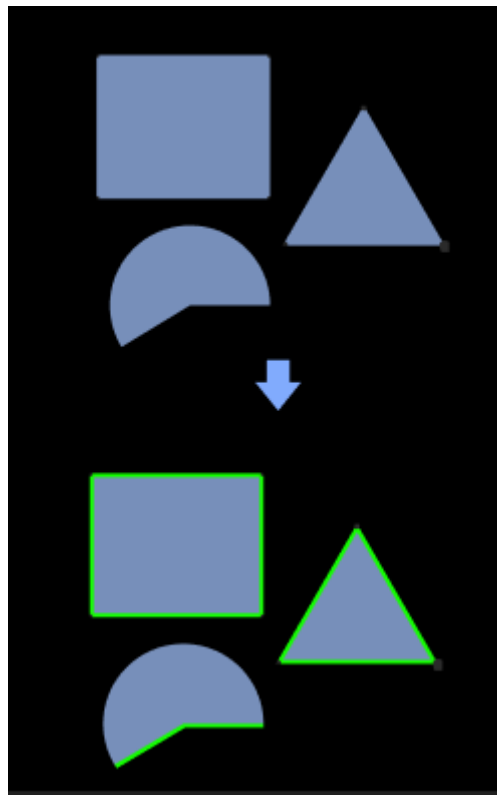
入力と出力

- 入力：
 1. このポートに入力されるマスクは、その内接円が計算されます。
 2. 内接円を可視化するための画像。
- 出力：
 1. 内接円を表示するためのマスク。
 2. 計算された内接円を含む可視化画像。

4.3.128. 線分を検出

機能

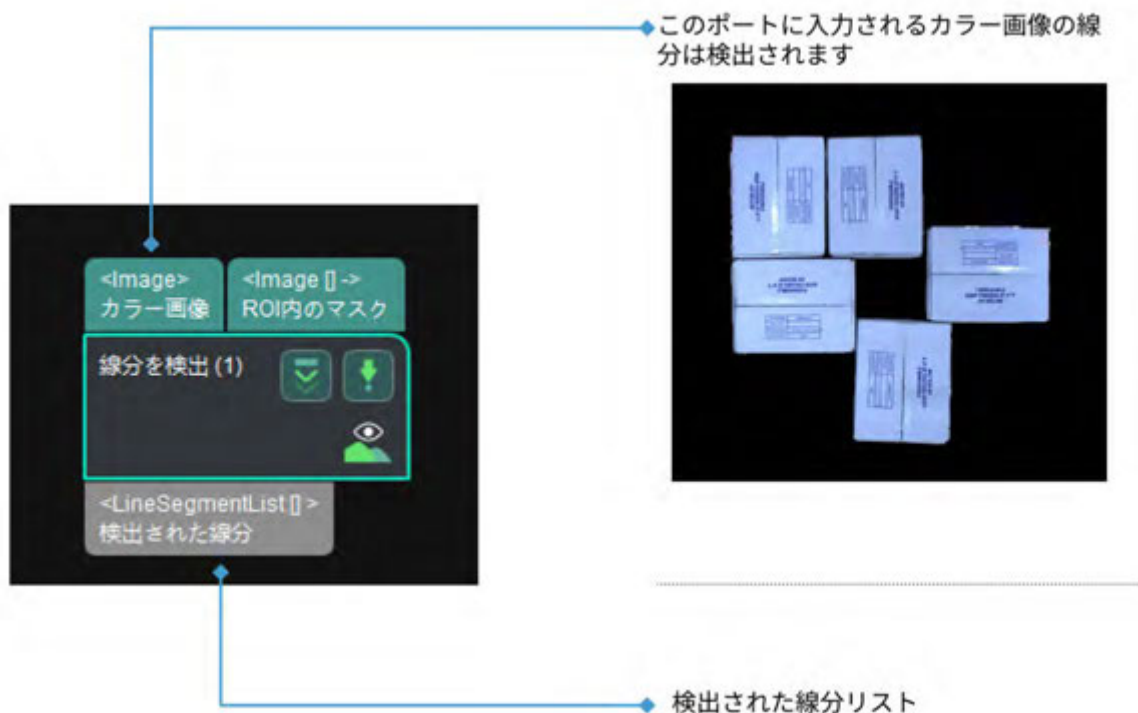
このステップは、画像から線分を検出できます。



使用シーン

その後のマッチングのために、2D画像内の線分特徴を抽出します。ステップ [\[vision-steps:2d-matching:::2d-matching\]](#) と併用できます。

入力と出力



よく使うパラメータの説明

直線検出器の設定

直線検出器タイプ

パラメータ説明：このパラメータは、直線検出器タイプを指定するために使用されます。

オプション：

- LineSegmentDetector
- FastLineDetector
- EdgeDrawingDetector

初期値：LineSegmentDetector

直線プロパティ

線分長さの下限しきい値

パラメータ説明：このパラメータは、検出する線分の最小長さ（ピクセル単位）を設定するために使用されます。検出する線分の長さがこの値より小さい場合、その線分は除去されます。

初期値：10px

推奨値：10px、20px、30px

マスク外の線分を除去

パラメータ説明：このパラメータは、マスク外で検出された線分を除去するかどうかを設定するために使用されます。

初期値：チェックを入れない。

調整アドバイス：実際の状況に応じて設定してください。

4.3.129. マスク内で最大面積の最初のN個の長方形を検出



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

マスク内で最大面積を持つ最初のN個の内接長方形を検出します。Nの値は、ステップパラメータで設定できます。

使用シーン

汎用のマスク計算ステップです。特定の使用シーンはありません。

入力と出力

- **入力：** このポートに入力されるマスクは、最大の内接長方形が検出されます。
- **出力：** マスクから検出された最初のN個の最大内接長方形。

4.3.130. 遮蔽された対象物を検出



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

対象物間の長方形の空き領域を検出し、遮られた対象物の位置姿勢と寸法を計算します。

使用シーン

特定のシーン向けのカスタマイズのステップです。

入力と出力

- **入力：**
 1. 遮られた対象物を含む点群。
 2. 対象物の中心位置姿勢。
- **出力：**

1. 遮られた対象物の中心位置姿勢。
2. 遮られた対象物の3D寸法。
3. 遮られた対象物の2D頂点の画像。

4.3.131. 領域の形状特徴を検出



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

このステップにより、マスクの非ゼロのピクセル領域の外接長方形を検出できます。

使用シーン

汎用のマスクデータ処理ステップです。特定の使用シーンはありません。

入力と出力

- **入力：** このポートに入力されるマスクには、非ゼロのピクセル領域の外接長方形が検出されます。
- **出力：** 検出された外接長方形のマスク。

4.3.132. 頂点を検出



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

検出する頂点を含む関心領域を手動で設定します。このステップでは、1つの対象物の頂点を検出し、頂点情報を出力します。

使用シーン

通常、計測シーンで対象物の頂点を見つけるために使用されます。

入力と出力

- **入力：**
 1. このポートに入力される画像の関心領域にある対象物は、1つの頂点が抽出されます。
 2. このポートに入力される2D変換は、以前設定したROIを適切な位置に変換するために使用されます。
- **出力：**

1. 可視化するための画像。パラメータ設定に従って、エッジポイント、関心領域のフレーム、頂点、およびベースラインを表示できます。
2. 頂点のピクセル座標。

4.3.133. ピクセルサイズを決定



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

このステップにより、入力された円または線分で、ピクセルの実際のサイズを確認できます。

使用シーン

汎用のピクセルサイズ計算ステップです。特定の使用シーンはありません。

入力と出力

- 入力：
 1. 可視化するための背景画像。
 2. ピクセルサイズを計算するための円または線分（ポートのタイプはパラメータで設定できます）。
- 出力：
 1. 線分または円を含む可視化画像。
 2. 計算によって得られたピクセルサイズ。

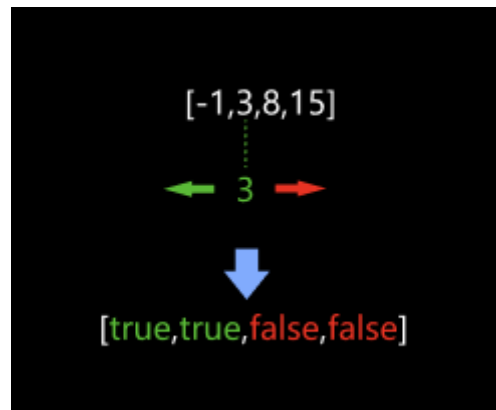
4.3.134. しきい値によって数値を二項分類



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

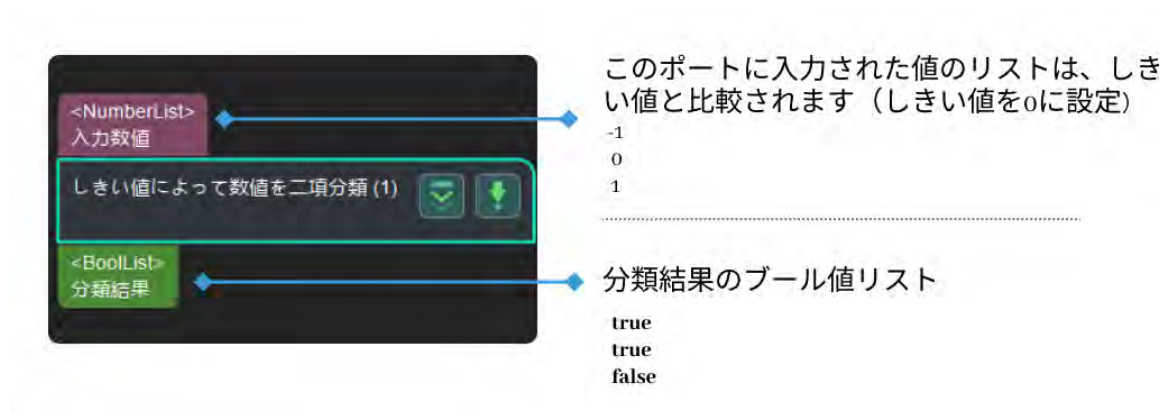
設定したしきい値によって入力リストの各数値の大小を判断します。



使用シーン

リストにある数値が対応するしきい値と等しいまたはそれより小さい場合、出力されたリストはTrueになります。それ以外の場合にはFalseになります。このステップは通常、ステップ [\[vision-steps:filter:::filter\]](#) と併用されます。

入力と出力



4.3.135. 点群を均等に分割



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

点群を均等に分割します。

使用シーン

特定のシーン向けのカスタマイズのステップです。

入力と出力

- **入力：** このポートに入力される点群は、分割されます。

- 出力：分割された点群。

4.3.136. 点群をダウンサンプリング



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

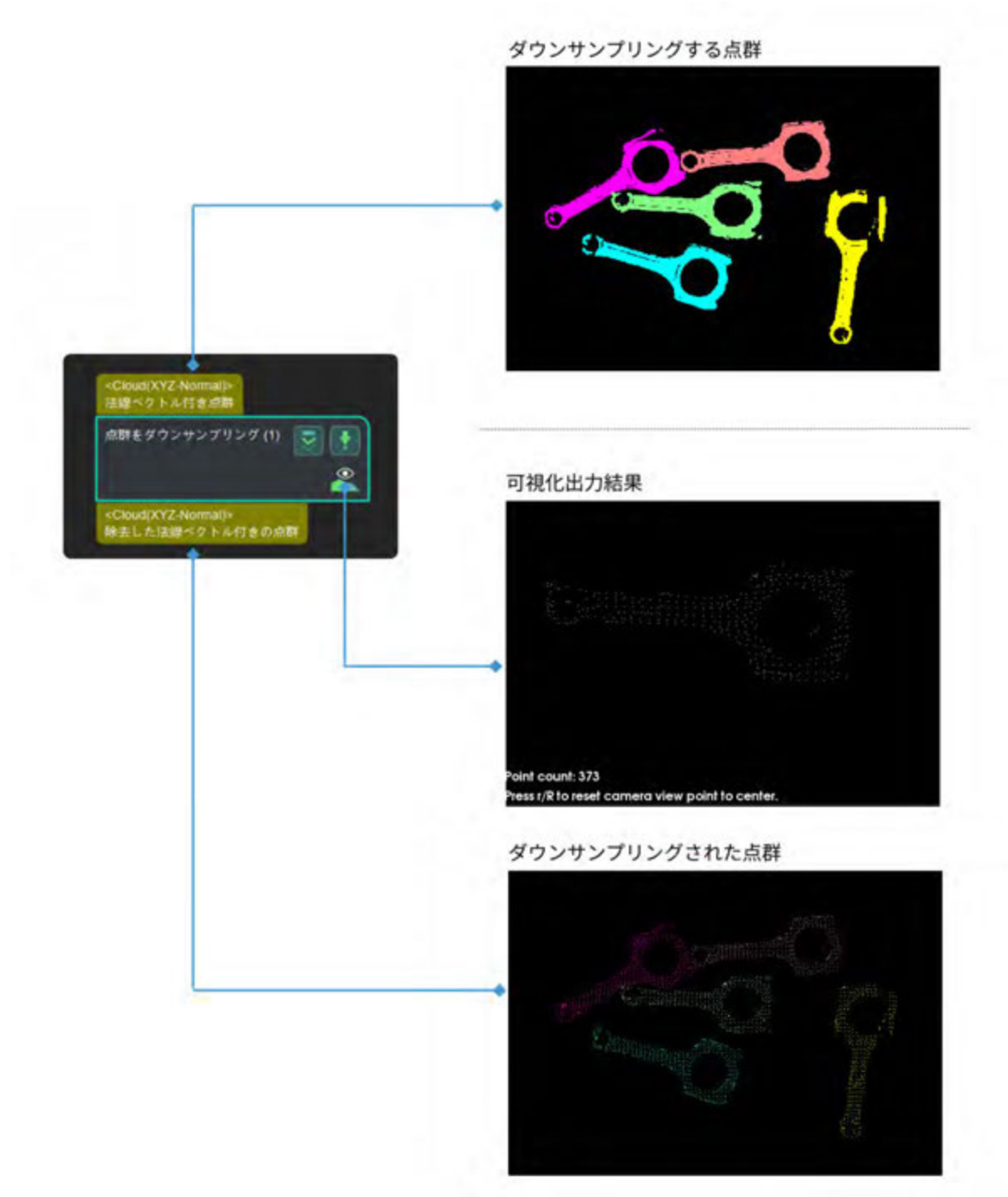
機能

点群をダウンサンプリングし、点の数を減らします。

使用シーン

通常は点群前処理に使用されます。処理する点群の点数が多すぎる場合に使用してプロジェクトの実行を加速させます。

入力と出力



4.3.137. マスクの最小外接長方形を描画



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

入力されたマスクで、マスクの最小外接長方形を描画します。

使用シーン

汎用のマスク処理ステップです。特定の使用シーンはありません。

入力と出力

- **入力：**
 1. このポートに入力される画像は、可視化するための背景として使用されます。
 2. このポートに入力されるマスクには、最小外接長方形が計算されます。
 3. 可視化画像をラベリングするための対象物の寸法。
- **出力：**
 1. 可視化された画像。

4.3.138. 座標系をクイック変換



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

入力位置姿勢を、カメラ座標系とロボット座標系の間で変換します。

使用シーン

汎用の位置姿勢の座標系の変換ステップです。特定の使用シーンはありません。

入力と出力

- **入力：** このポートに入力される位置姿勢には、基準座標系が変換されます。
- **出力：** 基準座標系が変換された後の位置姿勢。

4.3.139. 対象物のインデックスリストをクイック作成



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

インデックス型のデータを作成します。総数Nは、パラメータで設定できます。

使用シーン

汎用のデータ作成ステップです。特定の使用シーンはありません。

入力と出力

- **入力：** インデックスの開区間カットオフ範囲（選択可能。このポートが使用されていない場合、パラメータでの設定が使用されます）。
- **出力：** 作成されたインデックスリスト。

4.3.140. 数値リストをクイック作成



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

数値リスト型のデータを作成します。数値リストは、ステップのパラメータで設定できます。

使用シーン

汎用のデータ作成ステップです。特定の使用シーンはありません。

入力と出力

- **入力：** 数値リストの開区間カットオフ範囲（パラメータの入力タイプが「範囲に基づく」に設定された場合に入力されます。選択可能で、このポートを使用しない場合、パラメータで設定した範囲が使用されます）。
- **出力：** 作成された数値リスト。

4.3.141. 位置姿勢をクイック作成



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

位置姿勢リスト型のデータを作成します。位置姿勢リストは、ステップのパラメータで設定できます。

使用シーン

汎用のデータ作成ステップです。特定の使用シーンはありません。

入力と出力

- **入力：** なし。
- **出力：** 作成された位置姿勢リスト。

4.3.142. 四元数をクイック作成



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

四元数リスト型のデータを作成します。これは、パラメータで設定された回転ベクトルによって作成されます。

使用シーン

汎用のデータ作成ステップです。特定の使用シーンはありません。

入力と出力

- 入力：なし。
- 出力：作成された四元数。

4.3.143. ラベルリストをクイック作成



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

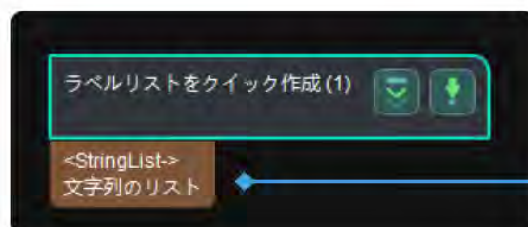
パラメータに入力された文字列リストからラベルリストを作成します。

使用シーン

汎用のラベルリスト作成のステップです。

入力と出力

入力なし



文字列リスト

4.3.144. 3Dベクトルをクイック作成



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

3Dベクトル型のデータを作成します。3Dベクトルは、ステップのパラメータで設定できます。

使用シーン

汎用のデータ作成ステップです。特定の使用シーンはありません。

入力と出力

- 入力：なし。
- 出力：作成された3Dベクトル。

4.3.145. 位置姿勢を基準点にクイック指向



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

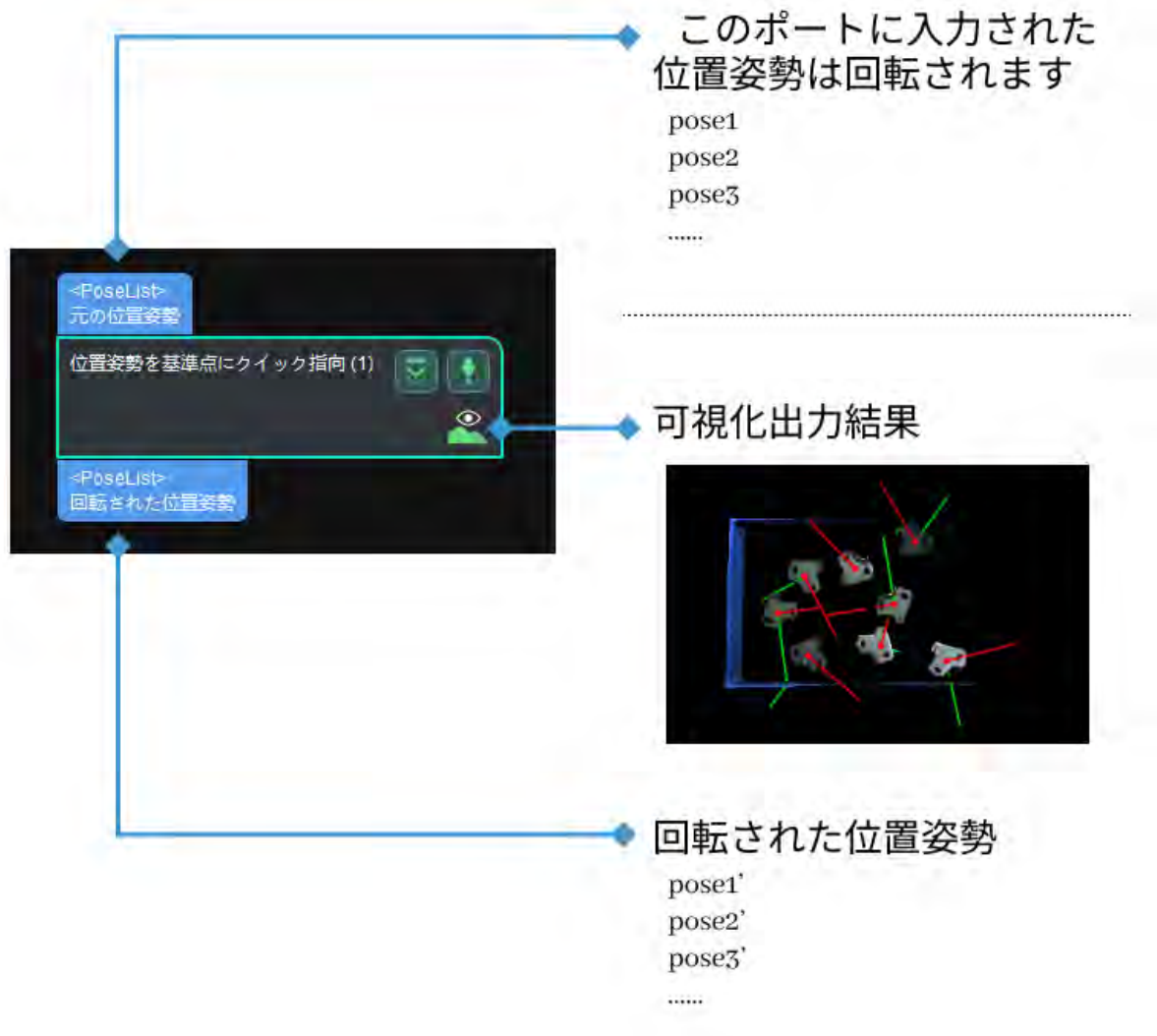
機能

入力位置姿勢の指定軸を、設定された基準点を指すようにします。

使用シーン

汎用の位置姿勢回転ステップです。特定の使用シーンはありません。

入力と出力



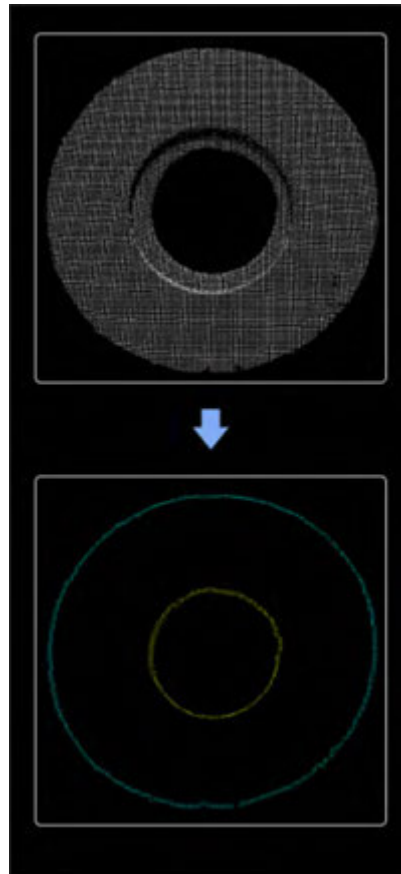
4.3.146. 2D方法による点群エッジを推定



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

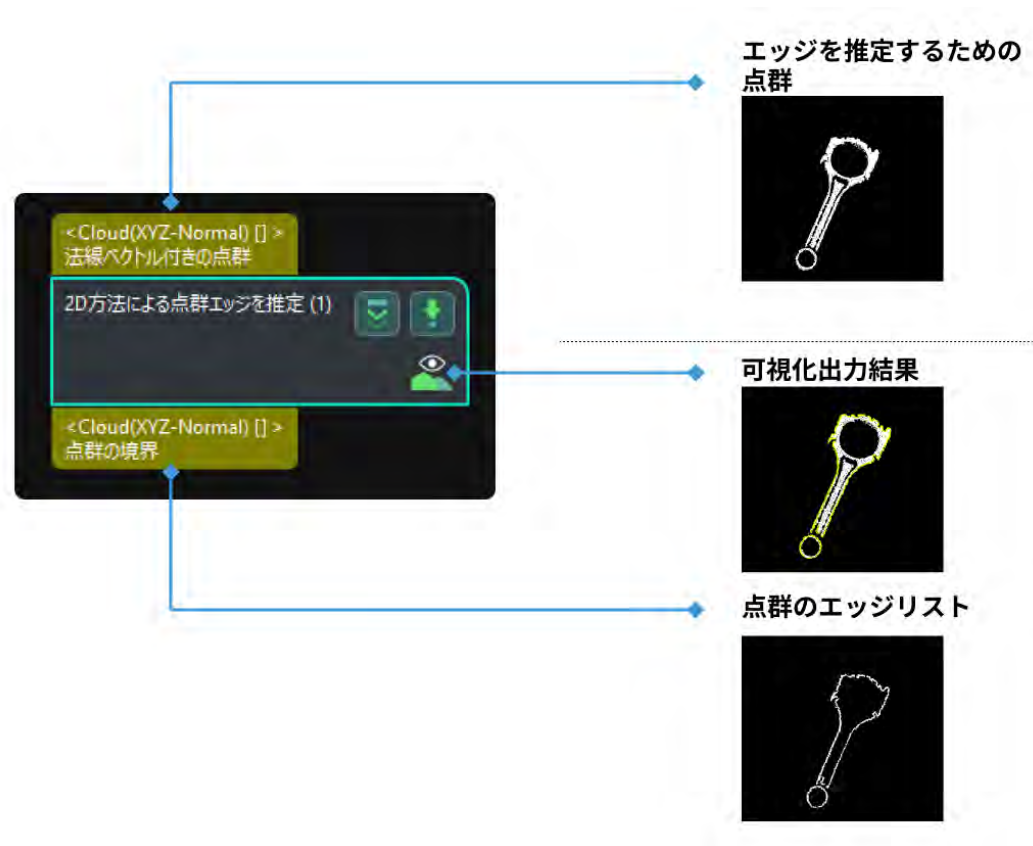
入力点群から点群エッジを抽出します。



使用シーン

このステップは、通常、平面部品の供給シーンに平面部品の点群エッジを取得するために使用されます。3D位置姿勢推定の関連ステップと併用されます。このステップの処理速度は、3Dエッジ抽出アルゴリズムよりも高速なので、実行時間が短く、3D点群の干渉が多く、対象物の最も外側の境界点のみを抽出する必要があるシーンに使用することを推奨します。

入力と出力



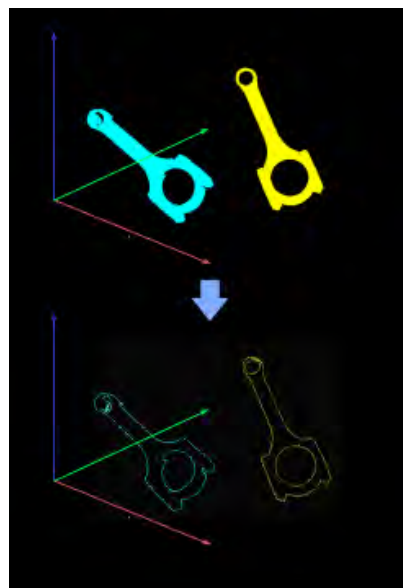
4.3.147. 3D方法による点群エッジを推定



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

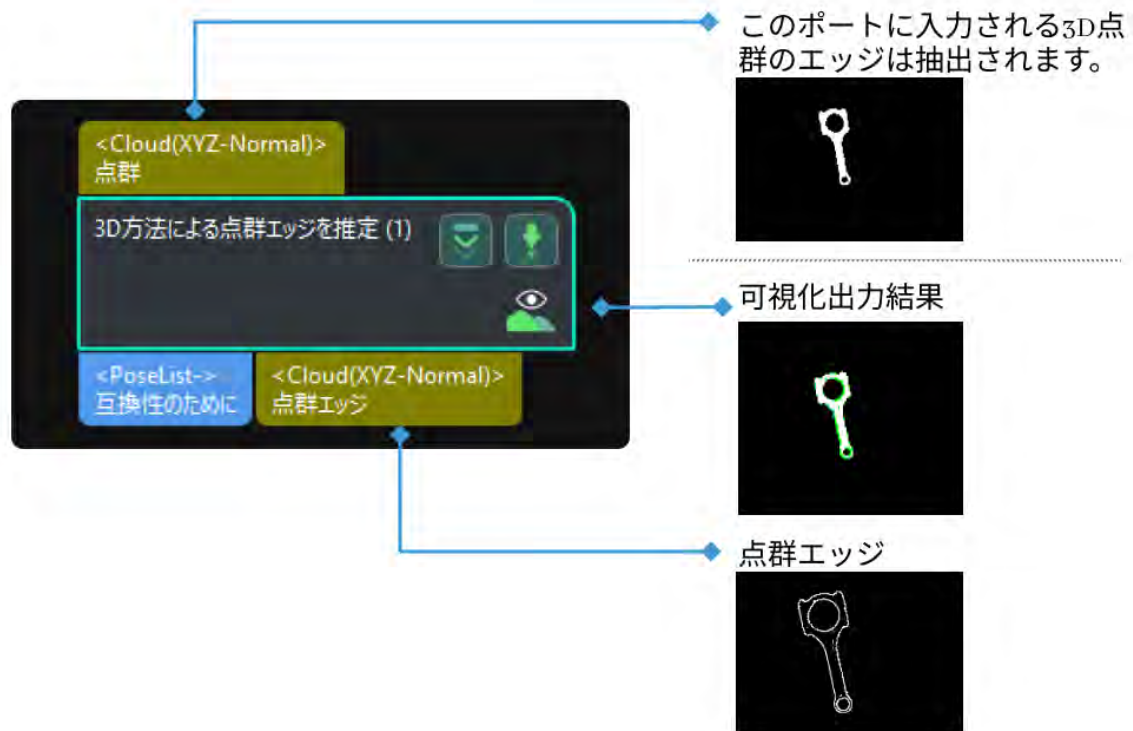
3D点群のエッジを抽出します。



使用シーン

通常、点群のエッジに基づくマッチングに使用され、全部の点群に基づくマッチングより高速です。

入力と出力



4.3.148. 画像の鮮明さを評価

機能

このステップは、異なる計算方法を指定することで、入力画像の鮮明さを定量的に評価することができます。

このステップでは、異なる合焦度でカメラによって撮影された1枚画像または画像群を入力し、出力結果に従ってカメラが合焦しているかどうかを評価することができます。画像の鮮明さの値が大きいほど、カメラのピントが正確に合っていることを意味します。

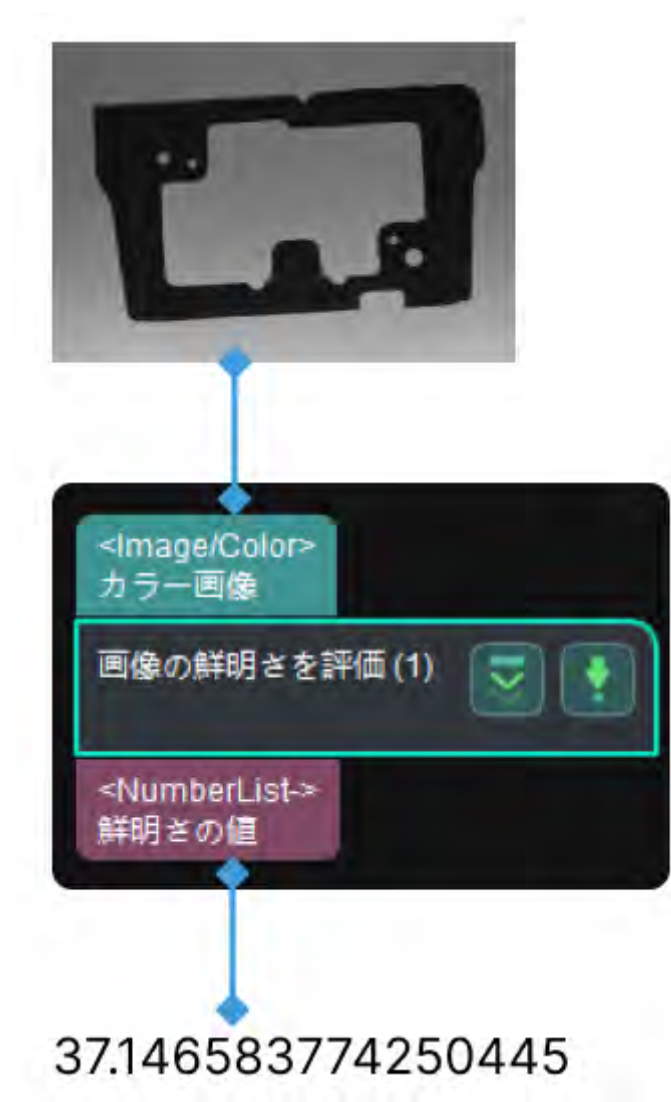


鮮明さの基準とスケールは方法によって異なるため、異なる方法で得られた鮮明さの値を比較することはできません。

使用シーン

通常、計測シーンでカメラのパラメータと位置調整に役立ちます。

入力と出力



パラメータの説明

方法

パラメータ説明：このパラメータは、画像の鮮明さの評価方法を選択するために使用されます。

オプション：Energy of Gradient、Brenner、Roberts Cross、SML、Sobel、Tenegrad

調整説明：各方法の詳細な説明は次の通りです。

- Energy of Gradient：画像の勾配エネルギーを関数化したもの（EOG関数と呼ぶ）です。次のように定義されています。

$$F_{\text{EOG}} = \sum_{i,j} [|g(i, j) - g(i, j - 1)| + |g(i, j) - g(i - 1, j)|]$$

- Brenner：Brenner関数を利用した方法です。次のように定義されています。

$$F_{\text{Brenner}} = \sum_{i,j} |g(i, j) - g(i + 2, j)|^2$$

- Roberts Cross：Robert演算子の差の絶対値和の関数（Robert関数と呼ぶ）に基づいています。次のように定義されています。

$$F_{\text{Robert}} = \sum_{i,j} [|g(i, j) - g(i + 1, j + 1)| + |g(i + 1, j) - g(i, j + 1)|]$$

- SML：ラプラス演算子を一部改良した手法で、ノイズ耐性が高いです。次のように定義されています。

$$F_{\text{SML}} = \sum_{i,j} [|2g(i, j) - g(i - 1, j) - g(i + 1, j)| + |2g(i, j) - g(i, j - 1) - g(i, j + 1)|]^2$$

- Sobel：Sobel演算子を用いた関数（Sobel関数と呼ぶ）です。次のように定義されています。式に含まれる画像のx、y方向の1次Sobel演算子差分です。

$$F_{\text{Sobel}} = \sum_{i,j} [|G_x(i, j)| + |G_y(i, j)|]$$

- Tenengrad：この方法も、ソーベル演算子に基づく関数です。次のように定義されています。

$$F_{\text{Tenengrad}} = \sum_{i,j} [G_x^2(i, j) + G_y^2(i, j)]$$

初期値：Brenner

調整アドバイス：画像の鮮明さの評価方法は、実際のニーズに応じて選択してください。

4.3.149. HDevEngineによる結果を評価



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

HDevEngineを介して事前定義されたHalconビジョンプログラムを実行し、計算結果をMech-

Visionに出力します。

使用シーン

Halconのアルゴリズムの一部を呼び出す必要がある場合は、このステップを使用できます。

入力と出力

- 入力：なし。
- 出力：なし。

4.3.150. JavaScriptエンジンによる結果を評価



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

JavaScriptエンジンを介してユーザーカスタマイズのスクリプトを実行し、計算結果をMech-Visionに出力します。

使用シーン

このステップは、複雑な論理判断を必要とするシーンで使用されます。これにより、ビジョンプロジェクトを簡素化することができます。

入力と出力

- 入力：なし。
- 出力：なし。

4.3.151. 深度画像の変動を評価



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

同じシーンで時間経過に伴う深度画像の変動を評価し、その結果をログに表示します。

使用シーン

開発者のテスト用のステップです。普通のユーザーは、一般的な使用シーンでこのステップを使用する必要はありません。

入力と出力

- 入力：
 1. このポートに入力される深度画像には、変動が評価されます。
 2. このポートに入力されるマスクは、評価領域を設定するために使用されます。
- 出力：
 1. なし。

4.3.152. 2D位置姿勢を評価



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

評価する位置姿勢を真値の位置姿勢と比較し、比較結果をシーンの画像に表示します。真値の位置姿勢は白で表示され、合格位置姿勢は緑で表示され、不合格位置姿勢は赤で表示されます。

使用シーン

特定のシーン向けのカスタマイズのステップです。

入力と出力

- 入力：
 1. このポートに入力される画像は、可視化するための背景として使用されます。
 2. 評価する位置姿勢。
 3. 真値の位置姿勢。
- 出力：
 1. 評価する位置姿勢および真値の位置姿勢を含む可視化画像。

4.3.153. 外部インターフェース



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

外部インターフェースから、パラメータの変化を受け入れます。

使用シーン

特定のシーン向けカスタマイズのステップです。オペレーター・インターフェイスの複雑な状

況を判断し、手動処理の結果をプロジェクトに送信するために使用されます。

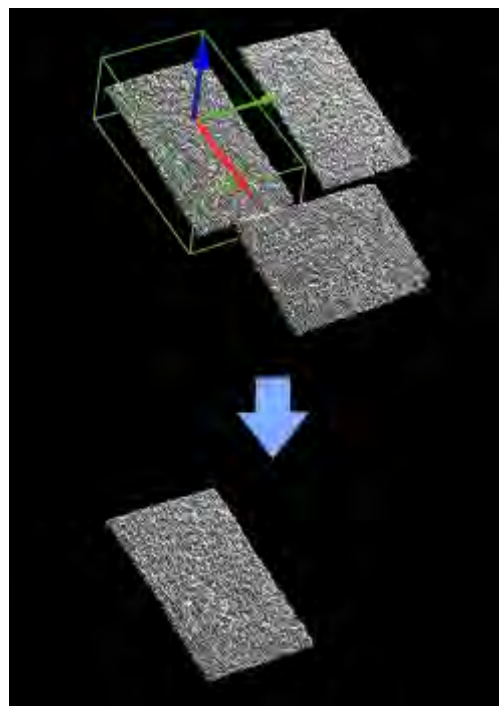
入力と出力

- 入力：なし。
- 出力：なし。

4.3.154. 直方体内の点群を抽出

機能

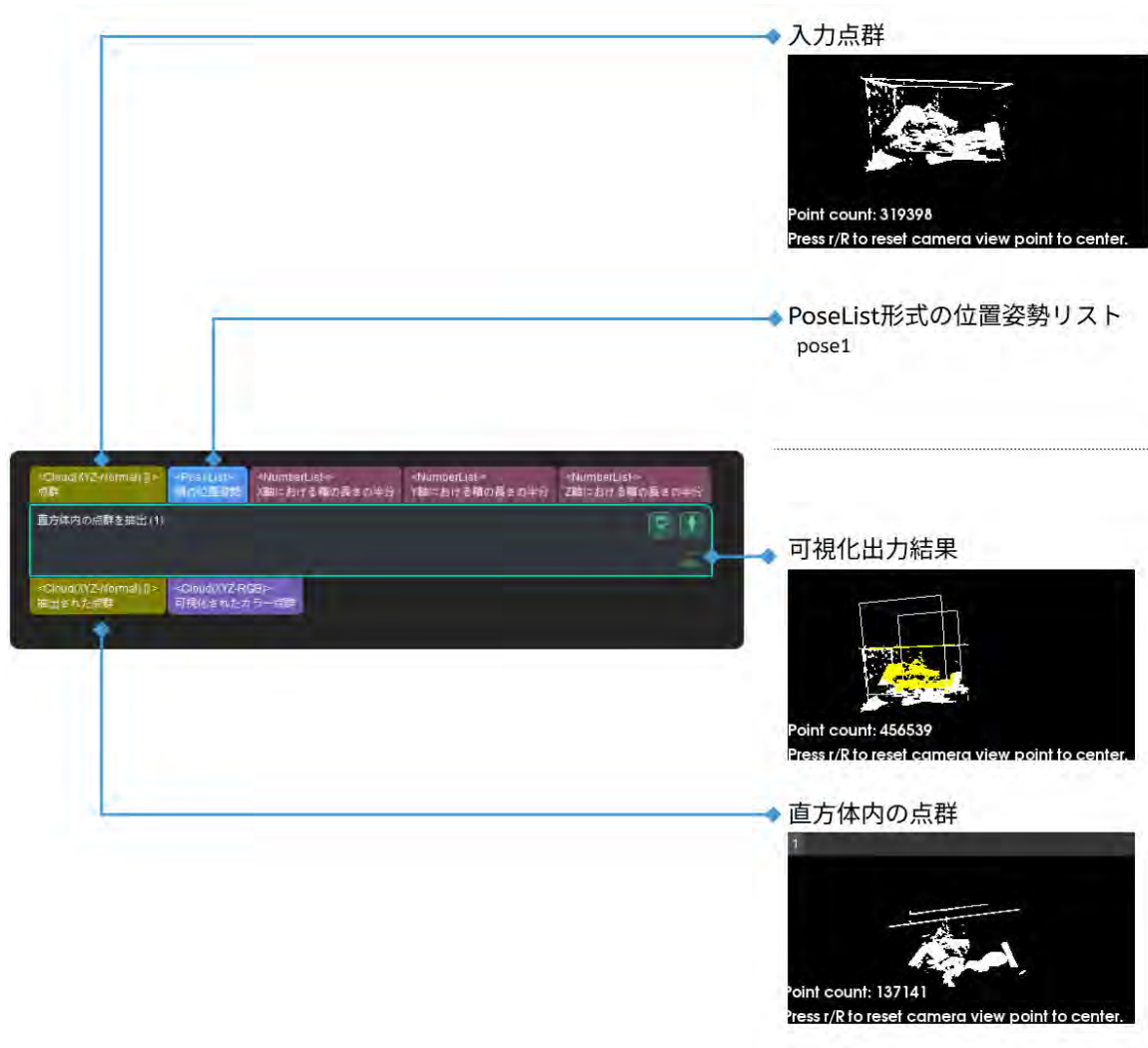
指定された直方体内の点群を抽出し、それ以外の点群は破棄されます。



使用シーン

汎用の点群抽出ステップで、特定の使用シーンはありません。直方体の寸法は、ポートから入力するか、パラメータで設定できます。

入力と出力



パラメータの説明

箱の寸法設定

X軸における長さの半分

初期値：100.000 mm

調整説明：このパラメータは、ワールド座標系のX軸における箱の長さの半分で、下図の **halfX** に対応します。単位はミリメートルです。

Y方向における長さの半分

初期値：100.000 mm

調整説明：このパラメータは、ワールド座標系のY軸における箱の長さの半分で、下図の **halfY** に対応します。単位はミリメートルです。

Z方向における長さの半分

初期値：100.000 mm

調整説明：このパラメータは、ワールド座標系のZ軸における箱の長さの半分で、下図の **halfZ** に対応します。単位はミリメートルです。

[extract 3d points in cuboid schematic diagram] | *extract-3d-points-in-cuboid-schematic_diagram.png*

入力点群の使用方法設定

初期値：OneToOne

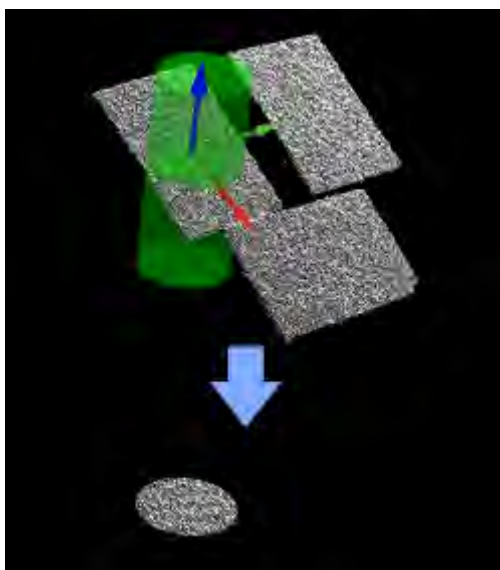
オプション：OneToOne、FirstToAll、AllToFirst

調整説明：異なるパラメータによって、入力点群の使用方法は異なります。

4.3.155. 円柱以内の点群を抽出

機能

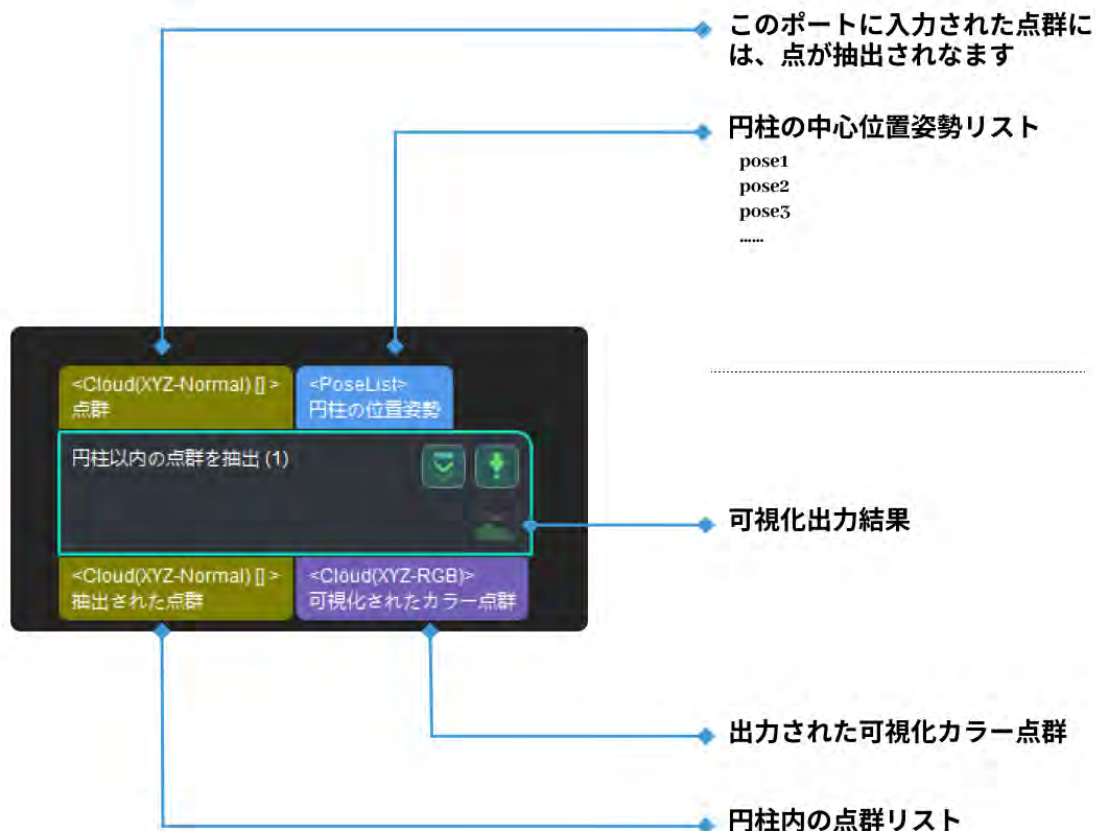
指定された円柱内の点群を抽出し、それ以外の点群は破棄されます。



使用シーン

汎用の点群抽出ステップで、特定の使用シーンはありません。

入力と出力



パラメータの説明

形状タイプ

初期値：中空円柱。

オプション：円柱、中空円柱、扇形の柱。

調整説明：異なるシーンのニーズに応じて、異なる形状の点群を抽出できます。形状を選択すると、指定した形状内の3D点群が抽出されます。

Z方向のパラメータ

最小Z値

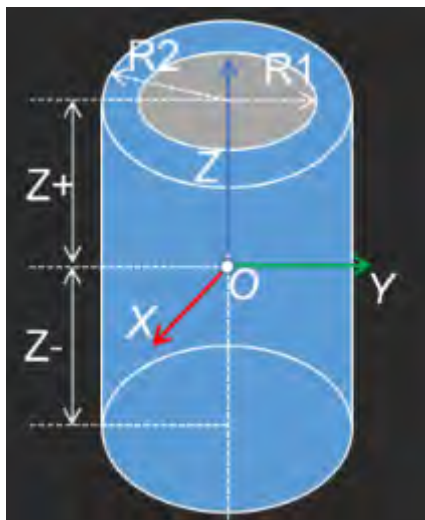
初期値：-100.000mm

調整説明：これはZ軸における最小座標（ミリメートル単位）で、下図の **Z-** に対応します。単位はミリメートル（mm）です。

最大Z値

初期値：100.000mm

調整説明：これはZ軸における最大座標で、下図の **Z** に対応します。単位はミリメートル（mm）です。



中空円柱と扇形の柱の半径のパラメータ

内円半径

初期値：50.000mm

調整説明：XY平面の最小距離で、上図の **R1** に対応します。単位はミリメートル（mm）です。このパラメータを **0** に設定すれば、円柱体を取得できます。

外円半径

初期値：100.000mm

調整説明：XY平面の最大距離で、上図の **R2** に対応します。単位はミリメートル（mm）です。

扇形の角度のパラメータ

形状のタイプ を **扇形の柱** に設定すれば、以下の2つのパラメータは有効になり、扇形の形状を設定できます。

開始角度

初期値：-180°

調整説明：このパラメータは、XY平面上における扇形の開始角度です。

終了角度

初期値：180°

調整説明：このパラメータは、XY平面上における扇形の終了角度です。

入力点群の使用方法設定

点群の中心によって物体が円柱内にあるかどうかを判断

初期値：チェックを入れない。

調整説明：チェックを入れると、対象点群の中心が円柱内にあるかどうかをチェックするために使用されます。それ以外の場合はチェックされません。

入力点群の使用方式

初期値：All

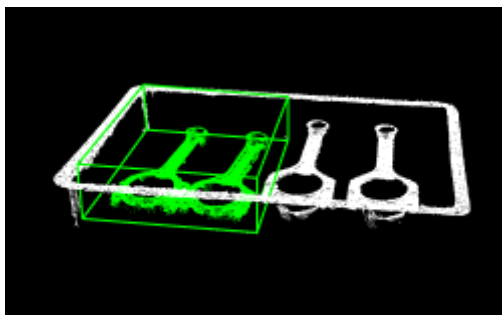
オプション：All、FirstOnly

調整説明：異なるパラメータの値を選択することで、異なる入力点群の使用方法を切り替えることができます。

4.3.156. 3D ROI内の点群を抽出

機能

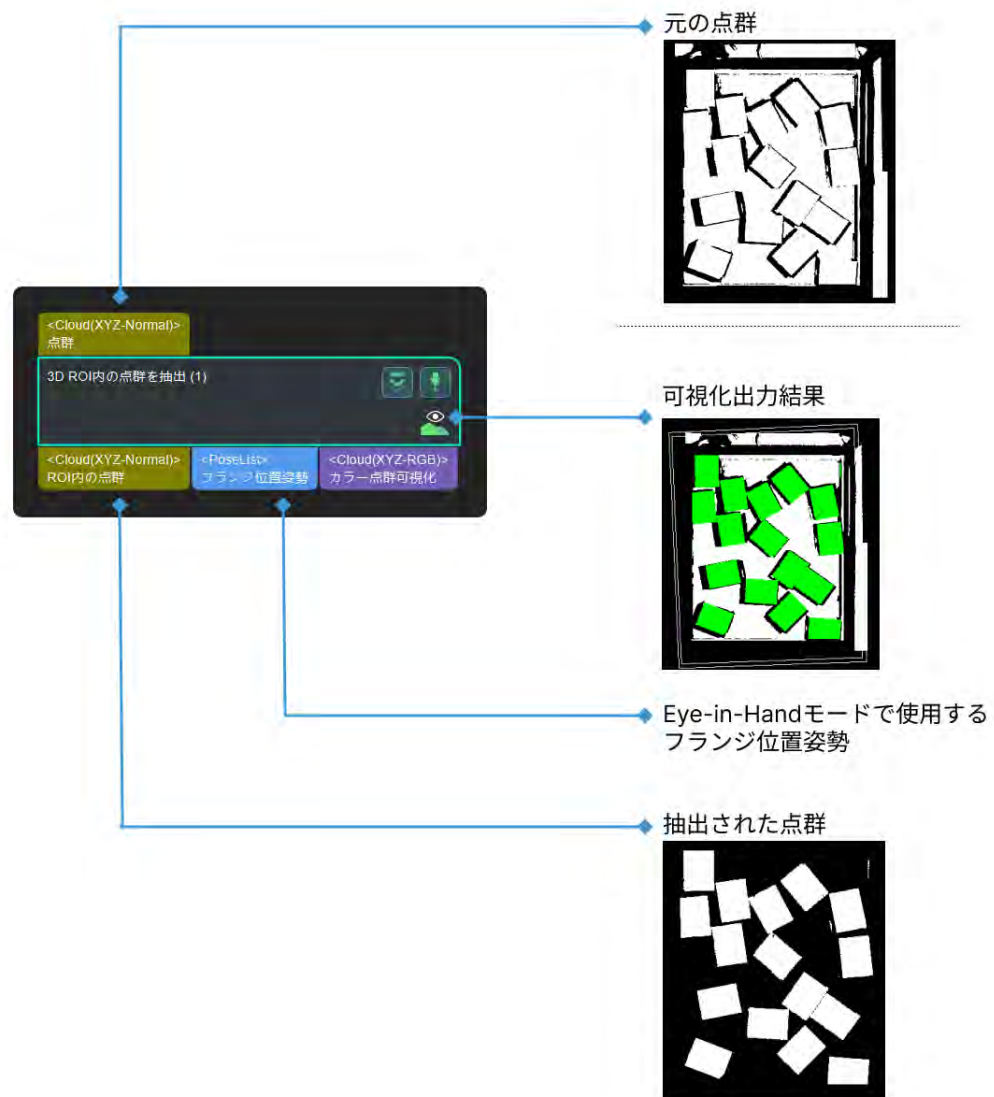
3D空間内に関心領域（ROI）を設定します。領域内の点群は保持され、領域外の点群は除去されます。



使用シーン

このステップを使用して対象物の点群に焦点を絞り、背景点群と干渉点群を無視します。

入力と出力

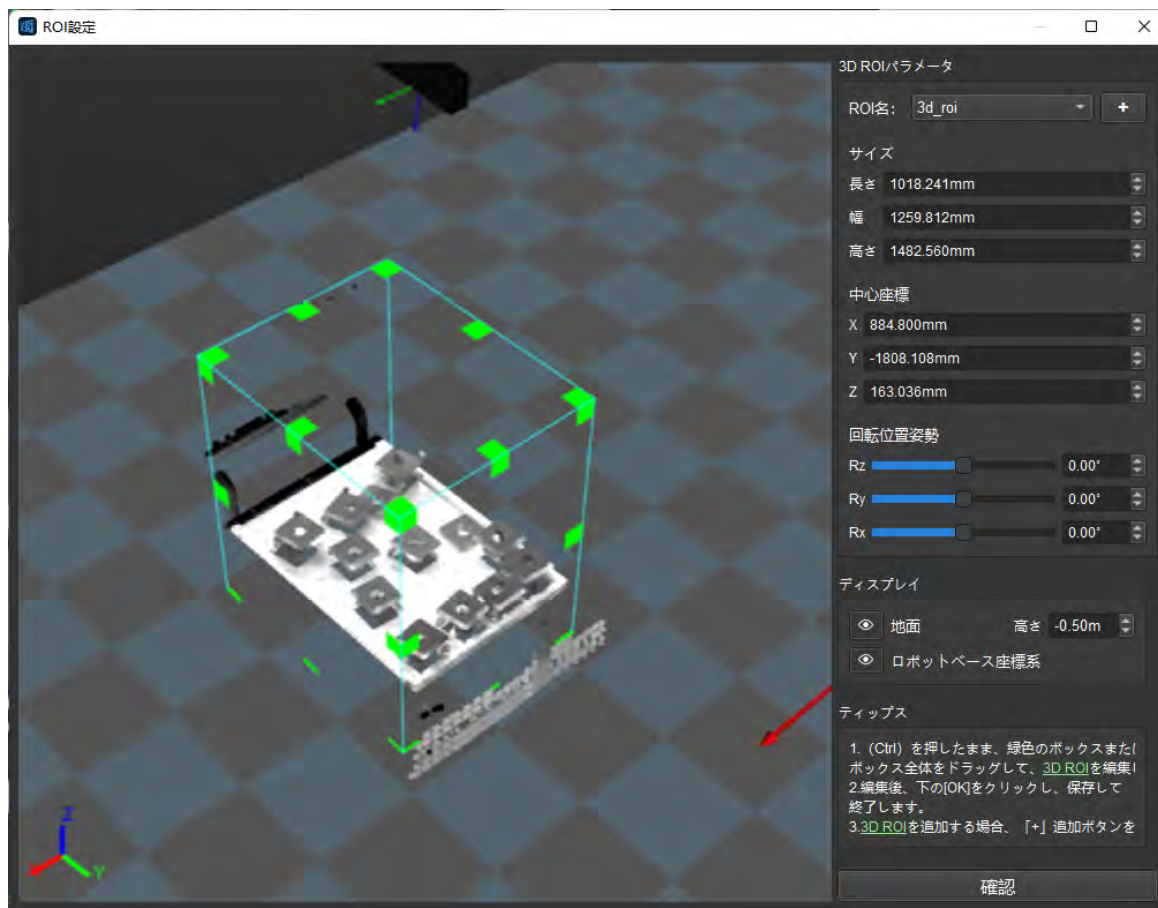


パラメータの説明

3D ROI設定

3D ROIファイル

調整説明：下図に示すように、[**3D ROIを設定**] をクリックしてその画面に入ります。詳細については、[3D ROIの設定手順](#) をご参照ください。

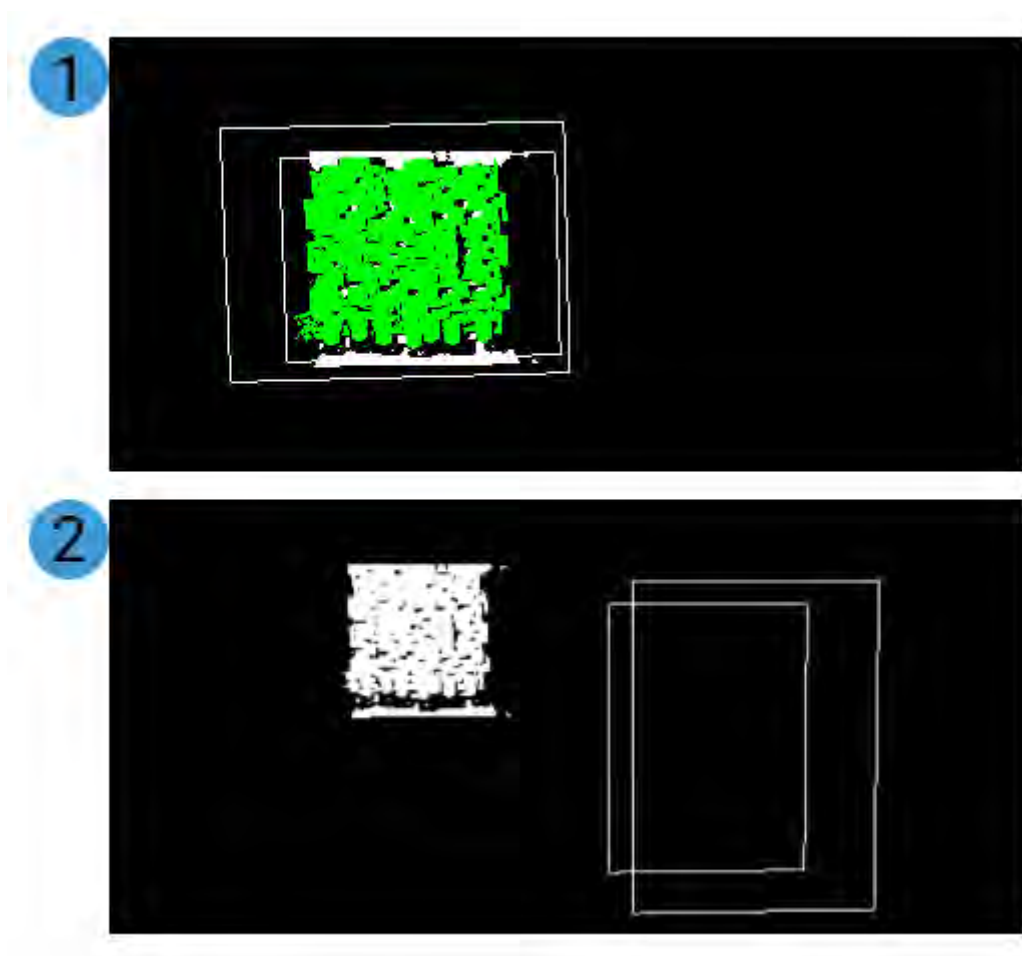


入力点群の座標系タイプ

初期値：カメラ座標系

オプション：カメラ座標系、ロボット座標系

調整説明：入力点群の座標系を変更します。下図に示すように、①はカメラ座標系での入力点群で、②はロボット座標系での入力点群です。



空の状態の設定

3D ROI内の最小点数

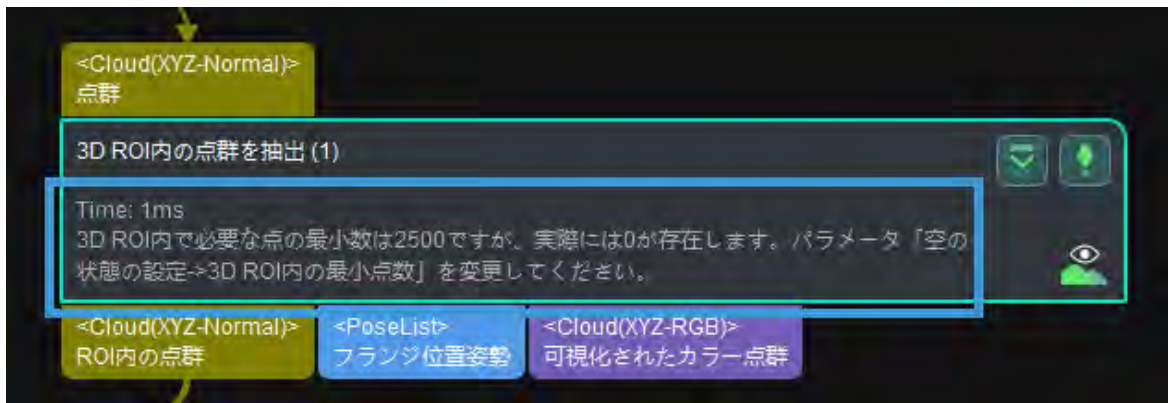
初期値：0

調整説明：このパラメータは3D ROIに3D点を抽出する数の最小値を設定するために使用されます。3D ROIで抽出した3D点の数はこの値より小さければ、設定した3D ROI内の点の数は最小値の要求を満たさないことになりますので、点群を送信しません。

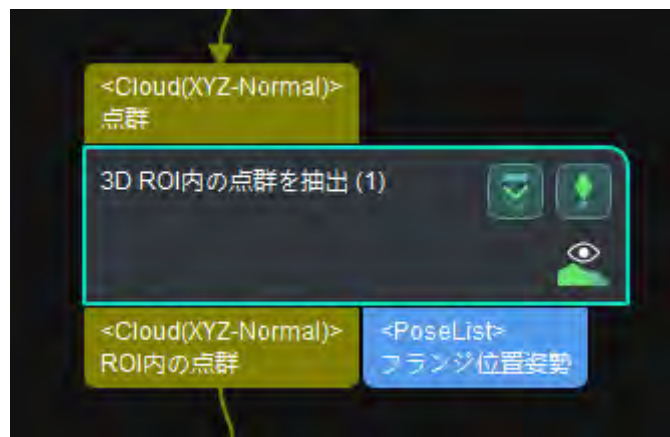
3D ROI内の点が空の状態であるかを送信

初期値：チェックを入れる

調整説明：チェックを入れれば、3D ROI内の点が空の状態であるかを送信します。初期値にしたら、3D ROI内の点の数は「3D ROI内の最小点数」に設定した値より少ない場合、下図のようなメッセージが出てきます。



下図に示すように、チェックを入れなければ、メッセージは表示されません。



4.3.157. 深度画像の3D ROIにゼロ深度値の領域を抽出



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社 (docs@mech-mind.net) までお問い合わせください。

機能

ステップパラメータで3D ROI設定し、深度画像のROIに深度値がゼロの領域を出力します。

使用シーン

特定のシーン向けのカスタマイズのステップです。

入力と出力

- **入力：** このポートに入力される深度画像の一部領域（ROIに深度値がゼロの領域）は抽出されます。
- **出力：** 深度画像のROIに深度値がゼロの領域。

4.3.158. 穴埋め処理

機能

入力二値画像の穴、すなわち非ゼロのピクセルで完全に囲まれた領域を埋めるために使用されます。



- 二値画像とは、画像上の各ピクセルが2つの可能な値しか持たない画像のことです。一般的には白黒画像、モノクロ画像などが含まれます。
- このステップでは、非ゼロピクセルは、ピクセル値255のピクセルとなります。

使用シーン

通常、画像分割に完全なターゲット領域の画像を取得し、穴領域の画像欠落による干渉を避けるために使用されます。

入力と出力



パラメータの説明

穴のタイプ

パラメータ説明：このパラメータは、充填する穴の種類を選択するために使用されます。

オプション：完全な穴、すべての穴。

- 完全な穴：非ゼロのピクセルで完全に囲まれた閉じた穴のみを埋めます。調整効果を [調整の例](#) に示します。
- すべての穴：非ゼロピクセルに囲まれた閉じた穴と、すべてのエッジが非ゼロピクセルに囲まれていない穴の両方を埋めます。調整効果を [調整の例](#) に示します。

初期値：完全な穴。

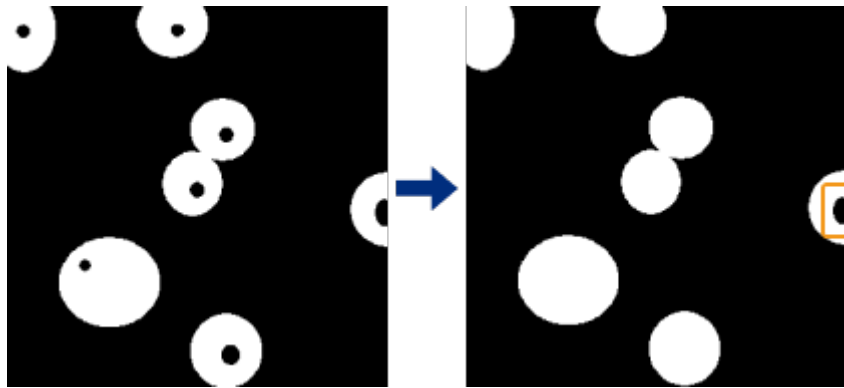
調整説明：実際の状況に応じて選択してください。

調整の例

完全な穴を埋める例

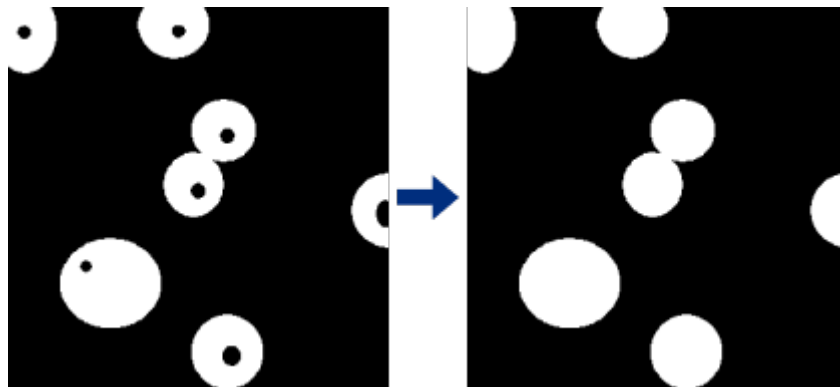
完全な穴を埋めた効果は下図の右側のとおりで、非ゼロピクセルで囲まれた閉じた穴だけが埋

められ、すべてのエッジが非ゼロピクセルで囲まれていないの穴（右側の黄色の枠で囲まれたもの）は埋められません。



すべての穴を埋める例

すべての穴を埋めた効果は下図の右側のどおりで、非ゼロピクセルで囲まれた閉じた穴も、すべてのエッジが非ゼロピクセルで囲まれていないの穴も埋められます。



4.3.159. ラベルによってフィルタリング



これは古いバージョンのステップであり、メンテナンスが停止されますので、新しいバージョンのステップ [\[vision-steps:validate-labels-and-output-flags::validate-labels-and-output-flags\]](#) を使用してください。このステップに関する詳細が必要な場合は、当社 (docs@mech-mind.net) までお問い合わせください。

機能

元ラベルリストにある全ての項目は基準ラベルリストにあるかを検査します。検査の結果はブール値の形式で出力されます。

元ラベルが基準ラベルリストにあれば、出力されるリストはTrueになります。

使用シーン

通常はステップ [\[vision-steps:filter::filter\]](#) と併用されます。このステップに出力されるブール値リストは配列をフィルタリングする基準になります。

入力と出力



4.3.160. 位置姿勢によってマスクをフィルタリング

機能

入力マスクが基準位置姿勢をカバーしているかどうかを判断します。基準位置姿勢をカバーするマスクは保持され、カバーしないマスクは削除されます。

使用シーン

特定のシーン向けのカスタマイズのステップです。

入力と出力

● 入力：

1. このポートに入力されるマスクは、フィルタリングされます。
2. マスクをフィルタリングするための位置姿勢。

● 出力：

1. 残りのマスク。

2. マスク内の位置姿勢。
3. マスク外の位置姿勢。

4.3.161. 箱外の位置姿勢を除去



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

候補位置姿勢（および対応する点群）、箱の2つの側面の位置姿勢を入力し、候補位置姿勢（および対応する点群）を計算して出力します。

使用シーン

箱以外の位置姿勢を除去するために使用されます。

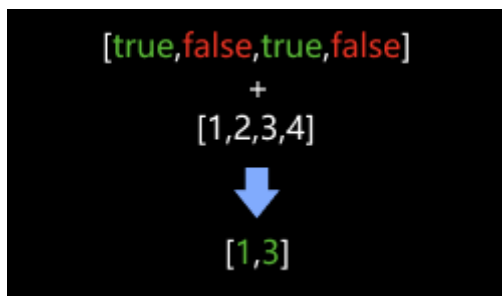
入力と出力

- 入力：
 1. このポートに入力される位置姿勢には、箱以外の位置姿勢が除去されます。
 2. このポートに入力される点群には、箱以外の位置姿勢が除去されます。
 3. 箱の2つの側面の3D中心位置姿勢。
- 出力：
 1. 箱内の位置姿勢。
 2. 箱内の位置姿勢の点群。
 3. 箱の2つの側面における立体的な対角線の点の位置姿勢。
 4. 箱以外の点群。

4.3.162. フィルタリング

機能

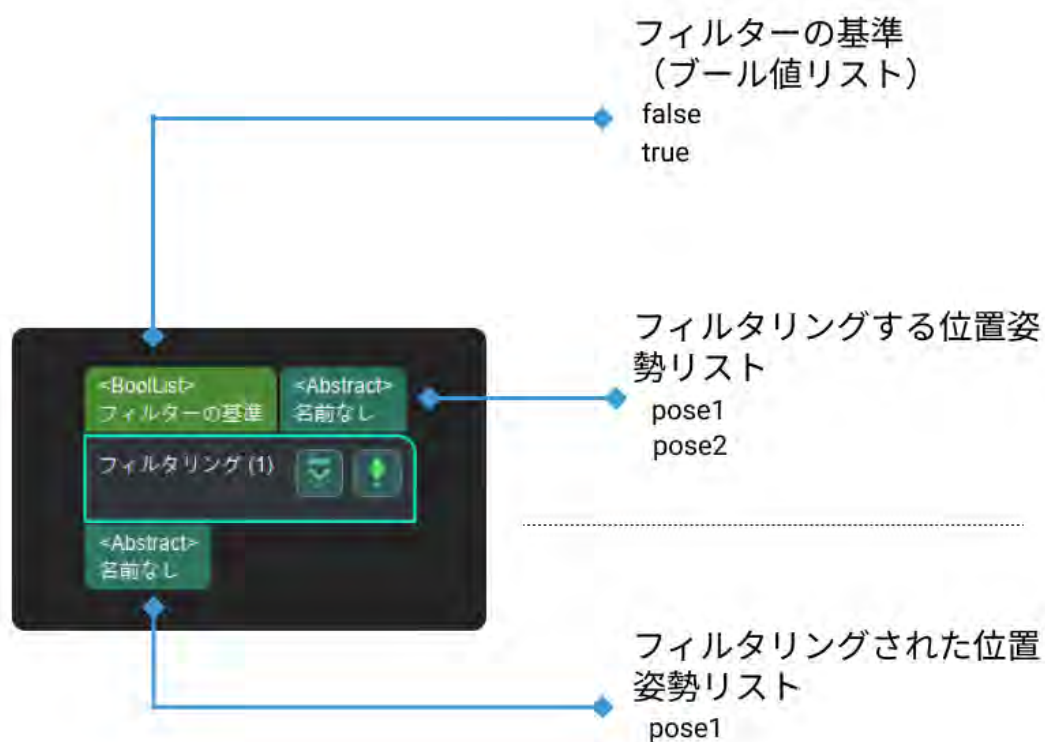
汎用配列フィルターとして機能します。入力された配列をTrue/Falseリストとマッチングし、Trueが対応するデータリストを出力します。



使用シーン

ブール値リストによって入力されたデータリストをフィルタリングします。このステップは通常、しきい値によって数値を二項分類、必要なラベルであるかどうかを検証、基準方向との角度によって位置姿勢が有効かどうかを検証、点群が要求を満たすかどうかを検証などブール値が出力できるステップと併用されます。

入力と出力



パラメータの説明

ポート設定

ポート数 (1-15)

初期値：1

調整可能な範囲：1~15

調整説明：ポートの数をカスタイズするために使用されます。

操作レイヤー

初期値：0

調整可能な範囲：0~14

調整説明：操作レイヤーが0の場合、リストにある全ての要素に対して操作します。

ブール値を反転するか

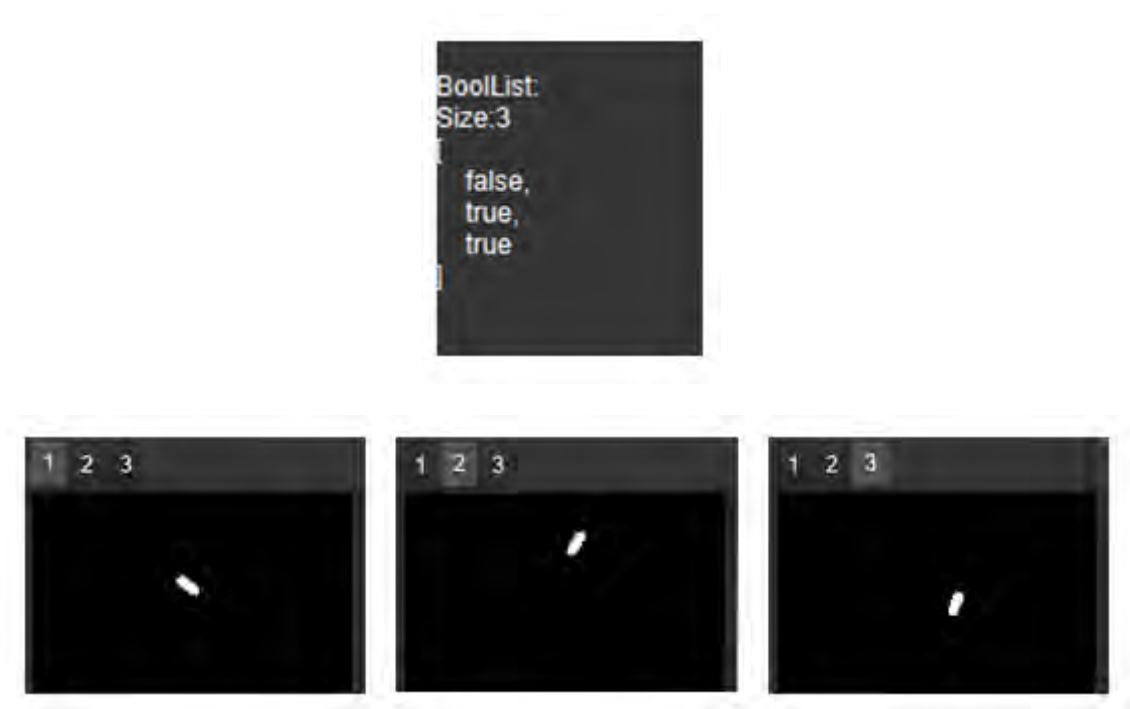
初期値：チェックを入れない。

調整説明：

- チェックを入れない：入力のブール値リストに「False」を持つ項目を除去します。
- チェックを入れる：入力のブール値リストに「True」を持つ項目を除去します。

調整の例：

ブール値と入力情報は下図に示します。



ブール値を反転するか にチェックを入れない場合、下図に示すように、入力ブール値リストにブール値がfalseである項目が除外されます。



ブール値を反転するか にチェックを入れる場合、下図に示すように、入力ブール値リストにブール値がtrueである項目が除外されます。



4.3.163. 2Dコーナーを取得



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社 (docs@mech-mind.net) までお問い合わせください。

機能

2D画像のコーナーを取得します。

使用シーン

2D輪郭のコーナーを検出するために使用されます。

これは古いバージョンのステップであり、メンテナンスが停止されていますので、新しいバージョンのステップ [\[vision-steps:detect-corners:::detect-corners\]](#) を使用してください。

入力と出力

● 入力：

1. このポートに入力される2D画像には、対象物のコーナーが検出されます。

● 出力：

1. 検出された対象物の輪郭。
2. コーナーを可視化するための画像。

4.3.164. 位置姿勢とオフセットの対応関係を求める



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場

合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

対象物の位置姿勢と把持位置姿勢との対応関係を求め、それらの間のオフセットを計算します。対象物の位置姿勢は、ステップパラメータでオフセットによって調整できます。

使用シーン

特定のシーン向けのカスタマイズのステップです。

入力と出力

● 入力：

1. 元の対象物の位置姿勢。
2. 対象物が対応する把持位置姿勢。
3. 対象物の寸法。

● 出力：

1. 対象物の位置姿勢。
2. 元の対象物の位置姿勢から把持位置姿勢までのオフセット。
3. 対象物の寸法。

4.3.165. 2D平面で丸穴を検出



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

元の画像およびマスク画像で対象物の丸穴を取得します。

使用シーン

特定のシーン向けのカスタマイズのステップです。

入力と出力

● 入力：

1. このポートに入力されるマスクには、丸穴が検出されます。
2. 対象物の輪郭の画像。

● 出力：

1. 検出された対象物の丸穴のマスク。

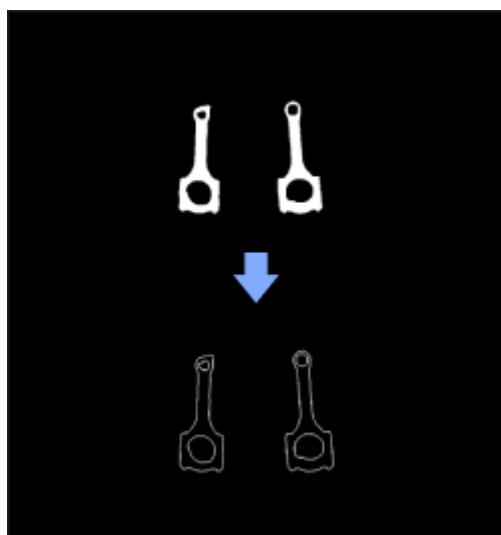
4.3.166. 指定した内外層で2D輪郭を検出



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

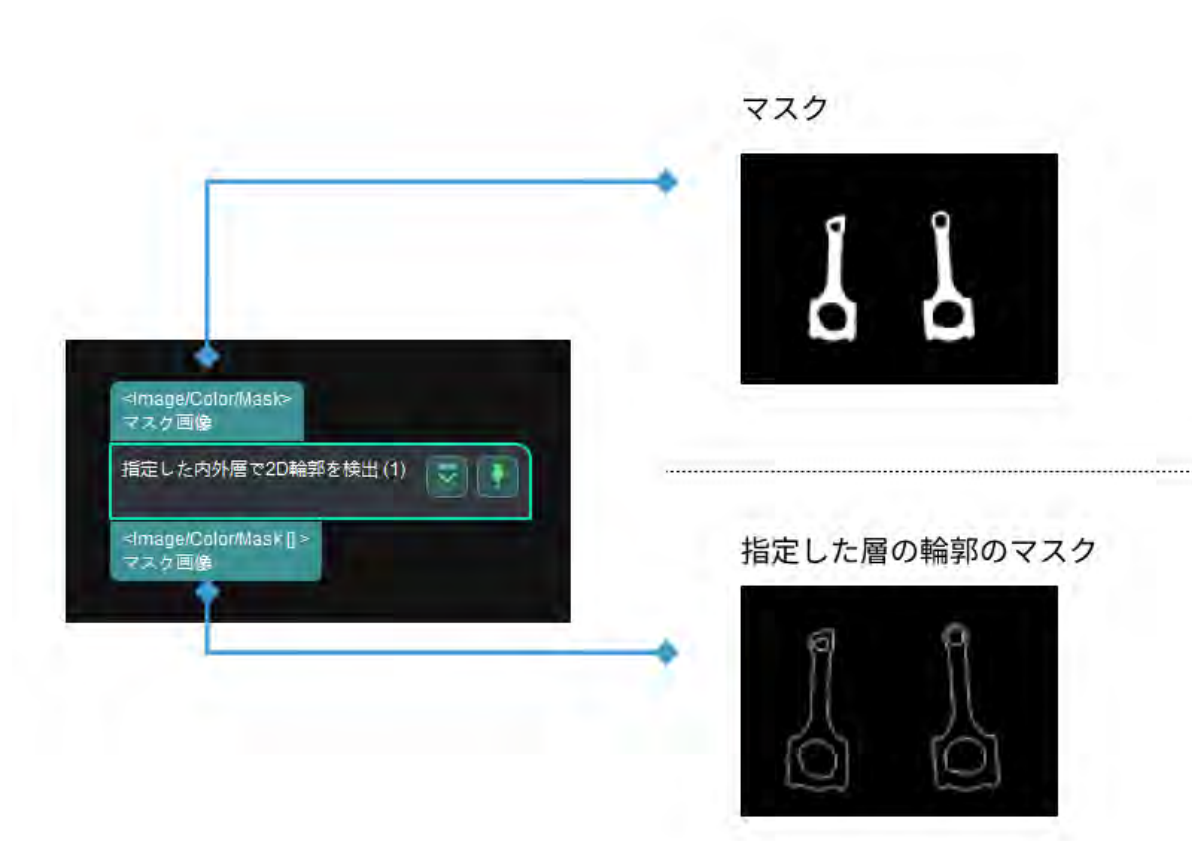
マスク画像の指定層の輪郭を取得します。



使用シーン

汎用の2D画像の輪郭抽出ステップです。3D点群の輪郭を取得するために、ステップ [\[vision-steps:orthographic-projection:::orthographic-projection\]](#) と [\[vision-steps:orthographic-projection-inverse:::orthographic-projection-inverse\]](#) の間で使用できます。

入力と出力



4.3.167. 円のフィッティング

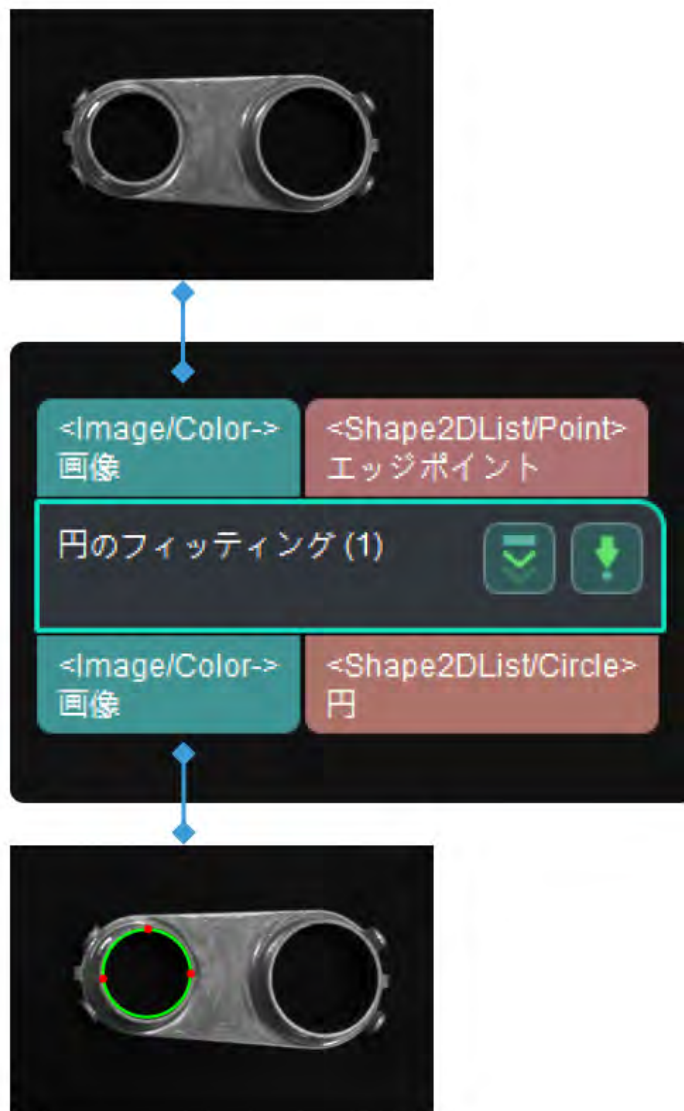
機能

入力される2D画像内の点を円にフィッティングします。

使用シーン

通常、計測シーンに使用されます。

入力と出力



パラメータの調整説明

フィッティング方法

パラメータ説明：円をフィッティングする方法を設定します。

オプション：Huber、最小二乗法。

初期値：Huber

- 最小二乗法：この方法を使用する場合、点とフィッティングによって得られた円の距離の和が最小になります。
- Huber：加重最小二乗法を使用して円のフィッティングを行います。この方法を使うと、フィッティングによって得られた円に対する外れ値の影響を軽減しています。

調整説明：通常の場合は初期設定を推奨します。

4.3.168. 直線のフィッティング

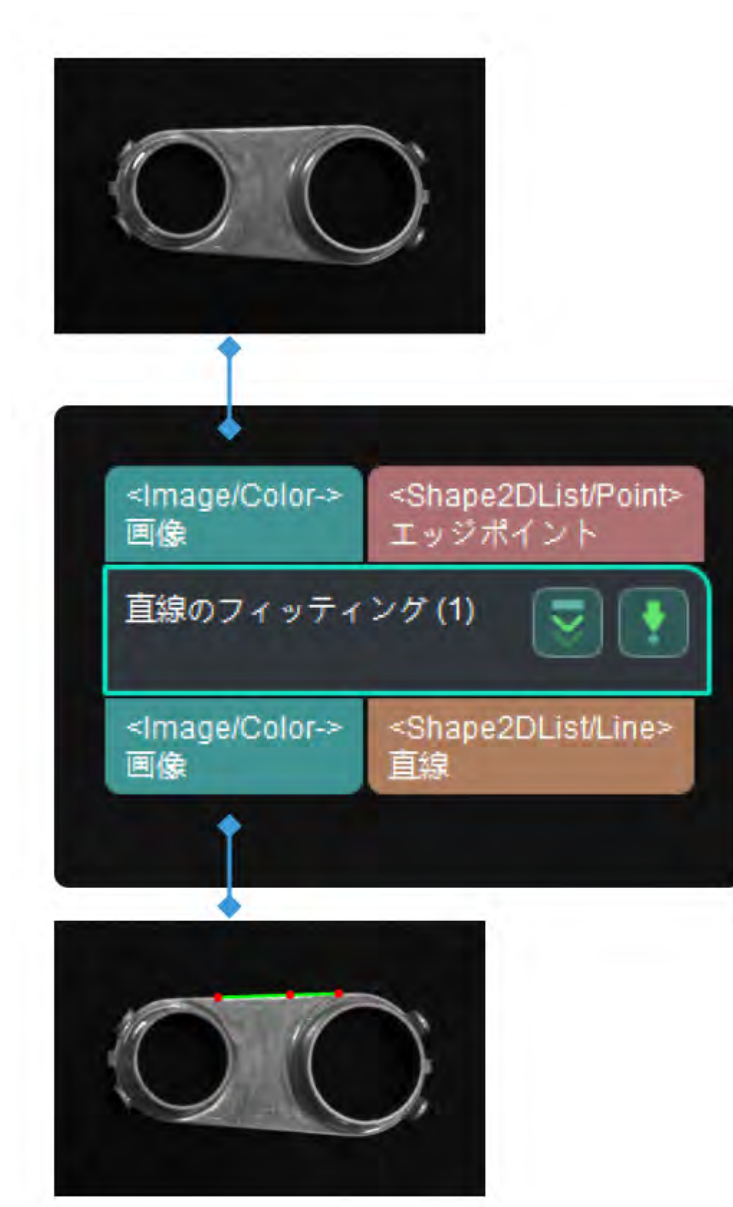
機能

入力2D画像内の点を直線にフィッティングします。

使用シーン

通常、計測シーンに使用されます。

入力と出力



パラメータの説明

フィッティング方法

パラメータ説明：直線をフィッティングする方法を設定します。オプション：Huber、最小二乗法。初期値：Huber

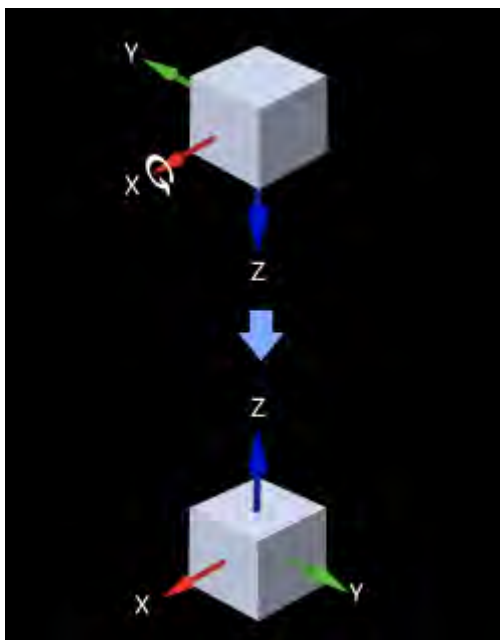
- 最小二乗法：この方法を使用する場合、点とフィッティングによって得られた直線間の距離の和が最小になります。
- Huber：加重最小二乗法を使用して直線のフィッティングを行います。この方法を使うと、フィッティングによって得られた直線に対する外れ値の影響を軽減しています。

調整説明：通常の場合は初期設定を推奨します。

4.3.169. 位置姿勢の座標軸方向を反転

機能

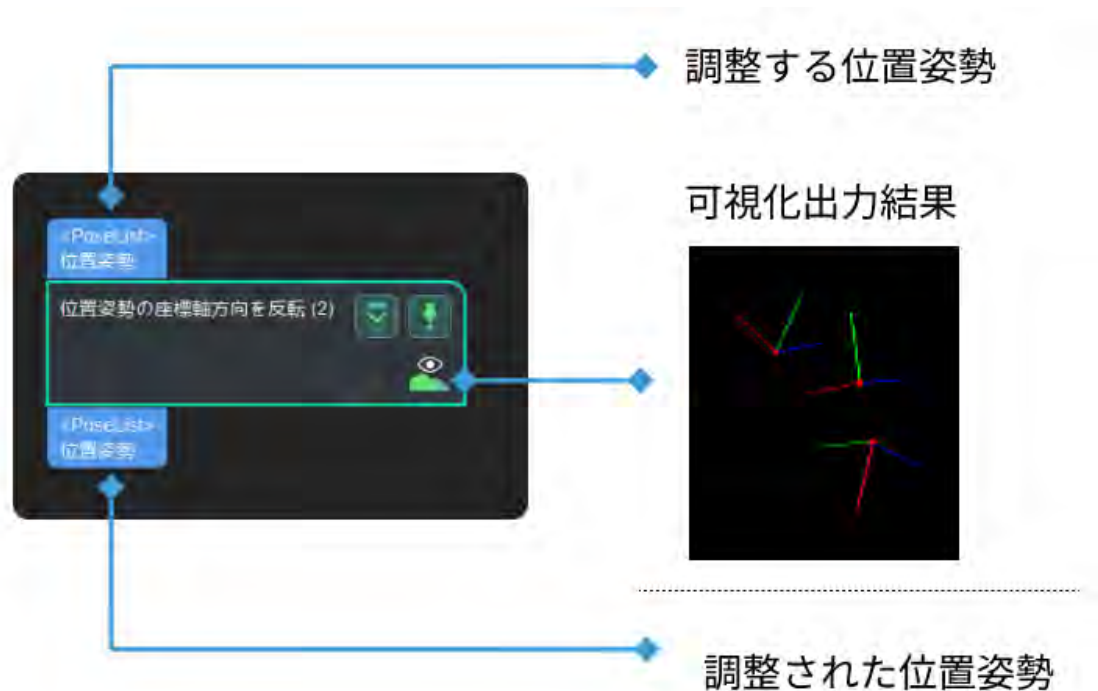
方向を指定し、入力位置姿勢のある軸の方向を調整します。



使用シーン

通常、位置姿勢の後処理に使用されます。位置姿勢のある軸を一つの向きに統一させます。

入力と出力



パラメータの説明

可視化設定

位置姿勢表示の設定

初期値：出力位置姿勢のみを表示。

値のリスト：入力位置姿勢のみを表示、出力位置姿勢のみを表示、入力と出力位置姿勢をすべて表示。

調整説明：処理後の位置姿勢、処理前の位置姿勢、処理前後の全ての位置姿勢をそれぞれ表示します。

軸の設定

反転する軸

初期値：Z

オプション：X、Y、Z

調整説明：反転する座標軸を選択します。

方向のタイプ

初期値：負方向。

オプション：負方向（位置姿勢のある軸を世界座標系の負方向に調整することを意味します。負方向との角度が90°以上の軸のみが調整されます）。正方向（位置姿勢のある軸を世界

座標系の正方向に調整することを意味します。正方向との角度が90°以上の軸のみが調整されます）。

調整説明：反転する軸の方向です。例えば、**反転する軸**が**Z**で、方向タイプが**正方向**の場合、世界座標系のZ軸の正方向との角度が90°よりも小さい位置姿勢のZ軸は調整されず、角度が90°よりも大きい位置姿勢のZ軸が基準軸を中心に180°反転します。

回転の基準軸

初期値：X

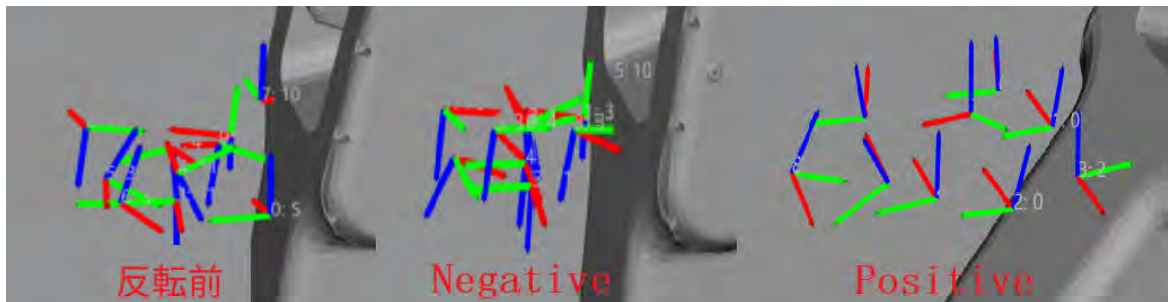
オプション：X、Y、Z

調整説明：どの軸を中心に回転するかを選択します。



反転する軸と回転の基準軸は同じ軸にしてはなりません。

調整の例：下図に示すように、ここでZ軸を反転軸、X軸を回転軸とします。



4.3.170. 実際のサイズをピクセルサイズに変換

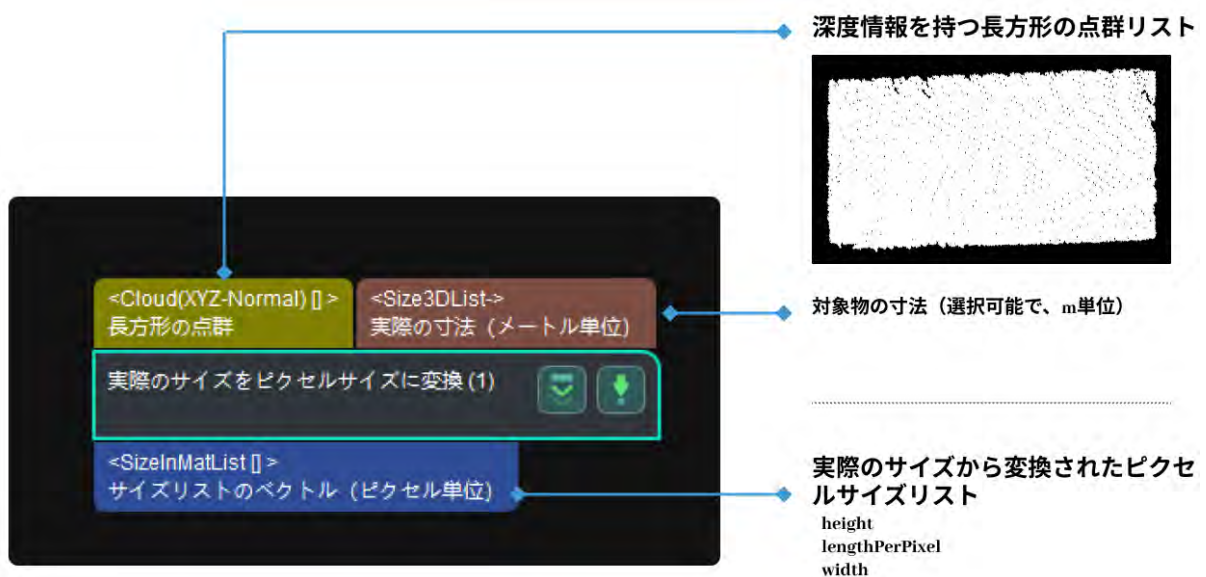
機能

長方形の点群の深度情報とその実際寸法を指定し、対応する対象物の寸法をピクセル単位で計算します。

使用シーン

その後の2Dマッチングにエッジテンプレートを提供するために、通常はステップ [\[vision-steps:generate-rectangular-edge-templates-of-specified-sizes::generate-rectangular-edge-templates-of-specified-sizes\]](#) と併用されます。

入力と出力



パラメータの説明

対応の設定

タイプ

パラメータ説明：このパラメータは、入力されたサイズと実際の点群サイズの対応タイプを設定するために使用されます。

オプション：OneActualSizeWithAllPointCloud、OneActualSizeWithOnePointCloud

- OneActualSizeWithAllPointCloud：最初の実際のサイズを、すべての点群に適用します。
- OneActualSizeWithOnePointCloud：入力されたサイズは、実際の点群に一つ一つに対応します。

初期値：OneActualSizeWithAllPointCloud 調整アドバイス：実際の状況に応じて設定してください。

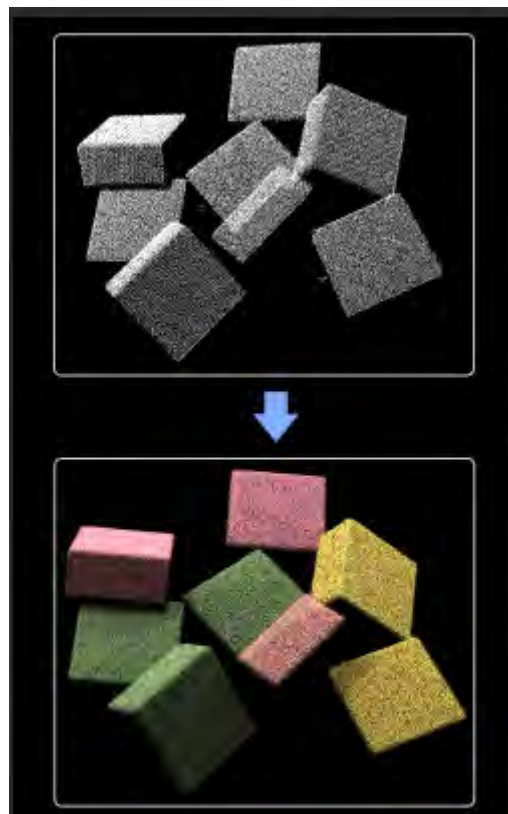
4.3.171. Cloud (XYZ-Normal)をCloud (XYZ-RGB)に変換



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

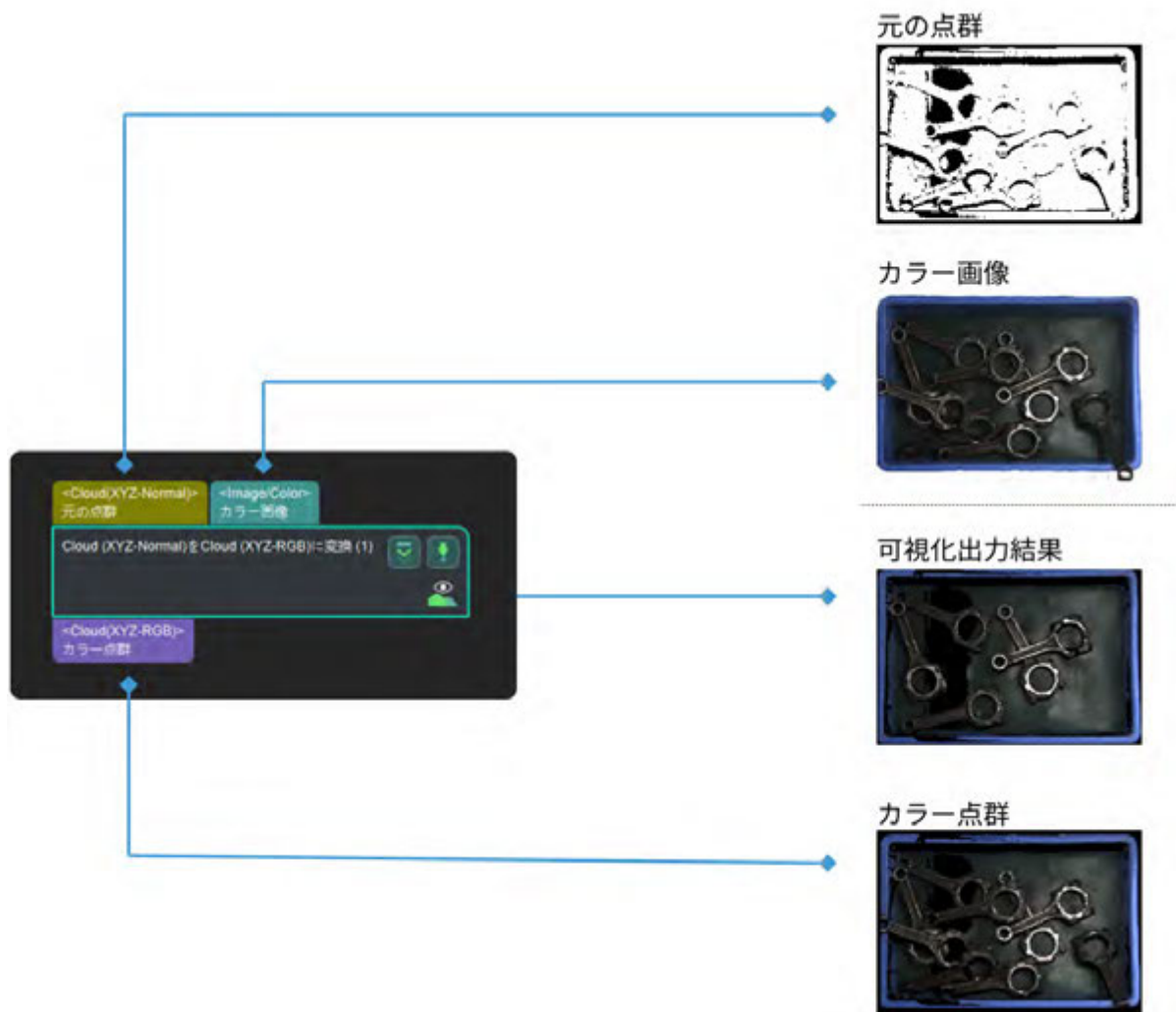
カラー画像を使用し、法線ベクトル付きの色がない点群をカラー点群に変換します。



使用シーン

仮想空間におけるカラー点群を表示するため、またその他のカラー点群を可視化する必要がある場合に使用されます。

入力と出力



4.3.172. Cloud (XYZ-RGB) をカラー画像に変換



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社 (docs@mech-mind.net) までお問い合わせください。

機能

カラー点群をカラー画像に変換します。

使用シーン

特定のシーン向けのカスタマイズのステップです。

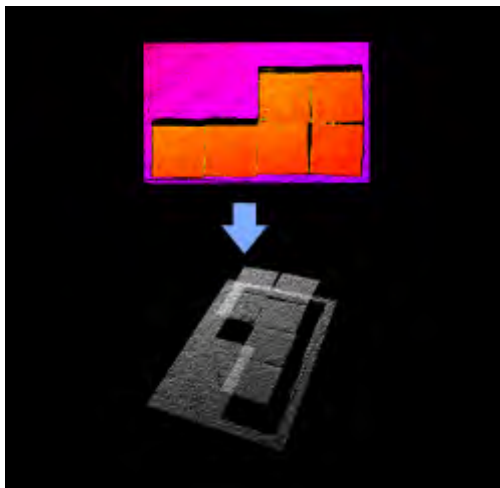
入力と出力

- 入力：カラー点群。
- 出力：カラー画像。

4.3.173. 深度画像を点群に変換

機能

このステップは、深度画像とカラー画像を使用して点群を生成できます。



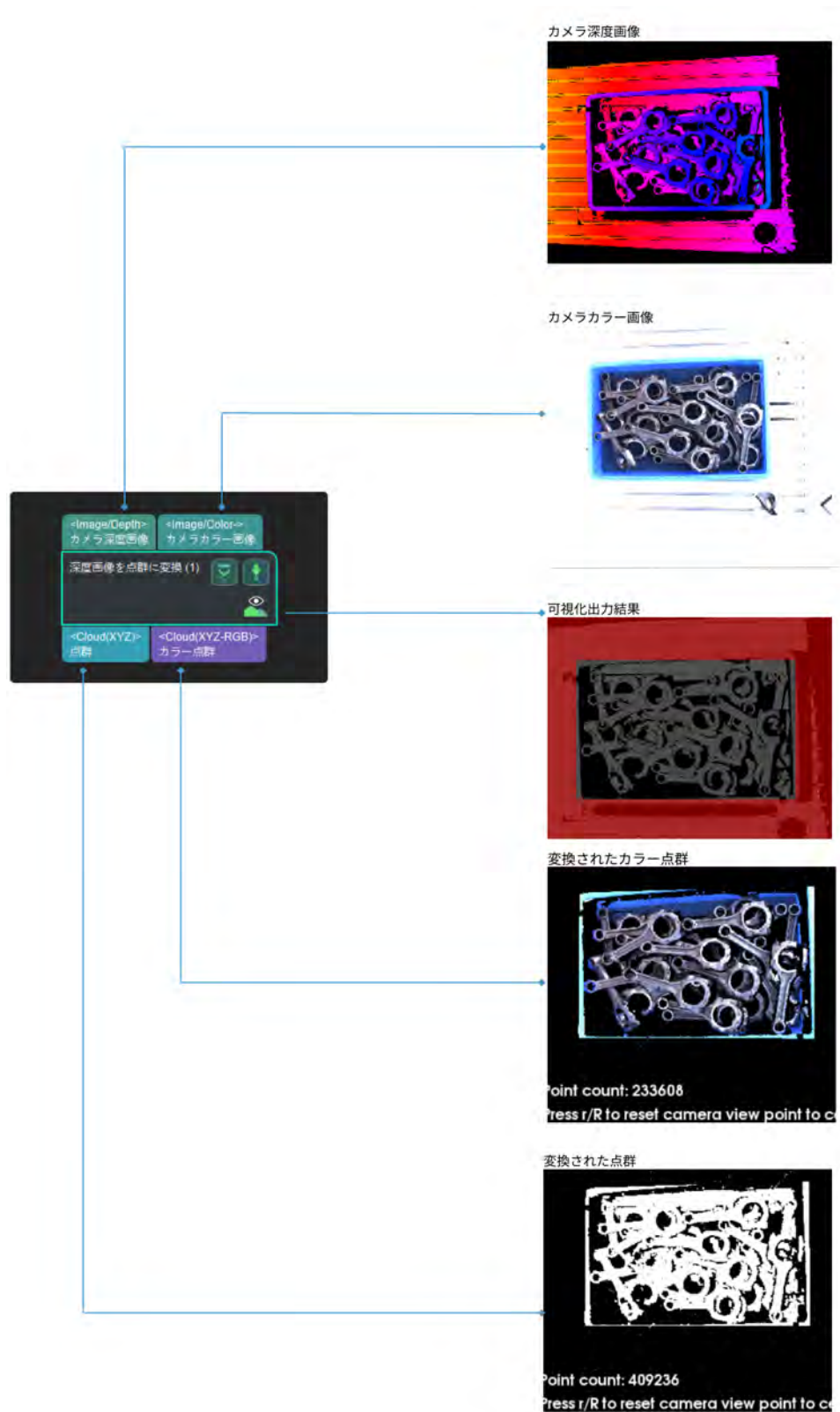
[カメラから画像を取得](#) も点群とカラー点群を直接的に出力できます。しかし、プロジェクトの実行速度を上げるために、一般的に **深度画像を点群に変換** ステップを使用して点群とカラー点群を取得します。

深度画像を点群に変換 を使用してROIを設定し、不要な点群を回避して実行の速度を向上させます。

使用シーン

通常は「カメラから画像を取得」ステップの後で使用され、カメラで取得した深度画像を点群画像に変換します。関心領域を使用して不要な点群を除去することでプロジェクトの実行速度を向上させます。

入力と出力

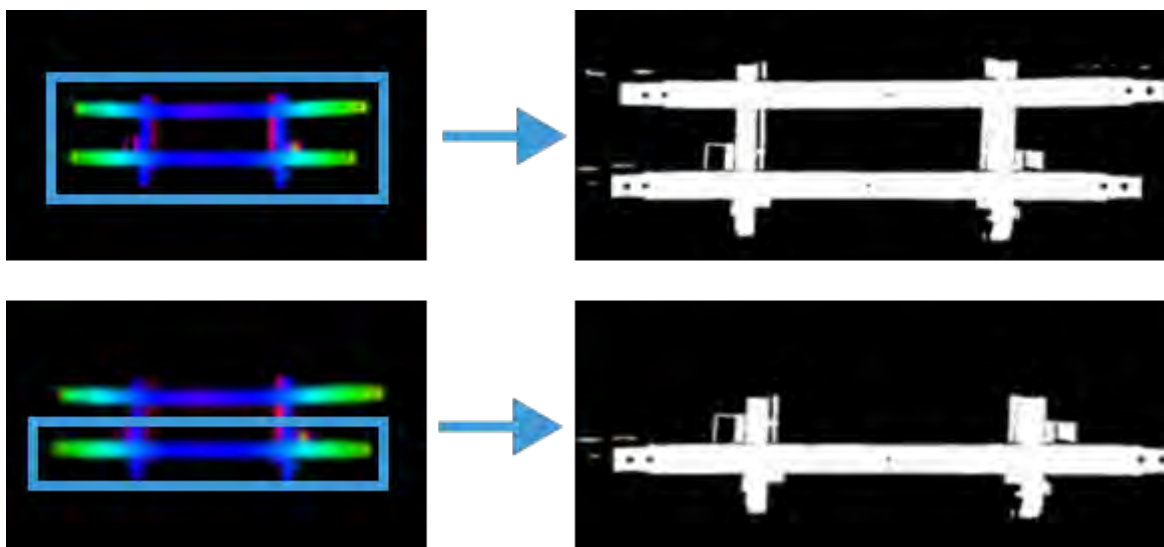


パラメータの説明

深度画像のROIファイル

調整説明：深度画像の関心領域を抽出します。

調整の例：異なるROIを設定して効果を対比します。



上図に示すように、左側は深度画像の関心領域で、右側は出力された点群です。

調整の手順：

1. 深度画像のROIファイルの右側にある **[2D ROIを設定]** ボタンをクリックして、**ROI設定** 画面に入ります。
2. 画面でROIを設定します。
3. **[確認]** をクリックして設定を保存します。同時に、プロジェクトのルートディレクトリに `depth_image_roi.json` ファイルは生成されます。

背景除去設定

深度画像による背景を除去

初期値：チェックを入れない。

調整説明： **深度画像による背景を除去** にチェックを入れてから以下のパラメータを設定可能です。

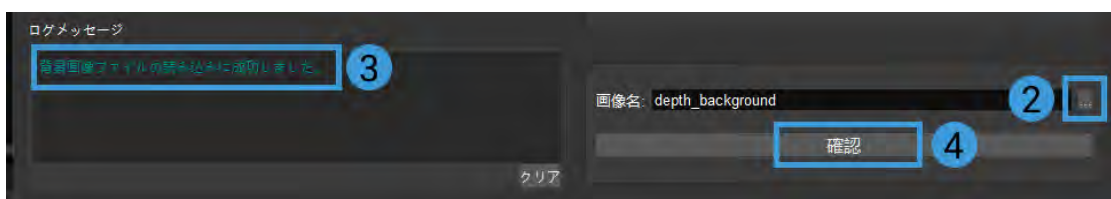
深度背景画像のファイル名

調整手順：背景深度画像設定の手順は以下の通りです。

1. 深度画像背景のファイル名の右側にある **[2D背景を設定]** ボタンをクリックして、**背景設定** 画面に入ります。



2. [...] をクリックして事前に準備された背景画像を選択します。メッセージで **背景画像ファイルの読み込みに成功しました。** が表示されたら、[確認] をクリックします。



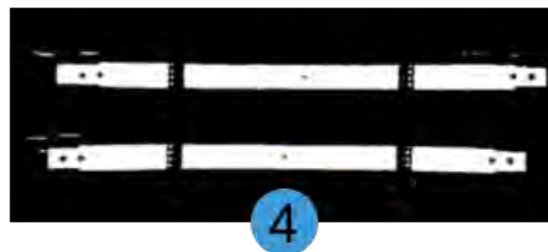
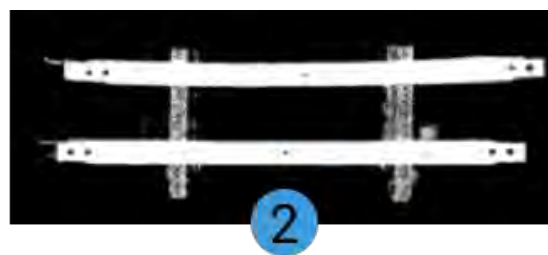
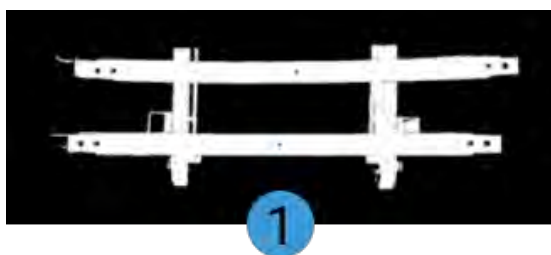
実際のカメラを接続して、現場で背景画像を取得することもできます。

背景深度の変動

初期値：10.000 mm

調整説明：深度画像と背景深度画像を入力して対比します。深度の差が変動範囲内にある背景は除去されます。

調整の例：下図では、1は背景が除去されなく、2、3、4は背景が除去され、背景深度の変動をそれぞれ1mm、10mm、30mmに設定した場合の調整効果は下図に示します。



背景深度の変動を小さく設定したら、背景は全部除去できません。背景深度の変動を大きく設定したら、部品の点群も除去されてしまいます。したがって、現場の状況に応じて背景深度の変動を適切に設定する必要があります。通常の場合は **10mm** に設定することを推奨します。

4.3.174. 視差画像によって深度画像を生成



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

視差画像を深度画像に変換します。

使用シーン

カメラのパッシブステレオモードをデバッグするための開発者機能です。普通のユーザーは、一般的な使用シーンでこのステップを使用する必要はありません。

入力と出力

- 入力：
 1. このポートに入力される視差画像は深度画像に変換されます。
 2. カメラの視差値から深度値へのマッピング行列（4 x 4行列である必要があります）。
- 出力：
 1. 視差画像から変換された深度画像。

4.3.175. NumberListをSize3DListに変換

機能

3つの数値リスト型の数値を、Size3DList型のデータに合成します。

使用シーン

汎用のデータ型変換・合成ステップです。特定の使用シーンはありません。

入力と出力

- 入力：
 1. Size3D要素を合成するためのX値リスト。
 2. Size3D要素を合成するためのY値リスト。
 3. Size3D要素を合成するためのZ値リスト。
- 出力：
 1. 入力された各X、Y、Zによって合成されたSize3Dデータのリスト。

4.3.176. NumberListをVariantListに変換

機能

NumberList型のデータをVariantList型に変換します。

使用シーン

汎用のデータ型の変換ステップです。特定の使用シーンはありません。

入力と出力

- **入力：** NumberList型のデータ。
- **出力：** 変換によって得られたVariantList型のデータ。

4.3.177. PoseListをMatrix4Dに変換

機能

PoseList型のデータをMatrix4D型に変換します。

使用シーン

汎用のデータ型の変換ステップです。特定の使用シーンはありません。

入力と出力

- **入力：** このポートに入力される位置姿勢リストデータはMatrix4D行列に変換されます。
- **出力：** 入力された位置姿勢リストデータの変換によって得られたMatrix4D行列。

4.3.178. PoseListをPoseListsに変換

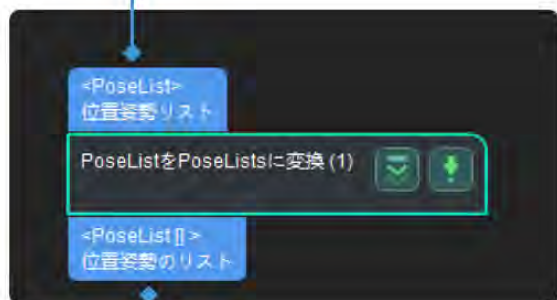
機能

位置姿勢リストの各位置姿勢を、一つの要素を含む位置姿勢リストに変えます。

使用シーン

汎用のデータ構造変換ステップです。ステップ [\[vision-steps:3d-coarse-matching:::3d-coarse-matching\]](#) と [3D位置姿勢高精度推定](#) の間もしくは [平面点群の位置姿勢とサイズを計算](#) と [フフイルタリング](#) の間などに使用されます。

入力と出力

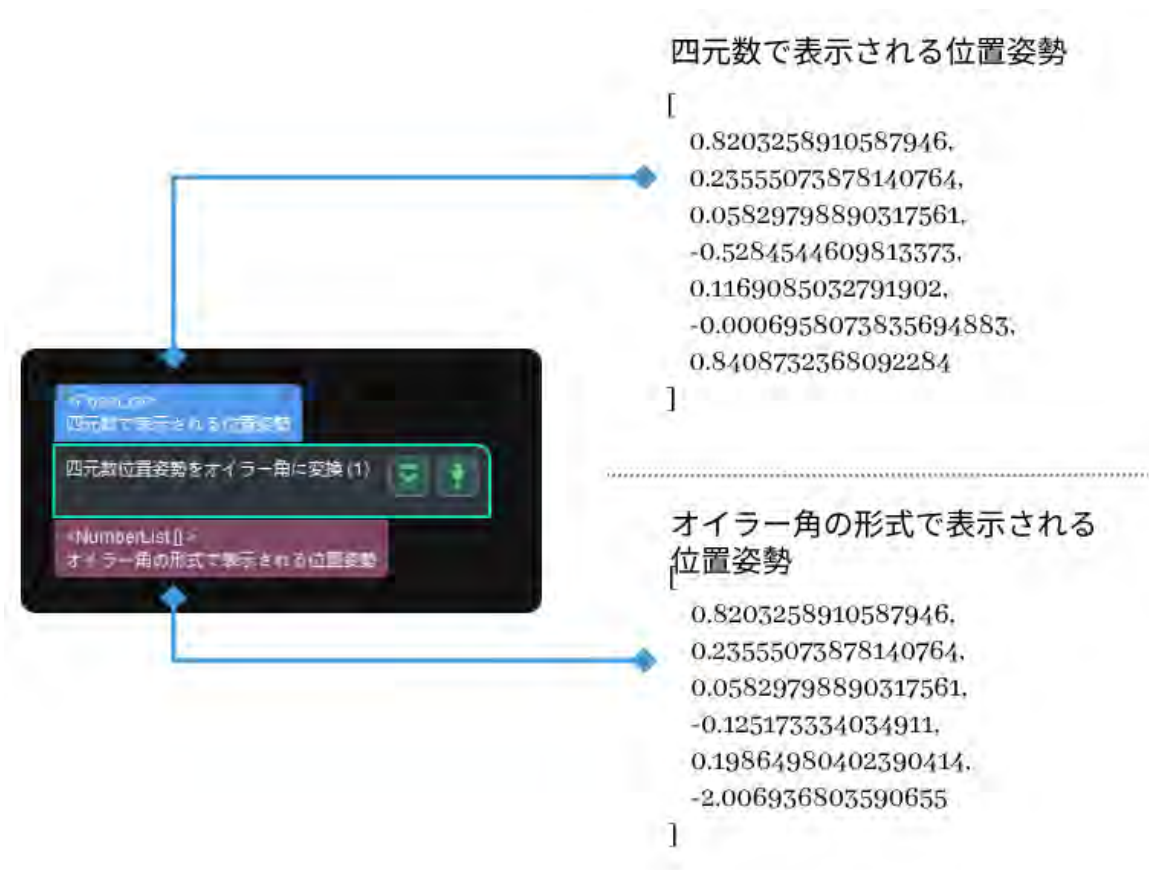


このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社 (docs@mech-mind.net) までお問い合わせください。

位置姿勢の表示方法を、四元数からオイラー角に変換します。

汎用の位置姿勢の表示方法の変換ステップで、特定の使用シーンはありません。

© 2023 Mech-Mind Robotics Technologies Ltd.



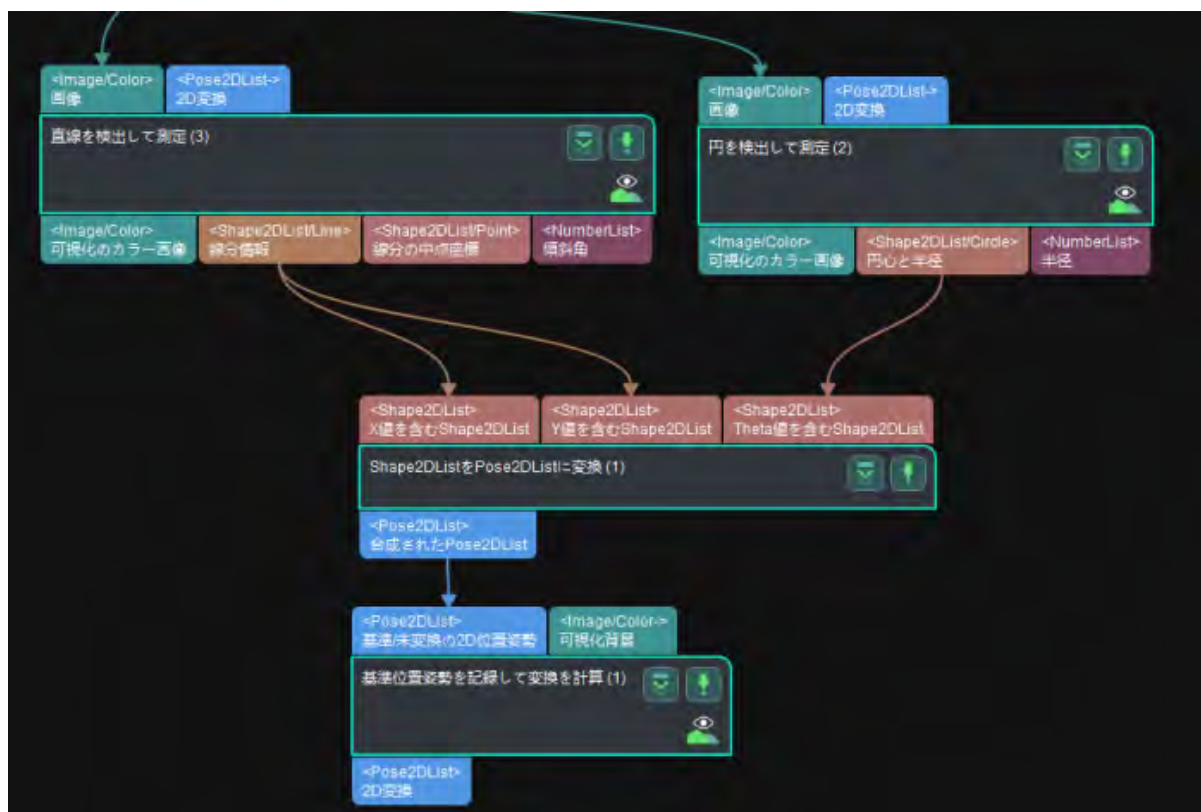
4.3.180. Shape2DListをPose2DListに変換

機能

入力された3つの2D形状情報リストから、X値リスト、Y値リスト、θ値リスト（θは傾斜角度）を取り出し、組み合わせて新しい2D位置姿勢リストを作成します。

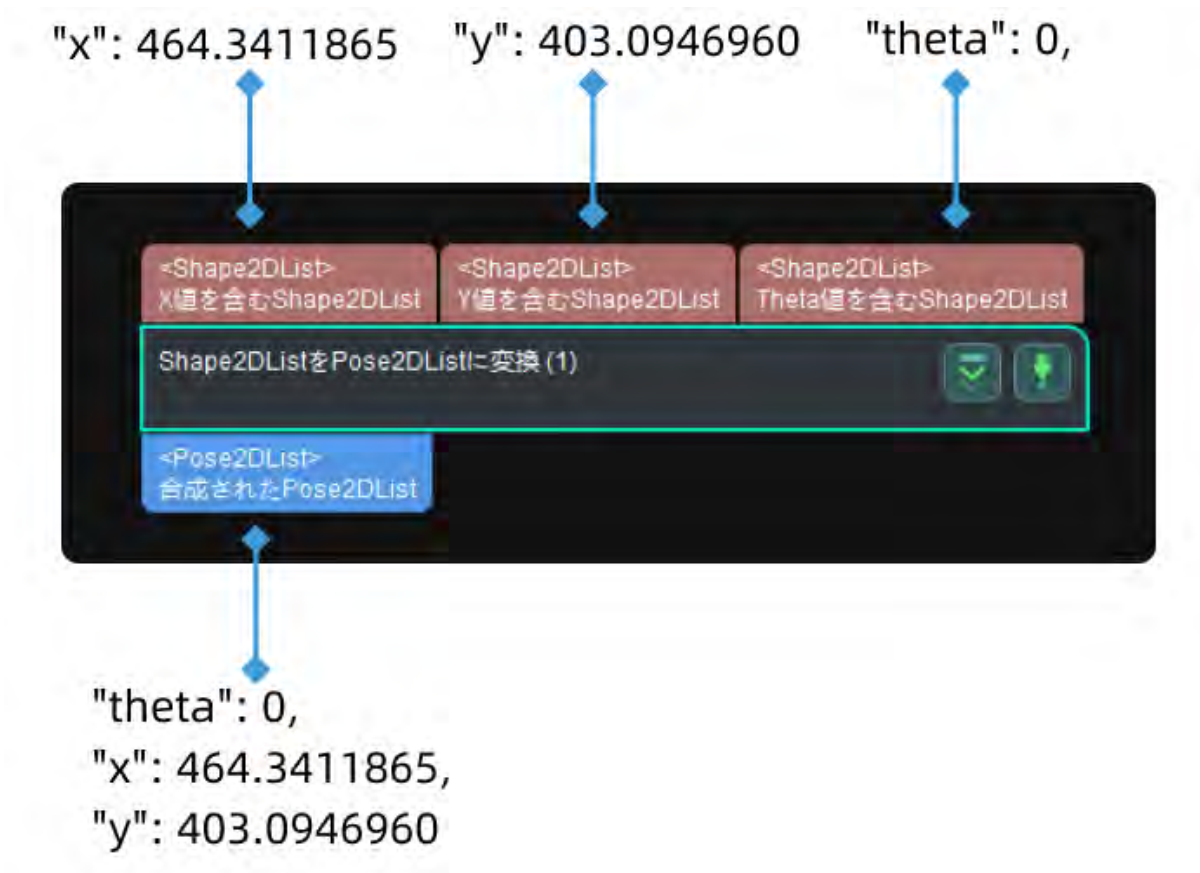
使用シーン

このステップの前には通常、[\[vision-steps:detect-and-measure-line:::detect-and-measure-line\]](#)（直線の傾斜角度θを出力）と [円を検出して測定](#)（円の中心のX、Y座標を出力）のステップがあり、画像中の対象物のより正確な2D位置位置情報を出力し、それを次でつながれる [\[vision-steps:record-criterion-pose-and-calc-transformation:::record-criterion-pose-and-calc-transformation\]](#) ステップに送信し、画像中の対象物を正確に合わせるために使用されます。



上記の操作は、通常 [\[vision-steps:template-matching:::template-matching\]](#) ステップの後に、[\[vision-steps:template-matching:::template-matching\]](#) の結果をより正確に調整するために使用されます。各画像の対象物が既にほぼ合わせる場合は、[\[vision-steps:template-matching:::template-matching\]](#) ステップを使用せずに、上記の操作を直接行うことができます。

入力と出力



パラメータの説明

調整可能なパラメータはありません。

4.3.181. VariantをVariantListに変換

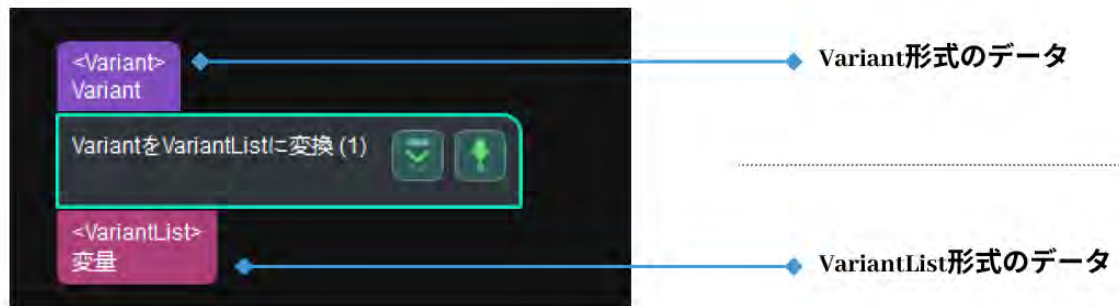
機能

入力を、VariantからVariantListに変換します。

使用シーン

汎用のデータ型変換のステップです。例えば、ステップ [\[vision-steps:count-elements-in-specified-dimension-in-data-list:::count-elements-in-specified-dimension-in-data-list\]](#) と [VariantListをNumberListに変換](#) の間に使用することで、要素数を数値リストに変換して出力することができるようになります。

入力と出力



4.3.182. 任意の変数からラベルに変換

機能

任意のフィールド形式で表示されるデータ型を、ラベルのデータ型に変換します（現在、DoubleList、QtVariantListの入力のみに対応します）。

使用シーン

汎用のデータ型の変換ステップです。特定の使用シーンはありません。

入力と出力

- **入力：** このポートに入力される点群はラベルに変換されます。
- **出力：** 変換によって得られたラベル。

4.3.183. VariantListをNumberListに変換

機能

入力データ型を、VariantListからNumberListに変換します。

使用シーン

汎用のデータ型の変換ステップです。特定の使用シーンはありません。

入力と出力

- **入力：** VariantList型のデータ。
- **出力：** 入力の変換によって得られたNumberList型のデータ。

4.3.184. VariantListをVariantに変換

機能

VariantList型のデータをVariant型に変換します。入力リストに複数の要素があれば、最初の要素が使用されます。

使用シーン

汎用のデータ型の変換ステップです。特定の使用シーンはありません。

入力と出力

- **入力：** このポートに入力されるVariantList型のデータは、Variant型のデータに変換されます。
- **出力：** 変換によって得られたVariant型のデータ。

4.3.185. 3D位置姿勢を2Dに変換



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

3D位置姿勢を2D位置姿勢に変換します。カメラの内部パラメータを考慮します。

使用シーン

汎用の位置姿勢変換ステップです。特定の使用シーンはありません。

入力と出力

- **入力：** このポートに入力される3D位置姿勢は、変換されます。
- **出力：** 変換によって得られた2D位置姿勢。

4.3.186. 点群で仮想側壁を生成



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

指定方向に沿って入力点群をコピーし、点群で仮想側壁を生成します。

使用シーン

特定のシーン向けのカスタマイズのステップです。

入力と出力

● 入力：

1. このポートに入力される点群は、点群で仮想側壁を生成するためにコピーれます。
2. 仮想側壁の高さ。
3. 上表面の点群をコピーするための指定方向（オプション。このポートが使用されていない場合は、パラメータで設定された方向が使用されます）。

● 出力：

1. 生成された仮想側壁。

4.3.187. 円柱形の点群モデルを生成



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

円柱形状の点群を生成します。円柱形状は、数値を入力するか、ステップパラメータで設定できます。

使用シーン

円柱形状の点群を自動的に生成するために使用されます。これは、その後のモデルとして使用されます。

入力と出力

● 入力：

1. 円柱形状の点群モデルの半径（オプション。このポートが使用されていない場合、パラメータの設定が使用されます）。
2. 円柱形状の点群モデルの長さ（オプション。このポートが使用されていない場合、パラメータの設定が使用されます）。

● 出力：

1. 生成された円柱形状の点群。

4.3.188. 基準位置姿勢を中心に回転した離散位置姿勢を生成



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

基準位置姿勢を中心に初期の位置姿勢を回転し、1つセットの位置姿勢を生成します。

使用シーン

汎用の位置姿勢生成ステップです。特定の使用シーンはありません。

入力と出力

● 入力：

1. 1つセットの位置姿勢を形成するために、このポートに入力される初期の位置姿勢が回転されます。
2. このポートに入力される位置姿勢は回転の基準位置姿勢として使用されます。

● 出力：

1. 基準位置姿勢を中心に初期の位置姿勢を回転させて形成した1つセットの位置姿勢。

4.3.189. 把持位置姿勢を生成



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

2Dテンプレート画像で把持位置姿勢を生成します。

使用シーン

通常、鋼板の把持シーンで、定義された吸盤の形状およびサイズに基づいて対応する平面の鋼板の把持位置姿勢を生成するために使用されます。

入力と出力

● 入力：

1. このポートに入力される対象物のテンプレートの二値化マスク画像は、把持位置姿勢を生成するために使用されます。
2. このポートに入力される吸盤の領域の二値化マスク画像は、把持位置姿勢を生成するために使用されます（対象物のテンプレートと吸盤の領域マスクとの重複は設定されたしきい値よりも小さい場合は、把持位置姿勢が生成されません）。

● 出力：

1. 生成された把持位置姿勢。

4.3.190. 点群モデルを生成



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

2D画像を3D点群に変換します。

使用シーン

通常、鋼板の把持シーンで2D画像を3D点群に変換するために使用されます。それは、後のモデルとして使用されます。

入力と出力

- **入力：** モデルを生成するための2D画像。
- **出力：** 2D画像によって生成された3D点群モデル。

4.3.191. リング形状の点群を生成



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

リング形状の点群を生成します。リングのサイズは、ステップパラメータで設定できます。

使用シーン

リング形状の点群を自動的に生成するために使用されます。それは、その後のモデルとして使用されます。

入力と出力

- **入力：** なし。
- **出力：** 生成されたリング形状の点群（法線ベクトル付き）。

4.3.192. 位置姿勢のオフセットを生成



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

対象物の元の位置姿勢を調整するために位置姿勢のオフセットを生成します。また、対象物範囲内のオフセット位置姿勢のみが保持されます。

使用シーン

特定のシーン向けのカスタマイズのステップです。

入力と出力

● 入力：

1. このポートに入力される元の位置姿勢は、オフセットされます。
2. 位置姿勢が対応する対象物の寸法リスト。

● 出力：

1. オフセット後の元の位置姿勢リスト。
2. オフセット位置姿勢リスト（パラメータの設定によって異なります）。
3. 位置姿勢が対応する対象物のインデックスリスト。1つの対象物は複数の位置姿勢（複数の把持位置姿勢）がある可能性があります。

4.3.193. 長方形の経路を生成



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

入力される対象物の点群によって長方形の経路を生成します。

使用シーン

通常、長方形のドア/ガラスなど、エッジ接着の応用シーンで使用されます。

入力と出力

● 入力：

1. 対象物の点群。
2. 対象物を含むシーンの点群。
3. 長方形の経路の開始点。

● 出力：

1. 長方形の経路上の点のリスト。
2. 経路の2D正投影図。

4.3.194. 長方形候補を生成



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

対象物のピクセルサイズおよびコーナー情報（位置と方向）によって候補の長方形の結果を生

成します。これは、後続のステップで検証する必要があります。

使用シーン

2Dマッチングが必要なテンプレートを生成するために使用されます。

入力と出力

● 入力：

1. 実際の対象物の寸法リストによって生成されたピクセルサイズリスト。
2. 入力されるサイズが対応するコーナーリスト（サイズとコーナーは、同じマスクまたは同じ点群から検出されます）。

● 出力：

1. 生成された長方形の頂点。

4.3.195. 指定したサイズの長方形のエッジテンプレートを生成

機能

対象物の寸法（ピクセル単位）を指定することで長方形のエッジテンプレートリストを生成することができます。これは通常、ステップ [2Dマッチング](#) の前に使用されます。

使用シーン

ステップ [2Dマッチング](#) のエッジテンプレートを生成するために、ステップ [\[vision-steps:from-actual-dimensions-to-dimensions-in-pixels:::from-actual-dimensions-to-dimensions-in-pixels\]](#) と併用することになっています。

入力と出力



パラメータの説明

テンプレートのプロパティ

テンプレートの対角線を定量化した後のセグメント数

調整説明：このパラメータは、テンプレートの対角線を定量化した後のセグメント数を設定するために使用されます。長方形のテンプレートの対角線の長さをテンプレートの対角線を定量化した後のセグメント数で割って、エッジ直線の定量化されたピクセルの数を取得します。

初期値：10

推奨値：[10,15]

有効範囲：[0,+∞)

4.3.196. 螺旋状経路を生成



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

入力点群に基づいて螺旋状の経路点を生成します。

使用シーン

通常、平面の物体への自動スプレーに使用されます。

入力と出力

● 入力：

1. このポートに入力される点群内の対象物の表面は、螺旋形状の経路点が生成されます。
2. 点群の深度画像。
3. フランジ座標系におけるロボットのTCPの位置姿勢。

● 出力：

1. 生成された経路を表示するための外側から内側への点（最後の点を除く）。
2. 経路の最後の点。
3. 経路の正投影2D画像。

4.3.197. テスト画像を生成



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

指定されたフォーマットとサイズの画像を生成します。

使用シーン

開発者のテスト用のステップです。普通のユーザーは、一般的な使用シーンでこのステップを使用する必要はありません。

入力と出力

- 入力：なし。
- 出力：生成された画像。

4.3.198. テスト点群を生成

機能

点群データを生成します。これは、ほかの機能をテストするために使用されます。

使用シーン

開発者のテスト用のステップです。普通のユーザーは、一般的な使用シーンでこのステップを使用する必要はありません。

入力と出力

- 入力：なし。
- 出力：生成された点群。

4.3.199. 輪郭から経路を生成

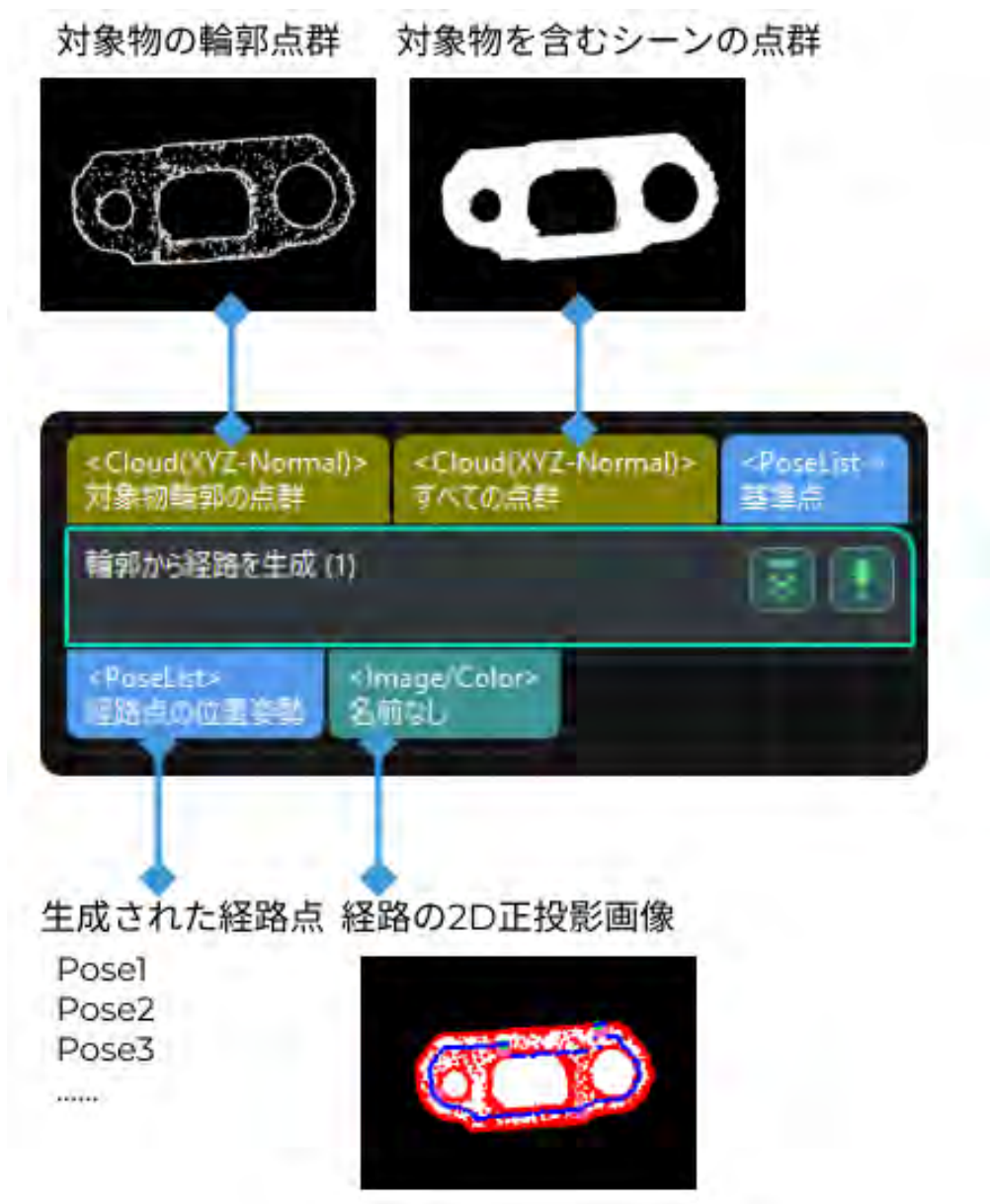
機能

入力された輪郭点群によって経路を生成します。

使用シーン

通常、長方形のドア/ガラスなど、エッジ接着の応用シーンで使用されます。

入力と出力



パラメータの説明

このパラメータには、以下の4つのパラメータ群があります。

- **一般設定**：このパラメータ群は、一般的な場合に使用されます。実際の状況に応じて設定してください。
- **エッジポイントの平滑化設定**：このパラメータ群は、部品のエッジ点群がギザギザに見える場合に調整する必要があります。
- **特殊場合向け収縮設定**：このパラメータ群は、特殊なプロジェクトでのみ使用され、一般に調整する必要はありません。
- **コーナー検出**：部品のエッジ点群に明らかなコーナーがある場合に調整される必要があります。

共通設定

マスククロージング処理のカーネルサイズ

パラメータ説明：このパラメータは、画像マスクに対してクロージング処理を行う際のカーネルのサイズ（ピクセル単位）を設定するために使用されます。

初期値：5 px

経路点の間隔

パラメータ説明：このパラメータは、生成された経路点の間隔（ピクセル単位）を設定するために使用されます。この値が大きいほど、生成された経路点の間隔が大きくなり、値が小さいほど、生成された経路点の間隔が小さくなります。調整効果を [調整の例](#) に示します。

初期値：10 px

エッジのスケーリング方法

パラメータ説明：このパラメータは、経路点を生成する際にエッジ点群のスケーリング方法を選択するために使用されます。

オプション：収縮、エッジポイントへの接線の垂直方向に沿ったスケーリング。

- 収縮：パラメータ **収縮カーネルサイズ** の設定により、スケーリングの距離と方向を調整します。
- エッジポイントへの接線の垂直方向に沿ったスケーリング：パラメータ **垂直方向の移動距離** の設定により、スケーリングの距離と方向を調整します。

初期値：収縮。

調整アドバイス：**収縮** を使用することを推奨します。

収縮カーネルサイズ

パラメータ説明：このパラメータは、**エッジのスケーリング方法** が **収縮** に設定された場合に、経路のスケーリングスケールと方向（ピクセル単位）を設定するために使用されます。この値が正の場合、経路は部品エッジの内側に向かって縮小し、値が大きくなるほど縮小します。この値が負の場合、経路は部品エッジの外側に向かって拡大し、値が小さいほど広がります。調整効果を [調整の例](#) に示します。

初期値：15px

垂直方向の移動距離

パラメータ説明：このパラメータは、**エッジのスケーリング方法** が **エッジポイントへの接線の垂直方向に沿ったスケーリング** に設定された場合に、スケーリングスケールと方向（ミリメートル単位）を設定するために使用されます。この値が正の場合、経路は部品エッジの内側に向かって縮小し、値が大きくなるほど縮小します。この値が負の場合、経路は部品エッジの外側に向かって拡大し、値が小さいほど広がります。調整効果を [調整の例](#) に示します。

初期値：8.000 mm

エッジポイントの平滑化設定

ガウシアンフィルタを使用

パラメータ説明：このパラメータは、部品のエッジ点群がギザギザに見える場合にガウシアンフィルタでエッジ点群を平滑化するかどうかを決定するために使用されます。

初期値：チェックを入れない。

調整アドバイス：実際の状況に応じて設定してください。

ガウシアンフィルタ半径

パラメータ説明：このパラメータは、ガウシアンフィルタを使用する場合にフィルタ半径（ピクセル単位）を設定するために使用されます。

初期値：20px

調整アドバイス：初期値を使用ことを推奨します。

ガウシアンフィルタのシグマ値

パラメータ説明：このパラメータは、ガウシアンフィルタを使用する場合にシグマ値を設定するために使用されます。

初期値：5.0000

調整アドバイス：初期値を使用ことを推奨します。

特殊場合向け収縮設定

このパラメータ群は、特殊なプロジェクトでのみ使用され、一般に調整する必要はありません。

コーナー検出

エッジ点群のダウンサンプリング係数

パラメータ説明：このパラメータは、部品のエッジ点群をダウンサンプリングするときのダウンサンプリング係数（ピクセル単位）を設定するために使用されます。部品のエッジ点群のダウンサンプリングにより、コーナーの検出が容易になります。

初期値：3

品質レベル

パラメータ説明：このパラメータは、対象物のコーナーを検出するために使用され、経路点の間隔がどのように調整されても、エッジのコーナーに経路点が存在することを保証します。

初期値：0.0100

調整アドバイス：初期値を使用ことを推奨します。

コーナー間の最小ピクセル

パラメータ説明：このパラメータは、コーナー検出時にコーナー間の最小ピクセル数を設定するために使用されます。

初期値：50.0000

コーナーの最大数

パラメータ説明：このパラメータは、対象物エッジのコーナーの最大数を設定するために使用されます。

初期値：4

検出領域サイズ

パラメータ説明：このパラメータは、部品の点群エッジにある目立たないコーナーを検出するために使用されます。点群のエッジが滑らかである場合に上げることができます。

初期値：10

Harris検出方法を使用

パラメータ説明：このパラメータは、Harris検出方法を使用するかどうかを設定するために使用されます。チェックを入れた後、**K** 値を設定する必要があります。

初期値：チェックを入れない。

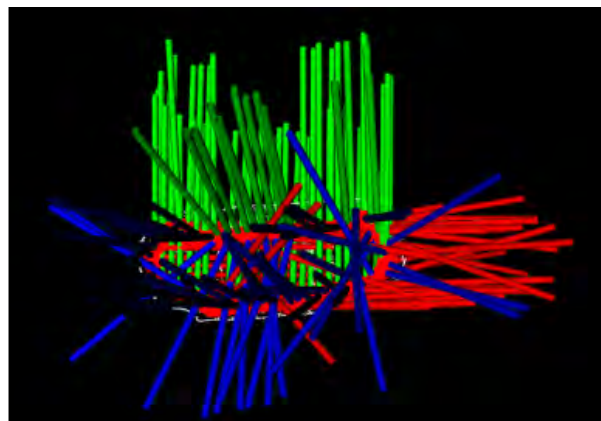
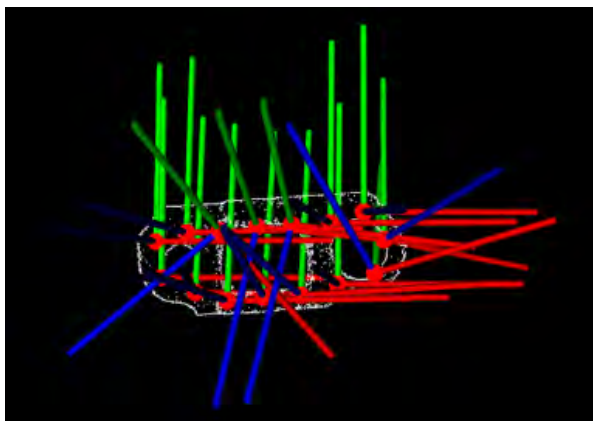
調整アドバイス：一般的には、この方法を使用する必要はありません。

調整の例

経路点の間隔

経路点の間隔をそれぞれ **20** と **5** に設定した場合、このステップによって生成された経路点を下図に示します。

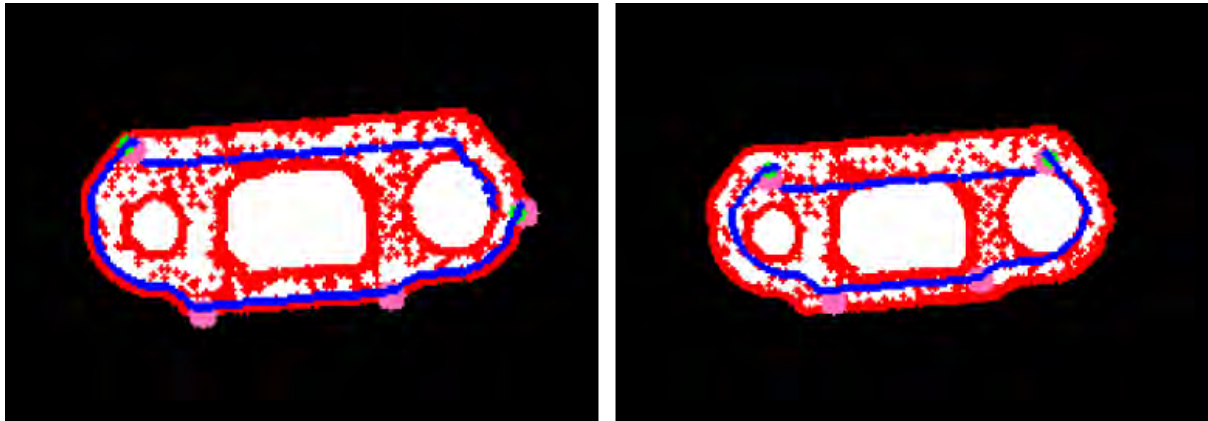
左側のは **20** に設定した場合の効果で、経路点がまばらになっています。右側のは **5** に設定した場合の効果で、経路点が密集しています。



収縮カーネルサイズ

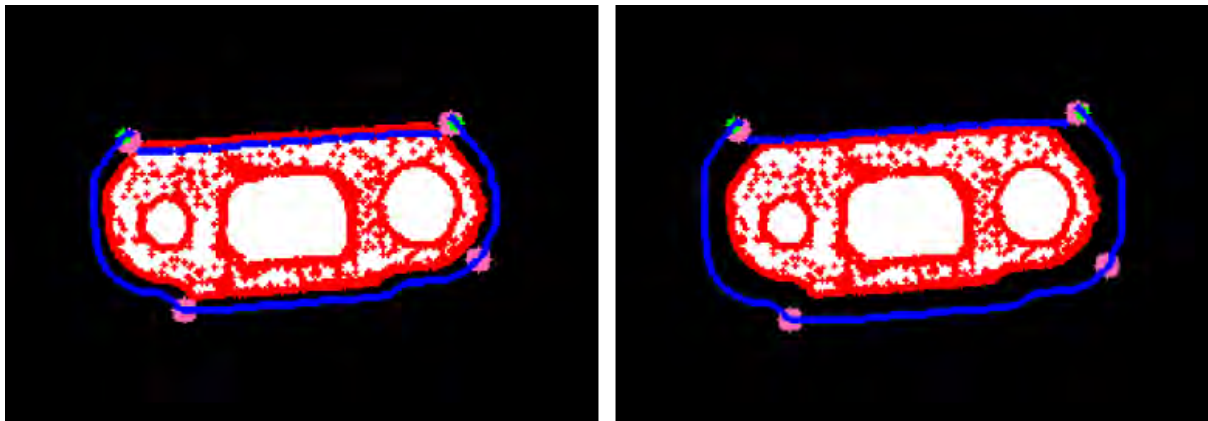
収縮カーネルサイズ をそれぞれ **5** と **15** に設定した場合、このステップによって生成された経路を下図に示します。

左側のは **5** に設定した場合の効果で、右側のは **15** に設定した場合の効果です。この正の数の値が大きくなると、経路は部品エッジの内側に向かって縮小します。



収縮カーネルサイズ をそれぞれ **-5** と **-20** に設定した場合、このステップによって生成された経路を下図に示します。

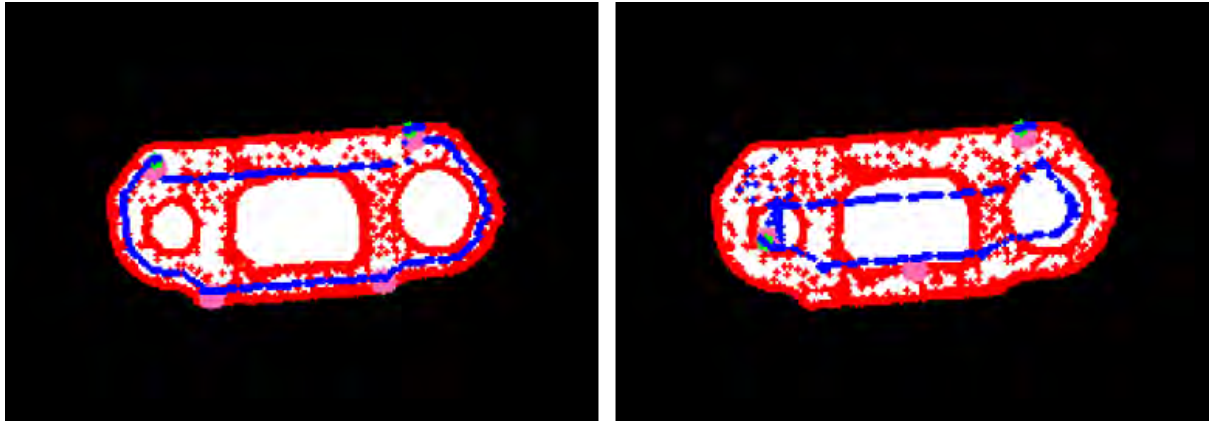
左側のは **-5** に設定した場合の効果で、右側のは **-20** に設定した場合の効果です。この負の値が小さくなると、経路は部品エッジの外側に向かって拡大します。



垂直方向の移動距離

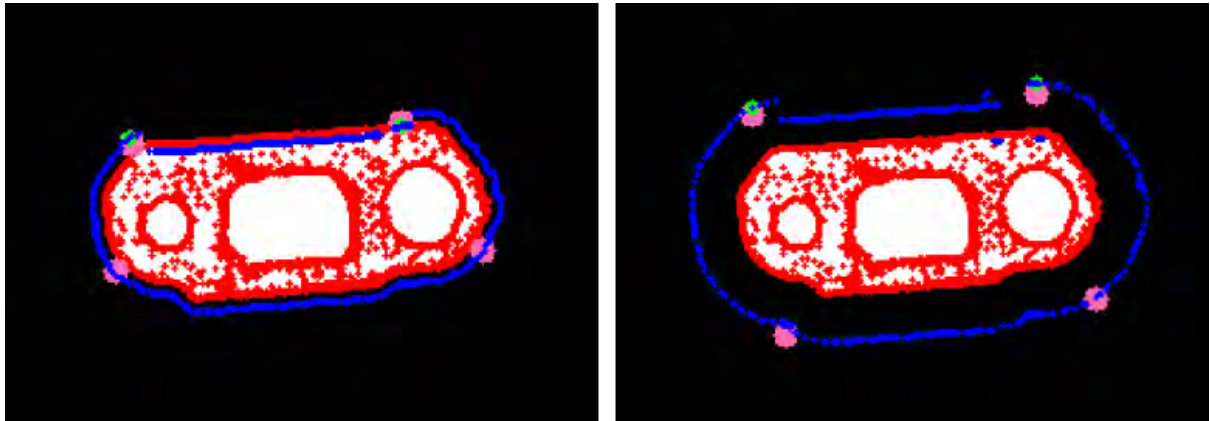
垂直方向の移動距離 をそれぞれ **5** と **15** に設定した場合、このステップによって生成された経路を下図に示します。

左側のは **5** に設定した場合の効果で、右側のは **15** に設定した場合の効果です。この正の数の値が大きくなると、経路は部品エッジの内側に向かって縮小します。



垂直方向の移動距離をそれぞれ **-5** と **-20** に設定した場合、このステップによって生成された経路を下図に示します。

左側のは **-5** に設定した場合の効果で、右側のは **-20** に設定した場合の効果です。この負の値が小さくなると、経路は部品エッジの外側に向かって拡大します。



4.3.200. 深度画像から経路を生成



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社 (docs@mech-mind.net) までお問い合わせください。

機能

対象物のエッジ点群および深度画像に基づいて、内側へ折り返される経路点を生成します。

使用シーン

通常、平らな対象物のエッジを接着するために使用されます。

入力と出力

- 入力：
 1. 対象物のマスク。

2. シーンの深度画像。

3. 対象物の点群。

● 出力：

1. 生成された経路を表示するための目標点。

4.3.201. ジグザグ経路を生成



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社 (docs@mech-mind.net) までお問い合わせください。

機能

入力点群によってジグザグの経路を生成します。

使用シーン

通常、平面の物体への自動スプレーに使用されます。

入力と出力

● 入力：

1. 対象物のマスク。
2. シーンの深度画像。
3. 対象物の点群。

● 出力：

1. 生成された経路を表示するための目標点。

4.3.202. 最初の画像を取得



このステップは古いバージョンであり、メンテナンスが停止されていますので、[出力の数を制限](#)、[アンパック](#) を組み合わせて使用することをお勧めします。このステップに関する詳細が必要な場合は、当社 (docs@mech-mind.net) までお問い合わせください。

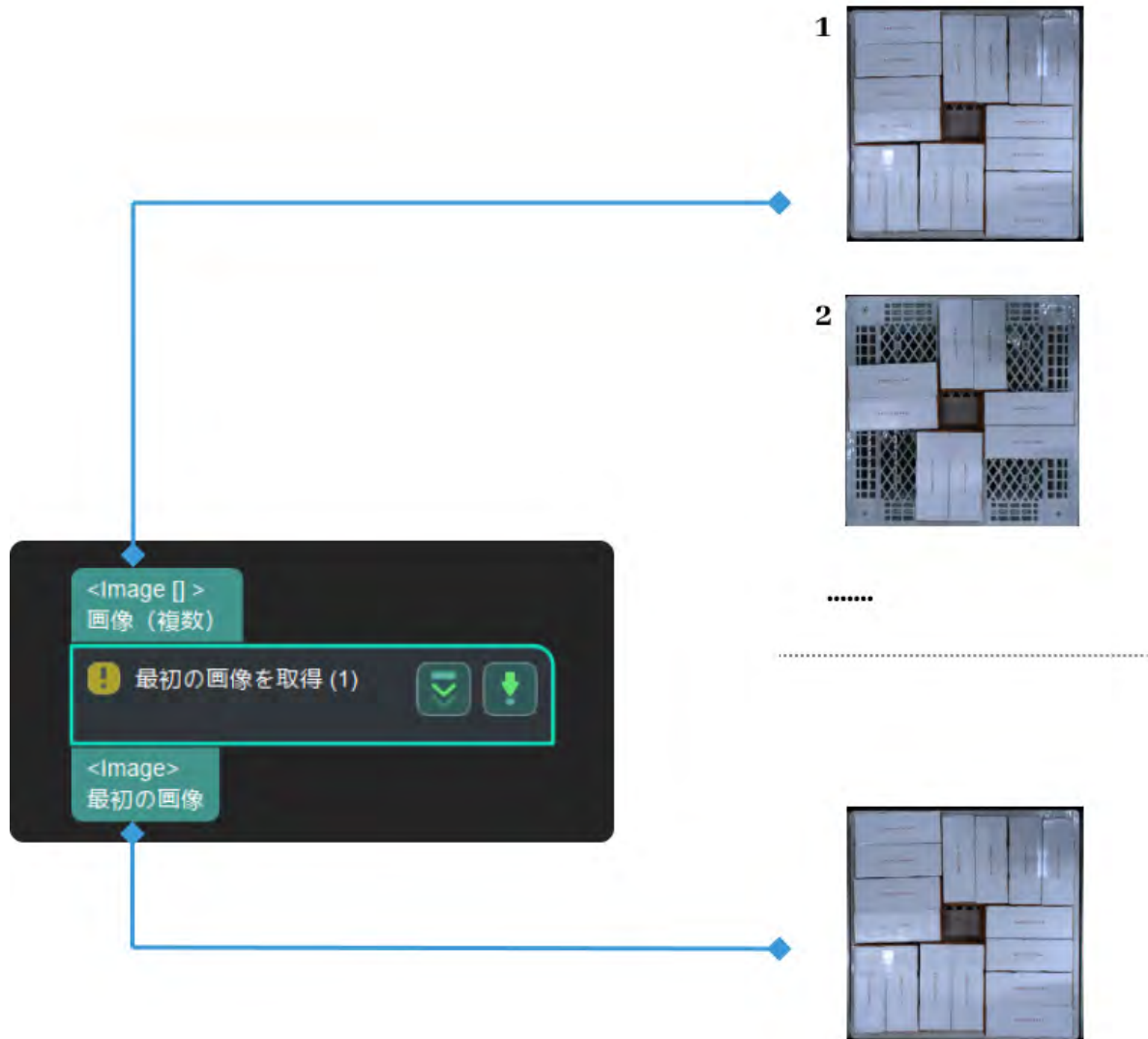
機能

画像リストの一番目の画像を取得します。

使用シーン

このステップは古いバージョンであり、新しいバージョンのステップを使用してください。

入力と出力



4.3.203. フランジ位置姿勢を取得

機能

ロボットのフランジ位置姿勢を取得します。

使用シーン

Eye In Handモード（1つのカメラの取り付け方式）で、カメラが画像を撮影しているときのロボットのフランジ位置姿勢を取得します。

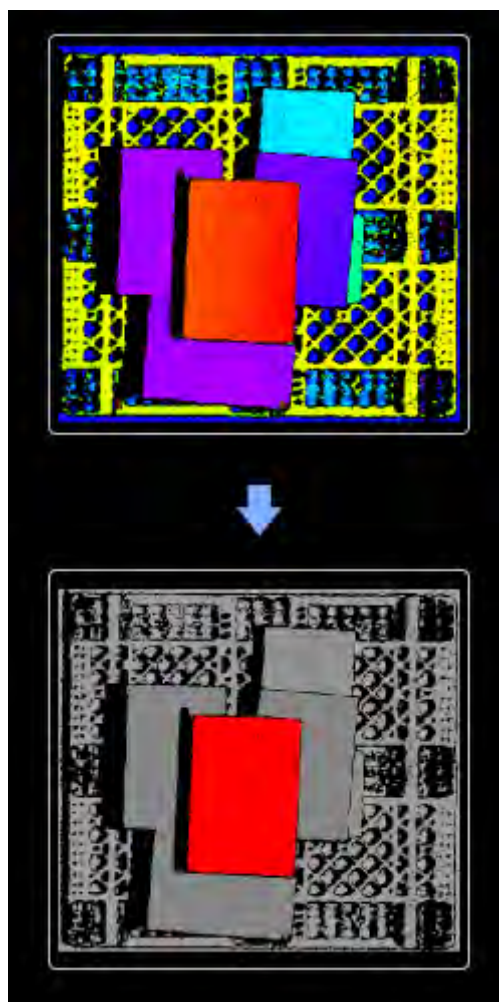
入力と出力

- 入力：なし。
- 出力：取得されたロボットのフランジ位置姿勢。

4.3.204. 深度画像で最高領域を取得

機能

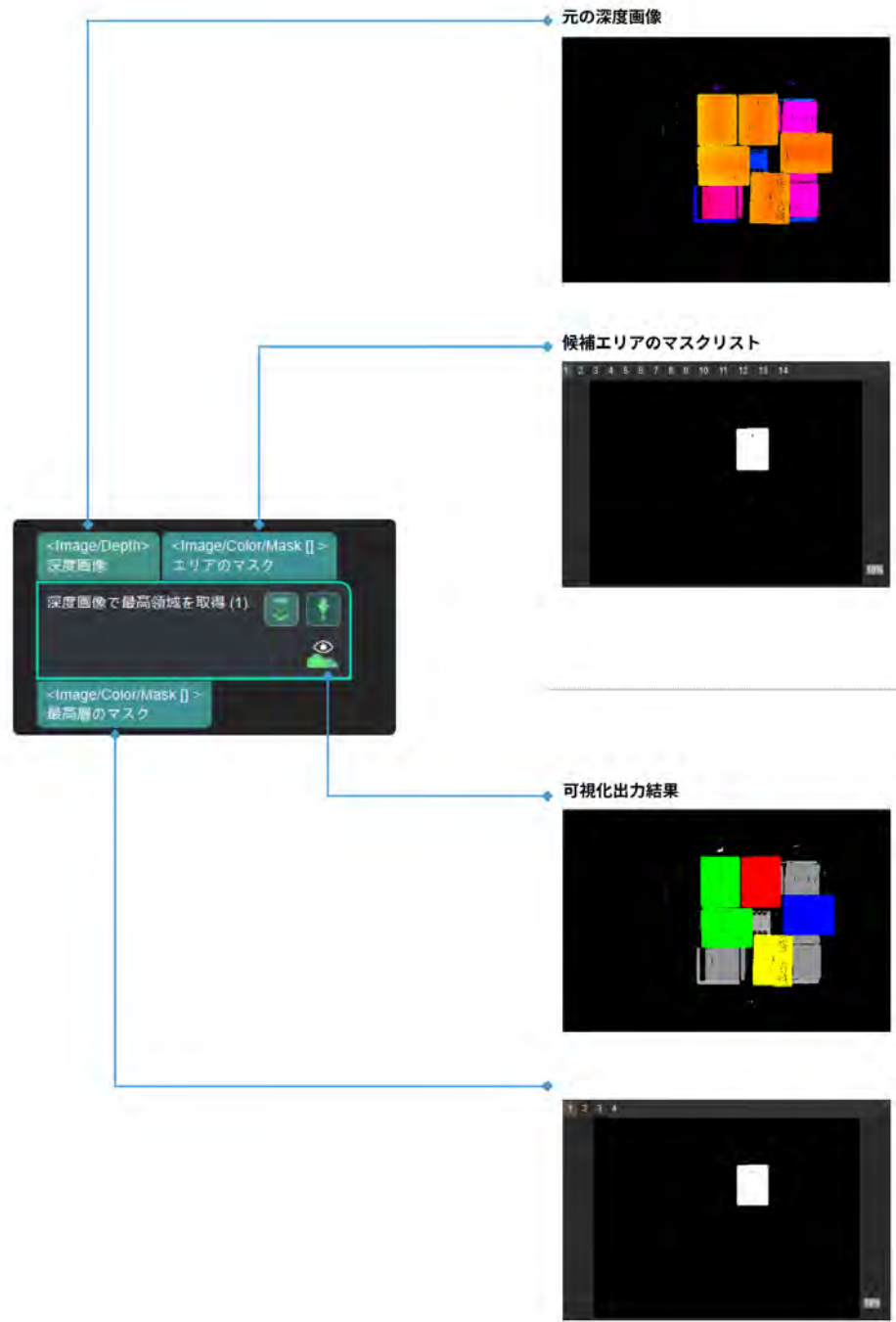
指定された深度画像とマスクリストを使用して画像を複数の領域に分割して、カメラに最も近い領域が保持され、残りは破棄されます。



使用シーン

このステップは、通常、段ボール箱のデパレタイジングシーンに最高層の対象物を認識するために使用されます。通常、[深度画像3D ROI外の領域を無効に設定](#)、[マスク画像をマージ](#)、[深度画像分割](#)などのステップと併用されます。

入力と出力



パラメータの調整説明

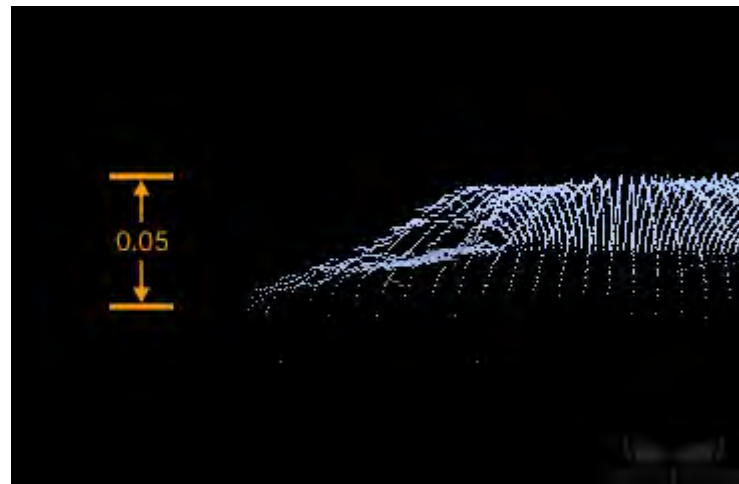
最高層の設定

層の高さ

初期値：50mm

調整説明：このパラメータは、取得された最高層領域の高さ範囲を示します。最高層の物体の縦方向の高さ範囲を設定することで（下図のように）、高さ範囲内の物体のマスクが出力されます。この高さの範囲内に他の層の物体がある場合は、この層の高さ範囲に最高層の物

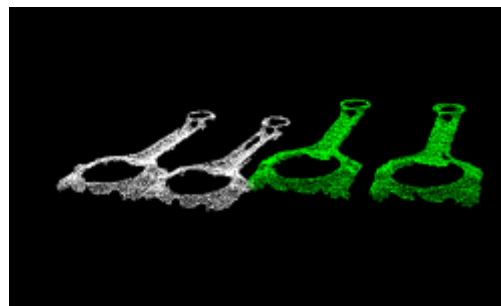
体のみが含まれるように、層の高さを調整する必要があります。



4.3.205. 最高層の点群を取得

機能

各点群を設定された方向に沿って降順でソートし、最高層の点群を取得します。



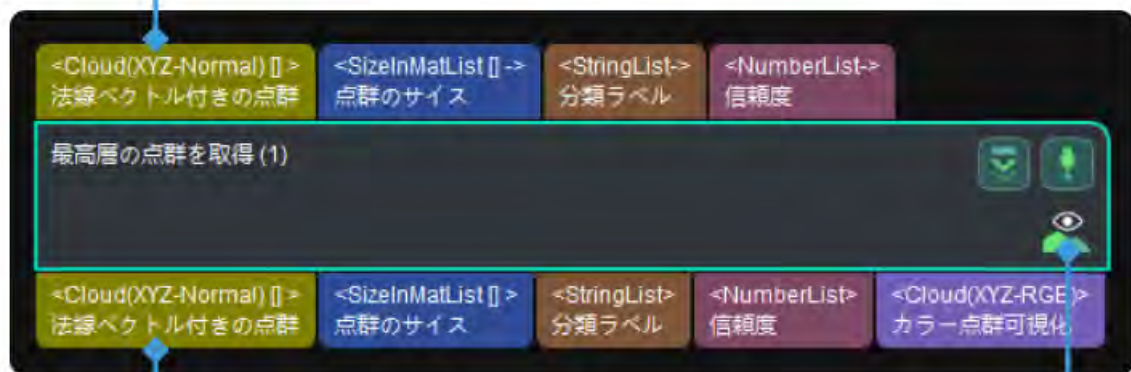
使用シーン

ユーザーに指定された基準方向で要求を満たす最高層の点群は保持されます。このステップは [最高層の点群を取得](#) と異なり、点群を直ちに操作し、[点群クラスタリング](#)、[マスク内の対応する点群を抽出](#) の後で使用されます。

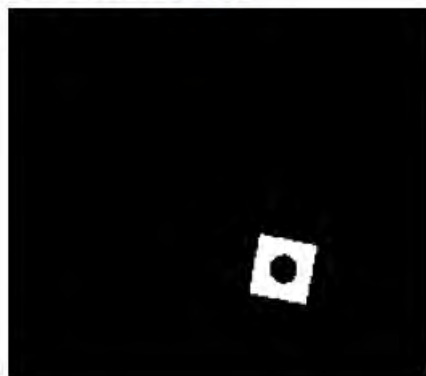
入力と出力



入力点群



処理後の点群



可視化出力結果



パラメータの説明

基準方向

X値、Y値、Z値

初期値：X値=0、Y値=0、Z値=1.0

調整説明：X、Y、Z値を設定することで基準方向を設定します。

層設定

最高層のみを出力

初期値：チェックを入れない

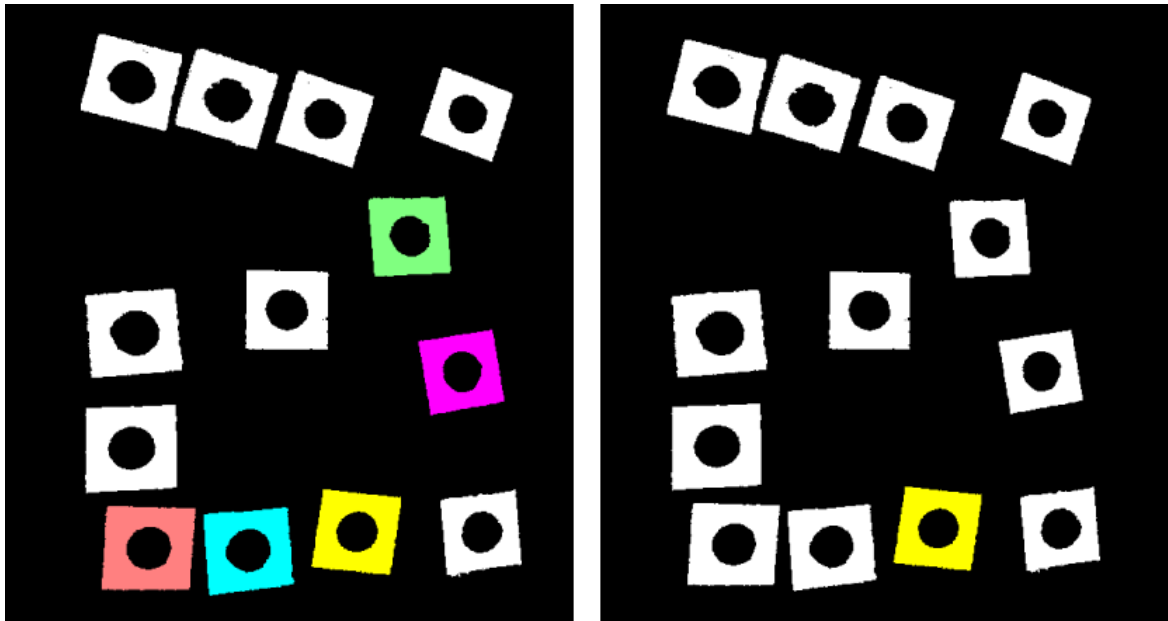
調整説明：チェックを入れれば、指定された方向に沿う最も高いのシングル点群を取得します。そうでなければ、最高層の結果を取得します。

層の高さ

パラメータ説明：このパラメータは、最高層の点群の層の高さ範囲を設定するために使用されます。層の高さの範囲内の点群は保持されます。

初期値：100.000 mm

調整の例：層の高さがそれぞれ100.000mmと35mmである場合、出力効果は下図の左側と右側に示します。



層の許容値

初期値：1.0000

調整説明：このパラメータは最高層と他の層との高さの差の上限を設定します。この上限は、層の高さに層の許容値を掛けた値です。項目が最高層までの距離がこの上限よりも小さい場合、それは最高層です。

点群設定

点数が最も多い点群の法線ベクトルを使用

初期値：チェックを入れない。

調整説明：チェックを入れれば、最大サイズの点群の法線ベクトルを基準方向として使用さ

れます。この時、**基準方向**は無効になります。

4.3.206. 最高層の点を取得

機能

設定された方向に最高層の点を取得します。

使用シーン

汎用の点群フィルタリングステップです。特定の使用シーンはありません。

入力と出力

● 入力

1. このポートに入力される点群の最高層の点は抽出されます。
2. 高／低方向の位置姿勢を定義するために使用されます（オプション。このポートが使用されていない場合は、パラメータの「基準方向」が使用されます。複数の位置姿勢が入力される場合、最初の位置姿勢を使用します）。

● 出力

1. 抽出された点群の最高層の点。

パラメータの説明

基準方向

調整説明：基準方向を選択します。x、y、zの三つの方向を含みます。

初期値：x=0、y=0、z=1.0。

推奨値：実際のニーズに応じて設定してください。

層設定

最高層のみを出力

調整説明：点群で最高の点のみを取得するかどうかを決定します。

初期値：チェックを入れない

推奨値：実際のニーズに応じて設定してください。

層の高さ

調整説明：層の高さ（ミリメートル単位）を設定します。

初期値：100.000 mm

推奨値：実際のニーズに応じて設定してください。

層の許容値

調整説明：最上層と他の層との高さの差の上限を設定します。この上限は、層の高さに層の許容値にかけた値です。ある層と最高層の高さの差がこの値より小さい場合、その層は最高層となります。

初期値：1.0

推奨値：実際のニーズに応じて設定してください。

4.3.207. 最高層の位置姿勢を取得



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

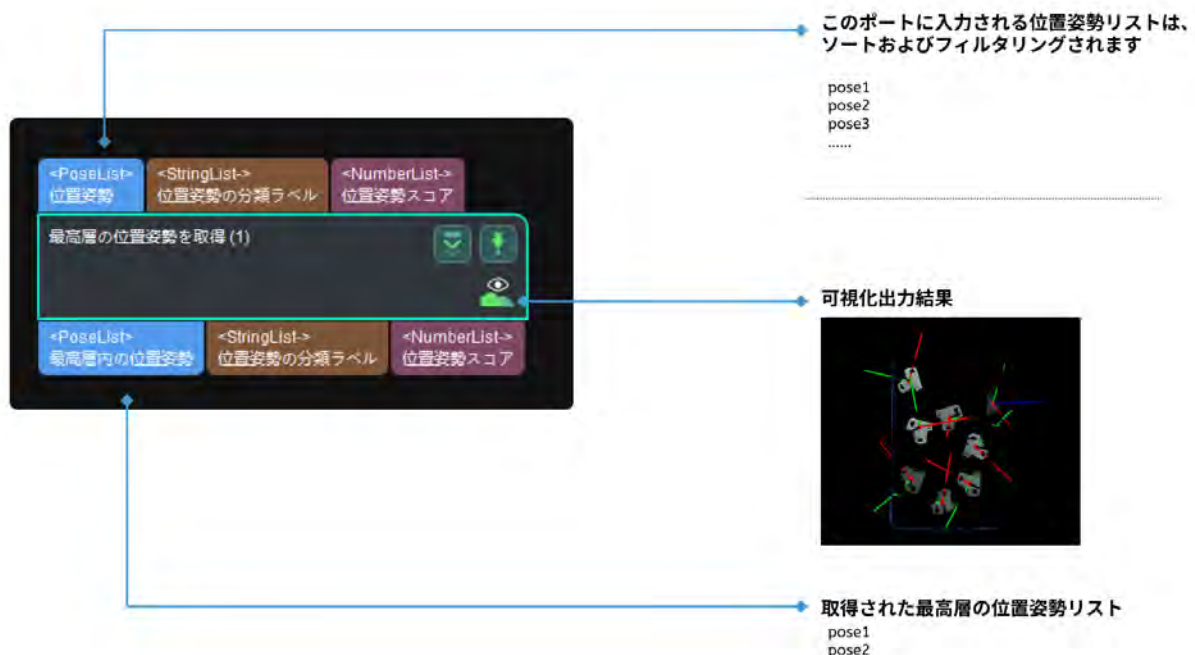
機能

指定方向に位置姿勢を降順ソートして、最高層の位置姿勢を出力します。

使用シーン

汎用の位置姿勢フィルタリングのステップです。設定されたパラメータによって要件を満たす位置姿勢リストを選出します。

入力と出力



4.3.208. 最高スコアの結果を取得



これは古いバージョンのステップであり、メンテナンスが停止されています。このステップに関する詳細が必要な場合は、当社 (docs@mech-mind.net) までお問い合わせください。

機能

2つの点群リストと対応する位置姿勢およびスコアを1対1で比較し、スコアの高い点群と位置姿勢を抽出して出力リストを作成します。

使用シーン

最適なマッチング結果を取得するために、異なるマッチング方法で得られた結果を比較するために使用されます。

これは古いバージョンのステップであり、メンテナンスが停止されますので、新しいバージョンのステップの組合せを使用してください。

入力と出力

● 入力：

1. 最初セットの点群。
2. 最初セットの点群の位置姿勢。
3. 最初セットの位置姿勢のマッチングスコア。
4. 最初セットの位置姿勢の分類ラベル。
5. 最初セットの位置姿勢の信頼度。
6. 二番目セットの点群。
7. 二番目セットの点群の位置姿勢。
8. 二番目セットの位置姿勢のマッチングスコア。
9. 二番目セットの位置姿勢の分類ラベル。
10. 二番目セットの位置姿勢の信頼度。

● 出力：

1. より高いスコアの点群。
2. より高いスコアの位置姿勢。
3. より高いスコア。
4. より高いスコアの位置姿勢の分類ラベル。
5. より高いスコアの位置姿勢の信頼度。

4.3.209. リング形状の点群リストをフィルタリング



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

ステップパラメータの設定によって、リング形状の点群リストをフィルタリングします。しきい値内の点群は保持され、しきい値外の点群は破棄されます。

使用シーン

特定のシーン向けのカスタマイズのステップです。

入力と出力

● 入力：

1. このポートに入力されるリスト内のリング形状の点群はフィルタリングされます。
2. 入力点群のラベルリスト。
3. 入力点群のスコアリスト。

● 出力：

1. 残りのリング形状の点群リスト。
2. 残りのリング形状の点群のラベルリスト。
3. 残りのリング形状の点群のスコアリスト。
4. 残りのリング形状の点群のマスクリスト。

4.3.210. データをグループ化



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

データリスト内の要素をインデックスでグループ化します。複数の入力ポートを設定して、同じインデックスリストを持つ各データリスト内で同時にグループ化操作を実行できます。例えば、インデックスは[[0, 2]], [1, 3, 4]]で、入力データは[100, 20, 300, 40, 500]の場合、[[100, 300]], [20, 40, 500]]として出力します。

使用シーン

汎用の機能です。

入力と出力

● 入力：

1. このポートに入力されるインデックスはデータをグループ化するために使用されます。
2. このポートに入力されるデータはグループ化されます。

● 出力：

1. グループ化されたデータ。

4.3.211. 2D位置姿勢をグループ化

機能

2D位置姿勢をグループ化します。同じマスク内の位置姿勢はリストにグループ化され、どのマスク内にもない位置姿勢は別のリストにグループ化されます。

使用シーン

特定のシーン向けのカスタマイズのステップです。

入力と出力

● 入力：

1. このポートに入力される2D位置姿勢がグループ化されます。
2. 2D位置姿勢をグループ化するためのマスク。

● 出力：

1. グループ化された2D位置姿勢。
2. グループ化のインデックス。

4.3.212. ヒストグラムマッチング



これは古いバージョンのステップであり、メンテナンスが停止されています。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

2D画像のヒストグラムを入力し、パラメータで設定されたテンプレートとしたヒストグラムとマッチングし、スコアに従って対応するラベルを出力します。

使用シーン

特定のシーン向けのカスタマイズのステップです。

入力と出力

- **入力：** このポートに入力されるヒストグラムは、テンプレートとマッチングされます。
- **出力：** マッチング結果のラベル。

4.3.213. 画像の明るさと色バランサー

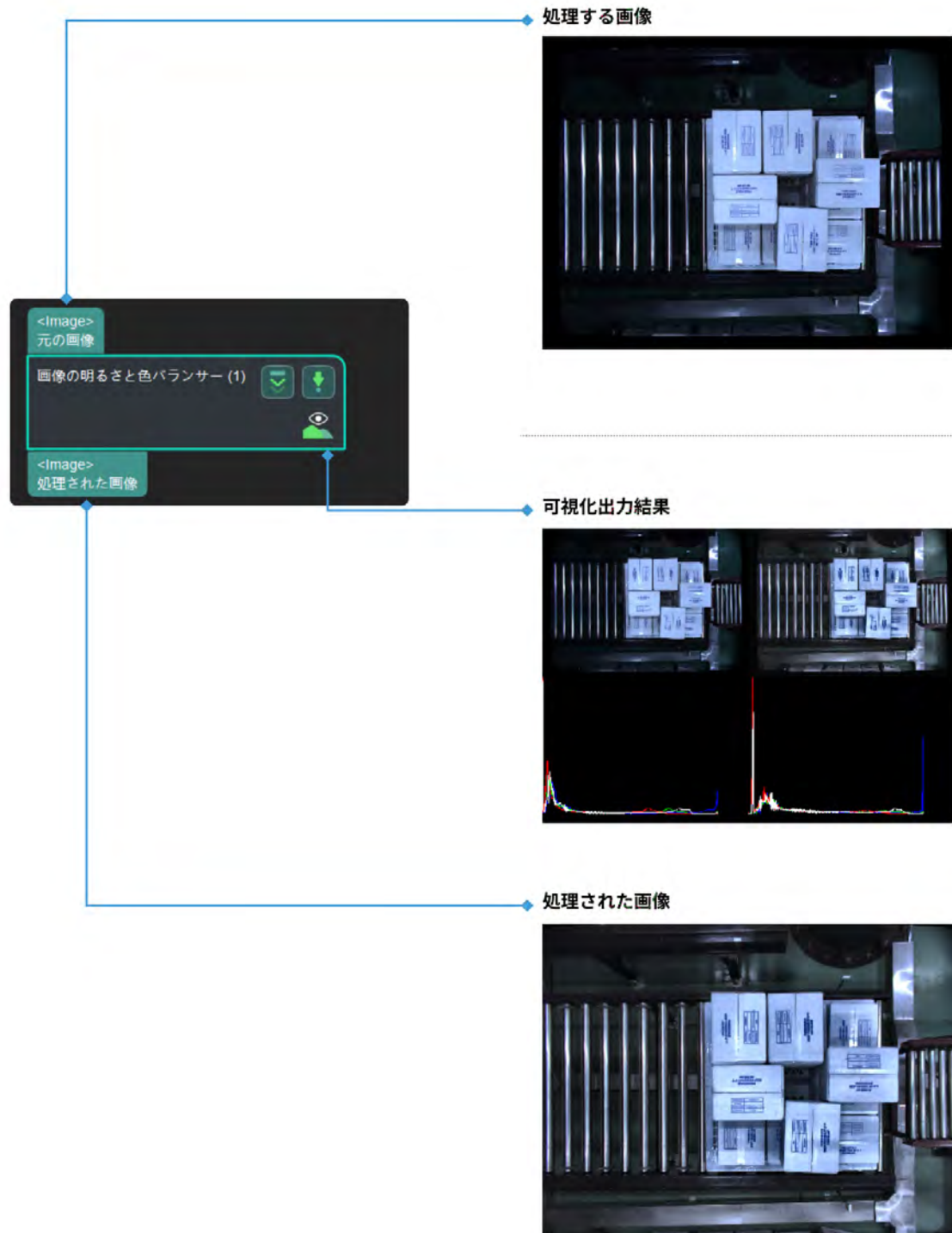
機能

異なるタイプのアルゴリズムを使用して画像をイコライズ処理します。これは主にコントラストの調整に使用されます。また、色と明るさのイコライズにも使用され、エッジ検出や認識などの後続の処理を容易にします。

使用シーン

照度の変化が大きい場合に使用することを推奨します。このステップでは、画像に対してイコライズの処理を実行し、画像の明るさまたは色が適切な範囲内になるようにし、後続のアルゴリズム処理を容易にします。

入力と出力



パラメータの説明

以下の5つのバランサーが利用可能です。調整可能なパラメータは、バランサーによって異なります。

- AdaptiveColorBalancer
- CLAHE (コントラスト限定Histogram適応等化)

- ColorBalancer
- GammaCorrection
- Illumination Normalization

AdaptiveColorBalancer

AdaptiveColorBalancerの場合、画像のRGBパラメータに従ってバランスが調整され、グレースケール画像では効果がはっきりしません。

調整可能なパラメータはありません。

CLAHE（コントラスト限定Histogtam適応等化）

初期値は **CLAHE** となります。

コントラストの最大しきい値

パラメータ説明：このパラメータは、画像全体のイコライジングとイコライジング効果の調整に使用されます。値が大きいほど、イコライズ効果が明らかで画像のコントラストが低くなります。値が小さいほど、イコライズ効果が顕著でなく、画像のコントラストが高くなります。

初期値：4

ColorBalancer

輝度

パラメータ説明：このパラメータは、画像の明るさを調整するために使用されます。値を大きくすると画像は明るくなり、値を小さくすると画像は暗くなります。

初期値：0

緑色から赤色へ

パラメータ説明：このパラメータは、色の調整に使用されます。値が高いほど画像は赤に傾き、値が低いほど画像は緑に傾きます。

初期値：0

青色から黄色へ

パラメータ説明：このパラメータは、色の調整に使用されます。値が高いほど画像は黄色に傾き、値が低いほど画像は青色に傾きます。

初期値：0

GammaCorrection

輝度係数

パラメータ説明：このパラメータの値が大きいほど、画像のコントラストが低くなります。

値が0の場合、一部のピクセルのみが表示されます。

初期値：1.00

Illumination Normalization

Illumination Normalizationを使用する場合、3つの方法があり、それぞれ調整するパラメータが異なります。

- [Retinex_SSR](#)
- [Retinex_MSR](#)
- [BgAdjust](#)

Retinex_SSR

カーネル設定

カーネルサイズ

パラメータ説明：ガウスカーネルサイズを設定するために使用され、奇数のみ設定可能です。値を大きくすると、画像内の対象物のエッジがより鮮明に表示され、値を小さくすると、画像内のエッジがより少なく表示されます。

初期値：21

推奨値：実際の状況に応じて設定してください。

共通設定

平均明るさ

パラメータ説明：このパラメータは、平均明るさを調整するために使用されます。大きくすると、画像は明るくなり、暗い照明条件で取得した画像の処理に適しています。小さくすると、結果の画像は暗くなり、強い照明条件で取得した画像の処理に適しています。

初期値：100

推奨値：実際の状況に応じて設定してください。

Retinex_MSR

カーネル設定

小・中・大の3つのスケールでカーネル処理を行った結果を合成した画像で、3つのカーネルのサイズ関係を調整することで、エッジ効果が大きく異なる画像が得られ、シングルスケール方式に比べてより完全なエッジ情報を得ることができます。

小/中/大カーネルサイズ

パラメータ説明：ガウスカーネルサイズを設定するために使用され、奇数のみ設定可能です。値を大きくすると、画像内の対象物のエッジがより鮮明に表示され、値を小さくすると、

と、画像内のエッジがより少なく表示されます。

小カーネルサイズの初期値：15

中カーネルサイズの初期値：81

大カーネルサイズの初期値：201

推奨値：実際の状況に応じて設定してください。

共通設定

平均明るさ

パラメータ説明：このパラメータは、平均明るさを調整するために使用されます。大きくすると、画像は明るくなり、暗い照明条件で取得した画像の処理に適しています。小さくすると、結果の画像は暗くなり、強い照明条件で取得した画像の処理に適しています。

初期値：100

推奨値：実際の状況に応じて設定してください。

BgAdjust

背景調整の設定

関心領域



元の画像の左上隅の座標は (0, 0) となります。

X

パラメータ説明：このパラメータは、関心領域の左上頂点座標のX値を示します。

初期値：30

Y

パラメータ説明：このパラメータは、関心領域の左上頂点座標のY値を示します。

初期値：700

幅

パラメータ説明：このパラメータは、関心領域の幅を設定するために使用されます。

初期値：200

高さ

パラメータ説明：このパラメータは、関心領域の高さを設定するために使用されます。

初期値：200

共通設定

平均明るさ

パラメータ説明：このパラメータは、平均明るさを調整するために使用されます。大きくすると、画像は明るくなり、暗い照明条件で取得した画像の処理に適しています。小さくすると、結果の画像は暗くなり、強い照明条件で取得した画像の処理に適しています。

初期値：100

推奨値：実際の状況に応じて設定してください。

4.3.214. 画像フィルタリング

機能

入力画像に対して異なるフィルターを使用してフィルタリングを行います。

使用シーン

汎用の画像処理ステップで、特定の使用シーンはありません。

パラメータの説明

以下のフィルターが利用可能です。

- [GaussianFilter](#)（ガウシアンフィルタ）
- [BilateralFilter](#)（バイラテラルフィルタ）
- [Boxfilter](#)（ボックスフィルタ）
- [ConditionalMeanFilter](#)（条件付き平均値フィルタ）
- [DftFilter](#)（離散フーリエ変換フィルタ）
- [GuidedFilter](#)（ガイド付きフィルタ）
- [HaarFilter](#)（Haarフィルタ）
- [MedianFilter](#)（メディアンフィルタ）

GaussianFilter（ガウシアンフィルタ）

ガウス関数を用いて画像処理を行います。

カーネルのX方向の幅

パラメータ説明：このパラメータは、ガウシアンカーネルのX方向の幅を設定するために使用されます。値を大きくすると、画像がよりぼやけます。

初期値：3

調整アドバイス：実際の状況に応じて設定してください。

カーネルのY方向の高さ

パラメータ説明：このパラメータは、ガウシアンカーネルのY方向の高さを設定するために使用されます。値を大きくすると、画像がよりぼやけます。

初期値：3

調整アドバイス：実際の状況に応じて設定してください。

カーネルのX方向の標準偏差

パラメータ説明：このパラメータは、ガウシアンカーネルのX方向の標準偏差を設定するために使用されます。値を大きくすると、画像がよりぼやけます。

初期値：3

調整アドバイス：実際の状況に応じて設定してください。

カーネルのY方向の標準偏差

パラメータ説明：このパラメータは、ガウシアンカーネルのY方向の標準偏差を設定するために使用されます。値を大きくすると、画像がよりぼやけます。

初期値：3

調整アドバイス：実際の状況に応じて設定してください。

BilateralFilter（バイラテラルフィルタ）

直径

パラメータ説明：このパラメータは、フィッティングの範囲を指定するために使用されます。値が大きいほど、元の画像から保持される情報が多くなり、画像が鮮明になります。値が小さいほど、元の画像から保持される情報が少なくなり、画像がぼやけます。

初期値：3

調整アドバイス：実際の状況に応じて設定してください。



フィッティングの効果は、**色空間フィルターの標準偏差**と**座標空間フィルターの標準偏差**によって異なります。

色空間フィルターの標準偏差

パラメータ説明：このパラメータは、画像のピクセル間の標準偏差を設定するために使用されます。

初期値：3

調整アドバイス：実際の状況に応じて設定してください。

座標空間フィルターの標準偏差

パラメータ説明：このパラメータは、画像のピクセル間の空間距離の標準偏差を設定するために使用されます。

初期値：3

Boxfilter（ボックスフィルタ）

カーネルサイズ

パラメータ説明：このパラメータは、フィルタリング処理時のカーネルの辺長の大きさ（ピクセル単位）を設定するために使用されます。値が大きいほどフィルタリングされた画像はぼやけ、値が小さいほど鮮明な画像になります。

初期値：3

調整アドバイス：実際の状況に応じて設定してください。

ConditionalMeanFilter（条件付き平均値フィルタ）

カーネルサイズ

パラメータ説明：このパラメータは、フィルタリング処理に使用されるカーネルサイズを設定するために使用されます。

初期値：3

調整アドバイス：実際の状況に応じて設定してください。

しきい値

パラメータ説明：このパラメータは、画像のピクセル間のしきい値（ピクセル単位）を設定するために使用されます。値がこれより大きい場合、そのピクセルは保持されます。値がこれより小さい場合、そのピクセルは除去されます。

初期値：100px

調整アドバイス：実際の状況に応じて設定してください。

DftFilter（離散フーリエ変換フィルタ）

離散フーリエ変換を使用して画像を処理し、高周波信号（ノイズやエッジなど）をフィルタリングすることでノイズ除去またはエッジエンハンスメントを行います。

GuidedFilter（ガイド付きフィルタ）

カーネルサイズ

パラメータ説明：このパラメータは、フィルタリング処理に使用されるカーネルサイズを設定するために使用されます。この値が大きいほど画像平滑化効果が強く、値が小さいほど画像平滑化効果が弱くなります。

初期値：3

調整アドバイス：実際の状況に応じて設定してください。

正則化項

パラメータ説明：このパラメータは、ガイド付きフィルタの正則化項を設定するために使用されます。

初期値：0.01

調整アドバイス：実際の状況に応じて設定してください。

HaarFilter (Haarフィルタ)

画像に対して平滑化とノイズリダクションを行って一部のノイズを除去します。

MedianFilter (メディアンフィルタ)

カーネルサイズ

パラメータ説明：このパラメータは、フィルタリング処理に使用されるカーネルサイズを設定するために使用されます。この値が大きいほど画像平滑化効果が強く、値が小さいほど画像平滑化効果が弱くなります。

初期値：3

調整アドバイス：実際の状況に応じて設定してください。

4.3.215. 画像調整



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社 (docs@mech-mind.net) までお問い合わせください。

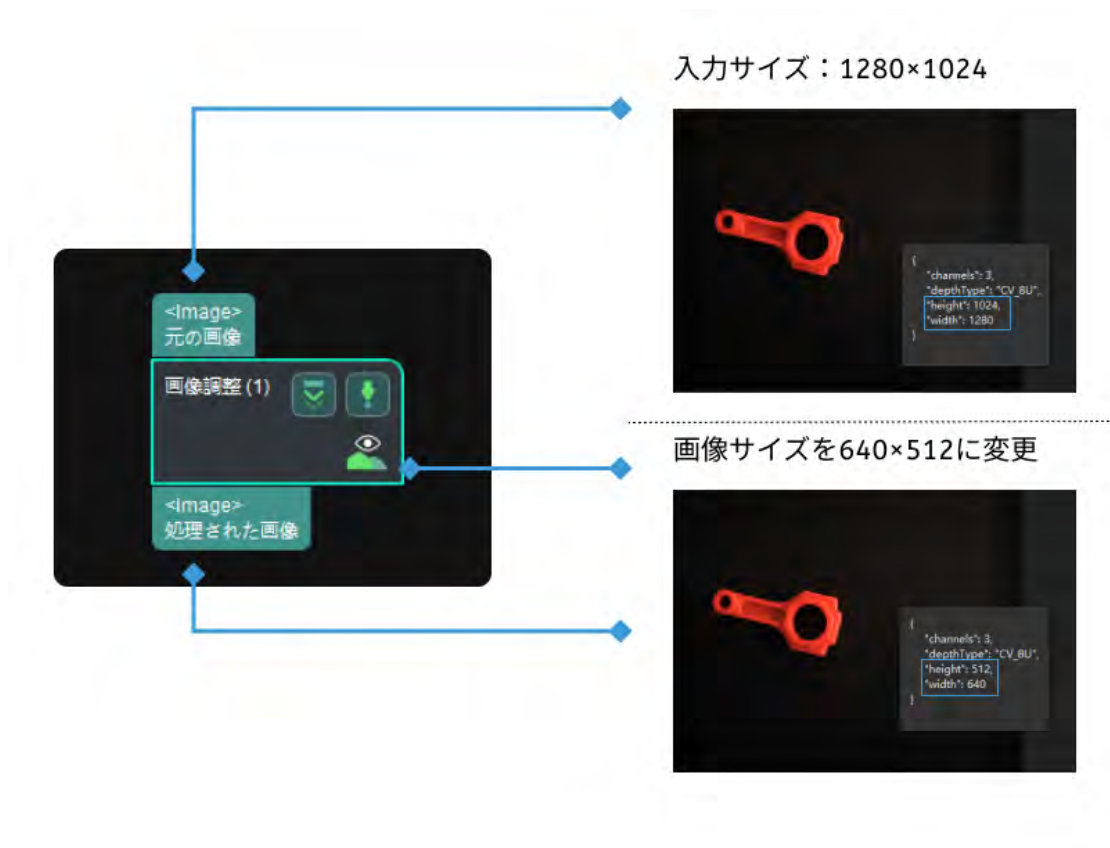
機能

入力された2D画像をスケーリングするか、カメラの歪み係数で画像を補正します。

使用シーン

汎用の画像処理ステップで、特定の使用シーンはありません。

入力と出力



4.3.216. 終了点を挿入して移動パラメータを送信



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社 (docs@mech-mind.net) までお問い合わせください。

機能

入力された経路の最後に、単一の経路点を挿入します。

使用シーン

通常、経路の応用シーンで使用されます。

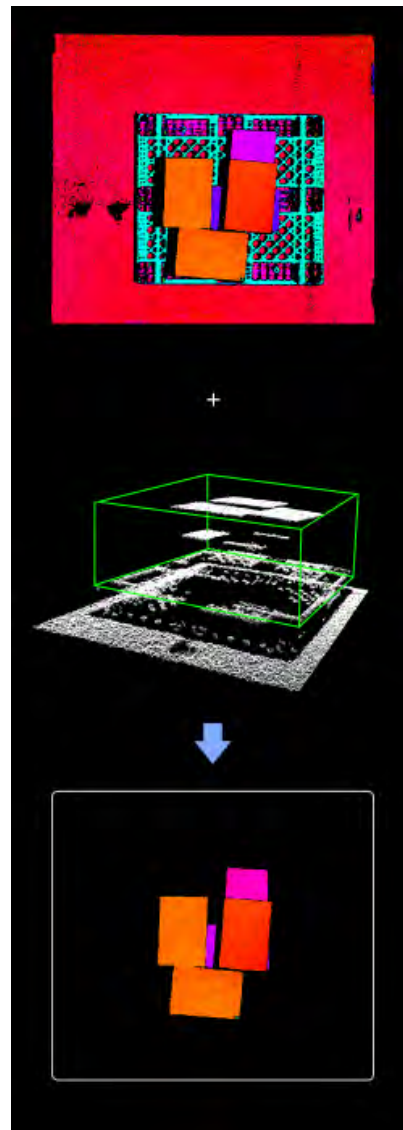
入力と出力

- 入力：
 1. このポートに入力される経路点のリストの最後に、新しい経路点が挿入されます。
 2. このポートに入力される新しい経路点は、リストの最後に挿入されます。
- 出力：新しい経路点が最後に挿入された後の経路点のリスト。

4.3.217. 深度画像の3D ROI外の領域を無効に設定

機能

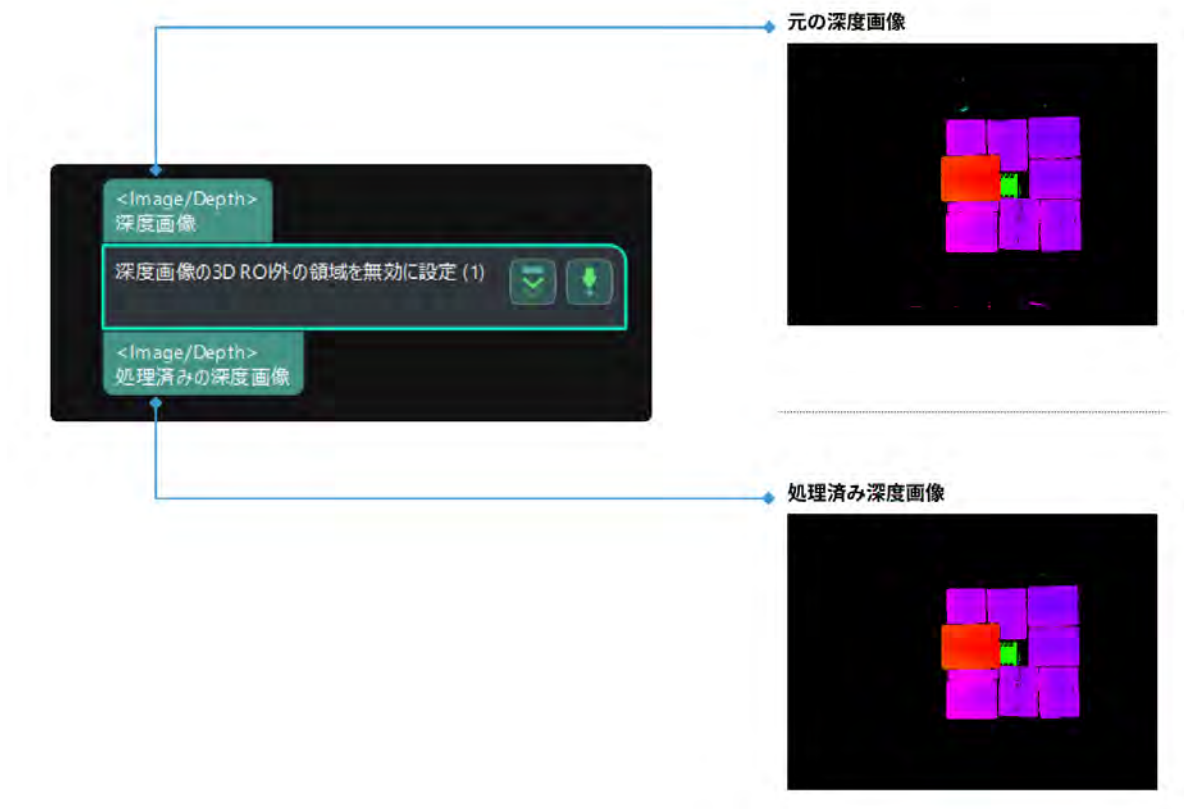
深度画像の関心領域（ROI）を設定し、関心領域以外のピクセル値はゼロに設定されます。



使用シーン

このステップを使用して、元の深度画像の対象領域を抽出し、背景や他の領域からの干渉を回避します。通常、[深度画像分割](#)、[深度画像で最高領域を取得](#)、[マスク画像をマージ](#)などのステップと併用されます。

入力と出力



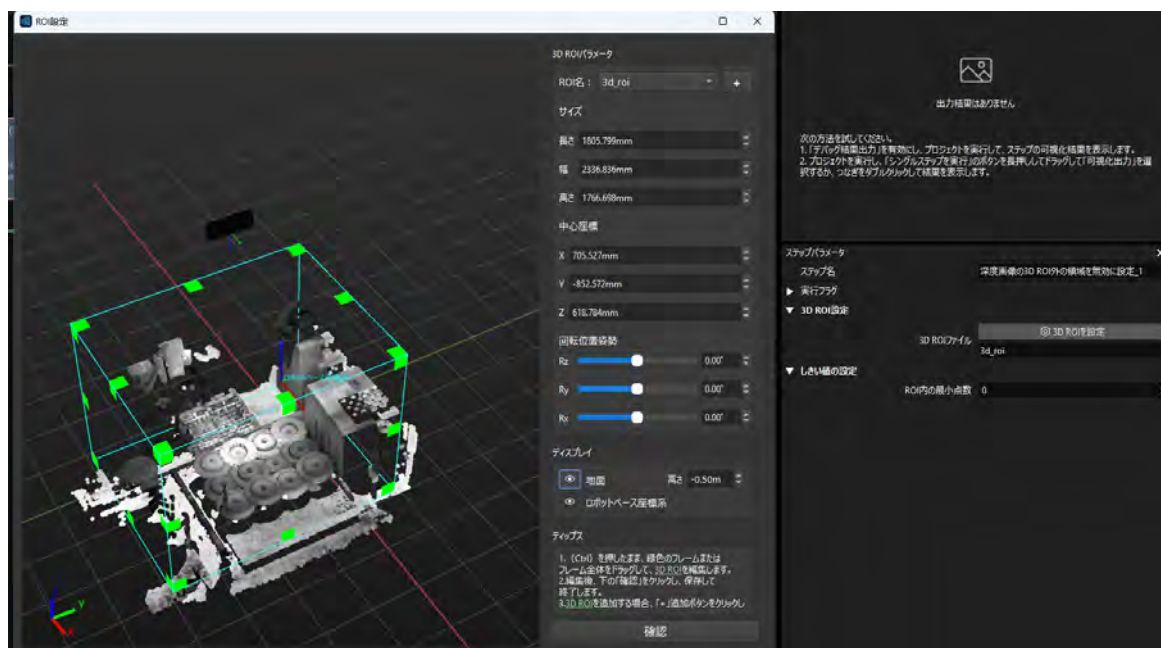
パラメータの説明

3D ROI設定

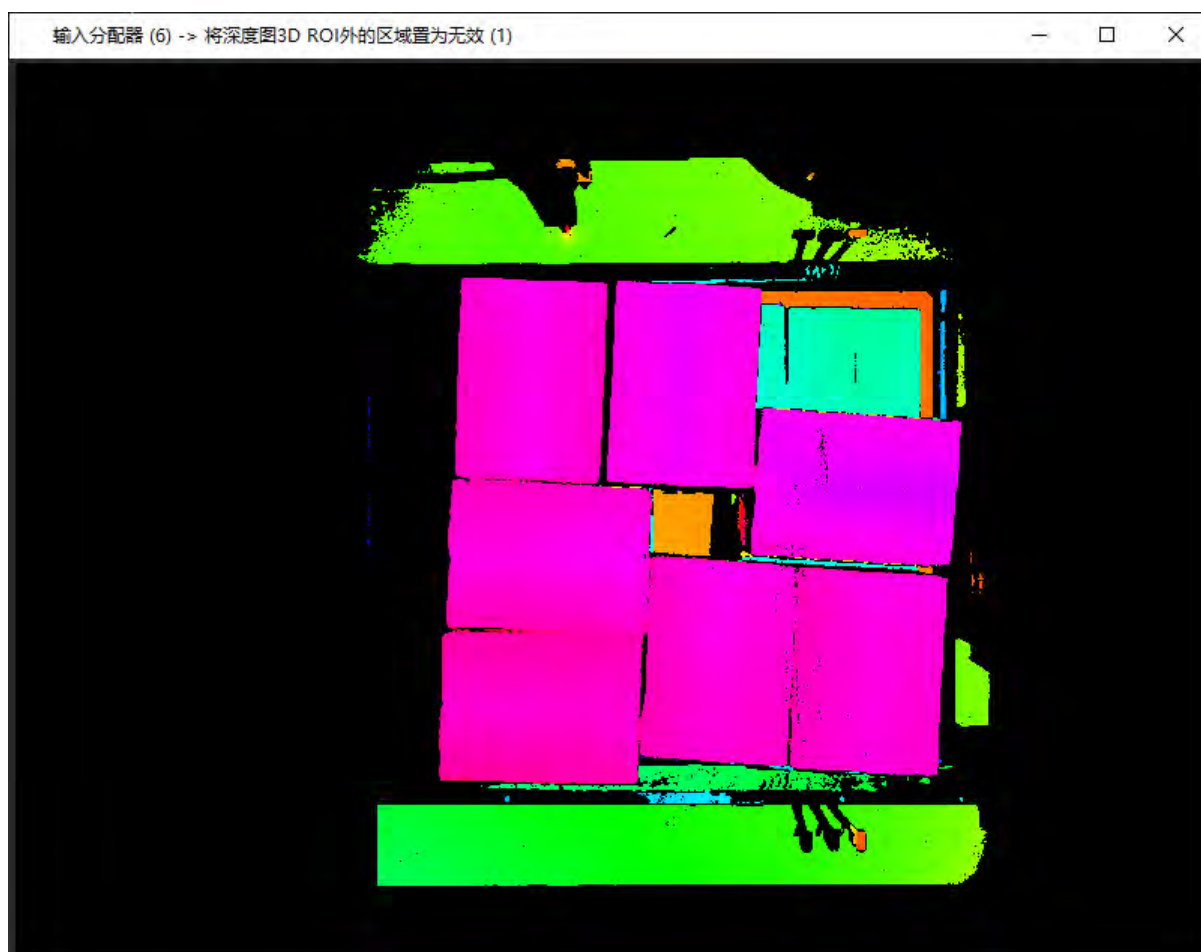
3D ROIファイル

パラメータ説明：このパラメータは、3D ROIを設定するために使用されます。

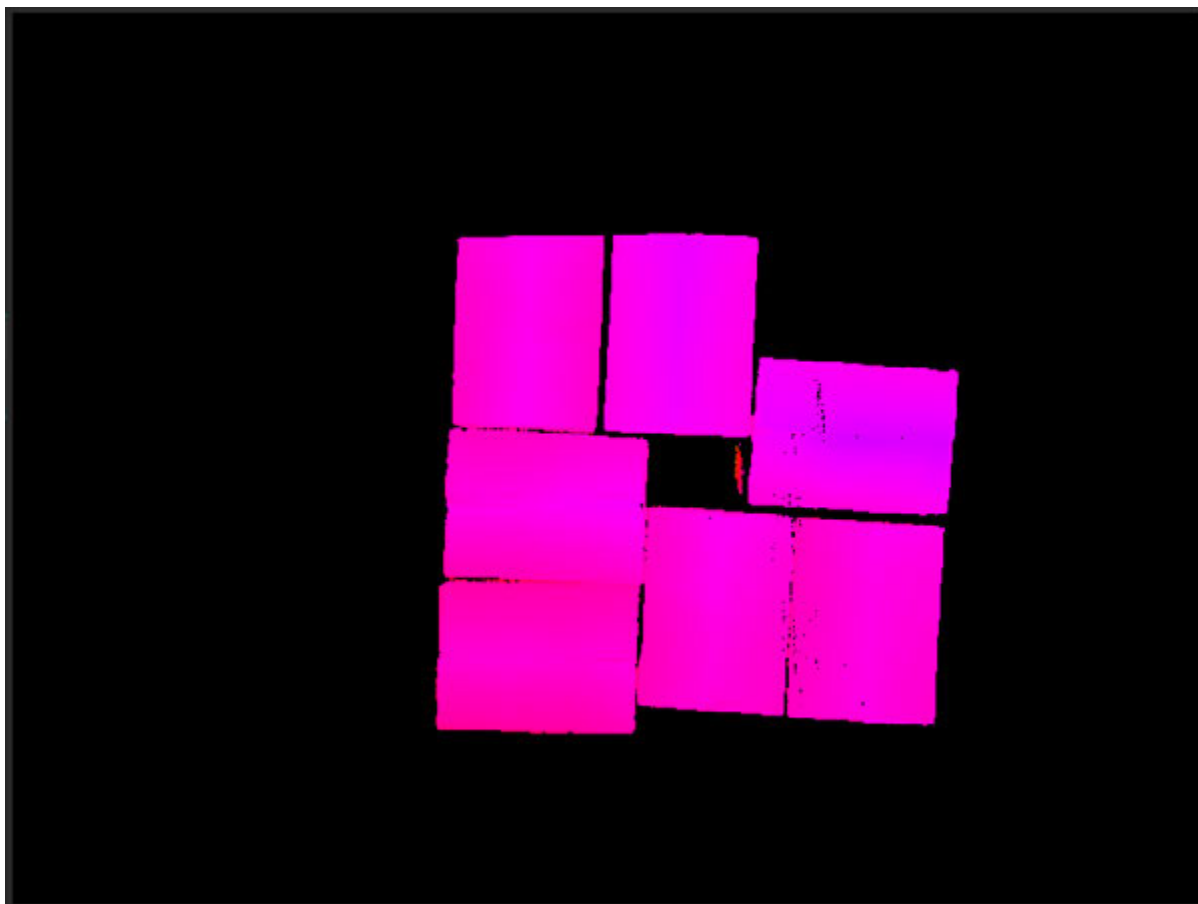
調整説明：ステップパラメータで **[3D ROIを設定]** をクリックして、下図のような **設定画面** に入ります。3D ROIの設定方法については、[3D ROIの設定手順](#) をご参照ください。



深度画像の関心領域の選択を完了すると、関心領域以外のピクセルがゼロに設定されます。下図は入力された深度画像です。



下図はROI設定後の深度画像です。ROI設定後、箱の周囲の無効な情報は除去されます。



しきい値設定

ROI内の最小点数

パラメータ説明：このパラメータは、3D ROI内の点数のしきい値を設定するために使用されます。点数がしきい値より少ない場合、下図のようなメッセージが表示されます。



調整アドバイス：実際の状況に応じて設定してください。

4.3.218. 位置姿勢を逆変換

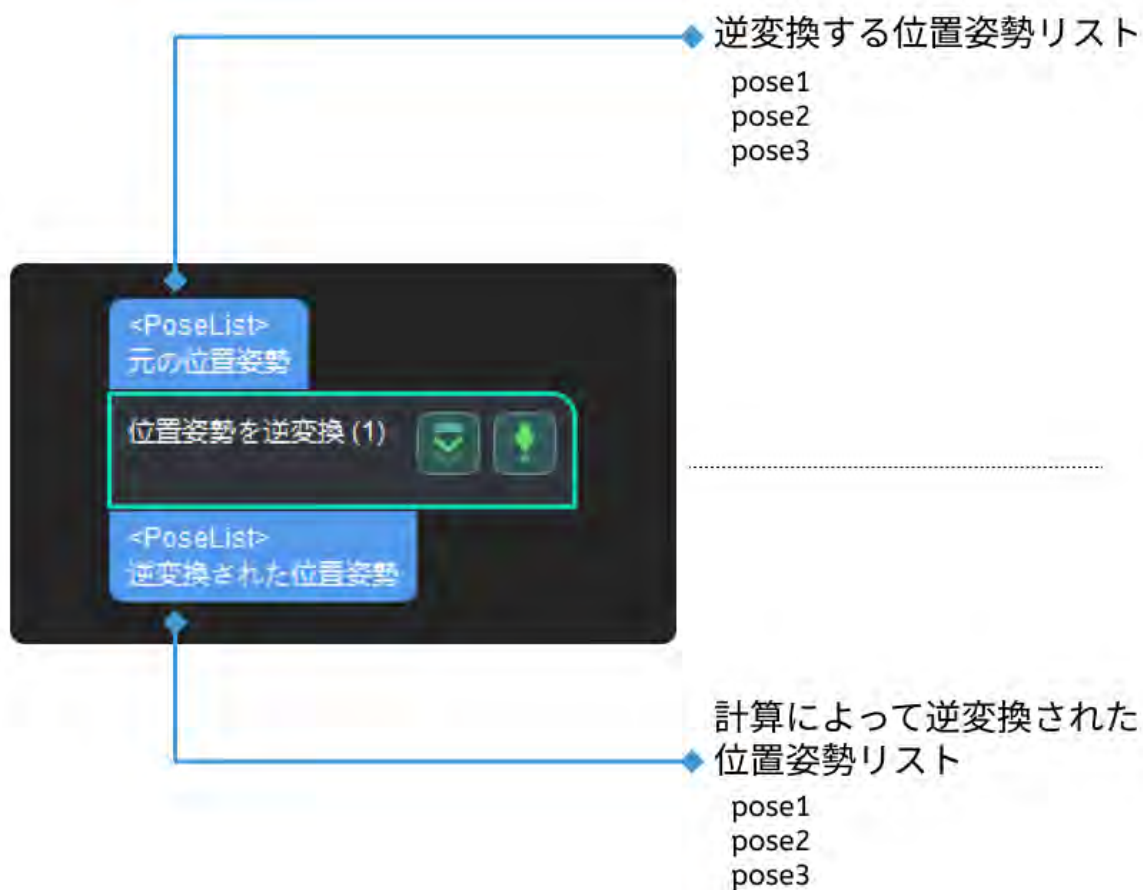
機能

入力リストにある位置姿勢を逆変換します。

使用シーン

汎用の位置姿勢計算ステップです。基準座標系間の変換に使用できます。

入力と出力



4.3.219. 四元数を逆変換

機能

入力された四元数を逆変換します。

使用シーン

汎用の四元数の逆変換ステップです。特定の使用シーンはありません。

入力と出力

- **入力：** このポートに入力される四元数は逆変換されます。
- **出力：** 計算された逆変換後の四元数。

4.3.220. 位置姿勢のZ値をしきい値と比較



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

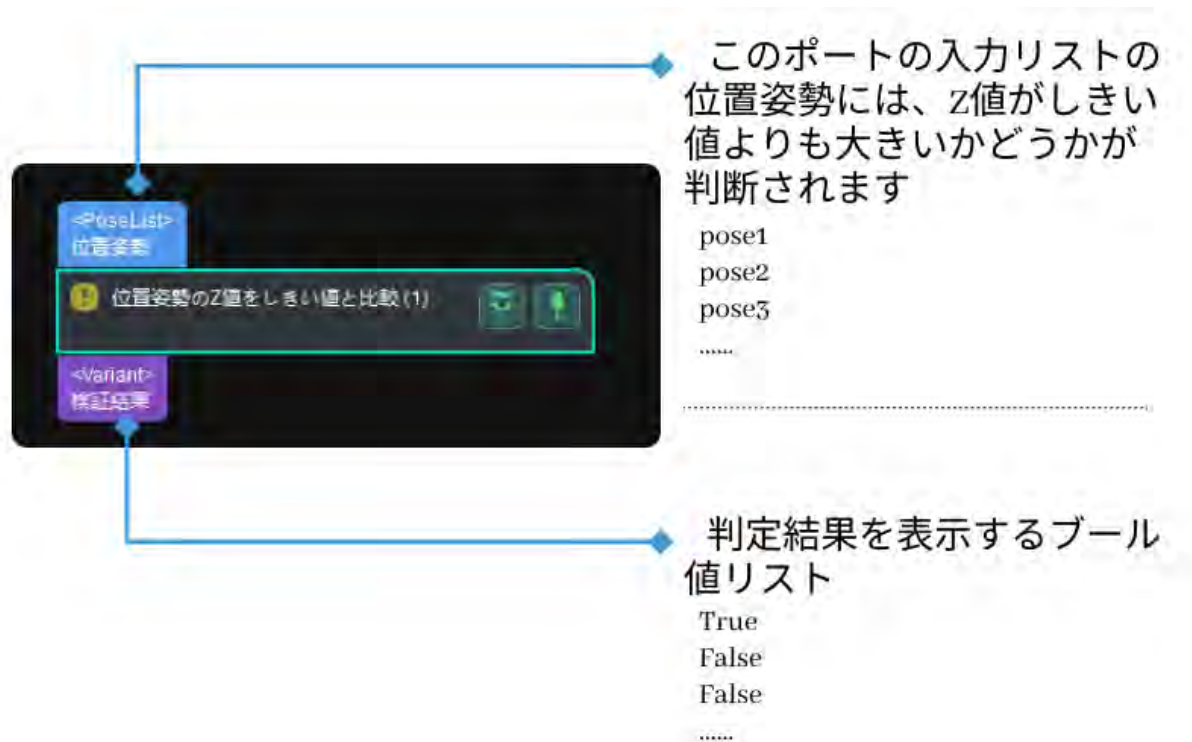
機能

入力位置姿勢のZ値を、パラメータで設定されたしきい値と比較して、判定結果を出力します。Z値はしきい値よりも大きい、または同じである場合、「True」を出力します。それ以外の場合は、「False」を出力します。

使用シーン

特定のシーン向けのカスタマイズのステップです。

入力と出力



4.3.221. ラベル接続



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

2つのラベルリストを1対1で1つにマージします。例えば、最初のポートに["a", "b"]を入力し、2番目のポートに["c", "d"]を入力する場合、["ac", "bd"]を出力します。

使用シーン

汎用のラベル編集ステップです。特定の使用シーンはありません。

入力と出力

- **入力：**
 1. このポートに入力されるラベルはマージされます。
 2. このポートに入力されるラベルはマージされます。
- **出力：**
 1. マージされたラベル。

4.3.222. 正ポリゴンに沿った分布する位置姿勢を保持



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

1つセットの位置姿勢を入力し、必要な位置姿勢のみを保持します。必要な位置姿勢は、正ポリゴンの頂点にあります。

使用シーン

特定のシーン向けのカスタマイズのステップで、ホイールハブ取り付け穴の識別のシーン専用です。

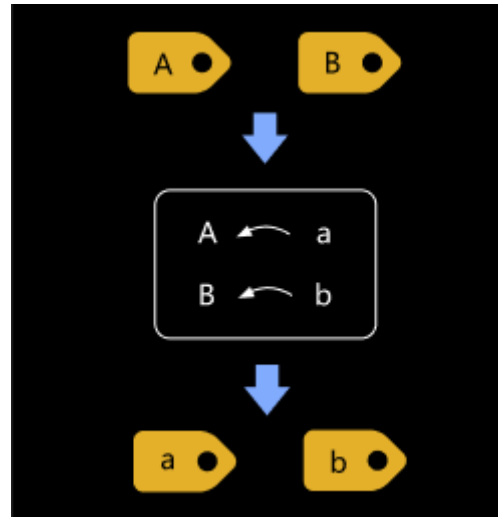
入力と出力

- **入力：** このポートに入力される位置姿勢リストはフィルタリングされます。
- **出力：** 正ポリゴンの頂点にある位置姿勢。

4.3.223. ラベルマッピング

機能

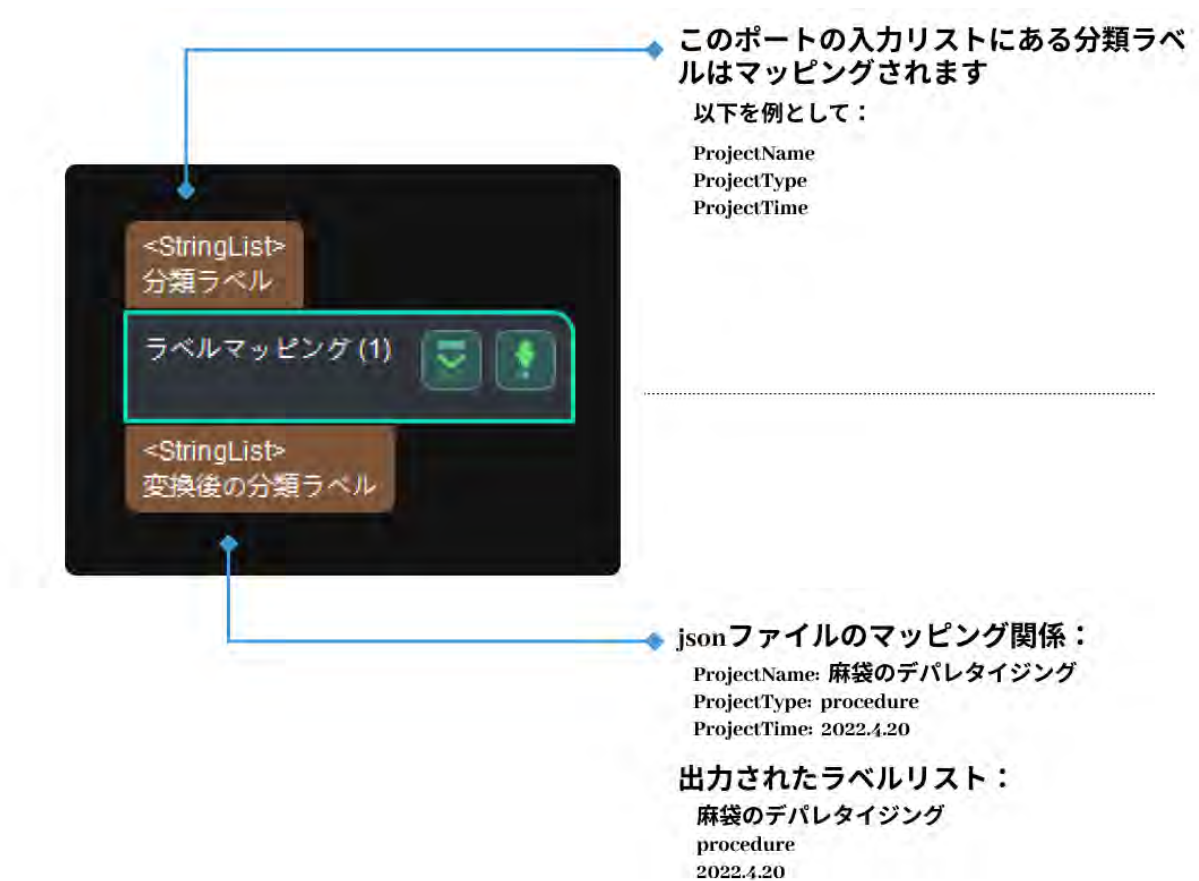
入力ラベルに対して1対1のマッピングを行い、ラベルの名前を変更します。



使用シーン

ラベル名の変更に使用されます。

入力と出力



パラメータの調整説明

ポート設定

入力ポート数 (1-8)

パラメータ説明：このパラメータは、このステップの入力ポートの数を設定するために使用されます。

初期値：1

値の範囲：1-8

ファイル設定




ラベルマッピングは、**入力ポート数** が1の場合のみ設定可能です。

ラベルファイル

パラメータ説明：このパラメータは、マッピング後のラベルファイルを設定するために使用されます。

初期値：labelmap.json

設定方法：

1. **[ラベルマッピングを設定]** をクリックし、JSONエディタの画面に入ります。
2.  をクリックし、**変換前のラベル** に変更するラベル名を入力し、**[OK]** をクリックします。その後、**変換後のラベル** に必要なラベル名を入力し、**[OK]** をクリックします。すると、ラベル名が正常に変更され、その時点で対応関係がマッピング欄に表示されます。



ラベル名に小数点以下の数字や特殊文字を入力すると、ラベルの追加に失敗し、リストの下に無効な入力が表示されます。

3. 変更が必要なラベルがすべて追加されるまで、必要に応じて手順2を繰り返してください。
4. **ディレクトリ** は、現在のプロジェクトのパスです。**[...]** をクリックしてプロジェクトの場所を変更することができます。実際の状況に応じて設定してください。
5. **JSONファイル名** にファイル名を入力し、**[保存]** をクリックします。すると、ラベルがこのファイルに保存されます。
6. **[確認]** をクリックし、ラベルの変更が完了します。



ラベルファイルが事前に設定されている場合は、**JSONファイル名** の **[...]** をクリックして、ファイルを選択し、マッピング関係を読み込むことができます。

4.3.224. ラインスキャンレーザーカメラ

機能

Mech-Visionを使ってレーザースキャナを制御してデータを取得します。

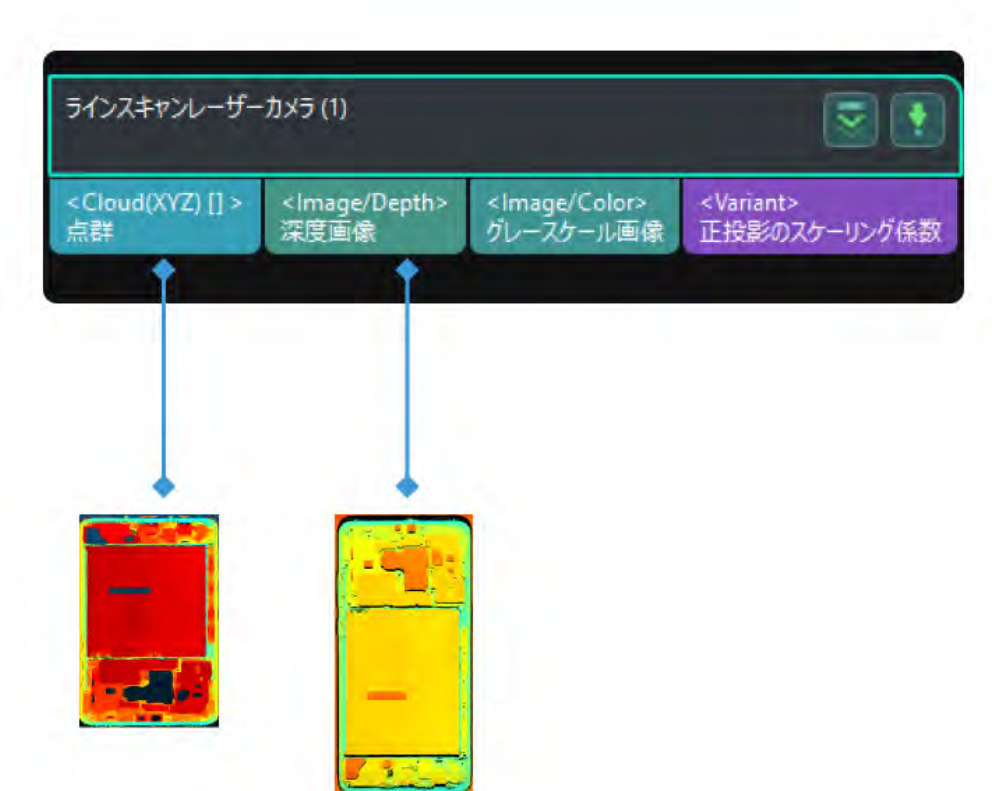


このステップは現在、**Mech-Eye 3D ラインレーザープロファイラー** と **SSZN製の3Dカメラ** のみ対応しています。

使用シーン

このステップは、一般的にビジョンプロジェクトの開始入力として使用され、データはラインスキャンレーザーカメラで取得されます。3C、新エネルギー、自動車、家電などの高精度な検査・測定用途に適しています。

入力と出力



パラメータの調整説明

LNXCamera（初期値）、**SSZNCamera** の2種類があります。

- Mech-Eye 3Dラインレーザープロファイラーを使用する場合、**LNXCamera** を選択してください。
- SSZN製の3Dカメラを使用する場合、**SSZNCamera** を選択してください。
- [LNXCamera](#)
- [SSZNCamera](#)

LNXCamera

3DラインレーザープロファイラーLNXシリーズを使用する場合、以下のパラメータを調整する必要があります。

IP

パラメータ説明：このパラメータは、接続するデバイスのIPアドレスを設定するために使用されます。

調整説明：[**エディタを開く**] をクリックしてIPアドレスを設定します。デバイスを選択し、ダブルクリックして接続します。

パラメータグループを使用

パラメータ説明：このパラメータを有効にすると、パラメータグループを読み込むことができます。

初期値：無効

パラメータグループ

パラメータ説明：このパラメータは、必要なパラメータグループを選択するために使用されます。

トリガー設定

等価パルス

パラメータ説明：このパラメータは、エンコーダから受信した信号の解像度、つまりエンコーダの2つのパルス信号間の距離をミリメートル（mm）単位で表示します。

初期値：0.004mm

モーションモード

パラメータ説明：このパラメータは、撮影される対象物がカメラに対する運動方向を選択するために使用されます。

オプション：

- 一方向運動：初期値です。X方向の測定範囲が認識する対象物をカバーできる場合に推奨します。
- 往復運動：一方向運動が要件を満たしていない場合に推奨します。

一般的な適用シーンは以下の通りです。

- カメラがコンベアと同時に移動する場合に使用すると、効率を向上させます。
- X方向の測定範囲が認識する対象物をカバーできない場合は、往復運動でスキャンし、ステッチングにより完全な深度データを出力します。



往復運動 は、Mech-Eye Viewerで **コーディングモード** が双方向の場合にのみ選択可能です。

横方向の移動距離

パラメータ説明：このパラメータは、コンベアまたはカメラが横方向に移動する距離（ミリ

メートル単位) を示します。

初期値：100mm



このパラメータは、**モーションモード** を **往復運動** に設定した場合にのみ設定できます。



パラメータグループを使用 を有効にする場合、**トリガースソース** と **エンコーダトリガー間隔** の2つのパラメータを設定できません。

トリガースソース

パラメータ説明：このパラメータは、取得のトリガー方式を指定するために使用されます。

オプション：

- Encoder：初期値で、エンコーダでトリガーします。
- Software：Mech-Eye Viewerでトリガーします。

エンコーダトリガー間隔

パラメータ説明：このパラメータは、1行のデータを取得するために必要なエンコーダ信号を指定するために使用されます。

初期値：10



このパラメータは、トリガースソースを **Encoder** に設定した場合にのみ設定できます。

取得停止の条件



以下のいずれかの場合、取得を中止します。

- 取得時間が **最大取得時間** より大きい場合。
- 設定した **移動距離** に到達した場合。
- 設定した **取得行数** に到達した場合。

最大取得時間

パラメータ説明：このパラメータは、最大取得時間（単位はミリ秒）をを設定するために使用されます。取得時間がこの値を超えると、取得は停止します。

初期値：5000ms

取得の制約条件

パラメータ説明：このパラメータは、取得の行数や移動距離を制限するために使用されます。

オプション：

移動距離

● 最大行数

初期値：移動距離。

移動距離

パラメータ説明：このパラメータは、Y方向における取得されたデータの長さ（ミリメートル単位）を示します。

調整説明：移動距離＝期待される行数×トリガー間隔×等価パルス

初期値：100mm



このパラメータは、**取得の制約条件**を**移動距離**に設定した場合にのみ設定できます。

最大行数

パラメータ説明：このパラメータは、カメラ取得の最大行数を設定するために使用されます。最大行数に到達すると、取得は停止します。

初期値：30000



このパラメータは、**取得の制約条件**を**最大行数**に設定した場合にのみ設定できます。

露光設定



このパラメータは、**パラメータグループを使用**を有効にした場合に使用できません。

露光モード

パラメータ説明：このパラメータは、ラインスキャンカメラの露光モードを選択するために使用されます。

オプション：

- Timed：固定露光、固定的な露光時間を使用してデータを取得します。このパラメータは、取得する対象物が同じタイプの場合に使用されます。
- HDR：複数の露光時間を用いて露光効果を融合させるモードです。取得した対象物に、反射する対象物に反射しやすいものと反射しないものがある場合に使用されます。

露光時間

調整説明：調整する場合は、反射しやすい対象物は露光時間を短くし、反射しにくい対象物は露光時間を長くしてください。



このパラメータは、**露光モード**を**Timed**に設定した場合にのみ設定できます。

HDR露光時間1 / 2 / 3

調整説明：対象物の反射性に応じて露光時間を設定してください。



このパラメータは、**露光モード**を**HDR**に設定した場合にのみ設定できます。

露光遅延

パラメータ説明：このパラメータは、露光の遅延時間（ μs 単位）を設定するために使用されます。トリガー信号が受信後、設定した時間を遅らせてから露光を開始します。

初期値：5 μs

ROI設定



このパラメータは、**パラメータグループを使用**を有効にした場合に使用できません。

Z方向のROI

パラメータ説明：このパラメータは、Z方向の関心領域を設定するために使用されます。

オプション：

- 画像全体の高さ：初期値で、Z方向の全体領域です。
- 画像高さの半分：Z方向の半分です。
- 画像高さの4分の1：Z方向の4分の1です。
- 画像高さの8分の1：Z方向の8分の1です。
- 画像高さの16分の1：Z方向の16分の1です。

SSZNCamera

SSZN製の3Dカメラを使用する場合、以下のパラメータを調整する必要があります。

IP

パラメータ説明：このパラメータは、接続するデバイスのIPアドレスを設定するために使用されます。

調整説明：[**エディタを開く**] をクリックしてIPアドレスを設定します。デバイスを選択し、ダブルクリックして接続します。

パラメータグループを使用

パラメータ説明：このパラメータを有効にすると、パラメータグループを読み込むことができます。

初期値：無効

コンフィグファイル名

パラメータ説明：このパラメータは、JSON形式のパラメータグループを読み込むために使用されます。



パラメータファイルは、**パラメータグループを使用**を有効にする場合にのみ読み込むことができます。

トリガー設定

等価パルス

パラメータ説明：このパラメータは、エンコーダから受信した信号の解像度、つまりエンコーダの2つのパルス信号間の距離をミリメートル（mm）単位で表示します。

初期値：0.004mm

モーションモード

パラメータ説明：このパラメータは、撮影される対象物がカメラに対する運動方向を選択するために使用されます。

オプション：

- 一方向運動：初期値です。X方向の測定範囲が認識する対象物をカバーできる場合に推奨します。
- 往復運動：一方向運動が要件を満たしていない場合に推奨します。

一般的な適用シーンは以下の通りです。

- カメラがコンベアと同時に移動する場合に使用すると、効率を向上させます。
- X方向の測定範囲が認識する対象物をカバーできない場合は、往復運動でスキャンし、ステッチングにより完全な深度データを出力します。

横方向の移動距離

パラメータ説明：このパラメータは、コンベアまたはカメラが横方向に移動する距離（ミリメートル単位）を示します。

初期値：100mm



このパラメータは、**モーションモード**を**往復運動**に設定した場合にのみ設定できます。

エンコーダトリガー間隔

パラメータ説明：このパラメータは、1行のデータを取得するために必要なエンコーダ信号を指定するために使用されます。

初期値：10

取得停止の条件



以下のいずれかの場合、取得を中止します。

- 取得時間が**最大取得時間**より大きい場合。
- 設定した**移動距離**に到達した場合。
- 設定した**取得行数**に到達した場合。

最大取得時間

パラメータ説明：このパラメータは、最大取得時間（単位はミリ秒）を設定するために使用されます。取得時間がこの値を超えると、取得は停止します。

初期値：5000ms

取得の制約条件

パラメータ説明：このパラメータは、取得の行数や移動距離を制限するために使用されます。

オプション：

- 移動距離
- 最大行数

初期値：移動距離。

移動距離

パラメータ説明：このパラメータは、Y方向における取得されたデータの長さ（ミリメートル単位）を示します。

調整説明：移動距離＝期待される行数×トリガー間隔×等価パルス

初期値：100mm



このパラメータは、**取得の制約条件**を**移動距離**に設定した場合にのみ設定できません。

最大行数

パラメータ説明：このパラメータは、カメラ取得の最大行数を設定するために使用されます。最大行数に到達すると、取得は停止します。

初期値：30000



このパラメータは、**取得の制約条件**を**最大行数**に設定した場合にのみ設定できません。

4.3.225. 2Dテンプレートを読み込む



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

入力される2D画像で2Dエッジテンプレートを生成します。

使用シーン

通常、2D形状マッチングに使用されます。

入力と出力

- **入力：** このポートに入力される2D画像のコンテンツは、2Dエッジテンプレートを生成するために使用されます。
- **出力：** 生成された2Dエッジテンプレート。

4.3.226. 経路点の読み込みと変換



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

経路の位置姿勢を読み込み、位置姿勢をカメラ座標系からロボット座標系に変換します。

使用シーン

通常、ハッチ/ガラス接着剤塗布の応用シーンで使用されます。

入力と出力

- **入力：**
 1. このポートに入力される剛体変換行列は、座標系の変換に使用されます。
 2. シーンの点群。シーンの点群。これは位置姿勢を点群表面に調整するために使用されます（パラメータ「位置姿勢の細分化」にチェックを入れた場合）。
- **出力：**
 1. 座標系変換後の位置姿勢。

4.3.227. 2D経路を読み込む



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

ファイルから2D経路点を読み込み、位置姿勢で表示される3D位置姿勢にマッピングします。

使用シーン

通常、経路の応用シーンで使用されます。

入力と出力

● 入力：

1. 対象物を含む2D画像。
2. 対象物を含む深度画像。
3. 対象物を含む点群。

● 出力：

1. 計算によって得られた3D経路点。

4.3.228. 校正球を位置決めする

機能

各入力点群に球（校正球）をフィッティングし、各校正球の中心位置姿勢と半径を出力するために使用されます。

使用シーン

計測シーンで、シーンに校正球を位置決めすることで、カメラとロボットの温度ドリフトを補正するために使用されます。

入力と出力



パラメータの説明

パラメータはありません。

4.3.229. 位置姿勢の軸の方向を基準点に指向



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

基準位置に向くように、入力位置姿勢の指定軸を回転させます。

使用シーン

汎用の位置姿勢回転ステップです。特定の使用シーンはありません。

入力と出力

● 入力：

1. このポートに入力される位置姿勢の指定軸は、基準方向に向くように回転されます。
2. 位置姿勢を回転するための基準位置（オプション。このポートが使用されていない場合、パラメータの「基準位置設定」を使用します）。

● 出力：

1. 回転後に基準位置を指す位置姿勢。

4.3.230. 2Dテンプレートを作成

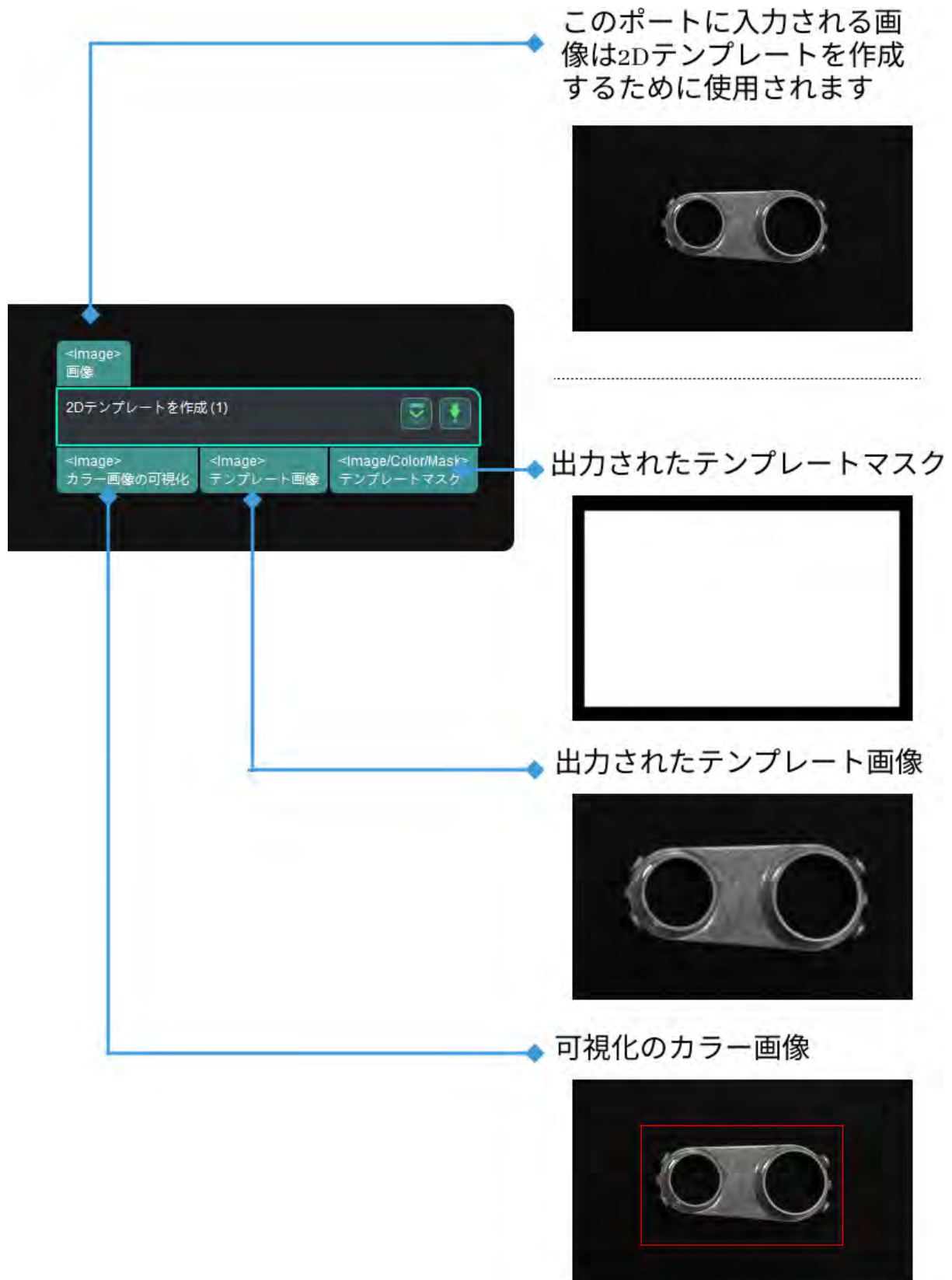
機能

描画エリアで入力画像に2D関心領域を選択します。この関心領域とそのマスクは、テンプレートとテンプレートマスクとして出力されます。

使用シーン

通常、計測シーンでステップ [2Dテンプレートマッチング](#) のための2Dテンプレートを作成し、2D画像内の対象物の2D位置姿勢を計算するために使用されます。

入力と出力

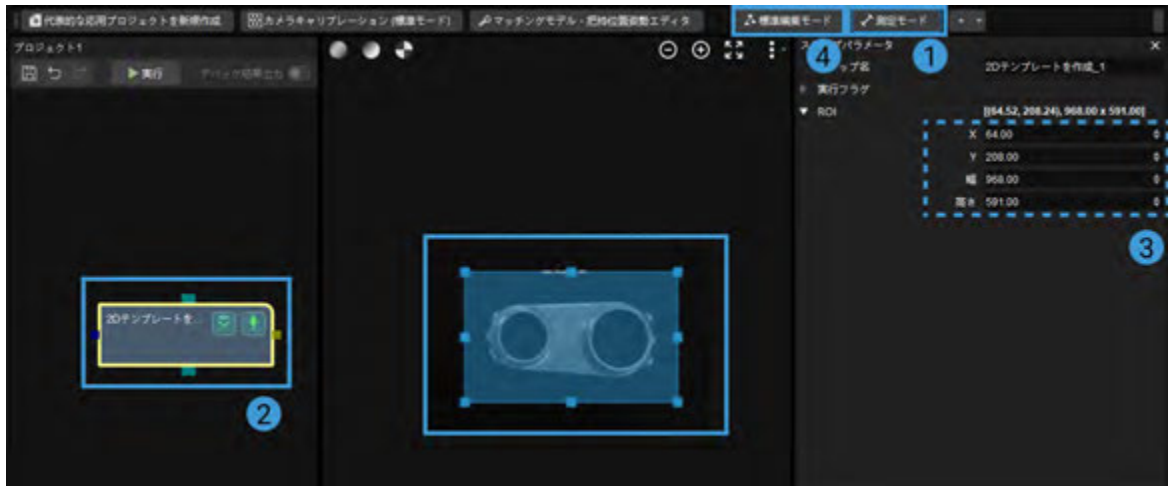


パラメータの説明

ROI

初期値： [(-1.00, -1.00) , 0.00 × 0.00]

調整説明：ツールバーで[測定モード]をクリックして測定モードに入ります。ステップ**2Dテンプレートを作成**を選択し、選択ボックスの頂点をドラッグしてサイズを調整して選択ボックスをドラッグして位置を調整するか、右側のROI値を調整して選択ボックスの位置とサイズを調整します。選択ボックスの位置を設定した後、標準編集モードに戻ります。



4.3.231. 深度画像をカラー画像にマッピング

機能

深度画像のピクセル座標をカラー画像に対応するように調整します。

使用シーン

特定のシーン向けのカスタマイズのステップです。

入力と出力

● 入力：

1. このポートに入力される深度画像は、カラー画像のピクセル座標に対応するように調整されます。
2. 深度画像が対応するカラー画像。
3. 深度画像が対応する点群。

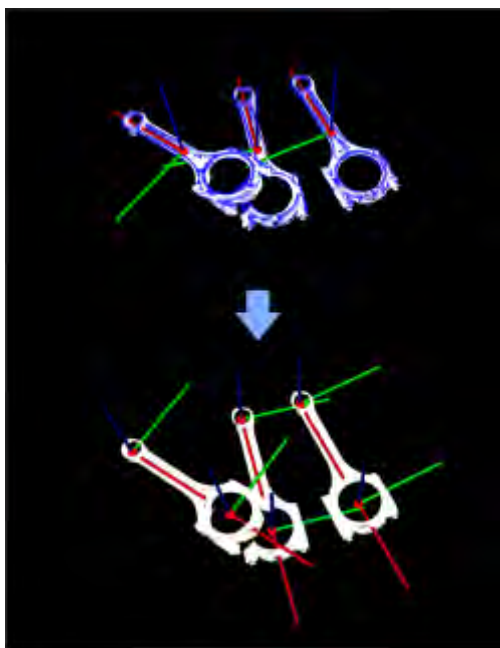
● 出力：

1. 調整された深度画像。

4.3.232. 複数の把持位置姿勢にマッピング

機能

物体の幾何学的中心と設定されたマッピング関係を使用して、物体の他の把持位置姿勢を計算します。



使用シーン

把持位置姿勢が幾何学的中心と一致しない場合、または複数の把持位置姿勢がある場合にこのステップが必要です。以下の特定方法を使用してシーン内の対象物にロボット制御ソフトウェアに送信する複数の把持位置姿勢を追加してください。

このステップから出力された「把持位置姿勢」は、[\[vision-steps:transform-poses::transform-poses\]](#) ステップの「基準位置姿勢」ポートにつながする必要があります。

このステップから出力された「位置姿勢オフセット」は、[\[vision-steps:transform-poses::transform-poses\]](#) ステップの「元の位置姿勢」ポートにつながする必要があります。

入力と出力



パラメータの説明

位置姿勢ファイル

幾何学的中心ファイル

モデルに対象物の幾何学的中心の位置姿勢

調整説明：幾何学的中心ファイルを保存する絶対パスあるいは相対パス。

配置位置姿勢ファイル

対象物の配置位置姿勢

調整説明：配置位置姿勢ファイルを保存する絶対パスあるいは相対パス。

把持位置姿勢ファイル

モデルに対象物の把持位置姿勢

調整説明：把持位置姿勢ファイルを保存する絶対パスあるいは相対パス。

ラベルのインポート (オプション)

位置姿勢ラベルファイル

把持位置姿勢に対応するラベルファイル

調整説明：ラベルファイルを保存する絶対パスあるいは相対パス。

4.3.233. マスクによって画像領域を抽出



これは古いバージョンのステップであり、メンテナンスが停止されています。このステッ

プに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

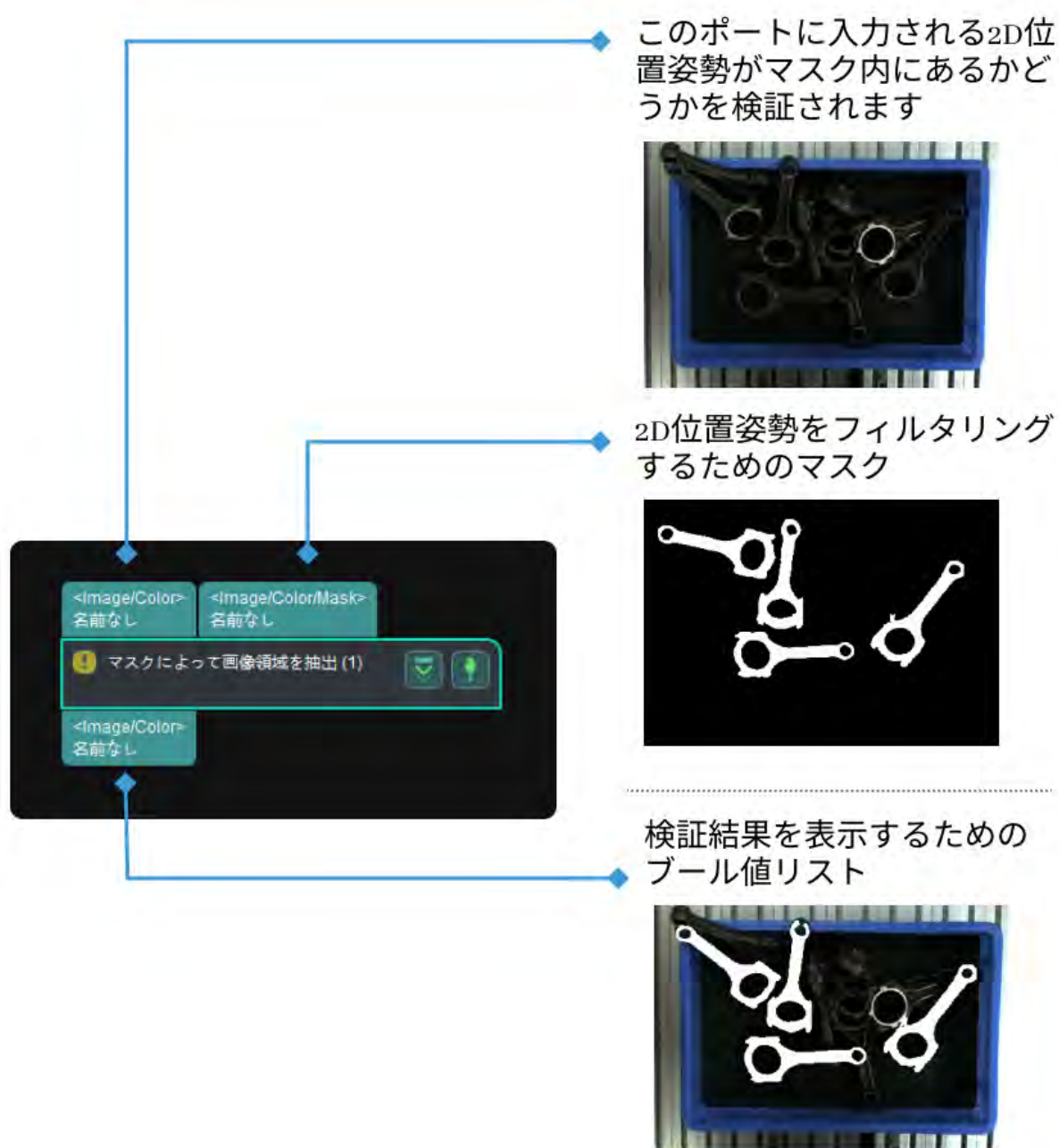
機能

入力された2D画像からマスクで覆われている領域を抽出し、残りのピクセル値を0に設定します。

使用シーン

特定のシーン向けのカスタマイズのステップです。

入力と出力



4.3.234. マスククラスタリング

機能

マスク内のピクセル値がゼロではない点をクラスタリングします。

使用シーン

画像内の同じ対象物に属する領域を抽出するためによく使われます。

入力と出力

- **入力：** このポートに入力されるリストのマスクがクラスタリングされます。
- **出力：** クラスタリングされたマスクリスト。

4.3.235. マスクフィルタリング



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

各領域のピクセル数によってマスク領域をフィルタリングします。

使用シーン

汎用のマスクフィルタリングのステップです。特定の使用シーンはありません。

入力と出力

- **入力：** このポートに入力されたマスクリストはフィルタリングされます。
- **出力：** フィルタリング後のマスクリスト。

4.3.236. マスクをメッシュ化



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

各入力マスクに対して、マスクをメッシュ化し、有効な点を持つ複数のマスクを生成します。

使用シーン

特定のシーン向けのカスタマイズのステップです。

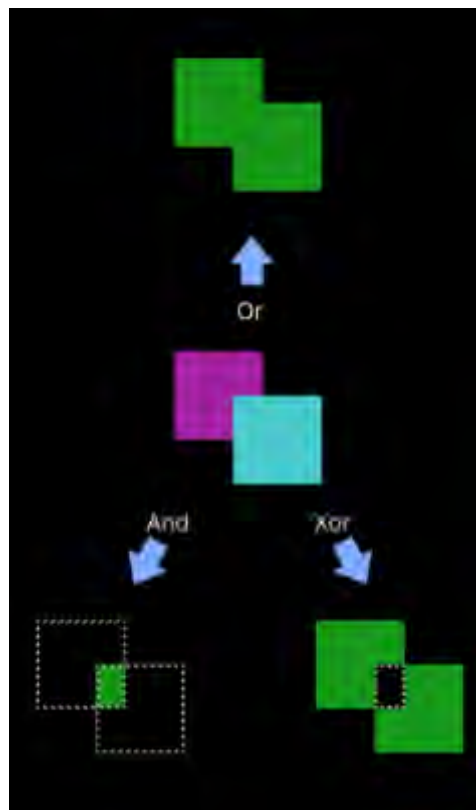
入力と出力

- **入力：** このポートに入力されるマスクは、メッシュ化されます。
- **出力：** メッシュ化されたマスク。

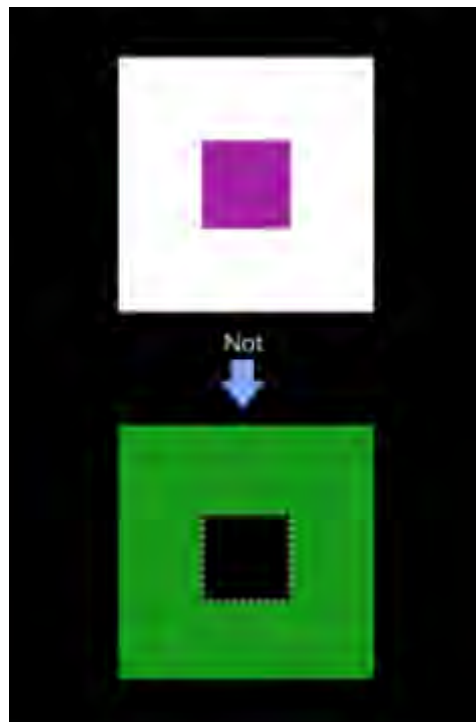
4.3.237. マスクの論理演算

機能

同じ画像サイズの2つのグループのマスクリストに対して論理演算を実行します。



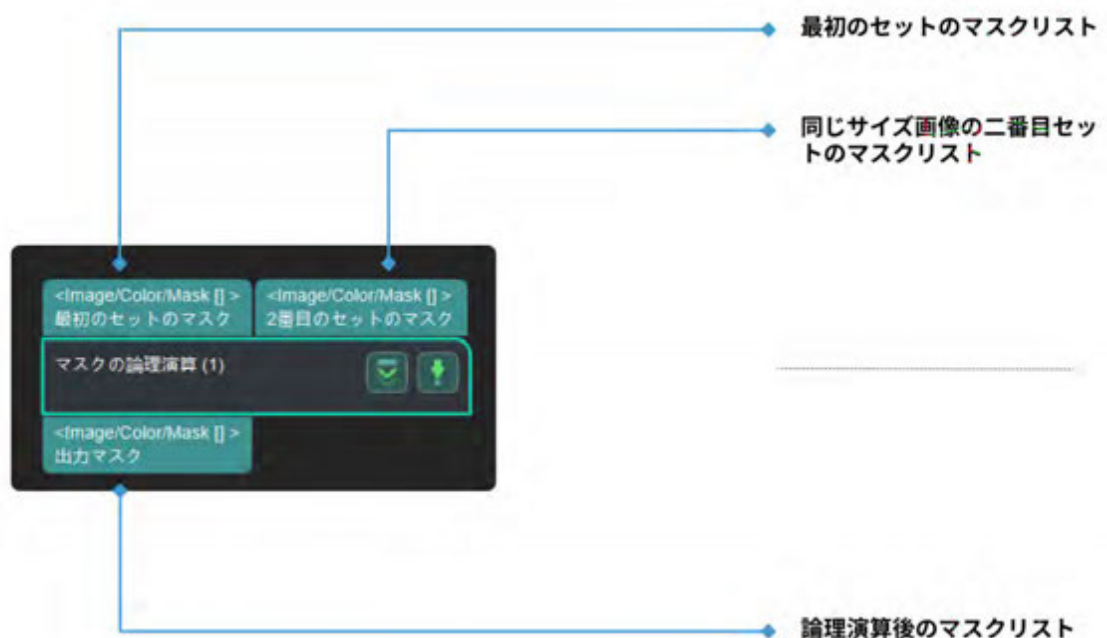
1つのマスクしかない場合は、逆演算（Not）は自身のマスク画像に対してのみ可能です。



使用シーン

汎用のマスクリストの論理演算ステップで、特定使用シーンはありません。

入力と出力



パラメータの説明

演算設定

論理演算

パラメータ説明：このパラメータは、マスクの論理演算の操作を選択するために使用されます。

オプション：

- And：初期値。2つのマスクの同じ部分を抽出します。
- Or：入力した2つのマスクをマージします。
- Xor：2つのマスクの同じ部分を除去します。
- Not：入力した1つのマスクに対して逆演算を実行します。黒い部分（通常は背景）が白くなり、白い部分（通常は対象物の輪郭またはエッジ）が黒くなります。

入力マスクの使用設定



論理演算が **Not** の場合、このパラメータは設定できません。

マッピングタイプ：

パラメータ説明：このパラメータは、入力された2つのマスクリストの対応関係を指定するために使用されます。

オプション：

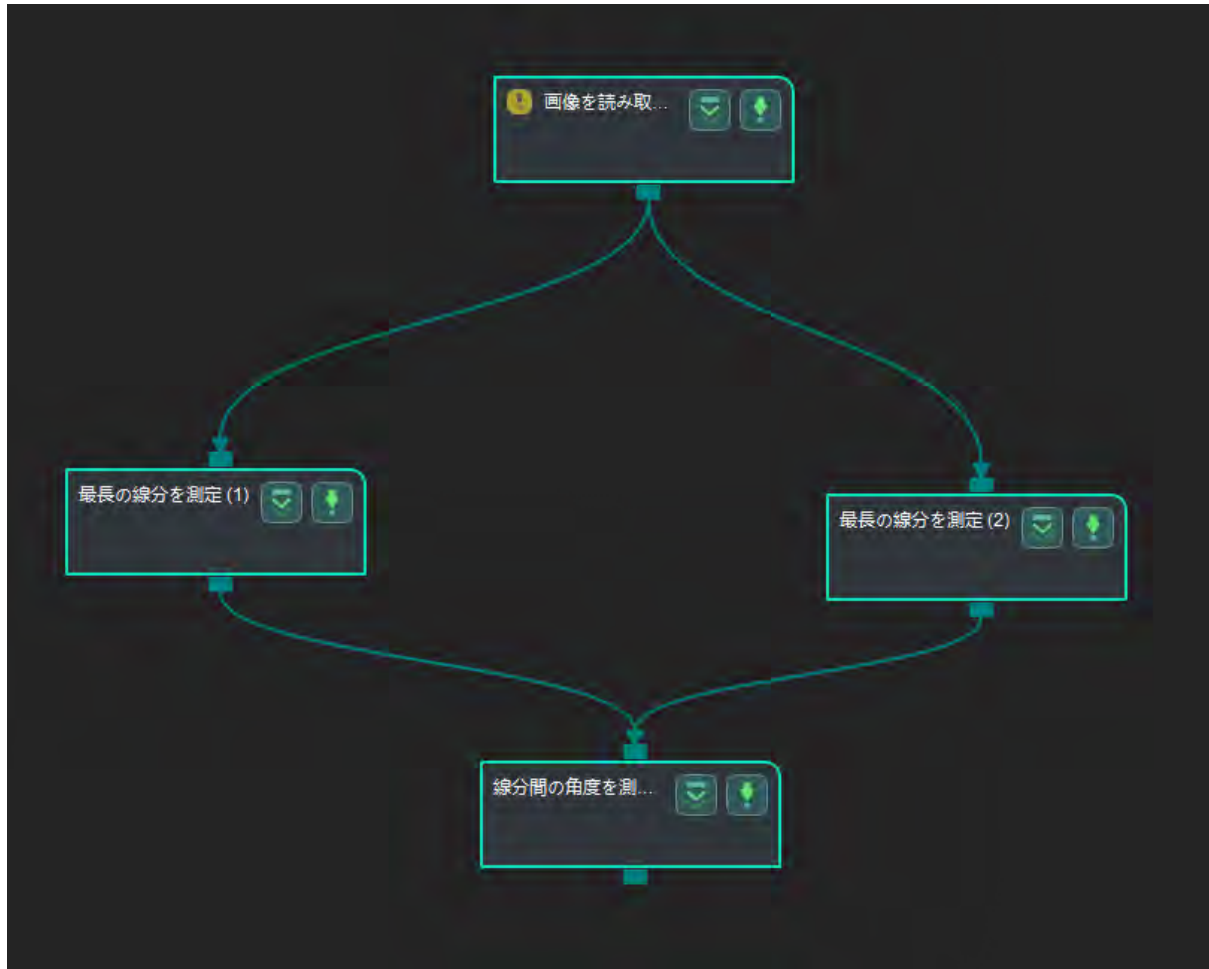
- FirstToAll：2番目グループの最初のマスクのみを使用して1番目グループのマスクに対応します。
- OneToOne：初期値。2番目グループのマスクをすべて使用して1番目グループのマスクと1対1で対応します。

4.3.238. 線分間の角度を測定

このステップは線分間の角度を測定するために使用されます。

このステップを実行する前に、[\[vision-measure-mode:getting-started:::start-use-measure-mode\]](#) をお読みください。

例



[vision-steps:read-images-v2:::read-images-v2]、最長の線分を測定、線分間の角度を測定 ステップと併用して線分間の角度を測定します。

ステップをつなぐ

1. [vision-steps:measure-longest-line-segment:::measure-longest-line-segment] (1) をクリックして、**ステップの入力・出力ポートの選択** 画面で「入力1(カラー画像)」のドロップダウンバーで「画像を読み取る_1_カラー画像」を選択します。
2. [vision-steps:measure-longest-line-segment:::measure-longest-line-segment] (2) をクリックして、**ステップの入力・出力ポートの選択** 画面で「入力1(カラー画像)」のドロップダウンバーで「画像を読み取る_1_カラー画像」を選択します。
3. **線分間の角度を測定** をクリックして、**ステップの入力・出力ポートの選択** 画面で「入力2(線分)」のドロップダウンバーで「最長の線分を測定_1_線分」を選択して「入力3(線分)」のドロップダウンバーで「最長の線分を測定_2_線分」を選択します。

パラメータの調整と結果の表示

- [vision-steps:read-images-v2:::read-images-v2] 設定：

このステップをクリックして **ステップパラメータ** 画面で、検出する対象物の画像のパスなどを

設定します（ユーザーは対象物を含むカラー画像を用意する必要があります）。[実行]をクリックしてプロジェクトを実行します。

- [\[vision-steps:measure-longest-line-segment:::measure-longest-line-segment\]](#) (1) 設定:

下図に示すように、このステップをクリックしてROIを選択します。



- [\[vision-steps:measure-longest-line-segment:::measure-longest-line-segment\]](#) (2) 設定:

下図に示すように、このステップをクリックしてROIを選択します。



- **測定結果の出力** または **結果ビュー** 画面で、線分間の角度（単位は°）を確認できます。また、上限・下限を設定することで測定の値が合格したかどうかを確認できます。



4.3.239. 円の測定

このステップは、円の半径を測定します。

このステップを実行する前に、[\[vision-measure-mode:getting-started::start-use-measure-mode\]](#) をお読みください。

例



[\[vision-steps:read-images-v2:::read-images-v2\]](#)、[円の測定](#) と併用してROI内の円を見つけてその半径を測定します。

ステップをつなぐ

[円の測定](#) をクリックして、**ステップの入力・出力ポートの選択** 画面で「入力1（カラー画像）」のドロップダウンバーで「画像を読み取る_1_カラー画像」に設定します。

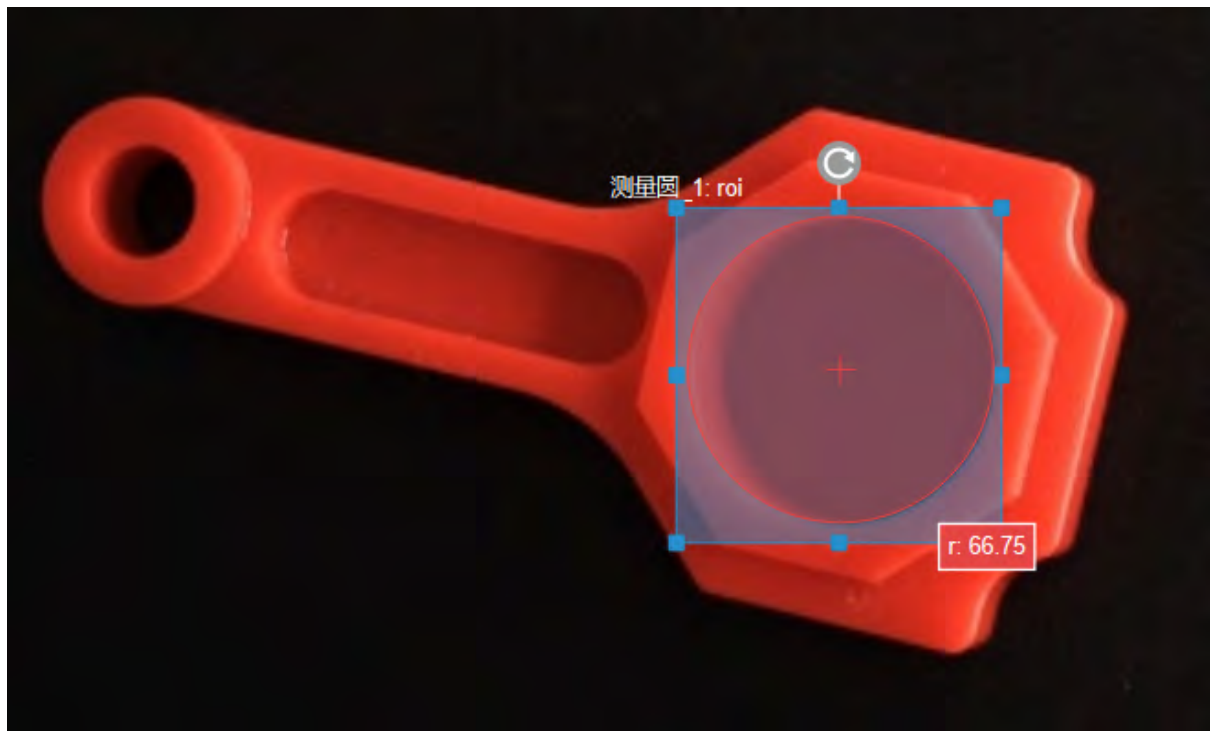
パラメータの調整と結果の表示

- [\[vision-steps:read-images-v2:::read-images-v2\]](#) 設定：

このステップをクリックして **ステップパラメータ** 画面で、検出する対象物の画像のパスなどを設定します（ユーザーは対象物を含むカラー画像を用意する必要があります）。設定を完了したら、**[実行]** をクリックしてプロジェクトを実行します。

- [円の測定](#) 設定：

下図に示すように、このステップをクリックしてROIを選択します。



- **測定結果の出力** または **結果ビュー** 画面で、円の半径（単位はピクセル）を確認できます。また、上限・下限を設定することで測定値が合格したかどうかを確認できます。

4.3.240. 円から円までの距離を測定

このステップは、円と円との距離を測定するために使用されます。

このステップを実行する前に、[\[vision-measure-mode:getting-started:::start-use-measure-mode\]](#) をお読みください。

例



[vision-steps:read-images-v2:::read-images-v2]、円の測定、円から円までの距離を測定 を併用して点から点までの距離を測定します。

ステップをつなぐ

1. 円の測定 (1) をクリックして、ステップの入力・出力ポートの選択 画面で「入力1(カラー画像)」のドロップダウンバーで「画像を読み取る_1_カラー画像」を選択します。
2. 円の測定 (2) をクリックして、ステップの入力・出力ポートの選択 画面で「入力1(カラー画像)」のドロップダウンバーで「画像を読み取る_1_カラー画像」を選択します。
3. 円から円までの距離を測定 をクリックして、ステップ入力ポート選択 画面で、「入力2(円)」のドロップダウンバーで「円の測定_1_円」を選択し、「入力3(円)」のドロップダウンバーで「円の測定_2_円」を選択します。

パラメータの調整と結果の表示

- [vision-steps:read-images-v2:::read-images-v2] 設定：

このステップをクリックして **ステップパラメータ** 画面で、検出する対象物の画像のパスなどを設定します（ユーザーは対象物を含むカラー画像を用意する必要があります）。設定を完了したら、[実行] をクリックしてプロジェクトを実行します。

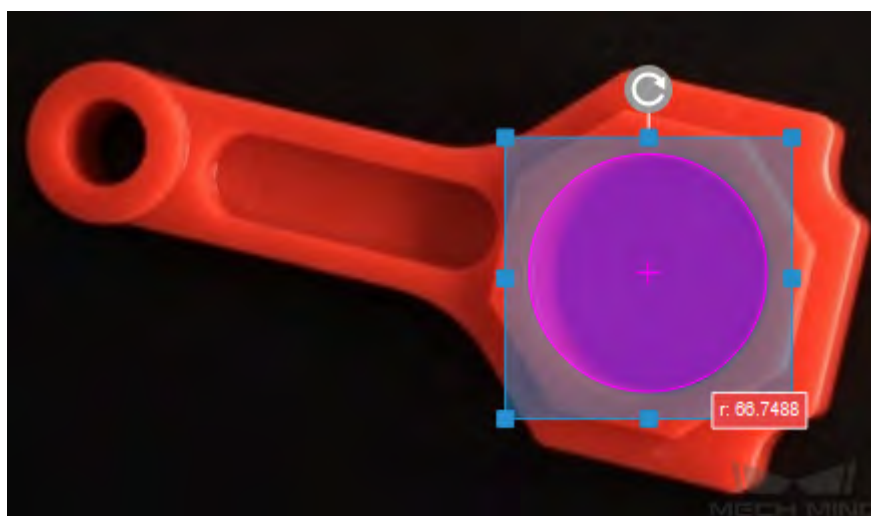
- 円の測定 (1) の設定：

このステップをクリックしてROIを選択します。



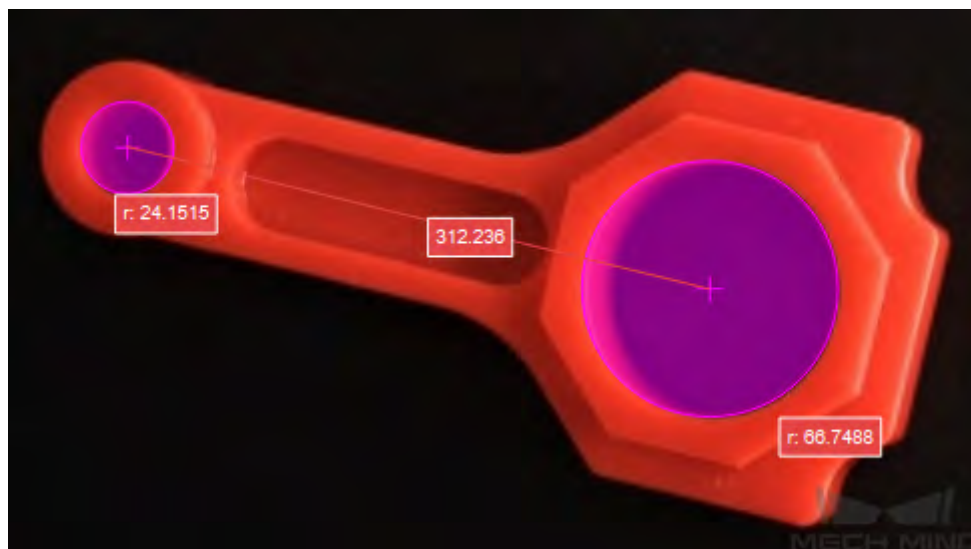
- 円の測定 (2) の設定：

このステップをクリックしてROIを選択します。

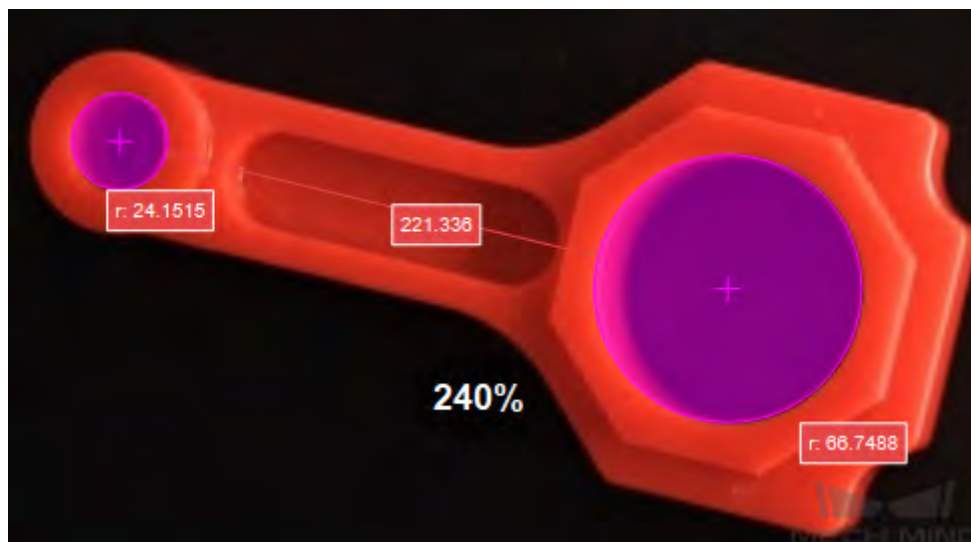


- 円から円までの距離を測定 の設定：

このステップをクリックして、描画エリアと測定結果出力画面で、デフォルトで二つの円の円心の距離（ピクセル単位）を表示します。



このステップのパラメータで、「距離モード」を「最短距離」に変更できます。プロジェクトを実行してから機能します。



- ユーザーは **測定結果の出力** または **結果ビュー** 画面で上限・下限を設定することで測定値が合格したかどうかを確認できます。

4.3.241. 円から線分までの距離を測定

このステップは、円から線分までの距離を測定するために使用されます。

このステップを実行する前に、[\[vision-measure-mode:getting-started:::start-use-measure-mode\]](#) をお読みください。

例



[[vision-steps:read-images-v2:::read-images-v2](#)]、[円の測定](#)、[最長の線分を測定](#)、[円から線分までの距離を測定](#)を併用して円から線分までの距離を測定します。

ステップをつなぐ

1. [円の測定](#) をクリックして、**ステップの入力・出力ポートの選択** 画面で「入力1(カラー画像)」のドロップダウンバーで「画像を読み取る_1_カラー画像」を選択します。
2. [[vision-steps:measure-longest-line-segment:::measure-longest-line-segment](#)] をクリックして、**ステップの入力・出力ポートの選択** 画面で「入力1(カラー画像)」のドロップダウンバーで「画像を読み取る_1_カラー画像」を選択します。
3. [円から線分までの距離を測定](#) をクリックして、**ステップ入力ポート選択** 画面で、「入力2(円)」のドロップダウンバーで「円の測定_1_円」を選択し、「入力3(線分)」のドロップダウンバーで「最長の線分を測定_1_線分」を選択します。

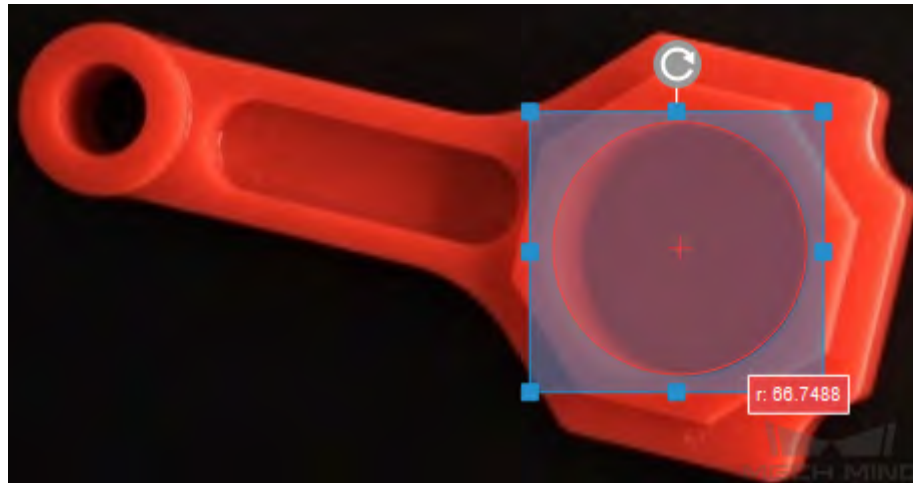
パラメータの調整と結果の表示

- [[vision-steps:read-images-v2:::read-images-v2](#)] 設定：

このステップをクリックして **ステップパラメータ** 画面で、検出する対象物の画像のパスなどを設定します（ユーザーは対象物を含むカラー画像を用意する必要があります）。設定を完了したら、[実行] をクリックしてプロジェクトを実行します。

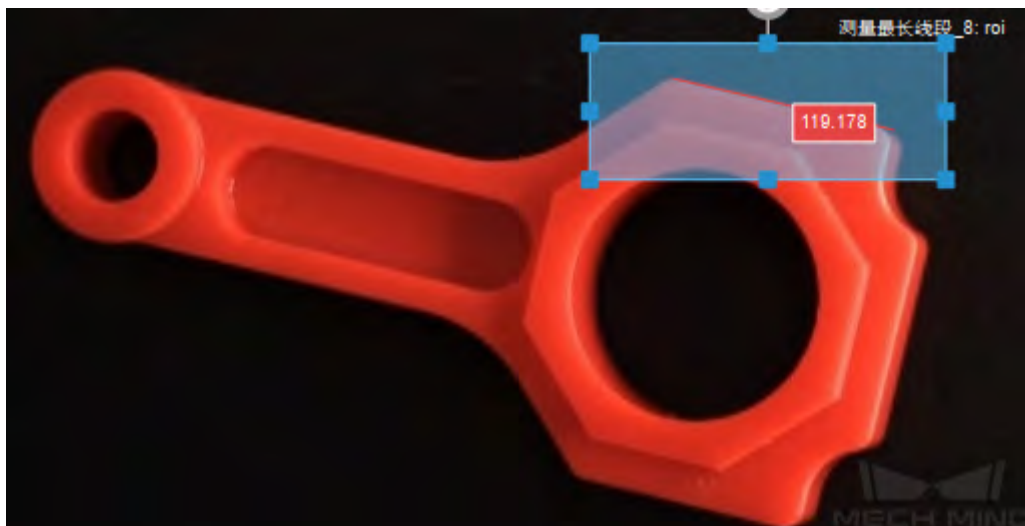
- [円の測定](#) 設定：

このステップをクリックしてROIを選択します。



- [\[vision-steps:measure-longest-line-segment:::measure-longest-line-segment\]](#) 設定：

このステップをクリックしてROIを選択します。

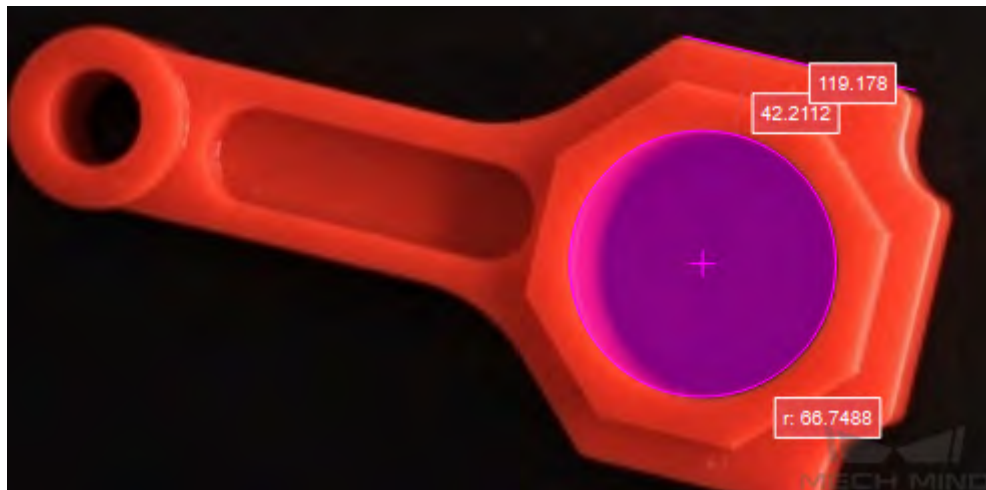


- [円から線分までの距離を測定](#) 設定：

このステップをクリックして、描画エリアと測定結果出力画面で、デフォルトで二つの円の円心の距離（ピクセル単位）を表示します。



このステップのパラメータで、「距離モード」を「最近距離」に変更できます。プロジェクトを実行してから機能します。



- ユーザーは **測定結果の出力** または **結果ビュー** 画面で上限・下限を設定することで測定値が合格したかどうかを確認できます。

4.3.242. 点から円までの距離を測定

このステップは、点から円までの距離を測定するために使用されます。

このステップを実行する前に、[\[vision-measure-mode:getting-started:::start-use-measure-mode\]](#) をお読みください。



現在、任意の点の測定に対応できません。 **円の測定** または **円心検出** によって得られた点の測定のみを実行できます。

例



[vision-steps:read-images-v2:::read-images-v2]、円の測定、点から円までの距離を測定 を併用して点から円までの測定を実行します。

ステップをつなぐ

1. 円の測定 (1) をクリックして、ステップの入力・出力ポートの選択 画面で「入力1(カラー画像)」のドロップダウンバーで「画像を読み取る_1_カラー画像」を選択します。
2. 円の測定 (2) をクリックして、ステップの入力・出力ポートの選択 画面で「入力1(カラー画像)」のドロップダウンバーで「画像を読み取る_1_カラー画像」を選択します。
3. 点から円までの距離を測定 をクリックして、ステップの入力・出力ポートの選択 画面で、「入力2(点)」のドロップダウンバーで「円の測定_1_円」を選択し、「入力3(円)」のドロップダウンバーで「円の測定_2_円」を選択します。

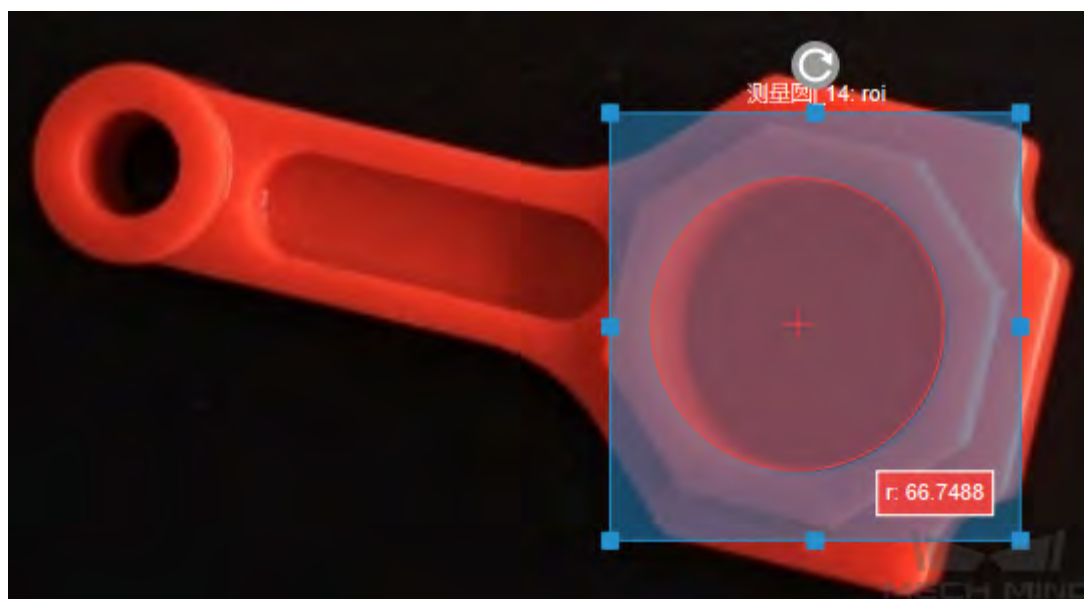
パラメータの調整と結果の表示

- [vision-steps:read-images-v2:::read-images-v2] 設定：

このステップをクリックして **ステップパラメータ** 画面で、検出する対象物の画像のパスなどを設定します（ユーザーは対象物を含むカラー画像を用意する必要があります）。設定を完了したら、**[実行]** をクリックしてプロジェクトを実行します。

- 円の測定 (1) の設定：

このステップをクリックしてROIを選択します。



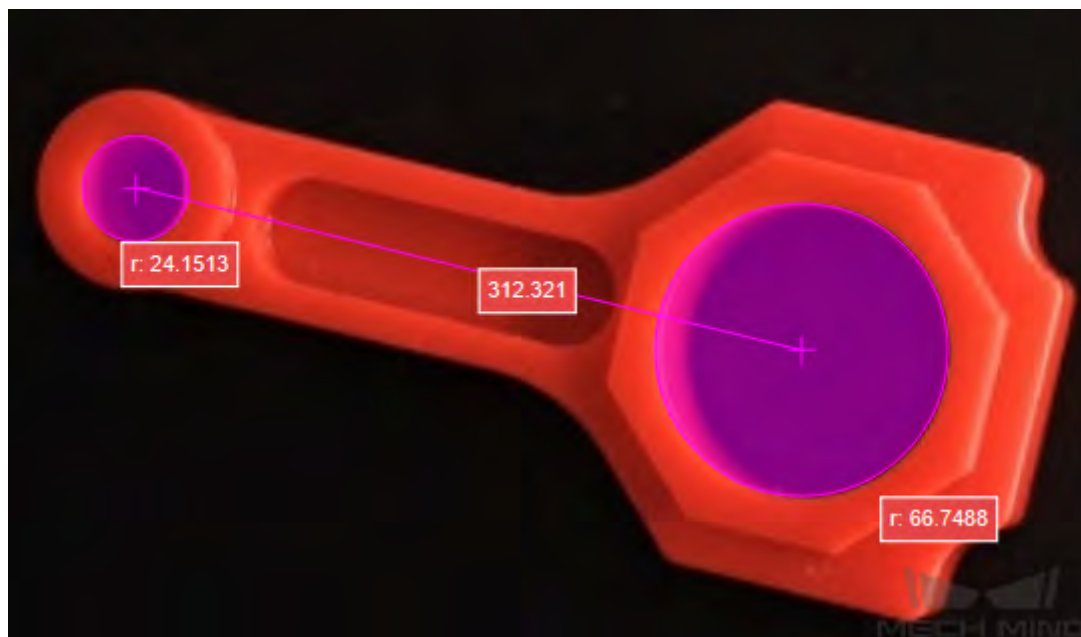
● 円の測定 (2) の設定：

このステップをクリックしてROIを選択します。



● 点から円までの距離を測定 の設定：

このステップをクリックして、描画エリアと測定結果出力画面で、デフォルトで二つの円の円心の距離（ピクセル単位）を表示します。



- ユーザーは **測定結果の出力** または **結果ビュー** 画面で上限・下限を設定することで測定値が合格したかどうかを確認できます。

4.3.243. 点から点までの距離を測定

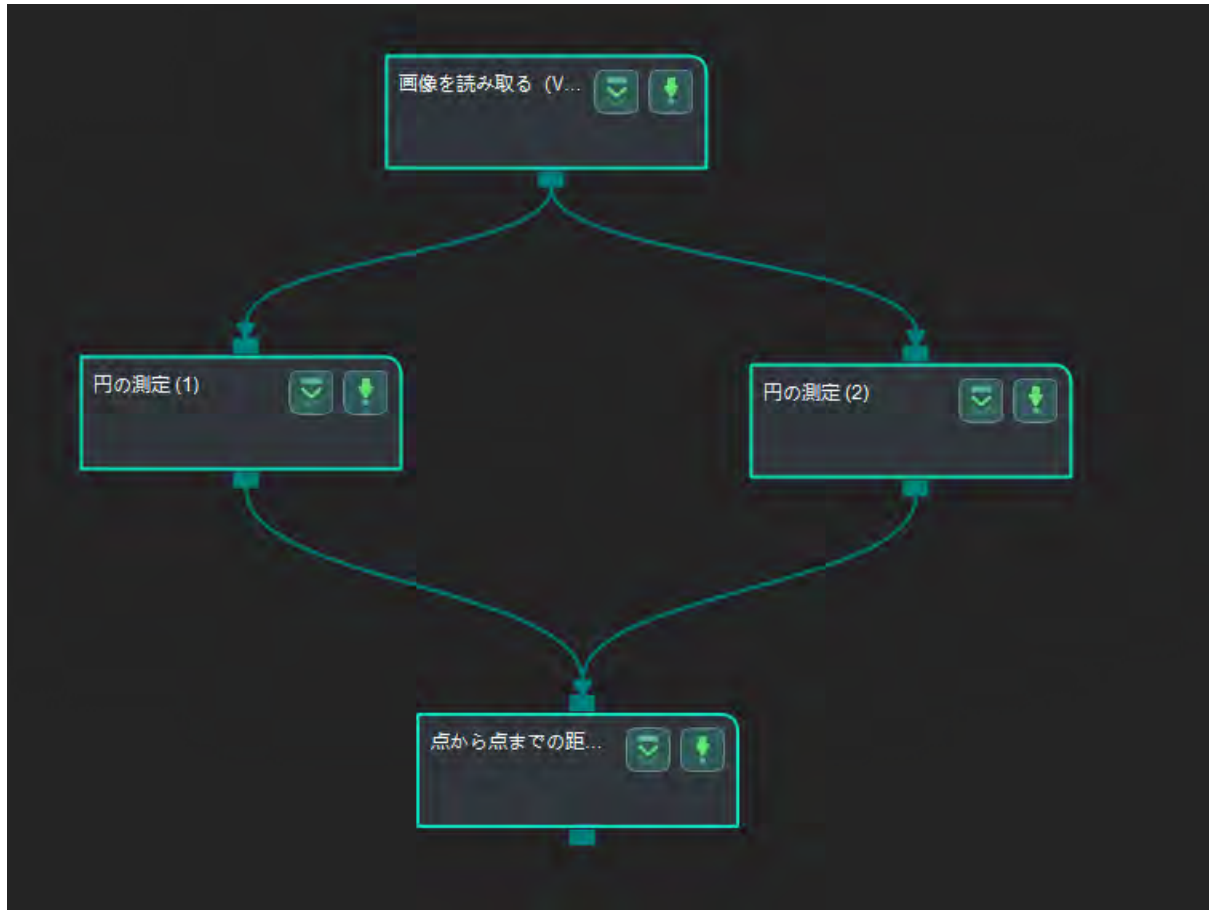
このステップは、点と点との距離を測定するために使用されます。

このステップを実行する前に、[\[vision-measure-mode:getting-started::start-use-measure-mode\]](#) をお読みください。



現在、任意の点の測定に対応できません。 **円の測定** または **円心検出** によって得られた点の測定のみを実行できます。

例



[vision-steps:read-images-v2:::read-images-v2]、円の測定、点から点までの距離を測定 を併用して点から点までの測定を実行します。

ステップをつなぐ

1. 円の測定 (1) をクリックして、ステップの入力・出力ポートの選択 画面で「入力1(カラー画像)」のドロップダウンバーで「画像を読み取る_1_カラー画像」を選択します。
2. 円の測定 (2) をクリックして、ステップの入力・出力ポートの選択 画面で「入力1(カラー画像)」のドロップダウンバーで「画像を読み取る_1_カラー画像」を選択します。
3. 点から点までの距離を測定 をクリックして、ステップの入力・出力ポートの選択 画面で、「入力2(点)」のドロップダウンバーで「円の測定_1_円」を選択し、「入力3(点)」のドロップダウンバーで「円の測定_2_円」を選択します。

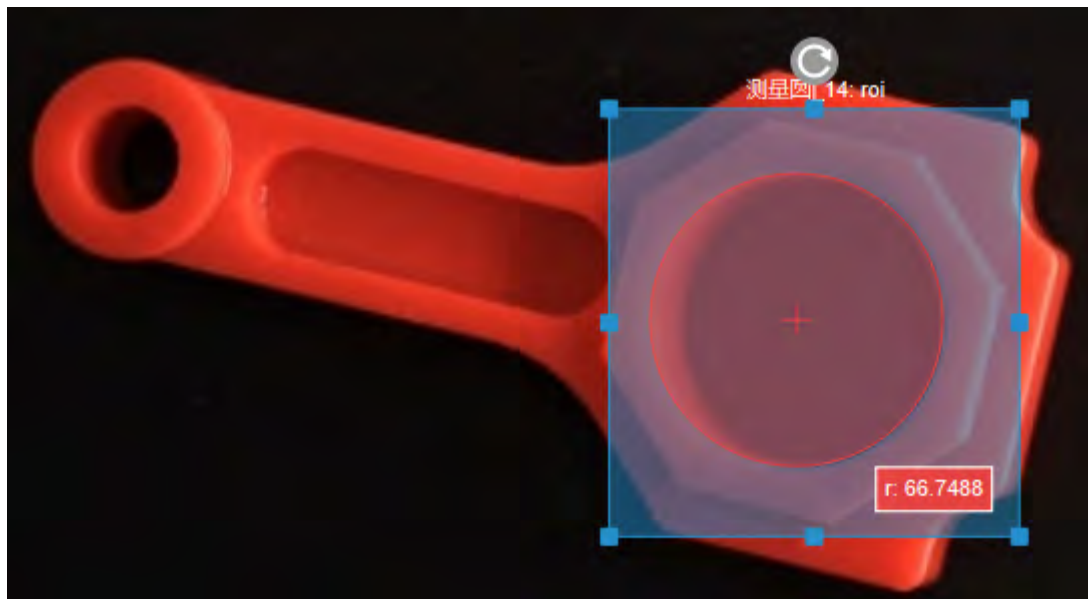
パラメータの調整と結果の表示

- [vision-steps:read-images-v2:::read-images-v2] 設定：

このステップをクリックして **ステップパラメータ** 画面で、検出する対象物の画像のパスなどを設定します（ユーザーは対象物を含むカラー画像を用意する必要があります）。設定を完了したら、[実行] をクリックしてプロジェクトを実行します。

- 円の測定 (1) の設定：

このステップをクリックしてROIを選択します。



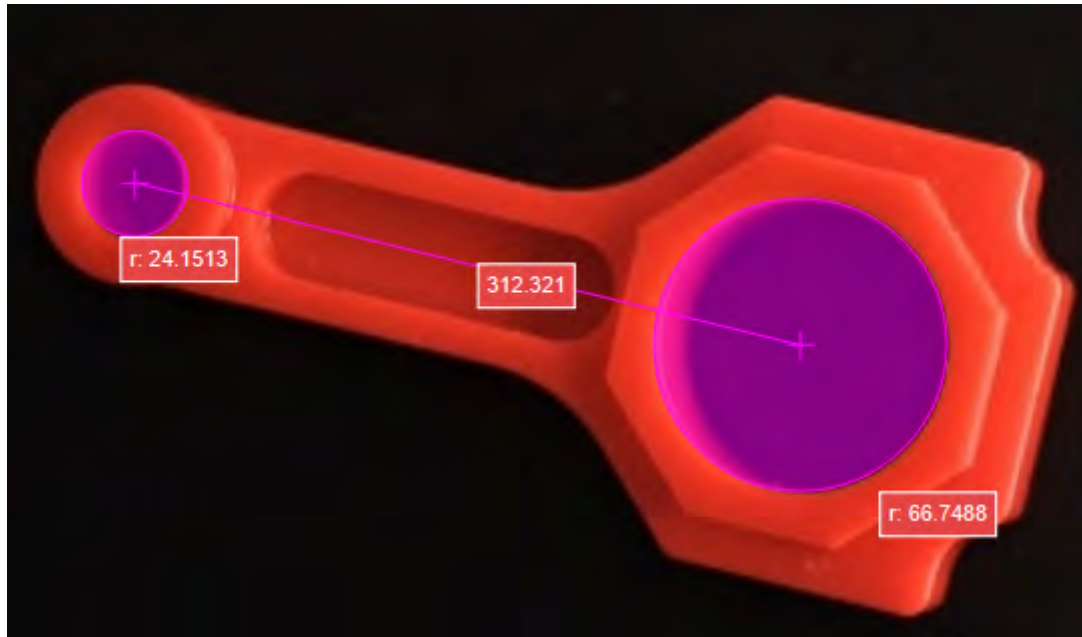
● 円の測定 (2) の設定：

このステップをクリックしてROIを選択します。



● 点から点までの距離を測定 の設定：

このステップをクリックして、描画エリアと測定結果出力画面で、デフォルトで二つの円の円心の距離（ピクセル単位）を表示します。



- ユーザーは **測定結果の出力** または **結果ビュー** 画面で上限・下限を設定することで測定値が合格したかどうかを確認できます。

4.3.244. 点から線分までの距離を測定

このステップは、点から線分までの距離を測定するために使用されます。

このステップを実行する前に、[\[vision-measure-mode:getting-started::start-use-measure-mode\]](#) をお読みください。



現在、任意の点の測定に対応できません。 [円の測定](#) または [円心検出](#) によって得られた点の測定のみを実行できます。

例



[[vision-steps:read-images-v2:::read-images-v2](#)]、[円の測定](#)、[最長の線分を測定](#)、[点から線分までの距離を測定](#) を併用して点から線分までの距離を測定します。

ステップをつなぐ

1. [円の測定](#) をクリックして、**ステップの入力・出力ポートの選択** 画面で「入力1(カラー画像)」のドロップダウンバーで「画像を読み取る_1_カラー画像」を選択します。
2. [[vision-steps:measure-longest-line-segment:::measure-longest-line-segment](#)] をクリックして、**ステップの入力・出力ポートの選択** 画面で「入力1(カラー画像)」のドロップダウンバーで「画像を読み取る_1_カラー画像」を選択します。
3. [点から線分までの距離を測定](#) をクリックして、**ステップの入力・出力ポートの選択** 画面で、「入力2(点)」のドロップダウンバーで「円の測定_1_円」を選択し、「入力3 (線分)」のドロップダウンバーで「最長の線分を測定_1_線分」を選択します。

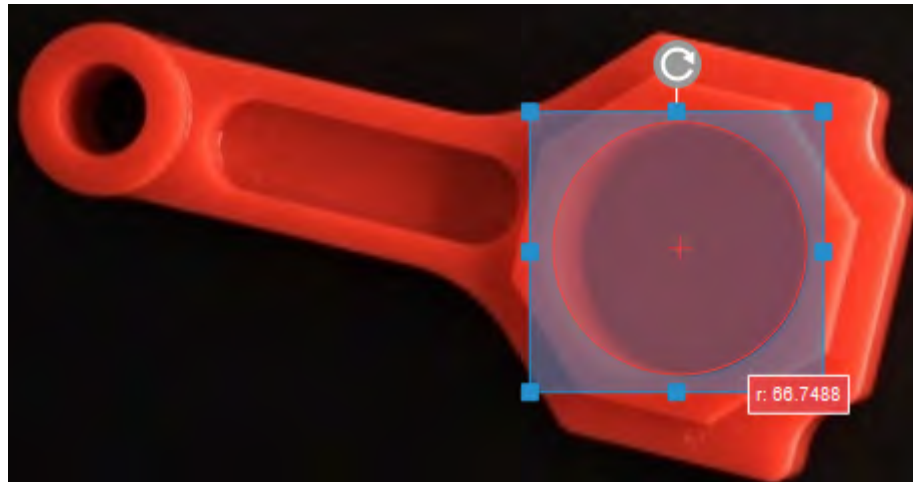
パラメータの調整と結果の表示

- [[vision-steps:read-images-v2:::read-images-v2](#)] 設定：

このステップをクリックして **ステップパラメータ** 画面で、検出する対象物の画像のパスなどを設定します（ユーザーは対象物を含むカラー画像を用意する必要があります）。設定後、[実行] をクリックしてプロジェクトを実行します。

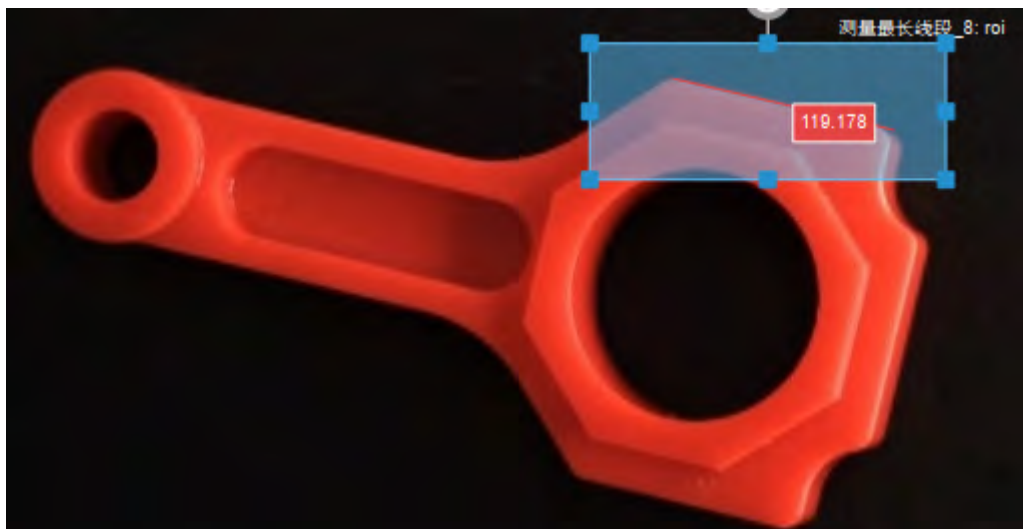
- **円の測定 設定：**

このステップをクリックしてROIを選択します。



- **[vision-steps:measure-longest-line-segment::measure-longest-line-segment] 設定：**

このステップをクリックしてROIを選択します。



- **点から線分までの距離を測定 設定：**

このステップをクリックして、描画エリアと測定結果出力画面で、デフォルトで二つの円の円心の距離（ピクセル単位）を表示します。



- ユーザーは **測定結果の出力** または **結果ビュー** 画面で上限・下限を設定することで測定値が合格したかどうかを確認できます。

4.3.245. 線分間の距離を測定

このステップは、線分間の距離を測定するために使用されます。

このステップを実行する前に、[\[vision-measure-mode:getting-started:::start-use-measure-mode\]](#) をお読みください。

例



[[vision-steps:read-images-v2:::read-images-v2](#)]、[最長の線分を測定](#)、[線分間の距離を測定](#) ステップと併用して線分間の距離を測定します。

ステップをつなぐ

1. [[vision-steps:measure-longest-line-segment:::measure-longest-line-segment](#)] (1) をクリックして、**ステップの入力・出力ポートの選択** 画面で「入力1(カラー画像)」のドロップダウンバーで「画像を読み取る_1_カラー画像」を選択します。
2. [[vision-steps:measure-longest-line-segment:::measure-longest-line-segment](#)] (2) をクリックして、**ステップの入力・出力ポートの選択** 画面で「入力1(カラー画像)」のドロップダウンバーで「画像を読み取る_1_カラー画像」を選択します。
3. [線分間の距離を測定](#) をクリックして、**ステップの入力・出力ポートの選択** 画面で、「入力2(線分)」のドロップダウンバーで「最長の線分を測定_1_線分」を選択し、「入力3(線分)」のドロップダウンバーで「最長の線分を測定_2_線分」を選択します。

パラメータの調整と結果の表示

- [[vision-steps:read-images-v2:::read-images-v2](#)] 設定：

このステップをクリックして **ステップパラメータ** 画面で、検出する対象物の画像のパスなどを設定します（ユーザーは対象物を含むカラー画像を用意する必要があります）。設定後、[実行] をクリックしてプロジェクトを実行します。

- [\[vision-steps:measure-longest-line-segment:::measure-longest-line-segment\]](#) (1) 設定
:

このステップをクリックしてROIを選択します。



- [\[vision-steps:measure-longest-line-segment:::measure-longest-line-segment\]](#) (2) 設定
:

このステップをクリックしてROIを選択します。



- [線分間の距離を測定](#) 設定:

このステップをクリックして、描画エリアと測定結果出力画面で、線分間の距離（ピクセル単位）を表示します。



- ユーザーは **測定結果の出力** または **結果ビュー** 画面で上限・下限を設定することで測定値が合格したかどうかを確認できます。

4.3.246. ギャップ幅を測定



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

2D画像で、対象物のあるギャップの幅を測定します。

使用シーン

特定のシーン向けのカスタマイズのステップです。

入力と出力

- **入力：**
 1. このポートに入力されるカラー画像は、可視化するための背景として使用されます。
 2. 対象物の2D位置姿勢（選択可能。テンプレートを元の画像に合わせるために使用されます。選択されたROIの画像コンテンツがテンプレートと一致する場合、入力は不要です）。
 3. 単一のピクセルで表示される実際のサイズ（選択可能。このポートが使用されていない場合、出力のサイズはピクセル単位です）。
- **出力：**
 1. 可視化するための画像。
 2. ギャップの2つのエッジを表示する線分。

3. ギャップの幅。

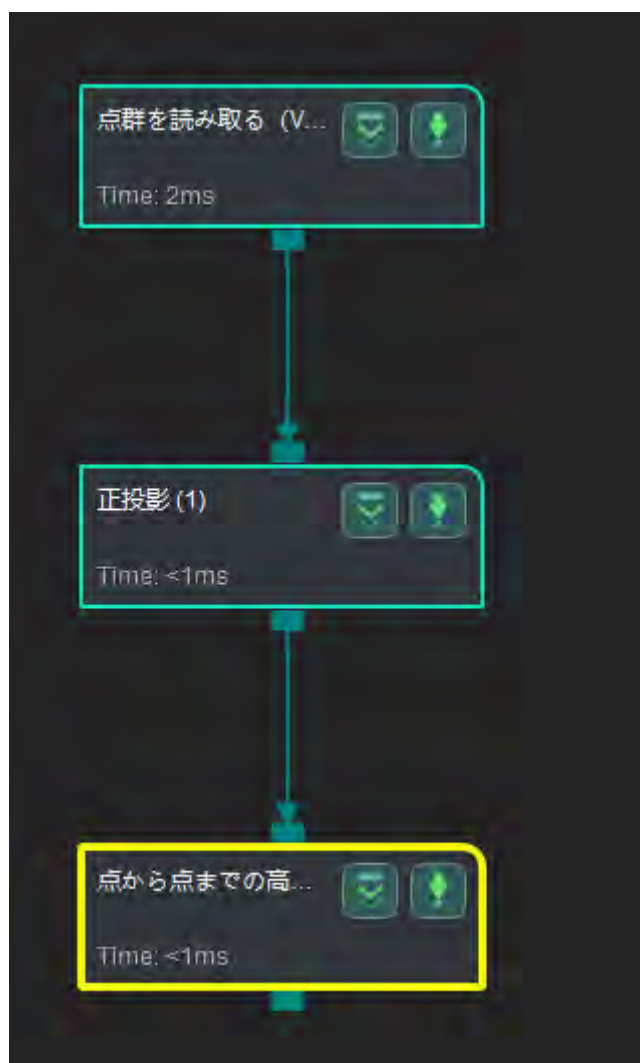
4.3.247. 点から点までの高さの差を測定

このステップは、二つの点の高さの差を計算します。

また、ユーザーが指定した二つの点の間に、線が生成されます。ビュープラグインに、この線の高さの変化が表示されます。

このステップを実行する前に、[\[vision-measure-mode:getting-started:::start-use-measure-mode\]](#) をお読みください。

使用例：



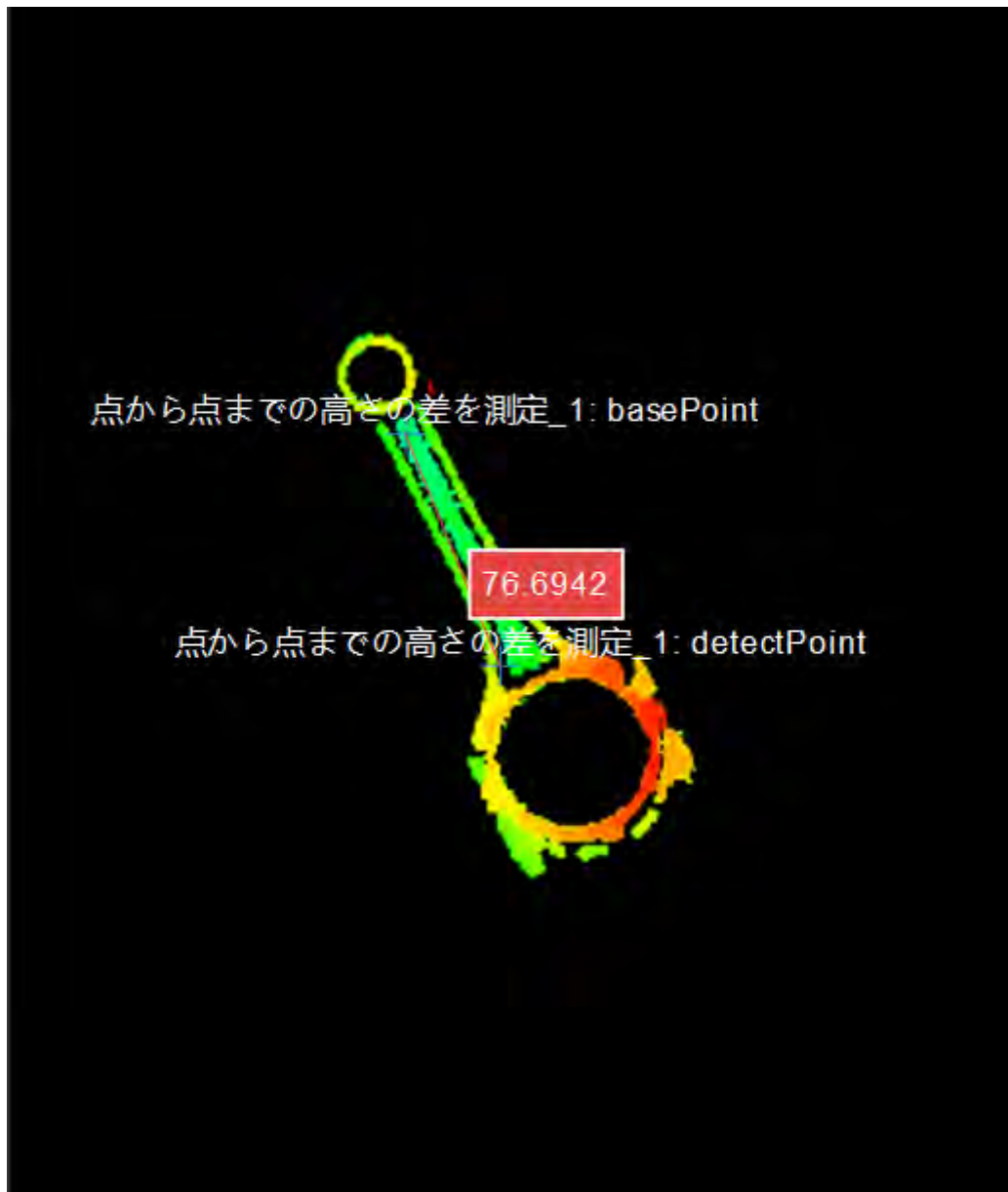
[\[vision-steps:read-point-cloud-v2:::read-point-cloud-v2\]](#)、[正投影](#)、[点から点までの高さの差を測定](#) を併用して [点 から 点 までの高度差の測定](#) を実行します。

ステップをつなぐ

1. [\[vision-steps:orthographic-projection:::orthographic-projection\]](#) をクリックして、**ステップの入力・出力ポートの選択** 画面で「入力1（法線ベクトル付きの点群）」のドロップダウンバーで「点群を読み取る_1_（法線ベクトル付きの点群）」に設定します。
2. [点から点までの高さの差を測定](#) をクリックして、**ステップの入力・出力ポートの選択** 画面で「入力1（可視化背景）」のドロップダウンバーで「正投影_1_深度画像」に設定します。

パラメータ設定

- [\[vision-steps:read-point-cloud-v2:::read-point-cloud-v2\]](#) 設定： このステップをクリックし、**ステップパラメータ** 画面で測定する対象物の点群ファイルパスなどを設定します（ユーザーは測定する対象物の点群を用意する必要があります）。設定を完了したら、**[実行]** をクリックしてプロジェクトを実行します。
- [点から点までの高さの差を測定](#) 設定： 一つの点を基準点で、もう一つの点は対象点です。下図に示すように、二つの点の元位置は画像の左上にあります。二つの点を指定した位置に移動します。



下図に示すように、**ステップパラメータ** 画面で、基準点と対象点の位置を調整できます。

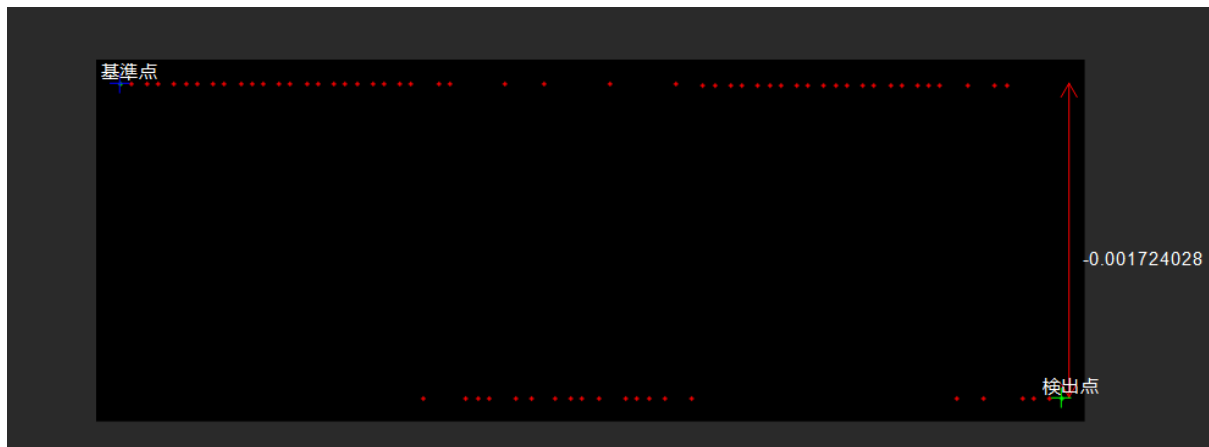
▼ 基準点	(122, 131)
X	122
Y	131
▼ 検出点	(150, 202)
X	150
Y	202

結果の確認

画面には、二つの点の距離が表示されます。**測定結果の出力** 画面で、二つの点の高さの差（単位はピクセル）が表示するします。

測定結果の出力				
名前	値	下限規格値	上限規格値	オフセット
点から点までの高さの差...	-0.002	0.000	0.000	0.000

下図に示すように、ビュープラグインには、二つの点を繋いだ線の高さの変化が表示されます。



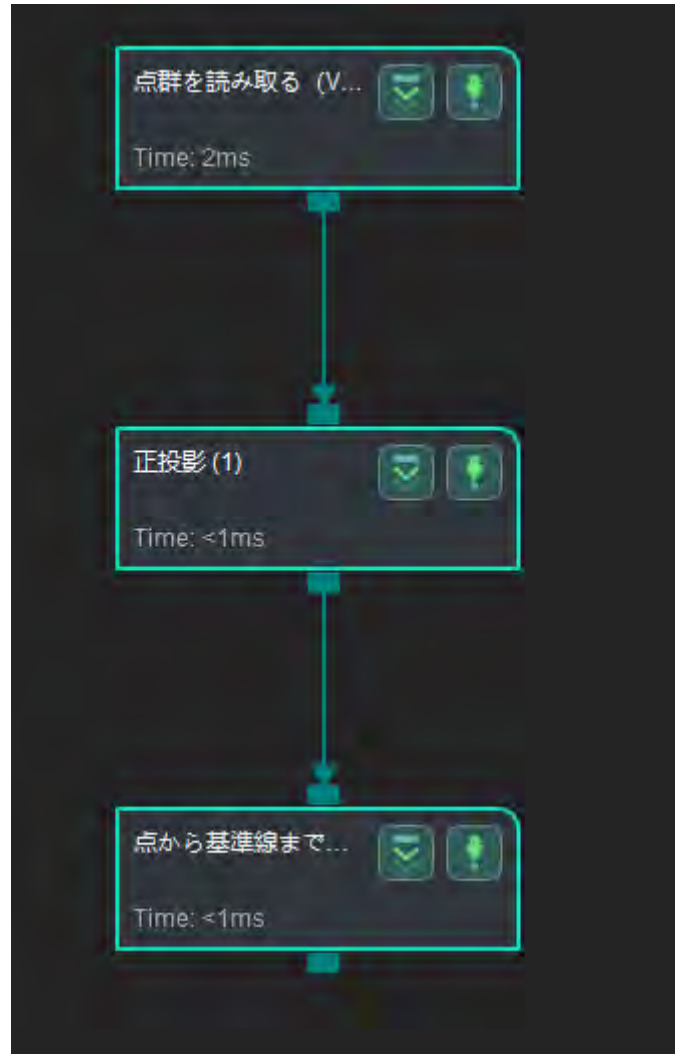
測定結果の出力 または 結果ビュー 画面で、上限・下限を設定することで測定の値が合格したかどうかを確認できます。

4.3.248. 点から基準線までの高さの差を測定

このステップは、点から基準線までの高さの差を計算します。

このステップを実行する前に、[測定モードを使ってみる](#) をお読みください。

例



点群を読み取る (V2) 、正投影、点から基準線までの高さの差を測定 を併用して点 から基準線 までの高さの差の測定を実行します。

ステップをつなぐ

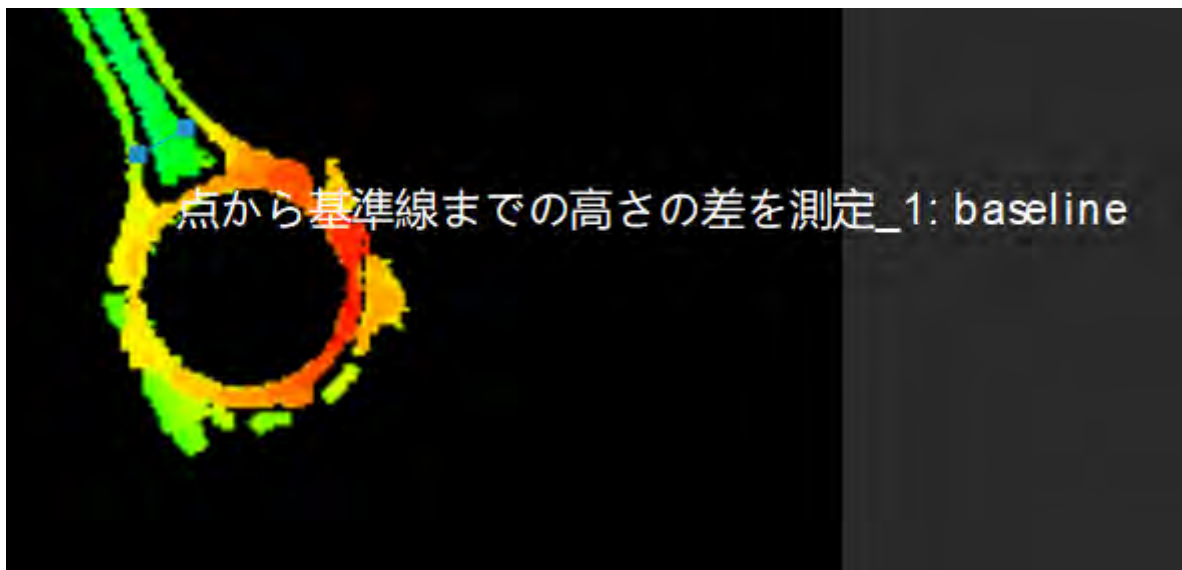
1. 正投影 をクリックして、ステップの入力・出力ポートの選択 画面で「入力1（法線ベクトル付きの点群）」のドロップダウンバーで「点群を読み取る_1_（法線ベクトル付きの点群）」に設定します。
2. 点から基準線までの高さの差を測定 をクリックして、ステップの入力・出力ポートの選択 画面で「入力1（可視化背景）」のドロップダウンバーで「正投影_1_深度画像」に設定します。

パラメータ設定

- 点群を読み取る (V2) 設定： このステップをクリックし、ステップパラメータ 画面で測定する対象物の点群ファイルパスなどを設定します（ユーザーは測定する対象物の点群を用意する必要があります）。設定を完了したら、[実行] をクリックしてプロジェクトを実行します。

点から基準線までの高さの差を測定 設定： 基準線設定：基準線の元位置は画面の左上にあ

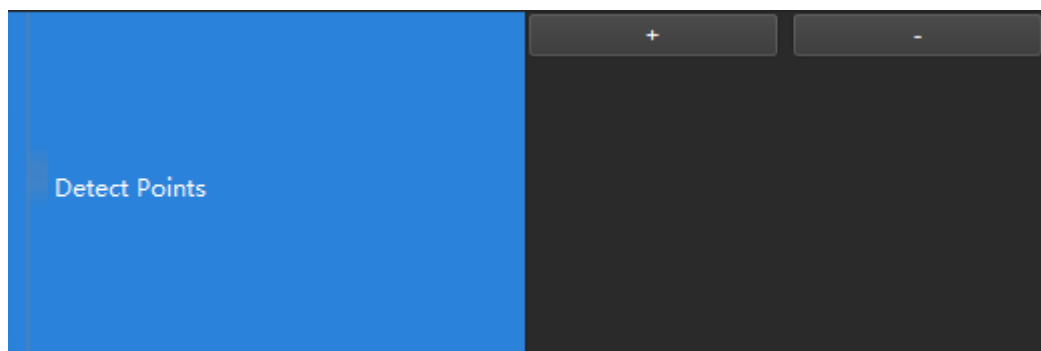
- ります。ユーザーが指定した位置に移動または調整することができます。



下図に示すように、**ステップパラメータ** 画面で基準線の位置を調整できます。

▼ Baseline		[(155, 141), (170, 223)]
X1		155
Y1		141
X2		170
Y2		223
		1: (169,244)

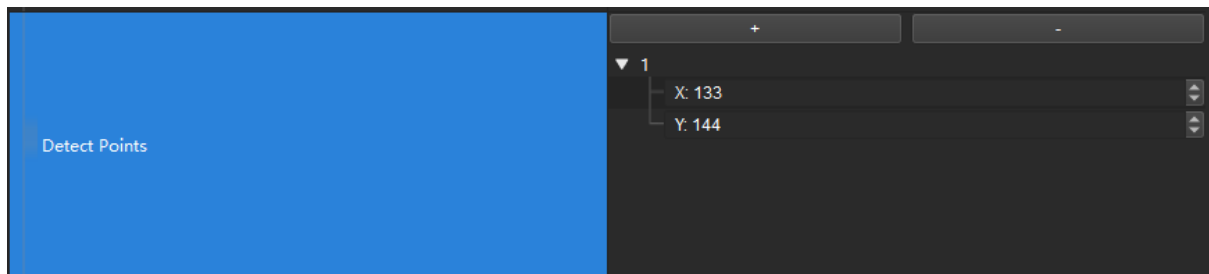
下図に示すように、**[+]**をクリックして、一つまたは複数の検出点を追加できます。



下図に示すように、追加された対象点は画面の左上にあります。検出点をユーザーが指定した位置に移動することができます。



下図に示すように、**ステップパラメータ** 画面で、検出点の位置を調整できます。



結果の確認

下図に示すように、**測定結果の出力** 画面で、点から基準線までの高さの差が表示されます。

測定結果の出力					📄 ✕
名前	値	下限規格値	上限規格値	オフセット	
点から基準線までの高...	1.031	0.000	0.000	0.000	



複数の検出点を追加すれば、複数の測定結果が取得されます。

ユーザーは **測定結果の出力** または **結果ビュー** 画面で上限・下限を設定することで測定値が合格したかどうかを確認できます。

4.3.249. 点から平面までの高さの差を測定

このステップは、**点** から **平面** までの高さの差を測定します。

このステップを実行する前に、[\[vision-measure-mode:getting-started:::start-use-measure-mode\]](#) をお読みください。

例



[[vision-steps:read-point-cloud-v2:::read-point-cloud-v2](#)]、[正投影](#)、[点から平面までの高さの差を測定](#)を併用して、**点** から **平面** までの高さの差の測定を実行します。

ステップをつなぐ

1. [[vision-steps:orthographic-projection:::orthographic-projection](#)] をクリックして、**ステップの入力・出力ポートの選択** 画面で「入力1（法線ベクトル付きの点群）」のドロップダウンバーで「点群を読み取る_1_（法線ベクトル付きの点群）」に設定します。
2. [点から平面までの高さの差を測定](#) をクリックして、**ステップの入力・出力ポートの選択** 画面で「入力1（可視化背景）」のドロップダウンバーで「正投影_1_深度画像」に設定します。

パラメータ設定

- [[vision-steps:read-point-cloud-v2:::read-point-cloud-v2](#)] 設定：

このステップをクリックし、**ステップパラメータ** 画面で測定する対象物の点群ファイルパスなどを設定します（ユーザーは測定する対象物の点群を用意する必要があります）。設定を完了

したら、**[実行]**をクリックしてプロジェクトを実行します。

- **点から平面までの高さの差を測定** 設定：
 - 高さを計算：初期値を使用します。
 - 基準点：平面を生成するには三つの基準点が必要なので、三つの基準点を追加します。
 - 対象点：一つ追加します。
- **測定結果の出力** または **結果ビュー** 画面で、**点** から **平面** までの高さの差を確認できます。また、上限・下限を設定することで測定値が合格したかどうかを確認できます。

4.3.250. 最長の線分を測定

このステップは、ROI内の最長線分を見つけて、その長さを測定するために使用されます。

このステップを実行する前に、[\[vision-measure-mode:getting-started:::start-use-measure-mode\]](#) をお読みください。

例



[\[vision-steps:read-images-v2:::read-images-v2\]](#)、**最長の線分を測定** を併用してROI内の最長線分を見つけて、その長さを測定します（この使用例では、長さの単位はピクセル）。

ステップをつなぐ

最長の線分を測定 をクリックして、**ステップの入力・出力ポートの選択** 画面で、「入力1(カラー画像)」のドロップダウンバーで「画像を読み取る_カラー画像」を選択します。

パラメータの調整と結果の表示

- [\[vision-steps:read-images-v2:::read-images-v2\]](#) 設定：

このステップをクリックし、**ステップパラメータ** 画面で測定する対象物の画像のパスなどを設定します（ユーザーは測定する対象物のカラー画像を用意する必要があります）。設定を完了したら、**[実行]** をクリックしてプロジェクトを実行します。

- **最長の線分を測定** 設定：下図に示すように、このステップをクリックしてROIを選択します。



- ユーザーは **測定結果の出力** または **結果ビュー** 画面で、線分の長さ（単位はピクセル）を確認できます。また、上限・下限を設定することで測定の値が合格したかどうかを確認できます。

4.3.251. 軸に平行な方向に沿った平面の高さの差を測定



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

深度画像で指定された方向に最大段差を計算します。

使用シーン

特定のシーン向けのカスタマイズのステップです。

入力と出力

- **入力：**
 1. 各座標点の高さを定義する深度画像。
- **出力：**

1. 平面度（最大段差）。
2. 最大段差に垂直で、最大段差から画像のエッジまで、指定軸に沿った線分の二つの部分。

4.3.252. 測定結果を計算して表示

機能

入力された深度画像と位置姿勢で、測定結果を計算して出力します。

使用シーン

開発者のテスト用のステップです。普通のユーザーは、一般的な使用シーンでこのステップを使用する必要はありません。

入力と出力

- 入力：
 1. 深度画像。
 2. マスク画像。
 3. 位置姿勢リスト。
 4. 位置姿勢リスト。
- 出力：
 1. カラー画像。

4.3.253. 点群リストをマージ



これは古いバージョンのステップであり、メンテナンスが停止されています。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

2つの点群リストを1つにマージします。

使用シーン

汎用の点群リストの編集ステップです。

これは古いバージョンのステップであり、メンテナンスが停止されますので、新しいバージョンのステップの組合せを使用してください。

入力と出力

入力： マージするための点群リスト。

- 出力： マージされた点群リスト。

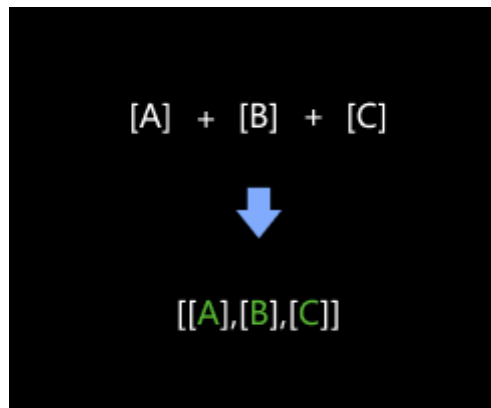
4.3.254. データをマージ



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

汎用のデータのマージステップです。

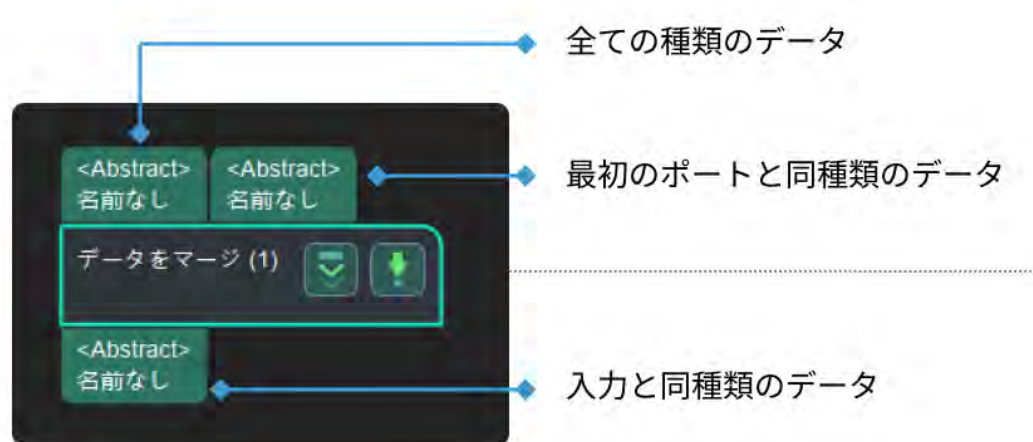


使用シーン

位置姿勢、データなどの数値をマージするために使用されます。

このステップは新しいバージョンで、古いバージョンの [\[vision-steps:merge-pose-lists::merge-pose-lists\]](#)、[点群リストをマージ](#) をこのステップに置き換えることを推奨します。

入力と出力



4.3.255. 深度画像をマージ



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

2枚のカメラの深度画像を1枚にマージします。

使用シーン

カメラのパッシブステレオモードをデバッグするための開発者機能です。普通のユーザーは、一般的な使用シーンでこのステップを使用する必要はありません。

入力と出力

● 入力：

1. マージするための最初のカメラ深度画像。
2. マージするための2番目のカメラ深度画像。
3. 左側面図のROIマスキリスト。
4. 右側面図のROIマスキリスト。
5. カメラの視差値から深度値へのマッピング行列（4 x 4行列である必要があります）。

● 出力：

1. マージによって形成された深度画像。

4.3.256. ラベルリストをマージ



これは古いバージョンのステップであり、メンテナンスが停止されていますので、新しいバージョンのステップ [\[vision-steps:merge-data:::merge-data\]](#) を使用してください。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

入力された2つセットのラベルリストを1つにマージします。

使用シーン

汎用のラベルのマージステップです。

入力と出力

● 入力：

1. このポートに入力されるラベルリストが出力にマージされます。

2. このポートに入力されるラベルリストが出力にマージされます。

● 出力：

1. ラベルのマージによって得られたラベルリスト。

4.3.257. 線分リストをマージ



これは古いバージョンのステップであり、メンテナンスが停止されていますので、新しいバージョンのステップ [\[vision-steps:merge-data:::merge-data\]](#) を使用してください。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

2つセットの線分リストを1つセットにマージします。

使用シーン

汎用の線分リストのマージステップです。特定の使用シーンはありません。

入力と出力

● 入力：

1. このポートに入力される線分リストはマージされます。
2. このポートに入力される線分リストはマージされます（最初ポートの入力リストの最後に）。

● 出力：

1. 入力された2つセットの線分をマージすることによって形成された線分リスト。

4.3.258. 近似高度の点群をマージ



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

指定方向に高さ値が近似している点群をマージします。

使用シーン

同じ対象物に属する点群リストをマージします。これは、ステップ [\[vision-steps:point-cloud-clustering:::point-cloud-clustering\]](#) の後で使用できます。

入力と出力

入力： このポートに入力される点群は、高さが近似しているかどうかに応じてマージされま

- す。
- 出力： マージされた点群。

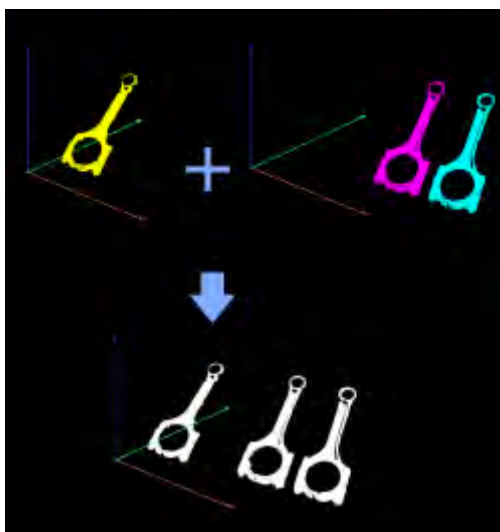
4.3.259. 点群をマージ



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

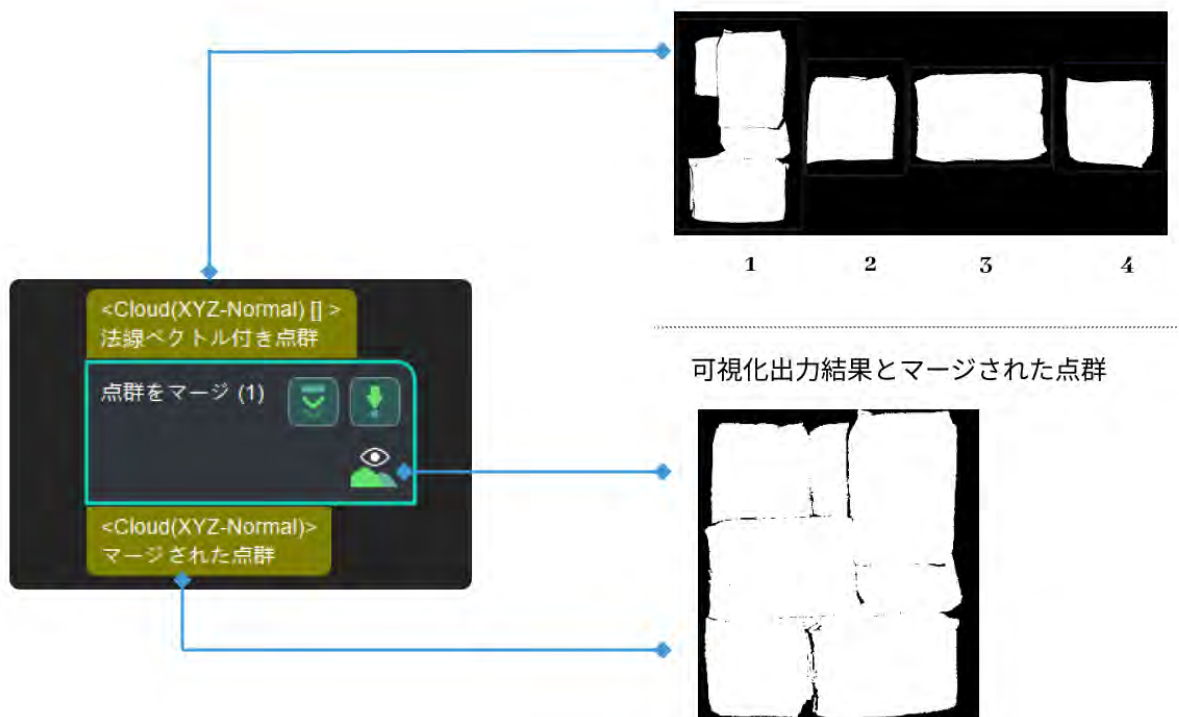
複数の点群を一つの点群にマージします。



使用シーン

通常、[\[vision-steps:point-cloud-clustering:::point-cloud-clustering\]](#)、[マスク内の対応する点群を抽出](#)などのステップの後で使用されます。点群リストにある複数の点群を全体として処理するために一つの点群をマージします。

入力と出力



4.3.260. 位置姿勢リストをマージ



このステップは古いバージョンであり、位置姿勢に関連するデータしかマージできず、他のタイプのデータを追加できません。新しいバージョンのステップ [\[vision-steps:merge-data::merge-data\]](#) を使用してください。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

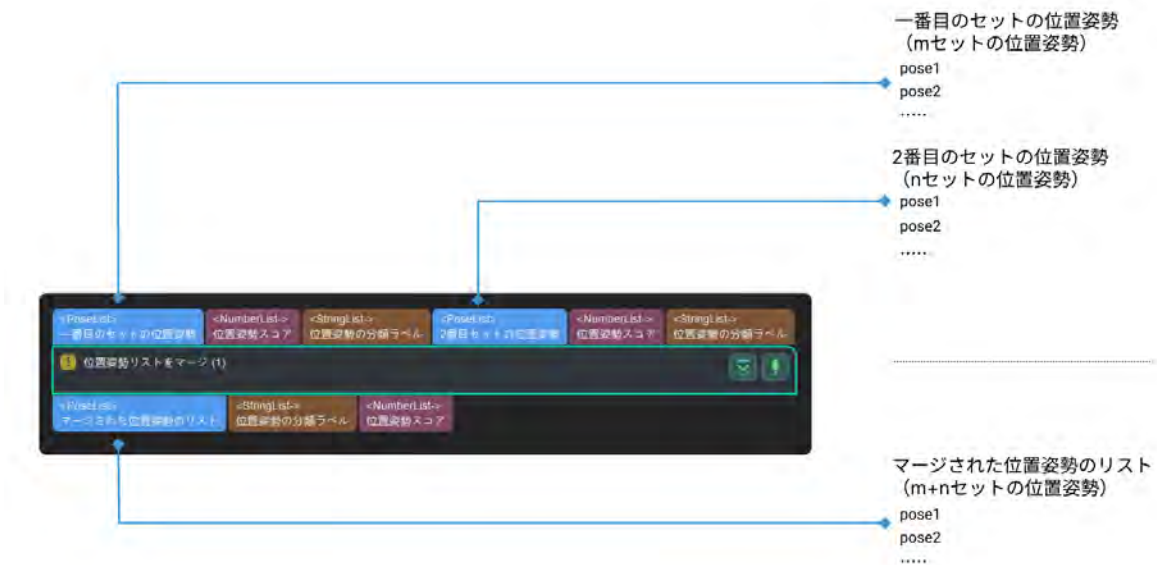
機能

二つの位置姿勢リストを繋ぎ、一つのリストにします。

使用シーン

通常、位置姿勢データの構造を編集する必要がある場合に使用されます。

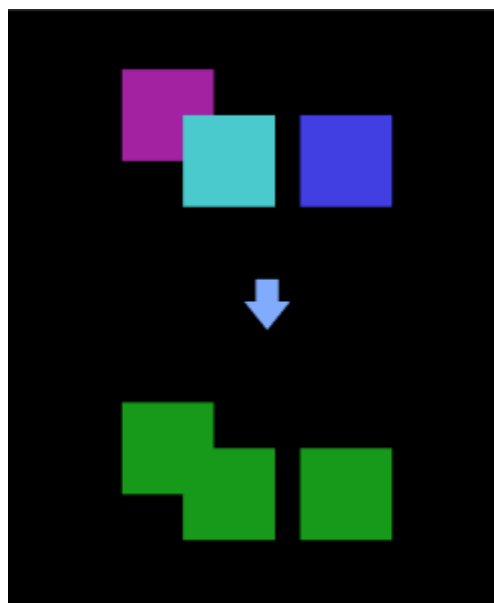
入力と出力



4.3.261. マスク画像をマージ

機能

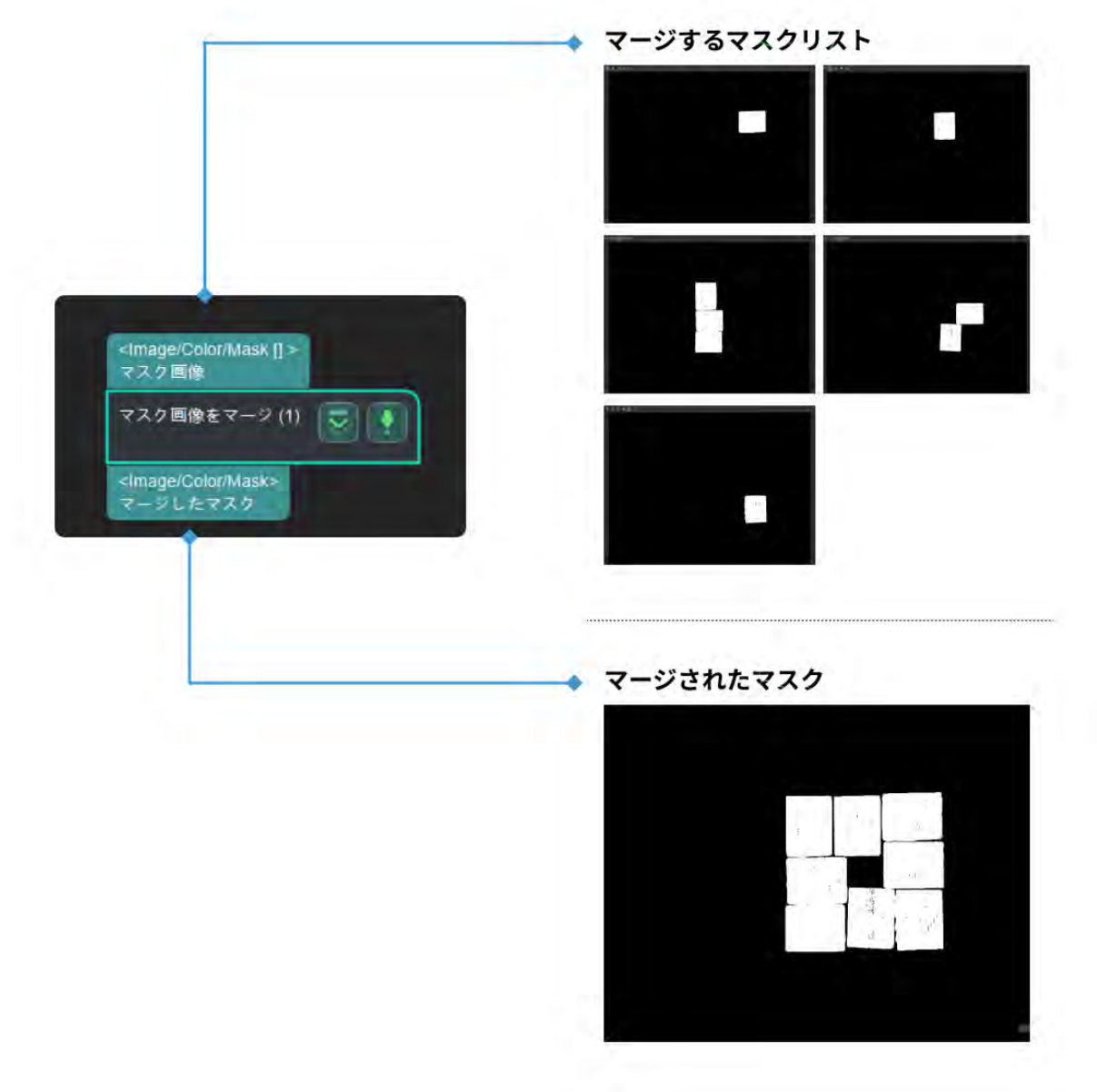
複数のマスクを1つのマスク画像にマージします。



使用シーン

汎用の深度画像の処理ステップです。リスト内の複数のマスクを1つのマスクにマージし、その後のマスクを全体として処理することを容易にします。

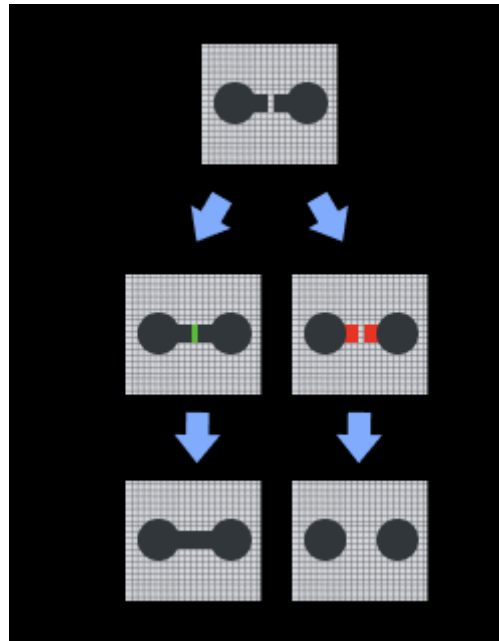
入力と出力



4.3.262. 画像のモフォロジー処理

機能

このステップは、モフォロジーを使用して画像を処理します。

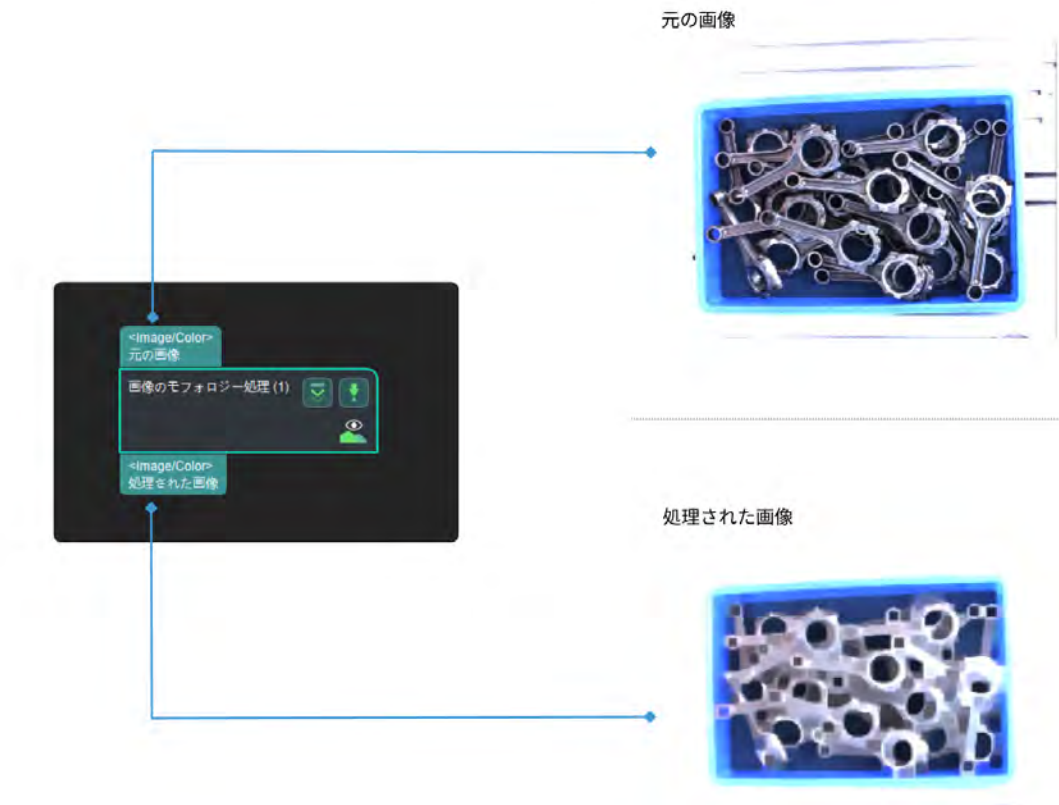


使用シーン

このステップは、一般的に画像前処理に使われます。画像品質が良くない、もしくはより詳細な調整を行う場合、ニーズに応じて膨張処理、収縮処理、オープニング・クロージング処理などにより、ノイズ除去や穴埋めなどを実現します。

入力と出力

クロージング処理を例として、画像のモフォロジー処理の入力と出力の効果は下図に示します。



パラメータの説明

モフォロジー処理のタイプ

パラメータ説明：このパラメータは、モフォロジー処理のタイプを選択するために使用されます。

初期値：DilateOperator

値のリスト：ClosingOperator(クロージング処理)、DilateOperator(膨張処理)、ErodeOperator(収縮処理)、OpeningOperator(オープニング処理)、Skeleton(スケルトン抽出)

調整説明：

1. ClosingOperator：膨張処理後に収縮処理することの別称です。前景の物体内部の穴又は黒い点を埋めることに役に立ちます。

カーネルのサイズ

初期値：21pixel

調整説明：検出要件に従って設定してください。操作カーネルが大きいほど、膨張後の収縮効果が強くなります。調整効果を [調整の例](#) に示します。

カーネルの形状

初期値：長方形 (MORPH_RECT)

オプション：長方形 (MORPH_RECT)、十字型 (MORPH_CROSS)、楕円 (MORPH_ELLIPSE)

調整説明：初期値を使用することをお勧めします。特別な場合は実際のニーズに応じて設定してください。

画像のディテール復元程度：MORPH_RECT < MORPH_CROSS < MORPH_ELLIPSE

◦ 操作に必要な時間：MORPH_RECT < MORPH_CROSS < MORPH_ELLIPSE

2. DilateOperator：膨張操作。対象物の面積を増やして対象物の損傷部分を繋ぎます。

カーネルのサイズ

初期値：3pixel

調整説明：操作カーネルが大きいほど、膨張効果が強くなります。検出要件に従って設定してください。調整効果を [調整の例](#) に示します。

カーネルの形状

初期値：長方形 (MORPH_RECT)

オプション：長方形 (MORPH_RECT)、十字型 (MORPH_CROSS)、楕円 (MORPH_ELLIPSE)

調整説明：初期値を使用することをお勧めします。特別な場合は実際のニーズに応じて設定してください。

画像のディテール復元程度：MORPH_RECT < MORPH_CROSS < MORPH_ELLIPSE

◦ 操作に必要な時間：MORPH_RECT < MORPH_CROSS < MORPH_ELLIPSE

3. ErodeOperator：収縮処理。小さいノイズを除去して二つのつながっている物体を離します。

カーネルのサイズ

初期値：3pixel

調整説明：操作カーネルが大きいほど、収縮効果が強くなります。検出要件に従って設定してください。調整効果を [調整の例](#) に示します。

カーネルの形状

初期値：長方形 (MORPH_RECT)

オプション：長方形 (MORPH_RECT)、十字型 (MORPH_CROSS)、楕円 (MORPH_ELLIPSE)

調整説明：初期値を使用することをお勧めします。特別な場合は実際のニーズに応じて設定してください。

画像のディテール復元程度：MORPH_RECT < MORPH_CROSS < MORPH_ELLIPSE

◦ 操作に必要な時間：MORPH_RECT < MORPH_CROSS < MORPH_ELLIPSE

4. OpeningOperator：収縮処理後に膨張処理することの別称です。ノイズ除去に役に立ちます。

カーネルのサイズ

初期値：21pixel

調整説明：検出要件に従って設定してください。調整効果を [調整の例](#) に示します。

カーネルの形状

初期値：長方形（MORPH_RECT）

オプション：長方形（MORPH_RECT）、十字型（MORPH_CROSS）、楕円（MORPH_ELLIPSE）

調整説明：初期値を使用することをお勧めします。特別な場合は実際のニーズに応じて設定してください。

画像のディテール復元程度：MORPH_RECT < MORPH_CROSS < MORPH_ELLIPSE

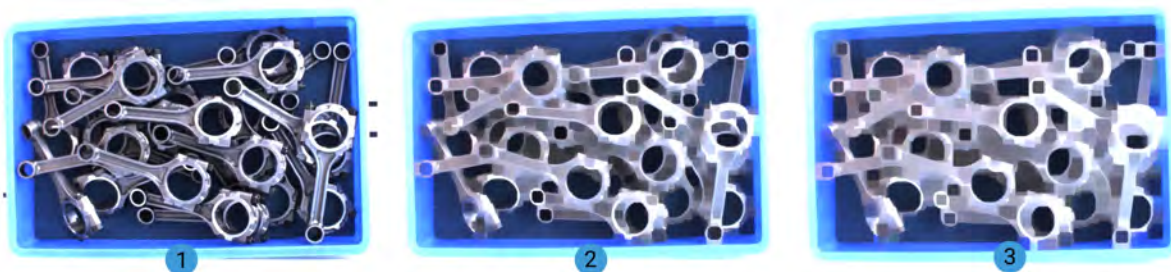
◦ 操作に必要な時間：MORPH_RECT < MORPH_CROSS < MORPH_ELLIPSE

5. Skeleton：対象輪郭を抽出します。調整効果を [調整の例](#) に示します。

調整の例

ClosingOperator処理

ClosingOperator を選択した場合、操作カーネルサイズが少しずつ増やしたときの画像効果を下図に示します（操作カーネルサイズが1から3に少しずつ増やします）。



DilateOperator処理

DilateOperator を選択した場合、操作カーネルサイズが少しずつ増やしたときの画像効果を下図に示します（操作カーネルサイズが1から3に少しずつ増やします）。



ErodeOperator処理

ErodeOperator を選択した場合、操作カーネルサイズが少しずつ増やしたときの画像効果を下図に示します（操作カーネルサイズが1から3に少しずつ増やします）。



OpeningOperator処理

OpeningOperator を選択した場合、操作カーネルサイズが少しずつ増やしたときの画像効果を下図に示します（操作カーネルサイズが1から3に少しずつ増やします）。



Skeleton処理

この処理実行前後の画像効果は下図に示します（1は実行前、2は実行後）。



4.3.263. 点群を指定方向に沿って移動



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社 (docs@mech-mind.net) までお問い合わせください。

機能

点群を、指定方向に沿って並進します。並進距離は、入力またはパラメータで設定できます。

使用シーン

汎用の点群並進ステップです。特定の使用シーンはありません。

これは古いバージョンのステップであり、メンテナンスが停止されますので、新しいバージョンのステップの組合せを使用してください。

入力と出力

● 入力：

1. このポートに入力される点群は並進されます。
2. 並進の基準となる対象物座標系（選択可能。パラメータで移動の基準座標系を対象物座標系に設定する場合は使用されます）。
3. 並進距離（選択可能。このポートが使用されていない場合、パラメータで設定された距離が使用されます）。

● 出力：

1. 並進された点群。

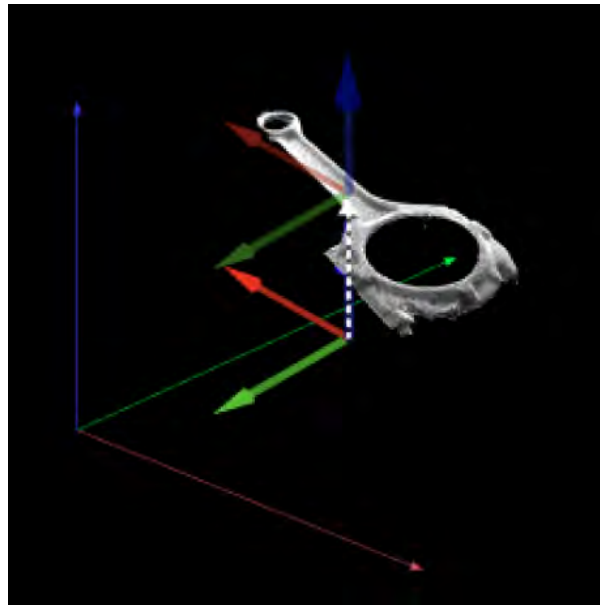
4.3.264. 位置姿勢をZ軸に沿って点群表面に移動



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社 (docs@mech-mind.net) までお問い合わせください。

機能

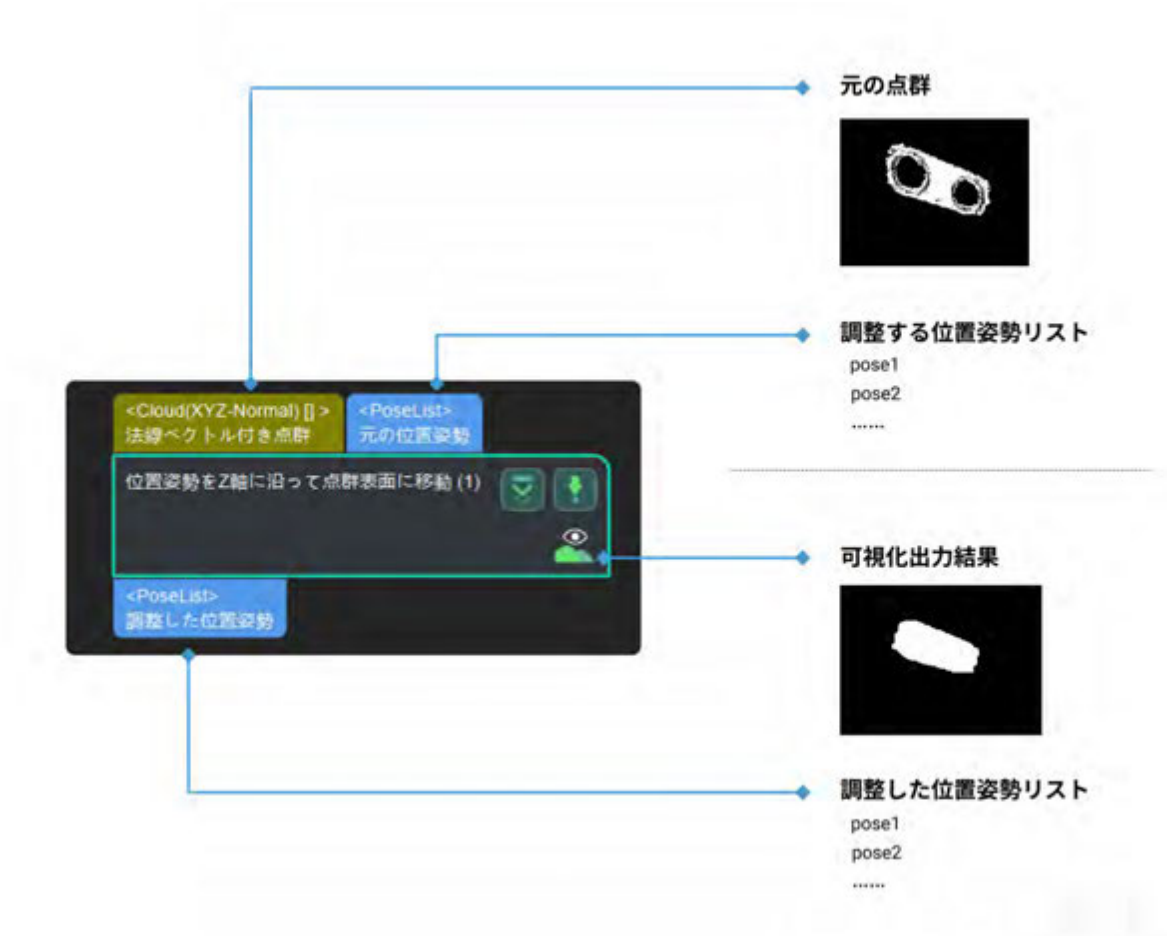
設定されたルールに従って、指定した座標系のZ軸に沿って位置姿勢を点群表面に移動します。



使用シーン

このステップは、対象物の表面にない位置姿勢を調整するために使用されます。このような位置姿勢は、対象物の傾きと凹面/凸面によって生成されます。通常、[\[vision-steps:calc-poses-and-dimensions-from-planar-point-clouds:::calc-poses-and-dimensions-from-planar-point-clouds\]](#) のステップの後で使用されます。

入力と出力



4.3.265. 法線ベクトルの計算



これは古いバージョンのステップで、新しいバージョンのステップ [\[vision-steps:calc-normals-of-point-cloud-and-filter-it::calc-normals-of-point-cloud-and-filter-it\]](#) での **NormalEstimation** という法線ベクトル計算方法を使用してください。このステップに関する詳細が必要な場合は、当社 (docs@mech-mind.net) までお問い合わせください。

機能

点群の法線ベクトルを計算します。

使用シーン

これは古いバージョンのステップで、新しいバージョンのステップを使用してください。

入力と出力



4.3.266. 通知



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

カスタマイズの通知を外部サービスに送信します。

使用シーン

通常は外部サービスとの通信に使用されます。制御フローで他のステップとつなぐと、ステップのステータスを監視でき、設定されたメッセージを外部サービスに送信することができます。

入力と出力

- 入力：なし。
- 出力：なし。

4.3.267. 数値スケーリング



これは古いバージョンのステップであり、メンテナンスが停止されていますので、新しいバージョンのステップ [\[vision-steps:numeric-operation:::numeric-operation\]](#) を使用してください。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

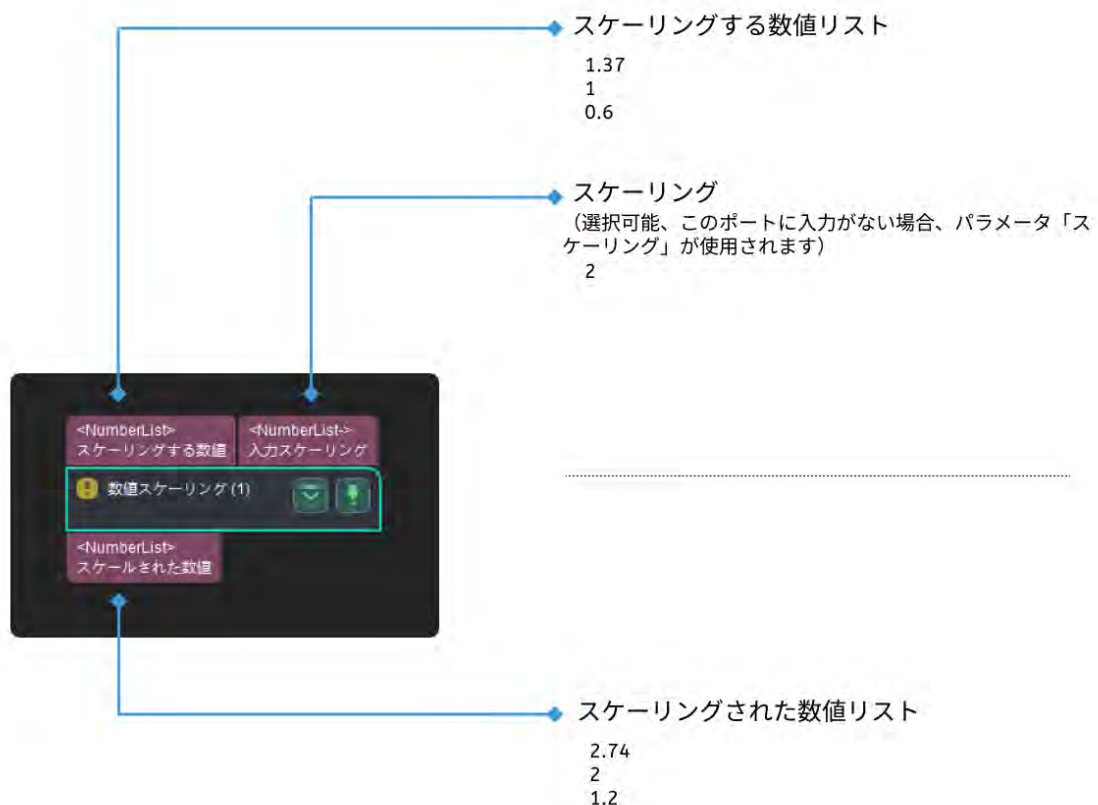
機能

最初のポートに入力された数値リストをスケーリングします。スケーリング係数は、2番目ポートから入力するか、パラメータで設定することができます。

使用シーン

これは古いバージョンのステップで、より多くの機能を備えた新しいバージョンのステップを使用してください。

入力と出力



4.3.268. 数値演算



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

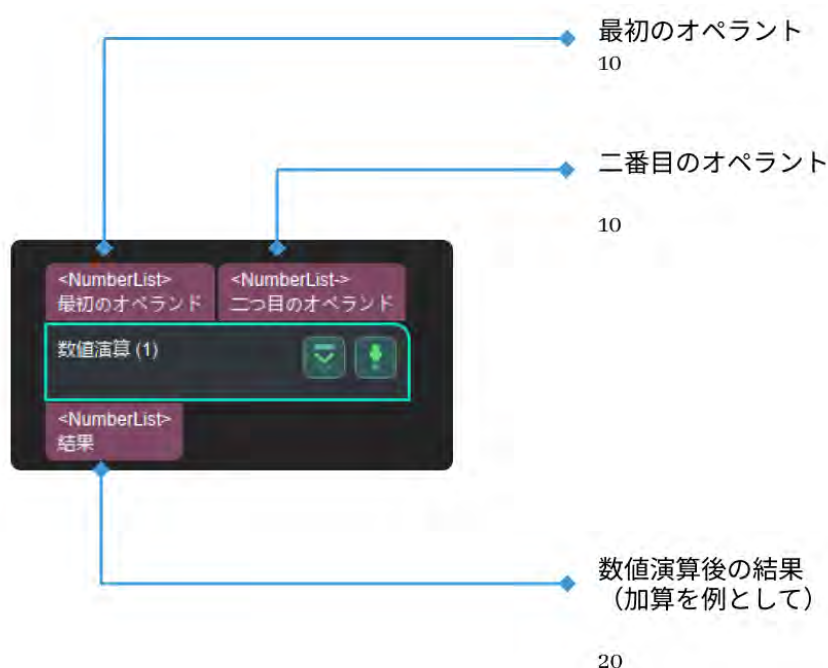
入力値に対して数値演算を実行します。加算、減算、乗算、除算などが実行可能です。

使用シーン

汎用の数値演算ステップです。一番目のオペランドを加算、減算、乗算、または二番目のオペランドで除算されます。

複雑な数値演算を必要とする場合は、[\[vision-steps:evaluate-results-by-javascript-engine::evaluate-results-by-javascript-engine\]](#) ステップを使用してください。

入力と出力



4.3.269. 光学文字認識



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社 (docs@mech-mind.net) までお問い合わせください。

機能

このステップにより、入力画像の中国語と英語の文字を認識できます。

使用シーン

汎用の画像処理ステップです。特定の使用シーンはありません。

入力と出力

● 入力：

1. このポートに入力される画像には、中国語と英語の文字が認識されます。

● 出力：

1. 認識された文字。
2. 認識の信頼度。
3. 画像内で認識された文字の位置。

4.3.270. 円柱の位置姿勢オフセット



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社 (docs@mech-mind.net) までお問い合わせください。

機能

位置姿勢を基準座標系のZ軸に沿って円柱表面に移動します。

使用シーン

通常、把持位置姿勢を点群表面に補正するために使用されます。

これは古いバージョンのステップであり、メンテナンスが停止されていますので、新しいバージョンのステップ「位置姿勢をZ軸に沿って点群表面に移動」を使用してください。

入力と出力

● 入力：

1. このポートに入力される円柱点群は位置姿勢の補正基準として使用されます。
2. このポートに入力される位置姿勢は補正されます。

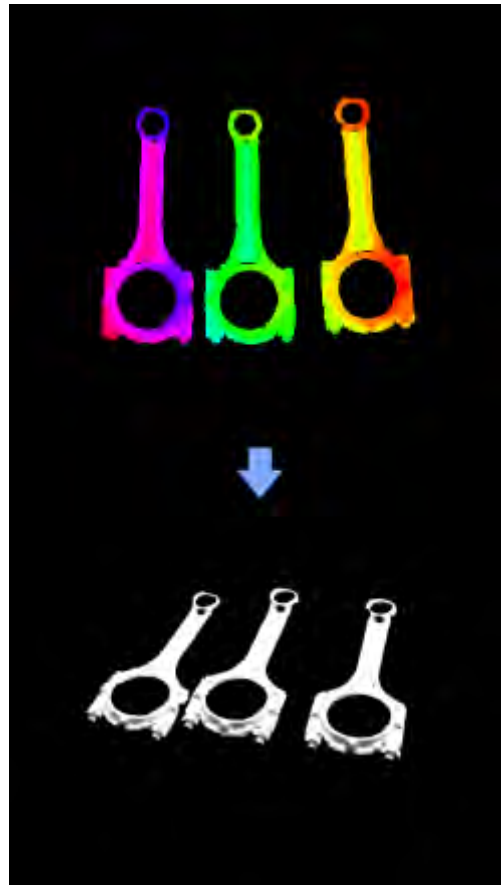
● 出力：

1. 円柱点群の表面に補正された位置姿勢。

4.3.271. 正投影の逆変換

機能

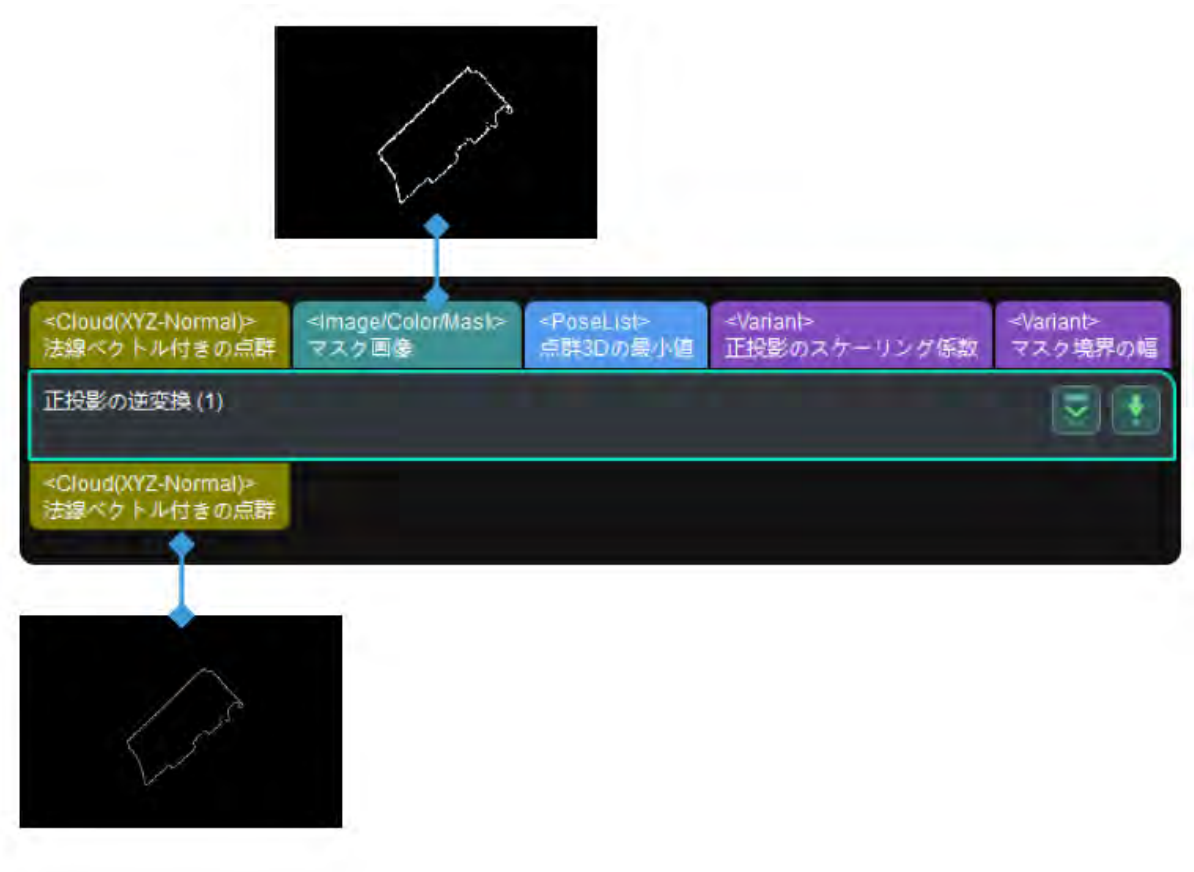
マスク画像の2D点を、3D点群に変換します。



使用シーン

通常、計測シーンによく使われます。通常はステップ [\[vision-steps:orthographic-projection:::orthographic-projection\]](#) の後で使用され、正投影後の2D画像を3D点群に変換します。

入力と出力



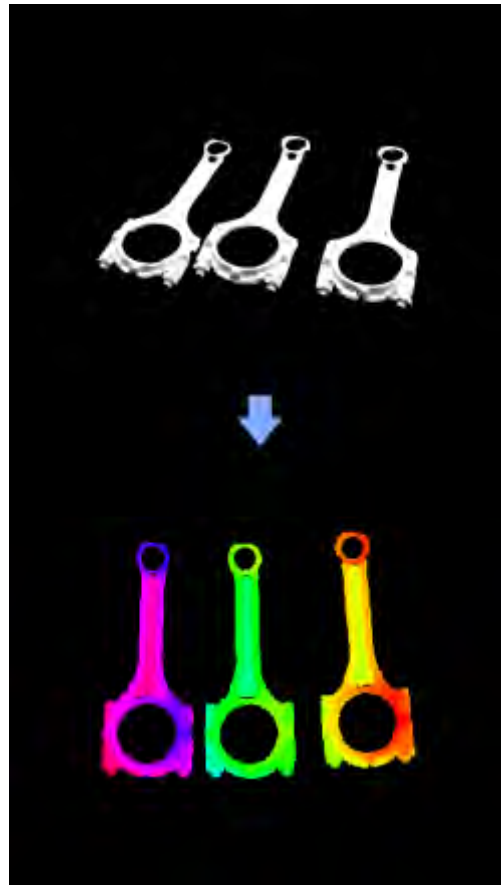
パラメータの説明

調整可能なパラメータはありません。

4.3.272. 正投影

機能

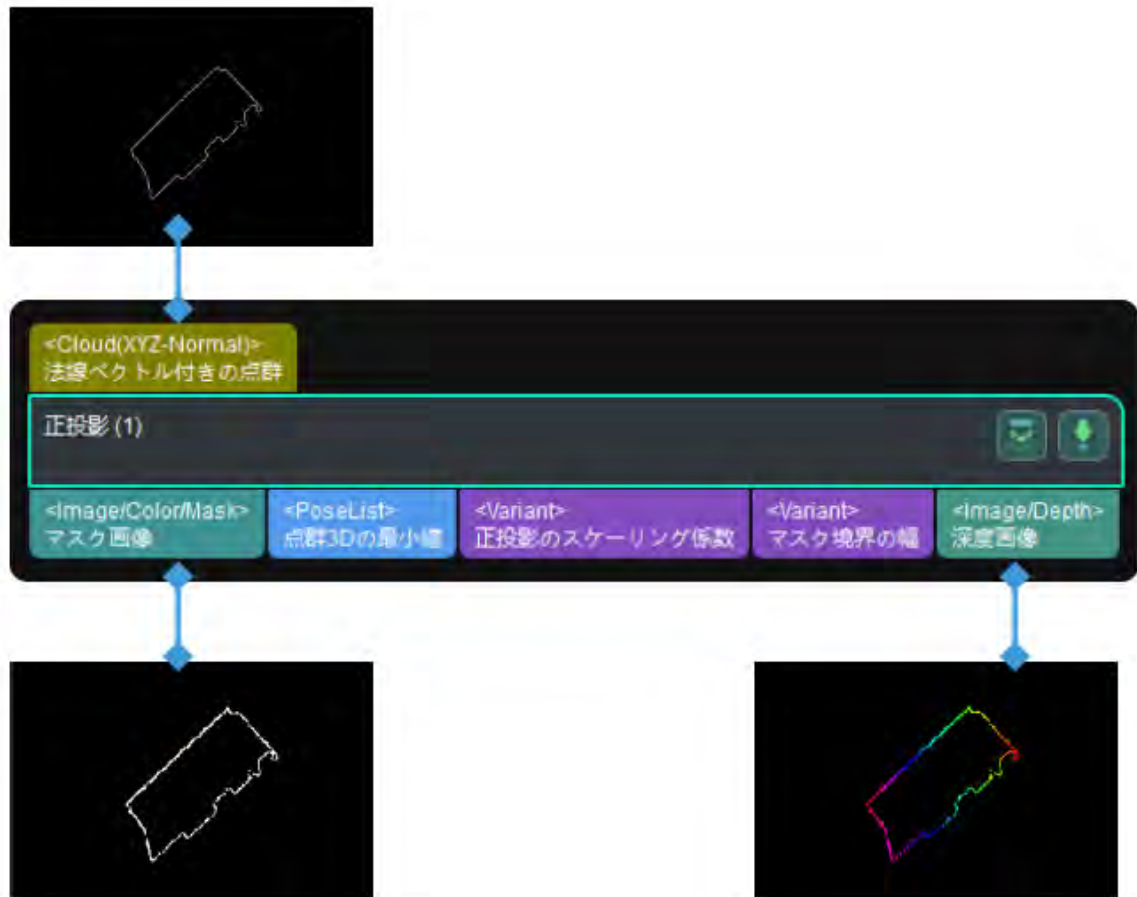
正投影で3D点群を2D深度画像に投影します。これにより、通常の投影の遠近法による歪みを回避できます。



使用シーン

通常、計測シーンに使用されます。3D点群を2D深度画像に投影してから、このステップと後の測定に関連するステップをつなぎます。

入力と出力



パラメータの説明



このステップのすべてのパラメータは、**すべてのパラメータを表示** に選択した場合にのみ表示されます。

正投影の設定

解像度

パラメータ説明：このパラメータは、正投影により形成される2D画像の単位長さ（1m）あたりのピクセル数を示します。

初期値：1000.0000

調整説明：調整効果を [調整の例](#) に示します。

マスクの空白の境界線の幅

パラメータ説明：正投影により形成された2D画像のマスクの周囲に空白の境界線が追加されます。このパラメータは、空白の境界線の幅を設定するために使用されます。

初期値：100px

調整説明：調整効果を [調整の例](#) に示します。

指定した最小点を使用

パラメータ説明：このパラメータは、点群の境界ボックスに最小のX、Y値を持つコーナーを示します。最小点と最大点を指定すると、最大点と最小点の範囲に正投影を行います。

初期値：チェックを入れない。

調整説明：このパラメータにチェックを入れるには、**最小点のX値** と **最小点のY値** を設定する必要があります。

最小点のX値

初期値：0.000 mm

調整アドバイス：パラメータ **指定した最小点を使用** にチェックを入れずにプロジェクトを実行すると、基準値がログバーに出力されます。基準値に従って正投影後の期待結果（干渉する点群がなく、対象物が位置する領域のみを保持）を得るまでこのパラメータを調整してください。

15:20:57.592 [orthographicProjection] 最小点のXとYは(-0.506839, -0.345616)です。
最大点のXとYは(0.450595, 0.316508)です。

最小点のY値

初期値：0.000 mm

調整アドバイス：詳細については、**最小点のX値** をご参照ください。

指定した最大点を使用

パラメータ説明：このパラメータは、点群の境界ボックスに最大のX、Y値を持つコーナーを示します。最小点と最大点を指定すると、最大点と最小点の範囲に正投影を行います。

初期値：チェックを入れない。

調整説明：このパラメータにチェックを入れるには、**最大点のX値** と **最大点のY値** を設定する必要があります。

最大点のX値

初期値：0.000 mm

調整アドバイス：詳細については、**最小点のX値** をご参照ください。

最大点のY値

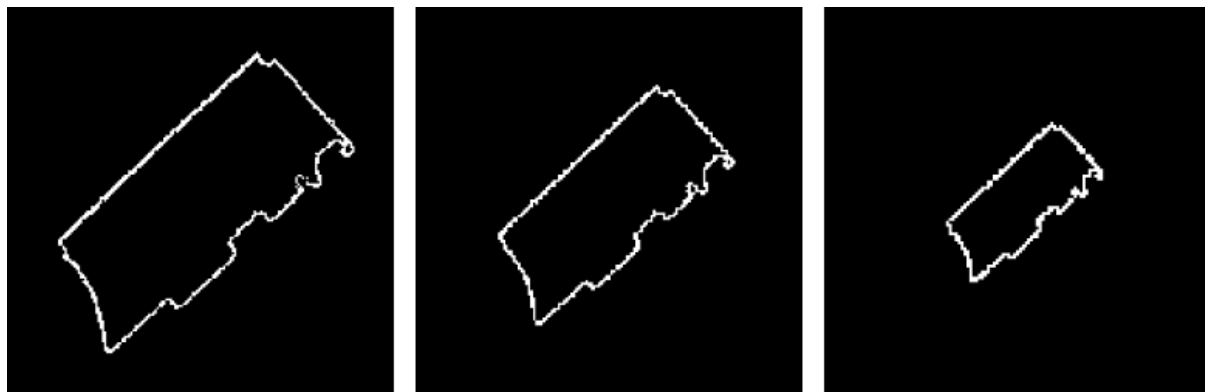
初期値：0.000 mm

調整アドバイス：詳細については、**最小点のX値** をご参照ください。

調整の例

解像度

正投影により出力されたマスク画像は、解像度がそれぞれ1000、700、400の場合、下図のようになります。



マスクの空白の境界線の幅

正投影により出力されたマスク画像は、マスクの空白の境界線の幅がそれぞれ100、400、700の場合、下図のようになります。



4.3.273. パック



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

複数の変数をパックし、変数配列にマージします。

使用シーン

汎用のデータをマージするステップです。

入力と出力

● 入力：

1. このポートに入力されるデータは、出力リストの要素になります。
2. このポートに入力されるデータは、出力リストの要素になります。

● 出力：

1. 各ポートに入力されたデータを要素として形成されたリスト。

4.3.274. パレット情報認識



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社 (docs@mech-mind.net) までお問い合わせください。

機能

パレットパターン情報とパレットパターンの点群に基づいて、最高層のパレタイジング状態を認識します。

使用シーン

通常、対象物が密集しているパレットパターン、および対象物の位置姿勢を認識するために使用されます。

入力と出力

● 入力：

1. パレットの点群。
2. 最高層の点群。
3. パレットの位置姿勢。
4. 単一のピクセルが対応する実際のサイズ。

● 出力：

1. 最高層の各箱の位置姿勢。
2. パレットの方向を表示するためのブール値。

4.3.275. 経路計画

機能

入力したビジョンポイントを用いてロボット動作経路を計画し、正常に計画したロボット動作経路を出力します。



- このステップを使用する前に、[ロボット通信設定](#) を完了させてください。

- 標準インターフェース通信を使用する場合、コマンド101でMech-Visionプロジェクトを実行する必要があります。送信したコマンドのパラメータは、101, Mech-Visionプロジェクト番号, ビジョンポイントの予期数, ロボット位置姿勢のタイプ, ロボット位置姿勢となります。

使用シーン

通常、標準インターフェースやAdapterの通信方式で、ビジョン目標点付近のロボット動作経路のみを計画する必要があるプロジェクトで使用されます。経路計画は、シーンを作成し、ビジョンポイントを送信し、点群衝突を検出し、最終的に衝突のないロボット動作経路を出力することによって行われます。

このステップの前後のステップは以下の通りです。

- 前のステップ：位置姿勢調整と処理に関連するステップ。
- 後のステップ：[vision-steps:procedure-out::procedure-out]（ポートタイプを「事前定義済み（ロボット経路）」に設定する必要があります）。



入力と出力

入力

- ビジョンポイント：このポートに入力される位置姿勢は、ロボット動作経路を計画するために使用されます。
- 衝突点群：このポートに入力される点群は、ロボットハンドとの衝突検出を行うために使用されます。
- 位置姿勢のラベル：このポートは、各ビジョンポイントに対応する対象物のラベルです。
- 対象物の寸法：このポートは、各ビジョンポイントに対応する対象物の寸法です。

出力

- 計画された経路：計画されたロボット動作経路です。
- フィルタリング結果：ビジョンポイントのフィルタリング結果のリストです。Trueは対応するビジョンポイントが要件を満たしていることを意味し、Falseは満たしていないことを意味します。

パラメータの説明

経路計画の設定

ワークフローの設定

調整説明：[[vision-steps:path-planning-configurator:::path-planing-tool](#)] を開いてワークフローの関連設定を行い、設定されたワークフローから1つを選択します。

衝突点群の設定

入力点群のタイプ

調整説明：経路計画設定ツールに入力する点群タイプを選択します。

オプション：CloudXYZRGB（カラー点群）、CloudXYZ（点群）、CloudXYZNORMAL（法線ベクトル付きの点群）。

入力点群がカメラ座標系にある

調整説明：チェックを入れると、カメラ座標系での点群がロボット座標系に変換されます。

4.3.275.1. 経路計画設定ツール

本節では、ロボット経路計画設定ツールとその使用方法について説明します。

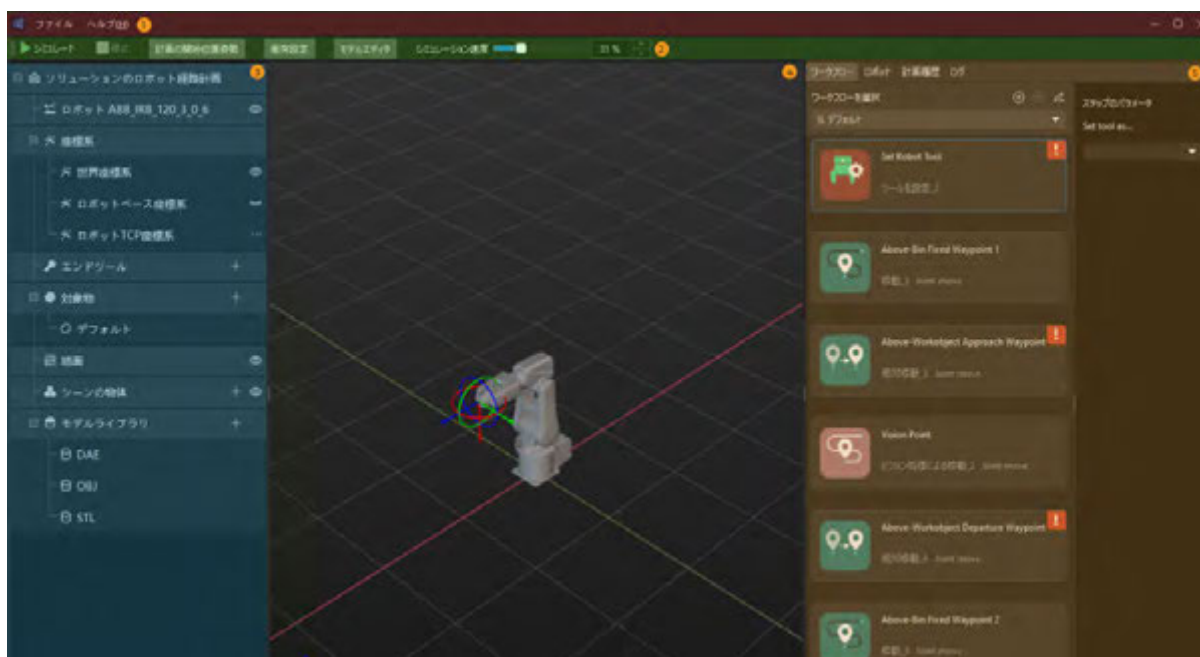
機能

経路計画設定ツールは、部品の周囲でロボットの衝突のない動作経路を計画するために使用されます。

ロボットの動作経路は、実際にロボットハンドがタスクを完了するまでの経路です。ロボットの動作に伴い、ロボットハンドが対象物やシーンの物体と衝突することもあります。したがって、経路計画や衝突検出に必要な各種リソースやデータを提供するためには、ロボットが位置するシーンの作成、ロボットハンドの設定、ビジョンポイントおよびシーンの点群の入力が必要です。経路計画設定ツールでは、これらのリソースやデータをインポートし、衝突検出や経路計画を完了させ、「経路計画」ステップでロボットの動作経路を出力し、把持などのタスクのためのロボットの制御を行うことができます。

画面

主に以下の5つの部分で構成されています。



メニューバー	設定のリセットと保存、設定ツールの終了のために使用されます。
ツールバー	シミュレーションの実行、開始位置姿勢の計画、衝突検出の設定、モデルエディタの起動のために使用されます。
リソース	ロボットハンド、対象物、シーンなど、プロジェクトで使用するリソースを管理します。
仮想空間	シミュレーションでは、ロボットの動作経路、ビジョン位置姿勢、点群などを表示します。
機能パネル	ワークフローとロボット位置姿勢を設定し、計画履歴とログを確認します。

使用手順

一般的な使用手順は以下の通りです。

プロジェクトリソースを設定

リソースとは、ロボットハンド、対象物、シーンの物体など、プロジェクトを構築するための要素を指します。

プロジェクトリソースを設定することで、以下のことが可能になります。

- 衝突検出に必要なロボットハンドやシーンの物体などのモデルを提供します。
- ロボットハンドの対称性、対象物の対称性と把持範囲を設定することで、経路計画をより柔

軟にし、より多くの経路点を選択できるようにします。


設定項目	説明
ロボットハンド	ロボットハンドとは、グリッパーや吸盤など、ロボットが動作を実現できるようにロボット先端に取り付ける装置のことです。 詳細については、 [vision-steps:end-tool:::path-planning-end-tool] をご参照ください。
対象物	対象物とは、段ボール箱や金属部品、接着剤塗布または溶接を実行する部品など、ロボットハンドが作用する物体です。 詳細については、 [vision-steps:work-object:::path-planning-work-object] をご参照ください。
地面	実際の地面をシミュレートします。デフォルトでは地面はロボットのベースと同じ平面にあります。実際のシーンの地面の高さに応じて地面の高さを調整する必要があります。 [地面] を右クリックして、表示されるウインドウでスライダーをドラッグするか、値を変更して地面の高さを調整します。
シーンの物体	シーンの物体とは、ロボット実機の作業環境にある様々な種類の対象物を指します。一般的にはロボットの安全柵、箱、パレット、カメラ、カメラスタンドなどが含まれます。 詳細については、 [vision-steps:scene:::path-planning-scene] をご参照ください。

ワークフローの構築

ワークフロー、すなわちフローチャート形式のロボット動作制御プログラムは、右側の「ワークフロー」タブで構築することができます。初期のワークフローを変更することで、対象物を把持する際のロボットの動作経路を計画することができます。

初期のワークフローは以下のステップが含まれています。

1. ロボットハンドを設定

ロボットハンドを選択するために使用されます。このステップをクリックし、右側の **ステップパラメータ** で  をクリックし、プロジェクトリソースに設定された対応するロボットハンドを選択します。

2. コンテナの上の固定点1

対象物に接近する前に経路点を設定するために使用されます。パラメータの詳細については、[移動](#)をご参照ください。

3. 対象物の上の接近点

対象物の上の接近点を設定するために使用されます。パラメータの詳細については、[相対移動](#)をご参照ください。

4. ビジョン目標点

ビジョン目標点を設定するために使用されます。パラメータの詳細については、[ビジョン目標点](#)をご参照ください。

5. 対象物の上の離れる点

対象物の上の離れる点を設定するために使用されます。パラメータの詳細については、[相対移動](#)をご参照ください。


6. コンテナの上の固定点2

対象物から離れた後の経路点を設定するために使用されます。パラメータの詳細については、[移動](#)をご参照ください。

実際の要件を満たすために、「移動」や「相対移動」のステップを追加することができます。

- ステップの追加：ステップ間のつながり線にマウスを移動し、をクリックして、追加するステップをクリックします。
- ステップの削除：ステップを右クリックし、表示されるメニューから[削除]をクリックします。もしくは、ステップを選択して **Delete** キーを押します。

また、同じプロジェクトリソースを用いて異なるワークフローを構成することで、異なる経路計画を実現することも可能です。同じプロジェクトに異なる対象物や箱が含まれるシーンでは、複数のワークフローを追加して、[パラメータレシピを切り替える](#)ことでワークフローを変更することが必要になる場合があります。

をクリックし、表示されるウィンドウでワークフローの名前を入力して[OK]をクリックすると、新しいワークフローが作成されます。

衝突検出の設定

デフォルトでは、ロボット、ロボットハンド、シーンの物体間の衝突を検出します。**点群衝突を検出**を有効にすると、入力点群とロボットハンド間の衝突をチェックします。

1. ツールバーの[衝突検出の設定]をクリックし、プロジェクト要件に応じて以下のパラメータを設定します。

点群の立方体の辺の長さ

衝突検出では、点群の各点を中心とした立方体を構築します。ロボットハンドと点群との衝突を検出することで、実際には個々の立方体との衝突を検出しています。点群立方体の辺の長さが大きいほど、ロボットハンドと衝突する可能性が高くなり、接している体積も大きくなります。

ロボットハンドと点群との衝突体積のしきい値

ロボットハンドと点群が接している体積がしきい値を超えた場合は衝突が発生したと判定し、それ以外の場合は衝突が発生しないと判定します。

点群衝突記録

- 記録：計画履歴には衝突が発生した点群の位置と数が記録されます。計画履歴を確認するとき、記録された点群は仮想空間にハイライト表示されます。記録は実行の速度に影響を与えます。
- 記録しない：計画履歴に衝突が発生した点群を確認できません。実行速度は速いです。

2. 設定後、[OK]をクリックします。

シミュレーションと最適化

以上の設定が完了したら、仮想ロボットを制御して走行させ、計画した経路に衝突などの問題がないことを確認します。



- 最初のシミュレーションでは、Mech-Visionのメイン画面で[実行]をクリックし、位置姿勢と点群情報を経路計画設定ツールに送信する必要があります。
- 経路計画設定ツールで[シミュレート]をクリックすると、前回のMech-Visionプロジェクト実行時に送信された位置姿勢と点群情報を使ってシミュレーションが実行されます。

1. ツールバーの[シミュレート]をクリックします。

2. シミュレーションで衝突のアラームや位置の未到達などの問題が発生した場合、計画履歴やログでエラーメッセージを確認し、問題を迅速に突き止めることができます。エラーメッセージに応じて、プロジェクトのリソース、ワークフローと衝突検出の設定を最適化することができます。

3. 問題がなければ、**ファイル**、**保存**をクリックして、設定を保存します。

経路計画設定ツールの使用が終わったら、**ファイル**、**終了**をクリックしてそれを完全に終了させます。次に、「経路計画」ステップのパラメータで、**ワークフロー** パラメータを設定します。

4.3.275.2. 画面

本節では、経路計画設定ツールの画面について説明します。主にメニューバー、ツールバー、プロジェクトリソース、仮想空間および機能パネルで構成されています。

メニューバー

オプション	説明	ショートカット
すべてをリセット	設定ツールの全設定をリセットします	Ctrl + N
保存	現在の設定をすべて保存します	Ctrl + S
終了	設定ツールを終了します	Ctrl + W
ユーザーズマニュアル	ユーザーマニュアルにアクセスします	なし

ツールバー

オプション	説明
シミュレート	仮想ロボットを実行します
停止	仮想ロボットの実行を停止します
計画の開始位置姿勢	シミュレートする際のロボットの初期位置姿勢を設定します
衝突検出の設定	点群衝突検出のパラメータを設定します
モデルエディタ	モデルエディタを開きます
シミュレーション速度	仮想ロボットの動作速度を設定します

リソース



プロジェクトリソースには、ロボット、ロボットハンド、部品（対象物）、シーンの物体などが含まれています。

ロボット

この項目は、現在使用されているロボットを表示します。



ロボット名を右クリックすると、以下のオプションが表示されます。


ロボットファイルディレクトリを開く	ロボットモデルファイルが格納しているフォルダを開きます
-------------------	-----------------------------

 をクリックするとロボットが半透明の状態になり、 をクリックすると元に戻ります。

座標系

この項目は、仮想空間におけるすべての座標系を表示します。

 をクリックすると対応する座標系が表示され、 をクリックすると非表示になります。

 をクリックすると、ドラッグボールの「並進」および「回転」機能の有効化や無効化を切り替えます。

ロボットハンド

この項目は、設定済みのロボットハンド名を表示します。

[+] をクリックすると、ロボットハンドを追加するためのロボットハンド設定画面が表示されます。

設定済みのロボットハンドの名前を右クリックすると、以下のオプションが表示されます。

オプション	説明	ショートカット
コピー	選択したロボットハンドをコピーします	Ctrl + C
貼り付け	コピーしたロボットハンドを貼り付けます	Ctrl + V
削除	選択したロボットハンドを削除します	Delete
名前を変更	選択したロボットハンドの名前を変更します	ロボットハンド設定
選択したロボットハンドの設定画面を開きます	現在のロボットハンドに設定	選択したハンドを現在のロボットハンドに設定します

 をクリックするとロボットが半透明の状態になり、 をクリックすると元に戻ります。

対象物

この項目は、設定済みの対象物名を表示します。

[+] をクリックすると、対象物を追加するための対象物設定画面が表示されます。

設定済みの対象物名を右クリックすると、以下のオプションが表示されます。

オプション	説明	ショートカット
削除	選択した対象物を削除します	Delete

オプション	説明	ショートカット
名前を変更	選択した対象物の名前を変更します	なし
対象物設定	選択した対象物の設定画面にアクセスします	なし





「Default」は、ラベルのないビジョン結果に使用される対象物設定となります。

地面

この項目は、地面の高さと表示状態を設定するために使用されます。

右クリックして地面の高さを調整することができます。

 をクリックすると地面が非表示になり、 をクリックすると元に戻ります。




シーンの物体

この項目は、設定済みのシーンの物体名を表示します。

[+] をクリックすると、シーンの物体を追加するための設定画面が表示されます。

設定済みのソーンの物体の名前を右クリックすると、以下のオプションが表示されます。

オプション	説明	ショートカット
コピー	選択したシーンの物体をコピーします	Ctrl + C
貼り付け	コピーしたシーンの物体を貼り付けます	Ctrl + V
切り取り	選択したシーンの物体を切り取ります	Ctrl + X
削除	選択したシーンの物体を削除します	Delete
名前を変更	選択したシーンの物体の名前を変更します	なし
シーンの物体設定	選択したシーンの物体の設定画面を開きます	なし

 をクリックするとロボットハンドが半透明の状態になり、 をクリックすると透明の状態になり、 をクリックすると通常の表示になります。

また、シーンの物体の名前をドラッグすることで、シーンの物体のモデルを子モデルや親モデルに変換することができます。

モデルライブラリ

モデルライブラリの項目には、ソフトウェアに追加されたすべての外部モデルが表示され、モデルの形式に従って分類されています。

[+] をクリックすると、外部モデルをインポートするためのファイル選択画面が表示されます。


インポートされた外部モデルを右クリックすると、以下のオプションが表示されます。

オプション	説明	ショートカット
コピー	選択したモデルをコピーします	Ctrl + C
貼り付け	コピーしたモデルを貼り付けます	Ctrl + V
削除	選択したモデルを削除します	Delete
モデル設定	選択したモデルのモデル変換の編集画面を開きます	なし
モデルエディタ	モデルエディタを開き、選択したモデルを基準モデルとして使用します	なし
フォルダを開く	モデルファイルが格納しているフォルダを開きます	なし

仮想空間

仮想空間には、ロボット、シーンの物体、ロボットハンド、ロボットの経路、把持点、衝突などが表示されます。また、シーンの物体のモデルはドラッグ＆ドロップで移動することが可能です。

仮想空間の表示は、以下のマウス操作で調整することができます。

視点を回転	マウスの左ボタンを長押しし、任意の方向にドラッグします	
-------	-----------------------------	---






視点を切り替える	マウスの右ボタンをクリックしてショートカットメニューで選択します	
視点を並進	マウスの中ボタンを長押しし、任意の方向にドラッグします	
スケーリング	ホイールを回します	

機能パネル

機能パネルには、ワークフロー、ロボット、計画履歴およびログが含まれています。

ワークフロー

ワークフローのパネルは、ワークフローを設定するために使用されます。

-  をクリックすると、ワークフローを新規作成することができます。
-  をクリックすると、ワークフローを削除することができます。
-  をクリックすると、ワークフローの名前を編集することができます。
-  をクリックすると、ワークフローを切り替えることができます。
- ワークフローでステップを選択すると、右側でステップパラメータが表示されます。
- マウスをステップ間のつながり線を移動し、 をクリックすると、ステップを追加することができます。
- ステップ名をダブルクリックすると、名前を変更することができます。

ロボット

ロボットパネルでは、関節角度、TCP、ロボットの初期位置を設定可能です。

計画履歴

詳細な計画履歴を確認することができ、問題の突き止めやプロジェクトの最適化に使用することができます。

計画履歴に関する設定は、この機能パネルの下部で行うことができます。

衝突を表示する期間	衝突による失敗結果を選択した場合、仮想空間には衝突を表示する期間を設定します。
保存期間	計画履歴を保存するための期間を設定します。この期間を超えると、現在の計画履歴が以前の計画履歴を上書きします。
計画履歴を読み込む	保存された計画履歴を読み込みます。
クリア	クリックすると現在の計画履歴がクリアされます。
ヘルプ	クリックすると計画履歴の紹介が表示されます。

さらに、任意場所を右クリックすると、以下のオプションが表示されます。

検索	[検索] をクリックすると、関連するオプションが表示され、計画履歴の結果を種類別にフィルタリングすることができます。
すべて折りたたむ	子ノードをすべて折りたたみ、親ノードのみを表示します。
選択された計画履歴を削除	選択された項目を削除します。

ログ

プロジェクト実行中の詳細情報がログに表示されます。また、ログパネルの下にパラメータを設定できます。

いずれかのコンテンツを選択すると、**Ctrl+C**を押してコピーすることができます。

D、i、W、E	ログのレベルによって表示するログを選択します。複数選択が可能です。 D：デバッグ情報 I：一般情報 W：ワーニング E：エラー
クリア	クリックすると現在のログメッセージがクリアされます。

エクスポート	現在表示されているログ情報をHTML形式でエクスポートします。ソフトウェアのインストールフォルダ中のlogsフォルダに格納されています。
ログフォルダを開く	ソフトウェアのインストールフォルダ中のlogsフォルダを開きます。中には時間付きのログファイルを確認可能です。

4.3.275.3. ロボットハンド

本節では、ロボットハンドとその関連設定について説明します。

概要

ロボットハンドとは、ロボットが作業を行うために特別に設計され、ロボットのフランジに取り付けられるグリッパーや吸盤などの装置です。

ロボットハンドモデル

仮想空間でロボットハンドを表示し、ロボットハンドの衝突検出を有効にするために、ロボットハンドモデルと衝突モデルを作成してソフトウェアのモデルライブラリにインポートする必要があります。

対応可能な形式は以下の通りです。

形式	STL	OBJ	DAE	Binvox
3Dモデル	○	○	○	×
衝突モデル	×	○	×	×


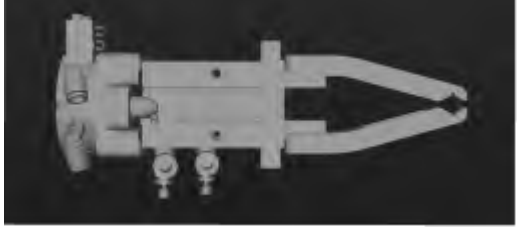
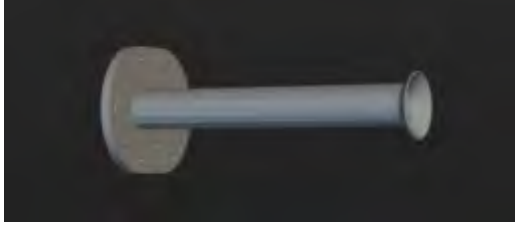


- ソフトウェアでは、ロボットハンドの衝突モデルが各部分の凸多面体（凸包）のobjモデルである必要があります。 [モデルエディタ](#) を使用してロボットハンドモデルの凸多面体を構築し、obj形式に変換することができます。
- 3Dモデルは仮想空間での表示にのみ使用でき、衝突検出が有効になりません。衝突モデルは衝突検出にのみ使用でき、仮想空間で表示されません。

ロボットハンドの対称性

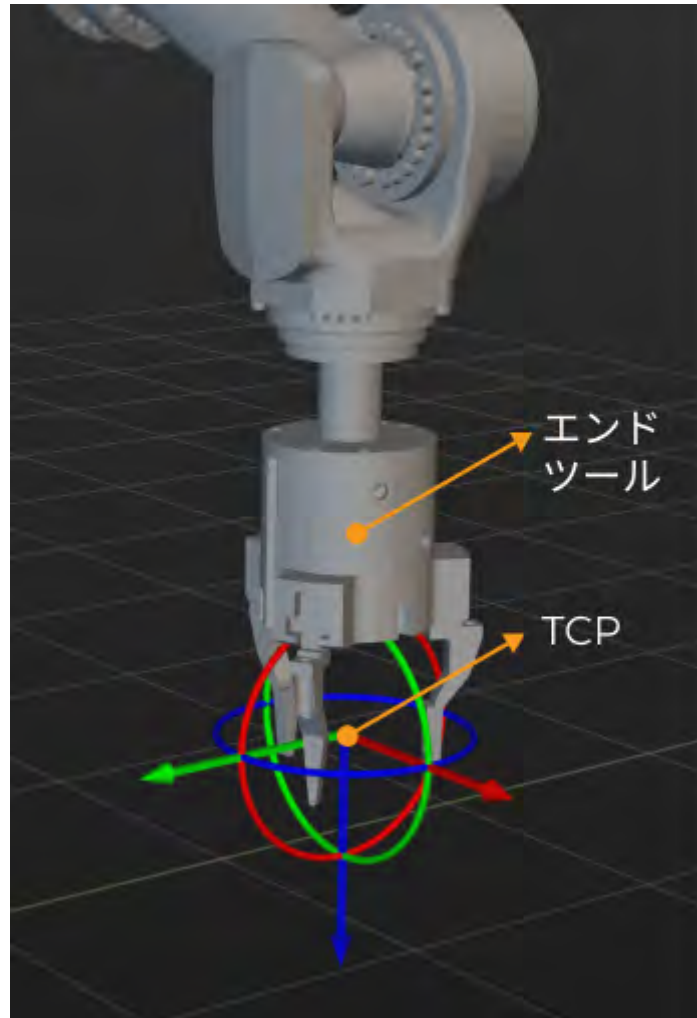
ロボットハンドを対称軸を中心に一定角度回転したら元の形状と完全に重なり合うことです。

ロボットハンドの対称性を設定することでロボットが把持や配置を実行する時の無駄な回転を減少し、動作実行の成功率を改善でき、またロボットがよりスムーズに動作します。

回転対称性無し	
2回対称	
円対称（任意の角度回転させても自らと重なる）	

ツール中心点（TCP）

ロボットハンドは対象物を把持・配置するために使用されます。把持・配置のためにロボットをある点に移動することは、実際にはTCPをその点に移動することです。TCPは普通、ロボット先端にあるので、実際の把持・配置を再現するためにTCPをロボットハンドの先端に調整する必要があります。



ロボットハンドモデルの設定

ロボットハンドの衝突モデルと3Dモデルをインポート

リソース ▶ モデルライブラリ のの [+] をクリックし、表示されたウィンドウで衝突モデルファイルと3Dモデルファイルを選択して [開く] をクリックします。

ロボットハンドの追加

リソース ▶ ロボットハンド のの [+] をクリックし、ロボットハンド設定ウィンドウを開きます。

1. **ロボットハンド名** に名前を付けて入力します。
2. **衝突モデル** を衝突検出のためのロボットハンドモデルを設定します。
3. **3Dモデル** を仮想空間に表示するためのロボットハンドモデルを設定します。3Dモデルの位置とサイズが実際と一致しない場合、**ロボットハンドモデルの位置またはサイズの調整** を参照して調整してください。
4. 実際の状況に応じて **回転対称** を、**回転対称性無し**、**N回対称** または **円対称** に設定します。**N回対称** に設定した場合、**対称回数** の設定も必要です。
- 5.

以下のいずれかの方法でTCPを設定します。

TCPパラメータを変更	オイラー角や四元数のパラメータを変更します。
正確なTCP位置姿勢の値を使用	他の方法で正確な位置姿勢の値を取得した場合、それを位置姿勢調整ウィンドウに貼り付けます。

6. 最後に [OK] をクリックします。

必要があれば、以上の手順を繰り返してロボットハンドを追加します。

ロボットハンドの削除

ロボットハンドを削除するには、次のいずれの方法を使用します。

- リソースツリー ▶ ロボットハンドでロボットハンド名をクリックして **Delete** を押します。
- リソースツリー ▶ ロボットハンドでロボットハンド名をクリックし、ドロップメニューで [削除] をクリックします。

ロボットハンドの変更

1. ロボットハンドの設定画面を開くには、次のいずれの方法を使用します。
 - リソースツリー ▶ ロボットハンドでロボットハンド名をダブルクリックします。
 - リソースツリー ▶ ロボットハンドでロボットハンド名を右クリックし、ドロップメニューで [ロボットハンド設定] をクリックします。
2. 実際に応じて各パラメータ値を調整します。
3. 最後に [OK] をクリックします。

現在のロボットハンドに設定

ロボットハンドが1つだけ追加されている場合、このハンドが現在のロボットハンドになります。

複数のロボットハンドを追加した場合、デフォルトでは最初に追加したロボットハンドが現在のロボットハンドとなります。

ロボットハンドを現在のロボットハンドに設定するには、**リソースツリー ▶ ロボットハンド**でロボットハンドを右クリックして、表示されるメニューで [現在のロボットハンドに設定] をクリックします。

ロボットハンドモデルの位置やサイズを調整

追加したロボットハンドモデルが正しい位置とずれた場合、以下の手順を実行します。

1. **リソースツリー**・**モデルライブラリ**のロボットハンドモデルファイルをダブルクリックします。
2. 表示されたモデル変換編集ウィンドウで位置姿勢を変更してロボットハンドモデルが仮想空間にある位置を調整します。

追加したロボットハンドモデルのサイズが実際と一致しない場合、以下の手順を実行します。

1. **リソースツリー**・**モデルライブラリ**のロボットハンドモデルファイルをダブルクリックします。
2. すると、モデル変換の編集画面が表示されます。
 - モデル全体のスケールを調整するには、**スケール**の値を変更します。
 - X、Y、Z方向でそれぞれスケールを設定するには、**X、Y、Zに同じスケールを適用**にチェックを外してからX、Y、Z方向のスケールをそれぞれ設定します。

4.3.275.4. シーンの物体

本節ではシーンの物体設定について説明します。

概要

シーンの物体とは、ロボット実機の動作環境の物体を指し、一般的にロボット安全柵や箱、パレット、カメラ、カメラブラケットなどが含まれます。

シーンの物体モデル

シーンの物体モデルはソフトウェアで実際の動作環境を再現し、衝突検出や経路計画を行うために使用されます。

ソフトウェアでは、直方体、円柱、箱のモデルを作成でき、また外部モデルをインポートして使用することもできます。外部モデルの3Dモデルと衝突モデルが対応する形式は下表のとおりです。

形式	STL	OBJ	DAE	Binvox
3Dモデル	○	○	○	×
衝突モデル	○	○	×	×

シーンの物体の「親子関係」

シーンの物体同士は、親子関係を作ることができます。親モデルを移動すると子モデルも追従し、子モデルを移動すると親モデルの位置は変わらず、親モデルを削除すると子モデルも削除されます。

シーンの物体の設定

リソースパネルでシーンの物体モデルを追加、編集、削除することができます。

シーンの物体を追加

直方体や円柱、箱をシーンの物体として使用するとき、以下の手順を実行します。

1. **リソース** > **シーンの物体**の[+]をクリックし、シーンの物体設定ウィンドウを開きます。
2. 物体設定画面で、実際の状況に応じて**シーンモデル**パラメータを設定します。
 - 直方体モデルの**X、Y、Z**を設定します。
 - 円柱モデルの**底面の半径、高さ**を設定します。
 - 箱モデルの**X、Y、Z、厚さ**を設定します。また、**箱のビジョン位置姿勢の有効範囲を設定**する必要もあります。
3. 物体名にカスタマイズのシーンの物体モデルの名前を入力します。
4. 追加したシーンの物体モデルはデフォルトでロボットベース座標系の原点のあります。[**物体の位置姿勢**]をクリックして物体の位置姿勢に関するパラメータを設定します。
5. [**OK**]をクリックします。

外部モデルを使用するとき、以下の手順を実行します。

1. **リソース** > **モデルライブラリ**のの[+]をクリックし、表示されたウィンドウで衝突モデルファイルと3Dモデルファイルを選択して[**開く**]をクリックします。
2. **リソース** > **シーンの物体**の[+]をクリックし、シーンの物体設定ウィンドウを開きます。
3. 物体設定画面で**シーンモデル**に**外部モデル**を選択します。
4. **衝突モデル**に衝突検出に衝突検出に使用するシーンの物体モデルを選択します。
5. **3Dモデル**に仮想空間に表示するシーンの物体モデルを選択します。
6. 物体名にカスタマイズのシーンの物体名を入力します。
7. 追加したシーンの物体モデルはデフォルトでロボットベース座標系の原点のあります。[**物体の位置姿勢**]をクリックして物体の位置姿勢に関するパラメータを設定します。
8. [**OK**]をクリックします。

必要があれば以上の手順を繰り返して複数のモデルを追加します。



リソースパネルでモデルを選択してから[+]をクリックして選択したモデルの子モデルを追加します。デフォルトでは、子モデルはその親モデルとは位置姿勢が同じです。

リソースパネルではモデルをドラッグしてほかのモデルの子モデルにすることと独立したモデルにすることができます。

シーンの物体を削除

以下のいずれかの手順を実行してシーンの物体を削除します。

- 仮想空間でシーンの物体モデルを選択するか、**リソース** > **シーンの物体**でシーンの物体モデル名を選択してから **Delete** を押します。

- 仮想空間でシーンの物体モデルを右クリックするか、**リソース・シーンの物体**で物体名を右クリックしてドロップダウンメニューで[削除]をクリックします。

シーンの物体を編集

1. 以下のいずれかの手順を実行してシーンの物体を編集します。
 - 仮想空間の物体モデルをダブルクリックします。
 - **リソース・シーンの物体**の物体名をダブルクリックします。
 - **リソース・シーンの物体**の物体名を右クリックし、表示されたドロップダウンメニューで[シーンの物体設定]をクリックします。
2. 実際の状況に応じて各パラメータを設定します。
3. [OK]をクリックします。



仮想空間でシーンの物体モデルをクリックして選択し、ドラッガーをドラッグして物体モデルの位置姿勢を調整できます。

箱のビジョン位置姿勢の有効範囲を設定

箱モデルを追加する場合、**箱の設定**画面で位置姿勢の有効範囲を設定する必要があります。

- 箱がただシーンの物体として使用し、位置姿勢の有効範囲を設定する必要がない場合、**位置姿勢の有効範囲を設定**のチェックを外します。
- 位置姿勢を箱の中に設定する場合、**箱内**を選択します。
- 位置姿勢を箱の上部に設定してもいい場合に、**箱内・外**を選択して**箱の上を超えた高さ範囲**パラメータを設定します。
- 位置姿勢の有効範囲をカスタマイズするとき、**カスタマイズ**を選択して**位置姿勢の有効範囲を設定**を設定します。

4.3.275.5. 対象物

本節では、対象物について説明します。

概要

対象物とは、段ボール箱や金属部品、接着剤塗布または溶接を実行する部品など、ロボットハンドが作用する物体です。

対象物の対称性

対称性とは、対象物がある軸で一定の角度回転したら元の形状と重なり合うことです。対象物の対称性を設定することでロボットハンドが把持や配置を実行する時の無駄な回転を回避し、時間を削減して把持計画の成功率を向上させることができます。

詳細については、[対象物の回転対称性](#)をお読みください。

把持範囲

対象物を把持できない場合でも、ロボットハンドの位置姿勢を一定角度の範囲内で微調整することで把持できる場合もあります。この範囲を把持範囲といいます。

詳細については、[対象物の把持範囲](#)をお読みください。

解の選択戦略

- **ロボットハンドの回転が最小**：この戦略を選択すると、「把持—配置」の過程でロボットハンドのZ軸の回転が最小な把持位置姿勢を優先的に把持点として使用します。これにより、ロボットが把持を実行した後、無駄な回転による対象物の落下を回避できます。
- **ロボットハンドと対象物の位置姿勢との差が最小**：この戦略を選択すると、ロボットハンドと対象物の位置姿勢との偏差角度が最小な点を優先的に把持点として使用します。
- **ロボットハンドと点群の衝突が最小**：この戦略を選択すると、ロボットハンドと対象物の点群の衝突体積が最小な点を優先的に把持点として使用します。

対象物設定

対象物を追加

1. **対象物名** に対象物の名前を付けて入力します。
2. 対象物の特徴に応じて **回転対称** を **Z軸を中心に**、**Y軸を中心に** または **X軸を中心に** に設定します（複数選択も可能）。その後、**対称回数** と **試行範囲** も設定します。対象物是对称性を持たない場合、このパラメータを設定する必要はありません。
3. 対象物の特徴に応じて **把持範囲** を **Y軸を中心に** と **X軸を中心に** に設定します（複数選択も可能）。その後、**試行範囲** と **試行間隔** も設定します。対象物の把持範囲がない場合、このパラメータを設定する必要はありません。
4. 持戦略の要求を満たすように **解選択戦略** を設定します。
5. **[OK]** をクリックします。

複数の対象物を追加する場合、以上の手順を繰り返してください。

対象物を削除

以下のいずれかの手順を実行して対象物を削除します。

- **リソース > 対象物** の下に表示されている対象物名を選択し、**Delete** を押します。
- **リソース > 対象物** の下に表示されている対象物名を右クリックし、表示されたドロップダウンメニューで **[削除]** をクリックします。

対象物を編集

1. 以下のいずれかの手順を実行して対象物設定ウィンドウを開きます。

- **リソース**、**対象物**の下に表示されている対象物名をダブルクリックします。
- **リソース**、**対象物**の下に表示されている対象物名を右クリックし、表示されたドロップダウンメニューで **[対象物設定]** をクリックします。

2. 対象物設定ウィンドウでパラメータを調整します。

3. **[OK]** をクリックします。

4.3.275.6. ステップパラメータ

移動

固定位置姿勢を経路点として設定します。

ロボットに送信するか

調整説明：このステップに対応する位置姿勢をロボットに送信するかどうかを設定します。

調整の例：計画するが送信しない場合、「経路計画」ステップはこのステップに対応する位置姿勢を出力しません。

運動タイプ

調整説明：このステップに対応する位置姿勢へのロボットの移動方式を示します。

調整の例：関節運動を選択した場合、ロボットは関節運動の方式でこのステップに対応する位置姿勢に移動します。

経路点の表示方式

調整説明：位置姿勢の調整方式を選択します。

ロボット実機の位置姿勢は、以下の操作でこのステップに入力することができます。

1. ティーチペンダントでロボットを移動させ、ツール位置姿勢または関節角度を記録します。
2. ステップパラメータで「位置姿勢を編集」（ツール位置姿勢を選択した場合）または「関節角度を編集」（関節角度を選択した場合）をクリックします。
3. 記録されたツール位置姿勢または関節角度を入力し、**[OK]** をクリックします。すると、仮想ロボットは対応する位置姿勢に移動します。

相対移動

前または次の経路点に対するロボットのZ軸方向の動きを設定します。

ロボットに送信するか

調整説明：このステップに対応する位置姿勢をロボットに送信するかどうかを設定します。

調整の例：計画するが送信しない場合、「経路計画」ステップはこのステップに対応する位置姿勢を出力しません。

運動タイプ

調整説明：このステップに対応する位置姿勢へのロボットの移動方式を示します。

調整の例：関節運動を選択した場合、ロボットは関節運動の方式でこのステップに対応する位置姿勢に移動します。

移動基準

調整説明：ロボット動作の基準位置姿勢を設定します。

調整の例：前の経路点を選択した場合、ロボットが前の経路点をもとに移動します。

移動方向

調整説明：ロボットの移動方向を設定します。

調整の例：対象物座標系のZ方向を選択した場合、ロボットが対象物座標系のZ方向に沿って移動します。

移動距離

調整説明：移動方向に沿ってロボットの移動距離を設定します。

ビジョン目標点

「経路計画」ステップに入力されたビジョン位置姿勢は、このステップに送信されます。

ロボットに送信するか

調整説明：このステップに対応する位置姿勢をロボットに送信するかどうかを設定します。

調整の例：計画するが送信しない場合、「経路計画」ステップはこのステップに対応する位置姿勢を出力しません。

運動タイプ

調整説明：このステップに対応する位置姿勢へのロボットの移動方式を選択します。

調整の例：関節運動を選択した場合、ロボットは関節運動の方式でこのステップに対応する位置姿勢に移動します。

4.3.276. サイクルトリガー

機能

トリガー信号を周期的に出力します。

使用シーン

これは通常、周期的なテストに使用されます。設定された期間に従ってプロジェクトを実行するようにトリガーします。

入力と出力

- 入力：なし。
- 出力：なし。

4.3.277. 透視変換



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

透視変換の前後の2つのセットの対応点に従って、入力画像に対して透視変換を行います。

使用シーン

通常、斜めの画像を水平の画像に補正するために使用されます。

入力と出力

- 入力：
 1. このポートに入力される画像は透視変換が行われます。
 2. 透視変換前の点。
 3. 2番目の入力ポートに対応する透視変換後の点。
- 出力：
 1. 透視変換後の画像。

4.3.278. 対象物の把持範囲

本節では、対象物の把持範囲について説明します。

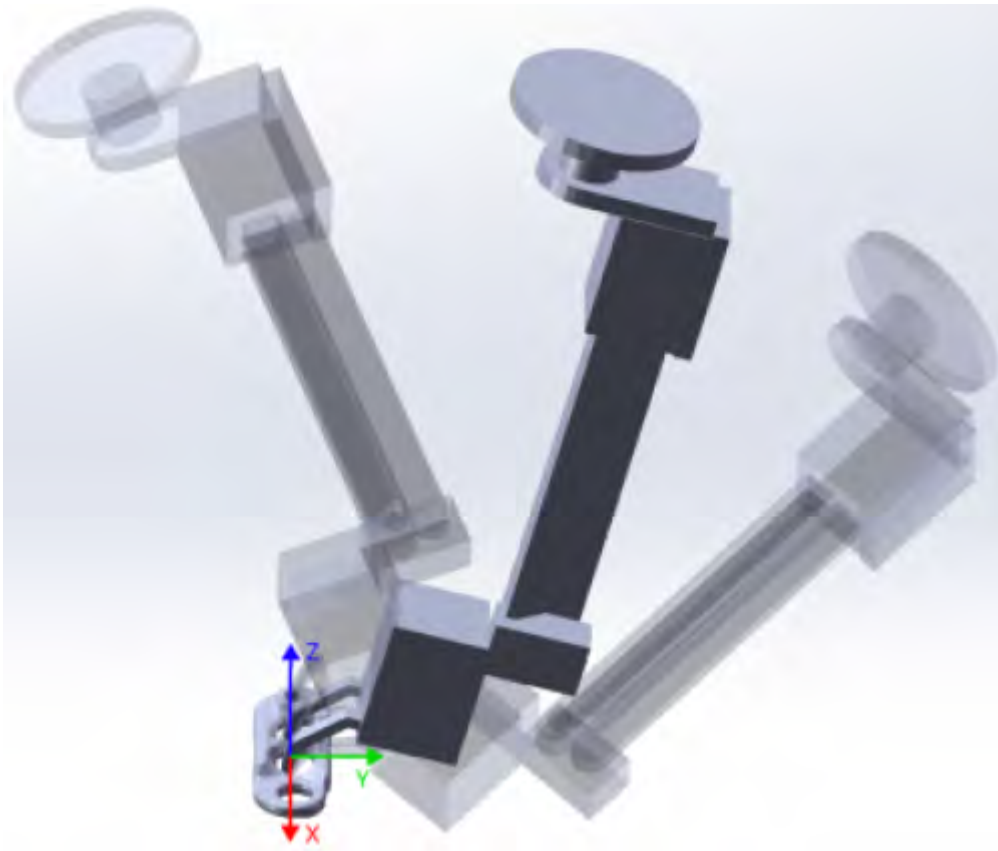
概要

対象物を把持できない場合でも、ロボットハンドの位置姿勢を一定角度の範囲内で微調整することで把持できる場合もあります。この範囲を把持範囲といいます。把持範囲を設定するとロボットが動作するときに衝突や特異点を回避することが可能になります。

配置位置姿勢を「対象物の位置姿勢」に設定した場合、把持範囲を利用して把持を実行するとき、ソフトウェアでは実際の把持位置姿勢と配置位置姿勢との角度が記録されます。配置を実行するとき、ソフトウェアはロボットハンドが正しい姿勢で対象物を配置するように、この角度によってロボットハンドの姿勢を調整します。

基準軸の確認

下図に示すようなシーンでは、ロボットハンドは対象物の位置姿勢のX軸を中心に一定の角度範囲内で把持を実行することができます。この場合、X軸が基準軸となります。



試行範囲と試行間隔

試行範囲は把持位置姿勢とビジョン位置姿勢の偏差角度の最大許容範囲です（下図では、BとCの間の角度が試行範囲となります）。

試行間隔は把持を試行する間隔です（下図では $\angle a$ は試行間隔となります）。

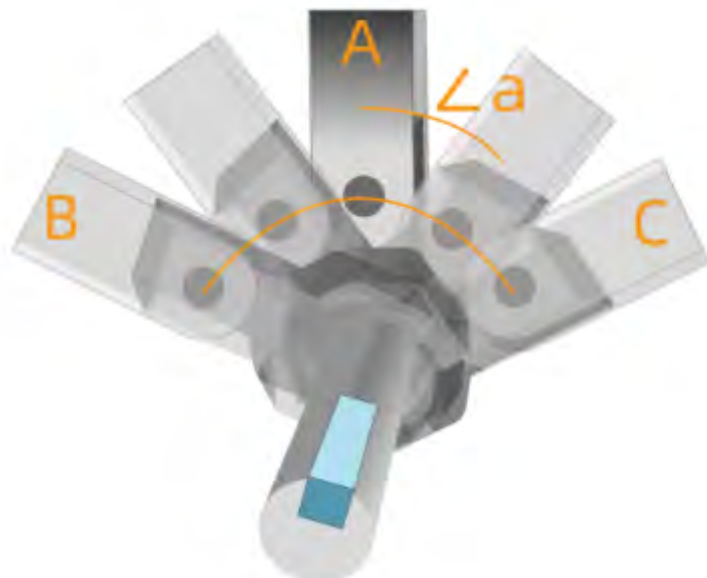


Figure 1. id7

試行回数

試行範囲と試行間隔によって自動的に計算されます。

試行間隔は 5° とし、試行範囲は $\pm 10^\circ$ とすると、ある方向の試行回数は $10/5 = 2$ となり、つまり試行回数は2となります（それぞれ 5° と 10° の位置で試行します）。もう一方の方向も併せて試行回数総計は $1+2*2 = 5$ となります（ -10° 、 -5° 、 0° 、 5° 、 10° の位置で試行します）。

4.3.279. 配置する対象物の位置姿勢を計画

機能

シーンの画像、箱と配置する対象物の情報を入力し、対象物が配置されたときの位置姿勢を計画します。

使用シーン

これは通常、対象物の把持やパレタイジングとデパレタイジングのシーンに、対象物が配置されたときの位置姿勢を計画するために使用されます。

入力と出力

● 入力：

1. ROI内のシーンのカラー画像（通常は `[vision-steps:scale-image-in-2d-roi::scale-image-in-2d-roi]` からのものです）。
2. ROI内のシーンの深度画像（通常は `[vision-steps:scale-image-in-2d-roi::scale-image-in-2d-roi]` からのものです）。

3. ROIパラメータ（通常は [\[vision-steps:scale-image-in-2d-roi:::scale-image-in-2d-roi\]](#) からのものです）。
4. 配置する対象物の3D寸法。パレタイジングとデパレタイジングシーンの場合は箱の寸法です。対象物の把持シーンの場合は対象物の3D境界ボックスの寸法です。
5. 箱の上面の平面位置姿勢。

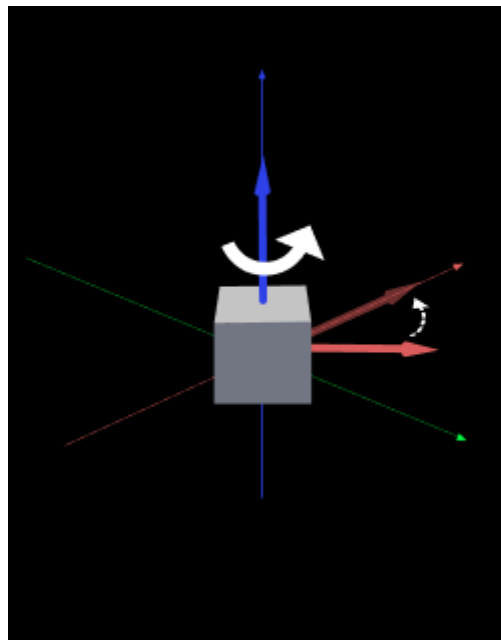
● 出力：

1. 計画した対象物が配置されたときの位置姿勢。順序は計画の配置順序です。
2. 可視化するための画像。

4.3.280. 位置姿勢のある軸を指定された方向に向ける

機能

位置姿勢のある軸を設定された基準方向に向くように調整します。

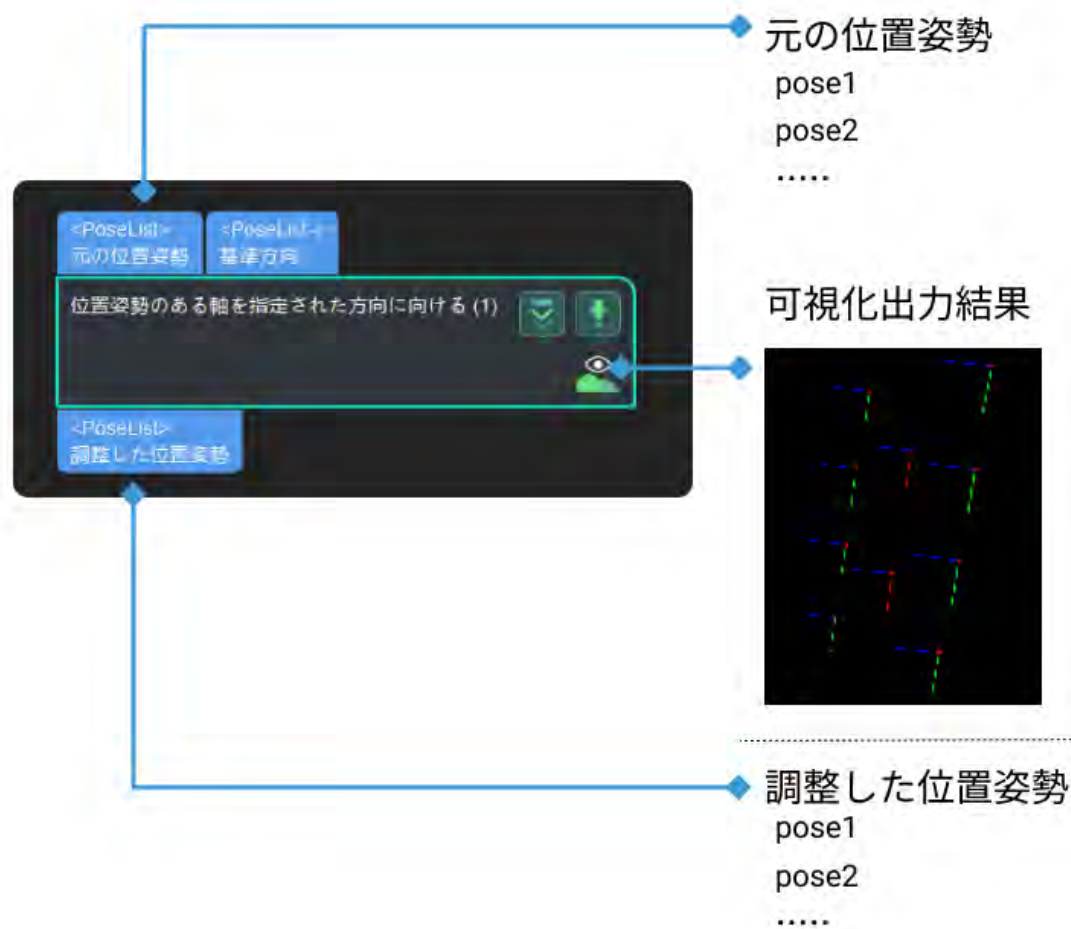


使用シーン

通常、位置姿勢の方向の調整が必要な各種のシーンに適用されます。

このステップは古いバージョンであり、対象物の対称性に応じて位置姿勢を回転する場合（このステップでパラメータ「補正方法」を「ROTATION」に設定してください）、新しいバージョンの [対称性制約で位置姿勢の軸を指定方向に回転](#) を使用し、ステップ [3Dベクトルをクイック作成](#) と併用してください。

入力と出力



パラメータの説明

可視化設定

位置姿勢表示の設定

初期値：出力位置姿勢のみを表示。

値のリスト：入力位置姿勢のみを表示、出力位置姿勢のみを表示、入力と出力位置姿勢をすべて表示。

調整説明：処理後の位置姿勢、処理前の位置姿勢、処理前後の全ての位置姿勢をそれぞれ表示します。

方法選択

補正方法

HARD_RECTIFY：初期値。二番目の入力（基準位置姿勢）があり、「useFirstRefPose」にチェックを入れた場合、基準方向は最初の位置姿勢に対応する軸の方向です。それ以外の場合は、すべての基準位置姿勢に対応する軸のタイプを基準方向とします。基準位置姿勢の入力がない場合、「reference Dir」は基準方向を決めます。

ROTATION：物体の対称性に応じて対称角度を設定し、調整する回転軸を指定した基準軸の方向とほぼ同じにします。

軸の設定

調整する軸

調整説明：補正する回転軸です。

オプション：X、Y、Z（初期値）

基準方向

X軸

初期値：0。

調整説明：基準軸Xの方向ベクトルです。

Y軸

初期値：0。

調整説明：基準軸Yの方向ベクトルです。

Z軸

初期値：1。

調整説明：基準軸Zの方向ベクトルです。

4.3.281. 点群クラスタリング

機能

指定されたルールに従って点群をクラスタリングします。対象物の分割に使用されます。



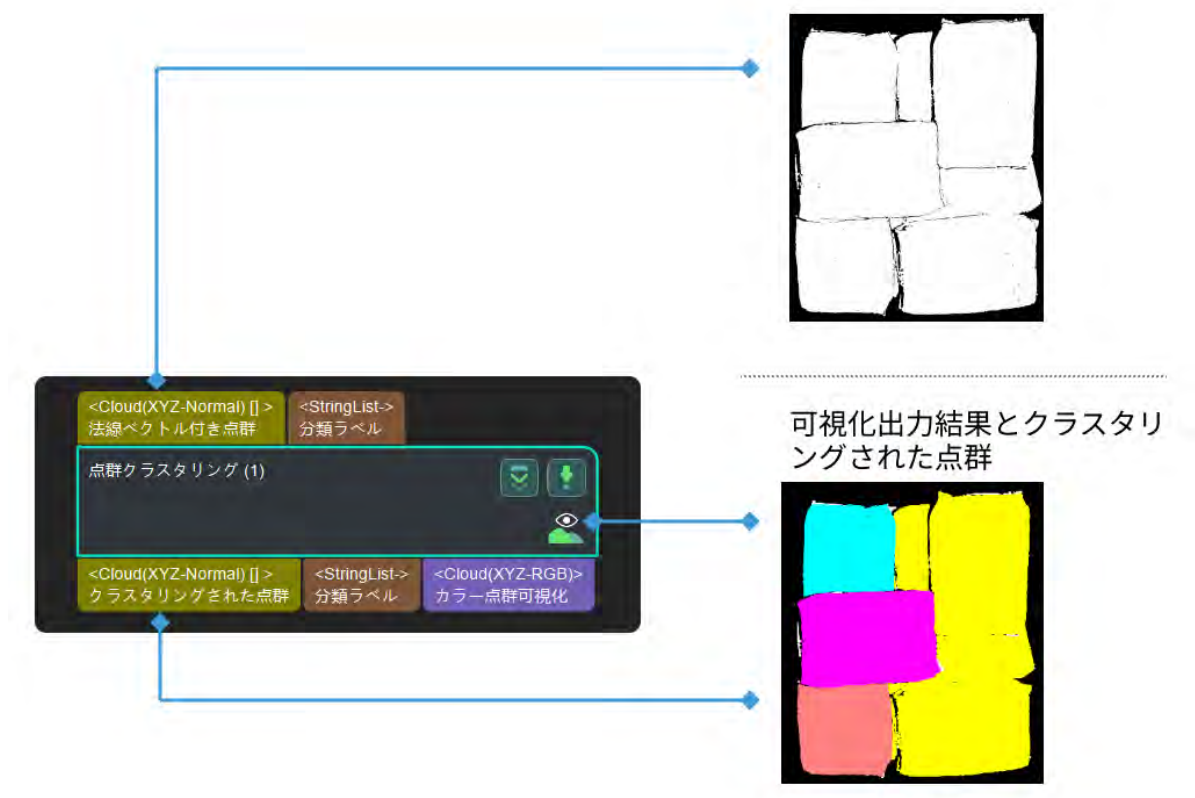
使用シーン

通常、点群前処理に使用され、干渉点群を除去します。

点群クラスタリングは、ユークリッド距離に基づいたクラスタリング（EuclideanCluster）と領域成長セグメンテーション（RegionGrowingSeg）の2つの方法があります。

点群が空間的に分離されている場合は、ユークリッド距離に基づいたクラスタリング方法（EuclideanCluster）を推奨します。点群が空間的に連続しているが、接続の曲率が大きく変化する場合は、領域成長セグメンテーション方法（RegionGrowingSeg）を推奨します。

入力と出力



パラメータの説明

クラスタリングアルゴリズム

初期値：EuclideanCluster

オプション：EuclideanCluster、RegionGrowingSeg

調整説明：このパラメータはクラスタリングの二つの方法を選択します。現場ではEuclideanClusterを使用することを推奨します。まずこれについて説明します。

- **EuclideanCluster**：距離によって同じ種類に属するかを判断します。
- **RegionGrowingSeg**：法線ベクトルと曲率によって同じ種類に属するかを判断します。

EuclideanCluster

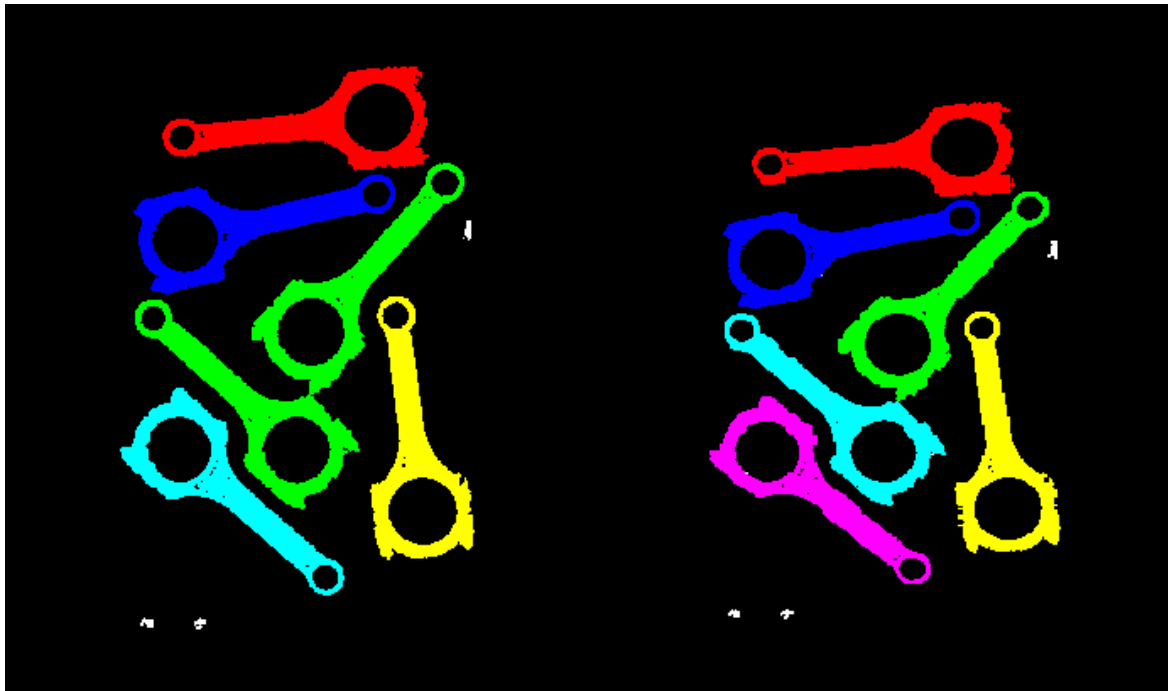
出力クラスター内の隣接点の最大距離

初期値：3.000 mm

パラメータ説明：このパラメータは、クラスタリングの許容誤差（ミリメートル単位）を設定するために使用されます。

調整説明：値を大きくしたら間隔が長い点は同じクラスに分類されます。小さくすれば間隔が近い点は異なるクラスに分類されます。

調整の例：下図に示します。左図は初期値3.000mmにした結果で、右図は5.000mmにした結果です。真ん中の緑の点群は異なるクラスに分類されます。



ユークリッドクラスター

初期値：4

調整説明：このパラメータはエリア成長プロセスに隣接点の法線ベクトル角度差のしきい値を調整します。値を大きくしたら隣接点の法線ベクトル角度差の許容値は大きくなり、角度差が大きい点でも同じクラスに分類されます。

RegionGrowingSeg

隣接点の個数

初期値：30

調整説明：このパラメータはエリア成長プロセスに検索ピクセル数を調整します。値を大きくしたら検索するピクセル数は多くなり、エリア成長の速度は速くなるのでクラスタリングによって取得するクラスターも少なくなります。

平滑しきい値

初期値：4

調整説明：このパラメータはエリア成長プロセスに隣接点の法線ベクトルの角度差のしきい

値を調整します。値を大きくしたら隣接点の法線ベクトルの角度差の許容値は大きくなり、角度差が大きい点でも同じクラスに分類されます。

曲率しきい値

初期値：1

調整説明：このパラメータはエリア成長プロセスに点の曲率のしきい値の上限を調整します。

クラスター内の最小点数

初期値：800

調整説明：このパラメータはクラスタリングされた結果をフィルタリングします。点数がクラスターの最小点数より大きいクラスターは出力されます。値を大きくしたら最終的に出力されるクラスターの数は少なくなります。小さくしたら最終的に出力されるクラスター数は多くなります。

クラスター内の最大点数

初期値：3000000

調整説明：このパラメータはクラスタリングされた結果をフィルタリングします。点数がクラスターの最小点数より小さいクラスターは出力されます。値を大きくしたら最終的に出力されるクラスター数は多くなりますが、多き過ぎると大きな影響は出ません。小さくしたら最終的に出力されるクラスター数は少なくなります。

調整の例：点群クラスタリングは5つのクラスターを生成し、点数は10000、20000、30000、40000、50000とします。クラスターの最大点数を45000に、最小点数を15000に設定したら、点数が10000と50000のクラスターは除去され、点数が20000、30000、40000のクラスターだけが出力されます。

GPUを使用

初期値：チェックを入れない

調整説明：このパラメータは、GPUを使用して計算を加速するかを設定するために使用されます。

4.3.282. 点群フィルタリング



これは古いバージョンのステップであり、メンテナンスが停止されますので、新しいバージョンのステップ [点群が要求を満たすかどうかを検証](#) を使用してください。このステップに関する詳細が必要な場合は、当社 (docs@mech-mind.net) までお問い合わせください。

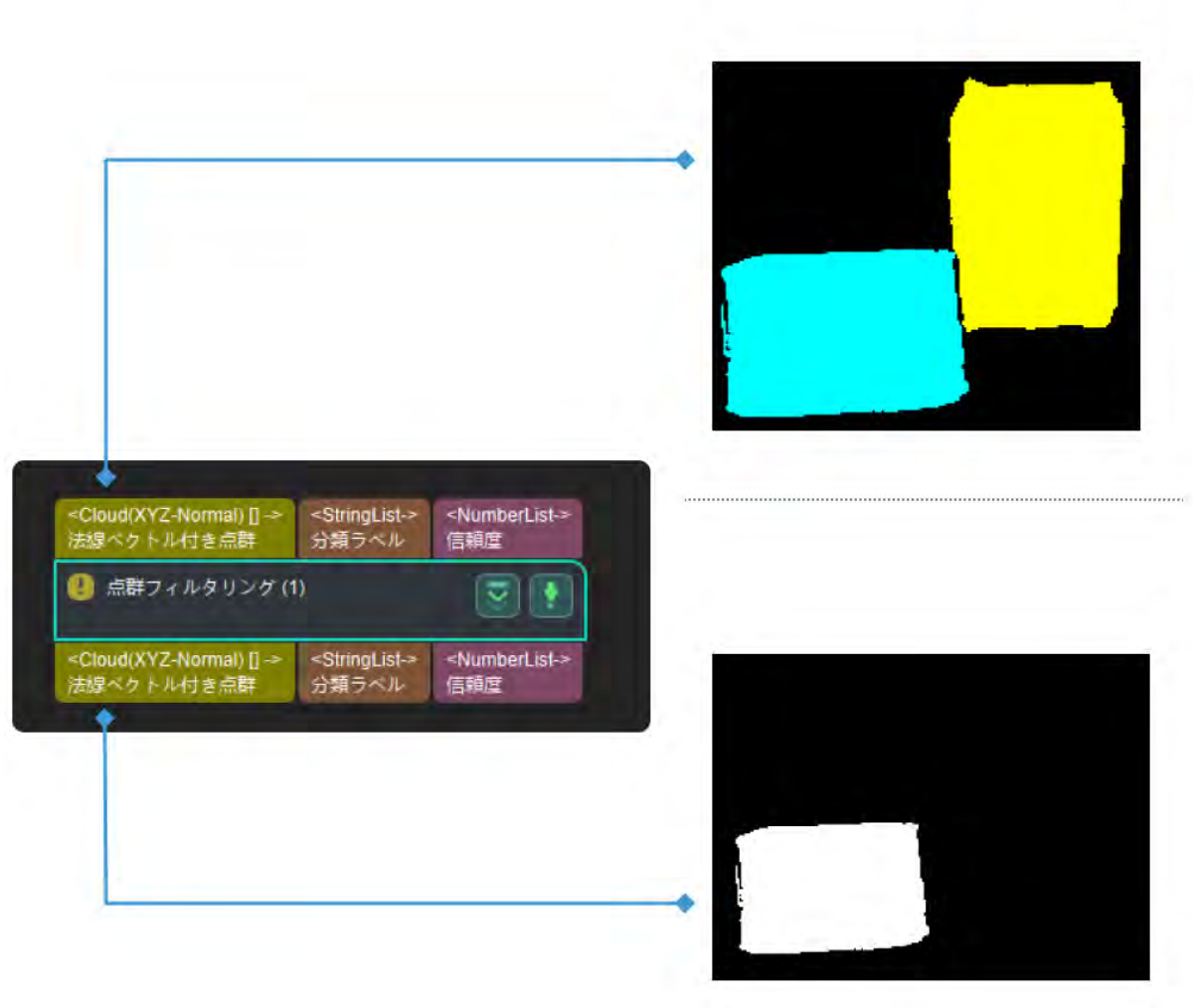
機能

設定されたルールに従って点群をフィルタリングします。

使用シーン

通所、点群の点数によって点群をフィルタリングします（パラメータ「フィルタリング方法」の「CloudCapacityFilter」に対応します）。

入力と出力



パラメータの説明

フィルタリング方法

パラメータ説明：このパラメータは、点群のフィルタリング方法を選択するために使用されます。CircleCloudsFilter、CloudCapacityFilter、RecCloudsFilter、SolidCloudsFilterが選択できます。

初期値：CircleCloudsFilter

推奨値：実際のニーズに応じて設定してください。

オプション：CircleCloudsFilter、CloudCapacityFilter、RecCloudsFilter、SolidCloudsFilter

CircleCloudsFilter

円形点群フィルタリング：3D点群を2Dマスク画像に変換し、マスクの外側輪郭を取得して、外側輪郭の円形度と周囲長を計算し、しきい値の範囲内にない点群を除去します。

輪郭検索モード

パラメータ説明：このパラメータは、輪郭検索モードを選択するために使用されます。RETR_EXTERNALとRETR_HULLの2つのタイプが含まれます。

初期値：RECT_EXTERNAL

推奨値：実際のニーズに応じて設定してください。

最小面積

パラメータ説明：このパラメータは、円形点群の最小面積を設定するために使用されます。

初期値：100

推奨値：実際のニーズに応じて設定してください。

最大面積

パラメータ説明：このパラメータは、円形点群の最大面積を設定するために使用されます。

初期値：1000

推奨値：実際のニーズに応じて設定してください。

最小真円度

パラメータ説明：このパラメータは、円形点群の最小真円度を設定するために使用されます。

初期値：0.5

推奨値：実際のニーズに応じて設定してください。

最大真円度

パラメータ説明：このパラメータは、円形点群の最大真円度を設定するために使用されます。

初期値：1

推奨値：実際のニーズに応じて設定してください。

最小凸性

パラメータ説明：このパラメータは、円形点群の最小凸性を設定するために使用されます。

初期値：0.8

推奨値：実際のニーズに応じて設定してください。

最大凸性

パラメータ説明：このパラメータは、円形点群の最大凸性を設定するために使用されます。

初期値：1

推奨値：実際のニーズに応じて設定してください。

最小慣性

パラメータ説明：このパラメータは、短軸長と長軸長との間の最小比率のしきい値を設定するために使用されます。

初期値：0.6

推奨値：実際のニーズに応じて設定してください。

最大慣性

パラメータ説明：このパラメータは、短軸長と長軸長との間の最大比率のしきい値を設定するために使用されます。

初期値：1

推奨値：実際のニーズに応じて設定してください。

CloudCapacityFilter

点群容量フィルタリング：点群リスト内の点数がしきい値範囲内でない点群を除去します。

最小点数しきい値

パラメータ説明：このパラメータは、点群の最小値を設定するために使用されます。

初期値：6000

推奨値：実際のニーズに応じて設定してください。

有効範囲：(0, maxPointsNum)

最大点数しきい値

パラメータ説明：このパラメータは、点数の最大値を設定するために使用されます。

初期値：360000

推奨値：実際のニーズに応じて設定してください。

有効範囲：(minPointsNum, $+\infty$)

RecCloudsFilter

長方形性の最小しきい値

パラメータ説明：このパラメータは、長方形性の最小値を設定するために使用されます。長方形度は、点群の実際面積と最小の外接長方形の面積の比率に等しくなります。

初期値：0.9000

推奨値：実際のニーズに応じて設定してください。

SolidCloudsFilter

比率しきい値

パラメータ説明：このパラメータは、塗りつぶされたソリッドに対する点群のサイズの比率の最小値です。

初期値：0.9000

推奨値：実際のニーズに応じて設定してください。

有効範囲：(0, 1]

ソリッド点群を除去

パラメータ説明：このパラメータは、ソリッド点群を除去するかどうかを設定するために使用されます。チェックを入れると、ソリッド点群が除去されます。それ以外の場合、中空点群は除去されます。

初期値：チェックを入れる。

推奨値：実際のニーズに応じて設定してください。

4.3.283. 点群形状検出器

機能

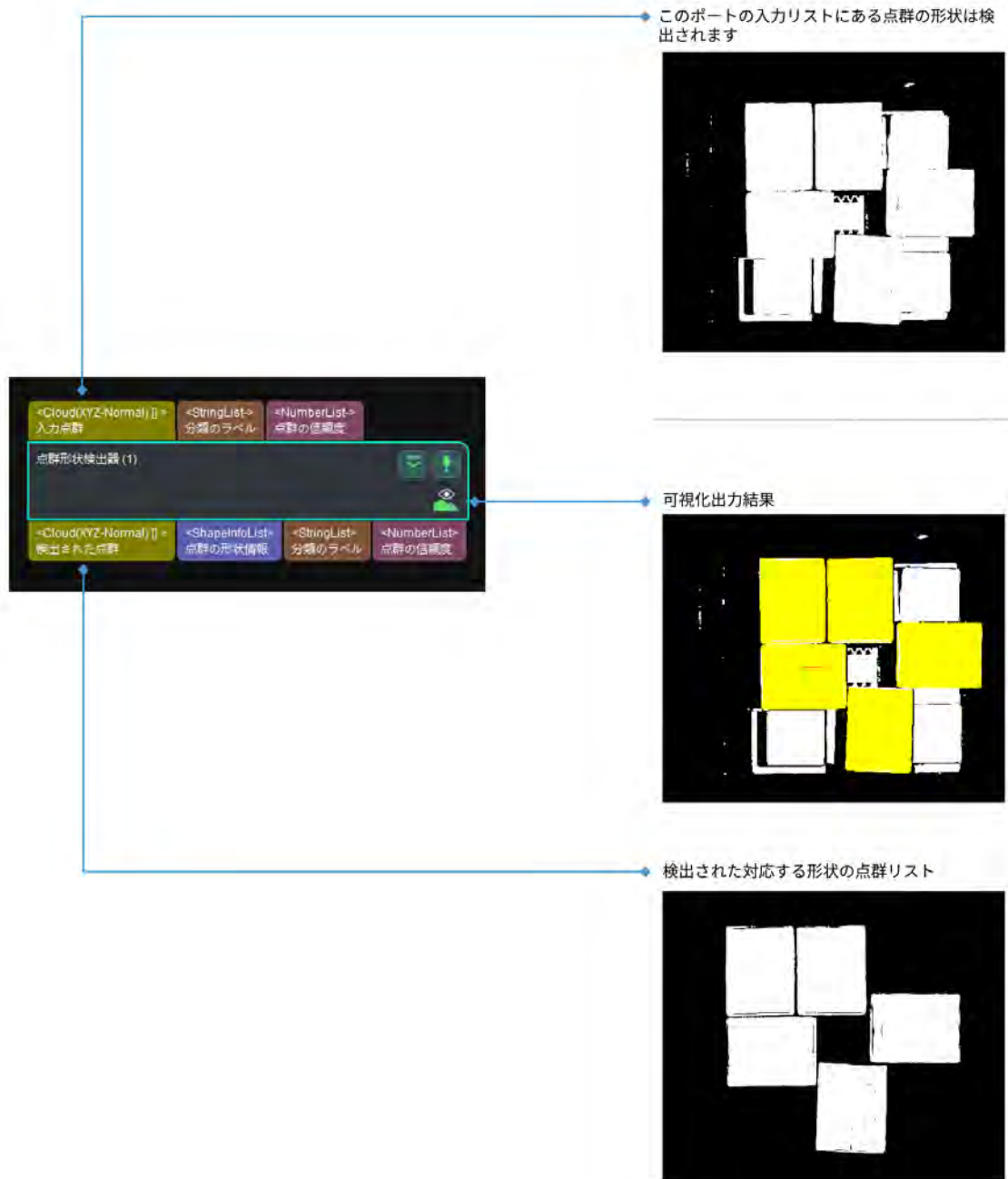
平面または円柱形状の点群を検出して出力します。その中、パラメータで必要な形状を選択できます。

使用シーン

一般的に平面、円柱形状の物体の検出に使われます。

円柱点群の把持位置姿勢を計算する場合、ステップ [\[vision-steps:calc-cylinder-poses:::calc-cylinder-poses\]](#) もしくは [オフセットによって位置姿勢を調整](#) をこのステップの後につなぐことができます。

入力と出力



パラメータの説明

検出する形状

調整説明：このパラメータは、検出する形状のタイプを選択するために使用されます。平面（plane）、円柱（cylinder）および球（sphere）が選択可能です。

初期値：平面

推奨値：実際のニーズに応じて設定してください。

検出設定

検出のアルゴリズム

調整説明：このパラメータは、検出中で使用するアルゴリズムを選択するために使用されます。方法1と方法2が選択可能です。（現在は方法1のみに対応し、後で最適化されます）

初期値：方法1

推奨値：方法1

法線ベクトルの偏差の最大しきい値

調整説明：このパラメータは、実際の画像内の点の法線ベクトルと、投影後の理論上に対応する標準形状上の点の法線ベクトルとの角度偏差の角度偏差の上限を表します。単位は°で、法線ベクトルと理論上の法線ベクトル間の誤差範囲の上限を与えます。

初期値：30.0000

推奨値：実際のニーズに応じて設定してください。

点群変動の誤差

調整説明：このパラメータは、点群変動の誤差の許容範囲を制御するために使用されます。

初期値：5.000mm

推奨値：[3.000, 5.000]

クラスタリングの最小間隔

調整説明：点群の異なる部分は同じクラスであるかどうかを判断するために使用されます。大きくすると、遠くの点群も同じクラスに分類されます。小さくすると、近くの点群の異なるクラスに分類されます。

初期値：5.000mm

推奨値：実際のニーズに応じて設定してください。

点群定量化の解像度

調整説明：このパラメータは、点の数および密度を制御するために使用されます。これを大きくすると、点群が密になり、計算精度は向上しますが、計算時間は長くなります。これを小さくすると、点群がまばらになり、計算精度が低下する可能性があります。計算時間は短縮されます。

初期値：5.000mm

推奨値：計算効果と時間の長さを共に確保する場合は、このパラメータの値を「クラスタリングの最小間隔」の2倍に設定するのが最適です。

出力結果の数

調整説明：このパラメータは、各入力点群が最終的に出力結果の数を制御するために使用さ

れます。

初期値：1

推奨値：実際のニーズに応じて設定してください。

点群の数によってソートするか

調整説明：点群の数によってソートするかどうかを決めるために使用されます。

初期値：チェックを入れる

推奨値：実際のニーズに応じて設定してください。

検出の成功率 (0-1.0)

調整説明：このパラメータは、検出の成功率を調整して、対応する形状を検出するのに十分な成功率があることを確認するために使用されます。

初期値：0.9900

推奨値：実際のニーズに応じて設定してください。

候補フィルター

最小被覆率 (0-1.0)

調整説明：すべて検出された形状のある点の合計数に対する、検出によって得られた形状のある点の数の比率です。このパラメータの設定値よりも小さい場合、対応する形状の点は除去されます。

初期値：0.1000

推奨値：実際のニーズに応じて設定してください。

最小平均距離誤差 (0-1.0)

調整説明：実際の点と、対応する検出された論理的な形状モデル内の点の間の平均距離誤差（メートル単位）です。大きくすると、誤差の大きい点群は除去されません。小さくすると、誤差の小さい点群でも除去されます。

初期値：10.000mm

推奨値：実際のニーズに応じて設定してください。

最小平均角度誤差 (0-90)

調整説明：実際の点と、対応する検出された論理的な形状モデル内の点の間の平均角度誤差（メートル単位）です。大きくすると、誤差の大きい点群は除去されません。小さくすると、誤差の小さい点群でも除去されます。

初期値：45.0000

推奨値：実際のニーズに応じて設定してください。

円柱のパラメータ

円柱の最小半径

調整説明：このパラメータは、円柱半径の最小値（ミリメートル単位）を設定するために使用されます。半径がこの値よりも小さい円柱はフィルターで除去されます。

初期値：10.000mm

推奨値：実際のニーズに応じて設定してください。

円柱の最大半径

調整説明：このパラメータは、円柱半径の最大値（ミリメートル単位）を設定するために使用されます。半径がこの値よりも大きい円柱はフィルターで除去されます。

初期値：100.000mm

推奨値：実際のニーズに応じて設定してください。

円柱の面の形状パラメータを最適化するか

調整説明：このパラメータは、円柱形状のパラメータ（中心点と軸）を最適化するかどうかを決定します。

初期値：チェックを入れない。

推奨値：チェックを入れる。

4.3.284. 点のフィルタリング

機能

ユーザーに指定されたルールに従って点群をフィルタリングし、ルールを満たさない点群は除去されます。

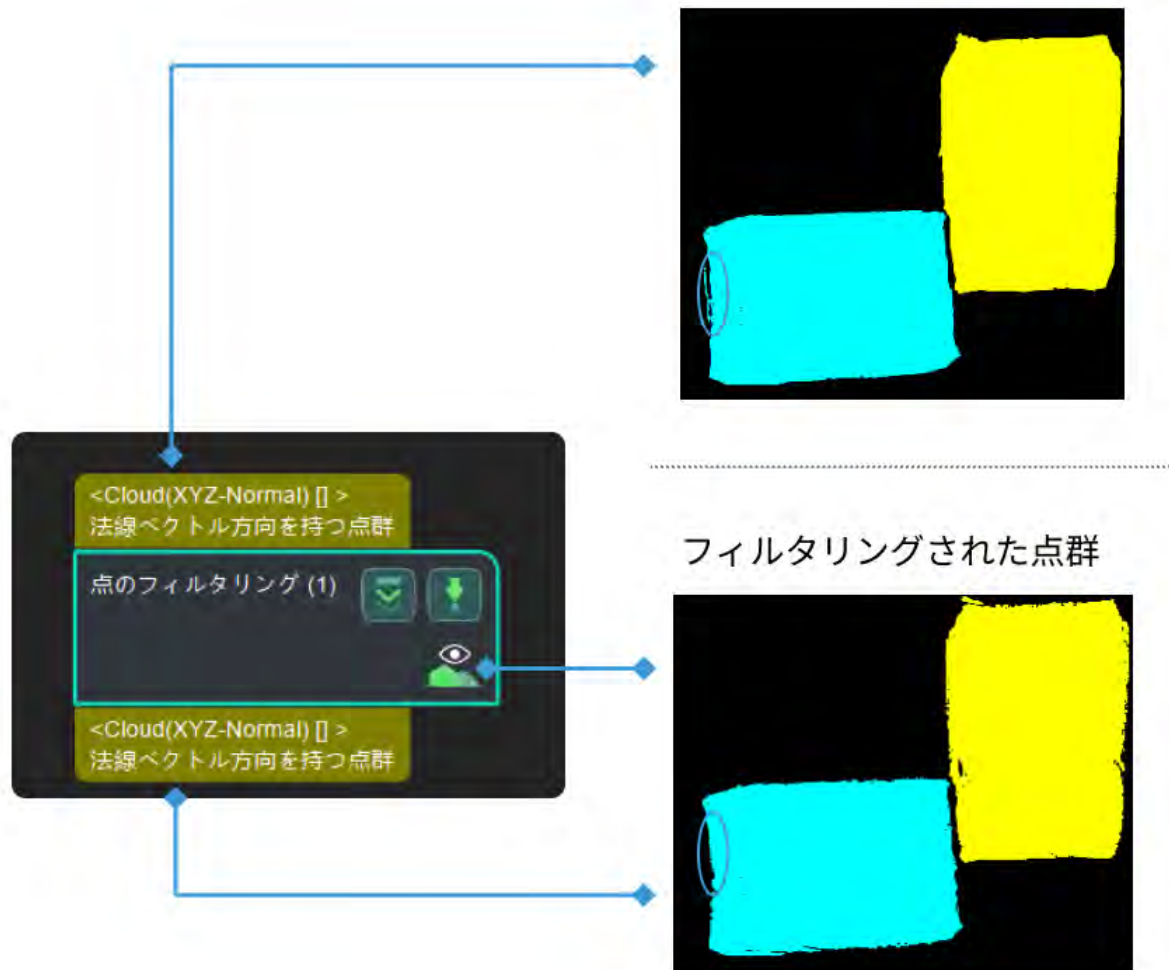


使用シーン

通常、点群前処理に使用され、点群内の明らかな外れ値やその他の不要な部分をフィルタリン

グして、その後の点群のクラスタリングを容易にします。

入力と出力

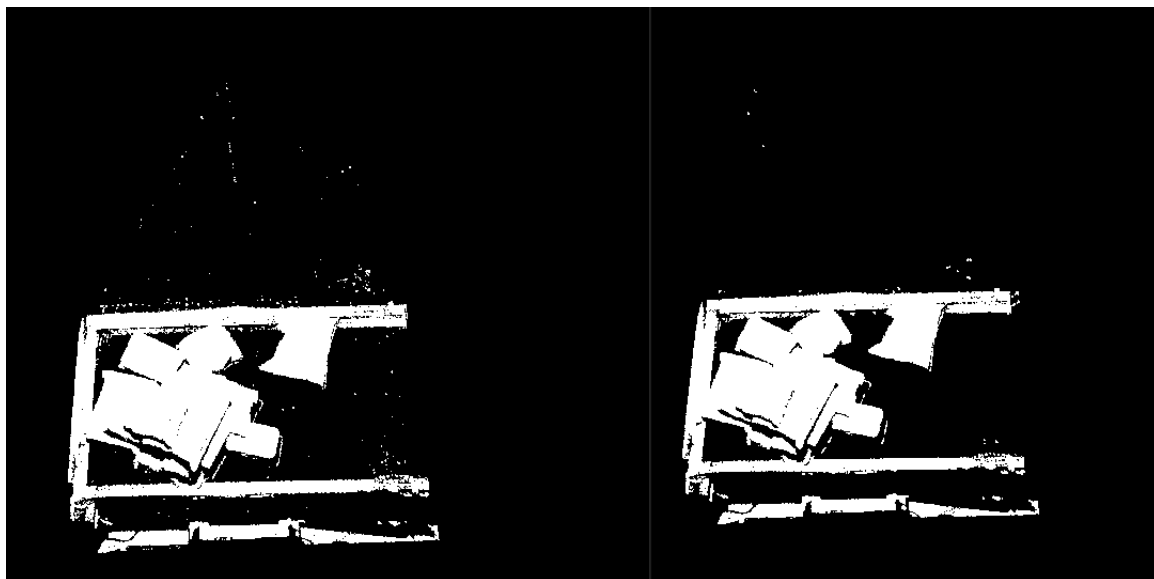


パラメータの説明

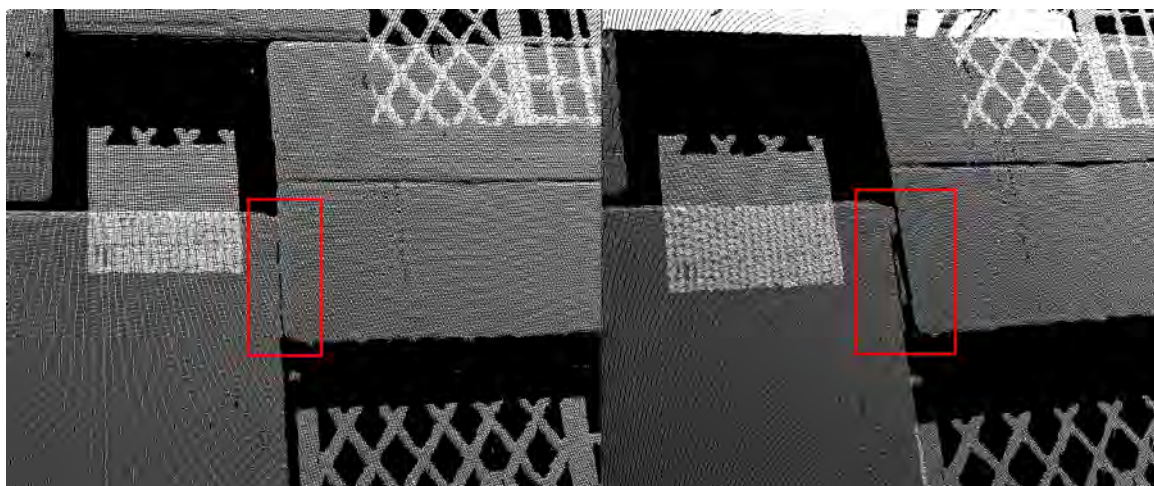
点のフィルタリング方法

調整説明：点群の点を除去する方法は二つあり、実際のニーズに応じて選択してください。

- [StatisticalOutlierFilter](#)：統計フィルター、指定したパラメータによって範囲を超えた点を、この方法で除去します。下図に示すように、明らか外れ値がある場合にこの方法を使用します。



- **NormalsFilter**:法線ベクトルフィルター、指定した基準方向によって、点群の各点と基準方向との角度を計算します。範囲を超えた点は除去されます。下図の赤枠に示すように、点群を階層化する必要があり、付着を回避する場合に使用されます。



StatisticalOutlierFilter

最近隣接点の数

初期値：30

調整説明：このパラメータは毎回検索する付近の点の数を調整します。値が大きいほど平均距離を計算するために使用される隣接点は多くなり、範囲も広がります。

標準偏差乗数

初期値：2

調整説明：このパラメータはしきい値判断条件の中に標準偏差の乗数を調整します。値が小さいほどしきい値は低くなり、保持される点も少なくなります。ただし、点の数が少なくなったら、情報も少なくなります。

NormalsFilter

基準方向

ロボットのZ軸を使用

初期値：チェックを入れない

調整説明：このパラメータは、カメラにあるロボットのZ軸を基準方向とするかどうかを決めます。チェックを入れれば、**基準方向X/Y/Z値** は全部無効になります。

基準方向のX値

初期値：0

調整説明：このパラメータは、法線ベクトルをフィルタリングするための基準軸を設定するために使用されます。XとY、Zの三つの値が一つのベクトルになり、基準軸の方向を設定します。**ロボットのZ軸を使用** にチェックを入れたら、三つの値は全部無効になり、ロボットのZ軸を基準軸とします。

基準方向のY値

初期値：0

調整説明：**基準方向のX値** と同様な方法で設定します。

基準方向のZ値

初期値：-1

調整説明：**基準方向のX値** と同様な方法で設定します。

角度しきい値の設定

最小角度差

初期値：0

調整説明：このパラメータは点をフィルタリングするプロセスに角度差の下限を調整し、単位は°です。点の法線ベクトルと基準ベクトルとの角度差がこの値より小さい場合に、点は除去されます。

最大角度差

初期値：90

調整説明：このパラメータは点をフィルタリングするプロセスに角度差の上限を調整し、単位は°です。点の法線ベクトルと基準ベクトルとの角度差がこの値を超えた場合に、点は除去されます。

4.3.285. 位置姿勢を一括調整

機能

位置姿勢に対して、座標系の変換、並進、回転およびソートなどの処理を行います。

使用シーン

汎用の位置姿勢調整ステップです。

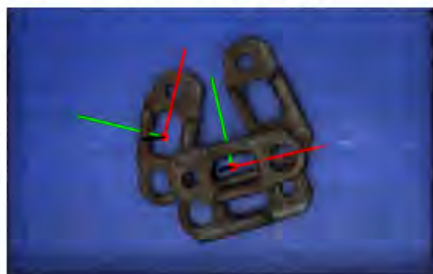
入力と出力

調整する位置姿勢リスト

Pose1
Pose2
Pose3
.....



可視化出力結果



調整後の位置姿勢リスト

Pose1'
Pose2'
Pose3'
.....

パラメータの説明

演算子ステップ

パラメータ説明：このパラメータから編集ツールを開いて位置姿勢を調整できます。

調整説明：パラメータで **[位置姿勢編集ツール]** をクリックしてその画面に入ります。このツールの使用方法については、[位置姿勢編集ツール](#) をご参照ください。

入力ポートの説明

パラメータ説明：このパラメータは、入力ポートの説明（英語）を設定するために使用され

ます。セミコロンで区切る必要があります。

調整説明：例えば、Pose; Indexは、それぞれPoselist; IndexListデータ型に対応します。

出力ポートの説明

パラメータ説明：このパラメータは、出力ポートの説明（英語）を設定するために使用されます。セミコロンで区切る必要があります。

調整説明：例えば、Pose; Indexは、それぞれPoselist; IndexListデータ型に対応します。

内部サブステップの完全な説明

パラメータ説明：このパラメータは、内部サブステップの完全な説明を入力するために使用されます。

内部サブステップ10つなぎの説明

パラメータ説明：このパラメータは、内部サブステップ10つなぎの説明を入力するために使用されます。

4.3.285.1. 位置姿勢編集ツール

機能

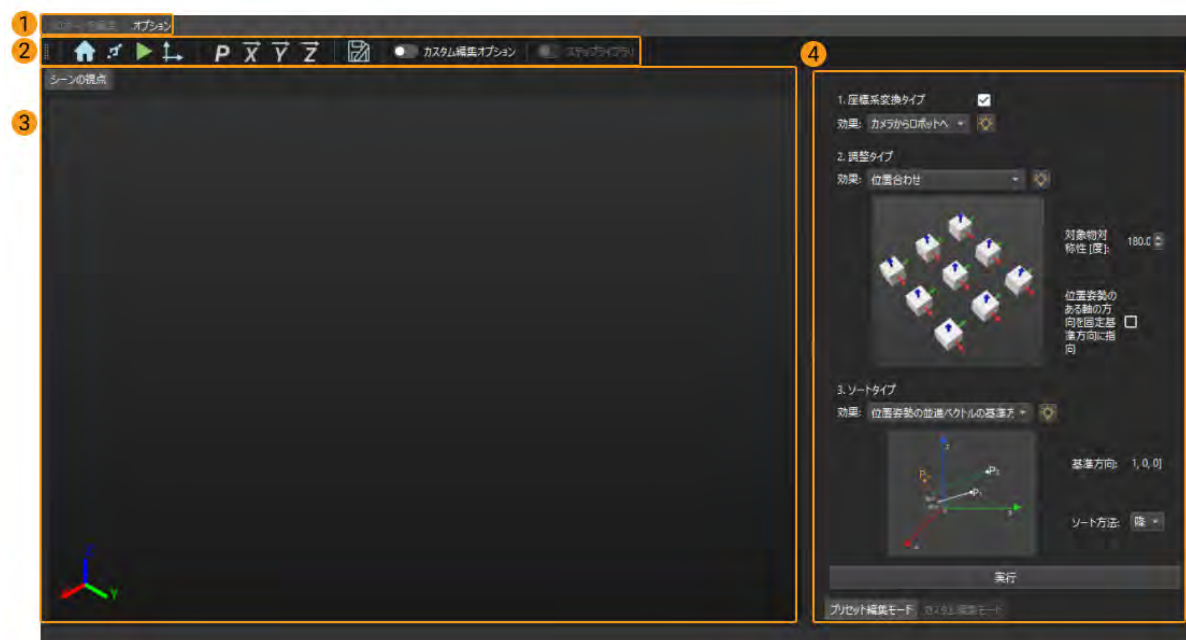
位置姿勢編集ツールの主な機能は、位置姿勢調整の使いやすさを向上させ、編集時の位置姿勢の可視化機能を追加、位置姿勢調整の学習コストを低減することです。

位置姿勢編集ツールには、以下の2つのモードがあります。

1. プリセット編集モード：座標系の変換や調整、ソートなど、プリセットされた一般的な機能により、位置姿勢をすばやく編集することができます。
2. カスタマイズモード：対応するステップを追加して位置姿勢の編集を行います。また、編集済みの位置姿勢が「ツール」で確認できます。

画面

下図に示すように、位置姿勢編集ツールの画面は、主に4つのエリアに分かれています。



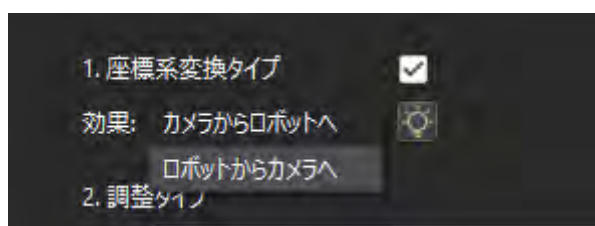
項目	説明
メニューバー	IOポートの編集やドラッグの種類の設定に使用されます。
ツールバー	位置姿勢をすばやく編集・調整するために使用されます。
シーンの視点（位置姿勢の可視化エリア）	その後にさらに調整するために、編集後の位置姿勢を確認します。
編集エリア	位置姿勢の調整、編集モードの切替をすばやく実行するために使用されます（編集エリアは、編集モードによって異なります）。

モード

プリセット編集モード

プリセット編集モードでは、以下の3つタイプがあります。

座標系変換：下図に示すように、カメラからロボットへまたはロボットからカメラへを選択し、位置姿勢の座標系変換を行います。



調整タイプ：「位置合わせ」、「パレットの中心から離れた位置を指す」、「一点を指す」の3種類があります。

ソートタイプ：5つのタイプを選択可能です。1.位置姿勢の並進ベクトルが基準方向に沿って投影する距離によってソートします。2.位置姿勢から基準点までの距離によってソートします。3.位置姿勢からXY平面上の基準点までの投影距離によってソートします。4.位置姿勢のXYZ値によってソートします。5.行または列によってソートします。

カスタム編集モード

カスタム編集モードを使用する場合、まずはツールバーのカスタム編集モードを有効にする必要があります。次に[**ステップライブラリ**]を有効にし、対応するステップを選択してから、[**追加**]をクリックします。




使用方法

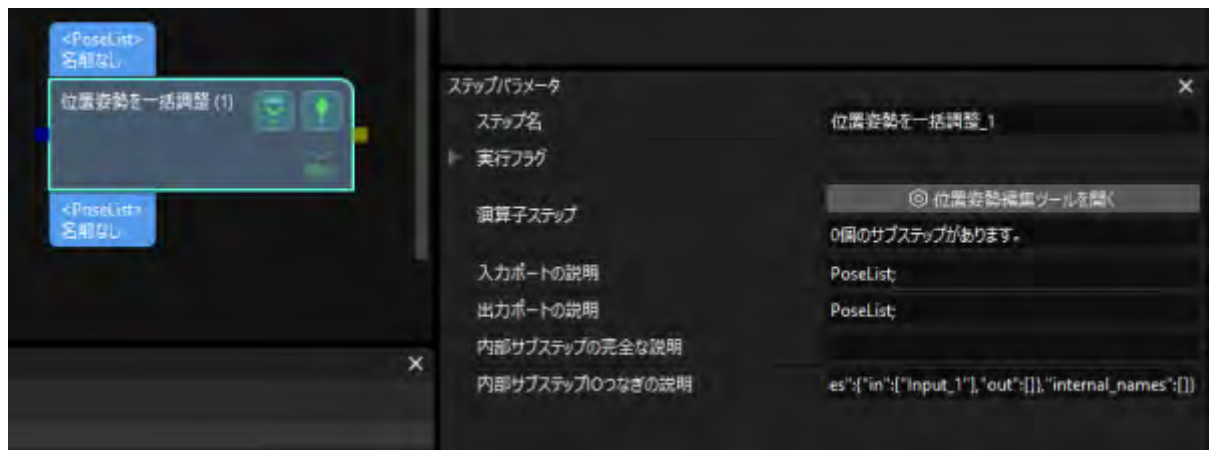
以下では、あるプロジェクトを例として位置姿勢編集ツールの使用手順について説明します。

プロジェクトに位置姿勢編集ツールを開くためのステップ [\[vision-steps:pose-adjustment-](#)

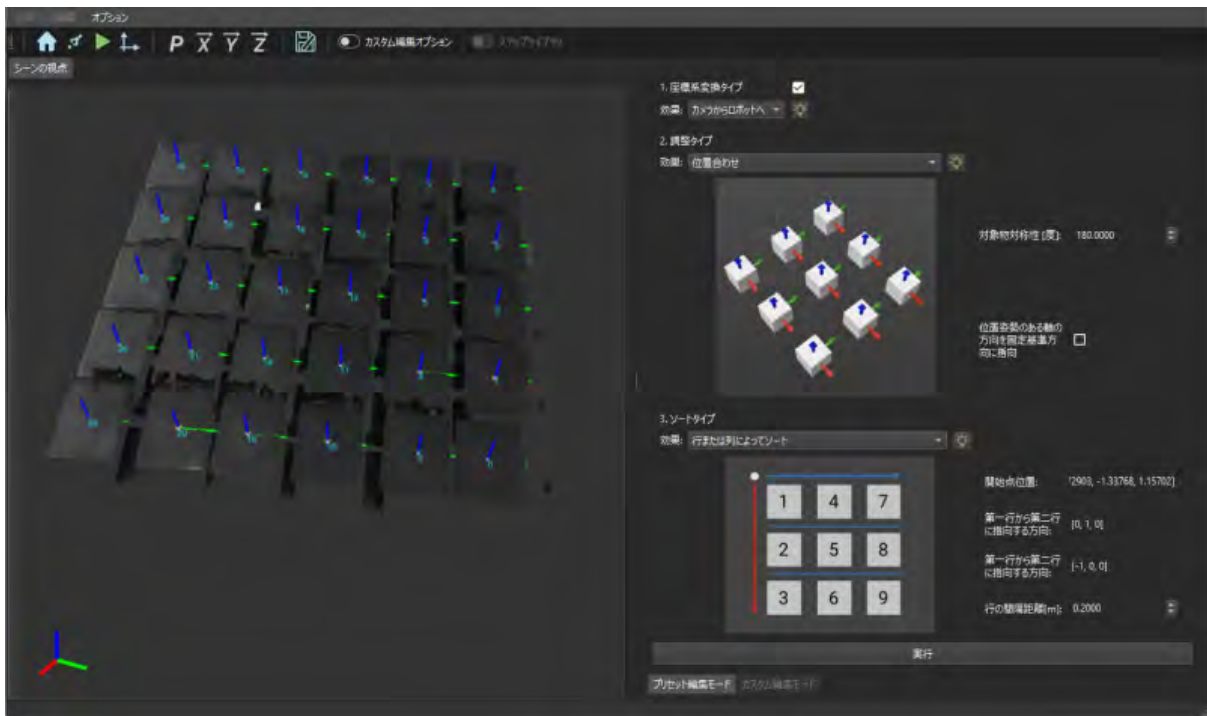
collection:::pose-adjustment-collection] を追加します。



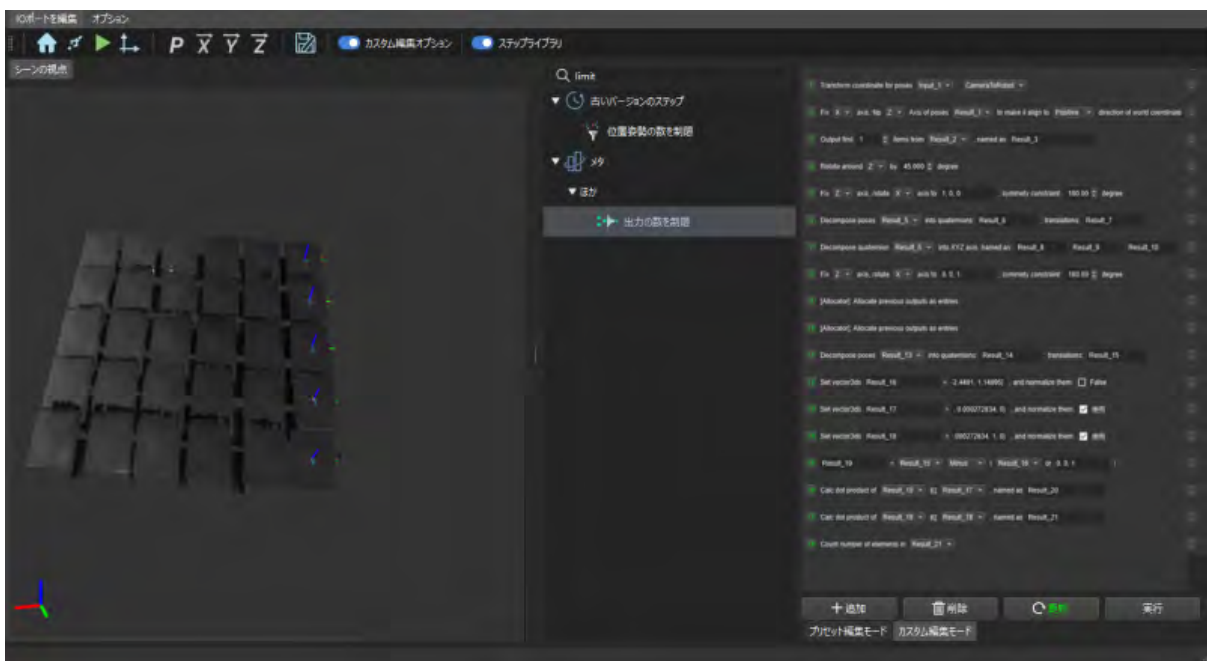
プロジェクトを実行し、入力位置姿勢を取得します。その後、ステップパラメータで  をクリックし、位置姿勢編集ツールを開きます。



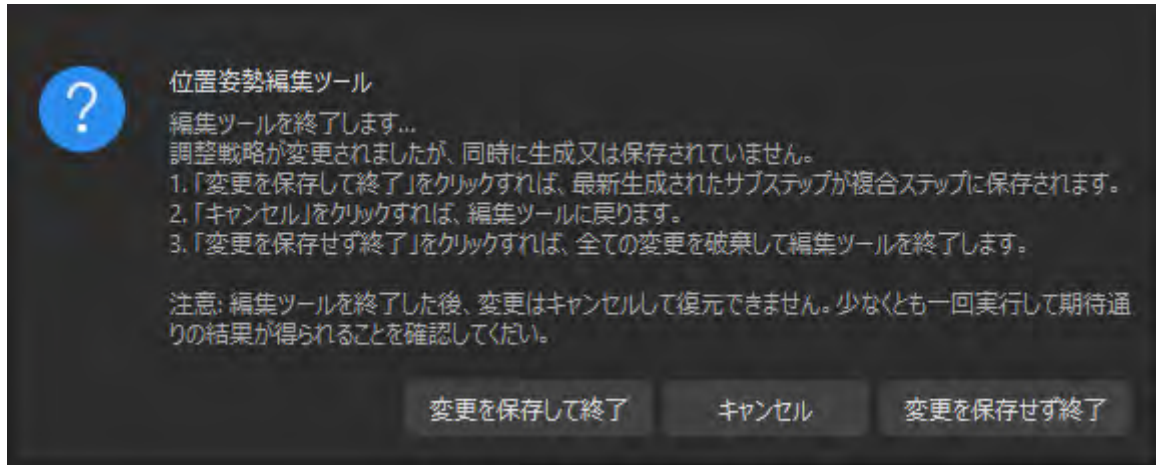
プリセット編集モードを使用する場合、座標系変換タイプ、調整タイプとソートタイプをそれぞれ設定してから、実行をクリックして結果を確認します。位置姿勢変換は下図に示します。



カスタム編集モードを使って出力数の制限を設定する手順は次の通りです。1.カスタム編集モードをクリックします。2.ステップライブラリを表示します。3.表示する出力数を選択します。4.「追加」をクリックします。5.制限数を記入します。6.「実行」をクリックして結果をプレビューします。位置姿勢変換は下図に示します。



位置姿勢が編集後、「変更を保存して終了」ボタンをクリックして、下図に示すように、編集済みの位置姿勢を保存します。



プロジェクトを実行して、プロジェクトの結果をすばやく確認します。

4.3.286. 位置姿勢フィルタリング



これは古いバージョンのステップであり、メンテナンスが停止されますので、新しいバージョンのステップ [\[vision-steps:validate-poses-by-included-angle-to-reference-direction:::validate-poses-by-included-angle-to-reference-direction\]](#) を使用してください。このステップに関する詳細が必要な場合は、当社 (docs@mech-mind.net) までお問い合わせください。

機能

入力位置姿勢のZ方向と基準方向との角度を計算します。設定された角度しきい値よりも小さい位置姿勢は保留され、設定された角度しきい値よりも大きい位置姿勢は除去されます。

使用シーン

これは古いバージョンのステップで、Z軸との角度しか計算できないため、任意の軸との角度によるフィルタリングに対応する新しいバージョンのステップ [\[vision-steps:validate-poses-by-included-angle-to-reference-direction:::validate-poses-by-included-angle-to-reference-direction\]](#) を使用してください。

入力と出力



4.3.287. 対象物座標系で四元数ベクトルによって位置姿勢を回転



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

回転ベクトルによって位置姿勢を回転します。

使用シーン

汎用の位置姿勢回転ステップです。特定の使用シーンはありません。

入力と出力



4.3.288. 対象物座標系で行列によって位置姿勢を変換



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

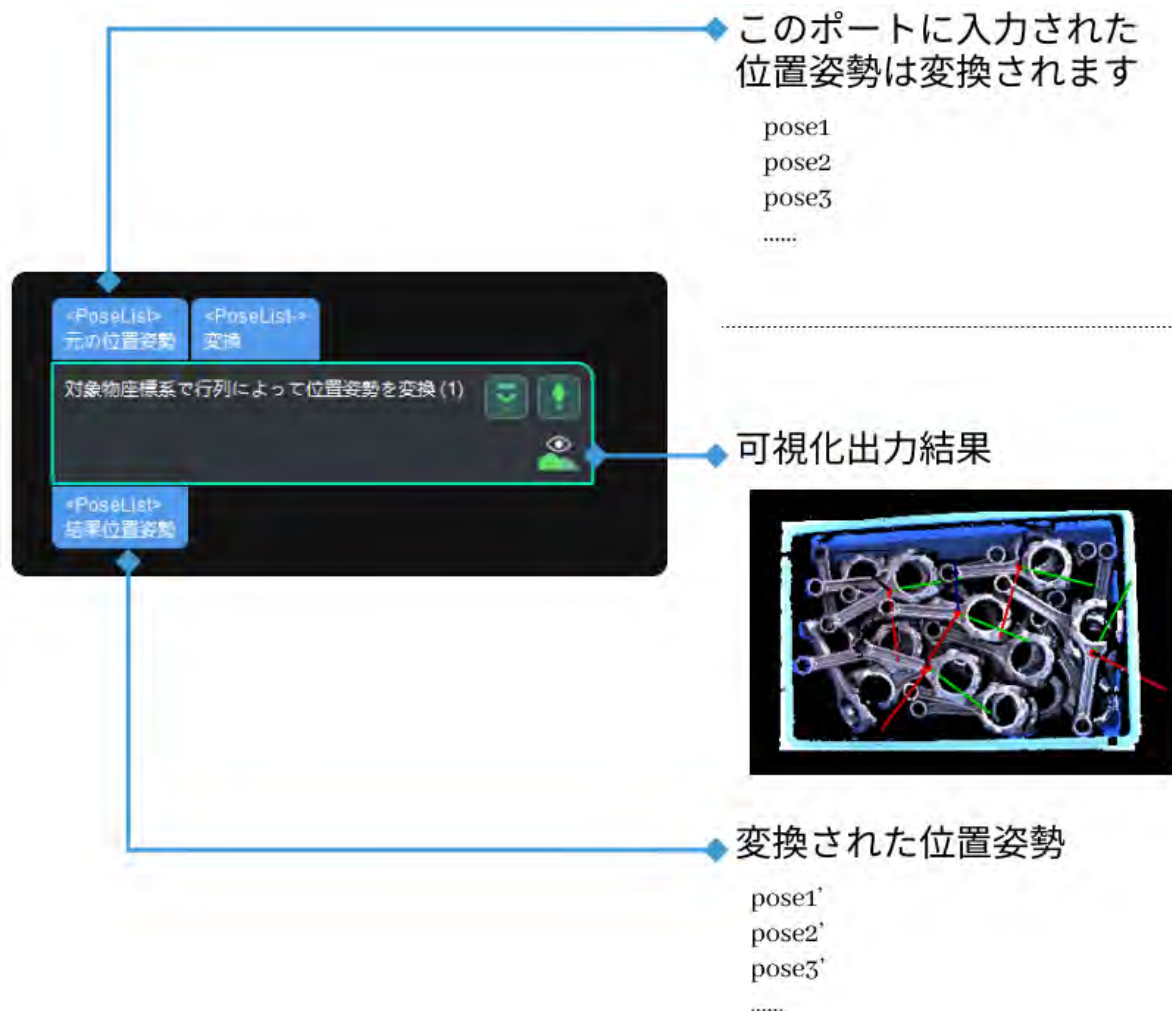
機能

ローカル座標系における、位置姿勢を変換します。

使用シーン

汎用の位置姿勢変換ステップです。特定の使用シーンはありません。

入力と出力



4.3.289. 統計データで位置姿勢の繰返し精度をチェック

機能

ビジョンプロジェクトの位置姿勢の安定性を検証できます。

同じワークを異なる位置で10～20回撮影することで、ワークがプロジェクトで出力する位置姿勢データ（最小値、最大値、平均値、範囲、標準偏差）を統計し、位置姿勢やその他のデータの安定性を判断します。



Eye-In-Hand、Eye-To-Handのどちらにカメラを設置しても、異なる位置でのワークの位置姿勢の安定性を確認する必要があります。異なる位置でワークを10～20回撮影する際に、ワークとカメラとの相対位置は固定されています。

- Eye-To-Handの場合、ワークの位置のみ移動できます。
- Eye-In-Handの場合、ワークが移動してもカメラもロボットと一緒に移動することができるため、同じワークを同じ位置で連続撮影する際にロボットとカメラの相対位置は固定されています。

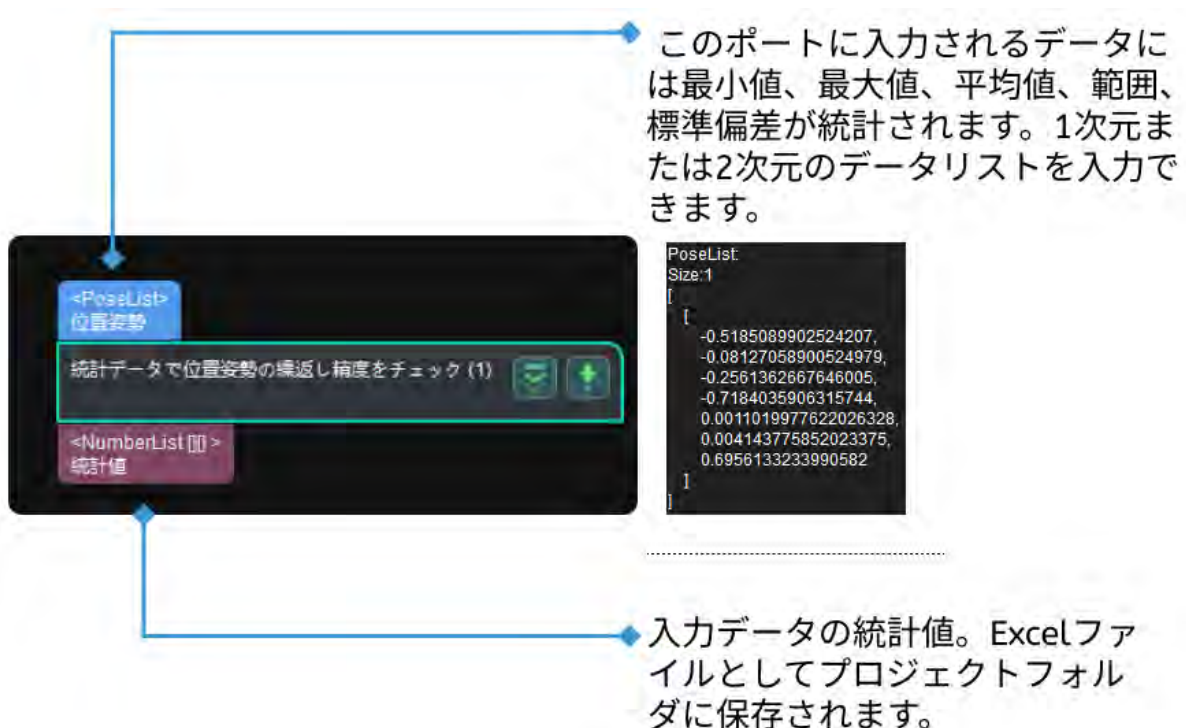
使用シーン

カメラの安定性をチェックしたり、ビジョンプロジェクトのパラメータ調整の有効性を検証するために使用されます。

3D位置姿勢推定に関連するステップの後につなぎ、モデルマッチングによって得られた位置姿勢の安定性を統計できます。また、**出力** ステップの前につなぎ、ワークの位置が変わらないようにしてから、カメラがワークの画像を取得して位置決め安定性と認識誤差を検証することもあります。

プロジェクトを実行後、このステップはExcel形式の「統計データファイル」を指定したパスに出力します。このファイルでは、位置姿勢のX、Y、Z値とオイラー角のX、Y、Z値の統計結果を確認できます。

入力と出力



パラメータの説明

統計するデータ

入力のデータ型

パラメータ説明：ステップに入力するデータ型を選択するために使用されます。異なる入力

データ型は、異なる入力ポートに対応します。

オプション：PoseListInput、NumberListInput

- PoseListInput：一つの位置姿勢のリスト、または複数の位置姿勢のリストを入力します。
- NumberListInput：その他の種類のデータを入力します。

初期値：PoseListInput

調整アドバイス：実際の状況に応じて設定してください。調整効果を [調整の例](#) に示します。

出力のデータ型

パラメータ説明：このパラメータは、異なる入力データ型を選択した後に出力されるデータ型を示します。

初期値：空間位置とオイラー角の統計値。

調整説明：調整する必要はありません。

位置姿勢座標偏差の許容範囲（0-1000）

パラメータ説明：入力した位置姿勢の位置座標の偏差の上限値（ミリメートル単位）を設定するために使用されます。値を入力した後、最大値と最小値との差はこのしきい値を超えた場合、このステップでは **異常値処理** パラメータの設定に従ってエラーを報告します。このチェックは、X、Y、Zの値について別々に行われます。

値の範囲：0-1000

初期値：1.0mm

調整説明：実際のニーズに応じて設定してください。調整効果を [調整の例](#) に示します。

オイラー角の偏差の許容範囲（0-360）

パラメータ説明：入力した位置姿勢のオイラー角の座標の偏差の上限値（単位は度）を設定するために使用されます。値を入力した後、最大値と最小値との差はこのしきい値を超えた場合、このステップでは **異常値処理** パラメータの設定に従ってエラーを報告します。このチェックは、 α 、 β 、 γ の値について別々に行われます。

値の範囲：0-360

初期値：0.100°

調整説明：実際のニーズに応じて設定してください。調整効果を [調整の例](#) に示します。

統計情報の保存

再開始

パラメータ説明：データを統計する際にワーク位置が変化した場合や、異常なデータが混入した場合に、このパラメータはデータを再統計するかどうかを決定するために使用されます。チェックを入れると、出力データに「Restart」がマークされます。また、「Restart」マーカーのタイムスタンプより前のデータが統計範囲に含まれず、グレー表示されます。

初期値：チェックを入れない。

調整説明：データを再統計する必要がある場合、このパラメータにチェックを入れてください。調整効果を [調整の例](#) に示します。



- 実行が1回成功すると、このパラメータチェックが外れている状態にリセットされます。
- このパラメータにチェックを外すと、保存された統計情報に異常データが常に表示されます。

異常値処理

パラメータ説明：このパラメータは、統計データに異常値がある場合に異常値の処理方式を選択するために使用されます。

オプション：異常値を発見したら即エラー報告、異常値を記録してマーク。

- 異常値を発見したら即エラー報告：異常値を発見した直後にエラーのみを報告し（エラー報告には以下の3種類がある）、異常データはファイルに書き込まれません。
 - 以前の数と現在の数の不一致によるエラー。これは、[データをアンパック](#) ステップを使用して解決することが可能です。
 - 位置座標の偏差とオイラー角の偏差がしきい値を超えた場合に報告するエラー。これは、[位置姿勢座標偏差の許容範囲](#)、[オイラー角の偏差の許容範囲](#) を調整して解決することが可能です。
 - 統計データの保存ファイルを開いてデータを保存できないために報告するエラー。これは、統計データの保存ファイルを閉じてからプロジェクトを再実行することで解決できます。
- 異常値を記録してマーク：異常値を発見しても、エラーは報告されません。異常データはファイルに書き込まれて統計に関与され、赤で表示されます。

初期値：異常値を発見したら即エラー報告。

調整説明：実際のニーズに応じて設定してください。調整効果を [調整の例](#) に示します。

小数点以下の桁数 (0-8)

パラメータ説明：このパラメータは、統計データの小数点以下の桁数を設定するために使用されます。

初期値：3

値の範囲：0-8

調整説明：実際のニーズに応じて設定してください。

ファイルベース名

パラメータ説明：このパラメータは、保存された統計データファイルの名前を設定するために使用されます。設定しない場合、統計データファイルの名前は「posesStatistics」になります。ファイルのベース名が変更された場合、この場合で自動的に **再開始** にチェックが入り、統計が再開されます。

初期値：なし。

調整説明：実際のニーズに応じて設定してください。



ワーク位置が変化した場合や、異常なデータが混入した場合、このステップの **再開始** パラメータにチェックを入れたら、 **ファイルベース名** を設定して、データ保存ファイルを再作成してデータを収集することが可能です。

保存ディレクトリ

パラメータ説明：このパラメータは、統計データファイルの保存ディレクトリを設定するために使用されます。設定しない場合、統計データファイルはプロジェクトフォルダに保存されます。

初期値：なし。

調整説明：右側にある  をクリックして保存ディレクトリを選択できます。

調整の例

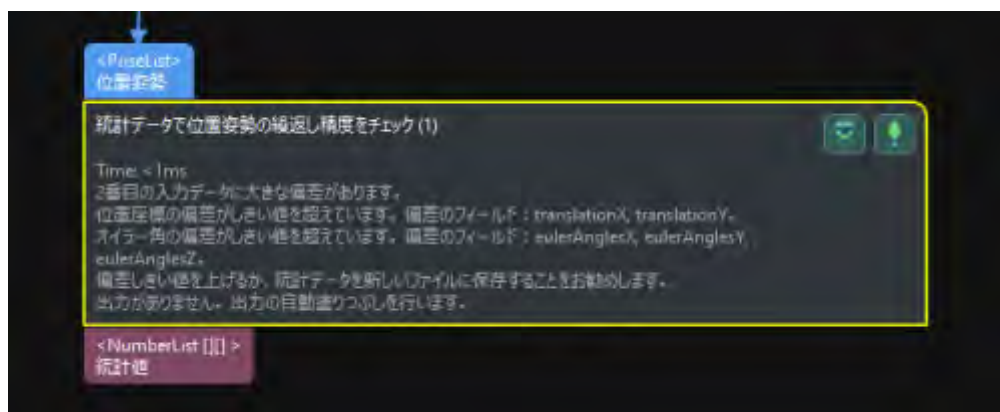
入力のデータ型

下図に示すように、このステップの入力ポートは、パラメータ **入力のデータ型** の設定によって異なります。



位置姿勢座標/オイラー角の偏差の許容範囲

下図に示すように、現在のデータと前のデータとの差は設定したしきい値を超えた場合、このステップではエラーを報告します。



再開始

再開始 にチェックを入れると、統計データファイルでは「Restart」マークのタイムスタンプ以降のデータが統計範囲に含まれ（下図の枠のように）、以前のデータが統計範囲に含まれず、グレー表示されます。

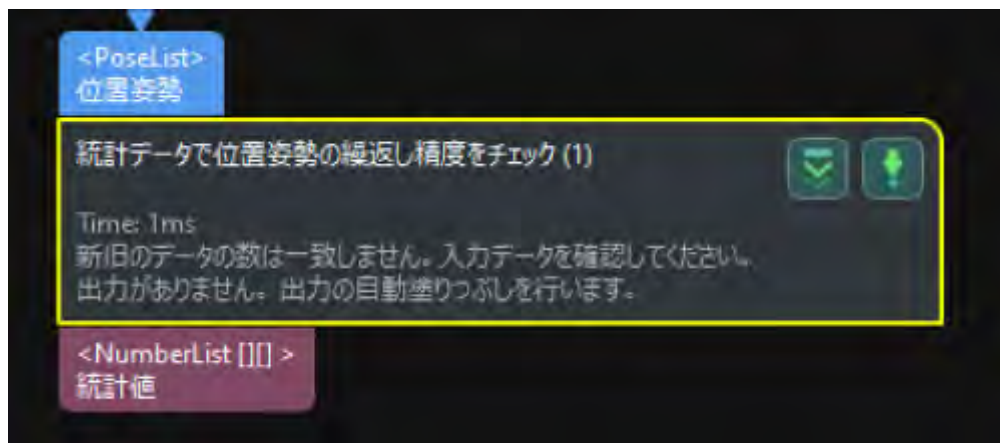
2022-11-10 14:10:15	0.000
2022-11-10 14:10:14	0.000
2022-11-10 14:10:14	0.000
Restart:2022-11-10 14:10:13	0.000
2022-11-10 14:10:17	3.000
2022-11-10 14:10:15	0.000
2022-11-10 14:10:14	0.000
2022-11-10 14:10:14	0.000

異常値処理

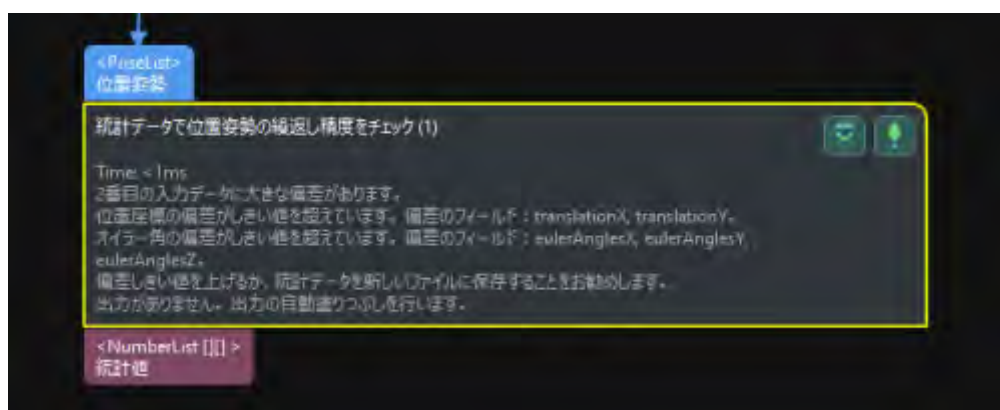
異常値を発見したら即エラー報告

このステップで発生する可能性のあるエラーは、以下の3種類です。

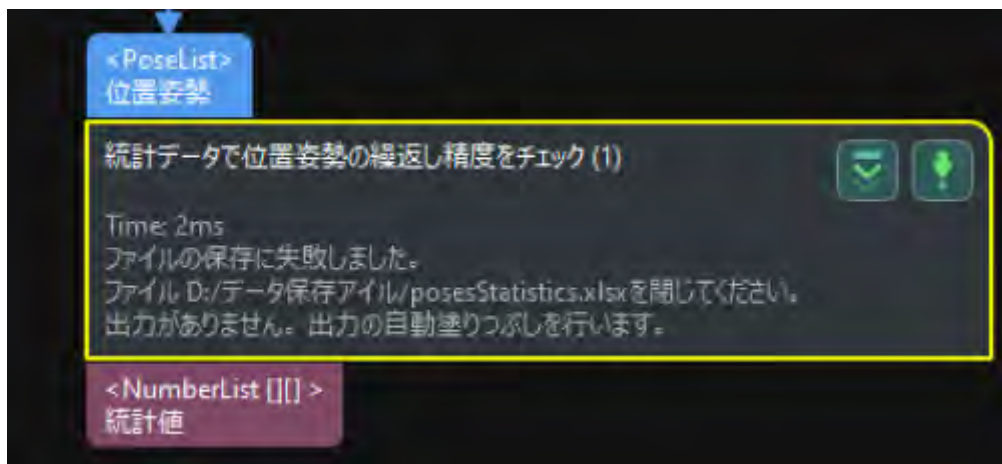
- 以前の数と現在の数の不一致によるエラーを下図に示します。



- 位置座標の偏差とオイラー角の偏差がしきい値を超えた場合に報告するエラーを下図に示します。



- 統計データの保存ファイルを開いてデータを保存できないために報告するエラー。



異常値を記録してマーク

このオプションを選択すると、異常値を発見したときに、異常データ（下図の1）およびこのフィールドのデータの統計値（下図の2）が赤色で表示されます。

mm & degree	translationX	translationY	translationZ
**Range	1 1.500	1.300	1.800
*Stdev	0.449	0.447	0.583
Min	0.100	1.500	2.000
Max	1.600	2.800	3.800
Mean	0.888	2.356	2.950
2023-01-10 12:12:51	1.000	2.150	3.100
2023-01-10 12:11:52	0.900	2.700	3.000
2023-01-10 12:11:07	0.800	2.600	2.900
2023-01-10 12:10:16	0.100	2.800	2.000
2023-01-10 12:10:00	0.500	1.500	2.300
2023-01-10 12:09:48	1.000	2.700	3.000
2023-01-10 12:09:38	1.200	2.400	3.500

4.3.290. 把持位置姿勢を予測（複数タイプ）

機能

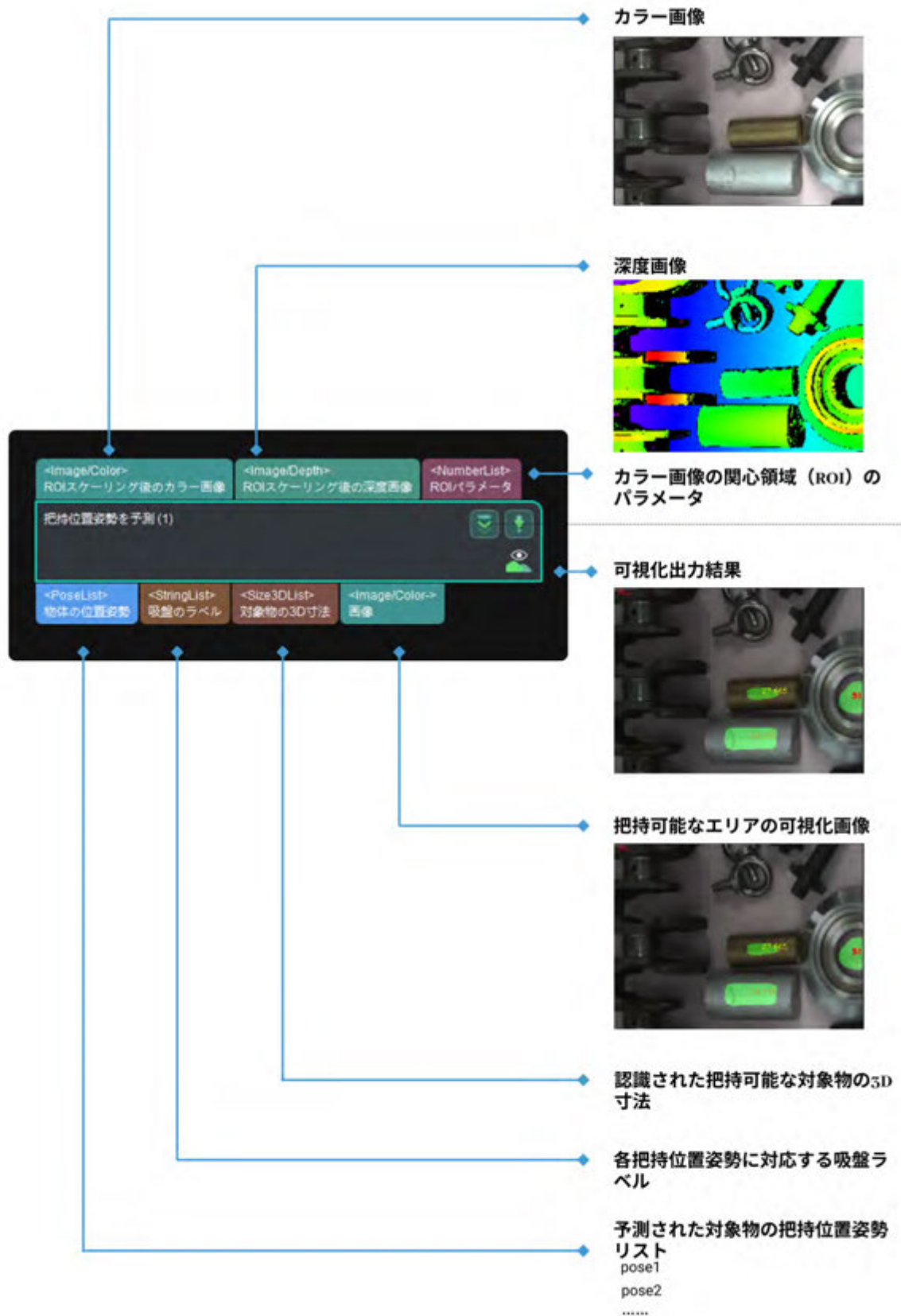
2D画像と3D深度画像に基づいて、画像内の把持可能な対象物を認識して、対応する把持位置姿勢を出力します。



使用シーン

このステップは通常、バラ積みされた様々な種類の対象物を仕分けるために使用されます。

入力と出力



パラメータの調整説明

サーバー

サーバーIP

初期値：127.0.0.1

調整説明：実際の状況に応じてディープラーニングサーバーIPを設定してください。

サーバーポート（1-65535）

初期値：60052

調整説明：対象物の吸着および対象物の吸着（箱なし）のプロジェクトで、60000~65535のポート番号を使用する必要があります。

稼働距離

最小稼働距離

初期値：0

調整説明：カメラからシーン内の物体までの最短距離（ミリメートル単位）です。箱を使用する場合、このパラメータをカメラの底部から箱の上端までの距離に設定してください。実際のニーズに応じて設定してください。

最大稼働距離

初期値：3000

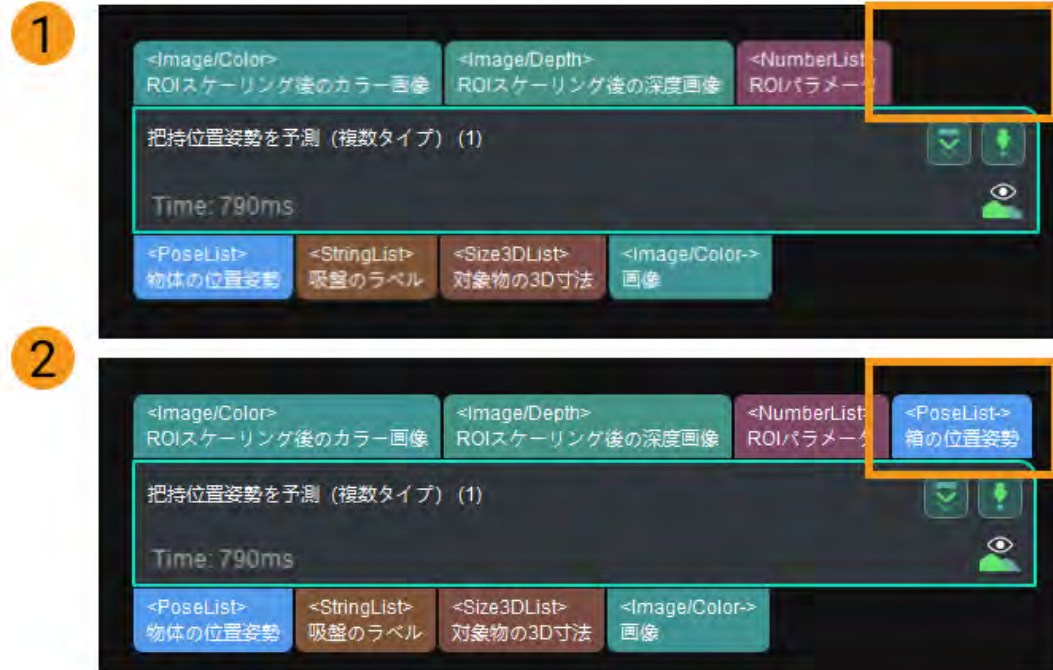
調整説明：カメラからシーン内の物体までの最長距離（ミリメートル単位）です。箱を使用する場合、このパラメータをカメラの底部から箱の下端までの距離に設定してください。実際のニーズに応じて設定してください。

箱の設定

箱を使用

初期値：チェックを入れない。

調整説明：箱が必要な場合はチェックを入れます。チェックを入れると、このステップの入力ポートに「箱の位置姿勢」が表示されます。下図に示すように、1がチェックを入れる前の状態で、2がチェックを入れた後の状態です。



箱の長さ

初期値：100mm

調整説明：パラメータ **箱を使用** にチェックを入れたら、箱の寸法に従ってこの値を設定する必要があります。

箱の幅

初期値：100mm

調整説明：パラメータ **箱を使用** にチェックを入れたら、箱の寸法に従ってこの値を設定する必要があります。

輪郭検出（自動配置と特殊形状部品）

輪郭検出

初期値：チェックを入れない。

調整説明：対象物の輪郭を検出するために使用されます。チェックを入れたら、下に **特殊形状部品のフィルタリング** のパラメータが表示されます。

特殊形状部品のフィルタリング（宅配便小包仕分けに適用可能）

狭すぎる対象物を除去

初期値：チェックを入れない。

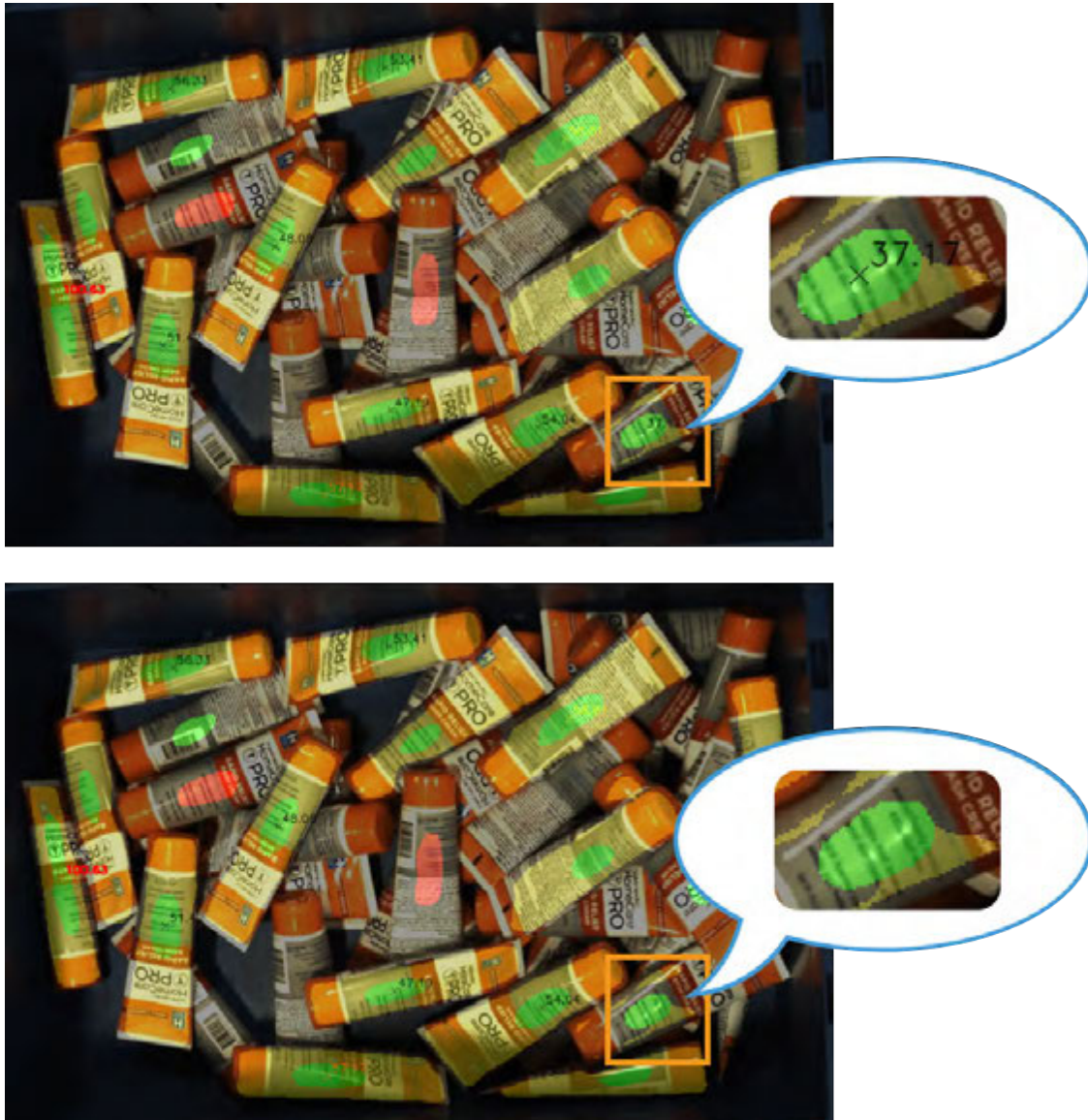
調整説明：チェックを入れた後、最小外接長方形の最短辺の長さが設定された「短辺の最小長さ」よりも小さい対象物は狭すぎる対象物と見なされ、無視されます。

短辺の最小長さ

初期値：0

調整説明：このパラメータは **狭すぎる対象物を除去** にチェックを入れた後に表示されます。最小外接長方形の最短辺の長さがこのしきい値よりも小さい対象物は無視されます。この値は現場の実際の状況に応じて設定する必要があります。

調整の例：この値を30に設定すると、実際の長さが30mm未満の対象物は除去されます。下図に示すように、上が除去前、下が除去後です。



長すぎる対象物を除去

初期値：チェックを入れない。

調整説明：チェックを入れた後、最小外接長方形の最長辺の長さが設定された「長辺の最大長さ」よりも大きい対象物は長すぎる対象物と見なされ、無視されます。

長辺の最大長さ

初期値：0

調整説明：このパラメータは **長すぎる対象物を除去** にチェックを入れた後に表示されます。最小外接長方形の最短辺の長さがこのしきい値よりも大きい対象物は無視されます。この値は現場の実際の状況に応じて設定する必要があります。

調整の例：この値を180に設定すると、実際の長さが180mmを超える対象物は除去されます。下図に示すように、上が除去前、下が除去後です。



積み重ねられた対象物の検出

積み重ねられた対象物の検出

初期値：チェックを入れない。

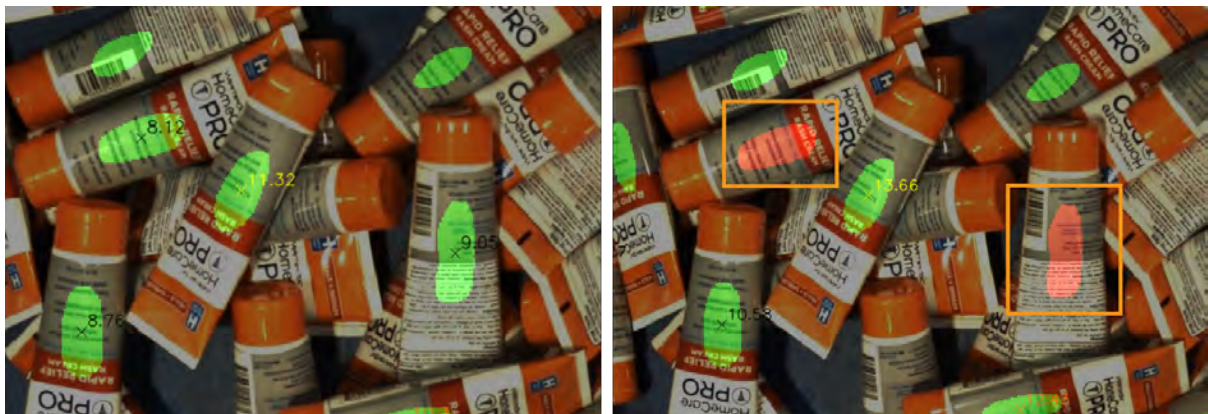
調整説明：積み重ねられた対象物がある場合はチェックを入れます。チェックを入れた後、積み重ねられた対象物が検出され、それらの把持優先度が下げられます。

対象物の最大数

初期値：6

調整説明：**積み重ねられた対象物の検出**にチェックを入れた後に表示されます。これは、積み重ねられた対象物の最大数と見なされます。この値は現場の実際の状況に応じて設定する必要があります。このパラメータがより大きな値に設定されている場合、より多くの対象物が積み重ねられていると見なされ、把持タスクを完了するのが難しくなりますが、対象物を損傷する可能性は低くなります。

調整の例：この値を6に設定すると、最大6つの積み重ねられた対象物が検出されます。下図に示すように、左が積み重ねられた対象物検出前、右が値を6に設定した後の検出結果であり、赤い領域の対象物が検出された積み重ねられた対象物です。



吸盤の設定

このパラメータを調整することで、「把持位置姿勢を予測」ステップの吸盤ラベルの出力が決定されます。



マスクの重さによって割り当てる

初期値：割り当てられない。

オプション：割り当てられない、2つのグループ、3つのグループ。

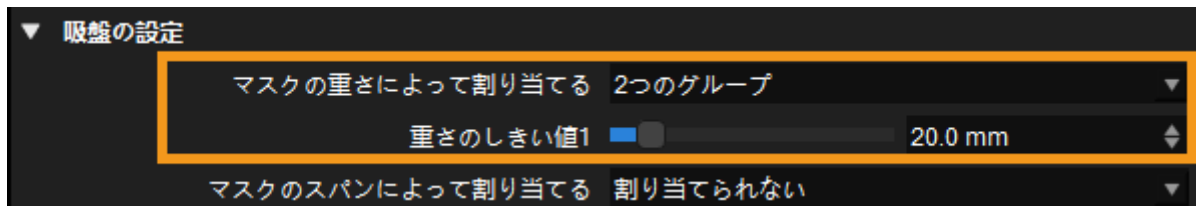
調整説明：対象物はマスクの最小内接円半径によって異なるグループに割り当てられ、各グループの対象物は異なる吸盤設定を使用します。

重さのしきい値1

初期値：0

調整説明：マスクの最小内接円半径の最初のしきい値です。このしきい値より小さい半径を持つ対象物は、最初のグループに割り当てられます。このしきい値より大きい半径を持つ対象物は、2番目のグループに割り当てられます。

調整の例：下図に示すように、対象物マスクの内接円の半径が20mm未満であることを検出した場合、吸盤ラベルが小と判断します。

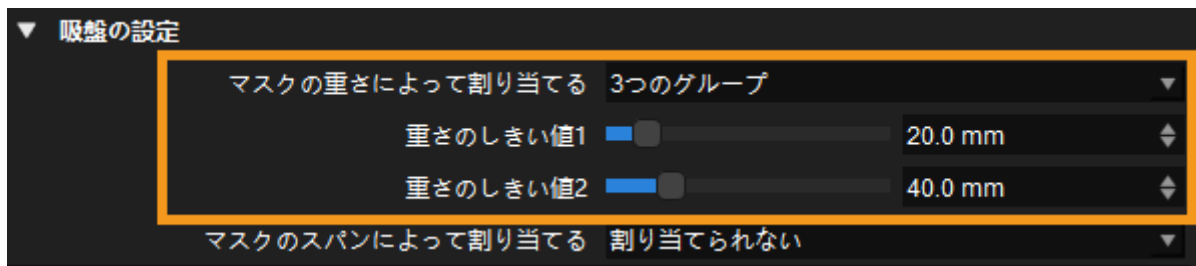


重さのしきい値2

初期値：0

調整説明：マスクの最小内接円半径の2番目のしきい値です。このしきい値より小さい半径を持つ対象物は、2番目のグループに割り当てられます。このしきい値より大きい半径を持つ対象物は、3番目のグループに割り当てられます。

調整の例：下図に示すように、マスクの内接円の半径が40mmを超えることを検出した場合は、吸盤ラベルが大と判断します。マスクの内接円の半径が20mmから40mmの範囲内にあることを検出した場合は、吸盤ラベルが中と判断します。



マスクのスパンによって割り当てる

初期値：割り当てられない。

オプション：割り当てられない、2つのグループ。

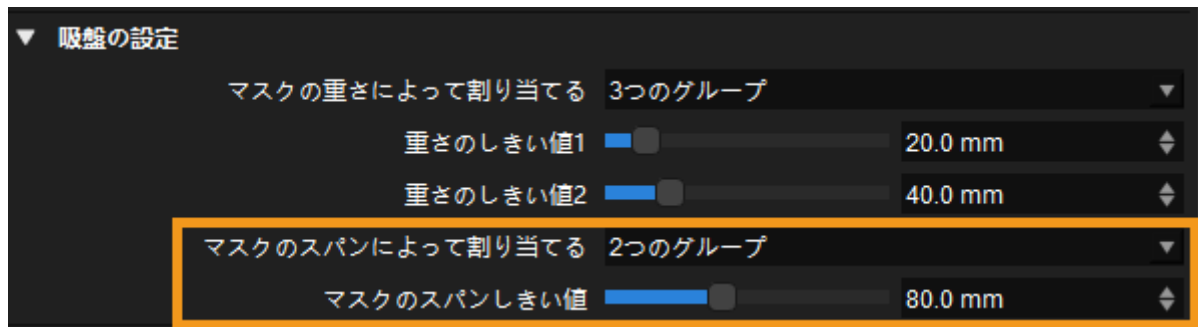
調整説明：対象物はマスクの最小外接長方形の対角線の長さによって異なるグループに割り当てられ、各グループの対象物は異なる吸盤設定を使用します。

マスクのスパンしきい値

初期値：80

調整説明：対象物の最小外接長方形の対角線の長さがこのしきい値より小さい場合は「短」とマークされ、それ以外の場合は「長」とマークされます。

調整の例：下図に示すように、対象物の長さが80mmを超えることを検出した場合は、吸盤ラベルが長と判断し、それ以外は短と判断します。



可視化設定

有効化

初期値：チェックを入れる。

調整説明：チェックを入れると、選択した項目が可視化で表示されます。

可視化のパラメータ

初期値：最後のスコア。

オプション：最後のスコア、吸盤直径、対象物の高さ、位置姿勢の高さ。

調整説明：可視化する項目を選択するために使用されます。

調整の例：

- 最後のスコアを選択した場合の出力結果を下図に示します。赤い文字の対象物は、優先把持対象物です。



- 吸盤直径を選択した場合の出力結果を下図に示します。赤い文字の対象物は、優先把持対象物です。



- 対象物の高さを選択した場合の出力結果を下図に示します。赤い文字の対象物は、優先把持対象物です。



- 位置姿勢の高さを選択した場合の出力結果を下図に示します。赤い文字の対象物は、優先把持対象物です。



位置姿勢のソート論理

位置姿勢高さの重み

初期値：3

オプション：1、2、3

吸盤のサイズの重み

初期値：1

オプション：0、1、2、3

対象物の長さの重み

初期値：1

オプション：0、1、2、3

調整説明：位置姿勢高さ、吸盤のサイズ、対象物の長さの重みを設定した後、最終スコアは位置姿勢高さ×位置姿勢高さの重み+吸盤のサイズ×吸盤のサイズの重み+対象物の長さ×対象物の長さの重みになります。ロボットは、最終スコアに応じて最も高いスコアを持つ対象物を把持します。

4.3.291. 把持位置姿勢を予測（V2）

機能

このステップは2D画像と深度画像から把持対象物を認識し、把持位置姿勢を出力できます。

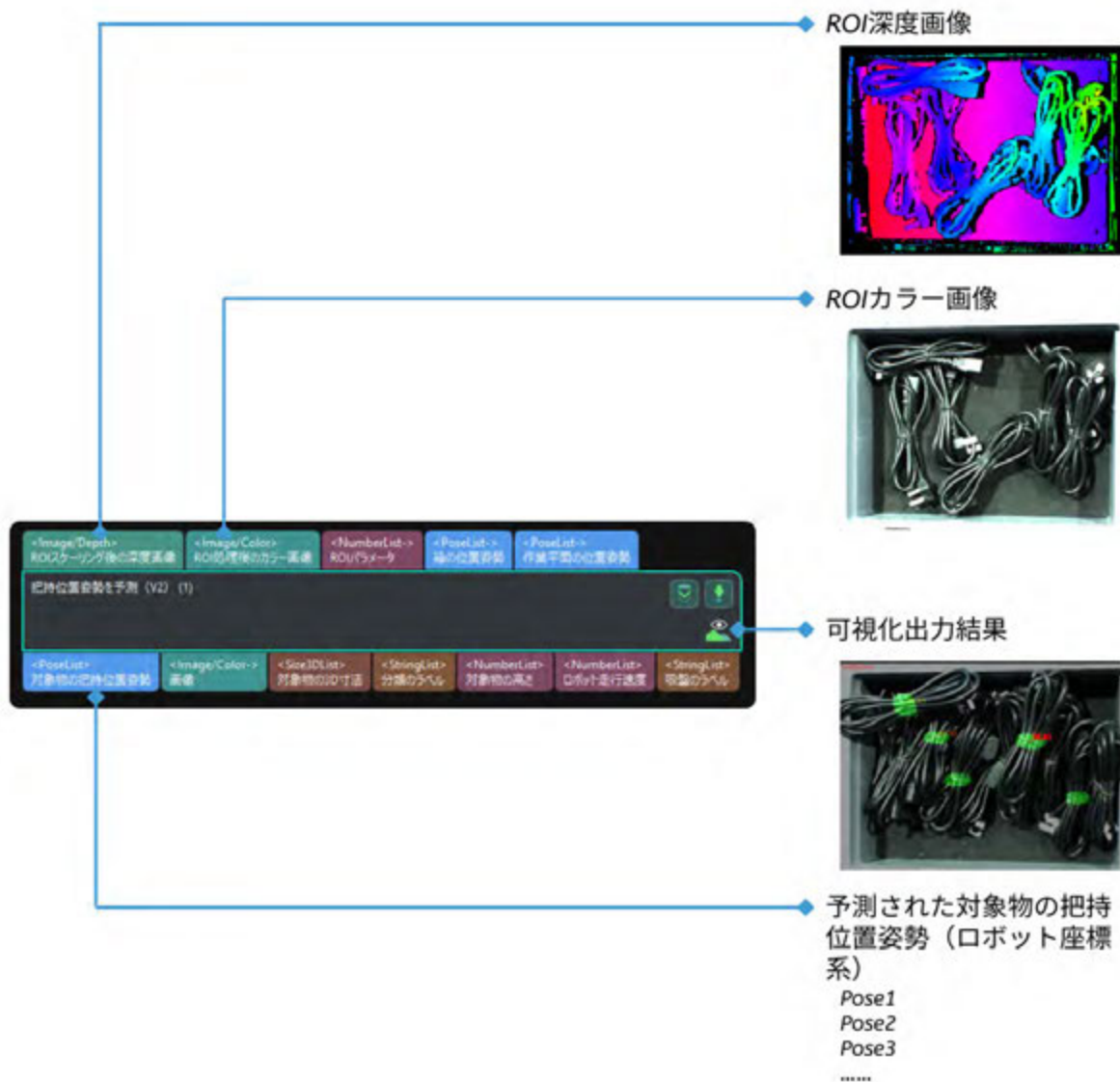


- このステップを使用する際に、ログバーに「ディープラーニングサーバーが正常に起動しました」というメッセージが表示されるのを待ってから、このステップを実行します。
- このステップを初めて実行する前に、**把持の設定ファイル**を追加する必要があります。読み込みに少し時間がかかりますので、しばらくお待ちください。

使用シーン

このステップは商品仕分けに使用され、物流、スーパーマーケット、ケーブルなどの産業で使用されています。スケーリングされた深度画像、点群、ROI情報を取得するために、ステップ [2D ROI内の画像をスケーリング](#) の後に使用されます。

入力と出力



パラメータの説明

サーバー

サーバーIP

パラメータ説明：このパラメータは、ディープラーニングサーバーのIPアドレスを設定するために使用されます。

初期値：127.0.0.1

調整説明：実際のニーズに応じて設定してください。

サーバーポート (1-65535)

パラメータ説明：このパラメータは、ディープラーニングサーバーのポート番号を設定するために使用されます。

初期値：60054

設定可能な範囲：60000~65535

調整説明：実際のニーズに応じて設定してください。

把持設定

把持設定のフォルダパス

パラメータ説明：このパラメータは、把持設定フォルダのパスを設定するために使用されます。

調整説明：プロジェクトを実行する前に、シーンに応じて **把持設定のフォルダ** を追加する必要があります。当社は、物流（セマンティックセグメンテーション）、物流（対象物検出）、スーパーマーケット、ケーブル、薬の外箱の5つの把持設定フォルダを提供しています。詳細は下表に示します。必要な場合はMech-Mind株式会社にお問い合わせください。

使用シーン	把持設定のフォルダ
物流業界（セマンティックセグメンテーション）	Logistics_Seg_RGBSuction
物流業界（対象物検出）	Logistics_OD_RGBSuction
スーパーマーケット業界	Supermarket_Seg_RGBSuction
ケーブル業界	Cable_Seg_RGBGrasp
薬の外箱	MedicineBox_Instance_3DSize_RGBSuction



把持の設定フォルダには、2つのJSONファイルと、1つのモデルが格納されているmodelフォルダがあります。パスを追加するときは、一番外側のフォルダを使用し、modelフォルダに選択する必要がありません。そうしないと、エラーが報告されます。

例えば：xxxx\Cable_Seg_RGBGraspのパスを選択します。



ディープラーニングモデルの選び方を判断するには、ある程度のディープラーニングの知識が必要なので、ご利用の際はMech-Mind株式会社に問い合わせることをお勧めします。

物流業界（セマンティックセグメンテーション）

詳細については、[物流業界（セマンティックセグメンテーション）に適応可能なパラメータ](#)をご参照ください。

物流業界（対象物検出）

詳細については、[物流業界（対象物検出）に適応可能なパラメータ](#)をご参照ください。

スーパーマーケット業界

詳細については、[スーパーマーケットに適応可能なパラメータ](#)をご参照ください。

ケーブル業界

詳細については、[ケーブルに適用可能なパラメータ](#)をご参照ください。

薬の外箱

詳細については、[薬の外箱に適用可能なパラメータ](#)をご参照ください。



上記のシーンでモデルを使用する場合、GeForce GTX 10シリーズを推奨し、ビデオメモリは4G以上が必要です。このステップを初めて実行すると、グラフィックカードに応じてモデルが最適化され、その処理には15分から35分程度かかります。

4.3.291.1. 物流業界（セマンティックセグメンテーション）に適用可能なパラメータ

使用する把持設定ファイルは **Logistics_Seg_RGBSuction** となります。これにより、画像内の対象物のすべてのピクセルを取得し、対象物の輪郭と状態（積み重ねられているかどうか）を取得できます。

調整ロジック

1. **稼働距離** を設定して、カメラがデータを取得する範囲を制限します。
2. **輪郭検出（自動配置と特殊形状部品）** グループのパラメータを調整して、対象物の輪郭を検出します。
3. 物流のシーンでは、対象物が一般的にバラ積みされています。**積み重ねられた対象物の検出** パラメータによって、積み重ねられたか対象物と積み重ねられない対象物に分類されます。
4. **位置姿勢処理** グループのパラメータを調整して、対象物の損傷を防ぐために、ロボットがZ軸に沿って上方にオフセットする距離を制御します。
5. **動的配置** グループのパラメータを調整して、ロボットの動作速度を調整します。
6. **優先把持の設定** と **位置姿勢のソート論理** グループのパラメータを調整し、把持順序を調整します。

パラメータの調整説明

稼働距離

最小稼働距離

パラメータ説明：このパラメータは、カメラからシーン内の対象物までの最小距離（ミリメートル単位）を指定するために使用されます。箱を使用する場合は、このパラメータをカメラ底面から箱の上面までの距離に設定してください。実際の状況に応じて設定してください。

初期値：0mm

設定可能な範囲：0~3000mm

最大稼働距離

パラメータ説明：このパラメータは、カメラからシーン内の対象物までの最大距離（ミリメートル単位）を指定するために使用されます。箱を使用する場合は、このパラメータをカメラ

ラ底面から箱の底面までの距離に設定してください。実際の状況に応じて設定してください。

初期値：3000mm

設定可能な範囲：0~3000mm

輪郭検出（自動配置と特殊形状部品）

外側輪郭が優先

パラメータ説明：このパラメータは、検出された対象物の外側輪郭を検出するために使用されます。対象物の輪郭が整っているほど把持の優先度は高く、輪郭が整っていないほど把持の優先度は低くなります。

初期値：チェックを入れる

調整アドバイス：重なり検出に対して高い要件がない場合、把持率を向上させ、把持時に頻繁な箱交換を避けるために、このパラメータにチェックを入れることを推奨します。重なり検出に対して高い要件がある場合、積み重ねられていない対象物のみを把持するために、このパラメータチェックを入れないことを推奨します。

積み重ねられた対象物の検出

対象物の最大数

パラメータ説明：このパラメータは、積み重ねられた状態と判定された対象物の最大数を設定するために使用されます。

初期値：6

設定可能な範囲：0~10

調整アドバイス：実際の状況に応じて設定してください。

位置姿勢処理

出力位置姿勢の最大数

パラメータ説明：このパラメータは、出力される位置姿勢の最大数を指定するために使用されます。

設定可能な範囲：0~20

Z軸に沿って上にオフセット

パラメータ説明：このパラメータは、把持中にロボットが対象物を押しつぶさないようにするために、把持位置姿勢からZ軸方向上方への距離を設定するために使用されます。単位はミリメートル（mm）です。

初期値：0

設定可能な範囲：0~10mm

調整アドバイス：実際の状況に応じて設定してください。

動的配置

このパラメータ群は、次の2つの状況を回避するために、対象物の高さに応じて対象物を配置するときに、ロボットハンドの高さを決定できます。

- ロボットが対象物を配置するときの位置が高すぎるため、高速で走行するコンベア上で対象物が転がってしまうのです。
- ロボットが対象物を配置するときの位置が低すぎるため、対象物がコンベアに衝突し、対象物が圧迫されて破損したり、コンベアが傷ついたりすることがあります。

ロボットが対象物を把持する際には、対象物の重量を確認する必要があります。このパラメータに関連するアルゴリズムは、対象物の体積を計算することで対象物の重量を推定します。ロボットが重い対象物を把持する場合、ロボットの速度が速すぎると、次のような問題が発生する可能性があります。

- 対象物が把持されていません。
- 把持された対象物は捨てられます。
- 対象物の表面が破れていましたが、対象物は把持されていません。

2D面積しきい値

パラメータ説明：このパラメータは、対象物の面積を検出するために使用されます。検出された対象物の面積がこの値より大きい場合、ロボットは配置時に速度を落とします。検出された物体の面積がこの値よりも小さい場合、ロボットは設定された **対象物を配置するときの速度** で対象物を配置します。

設定可能な範囲：0~10000000 px

対象物を配置するときの速度

パラメータ説明：このパラメータは、ロボットが対象物を配置する速度を設定するために使用されます。検出された対象物の面積が **2D面積しきい値** より小さい場合、ロボットは **対象物を配置するときの速度** で対象物を配置します。検出された物体の面積が **2D面積しきい値** よりも小さい場合、ロボットは配置時に速度を落とします。

初期値：100%

設定可能な範囲：0~100%

優先把持の設定

優先把持の方向

パラメータ説明：このパラメータは、優先に対象物を把持するときの方向を設定するために

使用されます。

オプション：

- 上
- 下
- 左
- 右

初期値：右

位置姿勢のソート論理

このステップは認識された位置姿勢を、位置姿勢スコアに従って降順にソートします。位置姿勢スコア計算式：位置姿勢スコア = 対象物高さ × 対象物高さの重み + 対象物の水平位置 × 対象物の水平位置の重み + 長方形性 × 長方形性の重み。

位置姿勢高さの重み

パラメータ説明：このパラメータは、対象物を把持するときの位置姿勢高さの重みを設定するために使用されます。高い対象物を優先的に把持します。この重みによって、ソートにおける割合が調整されます。

初期値：5

設定可能な範囲：0~5

対象物の水平位置の重み

パラメータ説明：このパラメータは、対象物を把持するときの対象物の水平位置の重みを設定するために使用されます。**優先把持の方向**の境界に近い対象物が優先に把持されます。この重みによって、ソートにおける割合が調整されます。

初期値：1

設定可能な範囲：0~5

長方形性の重み

パラメータ説明：このパラメータは、対象物を把持するときの長方形の重みを設定するために使用されます。長方形性の高い対象物を優先的に把持します。この重みによって、ソートにおける割合が調整されます。

初期値：0

設定可能な範囲：0~5

可視化設定



このパラメータは、データの表示のみに使用されます。

有効化

パラメータ説明：チェックを入れると、**デバッグ結果出力** 画面で選択した **可視化のパラメータ** の結果を確認できます。

初期値：チェックを入れる。

可視化のパラメータ

パラメータ説明：このパラメータは、可視化する項目を指定するために使用されます。

オプション：

- 最終スコア
- 位置姿勢の高さ
- 吸盤直径
- 対象物の長さ

初期値：最終スコア



このパラメータは、**有効化** にチェックを入れた場合にのみ設定できます。

その他の説明

長方形性

検出された対象物と長方形の間の類似度を指します。計算方法は対象物の面積を最小外接長方形（下図のオレンジ色の長方形）で割るというもので、範囲は0～1です。



4.3.291.2. 物流業界（対象物検出）に適応可能なパラメータ

使用する把持設定ファイルは **Logistics_OD_RGBSuction** となります。これにより、画像内の対象物の位置、サイズ、および状態（積み重ねられたかどうか）のみを予測します。対象物検出はピクセルを検出しないため、外側の輪郭を取得できません。

調整ロジック

1. **稼働距離** を設定して、カメラがデータを取得する範囲を制限します。

2. 物流のシーンでは、対象物が一般的にバラ積みされています。**積み重ねられた対象物の検出**によって、積み重ねられた対象物と積み重ねられていない対象物に分類されます。
3. **特殊形状部品のフィルタリング** グループのパラメータを調整して、細すぎるや大きすぎる対象物を除去し、把持できない可能性を減らします。
4. **位置姿勢処理** グループのパラメータを調整して、対象物の損傷を防ぐために、ロボットがZ軸に沿って上方にオフセットする距離を制御します。
5. **動的配置** グループのパラメータを調整して、ロボットの動作速度を調整します。
6. **優先把持の設定と位置姿勢のソート論理** グループのパラメータを調整して、把持順序を調整します。

パラメータの調整説明

稼働距離

最小稼働距離

パラメータ説明：このパラメータは、カメラからシーン内の対象物までの最小距離（ミリメートル単位）を指定するために使用されます。箱を使用する場合は、このパラメータをカメラ底面から箱の上面までの距離に設定してください。実際の状況に応じて設定してください。

初期値：0mm

設定可能な範囲：0~3000mm

最大稼働距離

パラメータ説明：このパラメータは、カメラからシーン内の対象物までの最大距離（ミリメートル単位）を指定するために使用されます。箱を使用する場合は、このパラメータをカメラ底面から箱の底面までの距離に設定してください。実際の状況に応じて設定してください。

初期値：3000mm

設定可能な範囲：0~3000mm

積み重ねられた対象物の検出

物流における一般的な対象物には、箱、袋、書類袋（封筒）などがあります。把持対象物が積み重ねられた状態かどうかを判断するために、箱状物、袋状物、書類袋の長方形性のしきい値を設定する必要があります。

対象物の最大数

パラメータ説明：このパラメータは、積み重ねられた状態と判定された対象物の最大数を設定するために使用されます。

初期値：6

設定可能な範囲：0~10

箱形対象物の長方形性しきい値

パラメータ説明：このパラメータは、箱形対象物の長方形性のしきい値を設定するために使用されます。箱形対象物の長方形性のしきい値がこの値より大きい場合、積み重ねられていない対象物と見なされます。この値より小さい場合、積み重ねられた対象物と見なされません。

初期値：10%

推奨値：60%

設定可能な範囲：0~100%

封筒形対象物の長方形性しきい値

パラメータ説明：このパラメータは、封筒形対象物の長方形性のしきい値を設定するために使用されます。封筒形対象物の長方形性のしきい値がこの値より大きい場合、積み重ねられていない対象物と見なされます。この値より小さい場合、積み重ねられた対象物と見なされます。

初期値：10%

推奨値：75%

設定可能な範囲：0~100%

袋状対象物の長方形性しきい値

パラメータ説明：このパラメータは、袋状対象物の長方形性のしきい値を設定するために使用されます。袋状対象物の長方形性のしきい値がこの値より大きい場合、積み重ねられていない対象物と見なされます。この値より小さい場合、積み重ねられた対象物と見なされません。

初期値：10%

推奨値：70%

設定可能な範囲：0~100%

特殊形状部品のフィルタリング

このパラメータ群は、細すぎたり大きすぎたりして、把持できない可能性を減らすために対象物をフィルタリングするために使用されます。

短辺の最小長さ

パラメータ説明：このパラメータは、検出される短辺の長さの最小値（ミリメートル単位）を設定するために使用されます。検出された対象物の短辺の長さがこの値より小さい場合、除去されます。

初期値：0mm

設定可能な範囲：0~1500mm

調整アドバイス：実際の状況に応じて設定してください。

長辺の最大長さ

パラメータ説明：このパラメータは、検出される長辺の長さの最大値（ミリメートル単位）を設定するために使用されます。検出された対象物の長辺の長さがこの値より大きい場合、除去されます。

初期値：1500mm

設定可能な範囲：0~1500mm

調整アドバイス：実際の状況に応じて設定してください。

位置姿勢処理

出力位置姿勢の最大数

パラメータ説明：このパラメータは、位置姿勢の出力最大数を指定するために使用されます。

設定可能な範囲：0~20

Z軸に沿って上にオフセット

パラメータ説明：このパラメータは、把持中にロボットが対象物を押しつぶさないようにするために、把持位置姿勢からZ軸方向上方への距離を設定するために使用されます。単位はミリメートル（mm）です。

初期値：0

設定可能な範囲：0~10mm

調整アドバイス：実際の状況に応じて設定してください。

動的配置

このパラメータ群は、次の2つの状況を回避するために、対象物の高さに応じて対象物を配置するときに、ロボットハンドの高さを決定できます。

- ロボットが対象物を配置するときの位置が高すぎるため、高速で走行するコンベア上で対象物が転がってしまうのです。
- ロボットが対象物を配置するときの位置が低すぎるため、対象物がコンベアに衝突し、対象物が圧迫されて破損したり、コンベアが傷ついたりすることがあります。

ロボットが対象物を把持する際には、対象物の重量を確認する必要があります。このパラメー

タに関連するアルゴリズムは、対象物の体積を計算することで対象物の重量を推定します。ロボットが重い対象物を把持する場合、ロボットの速度が速すぎると、次のような問題が発生する可能性があります。

- 対象物が把持されていません。
- 把持された対象物は捨てられます。
- 対象物の表面が破れていましたが、対象物は把持されていません。

2D面積しきい値

パラメータ説明：このパラメータは、対象物の面積を検出するために使用されます。検出された対象物の面積がこの値より大きい場合、ロボットは配置時に速度を落とします。検出された物体の面積がこの値よりも小さい場合、ロボットは設定された **対象物を配置するときの速度** で対象物を配置します。

設定可能な範囲：0~10000000 px

対象物を配置するときの速度

パラメータ説明：このパラメータは、ロボットが対象物を配置するときの速度を設定するために使用されます。検出された対象物の面積が **2D面積しきい値** より小さい場合、ロボットは**対象物を配置するときの速度**で対象物を配置します。検出された物体の面積が**2D面積しきい値**よりも大きい場合、ロボットは対象物を配置する時に速度を落とします。

初期値：100%

設定可能な範囲：0~100%

優先把持の設定

優先把持の方向

パラメータ説明：このパラメータは、優先に対象物を把持するときの方向を設定するために使用されます。

オプション：

- 上
- 下
- 左
- 右

初期値：右

位置姿勢のソート論理

このステップは認識された位置姿勢を、位置姿勢スコアに従って降順にソートします。位置姿勢スコア計算式：位置姿勢スコア = 対象物高さ × 対象物高さの重み + 対象物の水平位置 × 対象

物の水平位置の重み + 長方形性 × 長方形性の重み。

位置姿勢高さの重み

パラメータ説明：このパラメータは、対象物を把持するときの位置姿勢高さの重みを設定するために使用されます。高い対象物を優先的に把持します。この重みによって、ソートにおける割合が調整されます。

初期値：5

設定可能な範囲：0~5

対象物の水平位置の重み

パラメータ説明：このパラメータは、対象物を把持するときの対象物の水平位置の重みを設定するために使用されます。**優先把持の方向**の境界に近い対象物が優先に把持されます。この重みによって、ソートにおける割合が調整されます。

初期値：1

設定可能な範囲：0~5

長方形性の重み

パラメータ説明：このパラメータは、対象物を把持するときの長方形の重みを設定するために使用されます。長方形性の高い対象物を優先的に把持します。この重みによって、ソートにおける割合が調整されます。

初期値：0

設定可能な範囲：0~5

可視化設定



このパラメータは、データの表示のみに使用されます。

有効化

パラメータ説明：チェックを入れると、**デバッグ結果出力**画面で選択した**可視化のパラメータ**の結果を確認できます。

初期値：チェックを入れる。

可視化のパラメータ

パラメータ説明：このパラメータは、可視化する項目を指定するために使用されます。

オプション：

- 最終スコア
- 位置姿勢の高さ

- 吸盤直径
- 対象物の長さ

初期値：最終スコア



このパラメータは、**有効化**にチェックを入れた場合にのみ設定できます。

その他の説明

長方形性

検出された対象物と長方形の間の類似度を指します。計算方法は対象物の面積を最小外接長方形（下図のオレンジ色の長方形）で割るというもので、範囲は0～1です。



4.3.291.3. スーパーマーケットに適応可能なパラメータ

使用する把持設定ファイルは **Supermarket_Seg_RGBSuction** となります。読み込み後に調整可能なパラメータが表示されます。

このモデルを使用する場合、GeForce GTX 10シリーズを推奨し、ビデオメモリは4G以上が必要です。このステップを初めて実行すると、グラフィックカードに応じてモデルが最適化され、その処理には15分から35分程度かかります。

調整ロジック

1. **稼働距離** を設定して、カメラがデータを取得する範囲を制限します。
2. **輪郭検出（自動配置と特殊形状部品）** グループのパラメータを調整して、輪郭外の予測点に把持を行うかどうかを選択します。
3. スーパーマーケットのシーンでは、対象物が一般的にバラ積みされています。**積み重ねられた対象物の検出**によって、積み重ねられた対象物と積み重ねられていない対象物に分類されます。
4. **位置姿勢処理** グループのパラメータを調整して、対象物の損傷を防ぐために、ロボットがZ軸に沿って上方にオフセットする距離を制御します。
5. **吸盤の設定** グループのパラメータを調整し、吸盤をグループ化します。
6. **位置姿勢のソート論理** グループのパラメータを調整し、把持順序を調整します。

パラメータの調整説明

稼働距離

最小稼働距離

パラメータ説明：このパラメータは、カメラからシーン内の対象物までの最小距離（ミリメートル単位）を指定するために使用されます。箱を使用する場合は、このパラメータをカメラ底面から箱の上面までの距離に設定してください。実際の状況に応じて設定してください。

初期値：0mm

設定可能な範囲：0~3000mm

最大稼働距離

パラメータ説明：このパラメータは、カメラからシーン内の対象物までの最大距離（ミリメートル単位）を指定するために使用されます。箱を使用する場合は、このパラメータをカメラ底面から箱の底面までの距離に設定してください。実際の状況に応じて設定してください。

初期値：3000mm

設定可能な範囲：0~3000mm

輪郭検出（自動配置と特殊形状部品）

外側輪郭が優先

パラメータ説明：このパラメータは、検出された対象物の外側輪郭を検出するために使用されます。対象物の輪郭が整っているほど把持の優先度は高く、輪郭が整っていないほど把持の優先度は低くなります。

初期値：チェックを入れる

調整アドバイス：重なり検出に対して高い要件がない場合、把持率を向上させ、把持時に頻繁な箱交換を避けるために、このパラメータにチェックを入れることを推奨します。重なり検出に対して高い要件がある場合、積み重ねられていない対象物のみを把持するために、このパラメータチェックを入れないことを推奨します。

積み重ねられた対象物の検出

対象物の最大数

パラメータ説明：このパラメータは、把持する積み重ねられた対象物の最大数を指定するために使用されます。

初期値：6

設定可能な範囲：0~10

調整アドバイス：実際の状況に応じて設定してください。

位置姿勢処理

Z軸に沿って上にオフセット

パラメータ説明：このパラメータは、把持中にロボットが対象物を押しつぶさないようにするために、把持位置姿勢からZ軸方向上方への距離を設定するために使用されます。単位はミリメートル（mm）です。

初期値：0

設定可能な範囲：0~10mm

調整アドバイス：実際の状況に応じて設定してください。

吸盤の設定

マスクの重さによって割り当てる

パラメータ説明：このパラメータは、重さによって対象物をグループ化するために使用されます。吸盤設定はグループによって異なります。

オプション：

- 割り当てられない：初期値となります。グループ分けは行わず、1つのグループのみがあります。
- 2つのグループ：パラメータ **重さのしきい値1** によって、対象物を2つのグループに分けます。
- 3つのグループ：パラメータ **重さのしきい値1** と **重さのしきい値2** によって、対象物を3つのグループに分けます。

重さのしきい値1/重さのしきい値2

パラメータ説明：マスクをグループ化するための重さのしきい値で、単位はミリメートルです。

初期値：0mm

設定可能な範囲：0~200mm



- マスクの重さによって割り当てる が 2つのグループ の場合、 **重さのしきい値1** を設定できます。詳細は以下の通りです。
 - マスクの重さ < 重さのしきい値1の場合、1番目のグループに割り当てられません。
 - マスクの重さ > 重さのしきい値1の場合、2番目のグループに割り当てられません。
- マスクの重さによって割り当てる が 3つのグループ の場合、 **重さのしきい値1** と **重さのしきい値2** を設定できます。また、 **重さのしきい値2** は **重さのしきい値1** より大きい必要があります。詳細は以下の通りです。

- マスクの重さ < 重さのしきい値1の場合、1番目のグループに割り当てられます。
- 重さのしきい値1 < マスクの重さ < 重さのしきい値2の場合、2番目のグループに割り当てられます。
- 重さのしきい値2 < マスクの重さの場合、3番目のグループに割り当てられます。

マスクのスパンによって割り当てる

パラメータ説明：このパラメータは、スパンによって対象物をグループ化するために使用されます。吸盤設定はグループによって異なります。

オプション：

- 割り当てられない：初期値となります。グループ分けは行わず、1つのグループのみがあります。
- 2つのグループ：パラメータ **マスクのスパンしきい値** によって、マスクを2つのグループに分けます。

マスクのスパンしきい値

パラメータ説明：マスクをグループ化するためのスパンしきい値で、単位はミリメートルです。

初期値：0mm

設定可能な範囲：0~200mm



このパラメータは、**マスクのスパンによって割り当てる** が **2つのグループ** の場合にのみ設定できます。詳細は以下の通りです。

- マスクのスパン < マスクのスパンしきい値の場合、1番目のグループに割り当てられます。
- マスクのスパン > マスクのスパンしきい値の場合、2番目のグループに割り当てられます。

位置姿勢のソート論理

認識された位置姿勢は、位置姿勢スコアに従って降順にソートされます。位置姿勢スコアの計算公式は、 $\text{位置姿勢スコア} = \text{位置姿勢高さ} \times \text{位置姿勢高さの重み} + \text{吸盤サイズ} \times \text{吸盤サイズの重み} + \text{対象物長さ} \times \text{対象物長さの重み}$ となります。

位置姿勢高さの重み

パラメータ説明：このパラメータは、対象物を把持するときの位置姿勢高さの重みを設定するために使用されます。高い対象物を優先的に把持します。この重みによって、ソートにおける割合が調整されます。

初期値：3

設定可能な範囲：0~5

吸盤のサイズの重み

パラメータ説明：このパラメータは、対象物を把持するときの吸盤サイズの重みを設定するために使用されます。大きな吸盤で把持する必要のある対象物を優先的に把持します。この重みによってソートにおける割合が調整されます。

初期値：1

設定可能な範囲：0~5

対象物の長さの重み

パラメータ説明：このパラメータは、対象物を把持するときの対象物長さの重みを設定するために使用されます。長い対象物を優先的に把持します。この重みによって、ソートにおける割合が調整されます。

初期値：1

設定可能な範囲：0~5

可視化設定



このパラメータは、データの表示のみに使用されます。

有効化

パラメータ説明：チェックを入れると、**デバッグ結果出力**画面で選択した**可視化のパラメータ**の結果を確認できます。

初期値：チェックを入れる。

可視化のパラメータ

パラメータ説明：このパラメータは、可視化する項目を指定するために使用されます。

オプション：

- 最終スコア
- 位置姿勢の高さ
- 吸盤直径
- 対象物の長さ

初期値：最終スコア



このパラメータは、**有効化**にチェックを入れた場合にのみ設定できます。

4.3.291.4. ケーブルに適用可能なパラメータ

使用する把持設定ファイルは **Cable_Seg_RGBGrasp** となります。読み込み後に調整可能なパラメータが表示されます。

調整ロジック

1. **稼働距離** を設定して、カメラがデータを取得する範囲を制限します。
2. **位置姿勢処理** グループのパラメータを調整して、対象物の損傷を防ぐために、ロボットがZ軸に沿って上方にオフセットする距離を制御します。

パラメータの調整説明

稼働距離

最小稼働距離

パラメータ説明：このパラメータは、カメラからシーン内の対象物までの最小距離（ミリメートル単位）を指定するために使用されます。箱を使用する場合は、このパラメータをカメラ底面から箱の上面までの距離に設定してください。実際の状況に応じて設定してください。

初期値：0mm

設定可能な範囲：0~3000mm

最大稼働距離

パラメータ説明：このパラメータは、カメラからシーン内の対象物までの最大距離（ミリメートル単位）を指定するために使用されます。箱を使用する場合は、このパラメータをカメラ底面から箱の底面までの距離に設定してください。実際の状況に応じて設定してください。

初期値：3000mm

設定可能な範囲：0~3000mm

位置姿勢処理

Z軸に沿って上にオフセット

パラメータ説明：このパラメータは、把持中にロボットが対象物を押しつぶさないようにするために、把持位置姿勢からZ軸方向上方への距離を設定するために使用されます。単位はミリメートル（mm）です。

初期値：0

設定可能な範囲：0~10mm

調整アドバイス：実際の状況に応じて設定してください。

可視化設定



このパラメータは、データの表示のみに使用されます。

有効化

パラメータ説明：チェックを入れると、**デバッグ結果出力** 画面で選択した **可視化のパラメータ** の結果を確認できます。

初期値：チェックを入れる。

可視化のパラメータ

パラメータ説明：このパラメータは、可視化する項目を指定するために使用されます。

オプション：

- 最終スコア
- 位置姿勢の高さ

初期値：最終スコア

4.3.291.5. 薬の外箱に適応可能なパラメータ

使用する把持設定ファイルは **MedicineBox_Instance_3DSize_RGBSuction** となります。読み込み後に調整可能なパラメータが表示されます。

このモデルを使用する場合、GeForce GTX 10シリーズを推奨し、ビデオメモリは4G以上が必要です。このステップを初めて実行すると、グラフィックカードに応じてモデルが最適化され、その処理には15分から35分程度かかります。

調整ロジック

1. **把持設定フォルダのパス** を設定します。
2. **稼働距離** を設定して、カメラがデータを取得する範囲を制限します。
3. **箱の設定** のグループのパラメータを調整し、シーンにおける箱の寸法を設定します。
4. 対象物の損傷を防ぐために、**位置姿勢処理** グループのパラメータを調整し、ロボットがZ軸に沿って上方にオフセットする距離を設定します。
5. 対象物の点群の数を減らし、処理速度を向上させるために、**点群ダウンサンプリング** パラメータを調整します。
6. 最後に **平面分割** パラメータ調整を行います。平面分割のしきい値を設定することで、対象物の平面点群を分割します。

パラメータの調整説明

把持設定

把持設定のフォルダパス

パラメータ説明：このパラメータは、把持設定フォルダのパスを設定するために使用されます。

調整説明：このパラメータの右側のパス選択アイコンをクリックし、把持設定フォルダのパスを選択します。

稼働距離

最小稼働距離

パラメータ説明：このパラメータは、カメラからシーン内の対象物までの最小距離（ミリメートル単位）を指定するために使用されます。箱を使用する場合は、このパラメータをカメラ底面から箱の上面までの距離に設定してください。実際の状況に応じて設定してください。

初期値：0mm

設定可能な範囲：0~3000mm

最大稼働距離

パラメータ説明：このパラメータは、カメラからシーン内の対象物までの最大距離（ミリメートル単位）を指定するために使用されます。箱を使用する場合は、このパラメータをカメラ底面から箱の底面までの距離に設定してください。実際の状況に応じて設定してください。

初期値：3000mm

設定可能な範囲：0~3000mm

箱の設定

箱の長さ

パラメータ説明：このパラメータは、箱の長さ（ミリメートル単位）を設定するために使用されます。

箱の幅

パラメータ説明：このパラメータは、箱の幅（ミリメートル単位）を設定するために使用されます。

位置姿勢処理

出力位置姿勢の最大数

パラメータ説明：このパラメータは、Mech-Visionが出力できる位置姿勢の最大数を設定するために使用されます。

初期値：15

Z軸に沿って上にオフセット

パラメータ説明：このパラメータは、把持中にロボットが対象物を押しつぶさないようにするために、把持位置姿勢からZ軸方向上方への距離を設定するために使用されます。単位はミリメートル（mm）です。

初期値：0

設定可能な範囲：0~10mm

調整アドバイス：実際の状況に応じて設定してください。

位置姿勢たわみ角のしきい値

パラメータ説明：対象物の位置姿勢のX軸またはY軸のたわみがこのしきい値より大きい場合、その対象物の位置姿勢は除去されます。単位は度（°）です。

初期値：45.000°

点群ダウンサンプリング

点群ダウンサンプリング

パラメータ説明：このパラメータは、対象物の点群の数を減らし、処理速度を向上させるために使用されます。この値が大きいほど、対象物の点群が少なく、処理速度が上がります。

初期値：0.0010

平面分割

平面分割のしきい値

パラメータ説明：このパラメータは、対象物の平面点群を分割するときのしきい値を設定するために使用されます。値が小さいほど、得られる点群の数は少ないが、分割精度は高くなります。値が大きいほど、得られる点群の数は多いが、分割精度は低くなります。

初期値：0.0020

可視化設定

有効化

パラメータ説明：このパラメータは、このステップの出力結果を可視化で表示するかどうかを設定するために使用されます。

初期値：チェックを入れる。

4.3.292. 出力

機能

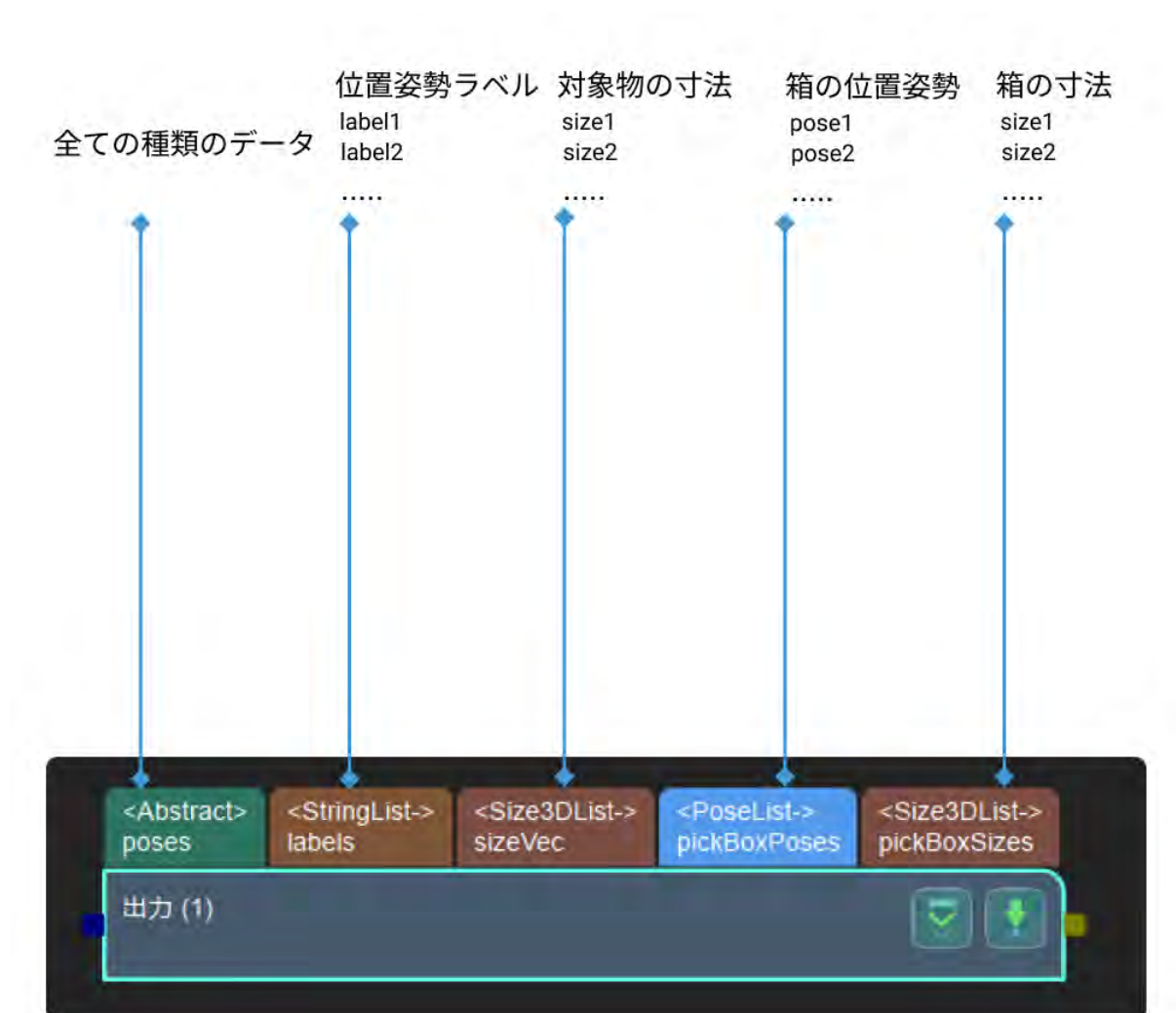
現在のプロジェクト結果をバックグラウンドに送信します。

使用シーン

ビジョンプロジェクトの結果を、サービスを介してMech-VizまたはMech-Centerに送信します。

入力と出力

プロジェクトに経路計画関連のステップが含まれていない場合、入力と出力は以下のようになります。



パラメータの説明

ポートタイプ

入力ポートのタイプを選択するために使用されます。

オプション：

- カスタム：実際の状況に応じて設定する必要があります。
- 事前定義済み（ビジョン結果）：初期値となります。プロジェクトに経路計画に関連するステップが含まれていない場合に使用します。
- 事前定義済み（ロボット経路）：プロジェクトに経路計画に関連するステップが含まれている場合に使用します。

カスタム

パラメータ説明：外部通信へ出力されるデータタイプを設定するために使用されます。

設定手順：

1. **[外部通信への出力データタイプを設定]** をクリックし、**出力設定アシスタント** の画面に入ります。
2. **事前定義された通信キー** または **カスタマイズの通信キー** でポートを追加します。
 - 事前定義された通信キー：必要なポートにチェックを入れます。
 - カスタマイズの通信キー：空白場所をクリックし、下部にある **[追加]** をクリックしてポート名を入力します。



カスタマイズの通信キーの名前を変更しないと、ポートが有効になりません。

3. **[確認]** をクリックすると、設定が完了します。

4.3.293. ステップの組合せ

機能

複数のステップの集まりです。

使用シーン

通常、複数のステップをステップの組合せにして、より複雑な機能を実現します。これにより、プロジェクトがより簡潔になり、ロジックと機能が分かりやすくなります。

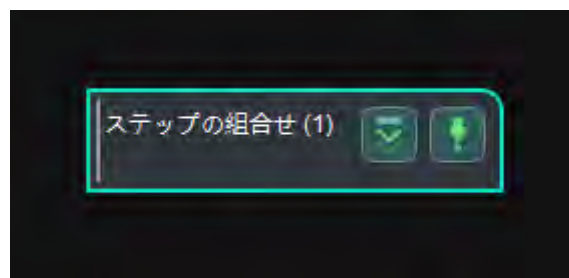


Figure 2. id5

複数のステップを「ステップの組合せ」として配置することができます。ダブルクリックまたは右クリックで「詳細を表示」を選択してこの「ステップの組合せ」の中に入り、必要なステップを「ステップの組合せ」にドラッグしてつなげます。

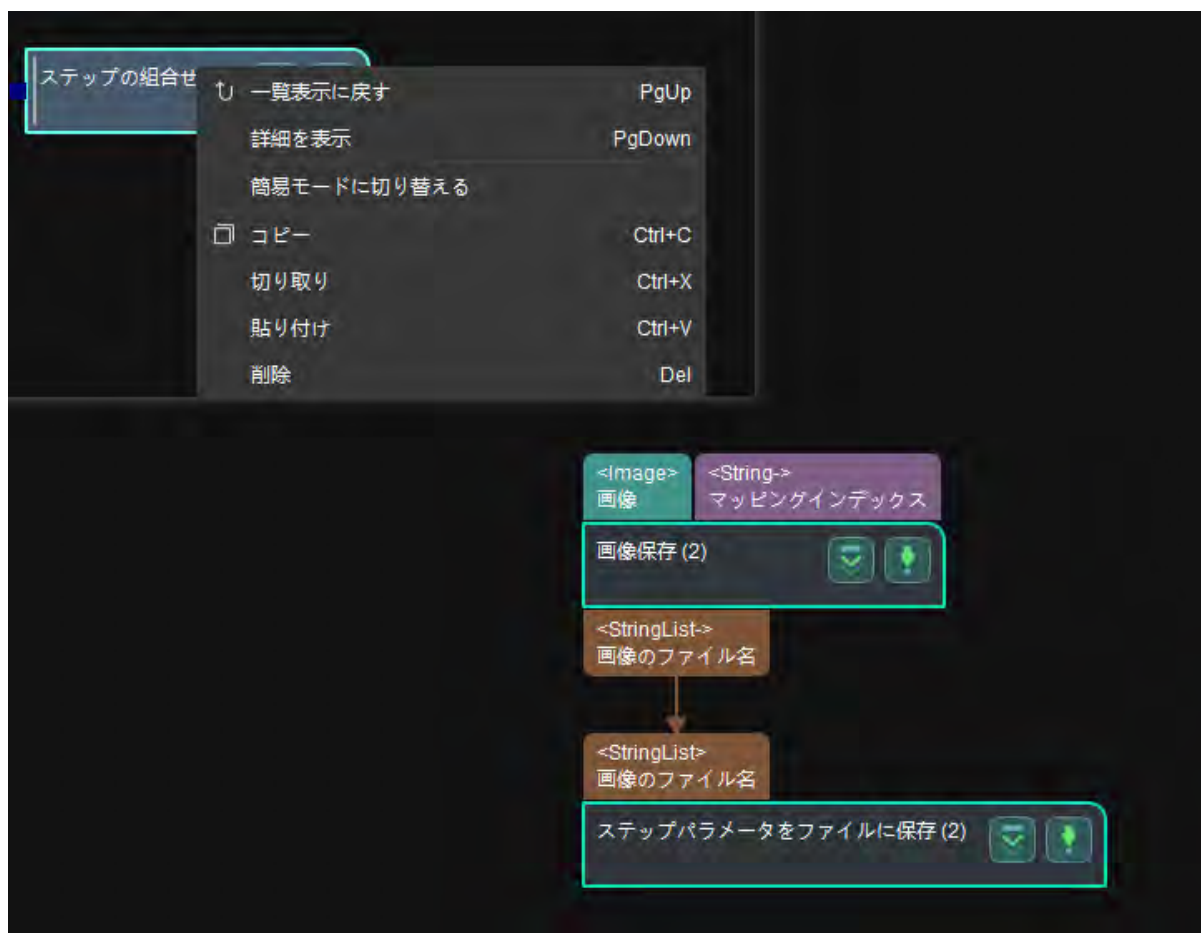


Figure 3. pictt2

右クリックで「一覧表示に戻す」を選択してこの「ステップの組合せ」を終了することができます。

「ステップの組合せ」の入力と出力を設定

「ステップの組合せ」を作成した後、その入力と出力を設定する必要があります。ステップの入力または出力をダブルクリックして、このステップの入力または出力を「ステップの組合せ」の入力および出力として使用できます。入力と出力の順序はダブルクリックによって異なります。設定方法と設定後の「ステップの組合せ」を下図に示します。

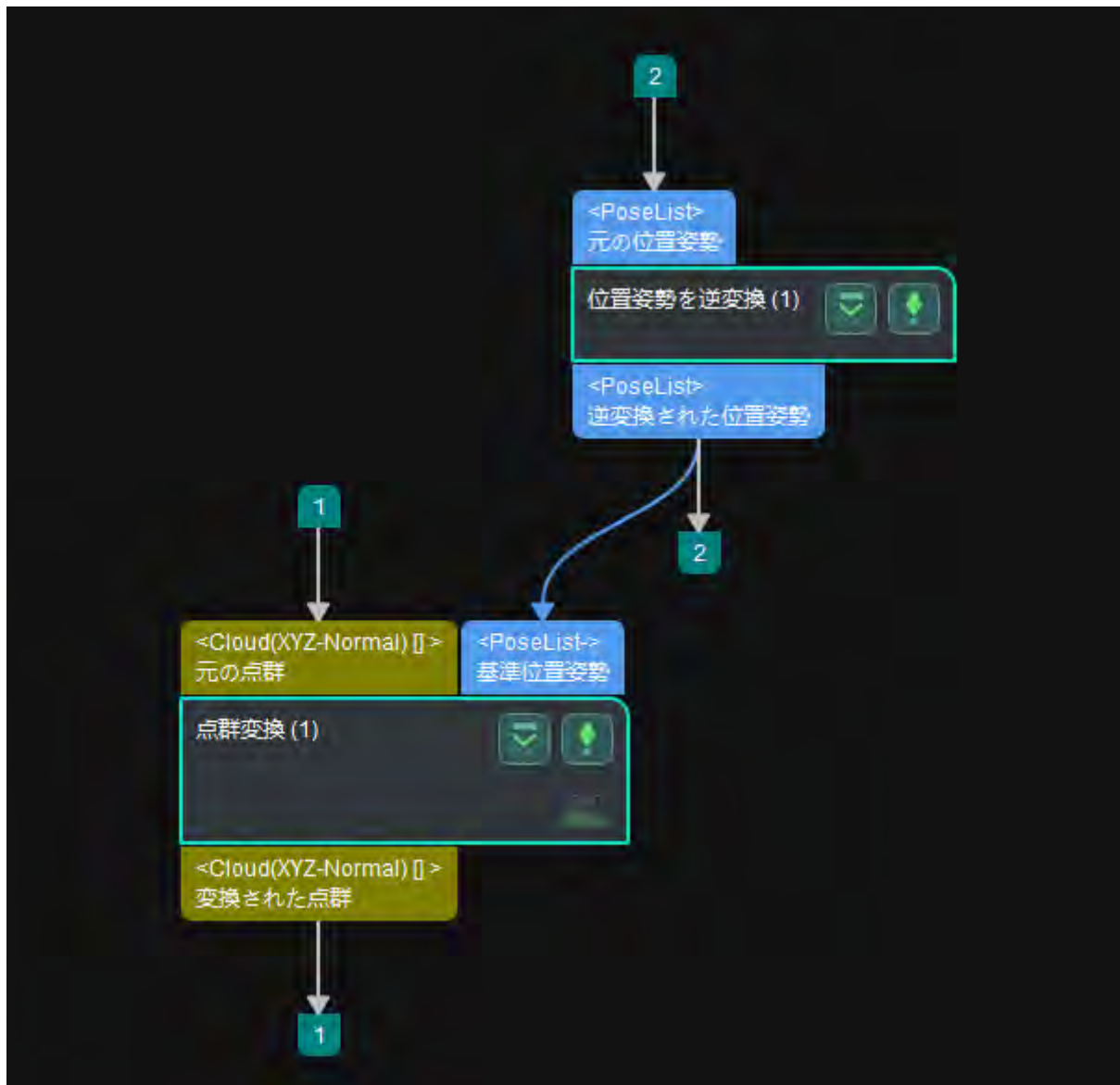


Figure 4. pic3

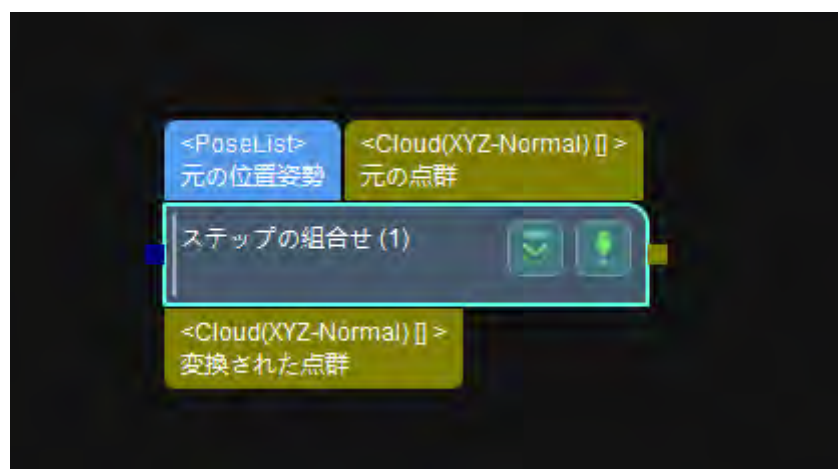


Figure 5. pic4

「ステップの組合せ」のパラメータを設定

「ステップの組合せ」のパラメータ調整を容易にするために、「ステップの組合せ」の各ステップの主要パラメータを「ステップの組合せ」のパラメータとして選択できます。「ステップの組合せ」を選択して右クリックし、「ステップの組合せのパラメータを編集」をクリックします。表示された画面で表示するパラメータを選択してから、「OK」をクリックします。設定方法と効果を下図に示します。

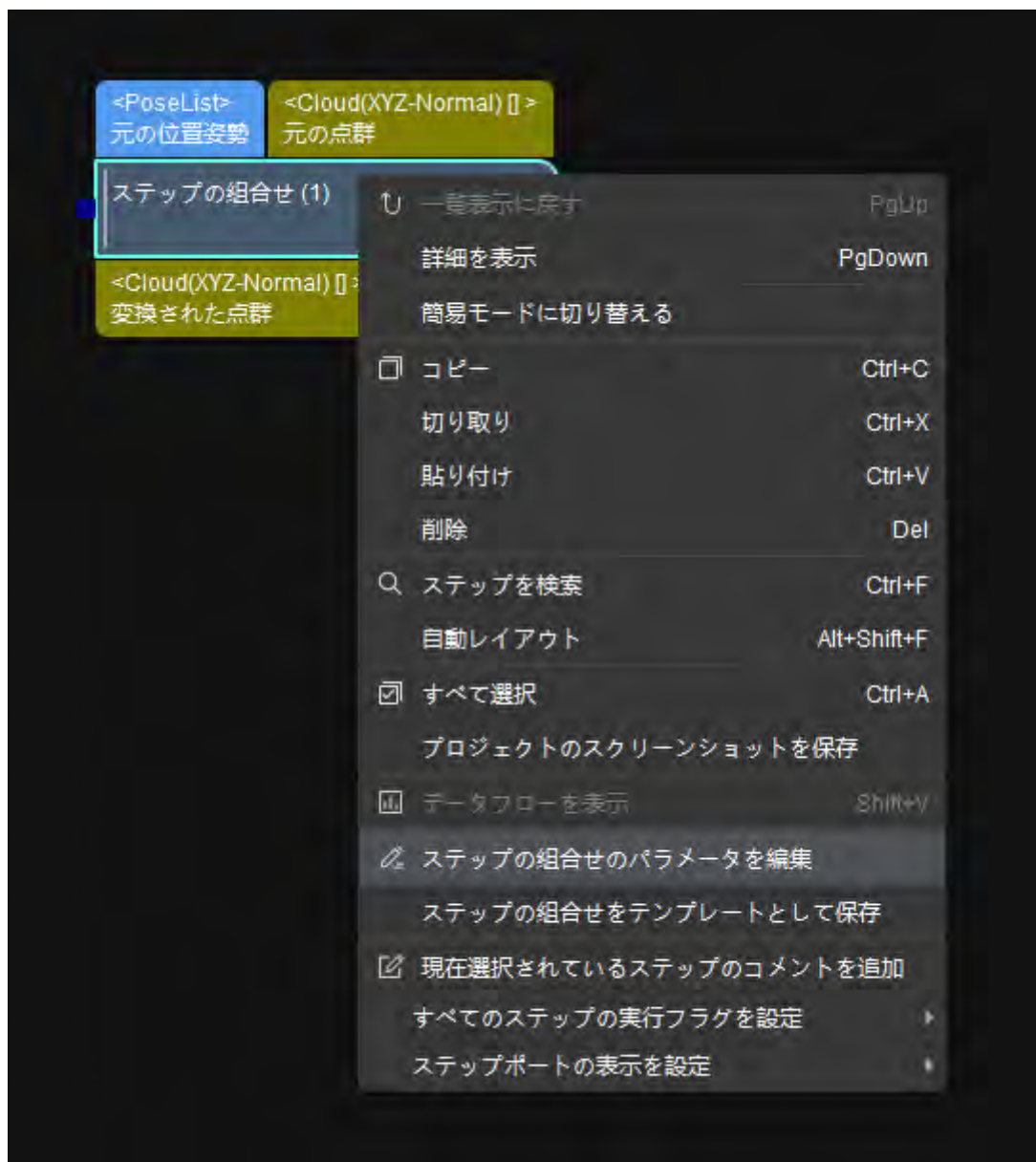


Figure 6. pic5

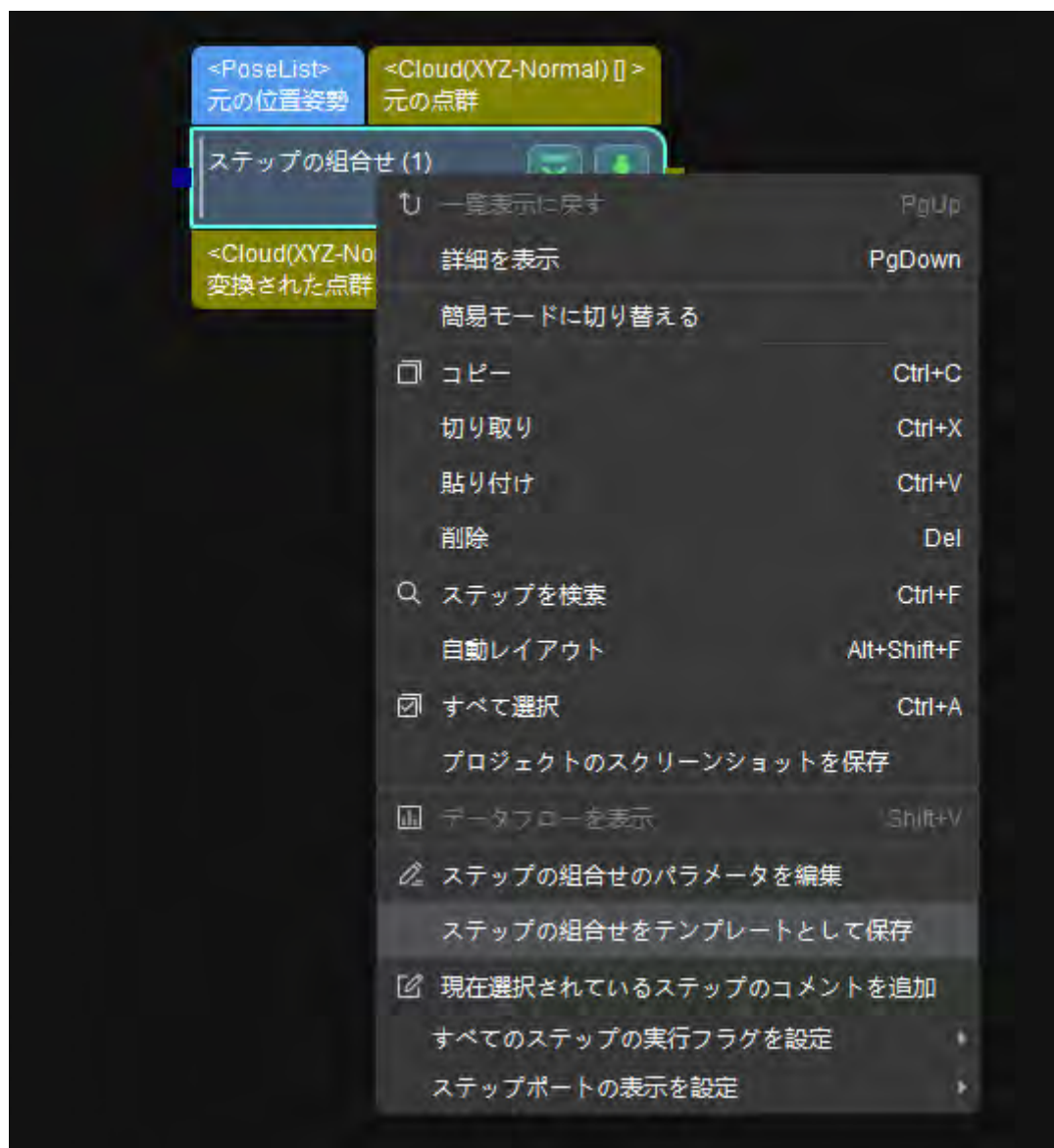


Figure 7. pic6



Figure 8. pic7

「ステップの組合せ」の保存と呼び出し

後で再利用するために、「ステップの組合せ」をテンプレートとして保存できます。「ステッ

「ステップの組合せ」を選択して右クリックし、「ステップの組合せをテンプレートとして保存」をクリックします。表示された画面で英語と日本語の名前をそれぞれ入力します。すると、「ステップの組合せ」のテンプレートが「カスタマイズ」グループの下に保存されます。

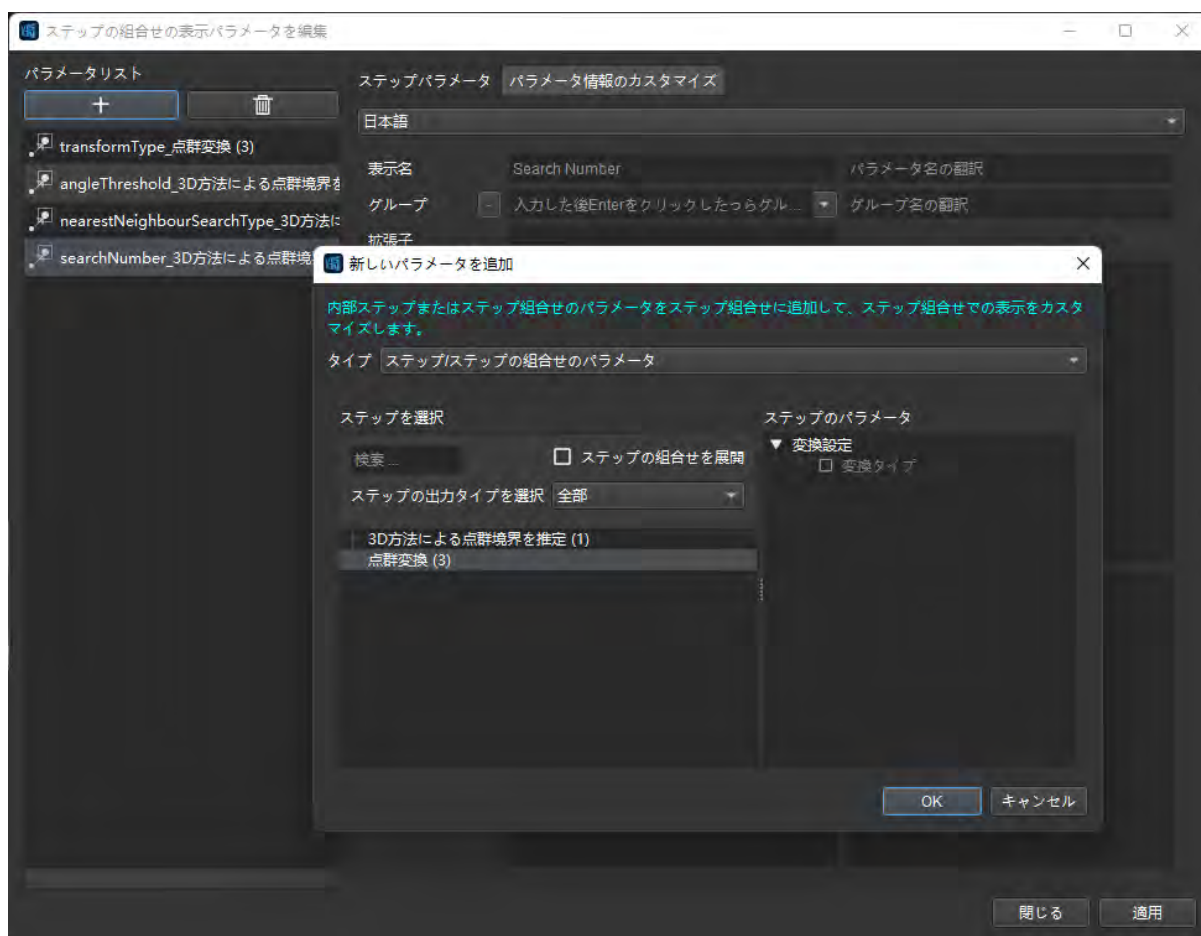


Figure 9. pic8

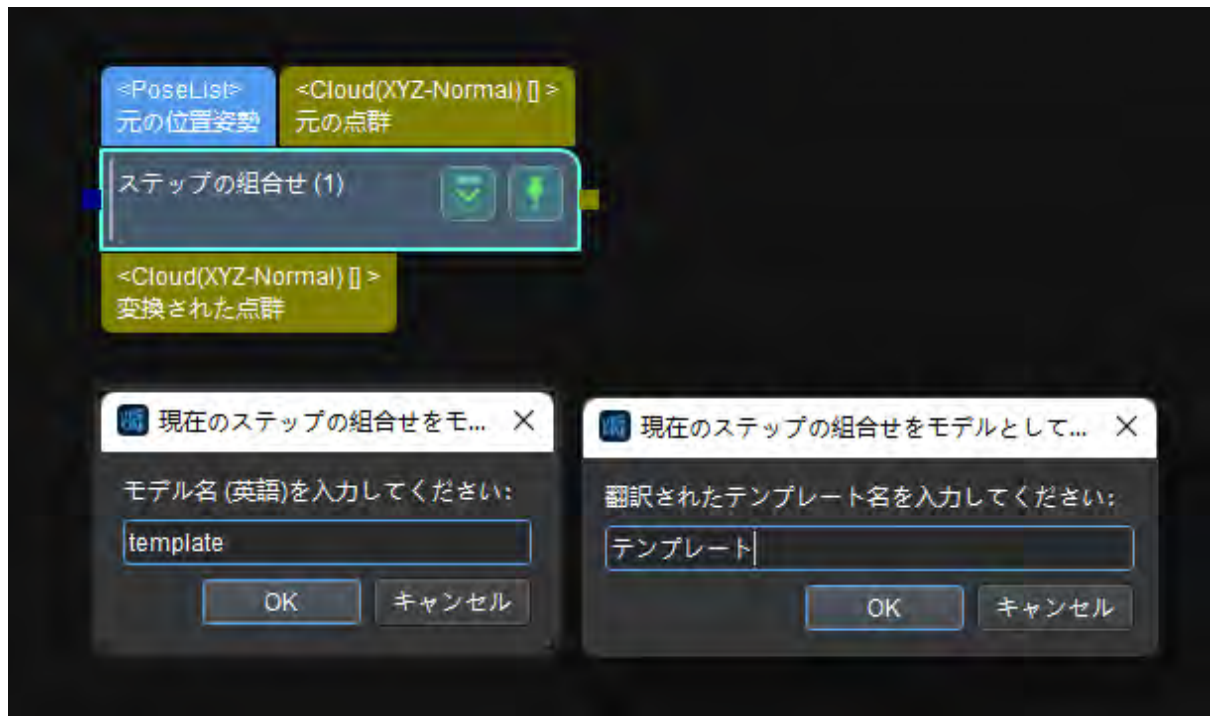


Figure 10. pic9



Figure 11. pic10

4.3.294. 2D形状を処理

機能

入力された2値画像内の形状を指定方法に従って処理します。

使用シーン

通常、計測シーンでの様々な計算を容易にするために、対象物の輪郭を処理するために使用されます。

入力と出力

このステップでは、**Ellipse** 方法を例に、入力と出力は下図のようになります。



パラメータの説明

対象形状

パラメータ説明：このパラメータは、2D形状の処理方法を選択するために使用されます。

オプション：Convex、Ellipse、Circumcircle、Incircle、Bounding box、Circumscribed rectangle、Inscribed rectangle、Inner center

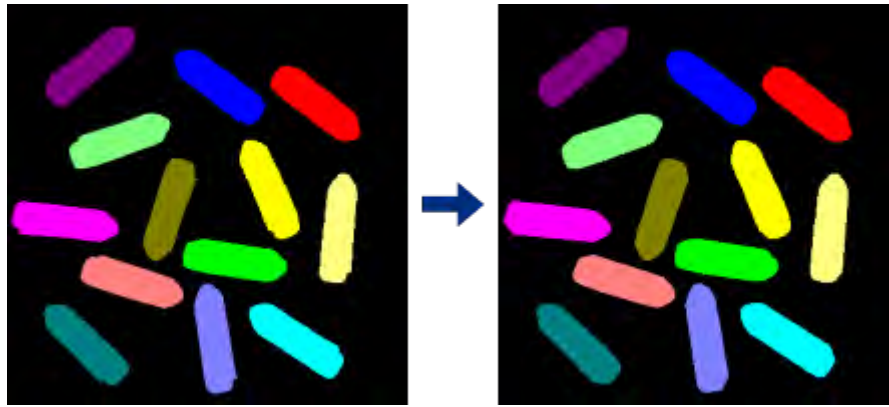
- Convex：元の形状の凸包を生成します。
- Ellipse：元の形状と同じ1次および2次モーメントを持つ楕円を生成します。
- Circumcircle：元の形状の最小内接円を生成します。
- Incircle：元の形状の最大内接円を生成します。
- BoundingBox：画像のエッジに平行な辺を持つ、元の形状の最小の境界ボックスを生成します。
- CircumscribedRectangle：元の形状の最小外接長方形を生成します。
- InscribedRectangle：画像のエッジに平行な辺を持つ、元の形状の最大内接長方形を生成します。
- InnerCenter：元の形状で重心に最も近いポイントを生成します。

調整アドバイス：実際のニーズに応じて選択してください。

調整の例

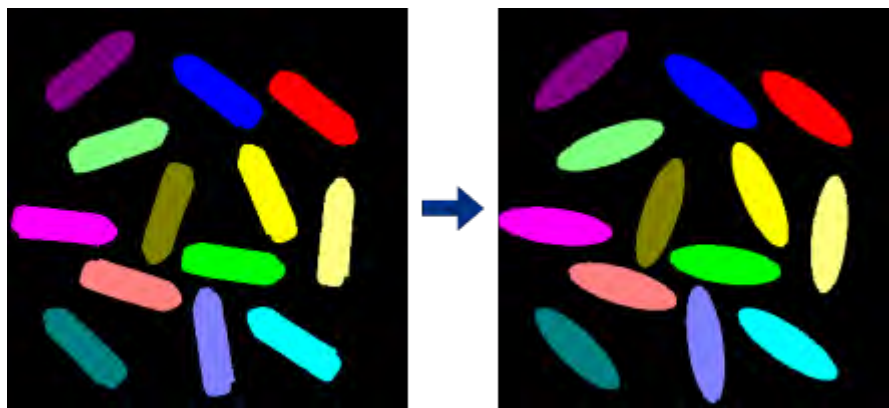
Convex

処理方法が **Convex** の場合、このステップの結果は下図の右側に表示されます。



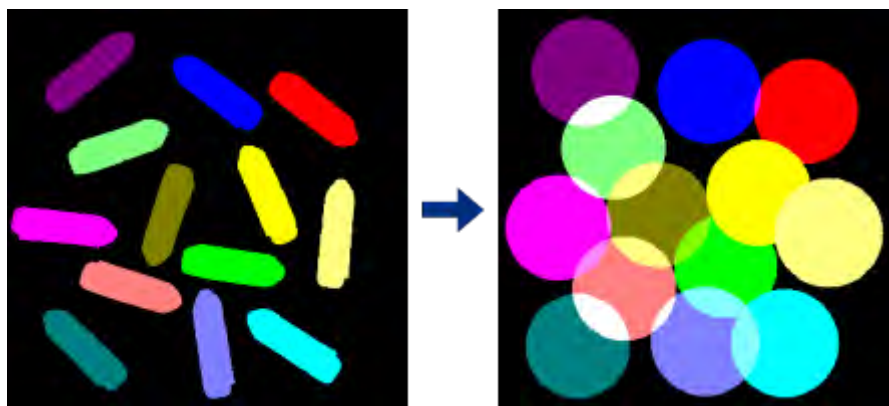
Ellipse

処理方法が **Ellipse** の場合、このステップの結果は下図の右側に表示されます。



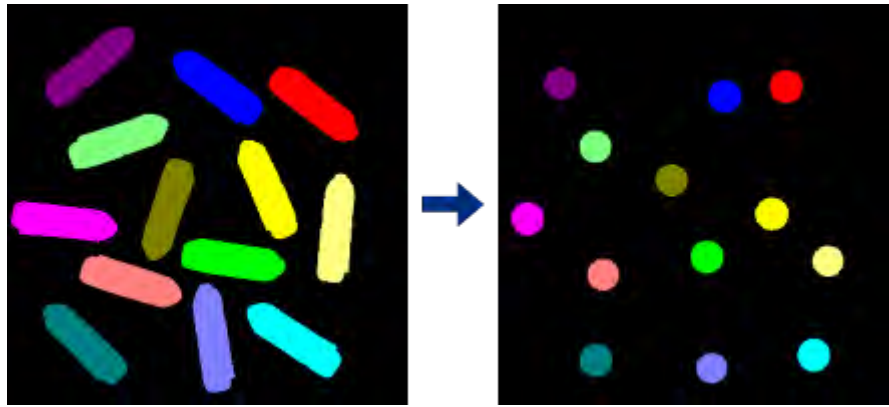
Circumcircle

処理方法が **Circumcircle** の場合、このステップの結果は下図の右側に表示されます。



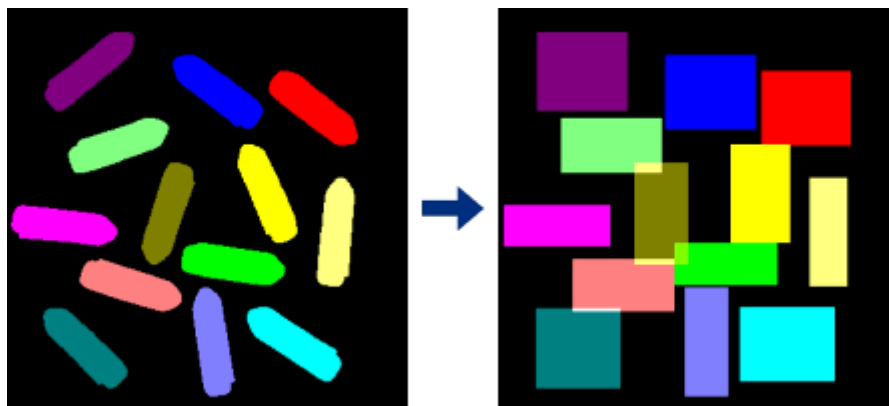
Incircle

処理方法が **Incircle** の場合、このステップの結果は下図の右側に表示されます。



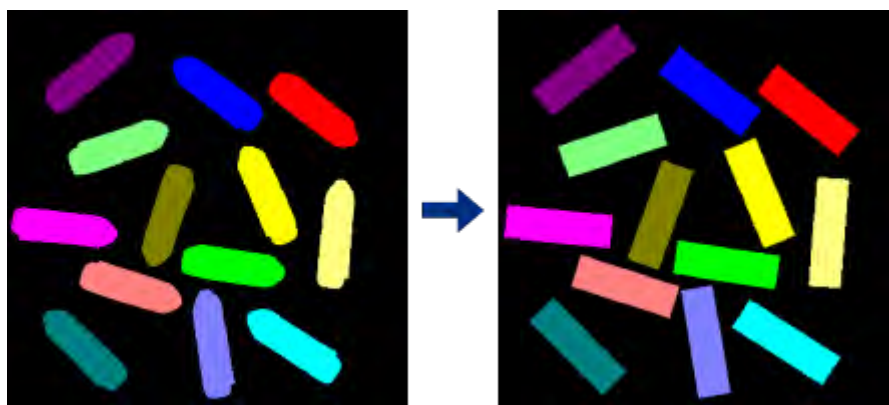
Bounding box

処理方法が **Bounding box** の場合、このステップの結果は下図の右側に表示されます。



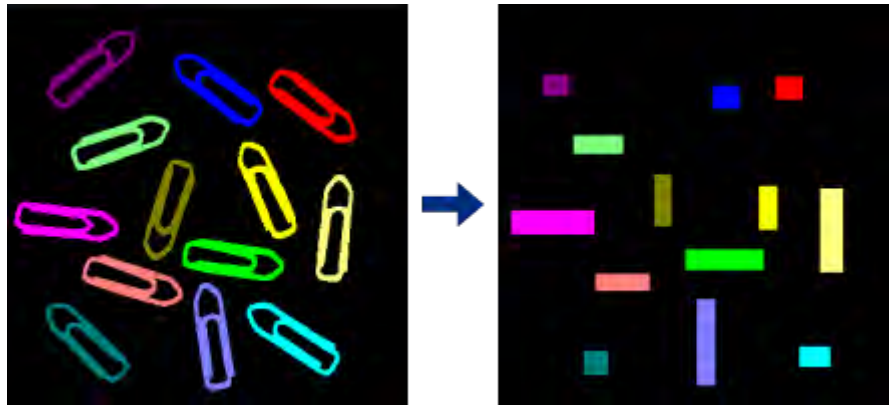
Circumscribed rectangle

処理方法が **Circumscribed rectangle** の場合、このステップの結果は下図の右側に表示されます。



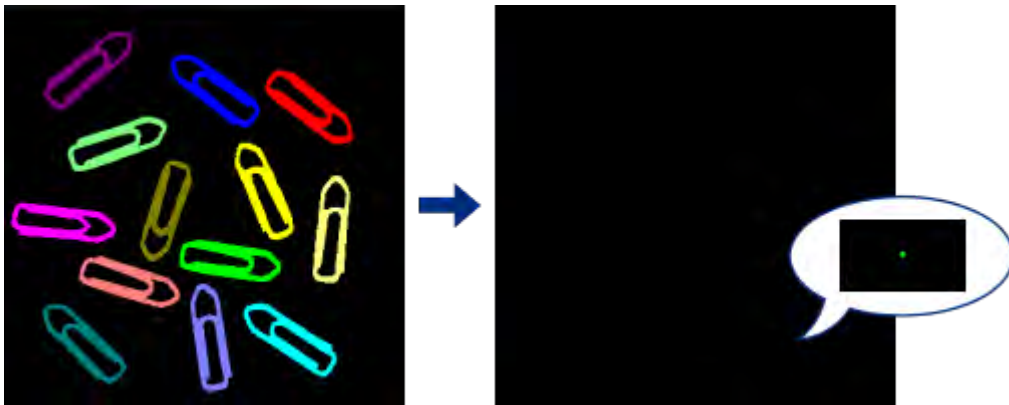
Inscribed rectangle

処理方法が **Inscribed rectangle** の場合、このステップの結果は下図の右側に表示されます。



Inner center

処理方法が **Inner center** の場合、このステップの結果は下図の右側に表示されます。



4.3.295. 点を平面に投影



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

点を、入力位置姿勢にある平面に投影します。

使用シーン

特定のシーン向けのカスタマイズのステップです。

入力と出力

● 入力：

1. このポートに入力された位置姿勢は平面を定義するために使用されます。
2. このポートに入力された位置姿勢は点を定義するために使用され、点は平面に投影されます。

● 出力：

1. 平面に投影された点。

4.3.296. 3D点群を2D画像に投影

機能

指定された投影タイプに従って点群を2D画像に変換します。

使用シーン

このステップは点群前処理に使用されます。まずは3D点群を2D画像に投影し、2D画像処理方法（2Dモフォロジー処理など）を使用した後に [マスク内の対応する点群を抽出](#) で点群に戻します。

把持シーンでは、**透視投影** の投影タイプを推奨します。

計測シーンでは、**正投影** の投影タイプを推奨します。このステップを [正投影](#) に置き換えることを推奨します。

入力と出力



パラメータの説明

投影のタイプ

パラメータ説明：このパラメータは、点群から2D画像への変換に使用する投影タイプを指定

するために使用されます。

オプション：

- **透視投影**：把持のシーンに適しています。
- **正投影**：計測のシーンに適しています。

初期値：透視投影

透視投影

画像のタイプ

パラメータ説明：このパラメータは、出力される2D画像のタイプを指定するために使用されます。

オプション：

- **カラー画像**：カラー画像を出力します。
- **深度画像**：深度画像を出力します。

初期値：カラー画像

膨張処理カーネルサイズ

パラメータ説明：このパラメータは、膨張処理カーネルサイズ（ピクセル単位）を設定するために使用されます。膨張処理カーネルは、対象物の面積を増やしたり、対象物の損傷した部分をつなげたりするために使用されます。値が大きいくほど、より多くの領域が埋められ、値が小さいほど、より少ない領域しか埋められません。

初期値：21px

調整の例：カーネルサイズを大きくした場合の効果を下図に示します（カーネルは1～3まで大きくなる）。



収縮処理カーネルサイズ

パラメータ説明：このパラメータは、収縮処理カーネルサイズ（ピクセル単位）を設定するために使用されます。収縮処理カーネルは、小さなノイズを除去し、つながった対象物を分離するために使用されます。値が大きいくほど、より多くの領域が除去され、値が小さいほど、より少ない領域が除去されます。

初期値：21px

調整の例：カーネルサイズを大きくした場合の効果を下図に示します（カーネルは1～3まで大きくなる）。



正投影

投影設定

対象物のZ軸に沿って投影

パラメータ説明：このパラメータは、点群をZ軸に沿って投影するかどうかを設定するために使用されます。

初期値：チェックを入れる

スケーリング

パラメータ説明：このパラメータは、投影後の画像のスケーリングを設定するために使用されます。

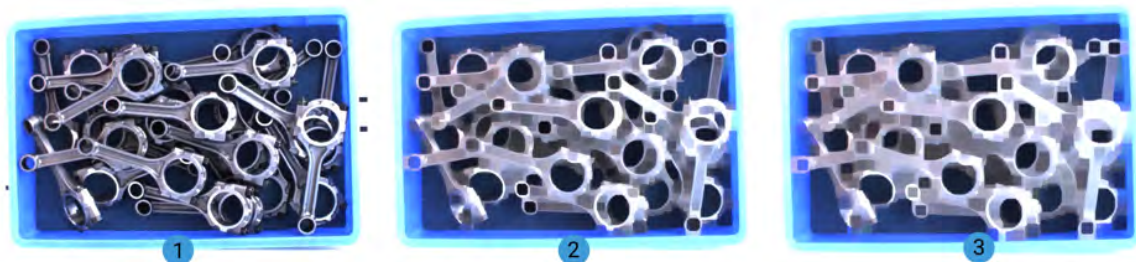
初期値：1000（1ピクセルは1mm）

クロージング処理のカーネルサイズ

パラメータ説明：このパラメータは、クロージング処理のカーネルサイズ（ピクセル単位）を設定するために使用されます。対象物の小さな空洞を埋め、隣接する対象物を接続し、そのエッジを滑らかにします（面積に大きな変化はありません）。値が高いほど、効果は顕著になります。

初期値：7px

調整の例：カーネルサイズを大きくした場合の効果を下図に示します（カーネルは1～3まで大きくなる）。



4.3.297. バーコードを読み取る



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

2D画像のバーコードを読み取ります。複数のコードタイプに対応します。

使用シーン

汎用の画像処理ステップです。特定の使用シーンはありません。

入力と出力

- 入力：
 1. このポートに入力される画像のバーコードは、読み取られます。
- 出力：
 1. 読み取られたバーコードのコンテンツ。
 2. 画像で各バーコードの位置。

4.3.298. 画像を読み取る（V2）



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

ファイルまたはフォルダーから2D画像を読み取ります。

使用シーン

汎用の画像読み取りステップです。特定の使用シーンはありません。

入力と出力

- 入力：なし。
- 出力：
 1. 読み取られた画像。
 2. 読み取られた画像のファイル名。

4.3.299. 画像を読み取る



これは古いバージョンのステップであり、メンテナンスが停止されていますので、新しい

バージョンのステップ [\[vision-steps:read-images-v2:::read-images-v2\]](#) を使用してください。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

2D画像を読み取ります。

使用シーン

汎用の画像読み取りステップです。特定の使用シーンはありません。

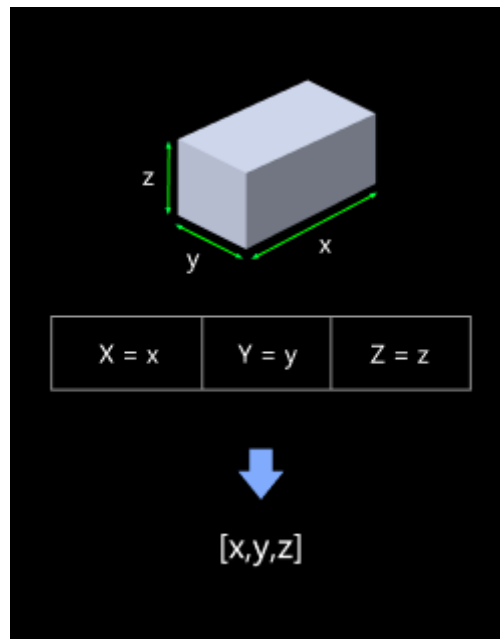
入力と出力

- 入力：
 1. なし。
- 出力：
 1. 読み取られた2D画像。
 2. 2D画像が対応するファイル名。

4.3.300. 対象物の寸法を読み込む

機能

このステップは、対象物の寸法リストを読み込みます。



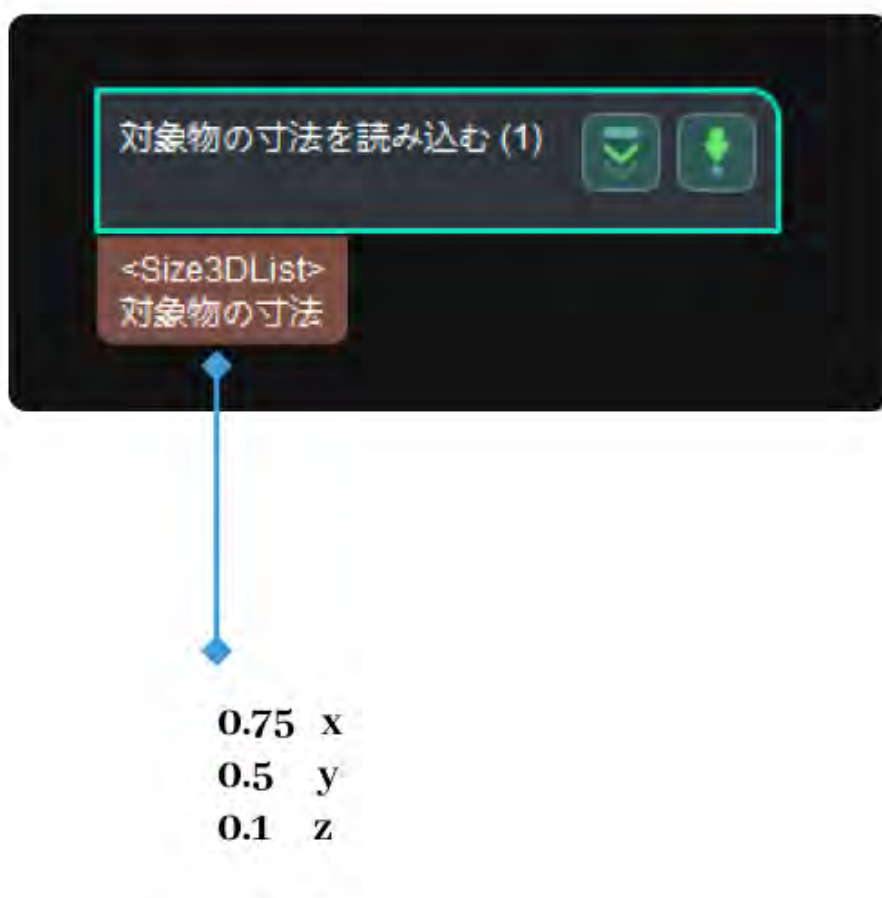
使用シーン

このステップでは、2つの方法で物体寸法を入力できます。

1. 物体寸法のJSONファイル。
2. パラメータ。

通常、**デパレタイジングプロジェクト**では段ボール箱の寸法を動的に設定する場合、または対象物の寸法を必要とする点群を分類する場合に使用されます。

入力と出力



パラメータの説明

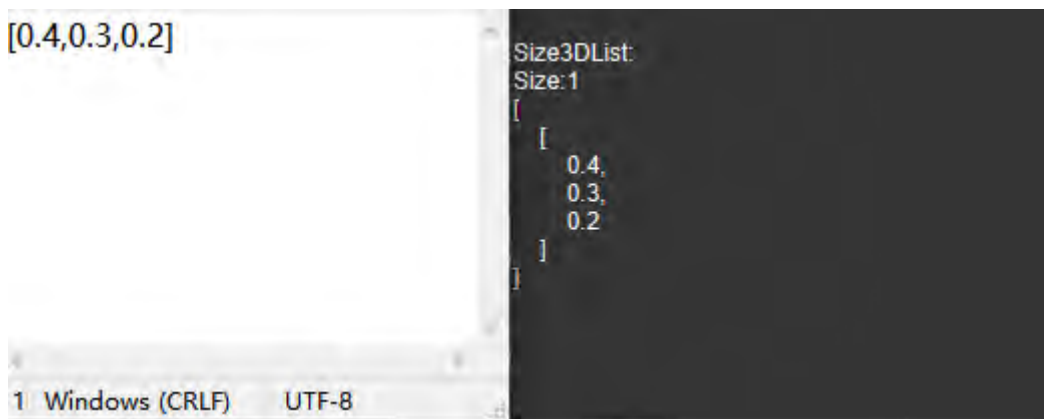
読込の設定

パラメータから対象物の寸法を読み込む

初期値：チェックを入れる。

調整説明：チェックを入れた場合、**箱の寸法設定**でX、Y、Z軸における箱の実際の長さを入力し、このステップを実行するときはパラメータから入力した寸法を読み込みます。**チェッ**

クを外した場合、指定したファイル「boxSizesList.json」のパラメータから対象物の寸法を読み込みます。下図に示すように、ファイル内は[0.4, 0.3, 0.2]で、対応する箱の長さ、幅、高さはそれぞれ0.4m、0.3m、0.2mです。

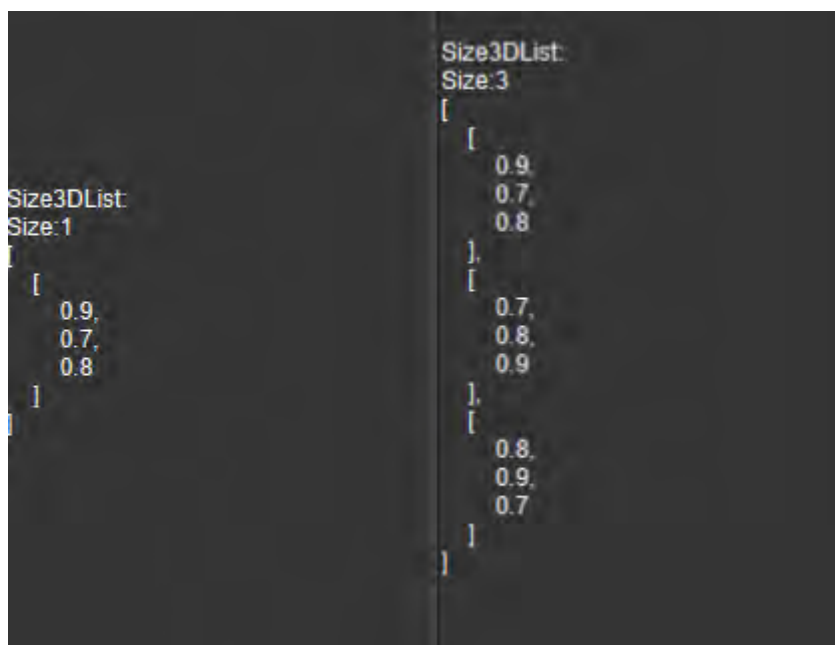


箱の寸法設定

箱のサイズを自動生成

初期値：チェックを入れない。

調整説明：このパラメータはマルチモデルのマッチングに使用され、一般的には使用されません。デフォルトでは、下図の左側に示すように、寸法値のセットが出力されます。チェックを入れると、3セットの寸法値が出力されます。下図の右側に示すように、元のx/y/z値はxyz/yzx/zxyの形式で出力されます。



X軸における長さ

初期値：100.000mm

調整説明：X軸に対象物の実際の長さ（ミリメートル単位）を入力します。有効な範囲は（0

， $+\infty$ ）です。

Y軸における長さ

初期値：100.000mm

調整説明：Y軸に対象物の実際の長さ（ミリメートル単位）を入力します。有効な範囲は（0， $+\infty$ ）です。

Z軸における長さ

初期値：100.000mm

調整説明：Z軸に対象物の実際の長さ（ミリメートル単位）を入力します。有効な範囲は（0， $+\infty$ ）です。

4.3.301. 点群を読み取る（V2）



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

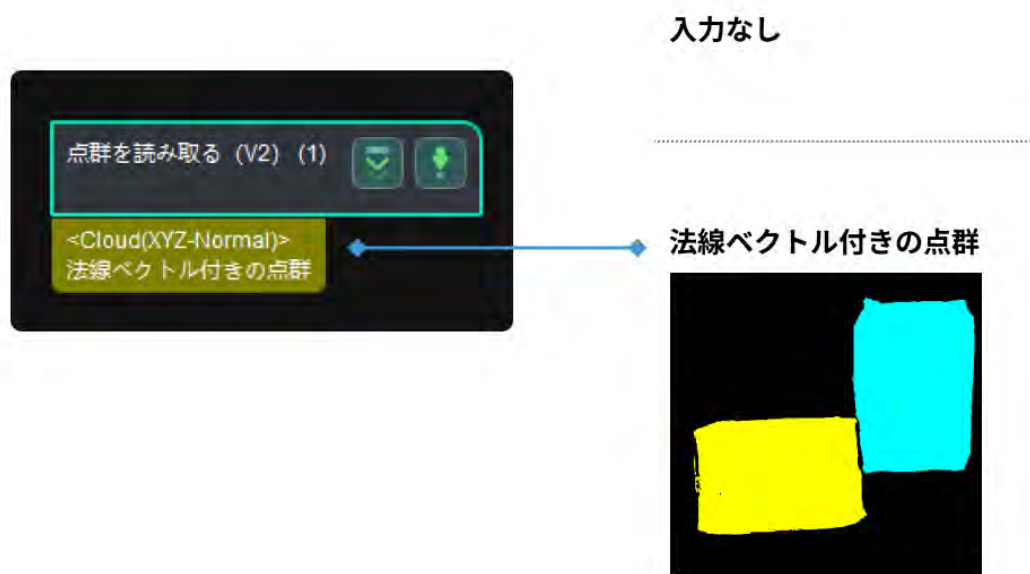
機能

ファイルまたはフォルダから点群を読み取ります。

使用シーン

汎用の点群を読み取るステップです。指定パスからPLY/PCD形式の点群を読み取ります。

入力と出力



4.3.302. 点群を読み取る



これは古いバージョンのステップであり、メンテナンスが停止されていますので、新しいバージョンのステップ [\[vision-steps:read-point-cloud-v2:::read-point-cloud-v2\]](#) を使用してください。このステップに関する詳細が必要な場合は、当社 (docs@mech-mind.net) までお問い合わせください。

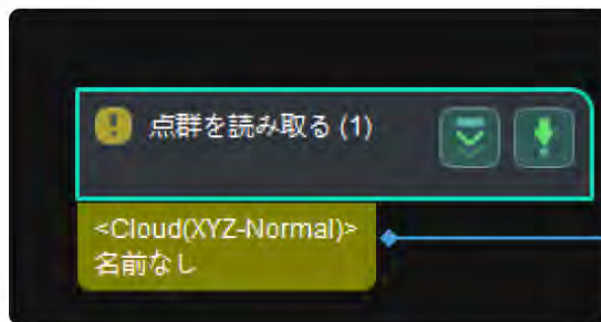
機能

ファイルまたはフォルダから点群を読み取ります。

使用シーン

指定パスからPLY形式の点群を読み取ります。

入力と出力



入力なし

法線ベクトル付きの3D点群



4.3.303. ファイルから位置姿勢を読み取る

機能

ローカルファイルから位置姿勢のリストを読み取ります。

使用シーン

このステップは基準位置姿勢を生成します。現在のプロジェクトには特定の読み込む位置姿勢ファイルがなければ、ステップ [\[vision-steps:easy-create-poses:::easy-create-poses\]](#) を使用してください。


入力と出力



パラメータの説明

位置姿勢ファイル

パラメータ説明：このパラメータは、位置姿勢ファイルのパスを選択するために使用されます。

調整説明：  をクリックして、必要なファイルを選択します。

4.3.304. QRコードを読み取る



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社 (docs@mech-mind.net) までお問い合わせください。

機能

2D画像のQRコードを解析し、QRコードの情報を文字列形式で出力します。

使用シーン

汎用のQRコード解析ステップです。特定の使用シーンはありません。

これは古いバージョンのステップであり、メンテナンスが停止されていますので、新しいバージョンのステップ [\[vision-steps:evaluate-results-by-hdev-engine::evaluate-results-by-hdevengine\]](#) を使用し、Halconをインストールしてください。

入力と出力

- **入力：** このポートに入力される2D画像のQRコードは解析されます。
- **出力：** 解析されたQRコードのコンテンツ。

4.3.305. STLファイルを読み取る



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社 (docs@mech-mind.net) までお問い合わせください。

機能

STL形式のファイルから点群データを読み取ります。

使用シーン

汎用の点群読み取りステップです。特定の使用シーンはありません。

入力と出力

- **入力：** なし。
- **出力：** 読み取られた点群データ。

4.3.306. 3D ROIの中心を読み取る



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社 (docs@mech-mind.net) までお問い合わせください。

機能

3D ROI形式のファイルを読み取り、3D ROIの中心位置姿勢を出力します。

使用シーン

特定のシーン向けのカスタマイズのステップです。

入力と出力

- **入力：** なし。
- **出力：** 3D ROI中心の位置姿勢。

4.3.307. 基準位置姿勢を記録して変換を計算

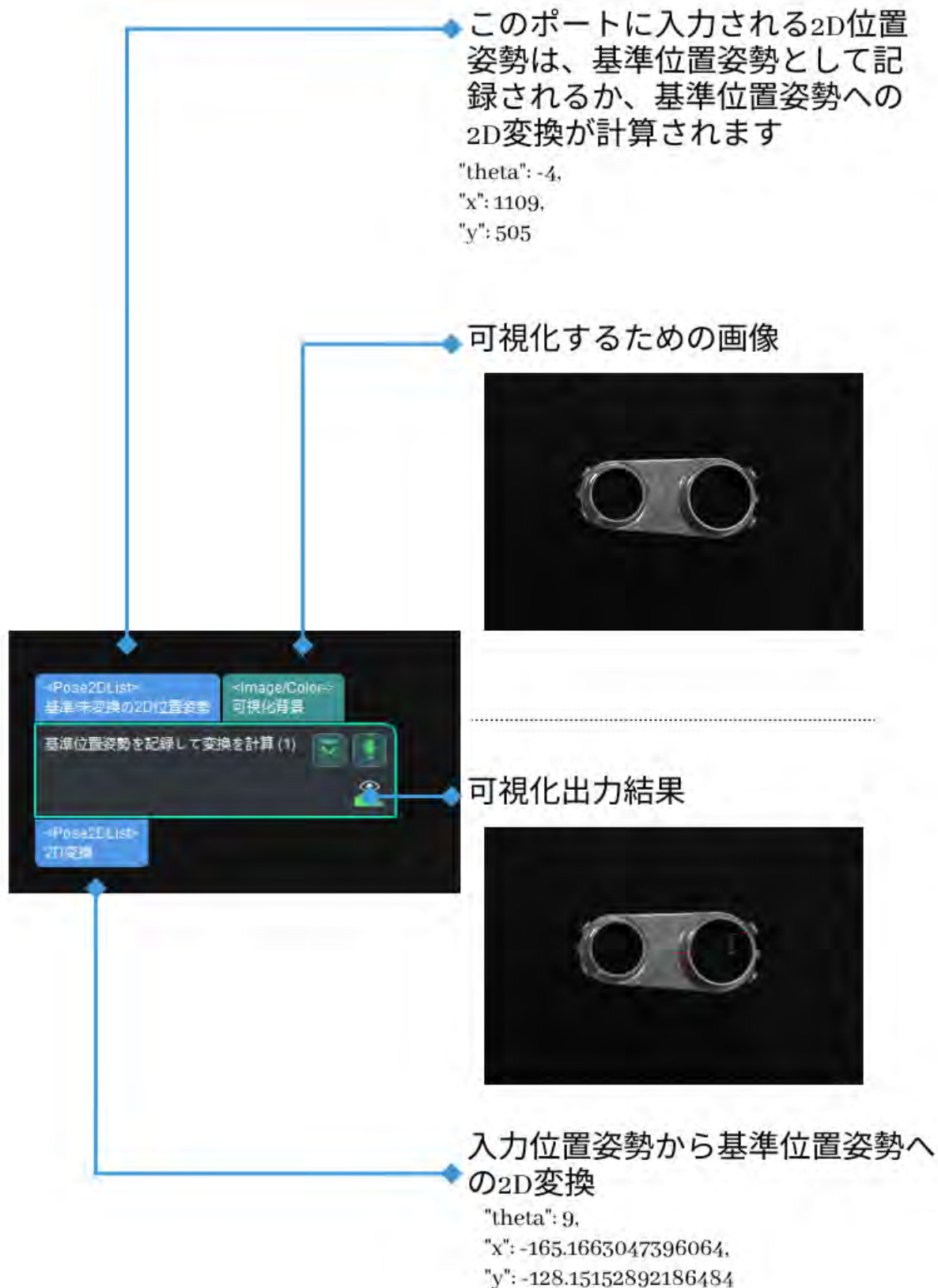
機能

特定の実行で入力された2D位置姿勢を画像の基準位置姿勢として指定し、後続の入力2D位置姿勢から基準位置姿勢への2D変換を計算します。

使用シーン

通常、計測シーンで画像を補正して、すべての画像内の対象物の位置と方向が一致していることを確保したり、関心領域の位置と方向を調整して画像内の対象物に収めたりするために使用されます。このステップは通常、ステップ [画像調整](#) と併用されます。

入力と出力



パラメータの説明

基準位置姿勢が作成済み

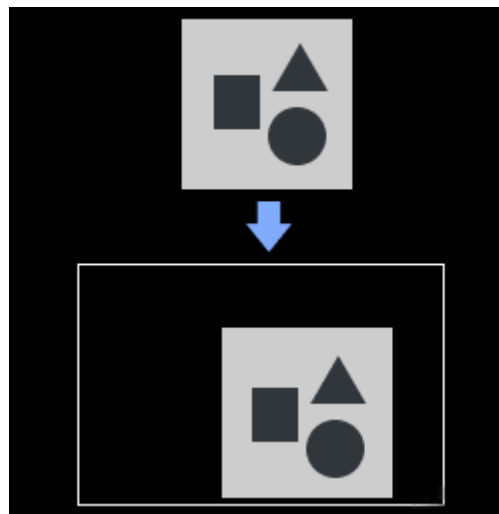
調整説明：以下の二つの状況があります。

- チェックを外す：入力される2D位置姿勢を基準位置姿勢として記録されます。
- チェックを入れる：入力される2D位置姿勢から基準位置姿勢への2D変換を計算します。

4.3.308. 2D ROI画像のスケール復元

機能

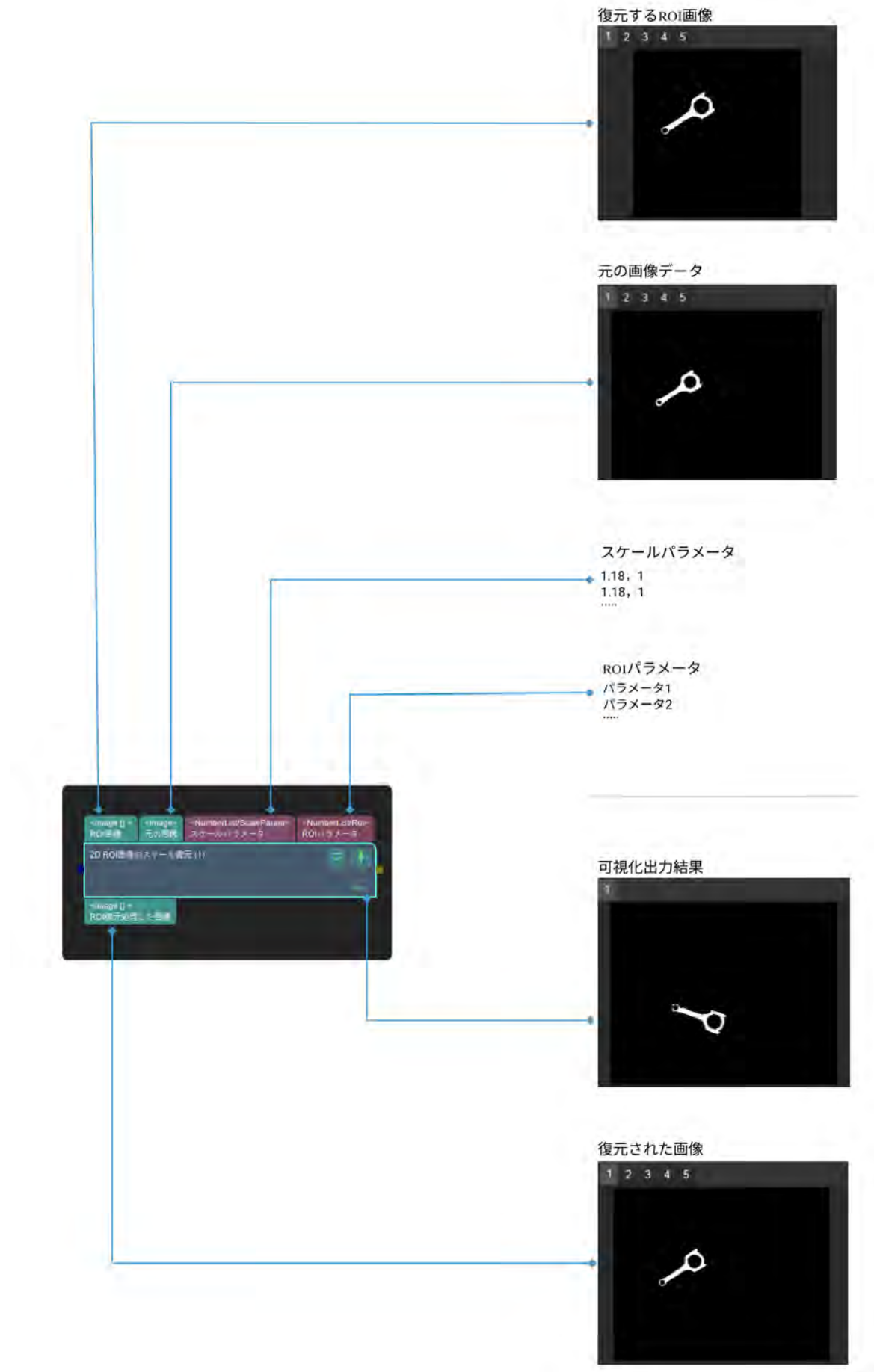
このステップを使用して入力された画像のサイズをスケーリングパラメータによって復元し、指定されたサイズの黒色背景画像に入れます。



使用シーン

通常、[\[vision-steps:scale-image-in-2d-roi:::scale-image-in-2d-roi\]](#) ステップとディープラーニングの関連ステップと併用されます。

入力と出力



パラメータの説明

パラメータはありません。

4.3.309. 画像の座標系を補正

機能

カラー画像を深度画像の座標系に補正するか、またはその逆に補正し、後続の画像処理のために2つのピクセルを1対1で対応させます。

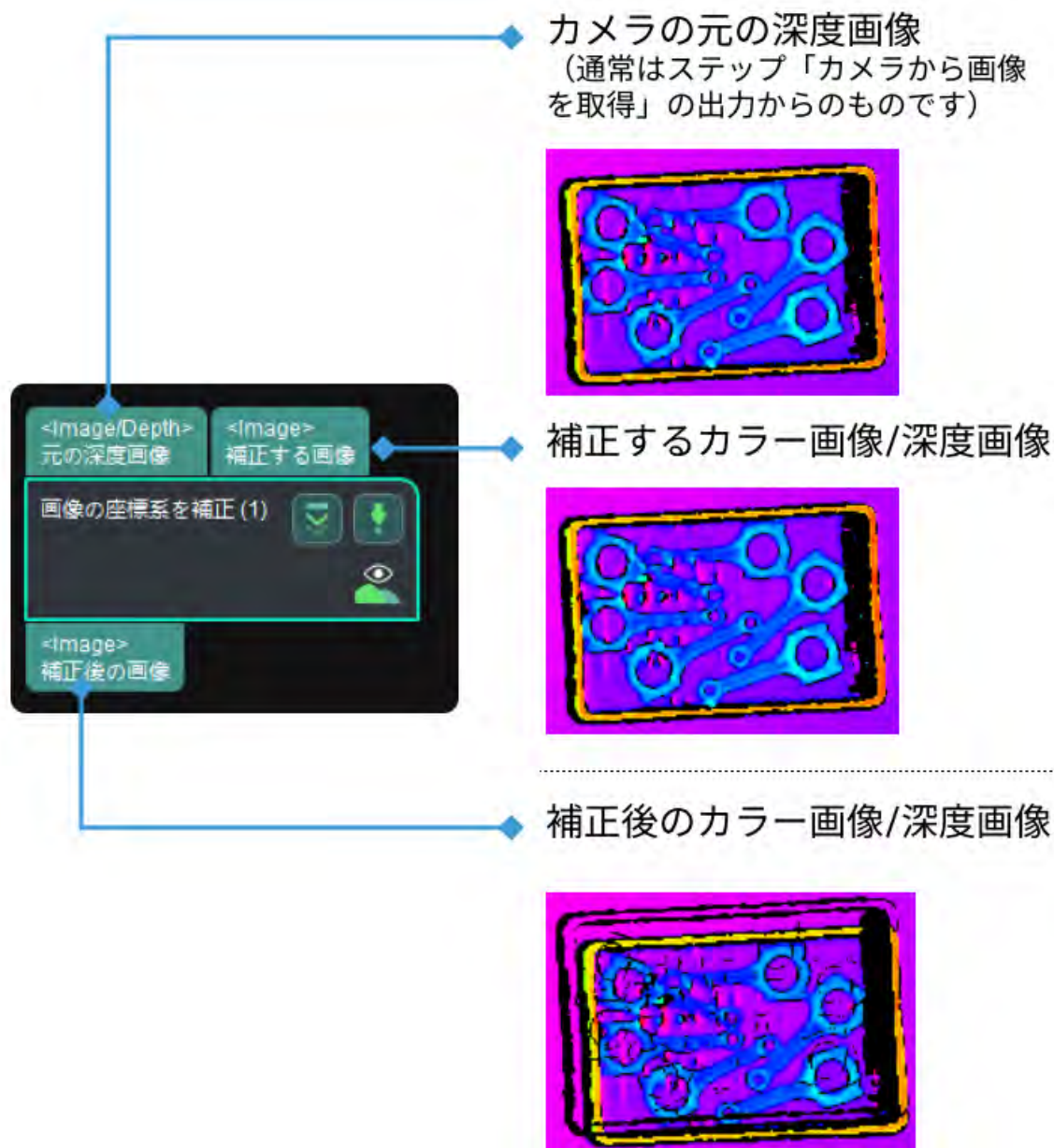
使用シーン

深度画像とカラー画像の座標系が一致しない場合に使用されます。



プロジェクト内の深度画像とカラー画像の座標系の関係がよくわかっていない場合、このステップはお勧めしません。

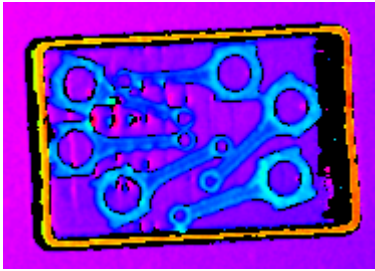
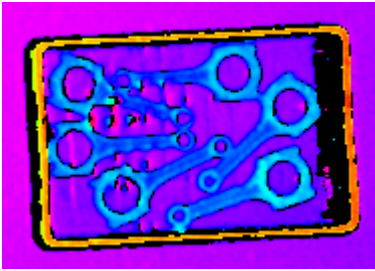
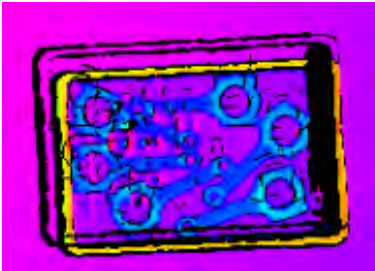
入力と出力



パラメータの説明

パラメータ		パラメータの説明
補正モード	深度画像からカラー画像の座標系へ	深度画像をカラー画像の座標系に補正します
	カラー画像から深度画像の座標系へ	カラー画像を深度画像の座標系に補正します

深度画像からカラー画像の座標系への変換を下図に示します。

元の深度画像	補正する深度画像	補正後の深度画像
		

カラー画像から深度画像の座標系への変換を下図に示します。

元の深度画像	補正するカラー画像	補正後のカラー画像
		

4.3.310. リング形状の対象物の位置姿勢を補正



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

入力されるリング形状の対象物の位置姿勢を、リングの中心位置に補正します。

使用シーン

特定のシーン向けのカスタマイズのステップです。

入力と出力

● 入力：

1. このポートに入力されるリング形状の対象物の位置姿勢は、リングの中心位置に補正されます。
2. リングのマスク。

● 出力：

1. 補正された把持位置姿勢。

4.3.311. 近すぎる位置姿勢を除去



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

両位置姿勢間の距離が設定されたしきい値よりも小さい場合、両位置姿勢はすべて除去されます。

使用シーン

汎用の位置姿勢フィルタリングのステップです。特定の使用シーンはありません。

入力と出力

- 入力：
 1. このポートに入力される元の位置姿勢リストはフィルタリングされます。
 2. 元の位置姿勢のラベルリスト。
 3. 元の位置姿勢の信頼度リスト。
- 出力：
 1. フィルタリング後の位置姿勢リスト。
 2. フィルタリング後の位置姿勢のラベルリスト。
 3. フィルタリング後の位置姿勢の信頼度リスト。

4.3.312. 点群の点を一部除去



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

点群内の指定した点を除去します。

使用シーン

シーンからモデルが対応する点群を除去／保持するためによく使われます。

入力と出力

- 入力：
 1. 元の点群。
 2. 元の点群から除去する一部の点群。

● 出力：

1. 一部の点群が除去された後の残りの元の点群。
2. 元の点群から除去された点。

4.3.313. マスクのノイズを除去



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

面積がしきい値よりも小さいノイズクラスターを削除します。ノイズクラスター領域のしきい値は、パラメータで設定できます。

使用シーン

汎用の画像処理ステップです。特定の使用シーンはありません。

入力と出力

- 入力：このポートに入力されるマスクのノイズが除去されます。
- 出力：ノイズが除去された後のマスク。

4.3.314. 重複対象物を除去（V2）

機能

指定されたルールに従って重ねられた対象物の位置姿勢を除去します。下図に示すように、右下隅にある赤い対象物は重ねられた対象物として除去されます。



使用シーン

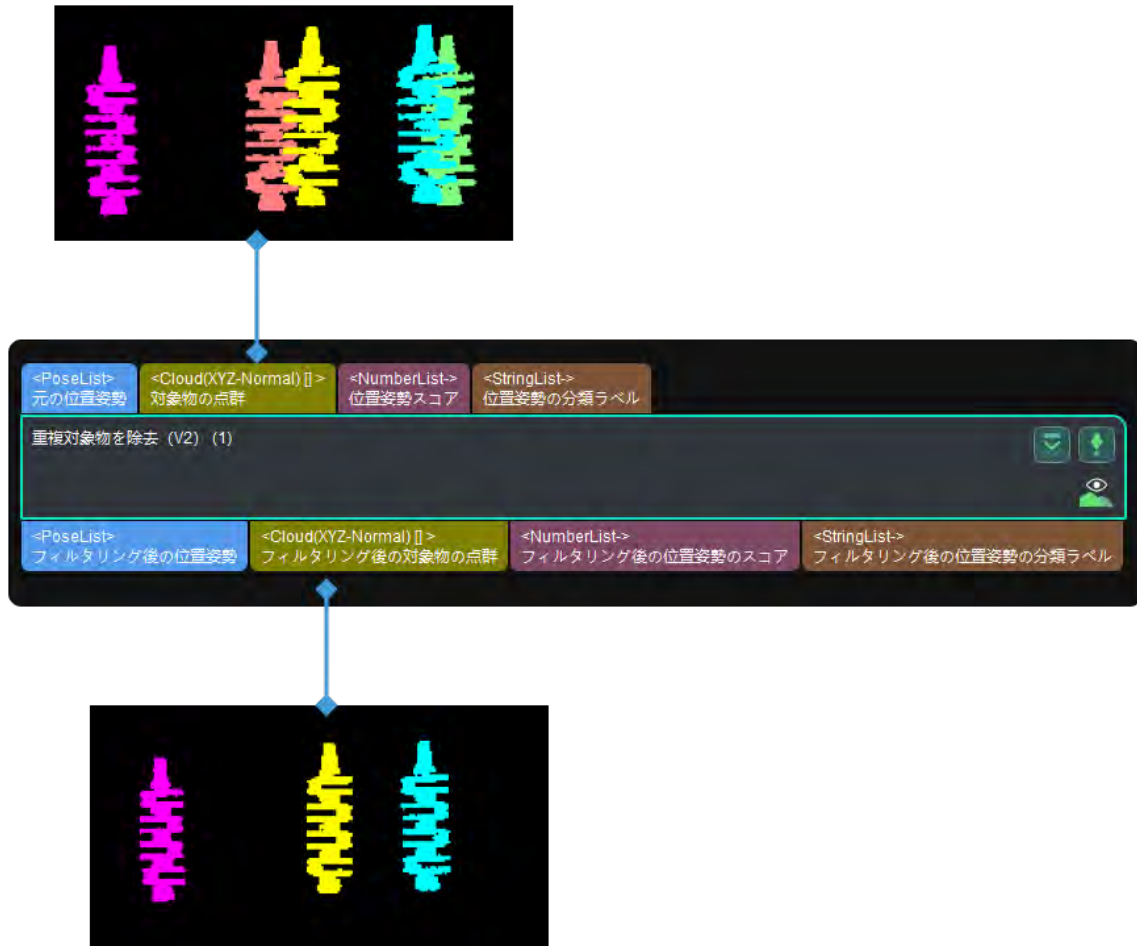
通常、ステップ [3D位置姿勢高精度推定](#) の後で使用されます。重なっているため、把持などの処理に適さない対象物の位置姿勢を除去するために使用されます。



[重複対象物を除去（V2）](#) ステップは投影法（2D）を最適化しました。投影法（2D）を使用する場合、[重複対象物を除去（V2）](#) を推奨します。

重複対象物を除去 (V2) は境界ボックス法 (3D) を変更しませんでした。境界ボックス法 (3D) を使用する場合、[重複対象物を除去](#) と [重複対象物を除去 \(V2\)](#) の両方を使用できます。

入力と出力



パラメータの説明

このステップでは、対象物が積み重ねられているかどうかを判断する方法が2つ用意されています。

- 境界ボックス (3D) : 点群の境界ボックスを小さな立方体に分割し、重なっている小さな立方体の数で重なり率を計算します。
- 投影 (2D) : 対象物が2D平面に投影する重なり領域の割合を計算し、積み重ねられているかどうかを判断します。

方法設定

方法

調整説明：このパラメータは、積み重ねられた対象物を除去する方法を選択するために使用されます。

オプション：境界ボックス（3D）、投影（2D）。

境界ボックス（3D）：プロジェクトで3Dマッチングを使用しない場合や、3Dマッチングのエッジマッチングを使用する場合に推奨されます。この方法は、対象物が3D境界ボックスによって対象物が重なっているかどうかを判断するします。3D境界ボックスとは、対象物の中心を中心とし、位置姿勢のX、Y、Z軸に平行な辺を持つ直方体のことを指します。

投影（2D）：プロジェクトで3Dマッチングを使用し、マッチング方式がサーフェスマッチングの場合に推奨されます。2D平面に投影された重なり領域の割合を計算することで、対象物が重なっているかどうかを判断するします。この方法を使用する場合、**重複比率のしきい値**を設定するだけで、他のパラメータの設定は必要ありません。

調整アドバイス：実際のニーズに応じて選択してください。

境界ボックス（3D）

しきい値設定

重複比率の計算方法

パラメータ説明：このパラメータは、対象物の重なり比率の計算方法を選択するために使用されます。

オプション：

- 単一の重なり面積を計算：対象物と、その上にある単一の対象物との重なり面積の比率を順番に計算します。
- すべての重なり領域を計算：対象物、とその上にあるすべての対象物との重なり領域の面積合計の比率を計算します。

調整アドバイス：実際のニーズに応じて選択してください。

重複比率のしきい値 (0~1.0)

パラメータ説明：重複比率とは、境界ボックスのと他の境界ボックスと重なる部分の割合のことです。このパラメータは、積み重ねられた対象物のビジョン結果を除去するかどうかを設定するために使用されます。対象物の点群と他の対象物の点群との重複領域の比率がこのしきい値を超えると、その対象物のビジョン結果は除去されます。

値の範囲：0.00~1.00

初期値：0.30

調整アドバイス：実際のニーズに応じて、0.01のステップサイズで **重複比率のしきい値** を調整できます。調整効果を [調整の例](#) に示します。

点群の解像度設定

対象物の点群の対角線長さの比率

パラメータ説明：このパラメータは、境界ボックスを小さな立方体に分割するために使用されます。重複比率しきい値は、重複する小さな立方体の数によって計算されます。小さな立方体の辺長は、対象物の点群の対角線長さの比率に対象物の点群の対角線長さを掛けたものにです。

初期値：2.00%

調整アドバイス：実際の状況に応じて設定してください。

対象物高さの設定

対象物高さの計算方法

パラメータ説明：このパラメータは、対象物高さの計算方法を選択するために使用されます。

オプション：高さを指定、点群から計算（初期値）。

- 高さを指定：パラメータ **指定された高さ** の値を設定することで、固定対象物の高さを指定します。
- 点群から計算：対象物の高さは入力点群に従って自動的に計算されます。

調整アドバイス：平らかな点群、または対象物の形状を反映できない点群の場合、対象物の高さを指定する必要があります。

高さを指定

パラメータ説明：このパラメータは、対象物の高さ（ミリメートル単位）を指定するために使用されます。**対象物高さの計算方法** が **高さを指定** に設定された場合、このパラメータを設定する必要があります。

初期値：100.000mm

調整アドバイス：実際の状況に応じて設定してください。

境界ボックスの拡張

境界ボックスのX方向の拡張率

パラメータ説明：境界ボックスを対象物の位置姿勢のX軸に沿って膨張し、重なりをより精確に検出することができます。

初期値：1.0000

調整アドバイス：実際の状況に応じて設定してください。

境界ボックスのY方向の拡張率

パラメータ説明：境界ボックスを対象物の位置姿勢のY軸に沿って膨張し、重なりをより精確に検出することができます。

初期値：1.0000

調整アドバイス：実際の状況に応じて設定してください。

境界ボックスのZ方向の拡張率

パラメータ説明：境界ボックスを対象物の位置姿勢のZ軸に沿って膨張し、重なりをより精確に検出することができます。**対象物高さの計算方法**が**高さを指定**に設定された場合、このパラメータを設定する必要はありません。

初期値：3.0000

調整アドバイス：実際の状況に応じて設定してください。

投影（2D）

しきい値設定

重複比率のしきい値 (0-1.0)

パラメータ説明：重複比率とは、他のマスクと重なる部分の割合のことです。このパラメータは、積み重ねられた対象物のビジョン結果を除去するかどうかを設定するために使用されます。対象物の点群と他の対象物の点群との重複領域の比率がこのしきい値を超えると、その対象物のビジョン結果は除去されます。

初期値：0.30

値の範囲：0.00~1.00

調整アドバイス：実際のニーズに応じて、0.01のステップサイズで**重複比率のしきい値**を調整できます。調整効果を [調整の例](#) に示します。

詳細設定



すべてのパラメータを表示 に設定された場合のみに表示されます。

投影のタイプ

パラメータ説明：このパラメータは、投影のタイプを選択するために使用されます。

オプション：透視投影（初期値）、正投影。

正投影：対象物の大きさが視野の深さに応じて伸びることがありません。

- 透視投影：対象物の大きさは視野の深さに応じて伸びます。

調整アドバイス：実際の状況に応じて設定してください。

膨張カーネルサイズ

パラメータ説明：このパラメータは、投影された2D画像上の空白ピクセルを避けるために、投影画像を膨張するためのカーネルサイズを設定するために使用されます。

初期値：1

調整アドバイス：実際の状況に応じて設定してください。

ダウンサンプリング係数

パラメータ説明：このパラメータは、ダウンサンプリング係数を設定するために使用されます。処理を高速化するために、投影後の画像をダウンサンプリングします。**投影のタイプ**が**透視投影**に設定された場合、このパラメータを設定する必要があります。例えば、係数を2に設定すると、100*100の画像が50*50になります。

初期値：2

調整アドバイス：実際の状況に応じて設定してください。

正投影の解像度

パラメータ説明：このパラメータは、正投影で形成された2D画像上の単位長さあたりのピクセル数です。**投影のタイプ**を**正投影**に設定した場合、このパラメータを設定する必要があります。

初期値：500.0000

調整アドバイス：実際の状況に応じて設定してください。

モデル周りの点群除去範囲

パラメータ説明：入力位置姿勢とモデルから点群を生成する際、生成された点群の周囲で、このパラメータで設定された範囲内のシーンの点群が除去されます。残りの点群は、重なり検出に関与することになります。単位はミリメートル（mm）です。

初期値：3.000mm



シーンの点群とは、カメラ視野内にあるすべての点群のことを指します。

可視化オプション

パラメータ説明：このパラメータは、可視化する項目を選択するために使用されます。

オプション：フィルタリング後の結果、ハイライト表示の除去されたシーンの点群、投影画像。

フィルタリング後の結果：デバッグ結果出力のウィンドウで、重複対象物が除去された後の結果を確認できます。

- ハイライト表示の除去されたシーンの点群：シーンの点群は、特に未認識の対象物の部分について、重なり検出を有効にする必要があります。重なり検出を実行する前に、シーン点群の認識された対象物を除去し、対象物の点群のためのスペースを確保する必要があります。対象物とは、入力位置姿勢とモデルによって生成された点群のことを指します。

- 投影画像：リストの最初の対象物の投影を確認し、投影の効果をプレビューします。
調整アドバイス：実際の状況に応じて設定してください。調整効果を [調整の例](#) に示します。

調整の例

重複比率のしきい値

重複比率のしきい値をそれぞれ0.60、0.10、0.03に設定した場合の可視化結果は以下の通りです。その中、除去された重ねられた対象物の点群は白で表示されます。

しきい値の範囲	0.60	0.01	0.03
可視化出力結果			

可視化オプション

可視化オプションが **フィルタリング後の結果** と **投影画像** の場合、このステップの可視化出力はそれぞれ下図の左側と右側に示されます。



4.3.315. 重複対象物を除去

機能

ユーザーに指定されたルールに従って積み重ねられた対象物の点群を除去します。

下図に示すように、右下隅にある赤い点群は重ねられた点群として除去されます。



使用シーン

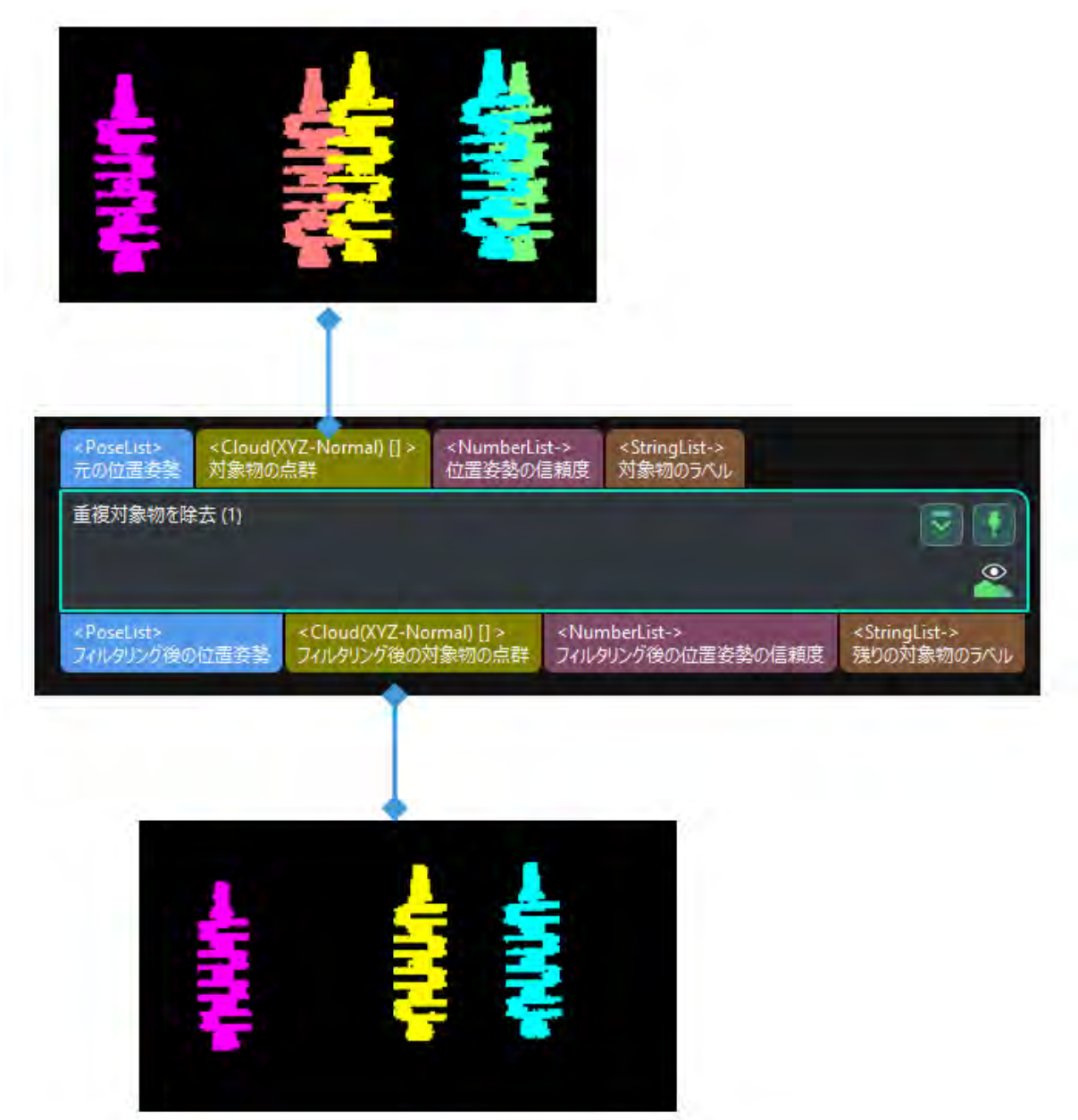
通常、[\[vision-steps:3d-fine-matching:::3d-fine-matching\]](#) ステップの後で使用されます。積み重ねられた対象物を除去するという目的を達成するために、様々な要件に従ってマッチングした位置姿勢をフィルタリングします。



[重複対象物を除去 \(V2\)](#) ステップは投影法 (2D) を最適化しました。投影法 (2D) を使用する場合、[重複対象物を除去 \(V2\)](#) を推奨します。

[重複対象物を除去 \(V2\)](#) は境界ボックス法 (3D) を変更しませんでした。境界ボックス法 (3D) を使用する場合、「重複対象物を除去」と [重複対象物を除去 \(V2\)](#) の両方を使用できます。

入力と出力



パラメータの調整説明

アルゴリズムは2種類ありますので、選択された種類に応じてパラメータを調整してください。

BoundingBoxOfObjectIn3DBased（境界ボックス）

このアルゴリズムは、点群の境界ボックスを小さな立方体に分割し、重なり合う対象物に重なる小さな立方体の数をカウントすることで、対象物と重なり合う対象物の比率を計算します。

しきい値設定

重複比率のしきい値 (0-1.0)

パラメータ説明：重複比率とは、対象物の境界ボックスと重複する対象物の重なり合う領域の体積比率を指します。このパラメータは、重複する対象物の点群を除去するかどうかを設

定するために使用されます。対象物の点群と重複する対象物の点群との重なり合う領域の比率がこの値を超える場合、対象物は重複する対象物と見なされ、対象物の点群が除去されます。

値の範囲：0~1.0

初期値：0.60

調整アドバイス：実際のニーズに応じて、0.01のステップサイズで **重複比率のしきい値** を調整できます。

点群の解像度設定

点群の境界ボックスの解像度

パラメータ説明：このパラメータは、点群の境界ボックスのサイズ、つまり対象物の点群の境界ボックスを分割する際のサイズを設定するために使用されます。そして、点群の境界ボックスをカウントすることで、対象物と重複する対象物の重なりを計算します。

初期値：3mm。

推奨値：実際のニーズに応じて設定してください。

対象物高さの設定

対象物高さの計算方法

パラメータ説明：このパラメータは、対象物高さの計算方法を選択するために使用されます。

オプション：

UsingInputCloudHeight：初期値で、入力した点群に従って高さを計算します。

UsingSpecifiedHeight：対象物の高さを指定します。点群が平らな場合や、点群が対象物の完全な形状を反映していない場合に使用されます。

高さを指定

パラメータ説明：このパラメータは、対象物の高さ（ミリメートル単位）を指定するために使用されます。**対象物高さの計算方法** が **高さを指定** に設定された場合、このパラメータを設定する必要があります。

初期値：100mm

調整アドバイス：実際の状況に応じて設定してください。

寸法拡張設定

境界ボックスのX方向の拡張率

パラメータ説明：境界ボックスを対象物の位置姿勢のX軸に沿って膨張し、重なりをより精

確に検出することができます。

初期値：1.0000

調整アドバイス：実際の状況に応じて設定してください。

境界ボックスのY方向の拡張率

パラメータ説明：境界ボックスを対象物の位置姿勢のY軸に沿って膨張し、重なりをより精確に検出することができます。

初期値：1.0000

調整アドバイス：実際の状況に応じて設定してください。

境界ボックスのZ方向の拡張率

パラメータ説明：境界ボックスを対象物の位置姿勢のZ軸に沿って膨張し、重なりをより精確に検出することができます。**対象物高さの計算方法**が**高さを指定**に設定された場合、このパラメータを設定する必要はありません。

初期値：3.0000

調整アドバイス：実際の状況に応じて設定してください。

ProjectionOfObjectIn2DBased（投影法）

2D平面に投影された重なり領域の割合を計算することで、対象物が重なっているかどうかを判断します。

しきい値設定

重複比率のしきい値 (0-1.0)

パラメータ説明：重複比率とは、対象物の境界ボックスと重複する対象物の重なり合う領域の体積比率を指します。このパラメータは、重複する対象物の点群を除去するかどうかを設定するために使用されます。対象物の点群と重複する対象物の点群との重なり合う領域の比率がこの値を超える場合、対象物は重複する対象物と見なされ、対象物の点群が除去されます。

値の範囲：0~1.0

初期値：0.60

調整アドバイス：実際のニーズに応じて、0.01のステップサイズで**重複比率のしきい値**を調整できます。

対象物高さの設定

対象物高さの計算方法

パラメータ説明：このパラメータは、対象物高さの計算方法を選択するために使用されま

す。

オプション：

UsingInputCloudHeight：初期値で、入力した点群に従って高さを計算します。

- UsingSpecifiedHeight：対象物の高さを指定します。点群が平らな場合や、点群が対象物の完全な形状を反映していない場合に使用されます。

高さを指定

パラメータ説明：このパラメータは、対象物の高さ（メートル単位）を指定するために使用されます。**対象物高さの計算方法**が**高さを指定**に設定された場合、このパラメータを設定する必要があります。

初期値：100mm

調整アドバイス：実際の状況に応じて設定してください。

詳細設定

マスククロージング処理のカーネルサイズ

パラメータ説明：このパラメータは、マスククロージング処理のカーネルサイズを調整するために使用されます。

初期値：3

推奨値：実際のニーズに応じて設定してください。

深度画像膨張カーネルサイズ

パラメータ説明：このパラメータは、深度画像膨張カーネルサイズを調整するために使用されます。

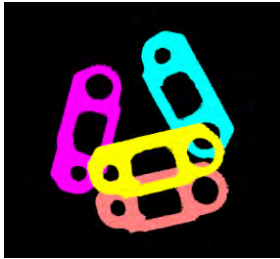
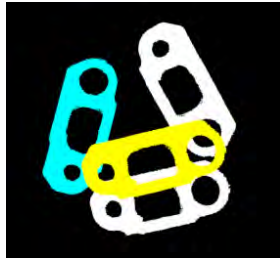

初期値：4

推奨値：実際のニーズに応じて設定してください。

調整の例

重複比率のしきい値

重複比率のしきい値をそれぞれ0.60、0.10、0.03に設定した場合の可視化結果は以下の通りです。その中、除去された重ねられた対象物の点群は白で表示されます。

しきい値の範囲	0.60	0.01	0.03
可視化出力結果			

4.3.316. 重複ポリゴンを除去

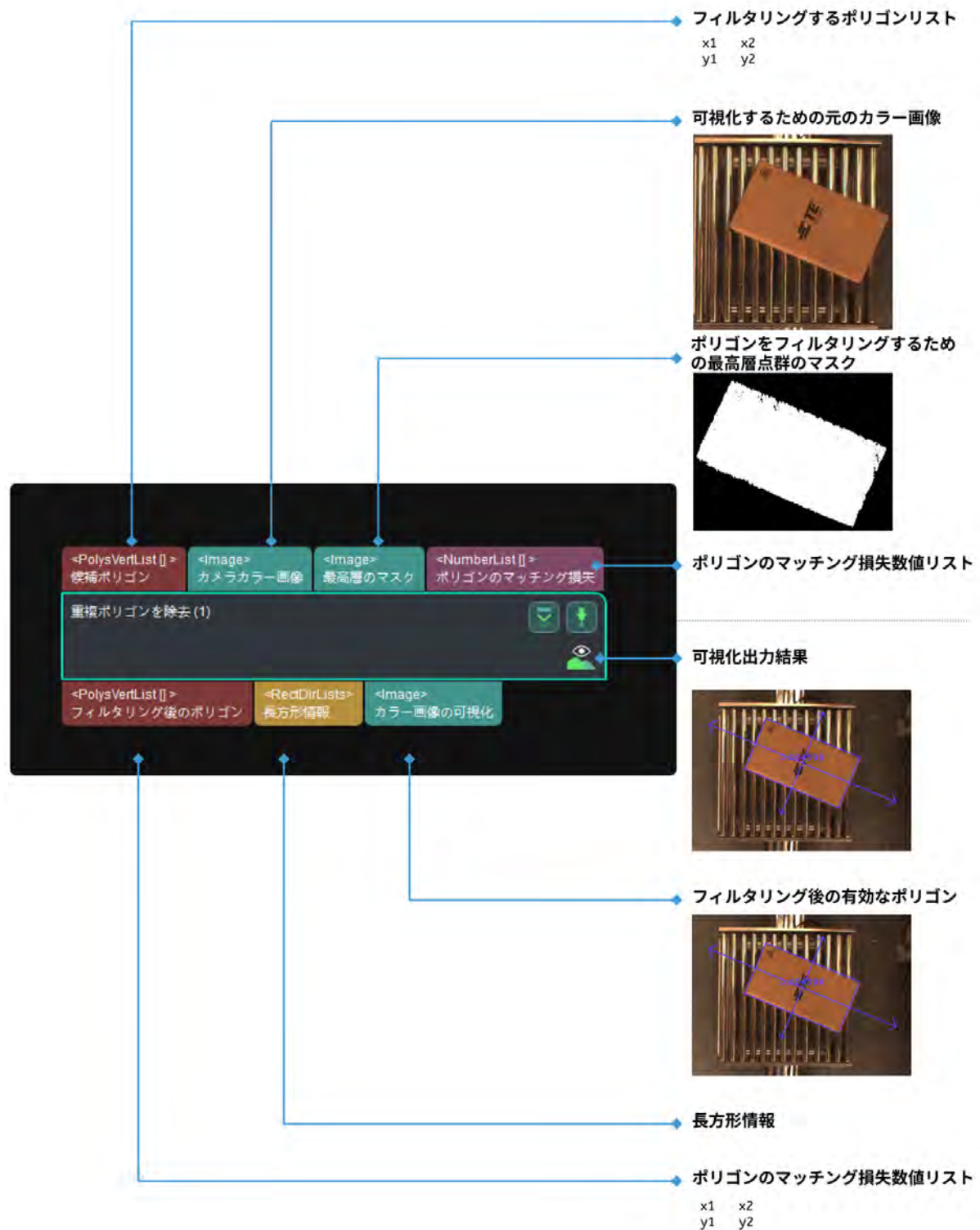
機能

重複するポリゴンやマスク以外の無効なポリゴンを除去します。

使用シーン

ステップ [マスク以外のポリゴンを除去](#) および [長方形の寸法と位置姿勢を計算](#) と併用することになっています。これは、実際の対象物の輪郭を反映しないポリゴンを除去するために使用されます。

入力と出力



パラメータの説明

マスク設定

入力マスクの使用方法

初期値：SingleMask

オプション：SingleMask（一つのマスクを全てのポリゴンに適用）、CorrespondingMasks（各ポリゴンに異なるマスクを適用）。

調整説明：マスクの使用方法を選択します。全てのポリゴンに一つのマスクを使用するか、各ポリゴンに異なるマスクを使用します。

重複除去の検査設定

エッジを超えた比率の最小しきい値

初期値：0.45

調整説明：箱のマスクを長軸と短軸の正と負の方向に沿ってそれぞれ長さの半分と幅の半分だけ並進し、マスク自体の面積に対する並進されたマスクが最高層のマスクを超えた部分の面積の比率を計算します。しきい値よりも大きい場合、箱は最高層の境界上にあると見なされます。一般的に、長方形の箱のマスクが最高層マスクのエッジにある場合、マスクが最高層から移動する方向にマスクの長さまたは幅の半分だけ並進されている限り、この比率は0.5である必要があります。したがって、デフォルトの最小しきい値は0.45です。この値を大きくすると、一般に0.5を超えると、すべてのポリゴンが除去されるようになります。このステップでは、コーナーの箱が優先されるため、最初に境界の箱のマスク領域を決定する必要があります。

エッジを超えた比率の最大しきい値

初期値：0.6

調整説明：箱のマスクを長軸と短軸の正と負の方向に沿ってそれぞれ長さの半分と幅の半分だけ並進し、マスク自体の面積に対する並進されたマスクが最高層のマスクを超えた部分の面積の比率を計算します。しきい値よりも小さい場合、箱は最高層の境界上にあると見なされます。一般的に、長方形の箱のマスクが最高層マスクのエッジにある場合、マスクが最高層から移動する方向にマスクの長さまたは幅の半分だけ並進されている限り、この比率は0.5である必要があります。したがって、デフォルトの最大しきい値は0.6です。このステップでは、コーナーの箱が優先されるため、最初に境界の箱のマスク領域を決定する必要があります。

ポリゴンの最大重複比

初期値：0.2

調整説明：両ポリゴンの重複比はこのしきい値よりも大きい場合は、その中の1つを除去する必要があります。

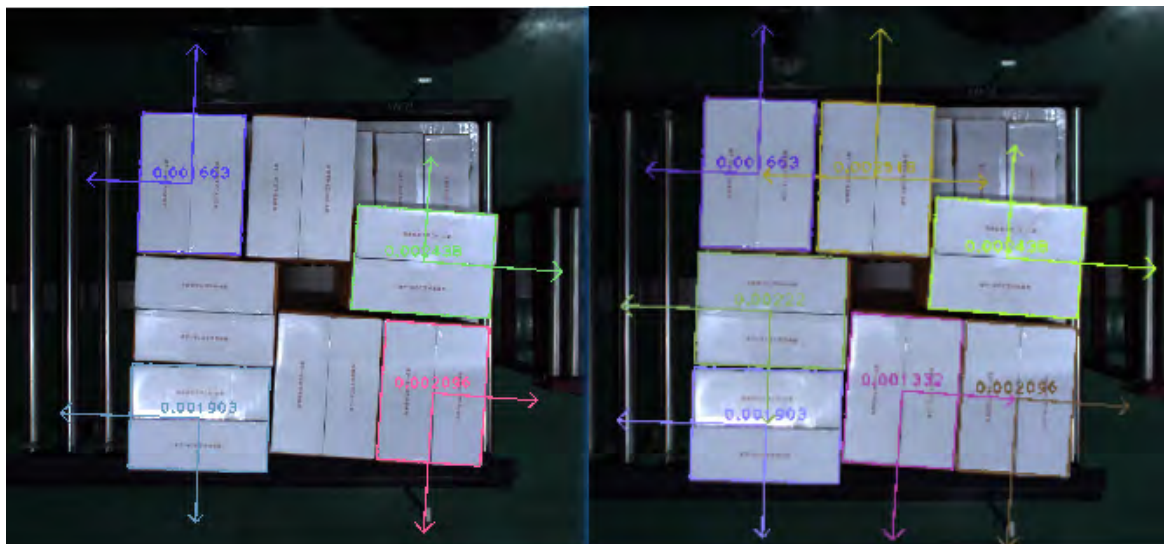
出力戦略

位置制限

初期値：AtCorners

オプション：AtCorners、None

調整説明：下図に示すように、左はデフォルト設定の出力画像で、右は **None** に設定した場合の出力画像です。



4.3.317. 点群の点を一部除去

機能

1つ点群から一部の点群を除去します。

4.3.318. マスク以外のポリゴンを除去

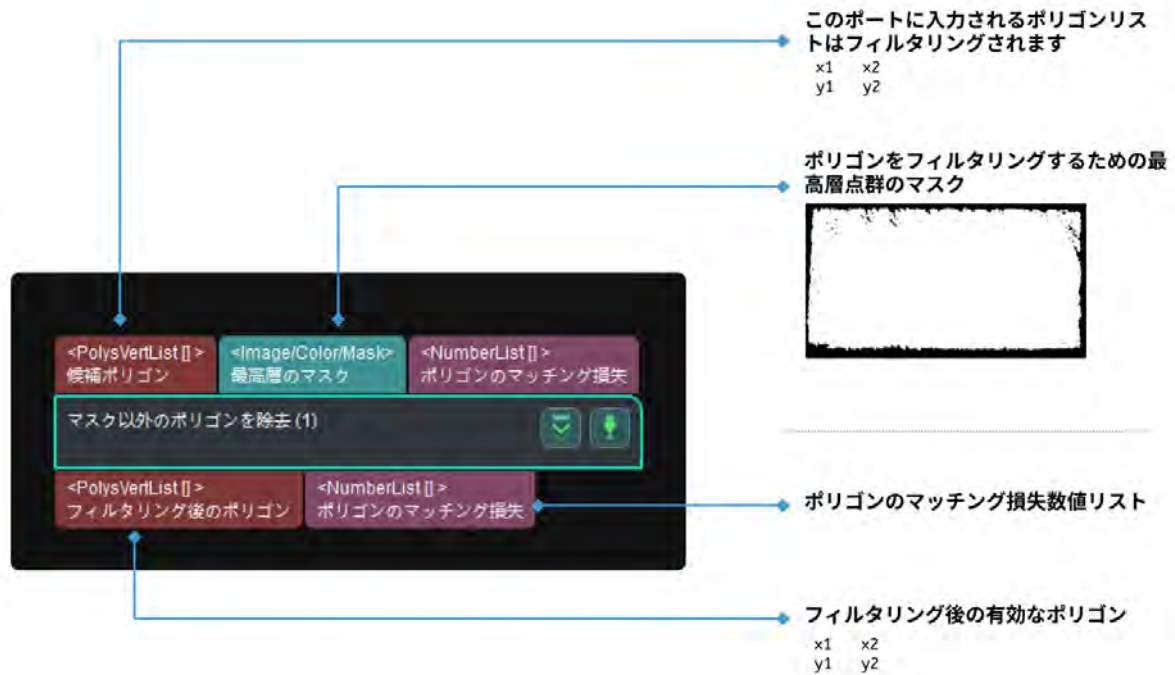
機能

マスクとの重複領域が設定値よりも小さい干渉ポリゴンを除去します。

使用シーン

ステップ [\[vision-steps:2d-matching:::2d-matching\]](#) および [重複ポリゴンを除去](#) と併用することになっています。マスクを使用してマッチングすることによって得られた干渉ポリゴンを除去します。

入力と出力



パラメータの説明

入力マスクの使用方式

初期値：SingleMask

オプション：SingleMask（一つのマスクを全てのポリゴンに使用される）、CorespondinMasks（各ポリゴンによって異なるマスクを使用する）。

調整説明：マスクの使用方法を選択します。全てのポリゴンに一つのマスクを使用するか、各ポリゴンに異なるマスクを使用します。

ポリゴンとマスクの面積の重なり率の最小値 (0-1.0)

初期値：0.8000

調整説明：このパラメータは、マスクとの重複面積比のしきい値を調整するために使用されます。マスクとの重複面積比がこのしきい値よりも大きいポリゴンは、保持されます。マスクとの重複面積比がこのしきい値よりも小さいポリゴンは、マスク以外のポリゴンは、除去されます。

4.3.319. 指定した下付き文字に従って並べ替える



これは古いバージョンのステップであり、メンテナンスが停止されていますので、新しいバージョンのステップ [\[vision-steps:reorder-by-index-list::reorder-by-index-list\]](#) を使用してください。このステップに関する詳細が必要な場合は、当社 (docs@mech-mind.net) までお問い合わせください。

機能

入力されたインデックスに従って、数値リストをソートします。

使用シーン

汎用の数値ソートのステップです。特定の使用シーンはありません。

入力と出力

● 入力：

1. このポートに入力されるデータリストはインデックスによってソートされます。
2. 入力のデータをソートするためのインデックス。

● 出力：

1. ソートされたデータリスト。

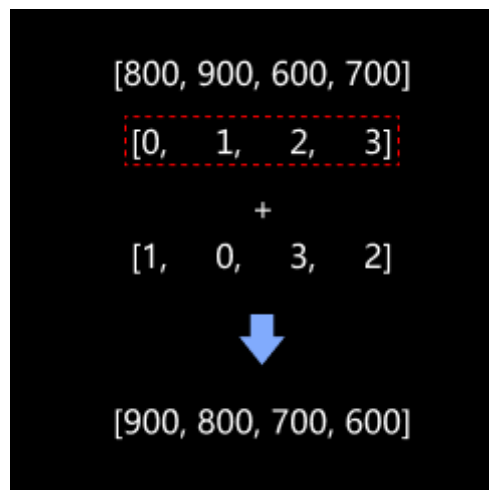
4.3.320. インデックスによって要素を並べ替える



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

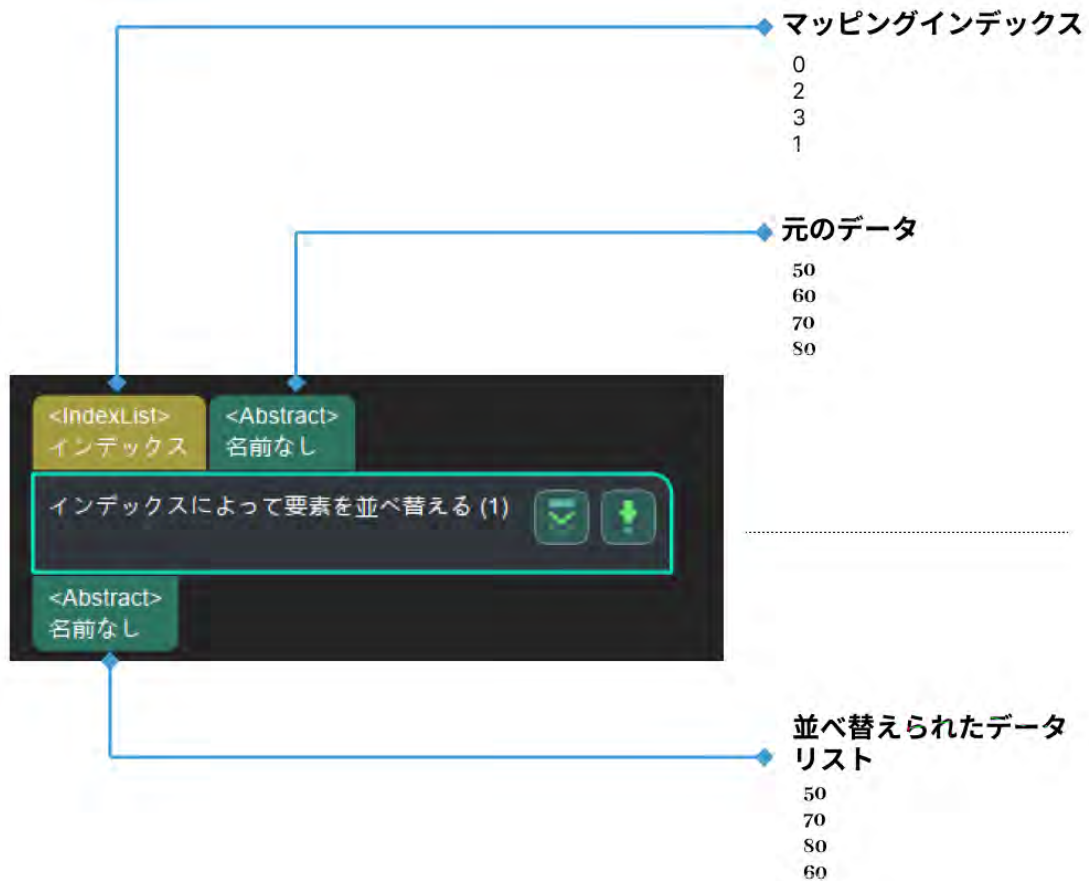
インデックスリストによって元のデータを並べ替えます。



使用シーン

入力インデックスリストによって元のデータを並べ替えます。通常、「ソート」クラスのステップの後に使用されます。

入力と出力



4.3.321. データを繰り返して連結



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

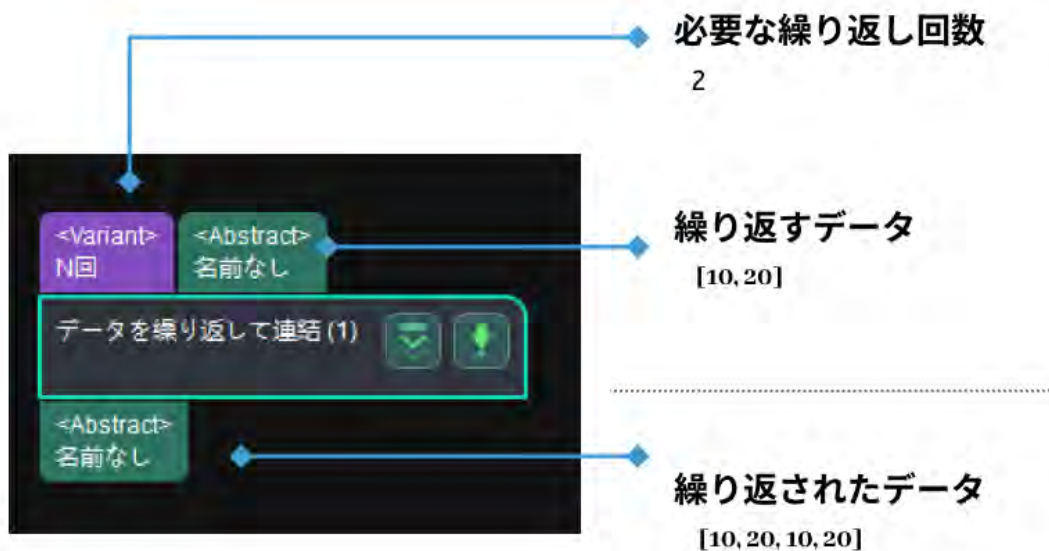
機能

入力データを指定回数だけコピーし、始まりと終わりを繋いで出力します。例えば、元のデータリストが[10, 20]で、2回繰り返す必要がある場合、出力は[10, 20, 10, 20]となります。

使用シーン

汎用のデータコピーのステップです。その後の入力データリストの要素数を等しくするために、一般的にステップ [\[vision-steps:count-elements-in-specified-dimension-in-data-list::count-elements-in-specified-dimension-in-data-list\]](#) と併用されます。

入力と出力



4.3.322. リスト内の要素を置き換える



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

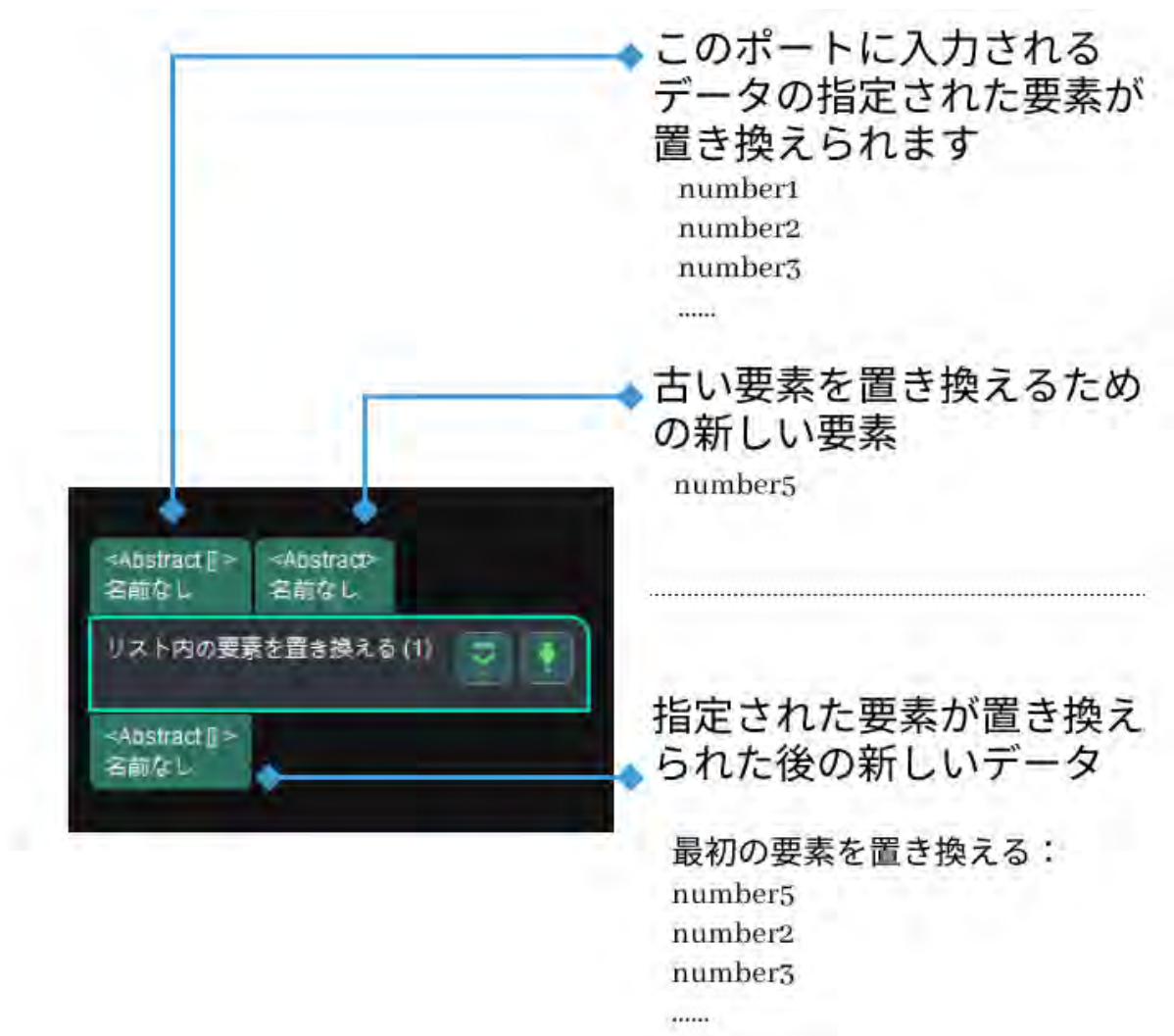
機能

入力データの要素を置き換えます。要素の次元と置き換え方法は、パラメータで設定できます。

使用シーン

開発者のテスト用のステップです。普通のユーザーは、一般的な使用シーンでこのステップを使用する必要はありません。

入力と出力



4.3.323. リストを逆順に並べる



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

データリスト内の要素を逆順に並べます。

使用シーン

汎用のデータソートの関連ステップです。特定の使用シーンはありません。

入力と出力

● 入力：

1. このポートに入力されるデータリストは逆順に並べられます。

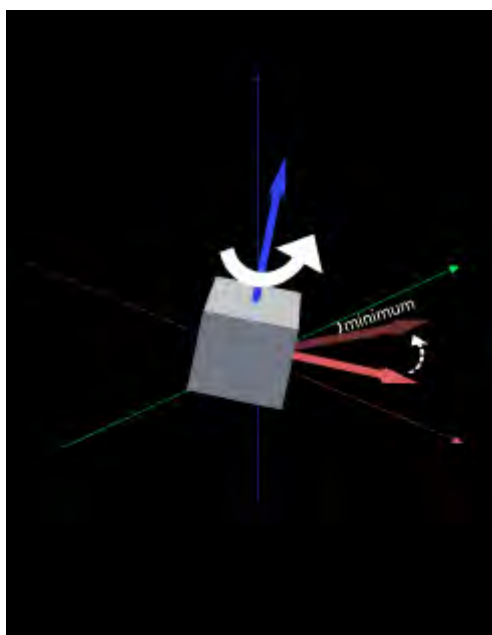
● 出力：

1. 逆順に並べられたデータリスト。

4.3.324. 指定軸が基準方向と最小角度になるように位置姿勢を調整

機能

このステップは、位置姿勢のある軸を中心に、もう一つの軸を基準方向との角度が最小になる方向に調整することができます。

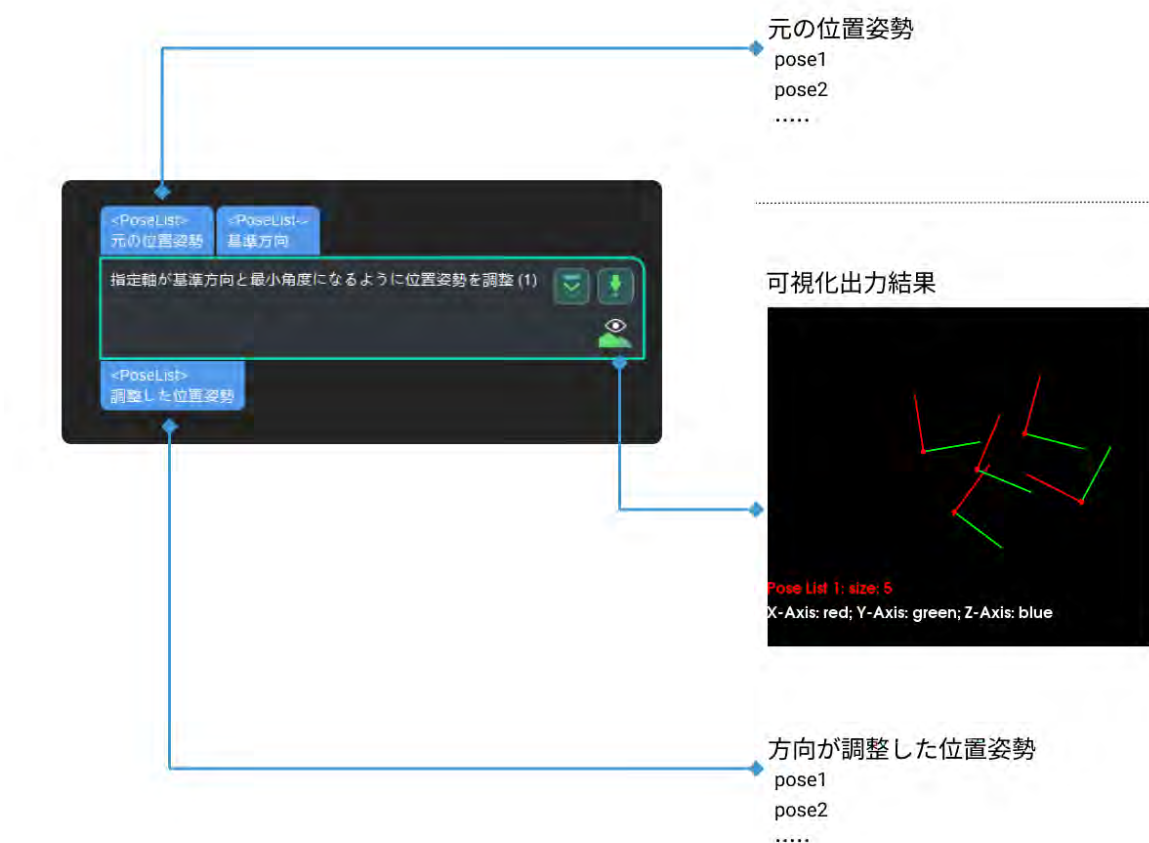


使用シーン

通常、位置姿勢の方向の調整が必要な各種のシーンに適用されます。

このステップは古いバージョンであり、更に機能が揃っている新しいバージョンのステップ [\[vision-steps:rotate-poses-to-directions-freely::rotate-poses-to-directions-freely\]](#) を使用し、ステップ [3Dベクトルをクイック作成](#) と組み合わせて使用してください。

入力と出力



パラメータの説明

可視化設定

位置姿勢表示の設定

パラメータ説明：このパラメータは、位置姿勢の表示タイプを設定するために使用されます。

オプション：

- 入力位置姿勢のみを表示
- 出力位置姿勢のみを表示
- 入力と出力位置姿勢をすべて表示

初期値：出力位置姿勢のみを表示。

基準点を使用

パラメータ説明：このパラメータは、方向の計算に基準点を使用するかどうかを選択するために使用されます。チェックを入れた後、**基準点の設定** パラメータ群を設定する必要があります。

初期値：チェックを入れない。

調整説明：基準点を使用して基準方向を計算する場合にのみチェックを入れます。構築される方向は、入力位置姿勢から基準点までの方向となります。

基準点を使用

相対Z値を使用

パラメータ説明：このパラメータは、基準点のZ軸を置き換えるかどうかを選択するために使用されます。基準点のZ値＝元の位置姿勢のZ値＋相対値。

初期値：チェックを入れない。

X

パラメータ説明：このパラメータは基準点のX値の座標を表します。

初期値：0.0000

Y

パラメータ説明：このパラメータは基準点のY値の座標を表します。

初期値：0.0000

Z

パラメータ説明：このパラメータは基準点のZ値の座標を表します。

初期値：0.0000

相対Z値

パラメータ説明：このパラメータは、Z方向における並進ベクトルを表します。**相対Z値を使用**にチェックを入れた場合に調整する必要があります。

初期値：0.0000

軸の設定

固定軸の方向

パラメータ説明：このパラメータは、位置姿勢の回転時に固定的な軸を指定するために使用されます。

オプション：X、Y、Z

初期値：Y

回転する軸の方向

パラメータ説明：このパラメータは、指定方向に回転する軸を選択するために使用されます。

オプション：X、Y、Z

初期値：Z

基準方向の設定

X

パラメータ説明：このパラメータは、基準軸Xの方向ベクトルを表します。

初期値：0.0000

Y

パラメータ説明：このパラメータは、基準軸Yの方向ベクトルを表します。

初期値：0.0000

Z

パラメータ説明：このパラメータは、基準軸Zの方向ベクトルを表します。

初期値：1.0000

4.3.325. 指定した位置姿勢によって画像を回転



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

入力される3D位置姿勢によって、2D画像を指定方向に回転させます。

使用シーン

対象物を位置決めした後、対象物に対応する画像を回転させ、それが元の画像の位置とほぼ同じように見えるために使用されます。

入力と出力

● 入力：

1. このポートに入力される対象物の画像が回転され、指定位置に配置されます。
2. 画像を配置するための対象物の位置姿勢。

● 出力：

1. 配置後の画像。

4.3.326. 指定した軸と角度を中心に位置姿勢を回転



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

指定軸を中心に、入力位置姿勢を指定角度に回転させます。

使用シーン

汎用の位置姿勢回転ステップです。特定の使用シーンはありません。

入力と出力

● 入力：

1. このポートに入力される位置姿勢は回転されます。
2. 位置姿勢はこのポートに入力された軸を中心に回転されます。
3. 位置姿勢の回転角度。

● 出力：

1. 回転された位置姿勢。

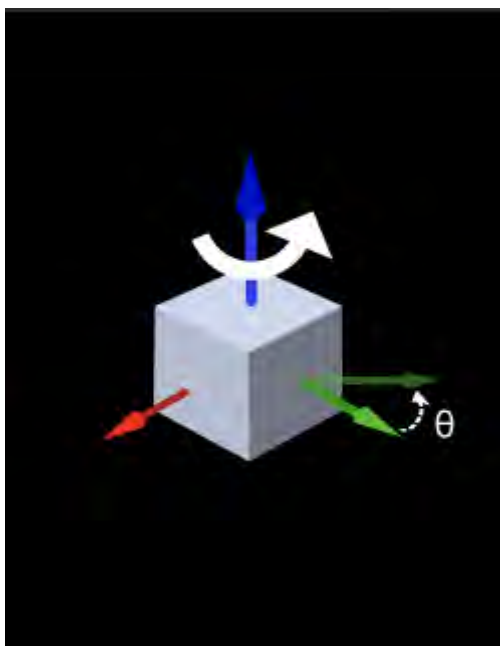
4.3.327. 指定された軸を中心に位置姿勢を回転



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

指定された軸を中心に位置姿勢を回転します。

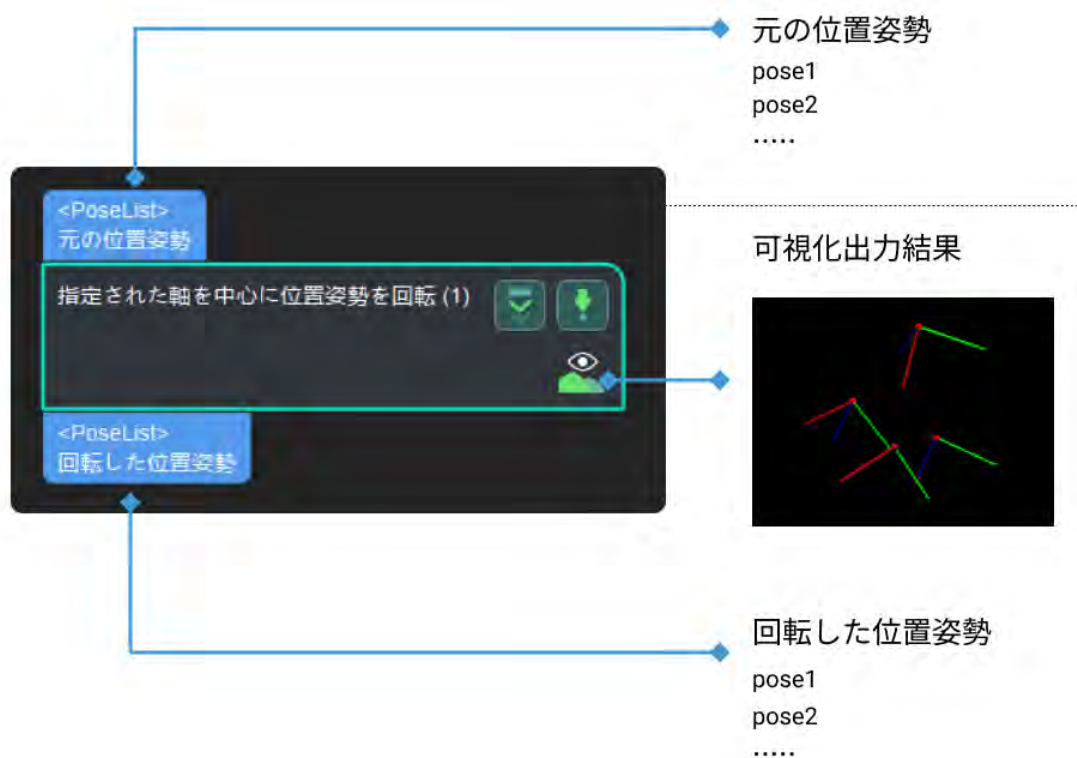


使用シーン

汎用位置姿勢調整ステップです。特定使用シーンはありません。

このステップは古いバージョンであり、新しいバージョンのステップ [\[vision-steps:rotate-poses-around-axis-by-angle:::rotate-poses-around-axis-by-angle\]](#) を使用して 3Dベクトルを [クイック作成](#) と併用してください。

入力と出力



4.3.328. ローカル座標系で位置姿勢を回転



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社 (docs@mech-mind.net) までお問い合わせください。

機能

対象物座標系で指定軸を中心に位置姿勢を回転させます。

使用シーン

汎用の位置姿勢回転ステップです。特定の使用シーンはありません。

入力と出力

- **入力：** このポートに入力される位置姿勢は回転されます。
- **出力：** 回転された位置姿勢。

4.3.329. 位置姿勢を目標方向に回転（制限なし）



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

指定軸の方向を基準方向に合わせるように、入力された元の位置姿勢を回転させます。対象物の対称角度は、入力から取得できます。または、パラメータで設定できます。

使用シーン

汎用の位置姿勢回転ステップです。特定の使用シーンはありません。

入力と出力

- **入力：**
 1. このポートに入力される位置姿勢は回転されます。
 2. 位置姿勢を回転するための目標方向。
- **出力：**
 1. 回転された位置姿勢。

4.3.330. 対称性制約で位置姿勢を指定方向に回転



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

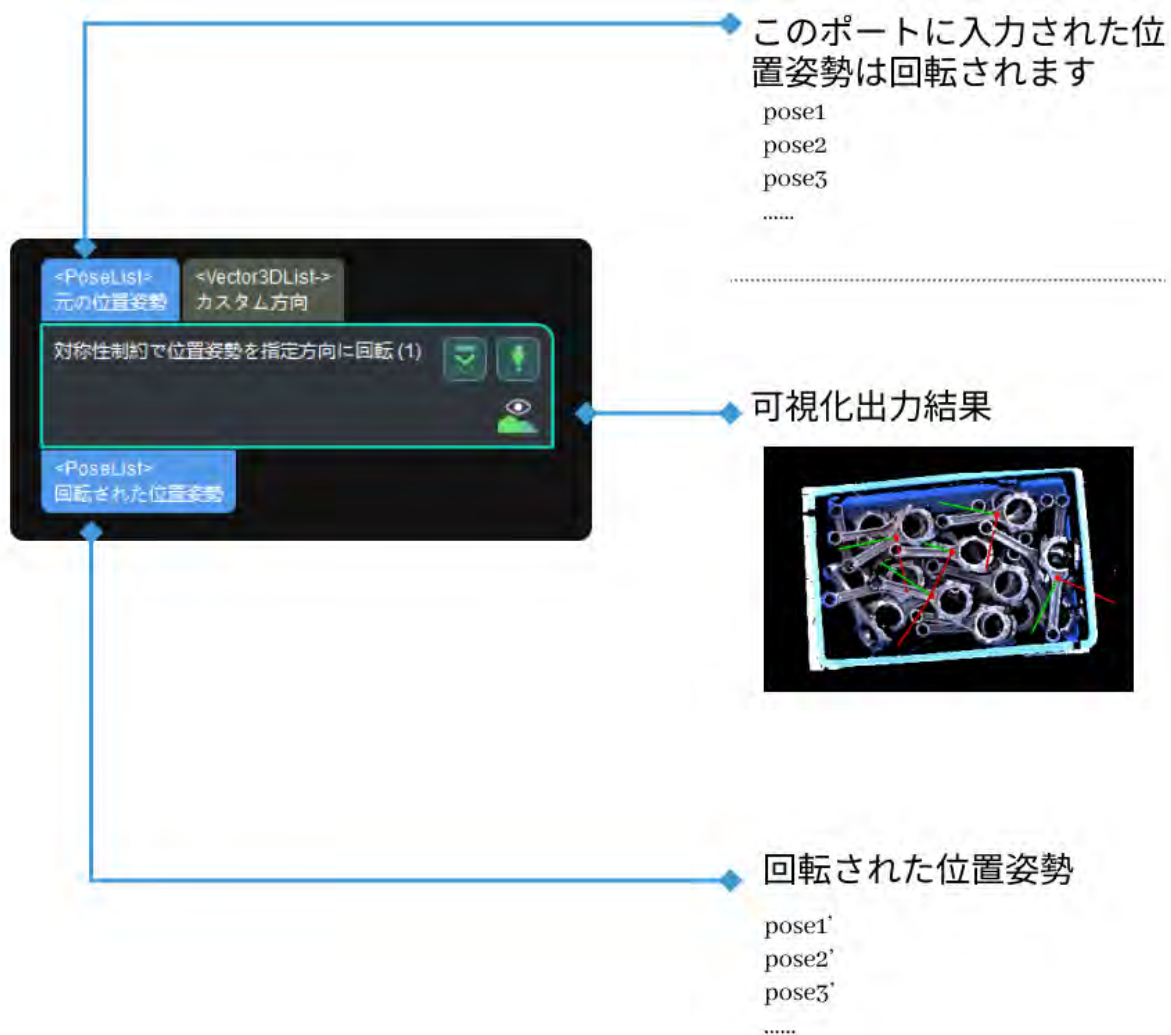
機能

指定軸の方向を基準方向に合わせるように、入力された元の位置姿勢を回転させます。対象物の対称角度は、パラメータで設定できます。

使用シーン

汎用の位置姿勢回転ステップです。特定の使用シーンはありません。

入力と出力



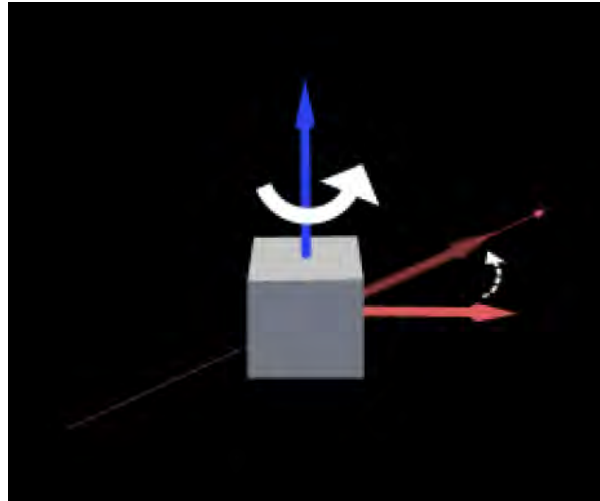
4.3.331. 位置姿勢の軸を指定方向に回転



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

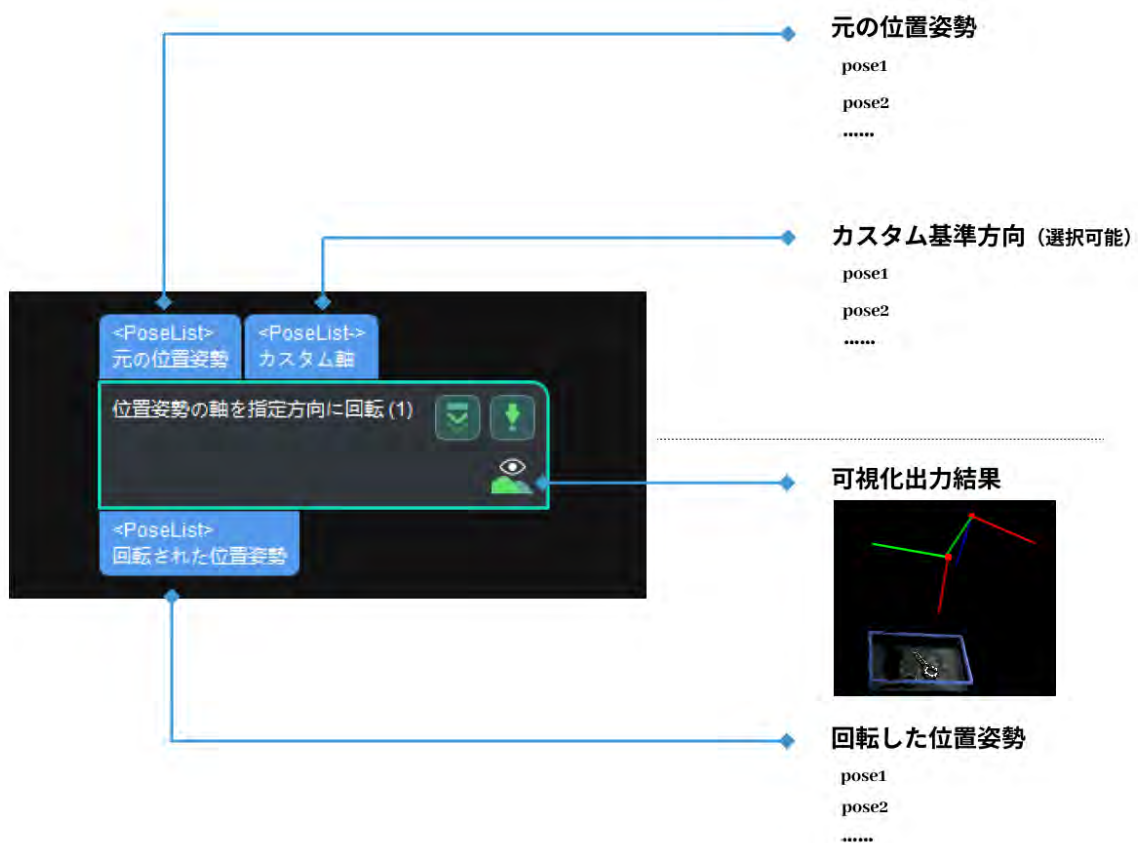
指定軸をカスタマイズの方向に合わせるように、固定軸を中心に位置姿勢を回転させます。



使用シーン

汎用の位置姿勢回転ステップで、特定の使用シーンはありません。

入力と出力



4.3.332. 画像保存

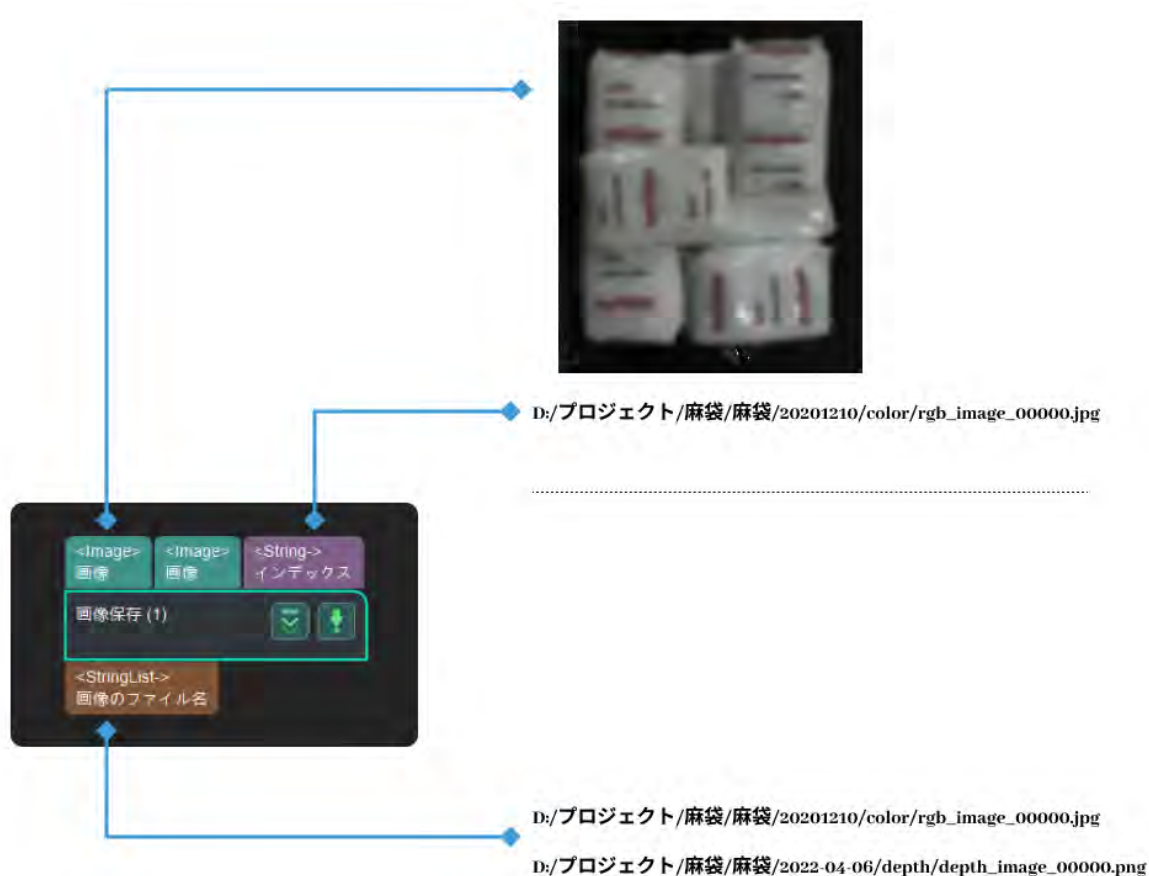
機能

カラー画像と深度画像を指定された場所に保存します。

使用シーン

通常、カメラでキャプチャした元のカラー画像と元の深度画像を保存するために使用されます。また、インスタンスセグメンテーションのグラフィックなどを保存するためにも使用できます。必要に応じて **保存ディレクトリ** や **サブフォルダ名** などのパラメータを変更できます。

入力と出力



パラメータの説明

ディレクトリ設定

保存ディレクトリ

パラメータ説明：このパラメータは、画像を保存するためのディレクトリを選択するために使用されます。

調整説明：実際のニーズに応じて選択してください。

対象物の名前

パラメータ説明：このパラメータは、対象物の名前を持つサブフォルダを作成するために使用されます。

初期値：なし。

サブフォルダ名

パラメータ説明：このパラメータは、異なる形式の画像フォルダを設定するために使用されます。フォルダとフォルダ間に「;」で区切る必要があります。

形式：XXXX;XXXX;XXXX;XXXX

初期値：depth;color

日付関連設定

日付サブフォルダを作成

パラメータ説明：このパラメータは、現在の日付で命名されたサブファイルを作成するかどうかを選択するために使用されます。

初期値：チェックを入れる。



日付サブフォルダを作成にチェックを入れない場合、**最大サブフォルダ数**と**時限切れのフォルダを自動削除**を設定する必要はありません。

最大サブフォルダ数

パラメータ説明：このパラメータは、サブフォルダを保存できる最大日数を設定するために使用されます。

初期値：7day

時限切れのフォルダを自動削除

パラメータ説明：このパラメータは、**最大サブフォルダ数**を超えたフォルダを削除するかどうかを設定するために使用されます。

初期値：チェックを入れる。

保存内容の設定

カメラパラメータを保存

パラメータ説明：このパラメータは、カメラパラメータをフォルダに保存するかどうかを選択するために使用されます。チェックを入れると、画像保存中にカメラパラメータも保存されます。

初期値：チェックを入れる。

調整説明：初期値を使用することをお勧めします。

フランジ位置姿勢を保存

パラメータ説明：このパラメータは、フランジ位置姿勢をフォルダに保存するかどうかなを選択するために使用されます。チェックを入れると、画像保存中にフランジ位置姿勢も保存されます。

初期値：チェックを入れる。

調整説明：初期値を使用することをお勧めします。

仮想カメラ使用時に保存しない

パラメータ説明：このパラメータは、仮想カメラのデータを保存するかどうかなを選択するために使用されます。チェックを入れると、プロジェクトに仮想カメラを使用する場合、入力された画像が保存されません。

初期値：チェックを入れる。

調整説明：初期値を使用することをお勧めします。

画像インデックス設定

同名画像を自動上書き

パラメータ説明：このパラメータは、同じ名前の画像を上書きするかどうかを設定するために使用されます。チェックを入れると、同じ名前の画像が自動的に上書きされます。チェックを入れないと、画像を上書きするかどうかを確認するためのポップアップウィンドウが表示されます。

初期値：チェックを入れる。

最大画像数

パラメータ説明：このパラメータは、保存する画像の最大数を設定するために使用されます。画像数が最大値を超えた場合、次の画像番号は0にリセットされます。

初期値：1000

設定可能な範囲：1以上の整数。

開始番号をカスタマイズ

パラメータ説明：このパラメータは、画像保存時の開始番号を設定するために使用されます。

初期値：-1

ファイル名設定

カラー画像をPNG形式で保存

パラメータ説明：このパラメータは、カラー画像を **.png** 形式で保存するかどうかを設定するために使用されます。

初期値：チェックを入れない。

画像ファイル名の接頭辞をカスタマイズ

パラメータ説明：このパラメータは、画像ファイル名の接頭辞を設定するために使用されます。

初期値：なし。

4.3.333. 位置姿勢の近傍領域を3D ROIとして保存



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

入力位置姿勢の近傍領域を、3D ROI形式のファイルに保存します。

使用シーン

古いバージョンの3D ROI形式のファイル生成ステップです。

これは古いバージョンのステップであり、メンテナンスが停止されますので、新しいバージョンの3D ROIの関連するステップを使用してください。

入力と出力

- **入力：** このポートに入力される位置姿勢の近傍領域は、3D ROI形式のファイルとして保存されます。
- **出力：** なし。

4.3.334. 結果をXMLファイルに保存



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

ほかのステップの出力結果を、XML形式のファイルに保存します。ファイル名は、パラメータで設定できます。

使用シーン

特定のシーン向けのカスタマイズのステップです。

入力と出力

- **入力：**

1. このポートに入力される位置姿勢は、保存されます。
2. このポートに入力される位置姿勢は、保存されます。
3. このポートに入力される文字列は、保存されます。
4. このポートに入力される数値は、保存されます。
5. このポートに入力される文字列は、保存されます。
6. このポートに入力される文字列は、保存されます。

● 出力：

1. なし。

4.3.335. 結果をファイルに保存



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社 (docs@mech-mind.net) までお問い合わせください。

機能

プロジェクトのデータをローカルファイルに保存します。

使用シーン

このステップは全てのタイプのデータを保存しますが、2D画像を保存する場合、ステップ [\[vision-steps:save-images:::save-images\]](#) を使用することをお勧めします。

入力と出力



4.3.336. 経路点を保存



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

経路点を、JSONファイルに保存します。

使用シーン

汎用の経路点の保存ステップです。特定の使用シーンはありません。

入力と出力

- **入力：** このポートに入力される経路点は、保存されます。
- **出力：** なし。

4.3.337. ステップパラメータをファイルに保存

機能

指定されたステップのパラメータをローカルファイルに保存します。

使用シーン

このステップはその他のステップのパラメータをエクスポートするために使用され、前にステップ [\[vision-steps:save-images:::save-images\]](#) を使用する必要があります。

入力と出力

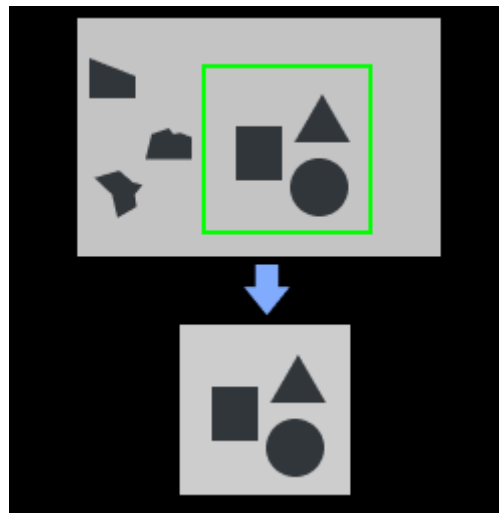


出力：無し

4.3.338. 2D ROI内の画像をスケーリング

機能

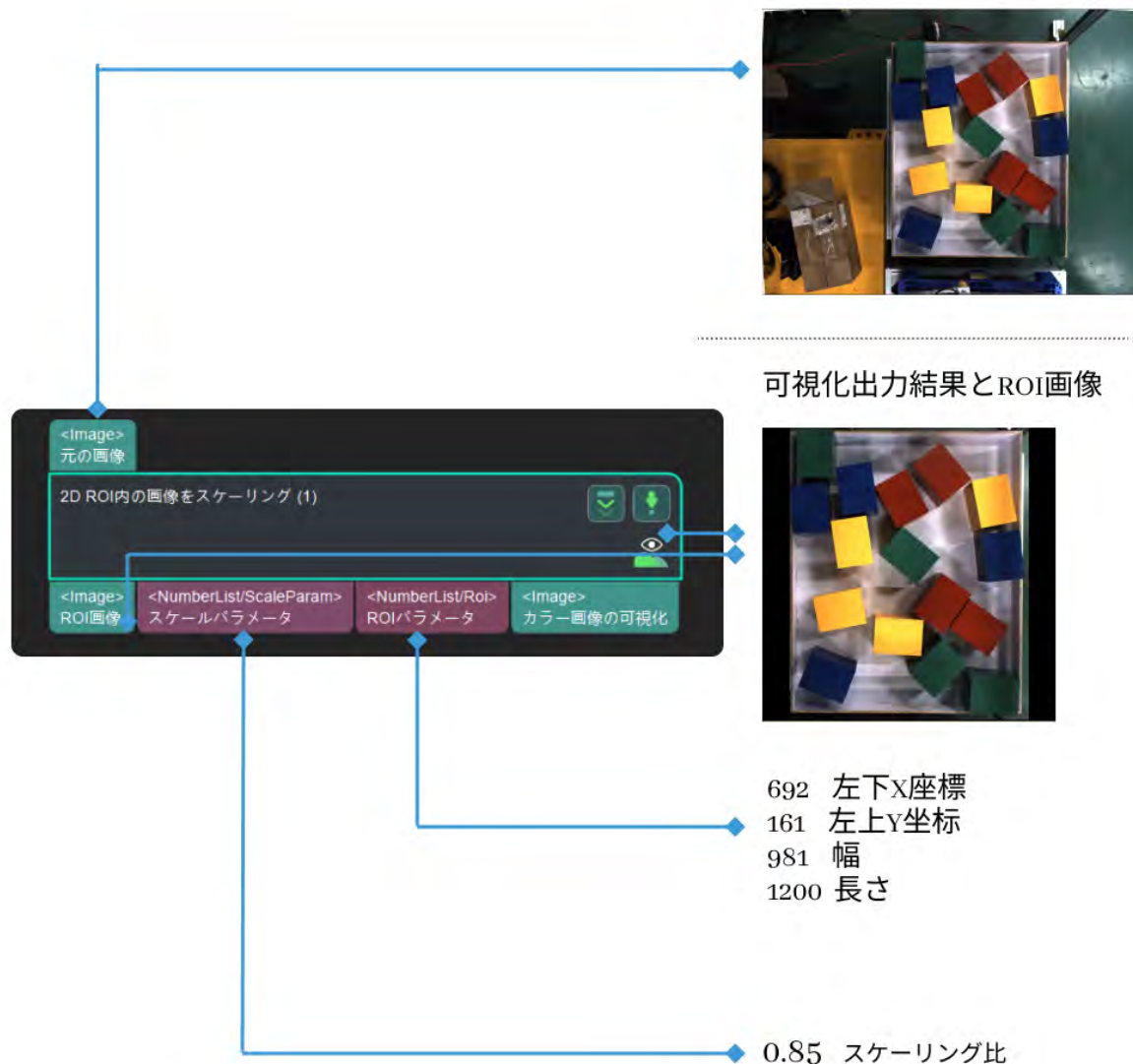
画像の関心領域を指定サイズにスケーリングします。



使用シーン

ディープラーニングの関連ステップの画像前処理に使用されます。通常、ディープラーニングの関連ステップと [2D ROI画像のスケール復元](#) と併用されます。

入力と出力



使用方法

2D ROI 内の画像のスケーリングは、関心領域の画像を指定サイズに拡大・縮小して表示するために使用されます。**カラー画像のROI設定**と**出力画像の理想的な長さ**と**幅の設定**を調整することで、必要なサイズを出力できます。

関心領域の画像を自動的に拡大・縮小

関心領域の画像を自動的に拡大・縮小するには、次の手順を実行します。

1. カラー画像のROIスケーリング設定の**自動スケーリング**にチェックを入れます。
2. 次に**出力画像の理想的な長さ**と**幅の設定**の入力カラー画像ROIと同じ長さ

元の関心領域の画像を出力

元の関心領域の画像を出力するには、次の手順を実行します。

1. カラー画像のROIスケーリング設定の **自動スケーリング** のチェックを外します。
2. **スケーリング係数** を1に設定します。
3. 次に **出力画像の理想的な長さとの幅の設定** の **入力カラー画像ROIと同じ長さとの幅（スケーリングなし）** にチェックを入れます。

指定した倍率で関心領域の画像を出力

指定した倍率で関心領域の画像を出力するには、次の手順を実行します。

1. カラー画像のROIスケーリング設定の **自動スケーリング** のチェックを外します。
2. 実際の状況に応じて **スケーリング係数** を設定してください。このパラメータが1より大きい場合、画像は拡大され、1より小さい場合、画像は縮小されます。
3. 次に **出力画像の理想的な長さとの幅の設定** の **入力カラー画像ROIと同じ長さとの幅（スケーリングなし）** にチェックを入れます。

指定された長さまたは幅の関心領域の画像を出力

指定された長さまたは幅の関心領域の画像を出力するには、次の手順を実行します。

1. **出力画像の理想的な長さとの幅の設定** の **入力カラー画像ROIと同じ長さとの幅（スケーリングなし）** のチェックを入れます。
2. 実際の状況に応じて **幅** と **長さ** を設定します。
3. その後、実際の状況に応じて **塗りつぶし** にチェックを入れるかどうかを設定します。



パラメータについては、対応するパラメータ説明をご参照ください。

パラメータの説明

カラー画像のROI設定

パラメータ説明：このパラメータは、カラー画像ROIの設定方法を背ていするために使用されません。

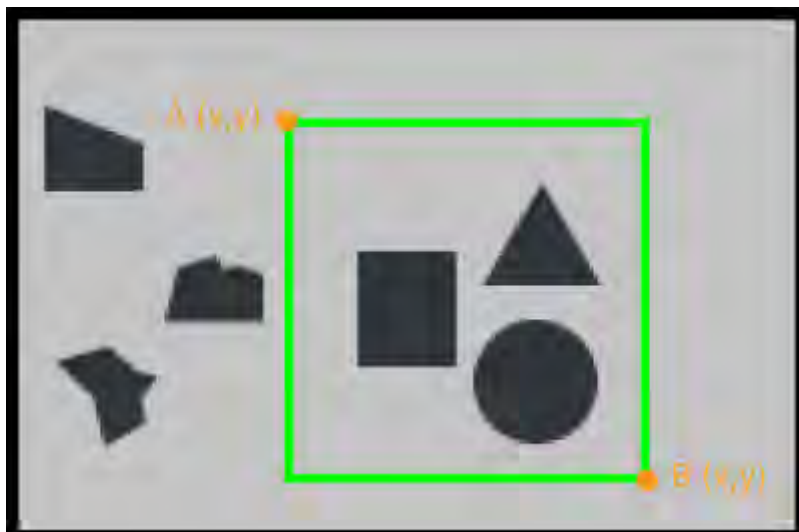
オプション：

- ROIByFile：初期値となります。関心領域を可視化で設定し、左上頂点と右下頂点の座標で関心領域を決定します。
- ROIByParam：パラメータを入力して関心領域を決定します。パラメータは左上頂点の座標と関心領域の幅・高さです。

ROIByFile

カラー画像ROIファイル

パラメータ説明：このパラメータは、カラー画像の関心領域を設定するために使用されます。詳細については、[2D ROIを設定](#) をご参照ください。下図では、Aは左上頂点、Bは右下頂点となります。



元の画像の左上隅の座標は (0, 0) となります。

ROIByParam

ROIの開始X座標

パラメータ説明：関心領域の左上頂点のX軸の座標です。

初期値：0

調整説明：ROIの開始X座標+幅<元の画像の幅

ROIの開始Y座標

パラメータ説明：関心領域の左上頂点のY軸の座標です。

初期値：0

調整説明：ROIの開始Y座標+高さ<元の画像の高さ

幅

パラメータ説明：関心領域の幅です。

初期値：0

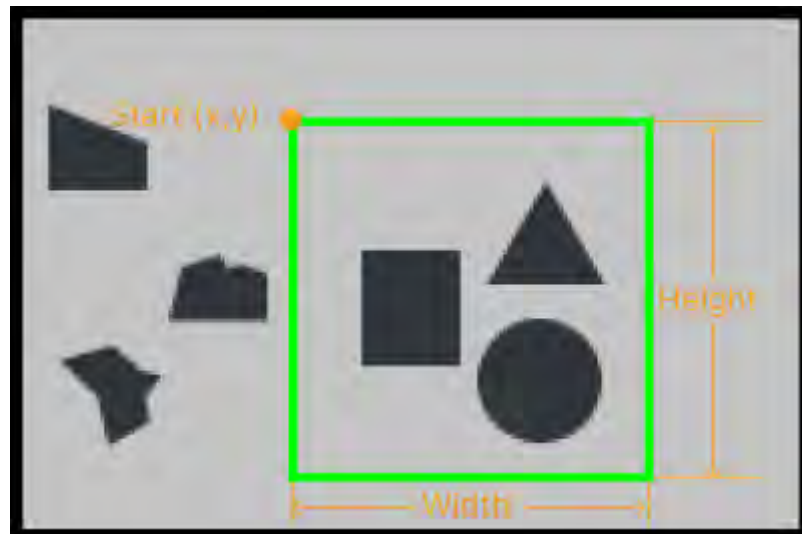
調整説明：ROIの開始X座標+幅<元の画像の幅

高さ

パラメータ説明：関心領域の高さです。

初期値：0

調整説明：ROIの開始Y座標+高さ<元の画像の高さ



元の画像の左上隅の座標は (0, 0) となります。

カラー画像のROIを更新する方法

パラメータ説明：このパラメータは、関心領域の画像の形式を指定するために使用されます。

オプション：

Origin：関心領域の元のカラー画像です。

AdaptiveModifiedRoiOriginP：画像の境界線を用いてカラー画像のROIを調整します。画像のゼロ以外の領域の中心をROI中心に適応的に調整します。

BoundRect：入力画像の境界長方形の領域を抽出します。画像のゼロ以外の外接長方形で元画像を切り替えてスケーリング比率を計算します。

カラー画像のROIスケーリング設定

自動スケーリング

パラメータ説明：実際の状況に応じて、関心領域の画像サイズを自動的に調整します。

初期値：チェックを入れる

スケーリング係数

パラメータ説明：このパラメータは、関心領域のスケーリング係数を設定するために使用されます。

調整説明：スケーリング後の画像=関心領域の元のサイズ×スケーリング係数。下図に示すように、左側が元のサイズ、右側がスケーリング係数を0.5に設定した場合の結果です。



自動スケーリングにチェックを入れない場合にのみ設定可能です。

出力画像の理想的な長さとの幅の設定

入力カラー画像ROIと同じ長さとの幅（スケーリングなし）

パラメータ説明：このパラメータは、**カラー画像のROIスケーリング設定**が行われた後の画像と同じ大きさの出力画像を作成するかどうかを設定するためのものです。

初期値：チェックを入れない



このパラメータにチェックを入れない場合、出力画像のサイズはスケーリング後のサイズになります。

幅/高さ

パラメータ説明：このパラメータは、関心領域の幅/高さ（ピクセル単位）を設定するために使用されます。

初期値：1024px

調整説明：実際の状況に応じて設定してください。



出力画像の理想的な長さとの幅の設定にチェックを入れない場合にのみ設定可能です。

画像を目標サイズにパッド



- 塗りつぶし機能は、関心領域の長さとの幅を指定した場合のみ必要です。
- 塗りつぶしにチェックを入れると、R、G、Bの値を設定可能です。

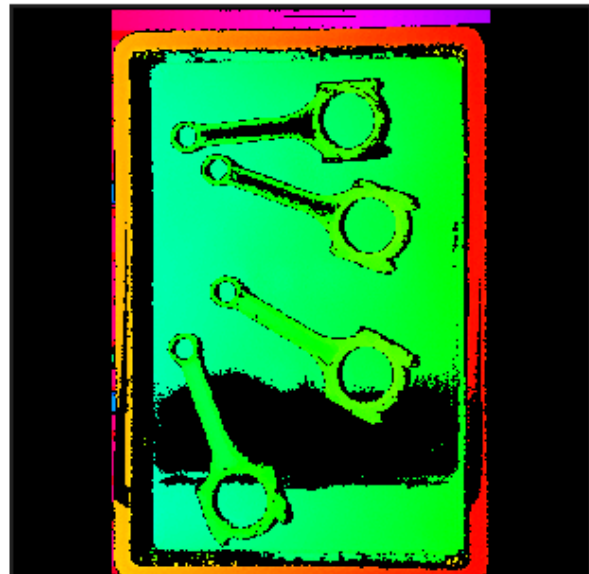
塗りつぶし

パラメータ説明：関心領域の画像が指定された長さ/幅に等倍で拡大された後、余分な領域は塗りつぶし機能で埋められます。

初期値：チェックを入れる

調整説明：通常、黒色(0,0,0)あるいは灰色(128,128,128)を推奨します。

調整の例：塗りつぶし機能を使用しない場合、下図の左側のように関心領域は指定された長さ/幅に等倍で拡大され、塗りつぶし機能を使用する場合、下図の右側のように関心領域は指定サイズでなければならず、残りの領域は塗りつぶし色で塗りつぶされます。



R/G/B値

パラメータ説明：カラー画像はR、G、Bで構成されており、異なるR/G/Bの値を設定することで異なる色を塗ることができます。

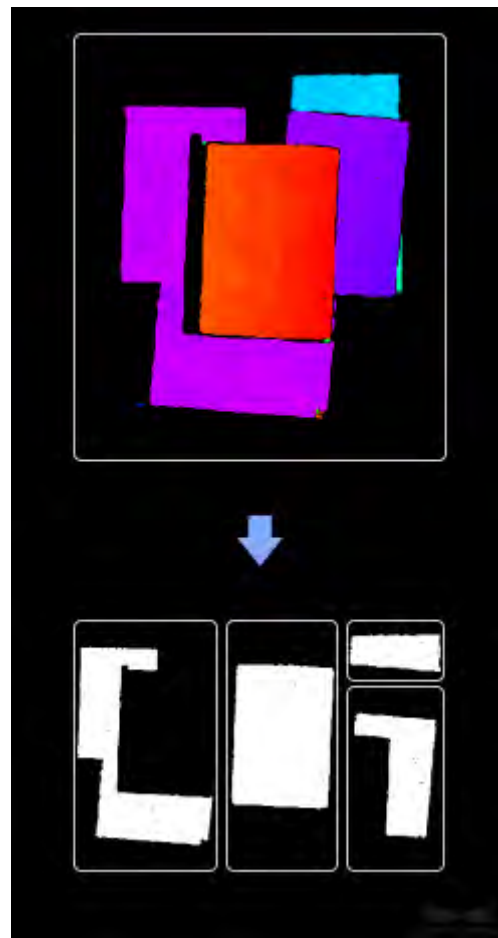
初期値：0

設定可能な範囲：0-255。

4.3.339. 深度画像分割

機能

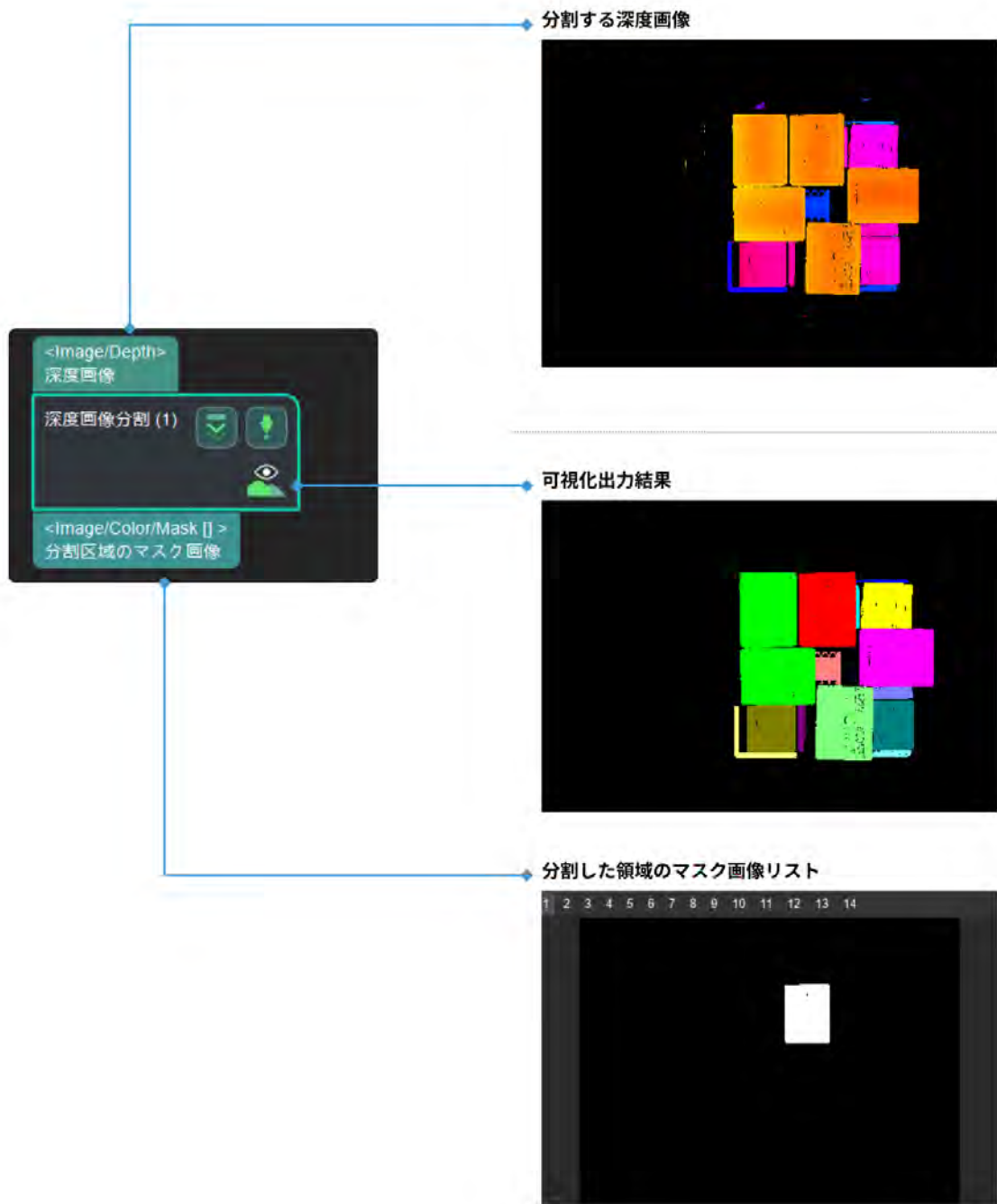
指定されたルールに従って深度画像を分割して、ルールに準拠した領域を出力します。



使用シーン

汎用の深度画像の処理ステップです。通常、[\[vision-steps:invalidate-depth-pixels-outside-3d-roi::invalidate-depth-pixels-outside-3d-roi\]](#)、深度画像で最高領域を取得、マスク画像をマージなどのステップと併用されます。

入力と出力



パラメータの説明

分割設定

隣接するピクセル領域間の最大値の差

パラメータ説明：深度画像を複数の領域に分割する処理において、隣接するピクセル間の最大高さの差（ミリメートル単位）を指定するために使用されます。隣接するピクセルの高さの差がこの値より小さい場合は同じ領域、大きい場合は異なる領域となります。値が大きいほど、より多くの領域が分割されます。

初期値：7.000mm

調整アドバイス：実際の状況に応じて設定してください。

対象物の最小面積

パラメータ説明：このパラメータは、深度画像の分割後に許可される最小領域（ピクセル単位）を指定するために使用されます。この値より少ないピクセル数の領域は除去されます。値が小さいほど残る領域が多くなり、値が大きいほど除去される領域が多くなります。

初期値：1500px

調整アドバイス：実際の状況に応じて設定してください。

深度画像のROIファイル

パラメータ説明：このパラメータは、カラー画像の関心領域を設定するために使用されます。

詳細については、[2D ROIを設定](#) をご参照ください。

4.3.340. 点群をMech-Vizに送信

機能

点群をMech-Vizに送信します。

使用シーン

通常、プロジェクトのデバッグまたはプロジェクトの実際な実行効果を確認するために使用されます。

入力と出力



パラメータの説明

点群設定

入力点群のタイプ

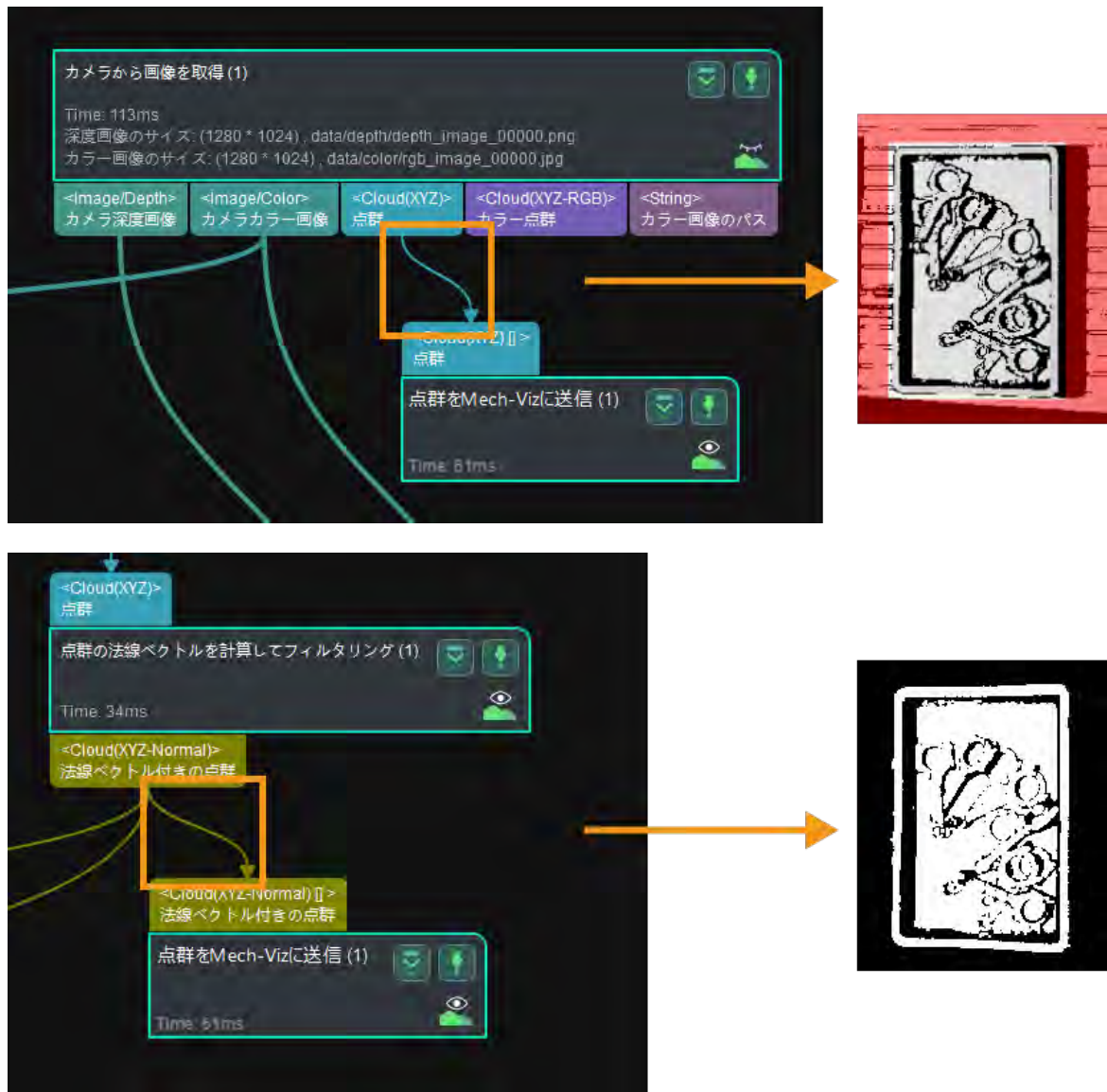
初期値：CloudXYZRGB（カラー点群）

オプション：CloudXYZRGB（カラー点群）、CloudXYZ（点群）、CloudNORMAL（法線ベクトル付きの点群）。

調整説明：実際のニーズに応じて送信する点群のタイプを選択します。

調整の例：CloudXYZRGB、CloudXYZ、CloudNORMALをそれぞれ選択した場合の入力点群のタイプを下図に示します。





対象物情報設定

対象物情報を送信

初期値：チェックを入れない

調整説明：チェックを入れない場合、**物体の位置姿勢**と**位置姿勢の分類ラベル**ポートは表示されません。チェックを入れると、**物体の位置姿勢**と**位置姿勢の分類ラベル**ポートが表示されます。実際のニーズに応じて設定してください。下図に示すように、①はチェックを入れる前の状態で、②はチェックを入れた後の状態です。



すべての入力カメラ座標系にある

初期値：チェックを入れる。

調整説明：点群と位置姿勢がカメラ座標系にある場合はチェックを入れ」ます。チェックを入れると、入力点群情報がロボット座標系に変換されて、Mech-Vizに送信されます。チェックを外すと、入力点群情報がMech-Vizに直接送信されます。

4.3.341. 位置姿勢の四元数を設定



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

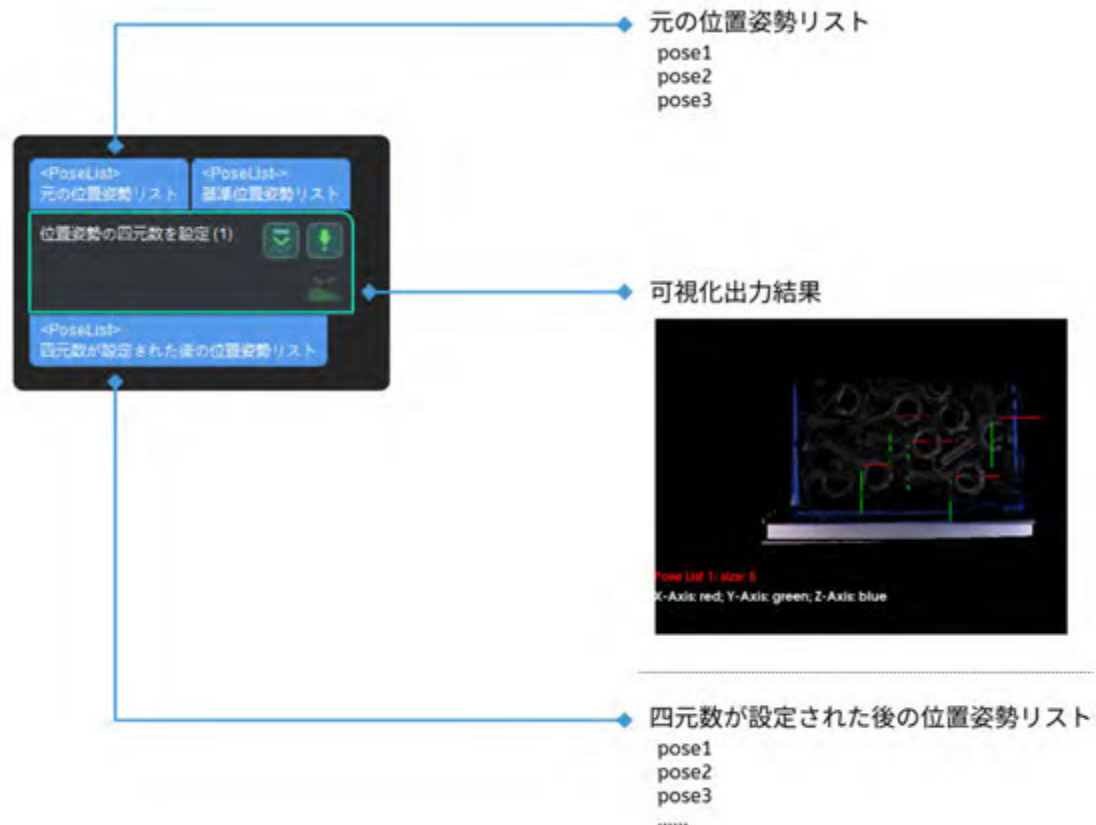
機能

位置姿勢の姿勢（四元数で表示される）を設定するには、2番目のポートに入力された基準位置姿勢もしくはパラメータで設定された四元数を使用できます。

使用シーン

汎用の位置姿勢の四元数設定ステップです。基準位置姿勢リストに複数の位置姿勢が含まれる場合は、最初の位置姿勢の四元数のみが設定のために使われます。

入力と出力



4.3.342. 位置姿勢の並進ベクトルを設定



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

基準位置姿勢もしくはパラメータで設定された基準方向を使用して元の位置姿勢の並進ベクトルを設定します。

使用シーン

汎用の位置姿勢の並進ベクトル設定ステップです。基準位置姿勢リストに複数の位置姿勢がある場合は、最初の位置姿勢の並進ベクトルのみが設定に使用されます。

入力と出力



4.3.343. 画像表示

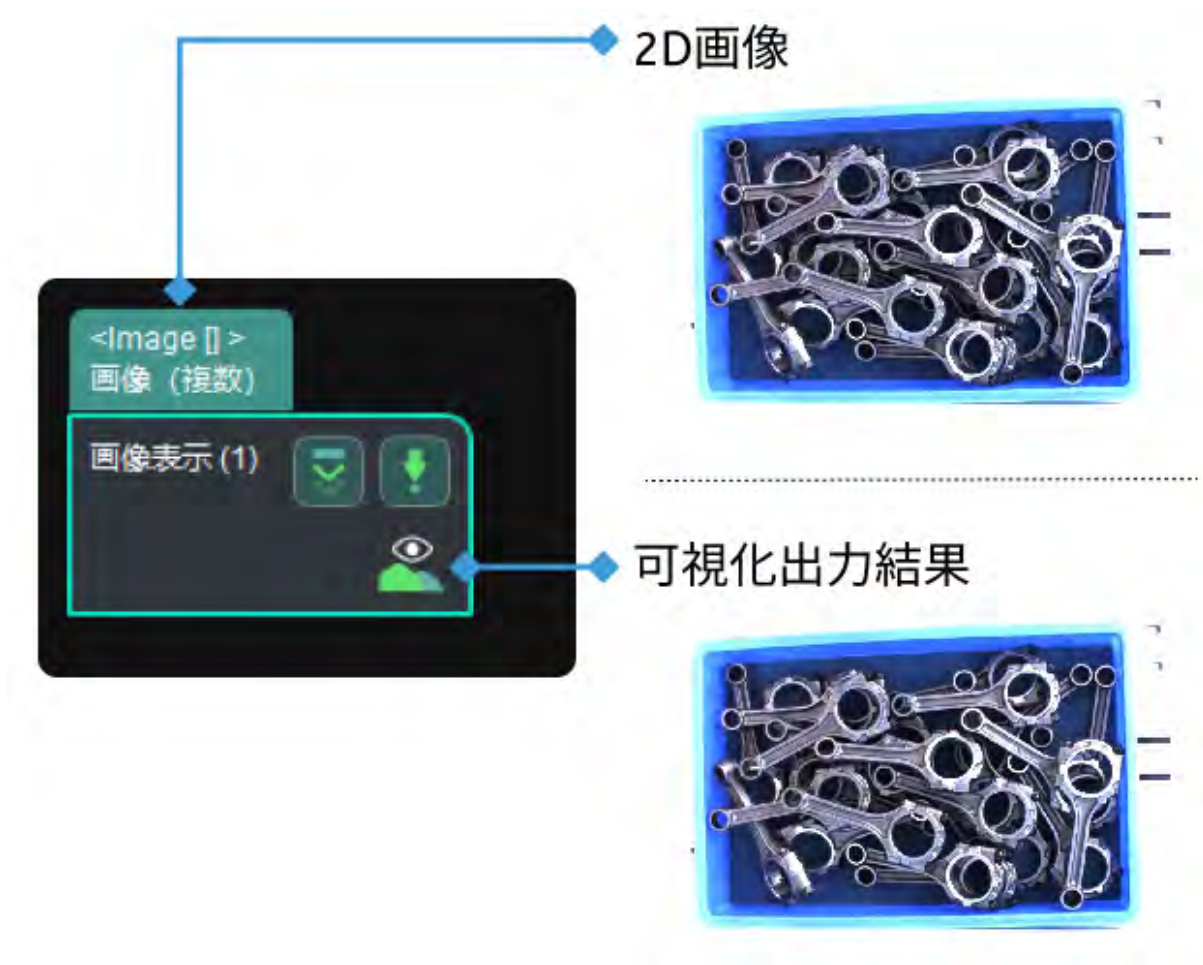
機能

その他のステップに出力された2D画像を見える化します。入力ステップは、**2Dマッチング**、**2D特徴抽出**、**2D汎用処理** などの中に2D画像を出力するステップです。

使用シーン

このステップでは、2D画像の表示効果により、対応する入力ステップのパラメータを画像が要求を満たすまでデバッグするために使用されます。

入力と出力



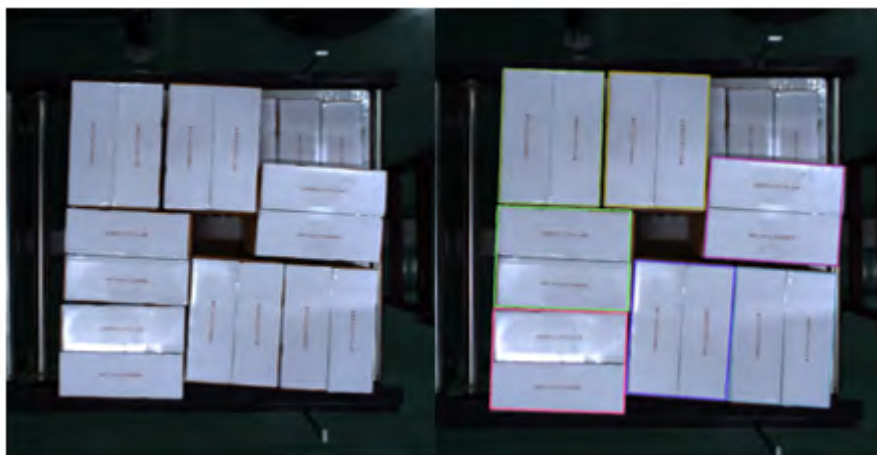
パラメータの説明

表示のタイプ

初期値：Multimages

オプション：Multimages、ImgAndPolyVerts

調整説明：下図の左側に示すように、「表示のタイプ」を **Multimages** に設定する場合は画像のみが表示されます。下図の右側に示すように、「表示のタイプ」を **ImgAndPolyVerts** に設定する場合は画像と候補ポリゴンが表示されます。



4.3.344. 点群と位置姿勢を表示

機能

このステップは、点群と位置姿勢の表示を可視化できます。

類似ステップ：

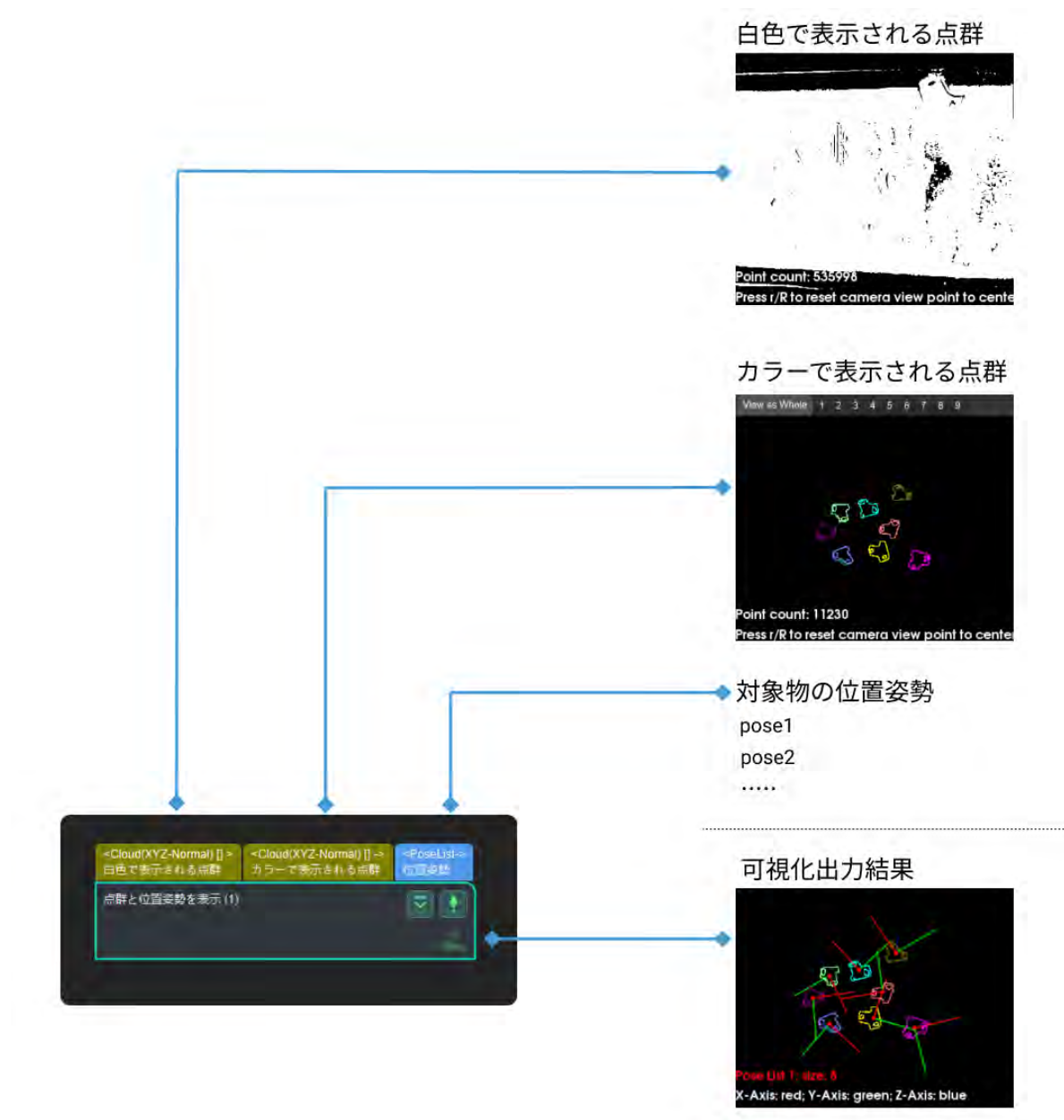


ステップ **点群と位置姿勢を表示** は、点群と位置姿勢を表示するために使用されます。ステップ [\[vision-steps:show-images:::show-images\]](#) は、カラー画像またはマスク画像を表示するために使用されます。

使用シーン

このステップは、位置姿勢と点群との相対位置関係を表示するために使用されます。例えば、対象物を把持するシーンでは、位置姿勢が正しいかを確認するために使用されます。通常はデバッグに使用されます。

入力と出力



パラメータの説明

法線ベクトルの可視化設定

法線ベクトルを表示

パラメータ説明：このパラメータは、可視化出力結果で点群の法線ベクトルを表示するかどうかを決定します。

初期値：チェックを入れない。

調整説明：チェックを入れれば、可視化出力結果で点群の法線ベクトルが表示されます。調整効果を [調整の例](#) に示します。

法線ベクトルの表示間隔

パラメータ説明：法線ベクトルがまばらか密集しているかは、このパラメータの値によって決めます。**法線ベクトル法線をべ表示**にチェックを入れると有効になります。

初期値：20

調整説明：このパラメータは、法線ベクトル間の距離（ミリメートル単位）を設定するために使用されます。この値が大きいほど、法線ベクトルの表示がまばらになります。調整効果を [調整の例](#) に示します。

Z値の可視化設定

値を可視化

パラメータ説明：このパラメータは、点群をグレースケールで表示するために使用されます。

初期値：チェックを入れない。

調整説明：チェックを入れると、Z値によって異なるグレースケールで点群を表示します。

上限

パラメータ説明：色付けるZ方向の最大値です。**値を可視化**にチェックを入れると有効になります。

初期値：0

調整説明：この値は、色を付けるZ方向の最大値を表します。この値が高いほど画像は黒になります。

下限

パラメータ説明：色付けるZ方向の最小値です。**値を可視化**にチェックを入れると有効になります。

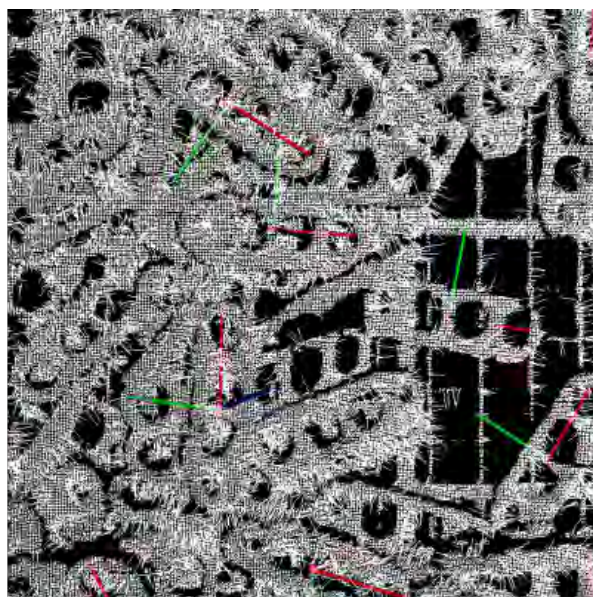
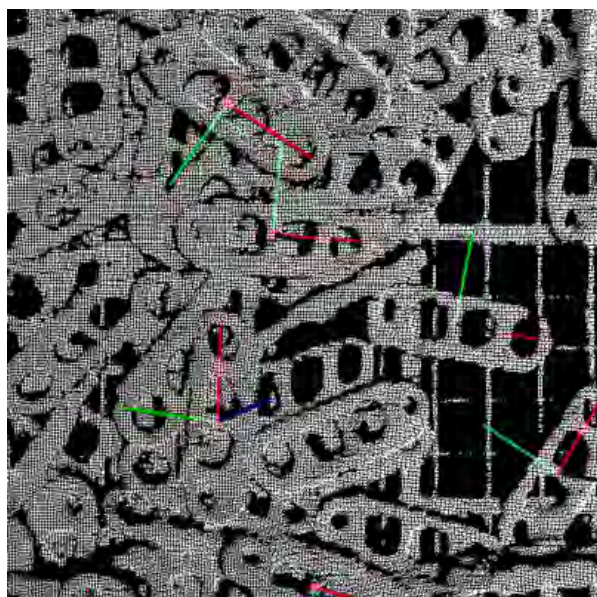
初期値：0

調整説明：この値は、色を付けるZ方向の最小値を表します。この値が低いほど画像は鮮明になります。

調整の例

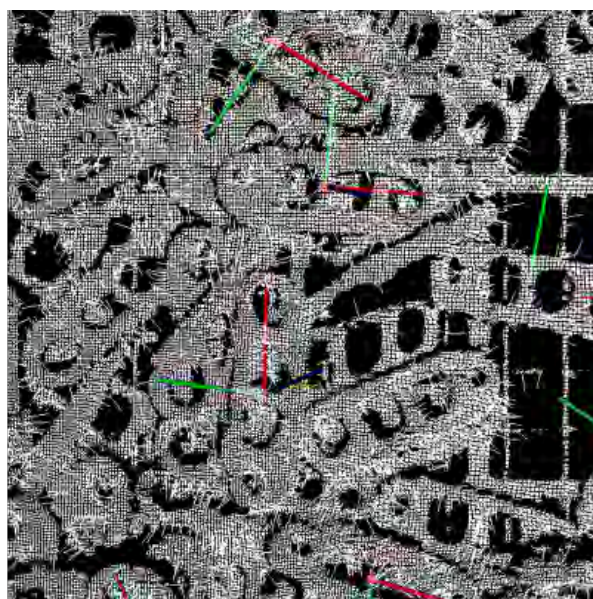
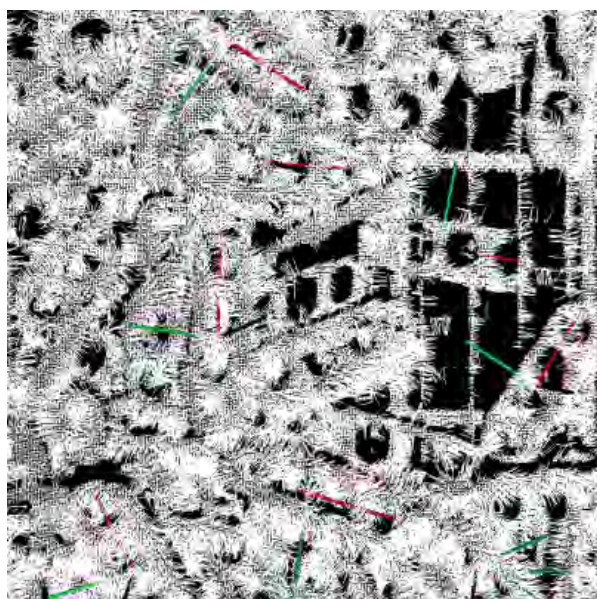
点群の法線ベクトルを表示

下図に示すように、左側は法線ベクトルを表示しない効果で、右側は20mm間隔で法線ベクトルを表示した効果です。



法線ベクトルの表示間隔を変更

下図に示すように、左側は5mm間隔で法線ベクトルを表示した効果で、法線ベクトルは密集しています。右側は40mm間隔で法線ベクトルを表示した効果で、法線ベクトルはまばらです。



4.3.345. 深度画像を平滑化処理



これは古いバージョンのステップであり、メンテナンスが停止されています。このステップに関する詳細が必要な場合は、当社 (docs@mech-mind.net) までお問い合わせください。

機能

設定した方法で深度画像を平滑化します。

使用シーン

通常は、点群が大きく変動する場合に使用されます。

入力と出力

- **入力**：このポートに入力される深度画像は平滑化されます。
- **出力**：平滑化された画像。

4.3.346. 経路を平滑化



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

位置姿勢リストの平滑化処理によって形成された運動経路です。

使用シーン

ロボットの激しい変動を伴う移動を防ぐために使用されます。

入力と出力

- **入力**：このポートに入力された位置姿勢リストが形成された運動経路は、平滑化されます。
- **出力**：平滑化された運動経路の位置姿勢リスト。

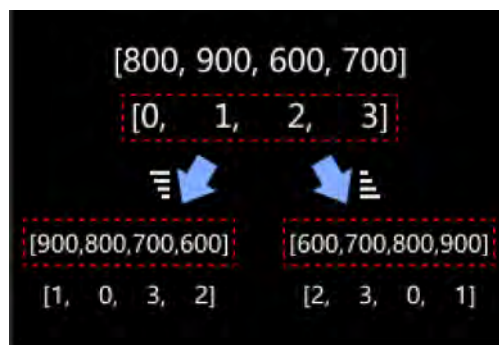
4.3.347. ソートしてインデックスリストを出力



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

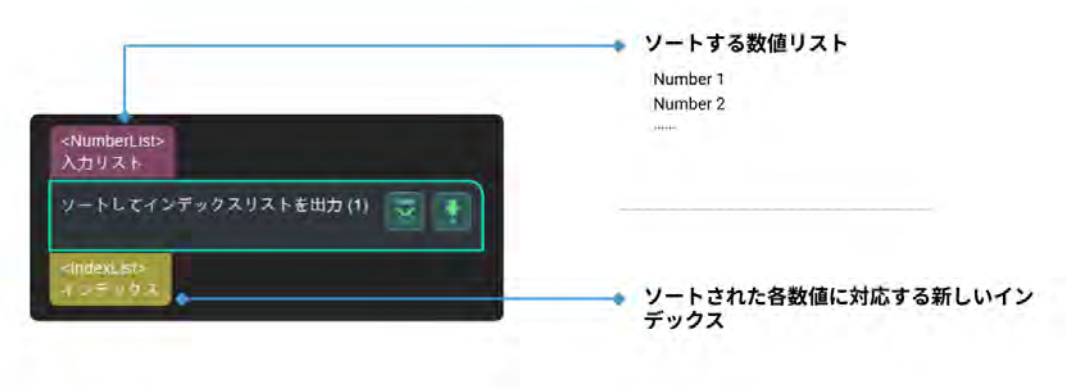
入力リストを昇順または降順でソートし、ソートに従ってインデックスリストを出力します。



使用シーン

様々なタイプのデータを昇順/降順でソートするシーンに使用されます。通常、[\[vision-steps:reorder-by-index-list::reorder-by-index-list\]](#) ステップと併用されます。

入力と出力



4.3.348. ソートと階層化

機能

入力データを指定されたルールに従ってソート・階層化し、ソート・階層化処理によってデータの元のインデックスを移動し、処理後のデータ、インデックス、ソート・階層化された指数を出力します。

入力が多次元データの場合、データの最後の次元の各配列が順番に独立して処理されます。

使用シーン

多次元ソートのために使用されます。このステップを他のステップと併用することで指定した次元をソートすることができます。

入力と出力

入力の処理するデータを以下の表に示します。

処理するデータ	[[30, 10, 20]], [20, 10]]
データの元のインデックス	[[4, 5, 6]], [3, 2]]

処理後の出力データを以下の表に示します。

処理後のデータ	\[[[10], [20], [30]], \[[[10], [20]]]
処理後のデータのインデックス	\[[[5], [6], [4]], \[[[2], [3]]]
ソートと階層化インデックス	\[[[1], [2], [0]], \[[[1], [0]]]

パラメータの説明

昇順

パラメータ説明：このパラメータは、ソート時にデータを昇順または降順でソートするかを指定するために使用されます。このオプションにチェックを入れると、データは昇順にソートされ、そうでない場合は降順にソートされます。

初期値：チェックを入れる。

階層間隔

パラメータ説明：データを階層化する際の階層間隔を指定するために使用されます。

初期値：50.0mm

各層の開始位置

パラメータ説明：このパラメータは、データをソート・階層化する際に、各層の開始位置を指定するために使用されます。

オプション：前層の範囲の最後、この層の最初の数。

- 前層の範囲の最後：各層の範囲は最初から最後まで接続され、データには依存しません。層間にはギャップはありません。
- この層の最初の数：各層の範囲がどこにあるかは、データによって異なります。層間にはギャップがあり、データの値が互いに大きく離れているほど大きくなります。

初期値：前層の範囲の最後。

4.3.349. 点群をソート



これは古いバージョンのステップであり、メンテナンスが停止されますので、新しいバージョンのステップ [\[vision-steps:calc-specified-property-of-point-clouds:::calc-specified-property-of-point-clouds\]](#) を使用してください。このステップに関する詳細が必要な場合は、当社 (docs@mech-mind.net) までお問い合わせください。

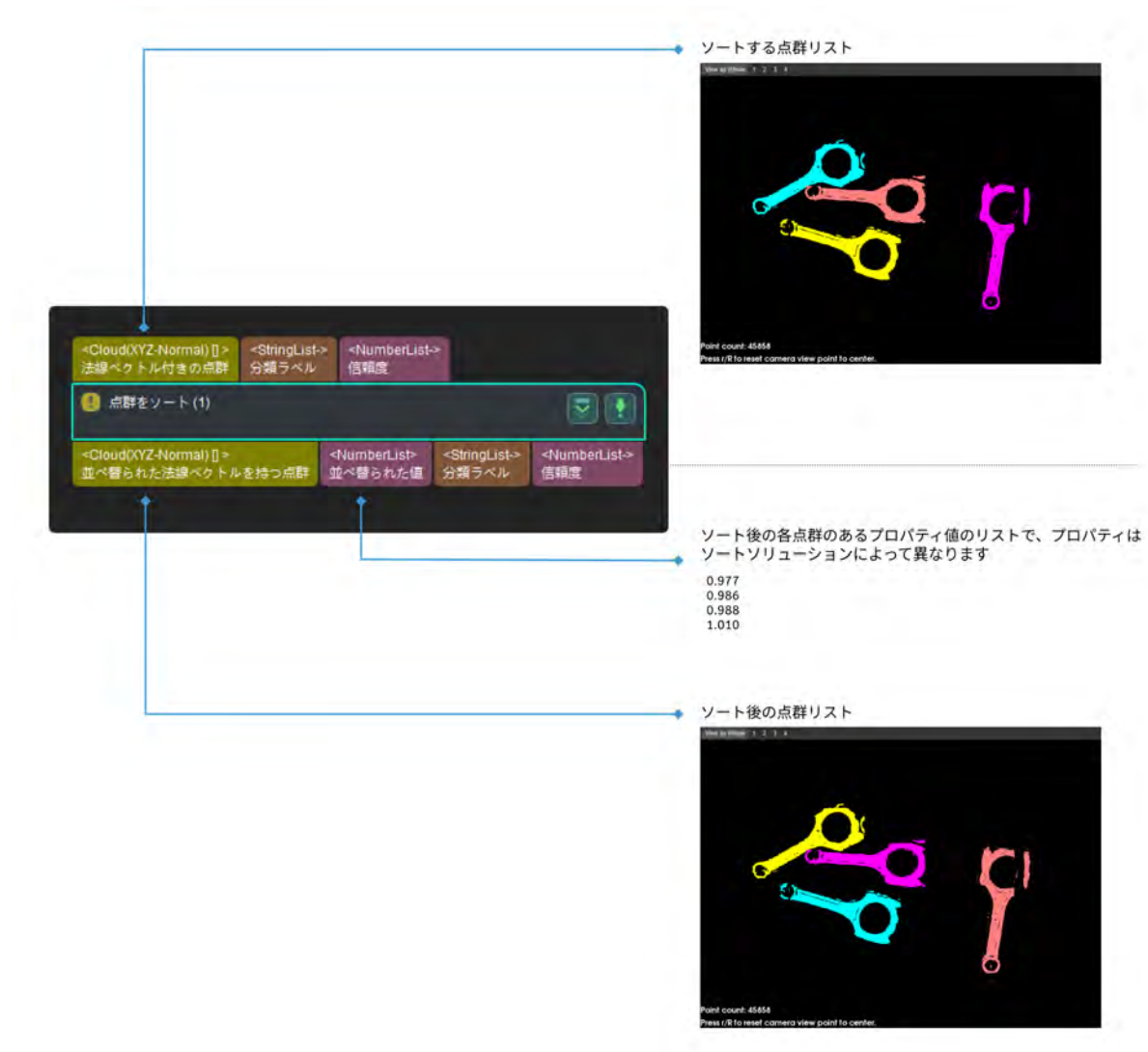
機能

指定されたルールに従って点群をソートしながら、対応するラベルおよび信頼度を出力します。

使用シーン

把持の順序を設定するために使用されます。

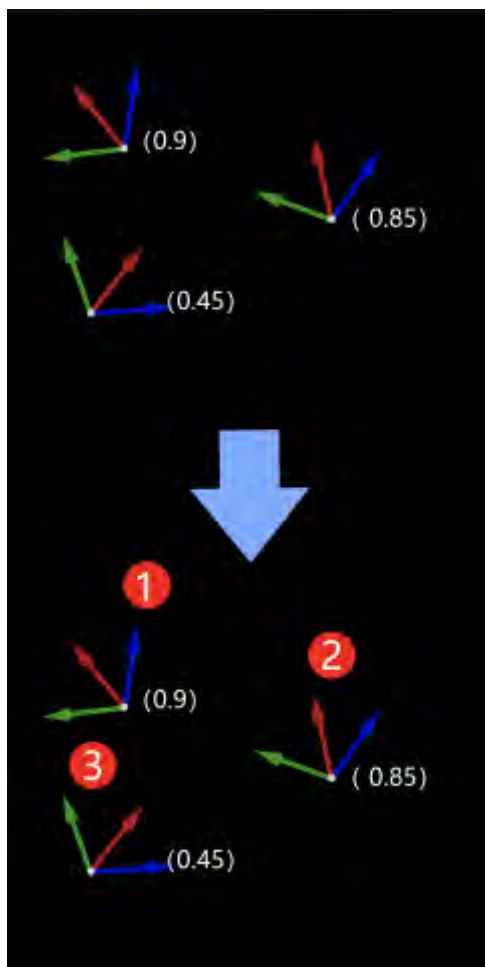
入力と出力



4.3.350. 入力したスコアで位置姿勢をソート

機能

このステップは、入力数値を使用して位置姿勢をソートすることができます。



使用シーン

通常は3D位置姿勢推定の関連ステップの後で使用され、マッチングスコアによってマッピングされた位置姿勢をソートします。

一般的な使用シーン：

- 金属部品の供給では、マッチングによって得られた把持位置姿勢に対して、位置姿勢のマッチングスコアによってソートします。マッチングスコアが高い部品を優先して把持します。
- ディープラーニング：ディープラーニングによって出力されたマスクのサイズによって位置姿勢とマスクをソートします。

入力と出力



パラメータの説明

ソート方式

パラメータ説明：このパラメータは、位置姿勢のソート方式を選択するために使用されます。入力したスコア（信頼度の値）に応じて、位置姿勢を昇順または降順にソートします。

オプション：

- 昇順：昇順にソートします。
- 降順：降順にソートします。

初期値：降順

4.3.351. 位置姿勢のXYZ値で位置姿勢をソート

機能

このステップは、入力位置姿勢のX、Y、Zのいずれかによって位置姿勢を再ソートし、ソート後の位置姿勢リストと対応するマッピングインデックスを出力することができます。

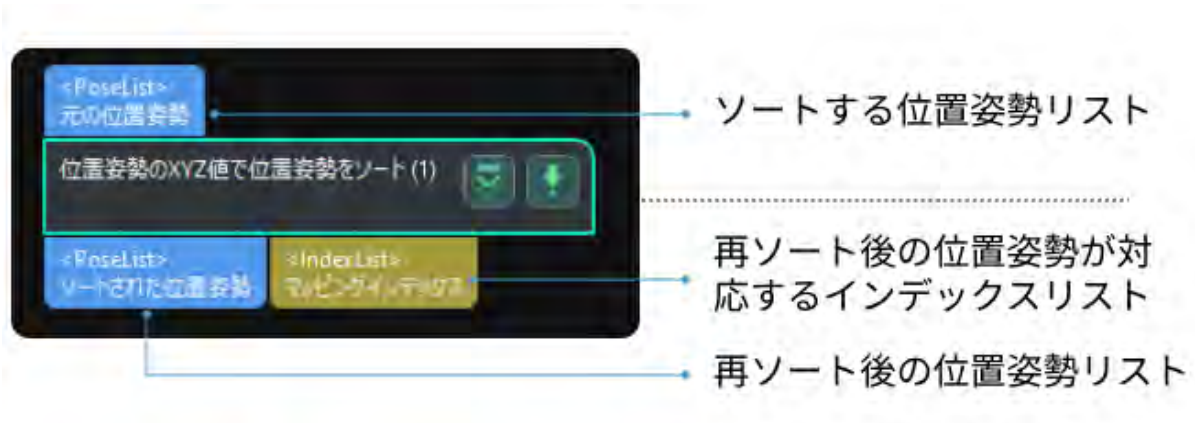


再ソートされた位置姿勢が対応するインデックスリストは、その他の数値の再ソートに使用できます。

使用シーン

汎用の位置姿勢のソートステップで、特定の使用シーンはありません。

入力と出力



パラメータの説明

ソート方式

パラメータ説明：このパラメータは、位置姿勢のソート方式を指定するために使用されます。

オプション：

- 昇順：昇順にソートします。
- 降順：降順にソートします。

初期値：降順

ソート値

パラメータ説明：このパラメータは、位置姿勢のソートに使用される値を設定するために使用されます。

オプション：

- X：位置姿勢のX値によってソートします。
- Y：位置姿勢のY値によってソートします。
- Z：位置姿勢のZ値によってソートします。

初期値：X

4.3.352. 2D位置姿勢をソート



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

X/Y座標によって入力された2D位置姿勢をソートします。

使用シーン

汎用の2D位置姿勢のソートステップです。特定の使用シーンはありません。

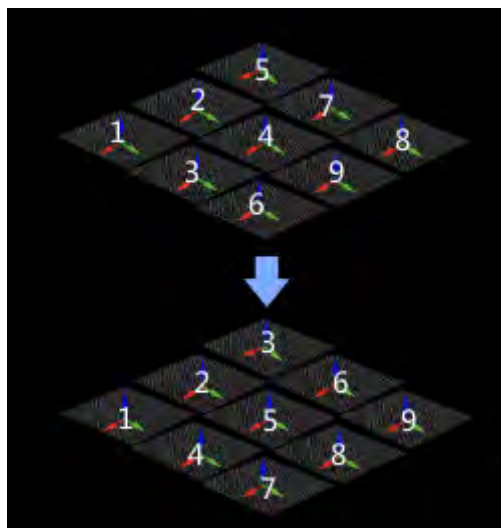
入力と出力

- **入力：** このポートに入力される位置姿勢はソートされます。
- **出力：** ソートされた位置姿勢。

4.3.353. 3D位置姿勢をソート

機能

このステップは、設定されたルールに従って3D位置姿勢をソートします。



使用シーン

その他のステップから出力された位置姿勢をルールに従ってソートします。通常、その後の位置姿勢の処理または把持戦略に使用されます。

入力と出力



パラメータの説明

ソート方法の設定

ソート方式

パラメータ説明：位置姿勢をソートするための方法を選択します。

初期値：NONE（操作なし）

オプション：SORT_BY_X、SORT_BY_Y、SORT_BY_Z、SORT_BY_DIS2BasePt、SORT_BY_DIS2BasePtOnXYPlane、SORT_BY_CONFIDENCE、SORT_BY_CUSTOM_DIR、SORT_BY_DIAGONAL_LENGTH

調整説明：

- SORT_BY_X：位置姿勢並進のX値に従って位置姿勢をソートします。調整の効果を [調整の例](#) に示します。
- SORT_BY_Y：位置姿勢並進のY値に従って位置姿勢をソートします。調整効果を [調整の例](#) に示します。
- SORT_BY_Z：位置姿勢並進のZ値に従って位置姿勢をソートします。調整効果を [調整の例](#) に示します。
- SORT_BY_DIS2BasePt：基準位置姿勢と入力位置姿勢間の距離に従って位置姿勢とスコアをソートします。これを選択する場合は基準位置の設定が必要です。調整効果を [調整の例](#) に示します。
- SORT_BY_DIS2BasePtOnXYPlane：XY平面上の基準位置姿勢と入力位置姿勢との距離に従って位置姿勢とスコアをソートします。これを選択すると、基準位置姿勢を設定する必要があります。調整効果を [調整の例](#) に示します。
- SORT_BY_CONFIDENCE：スコアに従って位置姿勢とスコアをソートします。これを選

択すると、スコアを入力する必要があります。調整効果を [調整の例](#) に示します。

- SORT_BY_CUSTOM_DIR：カスタマイズ方向上の位置姿勢並進に従って位置姿勢とスコアをソートします。これを選択すると、基準方向を設定する必要があります。調整効果を [調整の例](#) に示します。
- SORT_BY_DIAGONAL_LENGTH：入力サイズの対角線の長さに従って位置姿勢とスコアをソートします。これを選択すると、物体のサイズを入力する必要があります。調整効果を [調整の例](#) に示します。

昇順ソート

初期値：チェックを入れる

調整説明：実際のニーズに応じて昇順ソートするかを決めます。

調整の例

SORT_BY_X

位置姿勢のソート効果は下図に示します（降順ソートを例として）。



SORT_BY_Y

位置姿勢のソート効果は下図に示します（降順ソートを例として）。



SORT_BY_Z

位置姿勢のソート効果は下図に示します（降順ソートを例として）。



SORT_BY_DIS2BasePt

入力の基準位置姿勢は [-1,-1,-1]である場合、位置姿勢のソート効果は下図に示します（降順ソートを例として）。



SORT_BY_DIS2BasePtOnXYPlane

入力の基準位置姿勢は $[-1, -1, 0]$ である場合、位置姿勢のソート効果は下図に示します（降順ソートを例として）。



SORT_BY_CONFIDENCE

入力位置姿勢のスコアがa、b、cおよび $c > a > b$ である場合、位置姿勢のソート効果は下図に示します（降順ソートを例として）。



SORT_BY_CUSTOM_DIR

基準方向のX、Y、Z値が0、0、1である場合、位置姿勢のソート効果は下図に示します（降順ソートを例として）。



SORT_BY_DIAGONAL_LENGTH

対角線の長さがa、b、cおよび $c > a > b$ である場合、位置姿勢のソート効果は下図に示します（降順ソートを例として）。



4.3.354. リング形状の画像領域をまっすぐにする



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

入力画像のリング領域またはリング領域にある扇形領域をまっすぐにして出力します。

使用シーン

このステップは、リング形状に配置された文字を読み取るために使用されます。

入力と出力

● 入力：

1. このポートに入力される画像の指定された領域はまっすぐになります。
2. このポートに入力される2D変換は、以前設定したROIを適切な位置に変換するために使用されます。

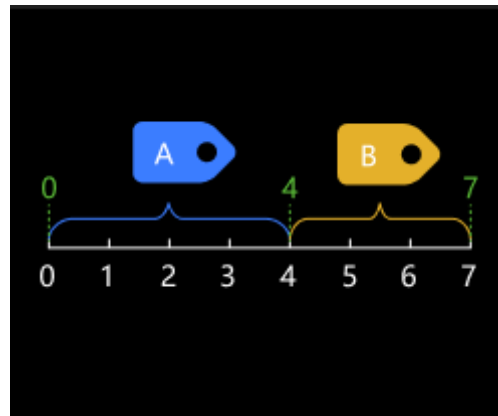
● 出力：

1. リング領域またはリング領域にある扇形領域をまっすぐにするによって生成された画像。

4.3.355. しきい値によって数値を分類

機能

設定されたしきい値によって入力された数値を分類し、分類ラベルを出力します。



使用シーン

数値を大小によって分類してラベルを追加します。通常、ステップ [必要なラベルであるかどうかを検証](#) と併用して数値区間によってリストをフィルタリングします。

あるしきい値によって数値リストを二つの部分に分けてその中の一つを除去する場合、ステップ [しきい値によって数値を二項分類](#) を使用して、[フィルタリング](#) と併用してください。

入力と出力



パラメータ説明

ファイル設定

しきい値ファイル

パラメータ説明：このパラメータは、しきい値を設定するために使用されます。入力された数値はしきい値によって分類され、しきい値はJSONファイルに保存されます。


値の種類：数値。





それ以外の種類のデータは無効とみなされ、正常に入力することができません。

調整アドバイス：1つまたは複数のしきい値を設定できます。しきい値は、小さいものから順に並べ替える必要があります。

設定方法：

1. [しきい値を設定] をクリックし、JSONエディタの画面に入ります。
2.  をクリックし、表示される画面で数値を入力して [OK] をクリックします。すると、1つのしきい値が追加されます。
3. 複数のしきい値を設定する必要がある場合、手順2を繰り返してください。



しきい値は小さいものから大きいものの順に並べる必要があります。追加されたしきい値の順序が条件を満たさない場合、 または  をクリックして、しきい値の順序を調整することができます。

4. ディレクトリの [...] をクリックし、JSONファイルの保存ディレクトリを選択します。デフォルトのディレクトリは、プロジェクトが配置されているディレクトリです。

5. **JSONファイル名** にファイル名を入力し、**[保存]** をクリックします。すると、しきい値がこのファイルに保存されます。
6. **[確認]** をクリックし、しきい値の追加が完了します。



しきい値のファイルが事前に設定されている場合は、**JSONファイル名** の [...] をクリックしてファイルを選択し、しきい値を読み取ることができます。

ラベルファイル

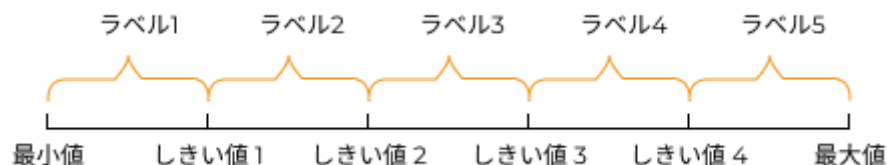
パラメータ説明：このパラメータは、数値分類後のラベルを指定するために使用されます。また、ラベルはJSONファイルに保存されます。

ラベル名：文字列（特殊文字不可）



入力されたデータが要件を満たさない場合は、無効なデータとみなされ、正常に入力することができません。



調整説明：ラベルの数は、しきい値の数に1を加えたものです。しきい値とラベルの対応関係は、ラベル名とは関係ありませんが、しきい値とラベルの順序で決定されます。



設定方法：

1. **[ラベルを設定]** をクリックし、JSONエディタの画面に入ります。
2. **+** をクリックし、表示される画面で文字列を入力して **[OK]** をクリックします。すると、1つのラベルが追加されます。
3. 手順2を繰り返してラベルを追加します。ラベルの数は、しきい値の数に1を加えたものです。



追加されたラベルの順序が条件を満たさない場合は、 または  をクリックして、ラベルの順序を調整することができます。

4. **ディレクトリ** は、現在のプロジェクトのパスです。[...] をクリックしてプロジェクトの場所を変更することができます。実際の状況に応じて設定してください。
5. **JSONファイル名** にファイル名を入力し、**[保存]** をクリックします。すると、ラベルがこのファイルに保存されます。
6. **[確認]** をクリックし、ラベルの追加が完了します。



ラベルファイルが事前に設定されている場合は、**JSONファイル名** の [...] をクリックしてファイルを選択し、ラベルを読み取ることができます。

4.3.356. 実数減算



これは古いバージョンのステップであり、メンテナンスが停止されていますので、新しいバージョンのステップ [\[vision-steps:numeric-operation:::numeric-operation\]](#) を使用してください。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

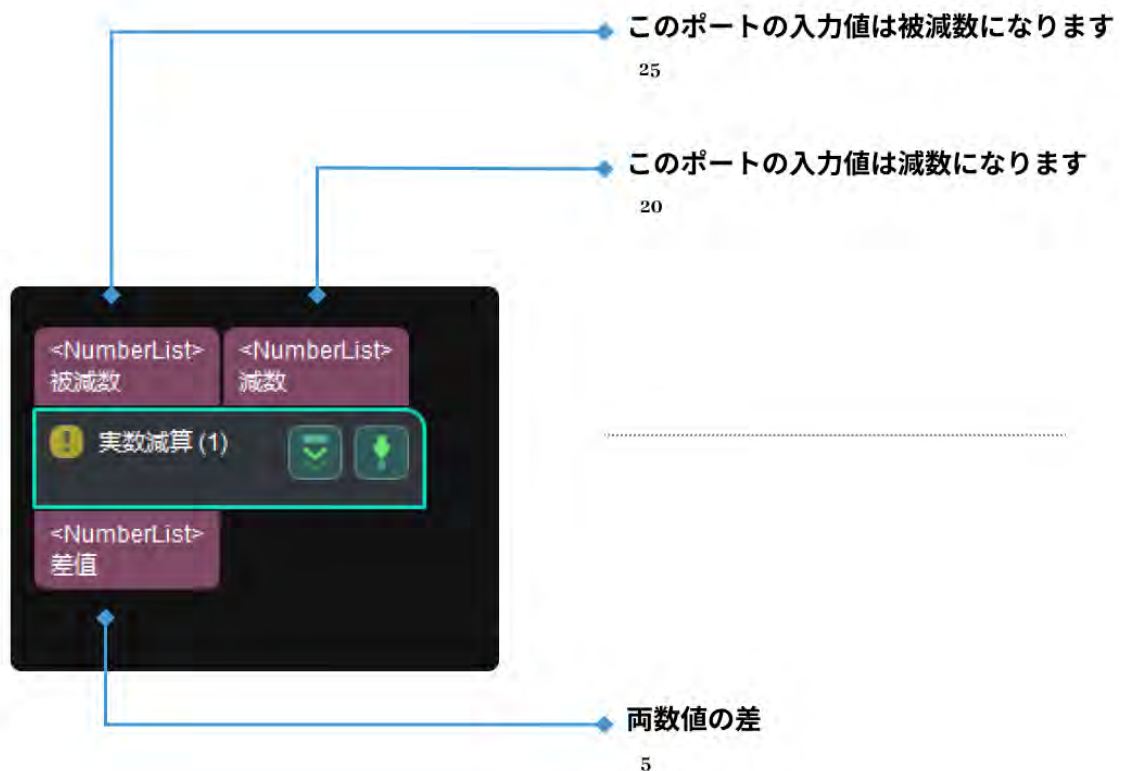
機能

2番目のポートの入力値から最初のポートの入力値を減算して、両数値の差を出力します。

使用シーン

これは古いバージョンのステップで、より完全な演算機能を備えた新しいバージョンのステップ [\[vision-steps:numeric-operation:::numeric-operation\]](#) を使用することを推奨します。

入力と出力



4.3.357. 一部領域で低いスコアの位置姿勢を除去(NMS)



これは古いバージョンのステップであり、メンテナンスが停止されていますので、新しいバージョンのステップ [\[vision-steps:3d-coarse-matching:::3d-coarse-matching\]](#) を使用してください。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

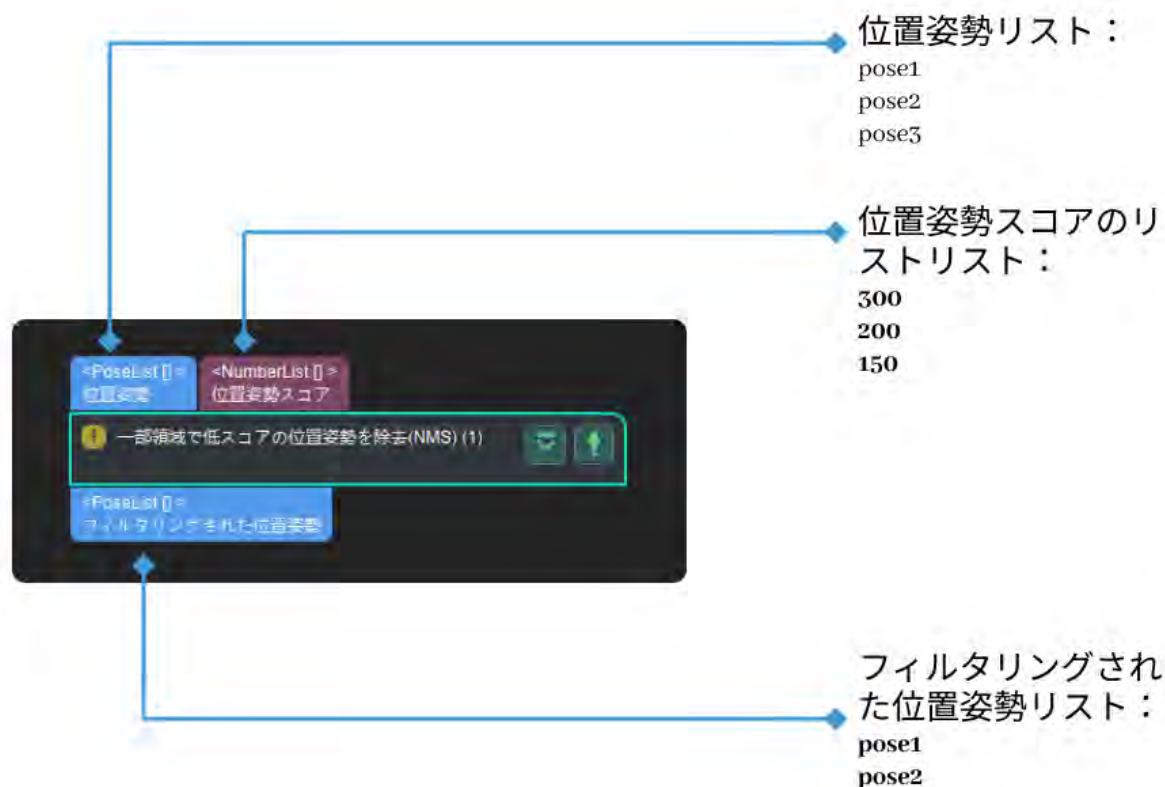
機能

設定された範囲内で最高スコアの位置姿勢を見つけ、設定されたしきい値に従ってその周りのスコアが低い位置姿勢を除去します。

使用シーン

位置姿勢をフィルタリングする方法の一つです。

入力と出力



パラメータの説明

局所検索半径

初期値：0.05

調整説明：一部領域の検索範囲のサイズを設定します。サイズが大きくなると、より多くの位置姿勢が計算されるため、後の段階での演算時間が長くなる可能性があります。

スコア比のしきい値

初期値：0.02

調整説明：位置姿勢のスコアがしきい値よりも小さい場合、それは考慮されません。このパラ

メータを大きくすると、位置姿勢のスコアを高くする必要があることを意味します。

4.3.358. 対象物の回転対称性

本節では、対象物の回転対称性について説明します。

概要



本節に説明した「対称性」は「回転対称性」を指します。

回転対称とは、2次元図形を一つの点を中心に回転させた場合に元の図形に完全に重なり合うことです。対称性がある対象物に対し、対象物の対称性パラメータを設定することができます。対象物の対称性を利用すると、ロボットが把持や配置を実行するときに発生するロボットハンドの回転を低減し、把持計画の成功率を向上させることが可能です。これによって、ロボットがさらにスムーズに動作します。

対称性のタイプ

回転対称軸を確認

[[vision-tools:matching-model-and-pick-point-editor:::matching-model-and-pick-point-editor](#)] で設定した幾何学的中心点の軸を回転対称軸とします。

回転対称軸は一つだけではありません。対象物の配置方式やロボットハンドの種類などによって、Mech-Vision で設定された対象物の幾何学的中心点は異なるので対称軸も異なります。

以下では一般的な対象物の対称タイプを示します。

対象物の対称タイプについては、[[vision-tools:matching-model-and-pick-point-editor:::matching-model-and-pick-point-editor](#)] を参照して幾何学的中心点と把持点の設定を確認してください。

対称性を持たない対象物

下表に示す対象物は対称性を持たない対象物です。

			
クローラシュー	コネクタ	自動車ドア板金	ハウジング

Z軸に関して対称性を持つ対象物

下表に示す対象物はZ軸に関して対称性を持つ対象物です。

			
ブレーキディスク	フランジ	歯車	段ボール箱

X/Y軸に関して対称性を持つ対象物





下表に示す対象物はX/Y軸に関して対称性を持つ対象物です。

			
整列して並べられた 反射する鋼棒	ばら積み鋼棒	ホースコネクタ	ボルト

対称回数

対象物を対称軸を中心に a° 回転させた後元の図形に完全に重なり合う時、対象物の対称回数は $N = 360^\circ / a^\circ$ です。

例えば、一般的な段ボール箱の対称回数（N）は2で、三角柱の対称回数（N）は3で、円柱の対称回数（N）は無限大（円対称）です。

			
N = 2	N = 4	N = 9	円対称

試行範囲

下図に示すように、BとCの間の角度が試行範囲となります。

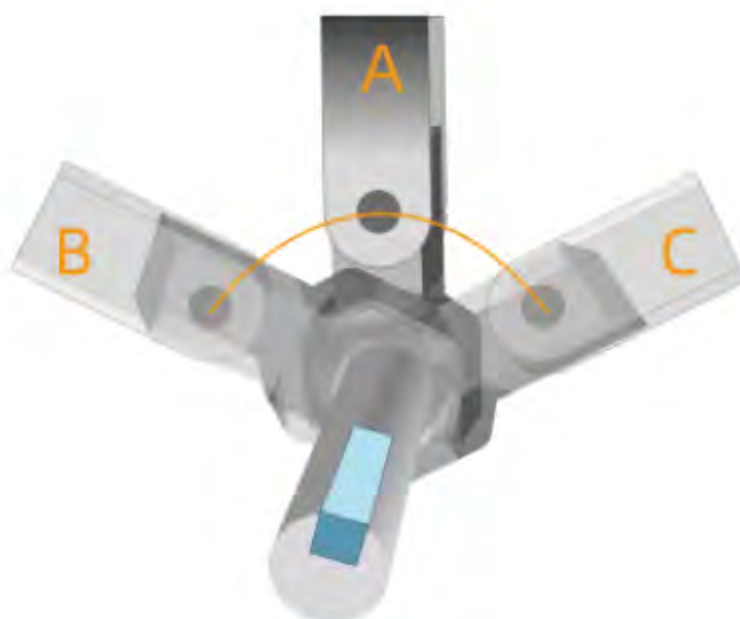


Figure 12. id10

試行範囲は対象物の配置方式やコンテナの形状、ロボットハンドの種類、タクトなどによって設定する必要があります。試行範囲を大きく設定すると計画の速度が遅くなりますが、小さく設定すると把持点に到達できなくなる可能性があります。

試行回数

対称回数と試行範囲によって自動的に計算されます。

対称回数Nは10とし、試行範囲は $\pm 80^\circ$ とすると、対称角度の間隔は $360^\circ/10=36^\circ$ となります。ある方向の試行回数は $80/36=2$ あまり8となり、つまり試行回数は2となります（それぞれ 36° と 72° の位置で試行します）。もう一方の方向も併せて試行回数総計は $1+2*2=5$ となります（ -72° 、 -36° 、 0° 、 36° 、 72° の位置で試行します）。

4.3.359. 2Dテンプレートマッチング

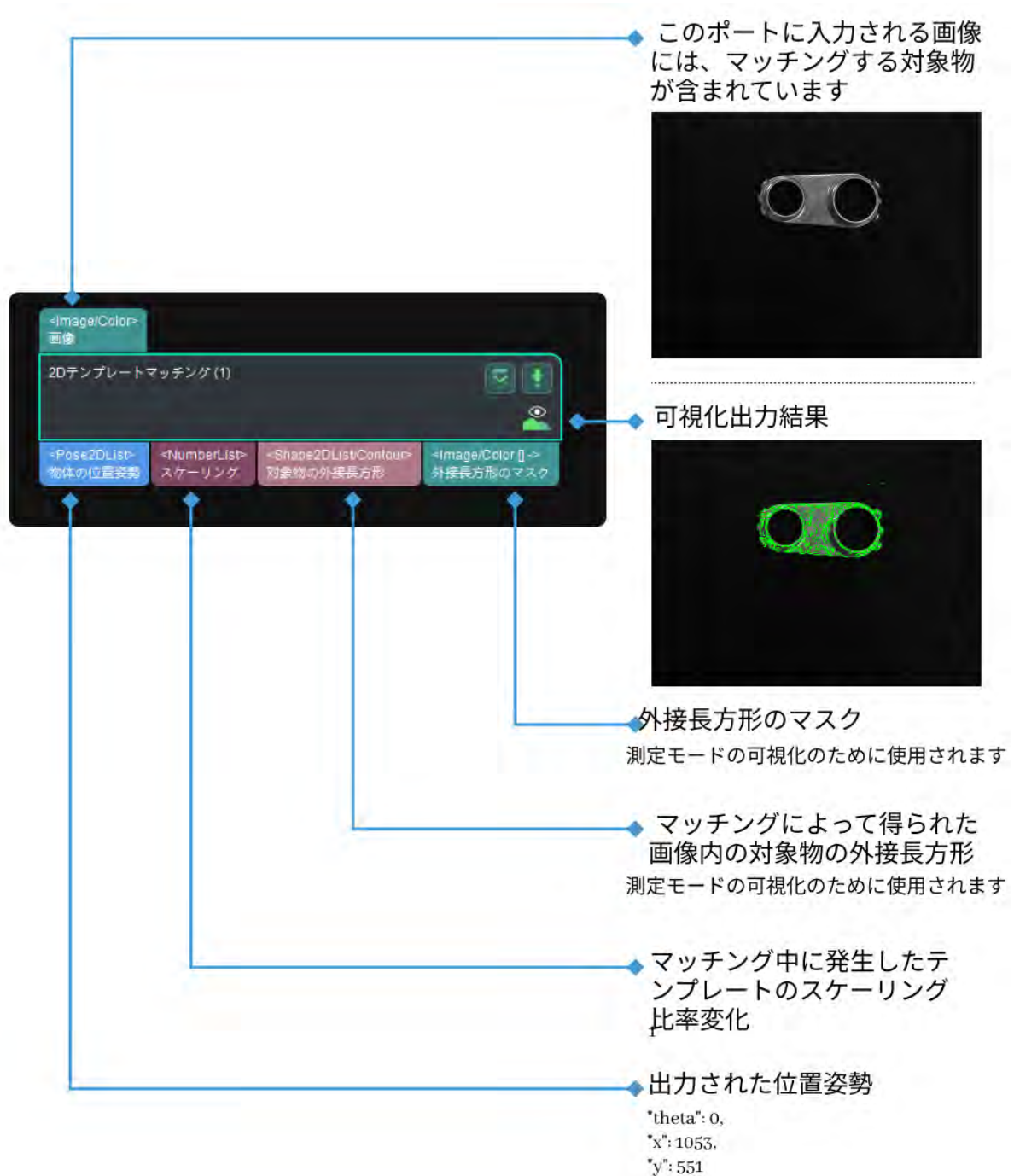
機能

画像内の対象物を2Dテンプレートとマッチングして、対象物の2D位置姿勢を計算します。

使用シーン

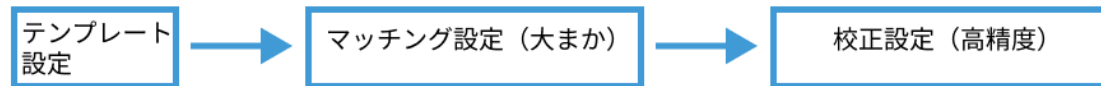
このステップは、対象物の位置姿勢を計算するために画像内の対象物の位置を取得し、または後続のステップで画像位置と向きとの校正のための2D変換を計算するために使用されます。

入力と出力



パラメータの説明

パラメータ調整の手順は以下の通りです。

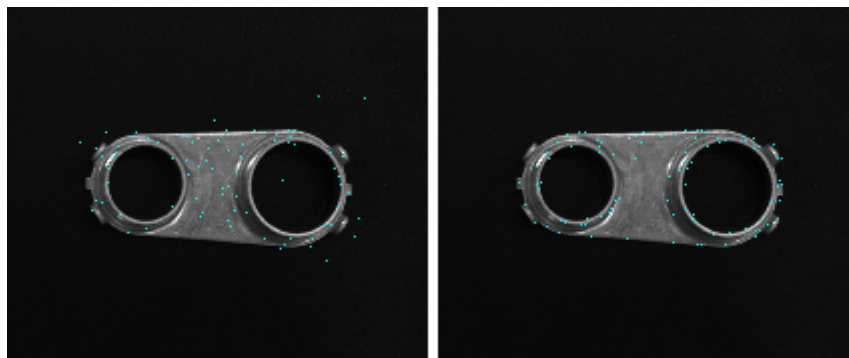


勾配の大きさの下限しきい値

初期値：60

調整説明：マッチングの前に、対象物の特徴を抽出するために勾配の大きさの下限しきい値を適切に設定する必要があります。このしきい値よりも大きな勾配の大きさを持つピクセルは、特徴ピクセルと見なされます。調整前後の効果比較を下図に示します。

下図では、左側のように勾配の大きさの下限しきい値を10に設定すると物体特徴抽出の効果が低くなります。右側のように60に設定するとより良い抽出が行われます。したがって、しきい値を適切に大きくすると、対象物の特徴を正確に抽出するのに役立ちます。



テンプレート特徴が生成済み

初期値：チェックを入れない。

調整説明：このステップを初めて実行する時に、ステップ「2Dテンプレートを作成」によって出力されたテンプレート画像の特徴ピクセルが抽出され、プロジェクトフォルダに保存されます。そして、マッチングする画像から抽出した特徴ピクセルとテンプレート画像の特徴ピクセルとをマッチングし、マッチング結果を出力します。このステップを後で実行すると、保存されたテンプレート画像の特徴ピクセルが直接呼び出されます。テンプレート特徴を再抽出する必要がある場合、たとえば、いくつかのパラメータとテンプレート画像を変更した後、このパラメータのチェックを外してこのステップを再度実行し、テンプレート画像の特徴ピクセルを再抽出してください。

テンプレート設定（**テンプレート特徴が生成済み** にチェックを入れない場合に表示）

テンプレート特徴のファイルパス

調整説明：テンプレート特徴のファイルを保存するパスを選択します。

テンプレート特徴画像のファイルパス

調整説明：ステップ [2Dテンプレートを作成](#) の2番目の出力ポートで保存されたテンプレート画像のファイルパスを選択します。

テンプレートマスクのファイルパス

調整説明：ステップ [2Dテンプレートを作成](#) の3番目の出力ポートで保存されたテンプレートマスクのファイルパスを選択します。

開始角度

初期値：-180.0

調整説明：マッチング時の画像内でのテンプレートの回転範囲は、開始角度と終了角度によって指定されます。

終了角度

初期値：180.0

調整説明：上記の「開始角度」をご参照ください。

角度のステップサイズ

初期値：1.0

調整説明：各マッチングの試行が行われる角度の数は、角度ステップサイズによって指定されます。

開始スケール

初期値：0.80

調整説明：マッチング時の画像内でのテンプレートのスケール変更範囲は、開始スケールと終了スケールによって指定されます。

終了スケール

初期値：1.20

調整説明：上記の「開始スケール」をご参照ください。

スケールのステップサイズ

初期値：0.20

調整説明：各マッチングの試行が行われるスケール変更の回数は、スケールのステップサイズによって指定されます。

マッチング設定（**テンプレート特徴が生成済み** にチェックを入れる場合に表示）

マッチングスコアの下限しきい値

初期値：50.0

調整説明：このしきい値より小さいスコアを持つマッチング結果は破棄されます。残りの結果は、このステップの高精度マッチングによってフィルタリングされた後に出力されます。



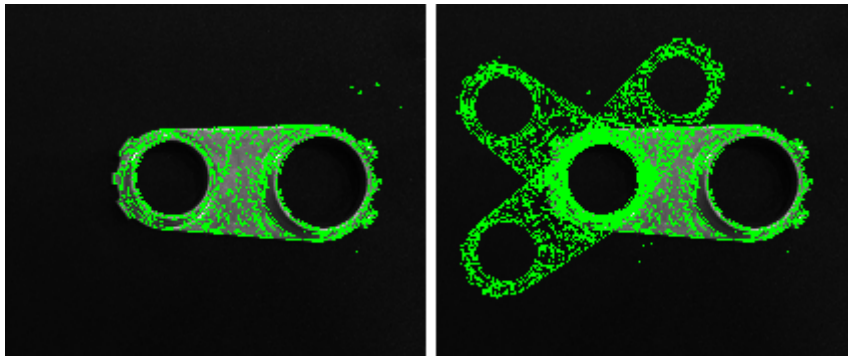
- 物体が遮られている場合は、このしきい値を適切に小さくする必要があります。
- このしきい値が高すぎると、結果が出力されない場合があります。

マッチング結果の最大数

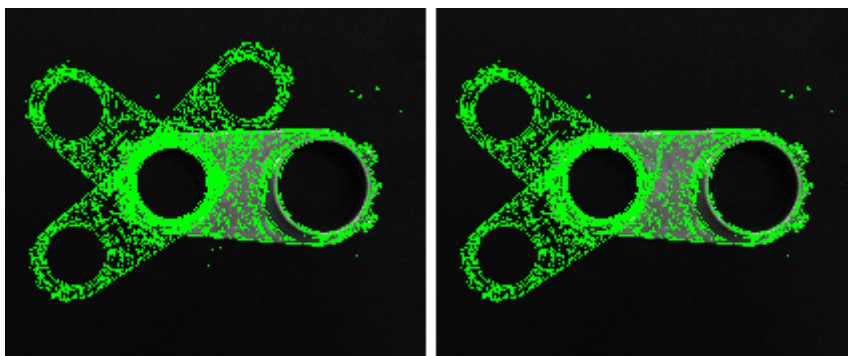
初期値：3

調整説明：実際のマッチング結果の数がこの値よりも多い場合、リストの最後にある結果は放棄されます。

- **マッチングスコアの下限しきい値** が適切な値に設定されている場合、**マッチング結果の最大数** が大きいほど、マッチング結果の数が多くなります。
- **マッチングスコアの下限しきい値** を25に設定した場合、**マッチング結果の最大数** をそれぞれ1と4に設定すると、マッチング結果は下図のようになります。その中、**マッチング結果の最大数** を1に設定した場合のマッチング結果は左側のように示し、4に設定した場合のマッチング結果は右側のように示しています。



- **マッチング結果の最大数** が適切な値に設定されている場合、**マッチングスコアの下限しきい値** が大きいほど、マッチング結果の数が少なくなります。**マッチング結果の最大数** を4に設定した場合、**マッチングスコアの下限しきい値** をそれぞれ25と50に設定すると、マッチング結果は下図のようになります。その中、**マッチングスコアの下限しきい値** を25に設定した場合のマッチング結果は左側のように示し、50に設定した場合のマッチング結果は右側のように示しています。





上図の右側では、設定した **マッチングスコアの下限しきい値** より高いスコアを持つマッチング結果は3つしかないので、最大3つの結果を出力できます。

重複比率の上限しきい値

初期値：0.40

調整説明：マッチング結果で2つのマッチング結果に重複があれば、重複率がこの値より大きい場合、高いマッチングスコアを持つ結果が保持され、他の結果は破棄されます。

補正設定（**テンプレート特徴が生成済み** にチェックを入れる場合に表示）

検索半径

初期値：10.0

調整説明：位置姿勢の高精度校正を行うときに対応点を検索するために使用されます。マッチング結果の精度が出ない場合はこの値を適切に大きくすることができます。

4.3.360. テスト



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

開発者のテストに使用されます。

使用シーン

開発者のテスト用のステップです。普通のユーザーは、一般的な使用シーンでこのステップを使用する必要はありません。

入力と出力

- 入力：なし。
- 出力：なし。

4.3.361. 経路点のマッチング



これは古いバージョンのステップであり、メンテナンスが停止されています。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

モデルを変換して、入力点群とマッチングします。

使用シーン

特定のシーン向けのカスタマイズのステップです。

入力と出力

- 入力：
 1. シーンの点群。
- 出力：
 1. 変換されたモデル点群。
 2. 変換関係行列。

4.3.362. 画像変換



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

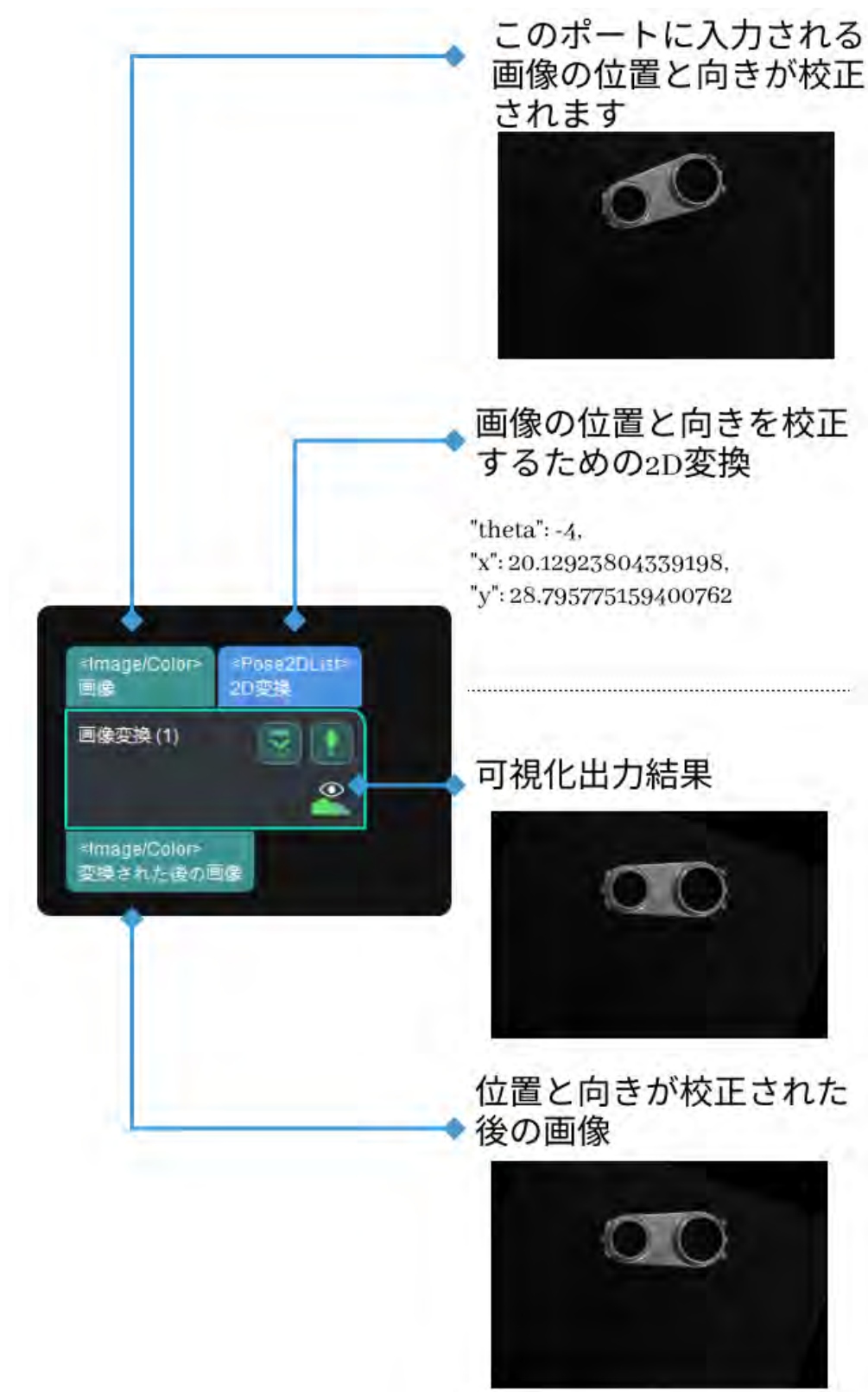
機能

入力された変換位置姿勢を使用して入力画像の位置と向きを補正し、補正された画像が基準画像と合わせられるようにします。

使用シーン

通常、計測シーンで画像を補正して、すべての画像ないの対象物の位置と向きが一致していることを確保するために使用されます。これは一般的にステップ [基準位置姿勢を記録して変換を計算](#) と併用されます。

入力と出力



4.3.363. 平面の点群を指定平面に合わせる



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

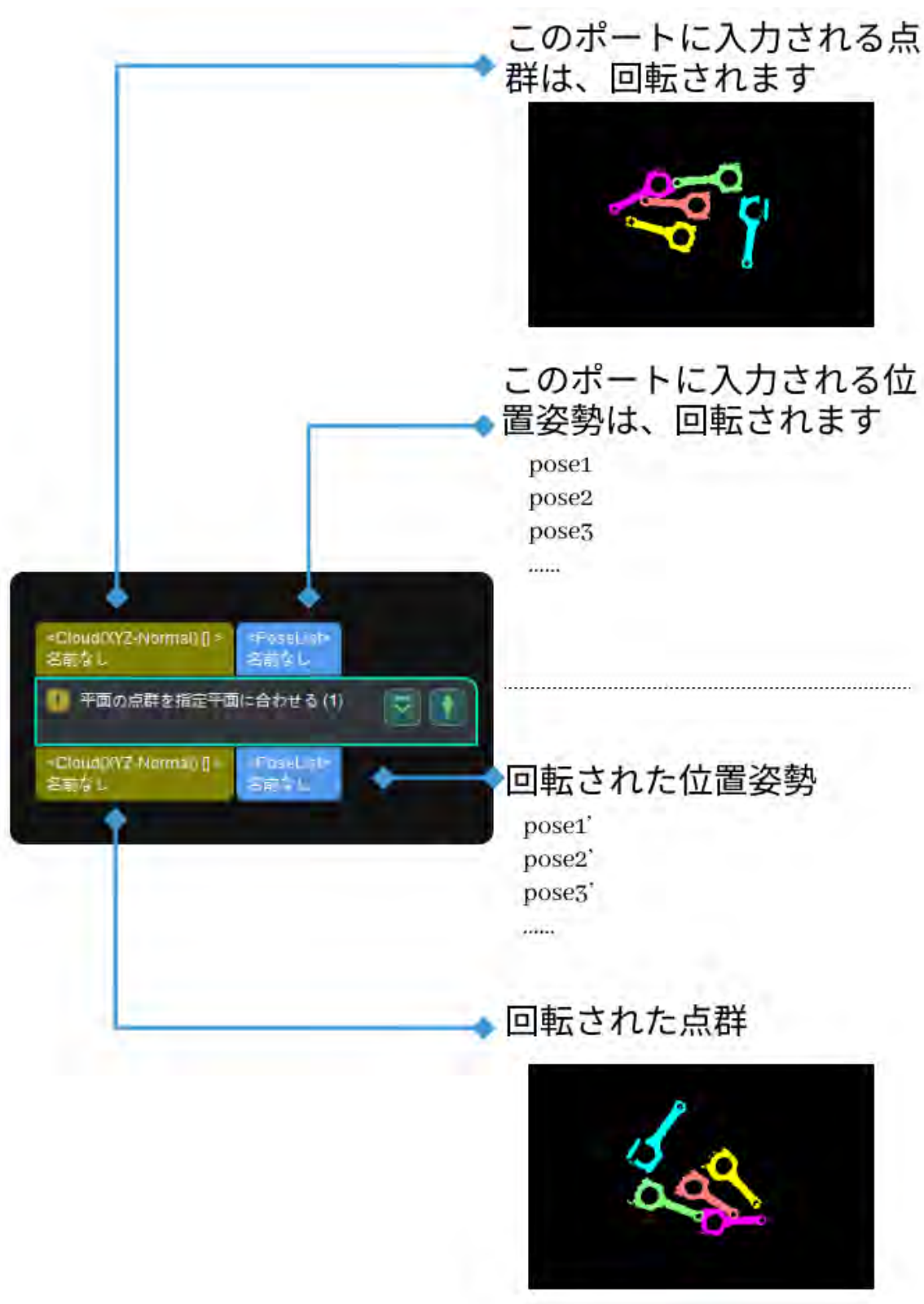
機能

設定された方向によって、入力点群および対応する位置姿勢を回転します。

使用シーン

特定のシーン向けのカスタマイズのステップです。

入力と出力



4.3.364. 点群変換（直行ロボット）

機能

このステップは、入力点群をカメラ座標系または直行ロボットの座標系に変換して出力できま

す。

使用シーン

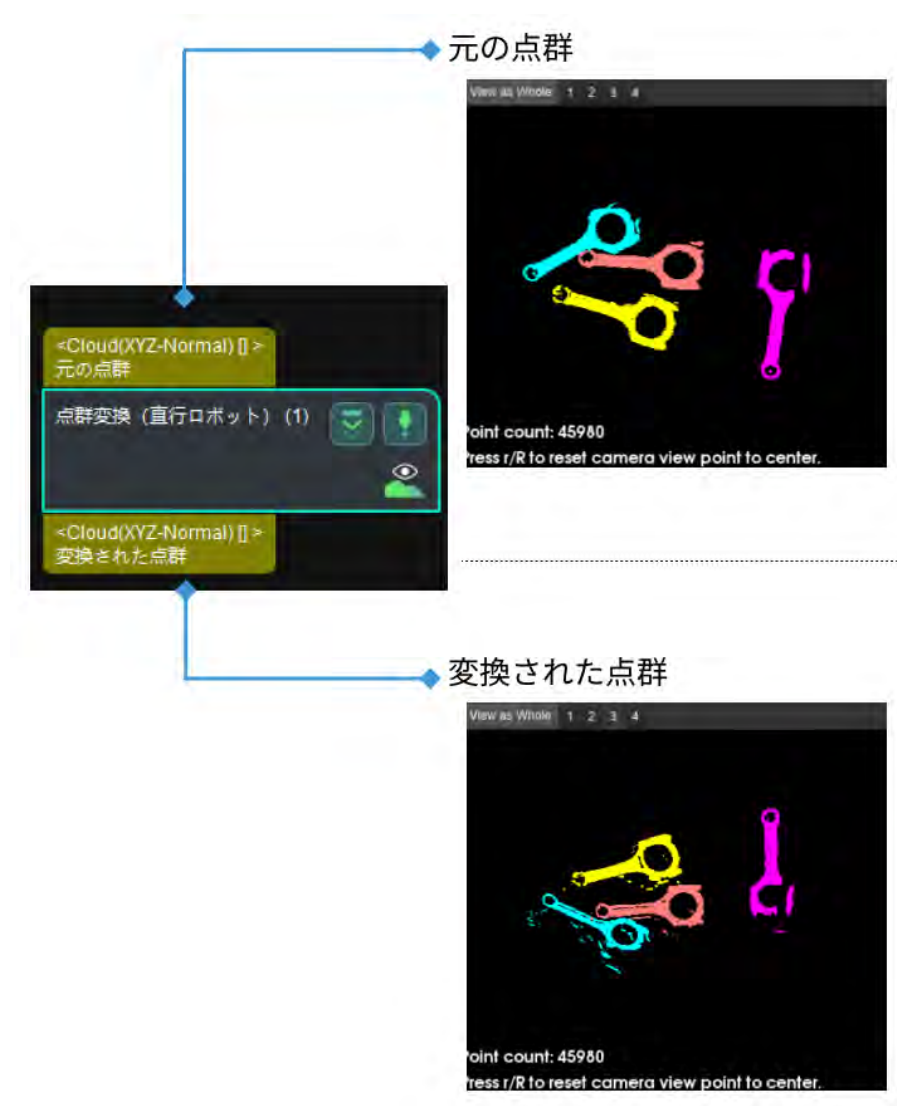
このステップは、直行ロボットを使用したプロジェクトに点群の座標系を変換するために使用されます。



プロジェクトに直行ロボット以外のロボットを使用している場合は、[\[vision-steps:transform-point-clouds::transform-point-clouds\]](#) ステップを使用してください。

誤ったステップを選択した場合、変換された点群の位置が正しくなくなり、誤った把持位置姿勢が出力される可能性があります。

入力と出力



パラメータの説明

変換方法

パラメータ説明：このパラメータは、点群の変換方法を選択するために使用されます。

オプション：CameraToRobot、RobotToCamera

- CameraToRobot：点群をカメラ座標系からロボット座標系に変換します。調整効果を [調整の例](#) に示します。
- RobotToCamera：点群をロボット座標系からカメラ座標系に変換します。調整効果を [調整の例](#) に示します。

初期値：CameraToRobot

4.3.365. 点群変換

機能

点群を他の座標系に変換したり、基準位置姿勢に従ってローカル座標系で変換したりできます。

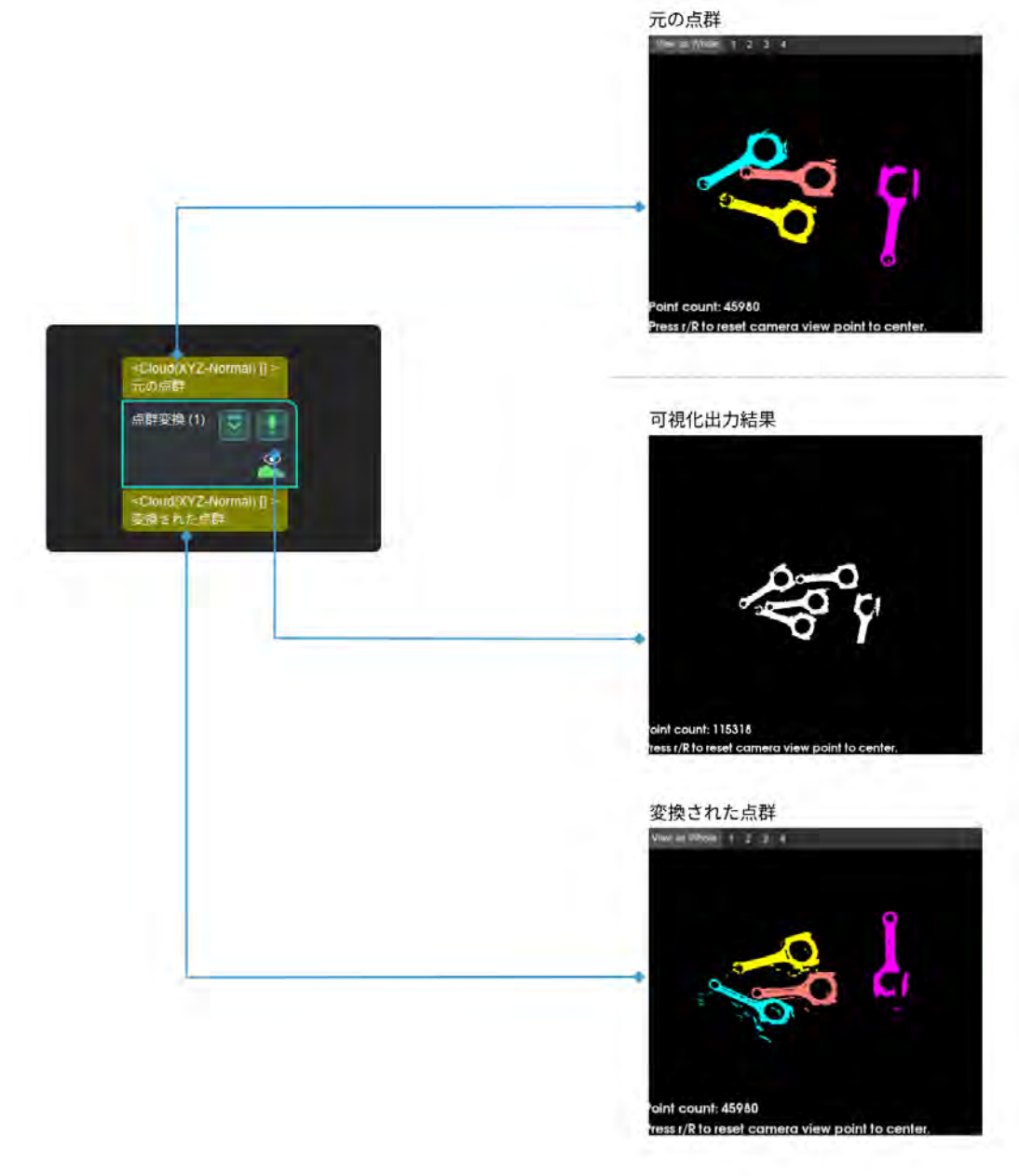


使用シーン

通常、点群がカメラ座標系とロボット座標系との間の相互変換に使用されます。

プロジェクトで直行ロボットが使用されている場合、[\[vision-steps:transform-point-clouds-for-truss::transform-point-clouds-for-truss\]](#) ステップを使用する必要があります。

入力と出力



パラメータの説明

変換設定

変換方法

パラメータ説明：このパラメータは、点群の変換方法を設定するために使用されます。

初期値：CameraToRobot

オプション

：CameraToRobot、RobotToCamera、AllWithFrst、FirstWithAll、UseCorrespondenceInput、AffineTransform

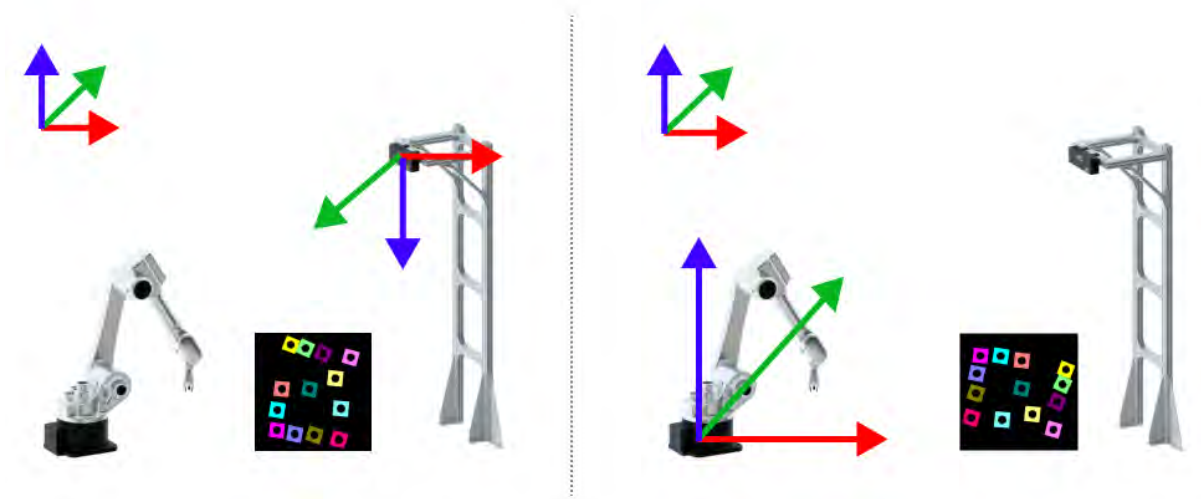
調整説明：

- CameraToRobot：点群をカメラ座標系からロボット座標系に変換します。調整効果を [調整の例](#) に示します。
- RobotToCamera：点群をロボット座標系からカメラ座標系に変換します。調整効果を [調整の例](#) に示します。
- AllWithFrst：最初の入力基準位置姿勢を使用して点群を変換します。基準位置姿勢リストの入力が必要です。調整効果を [調整の例](#) に示します。
- FirstWithAll：入力のすべての基準位置姿勢を使用して最初の点群を変換します。基準位置姿勢リストの入力が必要です。調整効果を [調整の例](#) に示します。
- UseCorrespondenceInput：全ての点群を基準位置姿勢の座標系に一对一に変換します。基準位置姿勢リストを入力する必要がある、入力位置姿勢の数は基準位置姿勢の数と同じです。調整効果を [調整の例](#) に示します。
- AffineTransform：変換行列を使用して点群を変換します。下図に示すように、「PoseListをMatrix4Dに変換」ステップにより、postlistをmatrix4D入力に変換してから、本ステップのMatrix4D入力ポートに入力できます。調整効果を [調整の例](#) に示します。

調整の例

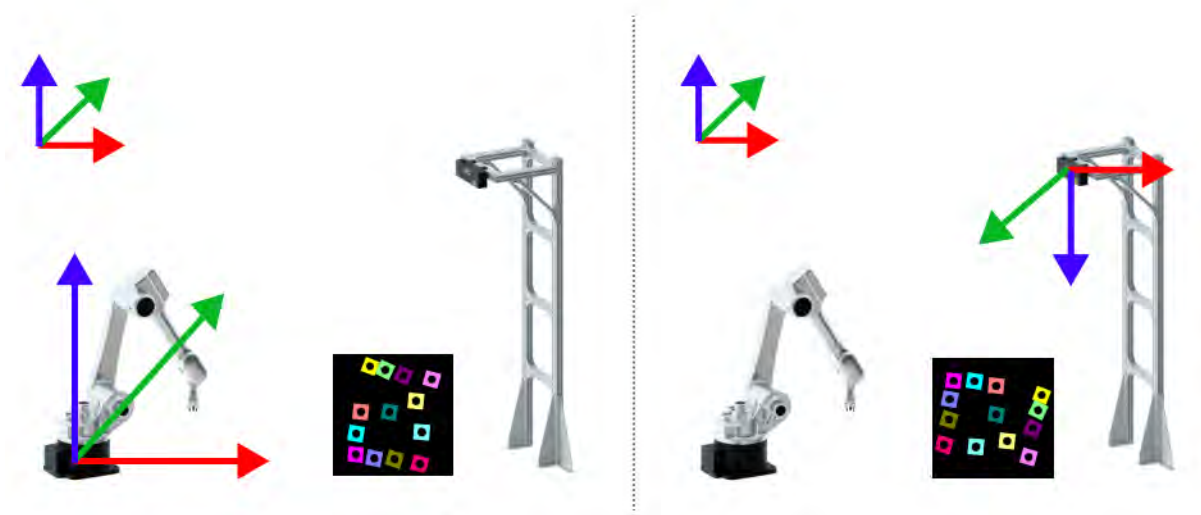
RobotToCamera

CameraToRobot に設定した場合の点群変換効果を下図に示します。左側は変換前で、右側は変換後です。



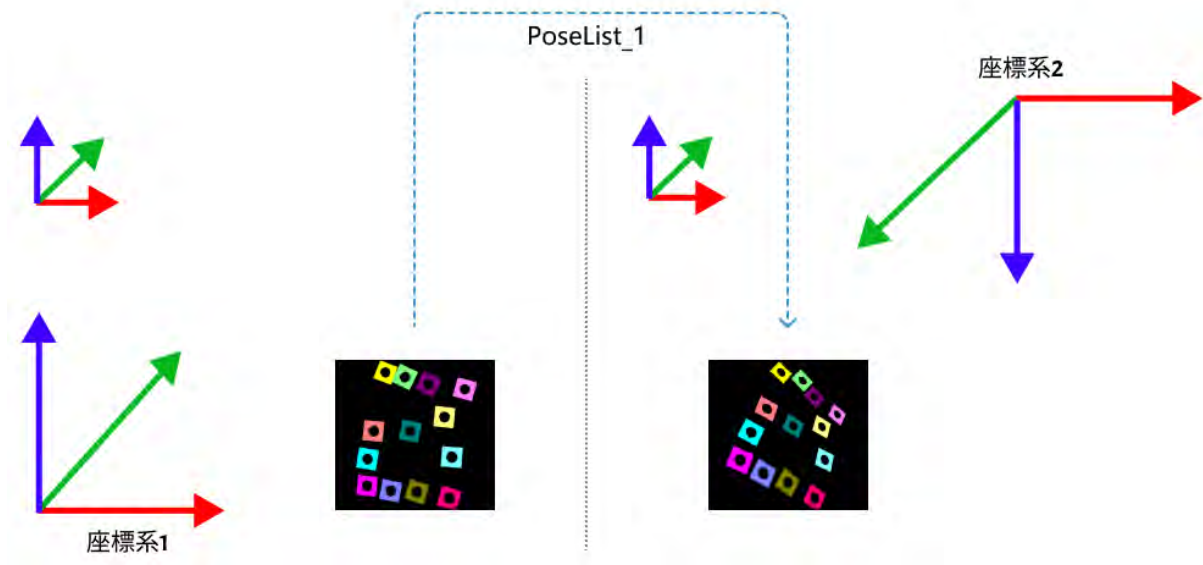
CameraToRobot

RobotToCamera に設定した場合の点群変換効果を下図に示します。左側は変換前で、右側は変換後です。



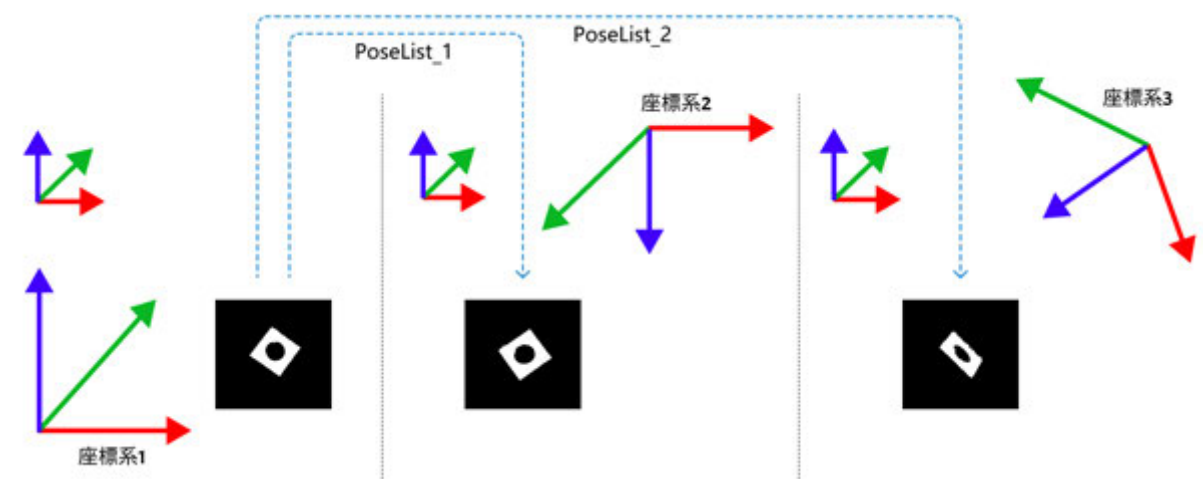
AllWithFirst

AllWithFirst に設定した場合の点群変換効果を下図に示します。左側は変換前で、右側は変換後です。



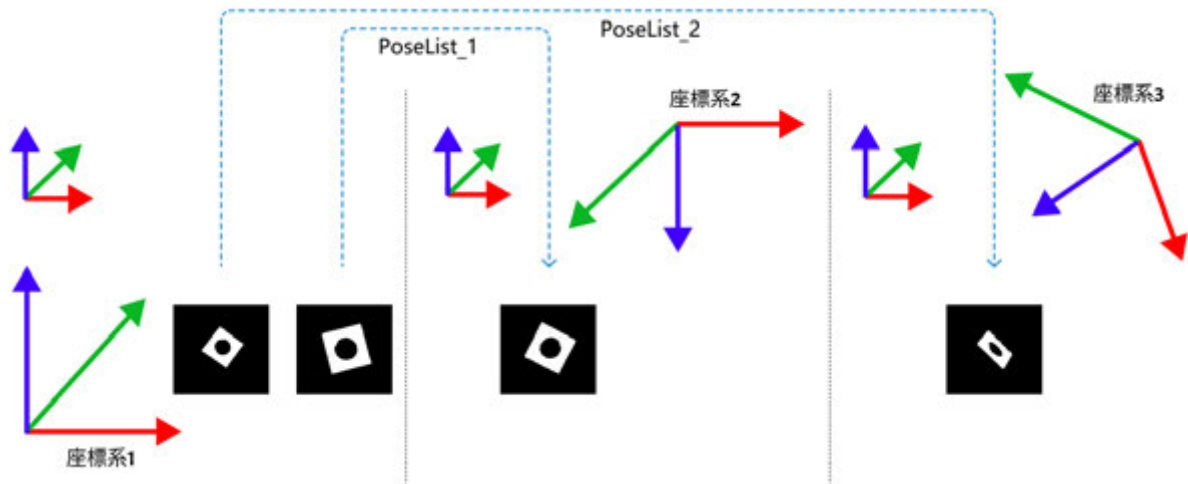
FirstWithAll

AllWithFirst に設定した場合の点群変換効果を下図に示します。



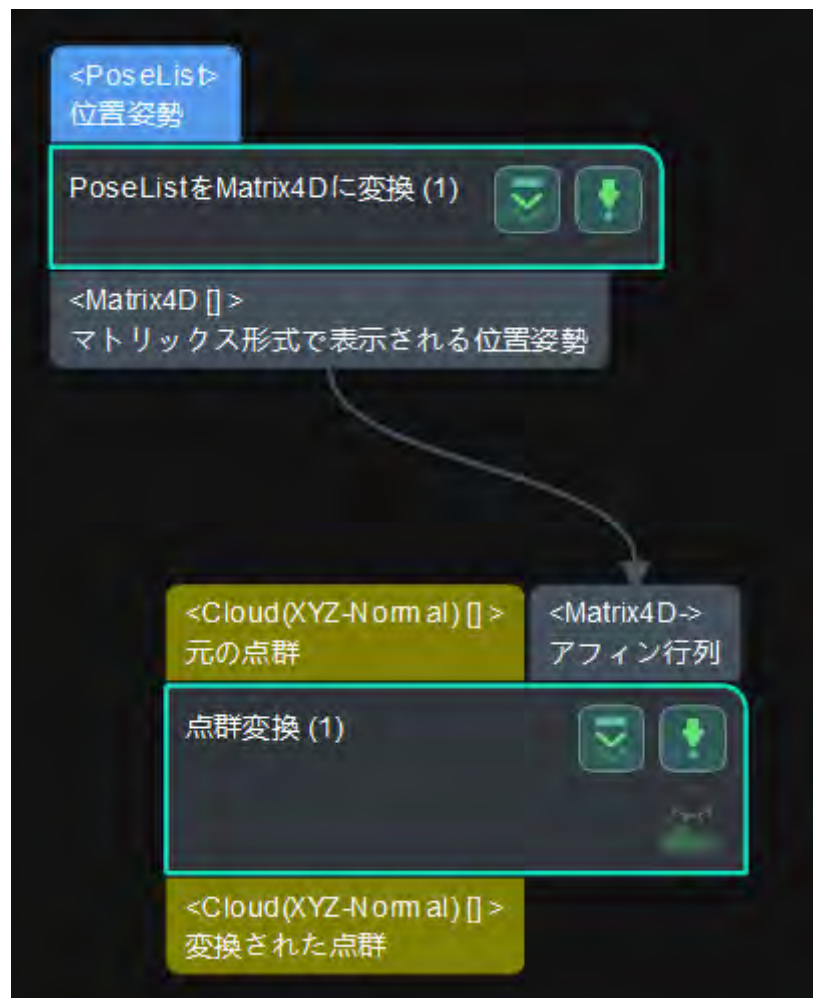
UseCorrespondenceInput

UseCorrespondenceInput に設定した場合の点群変換効果を下図に示します。



AffineTransform

AffineTransform に設定した場合の点群変換効果を下図に示します。



4.3.366. 位置姿勢を変換（直行ロボット）

機能

入力位置姿勢をカメラ座標系または直行ロボットの座標系に変換して出力します。

使用シーン

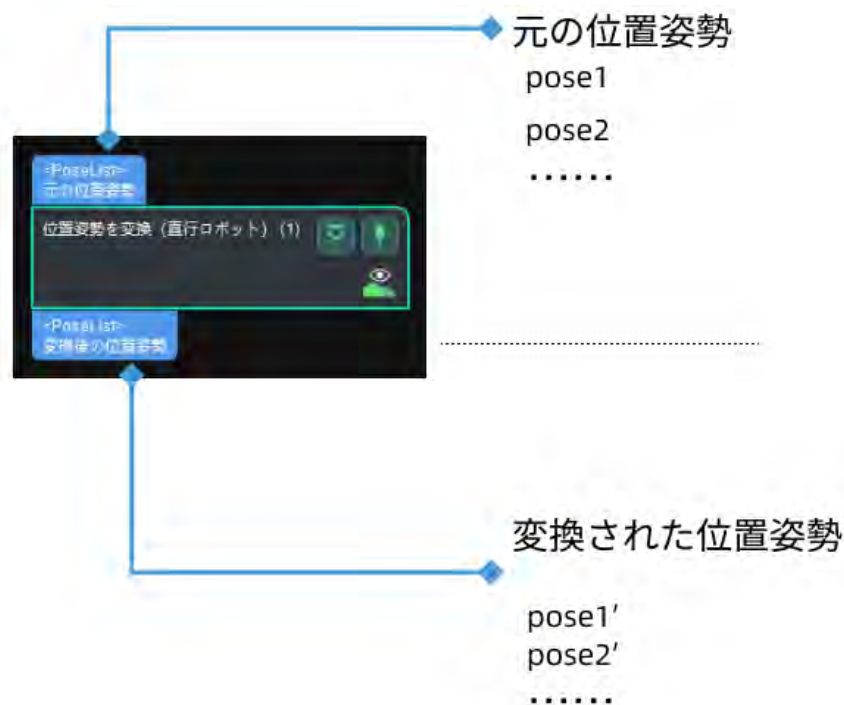
このステップは、直行ロボットを使用したプロジェクトに位置姿勢の座標系を変換するために使用されます。



プロジェクトに直行ロボット以外のロボットを使用している場合は、[\[vision-steps:transform-poses:::transform-poses\]](#) ステップを使用してください。

誤ったステップを選択した場合、変換された位置姿勢が正しくなくなり、誤った把持位置姿勢が出力される可能性があります。

入力と出力



パラメータの説明

変換方法

パラメータ説明：このパラメータは、位置姿勢の変換方法を選択するために使用されます。

オプション：CameraToRobot、RobotToCamera

- CameraToRobot：カメラ座標系で対象物の位置姿勢はPose1で、「変換方法」を **CameraToRobot** に設定すると、直行ロボット座標系での対象物の位置姿勢（Pose2）を出力できます。調整効果を [調整の例](#) に示します。
- RobotToCamera：直行ロボット座標系で対象物の位置姿勢はPose1で、「変換方法」を

RobotToCamera に設定すると、カメラ座標系での対象物の位置姿勢（Pose2）を出
力できます。調整効果を [調整の例](#) に示します。

初期値：CameraToRobot

4.3.367. 位置姿勢を変換

機能

このステップは、位置姿勢を他の座標系に変換し、又は基準位置姿勢によってローカル座標系
で変換できます。

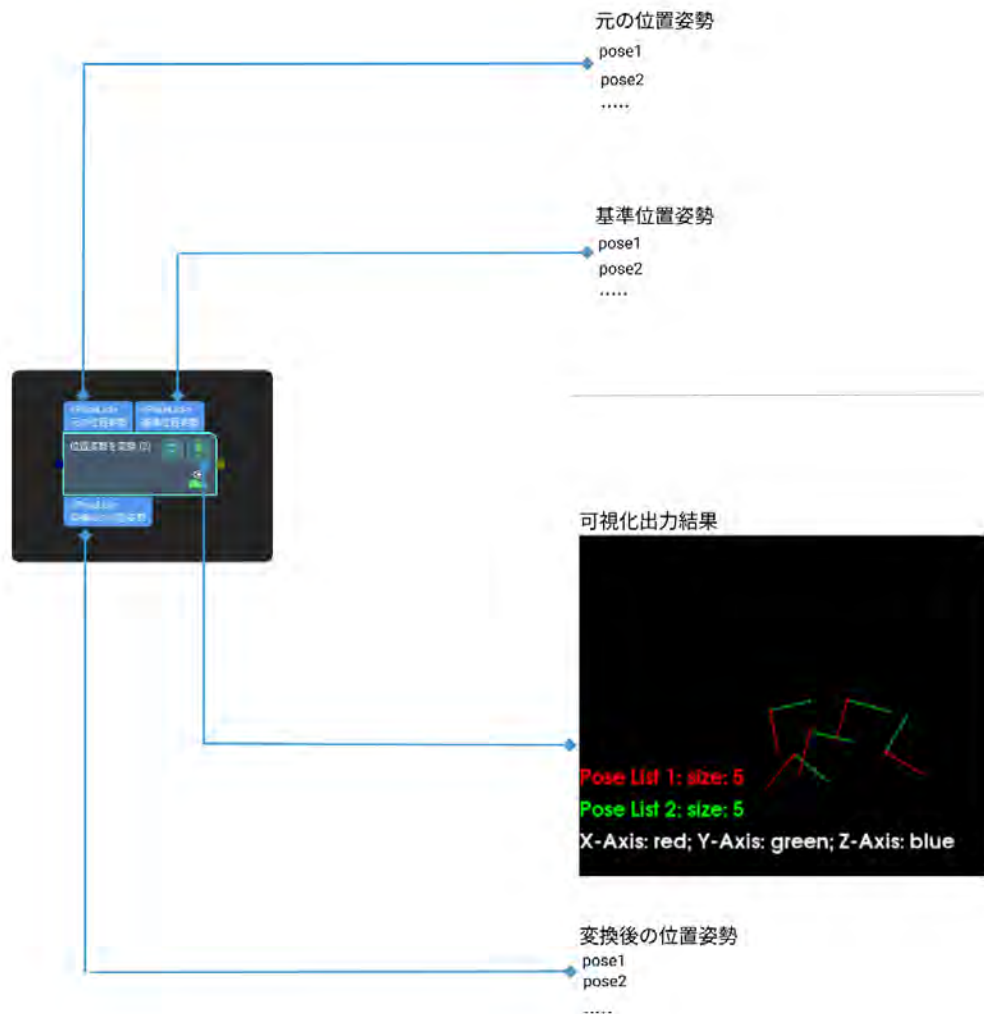


使用シーン

通常、位置姿勢がカメラ座標系とロボット座標系との間の相互変換に使用されます。

プロジェクトで直行ロボットが使用されている場合、[位置姿勢を変換（直行ロボット）](#) ステッ
プを使用する必要があります。

入力と出力



パラメータの説明

可視化設定

選択された座標系で点群を表示

パラメータ説明：ロボット座標系またはカメラ座標系で点群を表示します。

初期値：ロボット座標系。

オプション：ロボット座標系、カメラ座標系。

調整説明：**カメラ座標系**を選択する場合、デバッグ結果出力のウィンドウにカメラ座標系での可視化出力結果が表示されます。**ロボット座標系**を選択する場合、デバッグ結果出力のウィンドウにロボット座標系での可視化出力結果が表示されます。調整効果を [調整の例](#) に示します。



このパラメータは、プロジェクトにシーンの点群が設定された場合にのみ設定可能です。



す。

- **ロボット座標系** を選択すると、デバッグ結果出力のウィンドウが黒い画面になります。点群と座標を表示するには、視点を調整する必要があります。
- **カメラの座標系** を選択すると、デバッグ結果出力のウィンドウには点群のみが表示されます。座標を表示するには、点群の方向を調整する必要があります。

位置姿勢表示の設定

パラメータ説明：このパラメータは、位置姿勢の表示形式を設定するために使用されます。

初期値：出力位置姿勢のみを表示。

オプション：入力位置姿勢のみを表示、出力位置姿勢のみを表示、入力と出力位置姿勢をすべて表示。

調整説明：実際のニーズに応じて選択してください。調整効果を [調整の例](#) に示します。

変換設定

変換方法

パラメータ説明：このパラメータは、位置姿勢の変換方法を設定するために使用されます。

初期値：CameraToRobot

オプション

：CameraToRobot、RobotToCamera、AllWithFirst、FirstWithAll、UseCorrespondenceInput

調整説明：

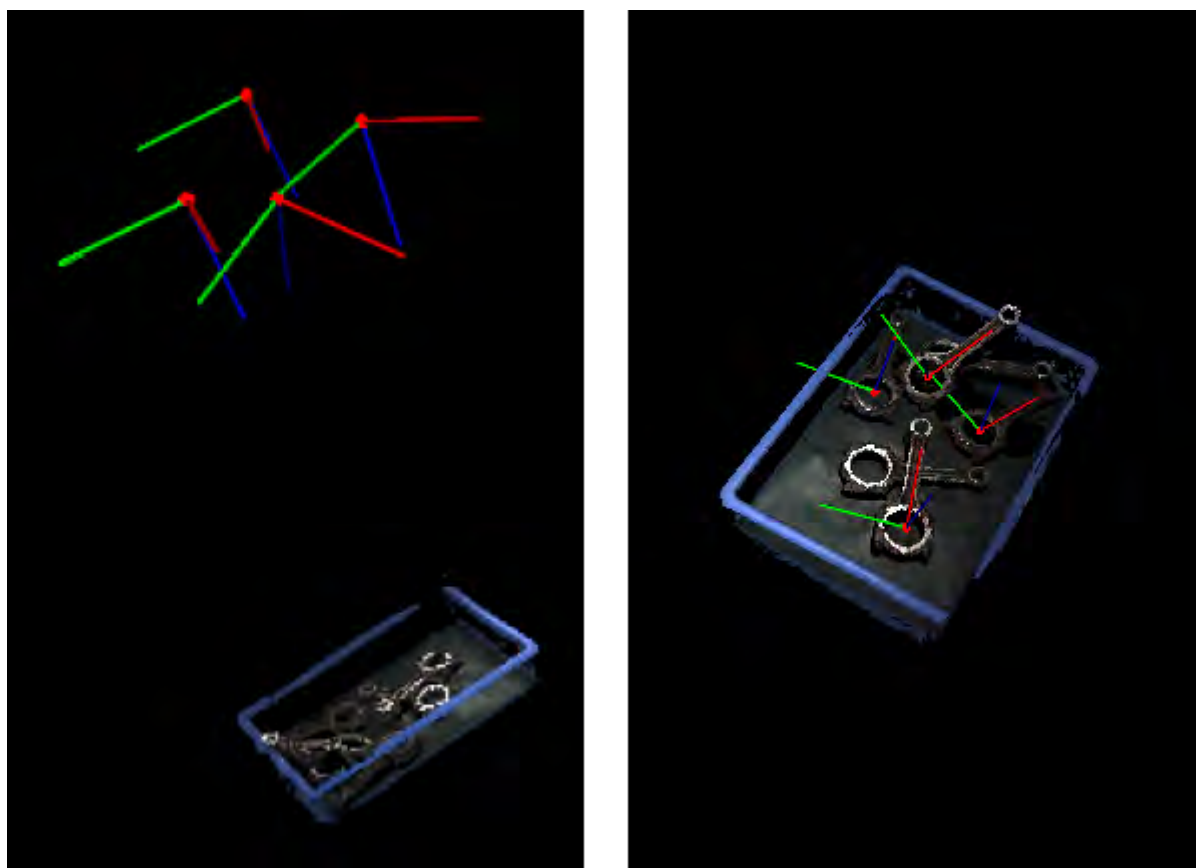
- CameraToRobot：カメラ座標系で対象物の位置姿勢はPose1で、「変換方法」を **CameraToRobot** に設定すると、ロボット座標系での対象物の位置姿勢（Pose2）を出力できます。調整効果を [調整の例](#) に示します。
- RobotToCamera：ロボット座標系で対象物の位置姿勢はPose1で、「変換方法」を **RobotToCamera** に設定すると、カメラ座標系での対象物の位置姿勢（Pose2）を出力できます。調整効果を [調整の例](#) に示します。
- AllWithFirst：座標系1で対象物A、Bの位置姿勢はそれぞれPose1A、Pose1Bで、「変換方法」を **AllWithFirst**（入力位置姿勢が必要）に設定すると、座標系2での対象物A、Bの位置姿勢（Pose2A、pose2B）を出力できます。この変換中に、変換されるすべての位置姿勢が最初の基準位置姿勢で変換されます。調整効果を [調整の例](#) に示します。
- FirstWithAll：座標系1で対象物A、Bの位置姿勢はそれぞれPose1A、Pose1Bで、「変換方法」を **FirstWithAll**（入力位置姿勢が必要）に設定すると、座標系2と座標系3における対象物Aの位置姿勢（Pose2AとPose3A）を出力できます。この変換中に、変換される最初の位置姿勢がすべての基準位置姿勢で変換されます。調整効果を [調整の例](#) に示します。
- UseCorrespondenceInput：座標系1で対象物A、Bの位置姿勢はそれぞれ

れPose1A、Pose1Bで、「変換方法」を **UseCorrespondenceInput**（入力位置姿勢が必要）に設定すると、座標系2での対象物Aの位置姿勢（Pose2A）と座標系3での対象物Bの位置姿勢（Pose3B）を出力できます。この変換中に、変換される位置姿勢と基準位置姿勢が1対1で変換されます。調整効果を [調整の例](#) に示します。

調整の例

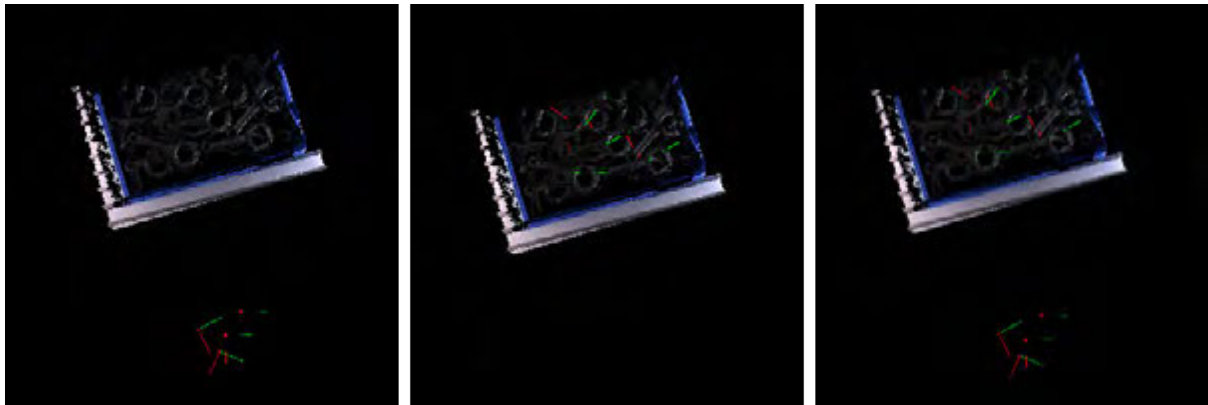
選択された座標系で点群を表示

下図に示すように、左側はカメラ座標系の点群で、右側はロボット座標系の点群です。



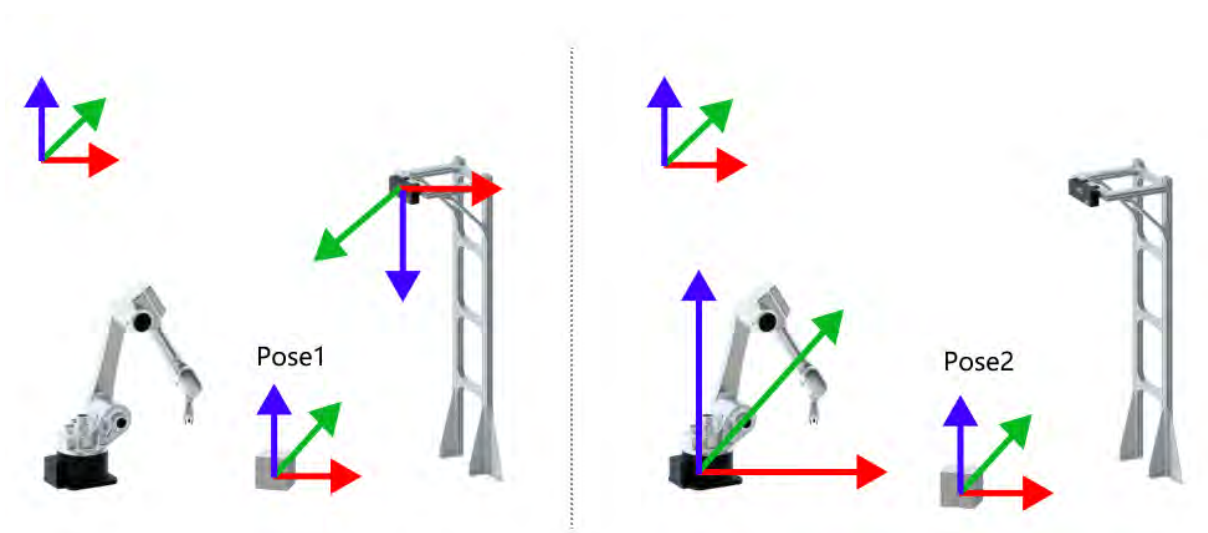
位置姿勢表示の設定

カメラ座標系を例として、**出力位置姿勢のみを表示**、**入力位置姿勢のみを表示**、**入力と出力を全て表示** を選択した場合の位置姿勢表示効果は下図のようになります。



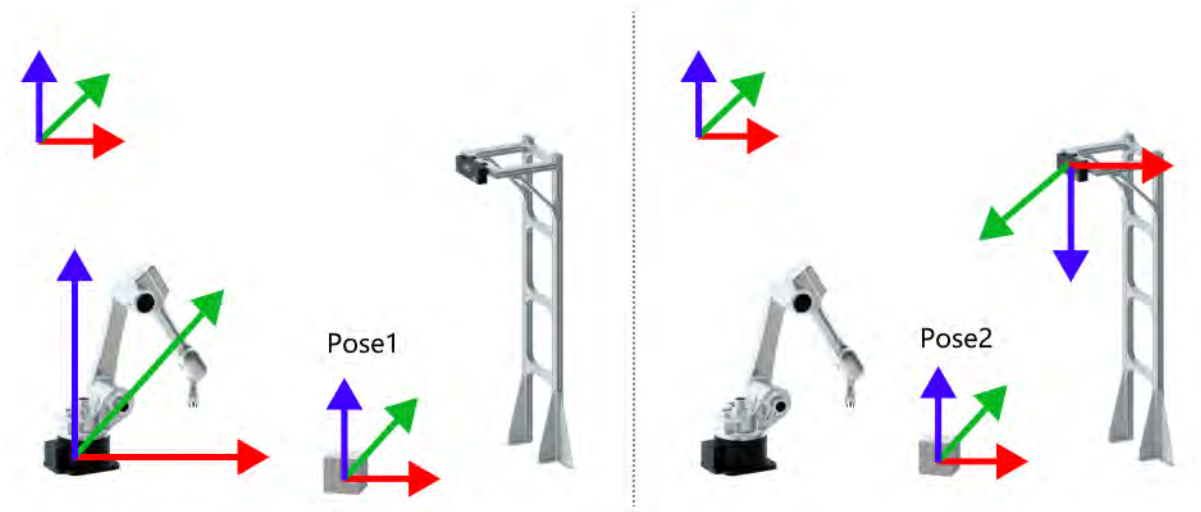
RobotToCamera

CameraToRobot に設定した場合の位置姿勢変換効果を下図に示します。左側は変換前で、右側は変換後です。



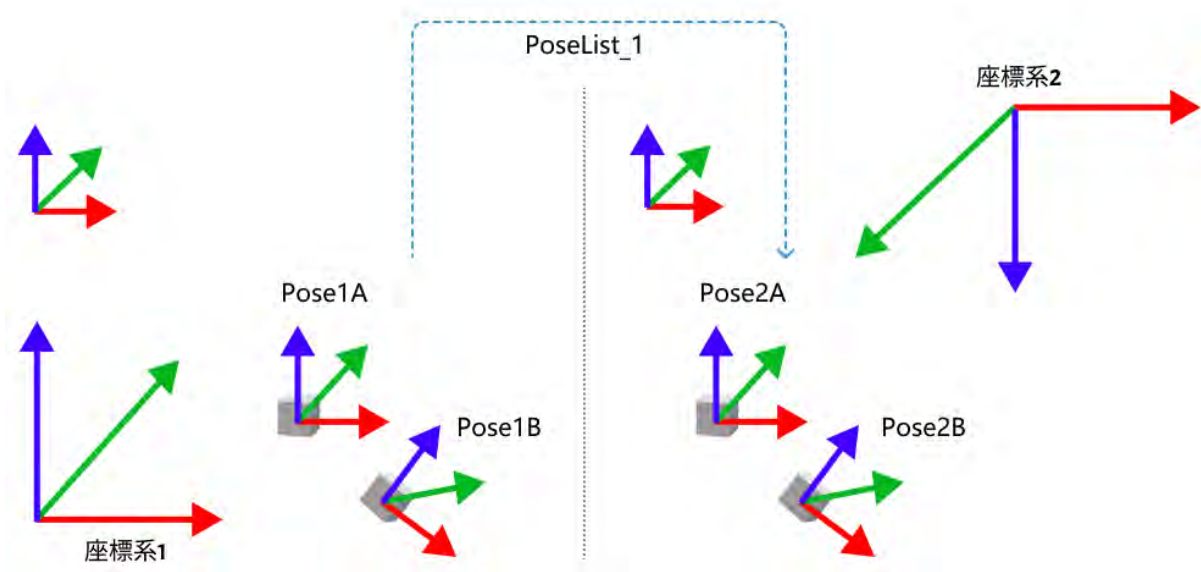
CameraToRobot

RobotToCamera に設定した場合の位置姿勢変換効果を下図に示します。左側は変換前で、右側は変換後です。



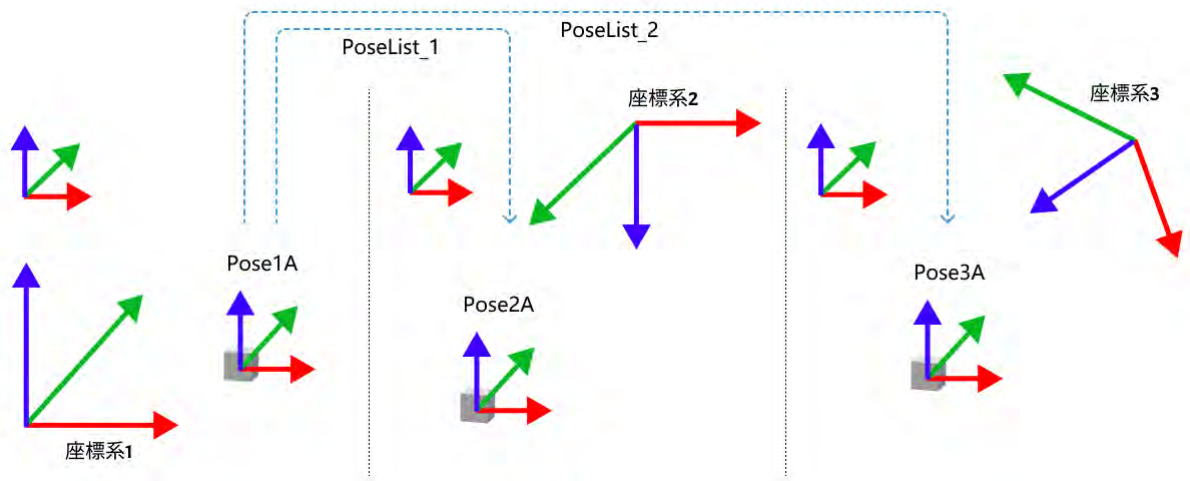
AllWithFirst

AllWithFirst に設定した場合の位置姿勢変換効果を下図に示します。左側は変換前で、右側は変換後です。



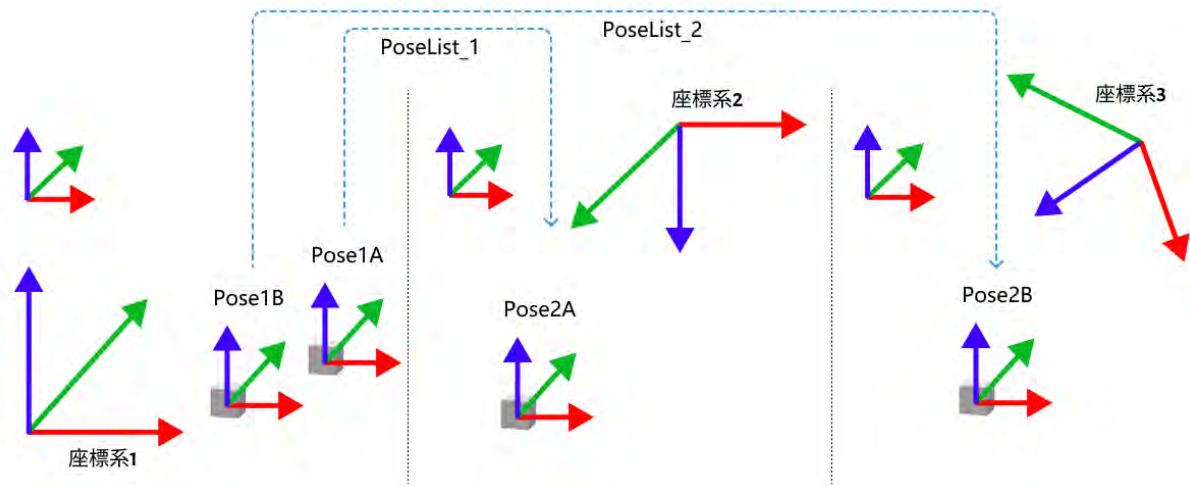
FirstWithAll

FirstWithAll に設定した場合の位置姿勢変換効果を下図に示します。



UseCorrespondenceInput

UseCorrespondenceInput に設定した場合の位置姿勢変換効果を下図に示します。



4.3.368. 基準方向に沿って位置姿勢を並進



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

入力位置姿勢を並進します。並進の方向および距離は、入力またはステップパラメータで設定できます。

使用シーン

汎用の位置姿勢並進ステップです。特定の使用シーンはありません。

入力と出力

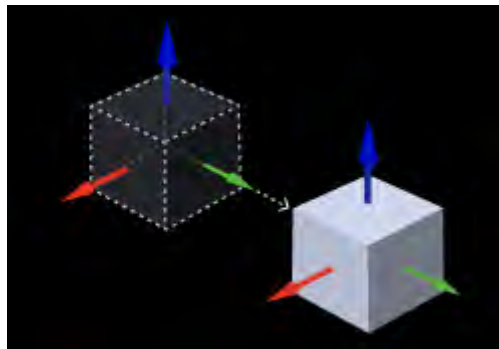
入力：

- 1. このポートに入力される位置姿勢は、並進されます。
- 2. カスタマイズされた位置姿勢の並進距離（選択可能。このポートが使用されていない場合、パラメータでの「移動距離」が使用されます）。
- 出力：
 1. 並進された位置姿勢。

4.3.369. 位置姿勢を指定方向に移動

機能

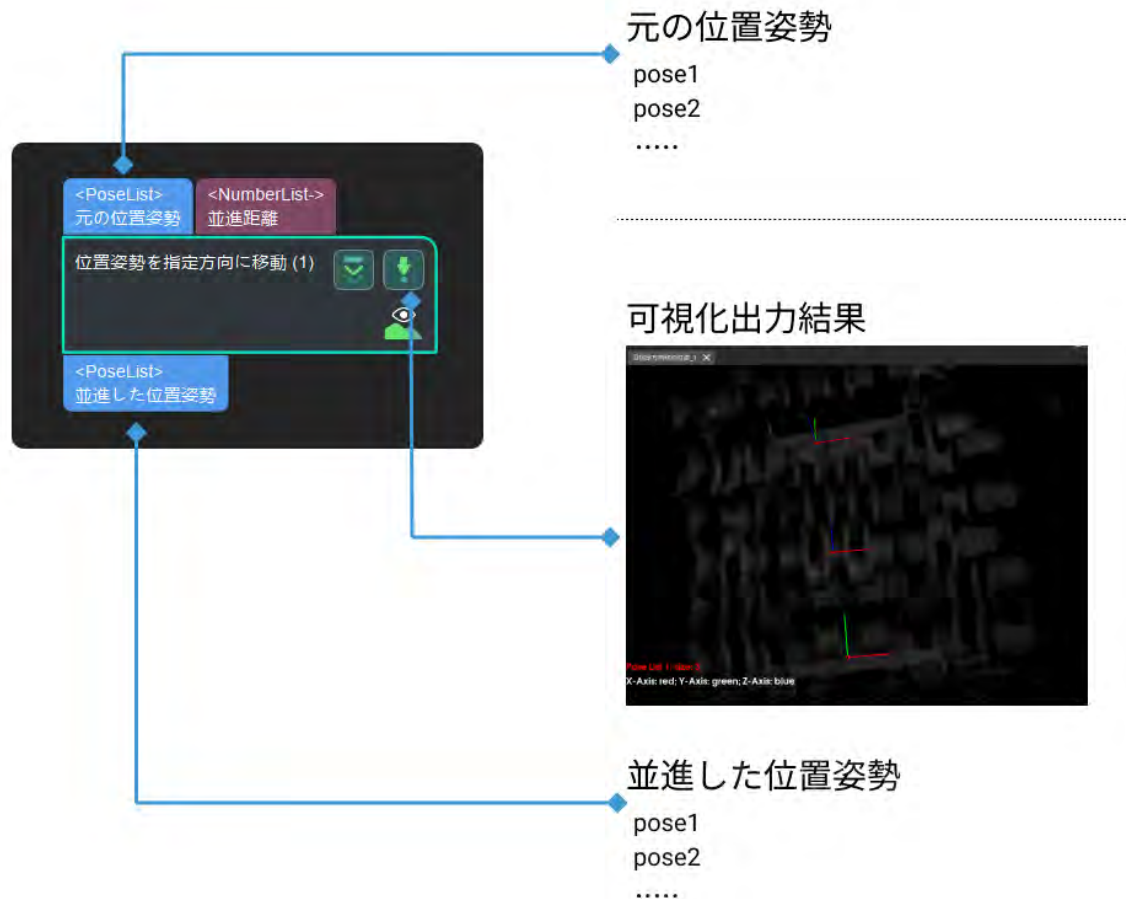
入力された位置姿勢を並進します。



使用シーン

ユーザーに指定された方向と距離によって位置姿勢を並進します。

入力と出力



パラメータの説明

可視化設定

選択された座標系で点群を表示

初期値：ロボット座標系。

オプション：ロボット座標系、カメラ座標系。

調整説明：詳細については、[選択された座標系で点群を表示](#) をご参照ください。

位置姿勢表示の設定

初期値：出力位置姿勢のみを表示

オプション：入力位置姿勢のみを表示、出力位置姿勢のみを表示、入力と出力位置姿勢をすべて表示。

調整説明：詳細については、[選択された座標系で点群を表示](#) をご参照ください。

並進方向の設定

移動距離

初期値：0

調整説明：実際の状況に応じて設定してください。

基準方向

初期値：CustomizedDirection（カスタマイズの方方向に沿って並進）

値のリスト：CustomizedDirection（カスタマイズの方方向に沿って並進）、ObjectAxis（物体の指定軸に沿って並進）

調整説明：実際の状況に応じて設定してください。

基準方向をObjectAxisに設定する場合、実際のニーズに指定軸を選択してください。



カスタマイズの方方向（基準方向をCustomizedDirectionに設定する場合のみに表示）

X

初期値：0

Y

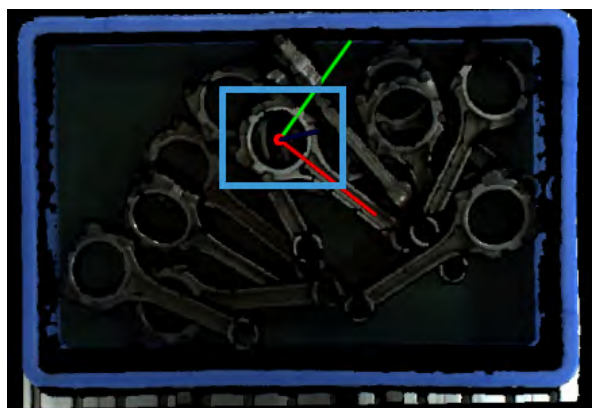
初期値：0

Z

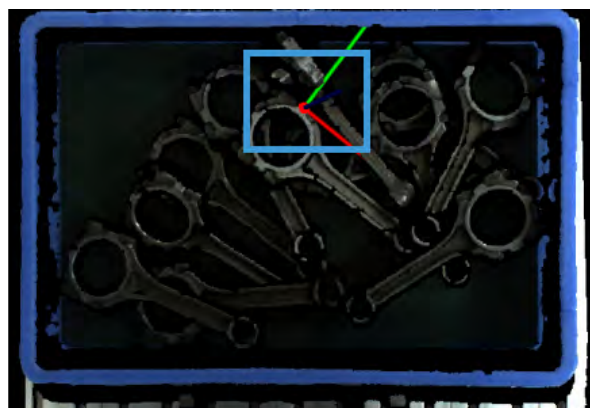
初期値：1

調整の例

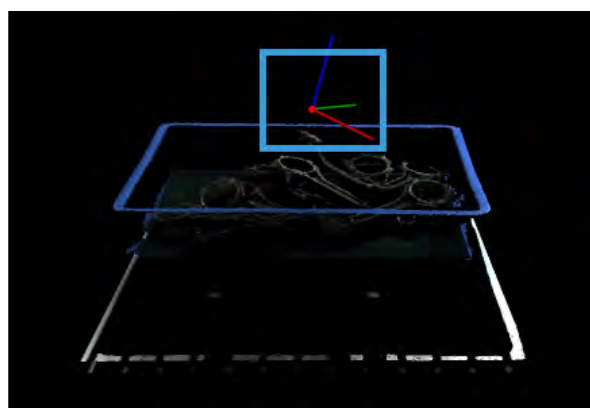
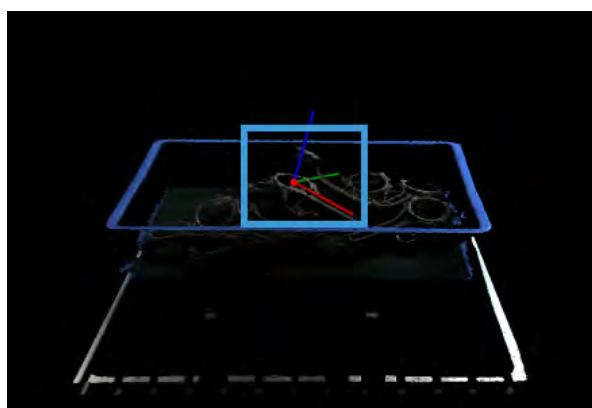
- 基準方向がObjectAxis、指定軸がX軸、並進距離がそれぞれ0と0.03の場合の位置姿勢並進効果は、下図のように示します。その中、左側の位置姿勢の並進距離は0で、右側は左側の位置姿勢をX軸に沿って0.03m移動させた効果です。



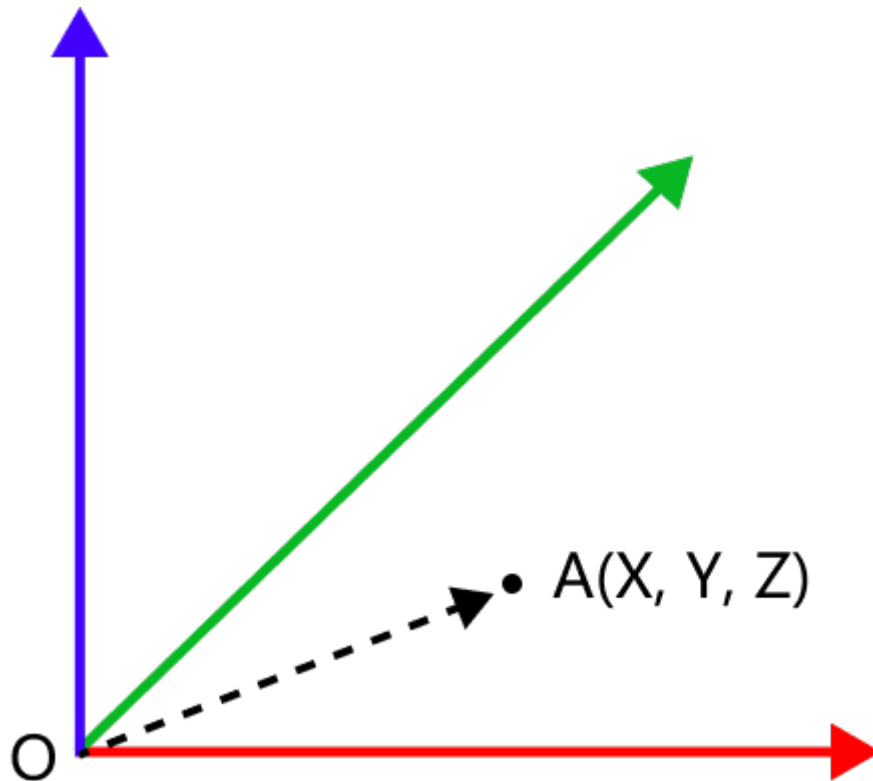
- 基準方向がObjectAxis、指定軸がY軸、並進距離がそれぞれ0と0.03の場合の位置姿勢並進効果は、下図のように示します。その中、左側の位置姿勢の並進距離は0で、右側は左側の位置姿勢をY軸に沿って0.03m移動させた効果です。



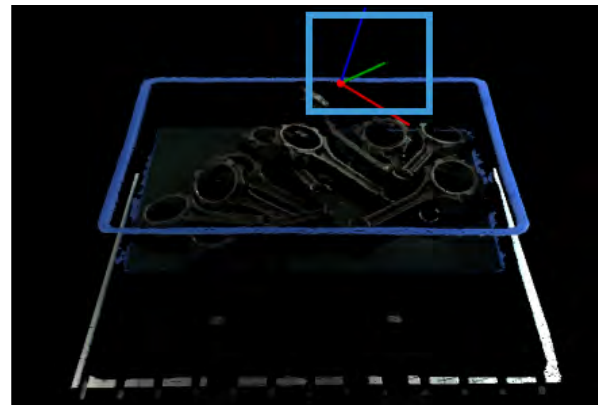
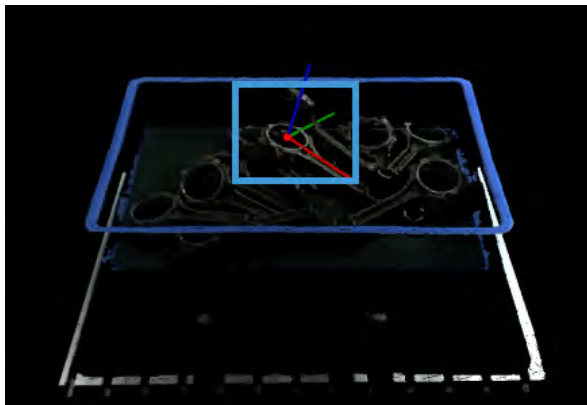
- 基準方向がObjectAxis、指定軸がZ軸、並進距離がそれぞれ0と0.08の場合の位置姿勢並進効果は、下図のように示します。その中、左側の位置姿勢の並進距離は0で、右側は左側の位置姿勢をZ軸に沿って0.08m移動させた効果です。



- 基準方向がCustomizedDirectionの場合、カスタマイズの方にX、Y、Z値を設定する必要があります。下図に示すように、原点OからA (X,Y,Z) に向かう方向が位置姿勢の並進方向となります。



X、Y、Zをそれぞれ5に設定し、並進距離を0.08mに設定した場合の位置姿勢並進効果は下図に示します。その中、左側は並進前の位置姿勢で、右側は左側に基づいてカスタマイズの方
向に0.08m並進した効果です。



4.3.370. 指定した方向と距離によって位置姿勢を並進



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

入力位置姿勢を並進します。並進の方向および距離は、入力またはステップパラメータで設定できます。

使用シーン

汎用の位置姿勢並進ステップです。特定の使用シーンはありません。

入力と出力

● 入力：

1. このポートに入力される位置姿勢は、並進されます。
2. 位置姿勢の並進方向を定義するためのカスタマイズベクトル（オプション。このポートが使用されていない場合、パラメータの設定が使用されます）。
3. カスタマイズされた位置姿勢の並進距離（オプション。このポートが使用されていない場合、パラメータでの「移動距離」が使用されます）。

● 出力：

1. 並進された位置姿勢。

4.3.371. 3Dベクトルで位置姿勢を並進



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

カスタマイズされた並進ベクトルで、入力位置姿勢を並進します。

使用シーン

汎用の位置姿勢並進ステップです。特定の使用シーンはありません。

入力と出力

● 入力：

1. このポートに入力される位置姿勢は、並進されます。
2. 位置姿勢を並進するためのカスタマイズされたベクトル。

● 出力：

1. 並進された位置姿勢。

4.3.372. ローカル座標系で位置姿勢を並進



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

入力された元の位置姿勢を、指定軸の方向に沿って並進します。並進距離は、入力もしくはパラメータで設定できます。

使用シーン

汎用の位置姿勢並進ステップです。特定の使用シーンはありません。

入力と出力

● 入力：

1. このポートに入力される位置姿勢は、並進されます。
2. 位置姿勢の並進距離（オプション。このポートが使用されていない場合、パラメータの設定が使用されます）。

● 出力：

1. 並進された位置姿勢。

4.3.373. ブール値によるトリガー制御



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

制御フローを、入力されたブール値（True/False）および指定されたルールによってトリガーします。

使用シーン

プロジェクトの分岐構造に使用され、様々なルールに従ってその後の様々なステップをトリガーします。通常、[\[vision-steps:dichotomize-values-by-threshold:::dichotomize-values-by-threshold\]](#) のようなブール値を出力するステップと併用されます。

入力と出力



4.3.374. トリガー

機能

制御フローで後続のステップの実行をトリガーします。

使用シーン

特定のシーン向けのカスタマイズのステップです。

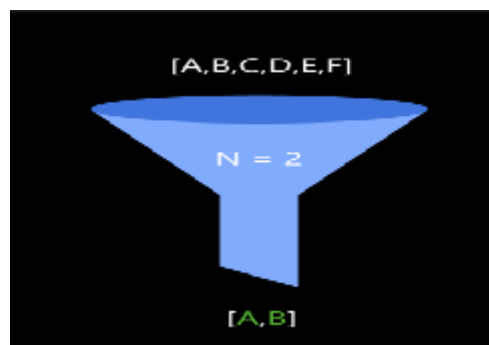
入力と出力

- 入力： なし。
- 出力： なし。

4.3.375. 出力の数を制限

機能

データリストの最初のN項目を保持して出力します。第N項目以降のデータは破棄されます。

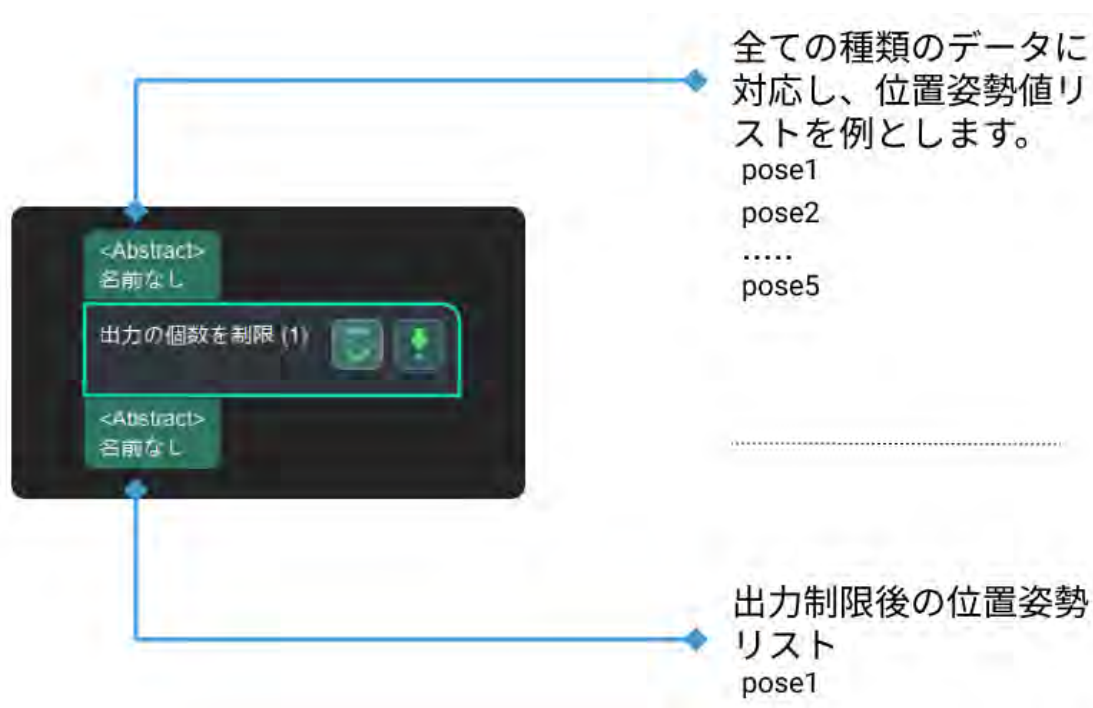


使用シーン

リストの最初のN項目を抽出し、Nの値はニーズに応じてパラメータで設定可能です。

入力と出力

このステップの入力と出力のデータタイプは同じです。入力位置姿勢データを例にとると、入力と出力を下图に示します。



パラメータの説明

ポート設定

ポート数 (1-15)

パラメータ説明：このパラメータは、このステップの入力・出力ポートの数を設定するために使用されます。

初期値：1

値の範囲：1-15

操作レイヤー

パラメータ説明：このパラメータは、リスト内のどのレベルの要素に対して操作を行うかを指定するために使用されます。操作レイヤーが0の場合、操作はリスト内の各要素に対して実行されます。このパラメータが0でない場合、リストのそのレベルの要素に対して操作が行われます。

初期値：0

出力結果の数

パラメータ説明：このパラメータは、出力結果の数を設定するために使用されます。

初期値：1

調整アドバイス：実際の状況に応じて設定してください。

4.3.376. 位置姿勢の数を制限



このステップは古いバージョンであり、メンテナンスが停止されていますので、[\[vision-steps:trim-input-list::trim-input-list\]](#) を使用してください。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

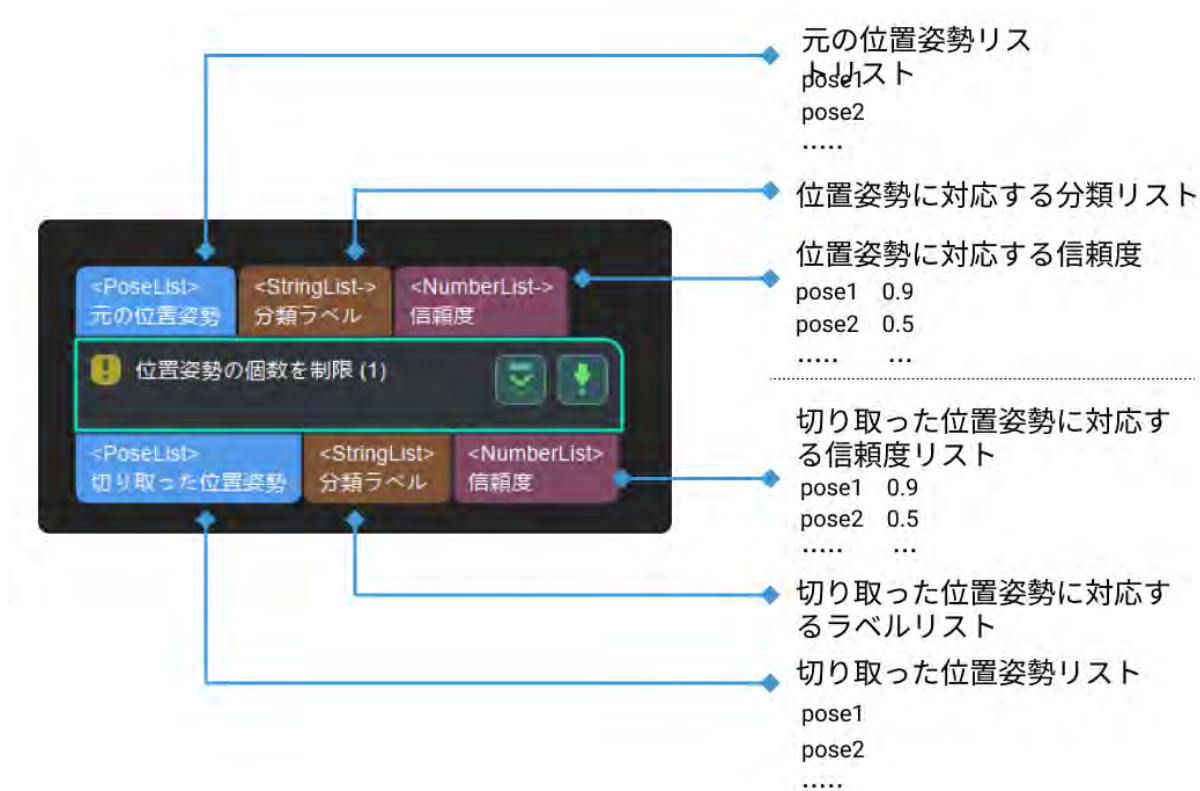
機能

位置姿勢リストの最初のN項目を保持して出力します。第N項目以降のデータは破棄されます。

使用シーン

リストの最初のN項目を抽出し、Nの値はニーズに応じてパラメータの中で設定可能です。

入力と出力



パラメータの説明

出力位置姿勢の数

初期値：1

調整説明：出力位置姿勢の数を指定します。

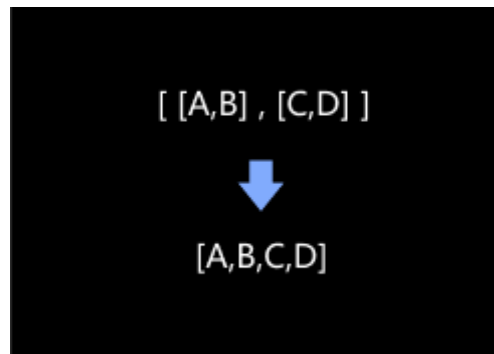
4.3.377. データをアンパックしてマージ



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

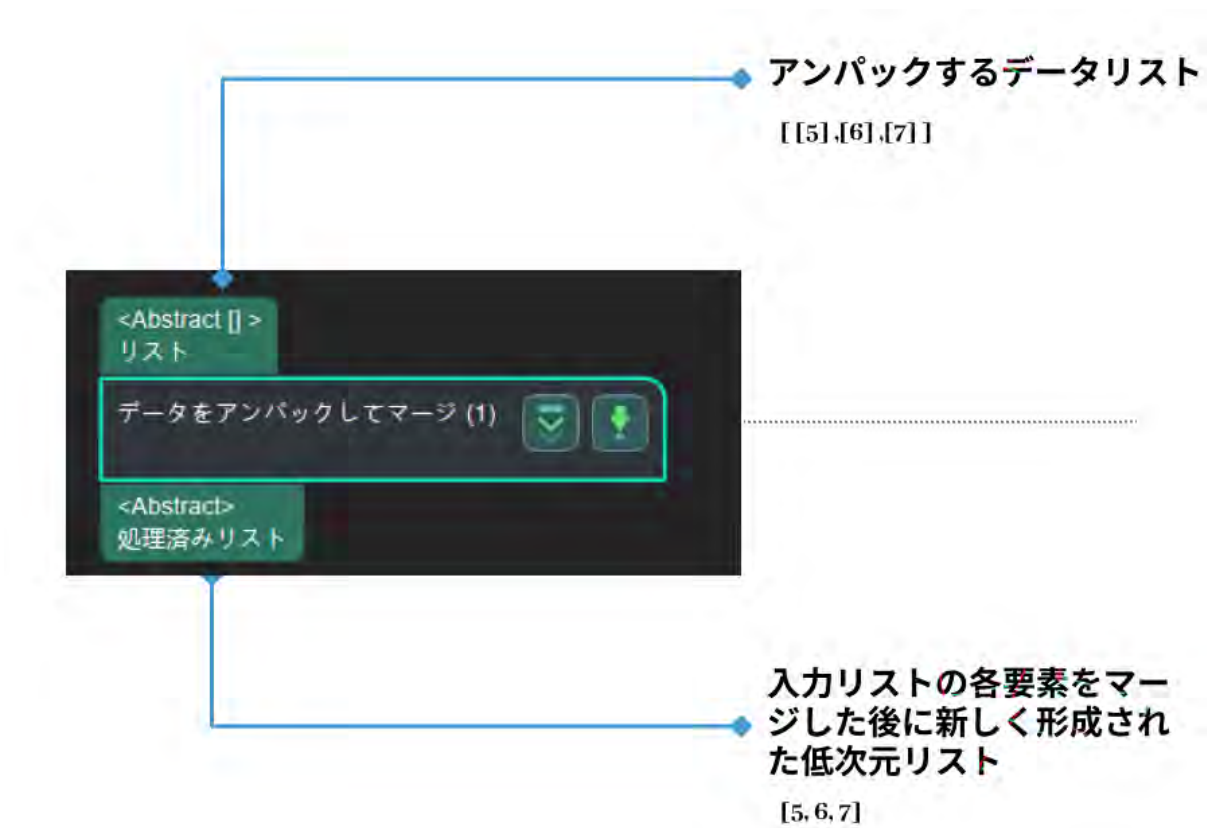
高次元リストを分解し、その中のサブリストをマージします。



使用シーン

汎用のデータ次元削減ステップです。このステップによって、データリストを分解してから、分解された要素をマージすることができます。これは通常、高次元データの次元削減に使用されます。例えば、入力リストが`[[5], [6], [7]]`の場合、`[5, 6, 7]`を出力できます。

入力と出力



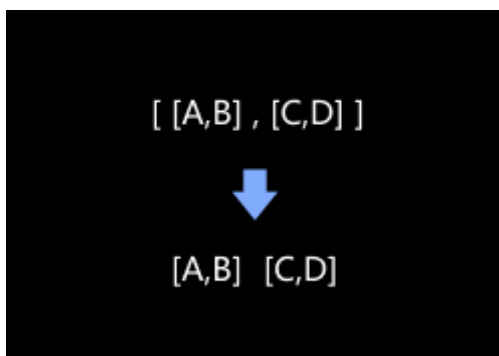
4.3.378. データをアンパック



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

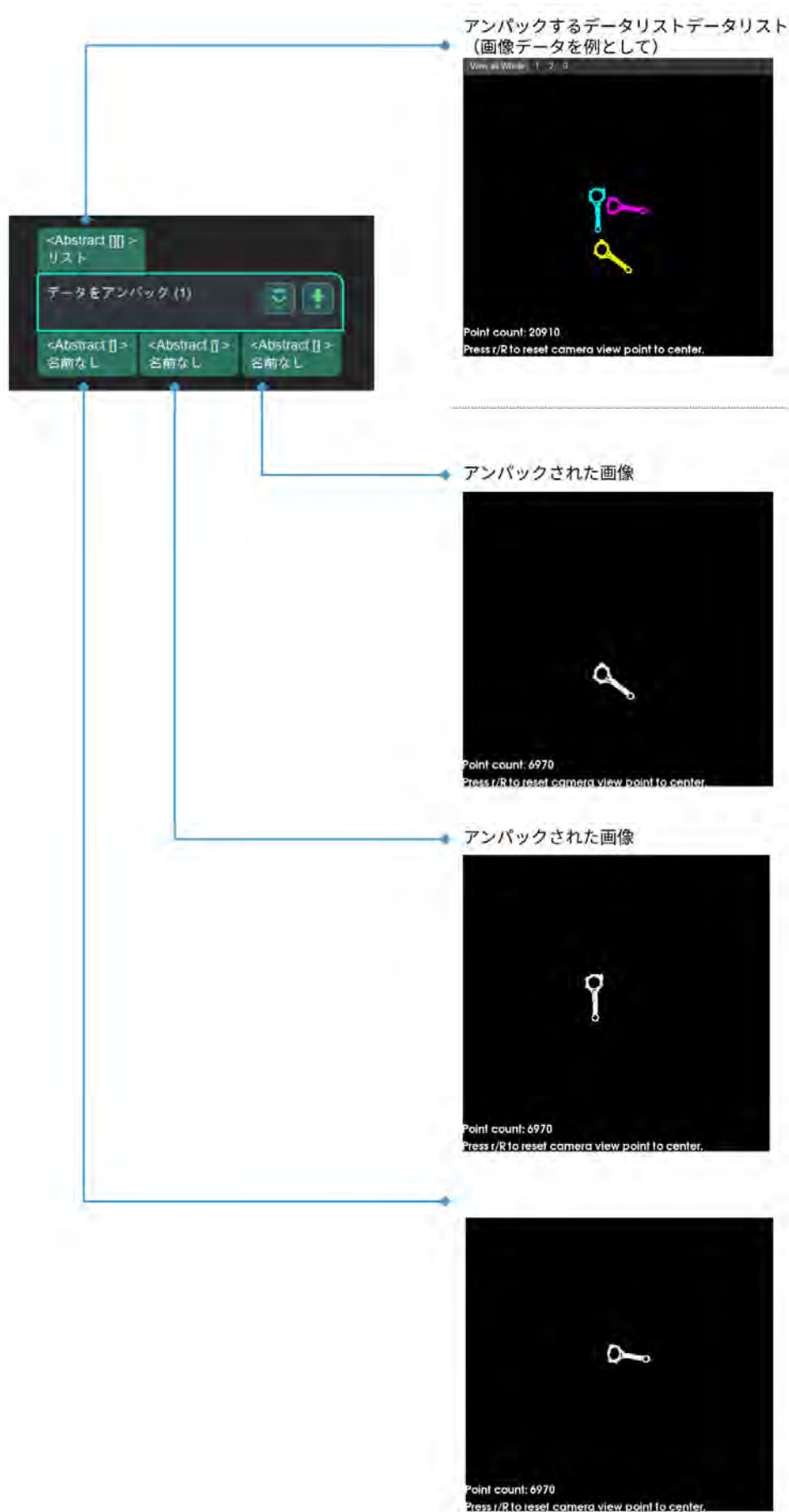
リストを分解し、リストの最初のN個の要素を取得します。N値はパラメータで設定可能です。



使用シーン

汎用のデータ次元削減ステップです。このステップでは、画像リスト、点群リスト、寸法リストなどのデータリストを分解できます。例えば、入力リストが\[[5], [6], [7]]の場合、[5], [6], [7]を出力できます。

入力と出力



4.3.379. マスクによって2D位置姿勢を検証

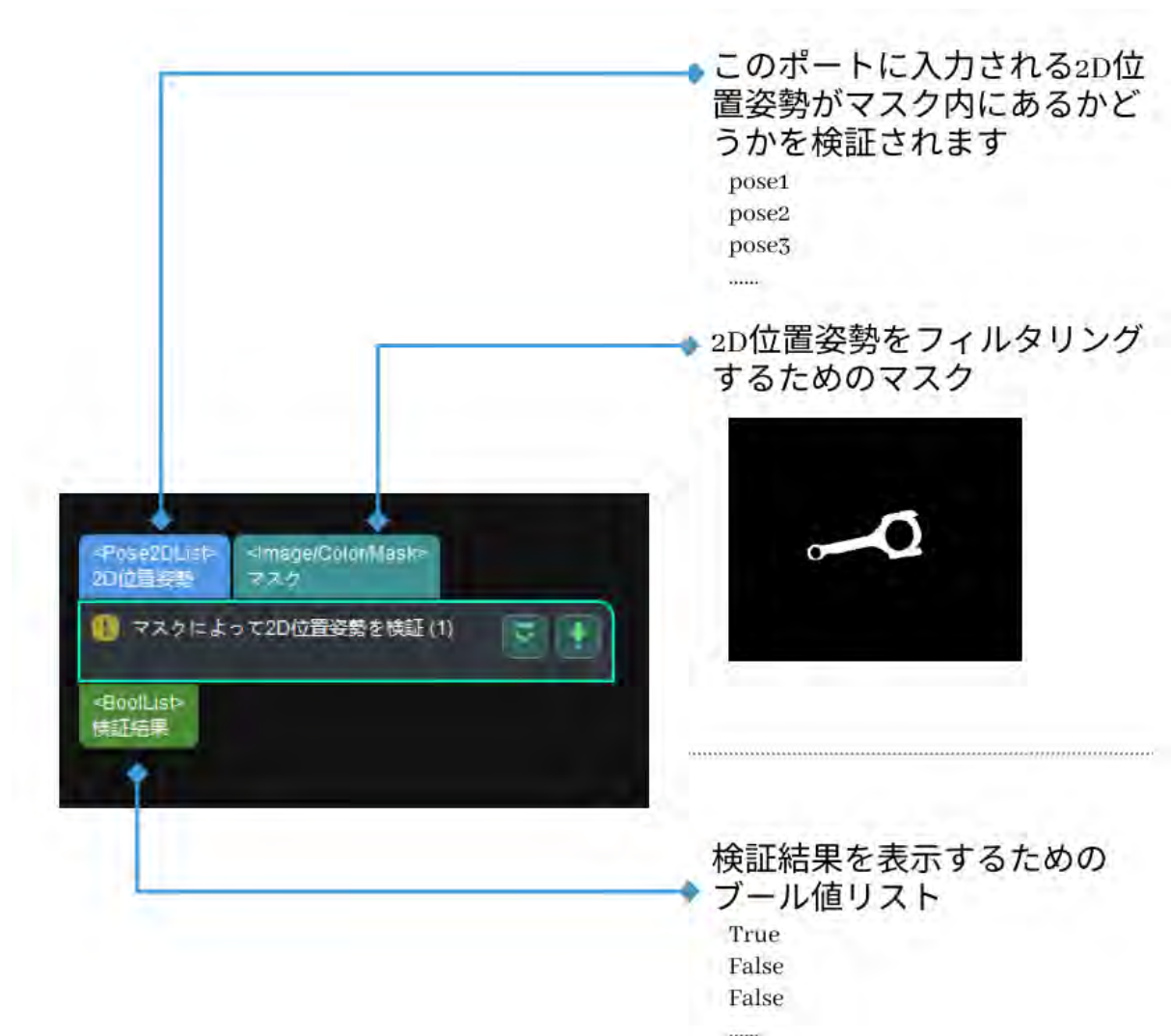
機能

各入力の2D位置姿勢が入力のマスク領域内にあるかどうかを検証します。

使用シーン

特定のシーン向けのカスタマイズのステップです。

入力と出力



4.3.380. カメラのパラメータ補正を検証して計算



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

パラメータで設定された基準位置姿勢を使用して、操作中にカメラパラメータを自動的に補正します。

使用シーン

時間の経過とともに大きくなるカメラの系統的誤差を補正するために使用されます。通常、ステップ「マーカーを検出」と併用されます。

入力と出力

- **入力：** なし。
- **出力：** 基準位置姿勢に基づいて計算されたカメラの外部パラメータへの補正。

4.3.381. 箱の寸法が有効であるかどうかを検証



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

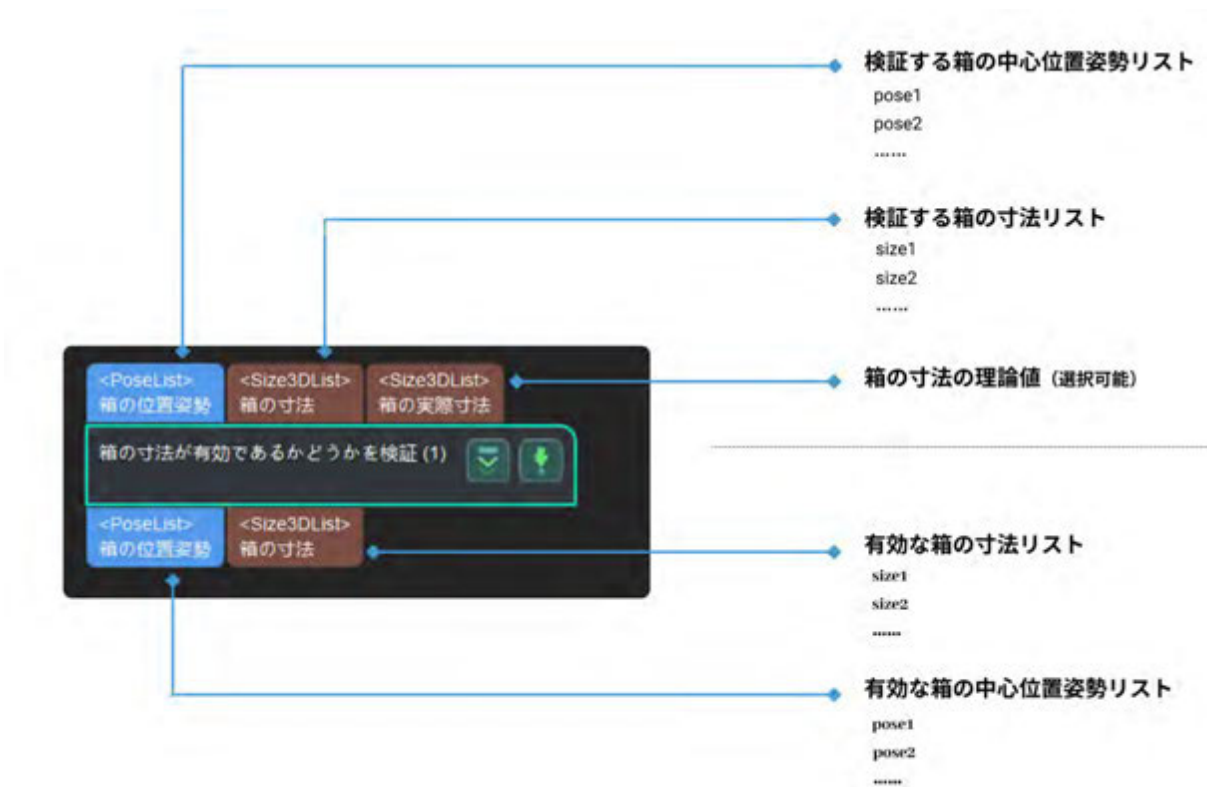
機能

基準寸法に基づいて、入力位置姿勢に対応する箱の寸法が有効かどうかを判断し、寸法誤差の大きい箱に異常ラベルを付けます。

使用シーン

このステップは、箱のデパレタイジングシーンに認識された箱が有効かどうかを判断するために使用されます。通常、ステップ [\[vision-steps:calc-poses-and-dimensions-from-planar-point-clouds:::calc-poses-and-dimensions-from-planar-point-clouds\]](#) の後で使用されます。

入力と出力



4.3.382. 箱形状の対象物のマスクを検証



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社 (docs@mech-mind.net) までお問い合わせください。

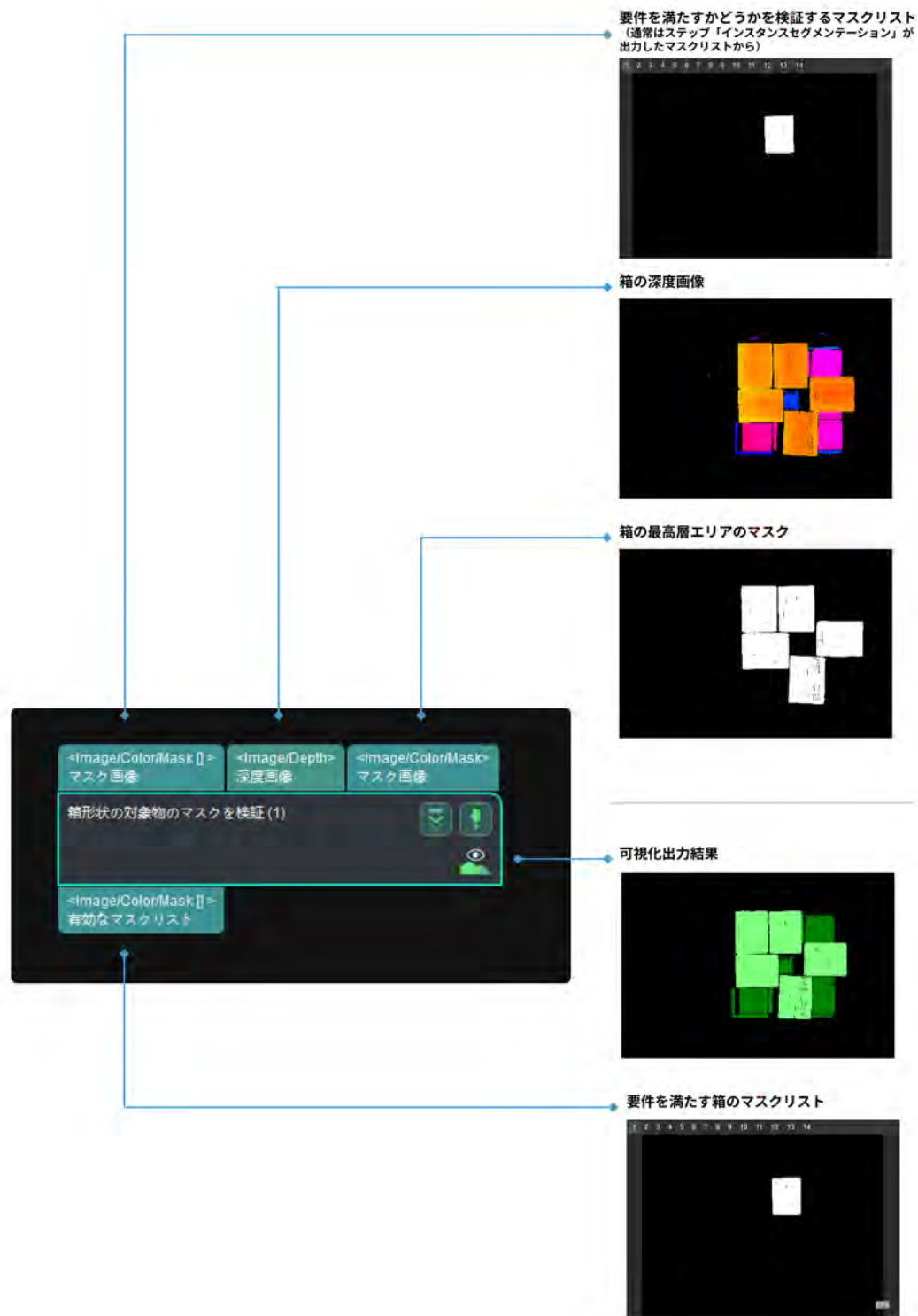
機能

指定されたルールに従って箱形状のマスクが要件を満たしているかどうかを検証して、要件を満たすマスクリストを出力します。

使用シーン

通常、箱のデパレタイジングに使用され、箱のマスクが要件を満たしているかどうかを検証します。

入力と出力



4.3.383. 位置姿勢が3D ROI内にあるかどうかを検証



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社 (docs@mech-mind.net) までお問い合わせください。

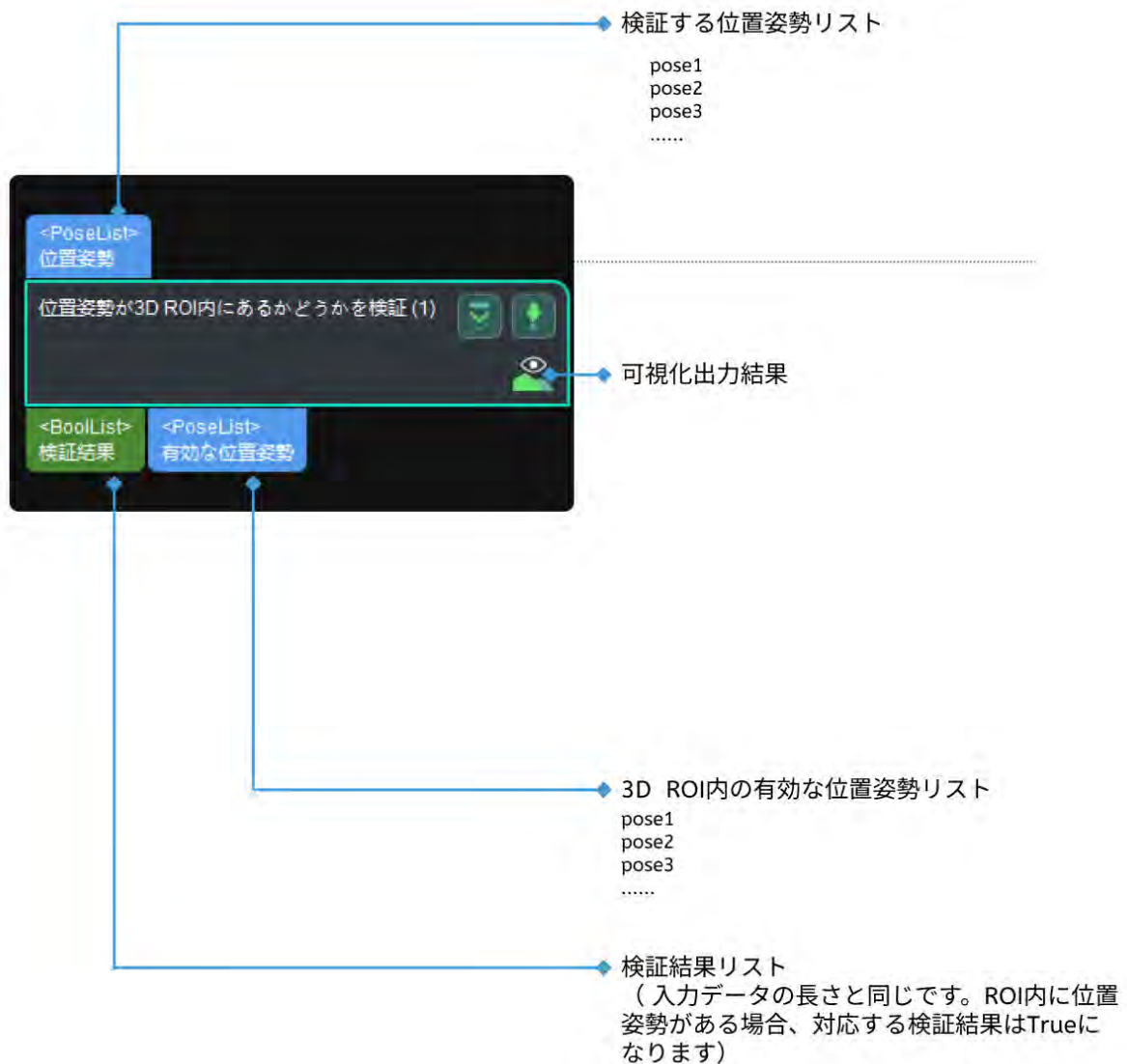
機能

入力された位置姿勢リストは設定され3D ROI内にあるかどうかを検証して、検証結果および3D ROI内の位置姿勢を出力します。

使用シーン

汎用の位置姿勢フィルタリングのステップです。3D ROIによってフィルタリングし、その結果を出力します。

入力と出力



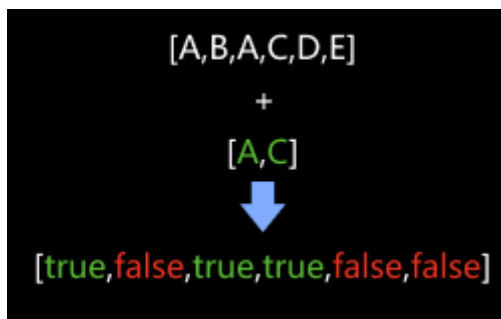
4.3.384. 必要なラベルであるかどうかを検証

機能

入力された元のラベルと基準ラベルリストを比較し、比較結果をブール値のリストとして出力

できます。

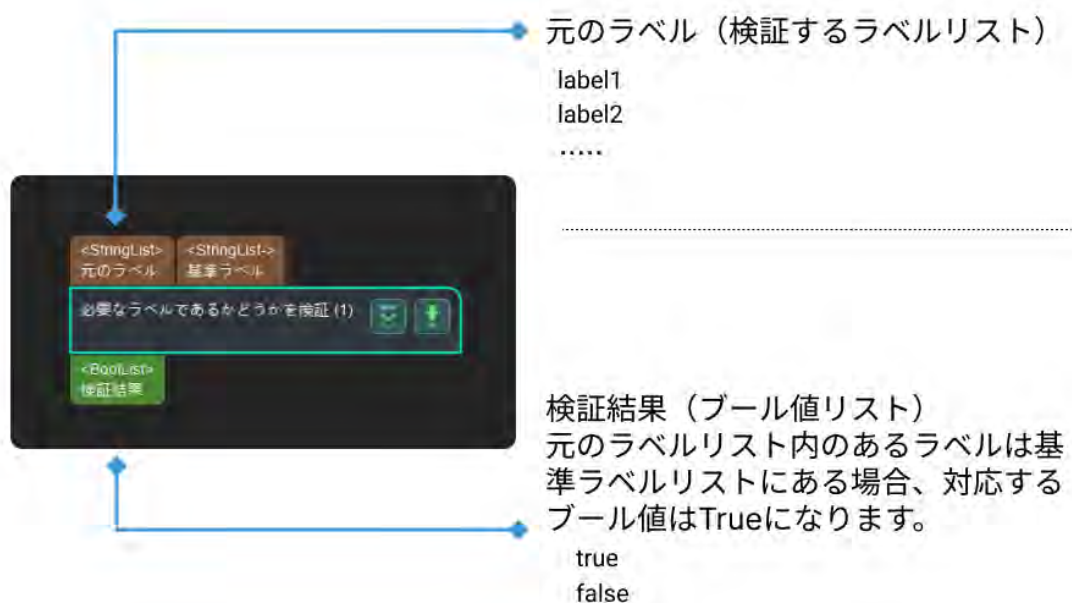
- 入力された元のラベルが基準ラベルリストにある場合、対応する出力はTrueとなります。
- 入力された元のラベルが基準ラベルリストにない場合、対応する出力はFalseとなります。



使用シーン

通常、`[vision-steps:filter:::filter]` ステップと併用されます。このステップで出力されるブール値リストは、配列のフィルタリングの基準として使用されます。

入力と出力



パラメータの説明

ラベルファイル名

パラメータ説明：このパラメータは、基準ラベルファイルを設定するために使用されます。

初期値：labelFilter.json

ラベル名：文字列（特殊文字不可）



入力されたデータが要件を満たさない場合は、無効なデータとみなされ、正常に入力することができません。

設定方法：

1. **[基準ラベルを設定]** をクリックし、JSONエディタの画面に入ります。
2. **+** をクリックし、**項目を追加** の画面で基準ラベルの名前を入力し、**[OK]** をクリックします。すると、基準ラベルが正常に追加されました。



ラベル名に小数点以下の数字や特殊文字を入力すると、ラベルの追加に失敗し、リストの下に無効な入力が表示されます。

3. 複数の基準ラベルを設定するが必要な場合、手順2を繰り返してください。
4. **ディレクトリ** は、現在のプロジェクトのパスです。**[...]** をクリックしてプロジェクトの場所を変更することができます。実際の状況に応じて設定してください。
5. **JSONファイル名** にファイル名を入力し、**[保存]** をクリックします。すると、ラベルがこのファイルに保存されます。
6. **[OK]** をクリックすると、基準ラベルファイルの設定が完了しました。



基準ラベルが事前に設定されている場合は、**JSONファイル名** の **[...]** をクリックし、ファイルを選択して基準ラベルファイルを読み込むことができます。

4.3.385. マスクに2D位置姿勢があるかどうかを検証

機能

各入力マスクに入力の2D位置姿勢が含まれているかどうかを検証します。

使用シーン

特定のシーン向けのカスタマイズのステップです。

入力と出力

● 入力：

1. このポートに入力されるマスクには、位置姿勢が含まれているかどうかを検証されます。
2. マスクを検証するための位置姿勢。

● 出力：

1. 検証結果を表示するブール値のリスト。Trueはマスクに位置姿勢があることを意味します。Falseはマスクに位置姿勢がないことを意味します。

4.3.386. 点群が要求を満たすかどうかを検証



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社 (docs@mech-mind.net) までお問い合わせください。

機能

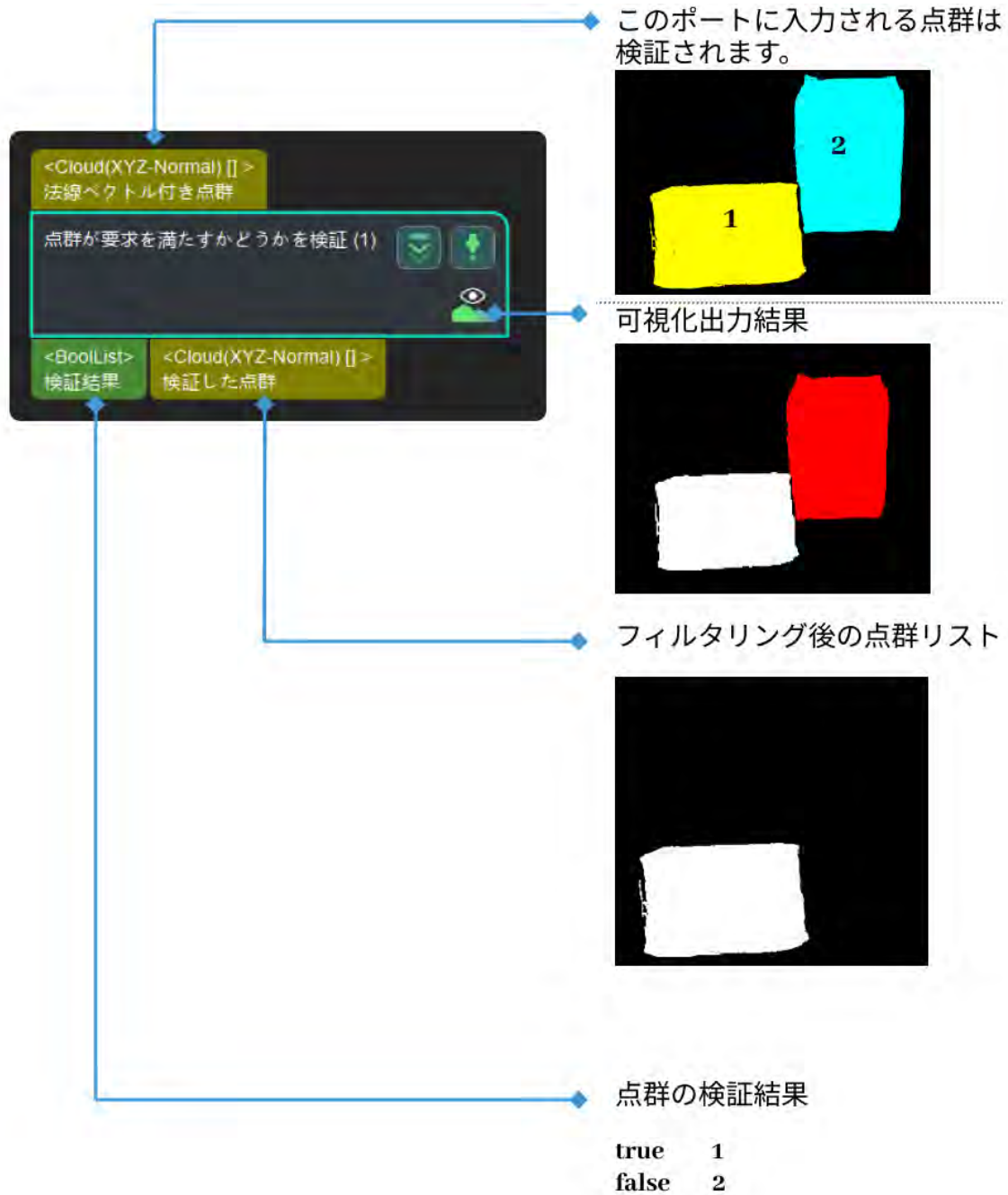
設定されたルールに従って点群が要件を満たしているかどうかを判断します。

使用シーン

通常、点群の点数によって点群が要件を満たしているかどうかを判断するために使用されます（パラメータ **フィルタリング方法** を **CloudCapacityFilter** に設定してください）。

検証が完了後、ステップ [\[vision-steps:filter:::filter\]](#) を使用して結果によって要件を満たさない点群を除去します。

入力と出力



4.3.387. 基準方向との角度によって位置姿勢が有効かどうかを検証



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

位置姿勢の指定軸と基準方向との角度を計算します。設定された最大角度差より小さい角度が対応する位置姿勢は保持され、判断結果はTrueになります。設定された最大角度差より大きい角度が対応する位置姿勢は除去され、判断結果はFalseになります。

使用シーン

汎用の位置姿勢フィルタリングのステップで、特定の使用シーンはありません。

入力と出力



4.3.388. 位置姿勢が遮蔽されているかを検証



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

入力位置姿勢が他の位置姿勢の下にあるかどうかを検証します。入力位置姿勢は異なるグループに分けられ、現在の位置姿勢が前の位置姿勢のグループの下にある場合にのみ、位置姿勢は対応する対象物が遮られているかどうかを検証するために使用されます。

使用シーン

特定のシーン向けのカスタマイズのステップです。

入力と出力

● 入力：

1. このポートに入力される位置姿勢は、対応する対象物が遮られているかどうかを検証するために使用されます（グループ内ではなく、グループとグループ間の検証です）。
2. 対象物が遮られているかどうかを検証するための半径の値。

● 出力：

1. 判定結果を表示するブール値リスト。

4.3.389. 法線ベクトルの偏差が大きい領域を抽出



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

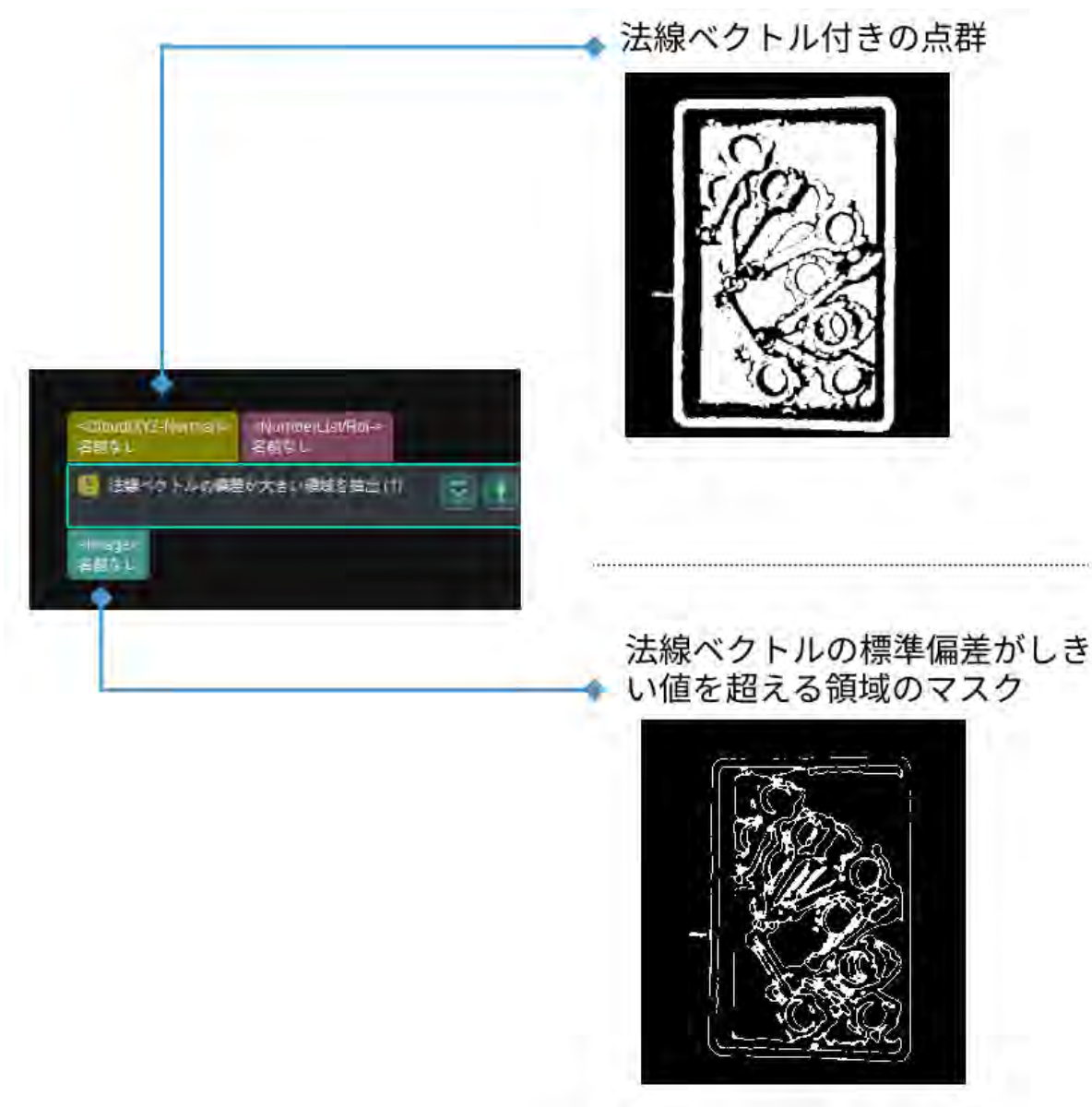
機能

法線ベクトルの標準偏差が大きい領域を抽出します。最初に法線ベクトルを2D画像にマッピングし、次に2Dフィルターで局所標準偏差を計算し、設定されたしきい値よりも大きい標準偏差を持つ領域のマスクを生成します。

使用シーン

特定のシーン向けのカスタマイズのステップです。

入力と出力



4.3.390. 3Dベクトル演算



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

ほかのセットの3Dベクトル（または数値）を使用して、1つセットのVector3D型の3Dベクトルに対して簡単な演算を行います。

使用シーン

汎用の3Dベクトルの計算ステップです。特定の使用シーンはありません。

入力と出力

- 入力：
 1. 演算するための最初セットの3Dベクトル。
 2. 演算するための2番目セットの3Dベクトル。
- 出力：
 1. 演算によって得られた3Dベクトル。

4.3.391. 把持位置姿勢を検証



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

入力された把持位置姿勢とファイル内の把持位置姿勢との誤差を計算します。

使用シーン

特定のシーン向けのカスタマイズのステップです。

入力と出力

- 入力：
 1. このポートに入力される把持位置姿勢と、パラメータで指定されたファイルの把持位置姿勢との誤差が計算されます。
- 出力：
 1. 計算された把持位置姿勢の誤差。

4.3.392. 範囲内の深度を可視化



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

ROI内の深度画像をカラー画像に変換します。その中、異なる深度値は異なる色のピクセル値に対応します。

使用シーン

これは通常、計測シーンで深度画像をカラー画像に変換して、後続の計測またはインターフェイス表示に使用されます。

入力と出力

- **入力：** このポートに入力される深度画像は、カラー画像に変換されます。
- **出力：** 変換によって得られたカラー画像。

4.3.393. 画像で情報を可視化



このステップの内容は、現在メンテナンス中です。このステップに関する詳細が必要な場合は、当社（docs@mech-mind.net）までお問い合わせください。

機能

カラー画像でマスク、位置姿勢およびラベルを描画します。

使用シーン

通常は、その後のインターフェイス表示に使用されます。

入力と出力

- **入力：**
 1. 他のポートの入力情報は、このポートに入力された画像で可視化されます。
 2. このポートに入力されるマスクは可視化されます。
 3. このポートに入力される位置姿勢は可視化されます。
 4. このポートに入力される分類ラベルは可視化されます。
- **出力：**
 1. 各ポートの入力情報が画像で可視化された結果。

4.3.394. 大型部品の測定

大型部品とは、長さが0.5mを超える部品を指します。通常は、自動車、飛行機、船舶などの製造業向けの部品です。測定の適用シーンでは、通常、部品の丸穴、長穴、平面などの適合性を検査するために使用されます。Mech-Visionでは、大型部品の測定に対応可能なステップの組合せが組み込まれています。これにより、プロジェクトを迅速に構築して実装することが可能です。



大型部品の測定には高い精度が要求されるため、高精度なカメラが必要です。Mech-Mindから発売されているUHPシリーズのカメラを推奨しています。

本節では、大型部品の測定プロジェクトの構築手順と関連するステップの組合せについて説明します。

大型部品の測定プロジェクトを構築する方法 については、以下の内容をお読みください。

大型部品の測定プロジェクトの構築手順

大型部品の測定プロジェクトに関連する **ステップの組合せ** については、以下の内容をお読みください。

- 深度画像から点群を分割
 - 平面点群を抽出
 - 丸穴の中心位置姿勢と直径を計算
 - 長穴の中心位置姿勢と長軸・短軸を計算
 - 位置姿勢をカスタマイズされた座標系に変換
-

本節の説明は終わります。次の節は、大型部品の測定プロジェクトの構築方法の説明です。

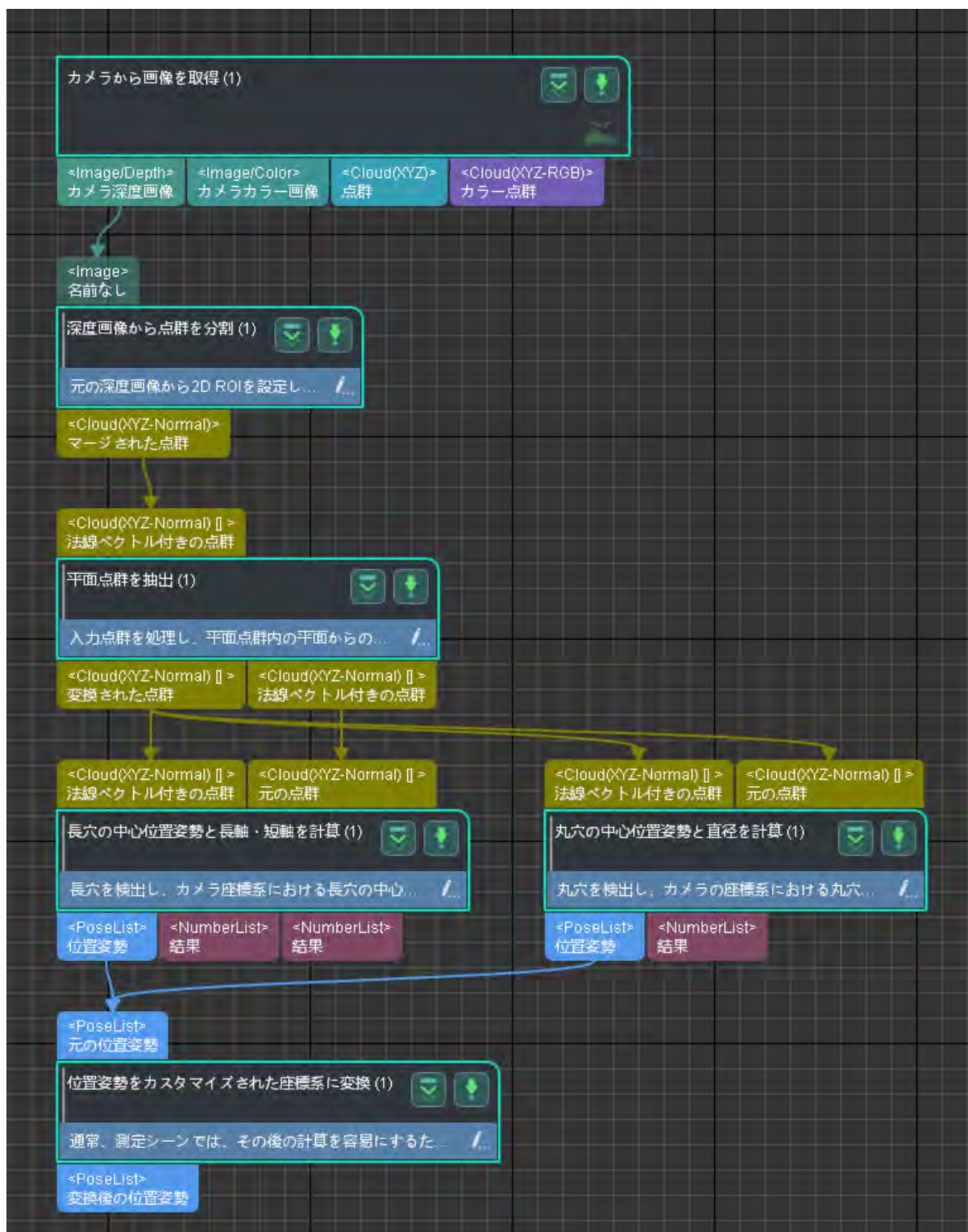
4.3.394.1. 大型部品の測定プロジェクトの構築手順

本節では、**大型部品** の測定プロジェクトの構築手順や原理、参考情報について説明します。

プロジェクトの構築手順

今回作成するプロジェクトは以下のような流れで処理を行います。

1. カメラから画像を取得
2. 深度画像から点群を分割
3. 平面点群を抽出
4. 丸穴の中心位置姿勢と直径を計算 または 長穴の中心位置姿勢と長軸・短軸を計算
5. 位置姿勢をカスタマイズされた座標系に変換



部品の特徴に応じて必要なステップを選択してください。丸穴を測定する場合、**丸穴の中心位置姿勢と直径を計算**を使用します。長穴を測定する場合、**長穴の中心位置姿勢と長軸・短軸を計算**を使用します。誤った使用すれば、誤った測定結果になることがあります。

原理

大型部品の測定プロジェクトでは、カメラ座標系での部品の位置姿勢を **基準座標系** にある位置姿勢を変換します。データ流れは以下の通りです。

1. 高精度のカメラで画像を取得し、Mech-Visionの処理によりカメラ座標系での部品の丸穴または長穴の位置姿勢を取得します。
2. お客様から提供された部品図面に基づいて部品座標系を確立し、カメラ座標系での特徴位置姿勢を部品座標系に変換して表示します。
3. 部品座標系での位置姿勢を **基準座標系** に変換します。



基準座標系での位置姿勢が出力されると、Mech-Visionの処理は終了します。

4. 実際値と図面の設計値との差を計算します。

参考情報

大型部品

大型部品とは、長さが0.5mを超える部品を指します。通常、自動車、飛行機、船舶などの製造業向けの部品です。

基準座標系

基準座標系とは、大型部品が取り付けられている対象物座標系を指します。大型部品が自動車部品の場合、自動車本体の座標系は基準座標系となります。

丸穴と長穴

説明図は以下の通りです。

丸穴	長穴
	

大型部品には複数の異なる穴（丸穴または長穴）がある場合がありますので、検出順番と特徴の種類に応じてプロジェクトや [\[vision-tools:parameter-recipe-configuration:::parameter-recipe-configuration\]](#) を追加することができます。

4.3.394.2. 丸穴の中心位置姿勢と直径を計算

機能

このステップの組合せにより、丸穴を検出し、直径とカメラ座標系での丸穴の中心位置姿勢を計算します。

使用シーン

通常、**大型部品** の測定シーンに使用されます。前に [\[vision-steps:extract-planar-point-clouds:::extract-planar-point-clouds\]](#) をつなぎ、平面点群を取得します。後に [位置姿勢をカス](#)

タマイズされた座標系に変換をつなぎ、基準座標系での位置姿勢を取得します。

入力と出力



パラメータの調整説明

円を検出して測定

ROI

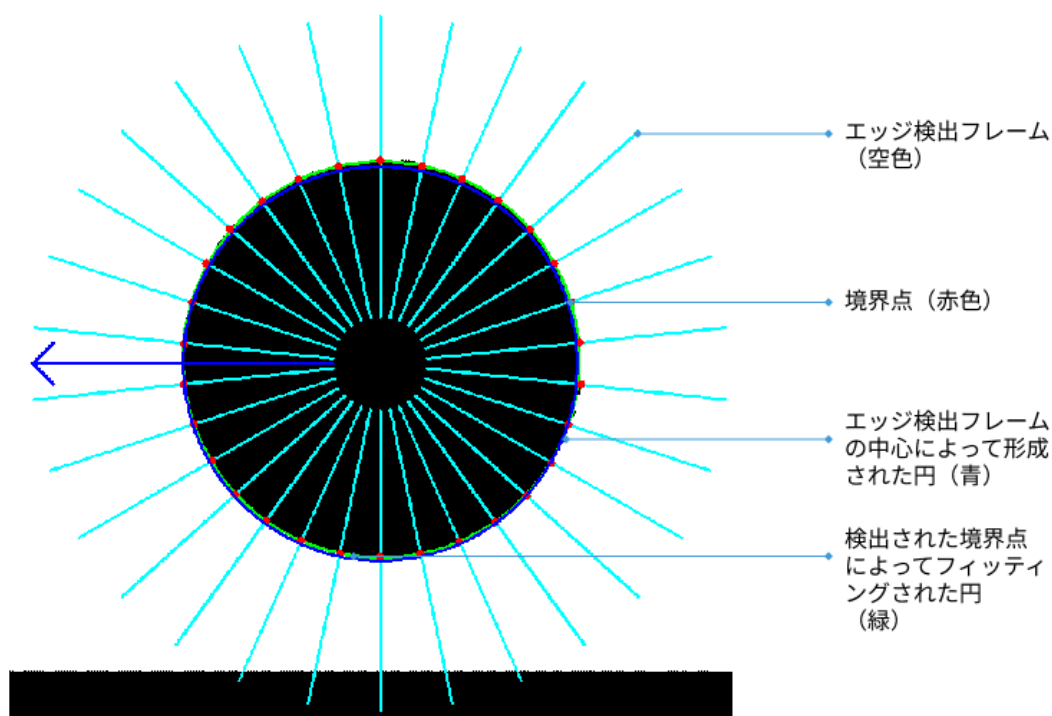
パラメータ説明：このパラメータは、ROIを設定し、丸穴の中心位置姿勢とサイズ情報を読み込むために使用されます。

調整アドバイス：測定モードでROIを選択し、関心領域を正しく設定することをお勧めします。

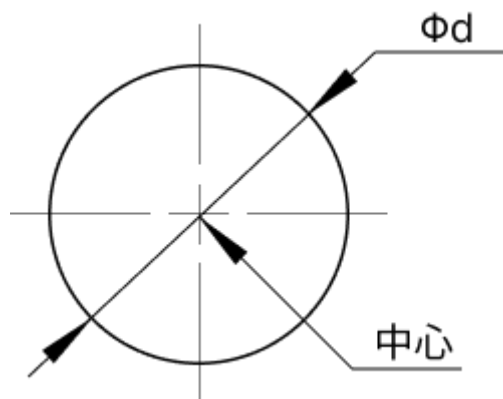


関心領域を設定する時、円に近づけるようにする必要があります。関心領域が不正確な場合、生成された円の精度に影響を与える可能性があります。

原理：下図に示すように、エッジ検出フレームは青い矢印に沿って黒と白の境界点を探し、そこから検出された境界点をフィッティングして円を生成します。そのため、エッジ検出フレームの数が多いほど境界点が多くなり、フィッティングされた円の精度が高くなります。エッジ検出フレームの高さの半分が円の中心に近いほど、検出可変範囲が大きくなり、前後の画像で円の位置が変化しても、正確な円を検出することができます。



- 中心X / Y：読み取った円中心の座標です。
- 幅/高さ：読み取った円の実寸です（下図dに示す）。
- 角度：ROIを設定するための角度です。



エッジ検出フレームの数

パラメータ説明：このパラメータは、円を生成するためのエッジ検出フレームの数を指定するために使用されます。値が高いほど、生成される円の精度が高くなり、計算時間が長くなります。

初期値：500

調整アドバイス：初期のパラメータ値では、円が要件を満たしている場合は適切に下げ、それ以外の場合は適切に上げます。

エッジ検出フレームの高さの半分

パラメータ説明：このパラメータは、円を生成するためのエッジ検出フレームの高さを設定するために使用されます。エッジ検出フレームが円の中心に近いほど、検出の可変範囲が大きくなります。パラメータが大きいほど計算時間は長くなり、小さいほど計算時間は短くなります。この値が大きすぎると、同じエッジ検出フレームが円全体を通過することになり、その際に2つの交点を検出したり、円以外の残りの輪郭点を検出したりして、円のフィッティングが不正確になる可能性があります。

初期値：110

調整アドバイス：ROIが円全体を選択する場合（半円しかない場合は円全体を選択することができない）、この値を円の半径（ROIの幅/高さの半分から計算）に設定することを推奨します。円全体をフレームに収めることができない場合や、画像に他の境界線がある場合は、実測値に応じて設定することができます。

円柱以内の点群を抽出



このステップの組合せを実行すると、点群はカメラ座標系から点の基準座標系に変換されます。このパラメータグループ下のパラメータは、点の基準座標系に従って点のZ軸の範囲を設定します。

最小Z値：

パラメータ説明：このパラメータは、点の基準座標系におけるZ軸の下限（ミリメートル単位）を指定するために使用されます。値が小さいほど抽出される点は多くなり、値が大きいほど抽出される点は少なくなります。

初期値：-0.500mm

最大Z値：

パラメータ説明：このパラメータは、点の基準座標系におけるZ軸の上限（ミリメートル単位）を指定するために使用されます。値が小さいほど抽出される点は少なくなり、値が大きいほど抽出される点は多くなります。

初期値：0.500mm

4.3.394.3. 長穴の中心位置姿勢と長軸・短軸を計算

機能

このステップの組合せにより、長穴の長軸と短軸、カメラ座標系での長穴中心の位置姿勢を計算します。

使用シーン

通常、大型部品の測定シーンに使用されます。前に [\[vision-steps:extract-planar-point-clouds:::extract-planar-point-clouds\]](#) をつなぎ、平面点群を取得します。後に [位置姿勢をカスタマイズされた座標系に変換](#) をつなぎ、[基準座標系](#) での位置姿勢を取得します。

入力と出力



パラメータの調整説明

長穴を検出して測定

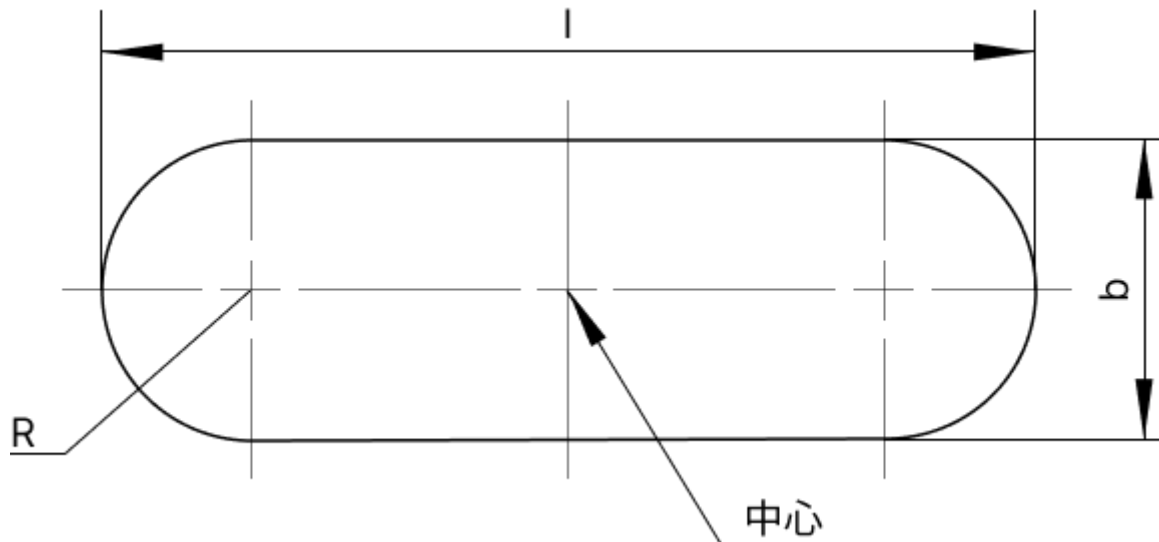
ROI

パラメータ説明：このパラメータは、ROIを設定し、長穴の中心位置姿勢とサイズ情報を読み込むために使用されます。調整アドバイス： **測定モード** でROIを選択し、関心領域を正しく設定することをお勧めします。



関心領域を設定する時、長穴全体を選択し、長穴に近づけるようにする必要があります。関心領域が不正確な場合、長穴データの精度に影響を与える可能性があります。

- 中心X/Y：長穴中心の座標です。
- 幅：読み取った長穴の幅です（下図lに示す）。
- 高さ：読み取った長穴の高さです（下図bに示す）。
- 角度：ROIを設定するための角度です。長穴の方向は水平または垂直の場合は設定する必要はありません。



円柱以内の点群を抽出



このステップの組合せを実行すると、点はカメラ座標系から点の基準座標系に変換されます。このパラメータグループ下のパラメータは、点の基準座標系に従って点のZ軸の範囲を設定します。

最小Z値

パラメータ説明：このパラメータは、点の基準座標系におけるZ軸の下限（ミリメートル単位）を指定するために使用されます。値が小さいほど抽出される点は多くなり、値が大きいほど抽出される点は少なくなります。初期値：-0.500mm

最大Z値

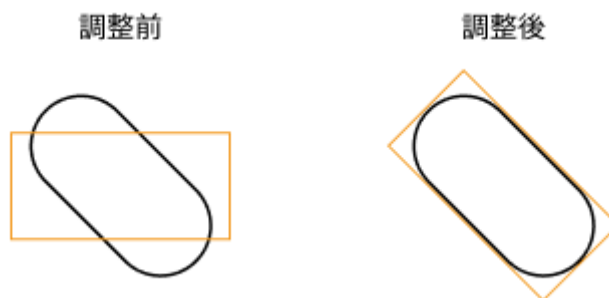
パラメータ説明：このパラメータは、点の基準座標系におけるZ軸の上限（ミリメートル単位）を指定するために使用されます。値が小さいほど抽出される点は少なくなり、値が大きいほど抽出される点は多くなります。初期値：0.500mm

調整の例

角度

このパラメータは、ROI選択ボックスの角度を設定するために使用されます。大型部品測定の場合、長穴は様々な向きになることがあるため、ROI選択ボックスは横向きか縦向きでしか設定できません。長穴が水平・垂直に配置されていない場合、関心領域の選択ボックスが実際の長穴に合うように **角度** を設定する必要があります。

角度 の調整前後の対比を下図に示します（オレンジ色の長方形はROI選択ボックスとなります）。



4.3.394.4. 平面点群を抽出

機能

このステップの組合せにより、平面点群内の平面からのずれが大きい点を除去し、より平らかな平面点群を取得できます。

使用シーン

通常、大型部品の測定シーンに使用されます。前に [\[vision-steps:select-point-cloud-from-depth-map:::select-point-cloud-from-depth-map\]](#) をつなぎ、点群を取得します。後に [丸穴の中心位置姿勢と直径を計算](#) をつなぐ場合は丸穴を検出し、[長穴の中心位置姿勢と長軸・短軸を計算](#) をつなぐ場合は長穴を検出します。

入力と出力



パラメータの調整説明

点のフィルタリング（点のフィルタリング） / 点のフィルタリング（最初のフィルタリング2） / 点のフィルタリング（再フィルタリング2）

最近隣接点の数

パラメータ説明：このパラメータは、平面点群に隣接する点を優先的に除去するために使用されます。値が大きいほど除去される点が多くなり、値が小さいほど除去される点が少なくなります。初期値：30

標準偏差乗数

パラメータ説明：このパラメータは、距離しきい値を計算するために使用されます。値が小さいほど除去される点が多くなり、値が大きいほど除去される点が少なくなります。初期値：2.0000

点のフィルタリング（最初のフィルタリング） / 点のフィルタリング（再フィルタリング）

最大角度差

パラメータ説明：このパラメータは、点の基準座標系のZ軸と点の法線ベクトルのZ軸との最大角度差（単位は度）を指定するために使用されます。点の基準座標系のZ軸とその点の法

線ベクトルのZ軸との角度差がこの値より大きい場合、その点は除去されます。値が大きいほど保持される点が多くなり、小さいほど保持される点が少なくなります。初期値：3°

4.3.394.5. 深度画像から点群を分割

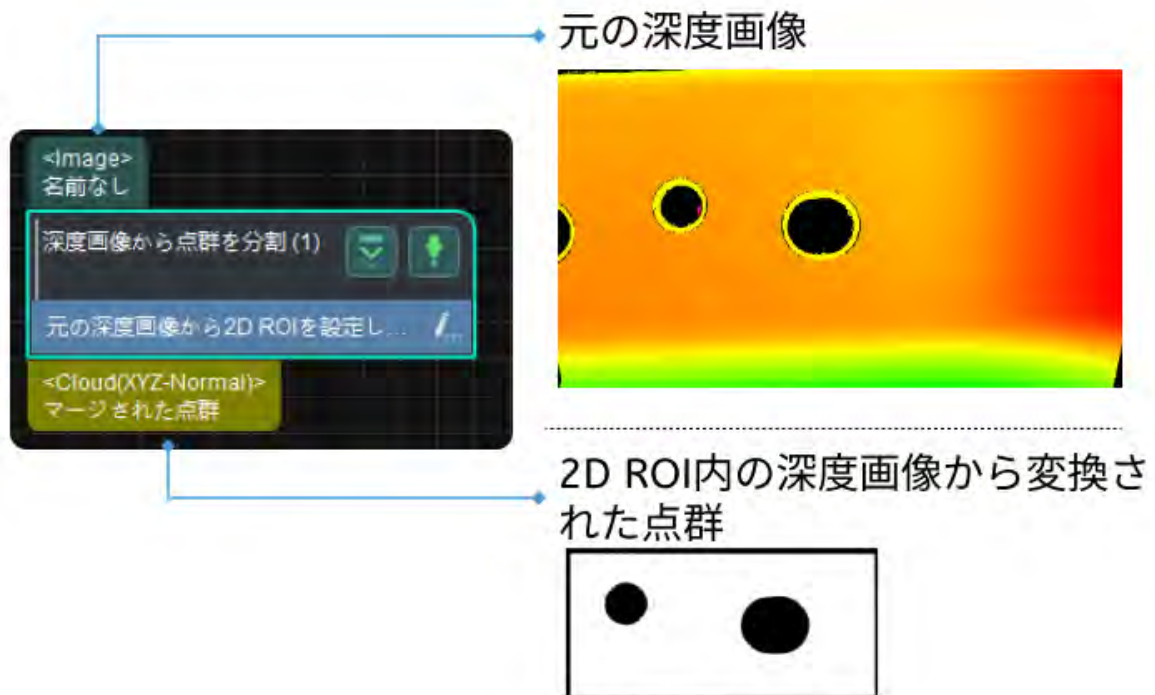
機能

このステップの組合せにより、元の深度画像から2D ROIを設定し、その後の処理を容易にするために2D ROI内の深度画像を点群に変換します。

使用シーン

通常、大型部品の測定シーンに使用されます。前に [\[vision-steps:capture-images-from-camera:::capture-images-from-camera\]](#) をつなぎ、深度画像を取得します。後に [平面点群を抽出](#) をつなぎ、点群を平面点群に変換します。

入力と出力



パラメータの説明

深度画像分割

以下のパラメータはすべて、深度画像を分割し、期待される点群を取得するために使用されます。

隣接するピクセル領域間の最大値の差

パラメータ説明：深度画像を複数の領域に分割する処理において、隣接するピクセル間の最

大高さの差を指定するため使用されます。隣接するピクセルの高さの差がこの値より小さい場合、そのピクセルは同じ領域にあります。値が大きいほど、分割する領域は多くなります。

初期値：0.5mm

調整アドバイス：平面を分割する場合、このパラメータを適切に上げます。ただし、値を大きくしすぎると、分割領域が小さくなりすぎて結果が出ない場合がありますので注意が必要です。曲面やネジ穴を分割する場合は、必要な点群が失われないよう、この値を小さく調整することを推奨します。点群の付着が激しい場合、この値を調整しても不要な点群が除去されないことがあります。

対象物の最小面積

パラメータ説明：このパラメータは、深度画像の分割後に許可される最小領域を指定するために使用されます。この値より少ないピクセル数の領域は除去されます。単位はピクセル (px) です。値が小さいほど残る領域が多くなり、値が大きいほど除去される領域が多くなります。

初期値：1500px

深度画像のROIファイル

パラメータ説明：このパラメータは、深度画像の関心領域を設定するために使用されます。

調整アドバイス：[**2D ROIを設定**] をクリックし、**2D関心領域** を正しく設定します。

深度画像で最高領域を取得

層の高さ

パラメータ説明：深度画像が点群に変換された後、このパラメータは、平面点群をカメラ座標系のZ軸に沿って下に移動させる距離（ミリメートル単位）を指定するために使用されます。値が大きいほど、保持される点群が多くなり、値が小さいほど保持される点群が少なくなります。

初期値：10.000mm

調整アドバイス：測定対象物の点群が上下に複数層ある場合、この値を調整して期待される点群を取得できます。

4.3.394.6. 位置姿勢をカスタマイズされた座標系に変換

機能

計算を容易にするために、測定対象物の中心位置姿勢を基準座標系に変換します。変換流れは以下の通りです。

1. カメラ座標系での位置姿勢を、最初にカスタマイズされた座標系での位置姿勢に変換します。

2. 最初にカスタマイズされた座標系での位置姿勢を、基準座標系での位置姿勢に変換します。

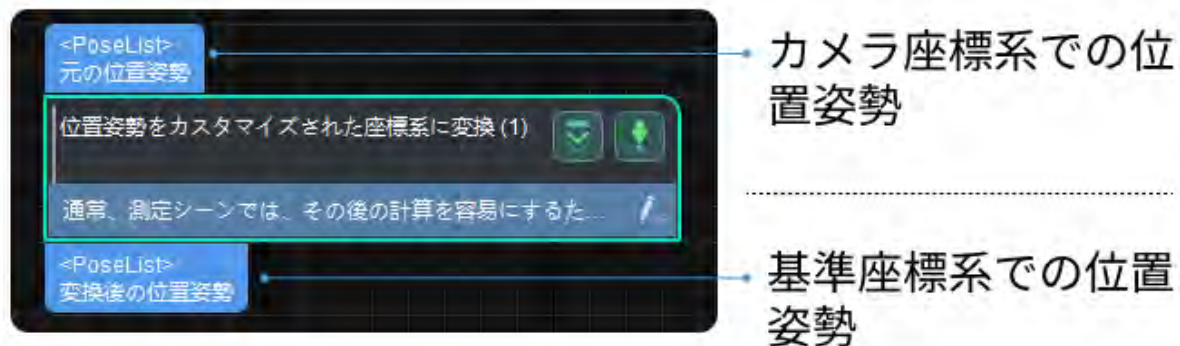


このステップが大型部品の測定シーンに使われる場合、最初にカスタマイズされた座標系は通常、部品座標系となります。

使用シーン

通常、大型部品の測定シーンに使用されます。前に [\[vision-steps:calc-hole-center-poses-and-diameters:::calc-hole-center-poses-and-diameters\]](#) をつなぐ場合、丸穴を検出してカメラ座標系での丸穴の中心位置姿勢を出力します。前に [長穴の中心位置姿勢と長軸・短軸を計算](#) をつなぐ場合、長穴を検出してカメラ座標系での長穴の中心位置姿勢を出力します。


入力と出力



パラメータの調整説明

ファイルから位置姿勢を読み込む（位置姿勢の原点がロボット座標系にある）

位置姿勢ファイル


 をクリックし、表示される画面で最初にカスタマイズされた座標系の **JSON** ファイルを選択します。



このステップが大型部品の測定に使われる場合、この座標系は通常、部品座標系となります。位置姿勢の変換ファイルを取得するには、Mech-Mind株式会社にお問い合わせください。

ファイルから位置姿勢を読み込む（位置姿勢の原点が自動車本体座標系にある）

位置姿勢のファイル

 をクリックし、表示される画面で2つ目のカスタマイズされた座標系の **JSON** ファイルを選択します。



このステップが大型部品の測定に使われる場合、この座標系は通常、基準座標系となります。位置姿勢の変換ファイルを取得するには、Mech-Mind株式会社にお問い合わせください。

4.3.395. ソート

ソートに関するステップの組合せについて説明します。

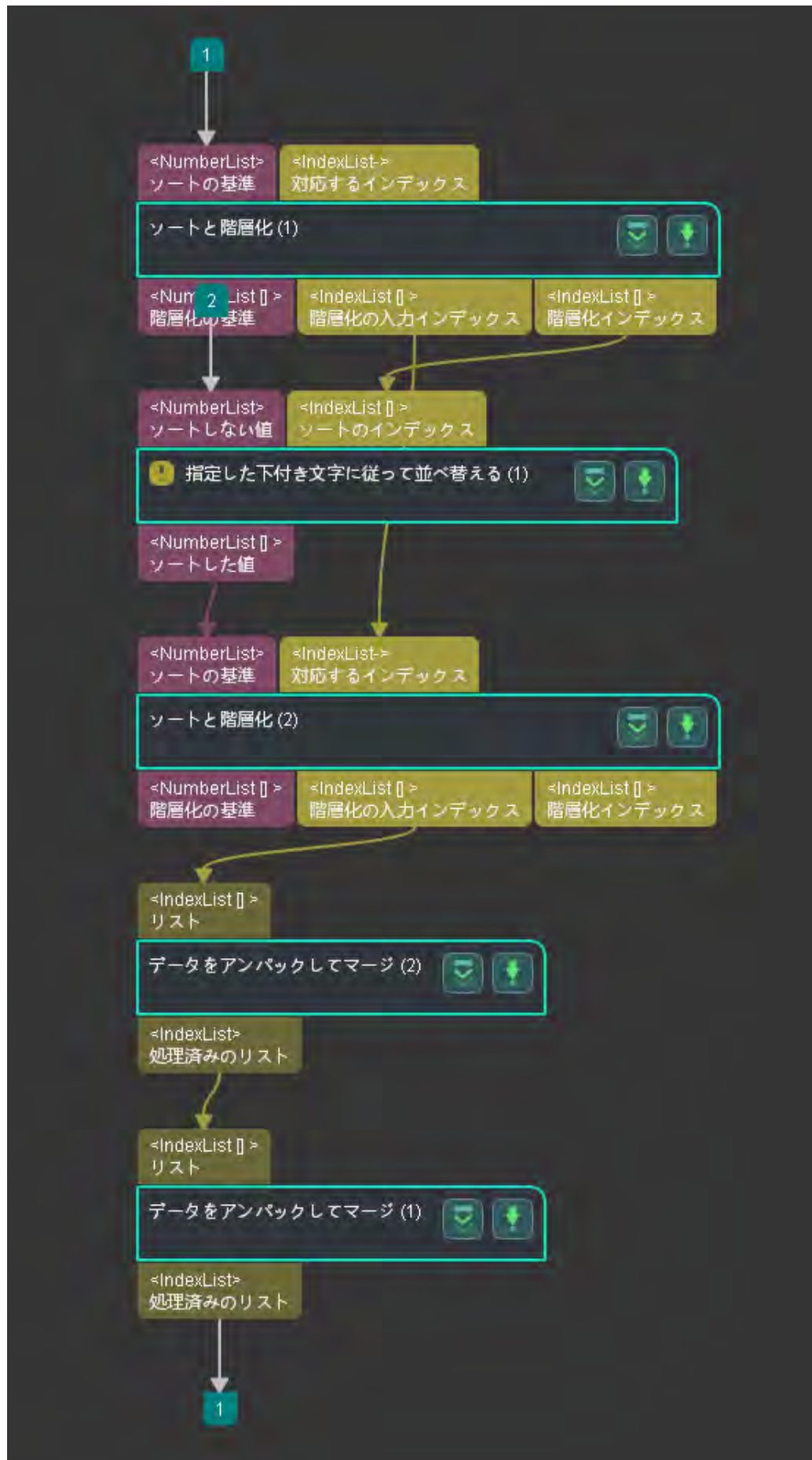
[\[vision-steps:2d-sort:::2d-sort\]](#)

[\[vision-steps:3d-sort:::3d-sort\]](#)

[\[vision-steps:sort-by-pickpoint:::sortby-pickpoint\]](#)

4.3.395.1. 2Dソート

同じ長さの2つの要素を入力し、異なるソート基準でソートされ、2回ソート後のインデックスが出力されます。その構成を下図に示します。

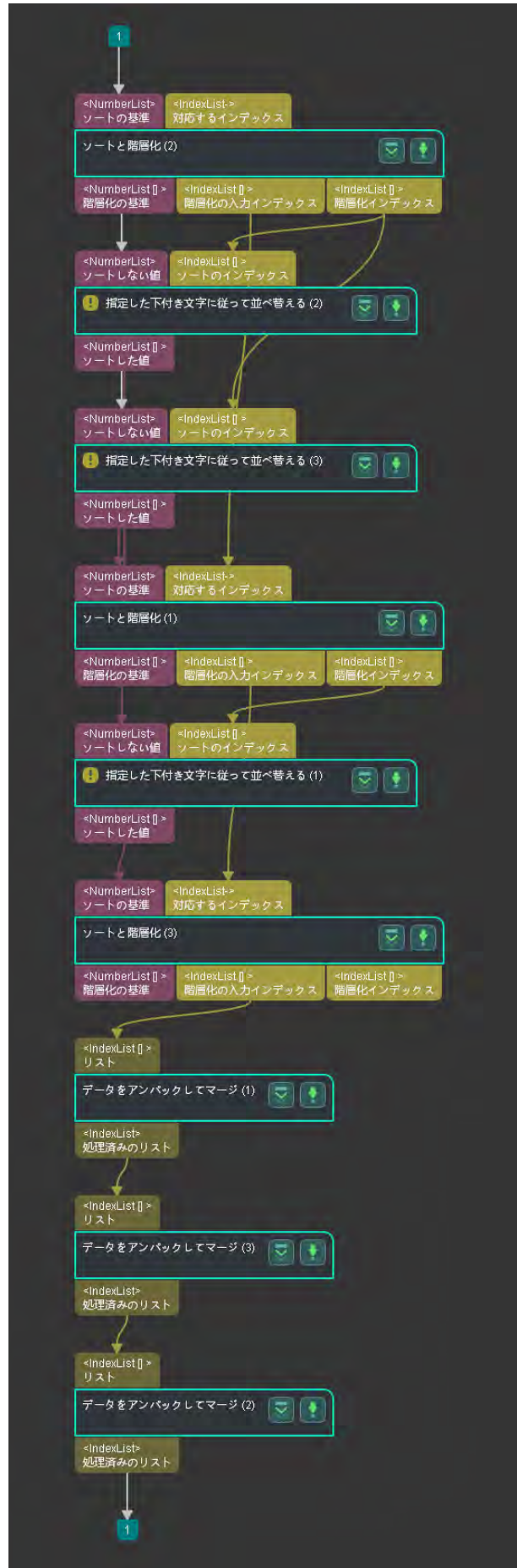


各ステップの機能は以下の通りです。

1. [\[vision-steps:sort-and-stratify:::sort-and-stratify\]](#) により、ソートする最初の要素を入力し、指定した層間隔（ソート基準1）でソート・分類し、最初のソート後のインデックスを出力します。
2. [\[vision-steps:reorder-according-to-given-indices:::reorder-according-to-given-indices\]](#) により、2番目のソートする要素と最初のソート後の要素のインデックスを入力し、インデックスを使用して2番目の要素をソートしてソート後の値を出力します。
3. [\[vision-steps:sort-and-stratify:::sort-and-stratify\]](#) により、ソート後の2番目の要素と最初のソートから得られたインデックスを入力し、指定した層間隔（ソート基準2）でソート後の2番目の要素を再ソート・分類します。2回のソート後のインデックスを出力します。
4. [\[vision-steps:unpack-and-merge-data:::unpack-and-merge-data\]](#) により、2回のソートによって得られたインデックスを入力し、3次元のクラス配列変数を2次元に変換します。
5. [\[vision-steps:unpack-and-merge-data:::unpack-and-merge-data\]](#) により、最初のマージ後の結果を入力し、2次元のクラス配列変数を1次元に変換してソート後のインデックスを出力します。

4.3.395.2. 3Dソート

同じ長さの2つの要素を入力し、異なるソート基準でソートを行い、3回ソート後のインデックスを出力します。その構成を下図に示します。



各ステップの機能は以下の通りです。

1. [\[vision-steps:sort-and-stratify:::sort-and-stratify\]](#) により、ソートする最初の要素を入力し、指定した層間隔（ソート基準1）でソート・分類し、最初のソート後のインデックスを出力します。
2. [\[vision-steps:reorder-according-to-given-indices:::reorder-according-to-given-indices\]](#) により、ソートする2番目の要素と最初のソート後のインデックスを入力し、インデックスで要素をソートし、1回ソート後の2番目の要素の結果を出力します。
3. [\[vision-steps:reorder-according-to-given-indices:::reorder-according-to-given-indices\]](#) により、ソートする3番目の要素と最初のソート後のインデックスを入力し、インデックスで要素をソートし、1回ソート後の3番目の要素の結果を出力します。
4. [\[vision-steps:sort-and-stratify:::sort-and-stratify\]](#) により、1回ソート後の2番目の要素（入力1）と最初のソート後のインデックス（入力2）を入力し、指定した層間隔（ソート基準2）で入力1を再ソート・分類します。2回ソート後のインデックス（出力2）と2回目ソートのインデックス（出力3）を出力します。
5. [\[vision-steps:reorder-according-to-given-indices:::reorder-according-to-given-indices\]](#) により、一回のソート後の3番目の要素と2回目のソートのインデックスを入力し、インデックスで要素をソートし、2回ソート後の3番目の要素の結果を出力します。
6. [\[vision-steps:sort-and-stratify:::sort-and-stratify\]](#) により、2回ソート後の3番目の要素（入力1）とインデックス（入力1）を入力し、指定した層間隔（ソート基準3）で入力1をソート・分類し、3回ソート後のインデックスを出力します。
7. [\[vision-steps:unpack-and-merge-data:::unpack-and-merge-data\]](#) により、3回のソートによって得られたインデックスを入力し、4次元のクラス配列変数を3次元に変換します。
8. [\[vision-steps:unpack-and-merge-data:::unpack-and-merge-data\]](#) により、最初のアンパックとマージの結果を入力し、3次元のクラス配列変数を2次元に変換します。
9. [\[vision-steps:unpack-and-merge-data:::unpack-and-merge-data\]](#) により、2回のアンパックとマージの結果を入力し、2次元のクラス配列変数を1次元に変換してソート後のインデックスを出力します。

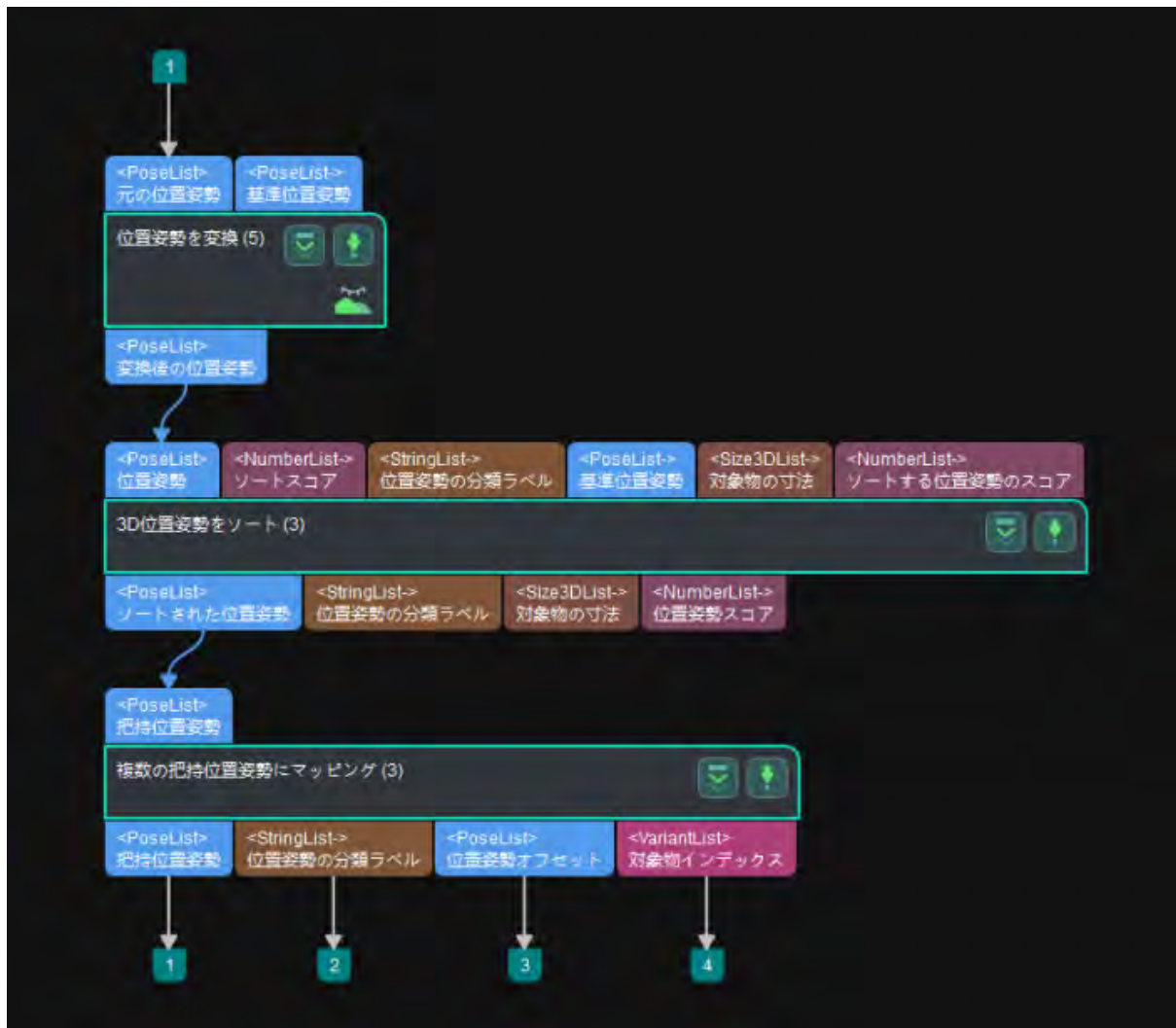
4.3.395.3. 把持位置姿勢をソート

機能

位置姿勢をカメラ座標系からロボット座標系に変換し、指定した基準で位置姿勢をソートし、それを複数の把持位置姿勢にマッピングします。

- 入力：ソートする位置姿勢リスト。
- 出力：把持位置姿勢リスト、位置姿勢の分類ラベル、位置姿勢オフセット、対象物のインデックス。

ステップ組合せの構成を下図に示します。



各ステップの機能は以下の通りです。

[[vision-steps:transform-poses:::transform-poses](#)] により、把持位置姿勢を入力し、ロボットが把持位置姿勢を読み込むために、その座標系をロボット座標系に変換します。

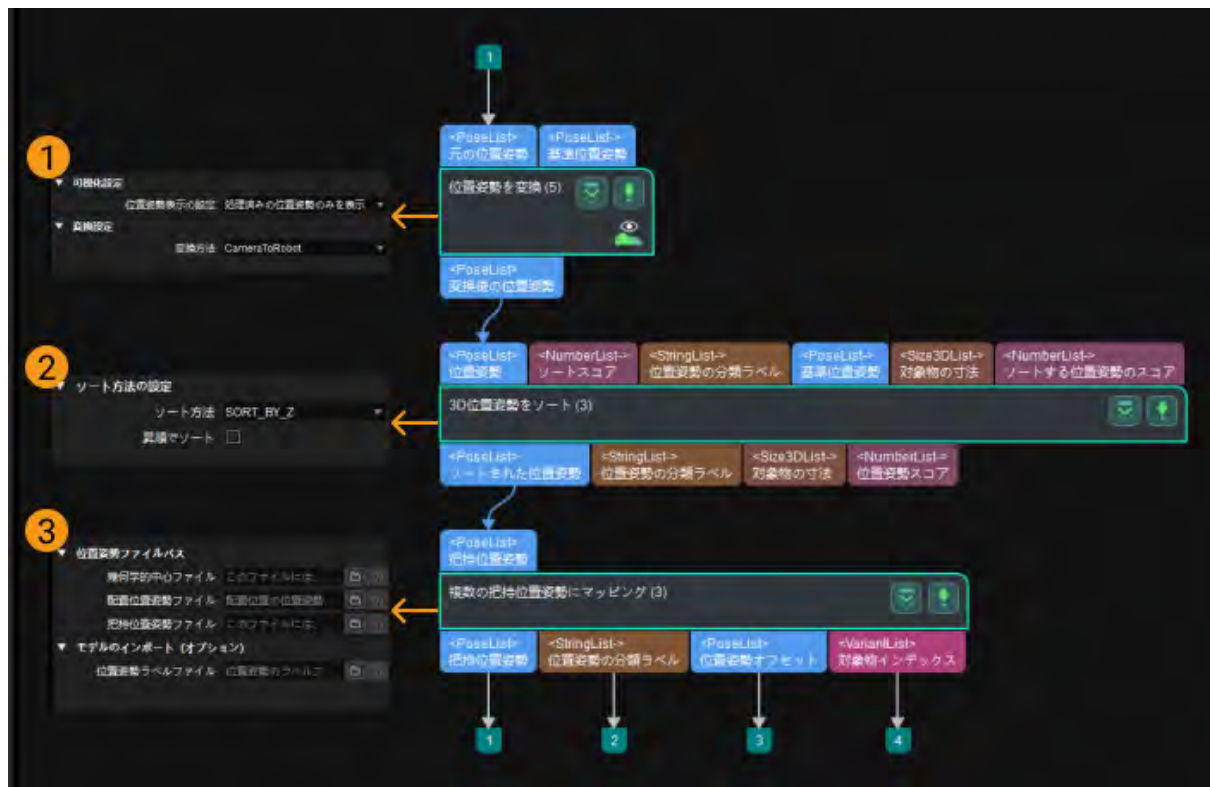
[[vision-steps:sort-3d-poses:::sort-3d-poses](#)] により、座標系変換後の把持位置姿勢を入力し、指定した基準で把持位置姿勢をソートします。例えば、位置姿勢のXで降順にソート、ソートのスコアで昇順にソートなど。

[[vision-steps:map-to-multi-pick-points:::map-to-multi-pick-points](#)] により、ソート後の把持位置姿勢を入力し、同じワークの異なる把持位置姿勢を計算して、ワークの把持率を向上させます。

調整の例

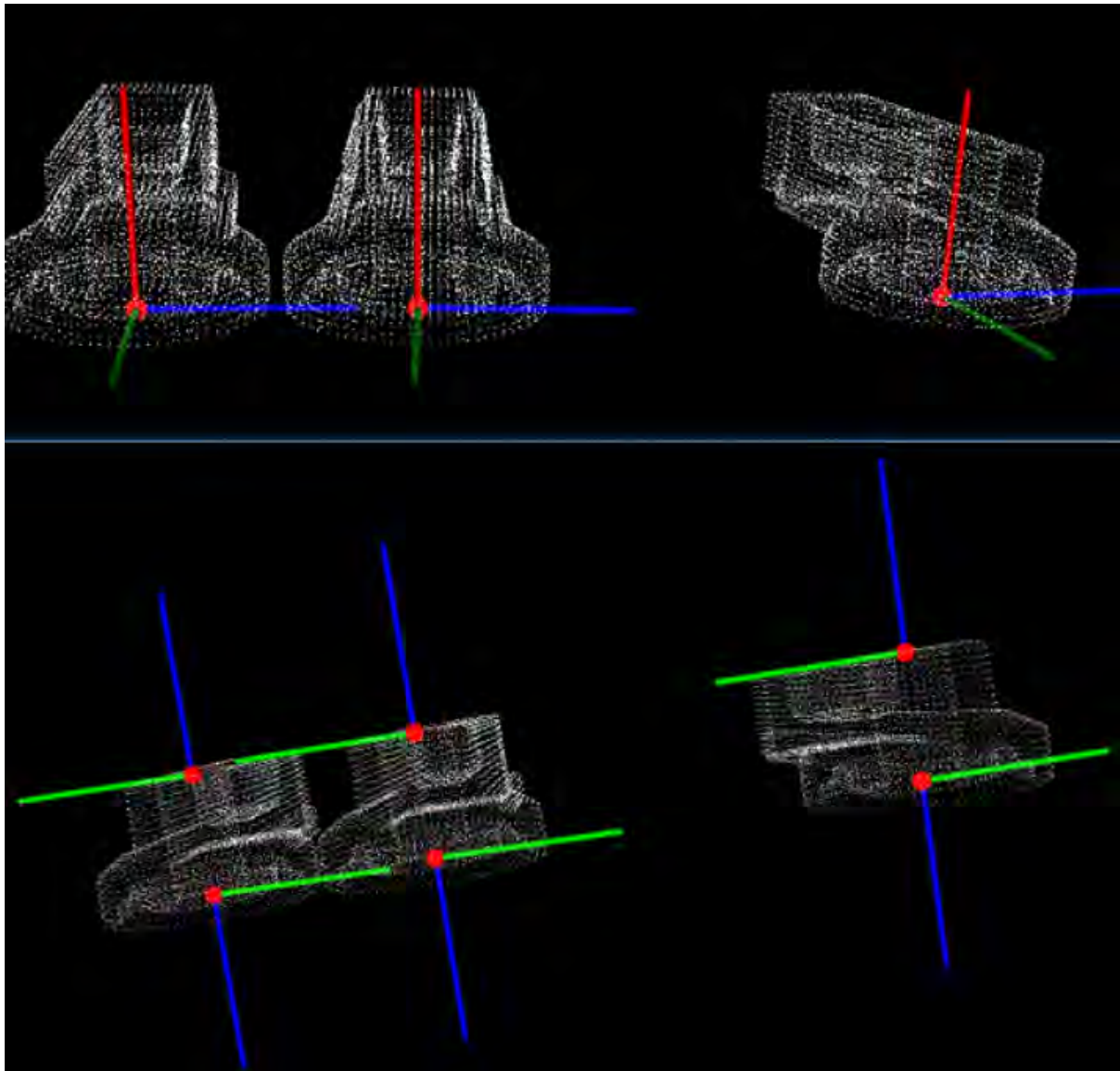
位置姿勢をカメラ座標系からロボット座標系に変換し、Z値で降順にソートし、複数の把持位置姿勢にマッピングして出力します。

ステップパラメータの設定は下図の通りです。



1. 位置姿勢をロボット座標系に変換します。
2. 位置姿勢のZの値によって降順にソートします。
3. 複数の把持位置姿勢にマッピングします。

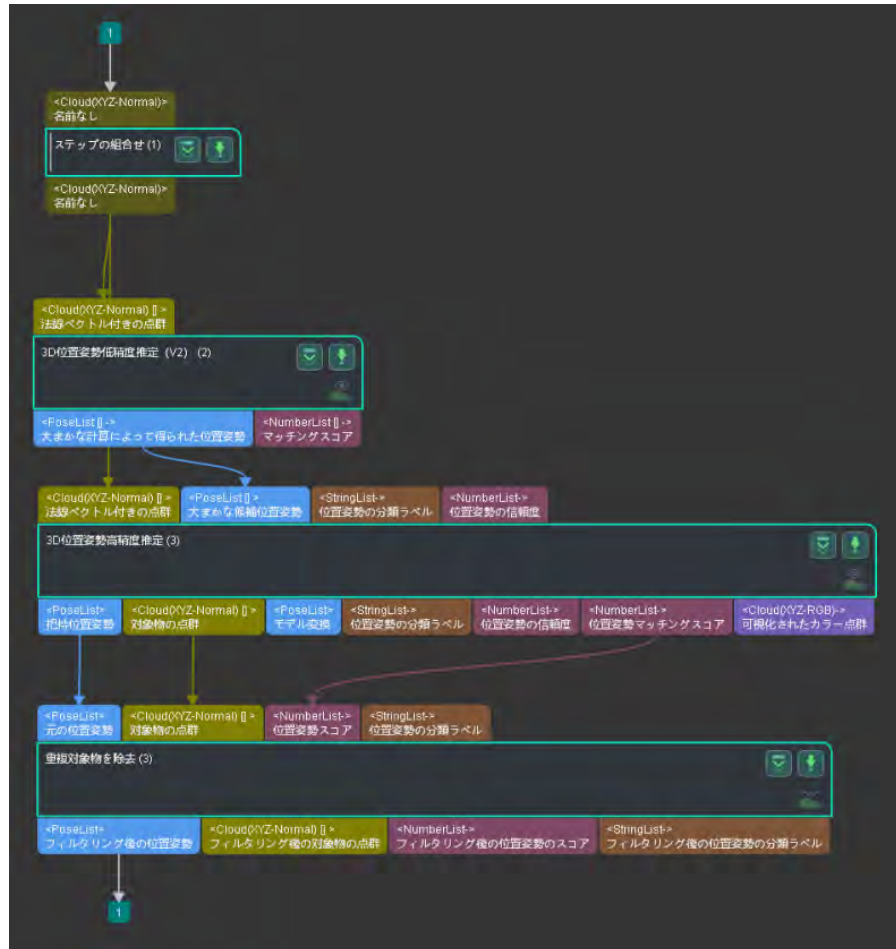
下図の上側と下側はそれぞれ複数の把持位置姿勢にマッピングする前と後の比較効果を示しています。



4.3.396. 3D位置姿勢推定

機能

主な機能は、対象物の点群モデルをマッチングすることにより、シーン内の物体を認識して位置姿勢を取得することです。その構成を下図に示します。

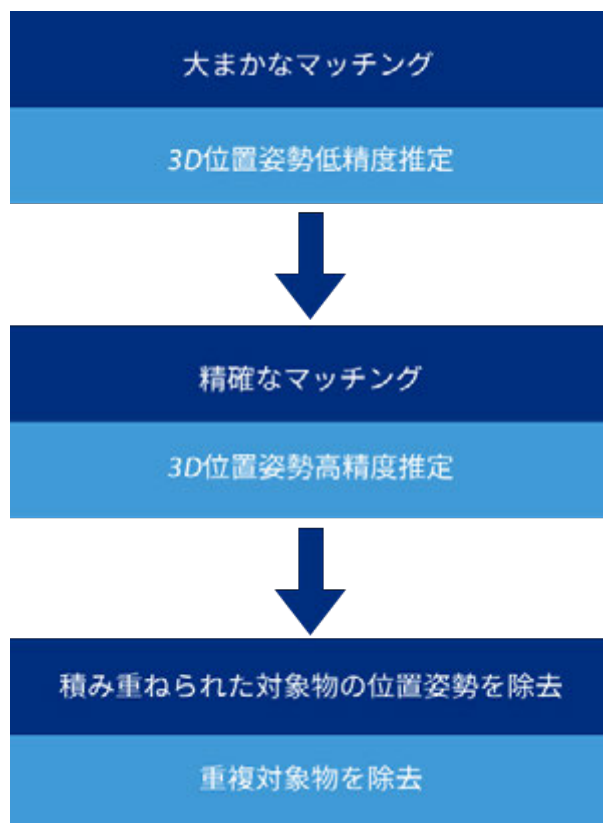


使用シーン

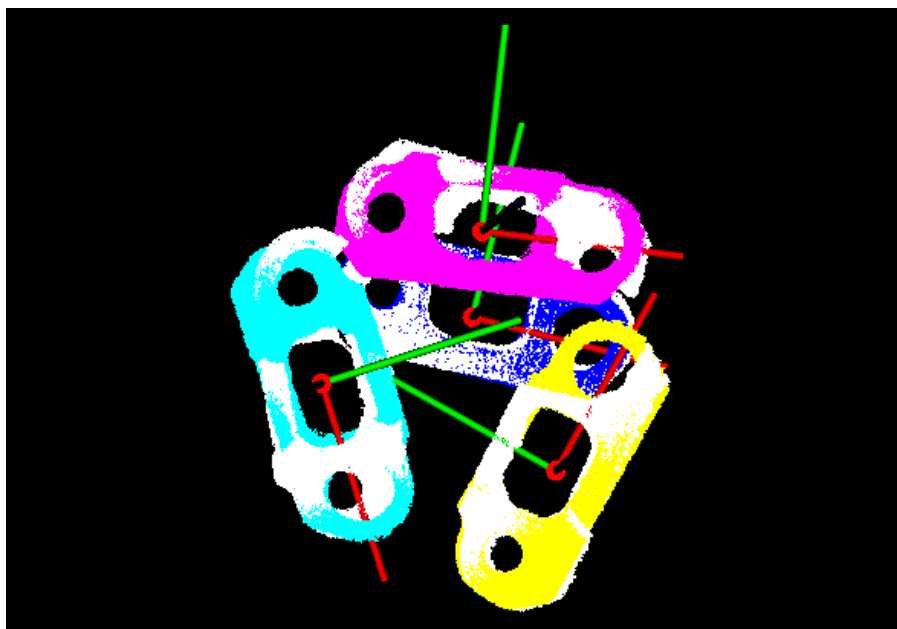
通常、このステップの組合せは、**点の数が制限を超えた点群を除外**の後で使用されます。**3D位置姿勢低精度推定 (V2)** と **3D位置姿勢高精度推定** により、把持位置姿勢を最適化し、積み重ねられた対象物の位置姿勢を除去することができます。

マッチング処理

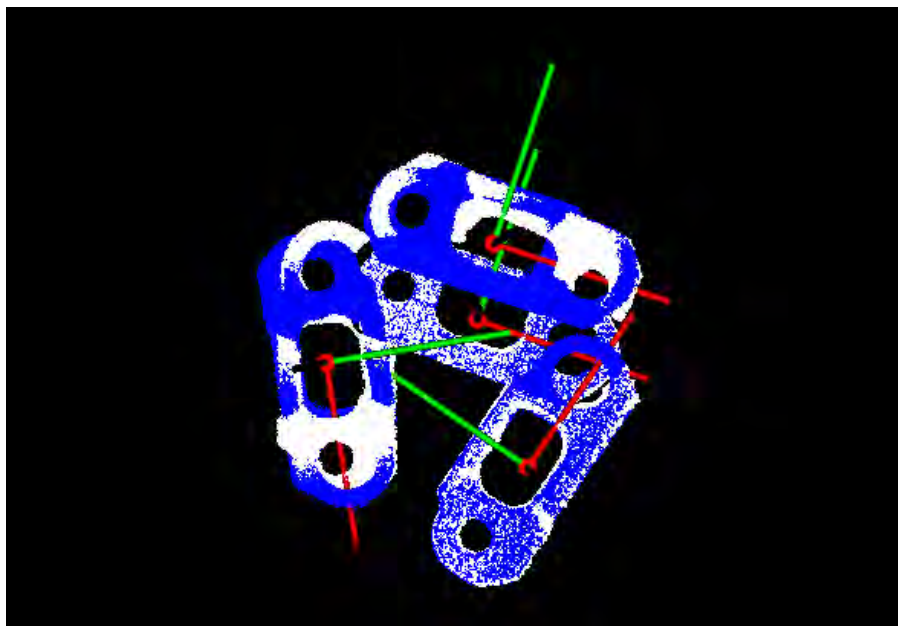
マッチング処理は下図のような流れで行われます。



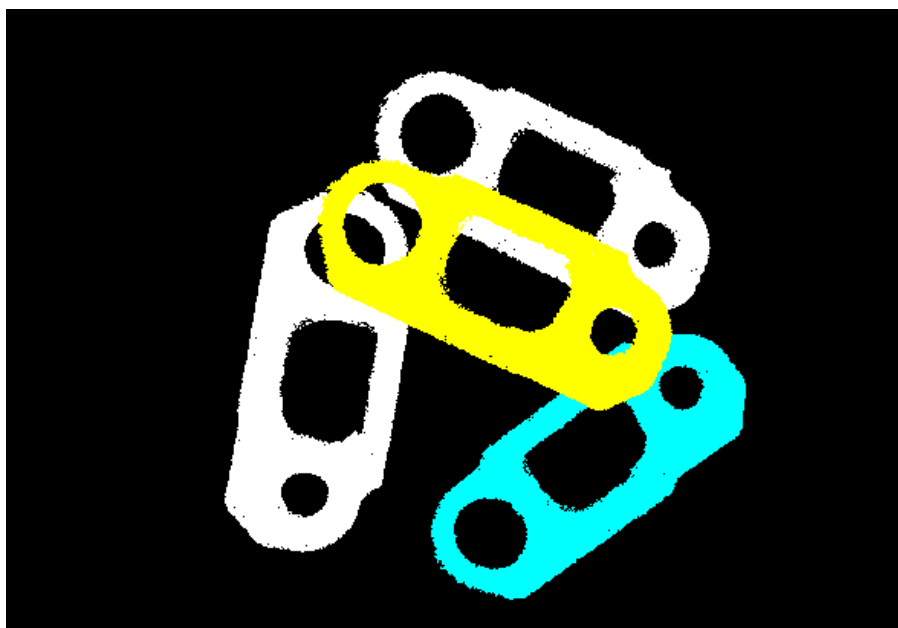
1. 下図に示すように、「点の数が制限を超えた点群を除外」を使用してフィルタリングされた点群を取得したら、フィルタリングされた点群を「3D位置姿勢推定」のステップの組合せに入力し、「3D位置姿勢低精度推定（V2）」で認識された対象物の大まかな位置姿勢を計算します。また、大まかな位置姿勢が複数ある場合があります。



2. 次に、下図に示すように、「3D位置姿勢高精度推定」を使用して大まかな位置姿勢に対して精確なマッチングを行い、精確な位置姿勢を取得します。



3. 最後に、水平方向に認識された対象物の位置姿勢を選出した後、**重複対象物を除去**をつなぎます。これにより、対象物のZ軸に沿って積み重ねられた対象物を除去し、上層で把持可能な位置姿勢を取得することができます。下図に白で表示されるのは、除去された重複対象物の点群です。



パラメータの調整説明

モデル設定

モデル名

初期値：なし。

調整説明：このステップの組合せを実行する前に、対象物の点群モデルを追加する必要があります。

ります。対象物の点群モデルを使用してシーン内の物体に対してマッチングを行い、シーン内の物体を認識して位置姿勢を取得します。[マッチングモデル・把持位置姿勢エディタ](#)で点群モデルを作成して保存した後、ドロップダウンバーから保存した点群モデルを選択できます。

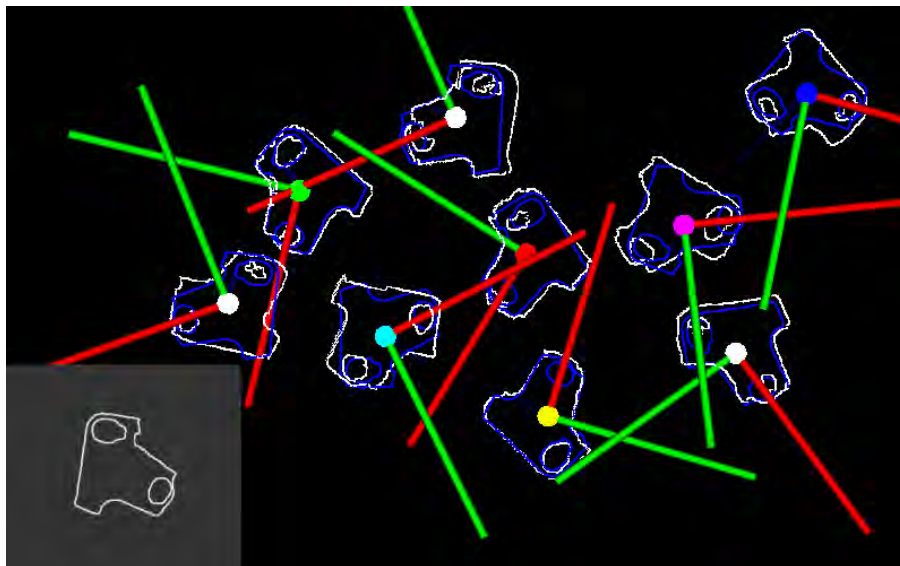
マッチング設定

マッチングモード

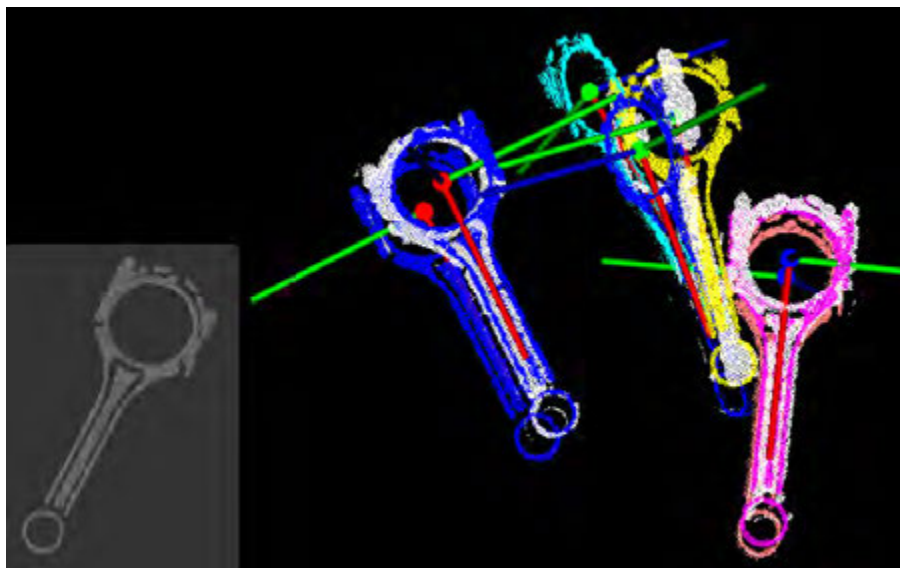
初期値：Edge

オプション：Surface（サーフェスマッチング）、Edge（エッジマッチング）。

調整説明：Surface（サーフェスマッチング）、Edge（エッジマッチング）の2つのマッチングモードを選択可能です。下図に示すように、使用するモデル点群が対象物のエッジである場合は、エッジマッチングを選択します。



下図に示すように、使用するモデル点群は対象物の表面である場合、サーフェスマッチングを選択してください。



サンプリングされたモデルの期待点数

初期値：300

調整説明：ダウンサンプリング後の点群モデルの数を設定するために使用されます。自動ダウンサンプリング後で得られた点数は、この値に近くなります。値が小さいほど、サンプリングされた点群内の点が少なくなり、マッチングの精度が低くなります。

パラメータ値	300	600
サンプル図		
点数	276	632

マッチング速度

初期値：標準

オプション：高速、標準、高精度

調整説明：プロジェクトのニーズに応じて選択してください。

- 高速：速度は速いですが精度は低いです。
- 標準：安定的に実行するモードです。
- 高精度：精度が高いですが実行速度は遅いです。

信頼度

初期値：0.6

調整説明：このパラメータは、位置姿勢のマッチングスコアと比較するために使用されます。さらに正確なマッチング結果を取得するために、**結果評価の検索半径** パラメータと組み合わせ使用されることがよくあります。位置姿勢のマッチングスコアがこの値より大きい場合、位置姿勢のマッチング結果は有効であると見なされます。他のパラメータが変更され

ていない場合、この値が大きいほど、位置姿勢のマッチング結果がより正確になります。

実際の使用では、最初に **結果評価の検索半径** を適切に設定する必要があります。設定するには、ダブルクリックしてその中に入り、「3D位置姿勢高精度推定」でこのパラメータを設定します。その後、実際の状況に応じて **信頼度** を設定します。

位置姿勢のマッチングスコアは、**結果評価の検索半径** の値によって決められます。この値が大きいほど位置姿勢のマッチング結果の評価は甘くなり、位置姿勢のマッチング精度が出なくても位置姿勢のマッチングスコアが高くなります。この値が小さいほど位置姿勢のマッチング結果の評価が厳しくなります。対象物の点群に応じて設定する必要があります。点群がまばらな場合は、**結果評価の検索半径** の値を適切に上げることができます。

パラメータを調整する際に、まずは初期の **結果評価の検索半径** を設定し、ステップを実行して可視化出力ウィンドウで位置姿勢のマッチング効果を確認し、位置姿勢のマッチングスコアと対比します。マッチング効果とマッチングスコアが一致していない場合、実際の状況に応じて **結果評価の検索半径** の値をマッチング効果とマッチングスコアが一致するまで調整することができます。

結果評価の検索半径 を設定した後、実際のニーズに応じて **信頼度** を設定し、信頼度より高い位置姿勢のマッチングスコアを持つ位置姿勢マッチング結果を取得します。

結果設定

出力の最大数

初期値：10

設定可能な範囲：1~100

調整説明：このパラメータは、出力可能な位置姿勢のマッチング結果の最大数を示します。複数の位置姿勢のマッチング結果がある場合、マッチングスコアに従ってマッチング結果を降順にソートし、設定された **出力の最大数** に従って最も高いスコアを持つマッチング結果を出力します。



位置姿勢のマッチング結果の実際の出力数は、設定された **出力の最大数** と違い場合もあります。例えば、**出力の最大数** が5に設定されていて、実際には合計3つの位置姿勢のマッチング結果しかない場合、最終的に出力される位置姿勢のマッチング結果の数は3です。

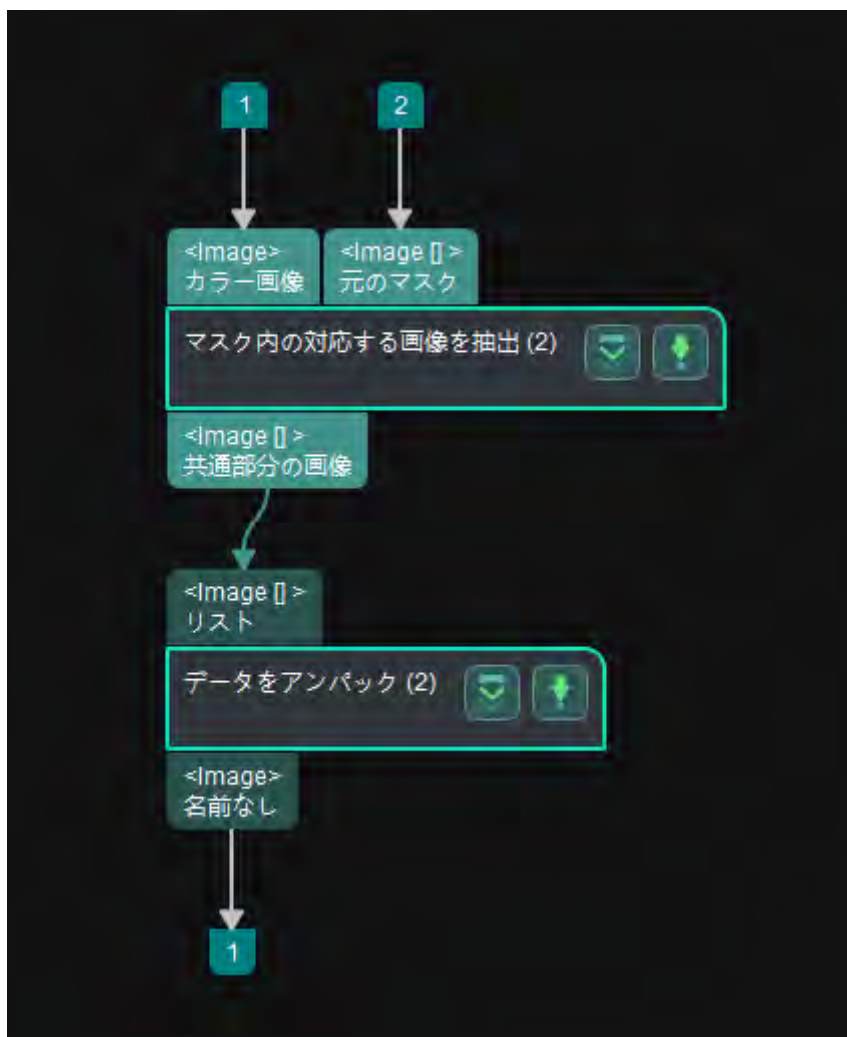
4.3.397. マスクが対応するカラー画像を取得

機能

マスク画像リストの最初のインデックスが対応するカラー画像を取得します。

- 入力：カラー画像、マスク画像リスト。
- 出力：マスク画像リストの最初のインデックスが対応するカラー画像。

ステップの構成を下図に示します。



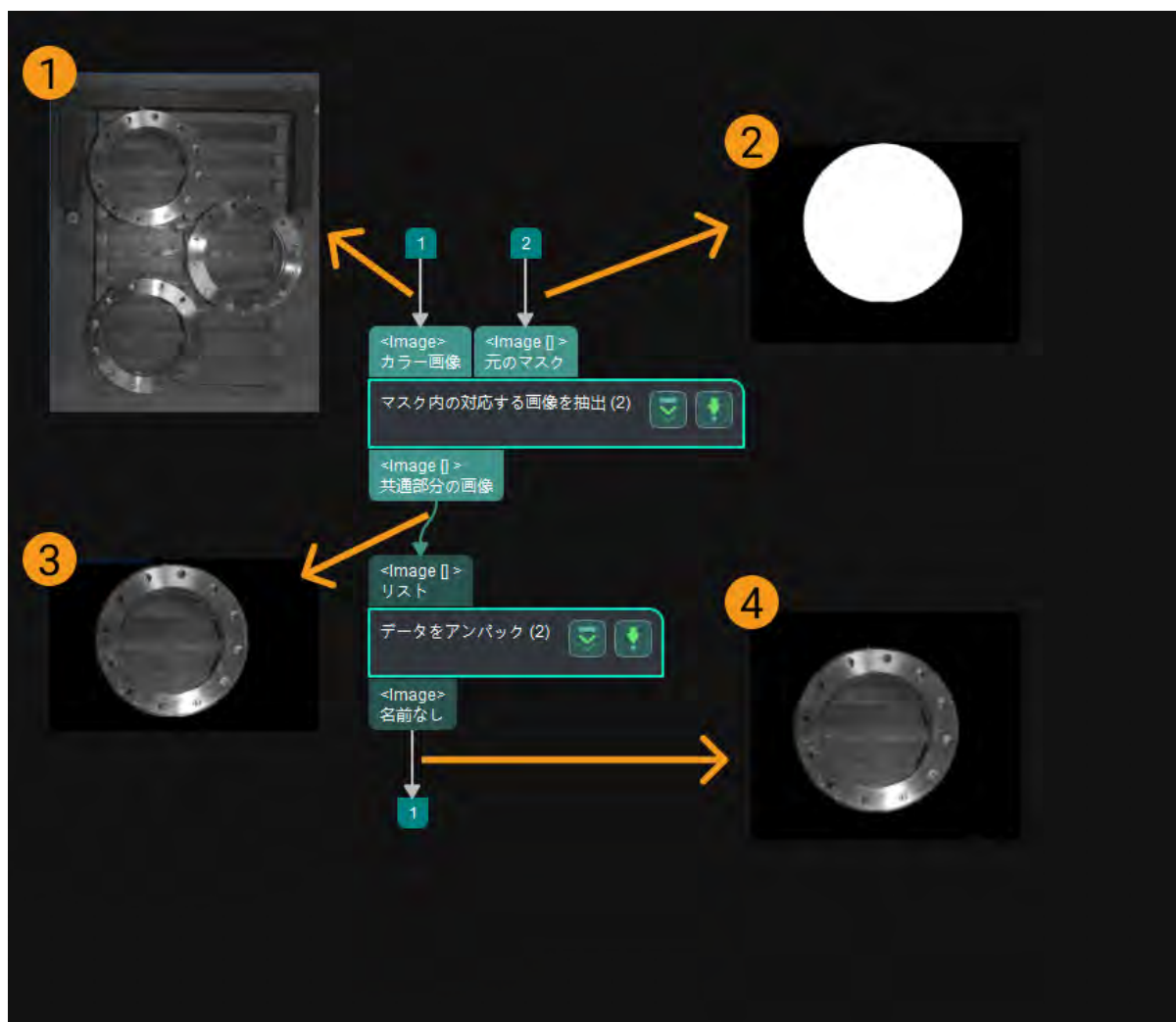
各ステップの機能

各ステップの機能は以下の通りです。

1. [\[vision-steps:apply-masks-to-image:::apply-masks-to-image\]](#) により、入力マスクで覆われた部分のカラー画像リストを取得します。
2. **データをアンパック** により、マスク画像リストの最初のインデックスが対応するカラー画像を取得します。

使用シーン

このステップの組合せは、インスタンスセグメンテーションによって出力されたマスクリストの最初のインデックスが対応するカラー画像を取得するために使用されます。



1. 入力カラー画像。
2. ディープラーニングによって出力されたワークのマスク画像で、3つのマスクが含まれています。
3. マスクが対応するカラー画像リストで、3つのカラー画像が含まれています。
4. 最初のインデックスが対応するカラー画像。

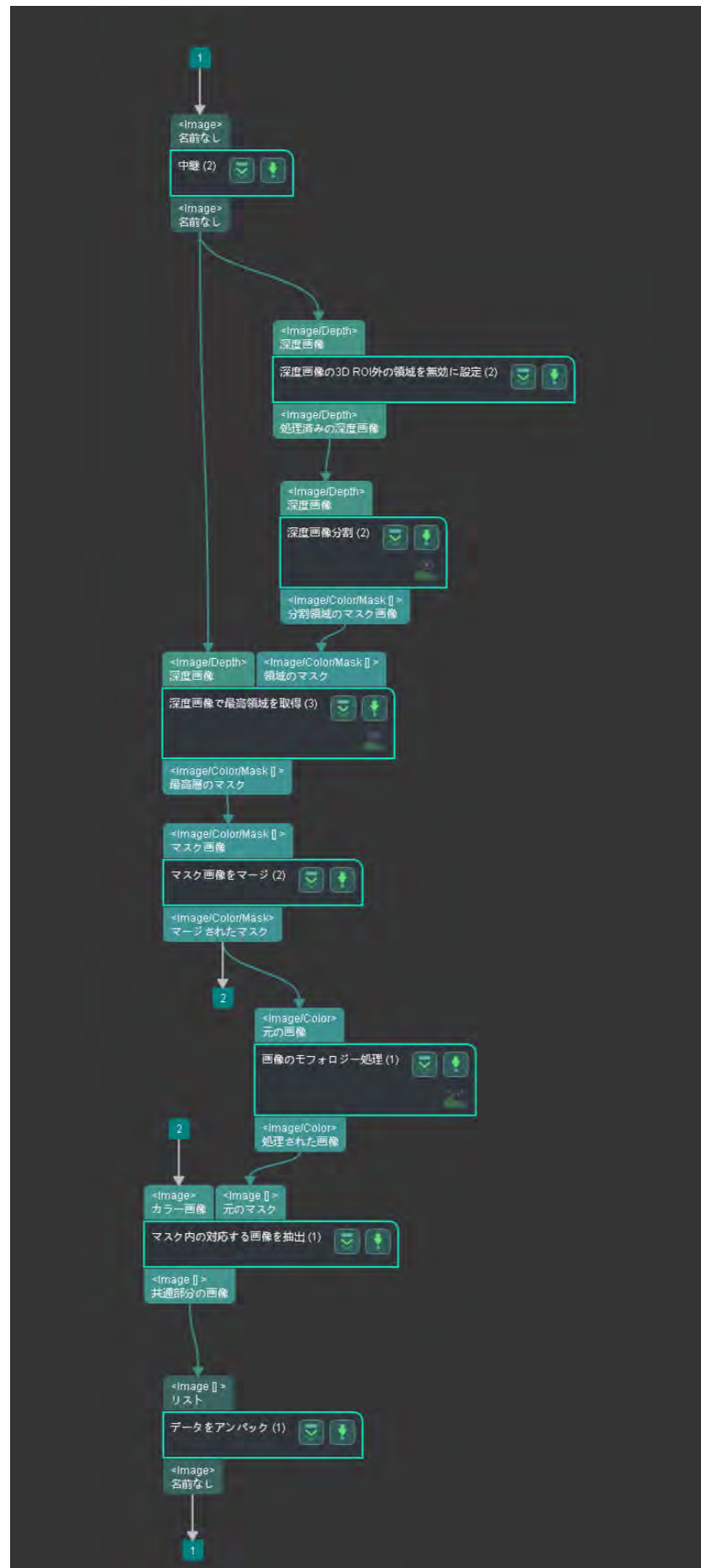
類似ステップ

マスクの対応するカラー画像を取得 によって出力されたものは、マスク画像リストの最初のインデックスが対応するカラー画像です。[vision-steps:apply-masks-to-image:::apply-masks-to-image] によって出力されたものは、すべてのマスクが対応するカラー画像リストです。

4.3.398. 最高層のカラー画像を取得

機能

このステップの組合せは、深度画像を通じて最高層領域を取得してから、対応する領域のカラー画像とマスクを取得するために使用されます。その構成を下図に示します。

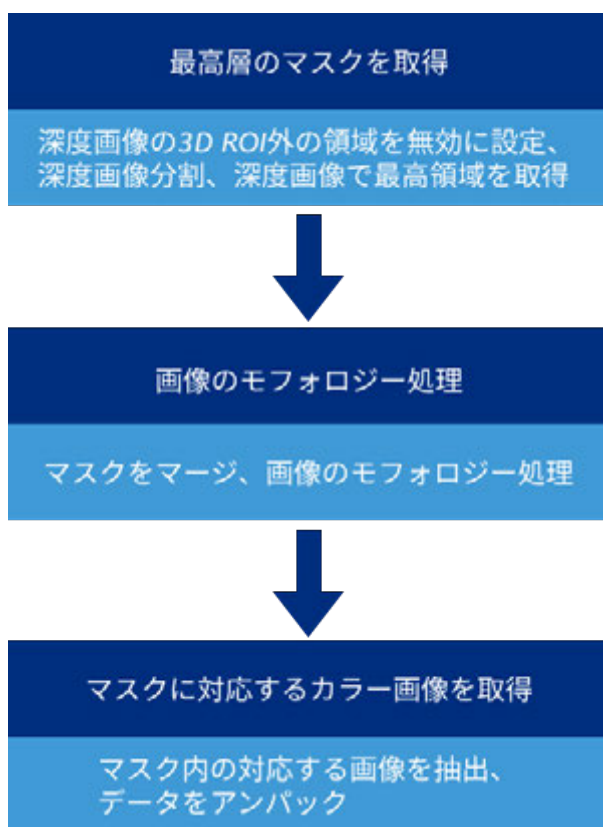


使用シーン

このステップの組合せによって抽出された最高層のカラー画像は、その後のディープラーニング処理に使用されます。

処理流れとパラメータ調整

このステップの組合せの処理流れは下図の通りです。

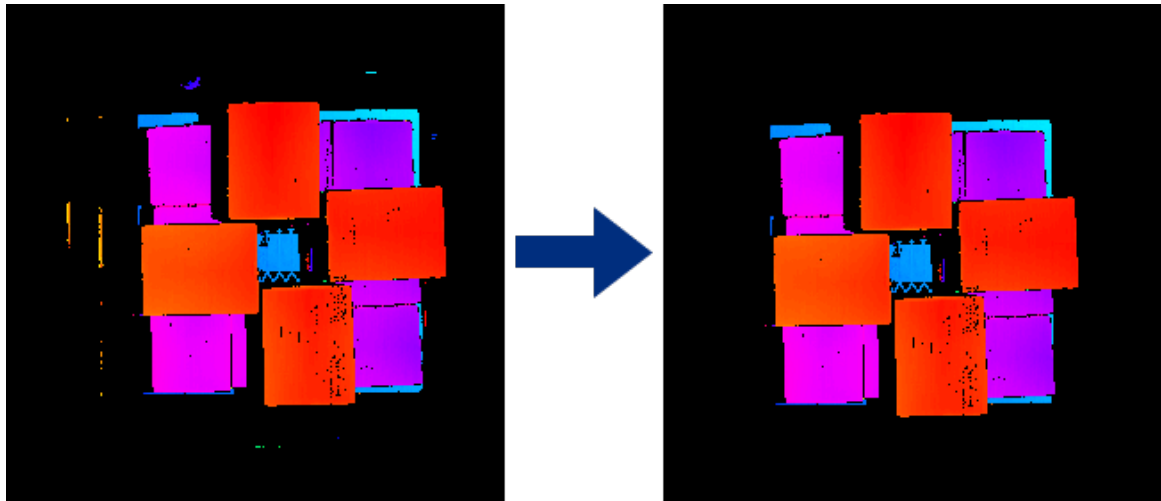


最高層のマスクを取得

1. 不要な点群を除去し、後続のステップの処理効率を向上させるために、必要な深度画像領域のみを保持し、元の深度画像で関心領域を選択する必要があります。

ダブルクリックしてその中に入ります。[vision-steps:invalidate-depth-pixels-outside-3d-roi::invalidate-depth-pixels-outside-3d-roi] ステップのパラメータで **[3D ROIを設定]** をクリックして、表示される画面で **3D ROIを設定** します。

3D ROI設定前後の点群を下図に示します。左側は入力された元の深度画像で、不要な点群が多いです。右側はROI設定後の深度画像で、ROI外の点群を除去します。

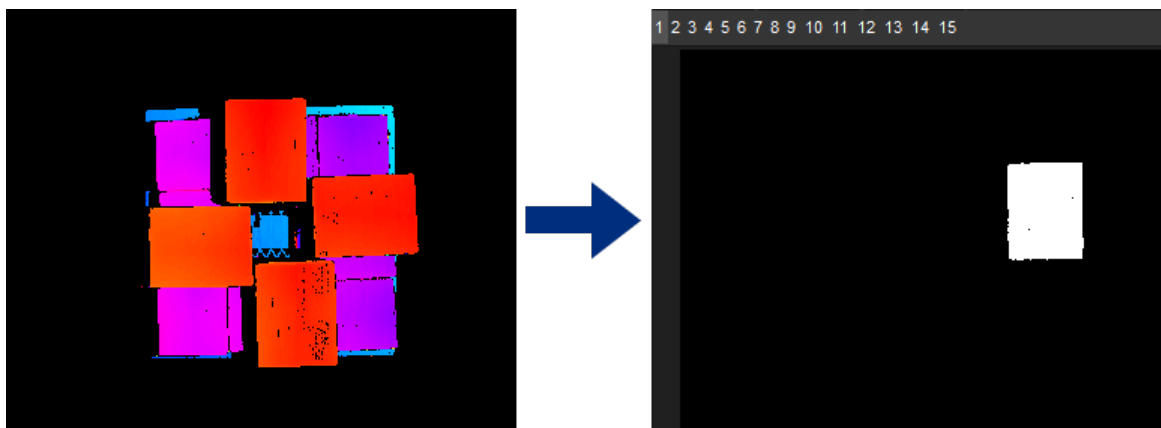


2. [\[vision-steps:segment-depth-image::segment-depth-image\]](#) ステップにより、深度に従って深度画像をいくつかの領域に分割し、マスクを生成します。

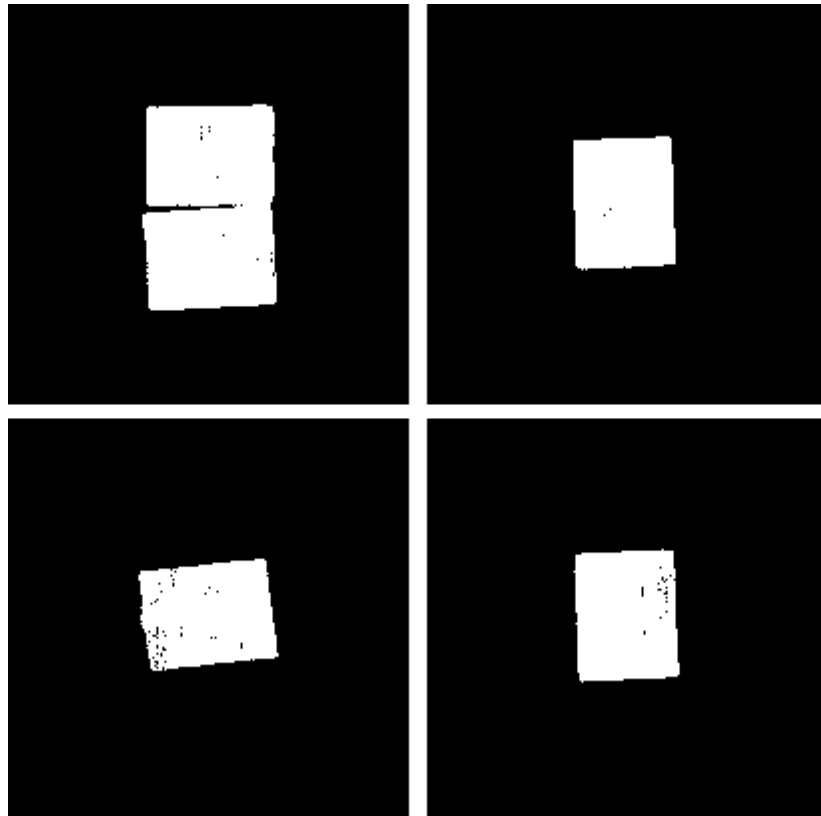
[\[vision-steps:segment-depth-image::segment-depth-image\]](#) のステップパラメータで **[2D ROIを設定]** をクリックして、表示される画面で **2D ROIを設定** します。

次に、**隣接するピクセル領域間の最大値の差** を設定し、隣接する2つのピクセルは同じ領域に属するかどうかを判断します。また、**対象物の最小面積** を設定し、有効な分割領域を判断します。

分割前後の効果を下図に示します。左側は分割前の深度画像で、右側は分割後の深度画像です。

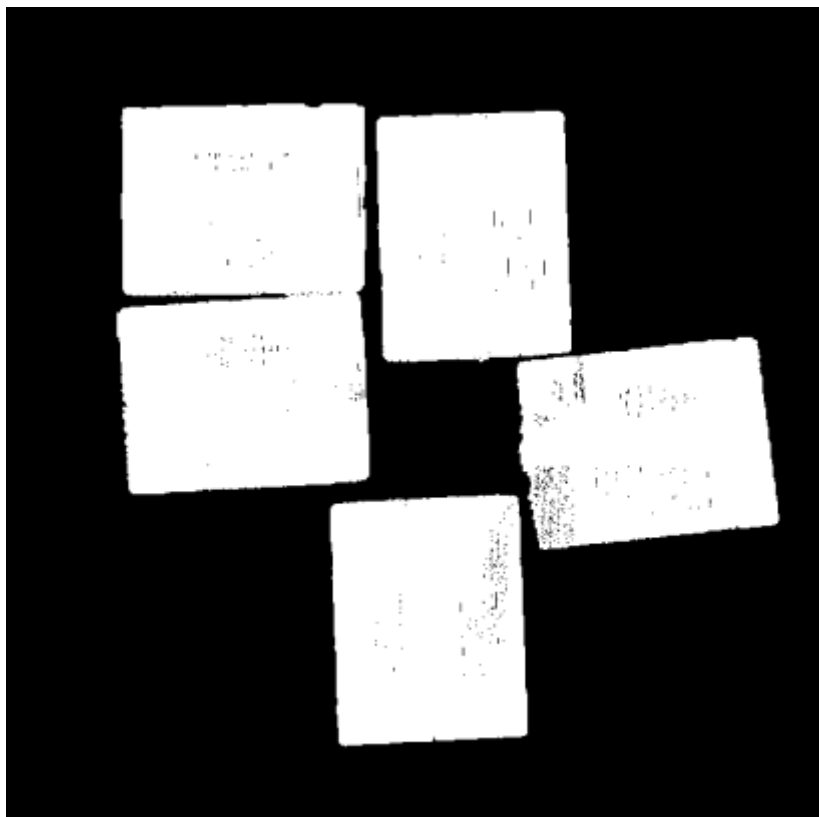


3. [\[vision-steps:get-highest-areas-in-depth-image::get-highest-area-in-depth-image\]](#) します。下図に示すように、前の手順で取得されたマスクおよび深度情報を使用し、**層の高さ** を設定して深度画像内の最高層のマスク領域を取得します。この領域は、1つまた複数がある可能性があります。

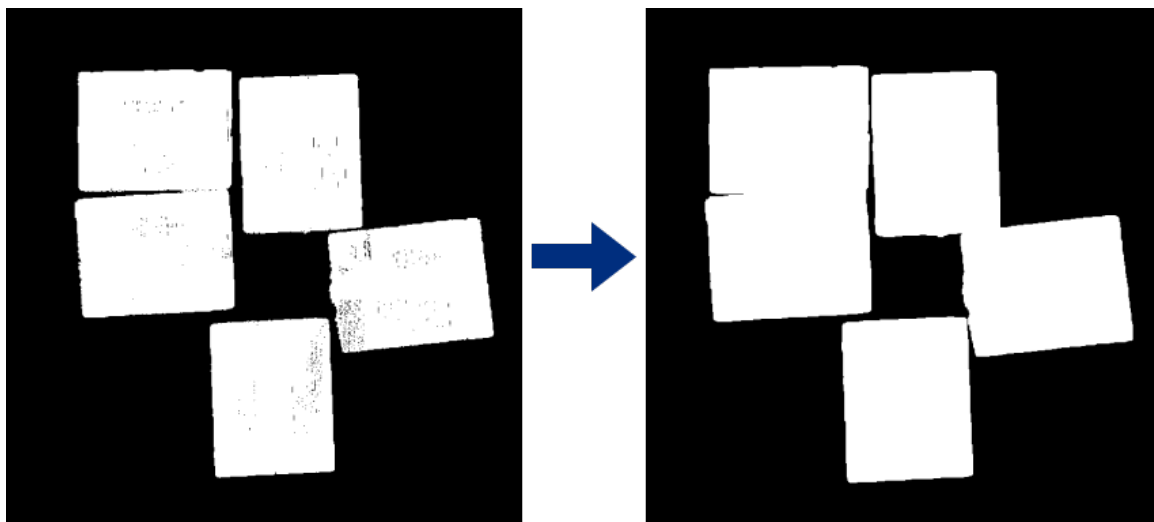


画像のモフォロジー処理

1. [\[vision-steps:merge-mask-images:::merge-mask-images\]](#) ステップを使用して、前の手順で取得された複数の最高層のマスクをマージします。これにより、マスク画像のモフォロジー処理を容易にします。マージされた最高層のマスクは下図の通りです。



2. マスクの領域を適切に拡大するために、マスクに対して [\[vision-steps:morphological-transformations:::morphological-transformations\]](#) を行う必要があります。マスクの **膨張処理**（モルフォロジー処理タイプはDilateOperator）により、その後の物体の認識や把持に便利になります。

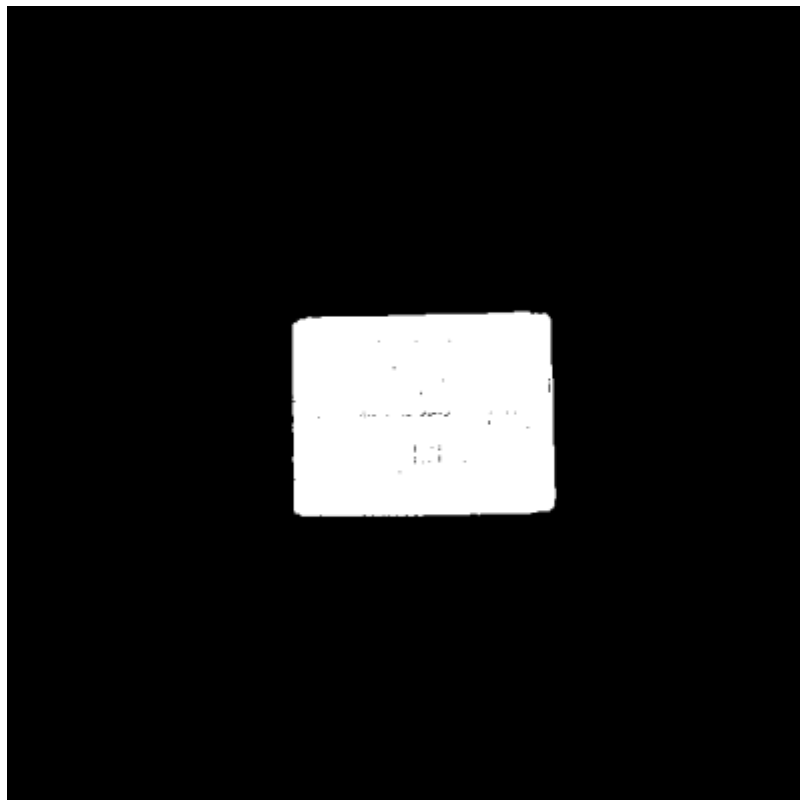


マスクの対応カラー画像を取得

1. 下図に示すように、カラー画像および処理後のマスクを使用して、[\[vision-steps:apply-masks-to-image:::apply-masks-to-image\]](#) ステップによりマスクに対応するカラー画像を抽出します。



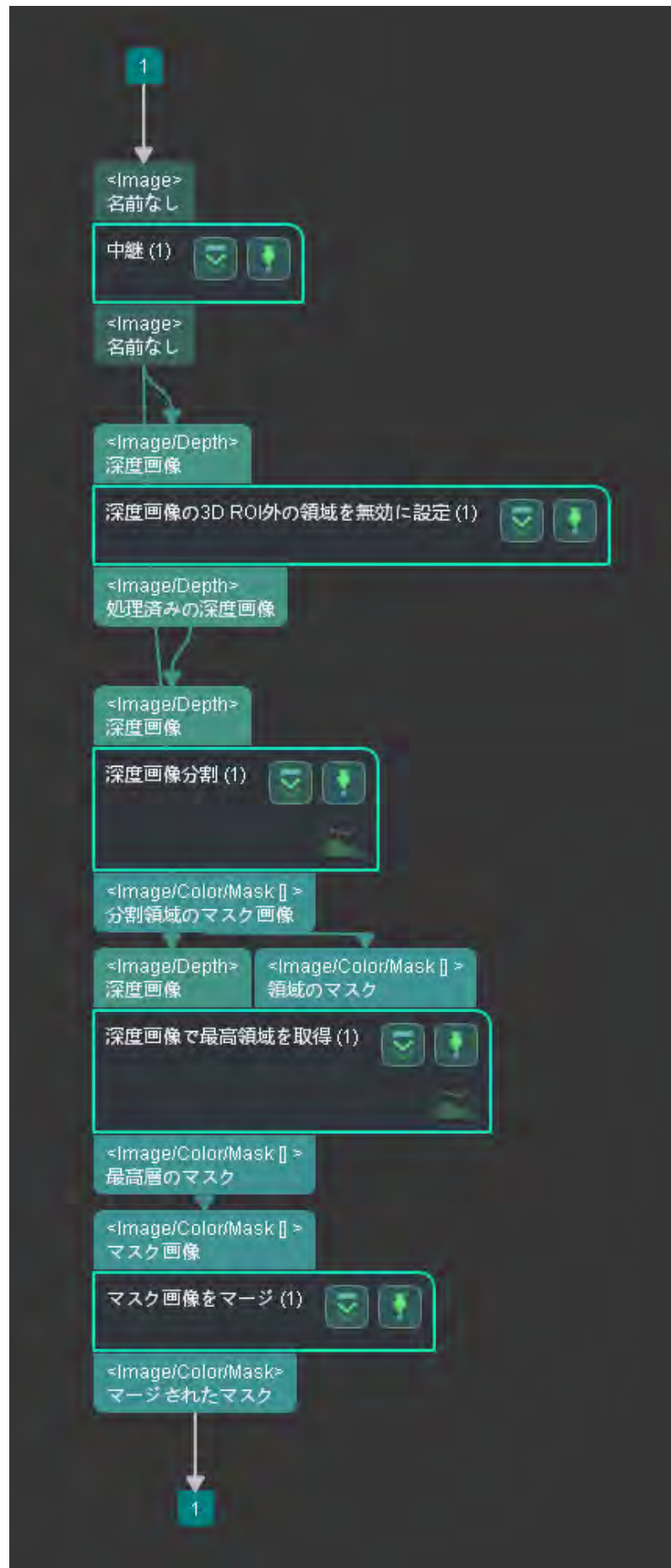
2. 下図に示すように、[\[vision-steps:unpack-data::unpack-data\]](#) ステップを使用してマスク画像リストの最初のインデックスに対応するカラー画像を取得します。



4.3.399. 最高層のマスクを取得

機能

このステップの組合せは、最高層のマスク画像を取得するために使用されます。その構成を下図に示します。

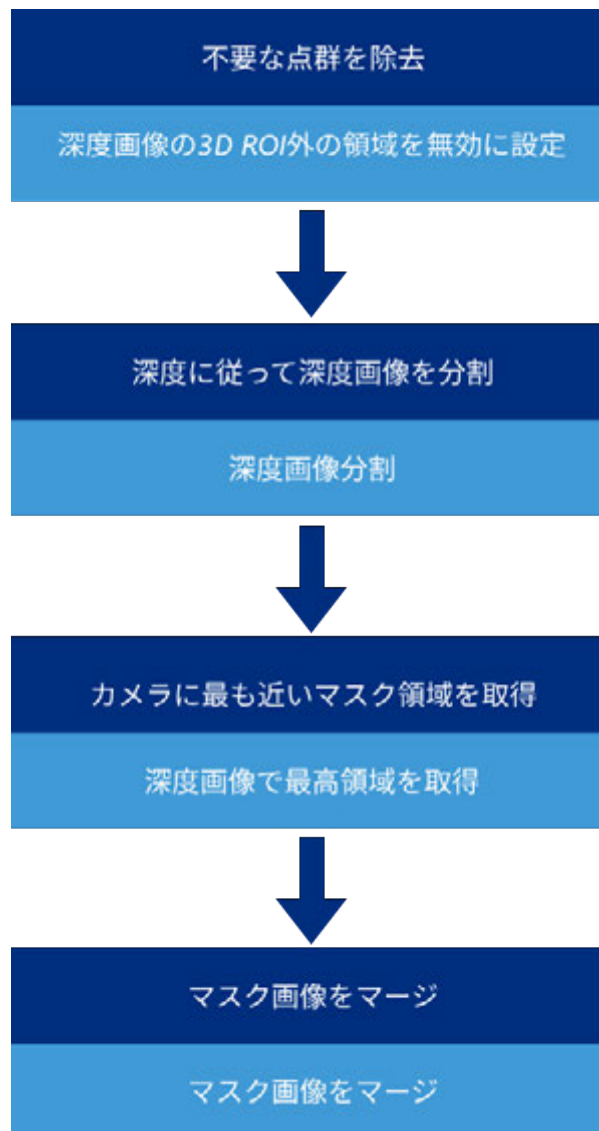


使用シーン

入力画像が深度方向に複雑で階層化する場合、このステップの組合せで無効な領域を除去し、深度画像を階層化して最高層マスクを取得します。

処理流れとパラメータ調整

このステップの組合せの処理流れは下図の通りです。

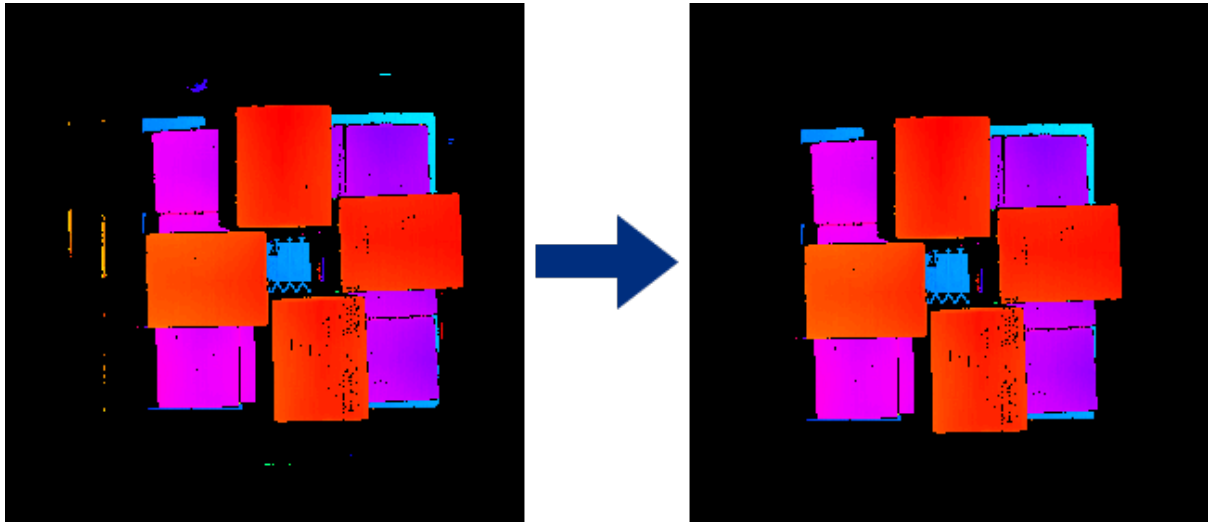


不要な点群を除去

不要な点群を除去し、後続のステップの処理効率を向上させるために、必要な深度画像領域のみを保持し、元の深度画像で関心領域を選択する必要があります。

ダブルクリックしてその中に入ります。 [\[vision-steps:invalidate-depth-pixels-outside-3d-roi::invalidate-depth-pixels-outside-3d-roi\]](#) ステップのパラメータで **[3D ROIを設定]** をクリックして、表示される画面で **3D ROIを設定** します。

3D ROI設定前後の点群を下図に示します。左側は入力された元の深度画像で、不要な点群が多いです。右側はROI設定後の深度画像で、ROI外の点群を除去します。



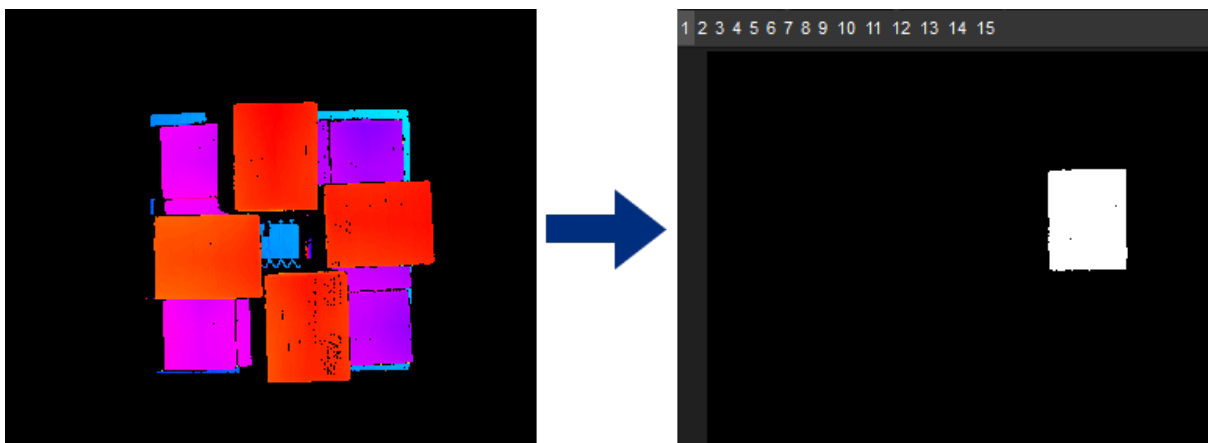
深度に従って深度画像を分割

[[vision-steps:segment-depth-image:::segment-depth-image](#)] ステップにより、深度に従って深度画像をいくつかの領域に分割し、マスクを生成することができます。

[[vision-steps:segment-depth-image:::segment-depth-image](#)] のステップパラメータで [2D ROIを設定] をクリックして、表示される画面で [2D ROIを設定](#) します。

次に、**隣接するピクセル領域間の最大値の差** を設定し、隣接する2つのピクセルは同じ領域に属するかどうかを判断します。また、**対象物の最小面積** を設定し、有効な分割領域を判断します。

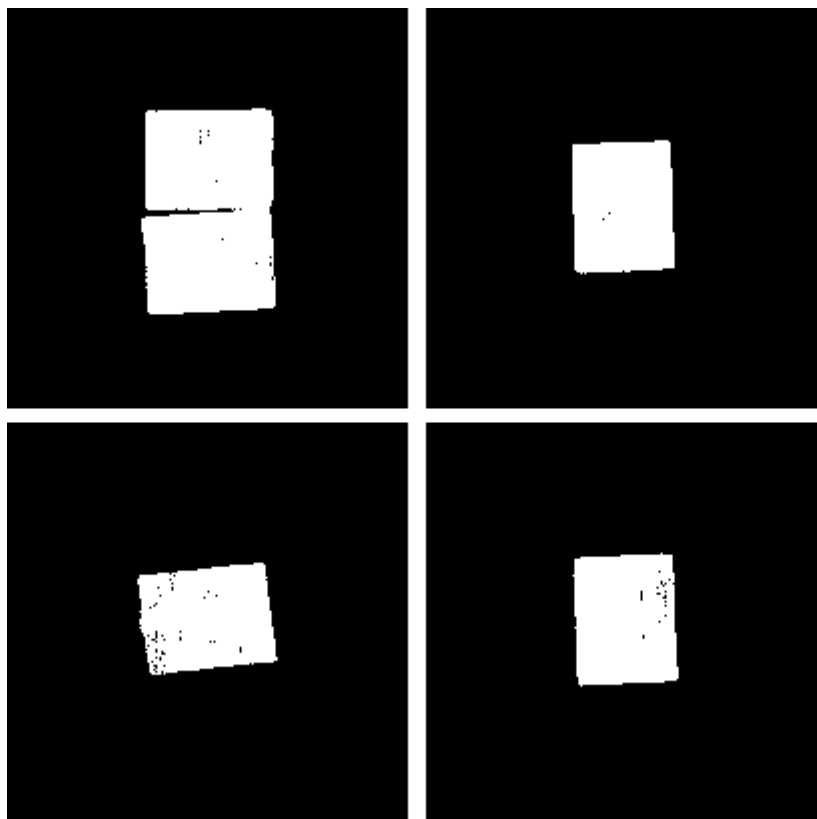
分割前後の効果を下図に示します。左側は分割前の深度画像で、右側は分割後の深度画像です。



カメラに最も近いマスク領域を取得

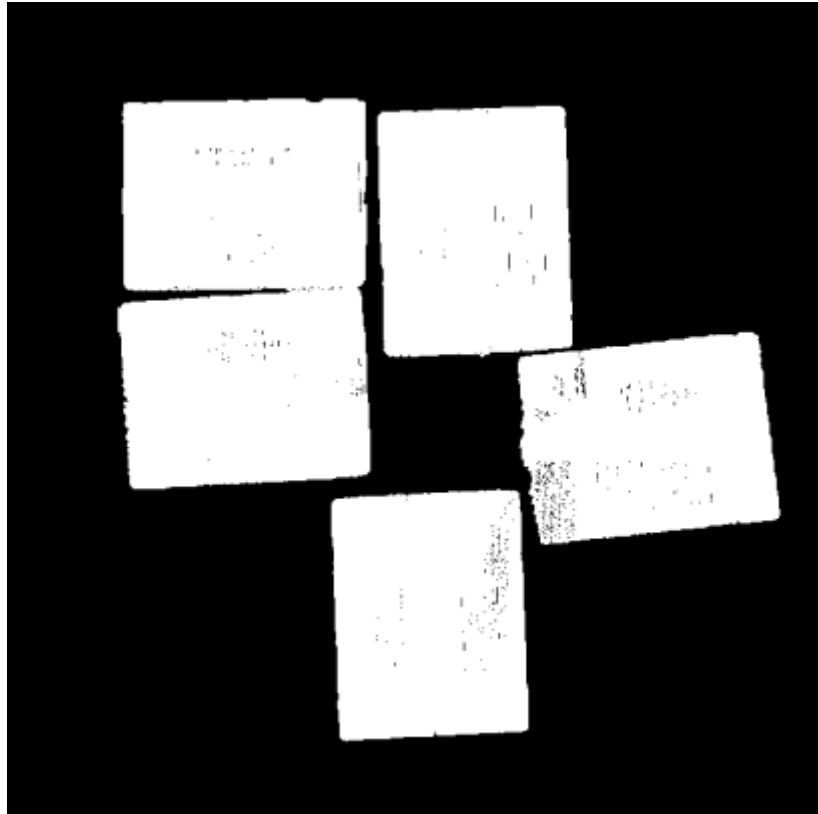
下図に示すように、前の手順で取得したマスクおよび深度情報に基づいて、 [[vision-steps:get-highest-areas-in-depth-image:::get-highest-area-in-depth-image](#)] ステップにより深度画像内

の最も高い（カメラ底部に最も近い）マスク領域を取得できます。**層の高さ**を適切に設定する必要があることに注意してください。得られたマスク領域は、1つまたは複数の場合があります。



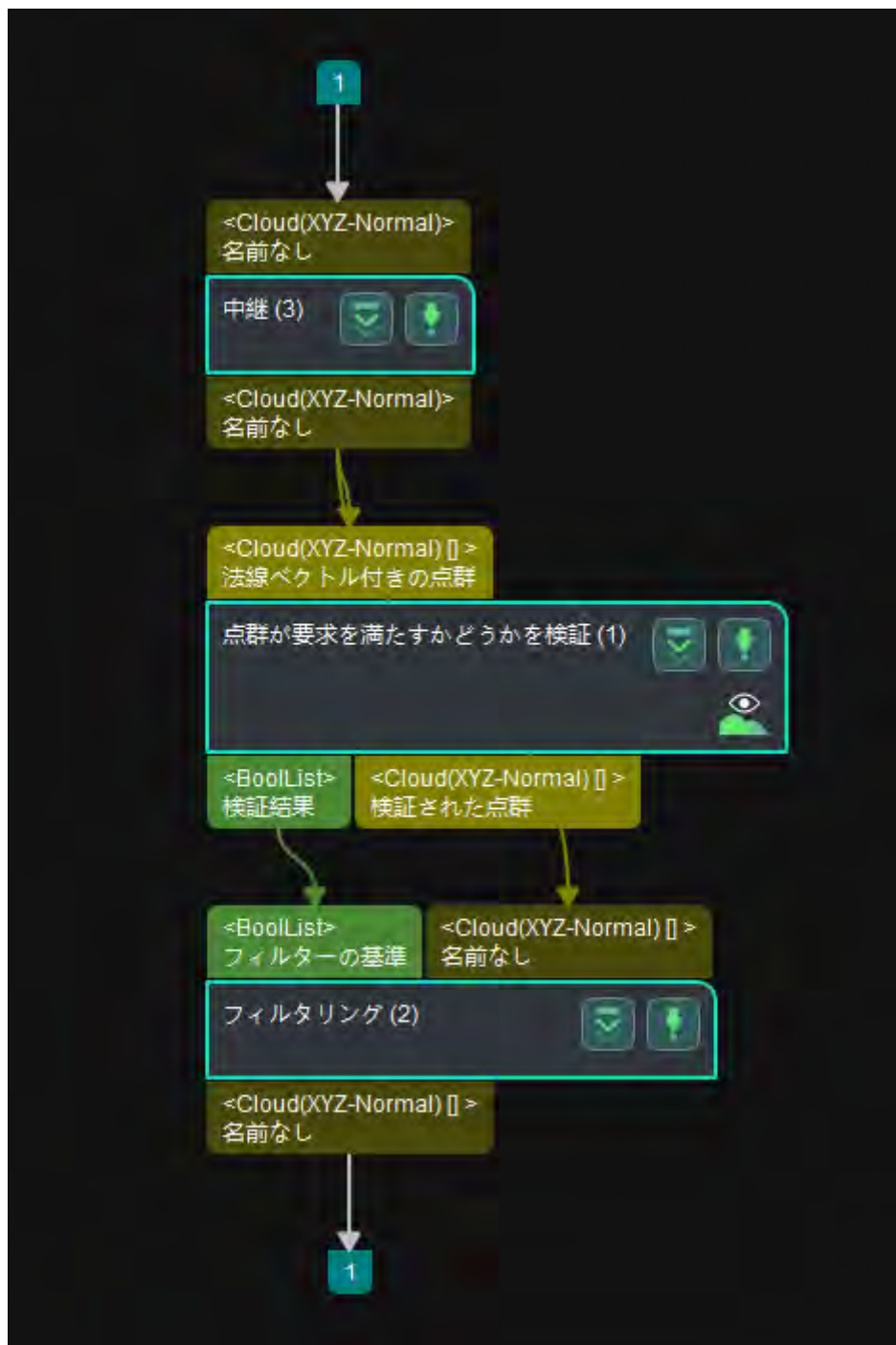
マスク画像をマージ

[[vision-steps:merge-mask-images:::merge-mask-images](#)] ステップを使用して前の手順で得られた複数の最高層のマスクをマージし、最高層のマスクを取得します。マージされた最高層のマスクを下図に示します。



4.3.400. 点の数が制限を超えた点群を除外

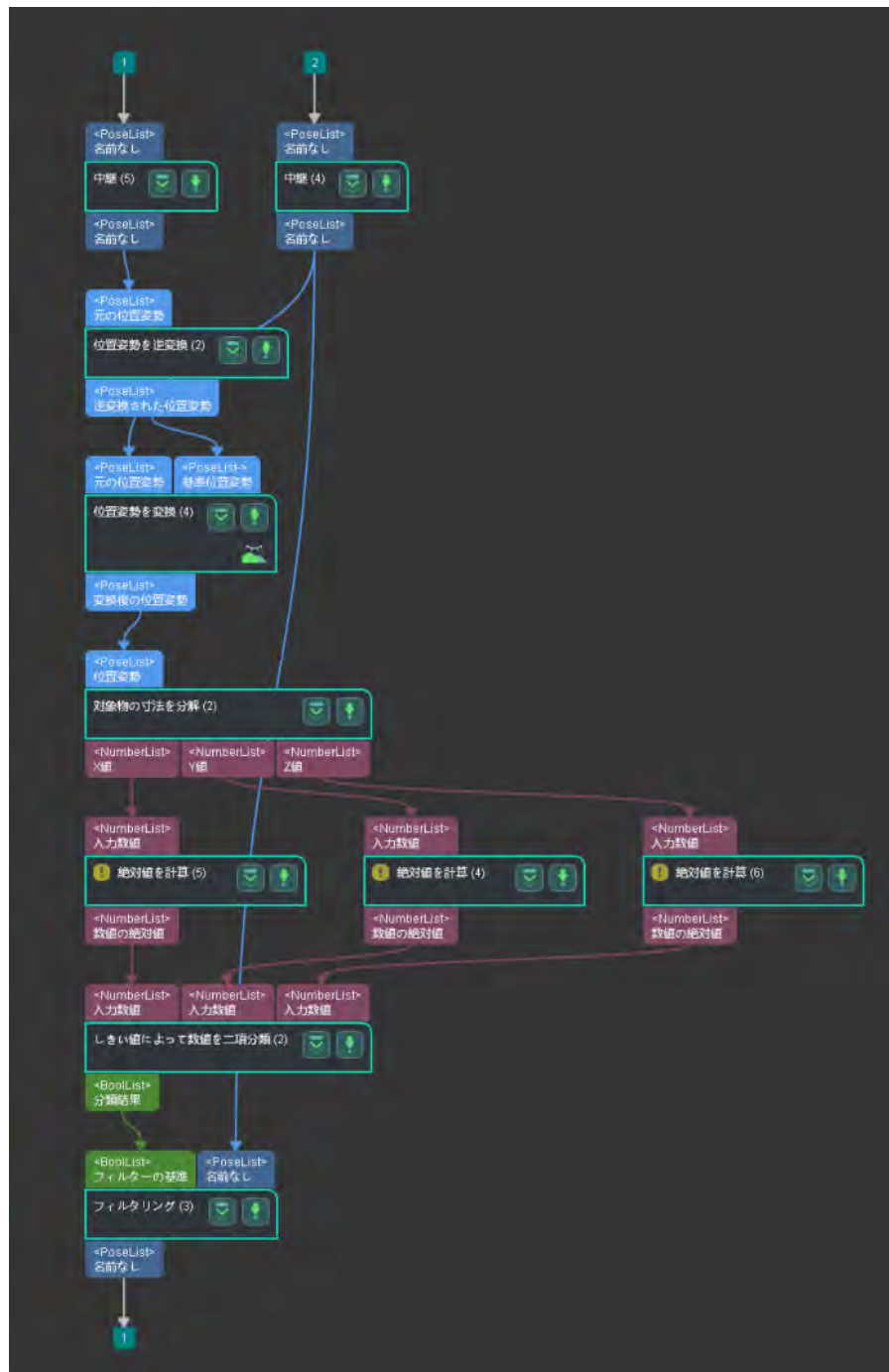
主な機能は、適切な点群サイズ範囲を設定して範囲外の点群を除去することです。それにより、プロジェクトの実行速度を高速化させます。その構成下図に示します。



[[vision-steps:validate-point-clouds:::validate-point-clouds](#)] により、点群が設定された範囲にあるかどうかを判断し、判断結果を **フィルタリング** ステップに入力します。フィルタリングステップにより、要件を満たしている点群は保持され、残りの点群は放棄されます。最後にフィルタリング後の点群を出力します。

4.3.401. ROI以外の位置姿勢を除去

主な機能は、関心領域外の位置姿勢を除去し、その後の位置姿勢処理を高速化させることです。その構成を下図に示します。



各ステップの機能は以下の通りです。

1. [\[vision-steps:inverse-poses:::inverse-poses\]](#) により、関心領域の中心位置姿勢に対して逆変換を行います。
2. [位置姿勢を変換](#) により、判断が必要な位置姿勢と逆変換後の関心領域の中心位置姿勢を入力し、判断が必要な位置姿勢の相対座標系を関心領域の中心位置姿勢の座標系に変換します。
3. [\[vision-steps:decompose-object-dimensions:::decompose-object-dimensions\]](#) と [絶対値を計算](#) により、位置姿勢のXYZ座標の絶対値を取得します。
4. [\[vision-steps:dichotomize-values-by-threshold:::dichotomize-values-by-threshold\]](#) によ

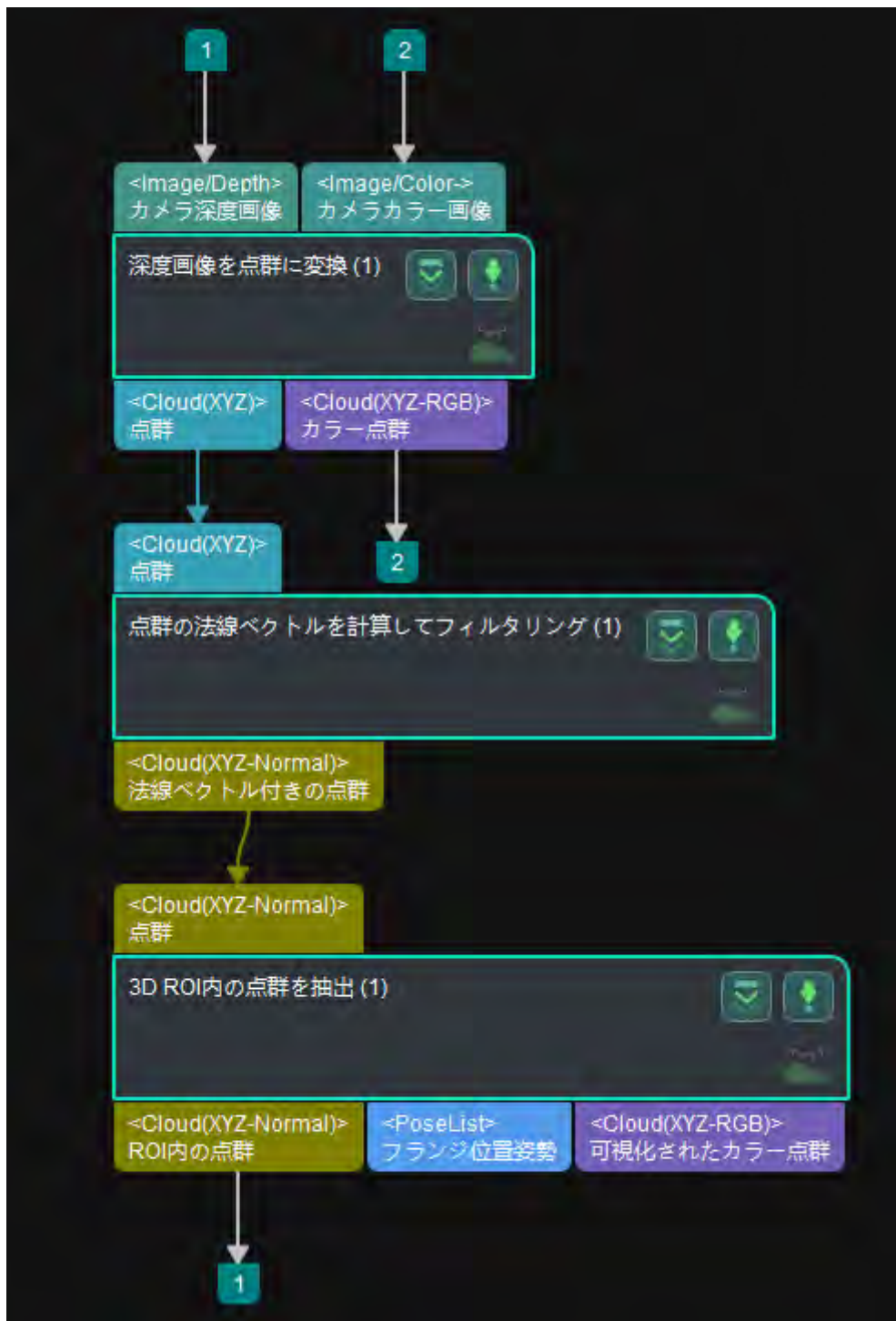
り、関心領域を設定し、XYZの絶対値が関心領域にあるかどうかを判断します。

5. **フィルタリング** により、関心領域外の位置姿勢を除去します。

4.3.402. 点群前処理

機能

このステップの組合わせの主な機能は、点群生成、干渉する点群の除去、関心領域内の点群の取得であり、これにより後続ステップの処理を高速化します。その構成を下図に示します。



使用シーン

このステップの組み合わせは、通常 [カメラから画像を取得](#) ステップの後に使用され、元の点群の前処理を行い、後続のステップの処理時間を短縮することができます。

処理流れとパラメータ調整

処理流れは、[深度画像を点群に変換](#)、[点群の法線ベクトルを計算してフィルタリング](#)、[3D ROI内の点群を抽出](#) で構成されています。

深度画像を点群に変換

まず、元のカラー画像と深度画像を [深度画像を点群に変換](#) ステップに入力し、シーンの点群を出力します。

不要な点群を除去するために、2D ROIを設定する必要があります。「2D ROIを設定」をクリックし、表示される画面で2D ROIを設定します。

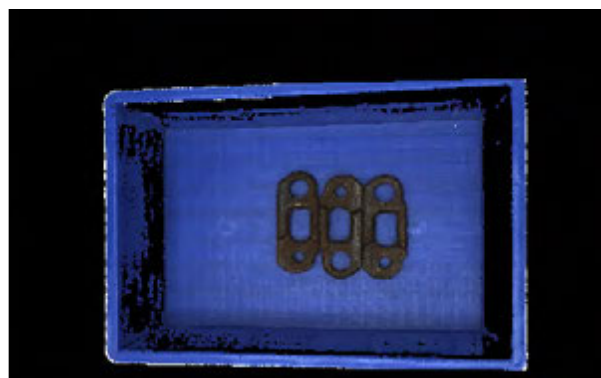
点群の法線ベクトルを計算してフィルタリング

次に、シーンの点群を [点群の法線ベクトルを計算してフィルタリング](#) ステップに入力し、法線ベクトルを計算して不要な点を除去し、フィルタリング後の法線ベクトル付きの点群を出力します。

3D ROI内の点群を抽出

最後に前のステップで処理された点群を [3D ROI内の点群を抽出](#) ステップに入力し、関心領域（ROI）の設定により、ROI内の点群を抽出して出力します。

点群前処理を行う前と後の効果を下図に示します。左側は前処理前の点群であり、関心領域を設定することではなく、不要なシーンの点群を含んでいます。右側は前処理後の点群であり、関心領域を設定して不要な点群が除去され、後のステップ処理を容易にします。



パラメータの調整説明

3D ROI内の点群を抽出

3D ROIを使用

初期値：チェックを入れる。

調整説明：チェックを入れると、3D ROIが使用されます。

3D ROI内の最小点数

初期値：0

調整説明：3D ROI内の点がこの値より少ない場合、点群は返されません。

3D ROI内の点が空であるかどうかを送信

初期値：チェックを入れる。

調整説明：このパラメータにチェックを入れると、3D ROI内の点群が空であるかどうかを判断し、結果を送信します。

深度画像を点群に変換

深度画像による背景を除去

初期値：チェックを入れない。

調整説明：このパラメータは背景除去に使用されます。チェックを入れると、入力深度画像は背景画像と比較し、深度差が背景深度の変動範囲内であれば、背景が除去されることになります。

深度背景画像のファイル名

調整説明：右側にある [2D ROIを設定] をクリックし、表示される画面で背景を設定します。

背景深度の変動

初期値：10mm

調整説明：このパラメータは、背景深度の変動のしきい値を設定するために使用されます。背景画像との深度差がこの値より小さい点は除去されます。

点群の法線ベクトルのを計算してフィルタリング

法線ベクトルを計算するときの半径

初期値：5.00 mm

調整説明：この半径範囲にある点は、法線ベクトルを計算するために使用されます。

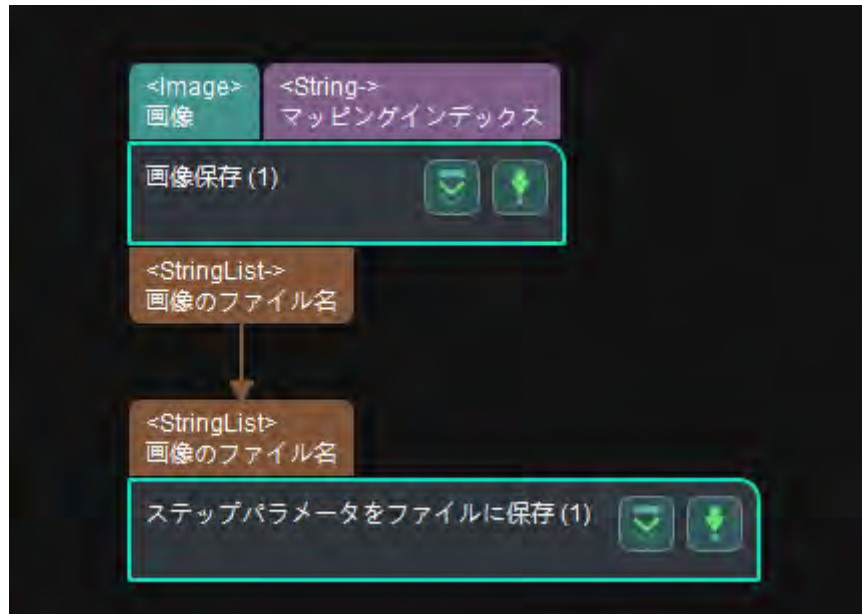
法線ベクトルの計算方法

初期値：CloudPreprocessingGPU

調整説明：このパラメータは、法線ベクトルの計算方法を選択するために使用されます。

4.3.403. 画像のデータとパラメータを保存

主な機能は、画像を指定したフォルダに保存することと、ステップのパラメータをプロジェクトに保存することです。その構成を下図に示します。



各ステップの機能は以下の通りです。

1. [\[vision-steps:save-images:::save-images\]](#) は保存する画像（カラー画像、深度画像、および保存する必要があるその他の画像を含む）を入力できます。これにより、必要に応じて複数の画像を入力でき、画像名のインデックスを追加できます。
2. [\[vision-steps:save-step-parameters-to-file:::save-step-parameters-to-file\]](#) は選択可能なステップです。このステップを追加すると、ステップのパラメータをファイルに保存できます。

4.4. ハンド・アイ・キャリブレーションの使用ガイド

本章では、ロボットハンド・アイ・キャリブレーションの実行方法について説明します。

ロボットハンド・アイ・キャリブレーション手順の選択については、以下の内容をお読みください。

[ロボットハンド・アイ・キャリブレーションの実行手順を選択](#)

ロボットハンド・アイ・キャリブレーションを実行するには、以下の内容をお読みください。

[自動キャリブレーション（Eye to Hand）](#)

[自動キャリブレーション \(Eye in Hand\)](#)

[標準キャリブレーション法を使用する手動キャリブレーション \(Eye to Hand\)](#)

[標準キャリブレーション法を使用する手動キャリブレーション \(Eye in Hand\)](#)

[TCPタッチ法を使用する手動キャリブレーション \(Eye to Hand\)](#)

[TCPタッチ法を使用する手動キャリブレーション \(Eye in Hand\)](#)

[直行ロボットのハンド・アイ・キャリブレーション](#)

[ハンド・アイ・キャリブレーション \(Eye to Eye\)](#)

ロボットのハンド・アイ・キャリブレーションの関連情報については、以下の内容をお読みください。

[キャリブレーションの関連概念](#)

[キャリブレーション原理](#)

[直行ロボットのキャリブレーション説明](#)

[キャリブレーション結果の確認と分析](#)

[よくある問題と解決策](#)

4.4.1. ロボットハンド・アイ・キャリブレーションの実行手順を選択

ハンド・アイ・キャリブレーションとは、**カメラ座標系とロボット座標系との対応関係を求めること**を指します。ビジョンシステムにより取得した対象物の位置姿勢をカメラ座標系に変換し、それによってロボットが正確に把持作業を完了するように制御します。ハンド・アイ・キャリブレーションの結果精度は、ロボットの把持精度に影響を与える最も重要な要素の1つです。

Mech-Mindビジョンシステムの構築が完了したら、ロボットのハンド・アイ・キャリブレーションを実行します。また、以下のような場合にはロボットのハンド・アイ・キャリブレーションを再度実行する必要があります。

- カメラは、ロボット架台（カメラがEye to Hand方式で取り付けられている）またはフランジ（カメラがEye in Hand方式で取り付けられている）に対して位置を変えます。
- カメラを交換します。
- ロボットを交換します。

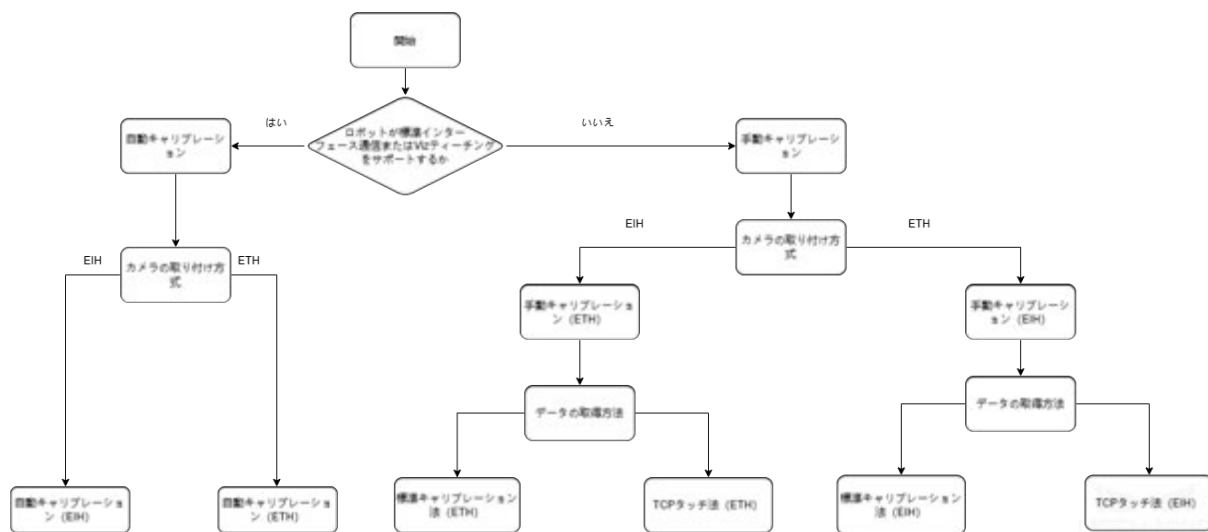
キャリブレーション手順を選択

ロボットの通信方式、カメラの取り付け方式、キャリブレーション実行方法、データの取得方法など、キャリブレーションの実行に影響します。Mech-Visionでは、カメラの内部および外部パラメータの迅速かつ正確なキャリブレーションを容易にするために、様々なシーンに対して標準化されたキャリブレーション手順が提供されています。



ロボットの通信方式、カメラの取り付け方式、キャリブレーション実行方法、データの取得方法などの詳細については、[キャリブレーションの関連概念](#)をご参照ください。

下図に従ってキャリブレーションの実行手順を選択してください。



1. ロボットが標準インターフェース通信やVizティーチング通信をサポートするかどうかを確認します。
 - サポートする場合、自動キャリブレーションを選択して手順2に進みます。
 - サポートしない場合、手動キャリブレーションを選択して手順3に進みます。
2. 自動キャリブレーションを選択する場合、カメラの取り付け方式を確認します。
 - カメラの取り付け方式がEye to Hand (ETH)の場合、[自動キャリブレーション \(Eye to Hand\)](#) を選択します。
 - カメラの取り付け方式がEye in Hand (EIH)の場合、[自動キャリブレーション \(Eye in Hand\)](#) を選択します。
3. 手動キャリブレーションを選択する場合、カメラの取り付け方式を確認します。
 - カメラの取り付け方式がEye to Hand (ETH)の場合、キャリブレーションデータの取得方法を確認します。
 - 標準キャリブレーション法でデータを取得する場合、[標準キャリブレーション法を使用する手動キャリブレーション \(Eye to Hand\)](#) を選択します。
 - TCPタッチ法でデータを取得する場合、[TCPタッチ法を使用する手動キャリブレーション \(Eye to Hand\)](#) を選択します。

- カメラの取り付け方式がEye in Hand (EIH)の場合、キャリブレーションデータの取得方法を確認します。
 - 標準キャリブレーション法でデータを取得する場合、[標準キャリブレーション法を使用する手動キャリブレーション \(Eye in Hand\)](#) を選択します。
 - TCPタッチ法でデータを取得する場合、[TCPタッチ法を使用する手動キャリブレーション \(Eye in Hand\)](#) を選択します。

2台カメラを使用する場合、[ハンド・アイ・キャリブレーション \(Eye to Eye\)](#) を参照してキャリブレーションを実行してください。直行ロボットを使用する場合、[直行ロボットのハンド・アイ・キャリブレーション](#) を参照してキャリブレーションを実行してください。

4.4.2. 自動キャリブレーション (Eye to Hand)

本節では、カメラがEye to Hand (ETH) 方式で取り付けられている場合に自動キャリブレーションを実行する方法について説明します。

事前準備

以下は、ロボットハンド・アイ・キャリブレーションを実行する前に必要な事前準備です。

- Mech-Mindビジョンシステムを構築します。
- キャリブレーションに必要なものを準備します。
- キャリブレーションボードの点群画像生成を調整します。
- ロボットと通信の設定を完了します。

ビジョンシステムを構築

Mech-Mindビジョンシステムの構築については、「[ビジョンシステムを構築](#)」節をご参照ください。

ハンド・アイ・キャリブレーションには、Mech-Eye Viewer、Mech-Center、Mech-VisionおよびMech-Vizが必要です。これらのソフトウェアがインストールされ、最新バージョンにアップグレードされていることを確認してください。

キャリブレーションに必要なものを準備

カメラがEye to Hand方式で取り付けられている場合、自動キャリブレーションを実行するにはキャリブレーションボードの使用が必要です。

キャリブレーションボードについては、以下の要件があります。

- キャリブレーションボードの白い円がはっきりと見えること、破損や曲げ変形がないことを確認してください。
- キャリブレーションボードのコネクタをロボット先端のフランジに取り付け、そのコネクタ

にキャリブレーションボードを取り付けます。キャリブレーションボードがしっかりと取り付けられていること、ロボット先端のXY平面と平行であることを確認してください。



ロボットフランジに取り外しができないハンドがある場合、キャリブレーションボードをハンドに直接固定します。

また、キャリブレーションを実行する前に、ロボットをキャリブレーションの開始位置に移動させてください。Eye to Handでは、キャリブレーションの開始位置はカメラ視野の最下部中央にあります。ロボットは、キャリブレーションボードを移動させて下から上にキャリブレーションを実行します。


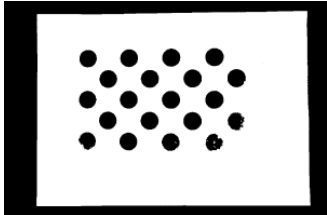
キャリブレーションボードの点群画像生成を調整

1. Mech-Eye Viewerソフトウェアを起動し、カメラのパラメータを調整します。
2. 2Dパラメータを調整して、2D画像にキャリブレーションボードがはっきり見られ、露光過度や露光不足などがないようにします。
3. 3Dパラメータを調整して、キャリブレーションボードの白い円の点群が完全に見られるようにします。点群の変動範囲を小さくするために、**点群後処理**の中の**点群平滑化**と**ノイズ除去**を**Normal**に設定することを推奨します。



現場の環境光が複雑な場合、2D画像や点群への影響を軽減するために、遮光・補光することを推奨します。

1. 上記の手順を完了することで、キャリブレーションボードの点群の画像品質が要件を満たしていることを確認します。

	正常	露光過度	露光不足
2D画像			
点群			

ロボット通信設定

ロボット側は標準インターフェース方式でビジョン側と通信する場合、ロボット側で標準インターフェースの通信設定を行ってください。詳細については、[標準インターフェース通信](#)をご参照ください。

ロボット側はVizティーチング方式でビジョン側と通信する場合、ロボット側でVizティーチングの通信設定を行ってください。詳細については、[Vizティーチング通信](#)をご参照ください。

キャリブレーション前の設定

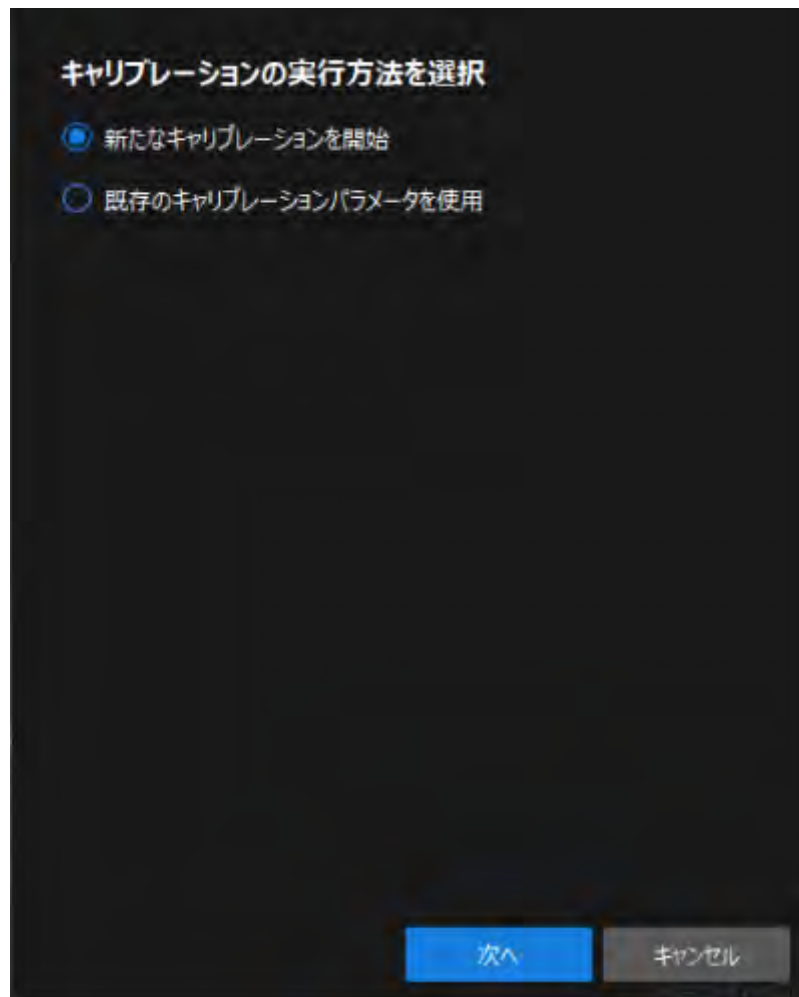
ロボット側は標準インターフェース方式でビジョン側と通信する場合、[キャリブレーション前の設定（標準インターフェース）](#)を行ってください。

ロボット側はVizティーチング方式でビジョン側と通信する場合、[キャリブレーション前の設定（Vizティーチング）](#)を行ってください。

キャリブレーション前の設定（標準インターフェース）

以下の手順を実行します。

1. Mech-Visionを起動し、ツールバーで[**カメラキャリブレーション（標準モード）**]をクリックします。すると、**キャリブレーション前の設定**画面が表示されます。
2. **キャリブレーションの実行方法を選択**画面で、**新たなキャリブレーションを開始**を選択し、[**次へ**]をクリックします。

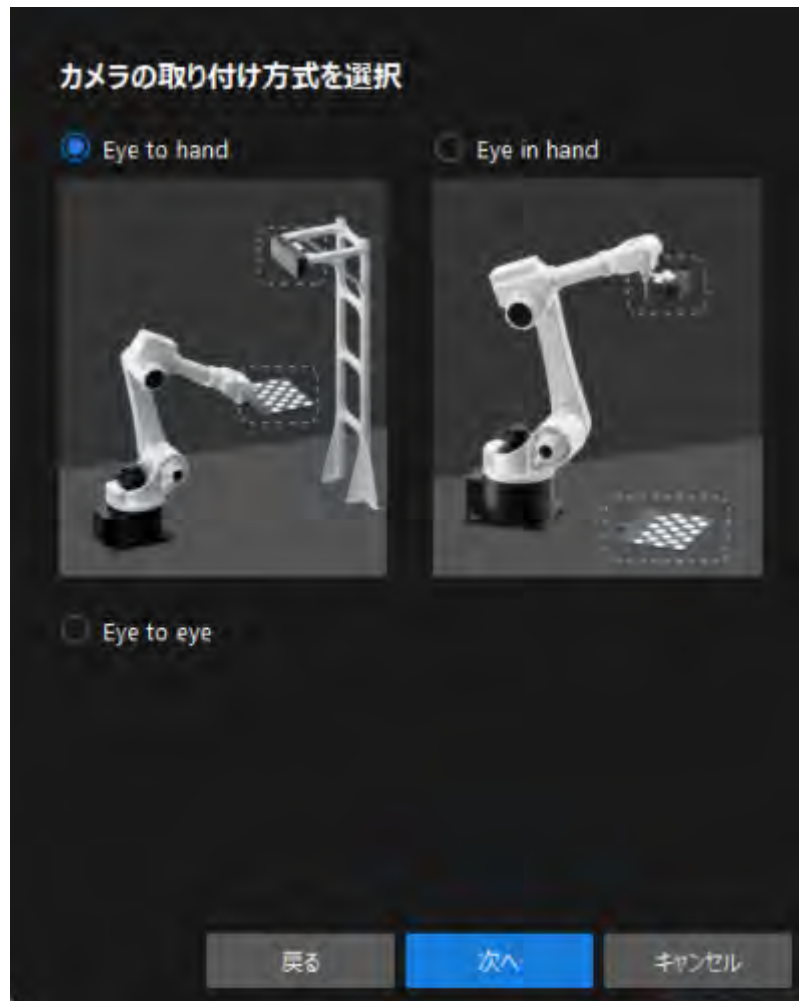


3. **キャリブレーションのタスクを選択**画面で、ドロップダウンリストから **適応可能なロボッ**

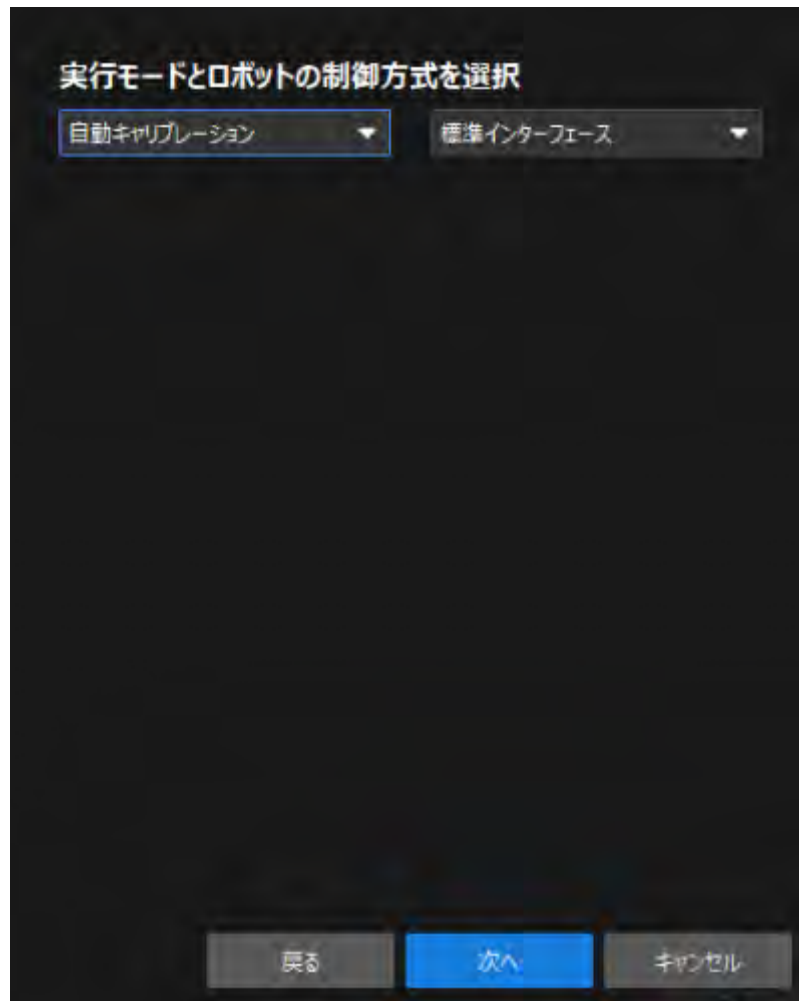
トのハンド・アイ・キャリブレーションを選択し、[ロボット型番を選択]をクリックして型番を選択してから、[次へ]をクリックします。



4. カメラの取り付け方式を選択画面で、**Eye to hand**を選択して[次へ]をクリックします。



5. 実行モードとロボットの制御方式を選択 画面で、自動キャリブレーションと標準インターフェースを選択してから、[次へ]をクリックします。



6. 通信設定画面で、実際の状況に応じて通信プロトコルとホストIPアドレスを設定してから[インターフェースサービスを起動]をクリックします。このボタンがロボットの接続を待機中…に変わります。



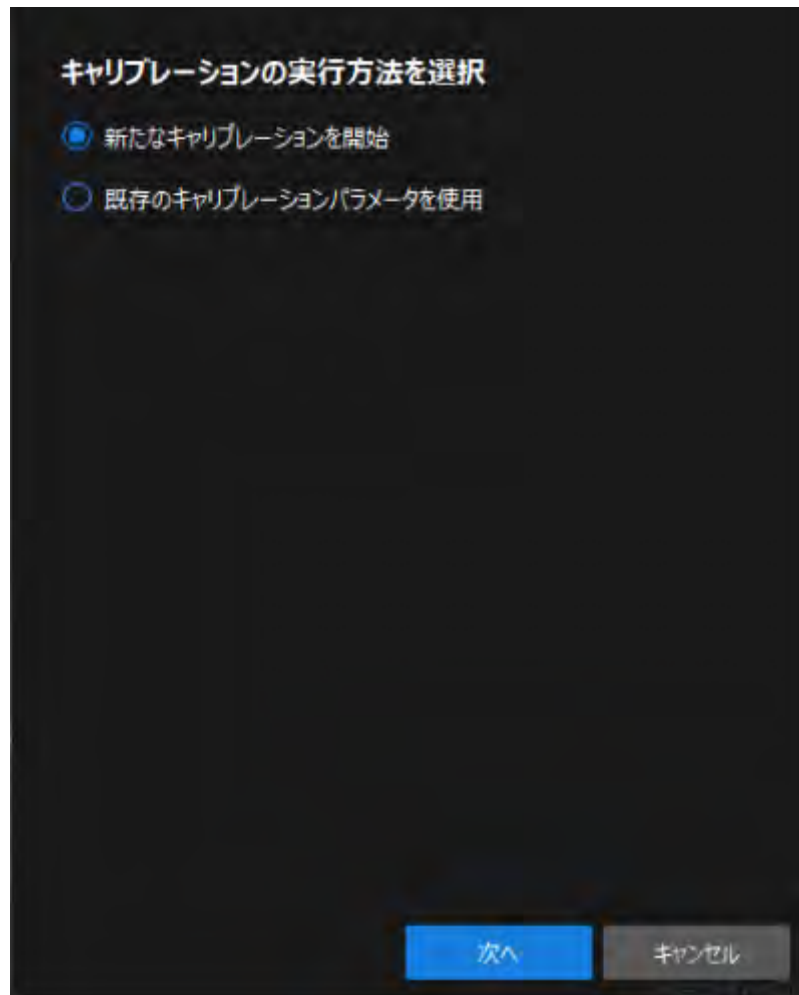
上図では、TCP Sever と Hex (リトルエンディアン) はABBロボットの**プロトコル設定**の例です。他のロボットの**プロトコル設定**は異なる場合があります。ソフトウェアは自動的にロボットの初期設定を追加しますので、初期設定のままにしておいてください。

7. ティーチペンダントで**自動キャリブレーションのプログラム**を選択し、**キャリブレーションの開始位置を定義してキャリブレーションプログラムを実行**します。様々なロボットに対応するキャリブレーション手順については、[標準インターフェース通信](#)をご参照ください。キャリブレーションプログラムが正常に起動されると、ログに「キャリブレーションプロセスに入ります。Mech-Visionでキャリブレーションを実行してください。」が表示されます。
8. Mech-Visionに戻り、**ロボットを接続**の下で「接続済み」が表示されることを確認してから**[実行]**をクリックします。すると、**キャリブレーション (Eye to Hand)** 画面が表示されます。

キャリブレーション前の設定 (Vizティーチング)

以下の手順を実行します。

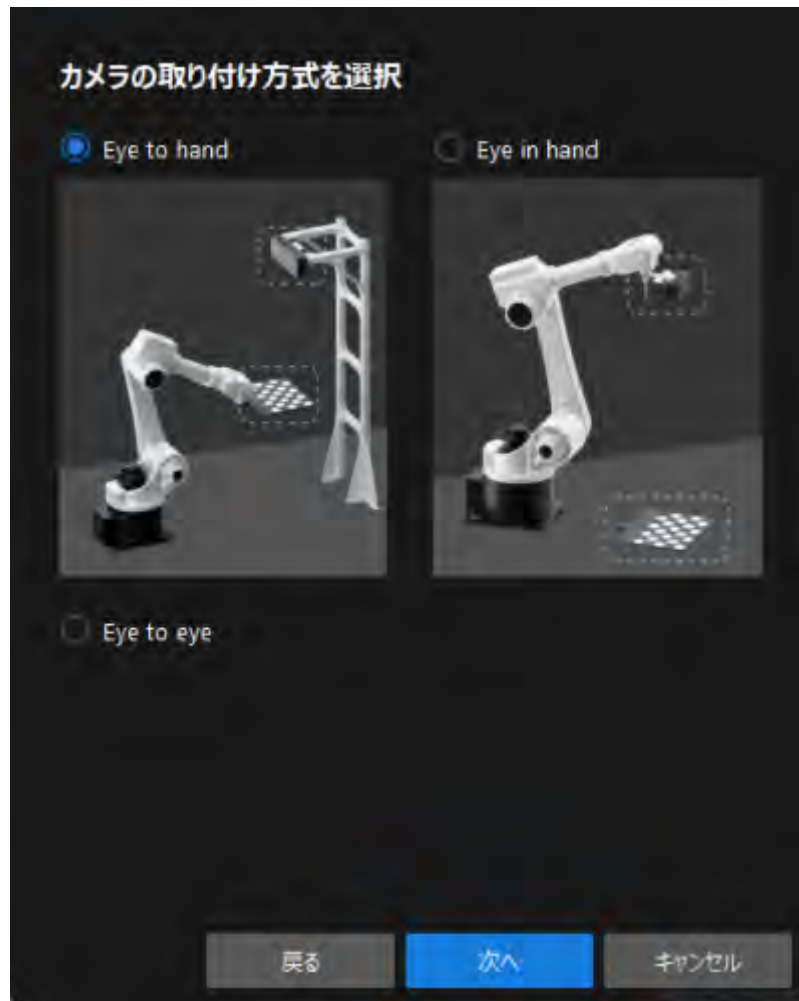
1. Mech-Visionを起動し、ツールバーで**[カメラキャリブレーション (標準モード)]**をクリックします。すると、**キャリブレーション前の設定**画面が表示されます。
2. **キャリブレーションの実行方法**を選択画面で、**新たなキャリブレーションを開始**を選択し、**[次へ]**をクリックします。



3. キャリブレーションのタスクを選択 画面で、ドロップダウンリストから **適応可能なロボットのハンド・アイ・キャリブレーション** を選択し、[**ロボット型番を選択**] をクリックして型番を選択してから、[**次へ**] をクリックします。



4. カメラの取り付け方式を選択画面で、**Eye to hand**を選択して[次へ]をクリックします。



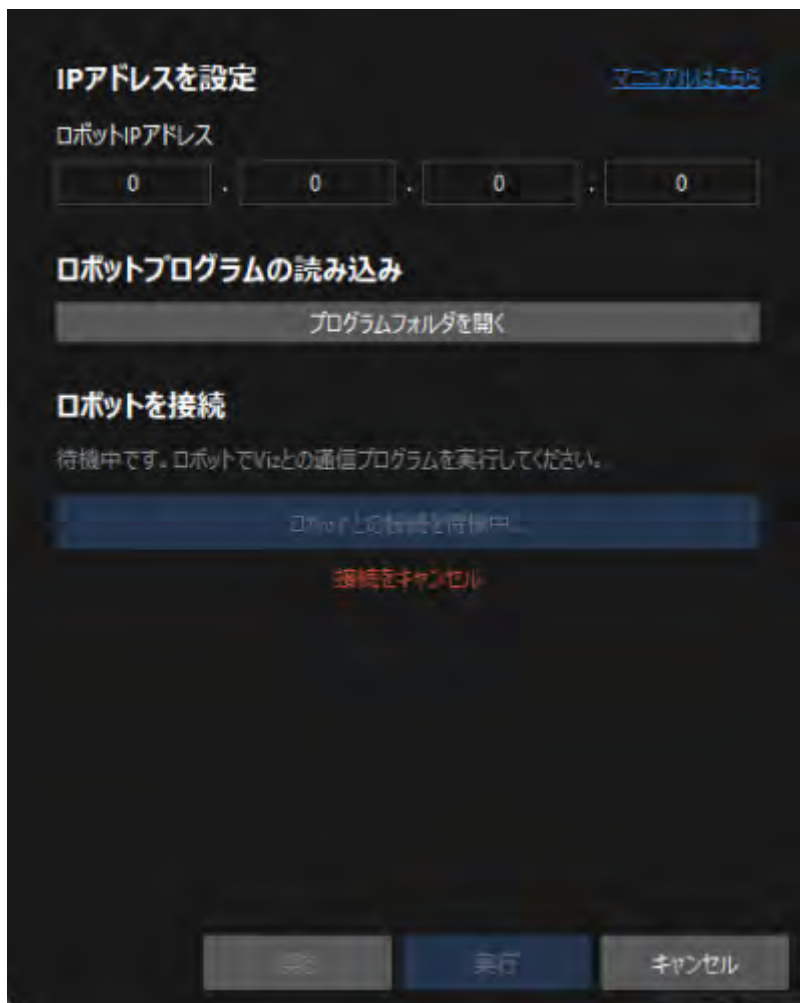
5. 実行モードとロボットの制御方式を選択画面で、自動キャリブレーションとVizティーチングを選択して[次へ]をクリックします。



6. 次の画面でロボットIPアドレスを設定します。



7. ティーチペンダントでVizとの通信プログラムを選択して実行します。様々なロボットに対応する実行方法については、 [Vizティーチング通信](#) をご参照ください。
8. Mech-Visionに戻り、**ロボットを接続**エリアの下に[**ロボットを接続**]をクリックします。このボタンが**ロボットの接続を待機中...**に変わります。




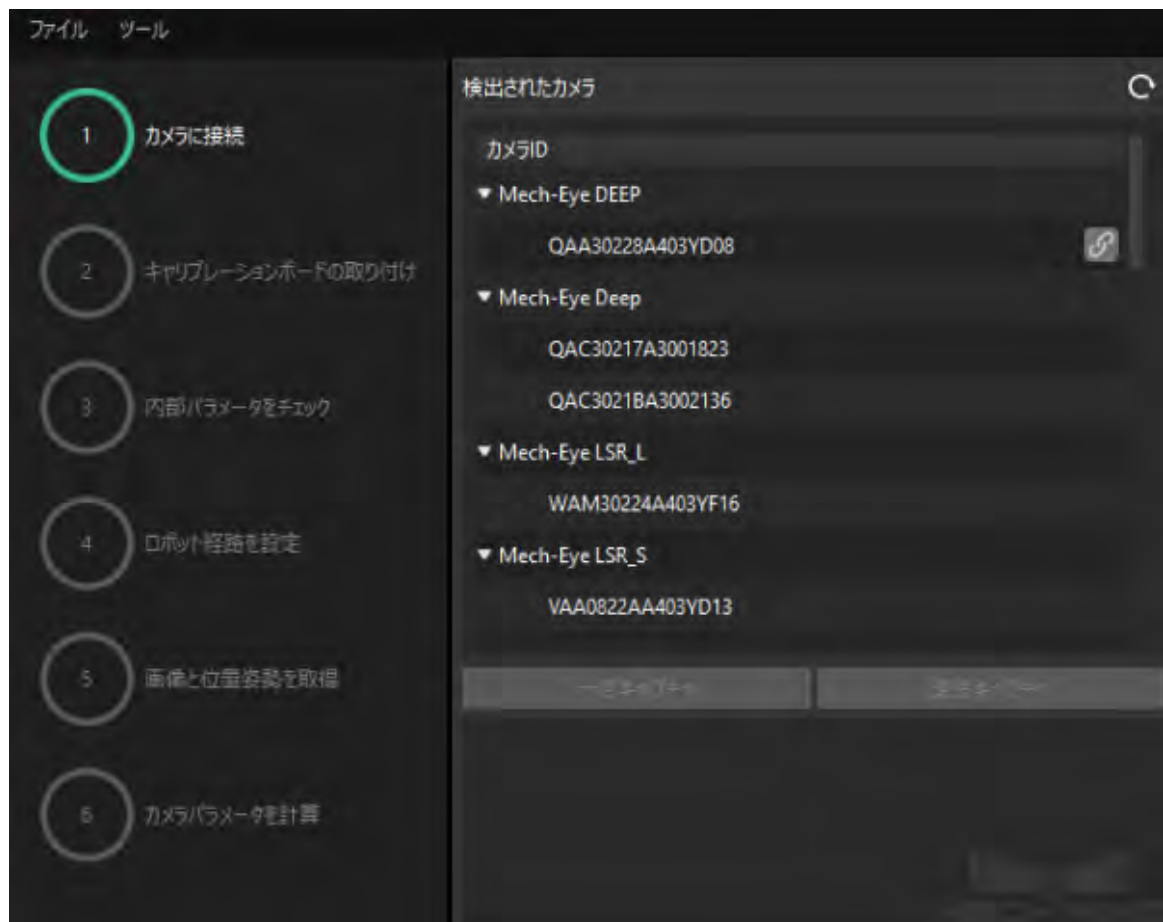
9. **ロボットを接続** の下で「接続済み」が表示されることを確認してから[**実行**]をクリックします。すると、**キャリブレーション（Eye to Hand）** 画面が表示されます。

これで、キャリブレーション前の設定が完了し、キャリブレーションに進むことになります。

キャリブレーションの実行手順

カメラの接続

1. **カメラに接続** 手順で、**検出されたカメラ**から接続するカメラを選択して  をクリックするか、ダブルクリックして接続します。



2. カメラ接続後、[一回キャプチャ]または[連続キャプチャ]をクリックします。



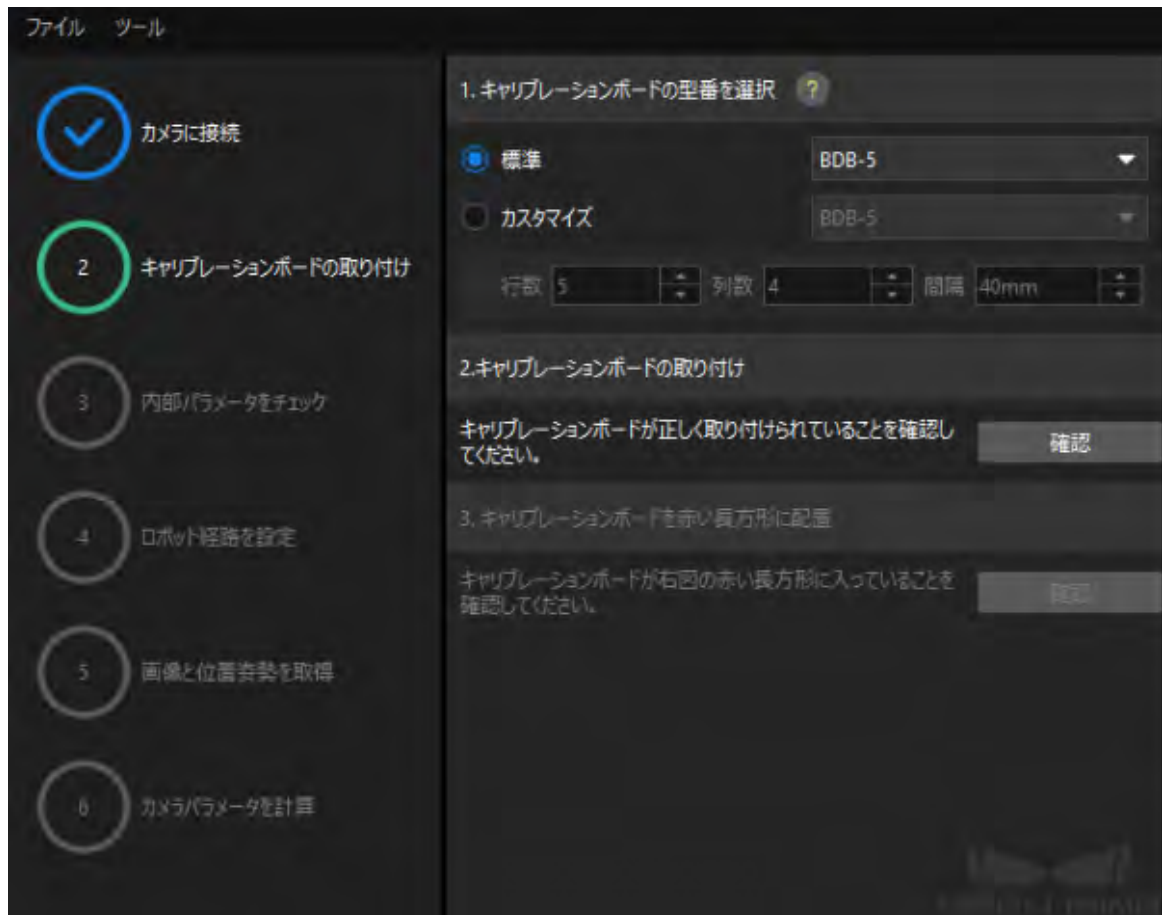
3. 画像ビューでは、取得した2D画像と深度画像が要件を満たしていることを確認してから、下部にある[次へ]をクリックします。



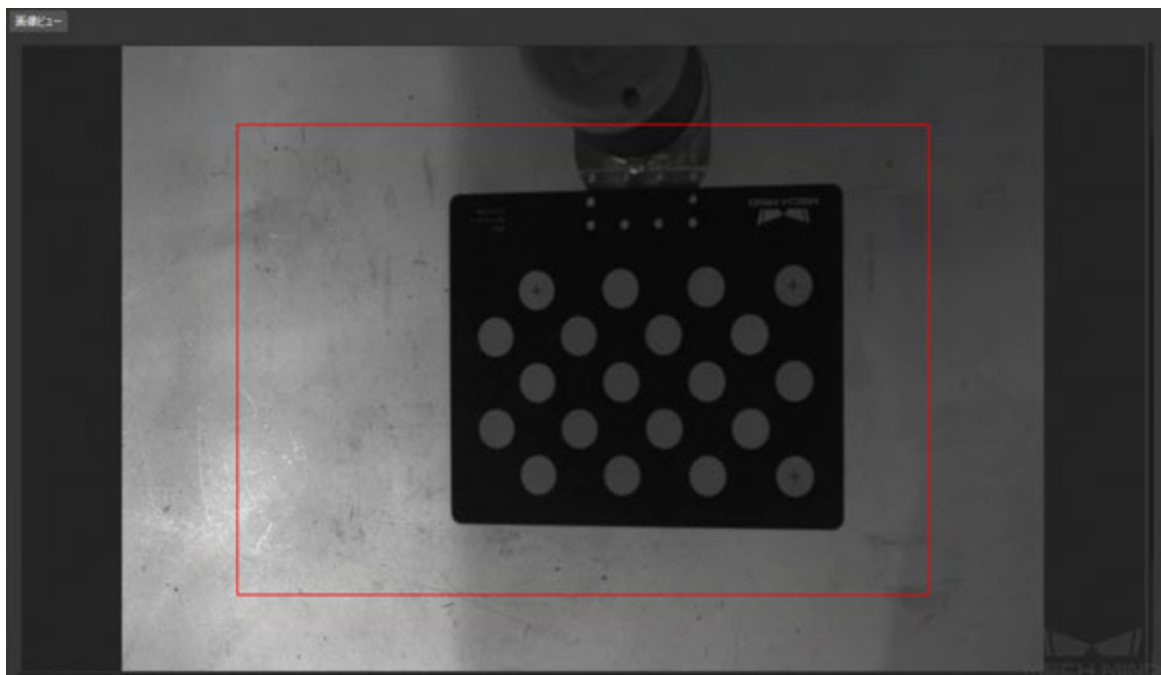
取得した画像が要件を満たしていない場合、Mech-Eye Viewerを起動し、[カメラの2Dと3D露光パラメータを調整](#)して再取得する必要があります。

キャリブレーションボードの取り付け

1. キャリブレーションボードの取り付け手順で、**1.キャリブレーションボードの型番を選択**で標準を選択し、キャリブレーションボードに貼られているラベルに従って型番を選択します。



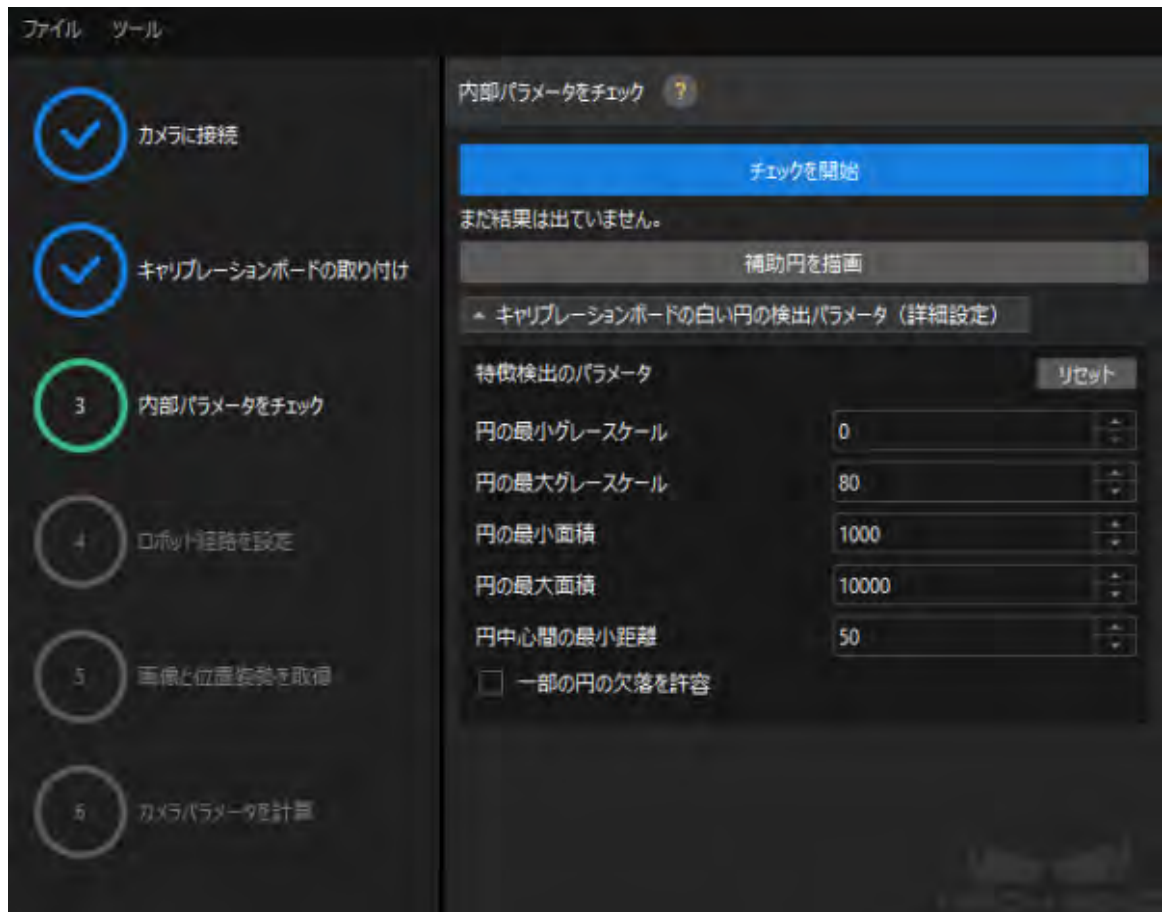
2. キャリブレーションボードが作業平面の中心に配置されていることを確認してから、**2. キャリブレーションボードの取り付け**で[確認]をクリックします。
3. キャリブレーションボードがカメラの視野中心（赤い長方形）にあることを確認してから、**3. キャリブレーションボードを赤い長方形に配置**で[確認]をクリックします。



4. キャリブレーションボードに関するすべての操作が完了したら、下部にある[次へ]をクリックします。

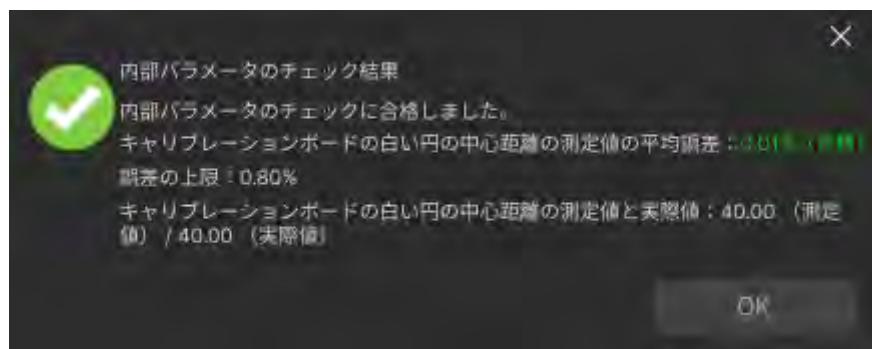
カメラの内部パラメータをチェック

1. 内部パラメータをチェック手順で[チェックを開始]をクリックします。



2. 内部パラメータのチェック結果を確認します。

- カメラの内部パラメータが合格した場合、[OK]をクリックして[次へ]をクリックします。

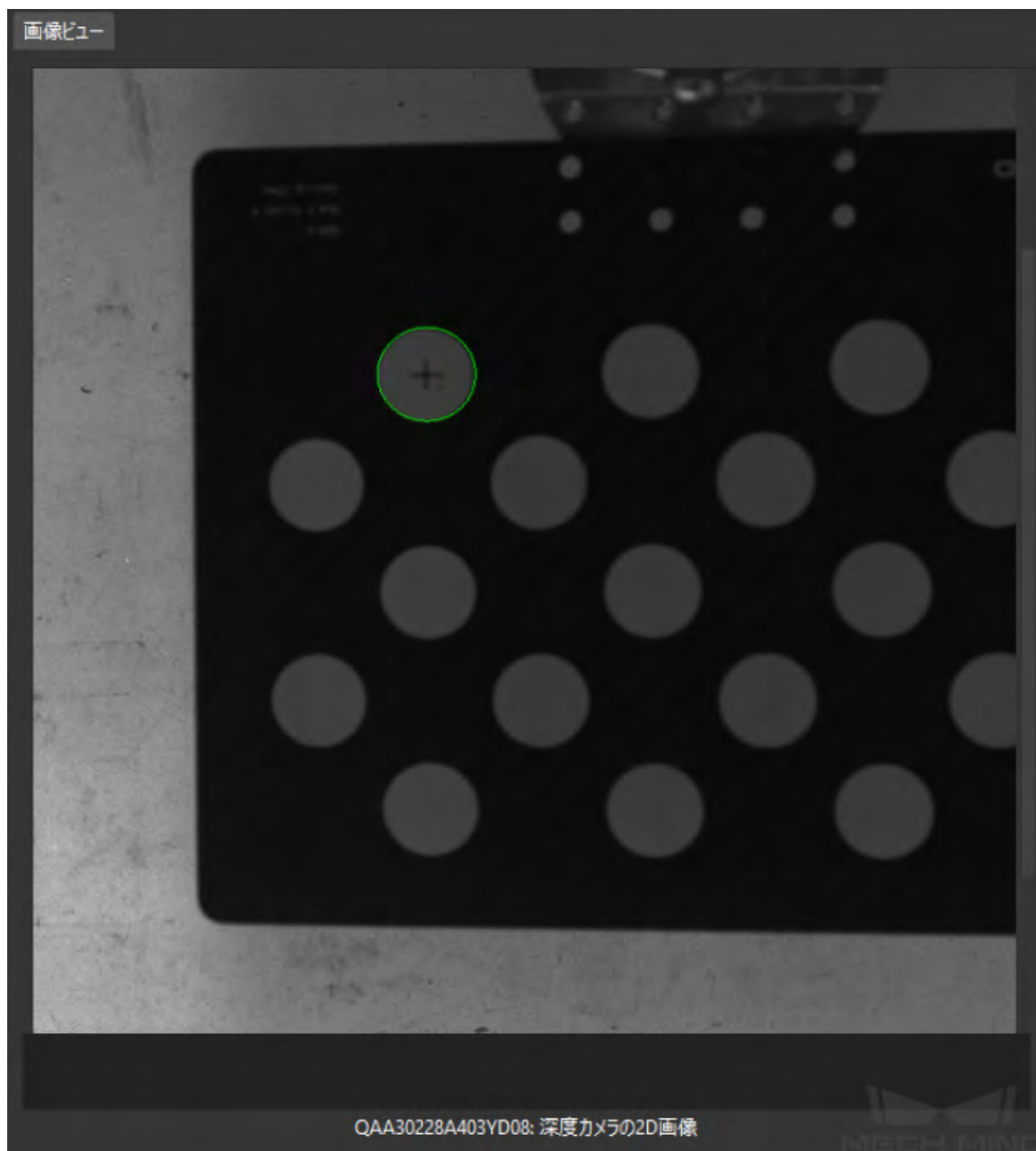


- カメラの内部パラメータが合格しなかった場合、補助円を描画するか、キャリブレーションボードの白い円の検出パラメータを手動で編集してから、[再度チェック]をクリック

クします。

補助円を描画

1. 補助円を描画する場合、[補助円を描画]をクリックします。
2. 右側の画像ビューパネルで、キャリブレーションボードの画像を右クリックし、**ウィンドウに合わせる**のチェックを外します。その後、**Ctrl**キーを押しながらスクロールホイールをドラッグして画像のサイズを適切な大きさに変更します。
3. キャリブレーションボードの白い円の十字型の中心にマウスポインタを合わせて、マウスの左ボタンを押し、補助円をキャリブレーションボードの白い円に完全に含ませてから離します。



4. [再度チェック]をクリックし、カメラ内部パラメータのチェックが合格したことを確認します。

検出パラメータを手動で調整

検出パラメータを手動で調整する場合、[キャリブレーションボードの白い円の検出パラメータ（詳細設定）] をクリックし、実際の状況に応じてパラメータ値を変更します。

それでもキャリブレーションボードの白い円が検出されない場合、現場の環境に応じてカメラの関連パラメータを調整する必要があります。カメラパラメータの調整については、 [パラメータ調整](#) をご参照ください。

ロボット経路を設定

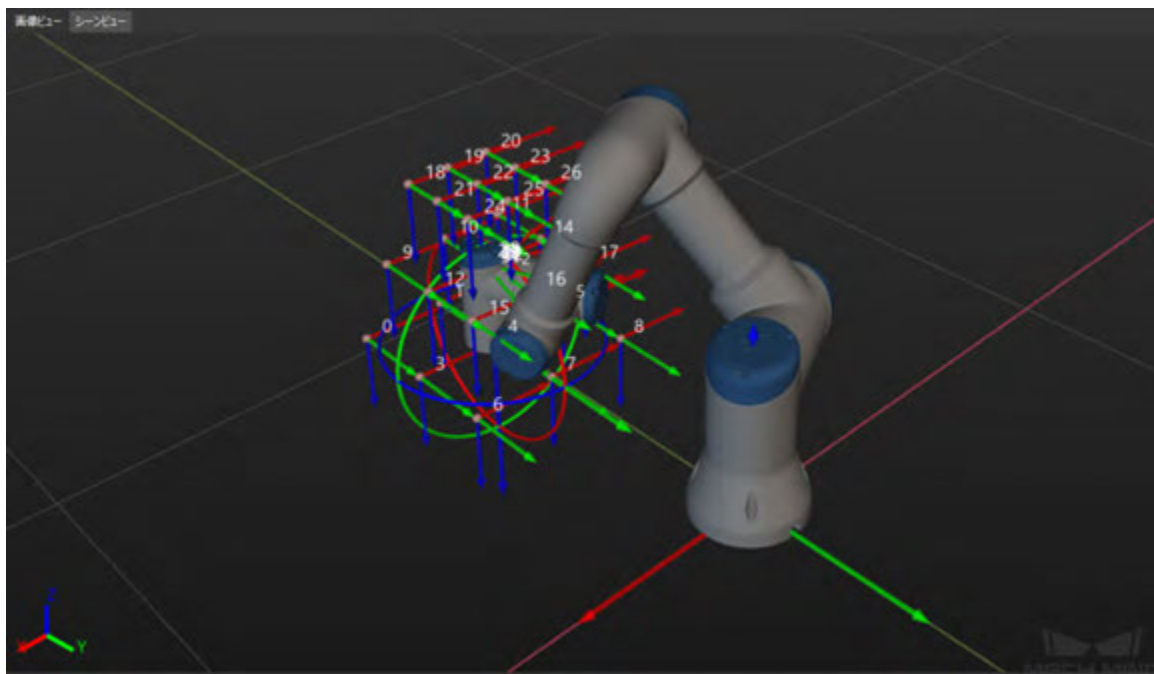
1. ロボット経路を設定手順で、高さ範囲を設定します。



高さ範囲は、カメラの推奨稼働範囲とロボットの動作領域に応じて設定する必要があります。

2. カメラとワールド座標系のZ軸が平行でない場合、[経路を自動的に合わせる] をクリックします。

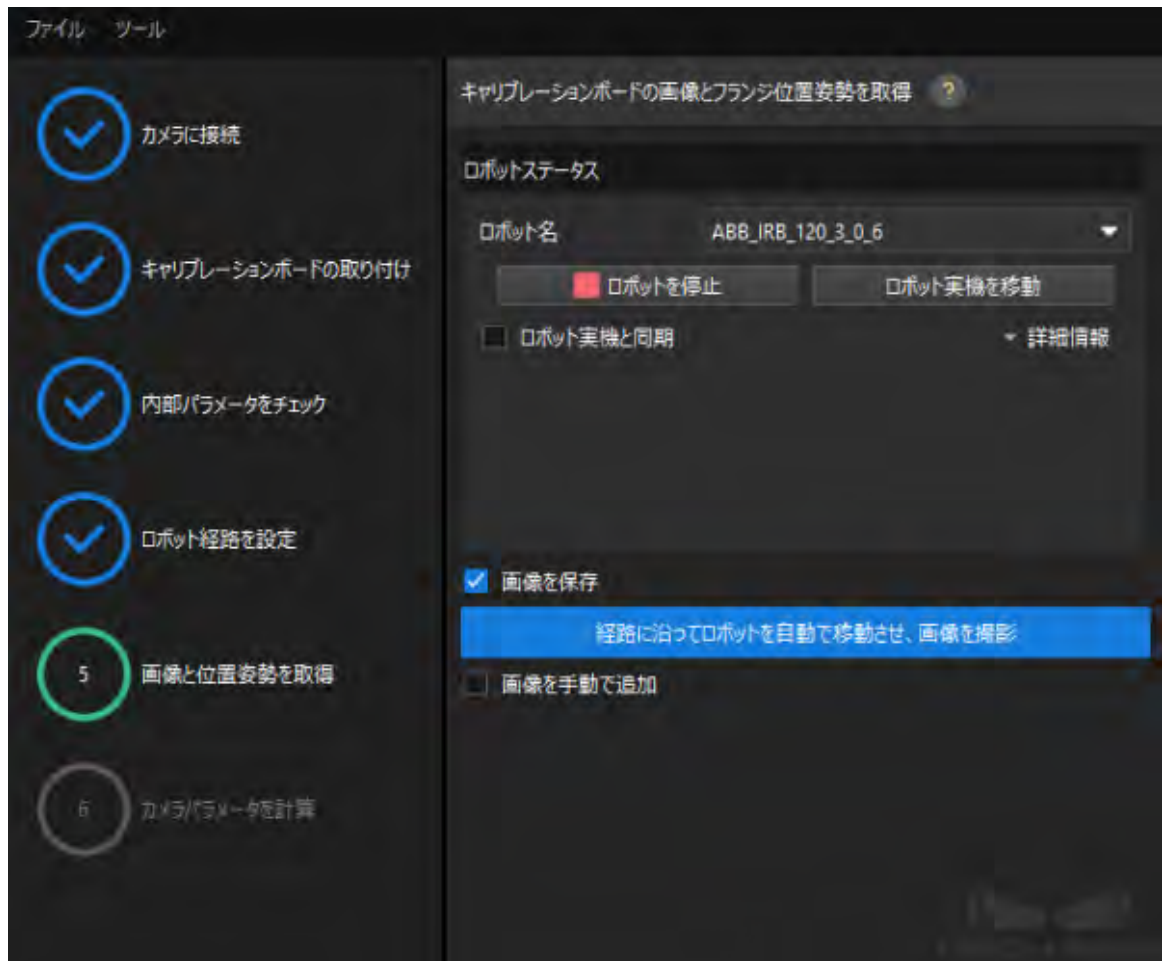
3. 実際の状況に応じて、**経路のタイプ**をToHandに指定し、ピラミッド型の経路パラメータ下の高さ範囲、層数、最下層サイズX/Y、最上層サイズX/Yと層ごとの移動グリッドの列と行数を設定し、**回転角度**を設定してから、[確認]をクリックします。
4. 右側のシーンビューのパネルで、各経路点の位置が正確で、周囲と衝突していないことを確認します。



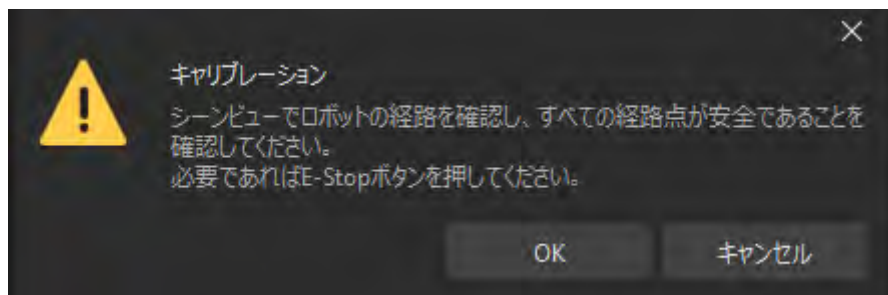
5. 確認後、下部にある[次へ]をクリックします。

画像と位置姿勢を取得

1. **画像と位置姿勢を取得**手順で、**画像を保存**にチェックを入れ、[経路に沿ってロボットを自動で移動させ、画像を撮影]をクリックします。



2. ロボット動作の安全に関する説明をよく読んでから、[OK]をクリックします。

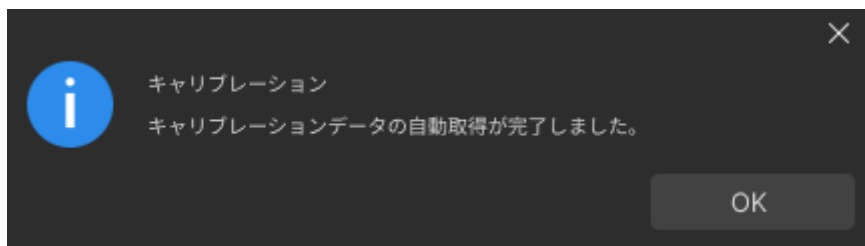


3. ロボットが設定された経路を移動し、カメラが各経路点で画像の取得を完了するのを待ちます。右側の画像ビューパネルで取得された画像が表示されます。



- ° ロボット動作中に、ロボットの動作範囲に近づかないようにしてください。
- ° 「ロボットを停止」ボタンをクリックするとキャリブレーションを終了しますが、ロボットはすぐには停止せず、現在の経路点に達するまで停止します。緊急時にロボットティーチペンダントの非常停止ボタンを押してください。非常停止を押した後、ロボットに再接続する必要があります。

4. 画像が取得した後、表示される画面で[OK]をクリックします。



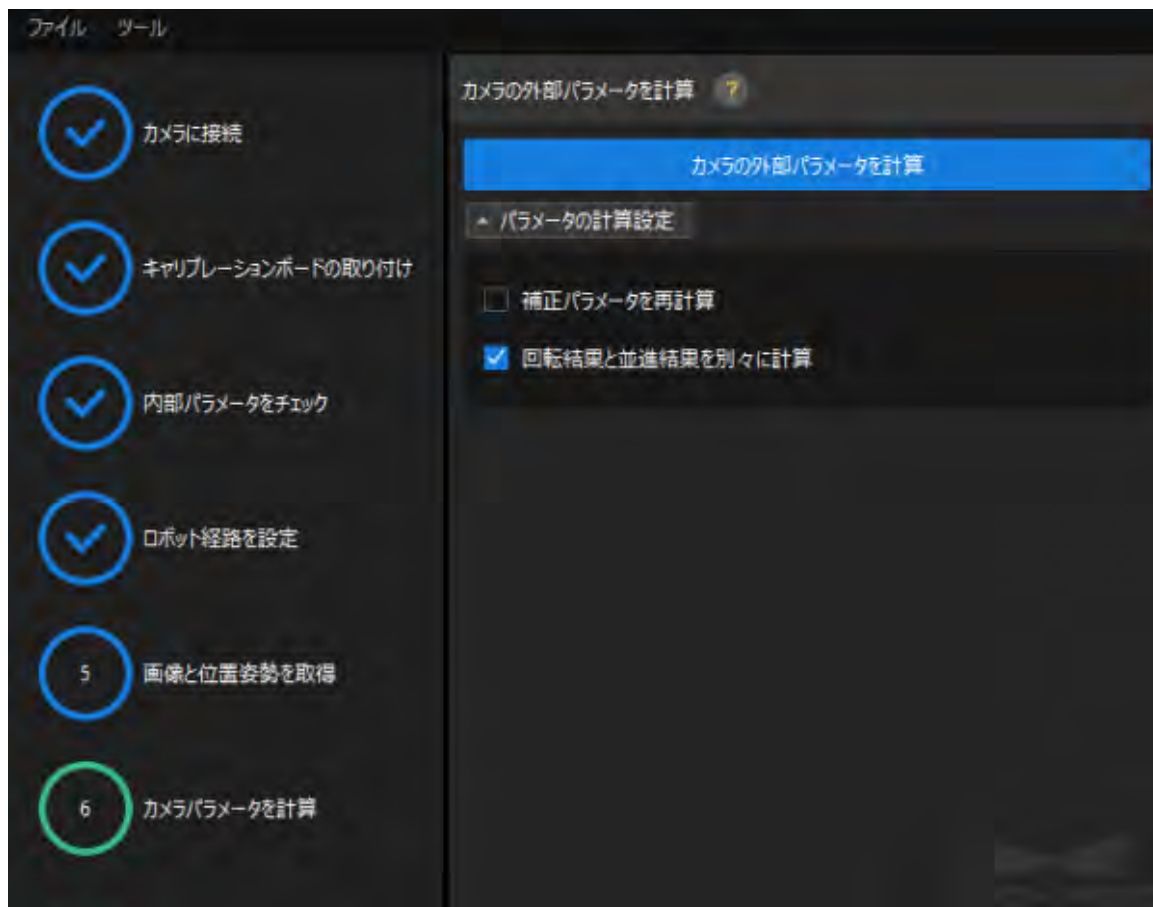
5. 認識したキャリブレーションポイントが要件を満たしていることを確認してから、下部にある[次へ]をクリックします。



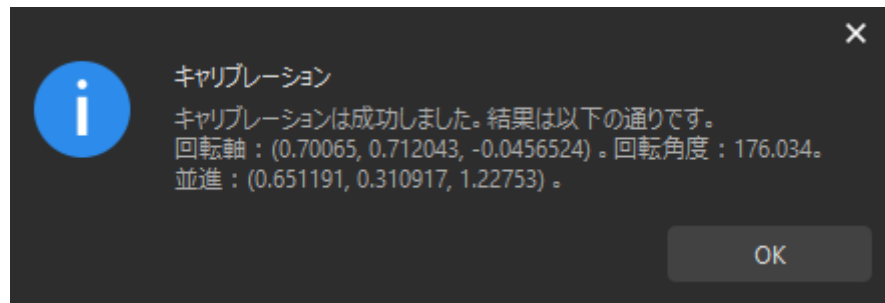
要件を満たしていない場合、ロボットを手動で移動した後（ティーチペンダントでもMech-Vizでも選択可能）、[画像を手動で追加]にチェックを入れ、[画像を追加してフランジ位置姿勢を記録]をクリックし、キャリブレーションボードの画像を追加してロボットのフランジ位置姿勢を入力する必要があります。

カメラパラメータを計算

1. カメラパラメータを計算手順で、「パラメータの計算設定」を展開し、**回転結果と並進結果**を別々に計算にチェックを入れてから、[カメラの外部パラメータを計算]をクリックします。



2. キャリブレーションが正常に実行されたことを示すダイアログで、**OK**をクリックします。



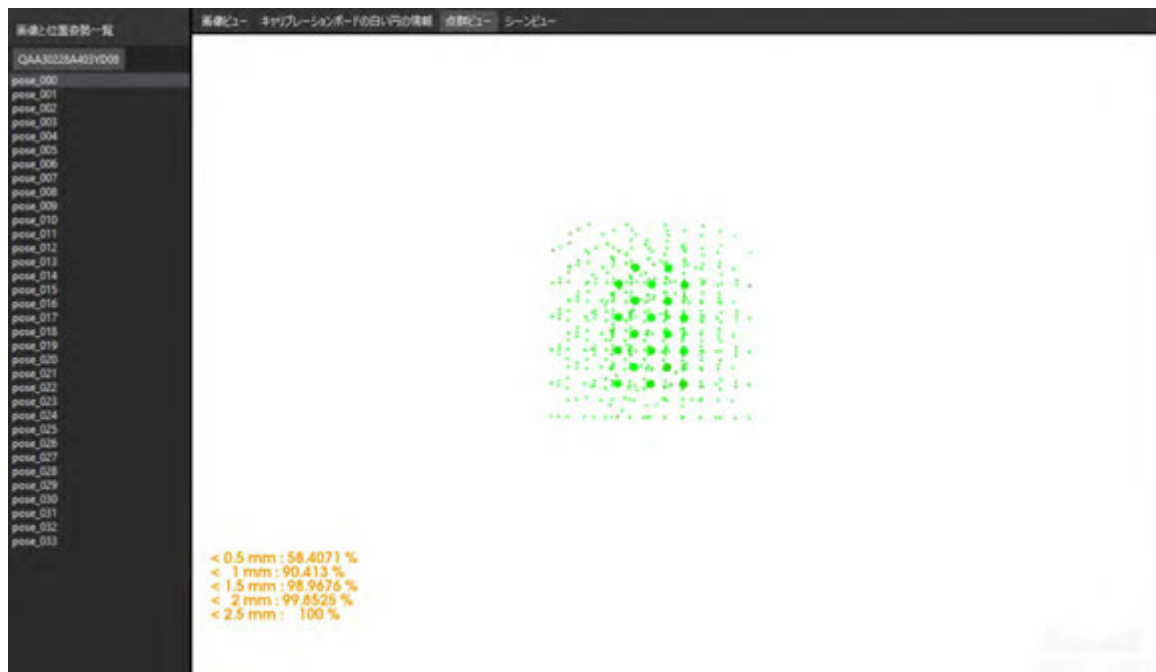
1. 右側の **点群ビュー** パネルでキャリブレーションの誤差点群が表示されます。



誤差点群は、各キャリブレーション位置姿勢におけるキャリブレーションボードの白い円の実測の値と計算値との偏差を示すために使用されます。

2. キャリブレーション精度が要件を満たしていることを確認します。

キャリブレーション精度は、100%を占める誤差の値を求めることで、おおよそ判断することができます。例えば、下図は2.5 mm以下の精度を示しています。



キャリブレーション精度を向上させるには、[キャリブレーション結果の分析](#)をご参照ください。

キャリブレーション結果の確認と保存

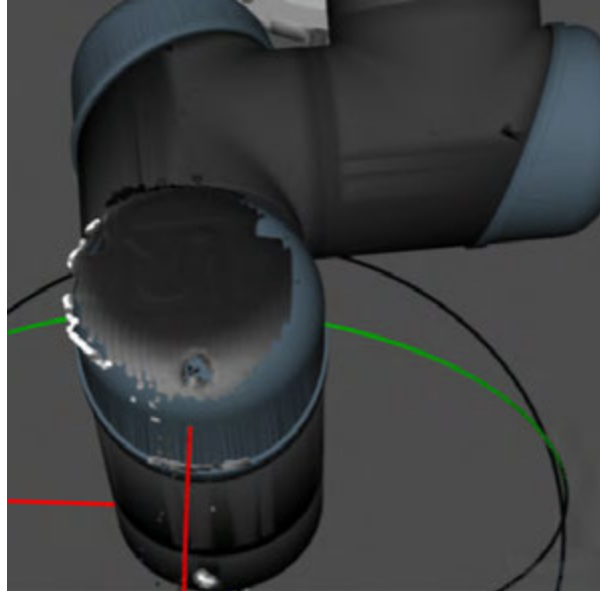
カメラの外部パラメータを計算した後、シーンビューでロボット点群とロボットモデルの点群との重なりを確認してキャリブレーション結果を大まかに判断します。具体的な流れは以下の通りです。

1. ロボットアームをカメラの視野範囲内に移動させます。
2. カメラパラメータを計算手順で[外部パラメータを再計算]をクリックします。クリックする

と、カメラが撮影を行います。

3. シーンビューに切り替え、ロボットの点群とロボットモデルの重なりを確認します。

ロボットの点群がロボットモデルとほぼ一致すれば、キャリブレーション結果が使用可能です。



1. ただし、ロボットモデルはロボット実機の外観と完全に一致しない場合があります、外部パラメータの微調整の基準とすることができないことに注意してください。
2. シーンビューに加え、Mech-Vizでロボットの点群とロボットモデルとの重なりを確認できます。

4. 下部にある[保存]をクリックし、ポップアップする **キャリブレーションファイルを保存** 画面で[OK]をクリックします。すると、カメラのキャリブレーション結果は、プロジェクトの「calibration」ディレクトリに自動的に保存されます。

これで、キャリブレーション手順が完了しました。

4.4.3. 自動キャリブレーション (Eye in Hand)

本節では、カメラがEye in Hand (EIH) 方式で取り付けられている場合に自動キャリブレーションを実行する方法について説明します。

事前準備

以下は、ロボットハンド・アイ・キャリブレーションを実行する前に必要な事前準備です。

- Mech-Mindビジョンシステムを構築します。
- キャリブレーションに必要なものを準備します。
- キャリブレーションボードの点群画像生成を調整します。
- ロボットと通信の設定を完了します。

ビジョンシステムを構築

Mech-Mindビジョンシステムの構築については、「ビジョンシステムを構築」節をご参照ください。

ハンド・アイ・キャリブレーションには、Mech-Eye Viewer、Mech-Center、Mech-VisionおよびMech-Vizが必要です。これらのソフトウェアがインストールされ、最新バージョンにアップグレードされていることを確認してください。

キャリブレーションに必要なものを準備

カメラがEye in Hand方式で取り付けられている場合、自動キャリブレーションを実行するにはキャリブレーションボードの使用が必要です。

キャリブレーションボードについては、以下の要件があります。

- キャリブレーションボードの白い円がはっきりと見えること、破損や曲げ変形がないことを確認してください。
- キャリブレーションボードを作業平面の中心に配置してください。

また、キャリブレーションを実行する前に、ロボットをキャリブレーションの開始位置に移動させてください。Eye in Handでは、キャリブレーションの開始位置はカメラの稼働位置となります。ロボットは、カメラを移動させて下から上にキャリブレーションを実行します。

キャリブレーションボードの点群画像生成を調整

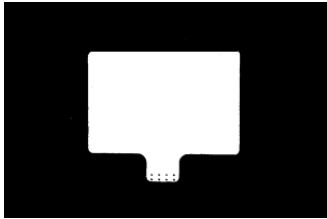
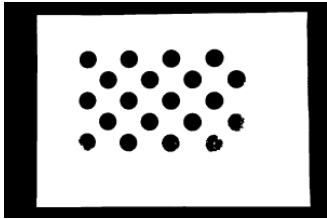
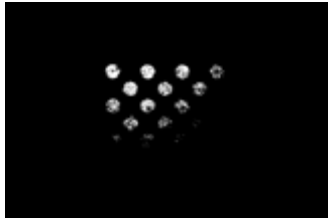
1. Mech-Eye Viewerソフトウェアを起動し、カメラのパラメータを調整します。
2. 2Dパラメータを調整して、2D画像にキャリブレーションボードがはっきり見られ、露光過度や露光不足などがないようにします。
3. 3Dパラメータを調整して、キャリブレーションボードの白い円の点群が完全に見られるようにします。点群の変動範囲を小さくするために、**点群後処理**の中の**点群平滑化**と**ノイズ除去**を**Normal**に設定することを推奨します。



現場の環境光が複雑な場合、2D画像や点群への影響を軽減するために、遮光・補光することを推奨します。

4. 上記の手順を完了することで、キャリブレーションボードの点群の画像品質が要件を満たしていることを確認します。

	正常	露光過度	露光不足
2D画像			

	正常	露光過度	露光不足
点群			

ロボット通信設定

ロボット側は標準インターフェース方式でビジョン側と通信する場合、ロボット側で標準インターフェースの通信設定を行ってください。詳細については、[標準インターフェース通信](#)をご参照ください。

ロボット側はVizティーチング方式でビジョン側と通信する場合、ロボット側でVizティーチングの通信設定を行ってください。詳細については、[Vizティーチング通信](#)をご参照ください。

キャリブレーション前の設定

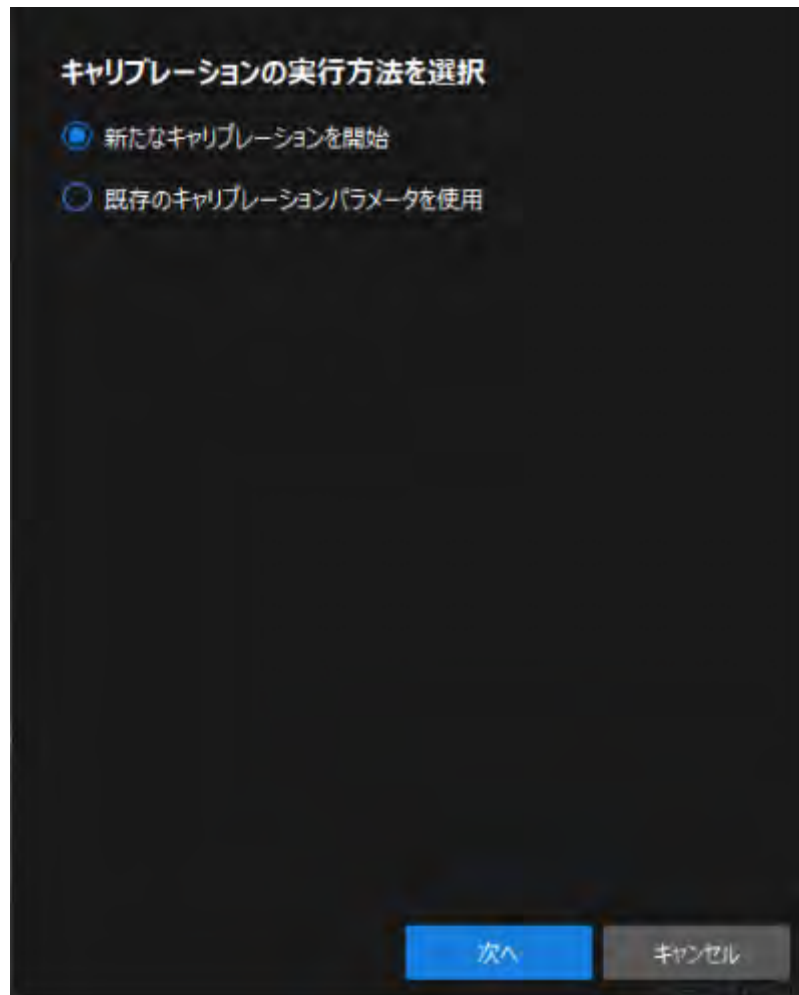
ロボット側は標準インターフェース方式でビジョン側と通信する場合、[キャリブレーション前の設定（標準インターフェース）](#)を行ってください。

ロボット側はVizティーチング方式でビジョン側と通信する場合、[キャリブレーション前の設定（Vizティーチング）](#)を行ってください。

キャリブレーション前の設定（標準インターフェース）

以下の手順を実行します。

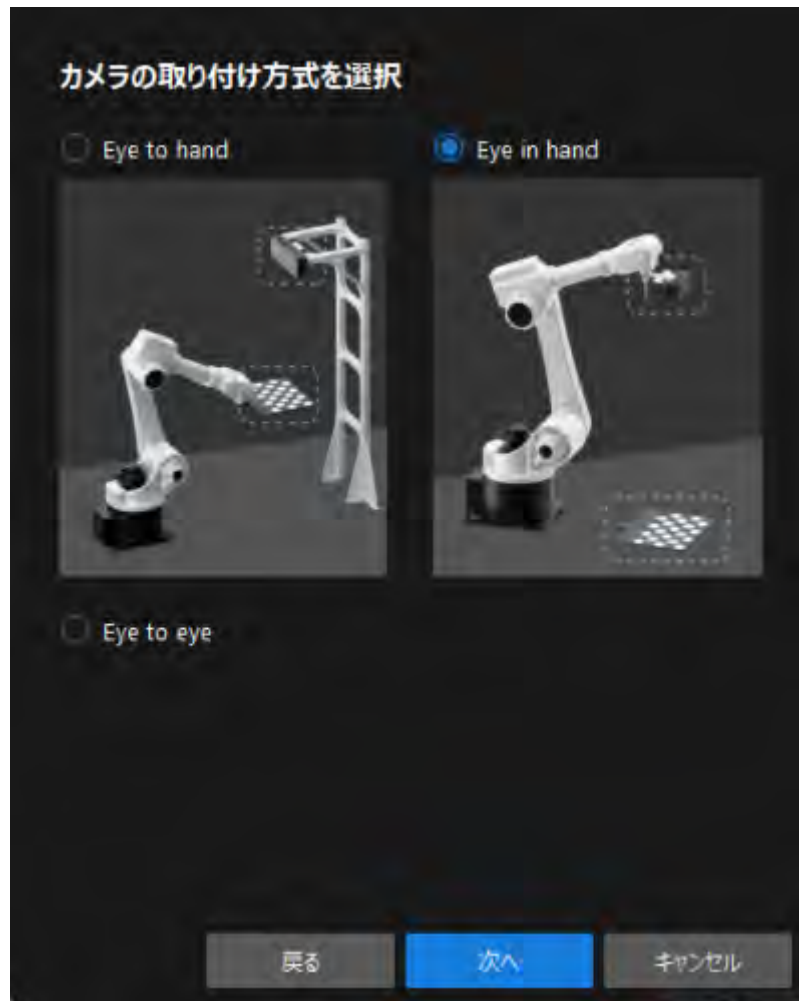
1. Mech-Visionを起動し、ツールバーで[**カメラキャリブレーション（標準モード）**]をクリックします。すると、**キャリブレーション前の設定** 画面が表示されます。
2. **キャリブレーションの実行方法を選択** 画面で、**新たなキャリブレーションを開始** を選択し、[**次へ**]をクリックします。



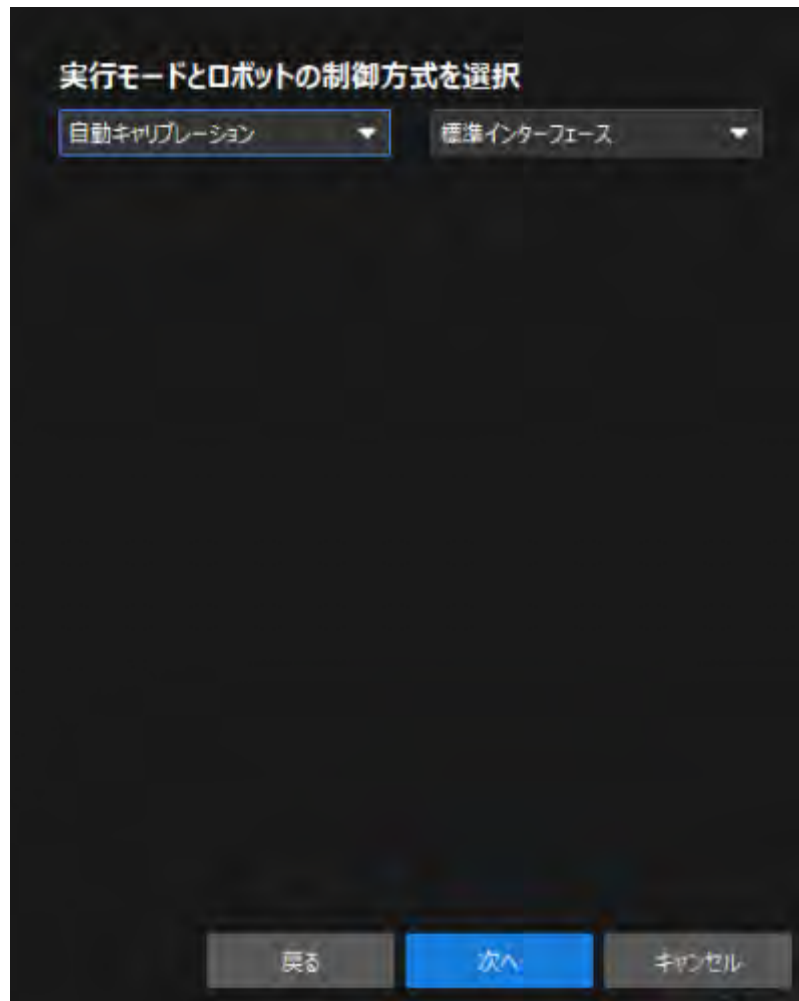
3. キャリブレーションのタスクを選択 画面で、ドロップダウンリストから **適応可能なロボットのハンド・アイ・キャリブレーション** を選択し、[**ロボット型番を選択**] をクリックして型番を選択してから、[**次へ**] をクリックします。



4. カメラ取り付け方式を選択 画面で、**Eye in hand** にチェックを入れ、[次へ]をクリックします。



5. 実行モードとロボットの制御方式を選択 画面で、自動キャリブレーションと標準インターフェースを選択してから、[次へ]をクリックします。



6. 通信設定画面で、実際の状況に応じて通信プロトコルとホストIPアドレスを設定してから [インターフェースサービスを起動] をクリックします。このボタンがロボットの接続を待機中…に変わります。



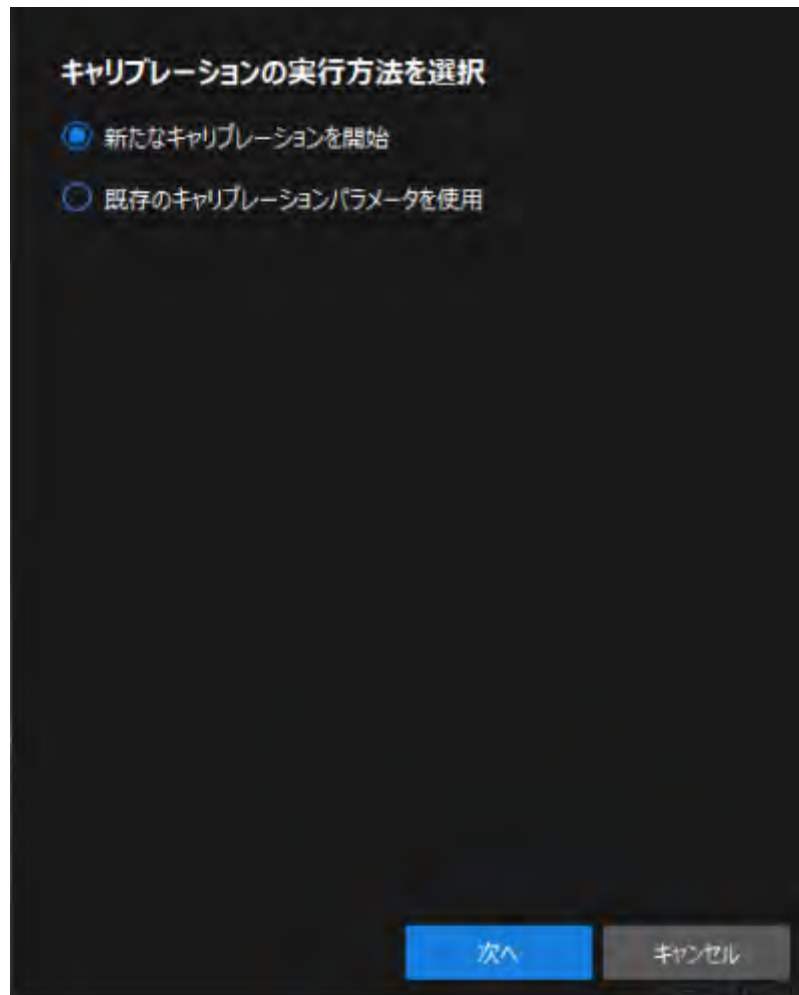
上図では、TCP Sever と Hex（リトルエンディアン）はABBロボットの**プロトコル**設定の例です。他のロボットの**プロトコル**設定は異なります。ソフトウェアは自動的にロボットの初期設定を追加しますので、初期設定のままにしておいてください。

7. ティーチペンダントで**自動キャリブレーションのプログラム**を選択し、**キャリブレーションの開始位置を定義してキャリブレーションプログラムを実行**します。様々なロボットに対応するキャリブレーション手順については、**標準インターフェース通信**をご参照ください。キャリブレーションプログラムが正常に起動されると、ログに「キャリブレーションプロセスに入ります。Mech-Visionでキャリブレーションを実行してください。」が表示されます。
8. Mech-Visionに戻り、**ロボットを接続**の下で「接続済み」が表示されることを確認してから**[実行]**をクリックします。すると、**キャリブレーション（Eye to Hand）**画面が表示されます。

キャリブレーション前の設定（Vizティーチング）

以下の手順を実行します。

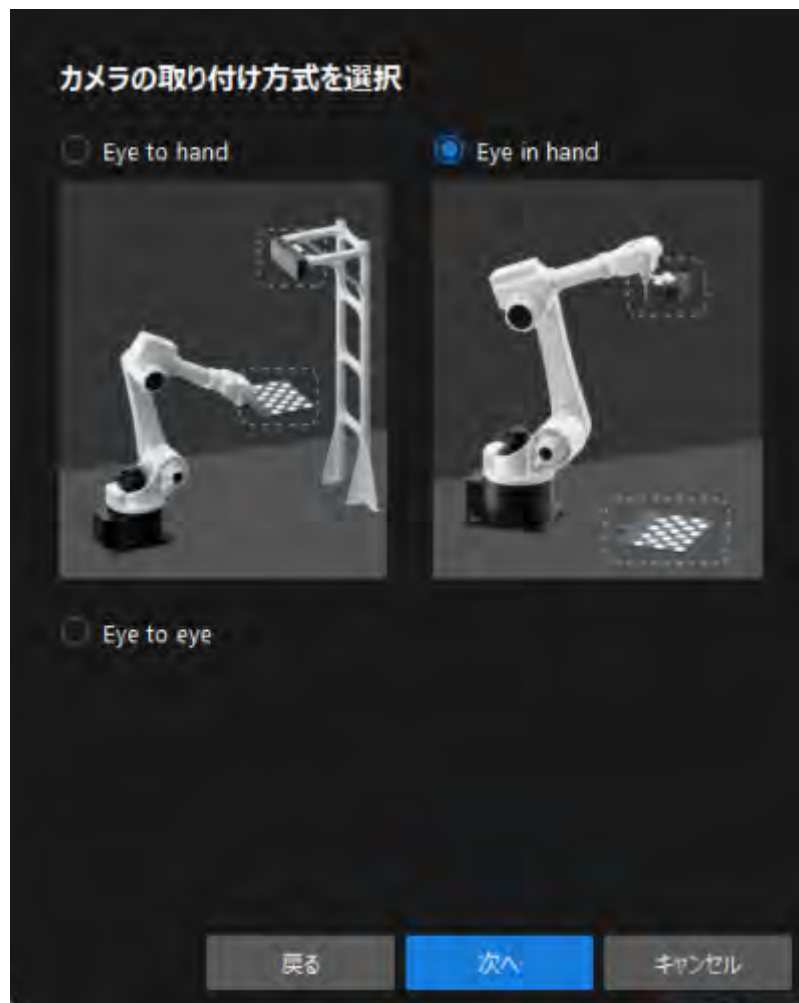
1. Mech-Visionを起動し、ツールバーで**[カメラキャリブレーション（標準モード）]**をクリックします。すると、**キャリブレーション前の設定**画面が表示されます。
2. **キャリブレーションの実行方法を選択**画面で、**新たなキャリブレーションを開始**を選択し、**[次へ]**をクリックします。



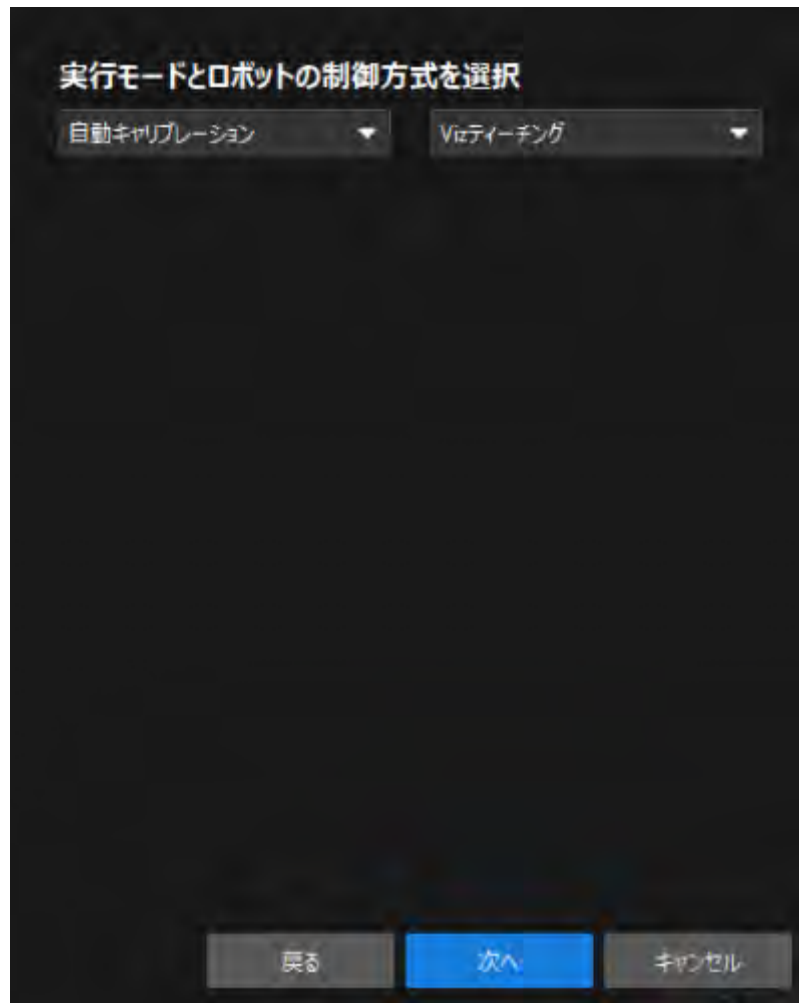
3. キャリブレーションのタスクを選択 画面で、ドロップダウンリストから **適応可能なロボットのハンド・アイ・キャリブレーション** を選択し、**[ロボット型番を選択]** をクリックして型番を選択してから、**[次へ]** をクリックします。



4. カメラ取り付け方式を選択 画面で、**Eye in hand** にチェックを入れ、[次へ]をクリックします。



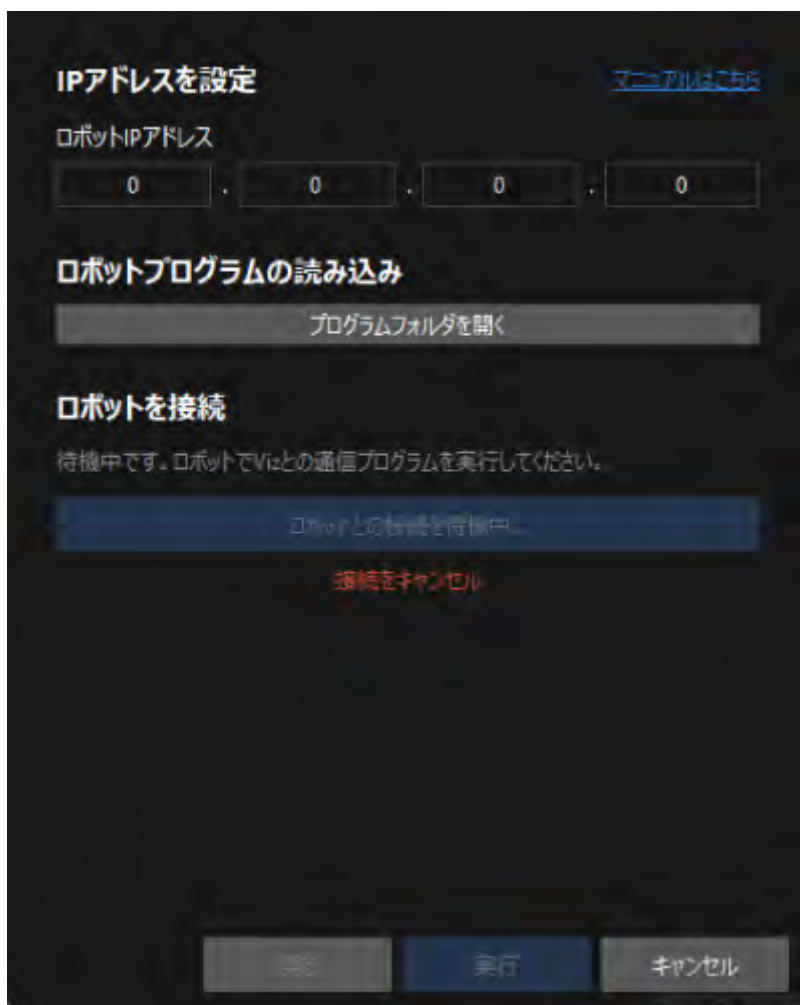
5. 実行モードとロボットの制御方式を選択 画面で、自動キャリブレーションと Vizティーチング を選択して [次へ] をクリックします。



6. 次の画面で **ロボットIPアドレス** を設定します。



7. ティーチペンダントで **Vizとの通信プログラム**を選択して実行します。様々なロボットに対応する実行方法については、 [Vizティーチング通信](#) をご参照ください。
8. Mech-Visionに戻り、**ロボットを接続**エリアの下に[**ロボットを接続**]をクリックします。このボタンが**ロボットの接続を待機中...**に変わります。




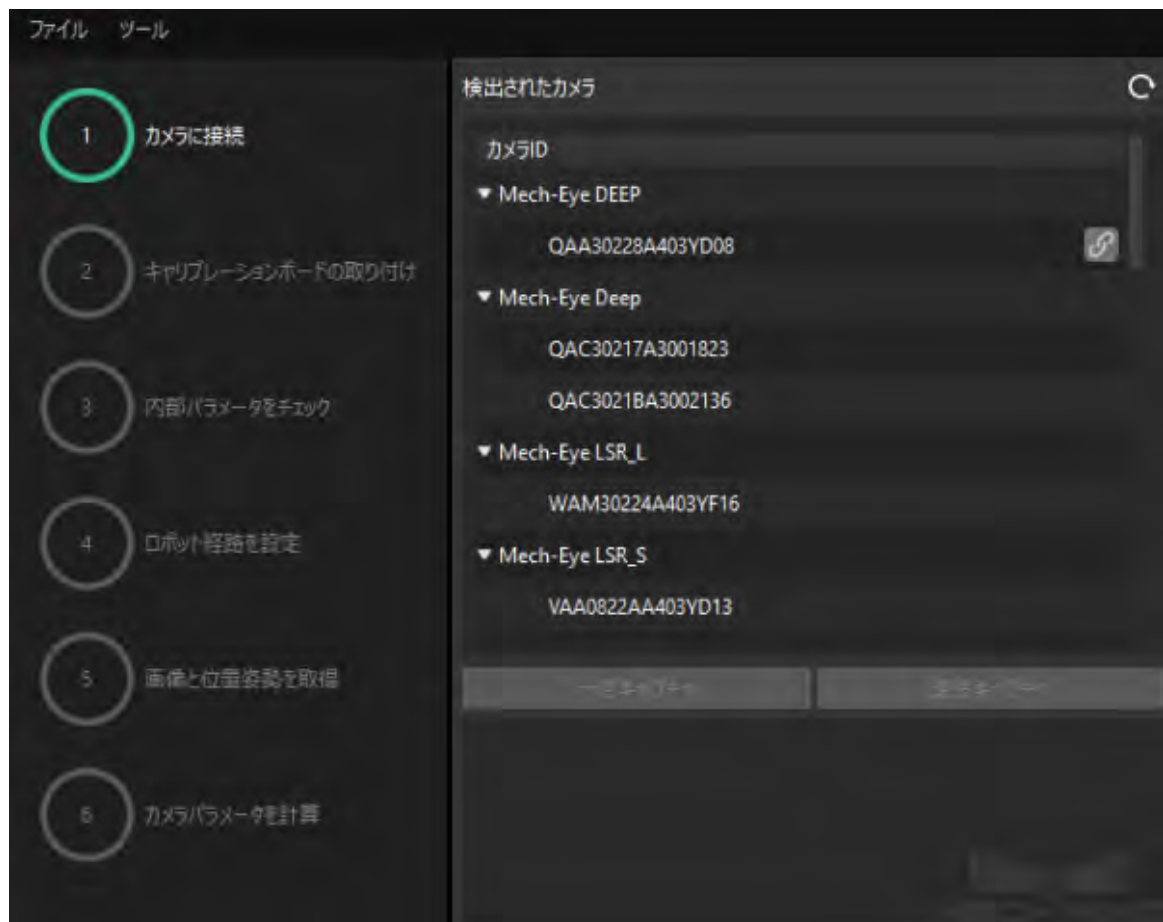
9. **ロボットを接続** の下で「接続済み」が表示されることを確認してから[**実行**]をクリックします。すると、**キャリブレーション（Eye to Hand）** 画面が表示されます。

これで、キャリブレーション前の設定が完了し、キャリブレーションに進むことになります。

キャリブレーションの実行手順

カメラ接続

1. **カメラに接続** 手順で、**検出されたカメラ** から接続するカメラを選択して  をクリックするか、ダブルクリックして接続します。



2. カメラ接続後、[一回キャプチャ]または[連続キャプチャ]をクリックします。



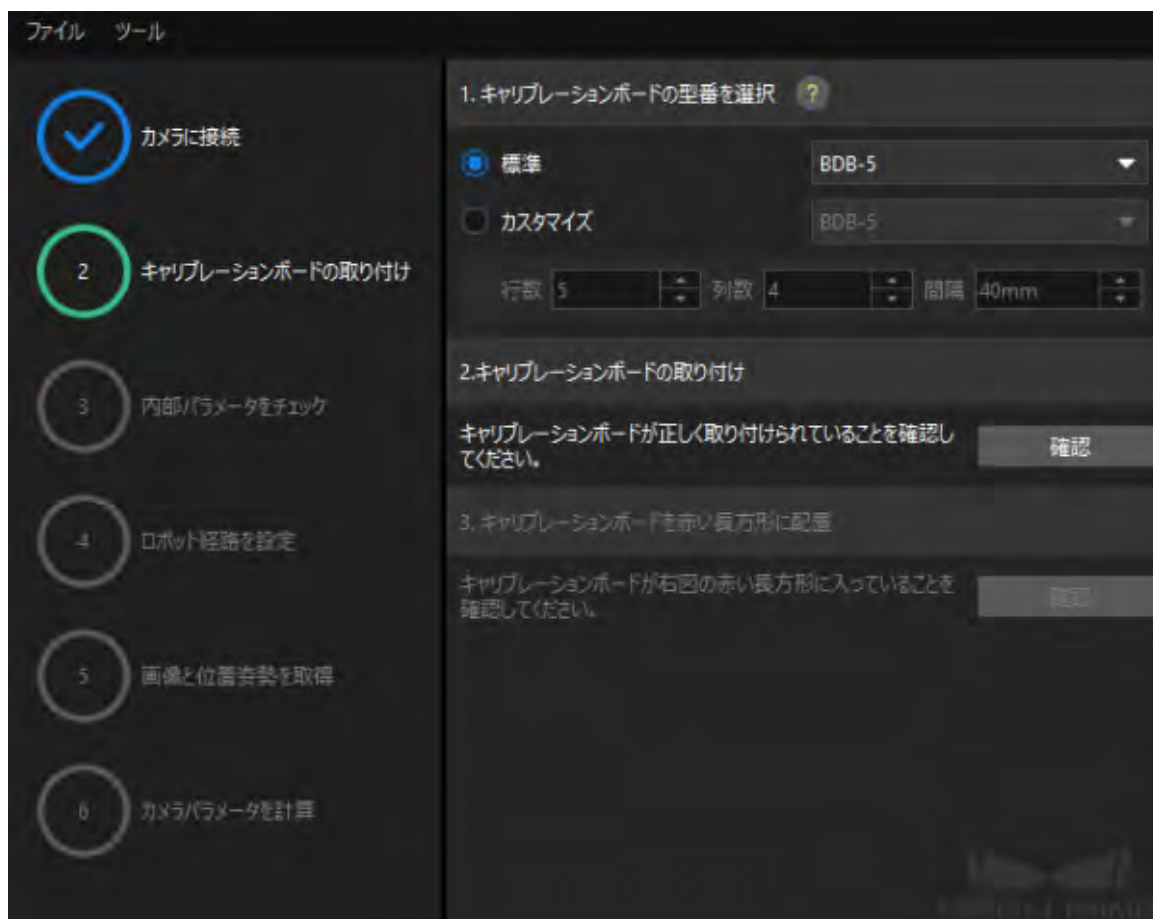
3. 画像ビューでは、取得した2D画像と深度画像が要件を満たしていることを確認してから、下部にある[次へ]をクリックします。



取得した画像が要件を満たしていない場合、Mech-Eye Viewerを起動し、[カメラの2Dと3D露光パラメータを調整](#)して再取得する必要があります。

キャリブレーションボードの取り付け

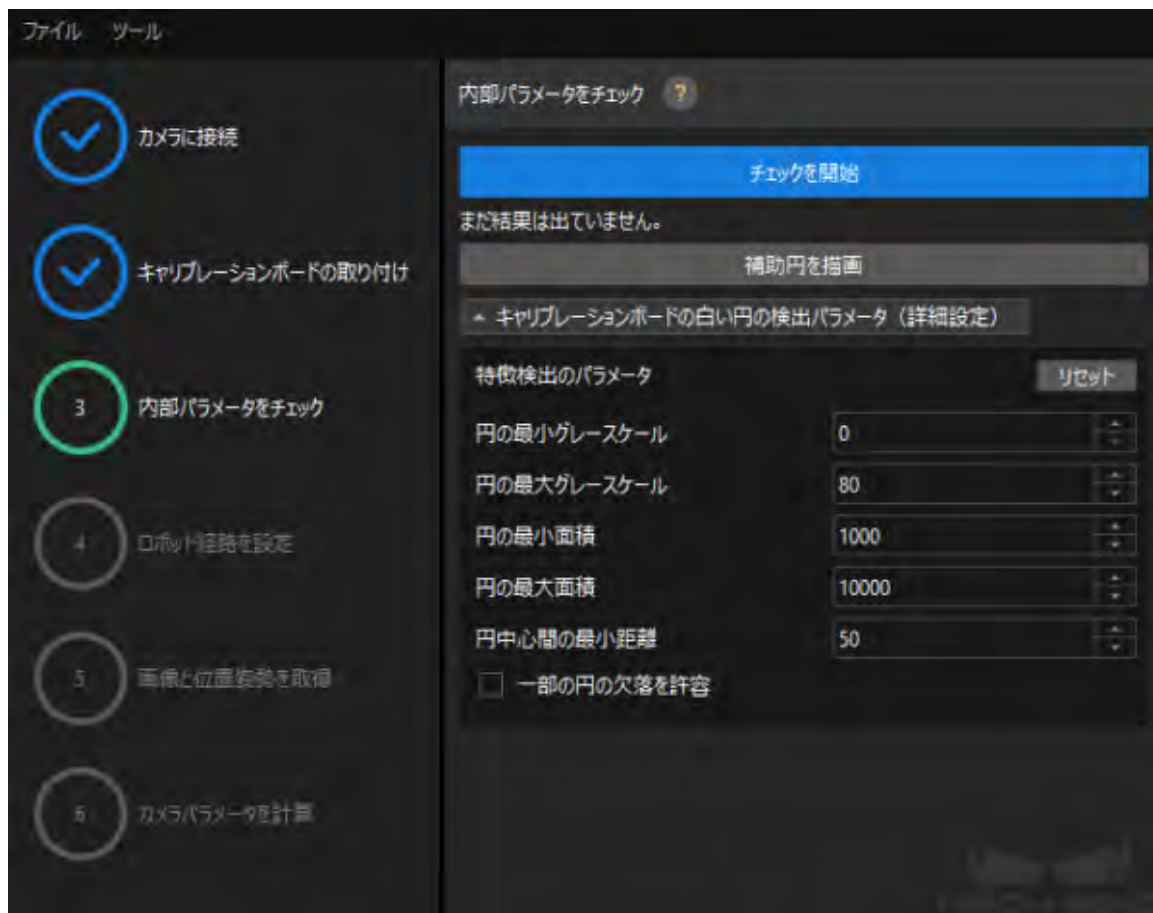
1. キャリブレーションボードの取り付け 手順で、**1. キャリブレーションボードの型番を選択**で **標準** を選択し、キャリブレーションボードに貼られているラベルに従って型番を選択します。



2. キャリブレーションボードが作業平面の中心に配置されていることを確認してから、**2. キャリブレーションボードの取り付け**で[確認]をクリックします。
3. キャリブレーションボードがカメラの視野中心（赤い長方形）にあることを確認してから、**3. キャリブレーションボードを赤い長方形に配置**で[確認]をクリックします。
4. キャリブレーションボードに関するすべての操作が完了したら、下部にある[次へ]をクリックします。

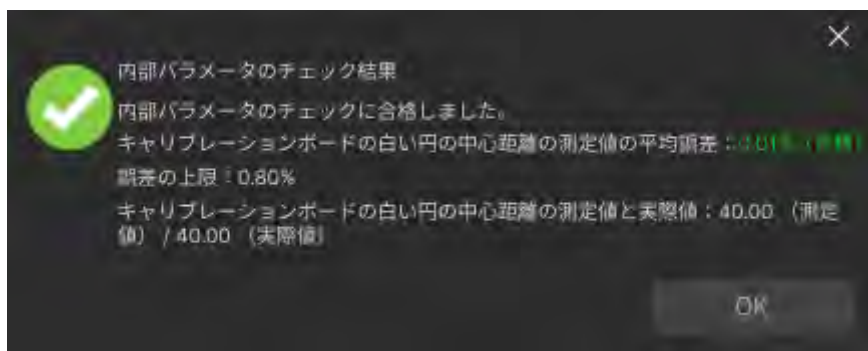
カメラの内部パラメータをチェック

1. **内部パラメータをチェック**手順で[**チェックを開始**]をクリックします。



2. 内部パラメータのチェック結果を確認します。

- カメラの内部パラメータが合格した場合、[OK]をクリックして[次へ]をクリックします。



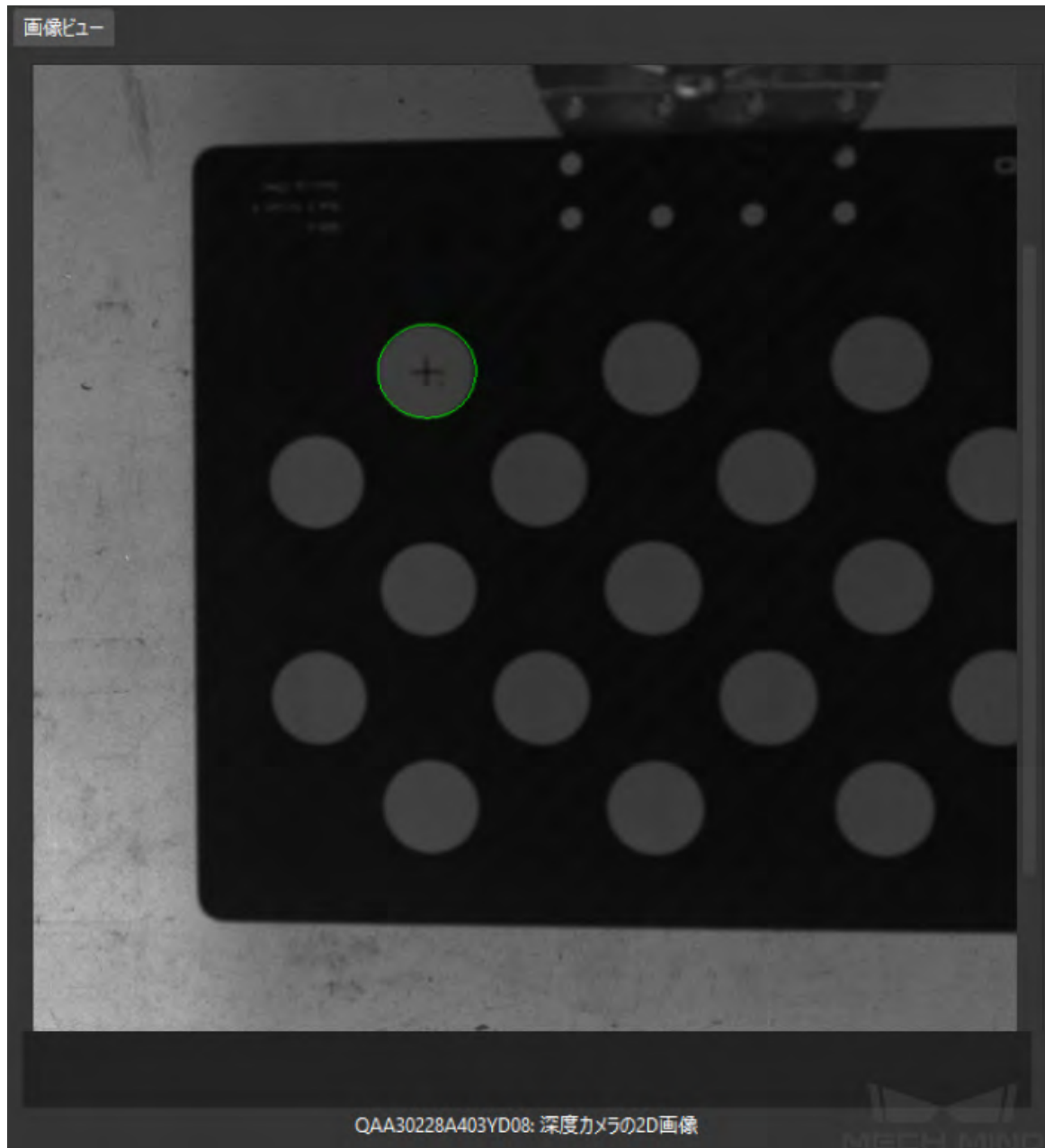
- カメラの内部パラメータが合格しなかった場合、補助円を描画するか、キャリブレーションボードの白い円の検出パラメータを手動で編集してから、[再度チェック]をクリックします。

補助円を描画

- 補助円を描画する場合、[補助円を描画]をクリックします。
- 右側の画像ビューパネルで、キャリブレーションボードの画像を右クリックし、**ウィンドウに合わせる**のチェックを外します。その後、**Ctrl** キーを押しながらスクロールホイールをド

ラッグして画像のサイズを適切な大きさに変更します。

3. キャリブレーションボードの白い円の十字型の中心にマウスポインタを合わせて、マウスの左ボタンを押し、補助円をキャリブレーションボードの白い円に完全に含ませてから離します。



4. **[再度チェック]**をクリックし、カメラ内部パラメータのチェックが合格したことを確認します。

検出パラメータを手動で調整

検出パラメータを手動で調整する場合、**[キャリブレーションボードの白い円の検出パラメータ (詳細設定)]**をクリックし、実際の状況に応じてパラメータ値を変更します。

それでもキャリブレーションボードの白い円が検出されない場合、現場の環境に応じてカメラ

の関連パラメータを調整する必要があります。カメラパラメータの調整については、[パラメータ調整](#)をご参照ください。

ロボット経路を設定

1. **ロボット経路を設定**手順で、**高さ範囲**を設定します。

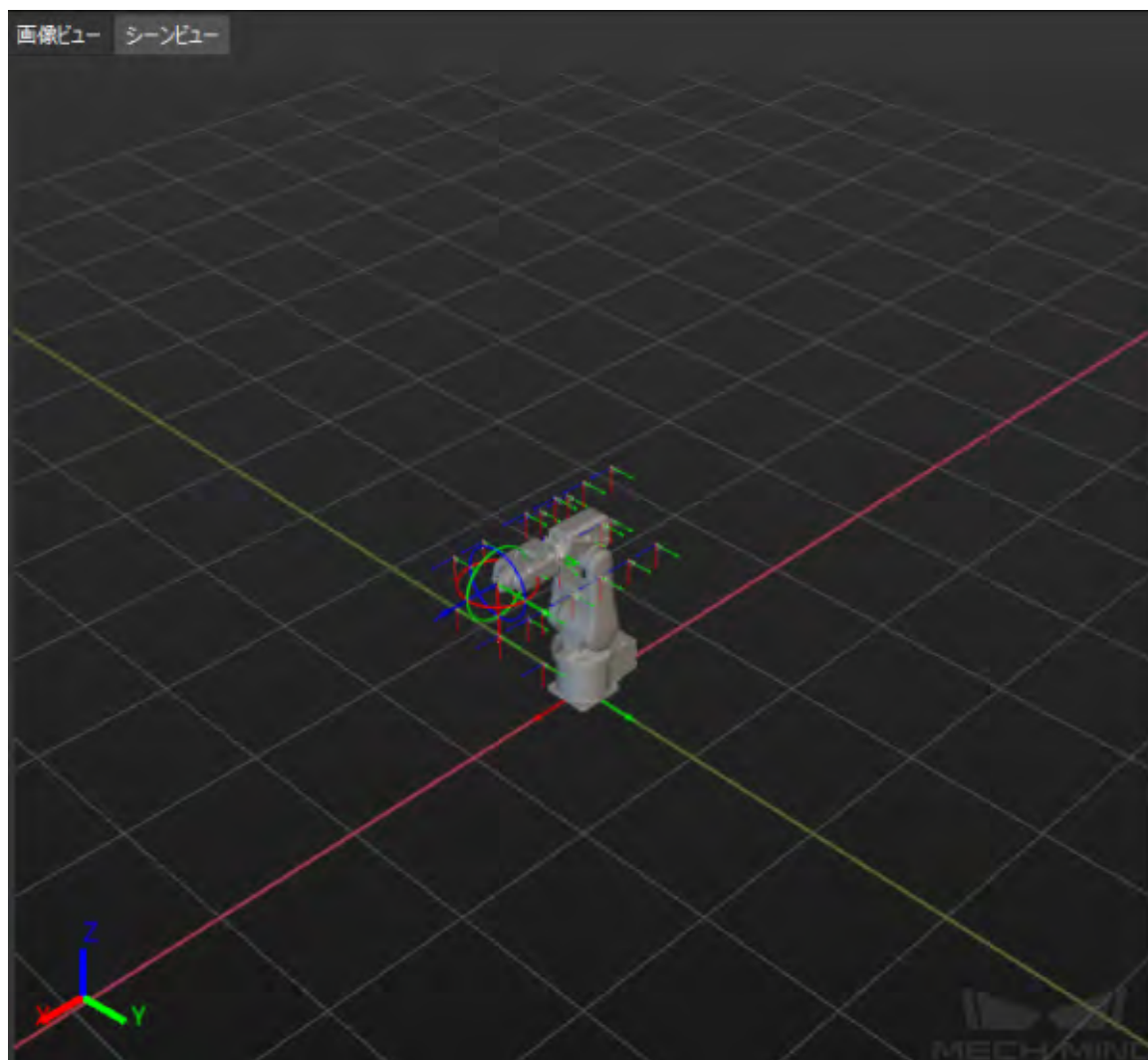


高さ範囲は、カメラの推奨稼働範囲とロボットの動作領域に応じて設定する必要があります。

2. カメラとロボットのフランジ座標系のZ軸が平行でない場合、[**経路を自動的に合わせる**]をクリックします。
3. 実際の状況に応じて**経路のタイプ**を設定する必要があります。ここで**InHand**に設定します。ピラミッド型の経路パラメータで**高さ範囲**、**層数**、**最下層サイズ**、**最上層サイズ**と**層ごとの移動グリッドの列と行数**を設定し、衛星パラメータで**ロボットの移動回数**、**中心からキ**

ャリブレーションボードまでの高さと半径を設定してから、[確認]をクリックします。

4. 右側のシーンビューのパネルで、各経路点の位置が正確で、周囲と衝突していないことを確認します。



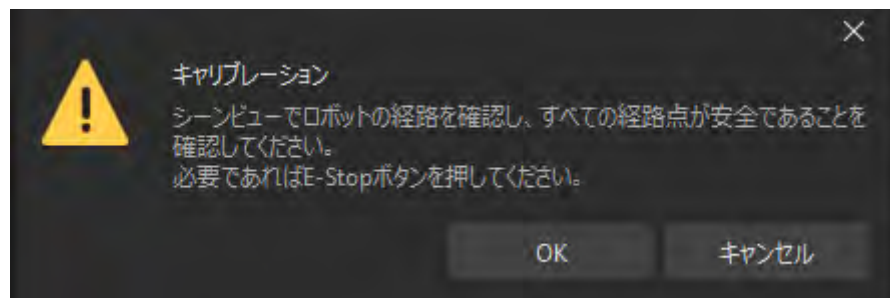
5. 確認後、下部にある[次へ]をクリックします。

画像と位置姿勢を取得

1. 画像と位置姿勢を取得手順で、画像を保存にチェックを入れ、[経路に沿ってロボットを自動で移動させ、画像を撮影]をクリックします。



2. ロボット動作の安全に関する説明をよく読んでから、[OK]をクリックします。

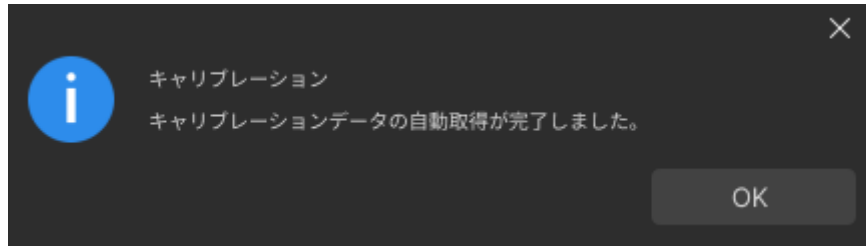


3. ロボットが設定した経路に沿って移動し、カメラが各経路点で画像取得を完了するのを待ちます。右側の画像ビューパネルで取得された画像が表示されます。



- ° ロボット動作中に、ロボットの動作範囲に近づかないようにしてください。
- ° 「ロボットを停止」ボタンをクリックするとキャリブレーションを終了しますが、ロボットはすぐには停止せず、現在の経路点に達するまで停止します。緊急時にロボットティーチペンダントの非常停止ボタンを押してください。非常停止を押した後、ロボットに再接続する必要があります。

4. 画像が取得した後、表示される画面で[OK]をクリックします。



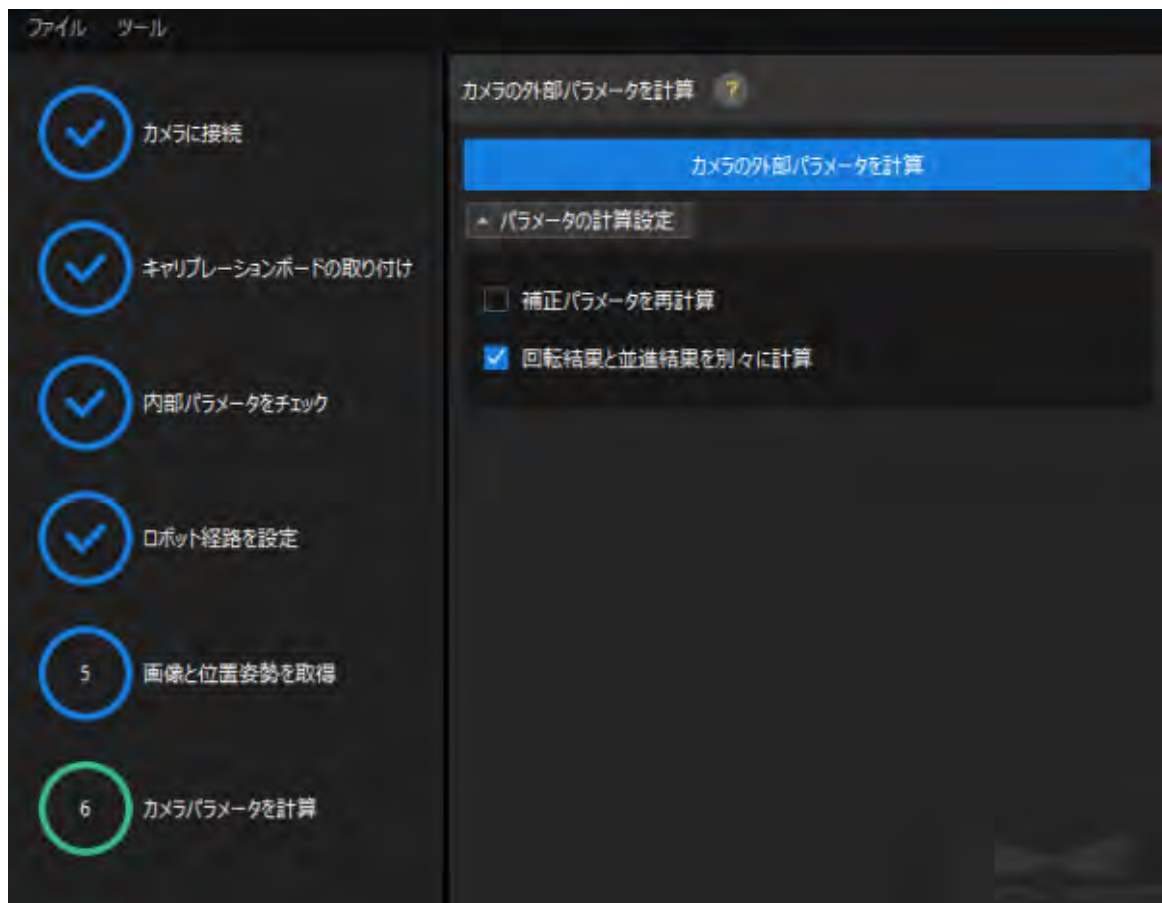
5. 認識したキャリブレーション位置姿勢が要件を満たしていることを確認してから、下部にある[次へ]をクリックします。



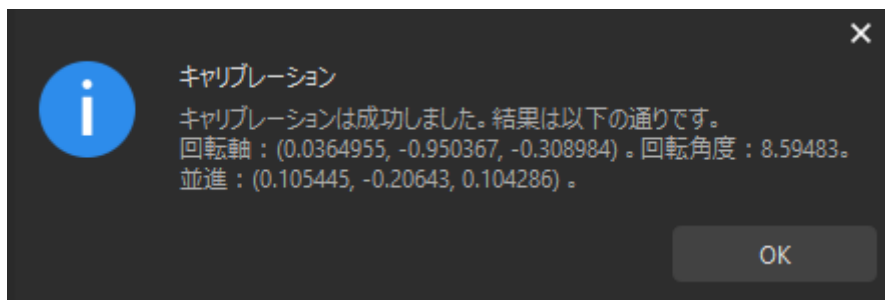
要件を満たしていない場合、ロボットを手動で移動した後（ティーチペンダントでもMech-Vizでも選択可能）、**画像を手動で追加**にチェックを入れ、[**画像を追加してフランジ位置姿勢を記録**]をクリックし、キャリブレーションボードの画像を追加してロボットのフランジ位置姿勢を入力する必要があります。

カメラパラメータを計算

1. カメラパラメータを計算手順で、「パラメータの計算設定」を展開し、**回転結果と並進結果**を別々に計算にチェックを入れてから、[**カメラの外部パラメータを計算**]をクリックします。



2. キャリブレーションが正常に実行されたことを示すダイアログで、[**OK**]をクリックします。



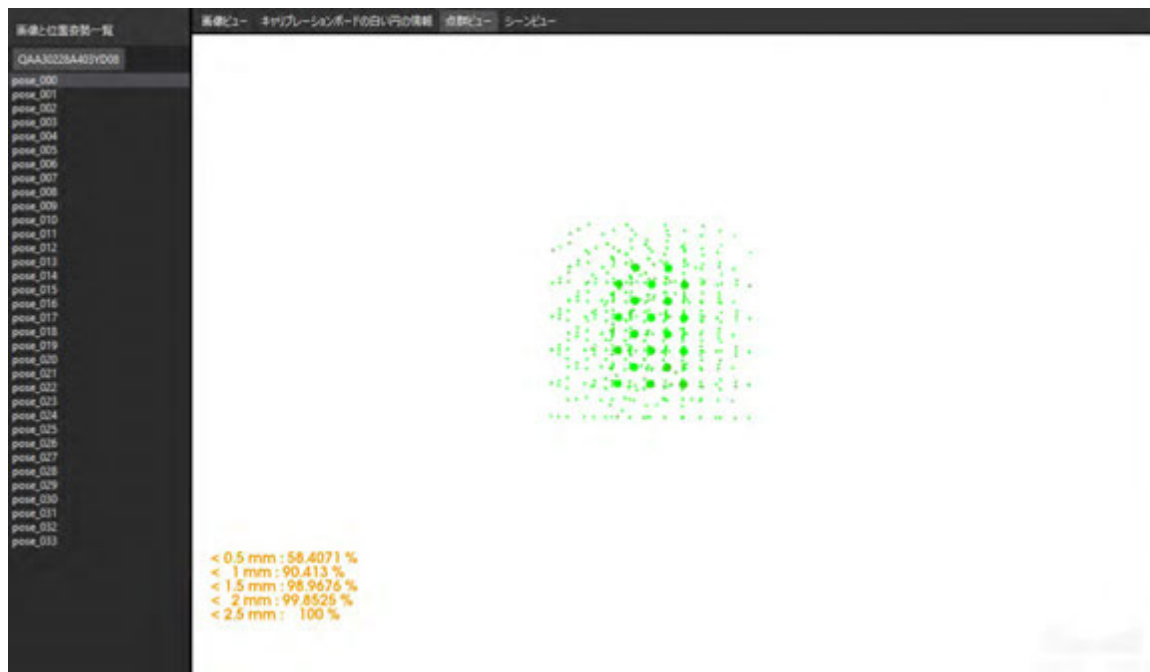
1. 右側の点群ビューパネルでキャリブレーションの誤差点群が表示されます。



誤差点群は、各キャリブレーション位置姿勢におけるキャリブレーションボードの白い円の実測の値と計算値との偏差を示すために使用されます。

2. キャリブレーション精度が要件を満たしていることを確認します。

キャリブレーション精度は、100%を占める誤差の値を求めることで、おおよそ判断することができます。例えば、下図は2.5 mm以下の精度を示しています。



キャリブレーション精度を向上させるには、[キャリブレーション結果の分析](#)をご参照ください。

キャリブレーション結果の確認と保存

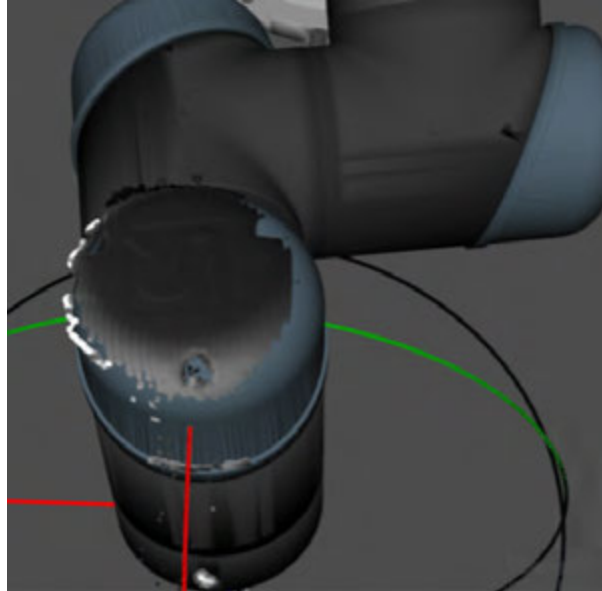
カメラの外部パラメータを計算した後、シーンビューでロボット点群とロボットモデルの点群との重なりを確認してキャリブレーション結果を大まかに判断します。具体的な流れは以下の通りです。

1. ロボットアームをカメラの視野範囲内に移動させます。
2. カメラパラメータを計算手順で[外部パラメータを再計算]をクリックします。クリックする

と、カメラが撮影を行います。

3. シーンビューに切り替え、ロボットの点群とロボットモデルの重なりを確認します。

ロボットの点群がロボットモデルとほぼ一致すれば、キャリブレーション結果が使用可能です。



1. ただし、ロボットモデルはロボット実機の外観と完全に一致しない場合があり、外部パラメータの微調整の基準とすることができないことに注意してください。
2. シーンビューに加え、Mech-Vizでロボットの点群とロボットモデルとの重なりを確認できます。

4. 下部にある [保存] をクリックし、ポップアップする **キャリブレーションファイル** を保存画面で [OK] をクリックします。すると、カメラのキャリブレーション結果は、プロジェクトの「calibration」ディレクトリに自動的に保存されます。

これで、キャリブレーション手順が完了しました。

4.4.4. 標準キャリブレーション法を使用する手動キャリブレーション (Eye to Hand)

本節では、カメラがEye to Hand (ETH)方式で取り付けられている場合に標準キャリブレーション法で手動キャリブレーションを実行する方法について説明します。

事前準備

以下は、ロボットハンド・アイ・キャリブレーションを実行する前に必要な事前準備です。

- Mech-Mindビジョンシステムを構築します。
- キャリブレーションに必要なものを準備します。
- キャリブレーションボードの点群画像生成を調整します。

- ロボットと通信の設定を完了します。

ビジョンシステムを構築

Mech-Mindビジョンシステムの構築については、「ビジョンシステムを構築」節をご参照ください。

ハンド・アイ・キャリブレーションには、Mech-Eye Viewer、Mech-Center、Mech-VisionおよびMech-Vizが必要です。これらのソフトウェアがインストールされ、最新バージョンにアップグレードされていることを確認してください。

キャリブレーションに必要なものを準備

カメラがEye to Hand方式で取り付けられている場合、手動キャリブレーションを実行するにはキャリブレーションボードの使用が必要です。

キャリブレーションボードについては、以下の要件があります。

- キャリブレーションボードの白い円がはっきりと見えること、破損や曲げ変形がないことを確認してください。
- キャリブレーションボードのコネクタをロボット先端のフランジに取り付け、そのコネクタにキャリブレーションボードを取り付けます。キャリブレーションボードがしっかりと取り付けられていること、ロボット先端のXY平面と平行であることを確認してください。



ロボットフランジに取り外しができないハンドがある場合、キャリブレーションボードをハンドに直接固定します。

また、キャリブレーションを実行する前に、ロボットをキャリブレーションの開始位置に移動させてください。

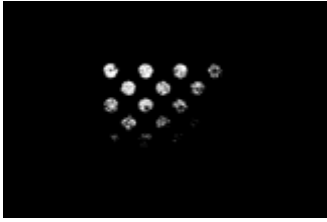
キャリブレーションボードの点群画像生成を調整

1. Mech-Eye Viewerを起動し、カメラのパラメータを調整します。
2. 2Dパラメータを調整して、2D画像にキャリブレーションボードがはっきり見られ、露光過度や露光不足などがないようにします。
3. 3Dパラメータを調整して、キャリブレーションボードの白い円の点群が完全に見られるようにします。点群の変動範囲を小さくするために、**点群後処理**の中の**点群平滑化**と**ノイズ除去**を**Normal**に設定することを推奨します。



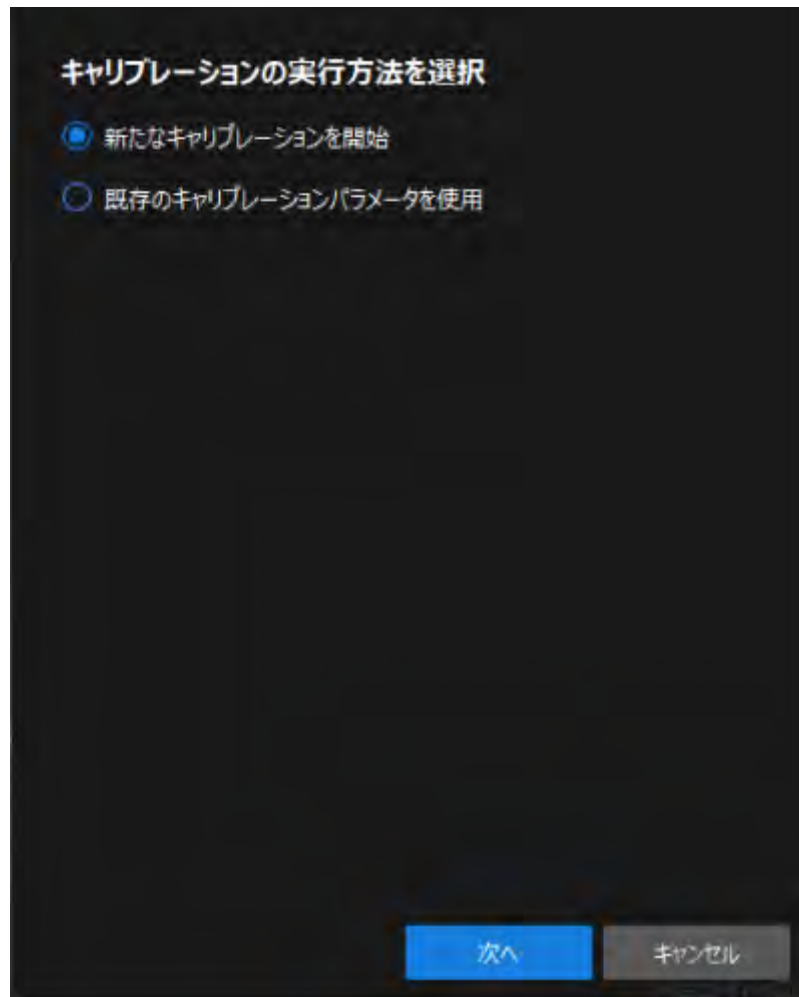
現場の環境光が複雑な場合、2D画像や点群への影響を軽減するために、遮光・補光することを推奨します。

4. 上記の手順を完了することで、キャリブレーションボードの点群の画像品質が要件を満たしていることを確認します。

	正常	露光過度	露光不足
2D画像			
点群			

キャリブレーション前の設定

1. Mech-Visionを起動し、ツールバーで[カメラキャリブレーション（標準モード）]をクリックします。すると、**キャリブレーション前の設定画面**が表示されます。
2. **キャリブレーションの実行方法**を選択画面で、**新たなキャリブレーションを開始**を選択し、[次へ]をクリックします。



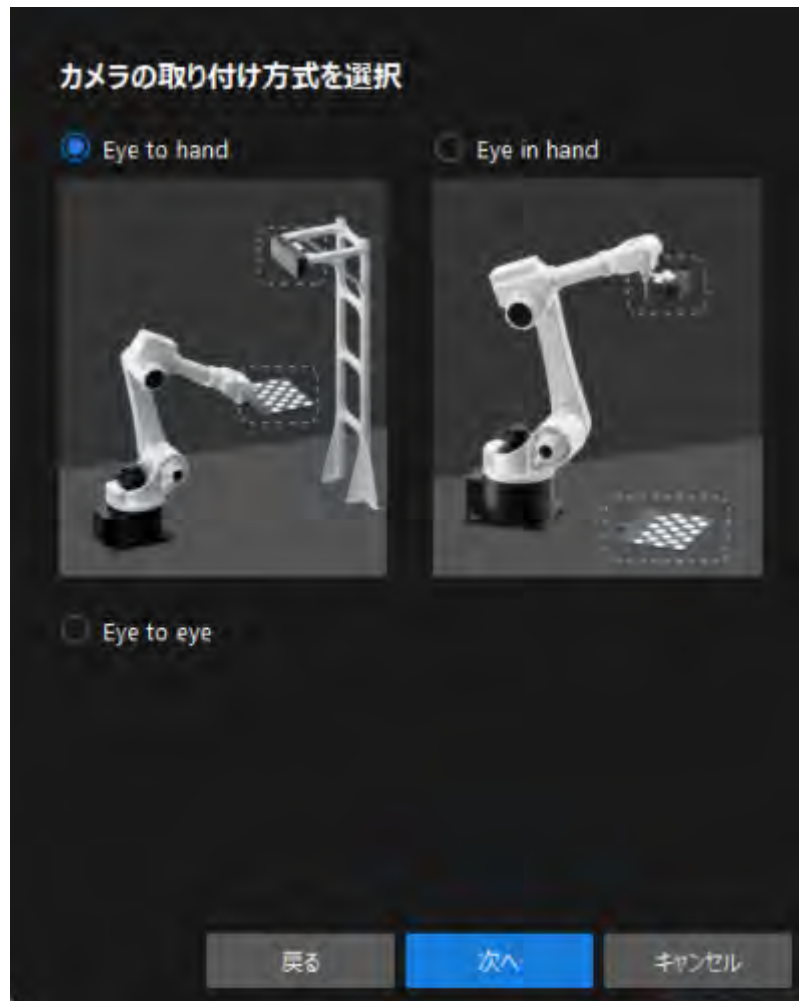
3. キャリブレーションのタスクを選択画面で、ドロップダウンリストから**その他のロボットのハンド・アイ・キャリブレーション**を選択し、必要に応じて**ロボットのオイラー角のタイプ**を指定し、ロボットの座標系を選択してから[次へ]をクリックします。



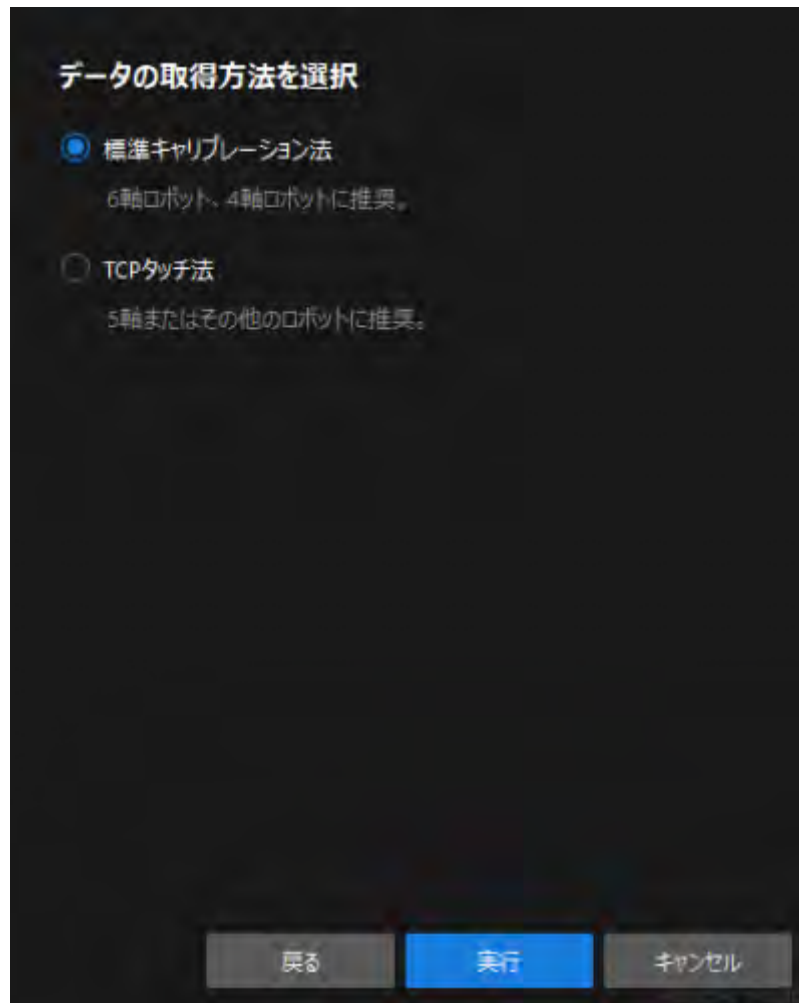
4. キャリブレーションするためのロボットを選択画面で、実際の状況に応じて6軸ロボット、4軸ロボットまたは5軸、または他のタイプのロボットを選択してから、[次へ]をクリックします。



5. カメラの取り付け方式画面で、**Eye to hand**を選択して[次へ]をクリックします。




6. データの取得方法を選択画面で、標準キャリブレーション法を選択してから、[実行]をクリックします。すると、キャリブレーション（Eye to Hand）画面が表示されます。



これで、キャリブレーション前の設定が完了し、キャリブレーションに進むことになります。

キャリブレーションの実行手順

カメラ接続

1. **カメラに接続** 手順で、**検出されたカメラ**から接続するカメラを選択して  をクリックするか、ダブルクリックして接続します。



2. カメラ接続後、[一回キャプチャ]または[連続キャプチャ]をクリックします。



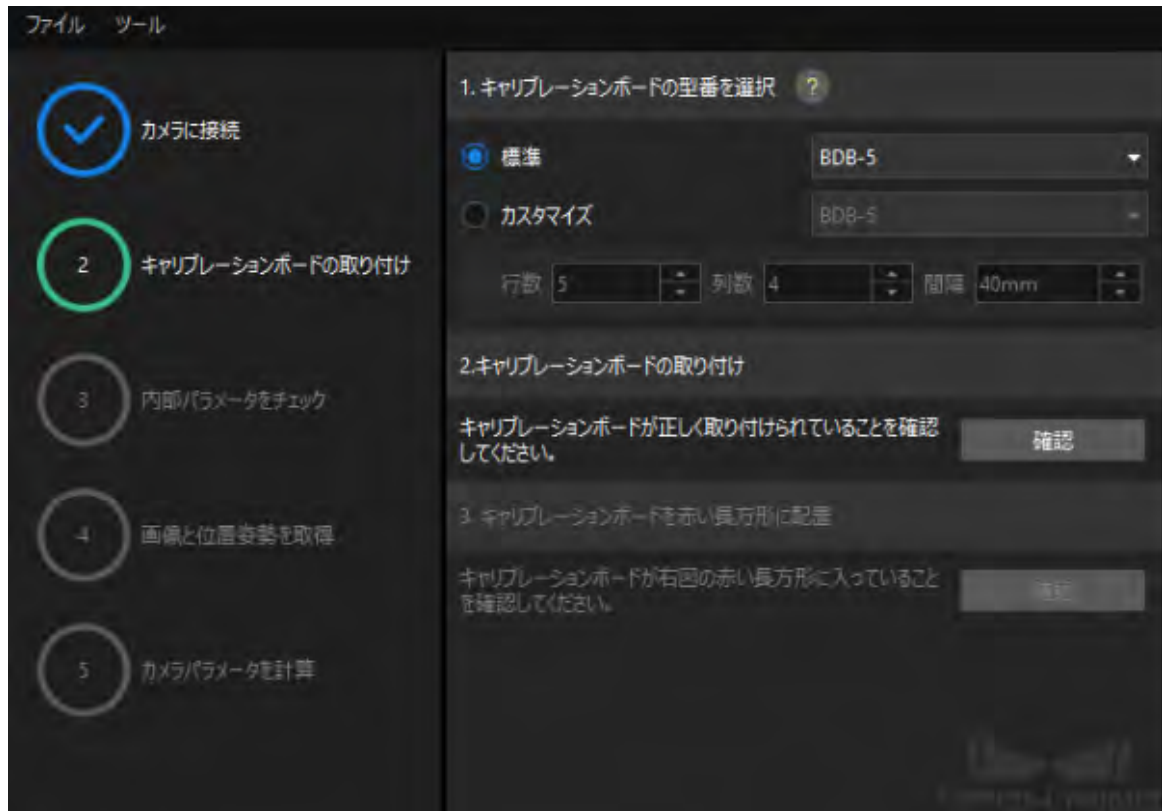
3. 画像ビューでは、取得した2D画像と深度画像が要件を満たしていることを確認してから、下部にある[次へ]をクリックします。



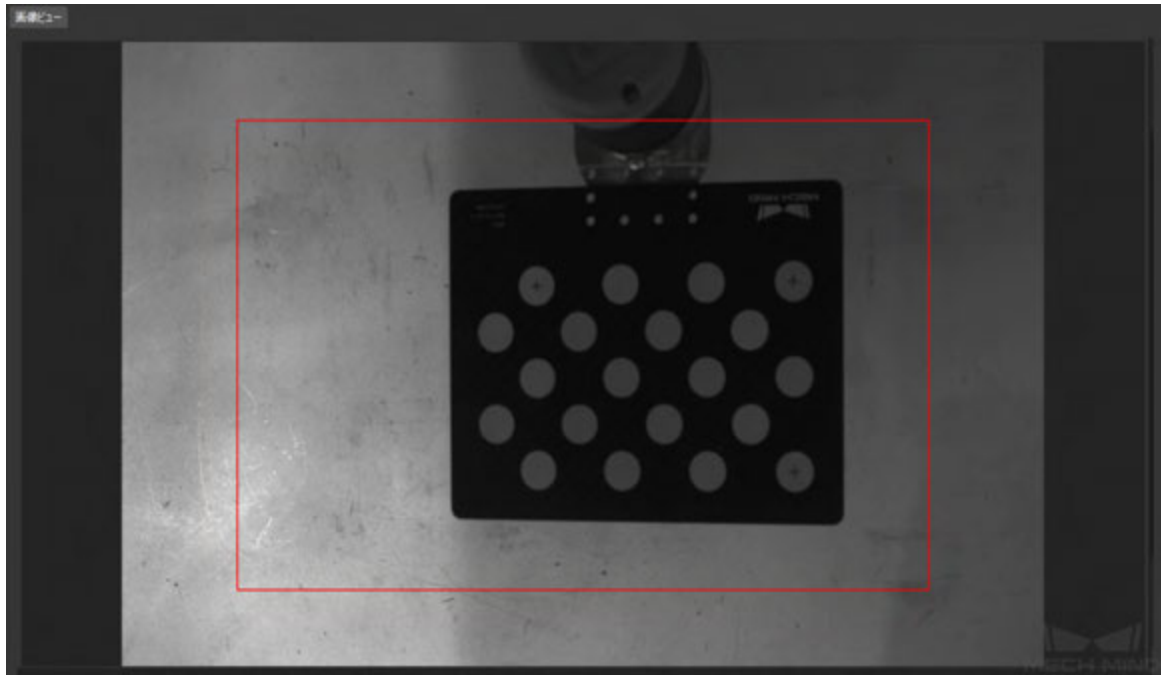
取得した画像が要件を満たしていない場合、Mech-Eye Viewerを起動し、[カメラの2Dと3D露光パラメータを調整](#)して再取得する必要があります。

キャリブレーションボードの取り付け

1. キャリブレーションボードの取り付け手順で、**1.キャリブレーションボードの型番を選択**で標準を選択し、キャリブレーションボードに貼られているラベルに従って型番を選択します。



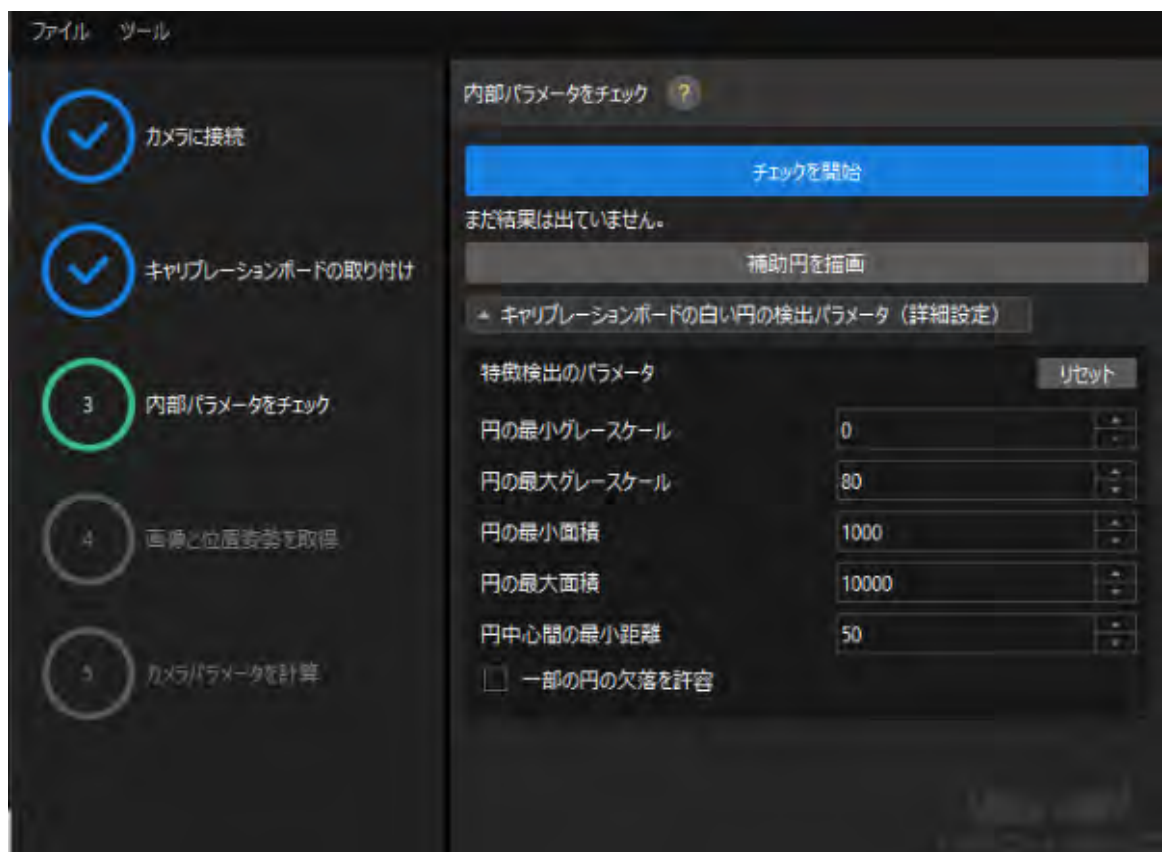
2. キャリブレーションボードが作業平面の中心に配置されていることを確認してから、**2. キャリブレーションボードの取り付け**で[確認]をクリックします。
3. キャリブレーションボードがカメラの視野中心（赤い長方形）にあることを確認してから、**3. キャリブレーションボードを赤い長方形に配置**で[確認]をクリックします。



4. キャリブレーションボードに関するすべての操作が完了したら、下部にある[次へ]をクリックします。

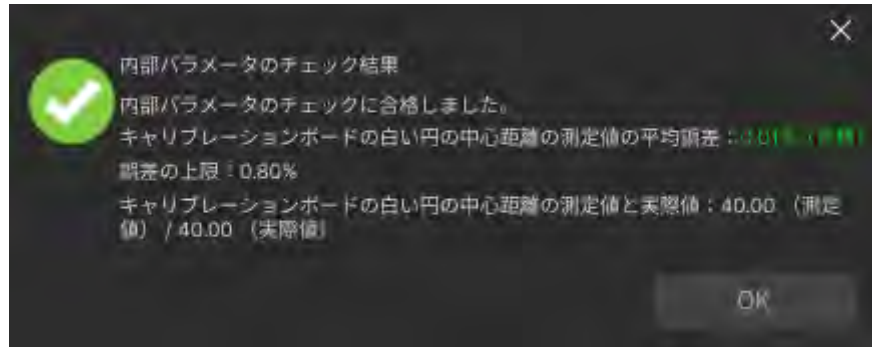
カメラの内部パラメータをチェック

1. 内部パラメータをチェック手順で[チェックを開始]をクリックします。



2. 内部パラメータのチェック結果を確認します。

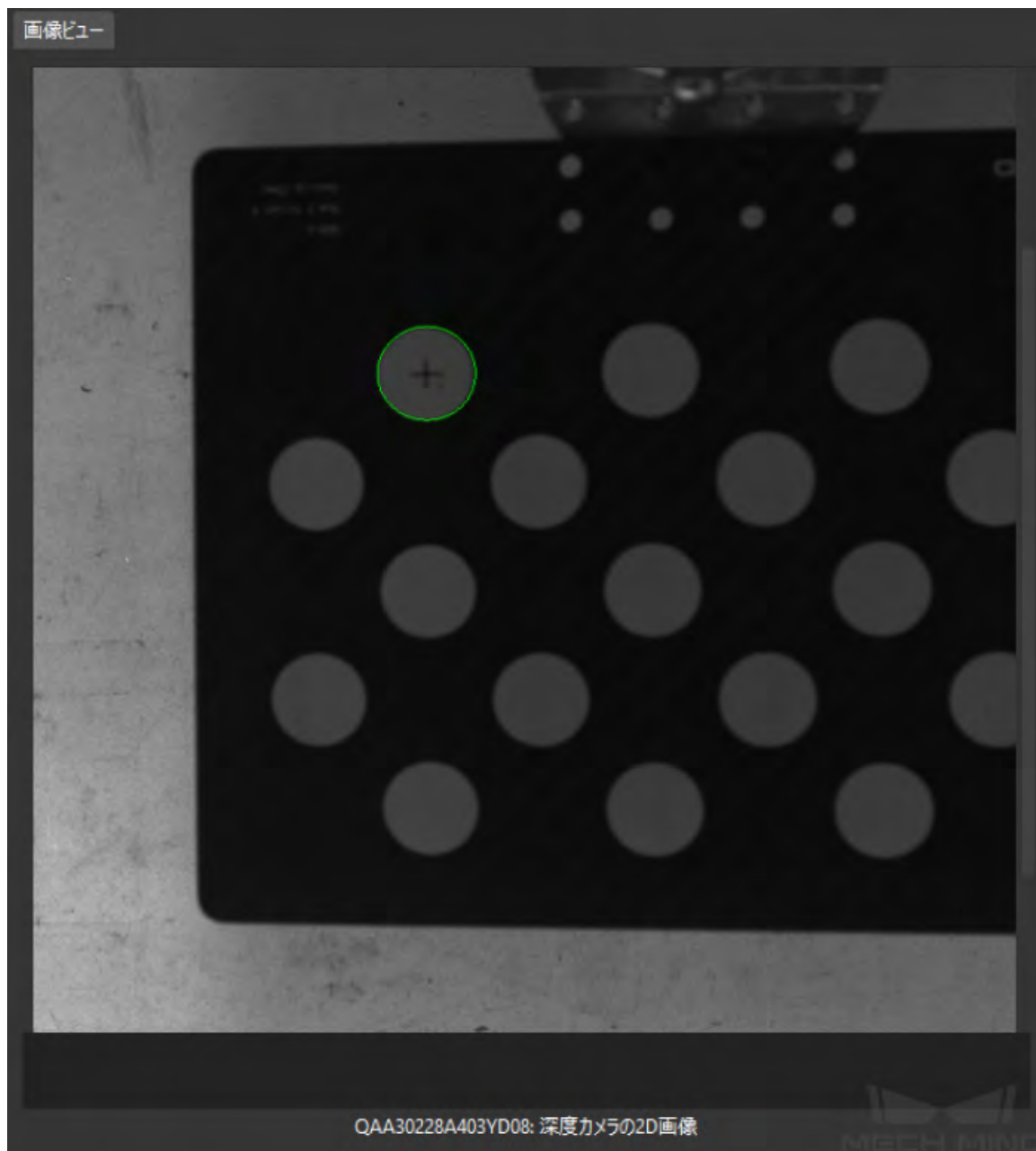
- カメラの内部パラメータが合格した場合、[OK]をクリックして[次へ]をクリックします。



- カメラの内部パラメータが合格しなかった場合、補助円を描画するか、キャリブレーションボードの白い円の検出パラメータを手動で編集してから、[再度チェック]をクリックします。

補助円を描画

- 補助円を描画する場合、[補助円を描画]をクリックします。
- 右側の画像ビューパネルで、キャリブレーションボードの画像を右クリックし、**ウィンドウに合わせる**のチェックを外します。その後、**Ctrl**キーを押しながらスクロールホイールをドラッグして画像のサイズを適切な大きさに変更します。
- キャリブレーションボードの白い円の十字型の中心にマウスポインタを合わせて、マウスの左ボタンを押し、補助円をキャリブレーションボードの白い円に完全に含ませてから離します。



4. **[再度チェック]**をクリックし、カメラ内部パラメータのチェックが合格したことを確認します。

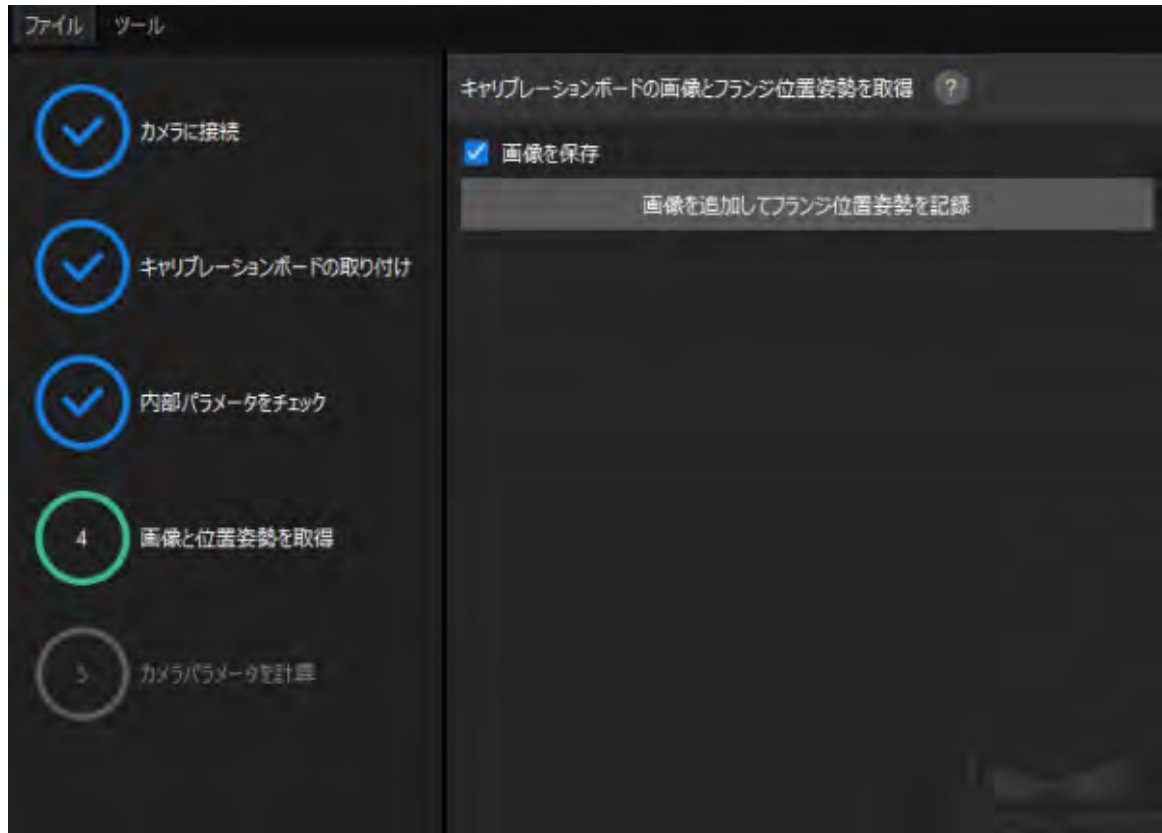
検出パラメータを手動で調整

検出パラメータを手動で調整する場合、**[キャリブレーションボードの白い円の検出パラメータ (詳細設定)]**をクリックし、実際の状況に応じてパラメータ値を変更します。

それでもキャリブレーションボードの白い円が検出されない場合、現場の環境に応じてカメラの関連パラメータを調整する必要があります。カメラパラメータの調整については、[パラメータ調整](#)をご参照ください。

画像と位置姿勢を取得

1. 画像と位置姿勢を取得手順で、[画像を保存]にチェックを入れます。



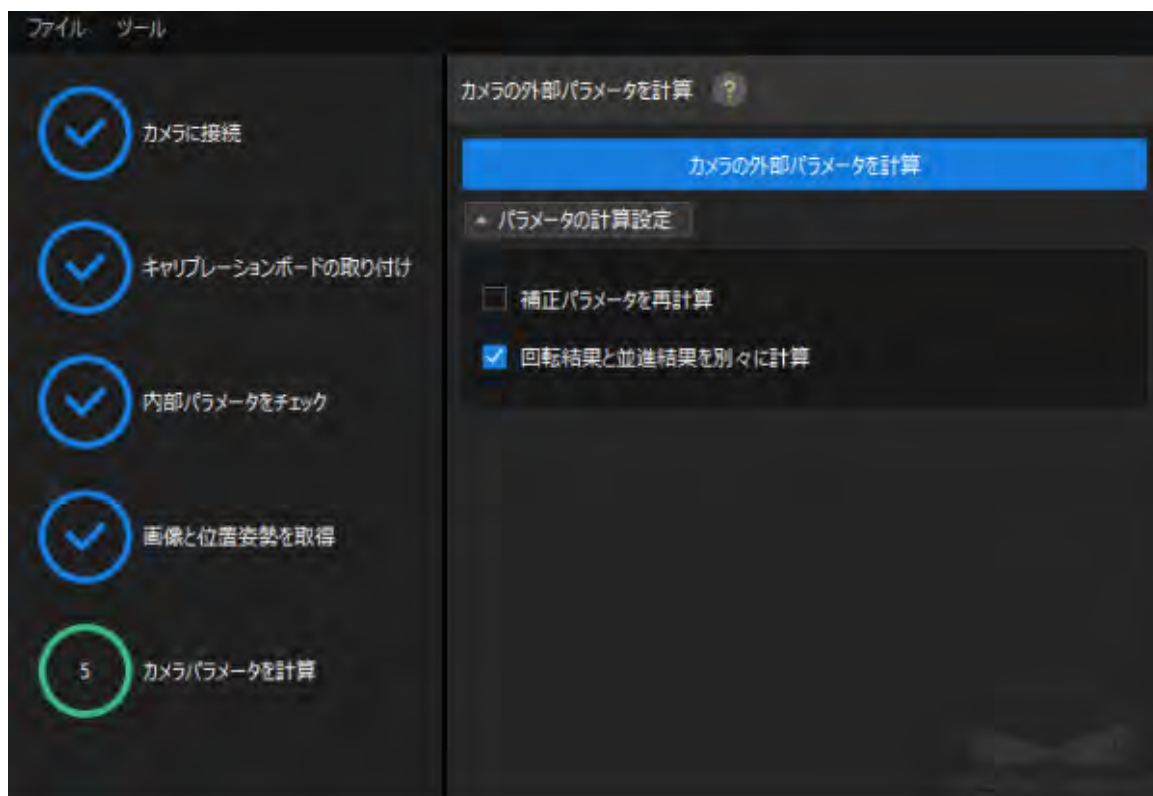
2. ロボットを異なる位置に移動させ、[画像を追加してフランジ位置姿勢を記録]をクリックします。
3. 表示される画面でロボットのフランジ位置姿勢を入力します。



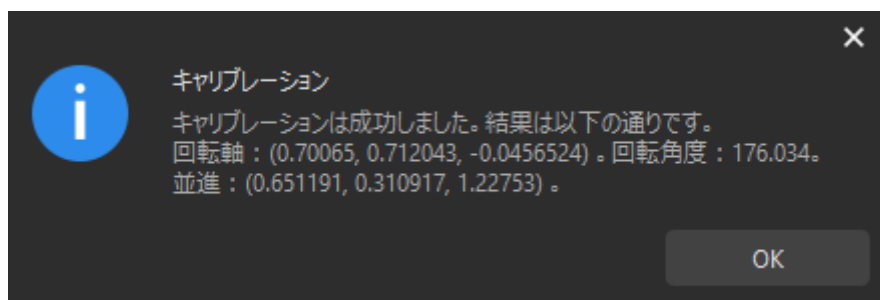
4. 認識したキャリブレーションポイントが要件を満たしていることを確認してから、下部にある[次へ]をクリックします。

カメラパラメータを計算

1. カメラパラメータを計算手順で、「パラメータの計算設定」を展開し、**回転結果と並進結果を別々に計算**にチェックを入れてから、[カメラの外部パラメータを計算]をクリックします。



2. キャリブレーションが正常に実行されたことを示すダイアログで、[OK]をクリックします。



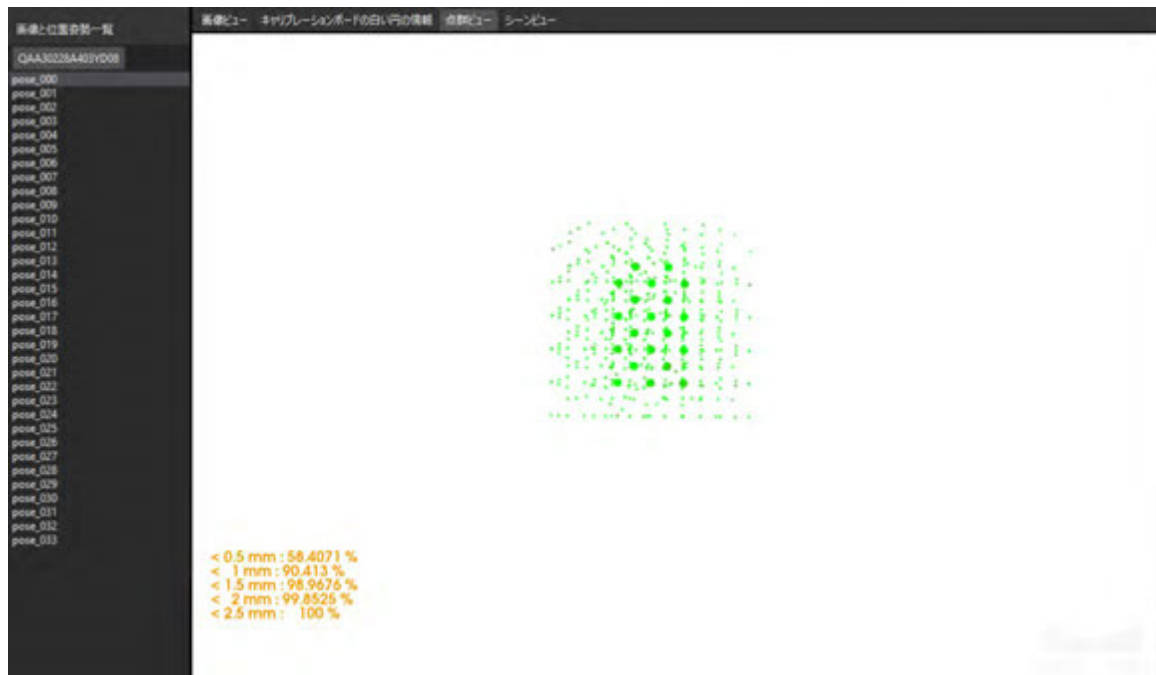
3. 右側の 点群ビュー パネルでキャリブレーションの誤差点群が表示されます。



誤差点群は、各キャリブレーション位置姿勢におけるキャリブレーションボードの白い円の実際の値と計算値との偏差を示すために使用されます。

4. キャリブレーション精度が要件を満たしていることを確認します。

キャリブレーション精度は、100%を占める誤差の値を求めることで、おおよそ判断することができます。例えば、下図は2.5 mm以下の精度を示しています。



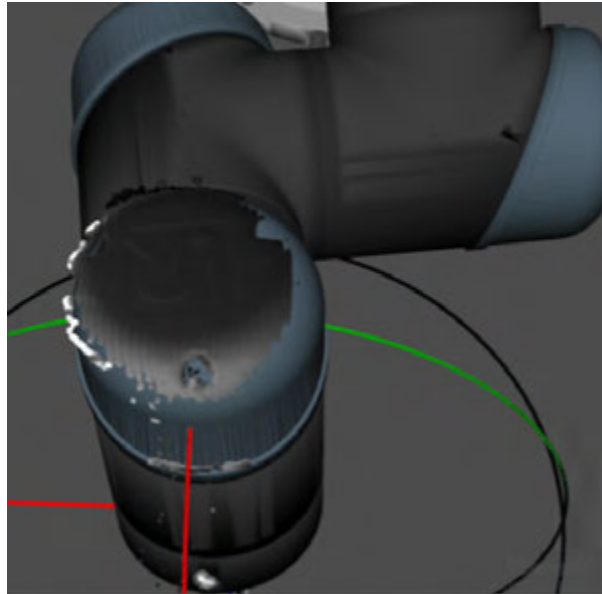
キャリブレーション精度を向上させるには、[キャリブレーション結果の分析](#)をご参照ください。

キャリブレーション結果の確認と保存

カメラの外部パラメータを計算した後、シーンビューでロボット点群とロボットモデルの点群との重なりを確認してキャリブレーション結果を大まかに判断します。具体的な流れは以下の通りです。

1. ロボットアームをカメラの視野範囲内に移動させます。
2. カメラパラメータを計算手順で[外部パラメータを再計算]をクリックします。クリックすると、カメラが撮影を行います。
3. シーンビューに切り替え、ロボットの点群とロボットモデルの重なりを確認します。

ロボットの点群がロボットモデルとほぼ一致すれば、キャリブレーション結果が使用可能です。



1. ただし、ロボットモデルはロボット実機の外観と完全に一致しない場合があり、外部パラメータの微調整の基準とすることができないことに注意してください。
2. シーンビューに加え、Mech-Vizでロボットの点群とロボットモデルとの重なりを確認できます。

4. 下部にある[保存]をクリックし、ポップアップする **キャリブレーションファイルを保存** 画面で[OK]をクリックします。すると、カメラのキャリブレーション結果は、プロジェクトの「calibration」ディレクトリに自動的に保存されます。

これで、キャリブレーション手順が完了しました。

4.4.5. 標準キャリブレーション法を使用する手動キャリブレーション (Eye in Hand)

本節では、カメラがEye in Hand (EIH)方式で取り付けられている場合に標準キャリブレーション法で手動キャリブレーションを実行する方法について説明します。

事前準備

以下は、ロボットハンド・アイ・キャリブレーションを実行する前に必要な事前準備です。

- Mech-Mindビジョンシステムを構築します。
- キャリブレーションに必要なものを準備します。
- キャリブレーションボードの点群画像生成を調整します。
- ロボットと通信の設定を完了します。

ビジョンシステムを構築

Mech-Mindビジョンシステムの構築については、「ビジョンシステムを構築」節をご参照ください。

ハンド・アイ・キャリブレーションには、Mech-Eye Viewer、Mech-Center、Mech-VisionおよびMech-Vizが必要です。これらのソフトウェアがインストールされ、最新バージョンにアップグレードされていることを確認してください。

キャリブレーションに必要なものを準備

カメラがEye in Hand方式で取り付けられている場合、手動キャリブレーションを実行するにはキャリブレーションボードの使用が必要です。

キャリブレーションボードについては、以下の要件があります。

- キャリブレーションボードの白い円がはっきりと見えること、破損や曲げ変形がないことを確認してください。
- キャリブレーションボードを作業平面の中心に配置してください。

また、キャリブレーションを実行する前に、ロボットをキャリブレーションの開始位置に移動させてください。




キャリブレーションボードの点群画像生成を調整

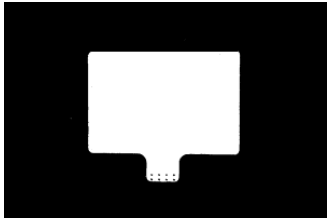
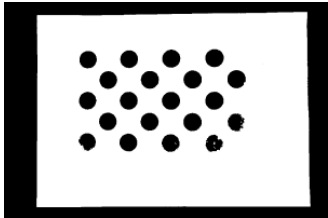
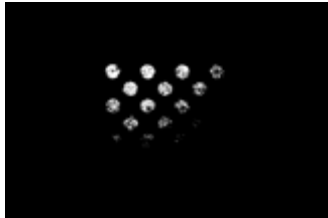
1. Mech-Eye Viewerを起動し、カメラのパラメータを調整します。
2. 2Dパラメータを調整して、2D画像にキャリブレーションボードがはっきり見られ、露光過度や露光不足などがないようにします。
3. 3Dパラメータを調整して、キャリブレーションボードの白い円の点群が完全に見られるようにします。点群の変動範囲を小さくするために、**点群後処理**の中の**点群平滑化**と**ノイズ除去**を**Normal**に設定することを推奨します。



現場の環境光が複雑な場合、2D画像や点群への影響を軽減するために、遮光・補光することを推奨します。

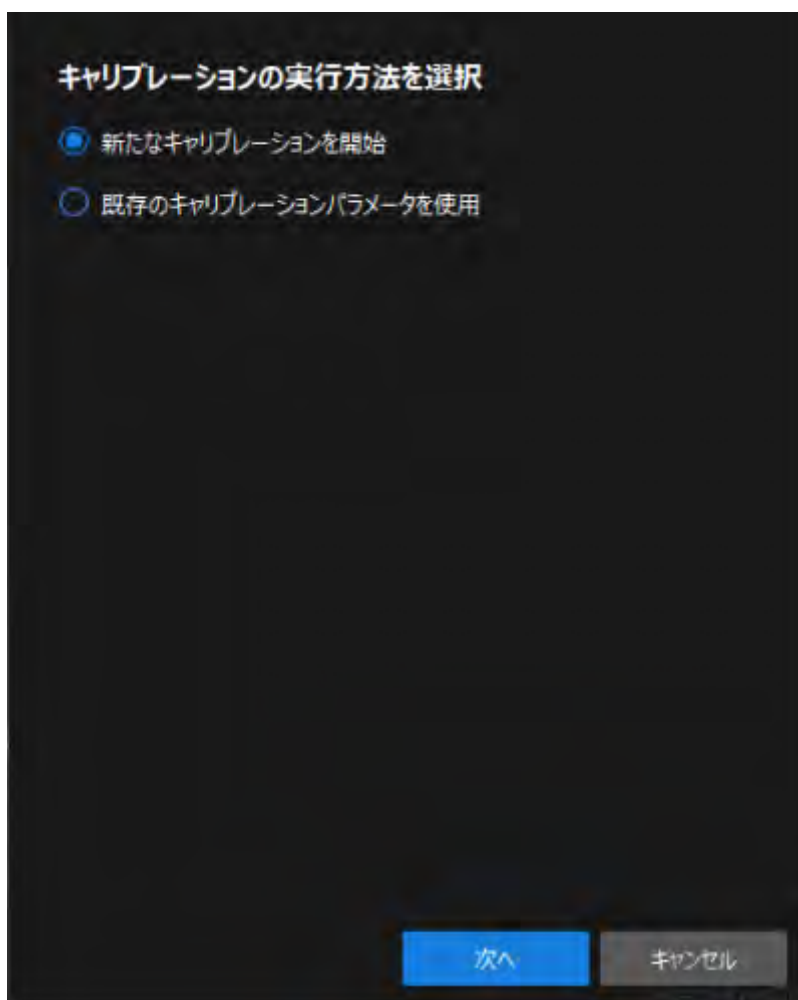
4. 上記の手順を完了することで、キャリブレーションボードの点群の画像品質が要件を満たしていることを確認します。

	正常	露光過度	露光不足
2D画像			

	正常	露光過度	露光不足
点群			

キャリブレーション前の設定

1. Mech-Visionを起動し、ツールバーで[カメラキャリブレーション（標準モード）]をクリックします。すると、キャリブレーション前の設定画面が表示されます。
2. キャリブレーションの実行方法を選択画面で、新たなキャリブレーションを開始を選択し、[次へ]をクリックします。



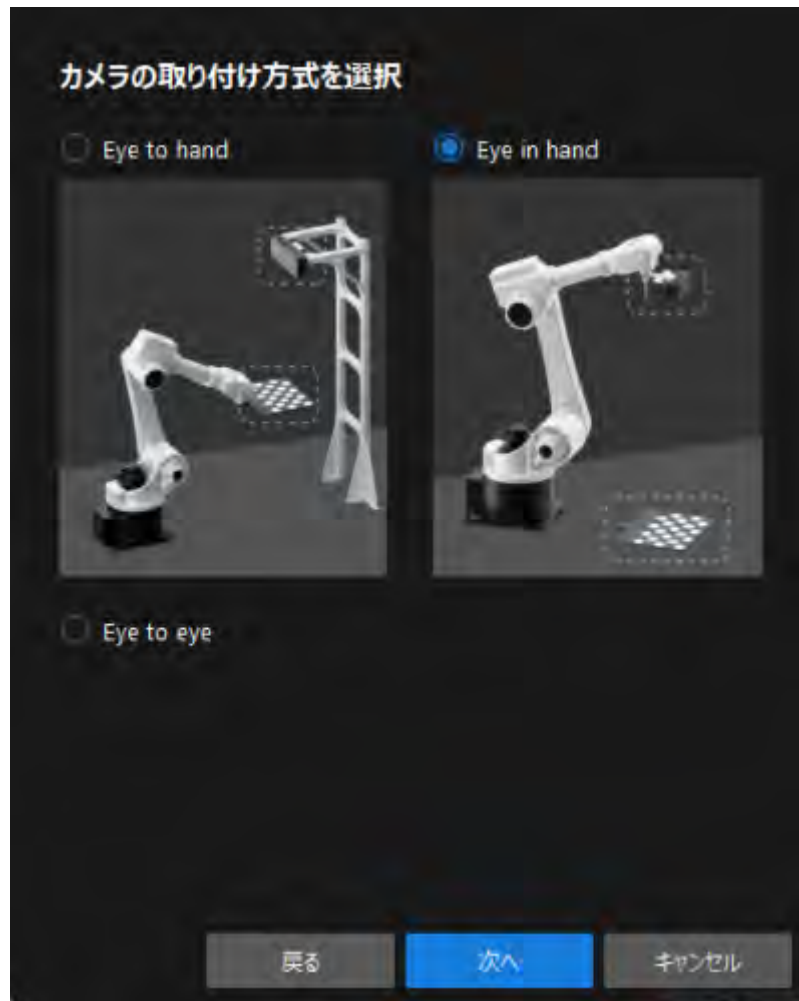
3. キャリブレーションのタスクを選択画面で、ドロップダウンリストからその他のロボットのハンド・アイ・キャリブレーションを選択し、必要に応じてロボットのオイラー角のタイプを指定し、ロボットの座標系を選択してから[次へ]をクリックします。



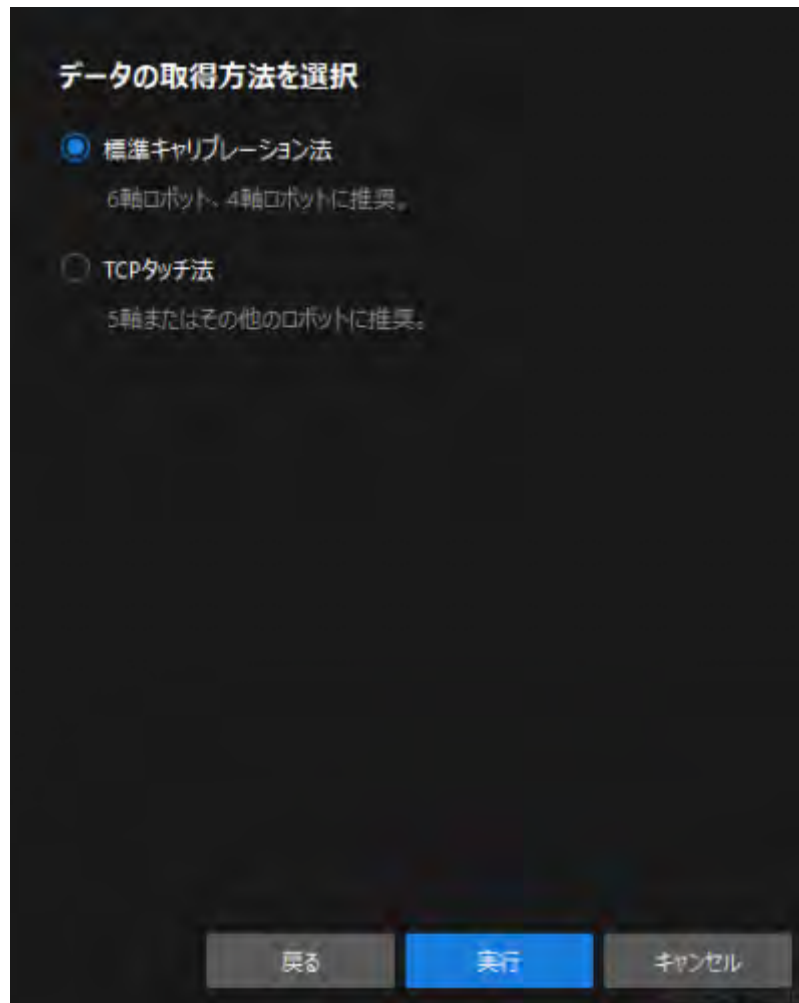
4. キャリブレーションするためのロボットを選択画面で、実際の状況に応じて6軸ロボット、4軸ロボットまたは5軸、または他のタイプのロボットを選択してから、[次へ]をクリックします。



5. カメラ取り付け方式を選択画面で、**Eye in hand**にチェックを入れ、[次へ]をクリックします。




6. データの取得方法を選択画面で、標準キャリブレーション法を選択してから、[実行]をクリックします。すると、キャリブレーション（Eye in Hand）画面が表示されます。



これで、キャリブレーション前の設定が完了し、キャリブレーションに進むことになります。

キャリブレーションの実行手順

カメラの接続

1. **カメラに接続** 手順で、**検出されたカメラ** から接続するカメラを選択して  をクリックするか、ダブルクリックして接続します。



2. 接続後、[一回キャプチャ]または[連続キャプチャ]をクリックします。



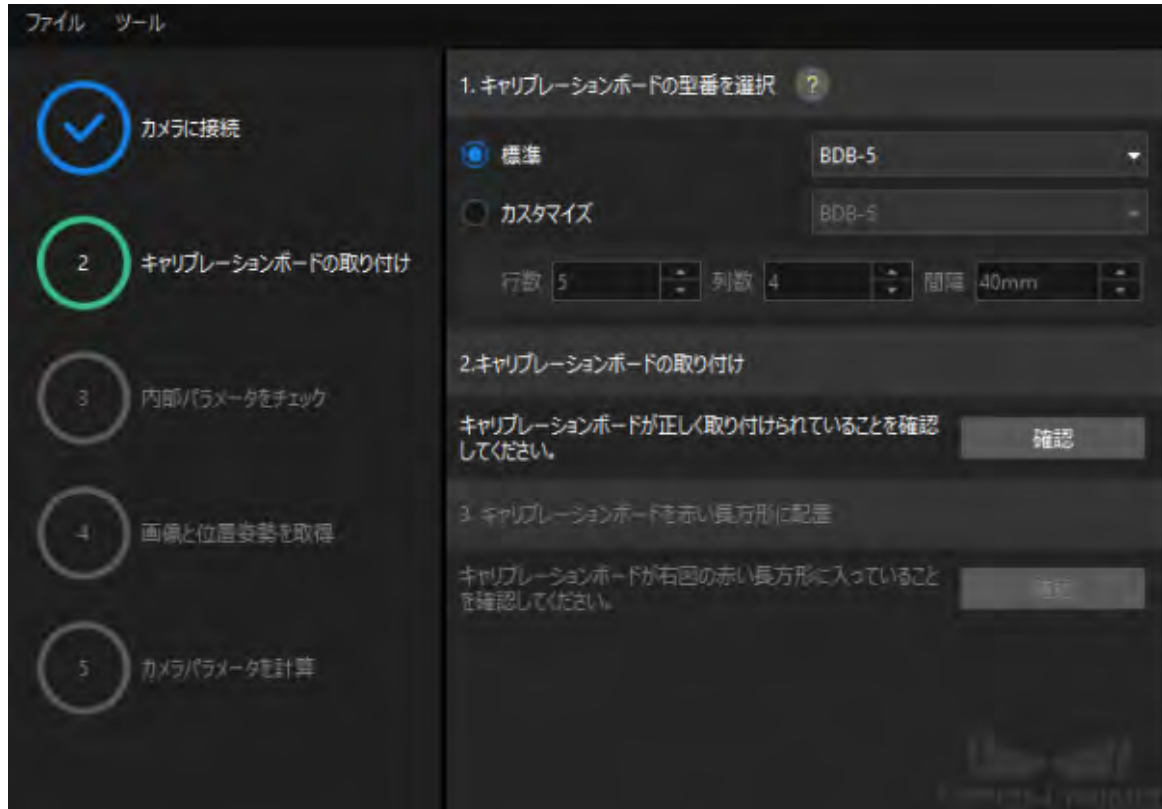
3. 画像ビューでは、取得した2D画像と深度画像が要件を満たしていることを確認した上で下部にある[次へ]をクリックします。



取得した画像が要件を満たしていない場合、Mech-Eye Viewerを起動し、[カメラの2Dと3D露光パラメータを調整](#)して再取得する必要があります。

キャリブレーションボードの取り付け

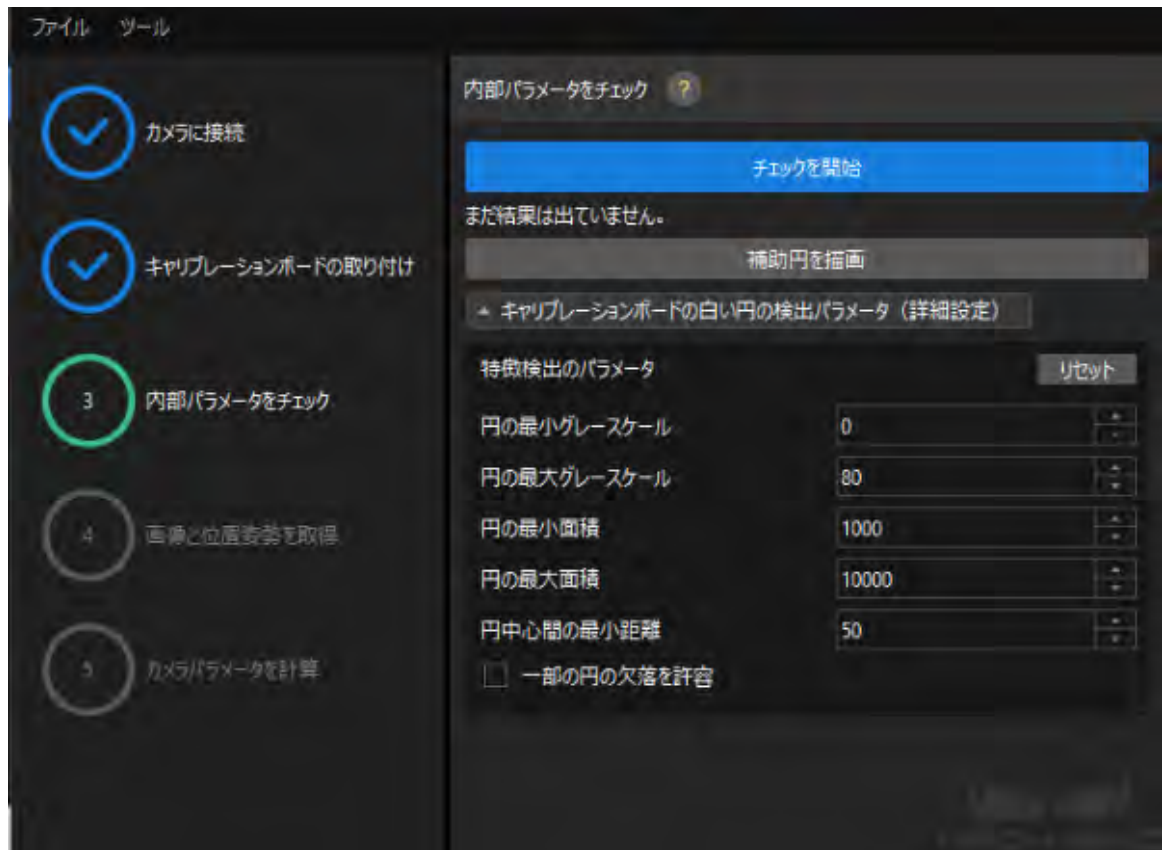
1. キャリブレーションボードの取り付け手順で、**1. キャリブレーションボードの型番を選択**で標準を選択し、キャリブレーションボードに貼られているラベルに従って型番を選択します。



2. キャリブレーションボードが作業平面の中心に配置されていることを確認してから、**2. キャリブレーションボードの取り付け**で[確認]をクリックします。
3. キャリブレーションボードがカメラの視野中心（赤い長方形）にあることを確認してから、**3. キャリブレーションボードを赤い長方形に配置**で[確認]をクリックします。
4. キャリブレーションボードに関するすべての操作が完了したら、下部にある[次へ]をクリックします。

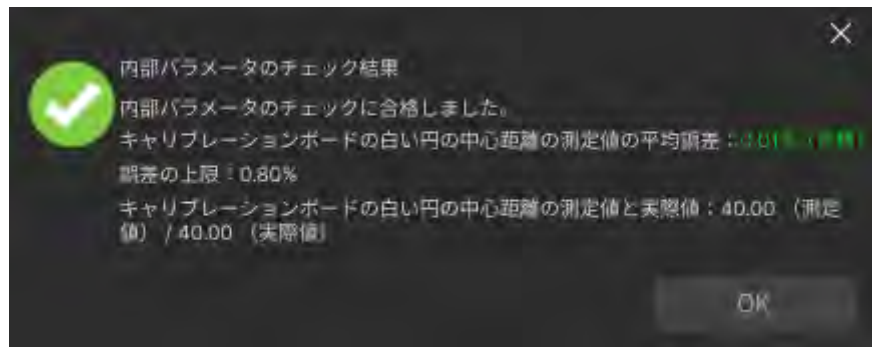
カメラの内部パラメータをチェック

1. 内部パラメータをチェック手順で[チェックを開始]をクリックします。



2. 内部パラメータのチェック結果を確認します。

- カメラの内部パラメータが合格した場合、[OK]をクリックして[次へ]をクリックします。

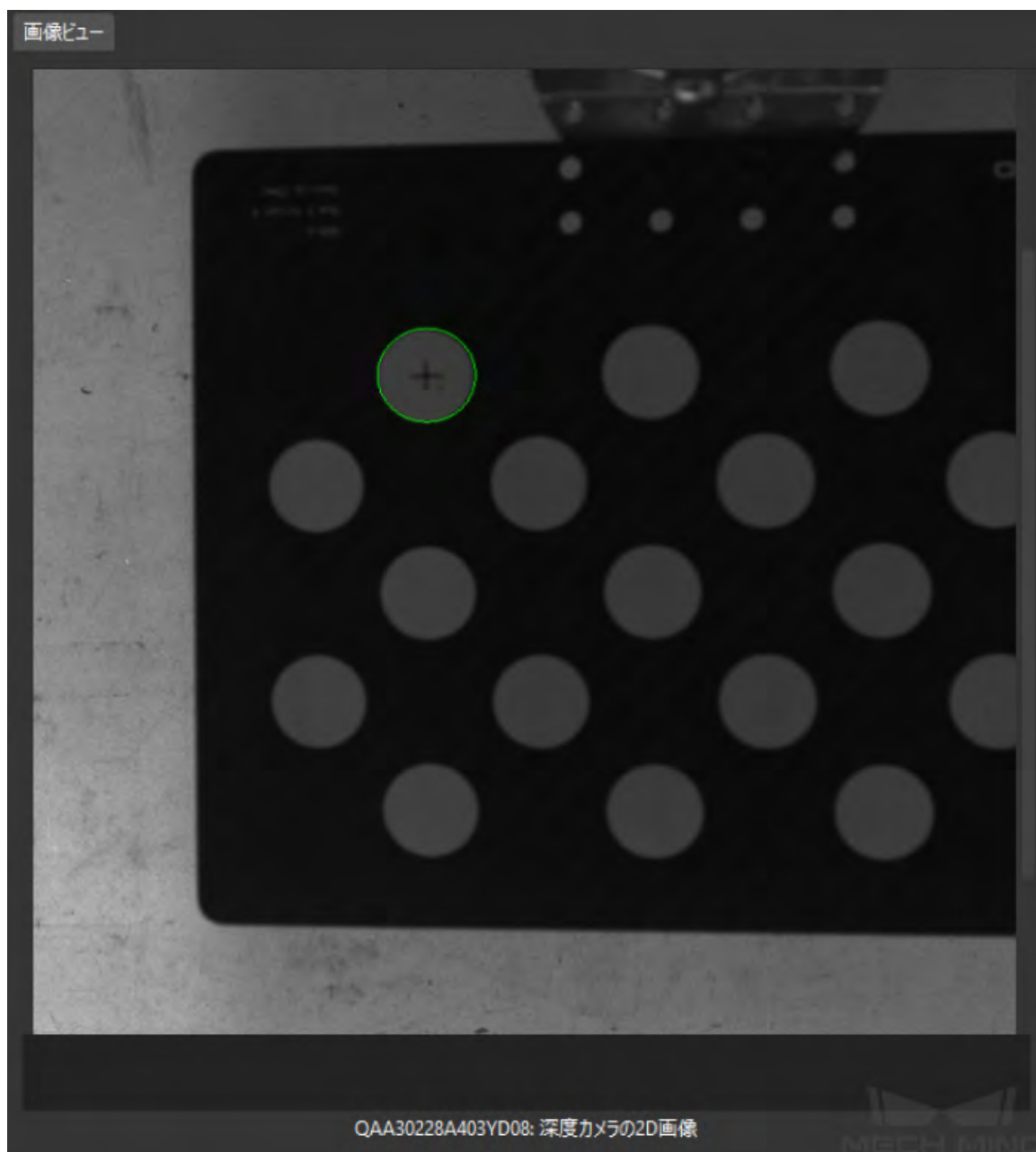


- カメラの内部パラメータが合格しなかった場合、補助円を描画するか、キャリブレーションボードの白い円の検出パラメータを手動で編集してから、[再度チェック]をクリックします。

補助円を描画

- 補助円を描画する場合、[補助円を描画]をクリックします。
- 右側の画像ビューパネルで、キャリブレーションボードの画像を右クリックし、**ウィンドウに合わせる**のチェックを外します。その後、**Ctrl**キーを押しながらスクロールホイールをドラッグして画像のサイズを適切な大きさに変更します。

3. キャリブレーションボードの白い円の十字型の中心にマウスポインタを合わせて、マウスの左ボタンを押し、補助円をキャリブレーションボードの白い円に完全に含ませてから離します。



4. **[再度チェック]**をクリックし、カメラ内部パラメータのチェックが合格したことを確認します。

検出パラメータを手動で調整

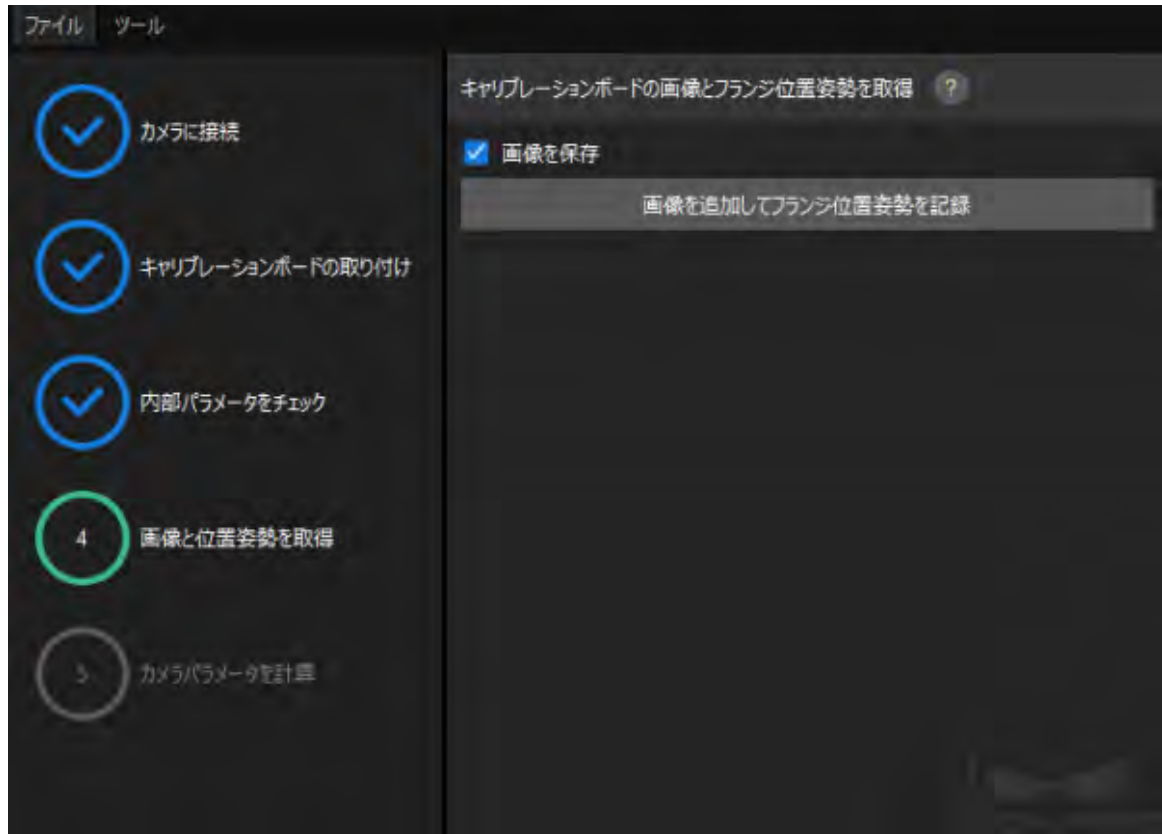
検出パラメータを手動で調整する場合、**[キャリブレーションボードの白い円の検出パラメータ (詳細設定)]**をクリックし、実際の状況に応じてパラメータ値を変更します。

それでもキャリブレーションボードの白い円が検出されない場合、現場の環境に応じてカメラの関連パラメータを調整する必要があります。カメラパラメータの調整については、[パラメー](#)

タ調整 をご参照ください。

画像と位置姿勢を取得

1. 画像と位置姿勢を取得手順で、**画像を保存**にチェックを入れます。



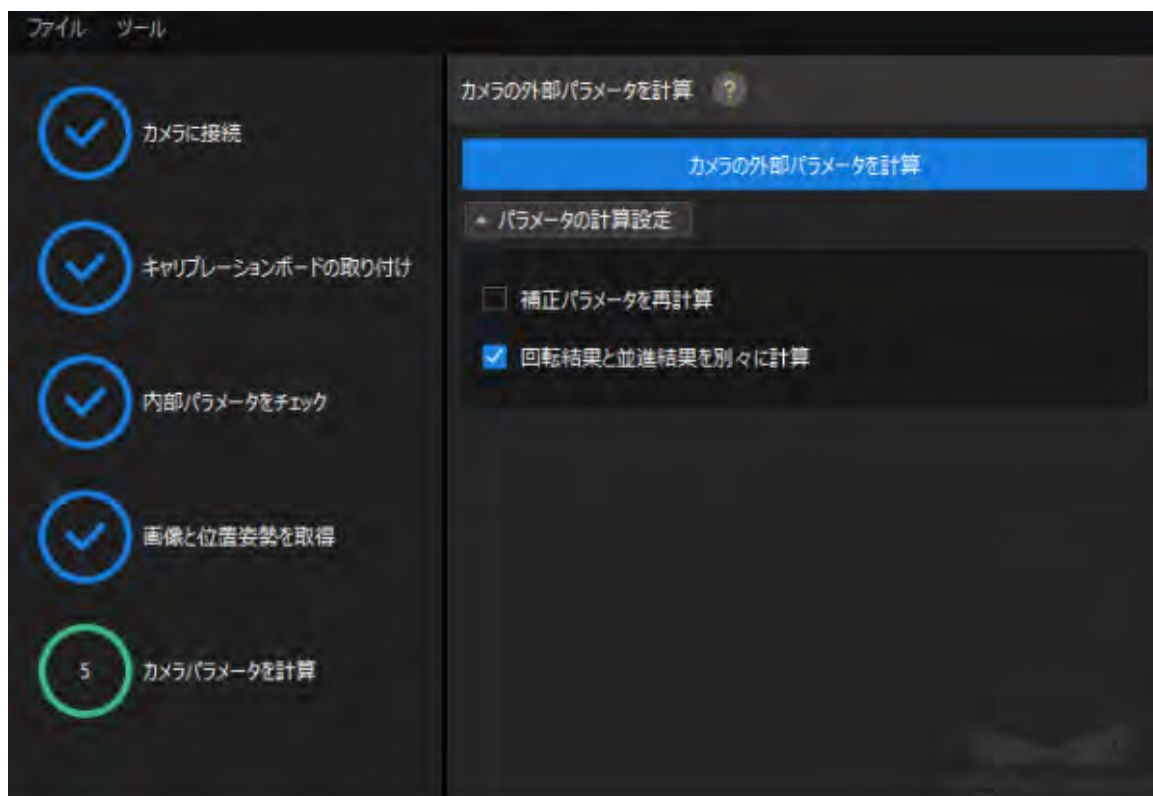
2. ロボットを異なる位置に移動させ、[**画像を追加してフランジ位置姿勢を記録**]をクリックします。
3. 表示される画面でロボットのフランジ位置姿勢を入力します。



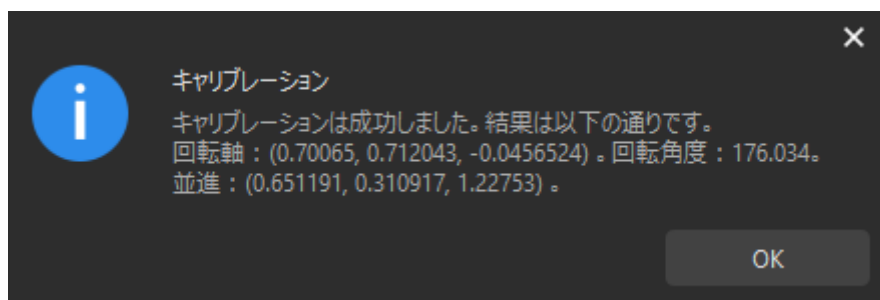
4. 認識したキャリブレーションポイントが要件を満たしていることを確認してから、下部にある[次へ]をクリックします。

カメラパラメータを計算

1. カメラパラメータを計算手順で、「パラメータの計算設定」を展開し、**回転結果と並進結果を別々に計算**にチェックを入れてから、[カメラの外部パラメータを計算]をクリックします。



2. キャリブレーションが正常に実行されたことを示すダイアログで、[OK]をクリックします。



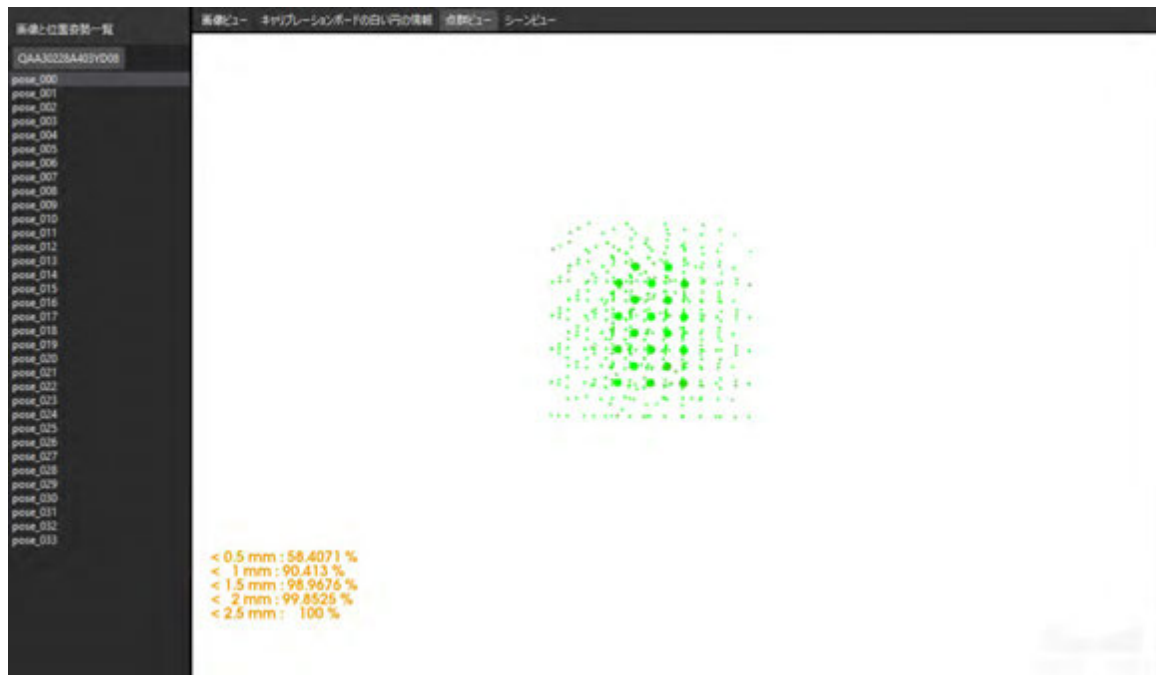
3. 右側の点群ビューパネルでキャリブレーションの誤差点群が表示されます。



誤差点群は、各キャリブレーション位置姿勢におけるキャリブレーションボードの白い円の実値と計算値との偏差を示すために使用されます。

4. キャリブレーション精度が要件を満たしていることを確認します。

キャリブレーション精度は、100%を占める誤差の値を求めることで、おおよそ判断することができます。例えば、下図は2.5 mm以下の精度を示しています。



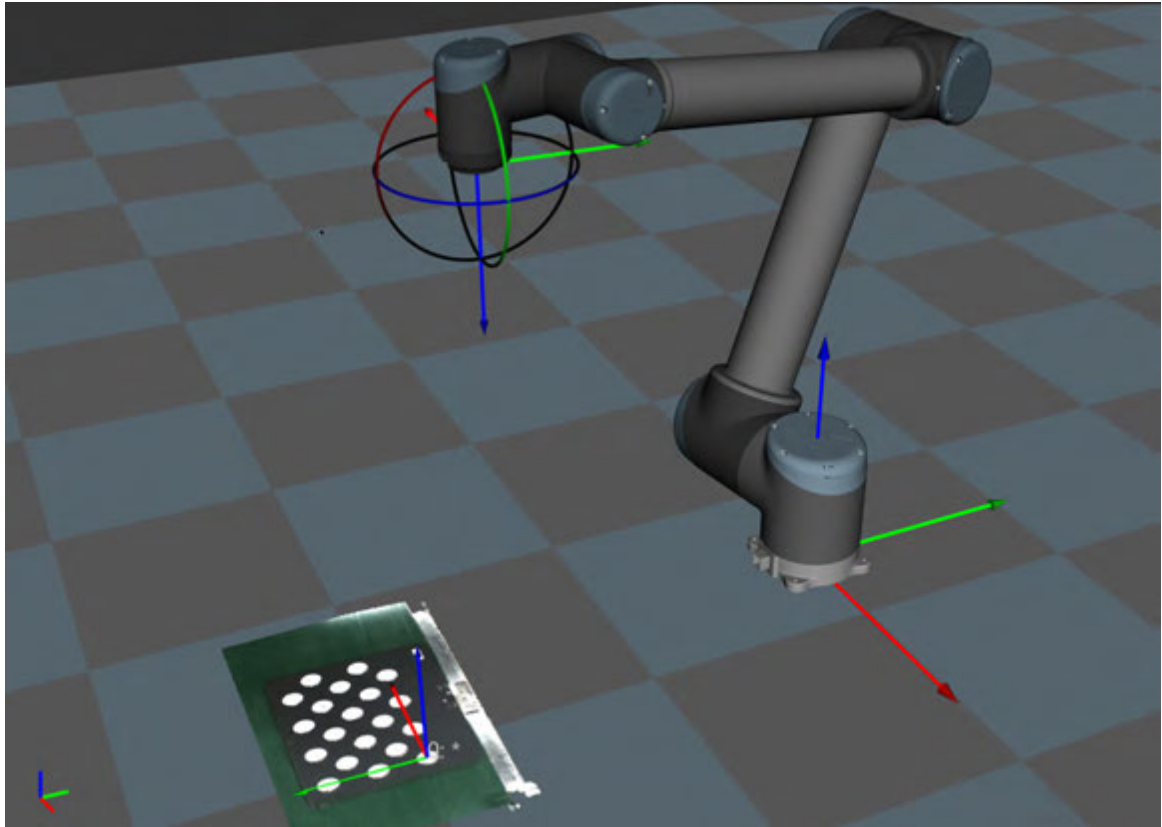
キャリブレーション精度を向上させるには、[キャリブレーション結果の分析](#)をご参照ください。

キャリブレーション結果の確認と保存

カメラの外部パラメータを計算した後、シーンビューで固定点に対するキャリブレーションボードの点群（キャリブレーションボードの白い円の十字中心）のずれを確認することで、キャリブレーション結果を大まかに判断することができます。詳細は以下の通りです。

1. キャリブレーションボードを固定位置に配置します。
2. Mech-Vizを起動し、固定点を追加し、それをキャリブレーションボードの白い円の1つの十字の中心に合わせます。
 - a. ステップライブラリから「移動」ステップを選択して、それをプロジェクト編集エリアにドラッグします。
 - b. このステップをクリックして選択し、ステップパラメータで**目標点タイプ**を「」に設定します。それがキャリブレーションボードの白い円のいずれかの十字中心と一致するように、位置姿勢のX/Y/Z座標値を調整します。
3. ロボットを制御してカメラの位置姿勢を数回変更し、キャリブレーションの**カメラパラメータを計算**手順で**[外部パラメータを再計算]**ボタンをクリックします。クリックすると、カメラが撮影を行います。
4. シーンビューで、キャリブレーションボードの点群が固定点に対して大きくなずれがあるかどうかを確認します。

キャリブレーションボードの点群が固定点に対して大きくなずれがない場合、キャリブレーション結果が使用可能であることを示します。



1. EIHの場合、ロボットが接続されていないとロボットの位置姿勢はリアルタイムで取得できないため、シーンビューまたはMech-Vizに表示される正確な点群を確認するには、画像撮影時のロボットの位置姿勢を記入する必要があります。
2. シーンビューに加え、キャリブレーションボードの点群が固定点に対して大きくずれていないかどうかをMech-Vizで確認することも可能です。

1. 下部にある[保存]をクリックし、ポップアップする**キャリブレーションファイルを保存**画面で[OK]をクリックします。すると、カメラのキャリブレーション結果は、プロジェクトの「calibration」ディレクトリに自動的に保存されます。

これで、キャリブレーション手順が完了しました。

4.4.6. TCPタッチ法を使用する手動キャリブレーション (Eye to Hand)

本節では、カメラがEye to Hand (ETH)方式で取り付けられている場合にTCPタッチ法で手動キャリブレーションを実行する方法について説明します。

事前準備

以下は、ロボットハンド・アイ・キャリブレーションを実行する前に必要な事前準備です。

- Mech-Mindビジョンシステムを構築します。
- キャリブレーションに必要なものを準備します。
- キャリブレーションボードの点群画像生成を調整します。

- ロボットと通信の設定を完了します。

ビジョンシステムを構築

Mech-Mindビジョンシステムの構築については、「ビジョンシステムを構築」節をご参照ください。

ハンド・アイ・キャリブレーションには、Mech-Eye Viewer、Mech-Center、Mech-VisionおよびMech-Vizが必要です。これらのソフトウェアがインストールされ、最新バージョンにアップグレードされていることを確認してください。

キャリブレーションに必要なものを準備

カメラがEye to Hand方式で取り付けられている場合、手動キャリブレーションを実行するにはキャリブレーションボードとTCP先端の使用が必要です。

キャリブレーションボードについては、以下の要件があります。

- キャリブレーションボードの白い円がはっきりと見えること、破損や曲げ変形がないことを確認してください。
- キャリブレーションボードを作業平面の中心に配置してください。

TCP先端をロボットフランジに取り付ける必要があります。ロボットフランジに取り外しができないハンドがある場合、TCP先端をハンドに直接固定します。

また、キャリブレーションを実行する前に、ロボットをキャリブレーションの開始位置に移動させて、カメラがキャリブレーションボードの画像を遮らないようにします。

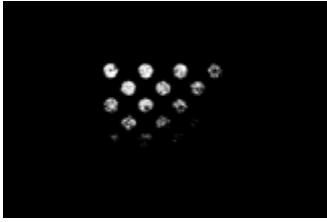
キャリブレーションボードの点群画像生成を調整

1. Mech-Eye Viewerソフトウェアを起動し、カメラのパラメータを調整します。
2. 2Dパラメータを調整して、2D画像にキャリブレーションボードがはっきり見られ、露光過度や露光不足などがないようにします。
3. 3Dパラメータを調整して、キャリブレーションボードの白い円の点群が完全に見られるようにします。点群の変動範囲を小さくするために、**点群後処理**の中の**点群平滑化**と**ノイズ除去**を**Normal**に設定することを推奨します。



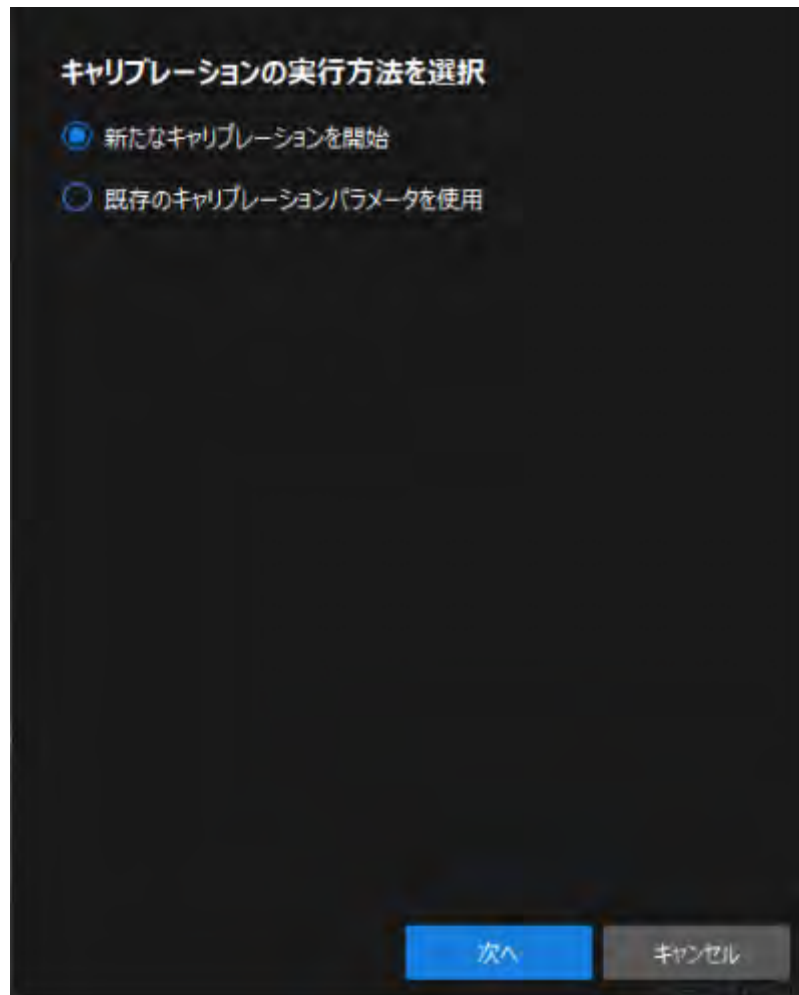
現場の環境光が複雑な場合、2D画像や点群への影響を軽減するために、遮光・補光することを推奨します。

4. 上記の手順を完了することで、キャリブレーションボードの点群の画像品質が要件を満たしていることを確認します。

	正常	露光過度	露光不足
2D画像			
点群			

キャリブレーション前の設定

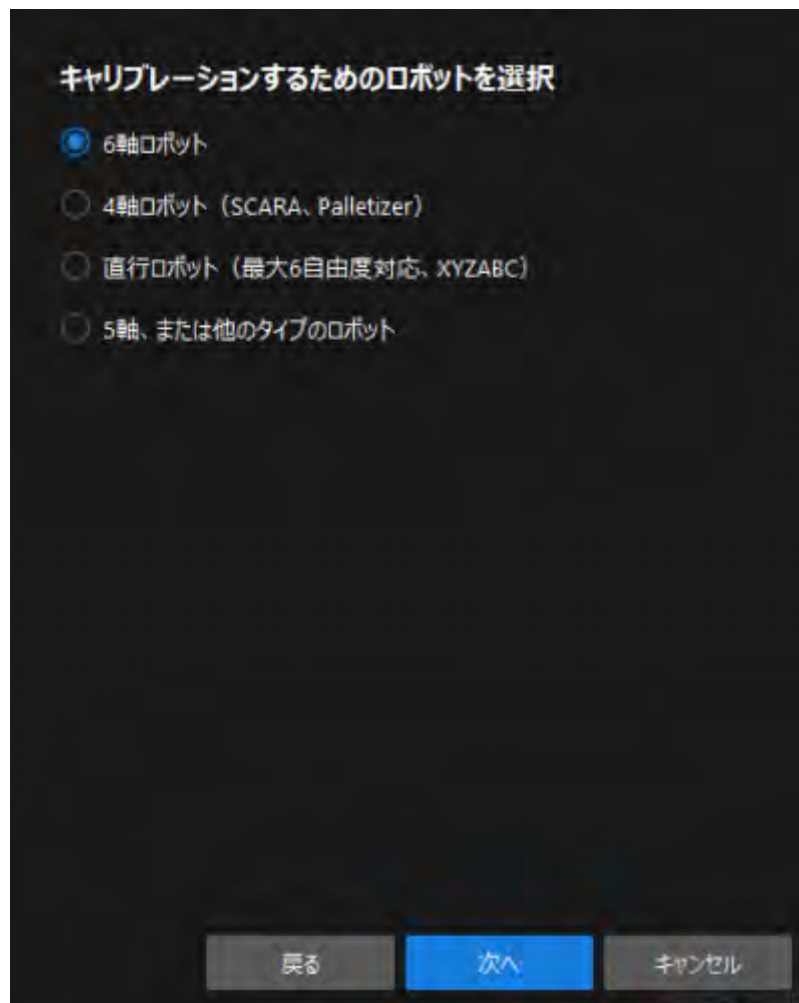
1. Mech-Visionを起動し、ツールバーで[カメラキャリブレーション（標準モード）]をクリックします。すると、キャリブレーション前の設定画面が表示されます。
2. キャリブレーションの実行方法を選択画面で、新たなキャリブレーションを開始を選択し、[次へ]をクリックします。



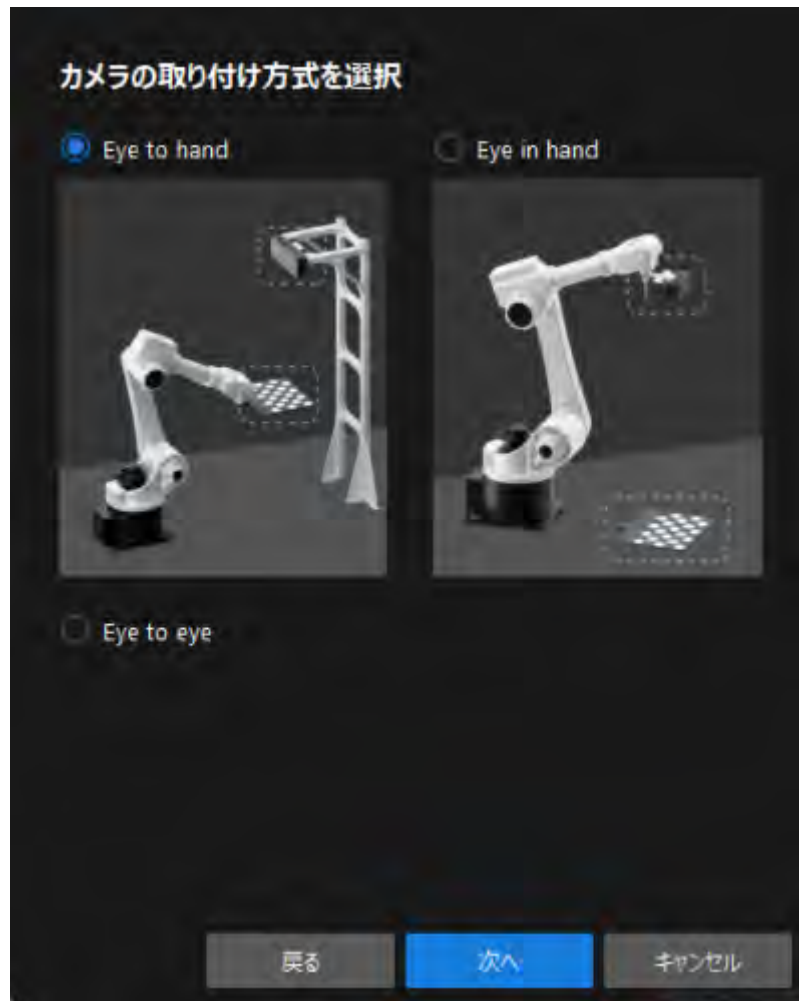
3. キャリブレーションのタスクを選択画面で、ドロップダウンリストから**その他のロボットのハンド・アイ・キャリブレーション**を選択し、必要に応じて**ロボットのオイラー角のタイプ**を指定し、ロボットの座標系を選択してから[次へ]をクリックします。



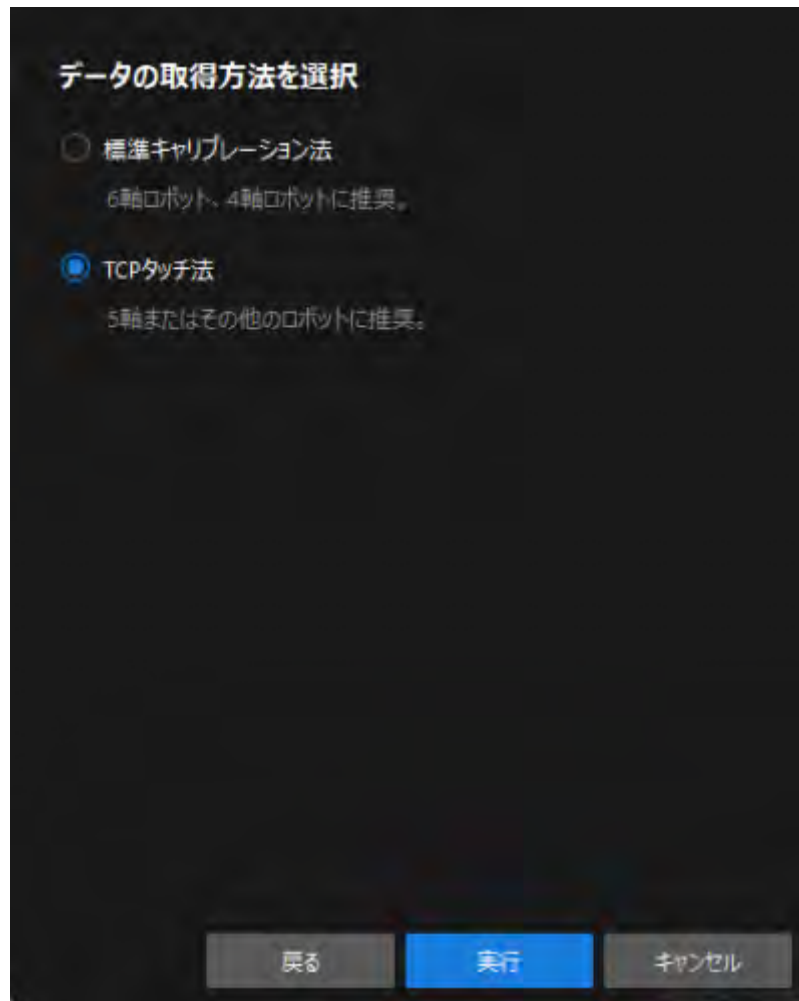
4. キャリブレーションするためのロボットを選択画面で、実際の状況に応じて6軸ロボット、4軸ロボットまたは5軸、または他のタイプのロボットを選択してから、[次へ]をクリックします。



5. カメラの取り付け方式を選択画面で、**Eye to hand**を選択して[次へ]をクリックします。




6. データの取得方法を選択画面で、TCPタッチ法を選択してから、[実行]をクリックします。
すると、キャリブレーション（Eye to Hand）画面が表示されます。



これで、キャリブレーション前の設定が完了し、キャリブレーションに進むことになります。

キャリブレーションの実行手順

カメラ接続

1. **カメラに接続** 手順で、**検出されたカメラ**から接続するカメラを選択して  をクリックするか、ダブルクリックして接続します。



2. カメラ接続後、[一回キャプチャ]または[連続キャプチャ]をクリックします。



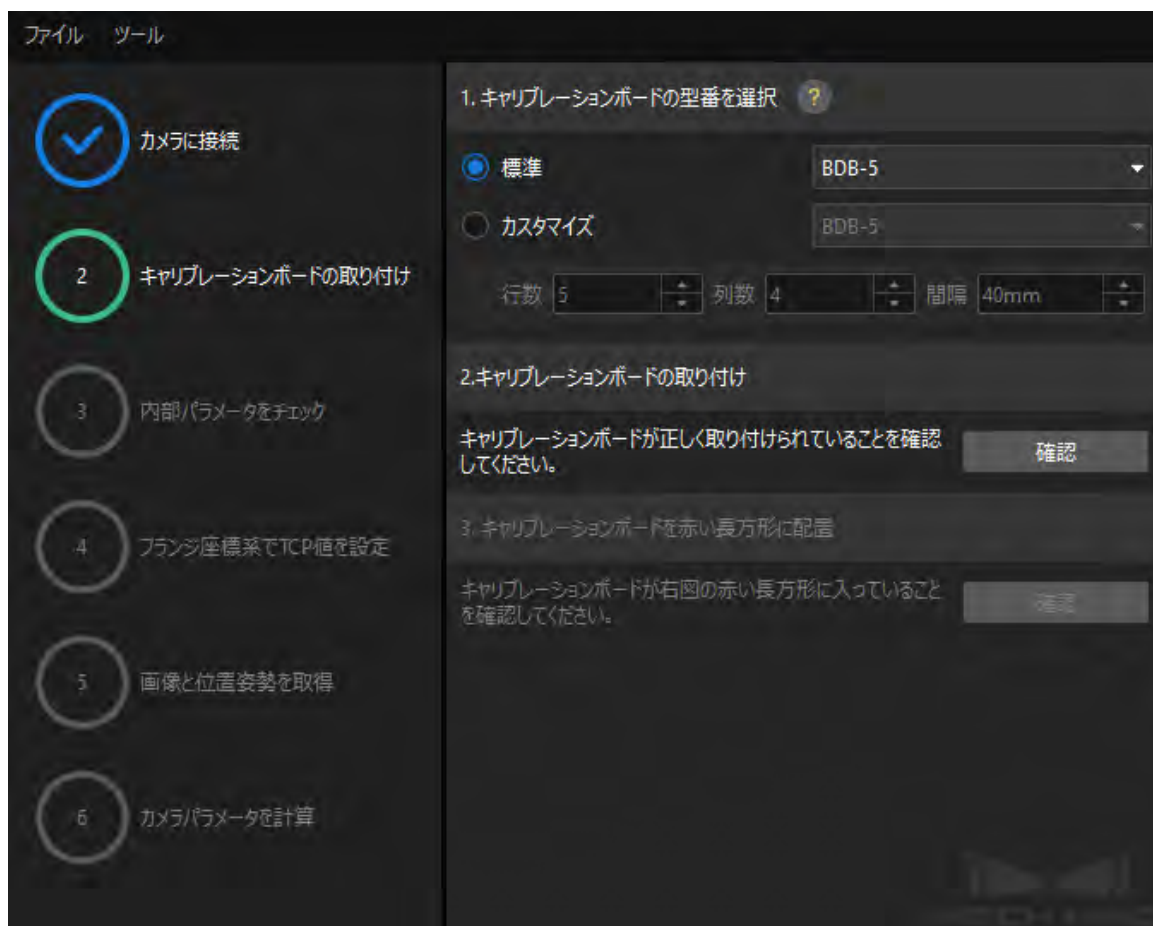
3. 画像ビューでは、取得した2D画像と深度画像が要件を満たしていることを確認してから、下部にある[次へ]をクリックします。



取得した画像が要件を満たしていない場合、Mech-Eye Viewerを起動し、[カメラの2Dと3D露光パラメータを調整](#)して再取得する必要があります。

キャリブレーションボードの取り付け

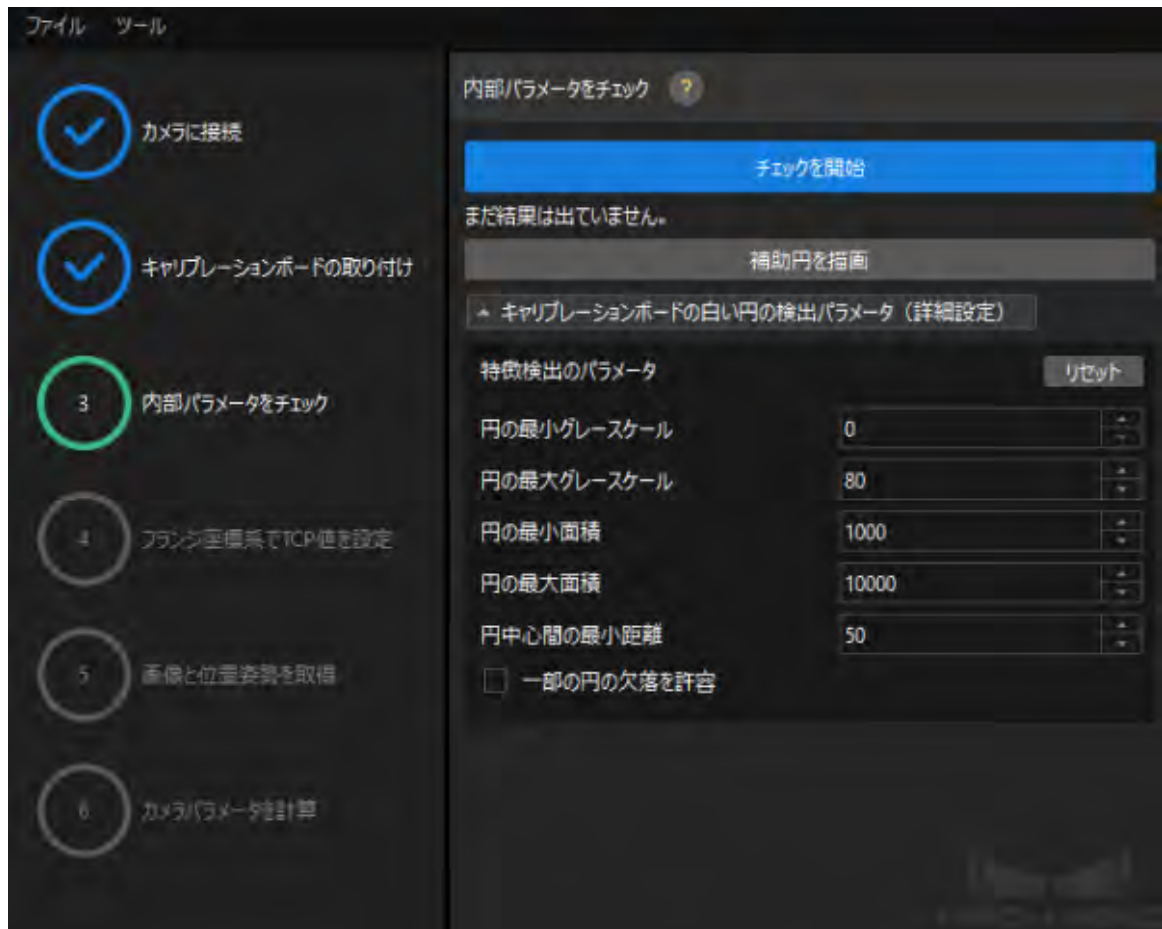
1. キャリブレーションボードの取り付け手順で、**1.キャリブレーションボードの型番を選択**で標準を選択し、キャリブレーションボードに貼られているラベルに従って型番を選択します。



2. キャリブレーションボードが作業平面の中心に配置されていることを確認してから、**2.キャリブレーションボードの取り付け**で[確認]をクリックします。
3. キャリブレーションボードがカメラの視野中心（赤い長方形）にあることを確認してから、**3.キャリブレーションボードを赤い長方形に配置**で[確認]をクリックします。
4. キャリブレーションボードに関するすべての操作が完了したら、下部にある[次へ]をクリックします。

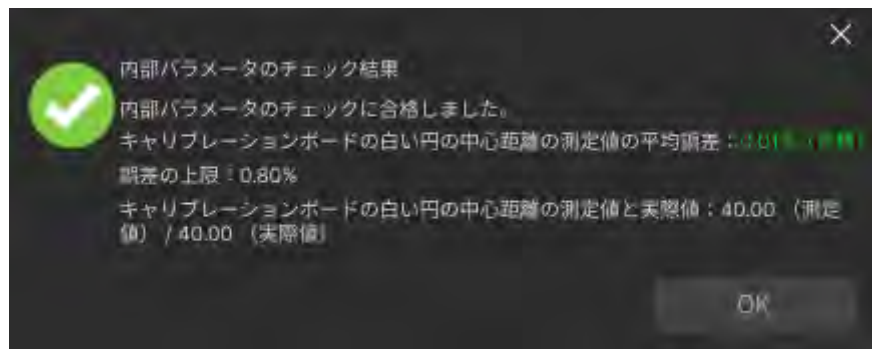
カメラの内部パラメータをチェック

1. 内部パラメータをチェック手順で[チェックを開始]をクリックします。



2. 内部パラメータのチェック結果を確認します。

- カメラの内部パラメータが合格した場合、[OK]をクリックして[次へ]をクリックします。



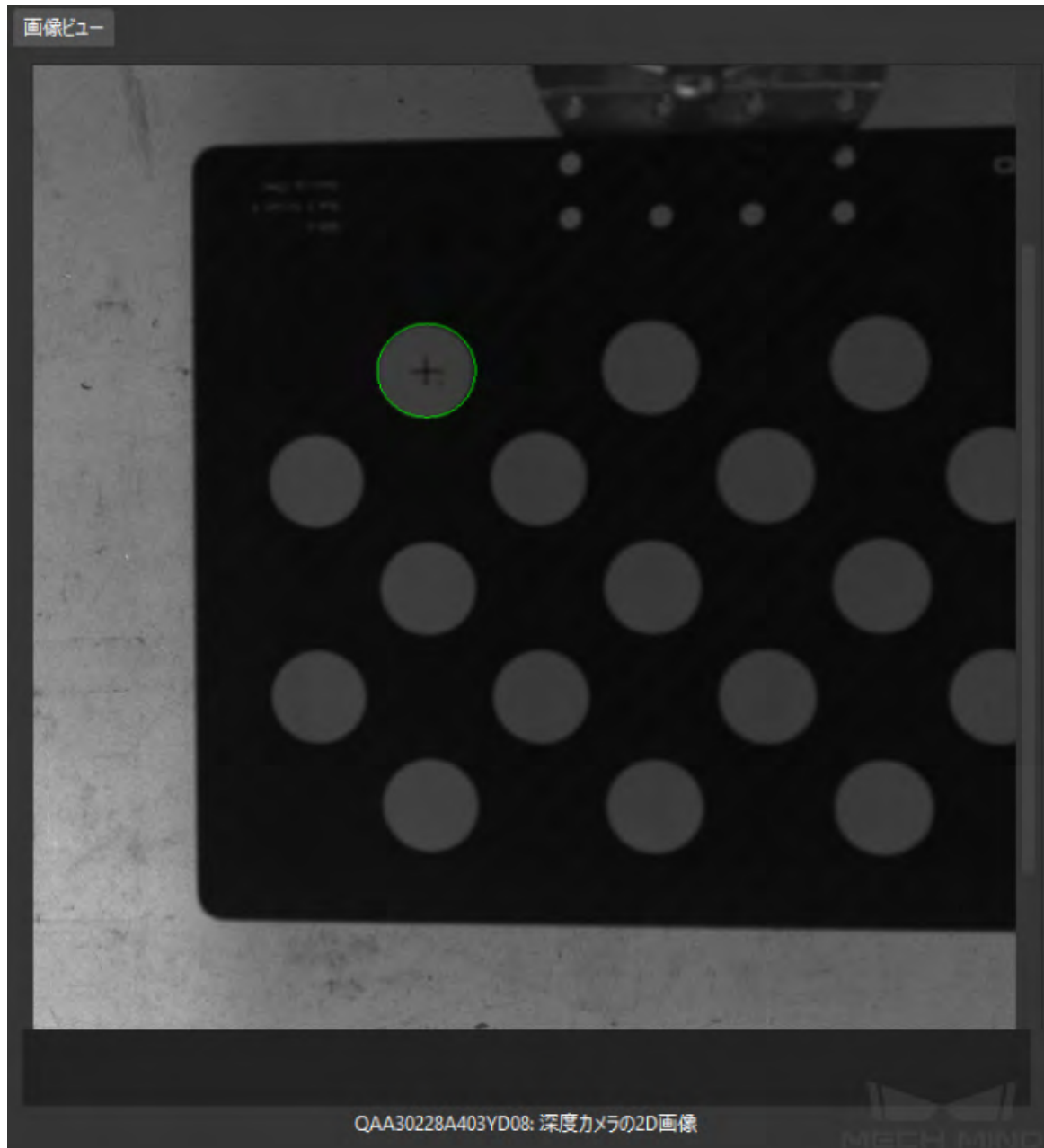
- カメラの内部パラメータが合格しなかった場合、補助円を描画するか、キャリブレーションボードの白い円の検出パラメータを手動で編集してから、[再度チェック]をクリックします。

補助円を描画

- 補助円を描画する場合、[補助円を描画]をクリックします。
- 右側の画像ビューパネルで、キャリブレーションボードの画像を右クリックし、**ウィンドウ**

に合わせるのチェックを外します。その後、**Ctrl** キーを押しながらスクロールホイールをドラッグして画像のサイズを適切な大きさに変更します。

3. キャリブレーションボードの白い円の十字型の中心にマウスポインタを合わせて、マウスの左ボタンを押し、補助円をキャリブレーションボードの白い円に完全に含ませてから離します。



4. **[再度チェック]** をクリックし、カメラ内部パラメータのチェックが合格したことを確認します。

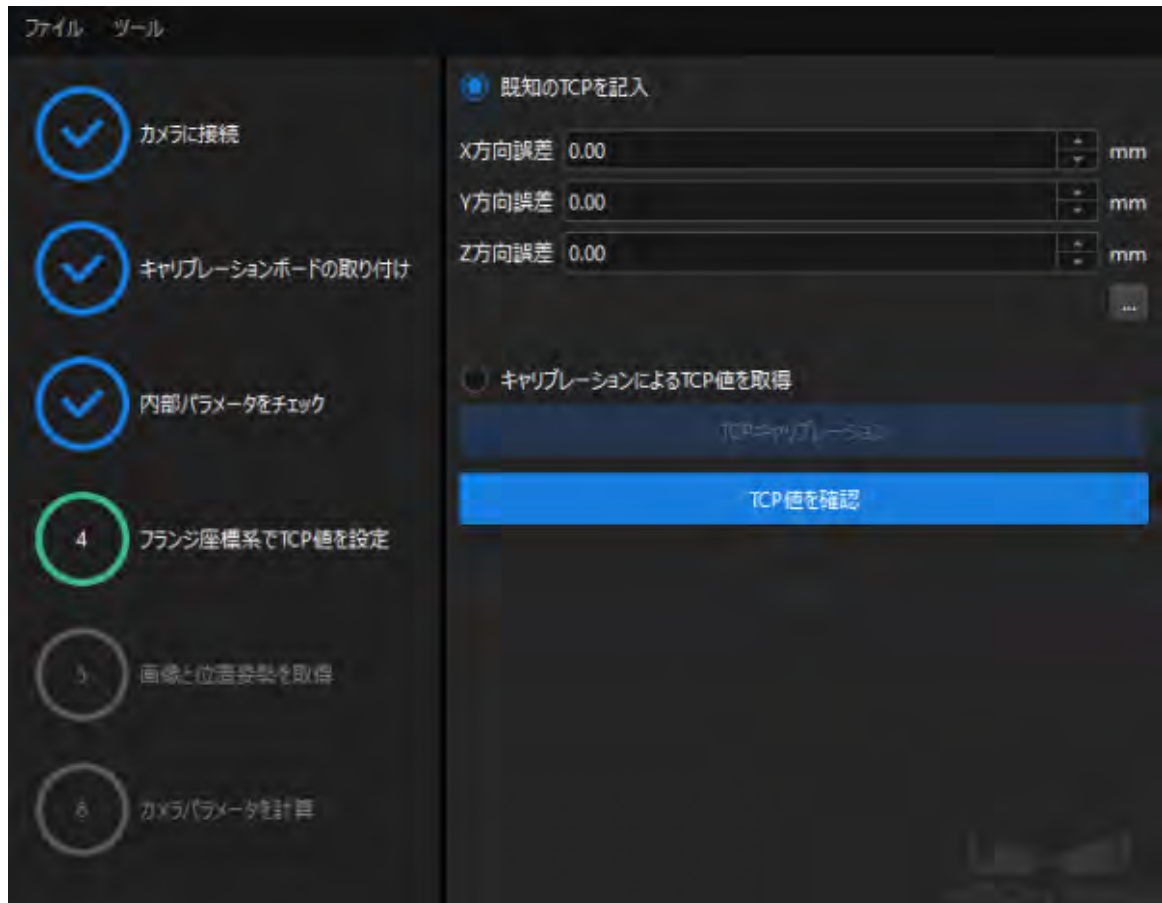
検出パラメータを手動で調整

検出パラメータを手動で調整する場合、**[キャリブレーションボードの白い円の検出パラメータ (詳細設定)]** をクリックし、実際の状況に応じてパラメータ値を変更します。

それでもキャリブレーションボードの白い円が検出されない場合、現場の環境に応じてカメラの関連パラメータを調整する必要があります。カメラパラメータの調整については、[パラメータ調整](#)をご参照ください。

フランジ座標系でTCP値を設定

1. フランジ座標系でTCP値を設定手順で、既知のTCPを記入を選択します。




2. [TCP値を確認]をクリックします。
3. 確認後、下部にある[次へ]をクリックします。




TCPが未知の場合、**キャリブレーションによるTCP値を取得**を選択して[**TCPキャリブレーション**]をクリックします。TCPキャリブレーションツールを使用してTCP値を計算できます。

画像と位置姿勢を取得

1. **画像と位置姿勢を取得**手順で、**画像を保存**にチェックを入れます。
2. TCP先端がキャリブレーションボードのポイント1の十字型の中心にタッチするようにロボットを移動させ、ティーチペンダントでのロボットのフランジ位置姿勢を読み取ります。
3. ポイント1の  をクリックし、表示される**ロボットのフランジ位置姿勢を入力**画面でロボットのフランジ位置姿勢を入力してから、[OK]をクリックします。



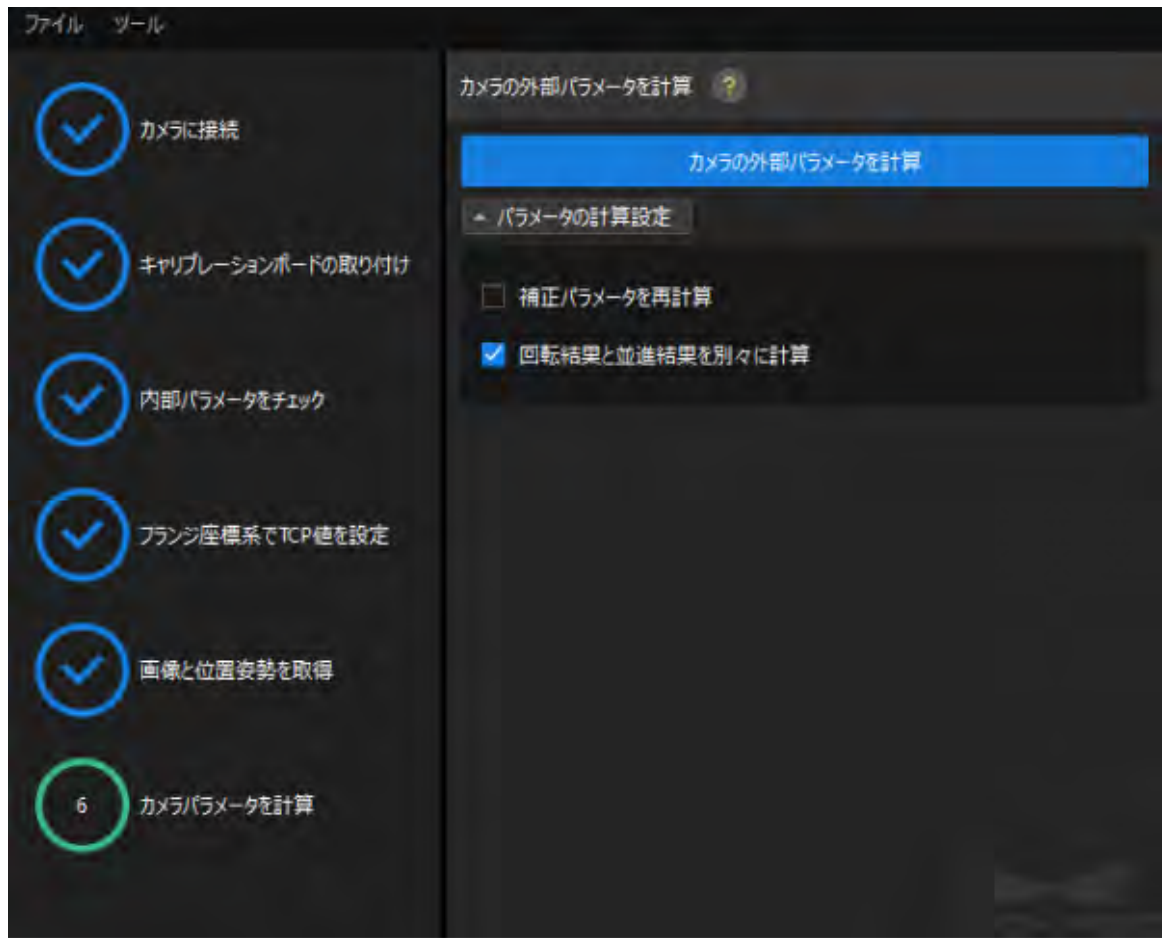
4. 上記の手順をクリックし、TCP先端がポイント2とポイント3での十字型の中心にタッチし、ロボットのフランジ位置姿勢を入力します。
5. キャリブレーションボードの  をクリックし、キャリブレーションボードの画像を取得します。
6. [データを更新]をクリックし、下部にある[次へ]をクリックします。



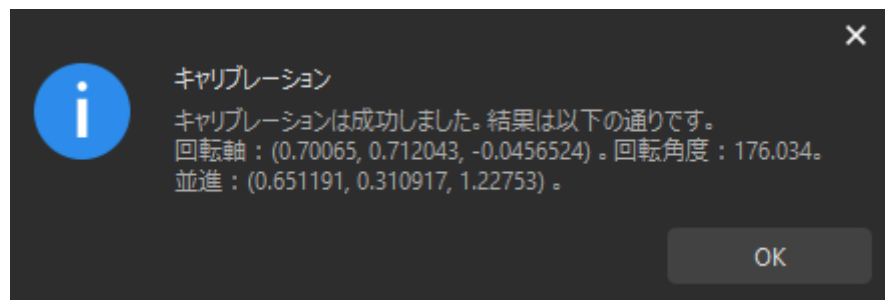
キャリブレーションでは、外部パラメータを計算するために、少なくとも3つの共線でないポイントをタッチする必要があります。

カメラパラメータを計算

1. カメラパラメータを計算手順で、「パラメータの計算設定」を展開し、**回転結果と並進結果**を別々に計算にチェックを入れてから、[**カメラの外部パラメータを計算**]をクリックします。



2. キャリブレーションが正常に実行されたことを示すダイアログで、[OK]をクリックします。



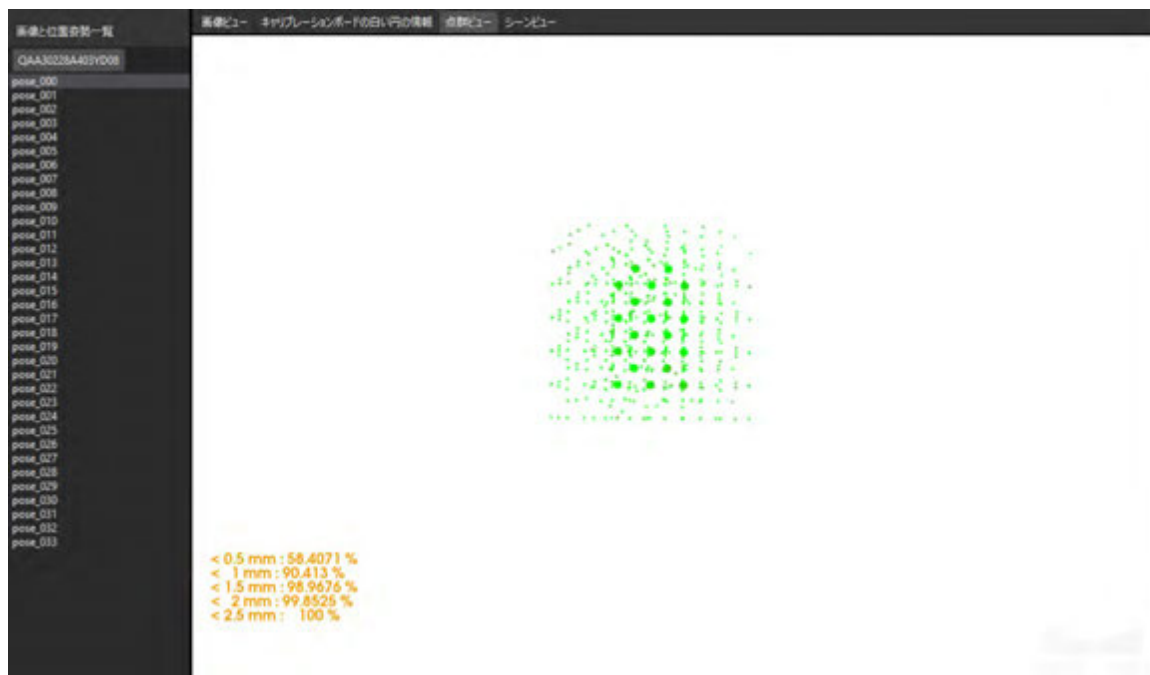
3. 右側の 点群ビュー パネルでキャリブレーションの誤差点群が表示されます。



誤差点群は、各キャリブレーション位置姿勢におけるキャリブレーションボードの白い円の実際の値と計算値との偏差を示すために使用されます。

4. キャリブレーション精度が要件を満たしていることを確認します。

キャリブレーション精度は、100%を占める誤差の値を求めることで、おおよそ判断することができます。例えば、下図は2.5 mm以下の精度を示しています。



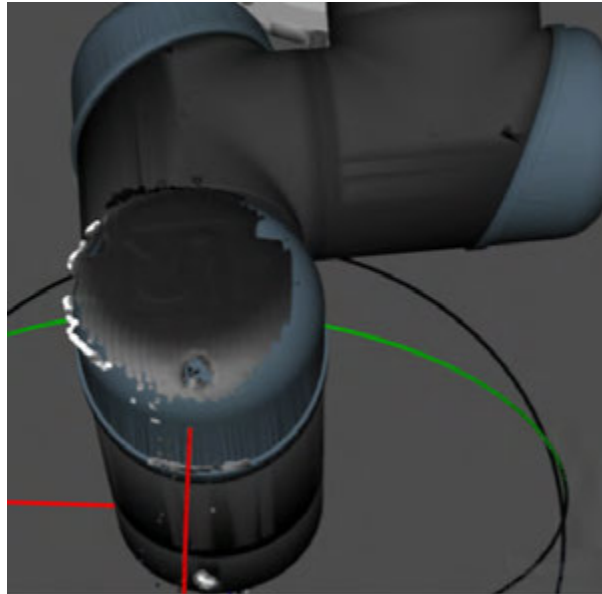
キャリブレーション精度を向上させるには、[キャリブレーション結果の分析](#)をご参照ください。

キャリブレーション結果の確認と保存

カメラの外部パラメータを計算した後、シーンビューでロボット点群とロボットモデルの点群との重なりを確認してキャリブレーション結果を大まかに判断します。具体的な流れは以下の通りです。

1. ロボットアームをカメラの視野範囲内に移動させます。
2. カメラパラメータを計算手順で[外部パラメータを再計算]をクリックします。クリックすると、カメラが撮影を行います。
3. シーンビューに切り替え、ロボットの点群とロボットモデルの重なりを確認します。

ロボットの点群がロボットモデルとほぼ一致すれば、キャリブレーション結果が使用可能です。



1. ただし、ロボットモデルはロボット実機の外観と完全に一致しない場合があり、外部パラメータの微調整の基準とすることができないことに注意してください。
2. シーンビューに加え、Mech-Vizでロボットの点群とロボットモデルとの重なりを確認できます。

4. 下部にある[保存]をクリックし、ポップアップする **キャリブレーションファイルを保存** 画面で[OK]をクリックします。すると、カメラのキャリブレーション結果は、プロジェクトの「calibration」ディレクトリに自動的に保存されます。

これで、キャリブレーション手順が完了しました。

4.4.7. TCPタッチ法を使用する手動キャリブレーション (Eye in Hand)

本節では、カメラがEye in Hand (EIH)方式で取り付けられている場合にTCPタッチ法で手動キャリブレーションを実行する方法について説明します。

事前準備

以下は、ロボットハンド・アイ・キャリブレーションを実行する前に必要な事前準備です。

- Mech-Mindビジョンシステムを構築します。
- キャリブレーションに必要なものを準備します。
- キャリブレーションボードの点群画像生成を調整します。
- ロボットと通信の設定を完了します。

ビジョンシステムを構築

Mech-Mindビジョンシステムの構築については、「ビジョンシステムを構築」節をご参照ください。

ハンド・アイ・キャリブレーションには、Mech-Eye Viewer、Mech-Center、Mech-VisionおよびMech-Vizが必要です。これらのソフトウェアがインストールされ、最新バージョンにアップグレードされていることを確認してください。

キャリブレーションに必要なものを準備

カメラがEye in Hand方式で取り付けられている場合、手動キャリブレーションを実行するにはキャリブレーションボードとTCP先端の使用が必要です。

キャリブレーションボードについては、以下の要件があります。

- キャリブレーションボードの白い円がはっきりと見えること、破損や曲げ変形がないことを確認してください。
- キャリブレーションボードを作業平面の中心に配置してください。

TCP先端をロボットフランジに取り付ける必要があります。ロボットフランジに取り外しができないハンドがある場合、TCP先端をハンドに直接固定します。

また、キャリブレーションを実行する前に、ロボットをキャリブレーションの開始位置に移動させてください。



キャリブレーションボードの点群画像生成を調整

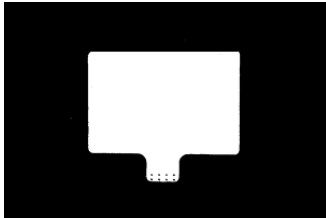
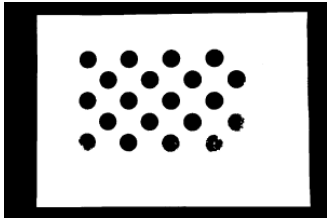
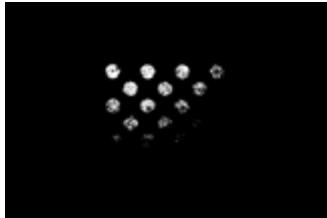
1. Mech-Eye Viewerを起動し、カメラのパラメータを調整します。
2. 2Dパラメータを調整して、2D画像にキャリブレーションボードがはっきり見られ、露光過度や露光不足などがないようにします。
3. 3Dパラメータを調整して、キャリブレーションボードの白い円の点群が完全に見られるようにします。点群の変動範囲を小さくするために、**点群後処理**の中の**点群平滑化**と**ノイズ除去**を**Normal**に設定することを推奨します。



現場の環境光が複雑な場合、2D画像や点群への影響を軽減するために、遮光・補光することを推奨します。

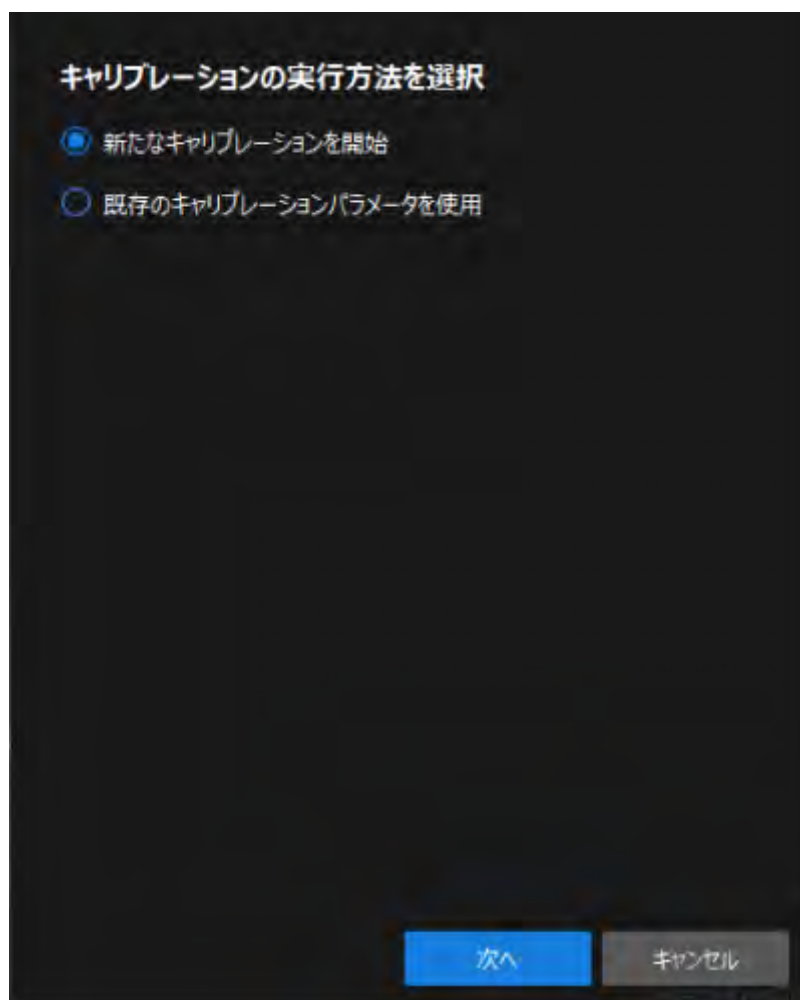
4. 上記の手順を完了することで、キャリブレーションボードの点群の画像品質が要件を満たしていることを確認します。

	正常	露光過度	露光不足
2D画像			

	正常	露光過度	露光不足
点群			

キャリブレーション前の設定

1. Mech-Visionを起動し、ツールバーで[カメラキャリブレーション（標準モード）]をクリックします。すると、キャリブレーション前の設定画面が表示されます。
2. キャリブレーションの実行方法を選択画面で、新たなキャリブレーションを開始を選択し、[次へ]をクリックします。



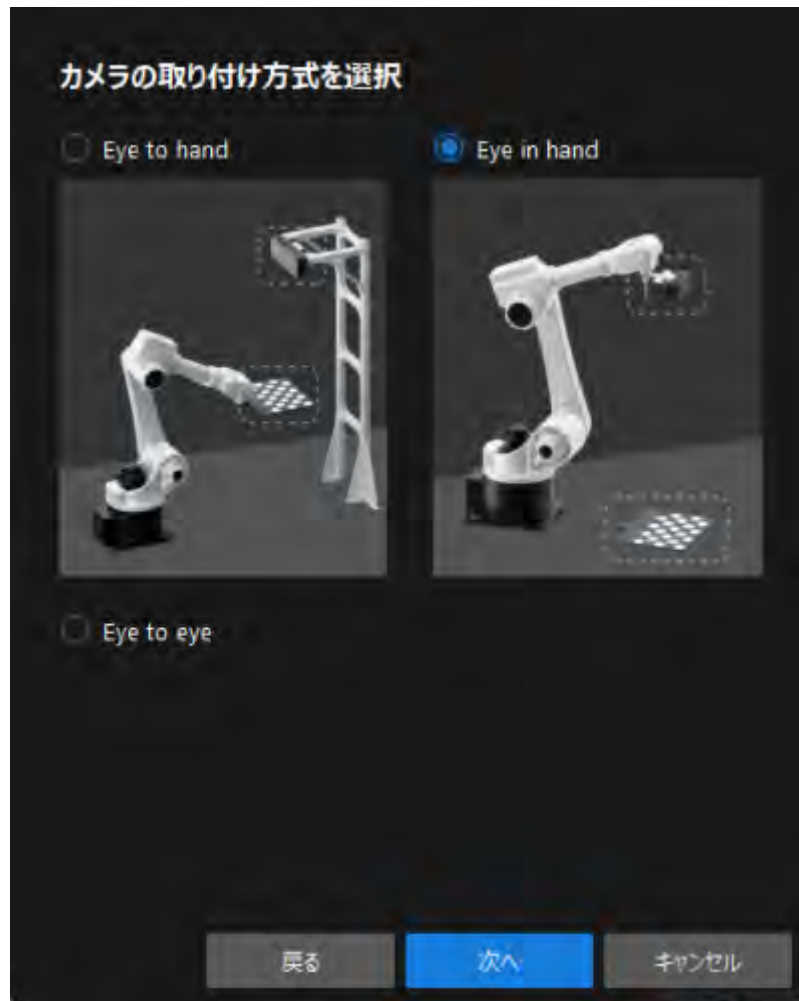
3. キャリブレーションのタスクを選択画面で、ドロップダウンリストからその他のロボットのハンド・アイ・キャリブレーションを選択し、必要に応じてロボットのオイラー角のタイプを指定し、ロボットの座標系を選択してから[次へ]をクリックします。



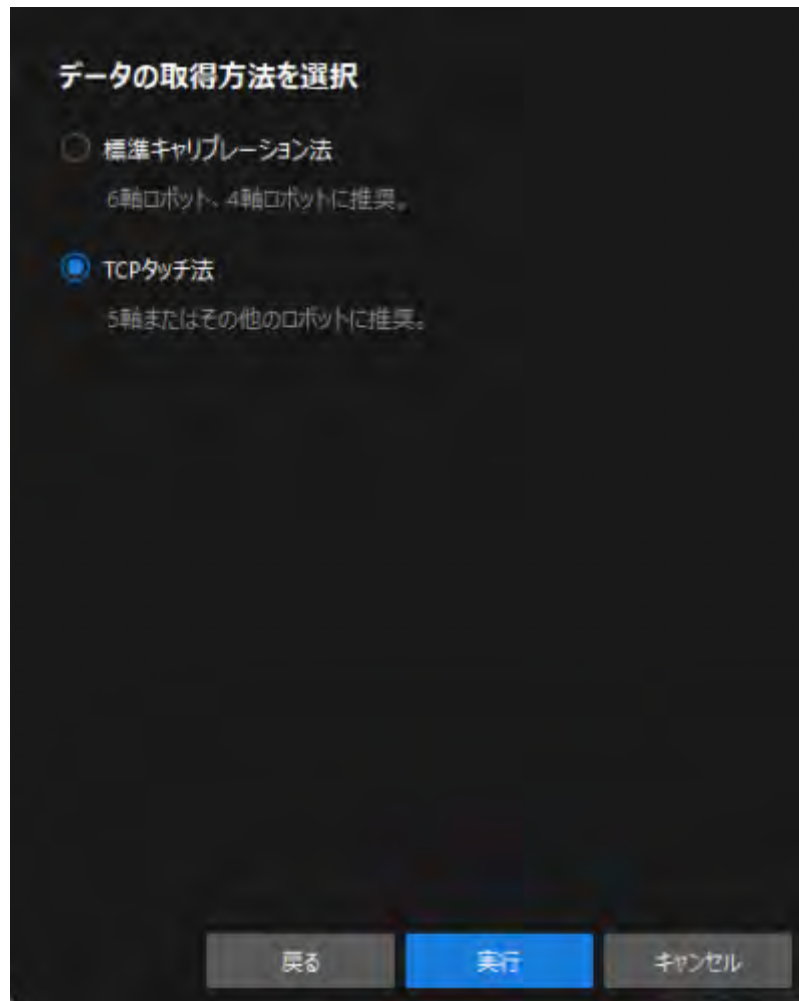
4. キャリブレーションするためのロボットを選択画面で、実際の状況に応じて6軸ロボット、4軸ロボットまたは5軸、または他のタイプのロボットを選択してから、[次へ]をクリックします。



5. カメラ取り付け方式を選択画面で、**Eye in hand**にチェックを入れ、[次へ]をクリックします。




6. データの取得方法を選択 画面で、TCPタッチ法を選択してから、[実行]をクリックします。すると、キャリブレーション（Eye in Hand）画面が表示されます。



これで、キャリブレーション前の設定が完了し、キャリブレーションに進むことになります。

キャリブレーションの実行手順

カメラ接続

1. **カメラに接続** 手順で、**検出されたカメラ** から接続するカメラを選択して  をクリックするか、ダブルクリックして接続します。



2. カメラ接続後、[一回キャプチャ]または[連続キャプチャ]をクリックします。



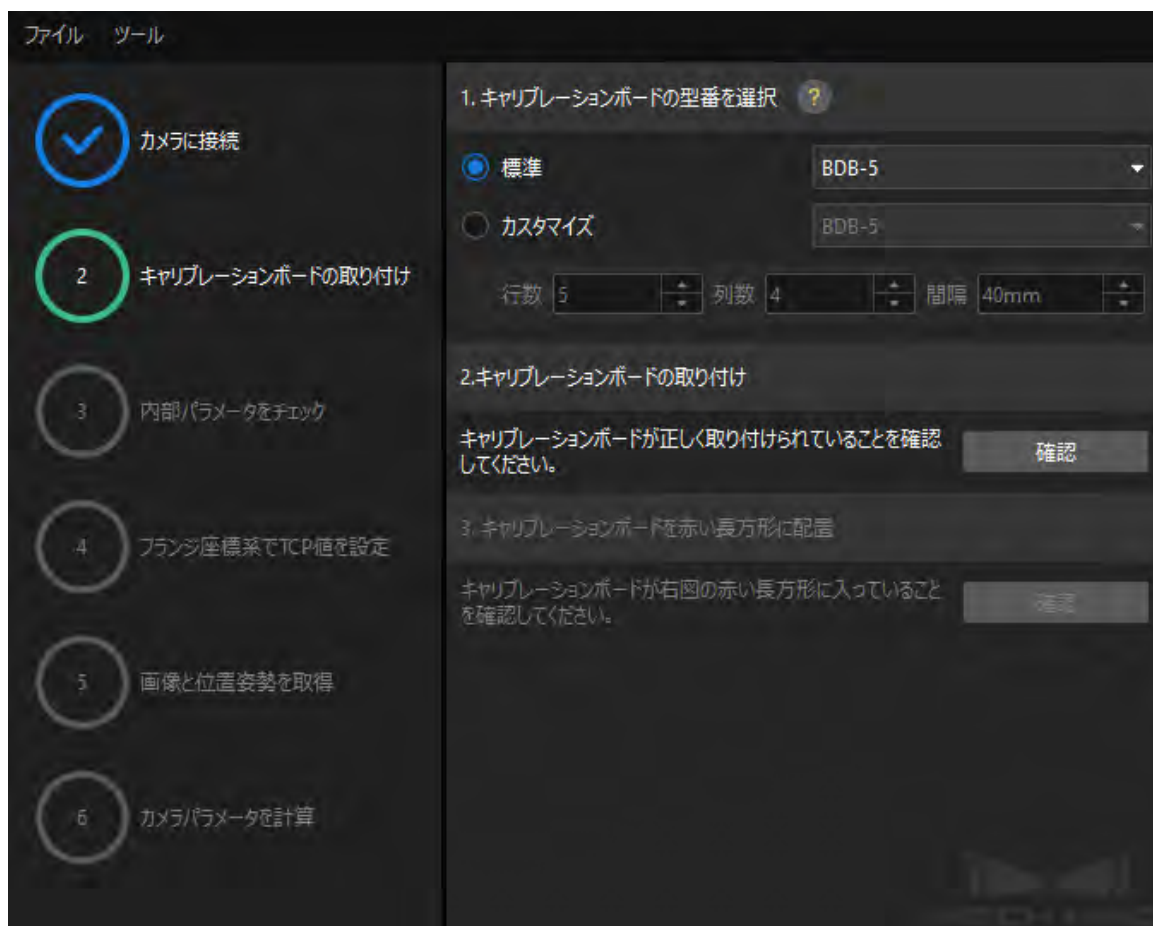
3. 画像ビューでは、取得した2D画像と深度画像が要件を満たしていることを確認した上で[次へ]をクリックします。



取得した画像が要件を満たしていない場合、Mech-Eye Viewerを起動し、[カメラの2Dと3D露光パラメータを調整](#)して再取得する必要があります。

キャリブレーションボードの取り付け

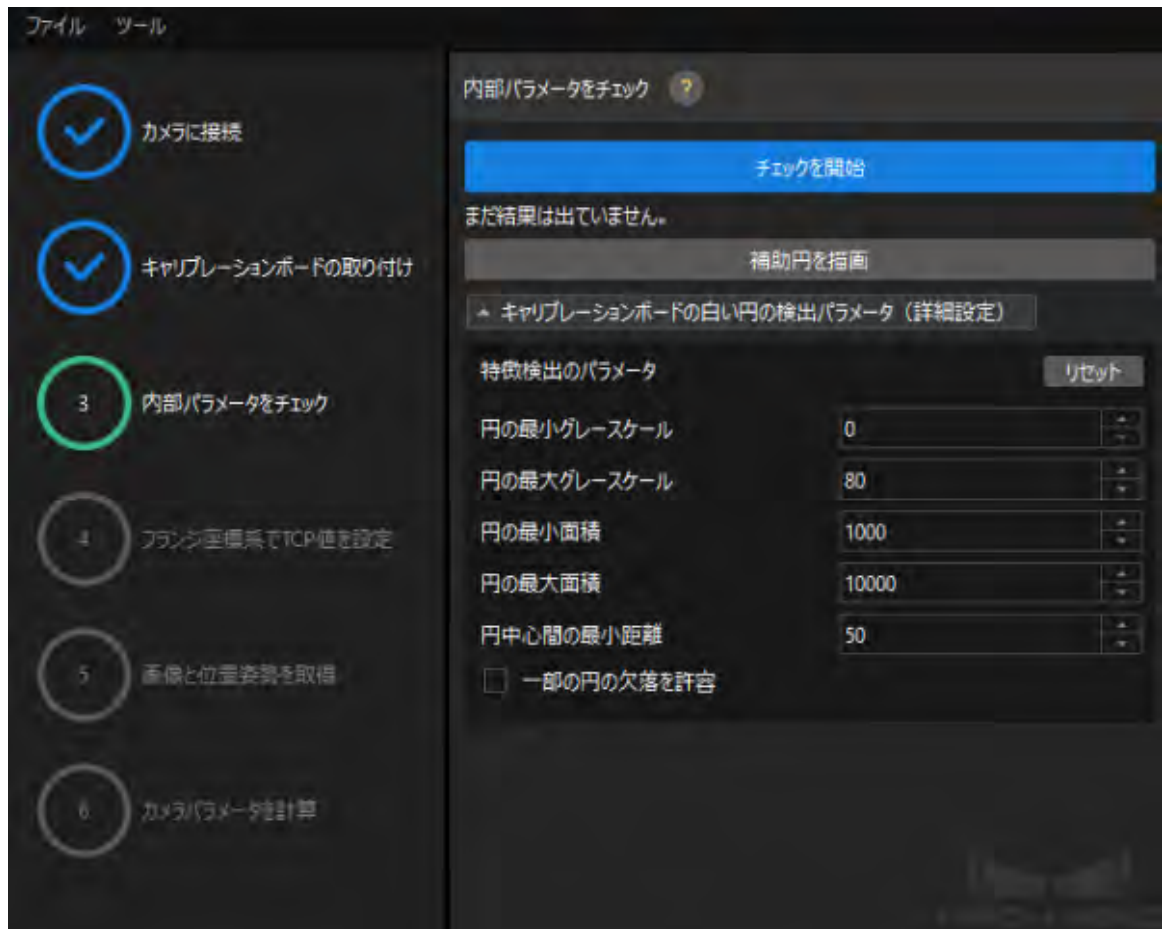
1. キャリブレーションボードの取り付け手順で、**1. キャリブレーションボードの型番を選択**で標準を選択し、キャリブレーションボードに貼られているラベルに従って型番を選択します。



2. キャリブレーションボードが作業平面の中心に配置されていることを確認してから、**2. キャリブレーションボードの取り付け**で[確認]をクリックします。
3. キャリブレーションボードがカメラの視野中心（赤い長方形）にあることを確認してから、**3. キャリブレーションボードを赤い長方形に配置**で[確認]をクリックします。
4. キャリブレーションボードに関するすべての操作が完了したら、下部にある[次へ]をクリックします。

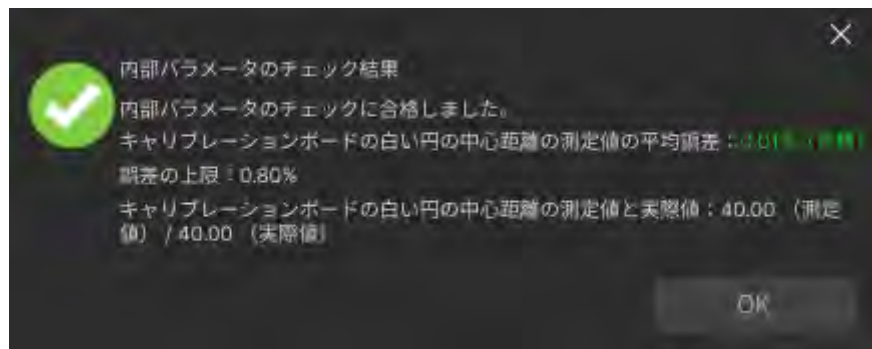
カメラの内部パラメータをチェック

1. 内部パラメータをチェック手順で[チェックを開始]をクリックします。



2. 内部パラメータのチェック結果を確認します。

- カメラの内部パラメータが合格した場合、[OK]をクリックして[次へ]をクリックします。



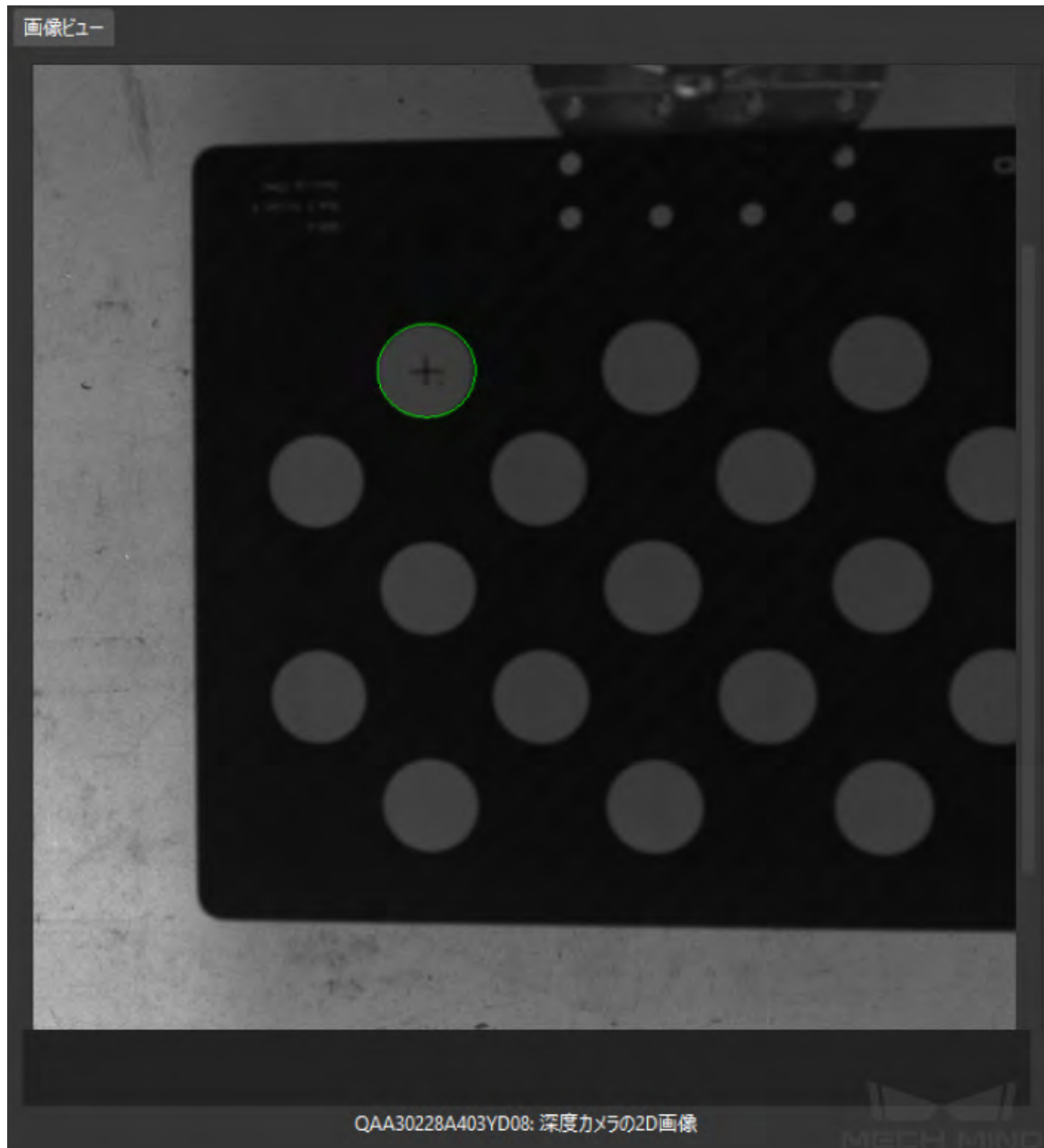
- カメラの内部パラメータが合格しなかった場合、補助円を描画するか、キャリブレーションボードの白い円の検出パラメータを手動で編集してから、[再度チェック]をクリックします。

補助円を描画

- 補助円を描画する場合、[補助円を描画]をクリックします。
- 右側の画像ビューパネルで、キャリブレーションボードの画像を右クリックし、**ウィンドウ**

に合わせるのチェックを外します。その後、**Ctrl** キーを押しながらスクロールホイールをドラッグして画像のサイズを適切な大きさに変更します。

3. キャリブレーションボードの白い円の十字型の中心にマウスポインタを合わせて、マウスの左ボタンを押し、補助円をキャリブレーションボードの白い円に完全に含ませてから離します。



4. **[再度チェック]** をクリックし、カメラ内部パラメータのチェックが合格したことを確認します。

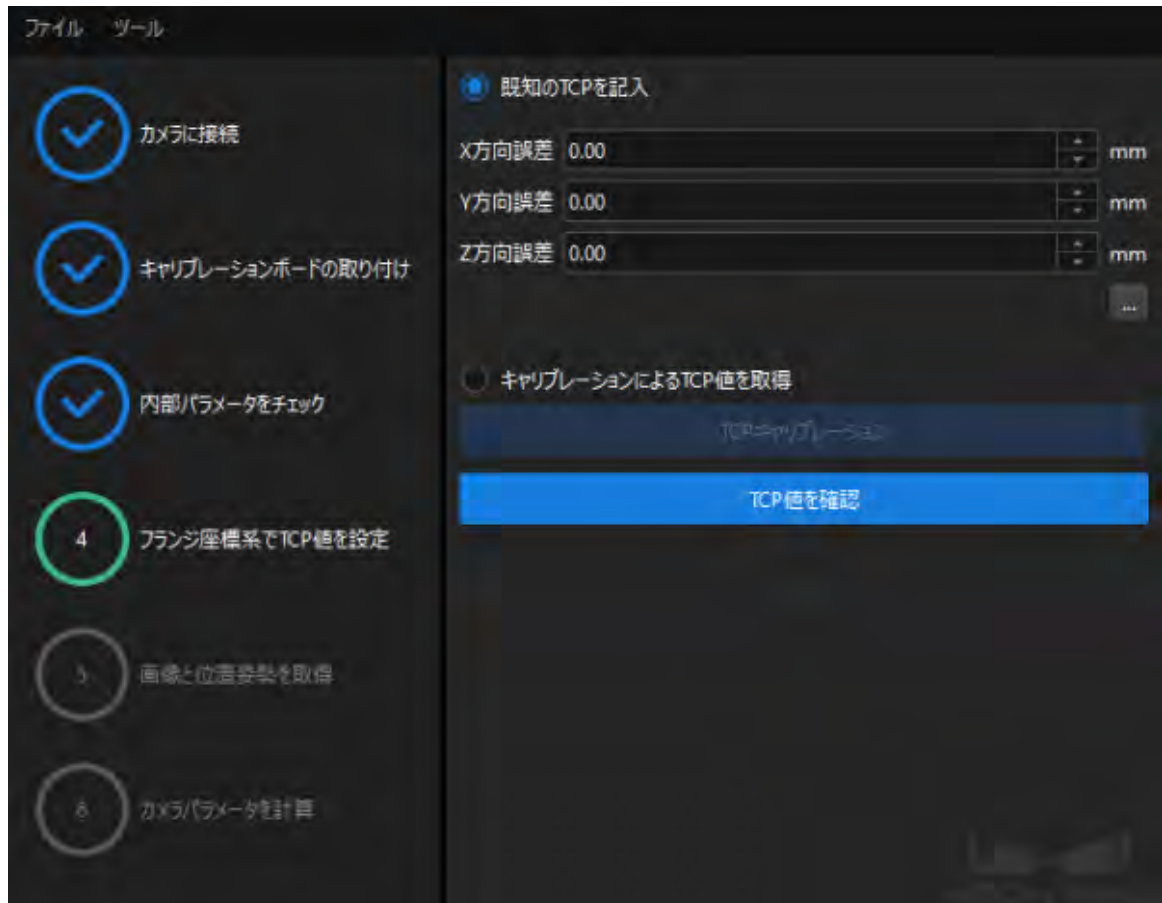
検出パラメータを手動で調整

検出パラメータを手動で調整する場合、**[キャリブレーションボードの白い円の検出パラメータ (詳細設定)]** をクリックし、実際の状況に応じてパラメータ値を変更します。

それでもキャリブレーションボードの白い円が検出されない場合、現場の環境に応じてカメラの関連パラメータを調整する必要があります。カメラパラメータの調整については、[パラメータ調整](#)をご参照ください。

フランジ座標系でTCP値を設定

1. フランジ座標系でTCP値を設定手順で、**既知のTCPを記入**を選択します。




2. **[TCP値を確認]**をクリックします。
3. 確認後、下部にある**[次へ]**をクリックします。




TCPが未知の場合、**キャリブレーションによるTCP値を取得**を選択して**[TCPキャリブレーション]**をクリックします。TCPキャリブレーションツールを使用してTCP値を計算できます。

画像と位置姿勢を取得

1. **画像と位置姿勢を取得**手順で、「画像を保存」にチェックを入れます。
2. TCP先端がキャリブレーションボードのポイント1の十字型の中心にタッチするようにロボットを移動させ、ティーチペンダントでのロボットのフランジ位置姿勢を読み取ります。
3. ポイント1の  をクリックし、表示される**ロボットのフランジ位置姿勢を入力**画面でロボットのフランジ位置姿勢を入力してから、**[OK]**をクリックします。



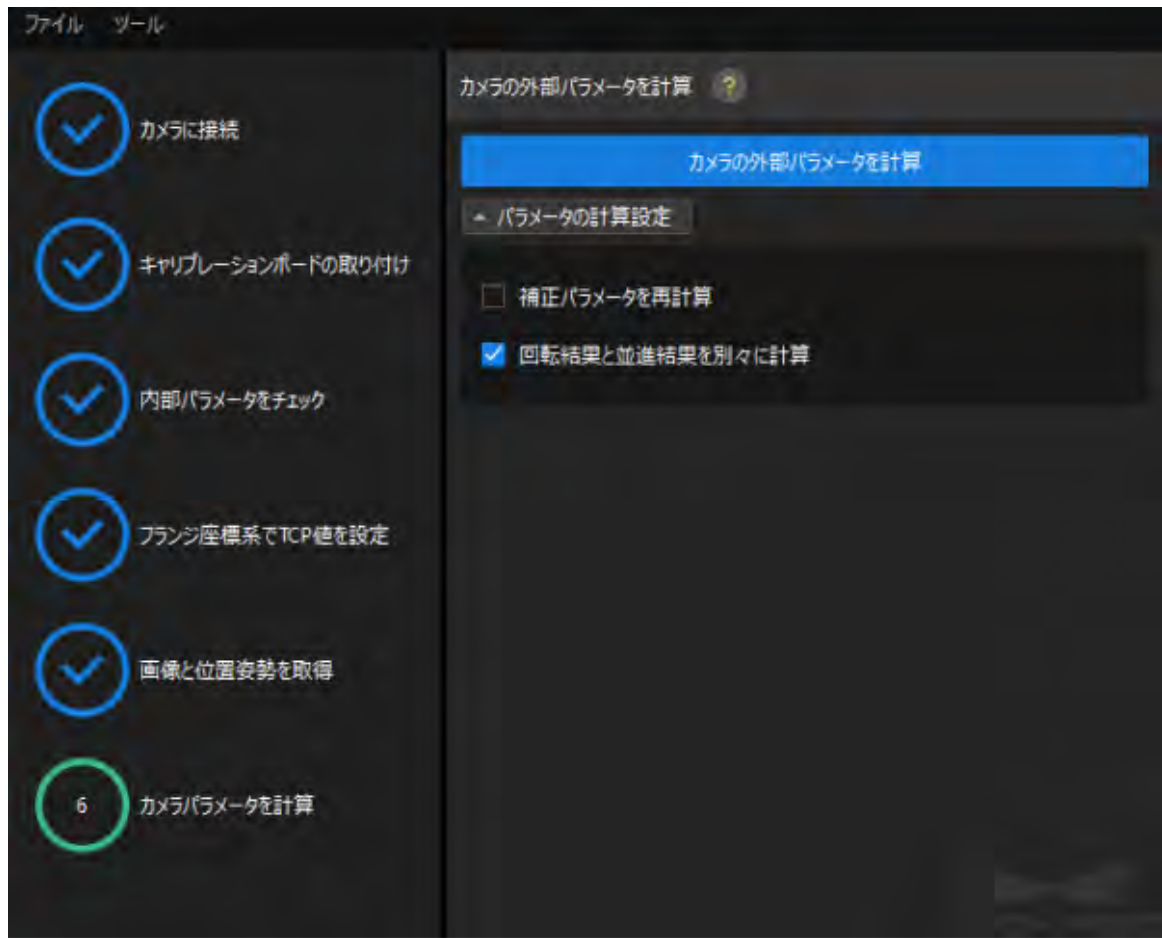
4. 上記の手順をクリックし、TCP先端がポイント2とポイント3での十字型の中心にタッチし、ロボットのフランジ位置姿勢を入力します。
5. キャリブレーションボードの  をクリックし、キャリブレーションボードの画像を取得します。
6. [データを更新]をクリックし、下部にある[次へ]をクリックします。



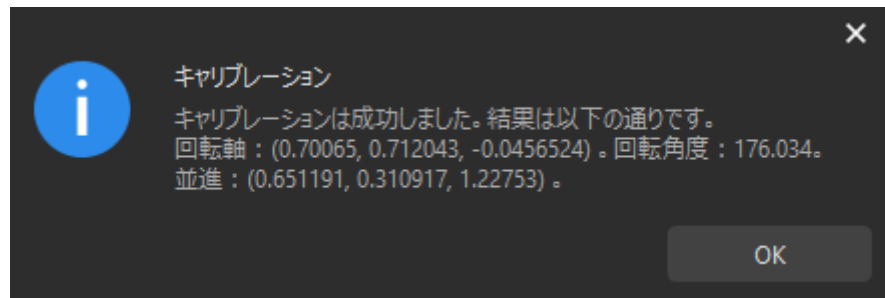
キャリブレーションでは、外部パラメータを計算するために、少なくとも3つの共線でないポイントをタッチする必要があります。

カメラパラメータを計算

1. カメラパラメータを計算手順で、「パラメータの計算設定」を展開し、**回転結果と並進結果**を別々に計算にチェックを入れてから、[**カメラの外部パラメータを計算**]をクリックします。



2. キャリブレーションが正常に実行されたことを示すダイアログで、[OK]をクリックします。



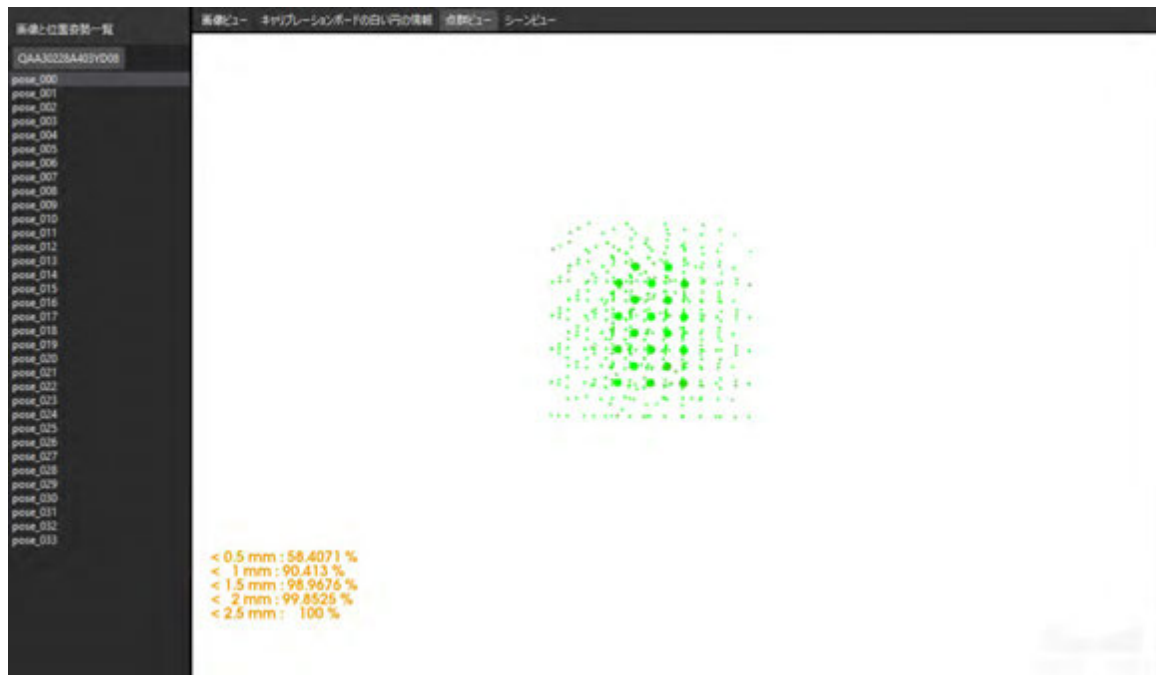
3. 右側の点群ビューパネルでキャリブレーションの誤差点群が表示されます。



誤差点群は、各キャリブレーション位置姿勢におけるキャリブレーションボードの白い円の実際の値と計算値との偏差を示すために使用されます。

4. キャリブレーション精度が要件を満たしていることを確認します。

キャリブレーション精度は、100%を占める誤差の値を求めることで、おおよそ判断することができます。例えば、下図は2.5 mm以下の精度を示しています。



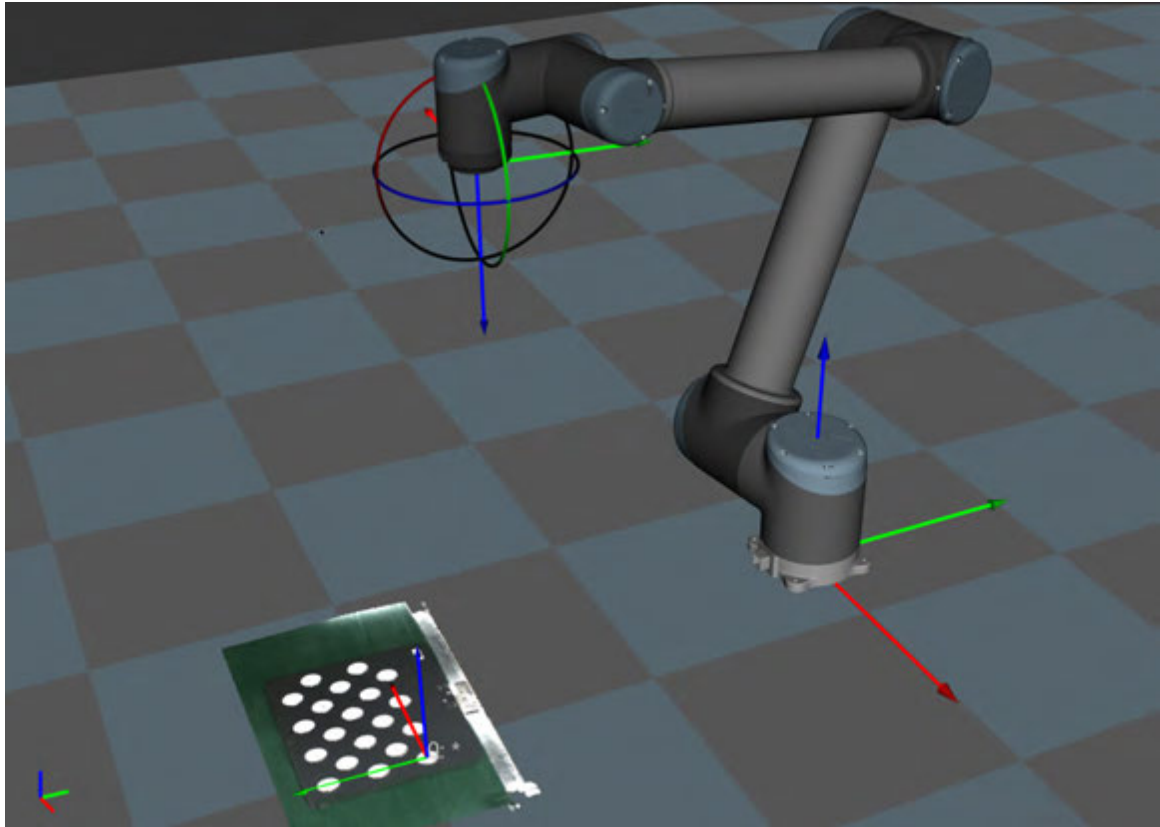
キャリブレーション精度を向上させるには、[キャリブレーション結果の分析](#)をご参照ください。

キャリブレーション結果の確認と保存

カメラの外部パラメータを計算した後、シーンビューで固定点に対するキャリブレーションボードの点群（キャリブレーションボードの白い円の十字中心）のずれを確認することで、キャリブレーション結果を大まかに判断することができます。詳細は以下の通りです。

1. キャリブレーションボードを固定位置に配置します。
2. Mech-Vizを起動し、固定点を追加し、それをキャリブレーションボードの白い円の1つの十字の中心に合わせます。
 - a. ステップライブラリから「移動」ステップを選択して、それをプロジェクト編集エリアにドラッグします。
 - b. このステップをクリックして選択し、ステップパラメータで**目標点タイプ**を「」に設定します。それがキャリブレーションボードの白い円のいずれかの十字中心と一致するように、位置姿勢のX/Y/Z座標値を調整します。
3. ロボットを制御してカメラの位置姿勢を数回変更し、キャリブレーションの**カメラパラメータを計算**手順で**[外部パラメータを再計算]**ボタンをクリックします。クリックすると、カメラが撮影を行います。
4. シーンビューで、キャリブレーションボードの点群が固定点に対して大きくなずれがあるかどうかを確認します。

キャリブレーションボードの点群が固定点に対して大きくなずれがない場合、キャリブレーション結果が使用可能であることを示します。



1. EIHの場合、ロボットが接続されていないとロボットの位置姿勢はリアルタイムで取得できないため、シーンビューまたはMech-Vizに表示される正確な点群を確認するには、画像撮影時のロボットの位置姿勢を記入する必要があります。
2. シーンビューに加え、キャリブレーションボードの点群が固定点に対して大きくずれていないかどうかをMech-Vizで確認することも可能です。

1. 下部にある[保存]をクリックし、ポップアップする**キャリブレーションファイルを保存**画面で[OK]をクリックします。すると、カメラのキャリブレーション結果は、プロジェクトの「calibration」ディレクトリに自動的に保存されます。

これで、キャリブレーション手順が完了しました。

4.4.8. 直行ロボットのハンド・アイ・キャリブレーション

本節では、直行ロボットのハンド・アイ・キャリブレーションを実行する方法について説明します。



直行ロボットのハンド・アイ・キャリブレーションの概念や外部パラメータファイルの使用については、[直行ロボットのハンド・アイ・キャリブレーションの説明](#)をご参照ください。

事前準備

以下は、ロボットハンド・アイ・キャリブレーションを実行する前に必要な事前準備です。

- Mech-Mindビジョンシステムを構築します。

- キャリブレーションに必要なものを準備します。
- キャリブレーションボードの点群画像生成を調整します。

ビジョンシステムを構築

Mech-Mindビジョンシステムの構築については、「ビジョンシステムを構築」節をご参照ください。

ハンド・アイ・キャリブレーションには、Mech-Eye Viewer、Mech-Center、Mech-VisionおよびMech-Vizが必要です。これらのソフトウェアがインストールされ、最新バージョンにアップグレードされていることを確認してください。

キャリブレーションに必要なものを準備

以下のものがが必要です。

- キャリブレーションボード
- TCP先端
- 巻尺

キャリブレーションボードについては、以下の要件があります。

- キャリブレーションボードの白い円がはっきりと見えること、破損や曲げ変形がないことを確認してください。
- 直行ロボットのハンド・アイ・キャリブレーションでは、キャリブレーションボードを作業平面の中心に平らに配置してください。

TCP先端は直行ロボットのハンドに取り付ける必要があります。巻尺を使用して、TCP先端からフランジの中心までのX、Y、Z方向のオフセットを測定します。



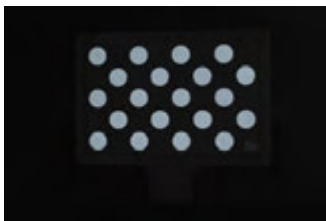


キャリブレーションボードの点群画像生成を調整

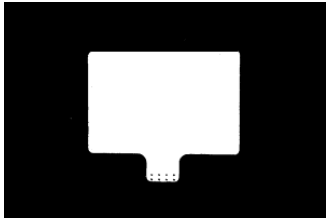
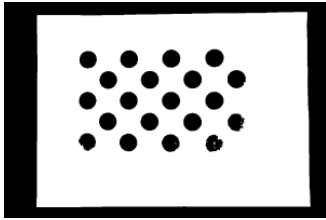
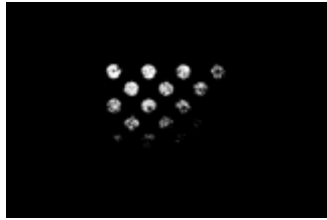
1. Mech-Eye Viewerソフトウェアを起動し、カメラのパラメータを調整します。
2. 2Dパラメータを調整して、2D画像にキャリブレーションボードがはっきり見られ、露光過度や露光不足などがないようにします。
3. 3Dパラメータを調整して、キャリブレーションボードの白い円の点群が完全に見られるようにします。点群の変動範囲を小さくするために、**点群後処理**の中の**点群平滑化**と**ノイズ除去**を**Normal**に設定することを推奨します。



現場の環境光が複雑な場合、2D画像や点群への影響を軽減するために、遮光・補光することを推奨します。

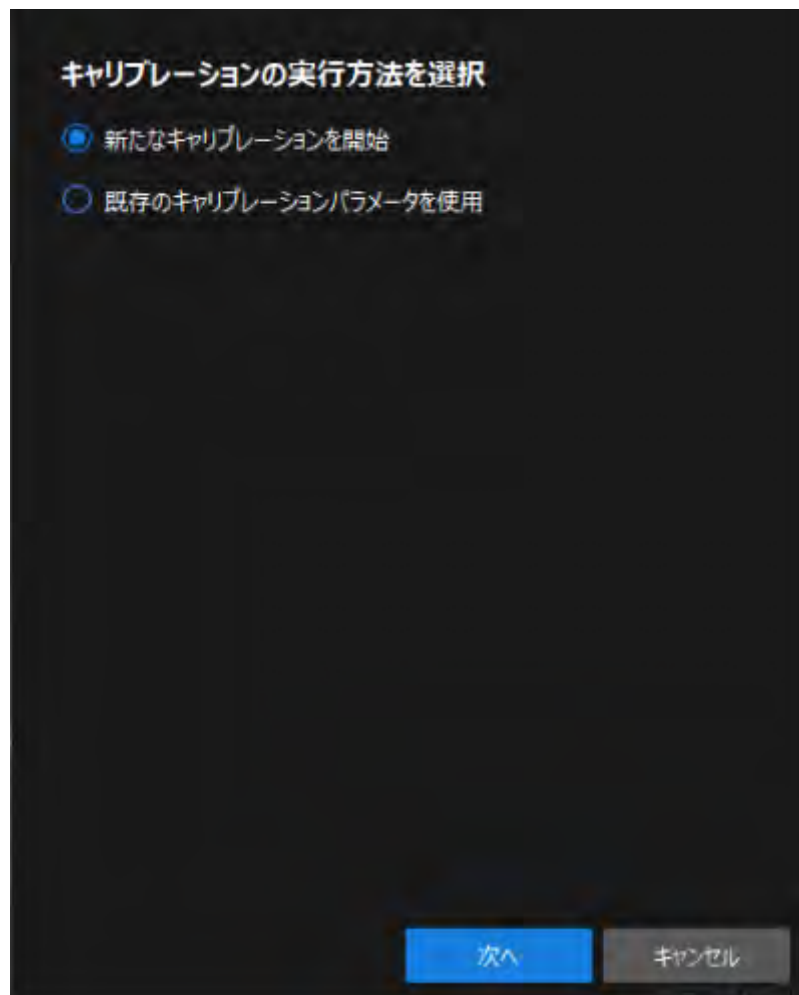
4. 上記の手順を完了することで、キャリブレーションボードの点群の画像品質が要件を満たしていることを確認します。

	正常	露光過度	露光不足
2D画像			

	正常	露光過度	露光不足
点群			

キャリブレーション前の設定

1. Mech-Visionを起動し、ツールバーで[カメラキャリブレーション（標準モード）]をクリックします。すると、**キャリブレーション前の設定画面**が表示されます。
2. **キャリブレーションの実行方法を選択画面**で、**新たなキャリブレーションを開始**を選択し、[次へ]をクリックします。



3. **キャリブレーションのタスクを選択画面**で、ドロップダウンリストから**その他のロボットのハンド・アイ・キャリブレーション**を選択し、必要に応じて**ロボットのオイラー角のタイプ**を指定し、ロボットの座標系を選択してから[次へ]をクリックします。

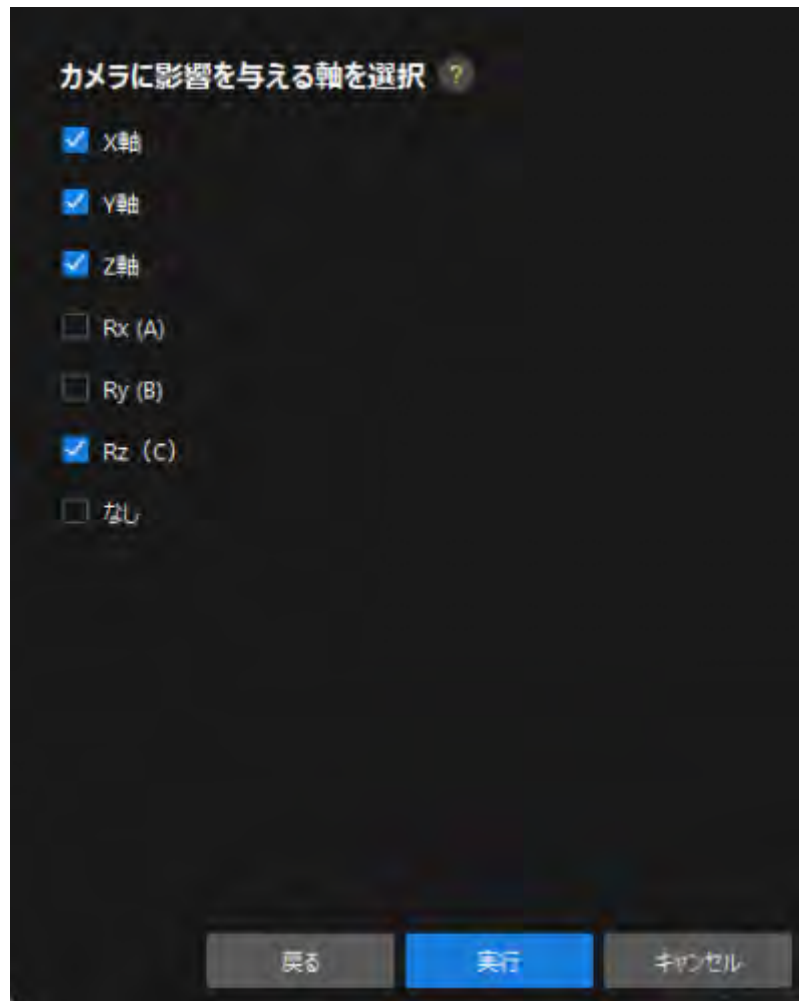


- ° ロボットの座標系タイプは、**左・右手系**を参照して確認できます。直行ロボットが左手系を使用している場合は、ここで左手系を選択してください。
- ° ロボットと通信する際、通信モジュールは送受信したロボットの位置姿勢を自動的に右手系に変換します。ロボット側からのロボット位置姿勢を受信した場合、左手座標系の位置姿勢を自動的に右手座標系の位置姿勢に変換します。ソフトウェアから出力された位置姿勢をロボット側に送信する前に、右手座標系の位置姿勢を左手座標系の位置姿勢に変換します。
- ° キャリブレーション中にロボットのフランジ位置姿勢を手動で入力する場合、Y軸の値を反転させる必要があります。

4. キャリブレーションするためのロボットを選択画面で、直行ロボット（最大6自由度に対応、XYZABC）を選択してから[次へ]をクリックします。




5. カメラに影響を与える軸を選択 画面で、実際の状況に応じて軸（X軸、Y軸、Z軸、Rzなど）を選択してから[実行]をクリックします。



キャリブレーションの実行手順

カメラ接続

1. カメラに接続手順で、検出されたカメラから接続するカメラを選択して  をクリックするか、ダブルクリックして接続します。



2. カメラ接続後、[一回キャプチャ]または[連続キャプチャ]をクリックします。



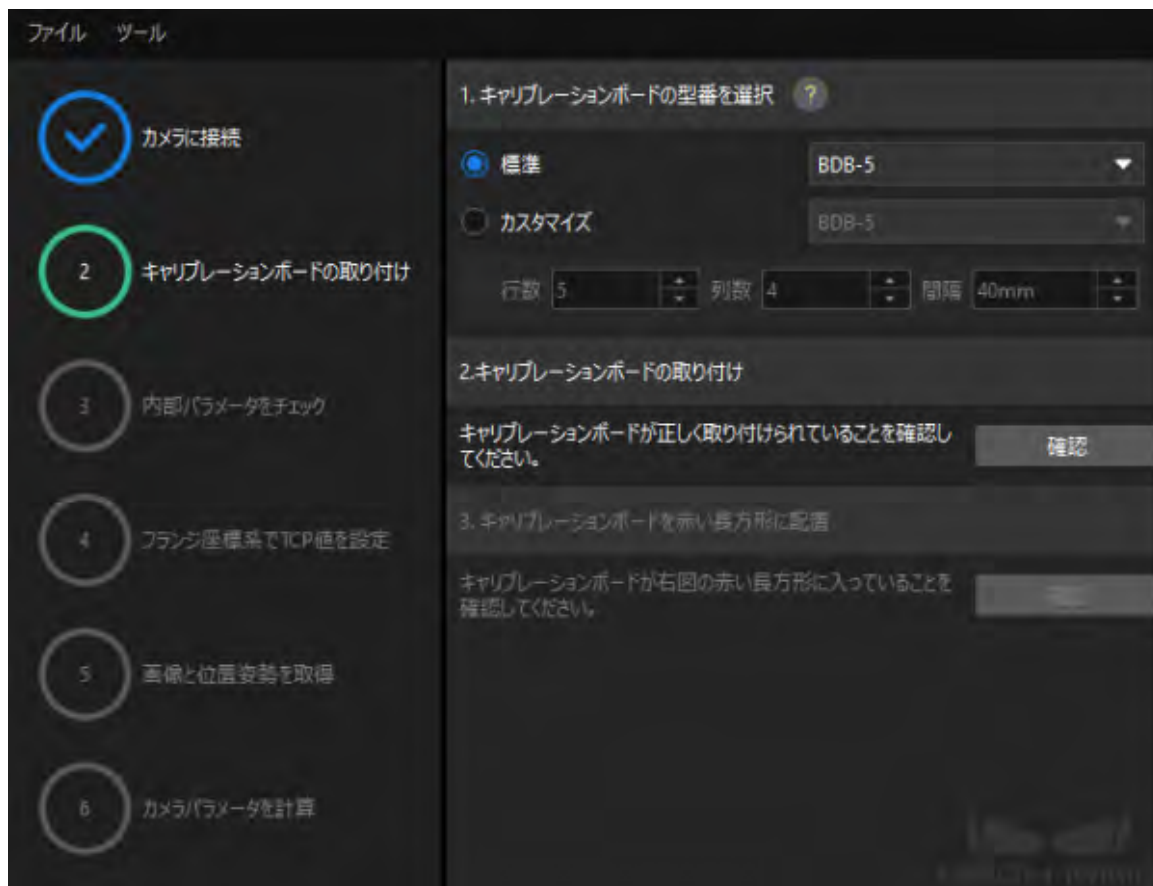
3. 画像ビューでは、取得した2D画像と深度画像が要件を満たしていることを確認してから、下部にある[次へ]をクリックします。



取得した画像が要件を満たしていない場合、Mech-Eye Viewerを起動し、[カメラの2Dと3D露光パラメータを調整](#)して再取得する必要があります。

キャリブレーションボードの取り付け

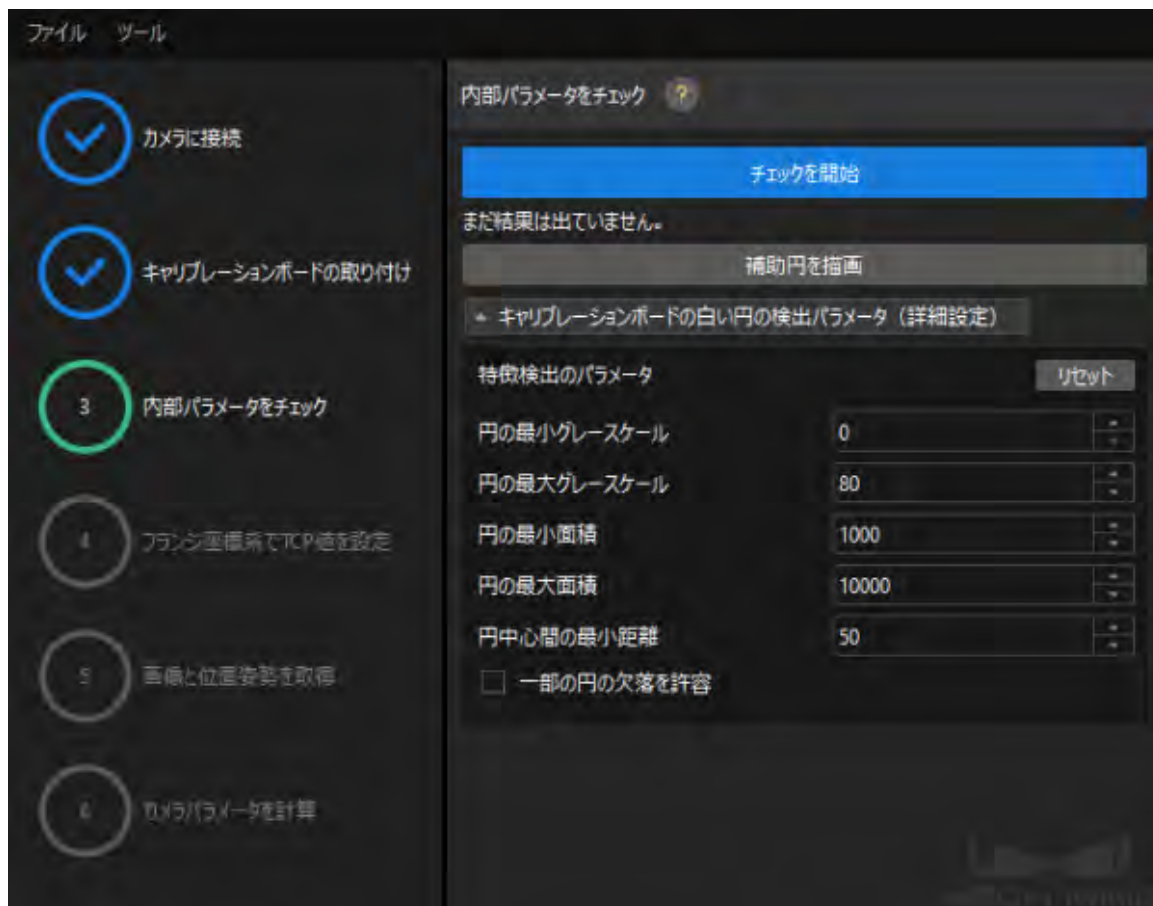
1. キャリブレーションボードの取り付け手順で、**1.キャリブレーションボードの型番を選択**で標準を選択し、キャリブレーションボードに貼られているラベルに従って型番を選択します。



2. キャリブレーションボードが作業平面の中心に配置されていることを確認してから、**2.キャリブレーションボードの取り付け**で[確認]をクリックします。
3. キャリブレーションボードがカメラの視野中心（赤い長方形）にあることを確認してから、**3.キャリブレーションボードを赤い長方形に配置**で[確認]をクリックします。
4. キャリブレーションボードに関するすべての操作が完了したら、下部にある[次へ]をクリックします。

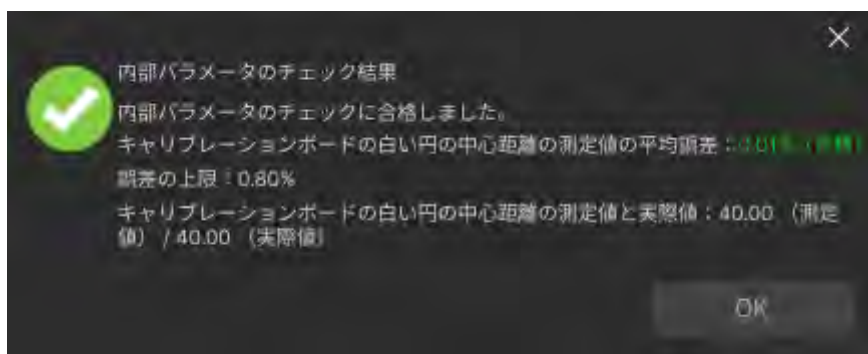
内部パラメータをチェック

1. 内部パラメータをチェック手順で[チェックを開始]をクリックします。



2. 内部パラメータのチェック結果を確認します。

- カメラの内部パラメータが合格した場合、[OK]をクリックして[次へ]をクリックします。



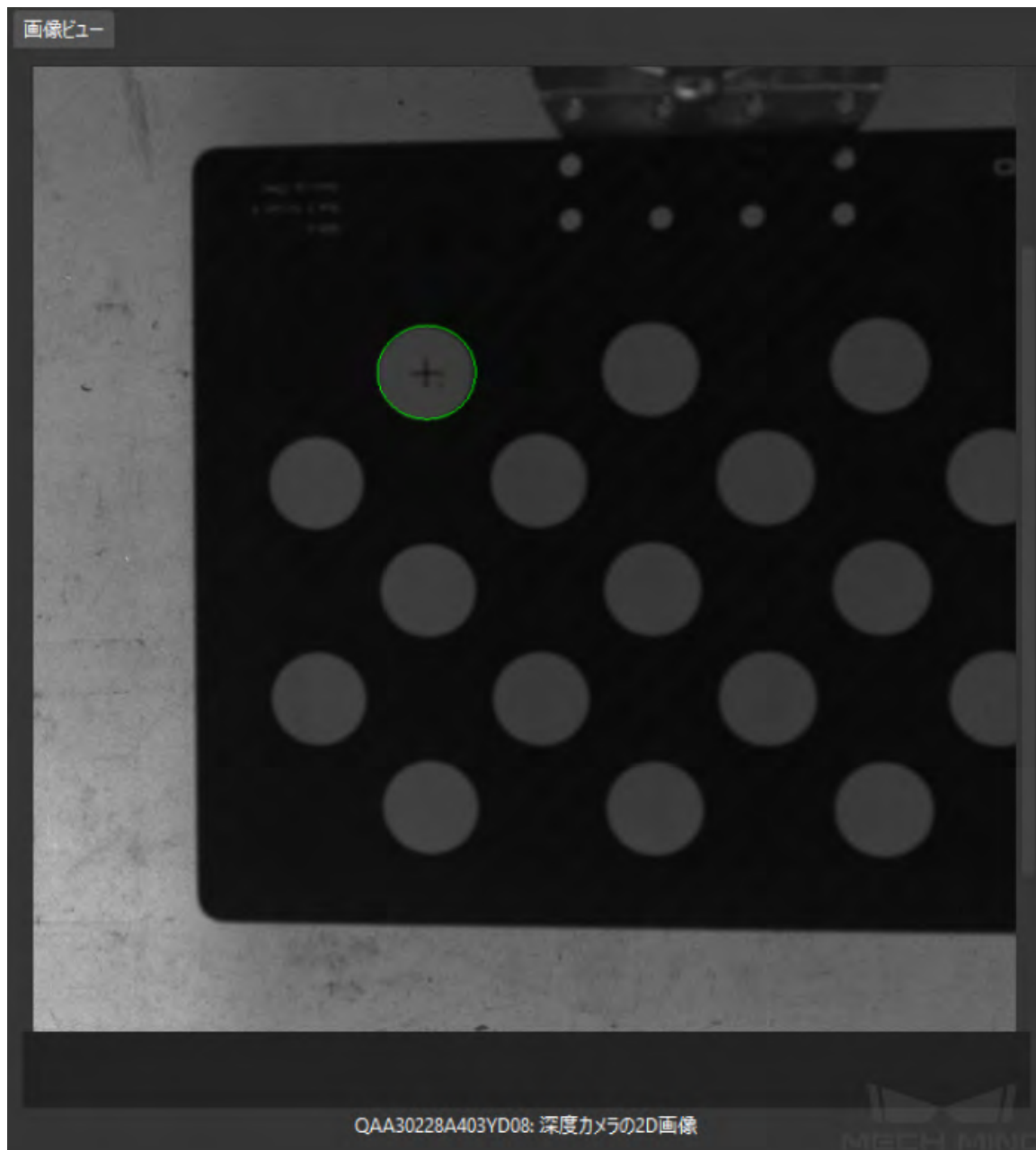
- カメラの内部パラメータが合格しなかった場合、補助円を描画するか、キャリブレーションボードの白い円の検出パラメータを手動で編集してから、[再度チェック]をクリックします。

補助円を描画

- 補助円を描画する場合、[補助円を描画]をクリックします。
- 右側の画像ビューパネルで、キャリブレーションボードの画像を右クリックし、**ウィンドウに合わせる**のチェックを外します。その後、**Ctrl** キーを押しながらスクロールホイールをド

ラッグして画像のサイズを適切な大きさに変更します。

3. キャリブレーションボードの白い円の十字型の中心にマウスポインタを合わせて、マウスの左ボタンを押し、補助円をキャリブレーションボードの白い円に完全に含ませてから離します。



4. **[再度チェック]**をクリックし、カメラ内部パラメータのチェックが合格したことを確認します。

検出パラメータを手動で調整

検出パラメータを手動で調整する場合、**[キャリブレーションボードの白い円の検出パラメータ (詳細設定)]**をクリックし、実際の状況に応じてパラメータ値を変更します。

それでもキャリブレーションボードの白い円が検出されない場合、現場の環境に応じてカメラ

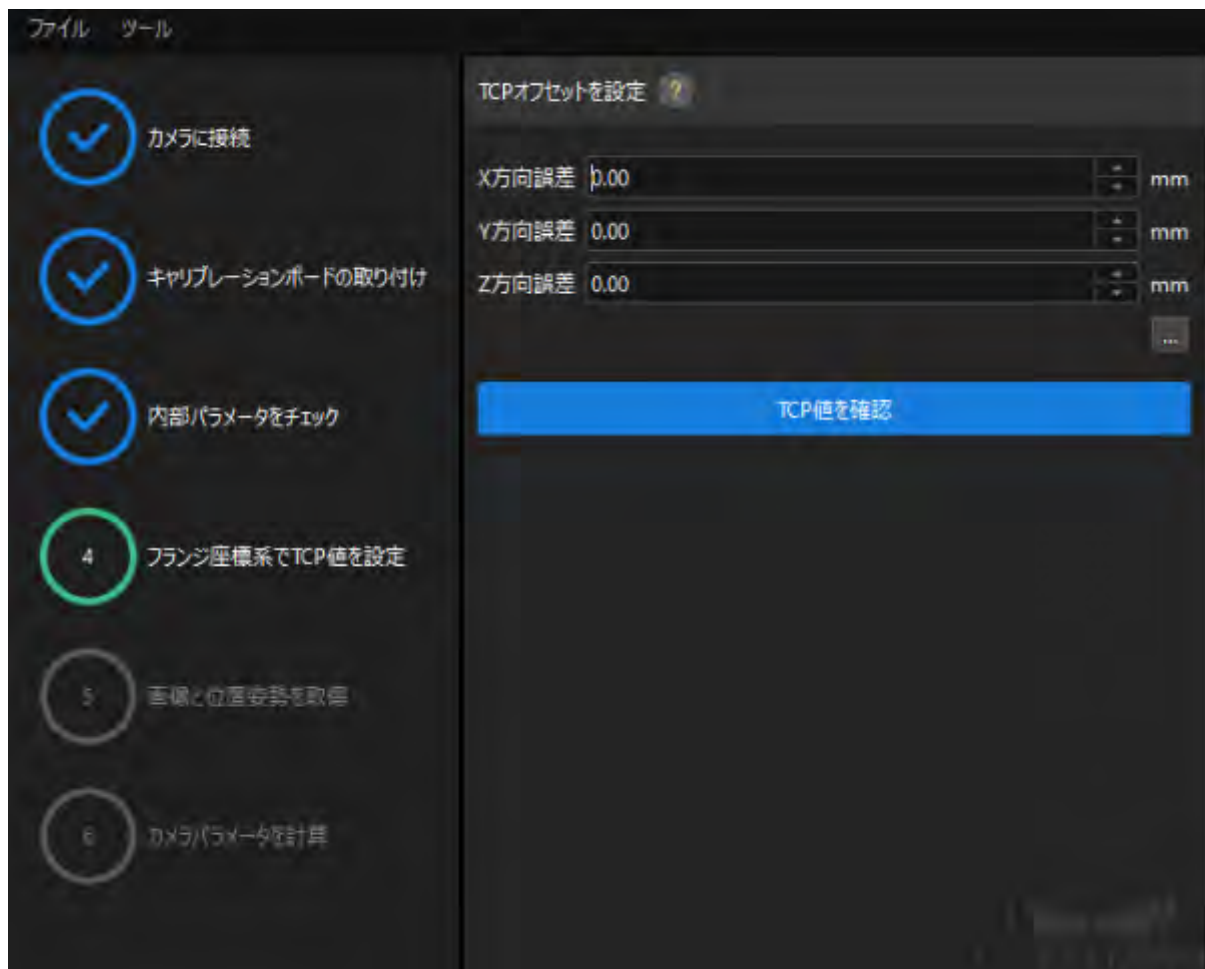
の関連パラメータを調整する必要があります。カメラパラメータの調整については、[パラメータ調整](#)をご参照ください。

フランジ座標系でTCP値を設定




この手順を実行する前に、巻尺を使用してTCP先端からフランジ中心までのX、Y、Z方向のオフセット値（ミリメートル単位）を測定する必要があります。

フランジ座標系でTCP値を設定 手順で、測定したX、Y、Z方向のオフセット値をX、Y、Zのテキストボックスに入力し、[**TCP値を確認**]をクリックしてから、下部にある[**次へ**]をクリックします。



左手系を使用する直行ロボットのの場合、入力前にY軸の値を反転させる必要があります。

画像と位置姿勢を取得


1. ハンドのTCP先端がキャリブレーションボードのポイント1での十字型の中心にタッチするように直行ロボットを移動させ、ティーチペンダントでのロボットのフランジ位置姿勢を読み取ります。
2. **画像と位置姿勢を取得**手順で、ポイント1の  をクリックし、表示されるロボットのフランジ位置姿勢を入力画面でロボットのフランジ位置姿勢を入力してから、[**OK**]をクリック

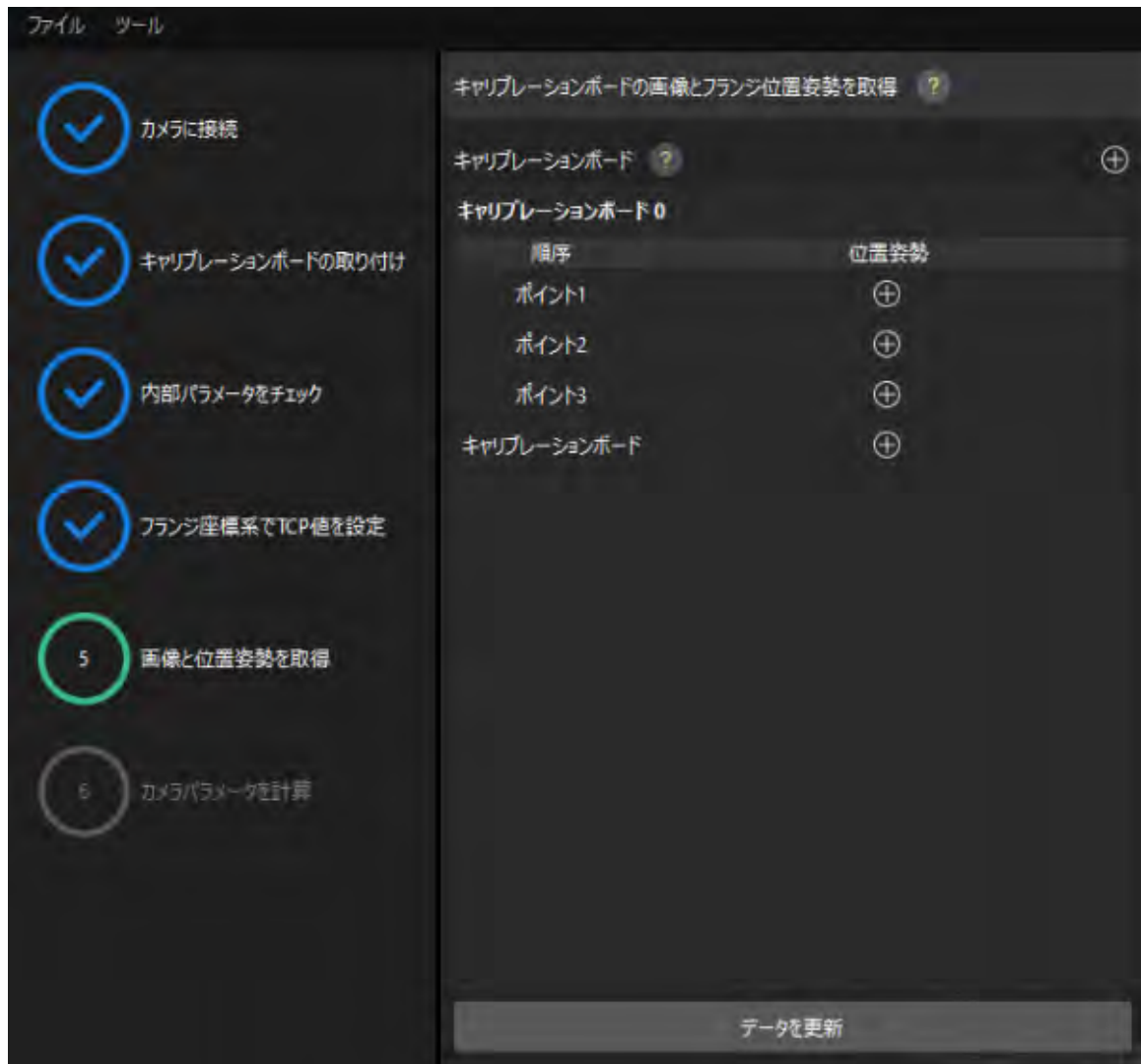
します。



左手系を使用する直行ロボットの場合、ロボットのフランジ位置姿勢を入力する前にY軸の値を反転させる必要があります。



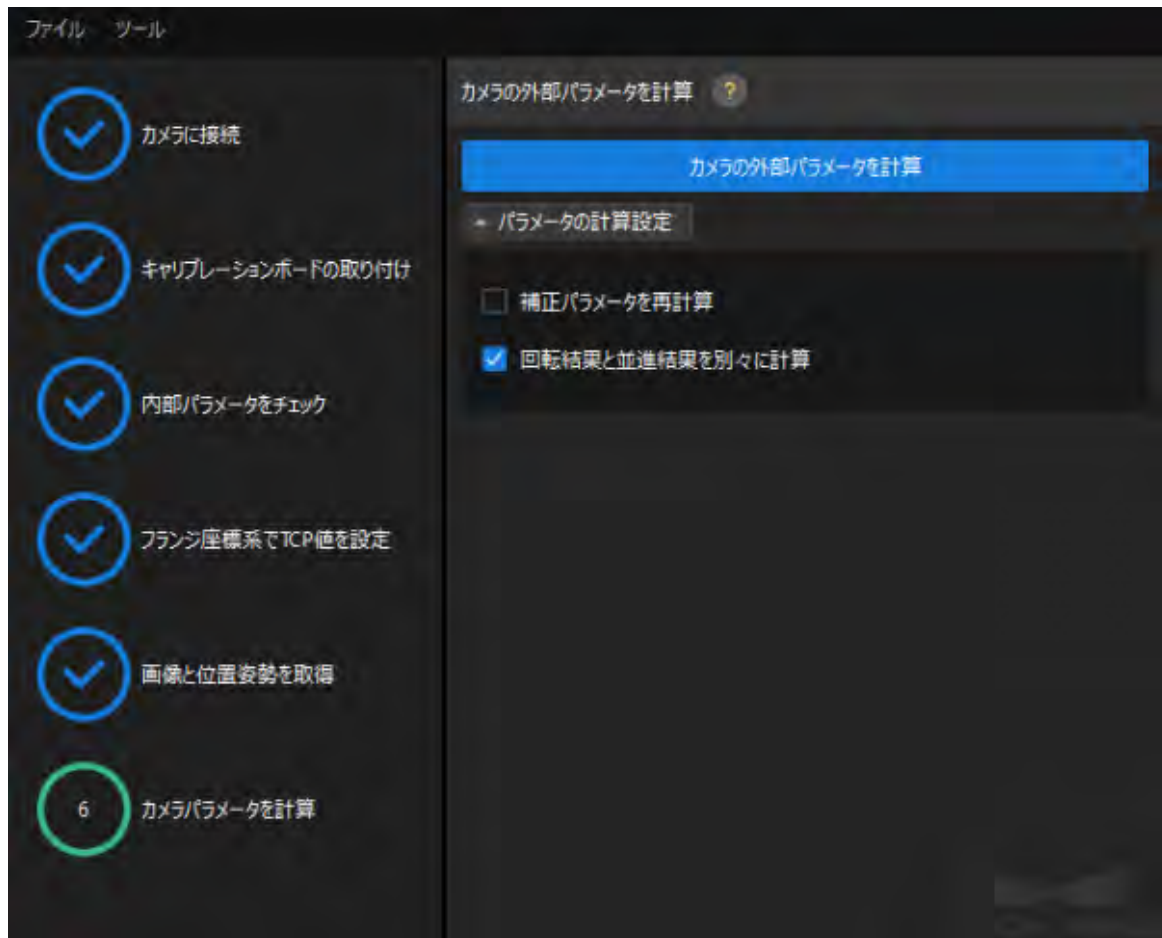
3. 上記の手順をクリックし、TCP先端がポイント2とポイント3での十字型の中心にタッチし、ロボットのフランジ位置姿勢を入力します。
4. キャリブレーションボードの  をクリックし、キャリブレーションボードの画像を取得します。
5. [データを更新]をクリックし、下部にある[次へ]をクリックします。



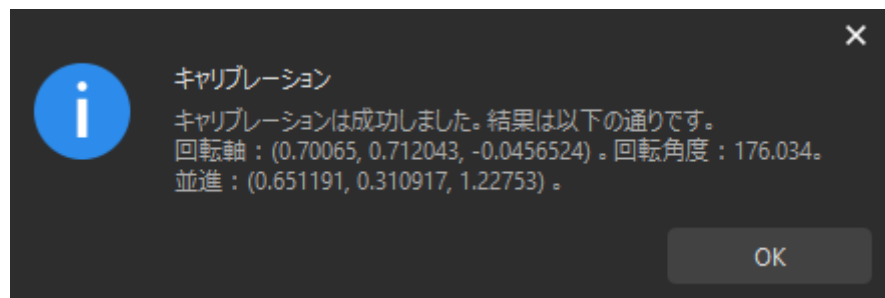
- キャリブレーションでは、外部パラメータを計算するために、少なくとも3つの共線でないポイントをタッチする必要があります。
- 直行ロボットがある方向の自由度を持たない場合、複数のキャリブレーションボードの位置姿勢を追加することができます。以下では、Y軸の自由度を持たないことを例として説明します。まず、TCP先端がキャリブレーションボード上の2つのポイントをタッチできるようにキャリブレーションボードの方向を調整し、ロボットのフランジ位置姿勢を入力してキャリブレーションボードの画像を取得します。次に、キャリブレーションボードの位置をZ方向に上に調整（または下に調整）し、X方向にキャリブレーションボード上のポイントをタッチし、ロボットのフランジ位置位置を入力してキャリブレーションボードの画像を取得します。移動されたキャリブレーションボードの位置が、前の位置と大きく異なっていることを確認してください。

カメラパラメータを計算

1. カメラパラメータを計算手順で、[カメラの外部パラメータを計算]をクリックします。



2. キャリブレーションが正常に実行されたことを示すダイアログで、[OK]をクリックします。



3. 右側の点群ビューパネルでキャリブレーションの誤差点群が表示されます。



誤差点群は、各キャリブレーション位置姿勢におけるキャリブレーションボードの白い円の実際の値と計算値との偏差を示すために使用されます。

4. キャリブレーション精度が要件を満たしていることを確認します。

キャリブレーション精度は、100%を占める誤差の値を求めることで、おおよそ判断することができます。例えば、下図の場合、精度は1.5mm以下となります。



キャリブレーション精度を向上させるには、[キャリブレーション結果の分析](#)をご参照ください。

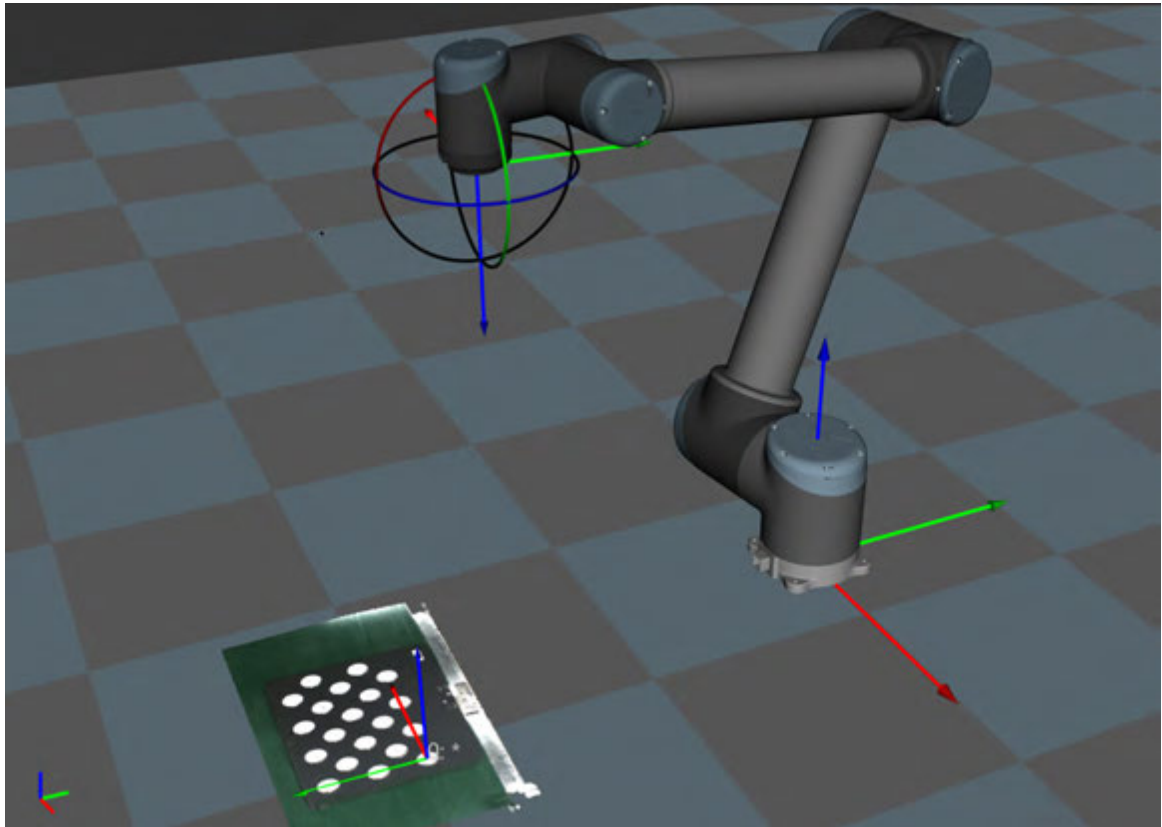
キャリブレーション結果の確認と保存

カメラの外部パラメータを計算した後、シーンビューで固定点に対するキャリブレーションボードの点群（キャリブレーションボードの白い円の十字中心）のずれを確認することで、キャリブレーション結果を大まかに判断することができます。詳細は以下の通りです。

1. キャリブレーションボードを固定位置に配置します。
2. Mech-Vizを起動し、固定点を追加し、それをキャリブレーションボードの白い円の1つの十字の中心に合わせます。
 - a. ステップライブラリから「移動」ステップを選択して、それをプロジェクト編集エリアにドラッグします。
 - b. このステップをクリックして選択し、ステップパラメータで**目標点タイプ**を「」に設定します。それがキャリブレーションボードの白い円のいずれかの十字中心と一致するように、位置姿勢のX/Y/Z座標値を調整します。
3. ロボットを制御してカメラの位置姿勢を数回変更し、キャリブレーションの**カメラパラメータを計算**手順で**[外部パラメータを再計算]**ボタンをクリックします。クリックすると、カメラが撮影を行います。
4. シーンビューで、キャリブレーションボードの点群が固定点に対して大きくなずれがあるかどうかを確認します。

キャリブレーションボードの点群が固定点に対して大きくなずれがない場合、キャリブレーション

ション結果が使用可能であることを示します。



1. EIHの場合、ロボットが接続されていないとロボットの位置姿勢はリアルタイムで取得できないため、シーンビューまたはMech-Vizに表示される正確な点群を確認するには、画像撮影時のロボットの位置姿勢を記入する必要があります。
2. シーンビューに加え、キャリブレーションボードの点群が固定点に対して大きくずれていないかどうかをMech-Vizで確認することも可能です。

1. 下部にある[保存]をクリックし、ポップアップする **キャリブレーションファイルを保存** 画面で[OK]をクリックします。すると、カメラのキャリブレーション結果は、プロジェクトの「calibration」ディレクトリに自動的に保存されます。

これで、キャリブレーション手順が完了しました。

直行ロボットのハンド・アイ・キャリブレーションについての説明は終わります。

4.4.9. ハンド・アイ・キャリブレーション (Eye to Eye)

本節では、カメラがEye to Eye方式で取り付けられている場合にキャリブレーションを実行する方法について説明します。



ハンド・アイ・キャリブレーション (Eye to Eye) の詳細な説明については、[2台カメラのキャリブレーション \(Eye to Eye\)](#) をご参照ください。

事前準備

以下は、ロボットハンド・アイ・キャリブレーションを実行する前に必要な事前準備です。

- Mech-Mindビジョンシステムを構築します。
- キャリブレーションに必要なものを準備します。
- キャリブレーションボードの点群画像生成を調整します。
- ロボットと通信の設定を完了します。

ビジョンシステムを構築

Mech-Mindビジョンシステムの構築については、「ビジョンシステムを構築」節をご参照ください。

ハンド・アイ・キャリブレーションには、Mech-Eye Viewer、Mech-Center、Mech-VisionおよびMech-Vizが必要です。これらのソフトウェアがインストールされ、最新バージョンにアップグレードされていることを確認してください。

キャリブレーションに必要なものを準備

カメラがEye to Eye方式で取り付けられている場合、キャリブレーションを実行するにはキャリブレーションボードの使用が必要です。キャリブレーションボードについては、以下の要件があります。

- キャリブレーションボードの白い円がはっきりと見えること、破損や曲げ変形がないことを確認してください。
- キャリブレーションボードのコネクタをロボット先端のフランジに取り付け、そのコネクタにキャリブレーションボードを取り付けます。キャリブレーションボードがしっかりと取り付けられていること、ロボット先端のXY平面と平行であることを確認してください。



ロボットフランジに取り外しができないハンドがある場合、キャリブレーションボードをハンドに直接固定します。

また、キャリブレーションを実行する前に、2つのカメラの視野が重なる領域の最下部中央をキャリブレーション開始位置として、ロボットを移動させます。



キャリブレーションボードの点群画像生成を調整

1. Mech-Eye Viewerソフトウェアを起動し、2つのカメラをそれぞれ接続し、カメラのパラメータを調整します。
2. 2Dパラメータを調整して、2D画像にキャリブレーションボードがはっきり見られ、露光過度や露光不足などがないようにします。
3. 3Dパラメータを調整して、キャリブレーションボードの白い円の点群が完全に見られるようにします。点群の変動範囲を小さくするために、**点群後処理**の中の**点群平滑化**と**ノイズ除去**を**Normal**に設定することを推奨します。



現場の環境光が複雑な場合、2D画像や点群への影響を軽減するために、遮光・補光することを推奨します。

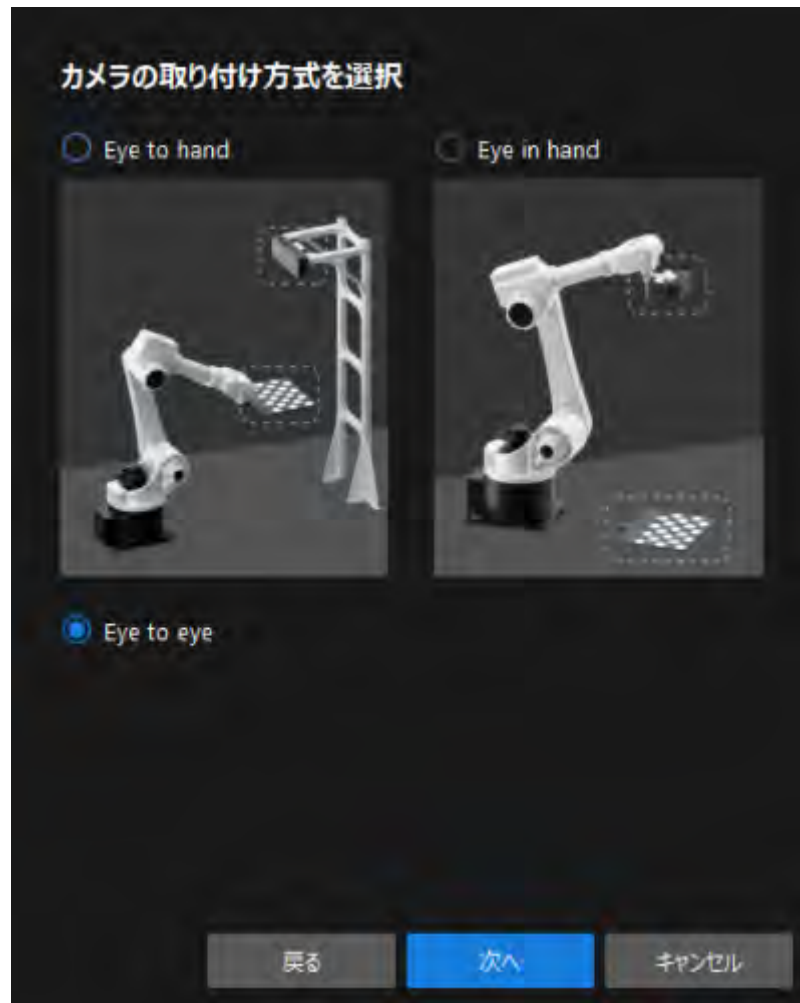
4. 上記の手順を完了することで、キャリブレーションボードの点群の画像品質が要件を満たしていることを確認します。

	正常	露光過度	露光不足
2D画像			
点群			

キャリブレーション前の設定

標準インターフェース通信またはVizティーチング通信の場合、自動キャリブレーション（Eye to Hand）の [キャリブレーション前の設定](#) を参照して設定を行います。

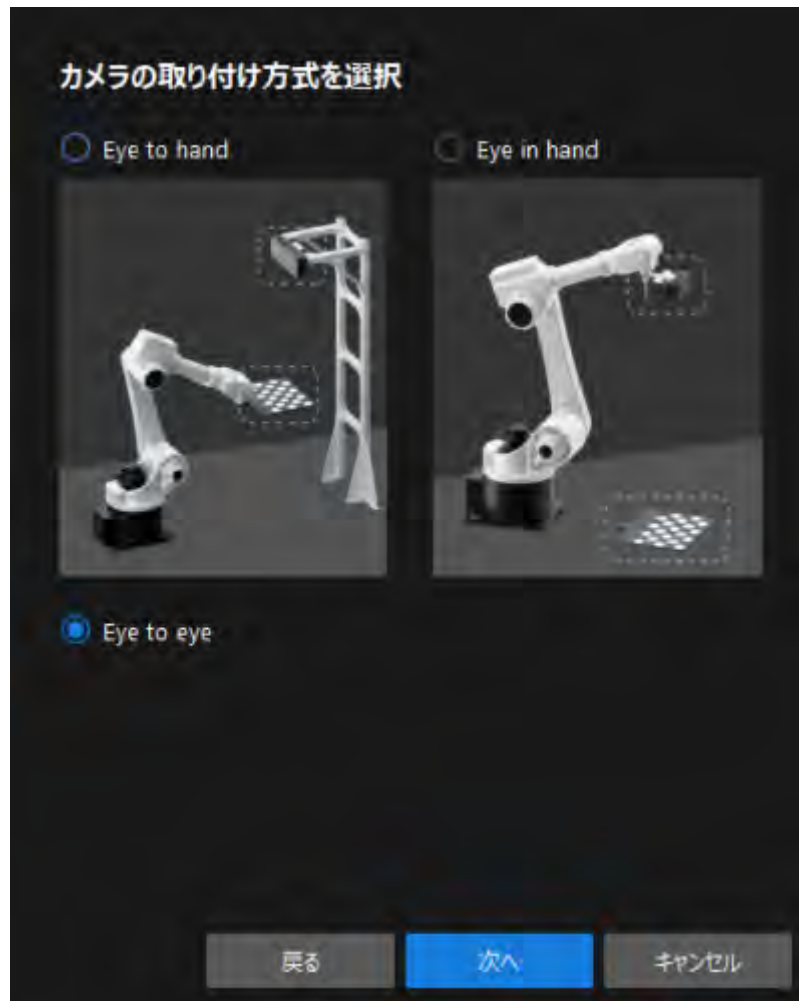
Eye to Eyeのキャリブレーション前の設定は、**カメラの取り付け方式を選択** 画面で**Eye to eye**を選択する必要があります。



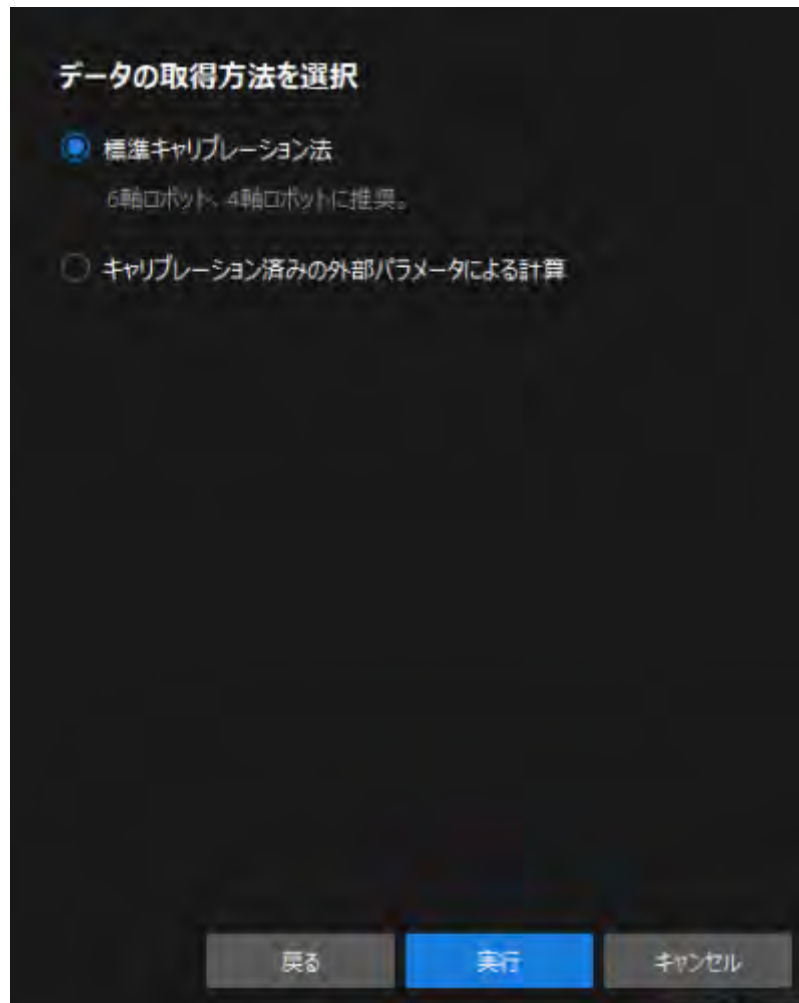
ロボットが標準インターフェース通信またはVizティーチング通信に対応していない場合、標準キャリブレーション法を使用する手動キャリブレーション（Eye to Hand）の [キャリブレーション前の設定](#) を参照して設定を行います。

その他のキャリブレーション手順と比較して、Eye to Eyeのキャリブレーション前の設定は、以下の手順が異なります。

1. カメラの取り付け方式を選択画面で**Eye to eye**を選択する必要があります。




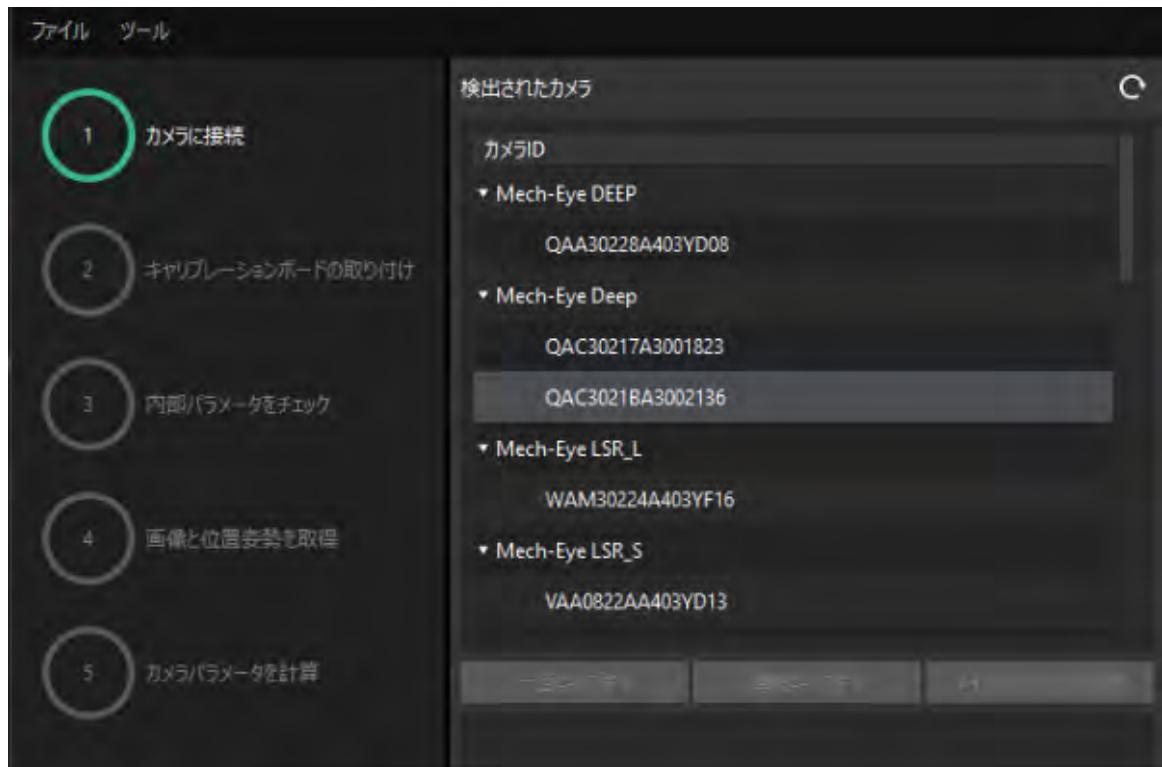
2. データの取得方法を選択画面で、標準キャリブレーション法を選択します。




キャリブレーションの実行手順

カメラ接続

1. **カメラに接続** 手順で、検出されたカメラから接続するサブカメラを選択して  をクリックするか、ダブルクリックして接続します。

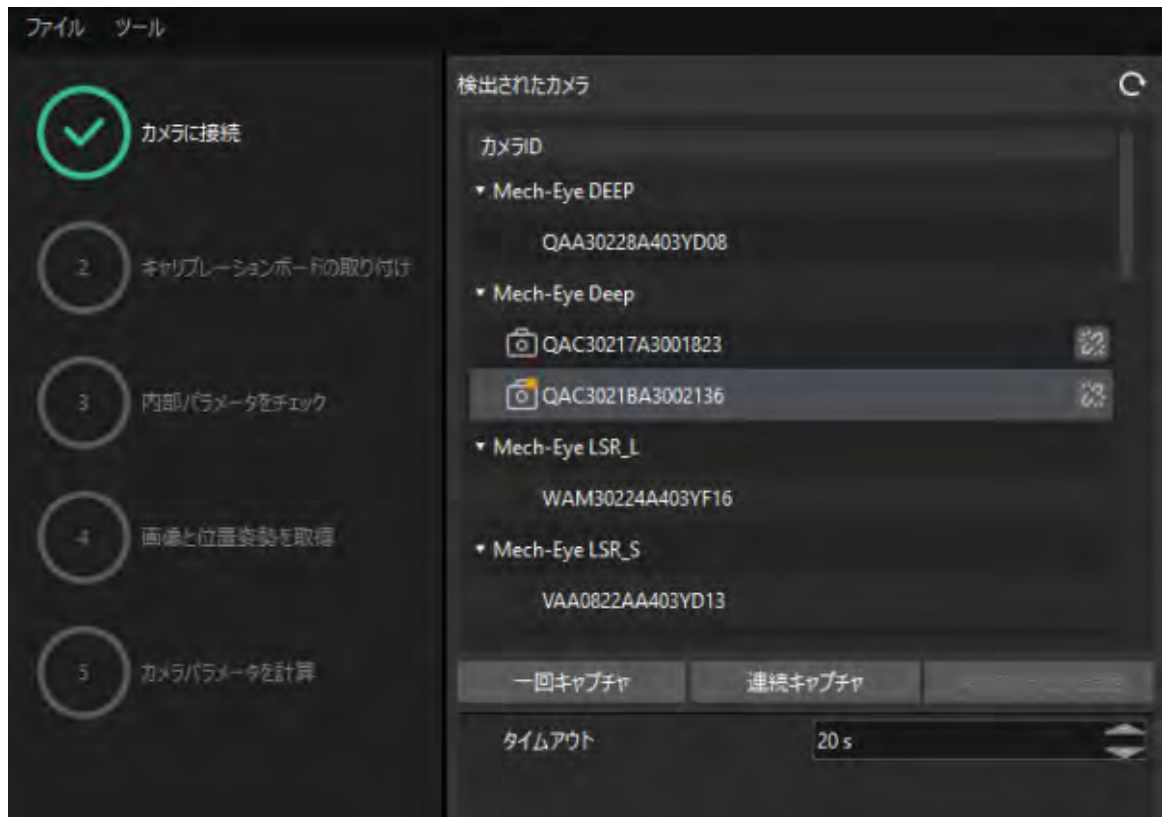


2. 上記の方法と同様にメインカメラに接続します。接続後、カメラIDの前に  が表示されます。



メインカメラを切り替えるには、カメラを選択し、[**メインカメラとして設定**]をクリックします。

3. カメラ接続後、[**一回キャプチャ**]または[**連続キャプチャ**]をクリックします。



4. 画像ビューでは、取得した2D画像と深度画像が要件を満たしていることを確認してから、下部にある[次へ]をクリックします。



- 取得した画像が要件を満たしていない場合、Mech-Eye Viewerを起動し、[カメラの2Dと3D露光パラメータを調整](#)して再取得する必要があります。
- この手順では、メインカメラの2D画像と深度画像のみが取得されます。サブカメラの画像取得効果を確認するには、それをメインカメラに設定し、画像取得効果を確認してから元の設定に戻ります。

その他の手順

ロボットが標準インターフェース通信またはVizティーチング通信に対応している場合、自動キャリブレーション（Eye to Hand）の[キャリブレーション手順](#)を参照して残りの設定を行います。

ロボットが標準インターフェース通信またはVizティーチング通信に対応していない場合、標準キャリブレーション法を使用する手動キャリブレーション（Eye to Hand）の[キャリブレーション手順](#)を参照して残りの設定を行います。



- 画像と位置姿勢を取得**手順で、ソフトウェアは2つのカメラを同時に使用して画像と位置姿勢を取得します。異なるカメラで取得したキャリブレーションボードの画像とキャリブレーションボードの白い円の情報を確認する場合、**画像と位置姿勢一覧**ネルでカメラを切り替えることができます。
- カメラパラメータを計算**手順で、ソフトウェアは2つのカメラの外部パラメータに加え、サブカメラがメインカメラの座標系における位置姿勢の変換関係も計算します。

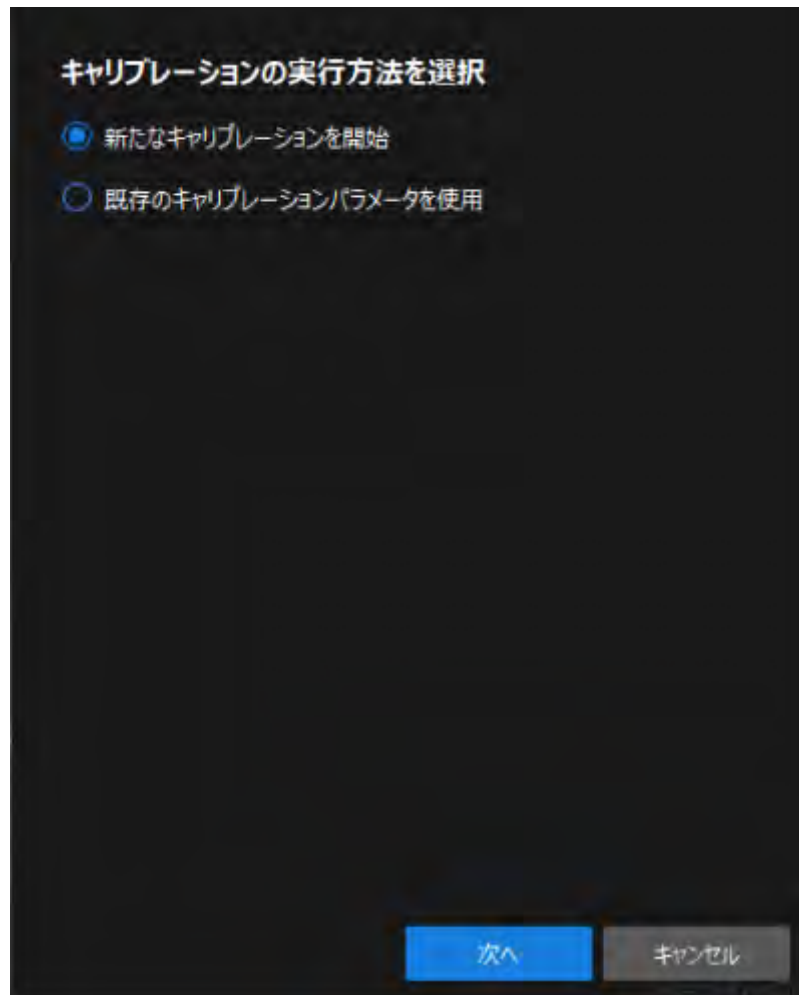
これで、キャリブレーション手順が完了しました。

キャリブレーション済みの外部パラメータによる2つのカメラの位置姿勢関係を計算

2つのカメラの外部パラメータを別々にキャリブレーションした後、Eye to Eyeのキャリブレーションで2つのカメラの位置姿勢関係を直接計算することができます。

キャリブレーション済みの外部パラメータで2つのカメラの位置姿勢関係を計算するには、以下の手順を実行します。

1. Mech-Visionを起動し、ツールバーで[**カメラキャリブレーション（標準モード）**]をクリックします。すると、**キャリブレーション前の設定画面**が表示されます。
2. **キャリブレーションの実行方法**を選択画面で、**新たなキャリブレーションを開始**を選択し、[**次へ**]をクリックします。



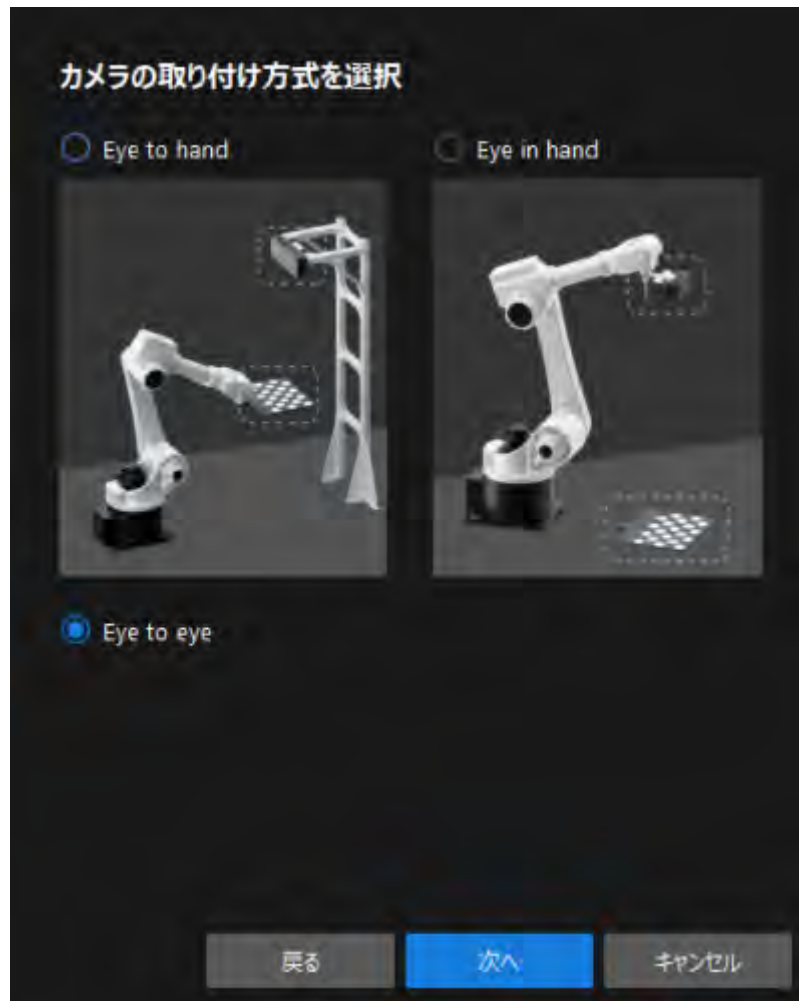
3. **キャリブレーションのタスク**を選択画面で、ドロップダウンリストから**その他のロボットのハンド・アイ・キャリブレーション**を選択し、必要に応じて**ロボットのオイラー角のタイプ**を指定し、ロボットの座標系を選択してから[**次へ**]をクリックします。



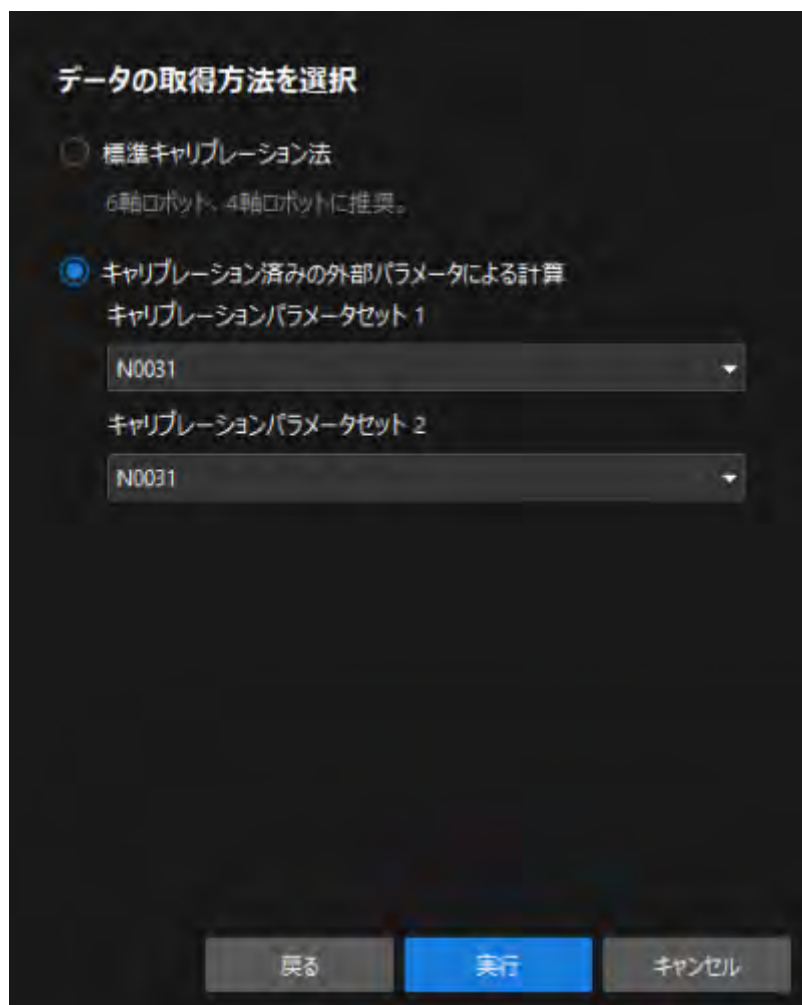
4. キャリブレーションするためのロボットを選択画面で、実際の状況に応じて6軸ロボット、4軸ロボットまたは5軸、または他のタイプのロボットを選択してから、[次へ]をクリックします。




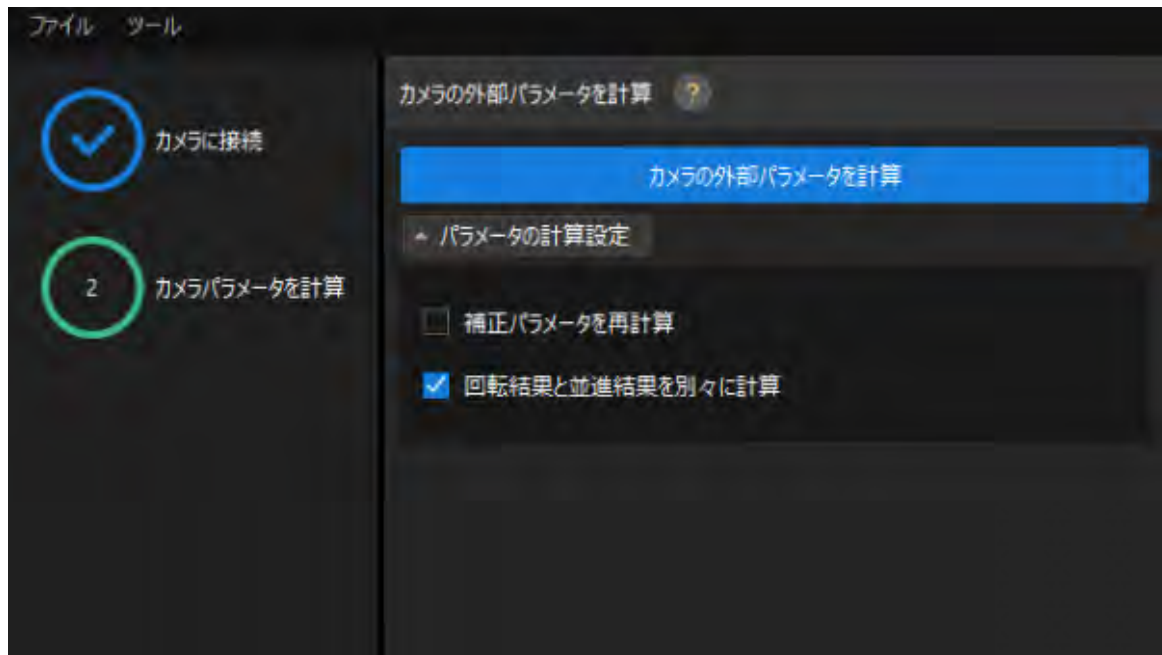
5. カメラの取り付け方式を選択画面で、**Eye to eye**を選択して[次へ]をクリックします。



- データの取得方法を選択画面で、キャリブレーション済みの外部パラメータによる計算を選択し、2つのカメラの外部パラメータファイルを選択してから[実行]をクリックします。すると、キャリブレーション（Eye to Eye）画面が表示されます。



7. **カメラに接続** 手順で、**検出されたカメラ** から接続するカメラを選択して  をクリックするか、ダブルクリックして接続します。
8. 繰り返して2番目のカメラに接続してから、下部にある[次へ]をクリックします。
9. **カメラパラメータを計算** 手順で、「パラメータの計算設定」を展開し、**回転結果と並進結果**を別々に計算にチェックを入れてから、[カメラの外部パラメータを計算]をクリックします。



外部パラメータが計算した後、**点群ビュー**パネルで融合された点群が表示されます。

Mech-Visionプロジェクトでキャリブレーションの融合効果を確認

下図のようなプロジェクトを構築し、「出力がない場合に制御フローをトリガー」と「出力がある場合に制御フローをトリガー」にチェックを入れます。



「データをマージ」ステップをシングル実行すると、融合された点群が表示されます。下図に示すように、データをマージした後の出力点群は、融合された全体点群となります。左上隅にある**View as Whole|1|2**をクリックして点群を切り替えることができます。



4.4.10. 参考情報

本節では、ロボットハンド・アイ・キャリブレーションに関連する参考情報を提供しています。

[キャリブレーションの関連概念](#)

[キャリブレーション原理](#)

[直行ロボットのキャリブレーション説明](#)

[キャリブレーション結果の確認と分析](#)

[よくある問題と解決策](#)

4.4.10.1. キャリブレーションの関連概念

ロボットの通信方式

通信方式とは、ロボットがどうの方式でビジョン側と通信することを指します。Mech-Mind ビジョンシステムは、標準インターフェース通信、Adapter通信とVizティーチング通信の3種類の通信方式に対応しています。詳細については、[通信方式の概要](#)をご参照ください。

カメラの取り付け方式

カメラの取り付け方式とは、カメラがセルに取り付けられている方式のことを指します。一般的な取り付け方式は、Eye to Hand (ETH) とEye in Hand (EIH) です。また、カメラ視

野の拡大と点群品質の向上のために、2台カメラ（Eye to Eyeと呼ぶ）を使用することも可能です。

キャリブレーション方式

キャリブレーション方式は、キャリブレーション画像や位置姿勢の取得を自動化するかどうかで、自動キャリブレーションと手動キャリブレーションに分けられます。手動キャリブレーションは複雑であるため、自動キャリブレーションを推奨します。

自動キャリブレーション（推奨）

キャリブレーション中にロボットを接続し、Mech-Visionが自動的にキャリブレーション経路を計画し、計画した経路に沿ってロボットを制御し、キャリブレーション画像と位置姿勢の取得を完了させます。

手動キャリブレーション

キャリブレーション中にロボットを接続せずに、ロボットを手動で移動し、ユーザーより計画した経路に沿って移動するか、キャリブレーションボードの白い円にタッチし、ロボットの位置姿勢を手動で入力してキャリブレーション画像を撮影します。

キャリブレーションデータの取得方法

キャリブレーションデータの取得方法は、キャリブレーション中にデータを取得する方法のことを指します。Mech-Visionでは、標準キャリブレーション法とTCPタッチ法に対応しています。

標準キャリブレーション法（推奨）

ロボットはソフトウェアで自動生成された、あるいは手動で計画された経路点を順次移動し、各経路点でキャリブレーションボードの画像を撮影し、キャリブレーションボードの白い円を認識してロボットのフランジの位置姿勢を取得し、最終的にキャリブレーション行列を形成します。これにより、キャリブレーションボード、カメラ、ロボット間の関係を確立します。この方法はシンプルなプロセスで高いキャリブレーション精度を実現します。6軸または4軸ロボットに使用することを推奨します。

TCPタッチ法

三点法を使用してキャリブレーションボードの位置姿勢を確認してから、キャリブレーションボード、カメラ、ロボット間の関係を確立します。この方法は、ロボット動作スペースが狭く、キャリブレーションボードが取り付けられない場合に場合に適します。5軸または他のタイプのロボットに使用することを推奨します。

カメラの内部パラメータ

内部パラメータとは、レンズの焦点距離、歪みなどのカメラ内部の基本的なパラメータです。通常、内部パラメータがカメラの工場出荷時に設定され、カメラに保存されています。

カメラの外部パラメータ

外部パラメータとは、ロボットとカメラとの位置姿勢の変換関係のことを指します。言い換えれば、ロボットのハンドとアイの関係なので、外部パラメータのキャリブレーションを口

ボットのハンド・アイ・キャリブレーションといいます。ロボットとカメラの相対位置姿勢はシーンによって異なるため、カメラとロボットのハンド・アイの関係を取得するには、現場でキャリブレーションを実行する必要があります。

キャリブレーションポイント

キャリブレーションポイントとは、キャリブレーション時にカメラがキャリブレーションボードの画像を取得する際のロボットの位置姿勢を指します。標準キャリブレーション法でデータを取得する場合、キャリブレーションポイントがキャリブレーション経路における点となります。TCPタッチ法でデータを取得する場合、キャリブレーションボードに3つの共線でないキャリブレーションボードの白い円の中心をタッチし、キャリブレーションボードの画像を一回撮影することでデータを取得できます。

キャリブレーションボードの白い円

キャリブレーションボードの白い円とは、キャリブレーションボード上の円形の特徴点のことを指します。キャリブレーション時には、ソフトウェアは取得したキャリブレーションボードの画像（2D画像と深度画像）からキャリブレーションボードの白い円の中心ピクセル座標とカメラ座標系での円の中心の座標を計算します。その後、取得したキャリブレーションボードの白い円のデータをもとに、カメラの外部パラメータを計算します。

4.4.10.2. キャリブレーション原理

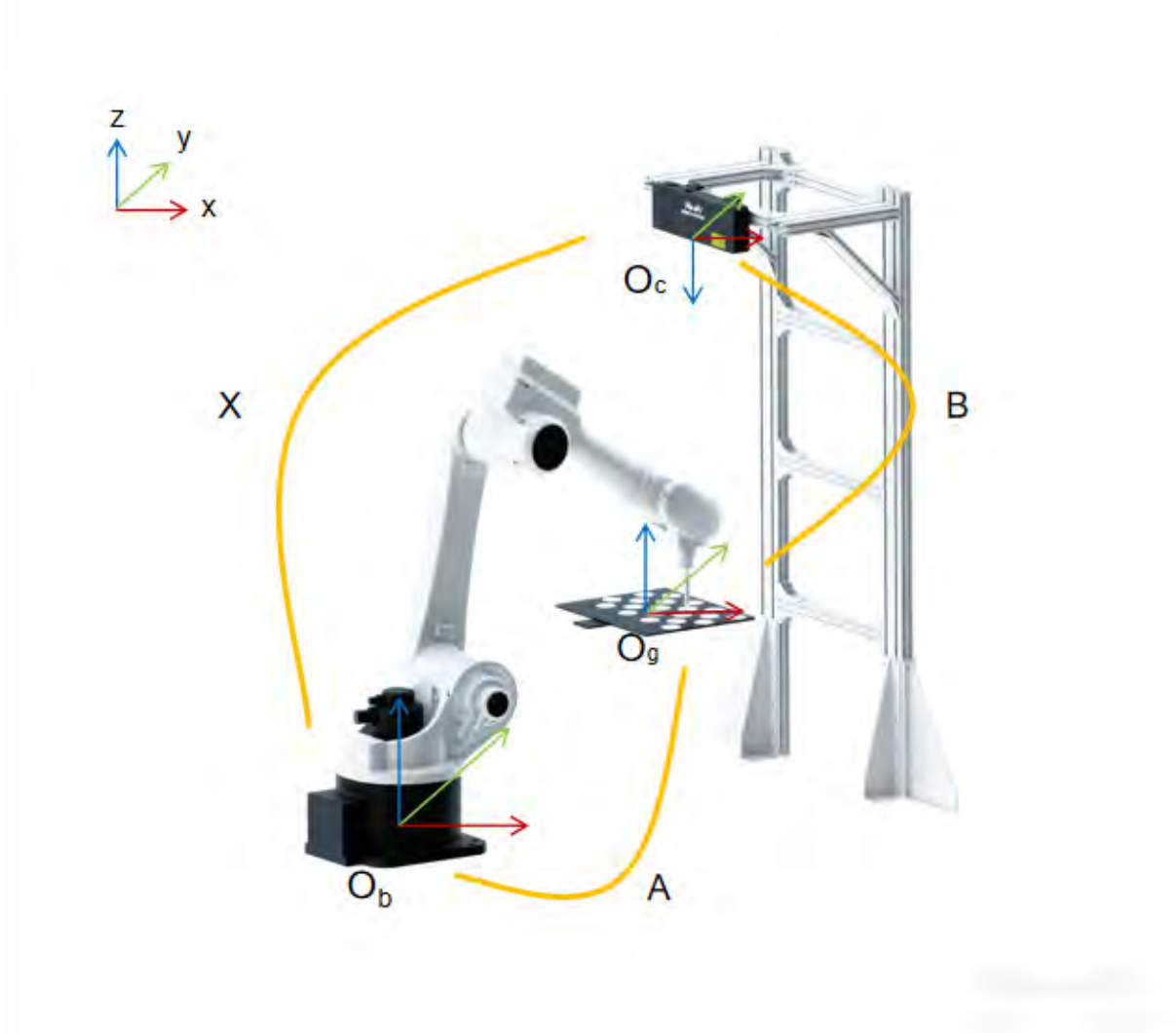
本節では、様々なシーンに適したキャリブレーションの動作原理について説明します。

標準キャリブレーション法（ETH）



フランジを介して寸法が確認されたキャリブレーションボードをロボット先端に取り付けて、キャリブレーションボードの各白い円がロボットベース座標に対する座標Aを取得します。カメラでキャリブレーションボードの各白い円の画像を撮影して、カメラ光学中心がキャリブレーションボードの各白い円に対する座標Bを取得します。カメラ光学中心とロボットベース座標との位置姿勢関係Xをとします。A、B、Xは方程式になり、Xが求められます。キャリブレーションボードとフランジ位置との位置関係Cは未知で、キャリブレーションボードを移動させて異な

る数値を取得してCを求め、さらにAが求められます。ロボットを移動させてキャリブレーションボードがカメラに対する位置姿勢を何回も変更して複数の方程式を得ます。これらの方程式の値をフィッティング計算して最適なX値を得ます。位置姿勢の関係は下図に示します。



カメラ座標系とロボット座標系の対応関係を表す式は以下の通りです。

$$\begin{aligned} B_1 \bullet X \bullet C_1 &= A \\ B_2 \bullet X \bullet C_2 &= A \\ B_1 \bullet X \bullet C_1 &= B_2 \bullet X \bullet C_2 \end{aligned}$$

標準キャリブレーション法 (EIH)



カメラスタンドを使用してカメラをロボット先端に固定します。この場合、ロボット先端のフランジ中心はカメラ光学中心との位置姿勢関係は固定されます。それは、下図の未知変数 X です。ロボットのベース座標系に対する先端のフランジ中心の位置姿勢は、既知変数 B とします。カメラはキャリブレーションボードの画像を撮影し、カメラの光学中心とキャリブレーションボード上の各白い円との位置姿勢の関係を取得し、既知変数 C を取得します。キャリブレーションボードは、カメラの視野内に平らに置かれて、ロボットのベース座標に対する相対位置姿勢は、固定値 A とします。このように、変数 A 、 B 、 C 、 X は方程式になります。以下の方程式には、 A が定数値なので、最初の2つの方程式から得た新しい方程式では、 X だけが未知数です。ロボット先端の位置姿勢を変更して、さまざまな角度から画像を撮影することにより、複数セットの A 、 B 、 C の値を取得します。それらを使用してフィッティング計算して X の最適値を取得します。



カメラ座標系とロボットフランジ座標系の対応関係を表す式は以下の通りです。

$$B_1 \bullet X \bullet C_1 = A$$

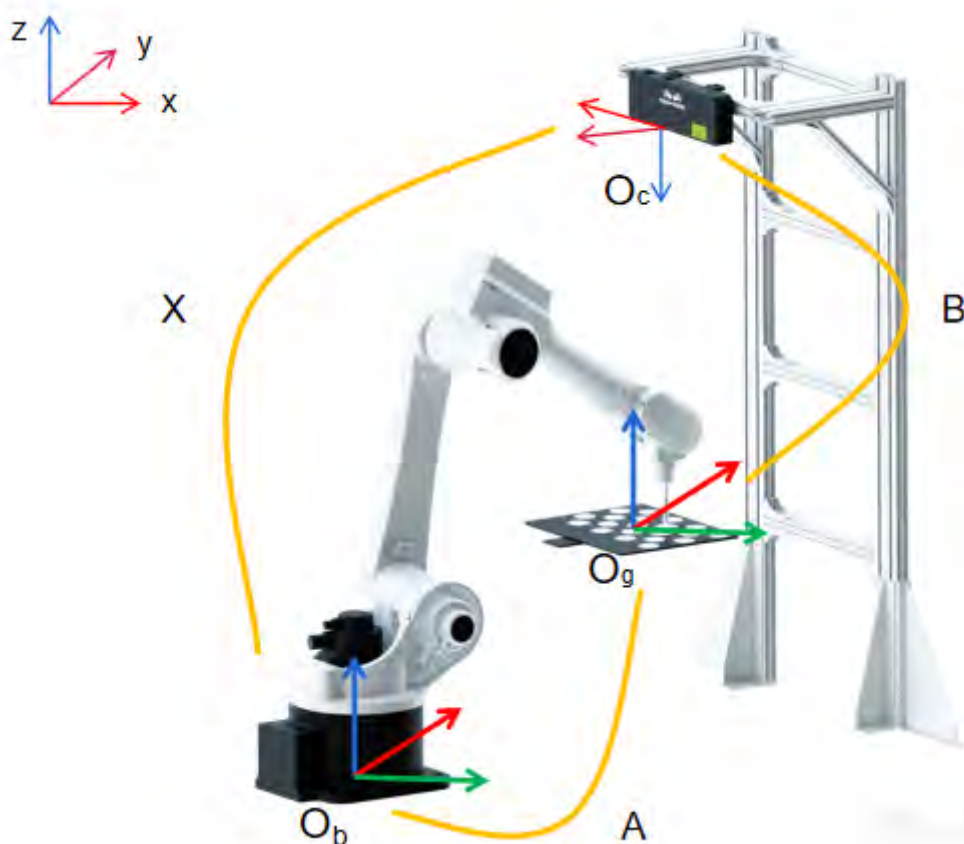
$$B_2 \bullet X \bullet C_2 = A$$

$$B_1 \bullet X \bullet C_1 = B_2 \bullet X \bullet C_2$$

TCPタッチ法 (ETH)



TCPタッチ方法でキャリブレーションする場合、キャリブレーションボードを作業平面に配置し、ロボット先端に既知寸法のTCP先端を取り付けます。TCP先端でキャリブレーションボードの白い円の十字型の中心にタッチします。その原理を下図に示します。A、Bの値が既知なので、Xの値を求めます。キャリブレーションボードとロボット先端は固定されておらず、TCP座標が既知のTCP先端をキャリブレーションボードの白い円にタッチさせることでAの値が求められます。



カメラ座標系とロボット座標系の対応関係を表す式は以下の通りです。

$$B_1 \bullet X \bullet C_1 = A$$

$$B_2 \bullet X \bullet C_2 = A$$

$$B_1 \bullet X \bullet C_1 = B_2 \bullet X \bullet C_2$$

TCPタッチ法 (EIH)



TCPタッチ方法でキャリブレーションする場合、キャリブレーションボードを作業平面に配置し、ロボット先端に既知寸法のTCP先端を取り付けます。TCP先端でキャリブレーションボードの白い円にタッチします。その原理を下図に示します。A、B、Cの値が既知なので、Xの値も求められます。



カメラ座標系とロボットフランジ座標系の対応関係を表す式は以下の通りです。

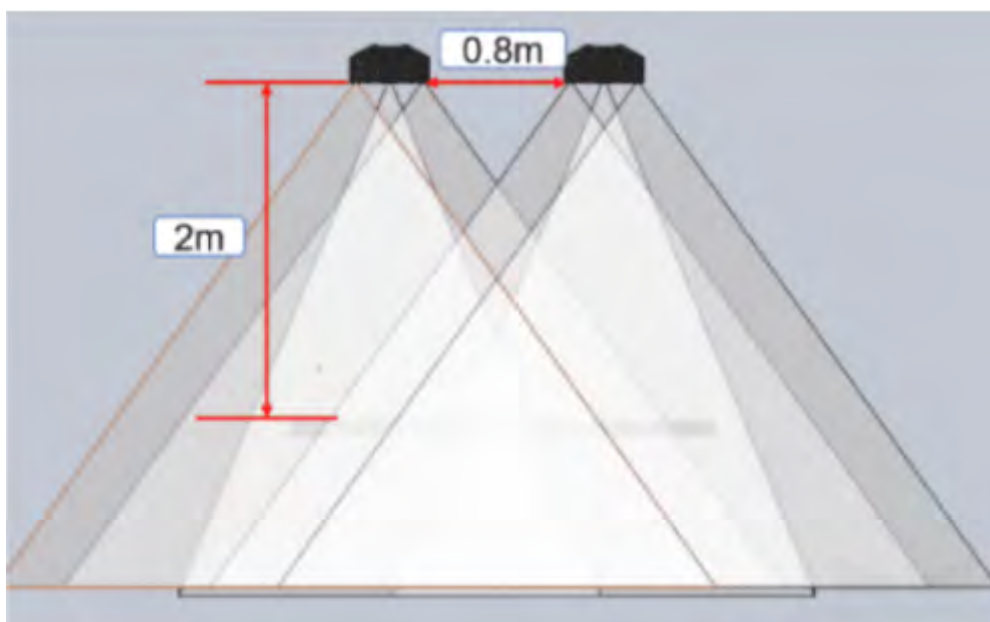
$$B_1 \bullet X \bullet C_1 = A$$

$$B_2 \bullet X \bullet C_2 = A$$

$$B_1 \bullet X \bullet C_1 = B_2 \bullet X \bullet C_2$$

2台カメラ（Eye to Eye）のキャリブレーション

下図に示すように、2台カメラにより、カメラの視野拡大と重なる点群品質向上を実現します。



カメラ（メインカメラとサブカメラを含む）がスタンドに固定されます。2台カメラのキャリブレーションでは、各カメラの外部パラメータのキャリブレーションと、2つのカメラ間の位置姿勢関係のキャリブレーションの両方が行われます。Mech-Visionでは、これに対応可能なEye to Eyeキャリブレーションを提供しています。



使用する2台のカメラは同じ解像度で、作業領域全体がカメラの視野（2D・3D）の重なり合う領域に入っている必要があります。

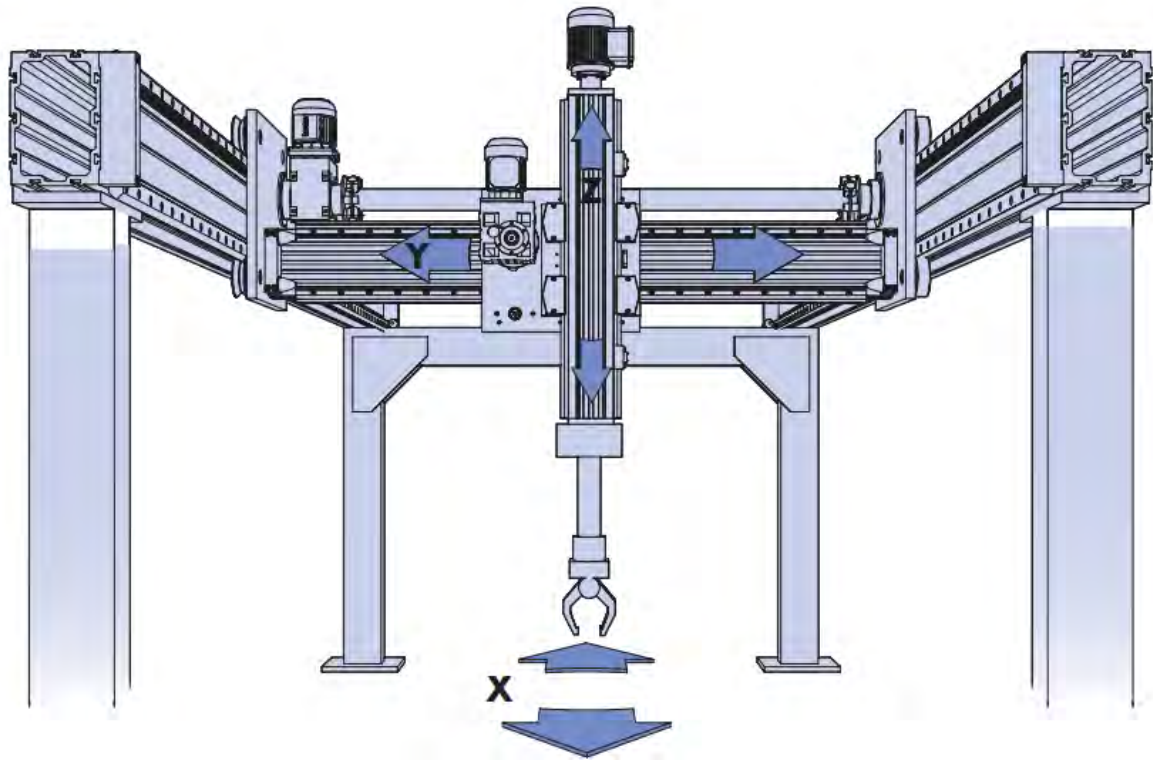
2台カメラのキャリブレーションは、標準キャリブレーション法によるキャリブレーションデータの取得にのみ対応しており、自動キャリブレーションと手動キャリブレーションの両方に対応しています。

4.4.10.3. 直行ロボットのキャリブレーション説明

関連概念

直行ロボット

直行ロボットは、空間XYZ直角座標系に基づく自動制御・再プログラム可能な多自由度自動化機械で、さまざまな作業に適しています。



直行ロボットの自由度

直行ロボットは、多自由度運動をサポートしています。各運動自由度の間の空間における角度を直角とします。

直行ロボットは通常、2-6自由度をサポートしています。詳細は以下の通りです。

- XYZ方向に3軸の直線運動が可能（3自由度）
- XYZ方向に3軸の直線運動、Z軸（RzまたはC軸を呼ぶ）の回転が可能（4自由度）
- XZ方向に2軸の直線運動が可能（2自由度）
- XZ方向に2軸の直線運動、Z軸の回転が可能（3自由度）

カメラは通常、直行ロボットのいずれかの軸（Z軸、C軸など）に取り付けられています。カメラの撮影位置は、軸の動きに影響されることになります。そのため、キャリブレーション前の設定において、どの軸の動きがカメラの位置に影響を与えるかを指定する必要があります。

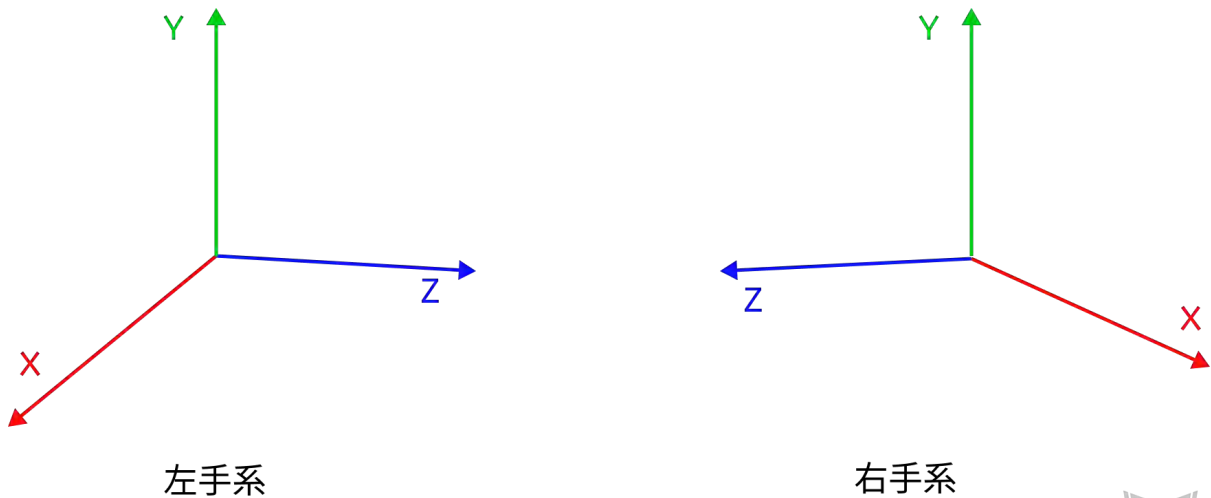
左手/右手系

Mech-Mindビジョンシステムでは、右手系でのロボットのフランジ位置姿勢を使用して外部パラメータを計算します。そのため、直行ロボットを使用する場合、その座標系が右手系であるかどうかを判断する必要があります。

左手系か右手系かは、以下のような方法で判断できます。

- 左手の親指をX軸の正方向に、人差し指をY軸の正方向に向け、中指がZ軸の正方向に向けることができれば、この座標系を左手系（下図左）と呼びます。

- 右手の親指をX軸の正方向に、人差し指をY軸の正方向に向け、中指がZ軸の正方向に向けることができれば、その座標系を右手系（下図右）と呼びます。



直行ロボットの座標系が左手系の場合、その後のキャリブレーションやビジョン処理を容易にするために、それを右手系に変更することを推奨します。通常、直行ロボットのいずれかのXYZ軸のエンコーダの運動の正方向を反転させる必要があります。これは通常、直行ロボットの架台が取り付けられている軸を反転し、Mech-Visionで右手系を選択してキャリブレーションを行います。

直行ロボットで設定できない場合、Mech-Visionで左手系を選択するだけで、ソフトウェアが自動的に入出力位置姿勢データに対して変換を行います。

キャリブレーション原理

カメラは通常、直行ロボットのある軸、例えばZ軸に取り付けられ、つまりカメラはEye in Hand方式で取り付けられています。したがって、直行ロボットのキャリブレーションは、カメラ座標系とロボットツール座標系との間の相対関係を求めることを目的としています。

直行ロボットの自由度や作業空間が限られた場合、Mech-VisionがTCPタッチ法でキャリブレーションデータを取得し、キャリブレーションボード、カメラ、ロボット間の関係を確立します。また、直行ロボットのキャリブレーションでは、キャリブレーションボードを複数枚追加することに対応します。自由度が限られ、キャリブレーションボード上の3つのポイントに触れるように直行ロボットを動かせない場合、直行ロボットがキャリブレーションボード上の3つのポイント（3ポイントは共線ではない）に触れるように複数のキャリブレーションボードを使用することが可能です。

直行ロボットのキャリブレーション原理については、[TCPタッチ法（EIH）](#) をご参照ください。

外部パラメータの使用

直行ロボットのキャリブレーションにより、キャリブレーション位置の外部パラメータファイルが生成され、Mech-Visionは直行ロボットの位置に基づいて動的な外部パラメータをリアルタイムで計算します。Mech-Visionが動的な外部パラメータを計算する時、カメラ座標系における点群

や位置姿勢を直行ロボットの座標系に変換するために、**点群変換（直行ロボット）**と**位置姿勢を変換（直行ロボット）**ステップと併用する必要があります。

プロジェクトを実行する際、直行ロボットがMech-Mindビジョンシステムと正しく通信できること、Mech-Visionプロジェクトが実行した際にロボットプログラムがロボットのフランジ位置姿勢を提供し、「カメラから画像を取得」ステップで画像撮影時にロボットのフランジ位置姿勢を正常に取得できることを確認してください。



直行ロボットの座標系が左手系を使用する場合、Mech-Centerはロボットプログラムから送信されたロボットのフランジ位置姿勢を自動的に右手系に変換してから、ビジョン処理のためにMech-Visionに送信します。ビジョン結果を返す際に、Mech-CenterはMech-Visionから出力された直行ロボットの座標系での位置姿勢を左手系に変換してから、ロボットプログラムに送信します。

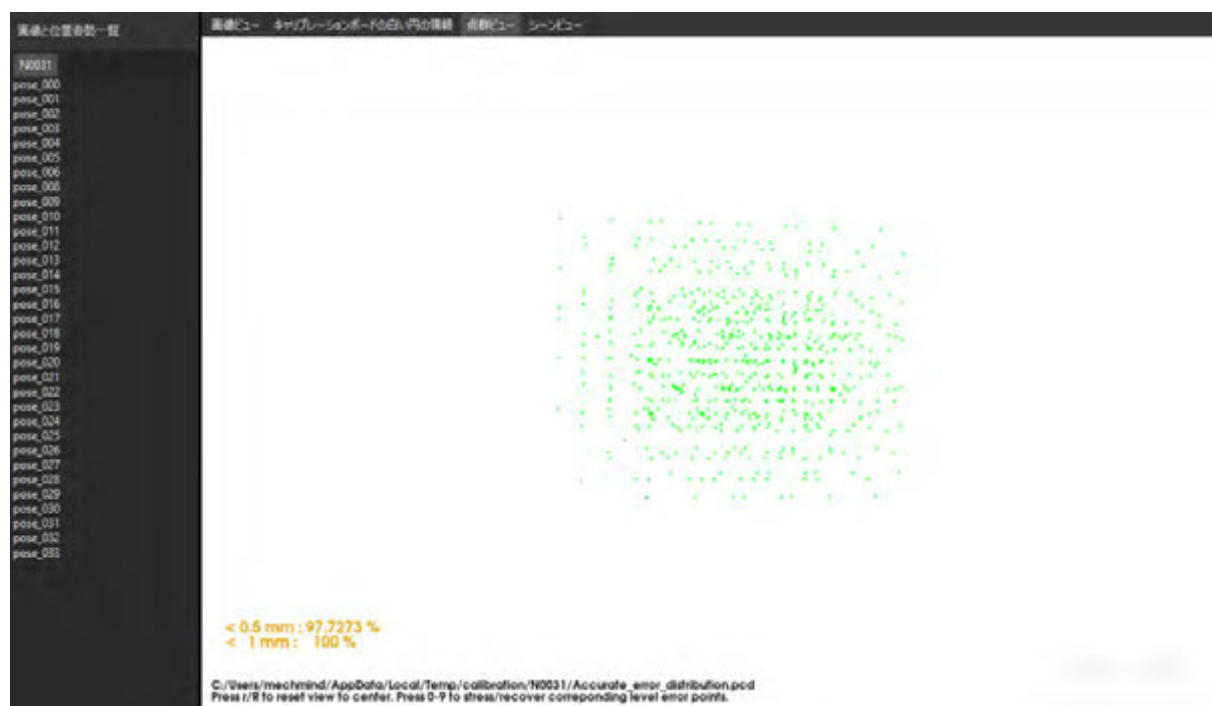
4.4.10.4. キャリブレーション結果の確認と分析

キャリブレーション済みの外部パラメータデータを確認し、その精度が要求を満たすかどうかを判断する必要があります。誤差が通常の範囲を超えた場合は、原因を突き止めて問題を解決し、外部パラメータが要求を満たすまでキャリブレーションを再度実行する必要があります。

キャリブレーション結果の確認

点群ビューで誤差点群を確認

カメラパラメータをチェック手順でキャリブレーション結果が計算した後、下図のような誤差点群が**点群ビュー**パネルに表示されます。誤差点群は、各キャリブレーション位置姿勢におけるキャリブレーションボードの白い円の実測の値と計算値との偏差を示すために使用されます。



誤差点群の色は誤差レベル（0.5mmにつき1つの誤差レベル）を示し、緑から赤にかけて色が濃いほど、その点での誤差が大きくなります。

誤差レベルごとの点を表示するには、**点群ビュー** パネルで、英語入力方式で数字キー0-9を押します。ソフトウェアは、点群の誤差レベルに対応するポイントを強調表示します。キー0は誤差0.5mm以下のポイントに、キー1は誤差0.5mm-1mmの点に対応します。これによって類推します。

下図では、誤差が0.5mm以下のポイントを示しています（数字キー0を押す）。



下図では、誤差が2.0-2.5 mmのポイントを示しています（数字キー4を押す）。

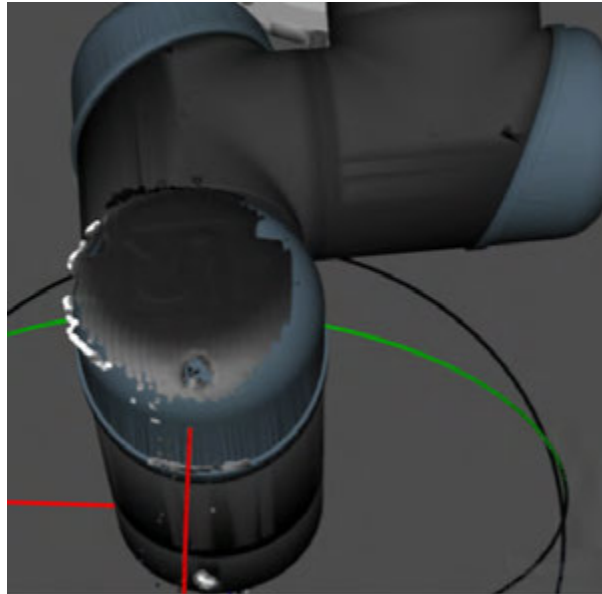


シーンビューでロボットの点群とロボットモデルとの重なりを確認（ETH場合のみ）

tag::calib_result_verification_eth[] カメラの外部パラメータを計算した後、シーンビューでロボット点群とロボットモデルの点群との重なりを確認してキャリブレーション結果を大まかに判断します。具体的な流れは以下の通りです。

1. ロボットアームをカメラの視野範囲内に移動させます。
2. カメラパラメータを計算手順で[外部パラメータを再計算]をクリックします。クリックすると、カメラが撮影を行います。
3. シーンビューに切り替え、ロボットの点群とロボットモデルの重なりを確認します。

ロボットの点群がロボットモデルとほぼ一致すれば、キャリブレーション結果が使用可能です。



1. ただし、ロボットモデルはロボット実機の外観と完全に一致しない場合があり、外部パラメータの微調整の基準とすることができないことに注意してください。
2. シーンビューに加え、Mech-Vizでロボットの点群とロボットモデルとの重なりを確認できます。

```
end::calib_result_verification_eth[]
```

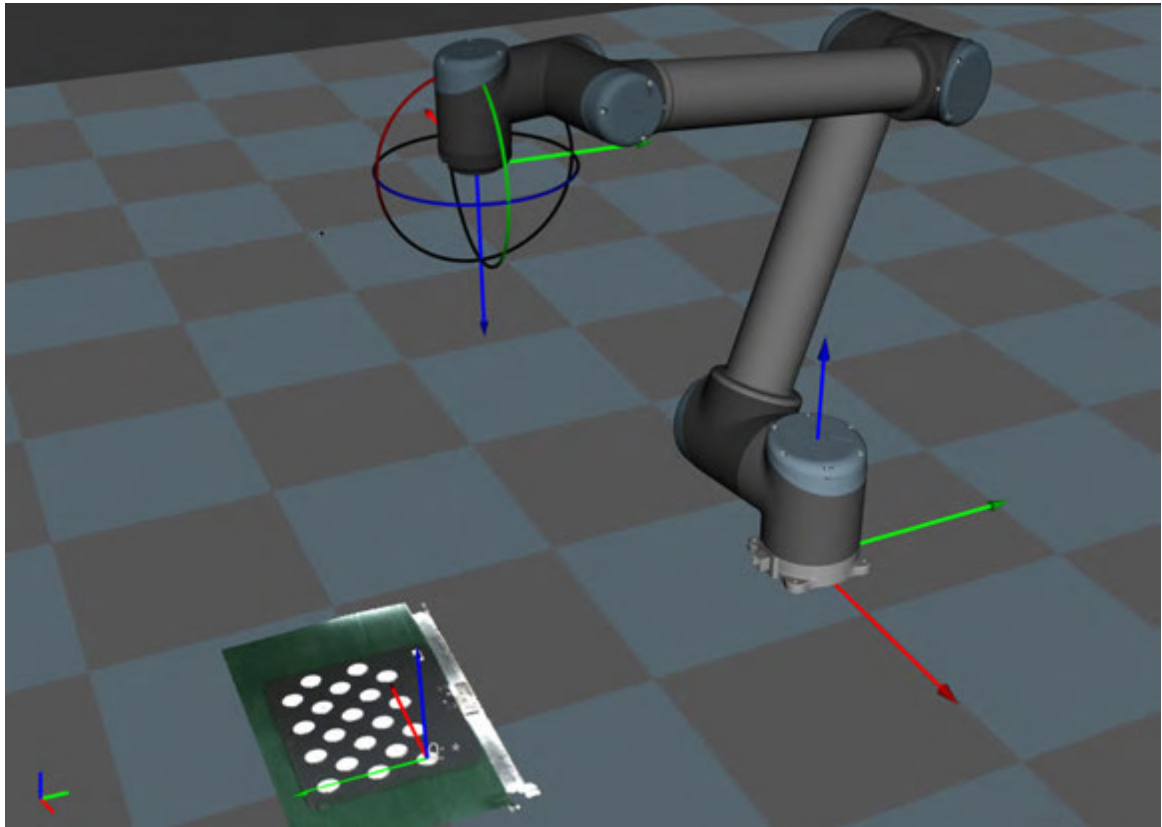
シーンビューで固定点に対するキャリブレーションボード点群のずれを確認（EIHの場合のみ）

tag::calib_result_verification_eih[] カメラの外部パラメータを計算した後、シーンビューで固定点に対するキャリブレーションボードの点群（キャリブレーションボードの白い円の十字中心）のずれを確認することで、キャリブレーション結果を大まかに判断することができます。詳細は以下の通りです。

1. キャリブレーションボードを固定位置に配置します。
2. Mech-Vizを起動し、固定点を追加し、それをキャリブレーションボードの白い円の1つの十字の中心に合わせます。
 - a. ステップライブラリから「移動」ステップを選択して、それをプロジェクト編集エリアにドラッグします。
 - b. このステップをクリックして選択し、ステップパラメータで**目標点タイプ**を「」に設定します。それがキャリブレーションボードの白い円のいずれかの十字中心と一致するように、位置姿勢のX/Y/Z座標値を調整します。
3. ロボットを制御してカメラの位置姿勢を数回変更し、キャリブレーションの**カメラパラメータを計算**手順で**[外部パラメータを再計算]**ボタンをクリックします。クリックすると、カメラが撮影を行います。
4. シーンビューで、キャリブレーションボードの点群が固定点に対して大きくなずれがあるかどうかを確認します。

キャリブレーションボードの点群が固定点に対して大きくなずれがない場合、キャリブレーション

ション結果が使用可能であることを示します。



1. EIHの場合、ロボットが接続されていないとロボットの位置姿勢はリアルタイムで取得できないため、シーンビューまたはMech-Vizに表示される正確な点群を確認するには、画像撮影時のロボットの位置姿勢を記入する必要があります。
2. シーンビューに加え、キャリブレーションボードの点群が固定点に対して大きくずれていないかどうかをMech-Vizで確認することも可能です。

```
end::calib_result_verification_eih[]
```

キャリブレーション結果の分析

キャリブレーション結果の評価指標

キャリブレーション済みの外部パラメータデータについては、精度の要求を満たすかどうかを判断する必要があります。点群誤差の基準は以下の通りです。

- 一般的なプロジェクトでは、すべてのデータポイントの誤差が3mm未満である必要があります。つまり、誤差が3mm未満のポイントが100%に達する必要があります。
- 高精度が求められるシーンでは、すべてのデータポイントの誤差が2mm未満である必要があります。つまり、誤差が2mm未満のポイントが100%に達する必要があります。
- パレタイジングのシーンでは、すべてのデータポイントの誤差が5mm未満である必要があります。つまり、誤差が5mm未満のポイントが100%に達する必要があります。

上記は基準値であり、具体的には作業現場の工程精度の要求に従うものとします。

キャリブレーション精度に影響する要因

キャリブレーション済みの外部パラメータデータの精度が要求を満たさない場合は、誤差の原因を突き止める必要があります。以下の通りに、キャリブレーションデータを1つずつ確認し、原因をチェックしてください。

キャリブレーションデータ問題の有無を確認

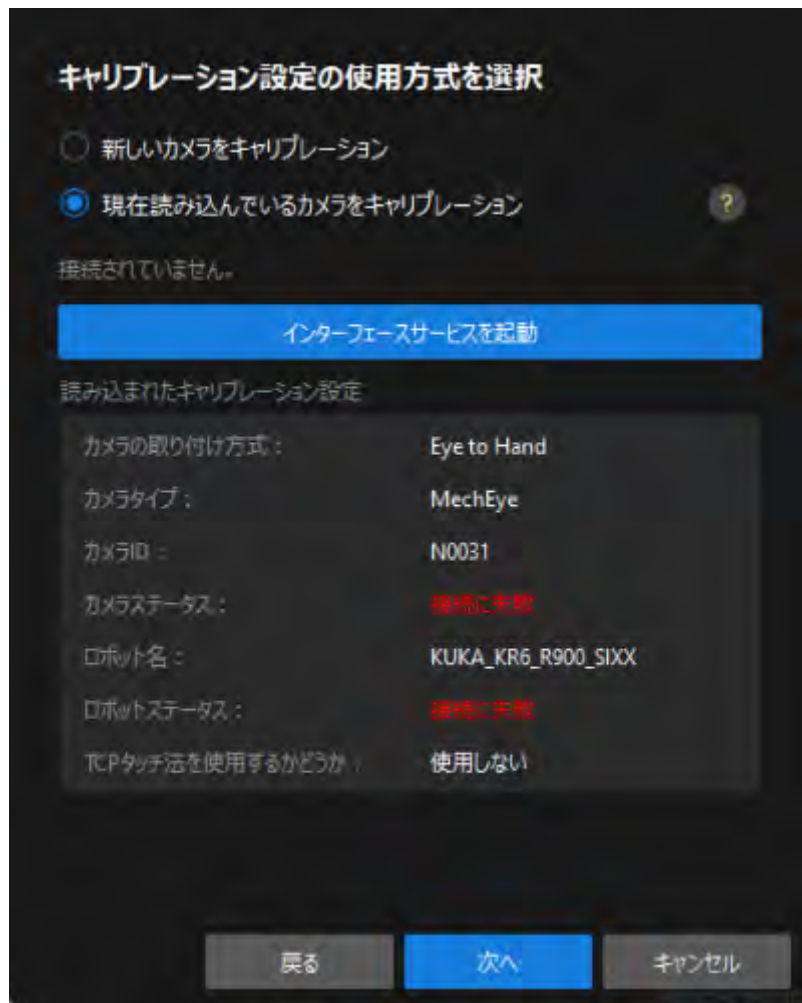
キャリブレーションデータとは、キャリブレーション中に生成された **キャリブレーションポイント** のデータのことを指します。それは、カメラIDフォルダの下のcalib_data.jsonファイルに格納されています。このファイルには、キャリブレーション経路点のフランジ位置姿勢、 **キャリブレーションボードの白い円** のデータなどが含まれています。

キャリブレーションポイントのデータを確認するには、以下の手順を実行します。

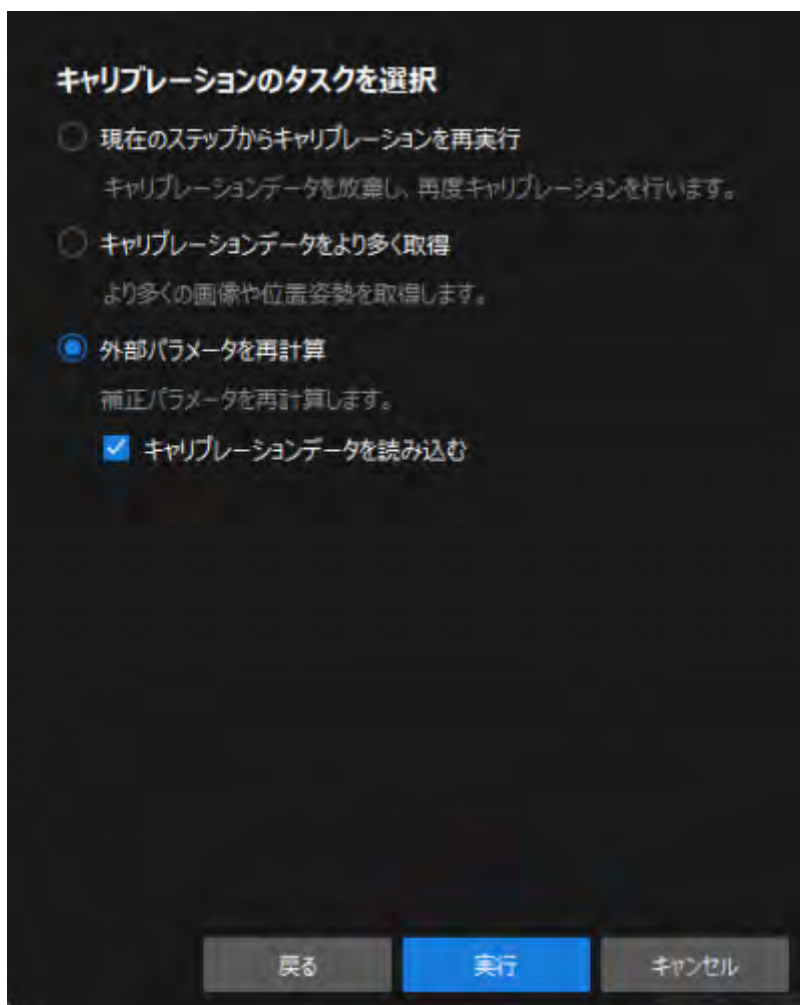
1. Mech-Visionを起動し、ツールバーで[**カメラキャリブレーション（標準モード）**]をクリックします。すると、**キャリブレーション前の設定画面**が表示されます。
2. **キャリブレーションの実行方法を選択** 画面で、 **既存のキャリブレーションパラメータを使用** を選択し、キャリブレーションのパラメータファイルを選択してから[**次へ**]をクリックします。



3. パラメータの使用法を選択 画面で、現在読み込んでいるカメラをキャリブレーション を選択してから[次へ]をクリックします。



4. キャリブレーションのタスクを選択 画面で、外部パラメータを再計算 を選択し、キャリブレーションデータを読み込む にチェックを入れてから [実行] をクリックします。



5. キャリブレーションのカメラパラメータを計算 手順で、画像と位置姿勢一覧 からキャリブレーションポイント（pose_000など）を1つ選択し、**キャリブレーションボードの白い円の情報** パネルでキャリブレーションポイントのフランジ位置姿勢とキャリブレーションボードの白い円のデータを確認します。

画像と位置姿勢一覧		画像ビュー キャリブレーションボードの白い円の情報 点群ビュー シーンビュー				
N0031		キャリブレーションボードの白い円の中心のピクセル座標とカメラ座標系での座標				
pose_000 pose_001 pose_002 pose_003 pose_004 pose_005 pose_006 pose_008 pose_009 pose_010 pose_011 pose_012 pose_013 pose_014 pose_015 pose_016 pose_017 pose_018 pose_019 pose_020 pose_021 pose_022 pose_023 pose_024 pose_025 pose_026 pose_027 pose_028 pose_029 pose_030 pose_031 pose_032 pose_033		X (px)	Y (px)	camera frame: X (r)	camera frame: Y (r)	camera frame: Z (r)
	1	1063.16	702.749	0.270909	-0.0794156	2.1468
	2	1062.02	786.623	0.26968	0.0212721	2.14786
	3	1060.85	870.484	0.268493	0.122082	2.1496
	4	1059.61	954.195	0.267249	0.2229	2.15159
	5	1020.59	744.21	0.220049	-0.0296808	2.14907
	6	1019.42	827.992	0.218753	0.0710287	2.15016
	7	1018.27	911.747	0.217604	0.171904	2.15246
	8	1017.06	995.205	0.216283	0.272524	2.15386
	9	979.093	701.717	0.170373	-0.0808309	2.15147
	10	977.971	785.579	0.169002	0.0200492	2.15121
	11	976.926	869.215	0.167864	0.120731	2.15272
	12	975.711	952.828	0.166484	0.221483	2.15379
	13	936.591	743.113	0.119384	-0.0310686	2.15397
	14	935.506	826.86	0.1181	0.069805	2.15438
	15	934.388	910.48	0.116768	0.170554	2.15467
	16	933.33	993.91	0.115558	0.271217	2.15586
	17	895.37	700.638	0.0697837	-0.082276	2.15533
	18	894.158	784.293	0.0683105	0.0185343	2.15494
	19	893.156	867.979	0.0671317	0.119418	2.15586
	20	892.028	951.575	0.0657991	0.220276	2.15675



下図では、X (pixel)とY (pixel)列は、キャリブレーションボードが現在のキャリブレーションボード経路点での2D画像におけるキャリブレーションボードの白い円の中心ピクセル座標を示します。Camera frame：X(m)/Y(m)/Z(m)列は、キャリブレーションボードが現在のキャリブレーションボード経路点での深度画像におけるキャリブレーションボードの白い円の中心がカメラ座標系での座標を示します。

キャリブレーションポイントのデータは、3つの方法で確認します。

1. ピラミッド型の経路のキャリブレーションポイントのフランジ位置姿勢において、前後のオイラー角の変化を確認



ピラミッド型の経路のキャリブレーションポイントのみが含まれ、回転のキャリブレーションポイントは含まれないことに注意してください。このチェックは、標準キャリブレーション法でキャリブレーションデータを取得する場合にのみ適しています。画像と位置姿勢一覧 からキャリブレーションポイント（pose_000など）を1つ選択し、**キャリブレーションボードの白い円の情報** パネルでキャリブレーションポイントのフランジ位置姿勢を確認します。ピラミッド型の経路に沿って移動する際、ロボットは自身のベース座標系またはフランジ座標系に沿って移動し、回転はしないので、オイラー角は基本的に変わりません。

ロボットの精度によっては、位置姿勢が上下に変動することがあります。変動が1度以上の場合、ロボット自体の原点が失われたり、ロボットの精度が低下することがあります。

解決策：この場合には、ハンド・アイ・キャリブレーションは使用できません。ロボ

ットの原点をチェック（再位置決め）して、問題を解決しなければ、続いてキャリブレーションすることはできません。

2. キャリブレーションポイントの「キャリブレーションボードの白い円の中心距離の測定値の平均誤差」が標準値を超えたかどうかを確認

画像と位置姿勢一覧 パネルからキャリブレーションポイントを選択し、画面の下部でキャリブレーションボードの白い円の中心距離の測定値の平均誤差を確認します。



誤差が上限を超えると、キャリブレーションポイントが黄色に変わり、警告が表示されます。キャリブレーションボードの白い円の中心距離の測定値の平均誤差は、内部パラメータのチェック結果の一部として、現在のカメラ内部パラメータの誤差値を部分的に反映させるために使用されます。



解決策：高精度が求められない場合では、標準値からわずかに外れた内部パラメータでも使用することができます。しかし、高い精度が求められるや大きな誤差がある場合は、カメラの内部パラメータを再キャリブレーションするか、メーカーに返送して交換することを推奨しています。



内部パラメータの誤差はキャリブレーションの結果に影響を与えますが、誤差が基準値以内であれば、一般的には問題ありません。

3. キャリブレーションポイントの「キャリブレーションボードの白い円の点群変動」が3mmを超えたかどうかを確認

画像と位置姿勢一覧 パネルからキャリブレーションポイントを選択し、画面の下部で **キャリブレーションボードの白い円の点群変動** を確認します。**キャリブレーションボードの白い円の点群変動** は、現在のキャリブレーションポイントですべてのキャリブレーションボードの白い円の中心座標がフィッティングする平面での変動を統計します。その大きさは外部パラメータの精度に直接影響します。



求められる精度が高いほど、点群変動の値は小さくなります。通常、最大値を3mm以上に設定しないでください。3つ以上のキャリブレーションポイントの点群変動の最大値が3mm

を超える場合、全体的な精度が保証できなくなります。この時点に、問題を突き止めて解決したら、キャリブレーションを再度実行する必要があります。

キャリブレーションボードの白い円の点群に変動が発生する原因は以下の通りです。

原因1：カメラの3D露光パラメータが正しく設定されていないため、点群に大きな変動が発生します。**点群平滑化**と**ノイズ除去**を有効にしないか、**カメラゲイン**を有効にします。**解決策：****点群平滑化**と**ノイズ除去**を有効にし、**カメラゲイン**を無効にすることで3D露出を調整します。調整してもキャリブレーションボードの白い円の点群が抜けている場合、遮光が必要となります。**原因2：**ETH方式で自動キャリブレーションを実行する際、キャリブレーションボードがロボットフランジにしっかりと固定されていません。キャリブレーション中にロボット動作が速すぎるため、カメラ撮影時にキャリブレーションボードが揺れ、点群に大きな変動が発生します。**解決策：**ロボットの移動速度を下げ、キャリブレーションボードをしっかりと取り付け、カメラ撮影の待ち時間を長くします。

補正パラメータを再計算

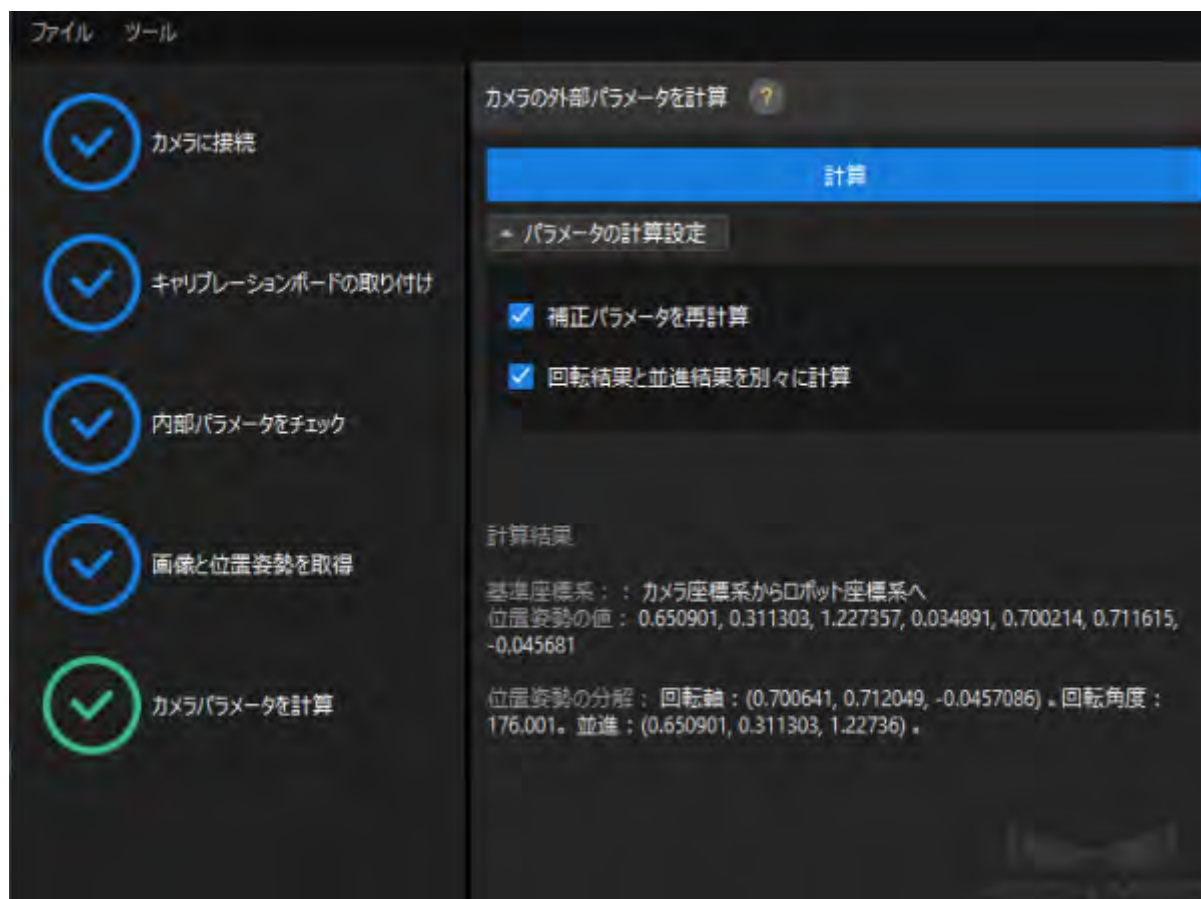
補正パラメータとは、キャリブレーション領域に対してブロック分割によって計算したパラメータ値を指します。通常は、工場出荷時にカメラの実際の最大稼働範囲に合わせた値を表示します。



実際の作業現場のキャリブレーション条件がカメラ出荷時と一致しない可能性もあるため、工場出荷時の補正パラメータが適用しない可能性があります。

上記の順序で一つずつキャリブレーションデータを確認し、正しいことを確認しても、まだ予期通りの結果が得られない場合は、補正パラメータを再計算することができます。

補正パラメータを再計算するには、キャリブレーションの**カメラパラメータを計算**手順で、**補正パラメータを再計算**にチェックを入れ、**[カメラの外部パラメータを計算]**をクリックします。



カメラパラメータを再計算する前に、元の外部パラメータファイル（extri_param.json）をバックアップし、それを工場出荷時の補正パラメータとしてコメントを付けてください。

4.4.10.5. よくある問題と解決策

工場出荷時の補正パラメータが無効な場合の解決策

工場出荷時の補正パラメータの説明

下図に示すように、補正パラメータがカメラの外部パラメータファイルextri_param.jsonに格納されています。


```

1  {
2      "depthInBase": [
3          1.432638082659246,
4          1.8377671238758124,
5          3.2883363420159943,
6          -0.009108137087217968,
7          0.7098468663853315,
8          -0.7042887897061081,
9          0.0034305708025669807
10     ],
11     "depthToCloud_offset": {
12         "Region0": {
13             "Layer0": [
14                 1.0077658936848384,
15                 0.0012689124404129978,
16                 0.0004306474909915748,
17                 2.3869795029440866e-15,
18                 0.0007093776048349514,
19                 1.0071560125601289,
20                 0.0004638496105526385,
21                 -4.440892098500626e-16,
22                 0.0017282657096284915,
23                 0.002712867205123279,
24                 1.0144842496796416,
25                 4.440892098500626e-16,
26                 -0.005293848730093753,
27                 -0.004644376122671545,
28                 -0.038295841686403695,
29                 0.99999999999999942
30             ],
31             "Layer1": [
32                 1.0077658936848384,
33                 0.0012689124404129978,
34                 0.0004306474909915748,
35                 2.3869795029440866e-15,
36                 0.0007093776048349514,
37                 1.0071560125601289,
38                 0.0004638496105526385,
39                 -4.440892098500626e-16,
40                 0.0017282657096284915,
41                 0.002712867205123279,
42                 1.0144842496796416,
43                 4.440892098500626e-16,
44                 -0.005293848730093753,
45                 -0.004644376122671545,
46                 -0.038295841686403695,
47                 0.99999999999999942
48         ],

```

下図に示すように、カメラに補正パラメータがない場合、その補正パラメータ値は単位行列になります。

```
"depthToCloud_offset": {
  "Region0": {
    "Layer0": [
      1,
      0,
      0,
      0,
      0,
      1,
      0,
      0,
      0,
      0,
      1,
      0,
      0,
      0,
      0,
      1
    ]
  }
}
```

「補正パラメータを再計算」にチェックを入れて計算すると、現在のすべてのキャリブレーションポイントのデータによって最適な補正パラメータがフィッティングされ、外部パラメータファイルの補正パラメータが更新されます。

工場出荷時の補正パラメータ：工場出荷時の補正パラメータとは、カメラの焦点距離に応じてカメラの最大視野範囲における計算された補正值です。工場出荷時の補正パラメータを使用することのメリットとデメリットは以下の通りです。

- メリットは、工場出荷時の補正パラメータを使用すれば、キャリブレーションポイントの個数が少なく、視野範囲が限られる場合に作業領域全体の精度を表す外部パラメータを計算できることです。TCPタッチ法を使用し、ロボットにキャリブレーションボードを取り付けた後、キャリブレーションの作業領域が制限され、作業領域が完全にキャリブレーションできない場合、工場出荷時の補正パラメータを使用することを推奨します。
- デメリットは、現場でロボットとキャリブレーション経路などが異なり、工場出荷時の補正パラメータを使用しては毎回も最適な結果が得られることを保証できないことです。作業領域を完全にキャリブレーションした場合、補正パラメータを再計算することで、より良い結果を得ることができます。



工場出荷時の補正パラメータを使用して精度要件を満たした外部パラメータを取得した後、作業領域を完全にキャリブレーションしていない場合、より高い精度のためにも、「補正パラメータを再計算」にチェックを入れて再計算することは推奨しません。

無効な工場出荷時の補正パラメータは、現場で工場出荷時の補正パラメータがバックアップせ

ずに更新された補正パラメータや、不明なカメラに工場出荷時補正パラメータなどが含まれています。

TCPタッチ法を使用した場合の解決策

この場合、不正確な補正パラメータに対し、複数セットのタッチポイントを追加して統合計算することで解決できます。

複数セットのタッチポイントを追加することは、キャリブレーションボードを作業領域の最下部中央に配置し、TCP先端を3つのポイントにタッチさせ、カメラで撮影して1セットのタッチポイントとします。その後、キャリブレーションボードをエッジに配置し、ポイントをタッチしてもう1セットのタッチポイントを生成します。上記のように、異なる層で複数セットのタッチポイントを追加します（通常、3つの層で、層ごとに2セットのタッチポイントが十分です）。

標準キャリブレーション法を使用した場合の解決策

標準キャリブレーション法を使用する場合、不正確な補正パラメータに対し、作業領域を完全にキャリブレーションして、**[補正パラメータを再計算]**にチェックを入れることで解決できます。

6軸以外のロボットのキャリブレーション

4軸ロボット

4軸ロボットは、一般的に直行ロボット、SCARAロボット、パレタイジングロボットに分けられます。

使用頻度に応じて、Mech-Vizは少数のSCARAロボット及びパレタイジングロボットにのみ対応します（対応範囲は今後拡大します）。

● TCPタッチ法を使用した場合のTCP先端の取り付け問題

4軸目にTCP先端を取り付けます。4軸目を回転させて、TCP先端が一点で固定しているかを確認して、フランジ中央に取り付けられているかを判断します。4軸ロボットのTCPはキャリブレーションにより取得できないので、XYZ値は手動で測定する必要があります。

● 4軸ロボットキャリブレーションにおけるZ方向を調整する方法

標準キャリブレーション法を使用する場合、4軸ロボットの回転自由度が不足しているため、キャリブレーション中に回転値が不十分です。キャリブレーション完了後、外部パラメータのZ方向を手動で調整する必要があります。

1.ロボットのベース座標の位置を確定します。

2.キャリブレーションボードを、ロボットのベース座標のXY平面と平行な作業面（通常は地面、作業台など）に配置します。

3.下図に示すように、ロボットのベース座標から作業面までの距離を測定します。Mech-Vizで地面の高さを、対象物の位置姿勢のZ方向とロボットのベース座標から作業面までの距離と同じ数値に設定します。下図のように示します。

```
+ image::calibration-reference-adjust_z.png[align="center"]
```

4.外部パラメータのZ方向を調整して、キャリブレーションボードの点群がMech-Vizの作業平面上に位置するようにして、Z方向の調整が完了します。

● 直行ロボットの使用説明

直行ロボットは種類によって、ベース座標の位置が異なります。そのため、ベース座標の位置に基づいてZ方向を調整するのは難しいので、トラスロボットの場合、TCPタッチ法のみを推奨します。

最も使用されるのは、カメラを3軸目に取り付ける方法です。

EIH方式を使用する場合、直行ロボットが3軸ロボットとなり、4軸目は使用できないため、Adapterを介して固定角度を送信するしかありません。

ETH方式を使用する場合、写真撮影点の個数には限りがあり、各写真撮影点間のオフセットは既知です。異なる写真撮影点を使用するたびに、ロボットのベース座標オフセットをAdapterに追加します。

7軸ロボット/スライドレール付き6軸ロボット/5軸ロボット



ここで、スライドレール付き6軸ロボットは、スライドレールがロボットティーチングペンダントに統合されているため、7軸ロボットになります。

上記の3種類のロボットを使用する場合、TCPタッチ法を最も推奨します。

● 標準キャリブレーション法を使用する7軸ロボットの操作要件

現場に適切なTCP先端がない場合、またはTCP先端を固定できない場合、7軸ロボットに対して、標準キャリブレーション法を使用してキャリブレーションできます。

キャリブレーション中に、1つの軸の移動を制限して、6軸ロボットとして操作します。他の操作は、基本的に6軸ロボットのキャリブレーションと同じです。

ロボットのオイラー角タイプが不明な場合のキャリブレーション方法

ロボットのオイラー角が不明な場合、カメラキャリブレーションでの「オイラー角のタイプを取得」ツールを使用して現在のロボットのオイラー角タイプを取得します。下図のように示します。



ロボットを3つの異なる位置姿勢で、ロボットハンドに取り付けられたTCP先端を作業面上のTCP先端にタッチさせます。タッチするたびに、ティーチングペンダントに表示されたロボットの位置姿勢を入力します。その後、「オイラー角のタイプを取得」をクリックして、推奨されるオイラー角を取得します。

現場では、TCPが既知のロボットハンドに取り付けられたTCP先端だけがあり、適切な固定TCP先端がない場合

正確なロボットのオイラー角タイプが取得できないため、TCPタッチ法を使用する時、オイラー角を正確なタイプで入力できないことがあります。

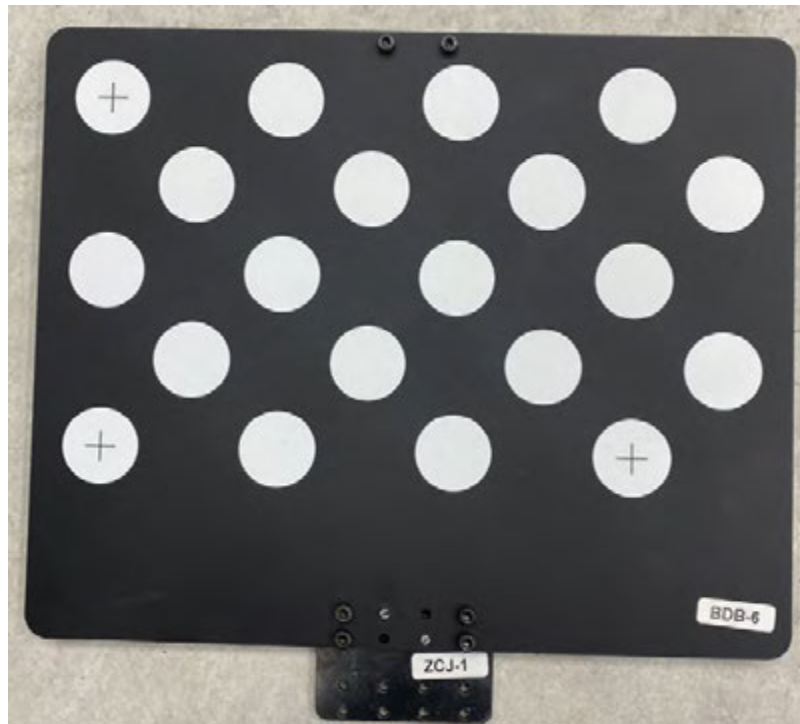
この場合、ティーチペンダントで位置姿勢をTCPに切り替えて、即ち、TCP先端の位置姿勢を読み取ります。

順次に3つのポイントにタッチし、ティーチングペンダントのXYZの値を読み取って入力します。いずれかのオイラー角タイプを選択し、いずれかの定数値を入力します。入力した3つの位

置姿勢のオイラー角が一致するようにしてください。

長距離のキャリブレーション時、大量のキャリブレーションポイントの点群変動が許容範囲を超え、カメラパラメータを調整しても改善できない場合の解決策

下図に示すように、キャリブレーションボードの状態を確認して、点群に大きな変動を引き起こし得るクロスがあるかどうかを確認します。



A4用紙などを切って、そのキャリブレーションボードの白い円の中心に貼り付けて（エッジではなく、通信のみに貼ります）。クロスによる点群の変動を軽減します。

上記の方法で軽減できまい場合、手動で位置姿勢を追加してから、点群変動が大きい位置姿勢を削除してみてください。

キャリブレーション経路についてよくある誤解

● キャリブレーションポイントの数が多いほど結果が良くなるか

キャリブレーションポイントが多すぎると、異常点が発生し、全体的な誤差率が高くなる可能性があります。

工場出荷時の補正パラメータを使用して計算する場合、カメラの焦点距離、キャリブレーションボードのサイズなどに応じて、各層のキャリブレーションポイントの個数を決定します。

- 焦点距離が300-2000mmの場合、2*2を推奨します。層ごとに4つのキャリブレーションポイント

ントを設定します。通常は3層ですが、パレットのタイプによっては4層を使用することもあります。

- 焦点距離が2000-3500mmの場合、3*3を推奨します。層ごとに9つのキャリブレーションポイントを設定します。通常は3層ですが、パレットのタイプによっては4、5層を使用することもあります。
- **キャリブレーション範囲はすべての作業領域を含める必要があるか**

工場出荷時の補正パラメータを使用する場合、カメラの焦点距離を中心位置として、焦点距離の範囲内で層ごとにキャリブレーションします。

工場出荷時の補正パラメータを使用しない場合、または条件が不十分でカメラの焦点距離の範囲内でキャリブレーションできないときに、作業領域を完全キャリブレーションすることを推奨します。

- **キャリブレーション済みの領域における外部パラメータは正確ですが、それ以外の領域における外部パラメータは不正確か**

キャリブレーション後に生成された誤差点群は、キャリブレーション済みの領域における外部パラメータの誤差を表示しますが、キャリブレーションしていない領域における外部パラメータが不正確とは限りません。逆に、基本的に、工場出荷時の補正パラメータを使用する場合、キャリブレーションしていない領域における外部パラメータも正確です。

4.5. 測定モードの使用ガイド

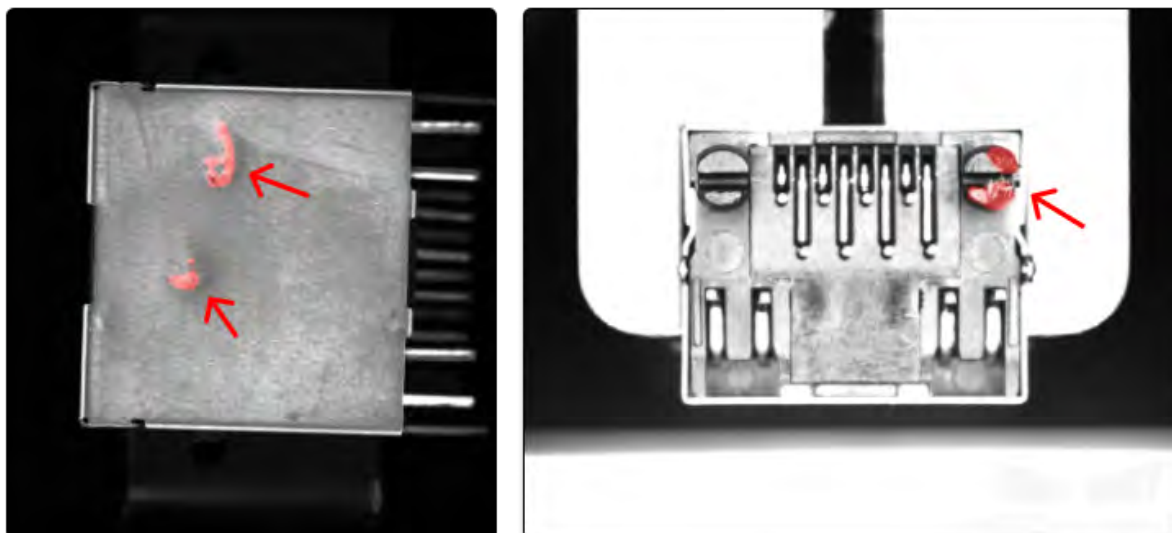
測定モードについて

測定、検出、位置合わせ、認識は工業業界におけるマシンビジョンの最も一般的な応用方向です。

Mech-Mind株式会社が提供する測定モードは、電子機器製造、リチウム電池、自動車、家電製品などの業界に適しています。

測定モードには、多種類の測定アルゴリズムツールが含まれ、描画エリアに基づくクイックデバッグに対応し、オペレーター・インターフェイス（カスタム）があり、実際のニーズに応じて**精密測定**および**欠陥検出**を迅速に展開できます。

下図は、コネクタの品質検査（測定+検査）のイメージ図です。



下図は、オペレーター・インターフェイス（カスタム）です。



測定モードの使用を開始して最初の応用を完成するには、以下の内容をお読みください。

測定モードを使ってみる

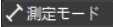
測定モードの**主要なアルゴリズム**については、以下の内容をお読みください。

主要なアルゴリズムの紹介

4.5.1. 測定モードを使ってみる

ユーザーは円形の部分を備えた部品を使用して最初の応用を完了します。つまり、部品の円形部分の寸法（円の半径）が合格するかどうかを検出します。

プロジェクトを新規作成して測定モードに入る

1. **設定** > **オプション** > **詳細設定**を順番にクリックし、「測定モード」をチェックしてソフトウェアを再起動します。
2. **ファイル** > **プロジェクトを新規作成**をクリックします。
3. ツールバーの  をクリックして測定モードに入ります。

入力

測定モードで生データを取得するには、いくつかの方法があります。

- [カメラから画像を取得](#)
- ラインスキャンレーザーカメラ
- [点群を読み取る \(V2\)](#)
- [画像を読み取る \(V2\)](#)

（2.5Dアルゴリズム処理の場合、「ラインスキャンレーザーカメラ」および [点群を読み取る \(V2\)](#) の2つのステップが [正投影](#) と併用します）。

本例では、[カメラから画像を取得](#) を使用して元の画像を取得します。

1. ステップライブラリから [カメラから画像を取得](#) を選択して、それをプロジェクト編集エリアにドラッグします。



ステップライブラリの右下隅にある[>]をクリックすると、表示モードを切り替えます。

2. [カメラから画像を取得](#) を参照して、このステップのパラメータバーで関連パラメータを設定します。

処理

本例では部品の円形部分の寸法が合格するかどうかを検出することであるため、[円の測定](#) を使用します。

1. ステップを探す：

ステップライブラリから [円の測定](#) を選択して、それをプロジェクト編集エリアにドラッグします。


2. ステップをつなぐ：

下図に示すように、このステップをクリックし、ソフトウェアインターフェイスの右側にある**ステップの入力・出力ポートの選択**画面で、「入力1（カラー画像）」のドロップダウンバーを「カラー画像」を選択すると、ステップは自動的につながります。



入力を選択するとき、データタイプは一致しなければなりません。一致しないデータタイプを選択すると、ポップアップウィンドウが表示されます。

3. プロジェクトの実行：

プロジェクト編集エリアの上部にある[**実行**]をクリックするか、最初のステップの  をクリックすると、画像が描画エリアに表示されます。

4. 検出する円形部分を選択：

円の測定 をクリックし、選択マーカーは画像の左上隅にあり、マウスを選択マーカーの上に移動する（最初に画像を拡大縮小することをお勧めします）と、矢印マウスが触手マウスに変わります。下図に示すように、マウスを長押ししてマーカーをドラッグし、マーカーサイズを設定します。

描画エリアは**可視化表示**、**関心領域選択**および **位置マーカー** に使用されます。

- a. 描画エリアの左上で、カラー画像、グレイ画像、二値画像の表示に切り替えることができます。
- b. 描画エリアの右上にあるツールバーで、画像の表示サイズ、ペンの幅、コントラストなどを調整できます。



画像の表示サイズはマウスホイールで調整することもできます。

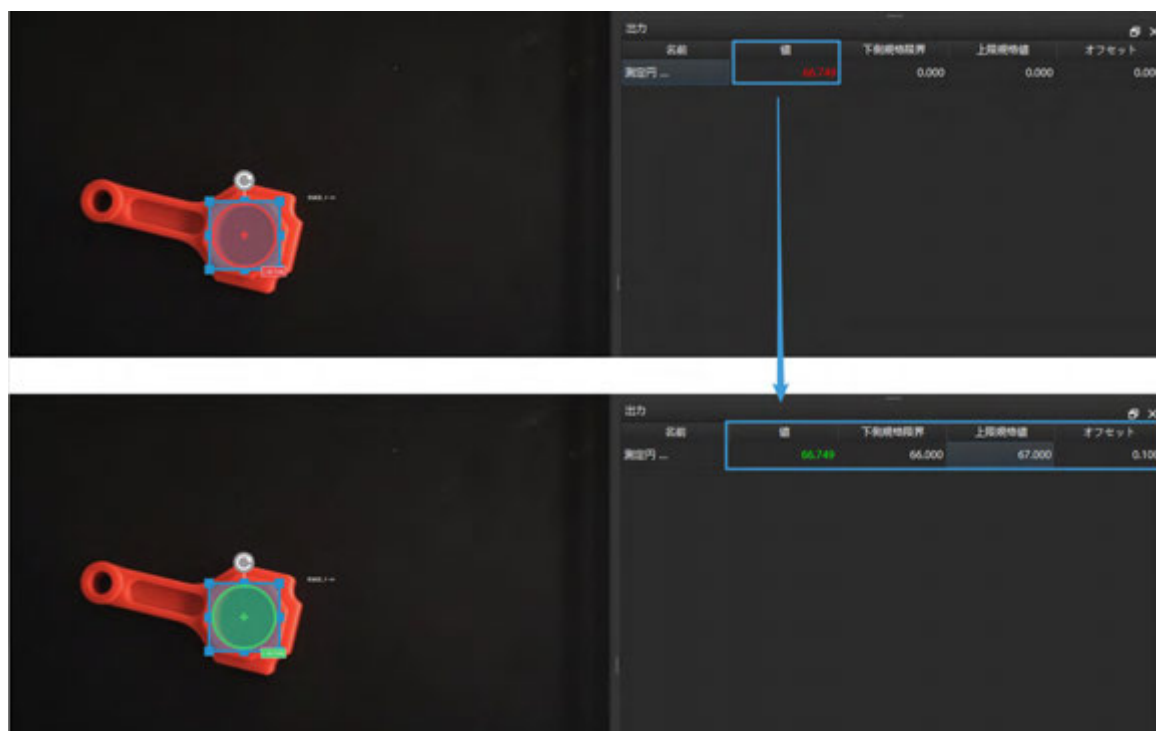
出力

1. 測定値を読み取る：

- 描画エリアに、関心領域の円形部分の測定値をリアルタイムで表示されます。
- インターフェイスの右側にある**測定結果の出力**の画面で測定値が表示されます。


2. 測定結果が合格かどうかを判断：

2つのエリアの測定値がすべて赤です。これは、上側規格限界と下側規格限界が設定されていないためです。この場合、上側規格限界と下側規格限界がそれぞれ設定されると（測定値は上側規格限界と下側規格限界の間にある必要があります）、測定値は緑色に変わります。



ユーザーが上側規格限界と下側規格限界、オフセット（合格基準）を設定し、測定値の色を確認することで、検出された物体の円形部分の寸法が合格かどうかを判断できます。

3. 結果のビュー：

測定モードでの**結果のビュー**はより見やすい判断結果を表示する方法を提供します。ツールバーの  をクリックし、ドロップダウンバーの **[結果のビュー]** をクリックします。ここでは、左上隅にアイテムが上側規格限界と下側規格限界に従って合格であるかどうか判断されて表示されます。合格はOK、不合格はNGです。



4. オペレーター・インターフェイス（カスタム）：

ツールバーの[オペレーター・インターフェイス（カスタム）]をクリックすると、オペレーター・インターフェイス（カスタム）に入ります。このインターフェイスでお客様のニーズに応じたカスタマイズをサポートしています。Mech-Mind株式会社ににお問い合わせください。

以上で最初の応用が完了しました。

一般的な使用手順

最初の応用を完了することにより、測定モードの一般的な使用手順が明らかになりました。

1. プロジェクトを新規作成して、測定モードに入ります。
2. アルゴリズム処理フローを構築し、入力ステップ、アルゴリズム処理ステップなどを設定します。測定モードの主要なアルゴリズムは、次の節で説明していきます。
3. 描画エリアで関心領域を選択し、基準点、検出点などを追加します。
4. 結果のビューに入り、出力するステップを選択して、合格基準を設定します。
5. カスタマイズされたオペレーター・インターフェイスでは、検出結果、製品画像、生産能力統計などの情報がリアルタイムで表示可能です。

4.5.2. 主要なアルゴリズムの紹介

異なるステップを使用して、様々なアルゴリズム処理を完了します。

主要なアルゴリズム：

1. 2Dアルゴリズム：2D画像に基づいて、測定や位置決めなどの機能を実現します。

関連ステップは、[円の測定](#)、[円から円までの距離を測定](#)、[円から線分までの距離を測定](#)、[点から円までの距離を測定](#)、[点から点までの距離を測定](#)、[点から線分までの距離を測定](#)、[線分間の距離を測定](#)、[最長の線分を測定](#)、[線分間の角度を測定](#) などがあります。

2. 2.5Dアルゴリズム：深度画像に基づいて、高さの差を計算します。

関連ステップは、[点から点までの高さの差を測定](#)、[点から基準線までの高さの差を測定](#)、[点から平面までの高さの差を測定](#) などがあります。

3. 3Dアルゴリズム：3D点群に基づいて距離を計算します。

関連ステップは、[対角線の長さを計算](#) などがあります。

4. 幾何公差：製品の外観の平坦度を計算するために使用されます。

関連ステップは、[平面度を計算](#)、[平行度を計算](#) などがあります。

5. ディープラーニング：ディープラーニングアルゴリズムを使用して、製品の外観の欠陥などを検出します。

4.6. 補助ツールの使用ガイド

本章では、Mech-Visionに組み込まれている補助ツールについて説明します。これらのツールを柔軟かつ合理的に使用することで、プロジェクトの実行スピードとビジョンソリューションの精度を向上させることができます。

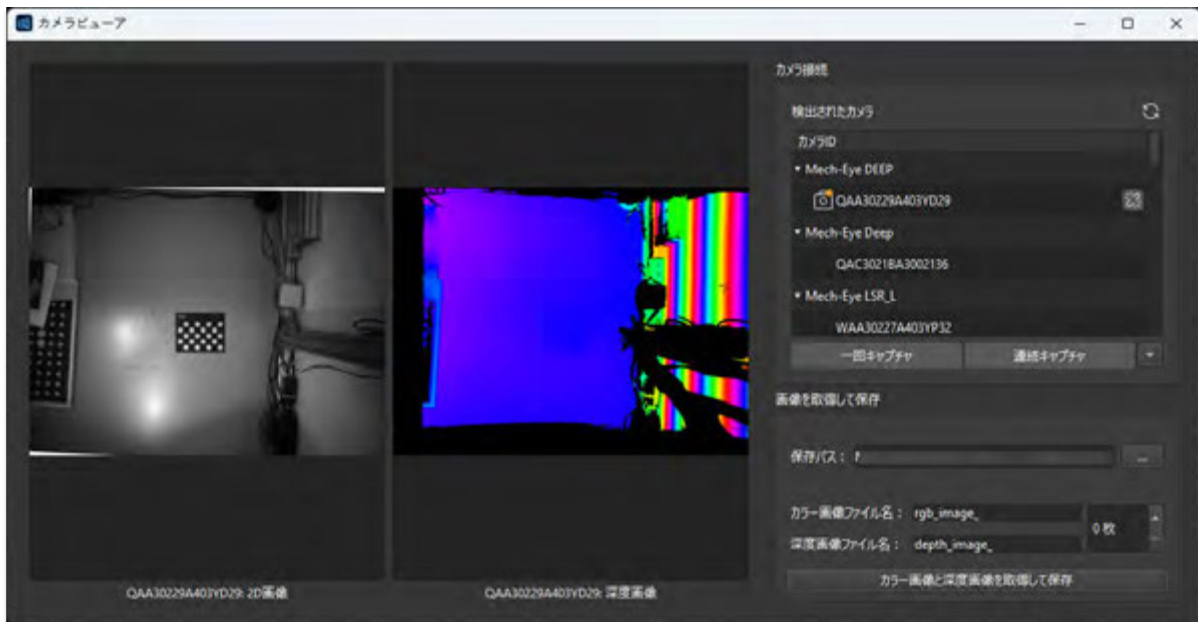
- [カメラビューア](#)
- [静的背景を取得して干渉を除去](#)
- [ROI 設定](#)
- [シーンの点群を設定](#)
- [ディープラーニングサーバー](#)
- [マッチングモデル・把持位置姿勢エディタ](#)
- [ディープラーニングモデルパッケージ管理ツール](#)
- [パラメータレシピ](#)
- [データ保存](#)
- [ログ](#)

4.6.1. カメラビューア

概要

カメラビューアは、カメラと接続して画像を撮影し、カラー画像と深度画像を表示することができます。また、保存パスを指定して画像を保存することもできます。

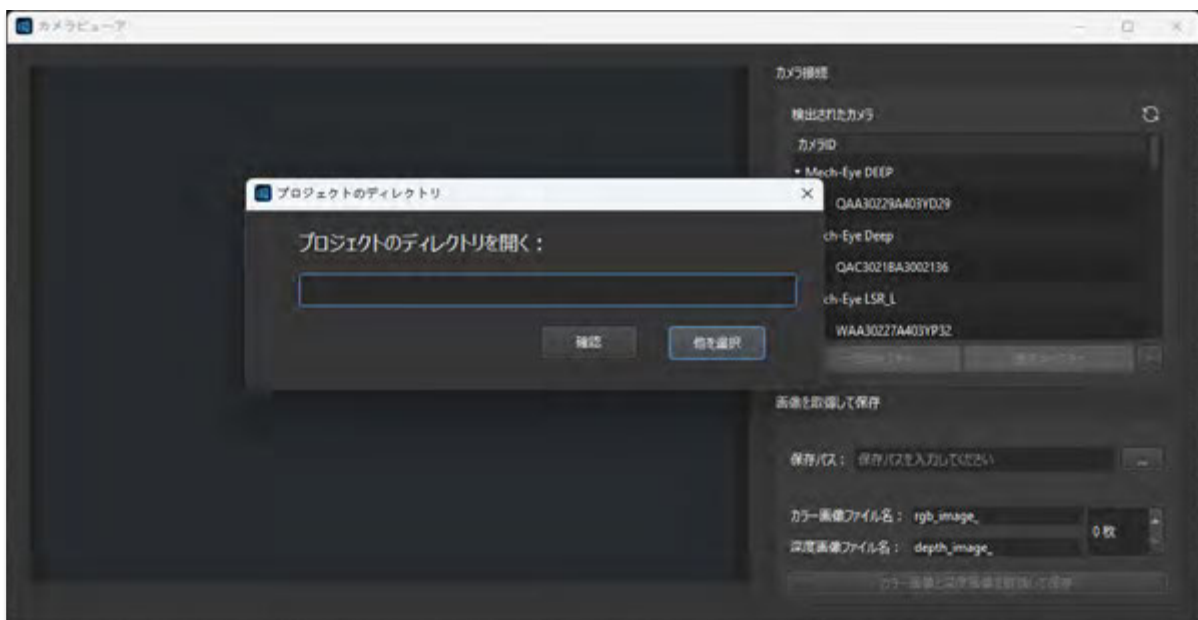
メニューバーで **カメラヘルパー** ➤ **カメラビューア** をクリックし、下図のような画面に入ります。



使用方法

カメラビューアを開く

カメラが撮影する時に、対応する内部・外部パラメータを読み込む必要があります。内部・外部パラメータが対応するプロジェクトディレクトリに格納されているため、対応するプロジェクトディレクトリを読み込めばいいです。また、ディレクトリを読み込まずにカメラビューアに入ることもできます。



カメラ接続

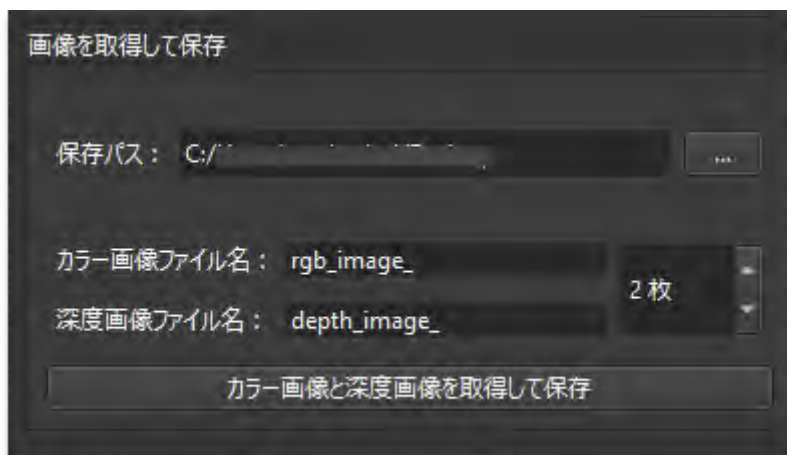
1. 検出されたカメラから接続するカメラを選択し、右側の ☒ をクリックして接続します。カメラが正常に接続された場合、☒ は ☐ に変わります。



2. 下部にある [一回キャプチャ] または [連続キャプチャ] をクリックすると、撮影が実行され、画像結果が表示されます。

画像取得と保存

画像を取得して保存する場合、**保存パス** を選択し、**カラー画像/深度画像ファイル名** と **画像名の開始番号 (X枚)** を設定して、[**カラー画像と深度画像を取得して保存**] をクリックします。すると、取得された画像が設定されたパスに保存されます。

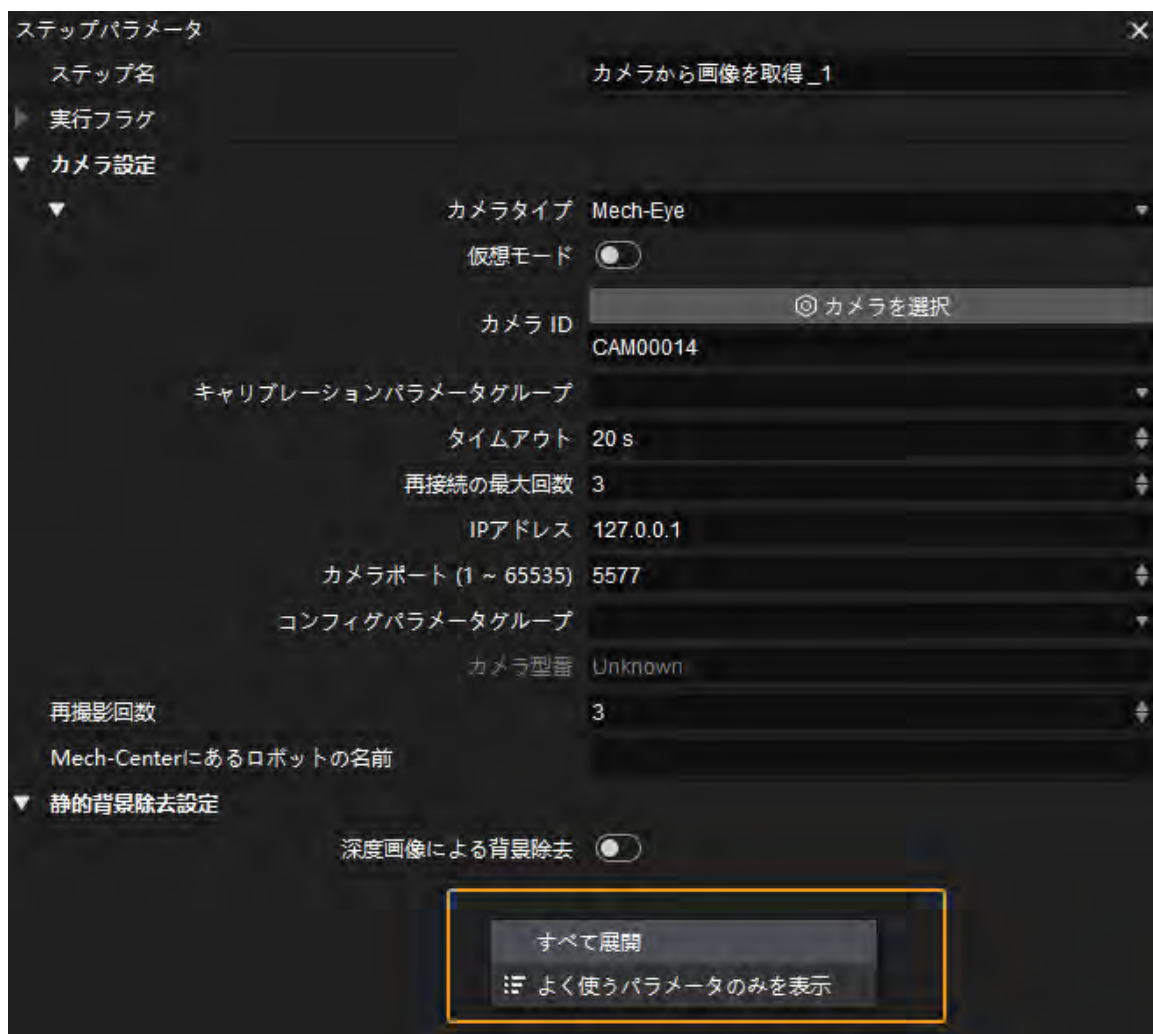


4.6.2. 静的背景を取得して干渉を除去

静的背景を設定して、背景深度の情報を取得します。それによって、後続ステップで背景の干渉を除去したり、物体の高さを算出したりすることができます。

静的背景の設定手順は以下の通りです。

1. Mech-Visionでプロジェクトを開き、**カメラから画像を取得** ステップをクリックします。下図に示すように、右側のパラメータエリアで空白場所を右クリックして、表示されるメニューから [**すべてのパラメータを表示**] を選択します。



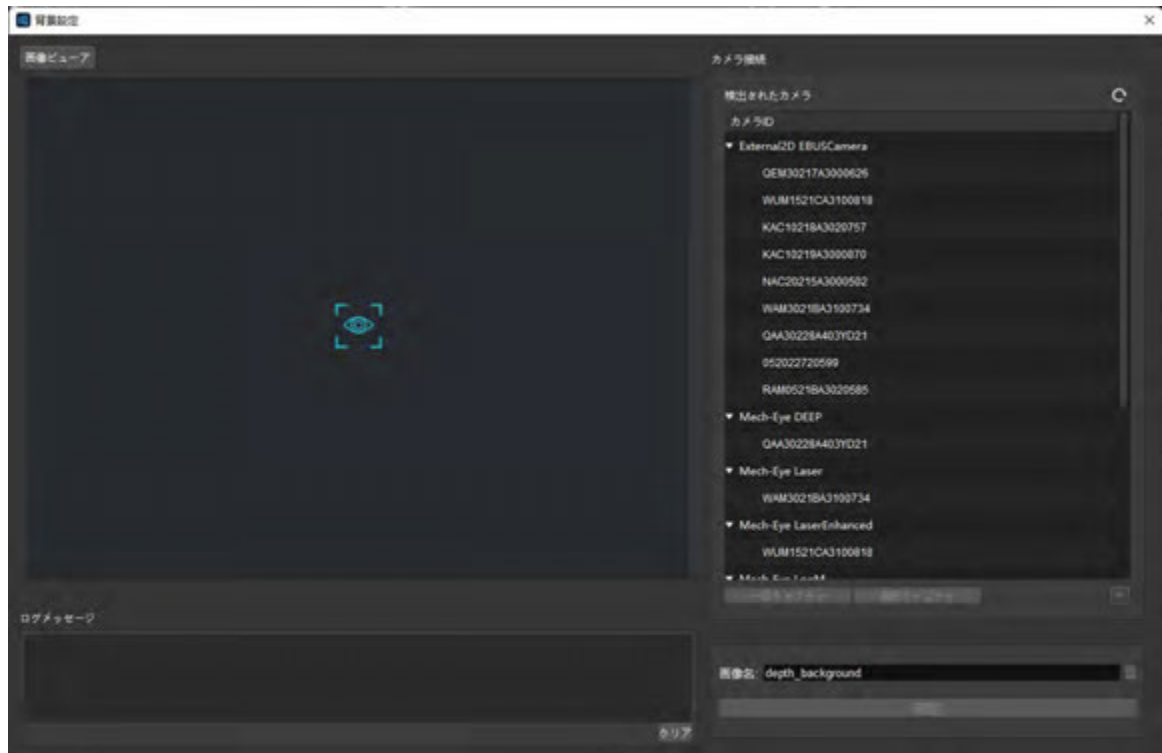
2. 下図に示すように、**カメラから画像を取得** ステップのパラメータで [深度画像による背景を除去] にチェックを入れます。




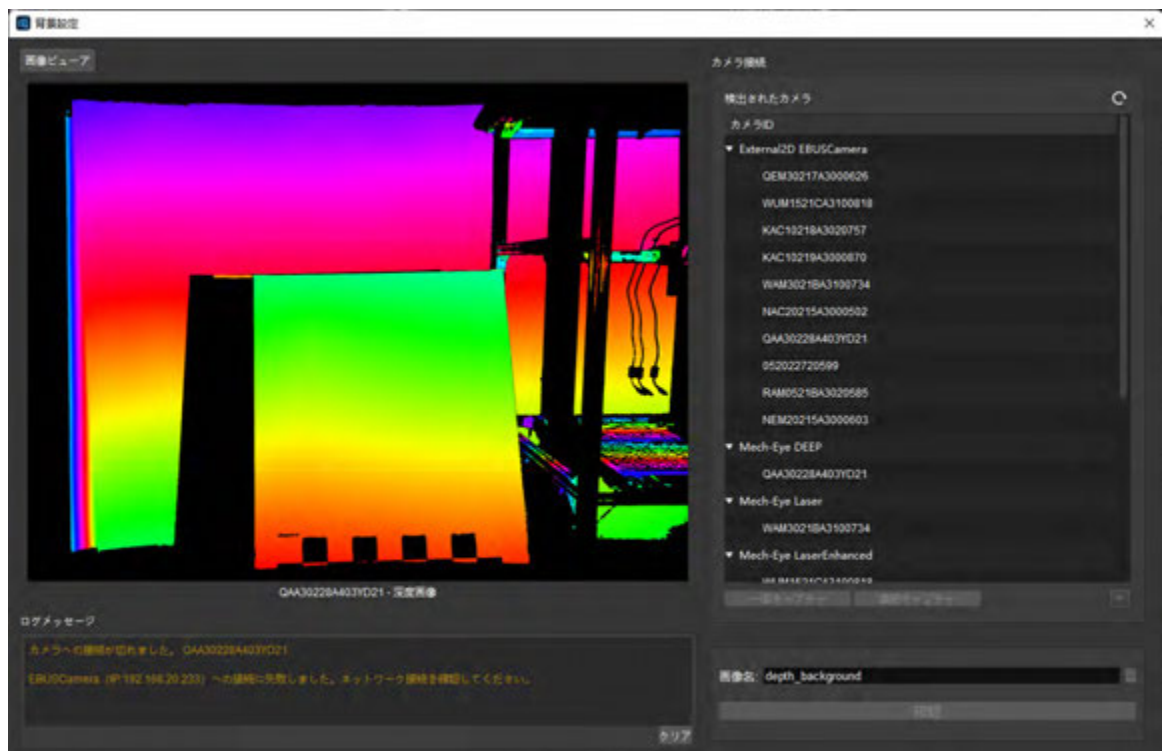
3. [深度背景画像ファイルを選択] をクリックし、背景設定ツールに入ります。



4. 下図に示すように、背景設定の画面にはカメラ接続、画像表示、情報記録、背景画像保存という4つのエリアがあります。



5. 背景を設定する前にカメラの接続が必要です。カメラ番号をダブルクリックするか、またはカメラを選択してから  をクリックして接続します。下図に示すように、正常に接続したら、[一回キャプチャ]または[連続キャプチャ]をクリックして画像をキャプチャします。必要な背景画像を取得した後、[確認]をクリックして画像を保存します。

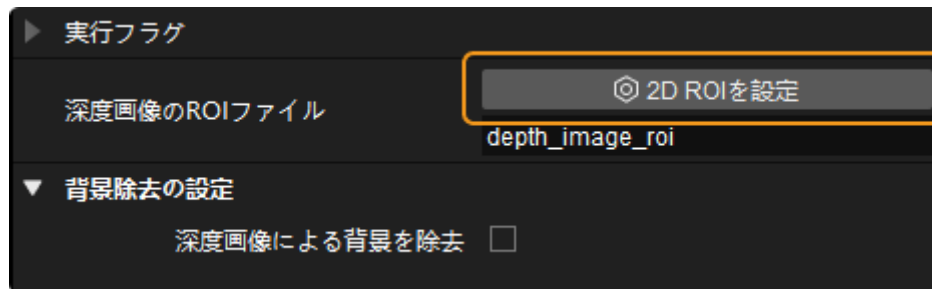


4.6.3. ROI 設定

Mech-Visionでは、2D ROIと3D ROIを設定可能です。詳細な操作手順は次の通りです。

2D ROIの設定手順

1. 下図に示すように、2D ROIを設定するステップのパラメータで[**2D ROIを設定**]をクリックしてROI設定の画面に入ります。



2. 下図に示すように、画像表示エリアで、マウスの左ボタンを押しながら、マウスをドラッグしてカラー画像または深度画像のROIを選択し、マウスの左ボタンをもう一度クリックして選択を完了します。ROIを再度選択する必要がある場合は、マウスの左ボタンをもう一度クリックしてROIを選択します。
3. 選択されたROI座標範囲がROIのパラメータで表示されます。[**確認**]をクリックして座標情報を保存し、画面が終了します。



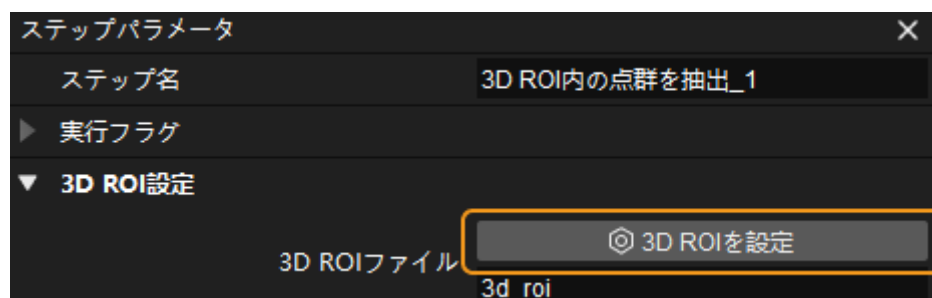
2D ROIの設定に関連するステップは、[2D ROI内の画像をスケーリン](#)、[深度画像分割](#)、[深度画像を点群に変換](#) などがあります。

3D ROIの設定手順



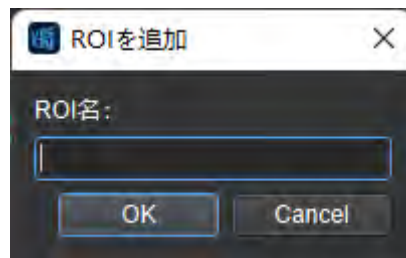
3D ROIを設定する前に、**プロジェクトアシスタント**で基準となるシーンの点群を選択してください。そうしないと、アラートウィンドウが右下隅にポップアップし、設定を行うことができません。

1. 下図に示すように、3D ROIを設定するステップのパラメータで[**3D ROIを設定**]をクリックしてROI設定の画面に入ります。



2. 下図に示すように、3D ROIの設定ツールには、点群表示、3D ROIパラメータ、表示設定、操作プロンプトの4つのエリアがあります。

- 点群表示エリアには、主にロボットベース座標系、カメラ座標系、目標エリア（3D ROI）、目標点群が含まれます。
 - 3D ROIパラメータエリアには、主に現在のROI名前、エリアサイズ、エリアの中心位置、エリアの位置姿勢、およびその他のパラメータが表示されます。
 - 表示設定エリアは、主に地面とロボットのベース座標系の表示状態を調整するために使用されます。
 - 操作プロンプトエリアは、3D ROIを設定するための操作プロンプトが表示されます。
3. 適切な位置に3D ROIを調整するために、**Ctrl** を押しながら選択フレームの面をドラッグします。そして、適切なサイズに3D ROIを調整するために、**Ctrl** を押しながら選択フレームの辺をドラッグします。また、3D ROIパラメータの中でサイズ、中心座標と回転姿勢のパラメータを調整することで、3D ROIを調整することもできます。
 4. デフォルトで生成された3D ROIを直接編集することに加えて、3D ROIを新規作成することもできます。下図に示すように、**+** をクリックすると、3D ROI追加の画面が表示されます。作成する3D ROIの名前を入力し、**[OK]** をクリックして作成を完了し、表示エリアで3D ROIを調整します。ROI名のドロップダウンメニューをクリックすれば、3D ROIを切り替えることができます。



3D ROIの設定に関連するステップは、「深度画像の3D ROI外の領域を無効に設定」、「深度画像の3D ROIにゼロ深度値の領域を抽出」、「3D ROI内の点群を抽出」、「位置姿勢が3D ROI内にあるかどうかを検証」、「3D ROI内の位置姿勢を取得」、「3D ROIの中心を読み取る」などがあります。



2D ROIと3D ROIを設定する際に以下のことに注意してください。

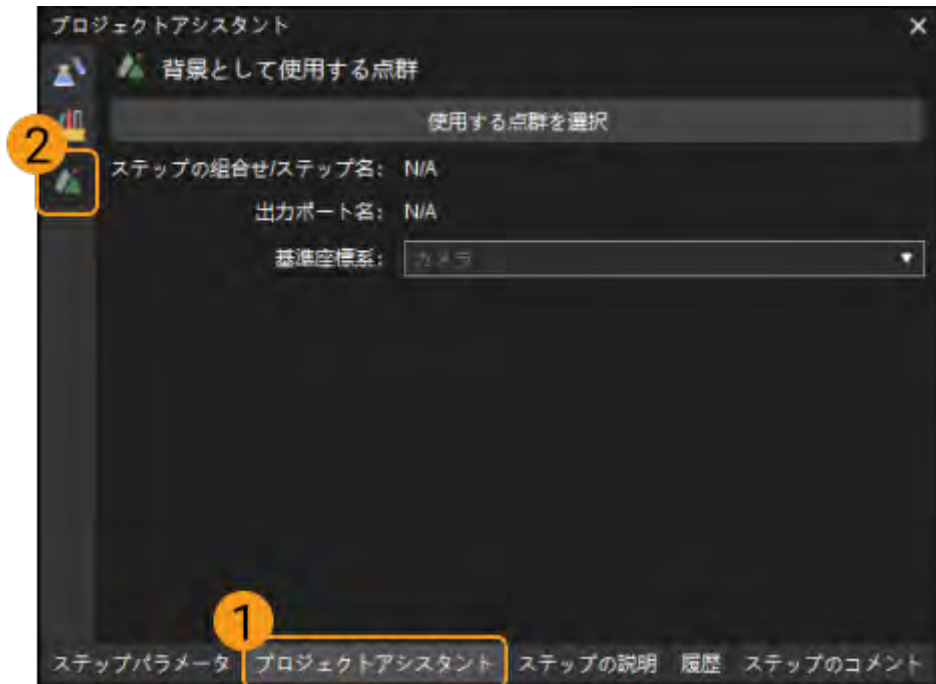
- 複数の3D ROIが作成された場合、ステップに適切な3D ROIが使用されていることを確認してください。
- 2D ROIと3D ROIを選択するとき、箱の損傷や揺れによる把持のずれを避けるために、ワークと箱をすべて選択する必要があります。また、選択範囲は応用現場よりも少し大きくする必要があります。

4.6.4. シーンの点群を設定

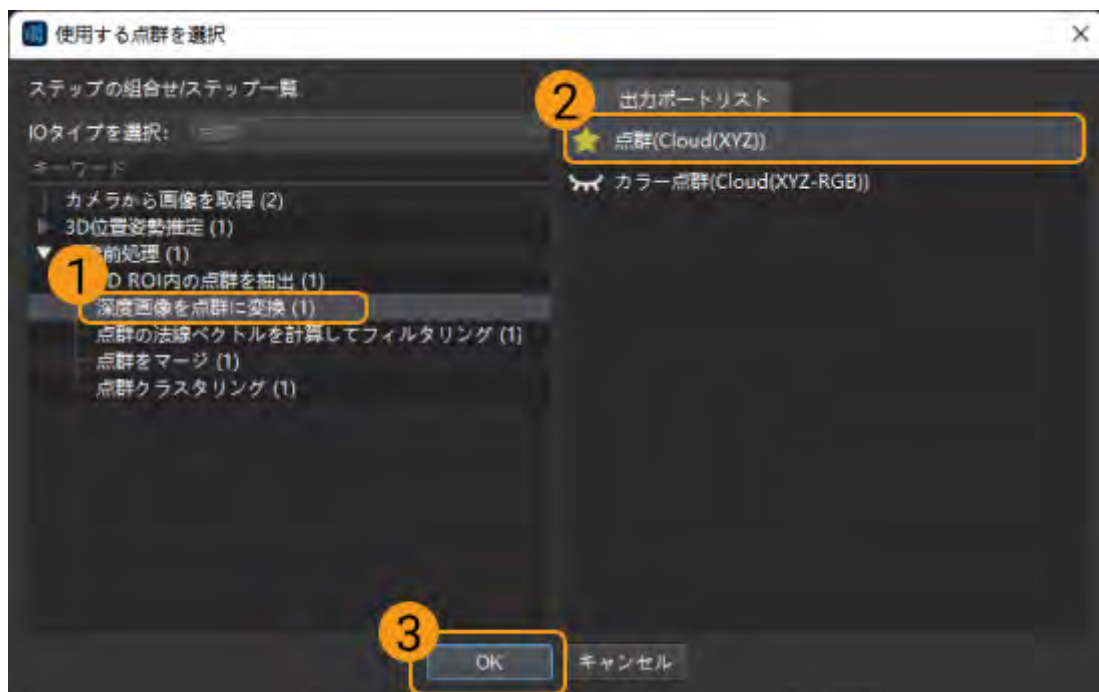
シーンの点群とは、カメラ視野内にあるすべての点群のことを指します。シーンの点群をもとにROIを設定することができるので、ROIを設定する前にシーン点群を設定する必要があります。

本節では、シーンの点群を設定する方法について説明します。

1. [プロジェクトアシスタント]をクリックし、をクリックしてシーンの点群の設定画面に入ります。





2. [使用する点群を選択]をクリックし、表示される画面でシーンの点群を出力するステップと出力ポートを順番に選択して、[OK]をクリックします。



シーンの点群を出力するステップを選択するとき、その後のステップの実行を高速化するために、通常はステップ **深度画像を点群に変換** から出力された点群をシーンの点群として使用されます。また、ステップ **カメラから画像を取得** から出力された点群の使用は推奨されません。



-  は、この出力ポートがその後のステップの入力ポートにつながっていることを意味します。
-  は、この出力ポートがその後のステップの入力ポートにつながっていないことを意味します。

3. シーンの点群が選択後、実際の状況に応じて基準座標系を選択できます。



シーンの点群が設定後、[ROI設定](#)を参照してROI設定に関連する内容を学ぶことができます。

4.6.5. マッチングモデル・把持位置姿勢エディタ

対象物の点群モデルの作成や編集、把持位置姿勢の追加を容易にするために、Mech-Visionはマッチングモデル・把持位置姿勢エディタのツールを用意します。このツールは、部品のピック&プレースなど、部品認識を必要とするシーンによく使われます。

マッチングモデル・把持位置姿勢エディタはプロジェクトのデータと設定を呼び出す必要があるため、使用する前に対応するプロジェクトを開くか、プロジェクトを新規作成してからツールの「マッチングモデル・把持位置姿勢エディタ」をクリックします。

マッチングモデル・把持位置姿勢エディタで把持位置姿勢を追加するには、次のいずれの方法を使用します。 マッチングモデル・把持位置姿勢エディタを使用する前に、[画面紹介](#)を参照してこのツールの画面と機能について詳しく学習することをお勧めします。


ティーチペンダントでロボットを移動して把持位置姿勢を取得

- 適用シーン：高い把持精度（0.5 mm以下）が求められている場合、ロボットハンドのTCP位置姿勢を正確に測定できない場合。
- 使用流れ：把持位置姿勢を取得して点群モデルを生成し、点群モデルを編集します。
 - [ティーチングによる把持位置姿勢を追加](#)
 - [点群モデルを生成](#)
 - [点群モデルを編集](#)

マッチングモデル・把持位置姿勢エディタで把持位置姿勢を追加

- 適用シーン：高い把持精度が求められない場合。
- 使用流れ：点群モデルを生成して編集し、把持位置姿勢を追加します。
 - [点群モデルを生成](#)
 - [点群モデルを編集](#)
 - [位置姿勢を追加して把持位置姿勢を追加](#)

上記2つの方法に加えて、既存の把持位置姿勢ファイルをインポートすることも可能です。ツール

ルバーの  をクリックし、表示される画面で既存の把持位置姿勢ファイルを選択することで、把持位置姿勢をすばやく追加することができます。

一部のステップでは、マッチングモデル・把持位置姿勢エディタによって作成されたファイルを使用する必要があります。



- 点群モデルファイルをモデルファイルとして使用する必要があるステップは、[3D位置姿勢低精度推定](#)、[3D位置姿勢低精度推定（マルチモデル）](#)、[3D位置姿勢高精度推定](#)、[3D位置姿勢高精度推定（マルチモデル）](#) です。
- 把持位置姿勢ファイルを使用する必要があるステップは、[複数の把持位置姿勢にマッピング](#) です。

4.6.6. ディープラーニングサーバー

本節では、ディープラーニングサーバーの使用方法について説明します。

概要

ディープラーニングサーバーは、Mech-Visionソフトウェアが提供するディープラーニングサービスを管理するためのツールです。ソフトウェアのメインウィンドウのメニューバーから **ディープラーニング** > **ディープラーニングサーバー** を選択することで開くことができます。

通常、ユーザーはディープラーニングサーバーの画面で何もする必要はありません。Mech-Visionプロジェクトを開くと、ディープラーニングサーバーがプロジェクト内のディープラーニングステップのパラメータ **サーバーIP** と **サーバーポート (1-65535)** の設定に従って自動的に起動されます。

互換性に関する注意事項

ディープラーニングステップでディープラーニングサーバーを有効にする必要があるかどうかについては、[互換性に関する注意事項](#) をご参照ください。

よく使われる操作

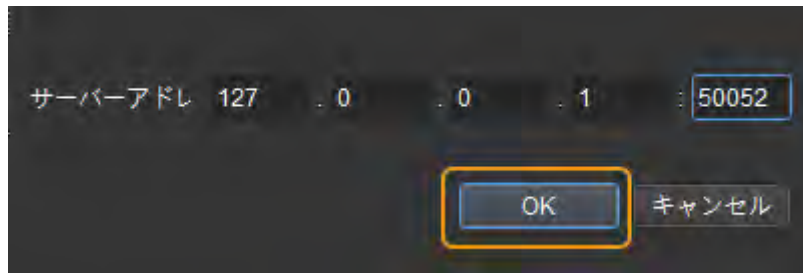
以下では、ディープラーニングサーバーでよく使われる操作について説明します。

新しいサーバーを起動

1. ディープラーニングサーバー画面で **[新しいサーバーを起動]** をクリックします。



- 表示される画面でサーバーアドレスとポート番号を入力して[OK]をクリックします。すると、新しいサーバーが起動されます。



選択したサーバーをシャットダウン

- 複数のディープラーニングサーバーが起動している場合、終了するディープラーニングサーバーを選択し、[選択したサーバーをシャットダウン]をクリックします。



- 表示される画面で[はい (Y)]をクリックします。すると、選択したディープラーニングサーバーが終了されます。



サーバーを再起動

ディープラーニングサーバーを再起動するには、まずディープラーニングサーバーを終了してから、同じサーバーアドレスとポート番号でディープラーニングサーバーを起動する必要があります。

4.6.7. ディープラーニングモデルパッケージ管理ツール

本節では、ディープラーニングモデルパッケージ管理ツールの使用方法と注意事項について説明します。

概要

ディープラーニングモデルパッケージ管理ツールは、Mech-Visionが提供するツールで、ディ-

プラーニングモデルパッケージの管理、Mech-DLK 2.2.0以降バージョンによってエクスポートされたディープラーニングモデルパッケージ（単体モデルまたは直列モデル）の最適化、実行モード、ハードウェアタイプ、モデル推論の効率、モデルパッケージの状態管理などを行います。また、このツールでは、IPCのGPU使用率監視にも対応可能です。

プロジェクトにディープラーニング関連のステップを使用する場合、まずディープラーニングモデルパッケージをディープラーニングモデルパッケージ管理ツールにインポートしてから、ディープラーニング関連のステップで 사용할 수 있습니다. 디ープラー닝 모델 패키지把这个工具导入到工具中，事先进行优化是可能的。



Mech-DLK 2.4.1以降、モデルパッケージには、単体モデルと直列モデルの2種類があります。

- 単体モデルパッケージ：1つのディープラーニングアルゴリズムモジュールのみを搭載したモデルパッケージのことを指します（例：「インスタンスセグメンテーション」モデル）。
- 直列モデルパッケージ：ディープラーニングアルゴリズムモジュールの複数のモデルを、前のモデルの出力が次のモデルの入力となる直列形式で搭載したモデルパッケージのことを指します。例えば、モデルパッケージに「対象物検出」と「インスタンスセグメンテーション」の2つのモデルがある場合、モデルの推論順序は**対象物検出**、**インスタンスセグメンテーション**となり、「対象物検出」の出力は「インスタンスセグメンテーション」の出力になります。「対象物検出」モデルの出力は、「インスタンスセグメンテーション」モデルの入力として使用されます。

画面

このツールを開くには、次のいずれの方法を使用します。

- メニューバーから**ディープラーニング**、**ディープラーニングモデルパッケージ管理ツール**をクリックします。
- プロジェクトで「ディープラーニングモデルパッケージを推論」ステップをクリックして選択し、ステップパラメータで[**モデルパッケージ管理ツールを開く**]をクリックします。

以下の表は、ディープラーニングモデルパッケージ管理ツールのオプションを説明するものです。

オプション	説明
キャッシュモデルパッケージ名	インポートされたディープラーニングモデルパッケージの名前を確認できます
モデルを使用するプロジェクト	ディープラーニングモデルパッケージを使用したMech-Visionプロジェクトを確認できます

オプション	説明
モデルパッケージのタイプ	「対象物検出（単体モデル）」や「対象物検出+欠陥セグメンテーション（直列モデル）」など、ディープラーニングモデルパッケージのタイプを確認できます
実行モード	モデルパッケージ推論の実行モード（「共有モード」または「パフォーマンスモード」）を選択できます
ハードウェアタイプ	GPU（デフォルト）、GPU（最適化）、CPUなど、モデルパッケージ推論に使用するハードウェアのタイプを確認できます
モデル推論の効率	モデル推論の効率を設定できます
モデルパッケージの状態	「最適化中」、「使用可能」や「最適化失敗」など、モデルパッケージの状態を確認できます

実行モード

- 共有モード：このオプションを選択すると、複数のステップが同じディープラーニングモデルパッケージを使用し、前の推論が完了してから次の推論を行い、より多くの実行時リソースを節約することができます。
- パフォーマンスモード：このオプションを選択すると、複数のステップが同じディープラーニングモデルパッケージを使用し、並行して推論します。実行速度を上げることができますが、より多くの実行時リソースを消費します。



ハードウェアタイプ

- CPU：ディープラーニングモデルパッケージの推論にCPUを使用すると、GPUと比較して推論時間が長くなり、認識精度が低下します。
- GPU（デフォルト）：このオプションを選択すると、ハードウェアタイプに応じてディープラーニングモデルパッケージを最適化する必要はなく、ディープラーニングモデルパッケージの推論が高速化されることはありません。
- GPU（最適化）：このオプションを選択すると、モデルパッケージはハードウェアタイプに応じて最適化されます。最適化は1回だけ実行する必要があり、5〜15分かかります。最適化後、モデルパッケージの推論時間が短縮されます。

使用ガイド

以下では、ディープラーニングモデルパッケージ管理ツールでよく使われる操作を説明します。

ディープラーニングモデルパッケージをインポート

1.

ディープラーニングモデルパッケージ管理ツールを開き、画面の左上隅にある[**インポート**]をクリックします。

- 表示されるファイル選択画面からインポートするディープラーニングモデルパッケージを選択して[**開く**]をクリックすると、選択したモデルパッケージがディープラーニングモデルパッケージ管理ツールのリストに表示されます。これで、ディープラーニングモデルパッケージのインポートは完了しました。



ディープラーニングモデルパッケージをインポートする場合、GPUドライバーの最低必要バージョンは472.50（バージョン500以上のGPUドライバーは推奨されておらず、ディープラーニングステップの実行時間が変動する可能性があります）、最低必要CPUはインテル第6世代Coreとなります。ハードウェアの条件を満たさない場合、ディープラーニングモデルパッケージのインポートに失敗します。

ディープラーニングのステップでディープラーニングモデルパッケージを選択

ディープラーニングモデルパッケージをインポートした後、「ディープラーニングモデルパッケージを推論」ステップでディープラーニングモデルパッケージを選択するには、そのパラメータの**モデルパッケージ名**のドロップダウンリストから、モデルパッケージ推論のためにインポートしたモデルパッケージを選択することができます。



ディープラーニングモデルパッケージを削除

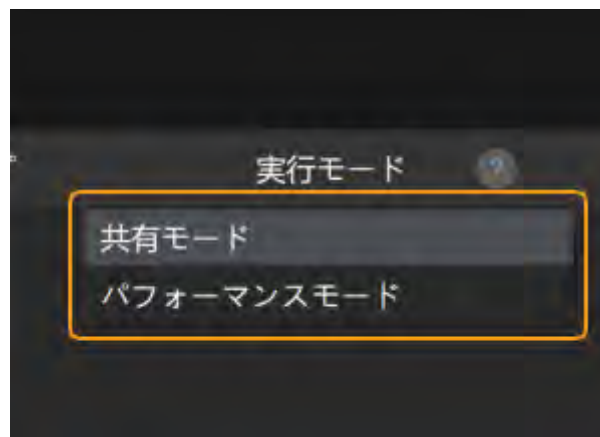
インポートされたディープラーニングモデルパッケージを削除するには、そのパッケージを選択し、右上にある[**削除**]ボタンをクリックします。



ディープラーニングモデルパッケージは、最適化中または使用中（このモデルパッケージを使用するプロジェクトが実行中）の場合、削除することはできません。

ディープラーニングモデルパッケージ推論の実行モードの切替

ディープラーニングモデルパッケージ推論の実行モードを切り替える必要がある場合、**実行モード**で▼をクリックし、**共有モード**または**パフォーマンスモード**を選択します。



- ディープラーニングモデルパッケージが最適化中または使用中（このモデルパッケージを使用するプロジェクトが実行中）の場合、実行モードを切り替えることはできません。
- ディープラーニングモデルパッケージが共有モードである場合は、ディープラーニングモデルパッケージを推論のステップパラメータでGPU IDを切り替えることはできません。

ディープラーニングモデルパッケージ推論のハードウェアタイプの切替

ディープラーニングモデルパッケージ推論のハードウェアタイプを、GPU（デフォルト）、GPU（最適化）、CPUに切り替えることができます。

ディープラーニングモデルパッケージ管理ツールのハードウェアタイプの▼をクリック

し、**GPU（デフォルト）**、**GPU（最適化）** または **CPU** を選択します。



- Mech-DLK 2.4.1によってエクスポートされたモデルパッケージのみ、CPU、GPUの切り替えに対応しています。
- **GPU（デフォルト）** ハードウェアタイプは、次のディープラーニングモデルパッケージをインポートした後は使用できません。
 - Mech-DLK 2.2.0によってエクスポートされたインスタンスセグメンテーションとインスタンスセグメンテーションのスーパーモデルパッケージ。
 - Mech-DLKによってエクスポートされた.dlkmファイルのみを含むディープラーニングモデルパッケージ。
- ディープラーニングモデルパッケージの最適化状態が**最適化中**または**使用中**である場合は、**ハードウェアタイプ**を切り替えることはできません。「使用中」は、このディープラーニングモデルパッケージを使用しているプロジェクトが実行中であることを意味します。

モデル推論の効率設定

Mech-DLK 2.4.1以降バージョンによってエクスポートされたモデルを使用する場合、モデル推論の効率を以下のように設定することができます。

1. ディープラーニングモデルパッケージをインポートします。
2. 「モデル推論の効率」の下にある[設定]ボタンをクリックし、「モデル推論の効率」の設定画面が表示されます。この画面で、「バッチサイズ」と「精度」を設定可能です。

モデルの実行効率は、「バッチサイズ」と「精度」パラメータによって異なります。

- バッチサイズ：モデル推論時にニューラルネットワークに一度に投入する画像の数で、1~128の範囲で設定します。この値を大きくするとモデル推論が高速化しますが、より多くのメモリを消費することになります。この値が正しく設定されていない場合、推論が遅くなります。インスタンスセグメンテーションモデルでは、「バッチサイズ」の設定はサポートされておらず、1でなければなりません。
- 精度（「ハードウェアタイプ」が「GPU最適化」の場合のみ有効）：
 - FP32：モデルの精度が高いですが、推論速度が遅いです。
 - FP16：モデルの精度が低いですが、推論速度が速いです。



「バッチサイズ」の値は、実際にニューラルネットワークに投入される画像数と

一致させることが推奨されます。

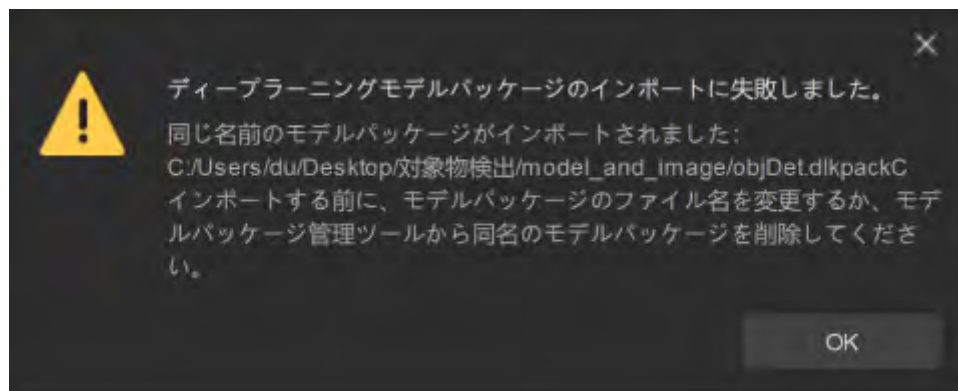
設定された「バッチサイズ」が、実際にニューラルネットワークに投入される画像数よりもはるかに大きい場合、一部のリソースが浪費され、推論が遅くなることがあります。

例えば、画像数が26枚で「バッチサイズ」が20に設定されている場合、推論は2回で行われ、1回目は20枚の画像をニューラルネットワークに投入し、2回目は6枚の画像をニューラルネットワークに投入します。2回目の推論では、設定された「バッチサイズ」が実際にニューラルネットワークに送信される画像数よりもはるかに大きいため、一部のリソースが浪費され、推論速度が遅くなる原因になります。したがって、リソースを合理的に使用するために、「バッチサイズ」を適切に設定する必要があります。

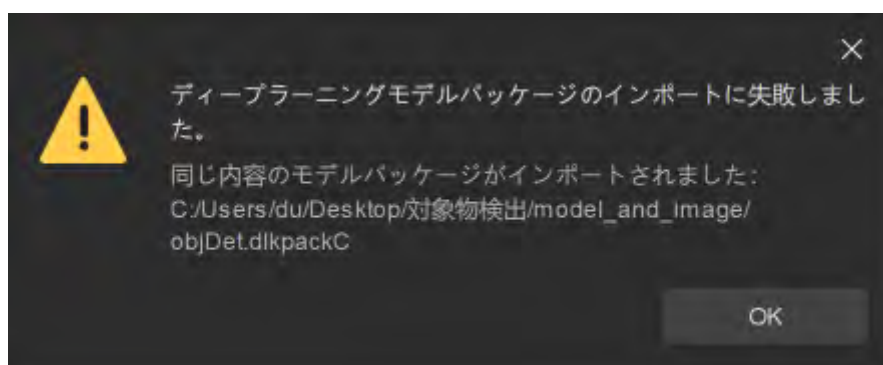
注意事項

ディープラーニングモデルパッケージのインポートに失敗

- ディープラーニングモデルパッケージをインポートした後、同じ名前のディープラーニングモデルパッケージをインポートする場合、2つのモデルの名前が異なるように**ディープラーニングモデルパッケージの名前を変更するか、同じ名前のモデルパッケージを削除して、再度インポートする必要があります。**



- 同じ内容のディープラーニングモデルパッケージを繰り返しインポートした場合、**ディープラーニングモデルパッケージのインポートに失敗しました**のメッセージが表示されます。

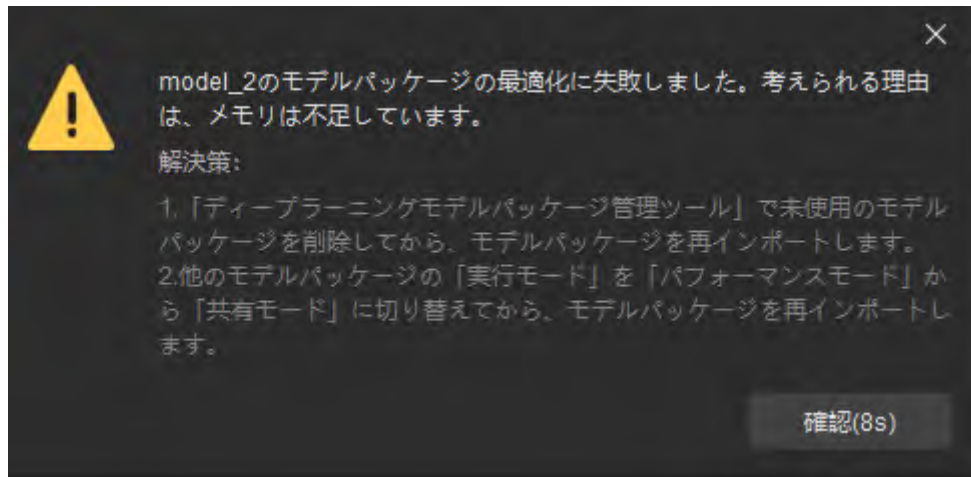


- ソフトウェア・ハードウェアの条件を満たさない場合、ディープラーニングモデルパッケージのインポートに失敗します。GPUドライバーの最低バージョンの要件は472.50、CPUの最

低要件はインテル第6世代Coreです。

ディープラーニングモデルパッケージの最適化に失敗

下図に示すように、メモリが不足している場合、ディープラーニングモデルパッケージの最適化に失敗することがあります。この問題は、プロンプトウィンドウの解決策に従うことで解決できます。



互換性に関する注意事項

- Mech-Vision 1.7.1バージョンでは、Mech-DLK 2.4.1バージョンによってエクスポートされたディープラーニングモデルパッケージも使用できますが、いくつかの互換性問題もあります。Mech-Vision 1.7.2バージョン以降とMech-DLK 2.4.1バージョン以降によってエクスポートされたディープラーニングモデルパッケージとの併用が推奨されています。
 - Mech-Visionは直列モデルを使用できません。
 - モデル推論の効率を設定できません。
 - 画像分類の効果が悪くなる可能性があります。
 - モデルパッケージをCPUデバイスでは使用できません。
- Mech-Vision 1.7.1バージョンで最適化されたモデルパッケージをMech-Vision 1.7.2バージョンに使用する場合、モデルパッケージが「ディープラーニングモデルパッケージを推論」ステップで初めて実行するときに速度が遅くなります。

4.6.8. パラメータレシピ

本節では、**パラメータレシピ**やその応用シーン、設定方法について説明していきます。

概要

パラメータレシピとは、様々な状況でプロジェクトを実行するための調整が必要なパラメータ設定の集りです。実際の状況に応じてパラメータレシピを設定し、複数のプロジェクトを構築せずに異なるパラメータレシピを切り替えることで様々な応用シーンに対応できます。それに

より、プロジェクトの適応性を高め、生産効率を向上させます。

Mech-Visionでは、パラメータレシピの機能をサポートする**パラメータレシピ・エディタ**のツールが組み込まれています。

応用シーン

パラメータレシピは、状況に応じて異なるパラメータ設定を必要とするシーンに使用されます。代表的なシーンは次の通りです。


- 同じMech-Visionプロジェクトで異なる種類のワークまたは部品を処理するシーン。このシーンでは、異なるパラメータレシピを設定することにより、ワークまたは対象物のパラメータ設定を動的に切り替えることができます。
- 同じMech-Visionプロジェクトで異なる箱内のワークを処理するシーン。このシーンでは、異なるパラメータレシピを設定することにより、箱に応じて異なるパラメータレシピを動的に切り替えることができます。
- Mech-Visionプロジェクトが照明条件（昼と夜など）に応じて異なるカメラ構成パラメータグループとその他のパラメータ設定を必要とするシーン。このシーンでは、異なるパラメータレシピを設定することにより、照明条件に応じてカメラ構成パラメータグループとその他のパラメータ設定を動的に切り替えることができます。
- Eye In Hand取り付け方式でMech-Visionプロジェクトが異なる場所に撮影する必要があるシーン。このシーンでは、異なるパラメータレシピを設定することにより、異なる撮影場所に応じてカメラ構成パラメータグループとその他のパラメータ設定を動的に切り替えることができます。

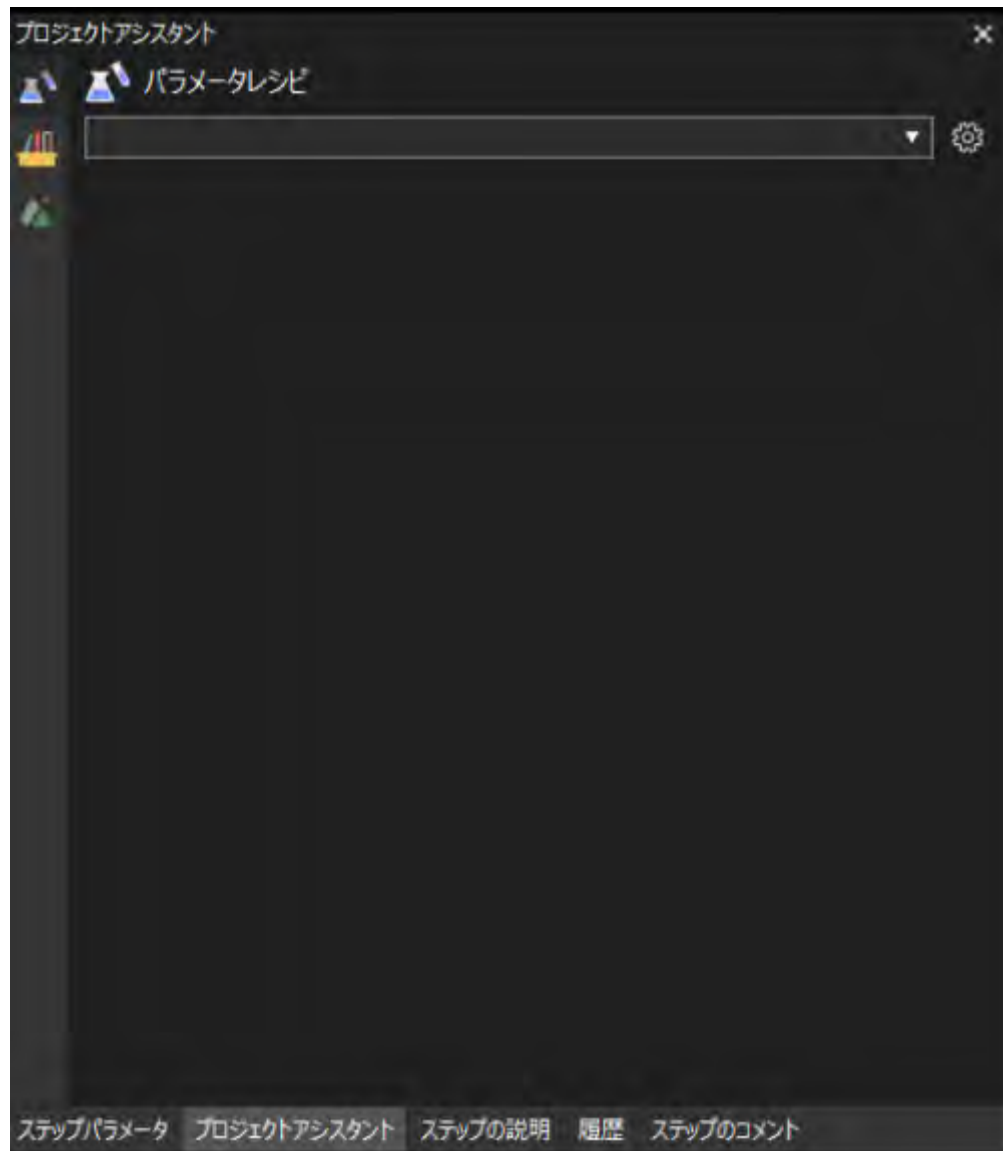
Vizティーチング通信の場合、手動でMech-Visionプロジェクトを対応するパラメータレシピに切り替える必要があります。また、標準インターフェースやAdapterを使用して通信する場合、ロボット側のプログラムから送信した切り替えコマンドが必要で、Mech-Visionは受信したコマンドに応じてパラメータレシピを自動で切り替えます。


パラメータレシピを構成

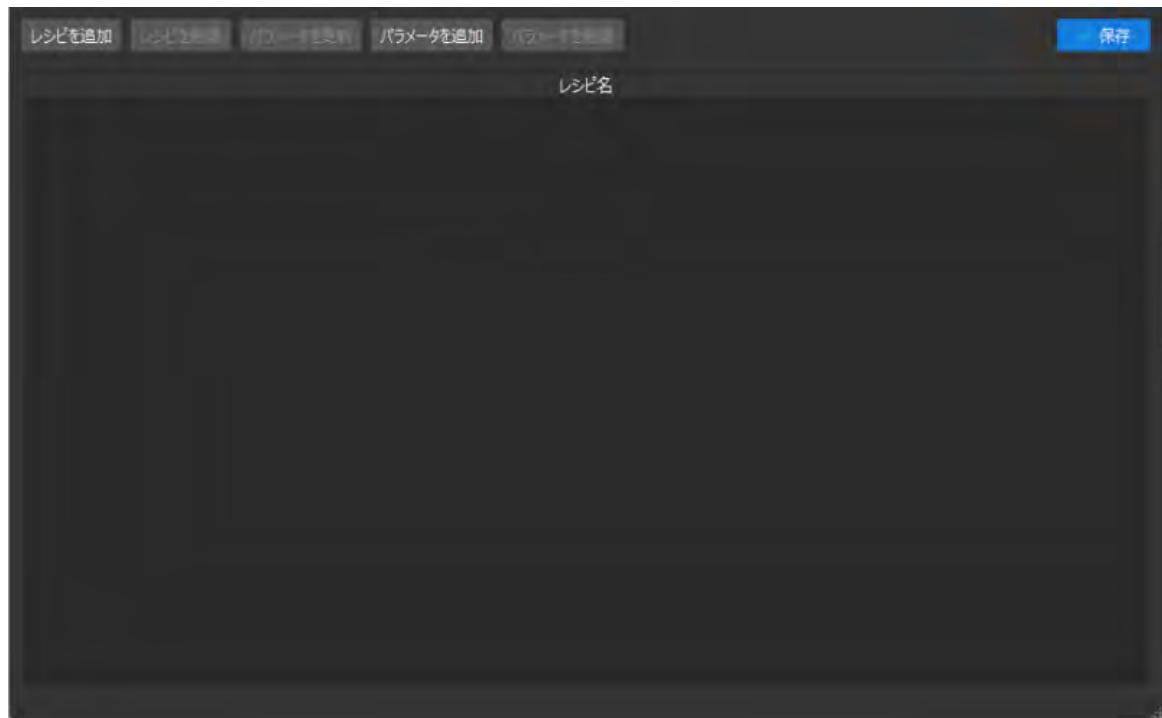
パラメータレシピを追加

パラメータレシピを追加するには、以下の手順を実行します。

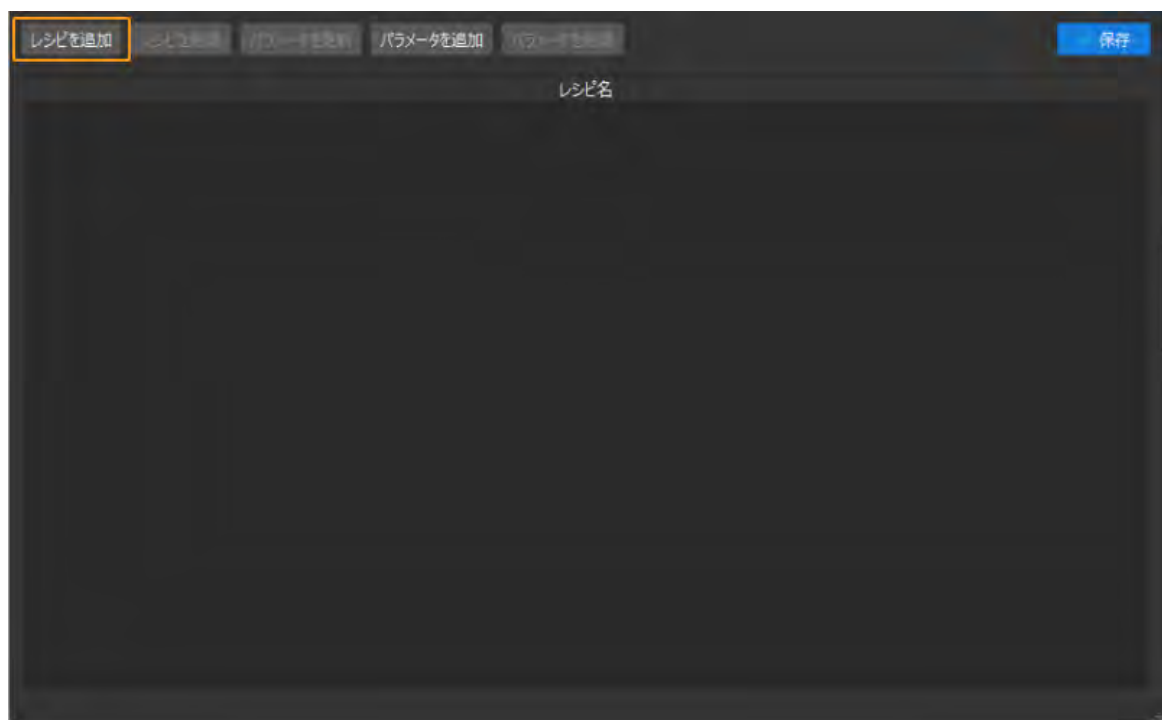
1. プロジェクトを開き、プロジェクト設定エリアで**プロジェクトアシスタント**タブを選択し、をクリックしてパラメータレシピの画面に入ります。



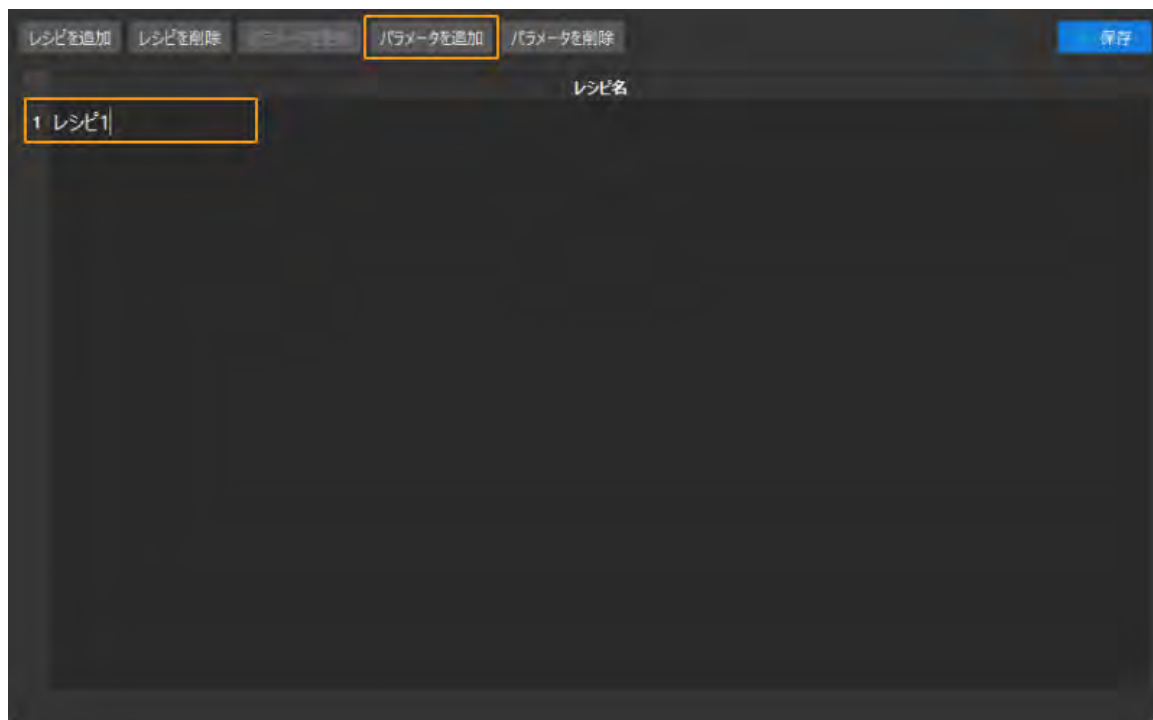
2. パラメータレシピの画面で  をクリックします。
3. すると、**パラメータレシピ・エディタ**画面が表示されます。この画面で[**レシピを追加**]をクリックします。



4. レシピ一覧で、追加されたレシピを選択し、ダブルクリックしてレシピ名を入力します。



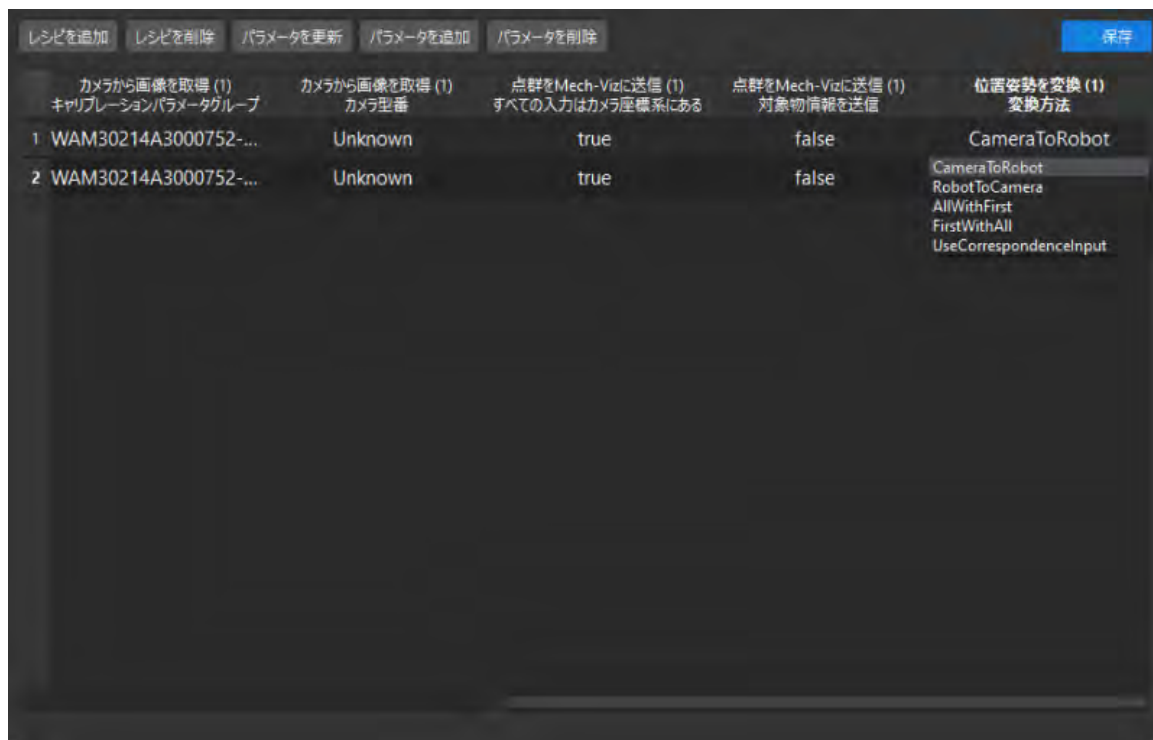
5. この画面の空白場所をクリックし、追加されたレシピを選択して[パラメータを追加]をクリックします。すると、**ステップパラメータを選択**画面が表示されます。



6. ステップパラメータを選択画面の左側から追加するステップを選択し、右側から追加するパラメータにチェックを入れます。



7. 必要なすべてのステップパラメータが選択されるまで上記の手順を繰り返し、[OK]をクリックします。
8. レシピ一覧で、レシピに追加されたパラメータを順番にダブルクリックして変更します。



パラメータの設定順序による特定のパラメータが正常に設定できないことを避けるため、レシピに設定されたパラメータ順序に注意してください。

- パラメータレシピ・エディタ画面で[保存]をクリックします。保存が成功すると、画面の左下隅に「保存に成功しました。」というメッセージが表示されます。



複数のパラメータレシピが必要な場合、上記の手順を繰り返して追加します。新しく追加されたパラメータレシピは、現在のプロジェクトのステップパラメータ設定を自動的に使用しま

す。

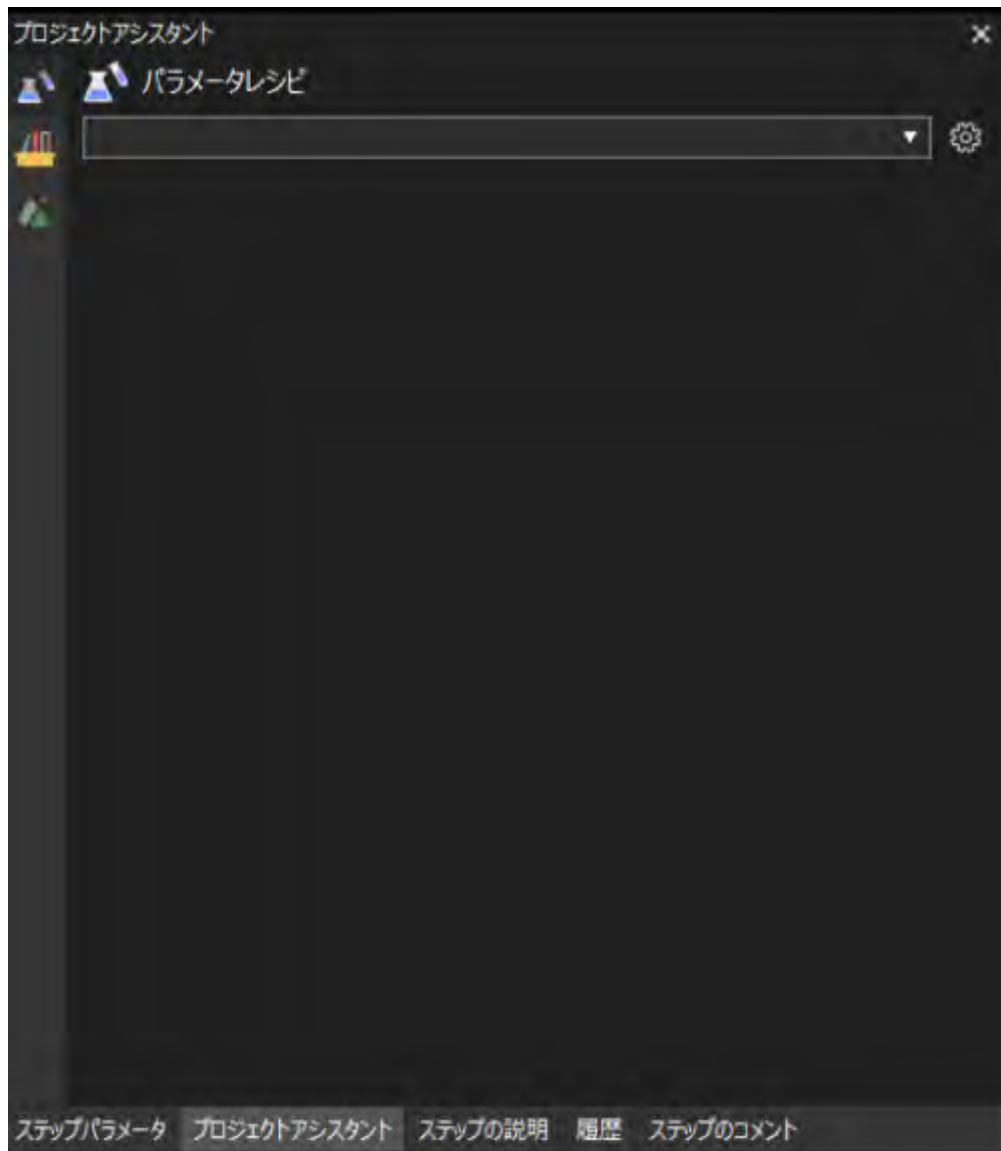
パラメータレシピの使用/切り替え



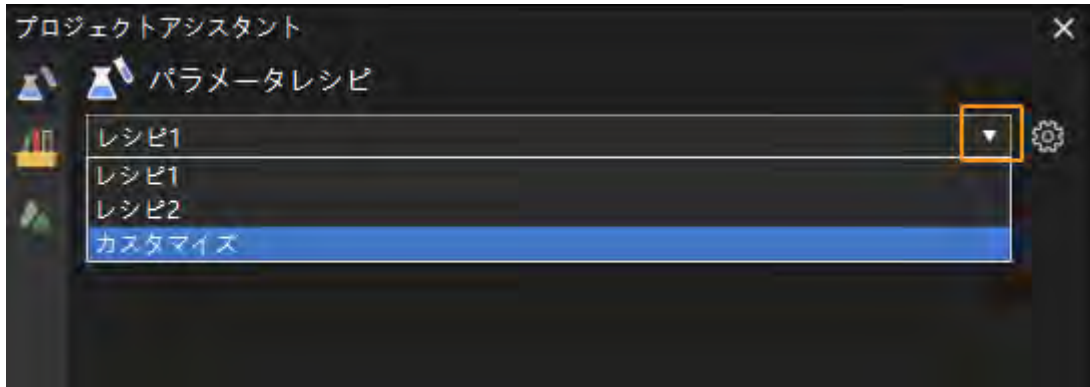
標準インターフェースやAdapterを使用して通信する場合、パラメータレシピを自動的に切り替えるためにロボット側のプログラムから切り替えコマンドを送信する必要があります。

パラメータレシピを手動で使用する/切り替えるには、以下の手順を実行します。

1. プロジェクト設定エリアで**プロジェクトアシスタント**タブを選択し、をクリックしてパラメータレシピの画面に入ります。



2. 次に、ドロップダウンリストからワークに対応するパラメータレシピを選択します。プロジェクトを実行すると、設定されたパラメータレシピが有効になります。




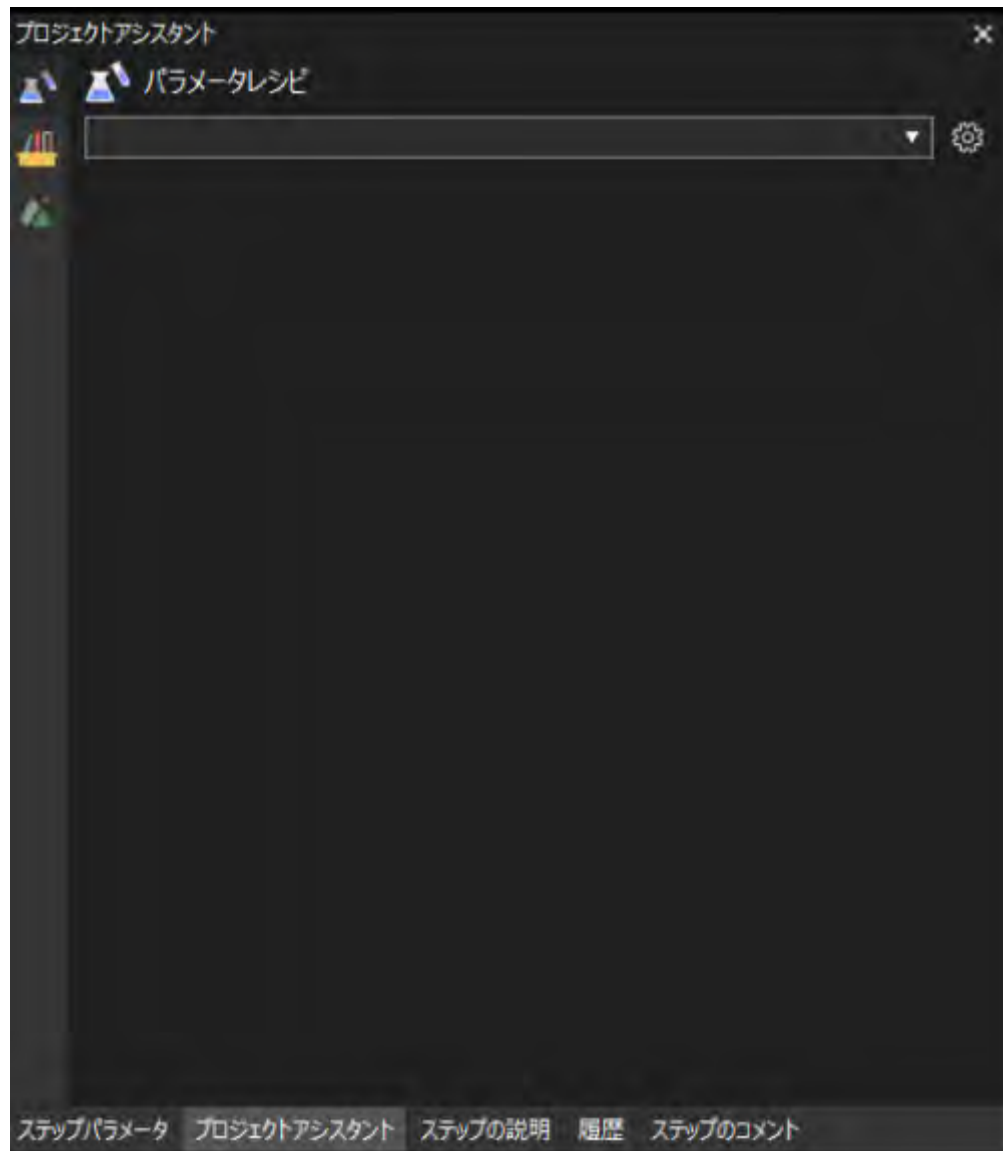
パラメータレシピを使用してプロジェクトを実行した後、レシピのパラメータ設定を変更したい場合、[パラメータレシピに指定したステップパラメータ設定を変更](#)を参照して変更してください。プロジェクトでパラメータを直接変更し、プロジェクトを再度実行すれば有効化されたレシピが自動的に「カスタマイズ」レシピに切り替えられます。


パラメータレシピの番号を確認

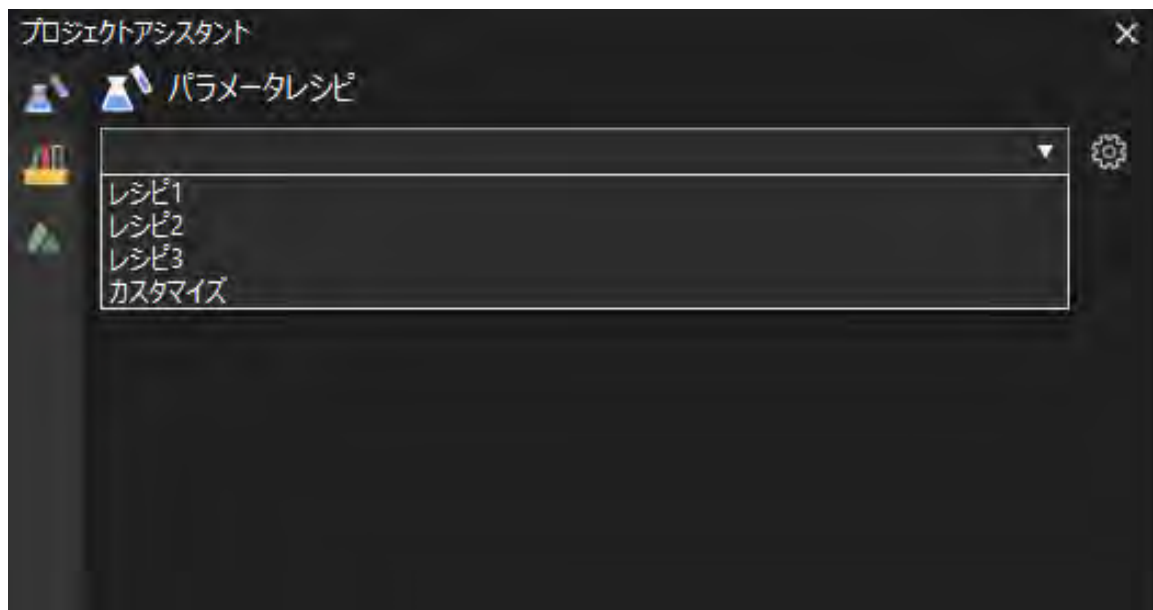
標準インターフェースやAdapterを使用して通信する場合、ロボット側のプログラムから送信した切り替えコマンドにパラメータレシの名前ではなく、パラメータレシピ番号を指定する必要があります。

パラメータレシピの番号を確認するには、以下の手順を実行します。

1. プロジェクトを開き、プロジェクト設定エリアで**プロジェクトアシスタント**タブを選択し、をクリックしてパラメータレシピの画面に入ります。



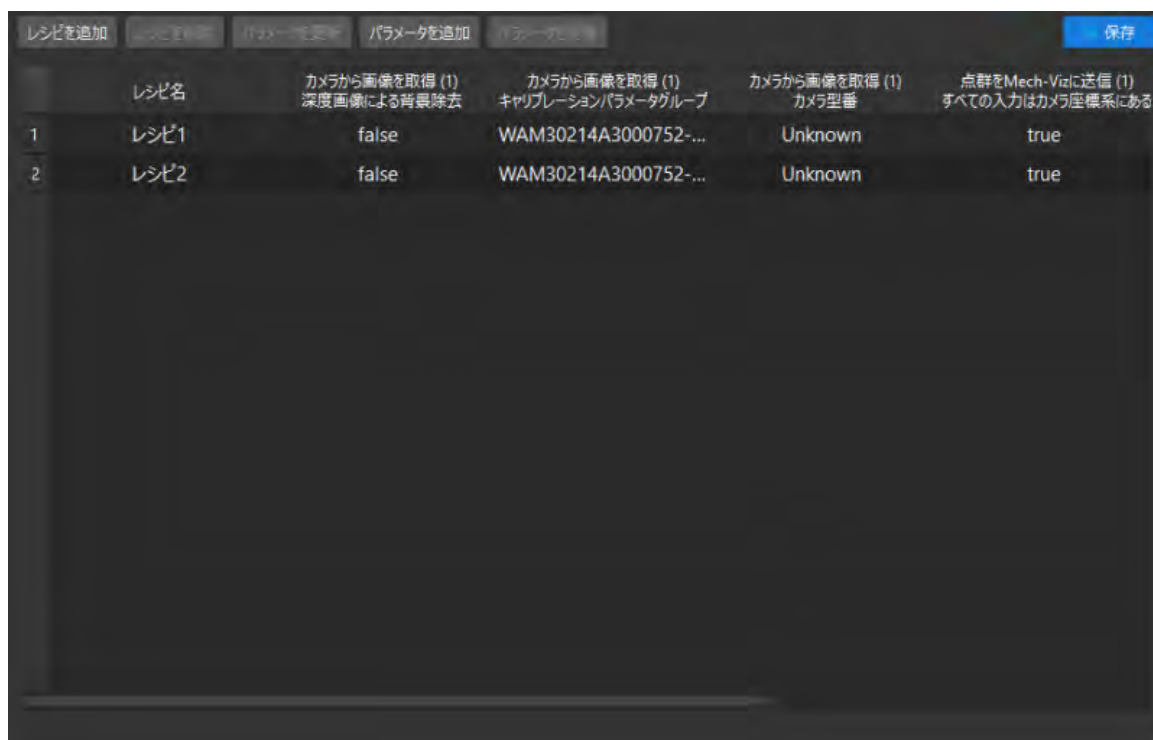
2. パラメータレシピの画面で  をクリックします。
3. レシピ一覧の左端からレシピ番号を確認することができます。



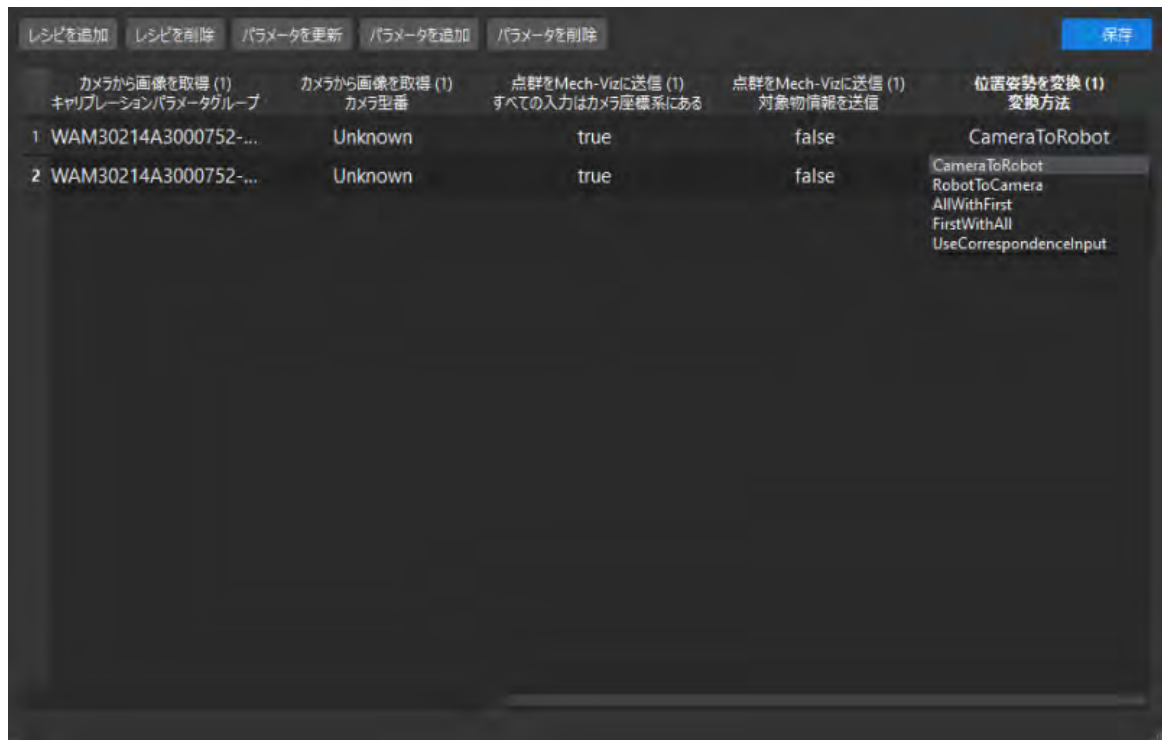
パラメータレシピにステップパラメータを追加

パラメータレシピにステップパラメータを追加するには、以下の手順を実行します。

1. パラメータレシピ・エディタ画面でレシピを選択して[パラメータを追加]をクリックします。



2. すると、ステップパラメータを選択画面が表示されます。この画面でステップパラメータを追加してから、[OK]をクリックします。
3. レシピ一覧で、追加されたパラメータをダブルクリックして変更します。



4. パラメータレシピ・エディタ画面で[保存]をクリックします。

パラメータレシピからステップパラメータを削除

パラメータレシピからステップパラメータを削除するには、以下の手順を実行します。

1. パラメータレシピ・エディタ画面でパラメータレシピの項目から削除するステップパラメータを選択して[パラメータを削除]をクリックします。

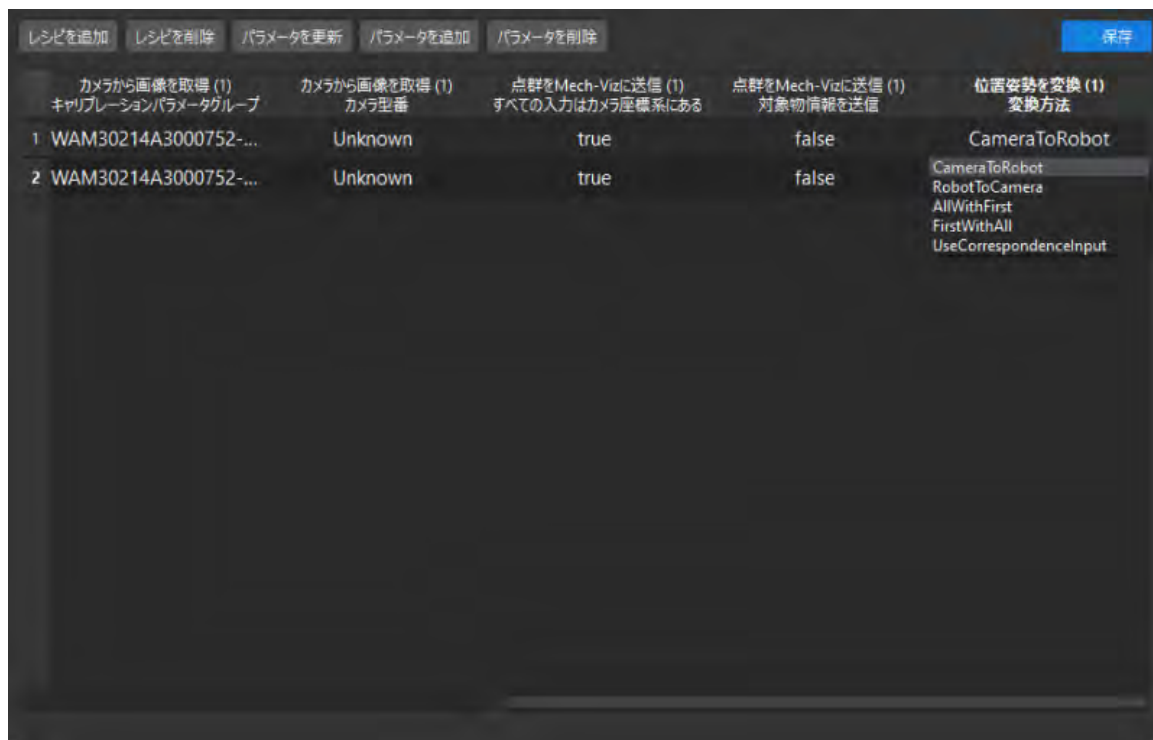


2. [保存]をクリックします。

パラメータレシピに指定したステップパラメータ設定を変更

パラメータレシピに指定したステップパラメータ設定を変更するには、以下の手順を実行します。

1. レシピ一覧で、変更するパラメータ設定をダブルクリックして変更します。



2. [保存]をクリックします。

プロジェクトの設定に応じてパラメータレシピのステップパラメータ設定を更新

パラメータレシピ・エディタは、現在のプロジェクト設定に従って、レシピの特定またはすべてのステップパラメータの設定を更新することをサポートします。

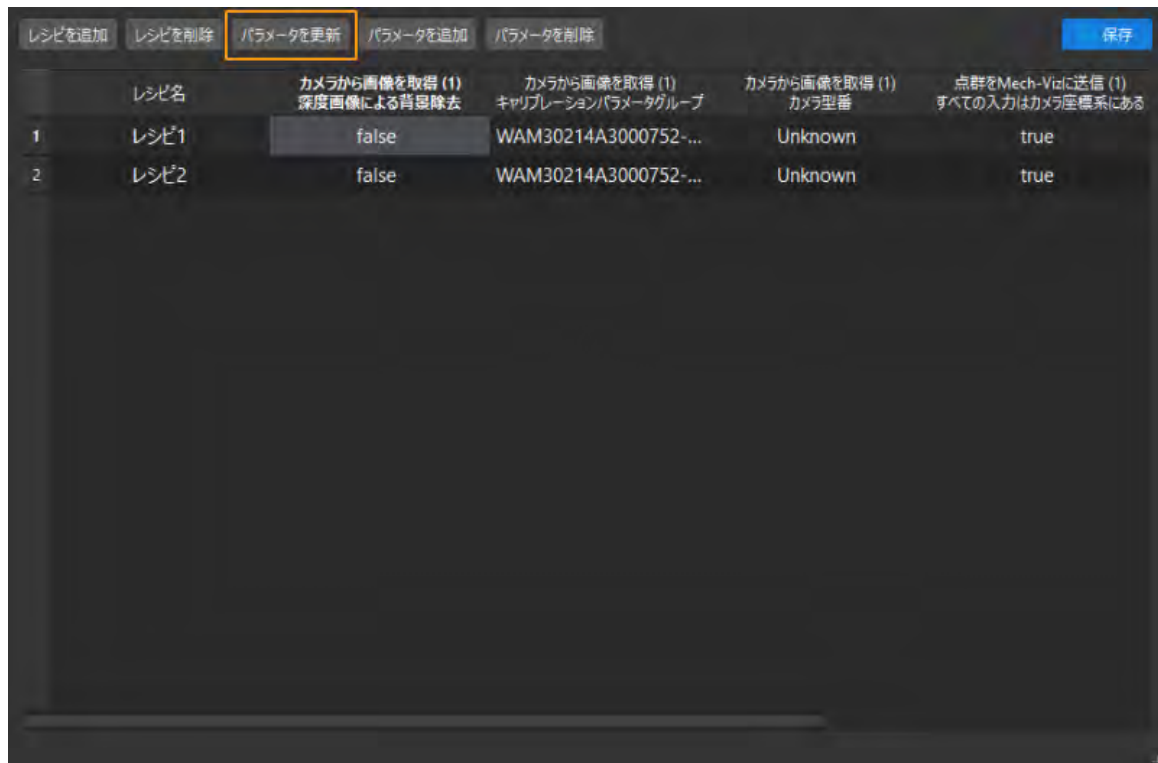
プロジェクトの設定に応じてパラメータレシピに指定したステップパラメータ設定を変更するには、以下の手順を実行します。

1. **パラメータレシピ・エディタ**画面でパラメータレシピの項目から更新するステップパラメータを選択します。



Ctrl キーを押したまま、複数のステップパラメータを選択することができます。

2. ツールバーの**パラメータを更新**をクリックします。



プロジェクトの設定に応じてパラメータレシピにすべてのステップパラメータ設定を更新するには、以下の手順を実行します。

1. **パラメータレシピ・エディタ**画面で、左端にあるレシピ番号をクリックして現在の項目（行全体）を選択することができます。
2. ツールバーの**パラメータを更新**をクリックします。

パラメータレシピのステップパラメータ設定を同期

パラメータレシピ・エディタは、あるパラメータレシピの指定されたパラメータまたはすべてのパラメータ設定を別のパラメータレシピに同期することをサポートします。

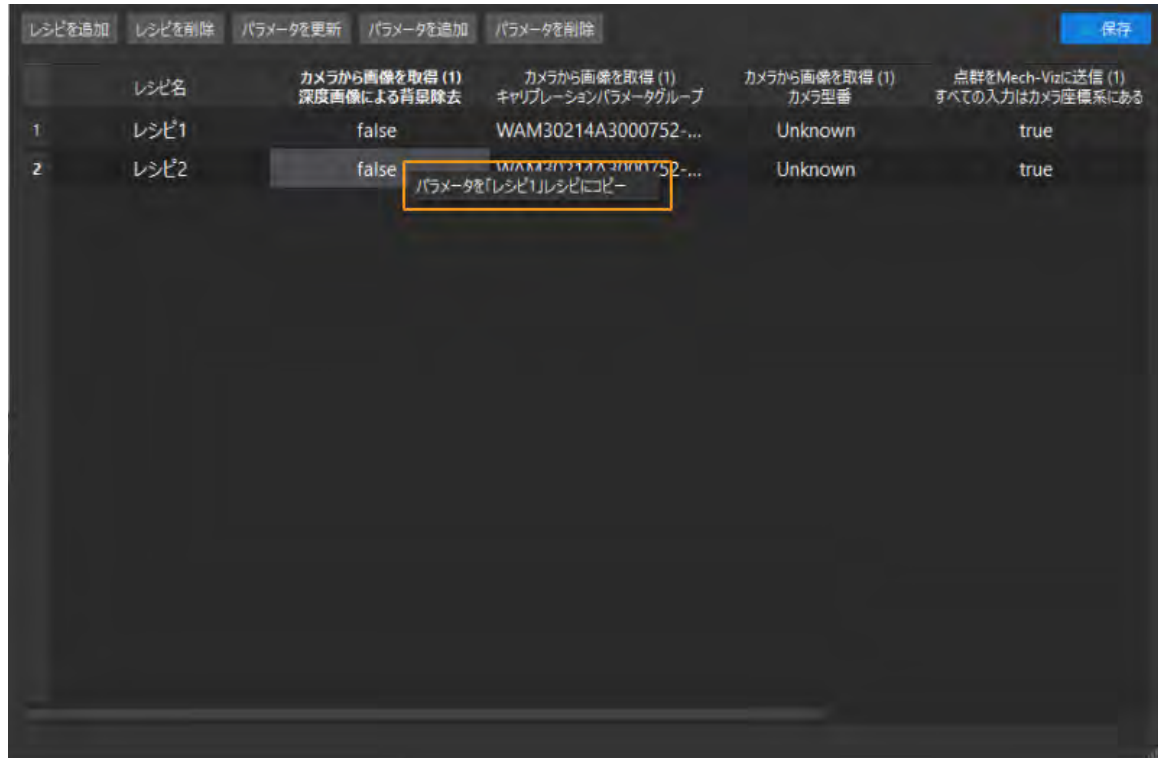
あるパラメータレシピの指定されたパラメータ設定を別のパラメータレシピに同期するには、以下の手順を実行します。

1. **パラメータレシピ・エディタ**画面でパラメータレシピの項目から同期するステップパラメータを選択します。



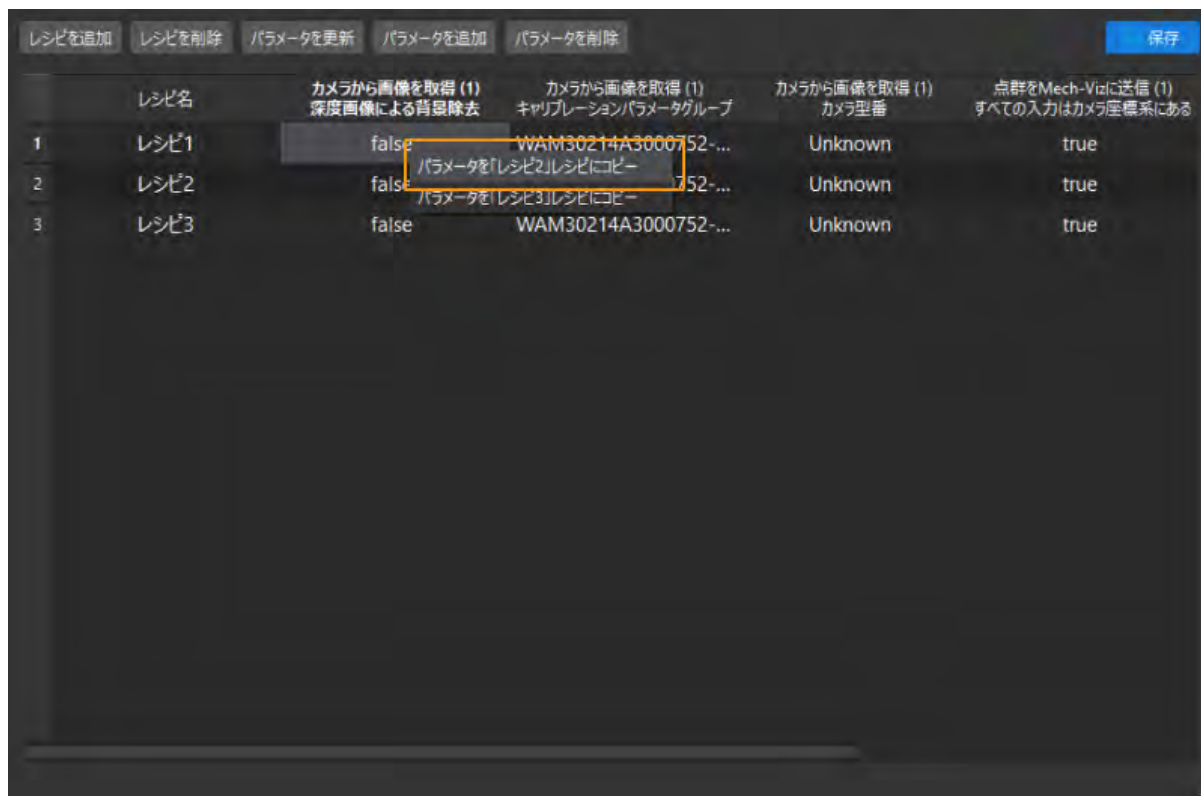
Ctrl キーを押したまま、複数のステップパラメータを選択することができます。

2. ステップパラメータを右クリックし、**パラメータを「○○」レシピにコピー**をクリックします。



あるパラメータレシピのすべてのパラメータ設定を別のパラメータレシピに同期するには、以下の手順を実行します。


1. **パラメータレシピ・エディタ**画面で、左端にあるレシピ番号をクリックして現在の項目（行全体）を選択することができます。
2. レシピ名を右クリックし、**パラメータを「○○」レシピにコピー**をクリックします。

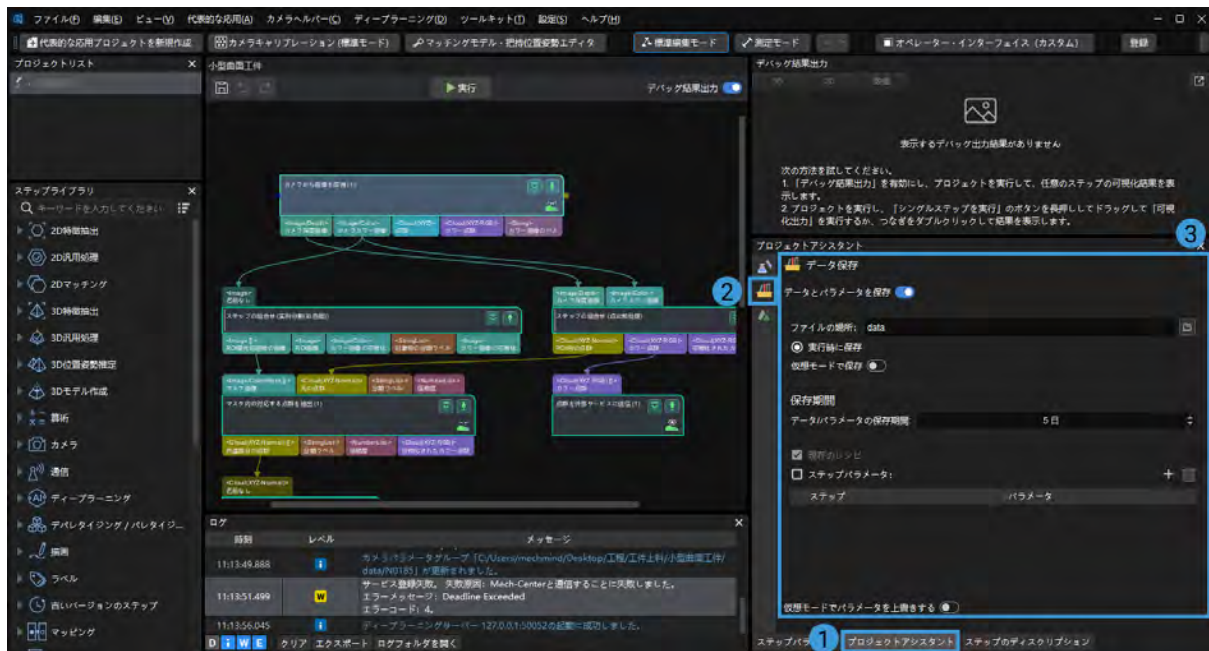


4.6.9. データ保存

機能

現場では、問題発生時の状況を再現し、十分な回帰検証を行うために、Mech-Visionがデータ保存機能を提供しています。

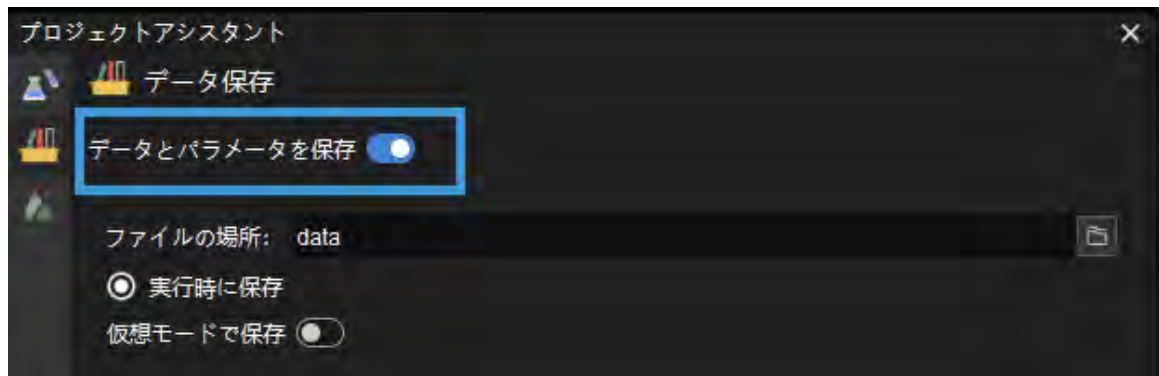
プロジェクトを開いた後、Mech-Visionのプロジェクト設定エリアの下部に[プロジェクトアシスタント]をクリックしてをクリックし、データ保存の画面に入ります。その画面を下図に示します。



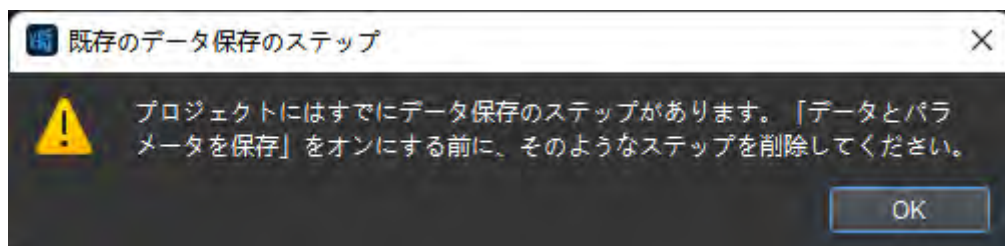
Mech-Visionのプロジェクト設定エリアの下部にプロジェクトアシスタントが表示されない場合は、Mech-Visionのメニューバーのビュータブでプロジェクトアシスタントにチェックが入っているかどうかを確認してください。

使用ガイド

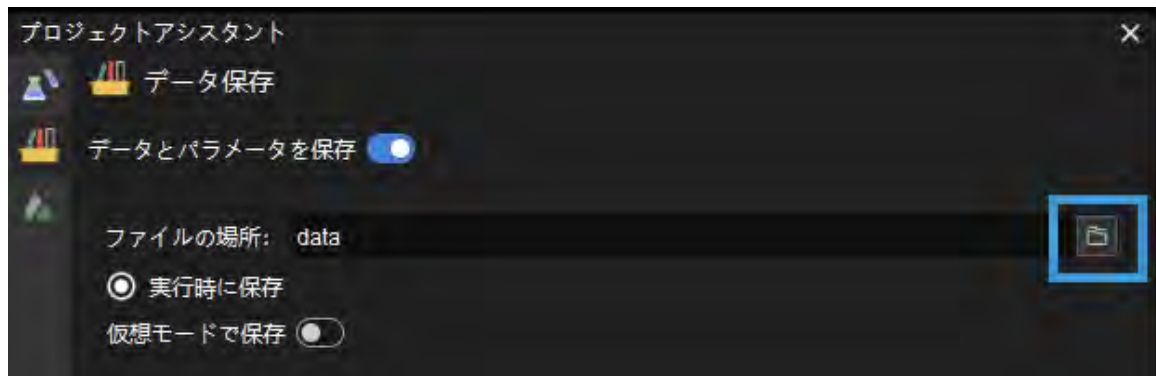
1. [データとパラメータを保存]を有効にすると、プロジェクトの実行時に実行中のデータが自動的に指定されたパスに保存されます。デフォルトの保存パスは、プロジェクトフォルダの中のdataフォルダです。



[データとパラメータを保存]を有効にすると、プロジェクトに「画像のデータとパラメータを保存」のステップ組合せがあるかどうかを検出されます。そのステップ組合せがある場合は、削除の確認メッセージが表示されます。



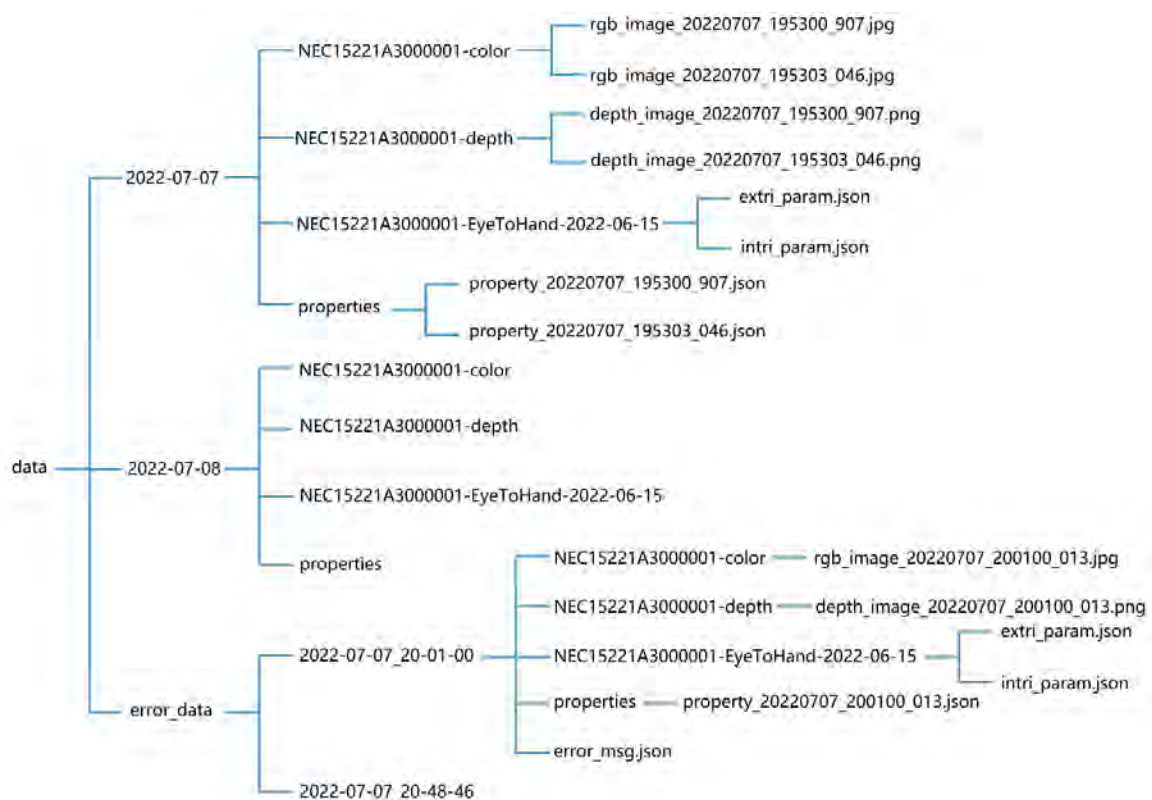
データの保存パスを調整するには、 をクリックして別のパスを選択します。




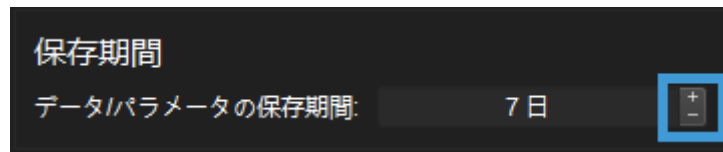
プロジェクトの実行中にEレベルのエラーが発生すると、異常なデータはdateフォルダのerror_dataフォルダに保存されます。



例としてプロジェクトを取り上げます（深度画像とカラー画像データは、「カメラから画像を取得」ステップから取得されています）。データのファイル名と保存パス構造は、下図に示されています。画像ファイルの拡張子はタイムスタンプです。これは、プロジェクトが実行されている時間です。



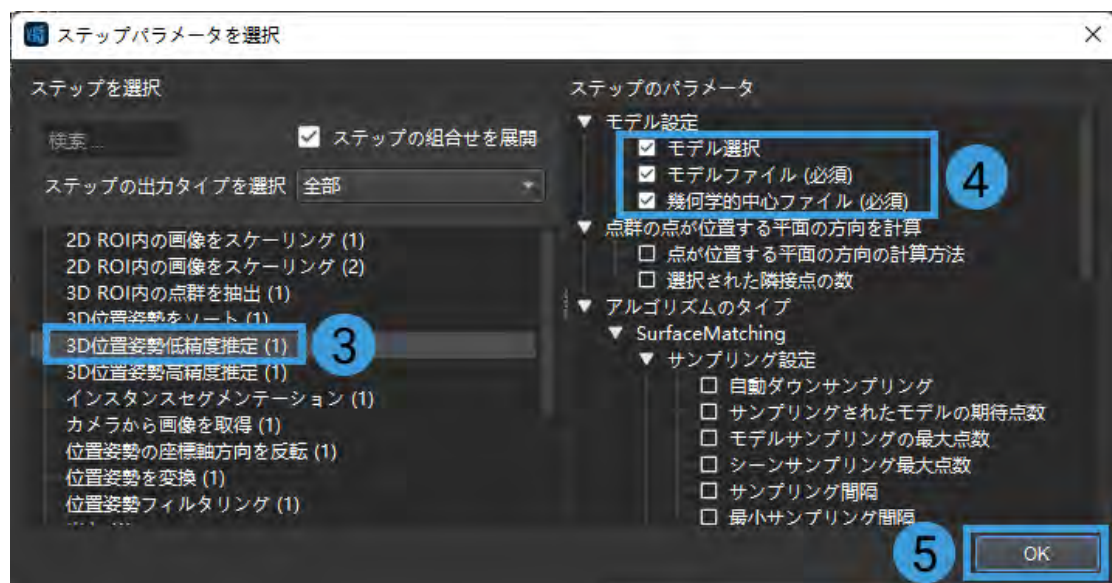
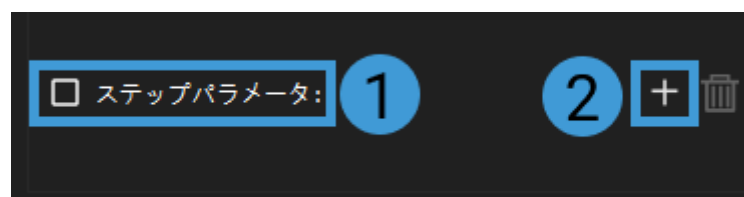
2. 保存期間で をクリックして、データの保存期間を設定します。設定した日数が経過すると、新しいデータが以前のデータを上書きします。



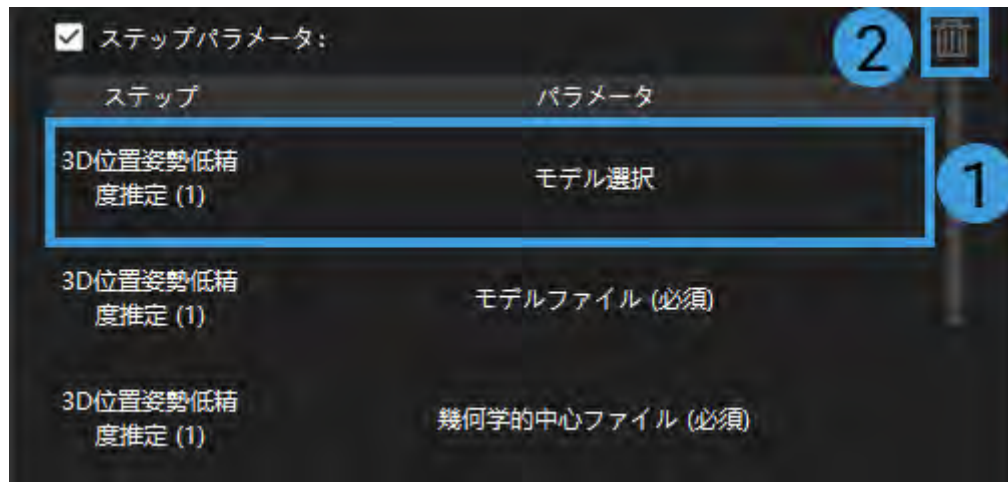
3. その他の設定を行います。

- 現在のレシピ：プロジェクトの実行時に使用されるレシピが保存されます。
- ステップパラメータ：保存するステップパラメータを選択できるので、データの再生時にステップが正常に実行されることが保証されます。

ステップパラメータにチェックを入れて、**+**をクリックし、ポップアップする**ステップパラメータを保存**の画面で保存するパラメータを選択し、**[確認]**をクリックします。



選択したステップのパラメータを削除するには、ステップパラメータリストでパラメータを選択し、**[-]**をクリックします。



- ° 仮想モードでは、再生中に保存されたパラメータを既存のパラメータで上書きする場合は、データ保存画面の下部にある**仮想モードでパラメータを上書きする**にチェックを入れることができます。



データ保存機能は複数のカメラデータを保存することをサポートします。これにより、Mech-Visionプロジェクトに複数の「カメラから画像を取得」ステップがある場合、複数のカメラからのデータが自動的に保存されます。

4.6.10. ログ

ログバーは、ビジョンソリューションやプロジェクト実行中の情報をリアルタイムに表示し、特定の時刻に何が実行されているかの記録を簡単に見つけることができます。

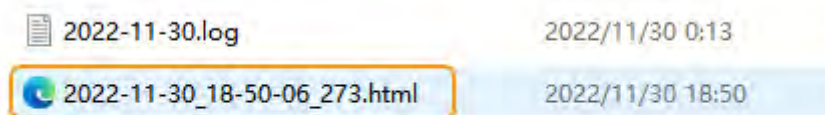
ログバーでは、2つのモジュールのログ、**Vision**と**Console**のログを切り替えることができます。**Vision**はビジョンソリューション/プロジェクトの実行中の情報を表示し、**Console**はコンソール情報を表示するために使用されます。

本節では、ログに関連する操作について説明します。

ログをエクスポート

ログバーの下部にある[**エクスポート**]ボタンをクリックすると、ポップアップウィンドウにHtml形式のログファイルが表示されます。



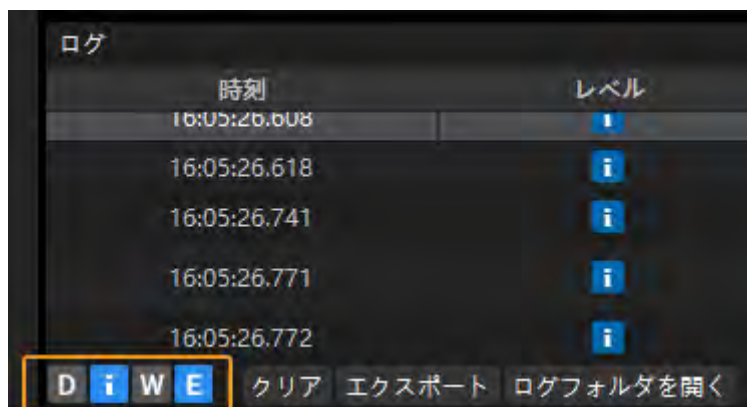


ログメッセージをフィルタリング

ログレベルの説明は次の表に示します。

ログレベル	説明
D	デバッグ情報
i	一般情報
W	ワーニング
E	エラー

特定のレベルのログメッセージを表示するには、ログバーの左下隅にあるログレベルをクリックします。ログレベルボタンの色が青色の場合は表示され、灰色の場合は表示されません。



その他の操作

ログをクリア

ログバーの下部にある[クリア]ボタンをクリックすると、現在のログバーに表示されている情報がクリアされます。

ログメッセージをコピー

- **ある行のログの一部メッセージをコピーするには**、ログバーのログメッセージをダブルクリックし、コピーする部分を選択し、右クリックで**コピー**をクリックすると、現在選択されているメッセージがコピーされます。
- **行全体のメッセージをコピーするには**、ログバーのある行のログをダブルクリックし、右ク

リックして**すべてを選択**をクリックすると、現在の行のすべてのログメッセージが選択されます。次にもう一度右クリックし、**コピー**をクリックすると、行全体のログメッセージがコピーされます。

5. Mech-Viz

Mech-Viz ユーザーズマニュアルへようこそ！

Mech-Viz はロボット経路計画用ソフトウェアとして、ビジョンシステムから受信したビジョン処理の結果に基づいて衝突のないロボット移動経路を生成します。グラフィカルユーザインタフェースにより、プログラミングの知識がなくてもかんたんに操作できます。ワンクリックでロボット移動をシミュレートすることができます。また、衝突検出や段ボール箱の組み合わせ、吸盤のスマートオフセット、混載パレタイジングなどのツールが組み込まれており、様々な現場で生産効率の向上に貢献できます。

本マニュアルでは、以下の内容について説明します：

番号	内容	意味
1	Mech-Vizの概要	基本的な概念の説明、ユーザーインターフェースの紹介、プロジェクト構築の手順
2	リソース設定	ロボットやロボットハンド、対象物、シーンなどのリソースの設定
3	ワークフローの構築	プロジェクト構築の概要と例
4	ステップライブラリ	ステップの詳細説明
5	プロジェクトのシミュレーションと最適化	衝突検出設定やプロジェクトシミュレーション、経路計画履歴、ログなどの機能の説明
6	ツールの使用	モデルエディタ、吸盤コンフィギュレータ、パレットエディタなど
7	付録	ロボットモデルの作成とインポート、オプション、表示の定など

5.1. Mech-Viz の概要

5.1.1. 基本概念

プロジェクト

Mech-Viz で作成するロボット経路計画プロジェクトです。Mech-Viz でプロジェクトに必要な設定を完了したら、このプロジェクトを使用してロボット経路を計画してロボットの動作を制御することができます。プロジェクトの設定は、プロジェクトと同名のフォルダに保存されます。

リソース

プロジェクトを構成するリソースです。ロボット、ロボットハンド、対象物、シーンの物体などがあります。

ロボット

Mech-Viz では、「ロボット」とは産業用の多関節ロボットまたはトラスロボットのことで

ロボットハンド

ロボットに作業を実行させるためにフランジに取り付ける装置です。グリッパや吸盤などがあります。

対象物

ロボットハンドが作用する物体です。段ボール箱や金属部品、接着剤塗布または溶接を実行する部品などがあります。

シーンの物体

ロボット実機が動作する空間にある物体です。ロボットの安全柵やコンテナ、パレット、カメラ、カメラブラケットなどがあります。

ワークフロー

Mech-Viz でフローチャートの形式で作成されたロボット動作制御プログラムです。

座標系

Mech-Viz では以下の座標系が使用されます。

世界座標系

仮想空間の中心を原点とする座標系。位置は固定されて変わりません。

ロボットベース座標系

原点はロボットベースの底面にあり、ロボットの位置とともに移動する座標系。ビジョン処理の結果はカメラの外部パラメータを通じてロボットの座標系に変換するとき、この座標系に変換します。デフォルトでは世界座標系と重なり合います。

ロボット TCP 座標系

ロボットハンドのエンドを原点とする座標系。

シーンの物体の座標系

シーンの物体モデルの中心を原点とする座標系。各シーンの物体モデルにシーンの物体の座標

系があります。

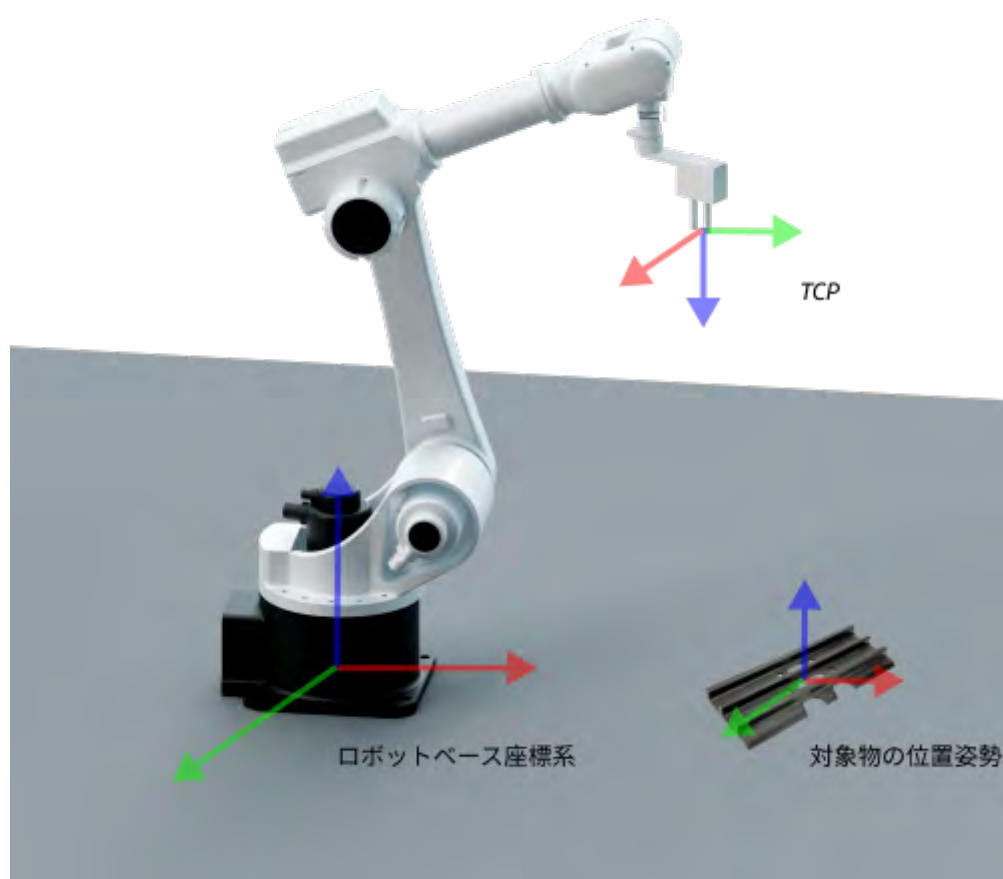
TCP（Tool Center Point、ツールセンターポイント）

ロボットハンドのエンドの点。

対象物把持などを実行する場合、ロボットをある点に移動するのは、TCP をこの点に移動することです。

対象物位置姿勢

対象物のある点がロボットとベースに対する位置姿勢。この点はロボットハンドの作用点です。



ロボットハンドの対称性

ロボットハンドを対称軸を中心に一定角度回転したら元の形状と完全に重なり合うことです。

対象物の対称性

対象物を対称軸を中心に一定角度回転したら元の形状と完全に重なり合うことです。

詳しくは [対象物の回転対称性](#) をお読みください。

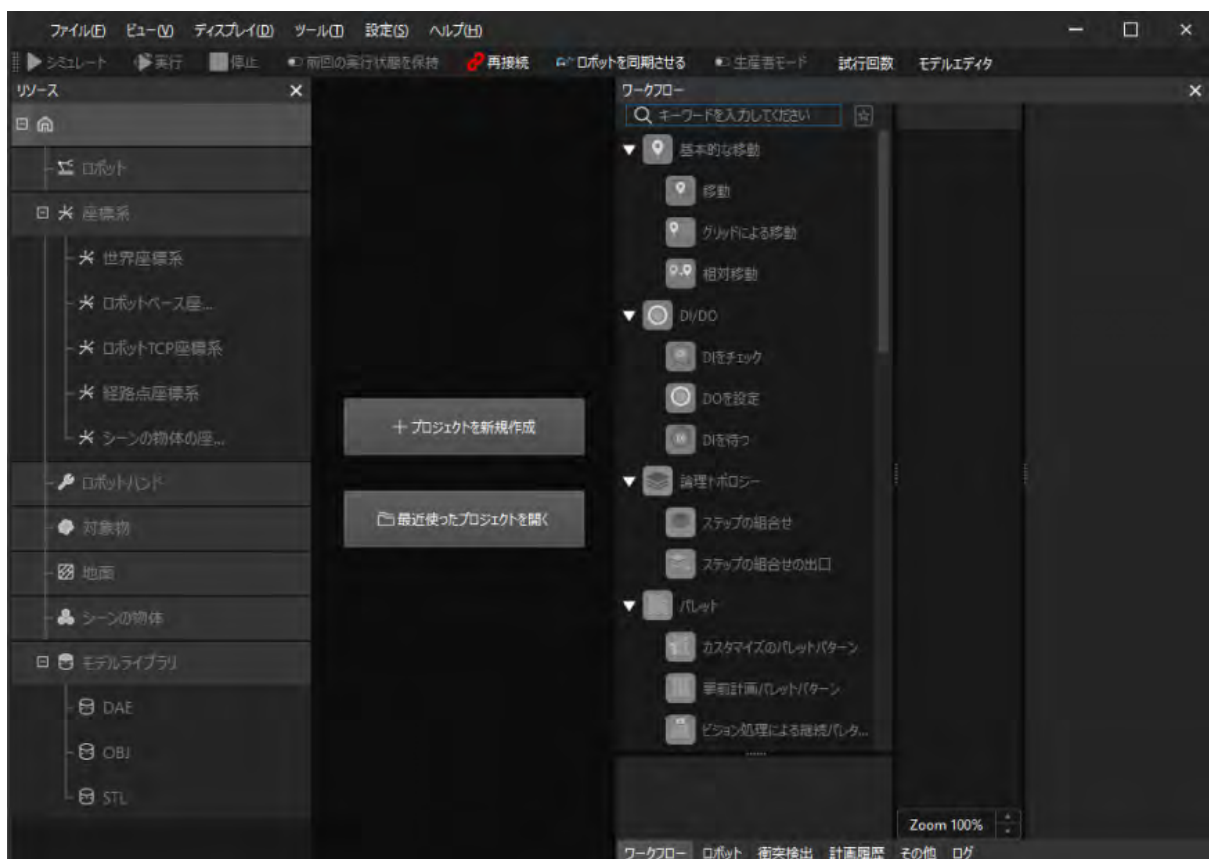
対象物の把持範囲

対象物を把持できない場合でも、ロボットハンドの位置姿勢を一定角度の範囲内で微調整することで把持できる場合もあります。この範囲を把持範囲といいます。

詳しくは [対象物の把持範囲](#) をお読みください。

5.1.2. ユーザーインターフェイス

Mech-Viz を起動した後、下図のような画面が表示されます。**プロジェクトを新規作成** または **最近使ったプロジェクトを開く** ことができます。



● プロジェクトを新規作成

1. [**プロジェクトを新規作成**] をクリックします。
2. 表示された画面の左上にロボット型番を入力するか、またはブランドなどからロボットを選択します。使用するロボットが見つからない場合、以下の手順を実行してください：
 - a. [オンラインロボットライブラリ](#) をクリックして使用するロボットのモデルをダウンロードします。オンラインロボットライブラリにもない場合、[ロボットモデルの作成とインポート](#) をお読みください。
 - b. **プロジェクトを新規作成** 画面を閉じてからロボットモデルをソフトウェアにドラッグ

グするか、または**ツール**、**ロボットライブラリ**、**ロボットをインポート**をクリックして表示された画面でダウンロードしたファイルを選択します。

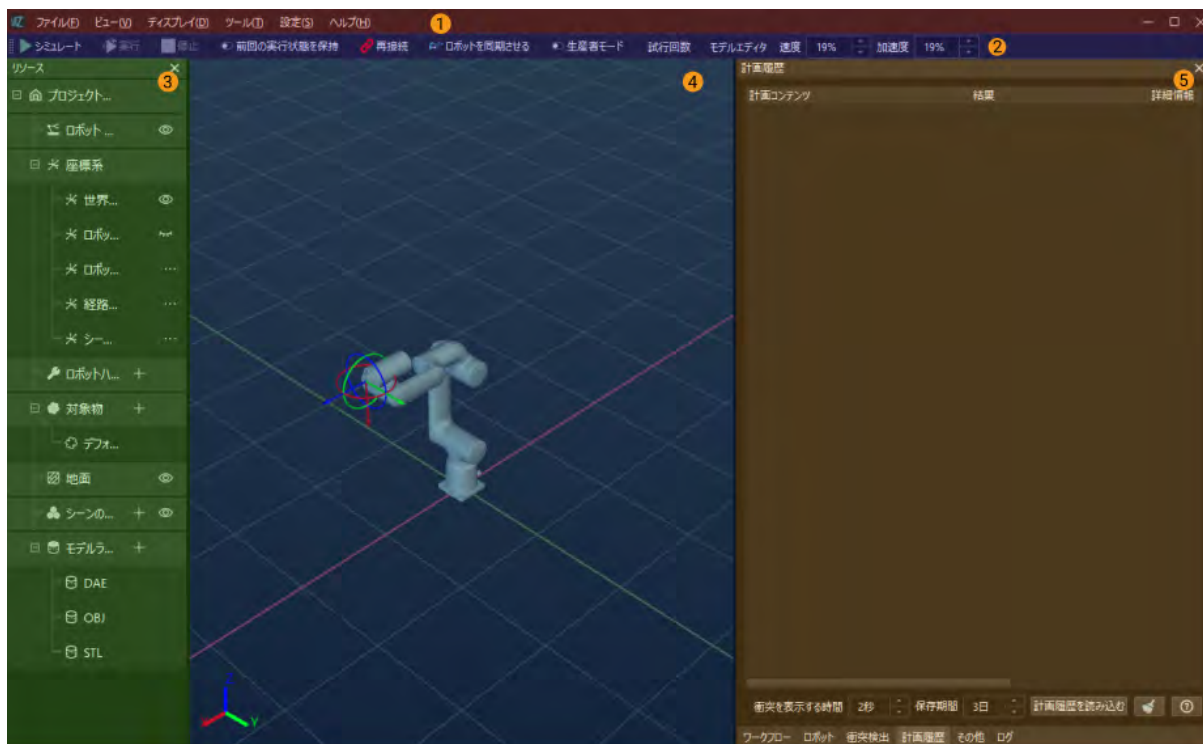
c. **[プロジェクトを新規作成]**をクリックしてロボットを再度検索します。

3. 使用するロボットをダブルクリックするか、またはカーソルを使用するロボットに合わせて**[選択]**をクリックします。

● 最近使ったプロジェクトを開く

[最近使ったプロジェクトを開く]をクリックして、使用するプロジェクトをダブルクリックするか、またはプロジェクトを選択してから**[開く]**をクリックします。

これで表示されているメインインターフェイスは下図のようです。



メインインターフェイスは五つの部分で構成されています：

1	メニューバー	ファイル、ビュー、ディスプレイ、ツール、設定、ヘルプ。 クリックして詳細な説明をお読みください
2	ツールバー	プロジェクトのシミュレートやロボットの制御、ロボット運動速度の調整などの機能がある 詳しくは ツールバー をお読みください
3	リソース	プロジェクトに使用するロボットハンドや対象物、シーンの物体などのリソースを管理する 詳しくは リソース をお読みください

4	仮想空間	プロジェクトを実行またはシミュレートするときにロボットの運動経路や衝突検出の結果、ビジョン位置姿勢、点群などを表示する 詳しくは 仮想空間 をお読みください
5	機能パネル	ワークフロー 、 ロボット 、 衝突検出 、 計画履歴 、 その他 、 ログ クリックして詳細な説明をお読みください

5.1.2.1. メニューバー

本節でメニューバーのボタンについて説明します。

ファイル(F) ビュー(V) ディスプレイ(D) ツール(T) 設定(S) ヘルプ(H)

ファイル

オプション	説明	ショートカット
プロジェクトを新規作成	新しいプロジェクトを作成する	Ctrl + N
プロジェクトを開く	既存のプロジェクトを開く	Ctrl + O
最近使ったプロジェクト	最近使ったプロジェクトを表示。プロジェクト名をクリックして開く	無し
実行可能ファイルの場所を開く	Mech-Viz のインストールフォルダを開く	無し
プロジェクトを保存	現在プロジェクトへの変更を保存する	Ctrl + S
プロジェクトを JSON 形式として保存	プロジェクトを保存し、プロジェクトのフォルダの .viz ファイルを .json 形式として保存する	無し
名前を付けてプロジェクトを保存	プロジェクトを指定場所に保存する	Ctrl + Shift + S
プロジェクトを閉じる	現在のプロジェクトを閉じる	Ctrl + W
終了	Mech-Viz を終了する	Ctrl + Q



Mech-Viz 1.4.0 以降のバージョンでは、プロジェクト実行ファイルを .json 形式の代わりに .viz 形式で保存します。

ビュー

ソフトウェアのインターフェイスを調整します。

オプション	説明
リソース	チェックすると「リソース」が表示される。デフォルトではチェックが入っている
ワークフロー	チェックすると「ワークフロー」パネルが表示される。デフォルトではチェックが入っている
ロボット	チェックすると「ロボット」パネルが表示される。デフォルトではチェックが入っている
衝突検出	チェックすると「衝突モデル」パネルが表示される。デフォルトではチェックが入っている
計画履歴	チェックすると「計画履歴」パネルが表示される。デフォルトではチェックが入っている
その他	チェックすると「その他」パネルが表示される。デフォルトではチェックが入っている
ログ	チェックすると「ログ」パネルが表示される。デフォルトではチェックが入っている
機能	チェックするとステップパラメータパネルにステップの機能の説明が表示される
デフォルトのレイアウト	クリックするとインターフェイスのレイアウトを初期設定に戻す

ディスプレイ

仮想空間のディスプレイを調整します。

オプション	説明
受信したビジョン位置姿勢を表示	チェックすると受信したビジョン位置姿勢とラベル、インデックスが表示される。デフォルトではチェックが入っている
点群を表示	チェックすると受信した点群が表示される。デフォルトではチェックが入っている
把持された対象物を表示	チェックすると把持された対象物のモデルが表示される。デフォルトではチェックが入っている
計画中に衝突を表示	チェックすると仮想空間に計画中に検出した衝突の位置が表示される。デフォルトではチェックが入っている

オプション	説明
八本木を表示	チェックすると物体の点群が八本木の形式で表示される。デフォルトではチェックが外れている
受信した箱モデルを表示	チェックすると仮想空間に受信した箱モデルが表示される。デフォルトではチェックが入っている
対象物の位置姿勢を表示	チェックすると仮想空間に把持対象物の位置姿勢が表示される。デフォルトではチェックが外れている
ディスプレイ設定	仮想空間のディスプレイをさらに詳細に設定する
位置姿勢の状態を表示する色	位置姿勢のインデックスとラベルの色が表示する位置姿勢の状態を確認する

ツール

オプション	説明
保存されたビジョン処理の記録を使用	チェックすると保存されたビジョン処理の記録を使用してプロジェクトを実行する
ビジョン処理の記録を設定	ビジョン処理の記録を保存して呼び出す。デバッグするときにトラブルを再現するために使用する
ビジョン処理の結果を検索	仮想空間で検索するインデックスが対応するビジョン処理の結果をハイライト表示する
ロボットライブラリ	ロボットモデルファイルをインポートする。また、お使いのロボットモデルを指定した場所にエクスポートする

設定

オプション	説明
Mech-Center アドレスを設定	Mech-Center の IP アドレスを設定する
プロジェクトをロック/ロック解除	チェックすると現在のプロジェクトをロックする
オプション	ソフトウェアの設定

オプション	説明
ログレベル	ログのレベルを設定する。ログパネルに選択したレベル以上のログが表示される

ヘルプ

ソフトウェアについて	ソフトウェアのバージョンを確認する
更新説明	ブラウザでソフトウェア更新説明を確認する
ユーザーズマニュアル	ブラウザでユーザーズマニュアルを閲覧する

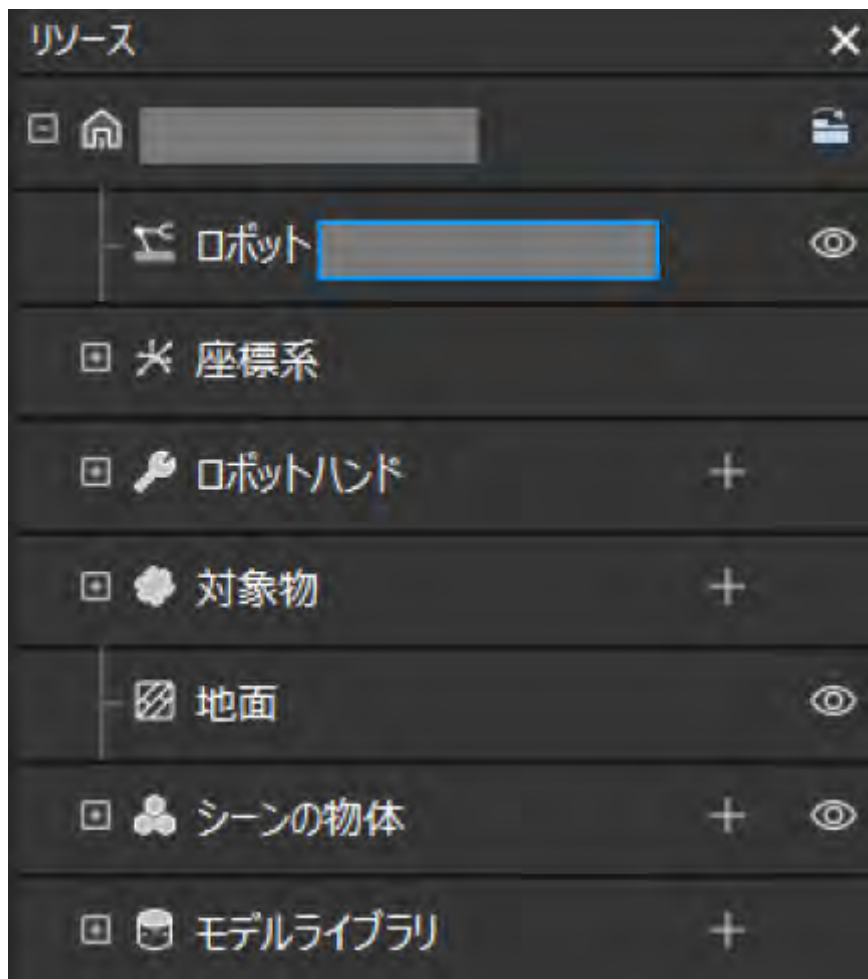
5.1.2.2. ツールバー

本節ではツールバーのオプションについて説明します。

シミュレート	仮想ロボットを実行する
実行	ロボット実機を実行する
停止	現在実行しているプロジェクトを終了する
前回の実行状態を保持	パレタイジングのデバッグに使用され、続けてパレタイジングを実行する
再接続	Mech-Center と接続し、ロボットとのデータ通信を実現する
ロボットを同期させる	仮想ロボットにロボット実機の位置姿勢を同期させる
生産モード	実行の速度を改善するために点群をダウンサンプリングする
試行回数	対象物ごとに把持を試行する回数
モデルエディタ	モデルエディタを開く
速度	ロボットの運動速度を調整する
加速度	ロボットの運動加速度を調整する

5.1.2.3. リソース

リソースパネルは下図に示します。



本節では、リソースのドロップダウンメニューのオプション及びボタンの機能について説明します。

リソースの設定については、[リソース設定](#)をお読みください。

プロジェクト

プロジェクト名とプロジェクトの変更の有無、自動的に読み込むかどうかを表示します。

プロジェクト名を右クリックすると、以下のオプションがあります：

プロジェクトフォルダを開く

プロジェクトの場所を開く

自動的に読み込む

チェックすると次回ソフトウェアを開くときに自動的にこのプロジェクトを開く

現在のプロジェクトをバックアップ

現在のプロジェクトをバックアップする

また、ワークフローの変更が保存されていない場合、自動的に読み込むように設定された場合、下図のようなマークが表示されます。





ロボット

現在のプロジェクトに使用するロボットを表示します。




ロボット名を右クリックすると、以下のオプションが表示されます：

ロボットファイルディレクトリを開く ロボットモデルファイルが格納しているフォルダを開く

また、をクリックするとロボットが半透明の状態になり、をクリックすると元に戻ります。

座標系

仮想空間にあるすべての座標系を表示します。

をクリックすると対応する座標系を表示します。をクリックすると非表示します。をクリックすると、ドラッグの「並進」と「回転」機能をオン/オフにします。



ロボットハンド

プロジェクトに使用するロボットハンドを表示します。ロボットハンドの設定については、[ロボットハンド](#)をお読みください。

[+]をクリックしてロボットハンド設定ウィンドウを開きます。ロボットハンドを追加できます。

追加済みのロボットハンドを右クリックすると、以下のオプションがあります：

オプション	説明	ショートカット
コピー	選択したロボットハンドをコピーする	Ctrl + C
貼り付け	コピーしたロボットハンドを貼り付ける	Ctrl + V
削除	選択したロボットハンドを削除する	Delete
名前を変更	選択したロボットハンドの名前を変更する	
ロボットハンド設定	選択したロボットハンドの設定画面にアクセスする	
現在のロボットハンドに設定	現在のロボットハンドを可動ロボットハンドに設定する	

をクリックするとロボットが半透明になり、をクリックすると元の状態に戻ります。

対象物

プロジェクトに使用する対象物を表示します。詳しくは [対象物](#) をお読みください。

[+] をクリックして対象物設定ウィンドウを開きます。対象物を追加します。

追加済みの対象物を右クリックすると、以下のオプションがあります：

オプション	説明	ショートカット
削除	現在の対象物を削除する	Delete
名前を変更	現在の対象物の名前を変更する	
対象物設定	現在の対象物を設定するために設定ウィンドウを開く	





デフォルトはラベルがないビジョン処理の結果に使用する対象物設定です。

地面

地面の高さを変更し、表示/非表示を選択します。

右クリックして地面の高さを調整することができます。

 をクリックすると地面が非表示になり、 をクリックすると元に戻ります。




シーンの物体

プロジェクトのシーンの物体を表示します。詳しくは [シーンの物体](#) をお読みください。

[+] をクリックして設定ウィンドウを開きます。シーンの物体を追加します。

追加済みシーンの物体を右クリックすると以下のオプションが表示されます：

オプション	説明	ショートカット
コピー	現在のシーンの物体をコピーする	Ctrl + C
貼り付け	シーンの物体を貼り付ける	Ctrl + V
切り取り	現在のシーンの物体を切り取る	Ctrl + X
削除	現在のシーンの物体を削除する	Delete
名前を変更	現在のシーンの物体の名前を変更する	
シーンの物体設定	現在のシーンの物体を設定するウィンドウを開く	

 をクリックするとシーンの物体が半透明になります。
 をクリックすると透明になります。
 をクリックすると元の状態に戻ります。

また、シーンの物体名をドラッグして子モデルまたは独立したモデルに変換することもできます。

モデルライブラリ

追加された外部モデルを表示します。モデルの形式によって分類します。

[+] をクリックしてファイル選択ウィンドウを開きます。外部モデルをインポートします。

インポートされた外部モデルを右クリックすると以下のオプションがあります：

オプション	説明	ショートカット
コピー	現在のモデルをコピーする	Ctrl + C
貼り付け	コピーしたモデルを貼り付ける	Ctrl + V
削除	選択したモデルを削除する	Delete
モデル設定	現在のモデルを編集するウィンドウを開く	
モデルエディタ	クリックして現在のモデルを基準モデルとする	
フォルダを開く	現在のモデルの場所を開く	

5.1.2.4. 仮想空間

仮想区間では、シーンの物体とロボットハンド、対象物モデル、ロボットの運動経路、把持位置姿勢、衝突などが表示されます。また、ここでシーンの物のモデルをドラッグすることもできます。

基本的な操作

以下の操作により仮想空間のディスプレイを調整します。

視点を回転する	マウスの左ボタンを押したままドラッグする	
視点を切り替える	マウスの右ボタンをクリックしてショートカットメニューで選択する	
視点を並進する	マウスのホイールを押したままドラッグする	
拡大・縮小	ホイールを回す	

シーンの物体の表示

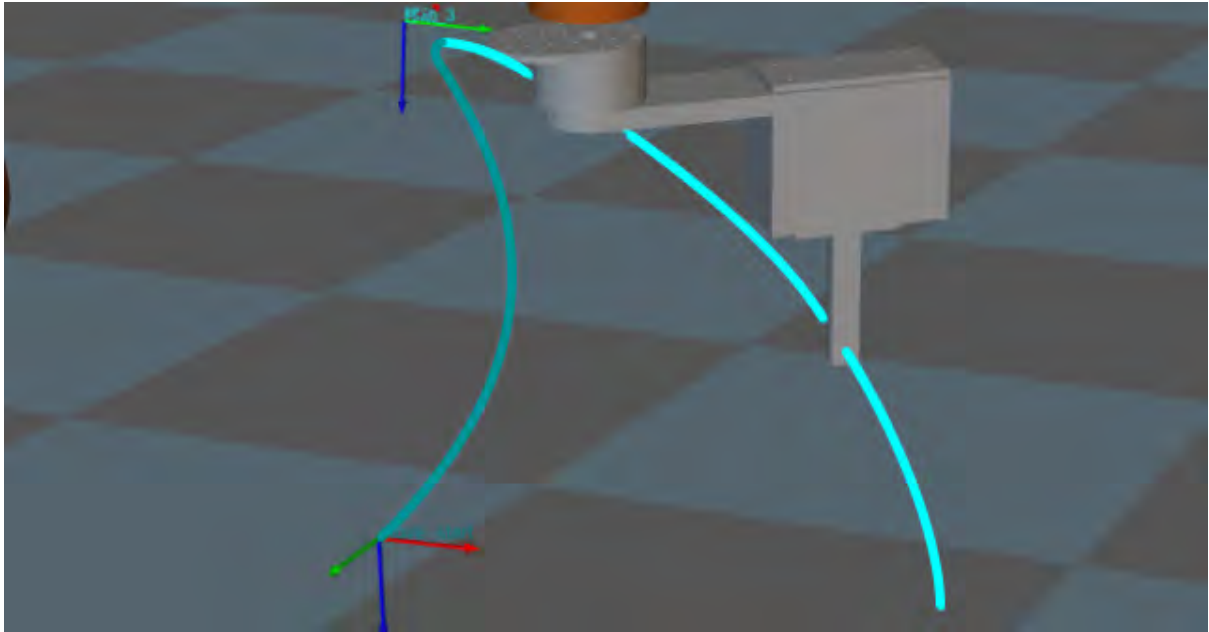
プロジェクトを構築する際に、仮想空間は下図のようになります。



- ① シーンの物体、② ロボット、③ ロボットハンド、④ 地面

運動経路の表示

ロボット実機を制御する時も、仮想ロボットの運動経路をシミュレートする時も、これからロボットが実行する経路が仮想空間に表示されます。経路をシミュレートすることでより合理的な動作経路を計画することができます。



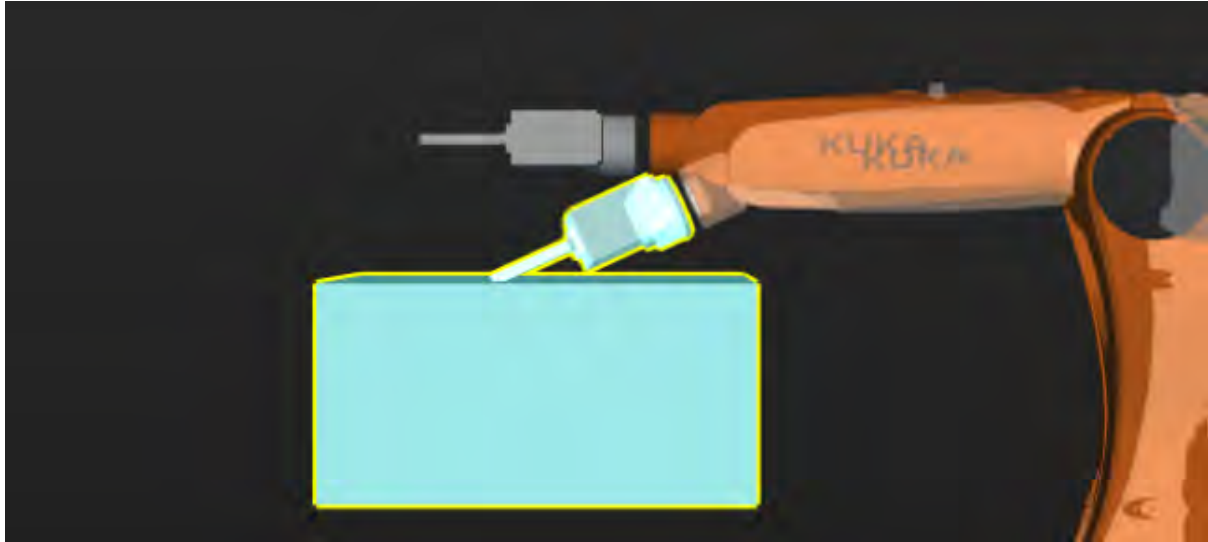
点群と把持点の表示

ビジョンサービスによって取得した結果が仮想空間に表示されます。普通、完全なビジョン結果には、カメラでキャプチャした対象物とシーンの点群、対象物の位置姿勢、対象物の番号、対象物の特徴を表示するラベルが含まれます。



衝突判断の表示

プロジェクト実行中、Mech-Viz はロボット運動経路にロボット実機とエンドエフェクタがシーンの物体、ワーク、箱と衝突するかを検出します。下図に示すように、衝突を検出したら、仮想空間では衝突が発生するものがハイライト表示されます。



5.1.2.5. ワークフロー

ワークフローパネルは下図に示します。



ワークフローパネルは①ステップライブラリ、②ワークフロー編集エリア、③パラメータ編集エリア、④操作履歴エリアで構成されます。

ロボットプログラミングを完成するためにワークフローを構築します。詳しくは [ワークフローの構築](#) をお読みください。

ステップライブラリ



ステップの機能によって複数のカテゴリに分類しています。例えば、「基本的な移動」カテゴリは、移動や相対移動、グリッドによる移動などのロボットの一般的な移動ステップを含みます。ステップカテゴリ名をクリックするとこのカテゴリのすべてのステップを展開/折りたたみます。

ステップは主に「移動ステップ」と「非移動ステップ」に分けられています。

- 移動ステップ：基本的な移動、パレタイジング、ビジョン処理による移動

- 非移動ステップ：DI DO、論理トポロジー、ロボットツール、サービス、ロボットハンド、ビジョン（ビジョン処理による移動を除く）、ほか

ステップライブラリパネルで以下の操作ができます。

- ステップライブラリの右上の  をクリックしてすべてのステップを表示します。
-  をクリックしてよく使うステップだけを表示します。
- 右クリックしてドロップダウンメニューの**すべて表示**をクリックしてすべてのステップを表示します。
- 右クリックしてドロップダウンメニューの**よく使うステップだけを表示**をクリックしてよく使うステップだけを表示します。
- 右クリックしてドロップダウンメニューの**展開**をクリックしてすべてのステップを展開します。
- 右クリックしてドロップダウンメニューの**折りたたむ**をクリックしてすべてのステップを折りたたみます。

また、ステップライブラリの検索窓にキーワードを入力してステップを検索することもできます。

ワークフロー編集エリア

ワークフロー編集エリアではステップを組み合わせてロボットプログラミングを実行します。

右クリックすると以下のオプションが表示されます：

前のページへ	ステップの組み合わせの上位ページへジャンプする
次のページへ	ステップの組み合わせを開く（ステップの組み合わせをクリックする）
コピー	選択したステップをコピーする
切り取り	選択したステップを切り取る
貼り付け	選択したステップを貼り付ける
削除	選択したステップを削除する
すべて選択	ワークフロー編集エリアにあるすべてのステップを選択する
プロジェクトのスクリーンショットを保存	スクリーンショットを保存する
自動レイアウト	すべてのステップを自動的にレイアウトする

ステップの組み合わせに合成	選択した複数のステップを一つの組み合わせに合成する
ステップの組合せをステップに分割	選択したステップの組み合わせを分割する
ステップの組み合わせをインポート・エクスポート	選択したステップの組み合わせをエクスポートし、またはステップの組み合わせをインポートする
ディスクリプションをカスタマイズする	選択したステップにディスクリプションをカスタマイズする
DO をリセット	DO を 0 にリセットする。普通、吸盤をオフにする時に使用する

プロジェクト編集エリアで左クリックして、**Ctrl + F** を押すと検索窓が表示されます。検索窓でワークフローのステップを検索・選択することができます。

また、**ctrl** を押したままマウスホイールを回転させると画面を拡大/縮小することができます。

パラメータ編集エリア

ワークフロー編集エリアでいずれかのステップを選択すると、パラメータ編集エリアにそのステップの機能と調整可能なパラメータが表示されます。

操作履歴エリア

ステップの追加・削除、つながりの追加・削除など、ワークフロー編集エリアで実行した操作を記録します。

いずれかの記録をクリックすると、ワークフロー編集エリアはこの記録の状態に戻ります。

5.1.2.6. ロボット

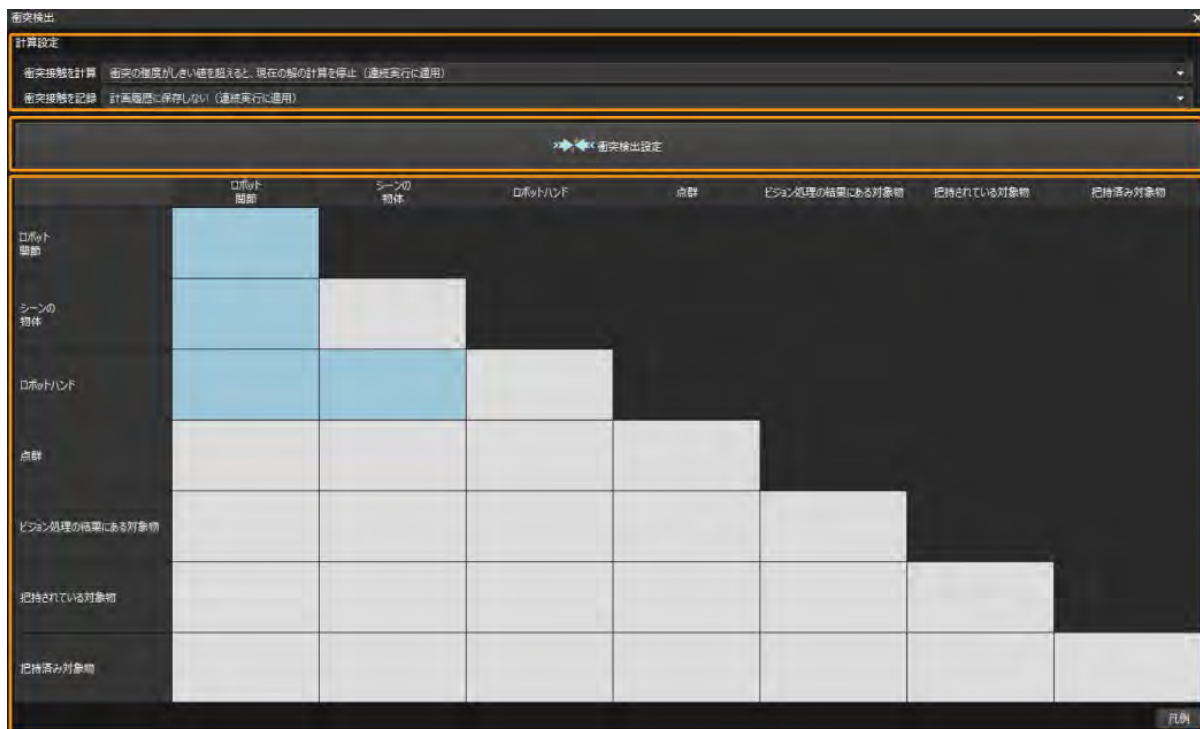
ロボットパネルは下図に示します。



ロボットパネルではロボットモデルの交換やロボットソフトリミットの設定、ロボット実機の制御、ロボット位置姿勢の調整などができます。詳しくは [ロボット設定](#) をお読みください。

5.1.2.7. 衝突検出

衝突検出パネルは下図に示します。



計算設定：実行の速度を改善できます。詳しくは [計算設定](#) をお読みください。

衝突検出設定：[[衝突検出設定](#)] をクリックして衝突検出設定ウィンドウを開きます。ここで点群やロボットハンド、ロボット関節、把持対象物の衝突検出を設定できます。詳しくは [衝突検出](#) をお読みください。

衝突検出テーブル：衝突検出の設定を表示します。[[凡例](#)] をクリックして異なる色の意味を確認します。

5.1.2.8. 計画履歴

計画履歴パネルは下図に示します。



計画履歴には、Mech-Viz の計画プロセスが詳細かつ完全に記録されます。トラブルシューティングやプロジェクトの最適化に使用できます。詳しくは、[計画履歴](#) をご参照ください。

計画履歴パネルの下にパラメータを設定できます：

衝突を表示する期間	衝突による失敗の結果を選択すると、仮想空間に衝突を表示する時間
保存期間	計画履歴のデータを保存する期間。この期間が切れたら新たなデータに上書きされる
計画履歴を読み込む	保存された計画履歴を読み込みむ
クリア	現在表示されている計画履歴をクリアする
ヘルプ	計画履歴に関する説明

計画履歴を右クリックすると以下のオプションが表示されます：

検索	種類から計画履歴を検索する
すべて折りたたむ	子ノードをすべて折りたたみ、親ノードのみを表示する

選択された計画履歴を削除 選択された計画履歴を削除する

5.1.2.9. その他

実行、ステップ収集、特異点、全体的関節制限などを設定できます。

実行設定

ワークフローの実行進捗を見積もる（Move IDを返信できないロボット）

ロボットが Move ID を返信できず、短いタクトタイムを必要とする場合、この機能を使用します。Mech-Viz は現在のロボットの位置姿勢に基づいてワークフローの実行進捗を判断します。ただ、この機能を使用すると、ワークフローにある隣接する移動ステップと非移動ステップの実行開始時間が間違う可能性があります。

隣接する経路点をスキップ

隣接する二つの経路点の前の経路点だけをロボットに送信します。

デフォルトの対象物設定だけを使用

チェックすると、対象物のラベルを受信した場合でもデフォルトの対象物設定を適用します。

ロボットサービスのタイムアウト時間

Mech-Viz とロボットの通信のタイムアウト時間（移動ステップと「制御を移転」ステップを除く）。デフォルトでは 1000ms となります。

ステップ収集

移動ステップの最大収集数

経路計画に使用する移動ステップの最大数です。プロジェクトに移動ステップの数が指定した値を超えると経路計画が中断されます。

非移動ステップの最大収集数

経路計画に使用する非移動ステップの最大数です。プロジェクトに非移動ステップの数が指定した値を超えると経路計画が中断されます。

特異点

ロボットが特異点の付近に到達すると直線運動を実行できない可能性があります。事前にソフトウェアで特異点を回避するために設定する必要があります。

特異点検出方法 には二つのオプションがあります：

- ロボットの各関節の速度を測定

関節運動の最大速度と減速比を設定することでロボットが特異点に移動する経路計画を回避します。

最大関節速度

関節運動の速度がこの値を超えると特異点と判断される可能性があります。

特異点の減速比

減速比がこの値を超えると特異点と判断される可能性があります。減速比 = 減速後の速度（加速度） / 設定した初期速度（加速度）。

● ロボットの関節角度を測定

関節の角度の範囲を設定してロボットが特異点に移動する経路計画を回避します。

関節を選択

ロボットの関節を選択します。

角度の下限

関節角度がこの値より小さい場合、特異点と判断される可能性があります。

角度の上限

関節角度がこの値より大きい場合、特異点と判断される可能性があります。

全体的関節制限

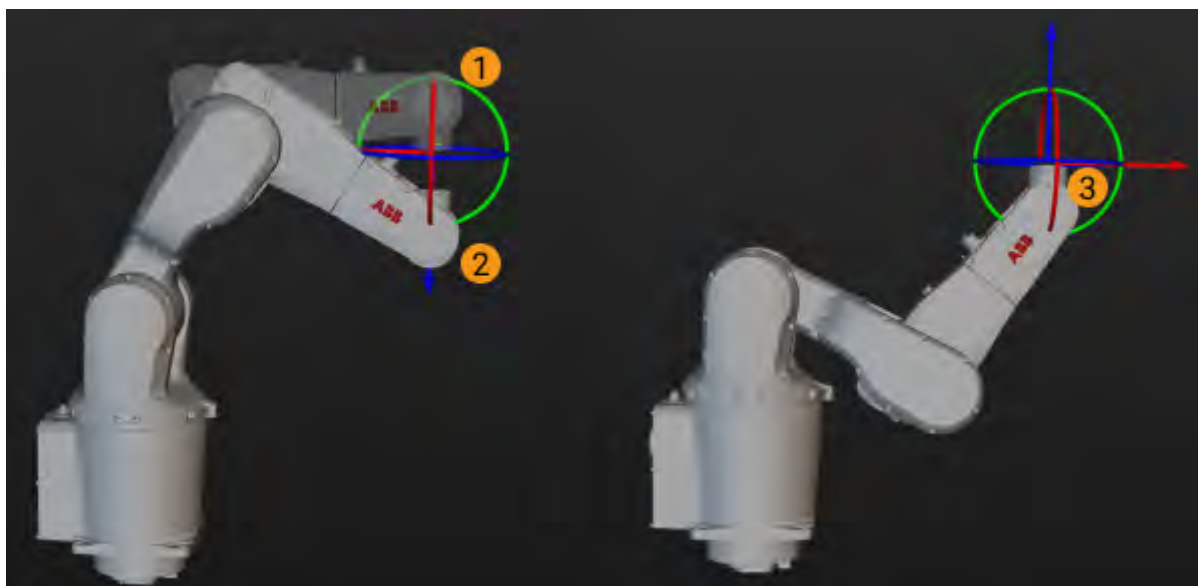


移動ステップに対し、パラメータの「関節角度の制約条件」を「自動」に設定した場合、Mech-Viz では全体的関節制限の設定を適用します。「自動」以外に設定した場合は「関節角度の制約条件」の設定を適用します。

肩/肘/手首を回転させない

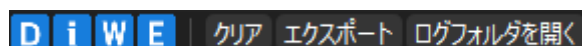
対象物を把持している時に と 実行全過程に が選択可能で、ロボットの不要な回転を低減します。

「肩/肘/手首を回転させない」 に設定しても不要な回転が発生する可能性があります。例えば：



ロボットが1から2まで移動する場合、「手首を回転させない」ように設定するとより大きな回転が起き、ロボットが3の姿勢になるかもしれません。この場合、「肩を回転させない」と「肘を回転させない」だけ設定してください。

5.1.2.10. ログ

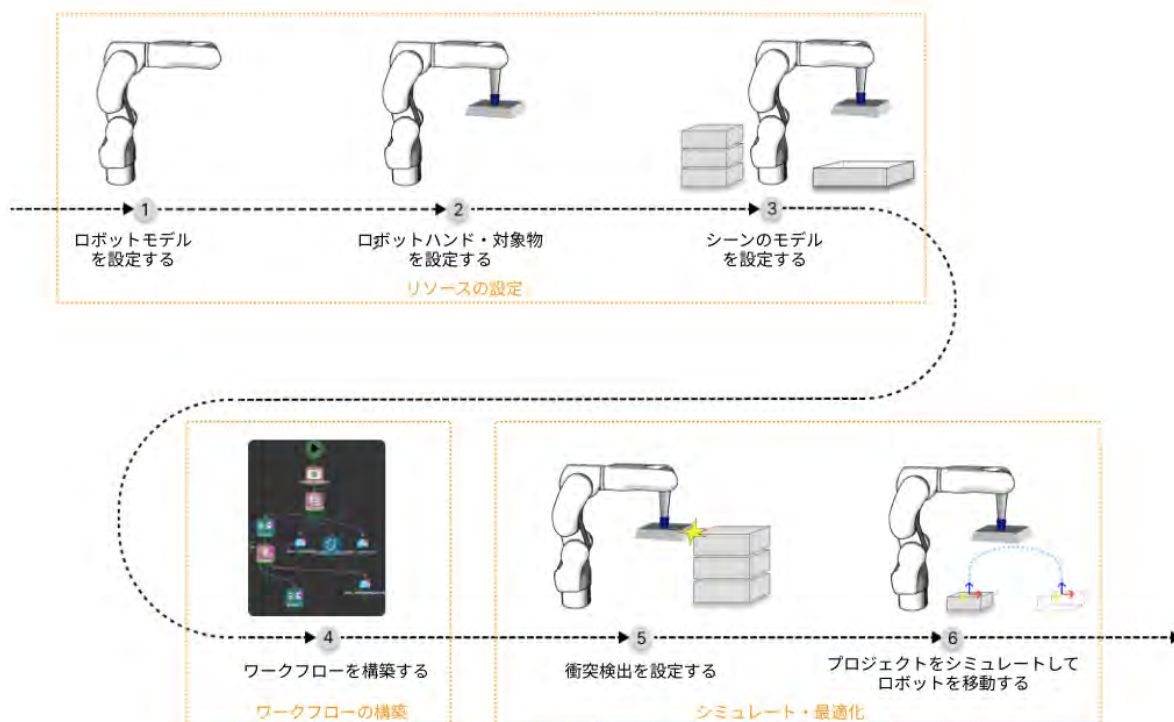


プロジェクト実行中の詳細情報がログに表示されます。また、ログパネルの下にパラメータを設定できます。いずれかのコンテンツを選択すると、**Ctrl + C**を押してコピーすることができます。

D、i、W、E	ログのレベルによって表示するログを選択する。複数選択可能
	D：デバッグ情報 i：一般情報 W：ワーニング E：エラー
クリア	クリックすると現在のログメッセージがクリアされる
エクスポート	現在表示されているログ情報を HTML 形式でエクスポートする。それは、Mech-Viz インストールフォルダ中の logs フォルダに保存される
ログフォルダを開く	Mech-Viz インストールフォルダ中の logs フォルダを開く。logs フォルダには時間付きのログファイルを確認可能

5.1.3. プロジェクトの構築手順

実際の応用では、Mech-Viz によるプロジェクト構築の手順は下記の通りです。



5.2. リソース設定

リソースとは、ロボットやロボットハンド、対象物、シーンの物体など、プロジェクトを構築するための要素を指します。

リソース設定の詳しい説明

以下の節で各リソースの設定について説明します。

- [ロボット](#)
- [ロボットハンド](#)
- [対象物](#)
- [地面](#)
- [シーンの物体](#)

また、プロジェクト名を右クリックしてドロップダウンメニューからプロジェクトファイルディレクトリを開き、自動的に読み込む、現在のプロジェクトをバックアップするように設定することができます。

リソースのモデル

仮想空間では、リソース（ロボットとロボットハンド、対象物、シーンの物体）のモデルをインポートする場合、**3D モデル**と**衝突モデル**をインポートしなければなりません。

3D モデル 仮想空間で表示されるだけで、衝突検出に使用されません。

衝突モデル 衝突検出に使用されるだけで、仮想空間では表示されません。

各リソースがサポートするモデル形式は下表に示します：

		STL	OBJ	DAE	Binvox
ロボット	3D モデル	√	√	√	×
	衝突モデル	√	√	×	×
シーンの物体	3D モデル	√	√	√	×
	衝突モデル	√	√	×	×
ロボットハンド	3D モデル	√	√	√	×
	衝突モデル	×	√	×	×
対象物	3D モデル	√	×	×	×
	衝突モデル	×	×	×	√

5.2.1. ロボット

本節ではロボット設定について紹介します。

概要

プロジェクトの構築はロボットの選択から始まります。ソフトウェアに使用するロボットモデルは、実際の動作環境に使用するロボットとは一致しなければなりません。

ロボットモデル

Mech-Viz のロボットライブラリには、ABB、FANUC、KAWASAKAI、KUKA、UR、YASKAWA などのロボットモデルが組み込まれています。また、[オンラインロボットライブラリ](#) にアクセスしてロボットモデルをダウンロードし、[ロボットモデルのインポート](#) することもできます。オンラインロボットライブラリに使用するモデルがない場合に、[ロボットモデルの作成とインポート](#) をお読みください。

ロボットの特徴

Mech-Viz ではロボットによって異なる機能をサポートしており、Mech-Viz のロボット制御に影響が出ます。詳しくは**ロボットの特徴**をお読みください。

ロボットのソフトリミット

ロボットのソフトリミットは、ロボットの軸の動作範囲を制限します。ロボットの軸の動作範囲内にある物体とは干渉する可能性があります。規格外品のロボットハンドとケーブルもロボットの動作に影響することがあります。前もってソフトリミットを設定してロボット径路に到達できない位置姿勢を回避することができます。

ロボットの初期位置

ロボットの初期位置とは、ユーザーが設定するロボット運動の初期位置です。

ロボットの設定

ロボットタブでは、仮想ロボットの交換やソフトリミットの設定、仮想ロボットの位置姿勢の調整、ロボット初期位置の設定、ロボット実機の移動、ロボットモデルのインポートなどの機能が実現できます。

仮想ロボットの切り替え

ようこそ画面でロボットを選択した場合、ロボットのを変更するとき、以下の手順を実行します：

1. **[ロボットライブラリ]**をクリックします。
2. 検索ボックスにロボットの型番を入力するか、ブランドや可搬重量、稼働範囲などを選択することで使用するモデルを選択します。見つからない場合は以下の手順を実行します：
 - a. **オンラインロボットライブラリ**をクリックして使用するロボットのモデルをダウンロードします。オンラインロボットライブラリに使用するモデルがない場合、**ロボットモデルの作成とインポート**をお読みください。
 - b. ロボットモデルファイルをソフトウェアにドラッグするか、**ツール > ロボットライブラリ > ロボットをインポート**をクリックして表示されたウィンドウでファイルを選択します。
 - c. 改めて検索します。
3. 使用するロボットをダブルクリックするか、カーソルをロボットに合わせて**[選択]**をクリックします。

ソフトリミットの設定

1. **[ソフトリミットを設定]**をクリックします。
2. 表示されたウィンドウで各軸の最大値と最小値を修正します。
- 3.

[OK] をクリックします。



この修正は Mech-Viz のソフトリミットを対象とし、Mech-Viz にのみ有効になります。ロボットティーチペンダントのソフトリミット設定に干渉しません。

仮想ロボット位置姿勢の調整

関節角度を調整することで仮想ロボットの位置姿勢を調整します。以下のいずれかの手順を実行します：

- 関節角度パラメータパレットでスライダをドラッグするか、または値を入力します。仮想ロボットの各軸の移動が見られます。
- [編集] をクリックして関節角度の値を入力し、[OK] をクリックします。

ロボットハンドの位置姿勢を調整することで仮想ロボットの位置姿勢を調整します。以下のいずれかの手順を実行します：

- TCP タブに切り替えてオイラー角または四元数を調整します。仮想ロボットの各軸の移動が見られます。
- [位置姿勢を調整] をクリックして TCP の値を入力し、[OK] をクリックします。
- ロボット実機を接続した場合に、[TCPを更新] をクリックしてロボット実機の TCP 位置姿勢を仮想ロボットに同期させます。

ロボットの初期位置の設定

1. ロボットの位置姿勢の修正については、[仮想ロボット位置姿勢の調整](#) をお読みください。
2. [関節角度] をクリックし、ロボットの初期位置を設定します。



仮想ロボットが他の位置姿勢にある場合、[仮想ロボットを初期位置に戻す] をクリックします。

ロボット実機を動かす

ロボット実機を接続したあと、ロボット実機を仮想ロボットの位置姿勢に移動させることができます。

1. ロボットモデルの位置姿勢の修正については、[仮想ロボット位置姿勢の調整](#) をお読みください。
2. 速度 を 5% 以下に設定します。
3. [ロボット実機を移動] をクリックします。

ロボットモデルのインポート

オンラインロボットライブラリからモデルファイルをダウンロードするか、またはモデルファイルを作成してから以下のいずれかの手順を実行します：

- モデルファイルをソフトウェアにドラッグします。
- ツール ▾ ロボットライブラリ ▾ ロボットをインポート をクリックして表示されたウィンドウでファイルを選択します。

5.2.2. ロボットハンド

本節では、ロボットハンドとその関連設定について説明します。

概要

ロボットハンドとは、ロボットが作業を行うために特別に設計され、ロボットのフランジに取り付けられるグリッパーや吸盤などの装置です。

ロボットハンドモデル

仮想空間でロボットハンドを表示し、ロボットハンドの衝突検出を有効にするために、ロボットハンドモデルと衝突モデルを作成してソフトウェアのモデルライブラリにインポートする必要があります。

対応可能な形式は以下の通りです。

形式	STL	OBJ	DAE	Binvox
3D モデル	√	√	√	×
衝突モデル	×	√	×	×



- ソフトウェアでは、ロボットハンドの衝突モデルが各部分の凸多面体（凸包）の obj モデルである必要があります。[モデルエディタ](#)を使用してロボットハンドモデルの凸多面体を構築し、obj 形式に変換することができます。
- 3D モデルは仮想空間での表示にのみ使用でき、衝突検出が有効になりません。衝突モデルは衝突検出にのみ使用でき、仮想空間で表示されません。


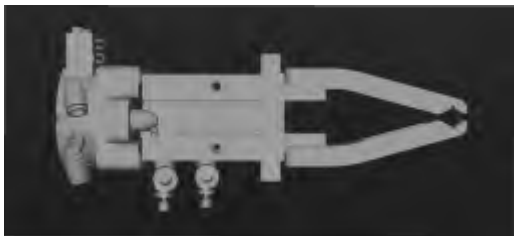
ロボットハンドの種類

Mech-Viz では、一般的なロボットハンド、デパレタイズ用吸盤、配列タイプグリッパに対応しています。

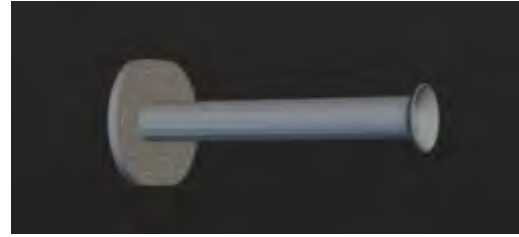
ロボット ハンドの 種類	説明	サンプル図
一般的な ロボット ハンド	デパレタイズ用吸盤と配列タイプグリッパ以外のロボットハンド。例えば：シリンダーグリッパー	
デパレタイズ用吸盤	長方形のデパレタイズ用吸盤です。複数のブロックに分けて制御することができる	
配列タイプグリッパ	複数のエンドで構成され、かつ同時に動作できるグリッパ	

ロボットハンドの対称性

ロボットハンドを対称軸を中心に一定角度回転したら元の形状と完全に重なり合うことです。ロボットハンドの対称性を設定することでロボットが把持や配置を実行する時の無駄な回転を減少し、動作実行の成功率を改善でき、またロボットがよりスムーズに動作します。

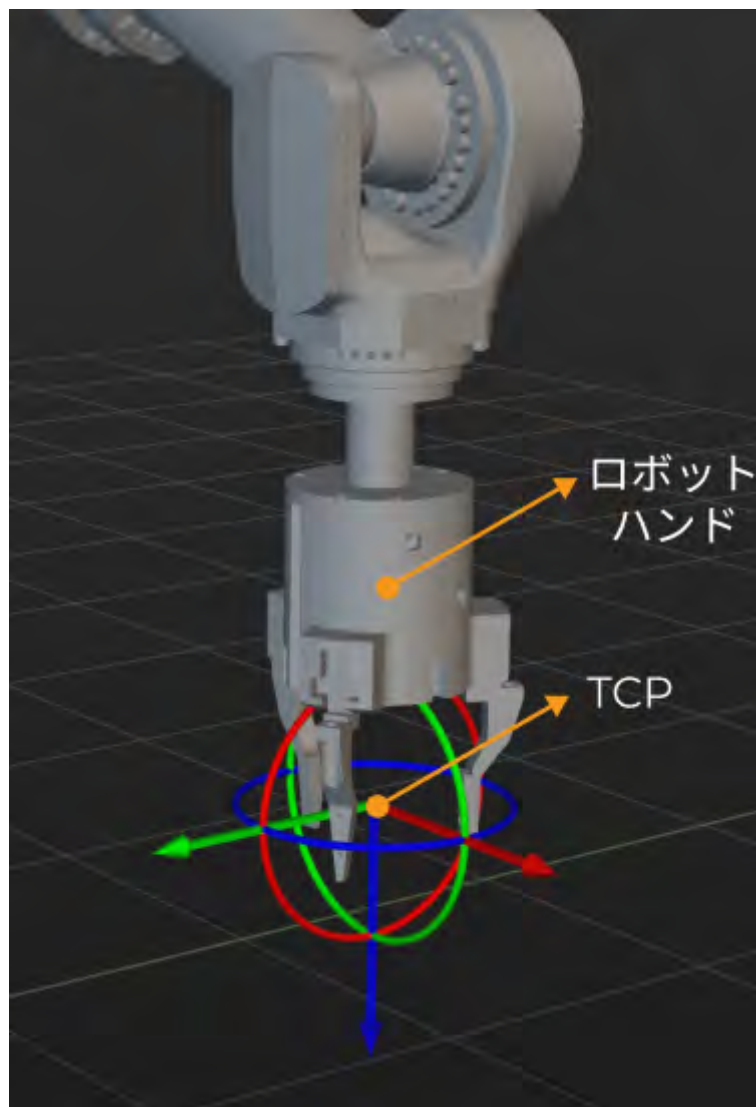
回転対称性無し	
2 回対称	

円対称（任意の角度回転させても自らと重なる）



ツール中心点（TCP）

ロボットハンドは対象物を把持・配置するために使用されます。把持・配置のためにロボットをある点に移動することは、本質的にはTCPをその点に移動することです。TCPは普通、ロボットのエンドにあるので、実際の把持・配置を再現するためにTCPをロボットハンドのエンドに調整する必要があります。



ロボットハンドモデルの設定

ロボットハンドの衝突モデルと 3D モデルをインポート

リソース ▶ モデルライブラリの[+]をクリックし、表示されたウィンドウで衝突モデルファイルと 3D モデルファイルを選択して[開く]をクリックします。

ロボットハンドの追加

リソース ▶ ロボットハンドの[+]をクリックし、ロボットハンド設定ウィンドウを開きます。

1. **ロボットハンド名** にカスタマイズの名前を入力します。
2. **ロボットハンドの種類** は、実際に応じて選択します。
3. **衝突モデル** を衝突検出のために使用されるロボットハンドモデルに設定します。
4. **3D モデル** を仮想空間に表示するために使用されるロボットハンドモデルに設定します。3D モデルの位置とサイズが実際と一致しない場合、**ロボットハンドモデルの位置またはサイズの調整**を参考して調整してください。
5. 実際に応じて **回転対称** を **回転対称性無し**、**N 回対称** または **円対称** に設定します。**N 回対称** に設定したら、**対称回数** を設定する必要があります。
6. 以下のいずれかの方法で TCP を設定します。

ロボットから TCP を更新

このボタンをクリックするとロボット実機の TCP の状態をソフトウェアに同期する

TCP キャリブレーション

ロボットをある点を中心に回転させて複数のフランジ位置姿勢を記録して TCP を計算する

TCP パラメータを変更

オイラー角や四元数のパラメータを変更する

正確な TCP 位置姿勢の値を使用


他の方法で正確な位置姿勢の値を取得した場合、それを位置姿勢調整ウィンドウに貼り付ける

7. 一般的なロボットハンドを使用するとき、制御ロジックを設定する必要はありません。吸盤または配列タイプグリッパを使用するとき、**ロボットハンド設定の制御ロジック**を参考して制御ロジックを完了させてください。
8. 最後に[OK]をクリックします。

必要があれば、以上の手順を繰り返してロボットハンドを追加します。

ロボットハンドの削除

以下のいずれかの手順を実行してロボットハンドを削除します。

- リソース ▶ ロボットハンドでロボットハンド名をクリックして **Delete** を押します。
- リソース ▶ ロボットハンドでロボットハンド名をクリックし、ドロップメニューで  をクリックします。

ロボットハンドの修正

1. 以下のいずれかの手順を実行してロボットハンド設定ウィンドウを開きます。
 - **リソース** > **ロボットハンド**でロボットハンド名をダブルクリックします。
 - **リソース** > **ロボットハンド**でロボットハンド名を右クリックし、ドロップメニューで[**ロボットハンド設定**]をクリックします。
2. 実際に応じて各パラメータ値を調整します。
3. [**OK**]をクリックします。

可動ロボットハンドの設定

ロボットハンドを一つだけ追加したとき、このロボットハンドは可動ロボットハンドです。複数のロボットハンドを追加したとき、デフォルトでは一番目のロボットハンドを可動ロボットハンドとします。

可動ロボットハンドを切り替える場合、**リソース** > **ロボットハンド**でロボットハンドを右クリックし、ドロップメニューで[**可動ロボットハンドに設定**]をクリックします。

ロボットハンドモデルの位置またはサイズの調整

追加したロボットハンドモデルが正しい位置とずれた場合、以下の手順を実行してください：

1. **リソース** > **モデルライブラリ**のロボットハンドモデルファイルをダブルクリックします。
2. 表示されたモデル変換編集ウィンドウで位置姿勢を変更してロボットハンドモデルが仮想空間にある位置を調整します。

追加したロボットハンドモデルのサイズが実際と一致しない場合以下の手順を実行してください：

1. **リソース** > **モデルライブラリ**のロボットハンドモデルファイルをダブルクリックします。
2. 表示されたモデル変換編集ウィンドウで、
 - モデルの全体のスケールを調整するには、**スケール** の値を変更します。
 - X、Y、Z 方向でそれぞれスケールを設定するには、**x,y,z を同じスケールにする** にチェックを外してから X、Y、Z 方向のスケールをそれぞれ設定します。

ロボットハンド設定の制御ロジック

一般的なロボットハンドを使用する場合、制御ロジックを設定する必要はありません。吸盤または配列タイプグリッパを使用する場合、実際に応じて制御ロジックを設定する必要があります。ロボットハンド設定ウィンドウで[**デパレタイズ用吸盤設定**]または[**配列タイプグリッパ設定**]をクリックして設定ウィンドウを開いて設定します。

デパレタイズ用吸盤 詳しくは[一般的なロボットハンド](#)をお読みください。

配列タイプグリッパ 詳しくは[配列タイプグリッパコンフィグレータ](#)をお読みください。

5.2.3. 対象物

本節では、対象物について説明します。

概要

対象物とは、段ボール箱や金属部品、接着剤塗布または溶接を実行する部品など、ロボットハンドが作用する物体です。

対象物の対称性

対称性とは、対象物がある軸で一定の角度回転したら元の形状と重なり合うことです。対象物の対称性を設定することでロボットハンドが把持や配置を実行する時の無駄な回転を回避し、時間を削減して把持計画の成功率を向上させることができます。

詳しくは、[対象物の回転対称性](#)をお読みください。

把持範囲

対象物を把持できない場合でも、ロボットハンドの位置姿勢を一定角度の範囲内で微調整することで把持できる場合もあります。この範囲を把持範囲といいます。

詳しくは、[対象物の把持範囲](#)をお読みください。

解の選択戦略

ロボットハンドの回転が最小

この戦略を選択すると、「把持－配置」の過程でロボットハンドのZ軸の回転が最小な把持位置姿勢を優先的に把持点として使用します。これにより、ロボットが把持を実行したあと、無駄な回転による対象物の落下を回避できます。

ロボットハンドと対象物の位置姿勢との差が最小

この戦略を選択すると、ロボットハンドと対象物の位置姿勢との偏差角度が最小な点を優先的に把持点として使用します。

ロボットハンドと点群の衝突が最小

この戦略を選択すると、ロボットハンドと対象物の点群の衝突体積が最小な点を優先的に把持点として使用します。

対象物設定

対象物を追加する

1. **リソース** > **対象物の[+]**をクリックして対象物設定ウィンドウを開きます。

2. **対象物名** に対象物の名前を入力します。
3. 対象物の特徴に応じて **回転対称** を **Z 軸を中心に**、**Y 軸を中心に**、**X 軸を中心に** を選択します（複数選択も可能）。その後、**対称回数** と **試行範囲** も設定します。対象物是对称性を持たない場合、このパラメータを設定する必要はありません。
4. 対象物の特徴に応じて **把持範囲** を **Y 軸を中心に** と **X 軸を中心に** を選択します（複数選択も可能）。その後、**試行範囲** と **試行間隔** も設定します。対象物の把持範囲がない場合、このパラメータを設定する必要はありません。
5. 把持戦略の要求を満たすように **解選択戦略** を設定します。
6. **[OK]** をクリックします。

複数の対象物を追加する場合、以上の手順を繰り返してください。

対象物を削除する

以下のいずれかの手順を実行して対象物を削除します。

- **リソース** > **対象物** の下に表示されている対象物名を選択し、**Delete** を押します。
- **リソース** > **対象物** の下に表示されている対象物名を右クリックし、表示されたドロップダウンメニューで **[削除]** をクリックします。

対象物を編集する

1. 以下のいずれかの手順を実行して対象物設定ウィンドウを開きます。
 - **リソース** > **対象物** の下に表示されている対象物名をダブルクリックします。
 - **リソース** > **対象物** の下に表示されている対象物名を右クリックし、表示されたドロップダウンメニューで **[対象物設定]** をクリックします。
2. 対象物設定ウィンドウでパラメータを調整します。
3. **[OK]** をクリックします。

5.2.3.1. 対象物の回転対称性

本節では、対象物の回転対称性について説明します。

概要



本節での「対称性」とは「回転対称性」を指します。

回転対称とは、2次元図形を一つの点を中心に回転させた場合に元の図形に完全に重なり合うことです。対称性がある対象物に対し、対象物の対称性パラメータを設定することができます。対象物の対称性を利用すると、ロボットが把持や配置を実行するときに発生するロボットハンドの回転を低減し、把持計画の成功率を向上させることが可能です。これによって、ロボットがさらにスムーズに動作します。

対称性のタイプ

回転対称軸を確認する

マッチングモデル・把持位置姿勢エディタで設定した幾何学的中心点の軸を回転対称軸とします。回転対称軸は一つだけではありません。対象物の配置方式やロボットハンドの種類などによって、Mech-Vision で設定された対象物の幾何学的中心点は異なるので対称軸も異なります。

以下では一般的な対象物の対称タイプを示します。対象物の対称タイプについては、マッチングモデル・把持位置姿勢エディタを参考して幾何学的中心点と把持点の設定を確認してください。

対称性を持たない対象物

下表に示す対象物は対称性を持たない対象物です。

			
クローラシュー	コネクタ	自動車ドア板金	ハウジング

Z 軸に関して対称性を持つ対象物

下表に示す対象物は Z 軸に関して対称性を持つ対象物です。

			
ブレーキディスク	フランジ	歯車	段ボール箱

X/Y 軸に関して対称性を持つ対象物





下表に示す対象物は X/Y 軸に関して対称性を持つ対象物です。

			
整列して並べられた 反射する鋼棒	ばら積み鋼棒	ホースコネクタ	ボルト

対称回数

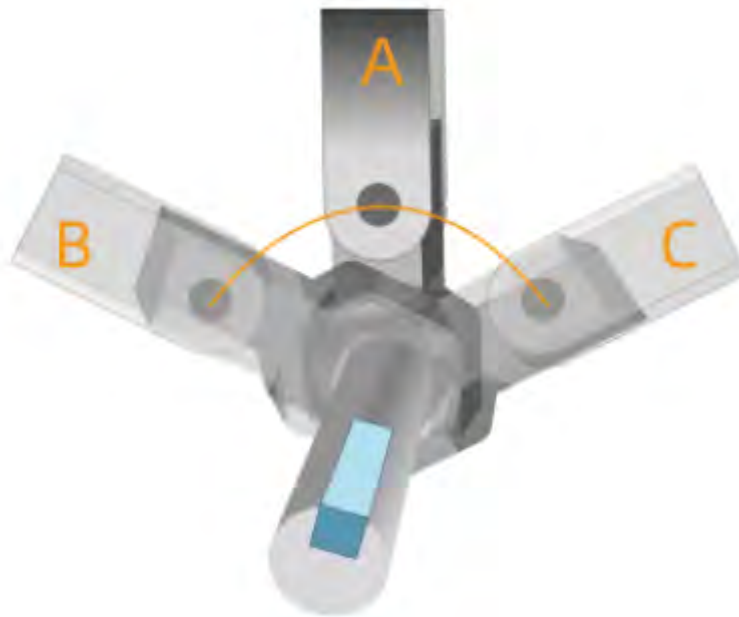
対象物を対称軸を中心に a° 回転させた後元の図形に完全に重なり合う時、対象物の対称回数は $N = 360^\circ / a^\circ$ です。

例えば、一般的な段ボール箱の対称回数（N）は2で、三角柱の対称回数（N）は3で、円柱の対称回数（N）は無限大（円対称）です。

			
N = 2	N = 4	N = 9	円対称

試行範囲

下図に示すように、BとCの間の角度が試行範囲となります。



Aはロボットハンドの把持位置姿勢で、BとCは把持を実行できる範囲の限界角度です。

試行範囲は対象物の配置方式やコンテナの形状、ロボットハンドの種類、タクトなどによって設定する必要があります。試行範囲を大きく設定すると計画の速度が遅くなりますが、小さく設定すると把持点に到達できなくなる可能性があります。

試行回数

対称回数と試行範囲によって自動的に計算されます。

対称回数 N は 10 とし、試行範囲は $\pm 80^\circ$ とすると、対称角度の間隔は $360^\circ/10 = 36^\circ$ となります。ある方向の試行回数は $80/36 = 2$ あまり 8 となり、つまり試行回数は 2 となります（それぞれ 36° と 72° の位置で試行します）。もう一方の方向も併せて試行回数総計は $1+2*2=5$ となります（ -72° 、 -36° 、 0° 、 36° 、 72° の位置で試行します）。

5.2.3.2. 対象物の把持範囲

本節では、対象物の把持範囲について説明します。

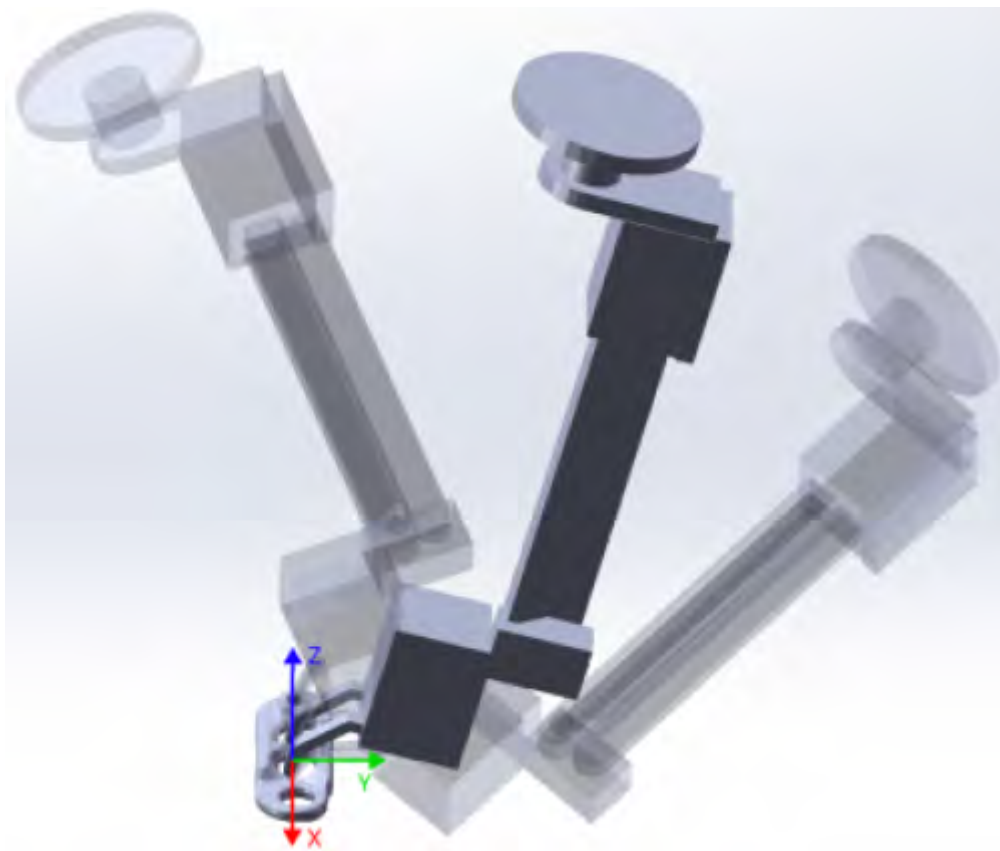
概要

一部分の対象物に対しては、ロボットの把持位置姿勢を一定の範囲で微調整することができます。この調整できる範囲を把持範囲といいます。把持範囲を設定するとロボットが動作するときに衝突や特異点を回避することが可能になります。

配置位置姿勢を「対象物の位置姿勢」に設定した場合、把持範囲を利用して把持を実行するとき、ソフトウェアでは実際の把持位置姿勢と配置位置姿勢との角度が記録されます。配置を実行するとき、ソフトウェアはロボットハンドが正しい姿勢で対象物を配置するように、この角度によってロボットハンドの姿勢を調整します。

基準軸を確認する

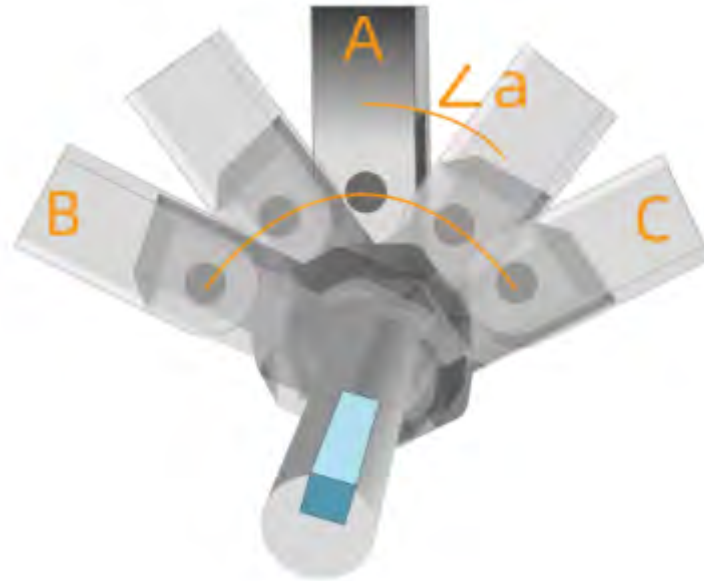
下図に示すようなシーンでは、ロボットハンドは対象物の位置姿勢の X 軸を中心に一定の角度範囲内で把持を実行することができます。この場合、X 軸が基準軸となります。



試行範囲と試行間隔

試行範囲は把持位置姿勢とビジョン位置姿勢の偏差角度の最大許容範囲です（下図では、B と C の間の角度が試行範囲となります）。

試行間隔は把持を試行する間隔です（下図では $\angle a$ は試行間隔となります）。



A はロボットハンドの把持位置姿勢で、B と C は把持を実行できる範囲の限界角度です。

試行回数

試行範囲と試行間隔によって自動的に計算されます。

試行間隔は 5° とし、試行範囲は $\pm 10^\circ$ とすると、ある方向の試行回数は $10/5 = 2$ となり、つまり試行回数は 2 となります（それぞれ 5° と 10° の位置で試行します）。もう一方の方向も併せて試行回数総計は $1+2*2=5$ となります（ -10° 、 -5° 、 0° 、 5° 、 10° の位置で試行します）。

5.2.4. 地面

実際の地面をシミュレートします。デフォルトでは地面はロボットのベースと同じ平面にあります。実際のシーンの地面の高さに応じて地面の高さを調整する必要があります。

地面の高さを調整する

リソース、**地面** を右クリックして、表示されたボックスにスライダーをドラッグするか、数値を編集して地面の高さを調整する

地面のパターンを変更する

設定、**オプション**、**詳細設定**、**地面のパターン** で選択する

5.2.5. シーンの物体

本節ではシーンの物体設定について紹介します。以下の内容について説明します。

概要

シーンの物体とは、ロボット実機の動作環境の物体を指し、普通はロボット安全柵や箱、パレット、カメラ、カメラブラケットなどが含まれます。

シーンの物体モデル

シーンの物体モデルはソフトウェアで実際の動作環境を再現し、衝突検出や経路計画を行うために使用されます。ソフトウェアでは、直方体、円柱、箱のモデルを作成でき、また外部モデルをインポートして使用することもできます。外部モデルの3Dモデルと衝突モデルが対応する形式は下表のとおりです。

形式	STL	OBJ	DAE	Binvox
3Dモデル	√	√	√	×
衝突モデル	√	√	×	×

シーンの物体の「親子関係」

シーンの物体同士は、親子関係を作ることができます。親モデルを移動すると子モデルもともに動きますが、子モデルを移動すると親モデルは動きません。親モデルを削除すると子モデルもともに削除されます。

シーンの物体の設定

リソースパネルでシーンの物体モデルを追加、編集、削除することができます。

シーンの物体を追加する

直方体や円柱、箱をシーンの物体として使用するとき、以下の手順を実行します：

1. **リソース** > **シーンの物体**の[+]をクリックし、シーンの物体設定ウィンドウを開きます。
2. 物体設定画面で、実際の状況に応じて**シーンモデル**パラメータを設定します。
 - **直方体** モデルの寸法 **X**、**Y**、**Z** を設定します。
 - **円柱** モデルの寸法 **底面の半径**、**高さ** を設定します。
 - **箱** モデルの寸法 **X**、**Y**、**Z**、**厚さ** を設定します。また、**箱のビジョン位置姿勢の有効範囲を設定する**必要もあります。
3. **物体名**にカスタマイズのシーンの物体モデルの名前を入力します。



- 今追加したシーンの物体モデルを衝突検出に使用しない場合、**衝突検出に使用する**のチェックを外します。
- 今追加したシーンの物体モデルを仮想空間でマウスで選択不能にする場合、**モデル選択可能**のチェックを外します。

4. 追加したシーンの物体モデルはデフォルトでロボットベース座標系の原点のあります。**[物体の位置姿勢]**をクリックして物体の位置姿勢に関するパラメータを設定します。
5. **[OK]**をクリックします。

外部モデルを使用するとき、以下の手順を実行します：

1. **リソース** ▶ **モデルライブラリ**の[+]をクリックし、表示されたウィンドウで衝突モデルファイルと 3D モデルファイルを選択して[**開く**]をクリックします。
2. **リソース** ▶ **シーンの物体**の[+]をクリックし、シーンの物体設定ウィンドウを開きます。
3. 物体設定画面で**シーンモデル**に**外部モデル**を選択します。
4. **衝突モデル**に衝突検出に衝突検出に使用するシーンの物体モデルを選択します。
5. **3D モデル**に仮想空間に表示するシーンの物体モデルを選択します。
6. **物体名**にカスタマイズのシーンの物体名を入力します。



- 今追加したシーンの物体を衝突検出に使用しない場合、**衝突検出に使用する**のチェックを外します。
- 今追加したシーンの物体を仮想空間でマウスで選択不能にする場合、**モデル選択可能**のチェックを外します。

7. 追加したシーンの物体モデルはデフォルトでロボットベース座標系の原点のあります。[**物体の位置姿勢**]をクリックして物体の位置姿勢に関するパラメータを設定します。
8. [**OK**]をクリックします。



「外部モデル」の「衝突モデル」が STL 形式の場合、「直方体」または「円柱」を作成して「外部モデル」を囲むようにしてください。

必要があれば以上の手順を繰り返して複数のモデルを追加します。



- リソースパネルでモデルを選択してから[+]をクリックして選択したモデルの子モデルを追加します。デフォルトでは、子モデルはその親モデルとは位置姿勢が同じです。
- リソースパネルではモデルをドラッグしてほかのモデルの子モデルにすることと独立したモデルにすることができます。

シーンの物体を削除する

以下のいずれかの手順を実行してシーンの物体を削除します。

- 仮想空間でシーンの物体モデルを選択するか、**リソース** ▶ **シーンの物体**でシーンの物体モデル名を選択してから[**Delete**]を押します。
- 仮想空間でシーンの物体モデルを右クリックするか、**リソース** ▶ **シーンの物体**で物体名を右クリックしてドロップダウンメニューで[**削除**]をクリックします。

シーンの物体を編集する

1. 以下のいずれかの手順を実行してシーンの物体を編集します。
 - 仮想空間の物体モデルをダブルクリックします。
 - **リソース** ▶ **シーンの物体**の物体名をダブルクリックします。
 - **リソース** ▶ **シーンの物体**の物体名を右クリックし、表示されたドロップダウンメニューで[**シーンの物体設定**]をクリックします。

2. 実際の状況に応じて各パラメータを設定します。

3. [OK]をクリックします。



仮想空間でシーンの物体モデルをクリックして選択し、ドラッガーをドラッグして物体モデルの位置姿勢を調整できます。

箱のビジョン位置姿勢の有効範囲を設定する

箱モデルを追加する場合、**箱の設定**画面で位置姿勢の有効範囲を設定する必要があります。

- 箱がただシーンの物体として使用し、位置姿勢の有効範囲を設定する必要がない場合、**位置姿勢の有効範囲を設定**のチェックを外します。
- 位置姿勢を箱の中に設定する場合、**箱内**を選択します。
- 位置姿勢を箱の上部に設定してもいい場合に、**箱内・外**を選択して**箱の上を超えた高さ範囲**パラメータを設定します。
- 位置姿勢の有効範囲をカスタマイズするとき、**カスタマイズ**を選択して**位置姿勢の有効範囲を設定**を設定します。

5.3. ワークフローの構築

本節では、ワークフローの構築について説明していきます。

概要

ワークフローの構築は、プロジェクトを実行するためのステップを配置して接続することです。また、各ステップのパラメータを調整してロボットの動作を指定します。

Mech-Viz には、様々な機能を実現する「ステップ」（ロボットプログラミング機能ユニット）が組み込まれており、ロボットの動作制御やビジョンシステムとの併用、ビジョンによる認識結果の処理、DI/DO 制御、論理トポロジー、ロボットの把持・配置、通信などを実現できます。

機能と操作

機能	操作
ステップを検索	ステップライブラリの検索ボックスにキーワードを入力するか、分類から検索する
ステップを追加	マウスの左ボタンを押したままステップをワークフロー編集エリアにドラッグしてから放す
ステップを削除	ステップをクリックしてからキーボードの Delete を押すか、右クリックしてドロップメニューの [削除] をクリックする

機能	操作
ステップを接続	カーソルをステップの出口に合わせてから左ボタンを押したままもう一つのステップにドラッグして放す
接続を削除	ステップ間の線をクリックしてキーボードの Delete を押すか、右クリックしてドロップメニューの [削除] をクリックする
ステップの説明をカスタマイズ	ステップを右クリックしてドロップメニューで [ディスクリプションをカスタマイズ] をクリックする

ワークフローの例

1. 仮想ロボットを動かしてみよう
2. ロボット実機を動かしてみよう

5.3.1. ワークフローの構築

本節では、ワークフローの構築について説明していきます。

概要

ワークフローの構築は、プロジェクトを実行するためのステップを配置して接続することです。また、各ステップのパラメータを調整してロボットの動作を指定します。

Mech-Viz には、様々な機能を実現する「ステップ」（ロボットプログラミング機能ユニット）が組み込まれており、ロボットの動作制御やビジョンシステムとの併用、ビジョンによる認識結果の処理、DI/DO 制御、論理トポロジー、ロボットの把持・配置、通信などを実現できます。

機能と操作

機能	操作
ステップを検索	ステップライブラリの検索ボックスにキーワードを入力するか、分類から検索する
ステップを追加	マウスの左ボタンを押したままステップをワークフロー編集エリアにドラッグしてから放す
ステップを削除	ステップをクリックしてからキーボードの Delete を押すか、右クリックしてドロップメニューの [削除] をクリックする
ステップを接続	カーソルをステップの出口に合わせてから左ボタンを押したままもう一つのステップにドラッグして放す

機能	操作
接続を削除	ステップ間の線をクリックしてキーボードの Delete を押すか、右クリックしてドロップメニューの [削除] をクリックする
ステップの説明をカスタマイズ	ステップを右クリックしてドロップメニューで [ディスクリプションをカスタマイズ] をクリックする

ワークフローの例

1. 仮想ロボットを動かしてみよう
2. ロボット実機を動かしてみよう

5.3.2. 仮想ロボットを動かしてみよう

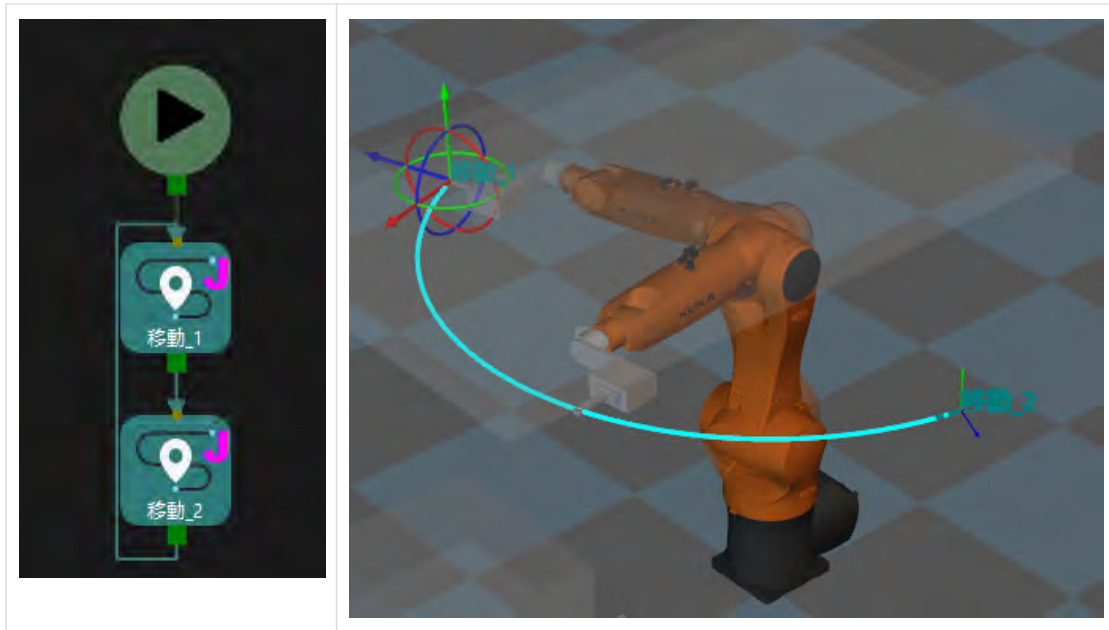
まず、簡単な経路を作って仮想ロボットを動かしてみましょう！「経路」は「経路点」によって指定されるので、ロボットの動作経路を作成するために「経路点」をいくつか設定すればいいです。

- 詳しく説明します。

▶ <https://docs.mech-mind.net/download/Mech-Viz/run-simulate-robot-ja.mp4> (video)

- 手順
 1. Mech-Viz を起動し、プロジェクトを新規作成します。ロボットをいずれか選択してください。
 2. ステップライブラリから「移動」ステップを編集エリアにドラッグします。
 3. この「移動」ステップを選択して右のパラメータパレットを見てください。下の「関節角度」は、今の仮想ロボットの各関節の角度数値が表示されます。いずれかの数値を変更すれば仮想ロボットの姿勢が変わります。変更した位置姿勢は経路点として使用できます。
 4. 「スタート」と「移動」ステップを接続し、**[シミュレート]** をクリックします。すると仮想ロボットが前の手順に変更した位置姿勢に移動します。
 5. 続けて経路点を設定してみましょう。二つ目の「移動」ステップをドラッグして関節角度の数値を変更します。「移動_1」の出口と「移動_2」の入口を接続してから**[シミュレート]** をクリックします。すると仮想ロボットが前の経路点から新しい経路点に移動します。
 6. 「移動_2」の出口と「移動_1」の入口を接続してから**[シミュレート]** をクリックします。すると仮想ロボットがこの2点の間を行き来します。**[停止]** をクリックすると仮想ロボットの動作が止まります。

ワークフロー	シミュレーション
--------	----------



これでロボット経路計画の第一歩を踏み出しました！

5.3.3. ロボット実機を動かしてみよう

仮想ロボットと異なり、ロボット実機を動かすには、まずソフトウェアとロボット間の通信を構築する必要があります。これにより、ロボットがソフトウェアから指令を受信して指令どおりに動作します。ここでソフトウェアティーチングの方式を例にして説明します。詳しくは [Vizティーチング通信](#) を参照してロボット通信の設定を行ってください。

通信を構築したあと、ロボット経路点を設定します。「移動_1」ステップを編集エリアにドラッグしてパラメータパレットで「関節角度」を変更します（**安全のために大幅な変更をしないでください**）。変更した位置姿勢はロボット実機の経路点になります。最後に「スタート」と「移動_1」ステップを接続します。



- ロボット実機を実行する前に、必ずロボットの動作速度を下げてください。ツールバーの「速度」と「速度」を 5% に設定してください
- ロボット実機を実行する前に、必ずロボットの動作速度を下げてください。ツールバーの「速度」と「速度」を 5% に設定してください

[実行]をクリックするとロボット実機が「移動_1」の位置姿勢に移動します。

これで Mech-Viz でロボットを移動させることができました！

5.4. ステップライブラリ

ステップについて詳しくは以下の内容をお読みください。

移動ステップ	動的移動
	移動
	グリッドによる移動
	リストによる移動
	外部移動
	相対移動
DI DO	DI をチェック
	DI リストをチェック
	DO を設定
	DO リストを設定
	DI を待つ
論理トポロジ ー	道標によって異なる分岐を実行
	メッセージによって異なる分岐を実行
	ステップの組合せ
	ステップの組合せの出口
	道標を設定
パレタイジ ングステップ	カスタマイズのパレットパターン
	混載パレタイジング
	事前計画パレットパターン
	ビジョン処理による継続パレタイジング
	複数把持のパレタイジング

ロボットハンドツール	ロボットハンドをチェック
	最大可搬質量設定
	ロボットハンドを切り替え
	把持状態を設定
	関節角度を取得
	制御を移転
サービス	通知
ツール	分類
	カウンター
	完了確認
	インデックスを変更
	ステップをリセット
	待つ
経路	経路ステップの組合せ
ビジョン	ビジョン処理の結果をチェック
	パレットの位置姿勢を更新
	把持済み対象物を更新
	シーンの物体を更新
	ビジョン結果を使い切る
	ビジョン処理による認識
	ビジョン処理による移動
ほか	実行を中止

5.4.1. 移動ステップ

5.4.1.1. 動的移動

機能

前回の「ビジョン処理による認識」で最も高い位置姿勢に基づいて移動し、カメラを認識対象

物から適切な距離に移動します。

使用シーン

EIH 方式に使用されます。

パラメータ

移動ステップの共通パラメータ

移動目標点を送信

デフォルトではチェックが入っています。相手側（ロボットなど）に移動目標点の位置姿勢を送信します。チェックを外したら送信しませんが、この目標点位置姿勢は依然として経路計画の一部です。

移動コマンドの後の非移動コマンドをスムーズに実行することを試行

デフォルトではチェックが外れており、**移動ステップ**の間に「ビジョン処理による認識」、「DO を設定」、「DI をチェック」などの**非移動ステップ**を接続すると、ロボット移動経路の計画を中断し、ロボット実機動作中に一時中止することがあります。チェックを入れると現在のステップが完了しなくても続行することが可能です。これにより、ロボットがよりスムーズに動作します。ただし、ステップが途中で終了する可能性があります。

ステップが途中で終了する原因は何ですか？



Mech-Viz 実行中、ロボットに同時に複数の位置姿勢を送信しますが、最後の位置姿勢が現在のロボットの関節角度と一致しているかのみを判断します。一致していると判断したら、ロボットが最後の位置に到達していると考えられます。例えば、10 の移動ステップがある経路では、移動ステップ 5 の位置姿勢は最後のステップの位置姿勢とは同じとします。ロボットが低速に動作し、移動ステップ 5 の位置に到達したらその位置の関節角度を Mech-Viz に送信します。経路では、移動ステップ 5 の位置姿勢は最後の移動ステップの位置姿勢とは同じなので、Mech-Viz はロボットの動作が完了したと判断して途中で実行を終了します。

配置された対象物との衝突を検出しない

デフォルトではチェックが外れており、配置された対象物との衝突を検出しません。チェックを入れると、ロボット本体・ロボットハンドと配置された対象物との衝突を検出します。

パレタイジングのシーンでは、以下場合があります。

1. ロボットが段ボール箱を配置するときに配置済みの箱と軽く衝突する（箱の凹みや変形などが発生しない）ことがあります。Mech-Viz ではこのような衝突を検出したら別の配置位置を計画するためパレットが満杯になりません。
2. 普通、吸盤を使用する場合、TCP を吸盤の表面でなく、モデル内部に設定するため、物体を吸着する時、吸盤と把持する箱のモデルと重なります（ソフトウェアでは吸盤と把持対象物との衝突を検出しない）。ロボットが箱を配置してから、把持された箱のモデルはシーンの

モデルになり、吸盤がシーンの箱のモデルと衝突すると判断してメッセージを表示してパレタイジングが続行できなくなります。

これをチェックするとロボット本体やロボットハンドと配置済み対象物モデルとの衝突を検出しないので上記の問題を解決できます。

点群との衝突検出モード

現場の状況に応じて設定してください。普通、デフォルトの **自動** を使用します。ロボットが物体を把持する前の移動ステップは、**チェックしない** を選択し、把持した後の移動ステップを **チェック** を選択します。

自動 初期値。「ビジョン処理による移動」ステップと「ビジョン処理による移動」ステップに依存する「相対移動」の点群衝突だけを検出します。

チェックしない 移動ステップの点群衝突をいっさい検出しません。

チェック 全ての移動ステップの点群衝突を検出します。



衝突検出、**衝突検出設定**、**点群と他の対象物間の衝突を検出** をオンにすれば、Mech-Viz は経路計画を行うときにロボットモデル、ツールモデルと点群との衝突を検出します。普通、点群衝突の設定は、ロボットが把持を実行するときに把持対象物との衝突を検出するためです。空間内にノイズがあれば、ソフトウェアは対象物を把持する前の経路を計画するときに、ノイズがロボットモデル、ツールモデルと接触して点群との衝突が誤って判断されて経路計画の誤りが発生します。

対象物の対称性を使用しない

このパラメータは、**目標点タイプ** が **対象物位置姿勢** の移動ステップ（目標点タイプが物体の位置姿勢の移動ステップ、パレタイジングステップなど）に対してのみ有効です。目標点のタイプが関節角度、TCP位置姿勢の移動ステップには無効です。

無し 無し：初期値。対称性を無効にしません。

Z 軸 Z 軸の対称性のみを使用しません。

XY 軸 X、Y 軸の対称性のみを使用しません。

全て 全ての対称性を使用しません。

物体の対称性を使用しないように設定すると、ロボットが物体の位置姿勢に正確に到達して対象物を配置します。



対象物を把持できない場合に **リソース**、**対象物設定** の **回転対称** を設定します。対象物に対称性がある場合、複数の候補位置姿勢があります。Mech-Viz では対象物の把持を計画するときに、デフォルト位置姿勢が把持できない場合、候補位置姿勢を試行します。物体対称位置姿勢と Mech-Vision によって出力された元位置姿勢とは一致しなければ、ロボットによる対象物の配置位置姿勢は一致しないことがあります。

把持された対象物との衝突検出モード

シーンの物体との衝突を検出しない

デフォルトではチェックが外れています。チェックすると把持された対象物とシーンの物体、ロボットとの衝突を検出せず、ソフトウェアの計画の速度を改善できます。普通、ロボットが対象物を把持したあとの移動ステップに使用します。

衝突を検出しなければ、衝突が発生するリスクがあるので注意して使用してください。



衝突検出 > **把持されている対象物の設定** > **把持されている対象物とその他の物体との衝突を検出** をオンにすれば、把持されている対象物とシーンの物体のモデル・ロボットとの衝突を検出します。パレタイジングのシーンでは、ビジョン処理により取得した箱の寸法にはささやかな誤差があり、把持を実行する時に箱同士の摩擦が発生しますが衝突は発生しません。

パレタイズのシーンでは、「シーンの物体との衝突を検出しない」をチェックしても対象物と配置済みの箱との衝突検出に干渉しません。箱の山の下にシーンの物体がある場合、経路計画のためにこれをチェックしてください。

点群との衝突を検出しない

デフォルトではチェックが外れています。チェックすると **把持対象物のモデル** と **シーンの点群** が衝突するかを検出しないので、Mech-Viz での衝突検出計算量を低減し、計画の効率を向上させることができます。また、把持対象物と点群との衝突の誤検出も回避できます。



- **衝突検出** > **把持されている対象物の設定** > **把持されている対象物とその他の物体との衝突を検出** と **点群設定** > **点群と他の対象物間の衝突を検出** をオンにすれば、把持されている対象物とシーンの物体のモデル・ロボットとの衝突を検出します。
- Mech-Vision が点群情報と対象物モデル情報を Mech-Viz に送信するときに、点群と対象物モデルは密着している状態です。ロボットが対象物を把持するときに、モデルはロボットの経路に沿って移動し、把持対象物モデルは点群と衝突します。
- 把持対象物は物体の点群と衝突する場合、この衝突をチェックすると Mech-Viz の計算量が増え、計画時間が長くなります。

Z 方向オフセット（経路点が視野内の最も高い対象物に対する）

初期値 0

調整説明 EIH 方式では、対象物把持を実行している過程でカメラの高さを固定してロボットと対象物との距離を固定します。設定したら、ロボットが「動的移動」を実行するたびに前回の [ビジョン処理による認識](#) の最高位置姿勢の Z 値に基づいて動的移動の Z 値を調整します。（経路点 Z 方向の座標 = Z 方向オフセット + 最高対象物位置姿勢の Z 方向の座標）

ロボットが到達できる Z 方向の許容最小値

初期値 -10 m

調整説明 ロボットが到達できる実際の Z 方向の許容最小値です。

基本的な移動設定

基本的な移動設定は、移動ステップの主なパラメータを設定し、ロボットが目標点に到達する**速度**と**運動方式**を決定します。

把持と配置設定

未指定 初期値

把持 「ビジョン処理による移動」の前の移動ステップ。

放置 「ビジョン処理による移動」の後の移動ステップ。

調整説明 Mech-Viz プロジェクトのロジック検査に使用されます。現場の実行手順に応じて、把持をしてから配置するという基本的な原則に従って設定します。

運動タイプ

関節運動 ロボットが円弧に沿って走行します。スムーズに走行できるので特異点を回避することが可能。ロボットの移動範囲が広くて高い精度を求めないシーンに適用しています。

直線運動 ロボットが直線に沿って走行します。溶接や接着剤塗布、把持など高い精度が求められるシーンに適用しています。

速度&加速度

速度&加速度は、ロボット動作のスピードを決定します。普通、加速度の値を速度より小さく設定します。加速度の値を速度よりも大きくしたらロボット動作がスムーズでなくなります。



ロボットが安定に把持を実行するように、「ビジョン処理による移動」とその前後のステップの速度を低く設定してください。

ブレンド半径

初期値 50.00mm

調整説明 調整説明：通常、初期値を使用すればいいです。

ブレンド半径とは、回転する位置が目標点までの距離であり、大きいほどロボットの動きはスムーズになります。ロボットが狭いスペースで動作する場合、ブレンド半径を大きく設定する必要はありません。

ロボットがより広いスペースで動作し、障害物がなくて 2つの経路の間に距離が遠い場合、ロボットがスムーズに動くように、ブレンド半径を大きく設定することができます。

関節角度の制約条件


概念：

- 肩** 腕関節の中心と軸 1 との相対関係。軸1とは、ロボット1軸の回転中心軸です。
- 肘** 腕関節と前腕との相対関係。前腕とはロボットの 2、3 軸の回転中心軸です。
- 手首** ロボット 5 軸です。5 軸角度の正負は、腕の回転を表します。



オプション：

- 自動** 関節を制約しません。各軸の回転が最小の位置姿勢を最適解とします。
- 保持** ロボットの現在の状態を状態Aとしてロボットを制約します。Mech-Viz プロジェクトでは、次の移動の解を選択するとき、状態 A と一致する解を有効なものとしてします。ロボット 3 軸を例に、現在の状態では 3 軸は正方向に向いているので移動の解を求めるときに 3 軸だけが正方向の解を有効とします。
- その前** 腕関節中心が軸 1 の前にあります。
- その後** 腕関節中心が軸 1 の後ろにあります。

 をクリックするとこの位置姿勢が対応するすべての関節角度が表示されます。いずれかをクリックしてシミュレーションエリアでロボットの姿勢を表示でき、異なる制約条件の下でその他の可能な関節角度を確認することができます。



1. 関節角度の制約条件は 6 軸ロボットにのみ有効です。4 軸ロボットは、肩や肘、手首の回転はないとされます。
2. ソフトウェアでは、この機能はカスタマイズパレットパターンと事前計画したパレットパターンに対応できません。デフォルトでは「自動」に設定し、肩、肘、手首を変えないのでロボットが動作中に特異点を経過しません。

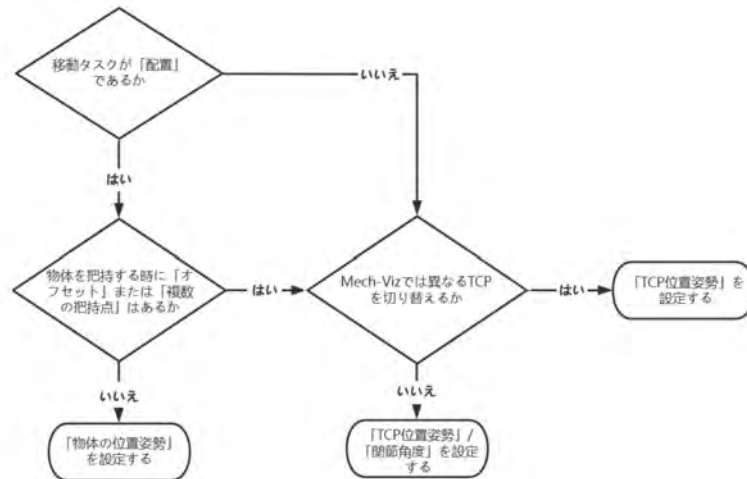
目標点タイプ

- TCP** TCP 座標系の X、Y、Z 値、およびオイラー角または四元数によって目標点を表示します。

関節角度 ロボットの各関節の数値で目標点を表示します。

対象物位置姿勢 対象物座標系の X、Y、Z 値、およびオイラー角または四元数によって目標点を表示します。

下図のように目標点タイプを選択します。

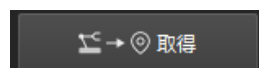


位置姿勢を編集 位置姿勢を編集します。コピーと貼り付けができ、四元数とオイラー角の二つの形式に対応しています。

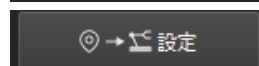
位置姿勢を変換 変換を定義することで現在の位置姿勢を新しい位置姿勢に変換します。

位置姿勢を校正 ABB ロボット三点法と類似しており、対象物の座標系を計算します。対象物の回転位置姿勢を確認できない場合に使用します。例えば、傾斜した直方体に対し、位置姿勢を校正することでその位置姿勢を計算し、ロボットが計算された位置姿勢に沿って動作させることが可能です。

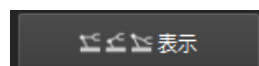
関節角度を編集 位置姿勢を調整する方法と同じく、コピーと貼り付けができます。ラジアンと角度の二つの形に対応可能です。



仮想ロボットを指定した経路点に移動します。



ロボットを移動する目標位置姿勢を読み取り、経路点に設定します。



可能な関節角度の解を全部表示します。

5.4.1.2. 移動

機能

ロボット経路にある目標位置姿勢とこの位置姿勢に到達する方式を設定します。

パラメータ

移動ステップの共通パラメータ

移動目標点を送信

デフォルトではチェックが入っています。相手側（ロボットなど）に移動目標点の位置姿勢を送信します。チェックを外したら送信しませんが、この目標点位置姿勢は依然として経路計画の一部です。

移動コマンドの後の非移動コマンドをスムーズに実行することを試行

デフォルトではチェックが外れており、**移動ステップ**の間に「ビジョン処理による認識」、「DOを設定」、「DIをチェック」などの**非移動ステップ**を接続すると、ロボット移動経路の計画を中断し、ロボット実機動作中に一時中止することがあります。チェックを入れると現在のステップが完了しなくても続行することが可能です。これにより、ロボットがよりスムーズに動作します。ただし、ステップが途中で終了する可能性があります。

ステップが途中で終了する原因は何ですか？



Mech-Viz 実行中、ロボットに同時に複数の位置姿勢を送信しますが、最後の位置姿勢が現在のロボットの関節角度と一致しているかのみを判断します。一致していると判断したら、ロボットが最後の位置に到達していると考えられます。例えば、10の移動ステップがある経路では、移動ステップ5の位置姿勢は最後のステップの位置姿勢と同じとします。ロボットが低速に動作し、移動ステップ5の位置に到達したらその位置の関節角度をMech-Vizに送信します。経路では、移動ステップ5の位置姿勢は最後の移動ステップの位置姿勢と同じなので、Mech-Vizはロボットの動作が完了したと判断して途中で実行を終了します。

配置された対象物との衝突を検出しない

デフォルトではチェックが外れており、配置された対象物との衝突を検出しません。チェックを入れると、ロボット本体・ロボットハンドと配置された対象物との衝突を検出します。

パレタイジングのシーンでは、以下場合があります。

1. ロボットが段ボール箱を配置するときに配置済みの箱と軽く衝突する（箱の凹みや変形などが発生しない）ことがあります。Mech-Vizではこのような衝突を検出したら別の配置位置を計画するためパレットが満杯になりません。
2. 普通、吸盤を使用する場合、TCPを吸盤の表面でなく、モデル内部に設定するため、物体を吸着する時、吸盤と把持する箱のモデルと重なります（ソフトウェアでは吸盤と把持対象物との衝突を検出しない）。ロボットが箱を配置してから、把持された箱のモデルはシーンのモデルになり、吸盤がシーンの箱のモデルと衝突すると判断してメッセージを表示してパレタイジングが続行できなくなります。

これをチェックするとロボット本体やロボットハンドと配置済み対象物モデルとの衝突を検出しないので上記の問題を解決できます。

点群との衝突検出モード

現場の状況に応じて設定してください。普通、デフォルトの **自動** を使用します。ロボットが物体を把持する前の移動ステップは、**チェックしない** を選択し、把持した後の移動ステップを **チェック** を選択します。

自動 初期値。「ビジョン処理による移動」ステップと「ビジョン処理による移動」ステップに依存する「相対移動」の点群衝突だけを検出します。

チェックしない 移動ステップの点群衝突をいっさい検出しません。

チェック 全ての移動ステップの点群衝突を検出します。



衝突検出、**衝突検出設定**、**点群と他の対象物間の衝突を検出** をオンにすれば、Mech-Viz は経路計画を行うときにロボットモデル、ツールモデルと点群との衝突を検出します。普通、点群衝突の設定は、ロボットが把持を実行するときに把持対象物との衝突を検出するためです。空間内にノイズがあれば、ソフトウェアは対象物を把持する前の経路を計画するときに、ノイズがロボットモデル、ツールモデルと接触して点群との衝突が誤って判断されて経路計画の誤りが発生します。

対象物の対称性を使用しない

このパラメータは、**目標点タイプ** が **対象物位置姿勢** の移動ステップ（目標点タイプが物体の位置姿勢の移動ステップ、パレタイジングステップなど）に対してのみ有効です。目標点のタイプが関節角度、TCP位置姿勢の移動ステップには無効です。

無し 無し：初期値。対称性を無効にしません。

Z 軸 Z 軸の対称性のみを使用しません。

XY 軸 X、Y 軸の対称性のみを使用しません。

全て 全ての対称性を使用しません。

物体の対称性を使用しないように設定すると、ロボットが物体の位置姿勢に正確に到達して対象物を配置します。



対象物を把持できない場合に **リソース**、**対象物設定** の **回転対称** を設定します。対象物に対称性がある場合、複数の候補位置姿勢があります。Mech-Viz では対象物の把持を計画するときに、デフォルト位置姿勢が把持できない場合、候補位置姿勢を試行します。物体対称位置姿勢と Mech-Vision によって出力された元位置姿勢とは一致しなければ、ロボットによる対象物の配置位置姿勢は一致しないことがあります。

把持された対象物との衝突検出モード

シーンの物体との衝突を検出しない

デフォルトではチェックが外れています。チェックすると把持された対象物とシーンの物体、ロボットとの衝突を検出せず、ソフトウェアの計画の速度を改善できます。普通、ロボットが対象物を把持したあとの移動ステップに使用します。

衝突を検出しなければ、衝突が発生するリスクがあるので注意して使用してください。



衝突検出 > **把持されている対象物の設定** > **把持されている対象物とその他の物体との衝突を検出** をオンにすれば、把持されている対象物とシーンの物体のモデル・ロボットとの衝突を検出します。パレタイジングのシーンでは、ビジョン処理により取得した箱の寸法にはささやかな誤差があり、把持を実行する時に箱同士の摩擦が発生しますが衝突は発生しません。

パレタイズのシーンでは、「シーンの物体との衝突を検出しない」をチェックしても対象物と配置済みの箱との衝突検出に干渉しません。箱の山の下にシーンの物体がある場合、経路計画のためにこれをチェックしてください。

点群との衝突を検出しない

デフォルトではチェックが外れています。チェックすると **把持対象物のモデル** と **シーンの点群** が衝突するかを検出しないので、Mech-Viz での衝突検出計算量を低減し、計画の効率を向上させることができます。また、把持対象物と点群との衝突の誤検出も回避できます。



- **衝突検出** > **把持されている対象物の設定** > **把持されている対象物とその他の物体との衝突を検出** と **点群設定** > **点群と他の対象物間の衝突を検出** をオンにすれば、把持されている対象物とシーンの物体のモデル・ロボットとの衝突を検出します。
- Mech-Vision が点群情報と対象物モデル情報を Mech-Viz に送信するときに、点群と対象物モデルは密着している状態です。ロボットが対象物を把持するときに、モデルはロボットの経路に沿って移動し、把持対象物モデルは点群と衝突します。
- 把持対象物は物体の点群と衝突する場合、この衝突をチェックすると Mech-Viz の計算量が増え、計画時間が長くなります。

基本的な移動設定

基本的な移動設定は、移動ステップの主なパラメータを設定し、ロボットが目標点に到達する **速度** と **運動方式** を決定します。

把持と配置設定

未指定	初期値
把持	「ビジョン処理による移動」の前の移動ステップ。
放置	「ビジョン処理による移動」の後の移動ステップ。
調整説明	Mech-Viz プロジェクトのロジック検査に使用されます。現場の実行手順に応じて、把持をしてから配置するという基本的な原則に従って設定します。

運動タイプ

- | | |
|-------------|---|
| 関節運動 | ロボットが円弧に沿って走行します。スムーズに走行できるので特異点を回避することが可能。ロボットの移動範囲が広くて高い精度を求めないシーンに適用しています。 |
| 直線運動 | ロボットが直線に沿って走行します。溶接や接着剤塗布、把持など高い精度が求められるシーンに適用しています。 |

速度&加速度

速度&加速度は、ロボット動作のスピードを決定します。普通、加速度の値を速度より小さく設定します。加速度の値を速度よりも大きくしたらロボット動作がスムーズでなくなります。



ロボットが安定に把持を実行するように、「ビジョン処理による移動」とその前後のステップの速度を低く設定してください。

ブレンド半径

初期値 50.00mm

調整説明 調整説明：通常、初期値を使用すればいいです。

ブレンド半径とは、回転する位置が目標点までの距離であり、大きいほどロボットの動きはスムーズになります。ロボットが狭いスペースで動作する場合、ブレンド半径を大きく設定する必要はありません。

ロボットがより広いスペースで動作し、障害物がなくて2つの経路の間に距離が遠い場合、ロボットがスムーズに動くように、ブレンド半径を大きく設定することができます。

自動障害物回避

ロボット経路計画の成功率を向上させるため、移動ステップ関連の経路点の近くに複数の候補点を設定します。

障害物回避のモード

経路計画の速度と成功率の要件によって **障害物回避のモード** を選択します。

- | | |
|----------------------|--------------------------------|
| 自動障害物回避を無効にする | 自動障害物回避機能を使用しません。 |
| 速度優先 | 経路を計画するとき速度を優先します。候補点が最も少ないです。 |
| 成功率優先 | 経路を計画するとき成功率を優先します。候補点が最も多いです。 |

バランス

速度と成功率のバランスを取ります。候補点が多くも少なくもないです。

位置調整の範囲半径

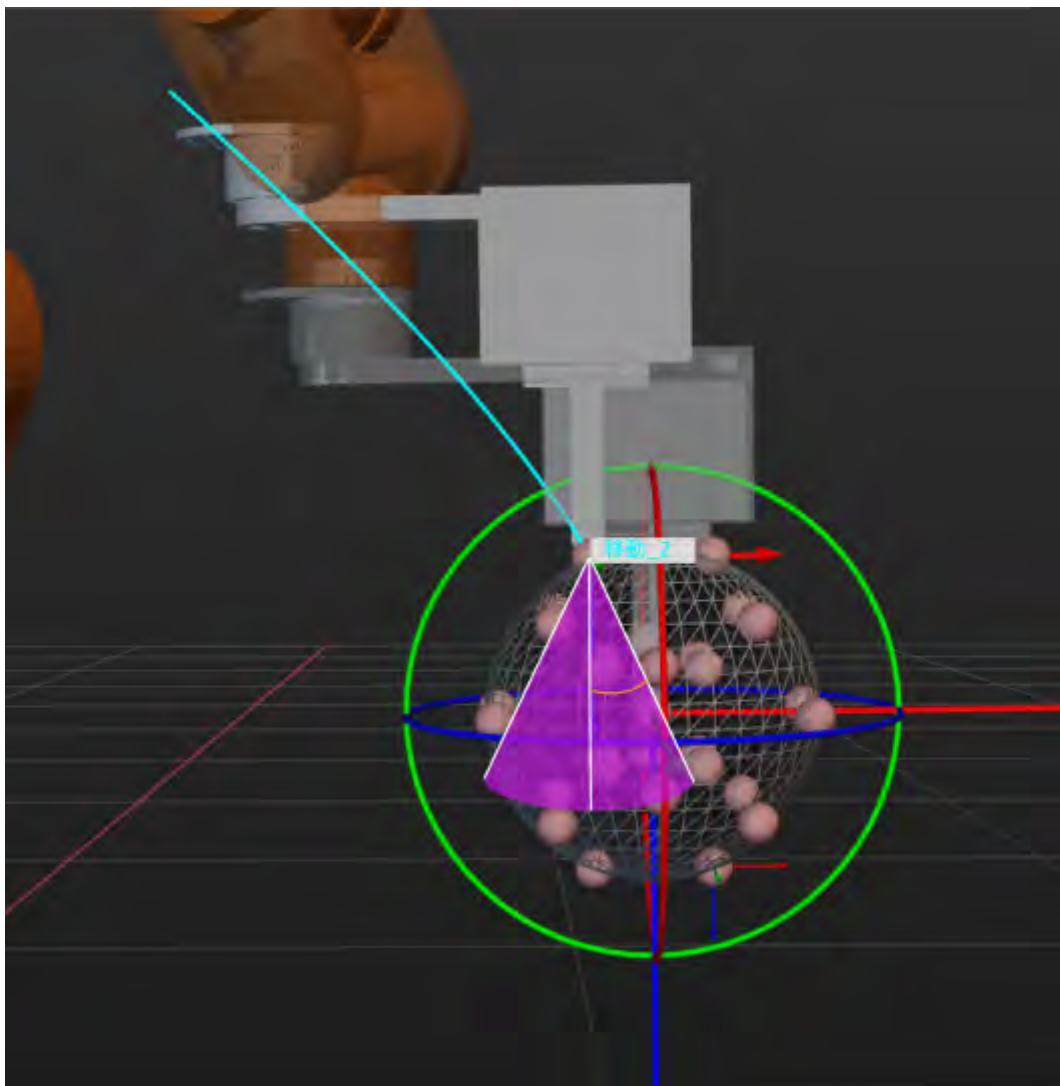
初期値：0.005

候補点はこの値を半径とする球体の内部に生成されます。

角度調整の範囲

初期値：0、調整の範囲：0 - 79

候補点に位置する TCP の角度の調整範囲。

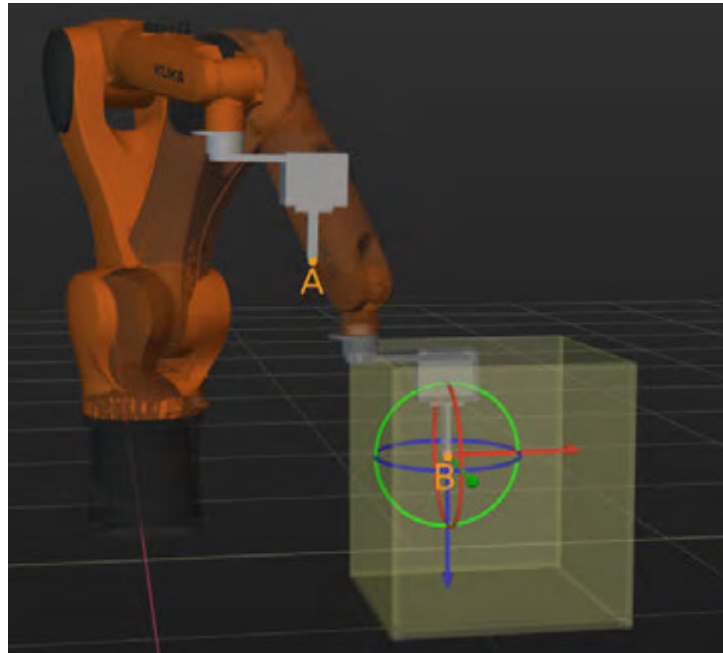


すべての候補位置姿勢を表示

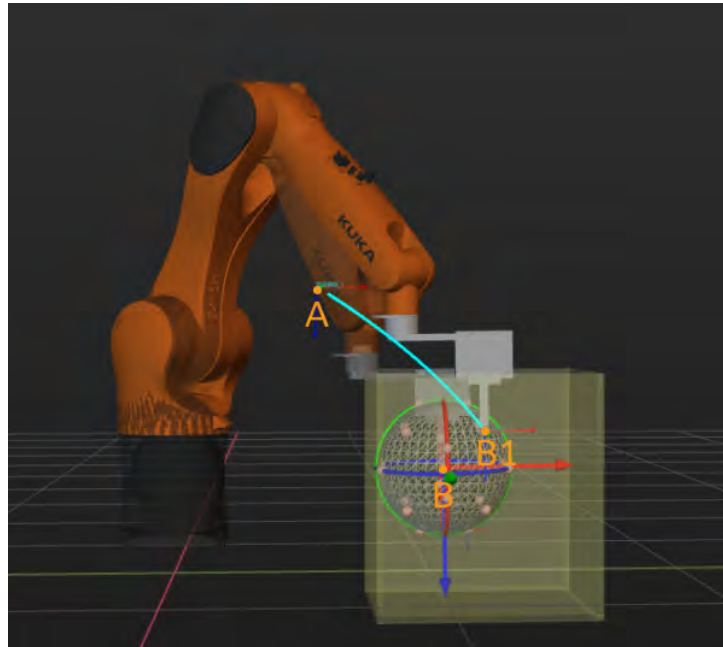
デフォルトではチェックが入っており、すべての候補点が表示されます。チェックを外すと表示されません。

例

下図に示すように、経路点 A から B まで移動すると、ロボットハンドは箱と衝突します。



経路点 B に障害物回避モードをオンにしてパラメータを設定すると、下図に示すように、候補点の中に衝突が発生しない B1 へ移動します。



5.4.1.3. グリッドによる移動

機能

ロボット経路にグリッドを形成する複数の **目標位置姿勢** とこの位置姿勢に到達する **運動方式** を設定し、順序に従って各点に移動します。

パラメータ

移動ステップの共通パラメータ

移動目標点を送信

デフォルトではチェックが入っています。相手側（ロボットなど）に移動目標点の位置姿勢を送信します。チェックを外したら送信しませんが、この目標点位置姿勢は依然として経路計画の一部です。

移動コマンドの後の非移動コマンドをスムーズに実行することを試行

デフォルトではチェックが外れており、**移動ステップ**の間に「ビジョン処理による認識」、「DOを設定」、「DIをチェック」などの**非移動ステップ**を接続すると、ロボット移動経路の計画を中断し、ロボット実機動作中に一時中止することがあります。チェックを入れると現在のステップが完了しなくても続行することが可能です。これにより、ロボットがよりスムーズに動作します。ただし、ステップが途中で終了する可能性があります。



ステップが途中で終了する原因は何ですか？

Mech-Viz 実行中、ロボットに同時に複数の位置姿勢を送信しますが、最後の位置姿勢が現在のロボットの関節角度と一致しているかのみを判断します。一致していると判断したら、ロボットが最後の位置に到達していると考えられます。例えば、10の移動ステップがある経路では、移動ステップ5の位置姿勢は最後のステップの位置姿勢とは同じとします。ロボットが低速に動作し、移動ステップ5の位置に到達したらその位置の関節角度をMech-Vizに送信します。経路では、移動ステップ5の位置姿勢は最後の移動ステップの位置姿勢とは同じなので、Mech-Vizはロボットの動作が完了したと判断して途中で実行を終了します。

配置された対象物との衝突を検出しない

デフォルトではチェックが外れており、配置された対象物との衝突を検出しません。チェックを入れると、ロボット本体・ロボットハンドと配置された対象物との衝突を検出します。

パレタイジングのシーンでは、以下場合があります。

1. ロボットが段ボール箱を配置するときに配置済みの箱と軽く衝突する（箱の凹みや変形などが発生しない）ことがあります。Mech-Vizではこのような衝突を検出したら別の配置位置を計画するためパレットが満杯になりません。
2. 普通、吸盤を使用する場合、TCPを吸盤の表面でなく、モデル内部に設定するため、物体を吸着する時、吸盤と把持する箱のモデルと重なります（ソフトウェアでは吸盤と把持対象物との衝突を検出しない）。ロボットが箱を配置してから、把持された箱のモデルはシーンのモデルになり、吸盤がシーンの箱のモデルと衝突すると判断してメッセージを表示してパレタイジングが続行できなくなります。

これをチェックするとロボット本体やロボットハンドと配置済み対象物モデルとの衝突を検出しないので上記の問題を解決できます。

点群との衝突検出モード

現場の状況に応じて設定してください。普通、デフォルトの **自動** を使用します。ロボットが物体を把持する前の移動ステップは、**チェックしない** を選択し、把持した後の移動ステップを **チェック** を選択します。

- 自動** 初期値。「ビジョン処理による移動」ステップと「ビジョン処理による移動」ステップに依存する「相対移動」の点群衝突だけを検出します。
- チェックしない** 移動ステップの点群衝突をいっさい検出しません。
- チェック** 全ての移動ステップの点群衝突を検出します。



衝突検出、**衝突検出設定**、**点群と他の対象物間の衝突を検出** をオンにすれば、Mech-Viz は経路計画を行うときにロボットモデル、ツールモデルと点群との衝突を検出します。普通、点群衝突の設定は、ロボットが把持を実行するときに把持対象物との衝突を検出するためです。空間内にノイズがあれば、ソフトウェアは対象物を把持する前の経路を計画するときに、ノイズがロボットモデル、ツールモデルと接触して点群との衝突が誤って判断されて経路計画の誤りが発生します。

対象物の対称性を使用しない

このパラメータは、**目標点タイプ** が **対象物位置姿勢** の移動ステップ（目標点タイプが物体の位置姿勢の移動ステップ、パレタイジングステップなど）に対してのみ有効です。目標点のタイプが関節角度、TCP位置姿勢の移動ステップには無効です。

- 無し** 無し：初期値。対称性を無効にしません。
- Z 軸** Z 軸の対称性のみを使用しません。
- XY 軸** X、Y 軸の対称性のみを使用しません。
- 全て** 全ての対称性を使用しません。

物体の対称性を使用しないように設定すると、ロボットが物体の位置姿勢に正確に到達して対象物を配置します。



対象物を把持できない場合に **リソース**、**対象物設定** の **回転対称** を設定します。対象物に対称性がある場合、複数の候補位置姿勢があります。Mech-Viz では対象物の把持を計画するときに、デフォルト位置姿勢が把持できない場合、候補位置姿勢を試行します。物体対称位置姿勢と Mech-Vision によって出力された元位置姿勢とは一致しなければ、ロボットによる対象物の配置位置姿勢は一致しないことがあります。

把持された対象物との衝突検出モード

シーンの物体との衝突を検出しない

デフォルトではチェックが外れています。チェックすると把持された対象物とシーンの物体、ロボットとの衝突を検出せず、ソフトウェアの計画の速度を改善できます。普通、ロボットが

対象物を把持したあとの移動ステップに使用します。

衝突を検出しなければ、衝突が発生するリスクがあるので注意して使用してください。



衝突検出 > **把持されている対象物の設定** > **把持されている対象物とその他の物体との衝突を検出** をオンにすれば、把持されている対象物とシーンの物体のモデル・ロボットとの衝突を検出します。パレタイジングのシーンでは、ビジョン処理により取得した箱の寸法にはささやかな誤差があり、把持を実行する時に箱同士の摩擦が発生しますが衝突は発生しません。

パレタイズのシーンでは、「シーンの物体との衝突を検出しない」をチェックしても対象物と配置済みの箱との衝突検出に干渉しません。箱の山の下にシーンの物体がある場合、経路計画のためにこれをチェックしてください。

点群との衝突を検出しない

デフォルトではチェックが外れています。チェックすると **把持対象物のモデル** と **シーンの点群** が衝突するかを検出しないので、Mech-Viz での衝突検出計算量を低減し、計画の効率を向上させることができます。また、把持対象物と点群との衝突の誤検出も回避できます。



- **衝突検出** > **把持されている対象物の設定** > **把持されている対象物とその他の物体との衝突を検出** と **点群設定** > **点群と他の対象物間の衝突を検出** をオンにすれば、把持されている対象物とシーンの物体のモデル・ロボットとの衝突を検出します。
- Mech-Vision が点群情報と対象物モデル情報を Mech-Viz に送信するときに、点群と対象物モデルは密着している状態です。ロボットが対象物を把持するときに、モデルはロボットの経路に沿って移動し、把持対象物モデルは点群と衝突します。
- 把持対象物は物体の点群と衝突する場合、この衝突をチェックすると Mech-Viz の計算量が増え、計画時間が長くなります。



- 普通、ロボットが対象物を把持した後の 1 つまたは 2 つの移動ステップに **衝突を検出しない** を選択します。選択すると衝突が発生する恐れがあります。
- パレタイズのシーンでは、**シーンの物体との衝突を検出しない** をチェックしても対象物と配置済みの箱との衝突検出に干渉しません。箱の山の下にシーンの物体がある場合、解選択に成功するためにこれをチェックしてください。

インデックス

開始インデックス

今到達しようとする位置姿勢のインデックス。例えば、一番目のインデックスは 0 となります。グリッドの三番目の位置姿勢から移動するように設定したい場合、2 に設定してください。

現在のインデックス

今移動を実行している点のインデックス。

グリッド設定

グリッドの方向にある点の数の間隔を設定します。

X 方向の点の数

グリッドの X 方向にある点の数。

X 方向の点の間隔 グリッドの X 方向にある隣接する点の間隔。

Y 方向の点の数 グリッドの Y 方向にある点の数。

Y 方向の点の間隔 グリッドの Y 方向にある隣接する点の間隔。

基本的な移動設定

基本的な移動設定は、移動ステップの主なパラメータを設定し、ロボットが目標点に到達する**速度**と**運動方式**を決定します。

把持と配置設定

未指定 初期値

把持 「ビジョン処理による移動」の前の移動ステップ。

放置 「ビジョン処理による移動」の後の移動ステップ。

調整説明 Mech-Viz プロジェクトのロジック検査に使用されます。現場の実行手順に応じて、把持をしてから配置するという基本的な原則に従って設定します。

運動タイプ

関節運動 ロボットが円弧に沿って走行します。スムーズに走行できるので特異点を回避することが可能。ロボットの移動範囲が広くて高い精度を求めないシーンに適用しています。

直線運動 ロボットが直線に沿って走行します。溶接や接着剤塗布、把持など高い精度が求められるシーンに適用しています。

速度&加速度

速度&加速度は、ロボット動作のスピードを決定します。普通、加速度の値を速度より小さく設定します。加速度の値を速度よりも大きくしたらロボット動作がスムーズでなくなります。



ロボットが安定に把持を実行するように、「ビジョン処理による移動」とその前後のステップの速度を低く設定してください。

ブレンド半径

初期値 50.00mm

調整説明 調整説明：通常、初期値を使用すればいいです。

ブレンド半径とは、回転する位置が目標点までの距離であり、大きいほどロボットの動きはスムーズになります。ロボットが狭いスペースで動作する場合、ブレンド半径を大きく設定する必要はありません。

ロボットがより広いスペースで動作し、障害物がなくて2つの経路の間に距離が遠い場合、ロボットがスムーズに動くように、ブレンド半径を大きく設定することができます。

目標点のグリッドを表示

チェックすると Mech-Viz 仮想空間にグリッドの点とインデックスが表示されます。

ロボットを表示

チェックすると Mech-Viz 仮想空間に **グリッドによる移動** の一番目の点に到達するロボットの位置姿勢または対象物位置姿勢が表示されます。

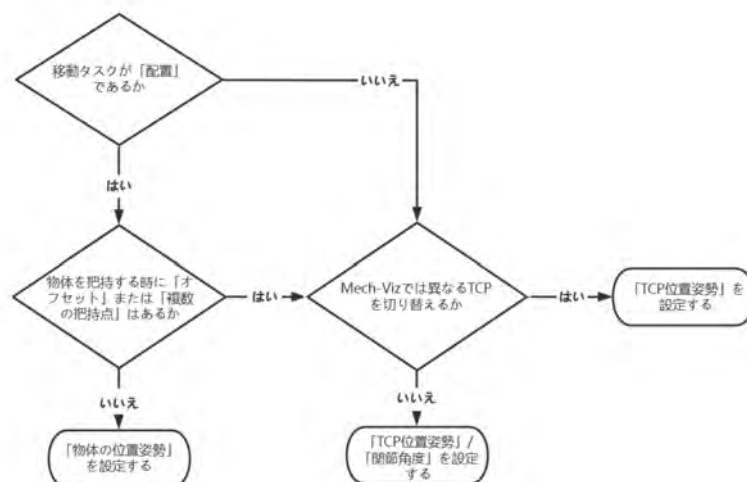
目標点タイプ

TCP TCP 座標系の X、Y、Z 値、およびオイラー角または四元数によって目標点を表示します。

関節角度 ロボットの各関節の数値で目標点を表示します。

対象物位置姿勢 対象物座標系の X、Y、Z 値、およびオイラー角または四元数によって目標点を表示します。

下図のように目標点タイプを選択します。



位置姿勢を編集

位置姿勢を編集します。コピーと貼り付けができ、四元数とオイラー角の二つの形式に対応しています。

位置姿勢を変換

変換を定義することで現在の位置姿勢を新しい位置姿勢に変換します。

位置姿勢を校正

ABB ロボット三点法と類似しており、対象物の座標系を計算します。対象物の回転位置姿勢を確認できない場合に使用します。例えば、傾斜した直方体に対し、位置姿勢を校正することでその位置姿勢を計算し、ロボットが計算された位置姿勢に沿って動作させることが可能です。

関節角度を編集

位置姿勢を調整する方法と同じく、コピーと貼り付けができます。ラジアンと角度の二つの形に対応可能です。



仮想ロボットを指定した経路点に移動します。



ロボットを移動する目標位置姿勢を読み取り、経路点に設定します。



可能な関節角度の解を全部表示します。

関節角度の制約条件

概念：

肩 腕関節の中心と軸 1 との相対関係。軸1とは、ロボット1軸の回転中心軸です。

肘 腕関節と前腕との相対関係。前腕とはロボットの 2、3 軸の回転中心軸です。

手首 ロボット 5 軸です。5 軸角度の正負は、腕の回転を表します。




オプション：

自動 関節を制約しません。各軸の回転が最小の位置姿勢を最適解とします。

保持 ロボットの現在の状態を状態Aとしてロボットを制約します。Mech-Viz プロジェクトでは、次の移動の解を選択するとき、状態Aと一致する解を有効なものとし
ます。ロボット3軸を例に、現在の状態では3軸は正方向に向いているので移動
の解を求めるときに3軸だけが正方向の解を有効とします。

その前 腕関節中心が軸1の前にあります。

その後 腕関節中心が軸1の後ろにあります。

 をクリックするとこの位置姿勢が対応するすべての関節角度が表示さ
れます。いずれかをクリックしてシミュレーションエリアでロボットの姿勢を表示でき、異な
る制約条件の下でその他の可能な関節角度を確認することができます。



1. 関節角度の制約条件は6軸ロボットにのみ有効です。4軸ロボットは、肩や肘、手首の
回転はないとされます。
2. ソフトウェアでは、この機能はカスタマイズパレットパターンと事前計画したパレ
ットパターンに対応できません。デフォルトでは「自動」に設定し、肩、肘、手首を変
えないのでロボットが動作中に特異点を経過しません。

5.4.1.4. リストによる移動

機能

ロボット経路に複数の **目標位置姿勢** とこれらの位置姿勢に到達する **運動方式** を設定し、順序に
従って各点に移動します。

パラメータ

移動ステップの共通パラメータ

移動目標点を送信

デフォルトではチェックが入っています。相手側（ロボットなど）に移動目標点の位置姿勢を
送信します。チェックを外したら送信しませんが、この目標点位置姿勢は依然として経路計画
の一部です。

移動コマンドの後の非移動コマンドをスムーズに実行することを試行

デフォルトではチェックが外れており、**移動ステップ** の間に「ビジョン処理による認
識」、「DOを設定」、「DIをチェック」などの **非移動ステップ** を接続すると、ロボット移動
経路の計画を中断し、ロボット実機動作中に一時中止することがあります。チェックを入れる
と現在のステップが完了しなくても続行することが可能です。これにより、ロボットがよりス
ムーズに動作します。ただし、ステップが途中で終了する可能性があります。



ステップが途中で終了する原因は何ですか？

Mech-Viz 実行中、ロボットに同時に複数の位置姿勢を送信しますが、最後の位置姿勢が現

在のロボットの関節角度と一致しているかのみを判断します。一致していると判断したら、ロボットが最後の位置に到達していると思なします。例えば、10の移動ステップがある経路では、移動ステップ5の位置姿勢は最後のステップの位置姿勢とは同じとします。ロボットが低速に動作し、移動ステップ5の位置に到達したらその位置の関節角度を Mech-Viz に送信します。経路では、移動ステップ5の位置姿勢は最後の移動ステップの位置姿勢とは同じなので、Mech-Viz はロボットの動作が完了したと判断して途中で実行を終了します。

配置された対象物との衝突を検出しない

デフォルトではチェックが外れており、配置された対象物との衝突を検出しません。チェックを入れると、ロボット本体・ロボットハンドと配置された対象物との衝突を検出します。

パレタイジングのシーンでは、以下場合があります。

1. ロボットが段ボール箱を配置するときに配置済みの箱と軽く衝突する（箱の凹みや変形などが発生しない）ことがあります。Mech-Viz ではこのような衝突を検出したら別の配置位置を計画するためパレットが満杯になりません。
2. 普通、吸盤を使用する場合、TCP を吸盤の表面でなく、モデル内部に設定するため、物体を吸着する時、吸盤と把持する箱のモデルと重なります（ソフトウェアでは吸盤と把持対象物との衝突を検出しない）。ロボットが箱を配置してから、把持された箱のモデルはシーンのモデルになり、吸盤がシーンの箱のモデルと衝突すると判断してメッセージを表示してパレタイジングが続行できなくなります。

これをチェックするとロボット本体やロボットハンドと配置済み対象物モデルとの衝突を検出しないので上記の問題を解決できます。

点群との衝突検出モード

現場の状況に応じて設定してください。普通、デフォルトの **自動** を使用します。ロボットが物体を把持する前の移動ステップは、**チェックしない** を選択し、把持した後の移動ステップを **チェック** を選択します。

自動 初期値。「ビジョン処理による移動」ステップと「ビジョン処理による移動」ステップに依存する「相対移動」の点群衝突だけを検出します。

チェックしない 移動ステップの点群衝突をいっさい検出しません。

チェック 全ての移動ステップの点群衝突を検出します。



衝突検出、**衝突検出設定**、**点群と他の対象物間の衝突を検出** をオンにすれば、Mech-Viz は経路計画を行うときにロボットモデル、ツールモデルと点群との衝突を検出します。普通、点群衝突の設定は、ロボットが把持を実行するときに把持対象物との衝突を検出するためです。空間内にノイズがあれば、ソフトウェアは対象物を把持する前の経路を計画するときに、ノイズがロボットモデル、ツールモデルと接触して点群との衝突が誤って判断されて経路計画の誤りが発生します。

対象物の対称性を使用しない

このパラメータは、**目標点タイプ**が**対象物位置姿勢**の移動ステップ（目標点タイプが物体の位置姿勢の移動ステップ、パレタイジングステップなど）に対してのみ有効です。目標点のタイプが関節角度、TCP位置姿勢の移動ステップには無効です。

- 無し** 無し：初期値。対称性を無効にしません。
- Z 軸** Z 軸の対称性のみを使用しません。
- XY 軸** X、Y 軸の対称性のみを使用しません。
- 全て** 全ての対称性を使用しません。

物体の対称性を使用しないように設定すると、ロボットが物体の位置姿勢に正確に到達して対象物を配置します。



対象物を把持できない場合に **リソース > 対象物設定** の **回転対称** を設定します。対象物に対称性がある場合、複数の候補位置姿勢があります。Mech-Viz では対象物の把持を計画するときに、デフォルト位置姿勢が把持できない場合、候補位置姿勢を試行します。物体対称位置姿勢と Mech-Vision によって出力された元位置姿勢とは一致しなければ、ロボットによる対象物の配置位置姿勢は一致しないことがあります。

把持された対象物との衝突検出モード

シーンの物体との衝突を検出しない

デフォルトではチェックが外れています。チェックすると把持された対象物とシーンの物体、ロボットとの衝突を検出せず、ソフトウェアの計画の速度を改善できます。普通、ロボットが対象物を把持したあとの移動ステップに使用します。

衝突を検出しなければ、衝突が発生するリスクがあるので注意して使用してください。



衝突検出 > 把持されている対象物の設定 > 把持されている対象物とその他の物体との衝突を検出 をオンにすれば、把持されている対象物とシーンの物体のモデル・ロボットとの衝突を検出します。パレタイジングのシーンでは、ビジョン処理により取得した箱の寸法にはささやかな誤差があり、把持を実行する時に箱同士の摩擦が発生しますが衝突は発生しません。

パレタイズのシーンでは、「シーンの物体との衝突を検出しない」をチェックしても対象物と配置済みの箱との衝突検出に干渉しません。箱の山の下にシーンの物体がある場合、経路計画のためにこれをチェックしてください。

点群との衝突を検出しない

デフォルトではチェックが外れています。チェックすると **把持対象物のモデル** と **シーンの点群** が衝突するかを検出しないので、Mech-Viz での衝突検出計算量を低減し、計画の効率を向上させることができます。また、把持対象物と点群との衝突の誤検出も回避できます。



● **衝突検出 > 把持されている対象物の設定 > 把持されている対象物とその他の物体との**



衝突を検出と点群設定、**点群と他の対象物間の衝突を検出**をオンにすれば、把持されている対象物とシーンの物体のモデル・ロボットとの衝突を検出します。

- Mech-Vision が点群情報と対象物モデル情報を Mech-Viz に送信するときに、点群と対象物モデルは密着している状態です。ロボットが対象物を把持するときに、モデルはロボットの経路に沿って移動し、把持対象物モデルは点群と衝突します。
- 把持対象物は物体の点群と衝突する場合、この衝突をチェックすると Mech-Viz の計算量が増え、計画時間が長くなります。

普通、ロボットが対象物を把持した後の1つまたは2つの移動ステップに**衝突を検出しない**を選択します。選択すると衝突が発生する恐れがあります。パレタイズのシーンでは、**シーンの物体との衝突を検出しない**をチェックしても対象物と配置済みの箱との衝突検出に干渉しません。箱の山の下にシーンの物体がある場合、解選択に成功するためにこれチェックしてください。

インデックス

開始インデックス

今到達しようとする位置姿勢のインデックス。

例えば、一番目のインデックスは0となります。グリッドの三番目の位置姿勢から移動するように設定したい場合、2に設定してください。

現在のインデックス

今移動を実行している点のインデックス。

一度に全ての目標点に移動

チェックすると、**インデックス**は機能しません。また、Mech-Viz プロジェクトの**グリッドによる移動**を実行するとき、ロボットは一度に順序に従って移動リストのすべての点に移動します。

全体に基本的な移動設定を適用

チェックすると、移動ステップリストのすべてのステップに**基本的な移動設定**を適用します。

基本的な移動設定

基本的な移動設定は、移動ステップの主なパラメータを設定し、ロボットが目標点に到達する**速度**と**運動方式**を決定します。

把持と配置設定

未指定 初期値

把持 「ビジョン処理による移動」の前の移動ステップ。

放置 「ビジョン処理による移動」の後の移動ステップ。

調整説明 Mech-Viz プロジェクトのロジック検査に使用されます。現場の実行手順に応じて、把持をしてから配置するという基本的な原則に従って設定します。

運動タイプ

関節運動 ロボットが円弧に沿って走行します。スムーズに走行できるので特異点を回避することが可能。ロボットの移動範囲が広くて高い精度を求めないシーンに適用しています。

直線運動 ロボットが直線に沿って走行します。溶接や接着剤塗布、把持など高い精度が求められるシーンに適用しています。

速度&加速度

速度&加速度は、ロボット動作のスピードを決定します。普通、加速度の値を速度より小さく設定します。加速度の値を速度よりも大きくしたらロボット動作がスムーズでなくなります。



ロボットが安定に把持を実行するように、「ビジョン処理による移動」とその前後のステップの速度を低く設定してください。

ブレンド半径

初期値 50.00mm

調整説明 調整説明：通常、初期値を使用すればいいです。

ブレンド半径とは、回転する位置が目標点までの距離であり、大きいほどロボットの動きはスムーズになります。ロボットが狭いスペースで動作する場合、ブレンド半径を大きく設定する必要はありません。

ロボットがより広いスペースで動作し、障害物がなくて2つの経路の間に距離が遠い場合、ロボットがスムーズに動くように、ブレンド半径を大きく設定することができます。

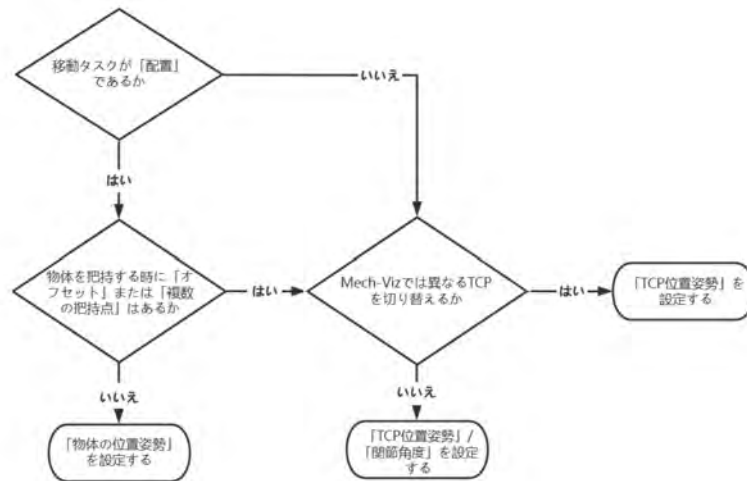
目標点タイプ

TCP TCP 座標系の X、Y、Z 値、およびオイラー角または四元数によって目標点を表示します。

関節角度 ロボットの各関節の数値で目標点を表示します。

対象物位置姿勢 対象物座標系の X、Y、Z 値、およびオイラー角または四元数によって目標点を表示します。

下図のように目標点タイプを選択します。



位置姿勢を編集

位置姿勢を編集します。コピーと貼り付けができ、四元数とオイラー角の二つの形式に対応しています。

位置姿勢を変換

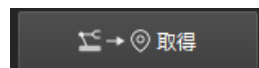
変換を定義することで現在の位置姿勢を新しい位置姿勢に変換します。

位置姿勢を校正

ABB ロボット三点法と類似しており、対象物の座標系を計算します。対象物の回転位置姿勢を確認できない場合に使用します。例えば、傾斜した直方体に対し、位置姿勢を校正することでその位置姿勢を計算し、ロボットが計算された位置姿勢に沿って動作させることが可能です。

関節角度を編集

位置姿勢を調整する方法と同じく、コピーと貼り付けができます。ラジアンと角度の二つの形に対応可能です。



仮想ロボットを指定した経路点に移動します。



ロボットを移動する目標位置姿勢を読み取り、経路点に設定します。



可能な関節角度の解を全部表示します。

移動リスト

手動で順序に従って移動目標点を追加し、追加するたびに移動パラメータを設定します。

`+` をクリックすると移動リストに移動目標点を追加します。`-` をクリックすると選択した移動目標点を削除します。

TCP 位置姿勢をリストに追加

一度に複数の移動目標点の位置姿勢を **TCP 位置姿勢** としてリストに追加します。

表示されたウィンドウで追加する位置姿勢を行ごとに入力します。

5.4.1.5. 外部移動

機能

外部サービスから取得する目標位置姿勢へ移動します。Mech-Viz ソフトウェアは外部サービスから移動する目標位置姿勢を取得し、ロボットをこの位置姿勢に移動させます。Adapter を使用する必要があります。

パラメータ

移動ステップの共通パラメータ

移動目標点を送信

デフォルトではチェックが入っています。相手側（ロボットなど）に移動目標点の位置姿勢を送信します。チェックを外したら送信しませんが、この目標点位置姿勢は依然として経路計画の一部です。

移動コマンドの後の非移動コマンドをスムーズに実行することを試行

デフォルトではチェックが外れており、**移動ステップ**の間に「ビジョン処理による認識」、「DO を設定」、「DI をチェック」などの**非移動ステップ**を接続すると、ロボット移動経路の計画を中断し、ロボット実機動作中に一時中止することがあります。チェックを入れると現在のステップが完了しなくても続行することが可能です。これにより、ロボットがよりスムーズに動作します。ただし、ステップが途中で終了する可能性があります。

ステップが途中で終了する原因は何ですか？



Mech-Viz 実行中、ロボットに同時に複数の位置姿勢を送信しますが、最後の位置姿勢が現在のロボットの関節角度と一致しているかのみを判断します。一致していると判断したら、ロボットが最後の位置に到達していると考えられます。例えば、10 の移動ステップがある経路では、移動ステップ 5 の位置姿勢は最後のステップの位置姿勢とは同じとします。ロボットが低速に動作し、移動ステップ 5 の位置に到達したらその位置の関節角度を Mech-Viz に送信します。経路では、移動ステップ 5 の位置姿勢は最後の移動ステップの位置姿勢と同じなので、Mech-Viz はロボットの動作が完了したと判断して途中で実行を終了します。

配置された対象物との衝突を検出しない

デフォルトではチェックが外れており、配置された対象物との衝突を検出しません。チェックを入れると、ロボット本体・ロボットハンドと配置された対象物との衝突を検出します。

パレタイジングのシーンでは、以下場合があります。

1. ロボットが段ボール箱を配置するときに配置済みの箱と軽く衝突する（箱の凹みや変形などが発生しない）ことがあります。Mech-Viz ではこのような衝突を検出したら別の配置位置を計画するためパレットが満杯になりません。

2. 普通、吸盤を使用する場合、TCP を吸盤の表面でなく、モデル内部に設定するため、物体を吸着する時、吸盤と把持する箱のモデルと重なります（ソフトウェアでは吸盤と把持対象物との衝突を検出しない）。ロボットが箱を配置してから、把持された箱のモデルはシーンのモデルになり、吸盤がシーンの箱のモデルと衝突すると判断してメッセージを表示してパレタイジングが続行できなくなります。

これをチェックするとロボット本体やロボットハンドと配置済み対象物モデルとの衝突を検出しないので上記の問題を解決できます。

点群との衝突検出モード

現場の状況に応じて設定してください。普通、デフォルトの **自動** を使用します。ロボットが物体を把持する前の移動ステップは、**チェックしない** を選択し、把持した後の移動ステップを **チェック** を選択します。

自動	初期値。「ビジョン処理による移動」ステップと「ビジョン処理による移動」ステップに依存する「相対移動」の点群衝突だけを検出します。
チェックしない	移動ステップの点群衝突をいっさい検出しません。
チェック	全ての移動ステップの点群衝突を検出します。



衝突検出、**衝突検出設定**、**点群と他の対象物間の衝突を検出** をオンにすれば、Mech-Viz は経路計画を行うときにロボットモデル、ツールモデルと点群との衝突を検出します。普通、点群衝突の設定は、ロボットが把持を実行するときに把持対象物との衝突を検出するためです。空間内にノイズがあれば、ソフトウェアは対象物を把持する前の経路を計画するときに、ノイズがロボットモデル、ツールモデルと接触して点群との衝突が誤って判断されて経路計画の誤りが発生します。

対象物の対称性を使用しない

このパラメータは、**目標点タイプ** が **対象物位置姿勢** の移動ステップ（目標点タイプが物体の位置姿勢の移動ステップ、パレタイジングステップなど）に対してのみ有効です。目標点のタイプが関節角度、TCP位置姿勢の移動ステップには無効です。

無し	無し：初期値。対称性を無効にしません。
Z 軸	Z 軸の対称性のみを使用しません。
XY 軸	X、Y 軸の対称性のみを使用しません。
全て	全ての対称性を使用しません。

物体の対称性を使用しないように設定すると、ロボットが物体の位置姿勢に正確に到達して対象物を配置します。



対象物を把持できない場合に **リソース**、**対象物設定** の **回転対称** を設定します。対象物に対称性がある場合、複数の候補位置姿勢があります。Mech-Viz では対象物の把持を計画するときに、デフォルト位置姿勢が把持できない場合、候補位置姿勢を試行します。物体対

称位置姿勢と Mech-Vision によって出力された元位置姿勢とは一致しなければ、ロボットによる対象物の配置位置姿勢は一致しないことがあります。

把持された対象物との衝突検出モード

シーンの物体との衝突を検出しない

デフォルトではチェックが外れています。チェックすると把持された対象物とシーンの物体、ロボットとの衝突を検出せず、ソフトウェアの計画の速度を改善できます。普通、ロボットが対象物を把持したあとの移動ステップに使用します。

衝突を検出しなければ、衝突が発生するリスクがあるので注意して使用してください。



衝突検出 > **把持されている対象物の設定** > **把持されている対象物とその他の物体との衝突を検出** をオンにすれば、把持されている対象物とシーンの物体のモデル・ロボットとの衝突を検出します。パレタイジングのシーンでは、ビジョン処理により取得した箱の寸法にはささやかな誤差があり、把持を実行する時に箱同士の摩擦が発生しますが衝突は発生しません。

パレタイズのシーンでは、「シーンの物体との衝突を検出しない」をチェックしても対象物と配置済みの箱との衝突検出に干渉しません。箱の山の下にシーンの物体がある場合、経路計画のためにこれをチェックしてください。

点群との衝突を検出しない

デフォルトではチェックが外れています。チェックすると **把持対象物のモデル** と **シーンの点群** が衝突するかを検出しないので、Mech-Viz での衝突検出計算量を低減し、計画の効率を向上させることができます。また、把持対象物と点群との衝突の誤検出も回避できます。



- **衝突検出** > **把持されている対象物の設定** > **把持されている対象物とその他の物体との衝突を検出** と **点群設定** > **点群と他の対象物間の衝突を検出** をオンにすれば、把持されている対象物とシーンの物体のモデル・ロボットとの衝突を検出します。
- Mech-Vision が点群情報と対象物モデル情報を Mech-Viz に送信するときに、点群と対象物モデルは密着している状態です。ロボットが対象物を把持するときに、モデルはロボットの経路に沿って移動し、把持対象物モデルは点群と衝突します。
- 把持対象物は物体の点群と衝突する場合、この衝突をチェックすると Mech-Viz の計算量が増え、計画時間が長くなります。

サービス名

Adapter を Mech-Center に登録したサービス名。Adapter と一致させなければなりません。

関節角度を取得（計画の初期関節角度として）

デフォルトではチェックが外れており、経路計画の初期位置は前回の計画終了時の位置を使用します。チェックすると、経路計画の初期位置は、外部から取得したロボットの関節位置に更新されます。普通、Mech-Viz セミティーチングの場合に使用します。

5.4.1.6. 相対移動

機能

ある目標点に向かう相対移動。座標系や移動の方向、回転が選択できます。

使用シーン

「ビジョン処理による移動」の前・後に使用します。ビジョン位置姿勢のZ軸に沿って相対移動します。

パラメータ

移動ステップの共通パラメータ

移動目標点を送信

デフォルトではチェックが入っています。相手側（ロボットなど）に移動目標点の位置姿勢を送信します。チェックを外したら送信しませんが、この目標点位置姿勢は依然として経路計画の一部です。

移動コマンドの後の非移動コマンドをスムーズに実行することを試行

デフォルトではチェックが外れており、**移動ステップ**の間に「ビジョン処理による認識」、「DOを設定」、「DIをチェック」などの**非移動ステップ**を接続すると、ロボット移動経路の計画を中断し、ロボット実機動作中に一時中止することがあります。チェックを入れると現在のステップが完了しなくても続行することが可能です。これにより、ロボットがよりスムーズに動作します。ただし、ステップが途中で終了する可能性があります。



ステップが途中で終了する原因は何ですか？

Mech-Viz 実行中、ロボットに同時に複数の位置姿勢を送信しますが、最後の位置姿勢が現在のロボットの関節角度と一致しているかのみを判断します。一致していると判断したら、ロボットが最後の位置に到達していると見なします。例えば、10の移動ステップがある経路では、移動ステップ5の位置姿勢は最後のステップの位置姿勢とは同じとします。ロボットが低速に動作し、移動ステップ5の位置に到達したらその位置の関節角度をMech-Vizに送信します。経路では、移動ステップ5の位置姿勢は最後の移動ステップの位置姿勢とは同じなので、Mech-Vizはロボットの動作が完了したと判断して途中で実行を終了します。

配置された対象物との衝突を検出しない

デフォルトではチェックが外れており、配置された対象物との衝突を検出しません。チェックを入れると、ロボット本体・ロボットハンドと配置された対象物との衝突を検出します。

パレタイジングのシーンでは、以下場合があります。

1. ロボットが段ボール箱を配置するときに配置済みの箱と軽く衝突する（箱の凹みや変形などが発生しない）ことがあります。Mech-Vizではこのような衝突を検出したら別の配置位置

を計画するためパレットが満杯になりません。

2. 普通、吸盤を使用する場合、TCP を吸盤の表面でなく、モデル内部に設定するため、物体を吸着する時、吸盤と把持する箱のモデルと重なります（ソフトウェアでは吸盤と把持対象物との衝突を検出しない）。ロボットが箱を配置してから、把持された箱のモデルはシーンのモデルになり、吸盤がシーンの箱のモデルと衝突すると判断してメッセージを表示してパレタイジングが続行できなくなります。

これをチェックするとロボット本体やロボットハンドと配置済み対象物モデルとの衝突を検出しないので上記の問題を解決できます。

点群との衝突検出モード

現場の状況に応じて設定してください。普通、デフォルトの **自動** を使用します。ロボットが物体を把持する前の移動ステップは、**チェックしない** を選択し、把持した後の移動ステップを **チェック** を選択します。

自動 初期値。「ビジョン処理による移動」ステップと「ビジョン処理による移動」ステップに依存する「相対移動」の点群衝突だけを検出します。

チェックしない 移動ステップの点群衝突をいっさい検出しません。

チェック 全ての移動ステップの点群衝突を検出します。



衝突検出、**衝突検出設定**、**点群と他の対象物間の衝突を検出** をオンにすれば、Mech-Viz は経路計画を行うときにロボットモデル、ツールモデルと点群との衝突を検出します。普通、点群衝突の設定は、ロボットが把持を実行するときに把持対象物との衝突を検出するためです。空間内にノイズがあれば、ソフトウェアは対象物を把持する前の経路を計画するときに、ノイズがロボットモデル、ツールモデルと接触して点群との衝突が誤って判断されて経路計画の誤りが発生します。

対象物の対称性を使用しない

このパラメータは、**目標点タイプ** が **対象物位置姿勢** の移動ステップ（目標点タイプが物体の位置姿勢の移動ステップ、パレタイジングステップなど）に対してのみ有効です。目標点のタイプが関節角度、TCP位置姿勢の移動ステップには無効です。

無し 無し：初期値。対称性を無効にしません。

Z 軸 Z 軸の対称性のみを使用しません。

XY 軸 X、Y 軸の対称性のみを使用しません。

全て 全ての対称性を使用しません。

物体の対称性を使用しないように設定すると、ロボットが物体の位置姿勢に正確に到達して対象物を配置します。



対象物を把持できない場合に **リソース**、**対象物設定** の **回転対称** を設定します。対象物に

対称性がある場合、複数の候補位置姿勢があります。Mech-Viz では対象物の把持を計画するときに、デフォルト位置姿勢が把持できない場合、候補位置姿勢を試行します。物体対称位置姿勢と Mech-Vision によって出力された元位置姿勢とは一致しなければ、ロボットによる対象物の配置位置姿勢は一致しないことがあります。

把持された対象物との衝突検出モード

シーンの物体との衝突を検出しない

デフォルトではチェックが外れています。チェックすると把持された対象物とシーンの物体、ロボットとの衝突を検出せず、ソフトウェアの計画の速度を改善できます。普通、ロボットが対象物を把持したあとの移動ステップに使用します。

衝突を検出しなければ、衝突が発生するリスクがあるので注意して使用してください。



衝突検出 > **把持されている対象物の設定** > **把持されている対象物とその他の物体との衝突を検出** をオンにすれば、把持されている対象物とシーンの物体のモデル・ロボットとの衝突を検出します。パレタイジングのシーンでは、ビジョン処理により取得した箱の寸法にはささやかな誤差があり、把持を実行する時に箱同士の摩擦が発生しますが衝突は発生しません。

パレタイズのシーンでは、「シーンの物体との衝突を検出しない」をチェックしても対象物と配置済みの箱との衝突検出に干渉しません。箱の山の下にシーンの物体がある場合、経路計画のためにこれをチェックしてください。

点群との衝突を検出しない

デフォルトではチェックが外れています。チェックすると **把持対象物のモデル** と **シーンの点群** が衝突するかを検出しないので、Mech-Viz での衝突検出計算量を低減し、計画の効率を向上させることができます。また、把持対象物と点群との衝突の誤検出も回避できます。



- **衝突検出** > **把持されている対象物の設定** > **把持されている対象物とその他の物体との衝突を検出** と **点群設定** > **点群と他の対象物間の衝突を検出** をオンにすれば、把持されている対象物とシーンの物体のモデル・ロボットとの衝突を検出します。
- Mech-Vision が点群情報と対象物モデル情報を Mech-Viz に送信するときに、点群と対象物モデルは密着している状態です。ロボットが対象物を把持するときに、モデルはロボットの経路に沿って移動し、把持対象物モデルは点群と衝突します。
- 把持対象物は物体の点群と衝突する場合、この衝突をチェックすると Mech-Viz の計算量が増え、計画時間が長くなります。

基本的な移動設定

基本的な移動設定は、移動ステップの主なパラメータを設定し、ロボットが目標点に到達する **速度** と **運動方式** を決定します。

把持と配置設定

未指定 初期値

把持 「ビジョン処理による移動」の前の移動ステップ。

放置 「ビジョン処理による移動」の後の移動ステップ。

調整説明 Mech-Viz プロジェクトのロジック検査に使用されます。現場の実行手順に応じて、把持をしてから配置するという基本的な原則に従って設定します。

運動タイプ

関節運動 ロボットが円弧に沿って走行します。スムーズに走行できるので特異点を回避することが可能。ロボットの移動範囲が広くて高い精度を求めないシーンに適用しています。

直線運動 ロボットが直線に沿って走行します。溶接や接着剤塗布、把持など高い精度が求められるシーンに適用しています。

速度&加速度

速度&加速度は、ロボット動作のスピードを決定します。普通、加速度の値を速度より小さく設定します。加速度の値を速度よりも大きくしたらロボット動作がスムーズでなくなります。



ロボットが安定に把持を実行するように、「ビジョン処理による移動」とその前後のステップの速度を低く設定してください。

ブレンド半径

初期値 50.00mm

調整説明 調整説明：通常、初期値を使用すればいいです。

ブレンド半径とは、回転する位置が目標点までの距離であり、大きいほどロボットの動きはスムーズになります。ロボットが狭いスペースで動作する場合、ブレンド半径を大きく設定する必要はありません。

ロボットがより広いスペースで動作し、障害物がなくて2つの経路の間に距離が遠い場合、ロボットがスムーズに動くように、ブレンド半径を大きく設定することができます。

移動基準

オプション：前の経路点に対して、次の経路点に対して。

調整説明：**次の経路点に対して** は次の経路点を基準に移動します。**次の経路点に対して** は次の経路点を基準に移動します。

目標点タイプ

ロボットハンド、ロボット、基準点 を選択できます。

ロボットハンド

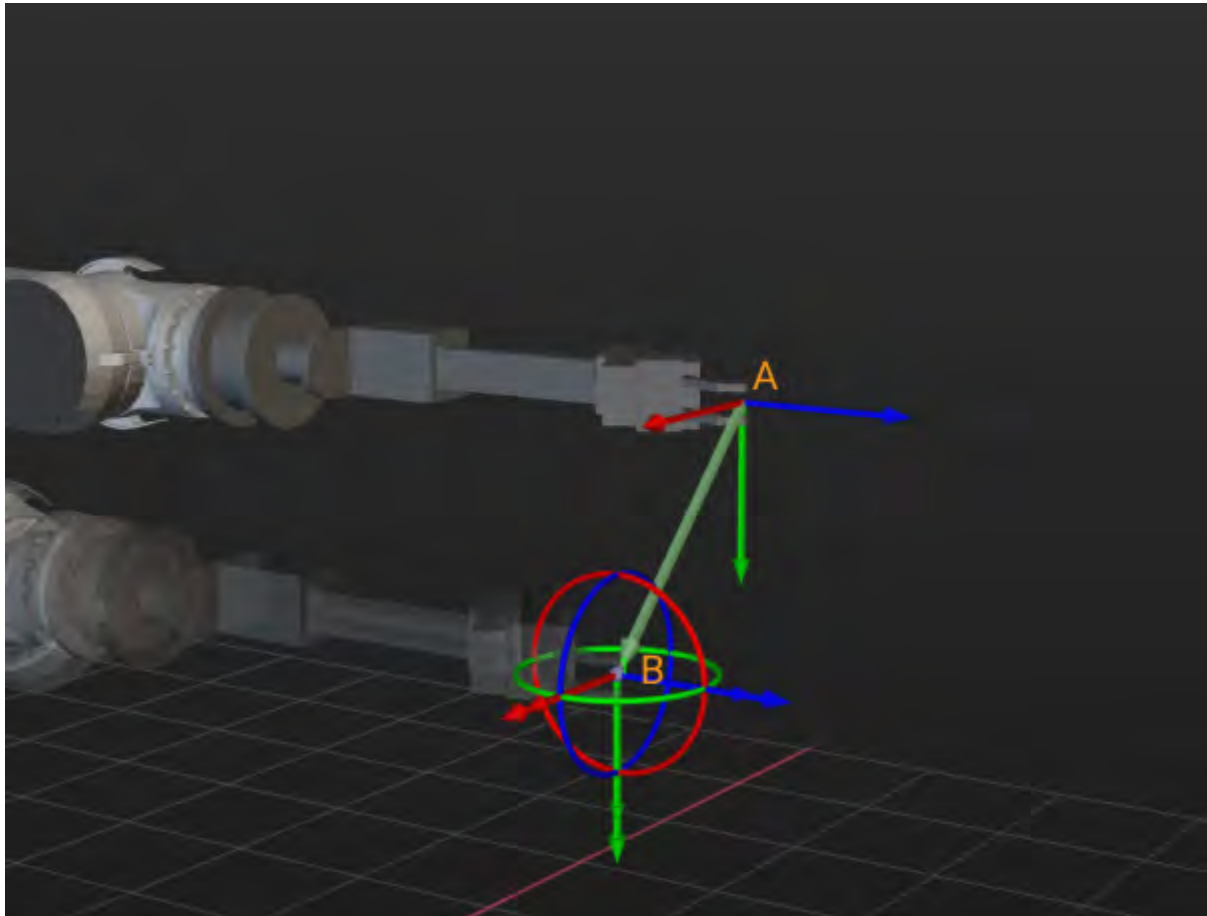
「基準経路点」の「ロボットハンド座標原点」からオフセットします。



「基準経路点」とは、「前の経路点」または「次の経路点」のことです。

座標 オフセット距離を設定するために使用されます。

下図に示すように、Aは基準経路点で、BはX、Yを300に設定した時のオフセット結果です。このステップを実行すると、ロボットはAからBに移動します。

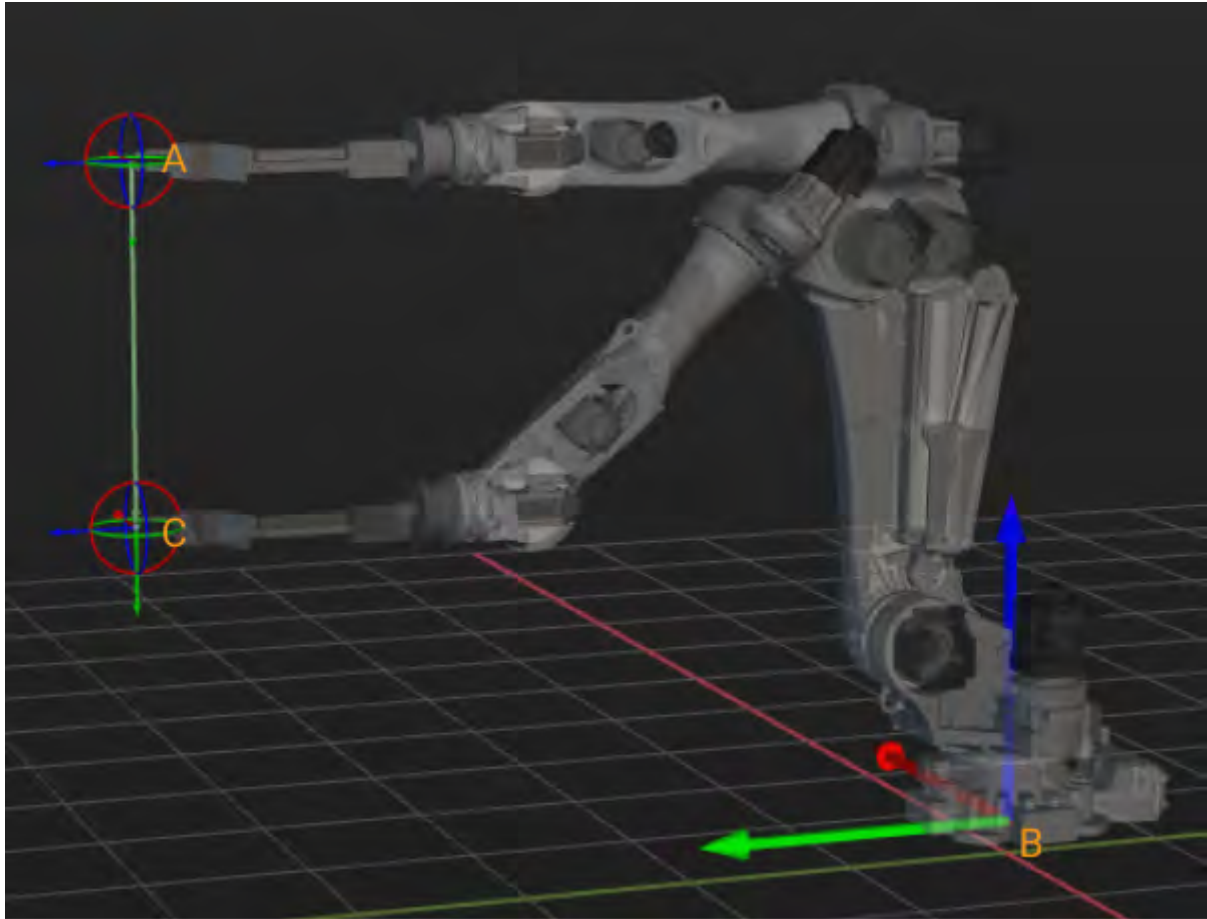


ロボット

「基準経路点」から「ロボット座標系の方向」に沿ってオフセットします。

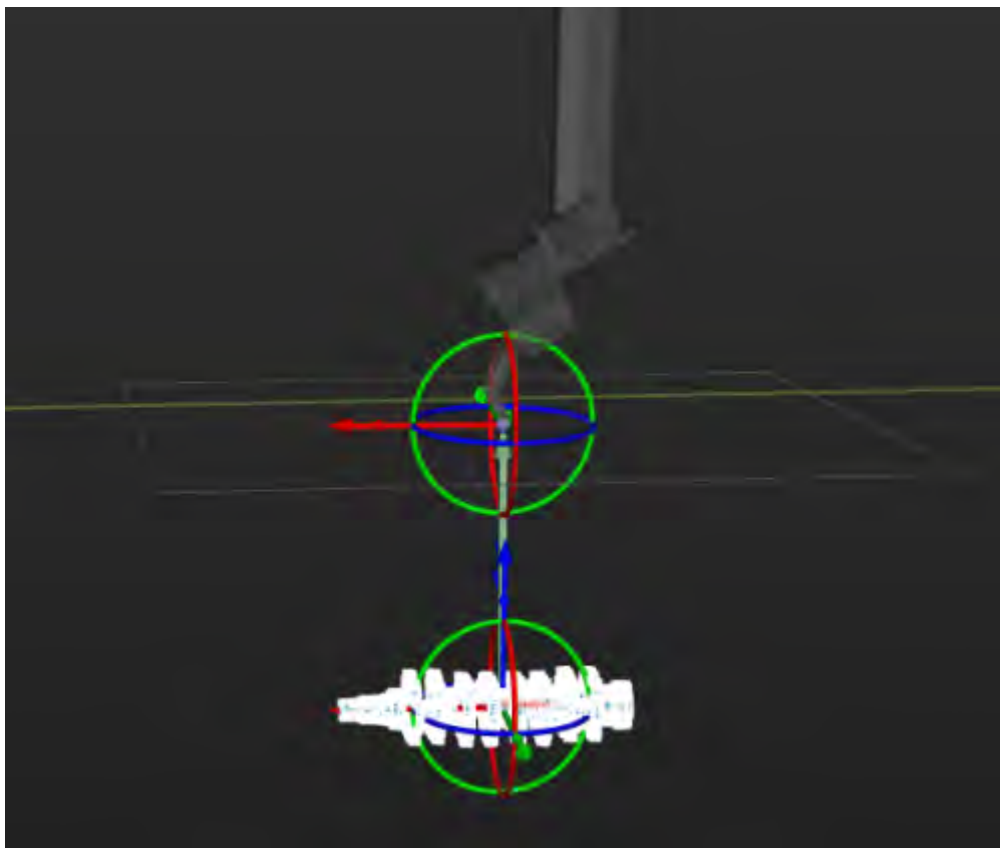
座標 オフセット距離を設定するために使用されます。

下図に示すように、Aは基準経路点で、Bはロボット座標系の原点で、CはZを-1000に設定した時のオフセット結果です。このステップを実行すると、ロボットはAからCに移動します。



X/Y/Z をロック

Zをロック を常にチェックします。**Zをロック** をチェックすると、下図に示すように平面が表示されます。**ビジョン処理による移動** と併用することで相対移動を指定した Z 値に固定します。これによってロボットが把持を実行したあとで所定の高さに移動します。



基準点

基準点は基準方向を指定します。対象物が深いコンテナに配置された場合、把持する時にロボットとコンテナとの衝突が発生する可能性があります。**基準点**を選択して設定することで相対移動をオフセットすることができます。これによって衝突回避が可能になります。普通、**ビジョン処理による移動**と組み合わせて使用します。

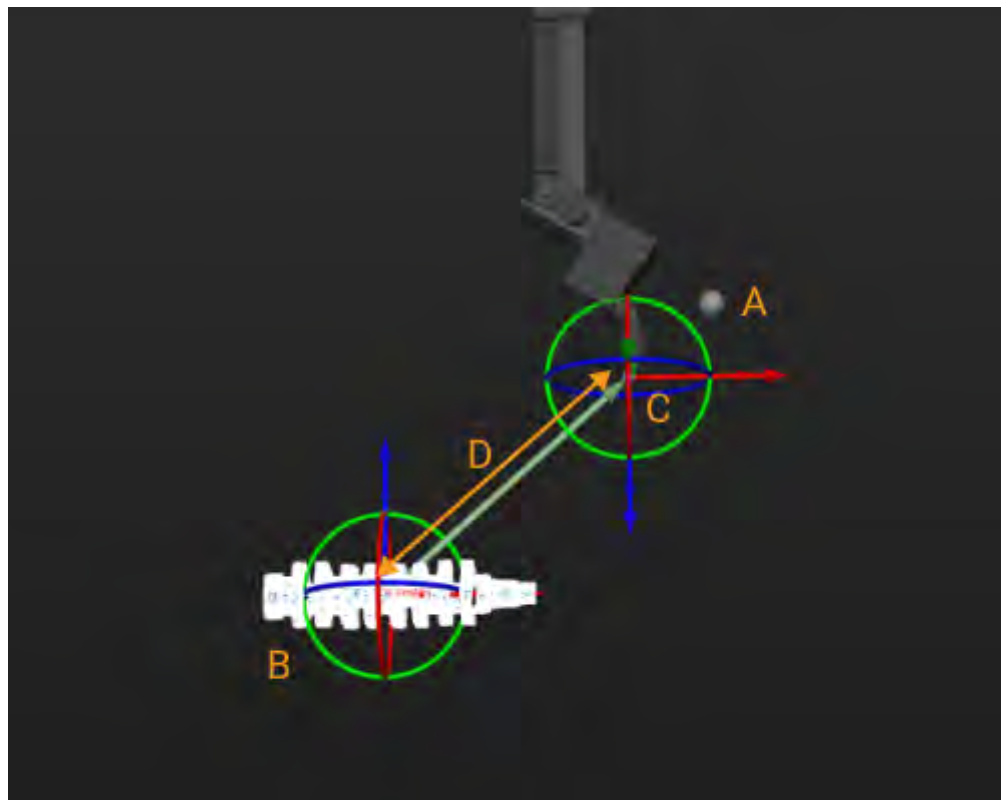
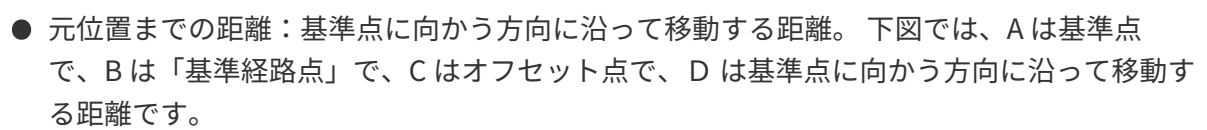
マークポイントX/Y/Z

基準点の位置を設定します。

経路点の位置

オフセット距離を設定するために使用されます。

- 世界座標系の Z 値：オフセット点から世界座標系の XOY 平面までの垂直距離。下図では、A は基準点で、B は「基準経路点」で、C はオフセット点で、D はオフセット点から世界座標系の XOY 平面までの垂直距離です。



XOY平面で移動 をチェックすると、「基準経路点」の XOY 平面にのみオフセット点を設定できます。

5.4.2. DI DO

5.4.2.1. DI をチェック

機能

指定された DI ポートの信号をチェックします。

パラメータ

非移動ステップの共通パラメータ

実行をスキップ

スキップしない	初期値、このステップをスキップしません。
シミュレーション時	シミュレーションを実行する時にこのステップをスキップします。出口は スキップ時の出口のインデックス によって決定されます。
いつも	シミュレーション時も実際に実行中にも実行をスキップします。出口は スキップ時の出口のインデックス によって決定されます。
調整説明	シミュレーション時 または いつも に設定すると、プロジェクト実行中にこのステップをスキップして後のステップを実行します。プロジェクトに「DI をチェック」ステップを配置したら、シミュレーション時に外部信号がなく、実行をスキップしなければプロジェクトの実行はここで停止することになります。このパラメータを設定すると、このステップを実行せずに次のステップを実行してシミュレーションを完了します。

スキップ時の出口のインデックス

実行をスキップ を **シミュレーション時** または **いつも** に設定すれば有効になります。このパラメータは、スキップしてから実行する出口を指定します。



- このステップの出口は 0 と 1 なので、**スキップ時の出口のインデックス** を設定する時、0 あるいは 1 にしか設定できません。
- 「実行をスキップ」を「スキップしない」に設定すると、シミュレートする時にソフトウェアは DI 値をランダムに出力し、ランダムにいずれかの出口から実行します。

DI ポート

初期値 0

調整説明 ロボット実機の配線に合わせて DI ポート番号を設定します。1 に設定すると DI1 を、2 に設定すると DI2 をチェックします。

事前計画する出口のインデックス

初期値 -1

オプション -1、0、1

調整説明 ロボットの DI をチェックする時、このステップの前と後のステップは依存関係にある場合、全体の計画が中断されるため同時に計画することはできません。

このパラメータを設定することで全体の計画を中断せずに信号を指定したポート（0 または 1）から出力します。

-1 に設定すると、DI 信号をチェックし、全体の計画を中断します。



0 に設定すると、信号はポート 0 から出力され、このステップの前と後のステップを接続して計画を実行します。これによってロボットがさらにスムーズに移動します。



1 に設定すると、信号はポート 1 から出力され、このステップの前と後のステップを接続して計画を実行します。これによってロボットがさらにスムーズに移動します。



5.4.2.2. DI リストをチェック

機能

お使いのロボットのすべての DI ポートの信号値が要求を満たしているをチェックし、かつ信号を相応のポートから出力します。すなわち、DI ポートの信号値が要求を満たしている時に、信号は **全部満たしている** ポートから実行し、満たさない時は **少なくとも一つが満たさない** ポートから実行します。

パラメータ

非移動ステップの共通パラメータ

実行をスキップ

スキップしない	初期値、このステップをスキップしません。
シミュレーション時	シミュレーションを実行する時にこのステップをスキップします。出口は スキップ時の出口のインデックス によって決定されます。
いつも	シミュレーション時も実際に実行中にも実行をスキップします。出口は スキップ時の出口のインデックス によって決定されます。
調整説明	シミュレーション時 または いつも に設定すると、プロジェクト実行中にこのステップをスキップして後のステップを実行します。プロジェクトに「DI をチェック」ステップを配置したら、シミュレーション時に外部信号がなく、実行をスキップしなければプロジェクトの実行はここで停止することになります。このパラメータを設定すると、このステップを実行せずに次のステップを実行してシミュレーションを完了します。

スキップ時の出口のインデックス

実行をスキップ を **シミュレーション時** または **いつも** に設定すれば有効になります。このパラメータは、スキップしてから実行する出口を指定します。

事前計画する出口のインデックス

初期値	-1
オプション	-1、0、1

調整説明

ロボットのDIをチェックする時、このステップの前と後のステップは依存関係にある場合、全体の計画が中断されるため同時に計画することはできません。このパラメータを設定することで全体の計画を中断せずに信号を指定したポートから出力します。

- -1 に設定すると、DI 信号をチェックし、全体の計画を中断します。
- 0 に設定すると、信号は **ポート 0** から出力され、このステップの前と後のステップを接続して計画を実行します。これによってロボットがさらにスムーズに動作します。
- 1 に設定すると、信号は **ポート 1** から出力され、このステップの前と後のステップを接続して計画を実行します。これによってロボットがさらにスムーズに動作します。

5.4.2.3. DI を待つ

機能

指定されたロボットのDIポートが設定した値になるまで待ちます。

パラメータ

非移動ステップの共通パラメータ

実行をスキップ

スキップしない

初期値、このステップをスキップしません。

シミュレーション時

シミュレーションを実行する時にこのステップをスキップします。出口は **スキップ時の出口のインデックス** によって決定されます。

いつも

シミュレーション時も実際に実行中にも実行をスキップします。出口は **スキップ時の出口のインデックス** によって決定されます。

調整説明

シミュレーション時 または **いつも** に設定すると、プロジェクト実行中にこのステップをスキップして後のステップを実行します。プロジェクトに「DIをチェック」ステップを配置したら、シミュレーション時に外部信号がなく、実行をスキップしなければプロジェクトの実行はここで停止することになります。このパラメータを設定すると、このステップを実行せずに次のステップを実行してシミュレーションを完了します。

スキップ時の出口のインデックス

実行をスキップ を **シミュレーション時** または **いつも** に設定すれば有効になります。このパラメータは、スキップしてから実行する出口を指定します。

事前計画する出口のインデックス

初期値 -1

オプション -1、0、1

調整説明 DI を待つ時に全体の計画が中断されます。このステップの前と後のステップは依存関係にある場合は、それらを接続することはできません。このパラメータを設定することで信号を所定のポートから出力し、かつ全体の計画を中断しません。

-1 に設定すると、現在の DI 信号を待ち、計画は中断されます。



0 に設定すると、このステップの前と後のステップを接続して計画します。これによって、ロボットがさらにスムーズに動作します。



1 に設定すると、このステップの前と後のステップを接続して計画します。これによって、ロボットがさらにスムーズに動作します。



このパラメータを設定する前に、ポートを追加する必要があります。

+		-	
ポート			値
1	0	<input checked="" type="checkbox"/>	1
2	1	<input checked="" type="checkbox"/>	1

タイムアウト時間

初期値 -1

調整説明 -1 に設定すると、現在の DI ポート信号が指定した値になるまで待ちます。



そのほかの値に設定すると、待ち時間が設定した値を超えたらタイムアウトのポートから実行します。



「ビジョン処理による移動」から DI リストを取得

ビジョン処理による移動の「ロボットハンドの種類」に **デパレタイズ用吸盤** または **配列タイプグリッパ** を選択すると、DI 信号を設定できます。対応する「ビジョン処理による移動」ステップを選択すると、このステップでは設定したパラメータが適用されます。

5.4.2.4. DO を設定

機能

指定されたロボットの DO ポートの信号を設定します。

パラメータ

非移動ステップの共通パラメータ

実行をスキップ

スキップしない 初期値、このステップをスキップしません。

シミュレーション時 シミュレーションを実行する時にこのステップをスキップします。出口は **スキップ時の出口のインデックス** によって決定されます。

いつも シミュレーション時も実際に実行中にも実行をスキップします。出口は **スキップ時の出口のインデックス** によって決定されます。

調整説明 **シミュレーション時** または **いつも** に設定すると、プロジェクト実行中にこのステップをスキップして後のステップを実行します。プロジェクトに「DI をチェック」ステップを配置したら、シミュレーション時に外部信号がなく、実行をスキップしなければプロジェクトの実行はここで停止することになります。このパラメータを設定すると、このステップを実行せずに次のステップを実行してシミュレーションを完了します。

スキップ時の出口のインデックス

実行をスキップ を **シミュレーション時** または **いつも** に設定すれば有効になります。このパラメータは、スキップしてから実行する出口を指定します。

移動の完全な到達を待つ

デフォルトではチェックが入っています。このステップの前に、ロボットが完全に指定された位置に到達してはじめてこのステップが実行されます。

DO 値

調整説明：必要に応じてチェックしてください。

DO ポート

初期値 0

調整説明 ロボットの IO 配線に応じてポート番号を入力してください。

遅延時間

初期値は 0 であり、必要に応じて信号入力の遅延時間を設定します。

5.4.2.5. DO リストを設定

機能

複数の DO ポートに信号を設定します。以下のシーンに使用できます：

- 複数の吸盤を同時にオンにします。ポート番号はユーザーによって設定されます。
- **DI リストをチェック** あるいは **ビジョン処理による移動** と組み合わせて使用します。**ビジョン処理による移動** から対応する DO 値を取得します。

パラメータ

非移動ステップの共通パラメータ

実行をスキップ

スキップしない	初期値、このステップをスキップしません。
シミュレーション時	シミュレーションを実行する時にこのステップをスキップします。出口は スキップ時の出口のインデックス によって決定されます。
いつも	シミュレーション時も実際に実行中にも実行をスキップします。出口は スキップ時の出口のインデックス によって決定されます。
調整説明	シミュレーション時 または いつも に設定すると、プロジェクト実行中にこのステップをスキップして後のステップを実行します。プロジェクトに「DI をチェック」ステップを配置したら、シミュレーション時に外部信号がなく、実行をスキップしなければプロジェクトの実行はここで停止することになります。このパラメータを設定すると、このステップを実行せずに次のステップを実行してシミュレーションを完了します。

スキップ時の出口のインデックス

実行をスキップ を **シミュレーション時** または **いつも** に設定すれば有効になります。このパラメータは、スキップしてから実行する出口を指定します。

「ビジョン処理による移動」から DO リストを取得

ビジョン処理による移動 の「ロボットハンドの種類」に **デパレタイズ用吸盤** または **配列タイプグリッパ** を選択すると、DO 信号を設定することができます。対応する「ビジョン処理による移動」ステップを選択すると、このステップでは設定したパラメータが適用されます。

5.4.3. 論理トポロジー

5.4.3.1. 道標によって異なる分岐を実行

機能

普通、「道標を設定」と組み合わせて使用します。異なる分岐に共通の部分がある場合に、それを複製する必要なく、**道標を設定** ステップ使用して現在の分岐を共通の部分として設定し、**道標によって異なる分岐を実行** ステップを実行すると元の分岐を実行することができます。

パラメータ

非移動ステップの共通パラメータ

実行をスキップ

スキップしない	初期値、このステップをスキップしません。
シミュレーション時	シミュレーションを実行する時にこのステップをスキップします。出口は スキップ時の出口のインデックス によって決定されます。
いつも	シミュレーション時も実際に実行中にも実行をスキップします。出口は スキップ時の出口のインデックス によって決定されます。
調整説明	シミュレーション時 または いつも に設定すると、プロジェクト実行中にこのステップをスキップして後のステップを実行します。プロジェクトに「DIをチェック」ステップを配置したら、シミュレーション時に外部信号がなく、実行をスキップしなければプロジェクトの実行はここで停止することになります。このパラメータを設定すると、このステップを実行せずに次のステップを実行してシミュレーションを完了します。

スキップ時の出口のインデックス

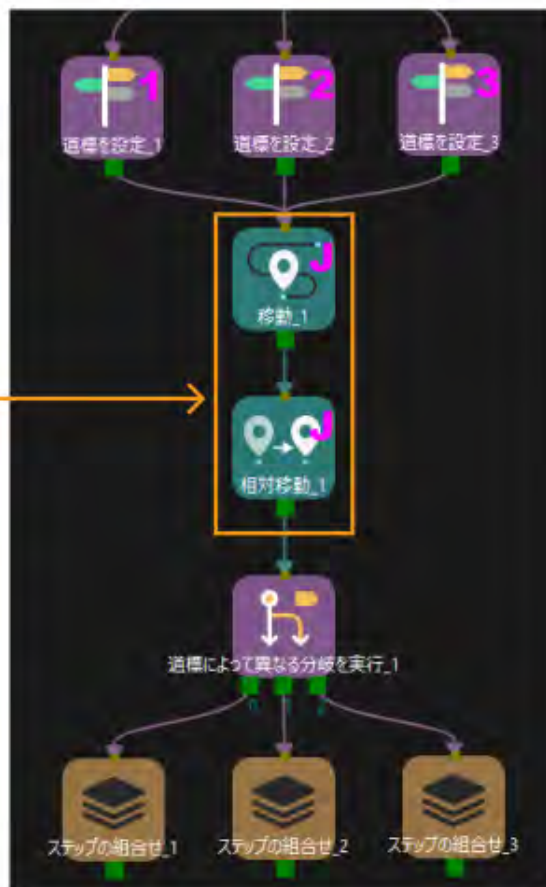
実行をスキップ を **シミュレーション時** または **いつも** に設定すれば有効になります。このパラメータは、スキップしてから実行する出口を指定します。

出口の数 このステップの出口の数を設定します。[道標を設定](#) の **タグ** と組み合わせて使用します。普通、道標の数と出口の数の値を一致させる必要があります。例えば、2つの道標を設定した場合、出口を2に設定しなければなりません。タグを出口0する必要があります。タグ1は [道標によって異なる分岐を実行](#) の出口0に対応し、タグ2は [道標によって異なる分岐を実行](#) の出口1に対応し、これによって類推することができます。

例

下図に示すように、3つの分岐があり、共通のステップがあり（移動と相対移動）、かつ共通ステップのパラメータ設定も一緒です。

1. 共通のステップを一つずつ保持します。
2. 分岐ごとに **道標を設定** ステップを追加し、タグの値をそれぞれ1、2、3に設定します。
3. 移動と相対移動を実行したあと、**道標によって異なる分岐を実行** ステップは設定した道標に従って実行します。



5.4.3.2. メッセージによって異なる分岐を実行

使用シーン

- 外部サービスから受信したデータによって異なる分岐を実行します
- プロジェクトの実行を中断して外部サービスを受信してから続行します

パラメータ

非移動ステップの共通パラメータ

実行をスキップ

スキップしない

初期値、このステップをスキップしません。

シミュレーション時

シミュレーションを実行する時にこのステップをスキップします。出口は **スキップ時の出口のインデックス** によって決定されます。

いつも

シミュレーション時も実際に実行中にも実行をスキップします。出口は **スキップ時の出口のインデックス** によって決定されます。

調整説明

シミュレーション時 または **いつも** に設定すると、プロジェクト実行中にこのステップをスキップして後のステップを実行します。プロジェクトに「DIをチェック」ステップを配置したら、シミュレーション時に外部信号がなく、実行をスキップしなければプロジェクトの実行はここで停止することになります。このパラメータを設定すると、このステップを実行せずに次のステップを実行してシミュレーションを完了します。

スキップ時の出口のインデックス

実行をスキップ を **シミュレーション時** または **いつも** に設定すれば有効になります。このパラメータは、スキップしてから実行する出口を指定します。

出口の数 このステップの出口の数。3つの分岐がある場合、このステップには3つの出口があり、3つのステップにそれぞれ接続します。プロジェクトの実行を中断したい場合、1 に設定してください。

出口の数 を 1 に設定した時、以下のパラメータを設定できます。

後続の移動に影響（計画が中断される）

チェックすると後続の実行が中断されます。プロジェクトを中断し、外部からの信号を待つ場合はチェックしないでください。

外部信号を受信せずに実行

ソフトウェアは外部サービスからの信号を一時的に保存します。このステップを実行する時は保存された分岐番号を実行します。チェックを外すと、このステップは外部サービスからの信号を受信せずに実行します。

待機タイムアウト

チェックすると、ステップにタイムアウトの出口が追加され、**待機タイムアウト時間** を設定できます。外部からの信号を待つ時間が **待機タイムアウト時間** を超えたら、タイムアウトの分岐を実行します。また、事前計画する出口も指定できます。タイムアウト以外の出口を指定できます。

出口の数 が 1 でない場合は、以下のパラメータを設定できます。

外部信号を受信せずに実行

ソフトウェアは外部サービスからの信号を一時的に保存します。このステップを実行する時は保存された分岐番号を実行します。チェックを外すと、このステップは外部サービスからの信号を受信せずに実行します。

事前計画する出口のインデックス

プロジェクトをスムーズに実行するために、事前計画する出口を指定する必要があります。

す。普通、可能性の高い分岐を指定します。

待機タイムアウト

チェックすると、ステップにタイムアウトの出口が追加され、**待機タイムアウト時間**を設定できます。外部からの信号を待つ時間が**待機タイムアウト時間**を超えたら、タイムアウトの分岐を実行します。

5.4.3.3. ステップの組合せ

機能

複数のステップを一つの組合せに合成できます。複数の出口がある場合、[ステップの組合せ](#)ステップと組み合わせて使用する必要があります。

パラメータ説明

調整可能なパラメータはありません。

名前

ステップの組合せの名前を編集します。

5.4.3.4. ステップの組合せの出口

機能

「ステップの組合せ」ステップを終了する時に実行する出口を指定します。

使用シーン

ステップの組合せの出口を指定します。

パラメータ

非移動ステップの共通パラメータ

実行をスキップ

スキップしない	初期値、このステップをスキップしません。
シミュレーション時	シミュレーションを実行する時にこのステップをスキップします。出口は スキップ時の出口のインデックス によって決定されます。
いつも	シミュレーション時も実際に実行中にも実行をスキップします。出口は スキップ時の出口のインデックス によって決定されます。

調整説明

シミュレーション時 または **いつも** に設定すると、プロジェクト実行中にこのステップをスキップして後のステップを実行します。プロジェクトに「DIをチェック」ステップを配置したら、シミュレーション時に外部信号がなく、実行をスキップしなければプロジェクトの実行はここで停止することになります。このパラメータを設定すると、このステップを実行せずに次のステップを実行してシミュレーションを完了します。

スキップ時の出口のインデックス

実行をスキップ を **シミュレーション時** または **いつも** に設定すれば有効になります。このパラメータは、スキップしてから実行する出口を指定します。

ステップの組合せを終了する時のポート番号

初期値 0

調整説明 異なるステップに接続するために、実行する「ステップの組合せ」ステップの出口を指定する

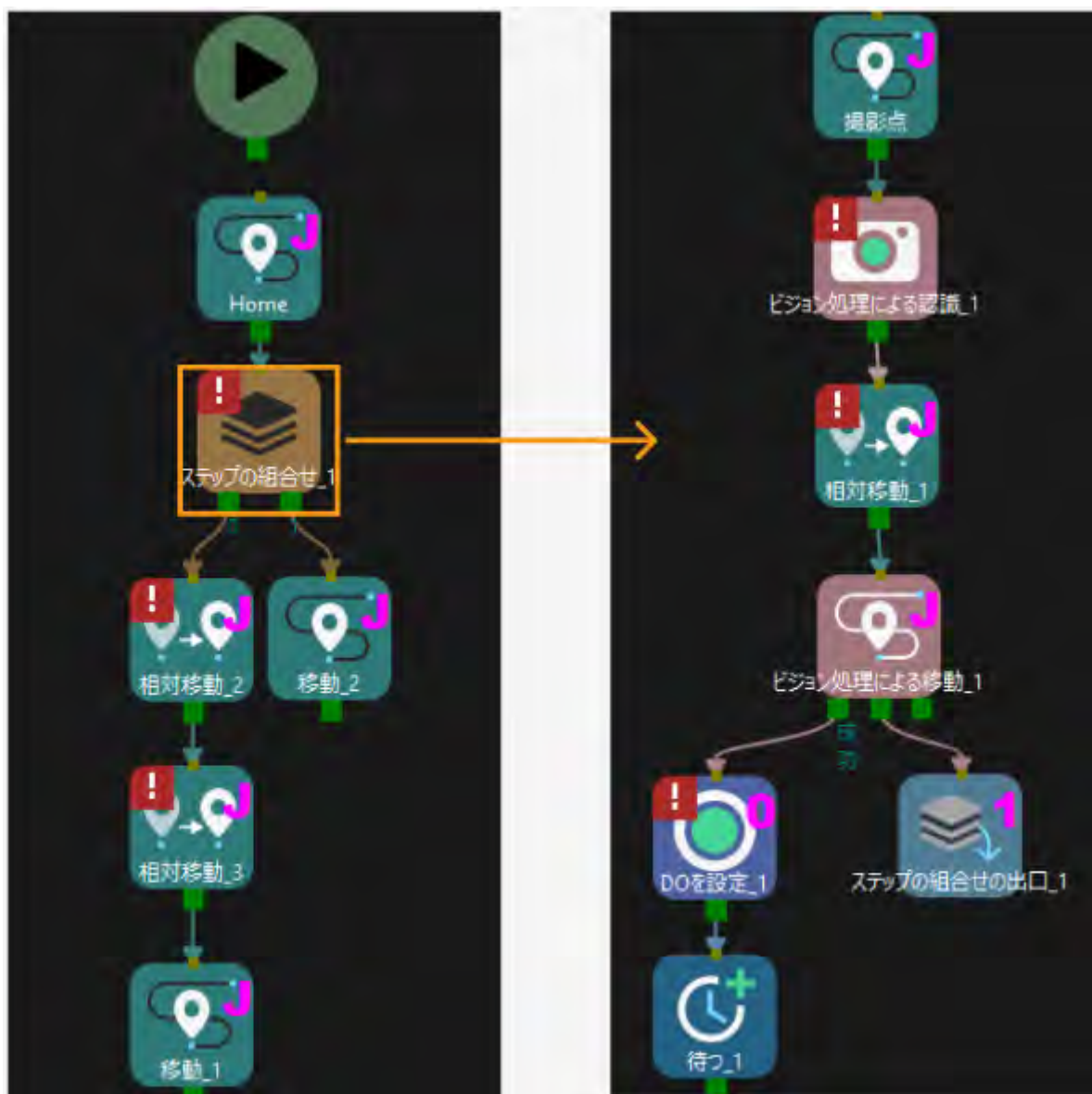
1 から順番に設定します。出口「0」は **ステップの組合せの出口** に接続されていない分岐に対応します。

例

ステップの組合せの「ビジョン処理による移動」は、ビジョン処理による認識に成功する、視覚点がないか把持に失敗する可能性があるため、異なるステップに繋ぐために二つの出口が必要です。

「ビジョン処理による移動」の「計画が失敗した時の出口」に **ステップの組合せの出口** を繋ぎ、**ステップの組合せの出口** ステップのパラメータ **ステップの組合せを終了する時のポート番号の値** を 1 に設定します。これで「ステップの組合せの出口」に出口 1 が追加され、ビジョン処理による移動 が失敗した時に出口 1 から実行します。

「ビジョン処理による移動」の成功出口は **ステップの組合せの出口** につながらない場合、実行した後は出口 0 からステップの組合せを終了します。



5.4.3.5. 道標を設定

機能

普通、「道標を設定」と組み合わせて使用します。複数の分岐に共通のステップがある場合、それを何回もコピーする必要なく、共通ステップの前後にそれぞれ **道標を設定** ステップと **道標によって異なる分岐を実行** ステップを接続すればいいです。

パラメータ

非移動ステップの共通パラメータ

実行をスキップ

スキップしない

初期値、このステップをスキップしません。

シミュレーション時	シミュレーションを実行する時にこのステップをスキップします。出口は スキップ時の出口のインデックス によって決定されます。
いつも	シミュレーション時も実際に実行中にも実行をスキップします。出口は スキップ時の出口のインデックス によって決定されます。
調整説明	シミュレーション時 または いつも に設定すると、プロジェクト実行中にこのステップをスキップして後のステップを実行します。プロジェクトに「DI をチェック」ステップを配置したら、シミュレーション時に外部信号がなく、実行をスキップしなければプロジェクトの実行はここで停止することになります。このパラメータを設定すると、このステップを実行せずに次のステップを実行してシミュレーションを完了します。

スキップ時の出口のインデックス

実行をスキップ を **シミュレーション時** または **いつも** に設定すれば有効になります。このパラメータは、スキップしてから実行する出口を指定します。

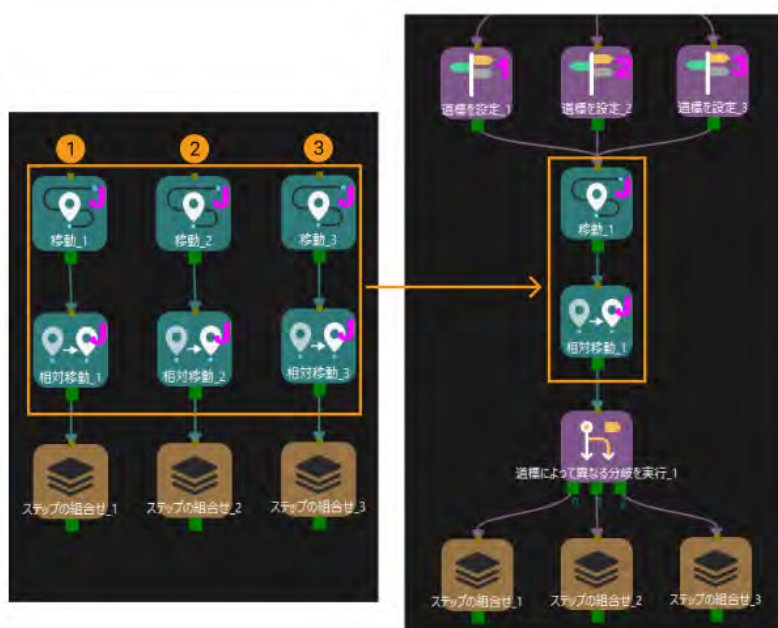
タグ

各分岐に番号を付けます。「道標によって異なる分岐を実行」の **出口の数** と合わせて使用します。道標を設定ステップの数と出口の数を一致させてください。例えば、2つの道標を設定すれば、出口も2に設定してください。また、タグ1は道標によって異なる分岐を実行の出口0を実行し、タグ2は出口1を実行します。これによって類推します。

例

下図に示すように、3つの分岐に共通ステップがあります（移動と相対移動、パラメータ値も同じです）。

1. 共通ステップを一つずつ残します。
2. 共通ステップの前に **道標を設定** ステップを配置し、それぞれ番号（1, 2, 3）を付けます。
3. **道標によって異なる分岐を実行** ステップを共通ステップの後に接続し、3つの出口を後続のステップに接続します。



5.4.4. パレタイジング

5.4.4.1. カスタマイズのパレットパターン

機能

パレットパターンをカスタマイズします。

使用シーン

対称性がなくて規則もないパレットパターンに適しています。

パラメータ

移動ステップの共通パラメータ

移動目標点を送信

デフォルトではチェックが入っています。相手側（ロボットなど）に移動目標点の位置姿勢を送信します。チェックを外したら送信しませんが、この目標点位置姿勢は依然として経路計画の一部です。

移動コマンドの後の非移動コマンドをスムーズに実行することを試行

デフォルトではチェックが外れており、**移動ステップ**の間に「ビジョン処理による認識」、「DOを設定」、「DIをチェック」などの**非移動ステップ**を接続すると、ロボット移動経路の計画を中断し、ロボット実機動作中に一時中止することがあります。チェックを入れると現在のステップが完了しなくても続行することが可能です。これにより、ロボットがよりスムーズに動作します。ただし、ステップが途中で終了する可能性があります。

ステップが途中で終了する原因は何ですか？



Mech-Viz 実行中、ロボットに同時に複数の位置姿勢を送信しますが、最後の位置姿勢が現在のロボットの関節角度と一致しているかのみを判断します。一致していると判断したら、ロボットが最後の位置に到達していると考えられます。例えば、10 の移動ステップがある経路では、移動ステップ 5 の位置姿勢は最後のステップの位置姿勢と同じとします。ロボットが低速に動作し、移動ステップ 5 の位置に到達したらその位置の関節角度を Mech-Viz に送信します。経路では、移動ステップ 5 の位置姿勢は最後の移動ステップの位置姿勢と同じなので、Mech-Viz はロボットの動作が完了したと判断して途中で実行を終了します。

配置された対象物との衝突を検出しない

デフォルトではチェックが外れており、配置された対象物との衝突を検出しません。チェックを入れると、ロボット本体・ロボットハンドと配置された対象物との衝突を検出します。

パレタイジングのシーンでは、以下場合があります。

1. ロボットが段ボール箱を配置するときに配置済みの箱と軽く衝突する（箱の凹みや変形などが発生しない）ことがあります。Mech-Viz ではこのような衝突を検出したら別の配置位置を計画するためパレットが満杯になりません。
2. 普通、吸盤を使用する場合、TCP を吸盤の表面でなく、モデル内部に設定するため、物体を吸着する時、吸盤と把持する箱のモデルと重なります（ソフトウェアでは吸盤と把持対象物との衝突を検出しない）。ロボットが箱を配置してから、把持された箱のモデルはシーンのモデルになり、吸盤がシーンの箱のモデルと衝突すると判断してメッセージを表示してパレタイジングが続行できなくなります。

これをチェックするとロボット本体やロボットハンドと配置済み対象物モデルとの衝突を検出しないので上記の問題を解決できます。

点群との衝突検出モード

現場の状況に応じて設定してください。普通、デフォルトの **自動** を使用します。ロボットが物体を把持する前の移動ステップは、**チェックしない** を選択し、把持した後の移動ステップを **チェック** を選択します。

自動	初期値。「ビジョン処理による移動」ステップと「ビジョン処理による移動」ステップに依存する「相対移動」の点群衝突だけを検出します。
チェックしない	移動ステップの点群衝突をいっさい検出しません。
チェック	全ての移動ステップの点群衝突を検出します。



衝突検出、**衝突検出設定**、**点群と他の対象物間の衝突を検出** をオンにすれば、Mech-Viz は経路計画を行うときにロボットモデル、ツールモデルと点群との衝突を検出します。普通、点群衝突の設定は、ロボットが把持を実行するときに把持対象物との衝突を検出するためです。空間内にノイズがあれば、ソフトウェアは対象物を把持する前の経路を計画するときに、ノイズがロボットモデル、ツールモデルと接触して点群との衝突が誤って判断

されて経路計画の誤りが発生します。

対象物の対称性を使用しない

このパラメータは、**目標点タイプ**が**対象物位置姿勢**の移動ステップ（目標点タイプが物体の位置姿勢の移動ステップ、パレタイジングステップなど）に対してのみ有効です。目標点のタイプが関節角度、TCP位置姿勢の移動ステップには無効です。

無し 無し：初期値。対称性を無効にしません。

Z 軸 Z 軸の対称性のみを使用しません。

XY 軸 X、Y 軸の対称性のみを使用しません。

全て 全ての対称性を使用しません。

物体の対称性を使用しないように設定すると、ロボットが物体の位置姿勢に正確に到達して対象物を配置します。



対象物を把持できない場合に **リソース**、**対象物設定** の **回転対称** を設定します。対象物に対称性がある場合、複数の候補位置姿勢があります。Mech-Viz では対象物の把持を計画するときに、デフォルト位置姿勢が把持できない場合、候補位置姿勢を試行します。物体対称位置姿勢と Mech-Vision によって出力された元位置姿勢とは一致しなければ、ロボットによる対象物の配置位置姿勢は一致しないことがあります。

インデックス

開始インデックス

説明 これから配置する箱のインデックス。

オプション 整数にしか設定できません。初期値は 0。

使用方法 空のパレットの場合に **0** に設定します。

パレタイジングを継続するとき、N 個の箱を配置した場合に値を N に設置します。これで N+1 個目の箱からパレタイジングを継続します。

現在のインデックス

説明 箱の位置を表示します。値が N であれば、**N+1** 個目に配置する箱を表示します。

オプション 整数。自動的に読み取ります。外部コマンドを受信した場合、コマンドに従って更新します。

パレットパターンの基本設定

経路を非表示

デフォルトではチェックが外れており、箱の進入経路を表示します。チェックすると箱の進入経路を非表示します。

対象物の数

パレット上の配置可能な対象物の数。編集できません。

動作制御

開始段階の強制関節運動

デフォルトではチェックが入っており、ロボットがパレタイジングする前の運動を関節運動にします。

進入/調整/配置段階の強制関節運動

デフォルトではチェック外れており、強制的に関節運動を設定しません。

空間が狭くて直線運動に設定する必要がある場合にチェックします。狭い現場では、進入/調整/配置段階の運動を関節運動に設定することで特異点を回避することが可能です。

加速度・速度のスケーリング比例

調整範囲 0~100%。初期値：100

使用シーン ロボットがパレットに近づく時と箱を配置する時の速度が異なる場合に使用します。

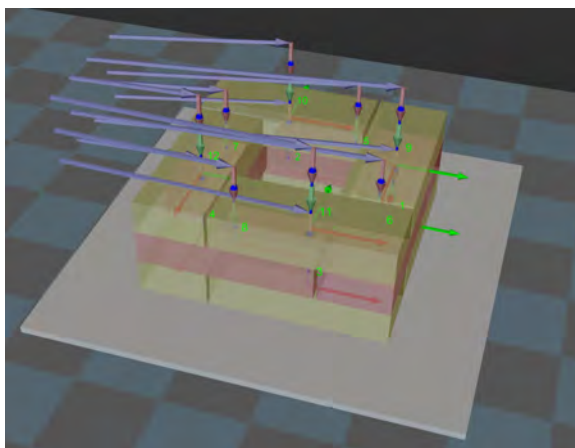
説明

箱を配置する時の加速度・速度。**加速度 & 速度 × 加速度 & 速度 & 速度のスケール比例**によって計算します。

パレットに近づく 3つの段階：

一段階目：紫（パレットに近づく）。二段階目：ピンク（箱を配置）。三段階目：緑（箱を配置）

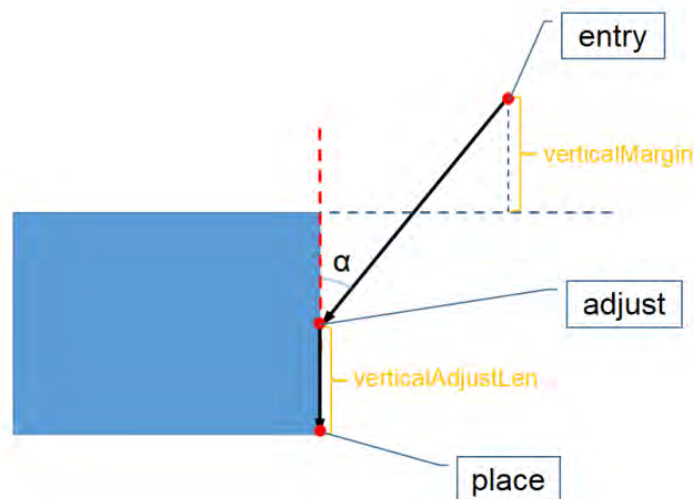
パレットに近づく速度と加速度は「基本的な移動設定」で指定されたあと、二、三段階目の速度/加速度は **加速度・速度 × 加速度・速度のスケール比例**。



進入と調整

3つのパラメータを設定して箱がパレットに進入する経路を指定します。箱が配置済みの箱の山に近づいてから垂直方向に沿って配置するような進入経路を調整します。これによって他の箱との衝突を回避できます。

箱ごとに、パレットにエントリーするには4つの位置があり、そのうち3つはこのパラメータグループによって制御されます。下図の赤い点に示すように、それらはentry、adjust、およびplaceです。下図の画像の視点は、箱を配置する正面図です。



垂直方向に長さの比例を調整

説明 調整する点（adjust）の位置を影響します。値 = verticalAdjustLen / 箱の高さ。

調整範囲 0~1

推奨値 0.5

垂直方向の範囲

説明 進入点（entry）の位置を指定します。値 = 高さの範囲。

調整範囲 0 ~ 無限大、単位：mm。実際に応じて調整してください。

Z方向の進入角度

説明 進入点（entry）から調整点（adjust）までの経路の垂直方向との角度を指定します（単位：°）。

調整範囲 -80°~80°

推奨値 30°~45°

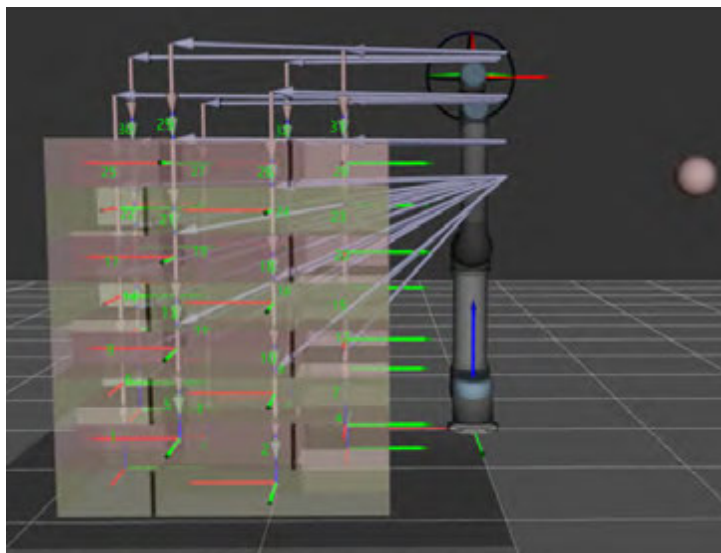
自動中間点

X/Y

ロボットベース座標系におけるピンクのボールの位置 x と y を指定します。この位置によって適切な中間点座標が算出されます。

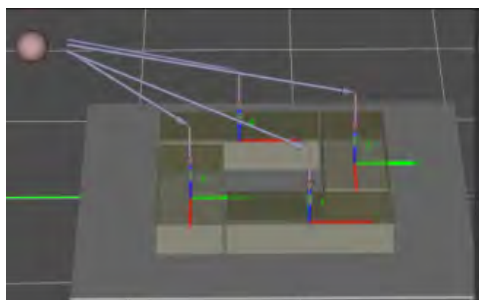
最小Z高さ

ロボットがパレットに進入する（紫の経路）時にZ方向の最小絶対高さ（Z方向高さはこの段の高さの差）。

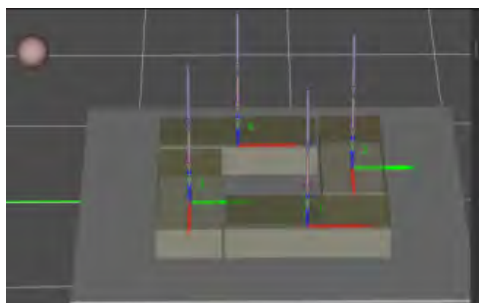


垂直な中間点の進入経路

デフォルトではチェックが外れており、進入段階は中間点方向に沿います。



チェックすると進入段階は中間点方向に沿わず、配置する位置の真上から進入します。



進入段階の延長距離

グリップが大きくて、進入段階が長いほど衝突発生確率は高くなる恐れがある現場に使用できません。安全に動作できるように進入段階を長くしてください。



自動中間点 はパレットに進入する方向だけを指定し、ロボットが到達する目標点を指定しないので、できるだけピンクのボールをパレットから遠ざけてください。

ビジョン処理によるパレットの位置を調整

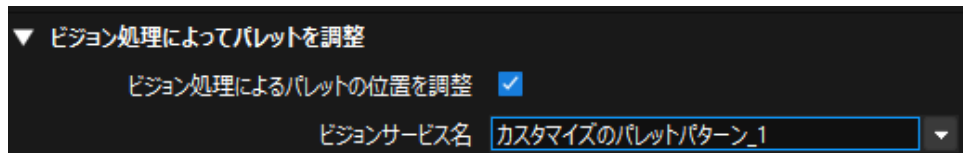
「ビジョン処理による認識」ステップを使用してパレットの位置を調整します。

ビジョン処理を使用してパレット位置を調整

デフォルトでチェックが外れています。チェックすると、動的にパレット位置を調整する必要がある場合にチェックします。「ビジョン処理による認識」ステップによって認識されたパレット位置を使用します。

ビジョンサービス名

パレット位置を認識するMech-Vision プロジェクト名（「ビジョン処理による認識」ステップ）を選択します。するとこのステップを実行するときに Mech-Vision プロジェクト名によって「ビジョン処理による認識」ステップを呼び出します。



パラメータ は外部設定（adapter）による調整も可能です。

デパレタイズされるパレットとして

ビジョンサービスとして

デフォルトではチェックが外れており、ビジョンサービスとして使用されません。チェックすると [ビジョン処理による認識](#) にビジョンサービスを提供し、カスタマイズのパレットパターンを生成します。すなわち、Mech-Vision からビジョン処理の結果を受信しない場合でも Mech-Viz でデパレタイジングをシミュレート・実行することができます。

動的に読み込む

デフォルトではチェックは外れています。チェックするとインターフェイスによって異なるパレットパターンを切り替えることができます。インターフェイスによって異なるファイルパスを設定すると異なるファイルのパレットパターンに基づいてパレタイジングを実行します。



オーダーメードの Adapter を使用しなければ動的読み込むことができません。set_task_property() 関数を使用してステップのパラメータ値を設定することでフォルダのパスとファイル名を設定します。

自動調整可能な水平方向のパレット侵入角度

デフォルトではチェックが外れています。チェックすると、手動で箱の進入角度を設定する必要がなく、事前設定したパレットパターンの進入角度を使用します。

パレットパターンエディタ

[パレットパターンエディタ](#) をご参照ください。編集されたパレットパターンは **.json** 形式でプロジェクトフォルダに保存されます。

パターンを読み込む

編集済みのパレットパターンを読み込みます。

5.4.4.2. 混載パレットパターン

機能

寸法が異なる対象物をパレタイズします。ビジョン処理による認識]ステップによって対象箱の寸法を認識して、かつパラメータを設定してパレットパターンを自動的に生成します。

使用シーン

寸法が異なる箱をパレタイズします。

パラメータ

移動ステップの共通パラメータ

移動目標点を送信

デフォルトではチェックが入っています。相手側（ロボットなど）に移動目標点の位置姿勢を送信します。チェックを外したら送信しませんが、この目標点位置姿勢は依然として経路計画の一部です。

移動コマンドの後の非移動コマンドをスムーズに実行することを試行

デフォルトではチェックが外れており、**移動ステップ** の間に「ビジョン処理による認識」、「DO を設定」、「DI をチェック」などの **非移動ステップ** を接続すると、ロボット移動経路の計画を中断し、ロボット実機動作中に一時中止することがあります。チェックを入れると現在のステップが完了しなくても続行することが可能です。これにより、ロボットがよりスムーズに動作します。ただし、ステップが途中で終了する可能性があります。



ステップが途中で終了する原因は何ですか？

Mech-Viz 実行中、ロボットに同時に複数の位置姿勢を送信しますが、最後の位置姿勢が現在のロボットの関節角度と一致しているかのみを判断します。一致していると判断したら、ロボットが最後の位置に到達していると考えられます。例えば、10 の移動ステップがある経路では、移動ステップ 5 の位置姿勢は最後のステップの位置姿勢とは同じとします。ロボットが低速に動作し、移動ステップ 5 の位置に到達したらその位置の関節角度を Mech-Viz に送信します。経路では、移動ステップ 5 の位置姿勢は最後の移動ステップの位置姿勢とは同じなので、Mech-Viz はロボットの動作が完了したと判断して途中で実行を終了します。

配置された対象物との衝突を検出しない

デフォルトではチェックが外れており、配置された対象物との衝突を検出しません。チェックを入れると、ロボット本体・ロボットハンドと配置された対象物との衝突を検出します。

パレタイジングのシーンでは、以下場合があります。

1. ロボットが段ボール箱を配置するときに配置済みの箱と軽く衝突する（箱の凹みや変形などが発生しない）ことがあります。Mech-Viz ではこのような衝突を検出したら別の配置位置を計画するためパレットが満杯になりません。
2. 普通、吸盤を使用する場合、TCP を吸盤の表面でなく、モデル内部に設定するため、物体を吸着する時、吸盤と把持する箱のモデルと重なります（ソフトウェアでは吸盤と把持対象物との衝突を検出しない）。ロボットが箱を配置してから、把持された箱のモデルはシーンのモデルになり、吸盤がシーンの箱のモデルと衝突すると判断してメッセージを表示してパレタイジングが続行できなくなります。

これをチェックするとロボット本体やロボットハンドと配置済み対象物モデルとの衝突を検出しないので上記の問題を解決できます。

点群との衝突検出モード

現場の状況に応じて設定してください。普通、デフォルトの **自動** を使用します。ロボットが物体を把持する前の移動ステップは、**チェックしない** を選択し、把持した後の移動ステップを **チェック** を選択します。

自動 初期値。「ビジョン処理による移動」ステップと「ビジョン処理による移動」ステップに依存する「相対移動」の点群衝突だけを検出します。

チェックしない 移動ステップの点群衝突をいっさい検出しません。

チェック 全ての移動ステップの点群衝突を検出します。



衝突検出、**衝突検出設定**、**点群と他の対象物間の衝突を検出** をオンにすれば、Mech-Viz は経路計画を行うときにロボットモデル、ツールモデルと点群との衝突を検出します。普通、点群衝突の設定は、ロボットが把持を実行するときに把持対象物との衝突を検出するためです。空間内にノイズがあれば、ソフトウェアは対象物を把持する前の経路を計画するときに、ノイズがロボットモデル、ツールモデルと接触して点群との衝突が誤って判断されて経路計画の誤りが発生します。

対象物の対称性を使用しない

このパラメータは、**目標点タイプ** が **対象物位置姿勢** の移動ステップ（目標点タイプが物体の位置姿勢の移動ステップ、パレタイジングステップなど）に対してのみ有効です。目標点のタイプが関節角度、TCP位置姿勢の移動ステップには無効です。

無し 無し：初期値。対称性を無効にしません。

- Z 軸** Z 軸の対称性のみを使用しません。
- XY 軸** X、Y 軸の対称性のみを使用しません。
- 全て** 全ての対称性を使用しません。

物体の対称性を使用しないように設定すると、ロボットが物体の位置姿勢に正確に到達して対象物を配置します。



対象物を把持できない場合に **リソース** > **対象物設定** の **回転対称** を設定します。対象物に対称性がある場合、複数の候補位置姿勢があります。Mech-Viz では対象物の把持を計画するときに、デフォルト位置姿勢が把持できない場合、候補位置姿勢を試行します。物体対称位置姿勢と Mech-Vision によって出力された元位置姿勢とは一致しなければ、ロボットによる対象物の配置位置姿勢は一致しないことがあります。

インデックス

開始インデックス

- 説明** これから配置する箱のインデックス。
- オプション** 整数にしか設定できません。初期値は 0。
- 使用方法** 空のパレットの場合に **0** に設定します。
- パレタイジングを継続するとき、N 個の箱を配置した場合に値を N に設置します。これで N+1 個目の箱からパレタイジングを継続します。

現在のインデックス

- 説明** 箱の位置を表示します。値が N であれば、**N+1** 個目に配置する箱を表示します。
- オプション** 整数。自動的に読み取ります。外部コマンドを受信した場合、コマンドに従って更新します。

パレットパターンの基本設定

経路を非表示

デフォルトではチェックが外れており、箱の進入経路を表示します。チェックすると箱の進入経路を非表示します。

対象物の数

パレット上の配置可能な対象物の数。編集できません。

動作制御

開始段階の強制関節運動

デフォルトではチェックが入っており、ロボットがパレタイジングする前の運動を関節運動にします。

進入/調整/配置段階の強制関節運動

デフォルトではチェック外れており、強制的に関節運動を設定しません。

空間が狭くて直線運動に設定する必要がある場合にチェックします。狭い現場では、進入/調整/配置段階の運動を関節運動に設定することで特異点を回避することが可能です。

加速度・速度のスケーリング比例

調整範囲 0~100%。初期値：100

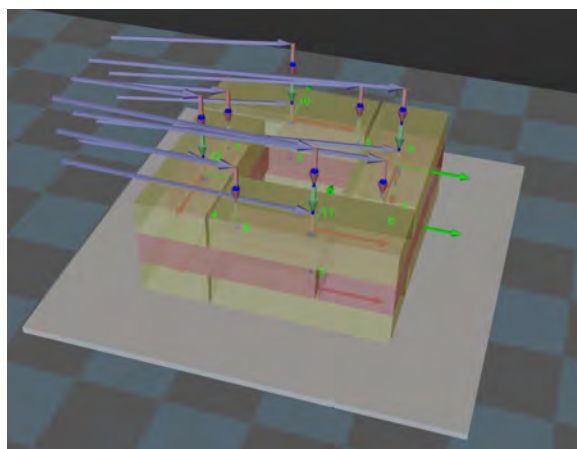
使用シーン ロボットがパレットに近づく時と箱を配置する時の速度が異なる場合に使用します。

説明 箱を配置する時の加速度・速度。**加速度 & 速度 × 加速度 & 速度 & 速度のスケーリング比例** によって計算します。

パレットに近づく 3つの段階：

一段階目：紫（パレットに近づく）。二段階目：ピンク（箱を配置）。三段階目：緑（箱を配置）

パレットに近づく速度と加速度は「基本的な移動設定」で指定されたあと、二、三段階目の速度/加速度は **加速度・速度 × 加速度・速度のスケーリング比例**。

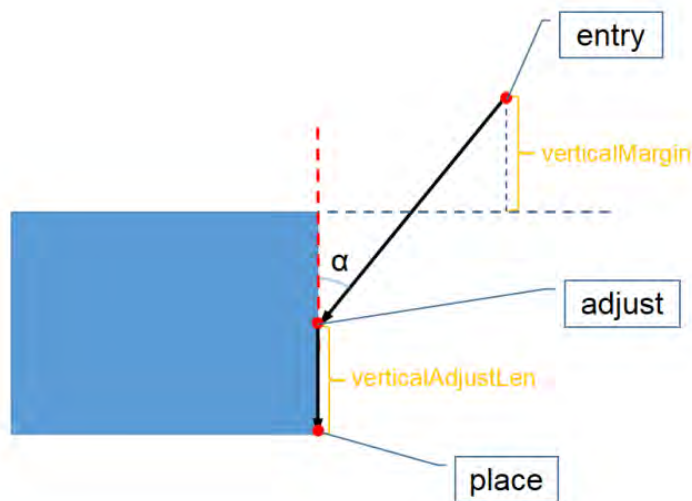


進入と調整

3つのパラメートを設定して箱がパレットに進入する経路を指定します。箱が配置済みの箱の山

に近づいてから垂直方向に沿って配置するような進入経路を調整します。これによって他の箱との衝突を回避できます。

箱ごとに、パレットにエントリーするには4つの位置があり、そのうち3つはこのパラメータグループによって制御されます。下図の赤い点に示すように、それらはentry、adjust、およびplaceです。下図の画像の視点は、箱を配置する正面図です。



垂直方向に長さの比例を調整

説明 調整する点（adjust）の位置を影響します。値 = verticalAdjustLen / 箱の高さ。

調整範囲 0~1

推奨値 0.5

垂直方向の範囲

説明 進入点（entry）の位置を指定します。値 = 高さの範囲。

調整範囲 0 ~ 無限大、単位：mm。 実際に応じて調整してください。

Z 方向の進入角度

説明 進入点（entry）から調整点（adjust）までの経路の垂直方向との角度を指定します（単位：°）。

調整範囲 -80°~80°

推奨値 30°~45°

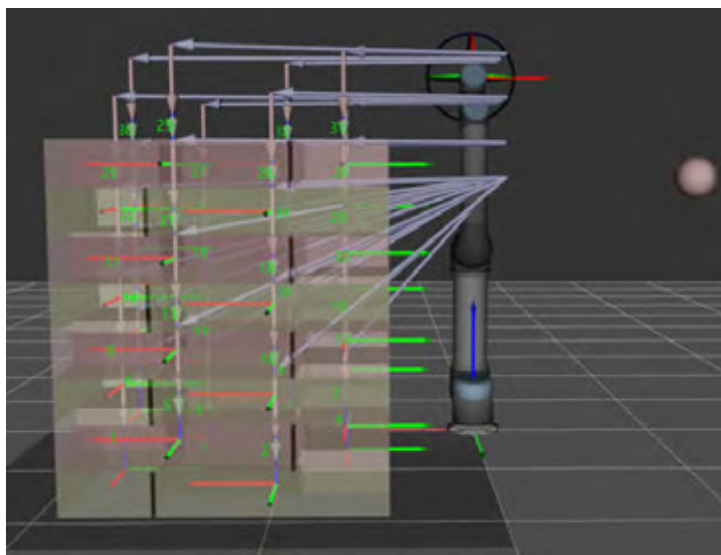
自動中間点

X/Y

ロボットベース座標系におけるピンクのボールの位置 x と y を指定します。この位置によって適切な中間点座標が算出されます。

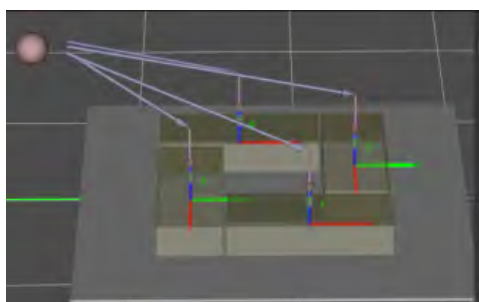
最小Z高さ

ロボットがパレットに進入する（紫の経路）時にZ方向の最小絶対高さ（Z方向高さはこの段の高さの差）。

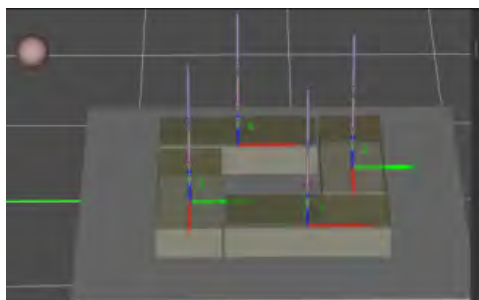


垂直な中間点の進入経路

デフォルトではチェックが外れており、進入段階は中間点方向に沿います。



チェックすると進入段階は中間点方向に沿わず、配置する位置の真上から進入します。



進入段階の延長距離

グリッパが大きくて、進入段階が長いほど衝突発生確率は高くなる恐れがある現場に使用できません。安全に動作できるように進入段階を長くしてください。



自動中間点 はパレットに進入する方向だけを指定し、ロボットが到達する目標点を指定しないので、できるだけピンクのボールをパレットから遠ざけてください。

ビジョン処理によるパレットの位置を調整

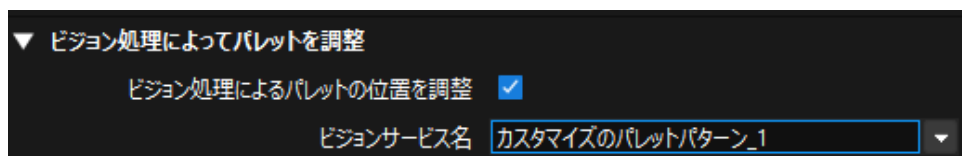
「ビジョン処理による認識」ステップを使用してパレットの位置を調整します。

ビジョン処理を使用してパレット位置を調整

デフォルトでチェックが外れています。チェックすると、動的にパレット位置を調整する必要がある場合にチェックします。「ビジョン処理による認識」ステップによって認識されたパレット位置を使用します。

ビジョンサービス名

パレット位置を認識するMech-Vision プロジェクト名（「ビジョン処理による認識」ステップ）を選択します。するとこのステップを実行するときに Mech-Vision プロジェクト名によって「ビジョン処理による認識」ステップを呼び出します。



パラメータ は外部設定（adapter）による調整も可能です。

パレットのサイズ

パレット X 辺長	パレットの長さを設定する
パレット Y 辺長	パレットの幅を設定する
パレット高さ	パレットパターンの最大高さを設定する
パレットの高さ制限を超える値の許容範囲	：箱を配置する時にパレット高さをを超える値の許容範囲。

例：

パレット高さを 600mm に、パレットの高さ制限を超える値の許容範囲を 100mm に設定すると、パレットの最大高さは $600\text{mm} + 100\text{mm} = 700\text{mm}$ になります。現在の箱の山の高さは 450mm の場合、これから続けて箱を配置できる高さは $700\text{mm} - 450\text{mm} = 250\text{mm}$ になります。

高さが 250mm 以上の箱：配置できません

- 高さが 250mm 以下の箱：配置できる

パレットパターン

Online これから配置する箱の寸法がわからない場合に選択する。次の一個の箱を配置する経路を計画する

Offline これから配置する箱の寸法が全部わかる場合に選択する。一回にすべての箱を配置する経路を計画する



- デバッグする時にパレットパターン関連パラメータを調整するために使用します。**json** ファイルを読み込みます。
- ロボット実機の実行をサポートしていません。

Online

パレットパターンの設定

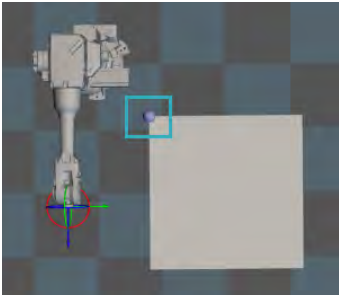
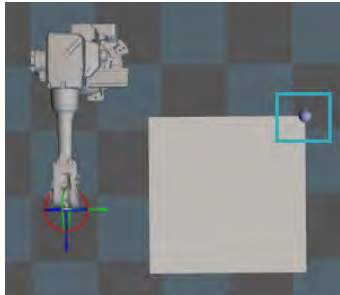
箱の最短間隔

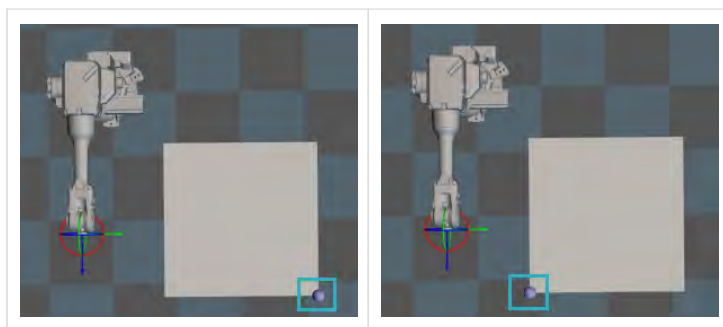
箱と箱の間の隙間の幅、mm を単位とします。衝突を回避するために使用します。推奨値：10 ~ 20mm。

優先コーナー

説明：配置する時に優先的に配置するパレットのコーナー。パレットの位置姿勢の変化とともに変化します。

- OO** パレットの基準コーナー。ロボット座標系の -Y、-X 方向にあるコーナー
- OY** OO コーナーを基準に、ロボット座標系の Y 方向にあるコーナー
- XY** OO コーナーを基準に、ロボット座標系の Y 方向と X 方向にあるコーナー
- XO** OO コーナーを基準に、ロボット座標系の X 方向にあるコーナー

OO	OY
	
XY	XO



落下配置

落下配置を有効にする

デフォルトではチェックが外れており、落下配置を有効にしません。

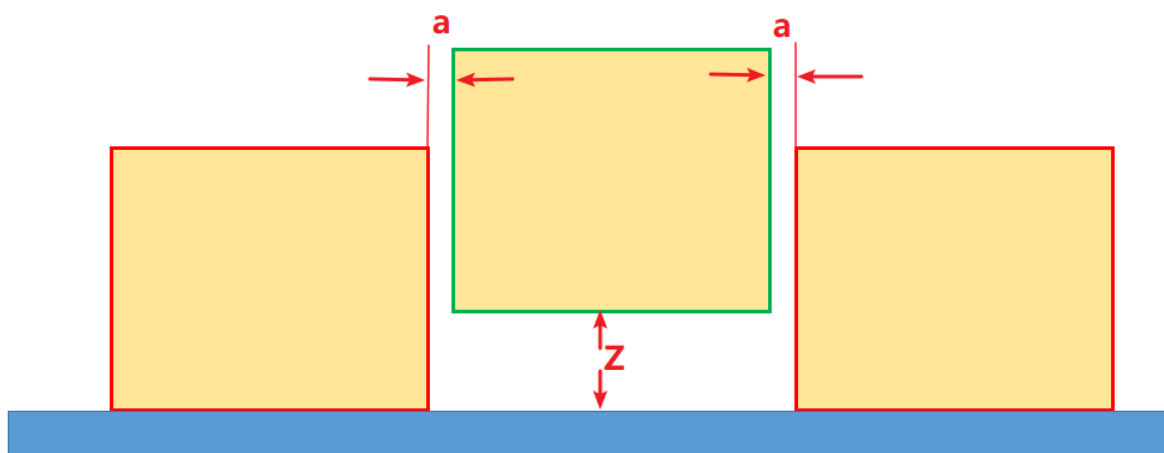
チェックすると、箱が配置位置から Z 方向に沿う距離がある範囲にある場合に、落下によって配置することができます。

使用シーン 低い箱を高い箱の間に配置するときに落下させる。これにより、衝突を回避可能

落下高さ 落下配置を有効にする をチェックすると表示される。箱を落下させる高さを設定する（単位：mm）

安全落下間隔 配置する箱の側面と隣接する箱の最小間隔（単位：mm）

下図では Z は **落下高さ** であり、a は **安全落下間隔** です。



再撮影して箱の寸法を更新

デフォルトでチェックが外れています。チェックすると、再撮影します。一回の撮影で完全な寸法を取得できないシーンにチェックします。

ビジョン処理による移動ステップは箱を把持したが箱の高さの情報を取得していない場合に、配置位置と重みを計算する時に箱の高さも推定し、推定された高さによって計画します。再撮影して箱の高さを取得してから2回目に計画します。



「把持済み対象物を更新」ステップと合わせて使用することができます。

候補位置の数の制限

候補位置の合計数の制限

初期値 -1。有効にしない。数を制限しない

説明 経路チェックの時間を削減するために、すべての箱の候補位置の数を制限する

単一の箱の候補位置の数の制限

初期値 -1。有効にしない。単一の箱の候補位置の数を制限しない

説明 経路チェックの時間を削減するために、単一の箱の候補位置の数を制限する

ラベル付き箱

箱のラベルを外に向ける

デフォルトでチェックが外れています。この機能を使用しません。

チェックすると、側面にラベルが付いている、かつラベルを外に向けるように配置するシーンに使用します。

ラベルのパレットのエッジまでの最大距離：箱のラベルがパレットエッジまでの距離の最大許容値。単位：mm。

候補位置のパラメータを計算

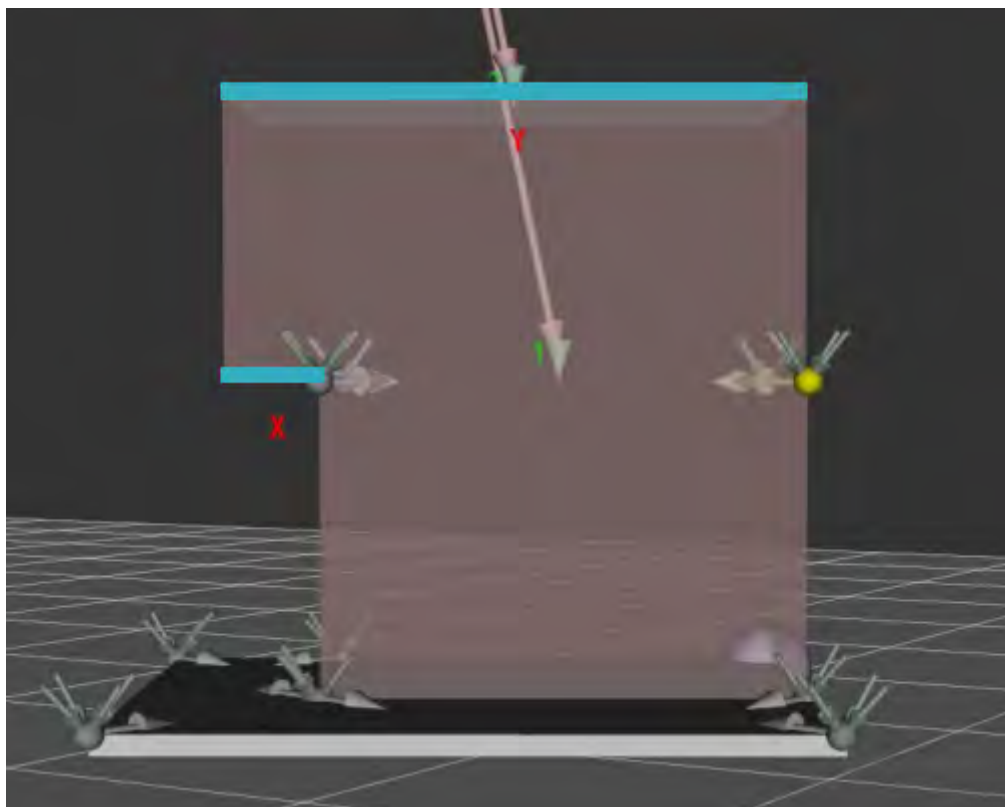
パレットエッジを超える幅の許容範囲

箱がパレットのエッジをはみ出す幅の許容範囲。推奨値：20～50mm。



箱が平面を超える比例の許容範囲

箱がその下の平面をはみ出す面積の最大比例。



段の平面高低差の最大値

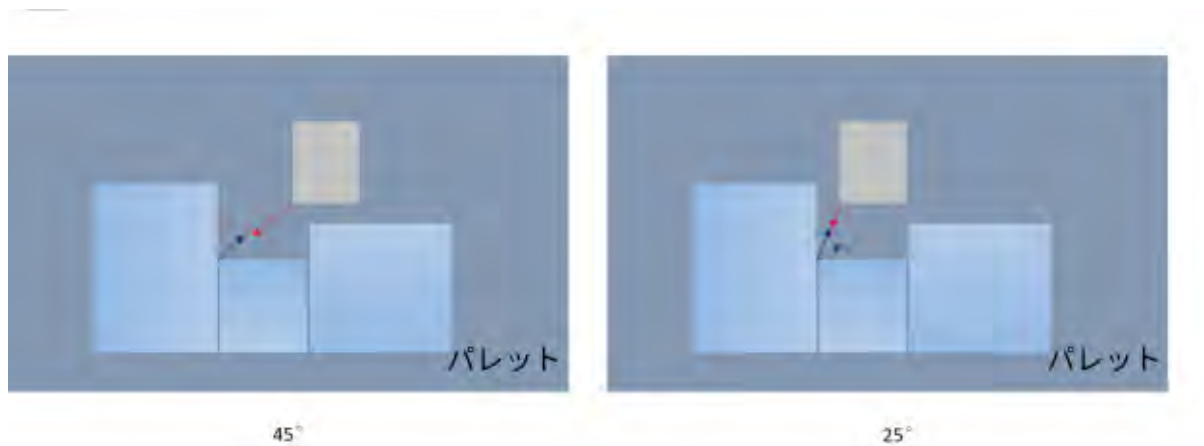
平面高低差がこの値より小さい段を平面と見なして箱を配置します。

サンプリング率

説明：箱の位置を計画する時のサンプリング率（sample/meter）です。サンプリング率が高いほど結果は精確になりますが、速度は遅くなります。推奨値：200、500、1000。

進入方向の X 軸角度

説箱の進入方向のXOY平面の投影と世界座標系X軸との角度です。



値をあんまり大きく設定すると、U字型の空き領域ができるかもしれません。あんまり小さく設定すると、箱の衝突が発生する恐れがあります。

推奨値：15°~30°。

中間点を使用

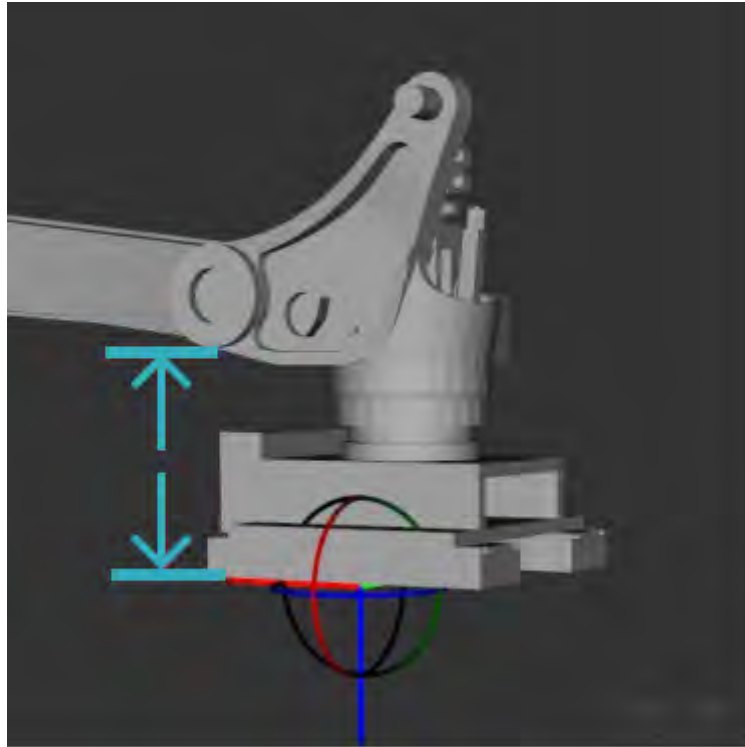
チェックするとコーナーとコーナーとの中点も候補位置とします。

コーナー安全半径

コーナーへの到達方向を決定するとき、この半径にあるコーナーに到達する指定方向がXOY投影平面に障害物があるかをチェックします。

グリッパーのZ方向高さ

グリッパーの底面からフランジから二番目のロボット関節の底面までのZ方向の距離。単位：mm。



中点安全間隔

箱を中間点候補位置に配置する時に箱の両側の隙間隔。単位：mm。



ロボットの運動誤差と箱の寸法誤差に許容されれば、このパラメータを間隔よりも小さい値に設定してもかまいません。

候補位置スコアの重み

このグループのパラメータを設定することでパレットに配置する箱の位置を指定します。

隣接面積

値が高いほど、側面が隣接する箱との接触する面積が大きい候補位置は使用される可能性は高くなります。

荷重面積

値が高いほど、配置する箱の荷重がかかわる面積が大きい（下の表面をはみ出す領域が狭い）位置が使用される可能性は高くなります。

箱の配置位置の高さ

配置する箱の底面からパレットまでのZ方向に沿う高さの差。値が高く、かつ目標位置が低いほど、箱が低い平面に配置される可能性は高くなります。

優先コーナーまでの投影距離

値が高いほど、優先コーナーまでの距離がパレットの対角線に投影する長さが短い位置が使用される可能性は高くなります。

荷重箱の数

値が高いほど、荷重をかかわる下段の箱の数が多い位置が使用される可能性は高くなります。この値を高く設定するとより安定的なパレットパターンを生成できますが空き領域が出てきます。値を調整するとき、倍にすることを推奨します。



例えば箱を優先コーナーに配置したい場合、**コーナーまでの投影距離**を倍にして効果を確かめてください。

底の面積の重み

値が高いほど、底面が広い箱が配置される可能性は高くなります。

パレットのエッジまでの距離

値が高いほど、箱をパレットのエッジに近づいて配置します。

Offline

パレットパターンの設定

箱の最短間隔

箱と箱の間隙の幅、mm を単位とします。

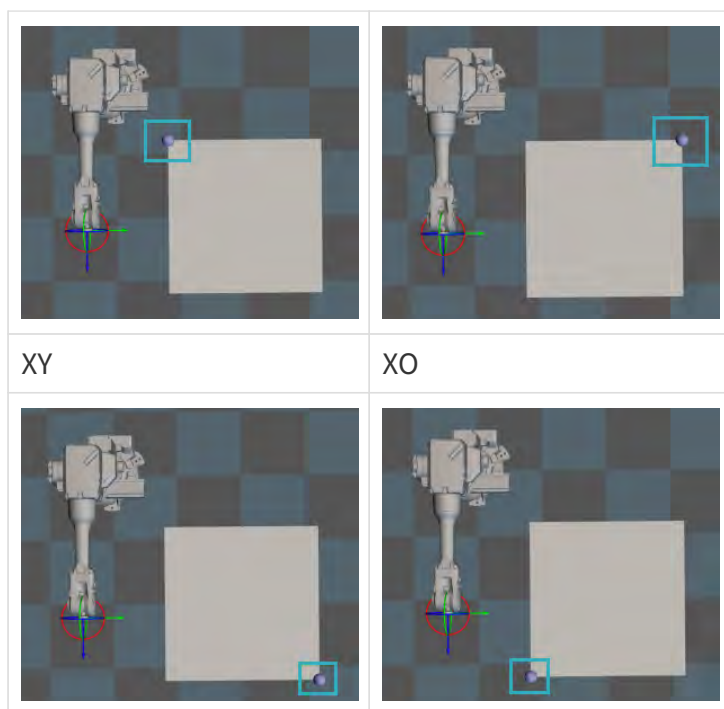
推奨値：10 ~ 20mm。

優先コーナー

説明：配置する時に優先的に配置するパレットのコーナー。パレットの位置姿勢の変化とともに変化します。

- 00** パレットの基準コーナー。ロボット座標系の -Y、-X 方向にあるコーナー
- OY** 00 コーナーを基準に、ロボット座標系の Y 方向にあるコーナー
- XY** 00 コーナーを基準に、ロボット座標系の Y 方向と X 方向にあるコーナー
- XO** 00 コーナーを基準に、ロボット座標系の X 方向にあるコーナー

00	OY
----	----



オフラインメソッド

- ベストフィット** 箱の寸法が大きく異なる場合やインラインに設定した場合に適用。レットパターンはほかの3種類ほど規則的ではない
- スタックごとに** 箱の寸法などによってスタックごとに配置する
- 段ごとに** 優先的に寸法が同じな箱を同じ段に配置する
- バッチごとに** SKU ごとに配置する。このようにして、AGV を使用する現場では、AGV の供給回数を削減可能

パレットのインデックス

説明：パレットを指定します。初期値：-1。

箱の最短辺長

入力された辺長がこの値より小さいときにプロンプトメッセージが表示されます。単位：mm。箱の寸法が小さい場合、オフライン混載パレタイジングの計画時間は長くなります。このパラメータを設定することで入力の間違いによって計算時間が長くなるのを防ぎます。

5.4.4.3. 複数把持のパレタイジング

機能

ターゲットパレットパターンのパレタイズ・デパレタイズ位置の箱の組合せ結果によって複数把持のパレタイジング経路を自動的に算出します。これによって、パレタイジングの効率向上が可能です。

使用シーン

ほとんどのパレタイジングシーンに適用されます。

パラメータの説明

移動ステップの共通パラメータ

移動目標点を送信

デフォルトではチェックが入っています。相手側（ロボットなど）に移動目標点の位置姿勢を送信します。チェックを外したら送信しませんが、この目標点位置姿勢は依然として経路計画の一部です。

移動コマンドの後の非移動コマンドをスムーズに実行することを試行

デフォルトではチェックが外れており、**移動ステップ**の間に「ビジョン処理による認識」、「DOを設定」、「DIをチェック」などの**非移動ステップ**を接続すると、ロボット移動経路の計画を中断し、ロボット実機動作中に一時中止することがあります。チェックを入れると現在のステップが完了しなくても続行することが可能です。これにより、ロボットがよりスムーズに動作します。ただし、ステップが途中で終了する可能性があります。

ステップが途中で終了する原因は何ですか？



Mech-Viz 実行中、ロボットに同時に複数の位置姿勢を送信しますが、最後の位置姿勢が現在のロボットの関節角度と一致しているかのみを判断します。一致していると判断したら、ロボットが最後の位置に到達していると見なします。例えば、10の移動ステップがある経路では、移動ステップ5の位置姿勢は最後のステップの位置姿勢とは同じとします。ロボットが低速に動作し、移動ステップ5の位置に到達したらその位置の関節角度をMech-Vizに送信します。経路では、移動ステップ5の位置姿勢は最後の移動ステップの位置姿勢とは同じなので、Mech-Vizはロボットの動作が完了したと判断して途中で実行を終了します。

配置された対象物との衝突を検出しない

デフォルトではチェックが外れており、配置された対象物との衝突を検出しません。チェックを入れると、ロボット本体・ロボットハンドと配置された対象物との衝突を検出します。

パレタイジングのシーンでは、以下場合があります。

1. ロボットが段ボール箱を配置するときに配置済みの箱と軽く衝突する（箱の凹みや変形などが発生しない）ことがあります。Mech-Vizではこのような衝突を検出したら別の配置位置を計画するためパレットが満杯になりません。
2. 普通、吸盤を使用する場合、TCPを吸盤の表面でなく、モデル内部に設定するため、物体を吸着する時、吸盤と把持する箱のモデルと重なります（ソフトウェアでは吸盤と把持対象物との衝突を検出しない）。ロボットが箱を配置してから、把持された箱のモデルはシーンのモデルになり、吸盤がシーンの箱のモデルと衝突すると判断してメッセージを表示してパレタイジングが続行できなくなります。

これをチェックするとロボット本体やロボットハンドと配置済み対象物モデルとの衝突を検出しないので上記の問題を解決できます。

点群との衝突検出モード

現場の状況に応じて設定してください。普通、デフォルトの **自動** を使用します。ロボットが物体を把持する前の移動ステップは、**チェックしない** を選択し、把持した後の移動ステップを **チェック** を選択します。

自動 初期値。「ビジョン処理による移動」ステップと「ビジョン処理による移動」ステップに依存する「相対移動」の点群衝突だけを検出します。

チェックしない 移動ステップの点群衝突をいっさい検出しません。

チェック 全ての移動ステップの点群衝突を検出します。



衝突検出、**衝突検出設定**、**点群と他の対象物間の衝突を検出** をオンにすれば、Mech-Viz は経路計画を行うときにロボットモデル、ツールモデルと点群との衝突を検出します。普通、点群衝突の設定は、ロボットが把持を実行するときに把持対象物との衝突を検出するためです。空間内にノイズがあれば、ソフトウェアは対象物を把持する前の経路を計画するときに、ノイズがロボットモデル、ツールモデルと接触して点群との衝突が誤って判断されて経路計画の誤りが発生します。

対象物の対称性を使用しない

このパラメータは、**目標点タイプ** が **対象物位置姿勢** の移動ステップ（目標点タイプが物体の位置姿勢の移動ステップ、パレタイジングステップなど）に対してのみ有効です。目標点のタイプが関節角度、TCP位置姿勢の移動ステップには無効です。

無し 無し：初期値。対称性を無効にしません。

Z 軸 Z 軸の対称性のみを使用しません。

XY 軸 X、Y 軸の対称性のみを使用しません。

全て 全ての対称性を使用しません。

物体の対称性を使用しないように設定すると、ロボットが物体の位置姿勢に正確に到達して対象物を配置します。



対象物を把持できない場合に **リソース**、**対象物設定** の **回転対称** を設定します。対象物に対称性がある場合、複数の候補位置姿勢があります。Mech-Viz では対象物の把持を計画するときに、デフォルト位置姿勢が把持できない場合、候補位置姿勢を試行します。物体対称位置姿勢と Mech-Vision によって出力された元位置姿勢とは一致しなければ、ロボットによる対象物の配置位置姿勢は一致しないことがあります。

インデックス

開始インデックス

説明 これから配置する箱のインデックス。

オプション 整数にしか設定できません。初期値は 0。

使用方法 空のパレットの場合に **0** に設定します。

パレタイジングを継続するとき、N 個の箱を配置した場合に値を N に設置します。これで N+1 個目の箱からパレタイジングを継続します。

現在のインデックス

説明 箱の位置を表示します。値が N であれば、**N+1** 個目に配置する箱を表示します。

オプション 整数。自動的に読み取ります。外部コマンドを受信した場合、コマンドに従って更新します。

パレットパターンの基本設定

経路を非表示

デフォルトではチェックが外れており、箱の進入経路を表示します。チェックすると箱の進入経路を非表示します。

対象物の数

パレット上の配置可能な対象物の数。編集できません。

動作制御

開始段階の強制関節運動

デフォルトではチェックが入っており、ロボットがパレタイジングする前の運動を関節運動にします。

進入/調整/配置段階の強制関節運動

デフォルトではチェック外れており、強制的に関節運動を設定しません。

空間が狭くて直線運動に設定する必要がある場合にチェックします。狭い現場では、進入/調整/配置段階の運動を関節運動に設定することで特異点を回避することが可能です。

加速度・速度のスケーリング比例

調整範囲 0~100%。初期値： 100

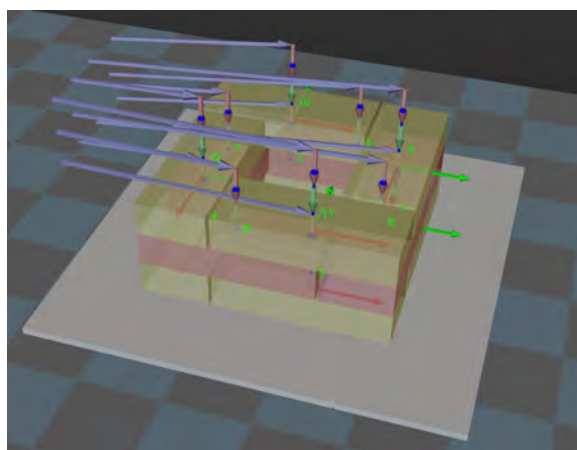
使用シーン ロボットがパレットに近づく時と箱を配置する時の速度が異なる場合に使用します。

説明 箱を配置する時の加速度・速度。**加速度 & 速度 × 加速度 & 速度 & 速度のスケーリング比例** によって計算します。

パレットに近づく 3つの段階：

一段階目：紫（パレットに近づく）。二段階目：ピンク（箱を配置）。三段階目：緑（箱を配置）

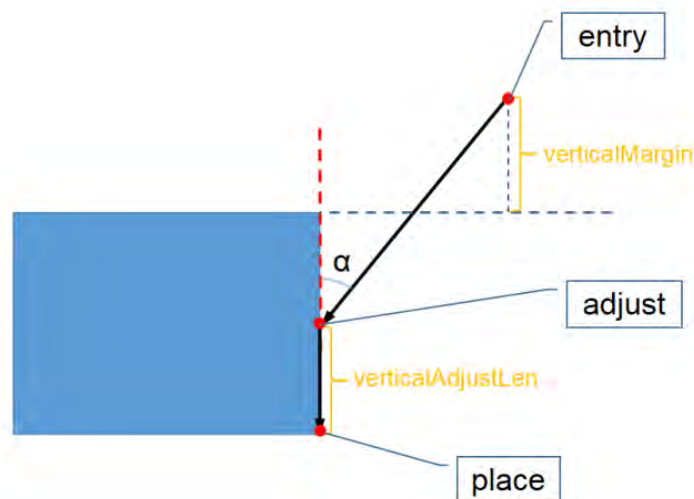
パレットに近づく速度と加速度は「基本的な移動設定」で指定されたあと、二、三段階目の速度/加速度は **加速度・速度 × 加速度・速度のスケーリング比例**。



進入と調整

3つのパラメータを設定して箱がパレットに進入する経路を指定します。箱が配置済みの箱の山に近づいてから垂直方向に沿って配置するような進入経路を調整します。これによって他の箱との衝突を回避できます。

箱ごとに、パレットにエントリーするには4つの位置があり、そのうち3つはこのパラメータグループによって制御されます。下図の赤い点に示すように、それらはentry、adjust、およびplaceです。下図の画像の視点は、箱を配置する正面図です。



垂直方向に長さの比例を調整

説明 調整する点（adjust）の位置を影響します。値 = verticalAdjustLen / 箱の高さ。

調整範囲 0~1

推奨値 0.5

垂直方向の範囲

説明 進入点（entry）の位置を指定します。値 = 高さの範囲。

調整範囲 0 ~ 無限大、単位：mm。実際に応じて調整してください。

Z 方向の進入角度

説明 進入点（entry）から調整点（adjust）までの経路の垂直方向との角度を指定します（単位：°）。

調整範囲 -80°~80°

推奨値 30°~45°

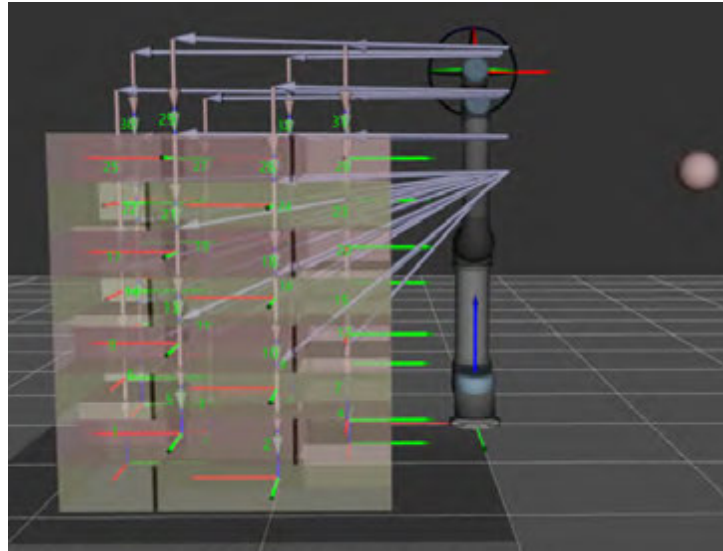
自動中間点

X/Y

ロボットベース座標系におけるピンクのボールの位置 x と y を指定します。この位置によって適切な中間点座標が算出されます。

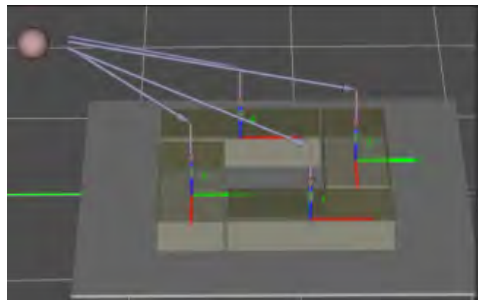
最小 Z 高さ

ロボットがパレットに進入する（紫の経路）時に Z 方向の最小絶対高さ（Z 方向高さはこの段の高さの差）。

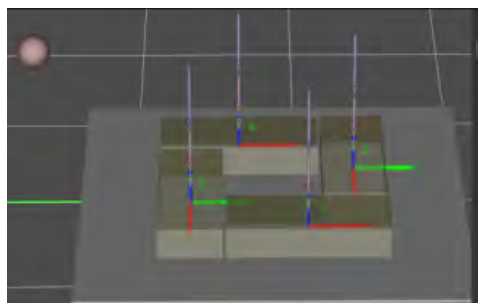


垂直な中間点の進入経路

デフォルトではチェックが外れており、進入段階は中間点方向に沿います。



チェックすると進入段階は中間点方向に沿わず、配置する位置の真上から進入します。



進入段階の延長距離

グリップが大きくて、進入段階が長いほど衝突発生確率は高くなる恐れがある現場に使用できません。安全に動作できるように進入段階を長くしてください。



自動中間点 はパレットに進入する方向だけを指定し、ロボットが到達する目標点を指定しないので、できるだけピンクのボールをパレットから遠ざけてください。

ビジョン処理によるパレットの位置を調整

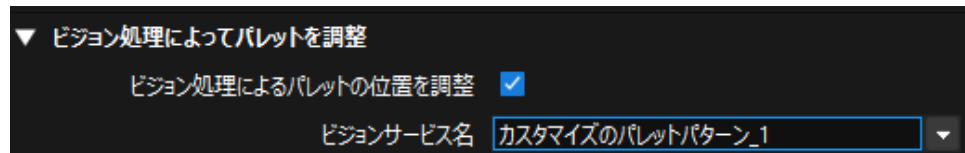
「ビジョン処理による認識」ステップを使用してパレットの位置を調整します。

ビジョン処理を使用してパレット位置を調整

デフォルトでチェックが外れています。チェックすると、動的にパレット位置を調整する必要がある場合にチェックします。「ビジョン処理による認識」ステップによって認識されたパレット位置を使用します。

ビジョンサービス名

パレット位置を認識するMech-Vision プロジェクト名（「ビジョン処理による認識」ステップ）を選択します。するとこのステップを実行するときに Mech-Vision プロジェクト名によって「ビジョン処理による認識」ステップを呼び出します。

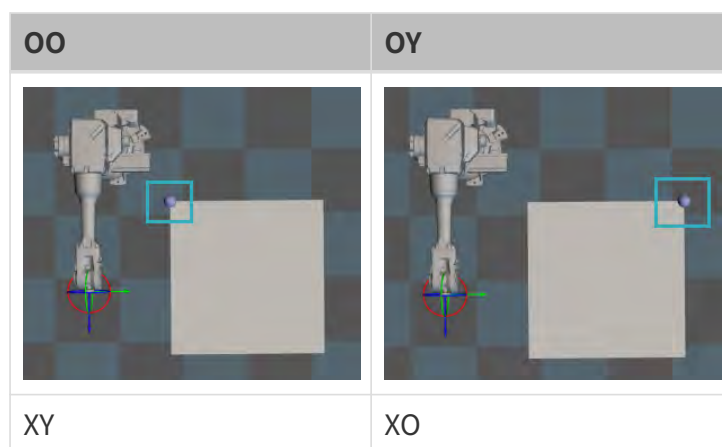


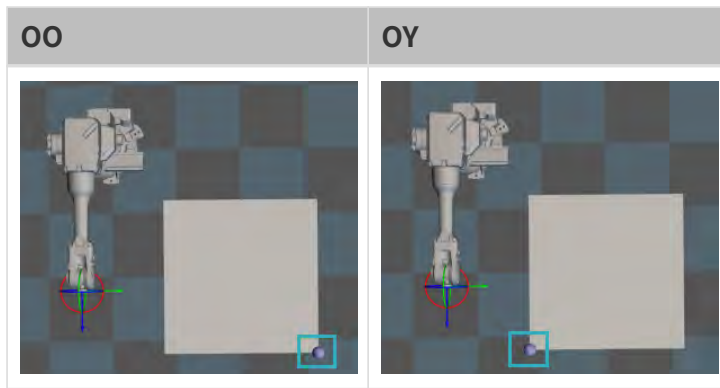
パラメータ は外部設定（adapter）による調整も可能です。

優先コーナー

パレタイジングを実行する時に、優先に使用するパレットのコーナーを指します。優先コーナーはパレットの位置姿勢とともに変わります。

- OO：ロボット座標系の-Y、-Xにあるコーナーを基準コーナーとします。
- OY：OOを基準コーナーとして、ロボット座標系のY正方向にあるコーナーです。
- XY：OOを基準コーナーとして、ロボット座標系のX・Y正方向にあるコーナーです。
- XO：OOを基準コーナーとして、ロボット座標系のX正方向にあるコーナーです。





箱の対称性なし

パレタイジングを実行するときに対称性を使用しません。

配置されたパレットパターンを再調整

デフォルトでは、配置済みの箱は移動できます。パレタイジング完了後、すべての箱の衝突モデルはターゲットパレットパターンの箱の位置と一致するように調整されます。

誤差マッチングのしきい値

把持される箱の組合せの各箱の座標と配置する位置の座標がXY平面における最大偏差値です。

この値を超えたらマッチングが失敗します。


パレタイジング記録の保存と読み込み

パレットパターンの記録と継続パレタイジングに対応します。

パレタイジングタスクにパレットパターンがなければ、json ファイルに保存されているパレットパターンを使用します。json ファイルにもパレットパターンがなければエラーメッセージが表示されます。

パレタイジングタスクにパレットパターンがあれば、それを使用して json ファイルに保存します。

パレットパターン情報のファイルパス

 をクリックして保存された json ファイルを選択します。

選択したオフラインパレット

パレタイジングに使用されるパレットパターンを選択します。**複数把持のパレタイジング**はパレットパターンを生成することはできないため、外部からパレットパターンを取得しなければなりません。

例

複数把持のパレタイジングと **ステップをリセット** と合わせて使用するとき、前回のパレタイジングの結果を放棄し、**選択したオフラインパレット** からパレットパターンを再取得します。

5.4.4.4. 事前計画パレットパターン

機能

よく使うパレットパターンを選択してパレタイジングを実行します。パレットパターンのエディタです。これにより、パレットパターンが自動的に生成されるため、使いやすく、手動で調整する必要はありません。

使用シーン

パレットが指定されて、ユーザーがパレットパターンを設定します。

パラメータ

移動ステップの共通パラメータ

移動目標点を送信

デフォルトではチェックが入っています。相手側（ロボットなど）に移動目標点の位置姿勢を送信します。チェックを外したら送信しませんが、この目標点位置姿勢は依然として経路計画の一部です。

移動コマンドの後の非移動コマンドをスムーズに実行することを試行

デフォルトではチェックが外れており、**移動ステップ**の間に「ビジョン処理による認識」、「DOを設定」、「DIをチェック」などの**非移動ステップ**を接続すると、ロボット移動経路の計画を中断し、ロボット実機動作中に一時中止することがあります。チェックを入れると現在のステップが完了しなくても続行することが可能です。これにより、ロボットがよりスムーズに動作します。ただし、ステップが途中で終了する可能性があります。



ステップが途中で終了する原因は何ですか？

Mech-Viz 実行中、ロボットに同時に複数の位置姿勢を送信しますが、最後の位置姿勢が現在のロボットの関節角度と一致しているかのみを判断します。一致していると判断したら、ロボットが最後の位置に到達していると考えられます。例えば、10の移動ステップがある経路では、移動ステップ5の位置姿勢は最後のステップの位置姿勢とは同じとします。ロボットが低速に動作し、移動ステップ5の位置に到達したらその位置の関節角度をMech-Vizに送信します。経路では、移動ステップ5の位置姿勢は最後の移動ステップの位置姿勢とは同じなので、Mech-Vizはロボットの動作が完了したと判断して途中で実行を終了します。

配置された対象物との衝突を検出しない

デフォルトではチェックが外れており、配置された対象物との衝突を検出しません。チェックを入れると、ロボット本体・ロボットハンドと配置された対象物との衝突を検出します。

パレタイジングのシーンでは、以下の場合があります。

1. ロボットが段ボール箱を配置するときに配置済みの箱と軽く衝突する（箱の凹みや変形など

が発生しない) ことがあります。Mech-Viz ではこのような衝突を検出したら別の配置位置を計画するためパレットが満杯になりません。

2. 普通、吸盤を使用する場合、TCP を吸盤の表面でなく、モデル内部に設定するため、物体を吸着する時、吸盤と把持する箱のモデルと重なります (ソフトウェアでは吸盤と把持対象物との衝突を検出しない)。ロボットが箱を配置してから、把持された箱のモデルはシーンのモデルになり、吸盤がシーンの箱のモデルと衝突すると判断してメッセージを表示してパレタイジングが続行できなくなります。

これをチェックするとロボット本体やロボットハンドと配置済み対象物モデルとの衝突を検出しないので上記の問題を解決できます。

点群との衝突検出モード

現場の状況に応じて設定してください。普通、デフォルトの **自動** を使用します。ロボットが物体を把持する前の移動ステップは、**チェックしない** を選択し、把持した後の移動ステップを **チェック** を選択します。

自動 初期値。「ビジョン処理による移動」ステップと「ビジョン処理による移動」ステップに依存する「相対移動」の点群衝突だけを検出します。

チェックしない 移動ステップの点群衝突をいっさい検出しません。

チェック 全ての移動ステップの点群衝突を検出します。



衝突検出、**衝突検出設定**、**点群と他の対象物間の衝突を検出** をオンにすれば、Mech-Viz は経路計画を行うときにロボットモデル、ツールモデルと点群との衝突を検出します。普通、点群衝突の設定は、ロボットが把持を実行するときに把持対象物との衝突を検出するためです。空間内にノイズがあれば、ソフトウェアは対象物を把持する前の経路を計画するときに、ノイズがロボットモデル、ツールモデルと接触して点群との衝突が誤って判断されて経路計画の誤りが発生します。

対象物の対称性を使用しない

このパラメータは、**目標点タイプ** が **対象物位置姿勢** の移動ステップ (目標点タイプが物体の位置姿勢の移動ステップ、パレタイジングステップなど) に対してのみ有効です。目標点のタイプが関節角度、TCP位置姿勢の移動ステップには無効です。

無し 無し：初期値。対称性を無効にしません。

Z 軸 Z 軸の対称性のみを使用しません。

XY 軸 X、Y 軸の対称性のみを使用しません。

全て 全ての対称性を使用しません。

物体の対称性を使用しないように設定すると、ロボットが物体の位置姿勢に正確に到達して対象物を配置します。



対象物を把持できない場合に **リソース**、**対象物設定** の **回転対称** を設定します。対象物に対称性がある場合、複数の候補位置姿勢があります。Mech-Viz では対象物の把持を計画するときに、デフォルト位置姿勢が把持できない場合、候補位置姿勢を試行します。物体対称位置姿勢と Mech-Vision によって出力された元位置姿勢とは一致しなければ、ロボットによる対象物の配置位置姿勢は一致しないことがあります。

インデックス

開始インデックス

- 説明** これから配置する箱のインデックス。
- オプション** 整数にしか設定できません。初期値は 0。
- 使用方法** 空のパレットの場合に **0** に設定します。
- パレタイジングを継続するとき、N 個の箱を配置した場合に値を N に設置します。これで N+1 個目の箱からパレタイジングを継続します。

現在のインデックス

- 説明** 箱の位置を表示します。値が N であれば、**N+1** 個目に配置する箱を表示します。
- オプション** 整数。自動的に読み取ります。外部コマンドを受信した場合、コマンドに従って更新します。

パレットパターンの基本設定

経路を非表示

デフォルトではチェックが外れており、箱の進入経路を表示します。チェックすると箱の進入経路を非表示します。

対象物の数

パレット上の配置可能な対象物の数。編集できません。

動作制御

開始段階の強制関節運動

デフォルトではチェックが入っており、ロボットがパレタイジングする前の運動を関節運動にします。

進入/調整/配置段階の強制関節運動

デフォルトではチェック外れており、強制的に関節運動を設定しません。

空間が狭くて直線運動に設定する必要がある場合にチェックします。狭い現場では、進入/調整/配置段階の運動を関節運動に設定することで特異点を回避することが可能です。

加速度・速度のスケーリング比例

調整範囲 0~100%。初期値：100

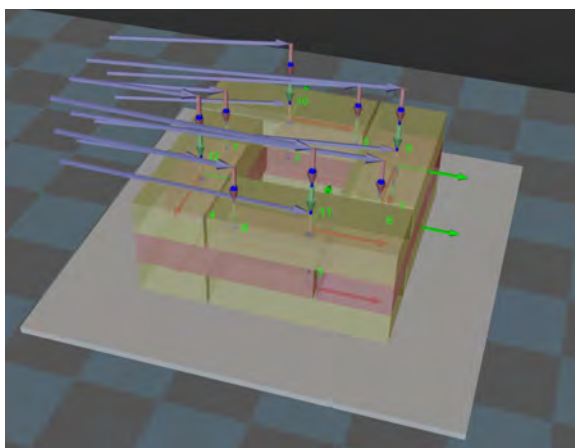
使用シーン ロボットがパレットに近づく時と箱を配置する時の速度が異なる場合に使用します。

説明 箱を配置する時の加速度・速度。**加速度 & 速度 × 加速度 & 速度 & 速度のスケーリング比例** によって計算します。

パレットに近づく 3つの段階：

一段階目：紫（パレットに近づく）。二段階目：ピンク（箱を配置）。三段階目：緑（箱を配置）

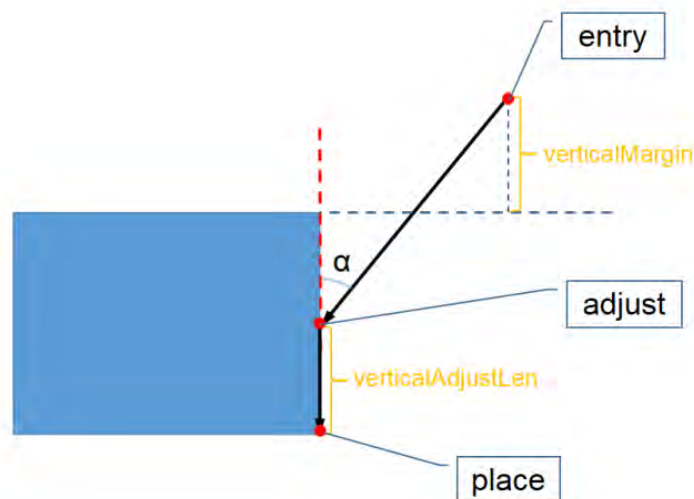
パレットに近づく速度と加速度は「基本的な移動設定」で指定されたあと、二、三段階目の速度/加速度は **加速度・速度 × 加速度・速度のスケーリング比例**。



進入と調整

3つのパラメータを設定して箱がパレットに進入する経路を指定します。箱が配置済みの箱の山に近づいてから垂直方向に沿って配置するような進入経路を調整します。これによって他の箱との衝突を回避できます。

箱ごとに、パレットにエントリーするには4つの位置があり、そのうち3つはこのパラメータグループによって制御されます。下図の赤い点に示すように、それらはentry、adjust、およびplaceです。下図の画像の視点は、箱を配置する正面図です。



垂直方向に長さの比例を調整

説明 調整する点（adjust）の位置を影響します。値 = verticalAdjustLen / 箱の高さ。

調整範囲 0~1

推奨値 0.5

垂直方向の範囲

説明 進入点（entry）の位置を指定します。値 = 高さの範囲。

調整範囲 0 ~ 無限大、単位：mm。実際に応じて調整してください。

Z方向の進入角度

説明 進入点（entry）から調整点（adjust）までの経路の垂直方向との角度を指定します（単位：°）。

調整範囲 -80°~80°

推奨値 30°~45°

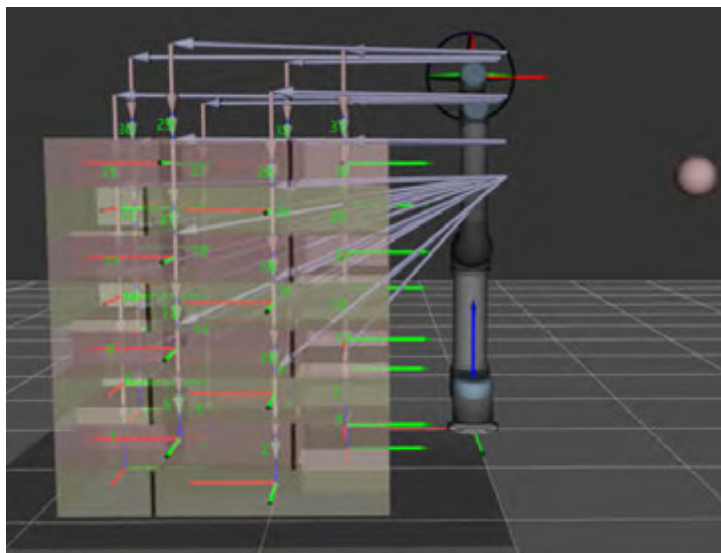
自動中間点

X/Y

ロボットベース座標系におけるピンクのボールの位置 x と y を指定します。この位置によって適切な中間点座標が算出されます。

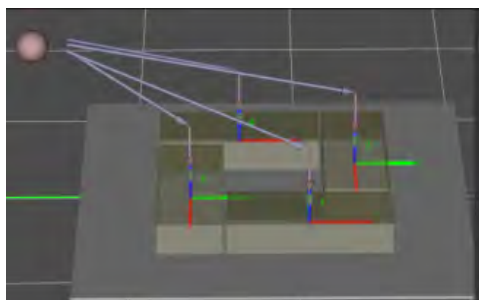
最小Z高さ

ロボットがパレットに進入する（紫の経路）時にZ方向の最小絶対高さ（Z方向高さはこの段の高さの差）。

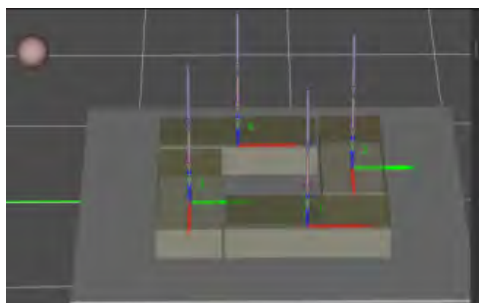


垂直な中間点の進入経路

デフォルトではチェックが外れており、進入段階は中間点方向に沿います。



チェックすると進入段階は中間点方向に沿わず、配置する位置の真上から進入します。



進入段階の延長距離

グリップが大きくて、進入段階が長いほど衝突発生確率は高くなる恐れがある現場に使用できません。安全に動作できるように進入段階を長くしてください。



自動中間点 はパレットに進入する方向だけを指定し、ロボットが到達する目標点を指定しないので、できるだけピンクのボールをパレットから遠ざけてください。

ビジョン処理によるパレットの位置を調整

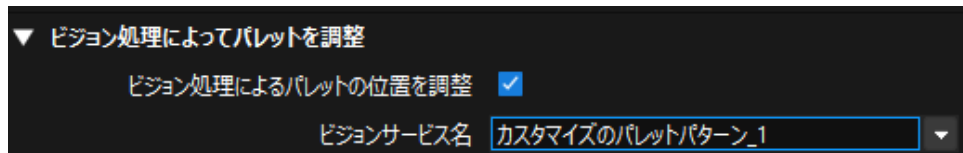
「ビジョン処理による認識」ステップを使用してパレットの位置を調整します。

ビジョン処理を使用してパレット位置を調整

デフォルトでチェックが外れています。チェックすると、動的にパレット位置を調整する必要がある場合にチェックします。「ビジョン処理による認識」ステップによって認識されたパレット位置を使用します。

ビジョンサービス名

パレット位置を認識するMech-Vision プロジェクト名（「ビジョン処理による認識」ステップ）を選択します。するとこのステップを実行するときに Mech-Vision プロジェクト名によって「ビジョン処理による認識」ステップを呼び出します。



パラメータ は外部設定（adapter）による調整も可能です。

パレットのサイズ

- パレットX辺長：パレットパターン（パレット）のX辺長を設定します。単位：mm。
- パレットY辺長：パレットパターン（パレット）のY辺長を設定します。単位：mm。

デパレタイズされるパレットとして

ビジョンサービスとして

デフォルトでチェックが外れています。ビジョンサービスとして使用しません。

チェックすると「ビジョン処理による認識」ステップにビジョンサービスを提供し、カスタマイズのパレットパターンを生成します。すなわち、Mech-Vision からビジョン処理の結果を受信しない場合でも Mech-Viz でデパレタイジングをシミュレート・実行することができます。

5.4.4.5. ビジョン処理による継続パレタイジング

機能

「ビジョン処理による認識」ステップによってパレットに配置された箱を確認し、事前計画したパレットパターンに基づいて続けてパレタイジングします。



「事前計画パレットパターン」、「カスタマイズのパレットパターン」、「スマートパレットパターン」ステップと合わせて使用します。「混載パレットパターン」と合わせて使用することはできません。

使用シーン

対象物を一部配置したがまだ終わっていないパレットに使用します。

パラメータ

移動ステップの共通パラメータ

移動目標点を送信

デフォルトではチェックが入っています。相手側（ロボットなど）に移動目標点の位置姿勢を送信します。チェックを外したら送信しませんが、この目標点位置姿勢は依然として経路計画の一部です。

移動コマンドの後の非移動コマンドをスムーズに実行することを試行

デフォルトではチェックが外れており、**移動ステップ**の間に「ビジョン処理による認識」、「DOを設定」、「DIをチェック」などの**非移動ステップ**を接続すると、ロボット移動経路の計画を中断し、ロボット実機動作中に一時中止することがあります。チェックを入れると現在のステップが完了しなくても続行することが可能です。これにより、ロボットがよりスムーズに動作します。ただし、ステップが途中で終了する可能性があります。

ステップが途中で終了する原因は何ですか？



Mech-Viz 実行中、ロボットに同時に複数の位置姿勢を送信しますが、最後の位置姿勢が現在のロボットの関節角度と一致しているかのみを判断します。一致していると判断したら、ロボットが最後の位置に到達していると考えられます。例えば、10の移動ステップがある経路では、移動ステップ5の位置姿勢は最後のステップの位置姿勢と同じとします。ロボットが低速に動作し、移動ステップ5の位置に到達したらその位置の関節角度をMech-Vizに送信します。経路では、移動ステップ5の位置姿勢は最後の移動ステップの位置姿勢と同じなので、Mech-Vizはロボットの動作が完了したと判断して途中で実行を終了します。

配置された対象物との衝突を検出しない

デフォルトではチェックが外れており、配置された対象物との衝突を検出しません。チェックを入れると、ロボット本体・ロボットハンドと配置された対象物との衝突を検出します。

パレタイジングのシーンでは、以下場合があります。

1. ロボットが段ボール箱を配置するときに配置済みの箱と軽く衝突する（箱の凹みや変形などが発生しない）ことがあります。Mech-Vizではこのような衝突を検出したら別の配置位置を計画するためパレットが満杯になりません。
2. 普通、吸盤を使用する場合、TCPを吸盤の表面でなく、モデル内部に設定するため、物体を吸着する時、吸盤と把持する箱のモデルと重なります（ソフトウェアでは吸盤と把持対象物との衝突を検出しない）。ロボットが箱を配置してから、把持された箱のモデルはシーンのモデルになり、吸盤がシーンの箱のモデルと衝突すると判断してメッセージを表示してパレタイジングが続行できなくなります。

これをチェックするとロボット本体やロボットハンドと配置済み対象物モデルとの衝突を検出しないので上記の問題を解決できます。

点群との衝突検出モード

現場の状況に応じて設定してください。普通、デフォルトの **自動** を使用します。ロボットが物体を把持する前の移動ステップは、**チェックしない** を選択し、把持した後の移動ステップを **チェック** を選択します。

自動 初期値。「ビジョン処理による移動」ステップと「ビジョン処理による移動」ステップに依存する「相対移動」の点群衝突だけを検出します。

チェックしない 移動ステップの点群衝突をいっさい検出しません。

チェック 全ての移動ステップの点群衝突を検出します。



衝突検出、**衝突検出設定**、**点群と他の対象物間の衝突を検出** をオンにすれば、Mech-Viz は経路計画を行うときにロボットモデル、ツールモデルと点群との衝突を検出します。普通、点群衝突の設定は、ロボットが把持を実行するときに把持対象物との衝突を検出するためです。空間内にノイズがあれば、ソフトウェアは対象物を把持する前の経路を計画するときに、ノイズがロボットモデル、ツールモデルと接触して点群との衝突が誤って判断されて経路計画の誤りが発生します。

対象物の対称性を使用しない

このパラメータは、**目標点タイプ** が **対象物位置姿勢** の移動ステップ（目標点タイプが物体の位置姿勢の移動ステップ、パレタイジングステップなど）に対してのみ有効です。目標点のタイプが関節角度、TCP位置姿勢の移動ステップには無効です。

無し 無し：初期値。対称性を無効にしません。

Z 軸 Z 軸の対称性のみを使用しません。

XY 軸 X、Y 軸の対称性のみを使用しません。

全て 全ての対称性を使用しません。

物体の対称性を使用しないように設定すると、ロボットが物体の位置姿勢に正確に到達して対象物を配置します。



対象物を把持できない場合に **リソース**、**対象物設定** の **回転対称** を設定します。対象物に対称性がある場合、複数の候補位置姿勢があります。Mech-Viz では対象物の把持を計画するときに、デフォルト位置姿勢が把持できない場合、候補位置姿勢を試行します。物体対称位置姿勢と Mech-Vision によって出力された元位置姿勢とは一致しなければ、ロボットによる対象物の配置位置姿勢は一致しないことがあります。

ビジョンサービス名

「ビジョン処理による認識」ステップを選択します。 **ビジョン処理による継続パレタイジング**

ステップはビジョンプロジェクトによってパレットに配置された箱を認識し、位置決めします。

ビジョン処理の結果は普通、2つの部分、**認識した箱の位置姿勢**と**認識したパレットの位置姿勢**で構成されています。

認識したパレットの位置姿勢は必須ではありません。ビジョン処理の結果に認識したパレットの位置姿勢がない場合、指定したパレットパターンのパレット位置姿勢を使用します。



パレットの実際の位置は設定したものと大きくずれている場合、ビジョンプロジェクトで認識した位置姿勢を使用してください。これはマッチングの精度を上げるためです。

ビジョン処理による結果を取得したあと、認識した箱の位置姿勢を指定したパレットパターン
の箱の位置姿勢とマッチングします。**ビジョン処理によるマッチング**を調整できます。マッチ
ングに失敗した場合、**マッチングに失敗** 出口から実行します。失敗の原因はログで確認できま
す。

ビジョン処理によるマッチング

物体XY オフセットのしきい値

指定したパレットパターンの箱とビジョンサービスによって認識された箱がXY 方向に沿うオフ
セットの最大値です。

対象物Z方向に沿う偏差距離

指定したパレットパターンの箱とビジョンサービスによって認識された箱がZ 方向に沿うオフ
セットの最大値です。

対象物がZ軸を中心に回転するしきい値

指定したパレットパターンの箱とビジョンサービスによって認識された箱がZ 方向に回転する
オフセットの最大値です。

対称性マッチング

ビジョンサービスから提供された箱の位置姿勢の向きとマッチングするパレットパターンの箱
の向きが一致しないために失敗する問題を解決できます。

自動	初期値
対称性無し	対称性を使用しない
2回対称	長方形に使用する
4回対称	正方形に使用する

位置姿勢の補正

位置姿勢補正設定

ビジョン処理の結果によるマッチング 実行後、箱の平均偏差に基づいてパレットパターンを補正します。

- 全てを補正：並進と回転をともに補正します。
- 並進だけを補正
- 回転だけを補正

判断条件

物体のラベル

ビジョンサービスから受信した箱のラベルによって物体をフィルタリングします。指定したラベルの物体にのみ **ビジョン処理の結果によるマッチング** を実行します。

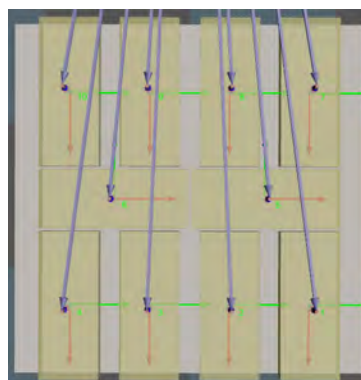
ラベルを設定しないと、ビジョンサービスによって認識された箱は全部パレットパターンのマッチングに使用されます。

配置された対象物の順序をチェック

デフォルトではチェックが入っています。継続パレタイジングを実行する時に配置済み箱の順序をチェックします。

継続パレタイジング番号：連続する番号でなければなりません。また、番号 1 は空ではありません。要件を満たさない時にプロンプトメッセージが表示されます。

例：下図に示すように、10 個の箱がある段に対し、把持する順番を計算します。



- 番号 1 に箱がまだ配置されず、2 と 3 には箱が配置されています。するとプロンプトメッセージが表示されます。
- 番号 1、2、3、4、6 に箱が配置され、5 には箱が配置されていません。するとプロンプトメッセージが表示されます。

番号 1 から箱が連続して配置されている場合に正常に実行できます。

5●4.4.6. パレットパターンの基本設定

インデックス

開始インデックス

説明 これから配置する箱のインデックス。

オプション 整数にしか設定できません。初期値は 0。

使用方法 空のパレットの場合に **0** に設定します。

パレタイジングを継続するとき、N 個の箱を配置した場合に値を N に設置します。これで N+1 個目の箱からパレタイジングを継続します。

現在のインデックス

説明 箱の位置を表示します。値が N であれば、**N+1** 個目に配置する箱を表示します。

オプション 整数。自動的に読み取ります。外部コマンドを受信した場合、コマンドに従って更新します。

パレットパターンの基本設定

経路を非表示

デフォルトではチェックが外れており、箱の進入経路を表示します。チェックすると箱の進入経路を非表示します。

対象物の数

パレット上の配置可能な対象物の数。編集できません。

動作制御

開始段階の強制関節運動

デフォルトではチェックが入っており、ロボットがパレタイジングする前の運動を関節運動にします。

進入/調整/配置段階の強制関節運動

デフォルトではチェック外れており、強制的に関節運動を設定しません。

空間が狭くて直線運動に設定する必要がある場合にチェックします。狭い現場では、進入/調整/配置段階の運動を関節運動に設定することで特異点を回避することが可能です。

加速度・速度のスケーリング比例

調整範囲 0~100%。初期値：100

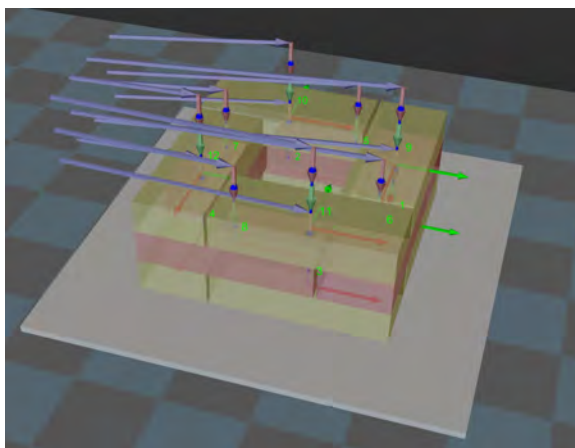
使用シーン ロボットがパレットに近づく時と箱を配置する時の速度が異なる場合に使用します。

説明 箱を配置する時の加速度・速度。**加速度 & 速度 × 加速度 & 速度 & 速度のスケーリング比例** によって計算します。

パレットに近づく 3つの段階：

一段階目：紫（パレットに近づく）。二段階目：ピンク（箱を配置）。三段階目：緑（箱を配置）

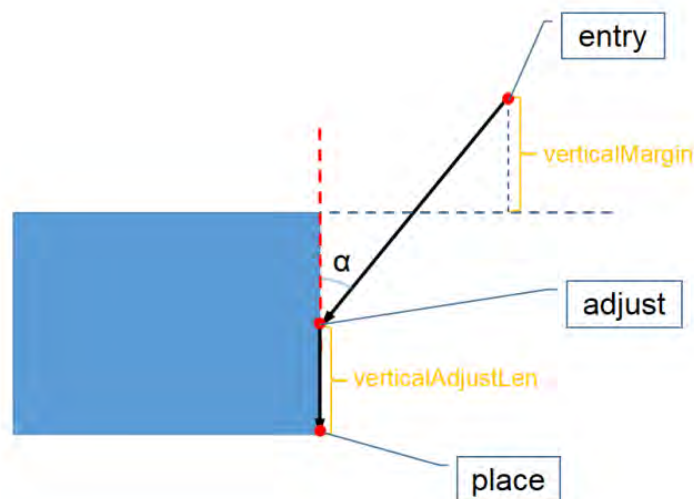
パレットに近づく速度と加速度は「基本的な移動設定」で指定されたあと、二、三段階目の速度/加速度は **加速度・速度 × 加速度・速度のスケーリング比例**。



進入と調整

3つのパラメータを設定して箱がパレットに進入する経路を指定します。箱が配置済みの箱の山に近づいてから垂直方向に沿って配置するような進入経路を調整します。これによって他の箱との衝突を回避できます。

箱ごとに、パレットにエントリーするには4つの位置があり、そのうち 3つはこのパラメータグループによって制御されます。下図の赤い点に示すように、それらはentry、adjust、およびplaceです。下図の画像の視点は、箱を配置する正面図です。



垂直方向に長さの比例を調整

説明 調整する点（adjust）の位置を影響します。値 = verticalAdjustLen / 箱の高さ。

調整範囲 0~1

推奨値 0.5

垂直方向の範囲

説明 進入点（entry）の位置を指定します。値 = 高さの範囲。

調整範囲 0 ~ 無限大、単位：mm。実際に応じて調整してください。

Z方向の進入角度

説明 進入点（entry）から調整点（adjust）までの経路の垂直方向との角度を指定します（単位：°）。

調整範囲 -80°~80°

推奨値 30°~45°

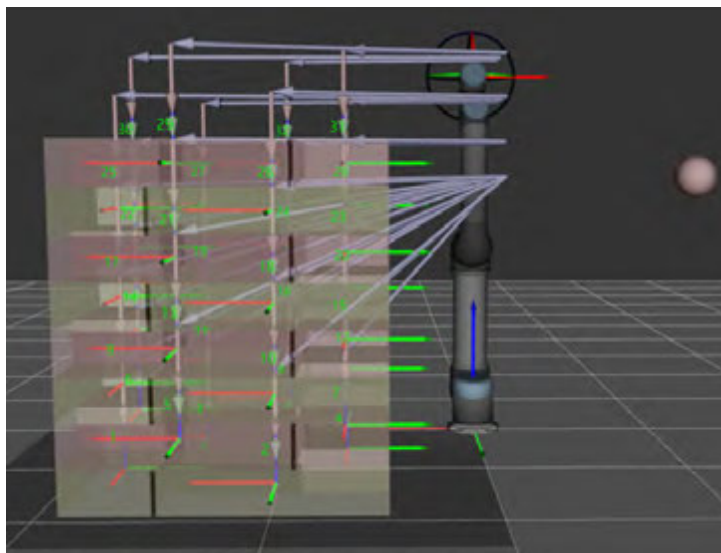
自動中間点

X/Y

ロボットベース座標系におけるピンクのボールの位置 x と y を指定します。この位置によって適切な中間点座標が算出されます。

最小Z高さ

ロボットがパレットに進入する（紫の経路）時にZ方向の最小絶対高さ（Z方向高さはこの段の高さの差）。

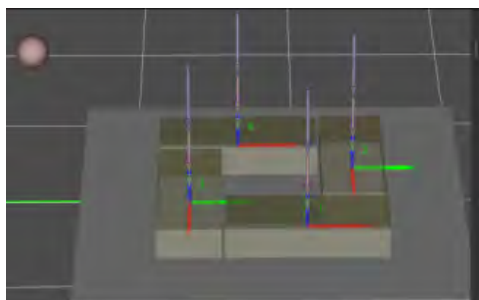


垂直な中間点の進入経路

デフォルトではチェックが外れており、進入段階は中間点方向に沿います。



チェックすると進入段階は中間点方向に沿わず、配置する位置の真上から進入します。



進入段階の延長距離

グリップが大きくて、進入段階が長いほど衝突発生確率は高くなる恐れがある現場に使用できません。安全に動作できるように進入段階を長くしてください。



自動中間点 はパレットに進入する方向だけを指定し、ロボットが到達する目標点を指定しないので、できるだけピンクのボールをパレットから遠ざけてください。

ビジョン処理によるパレットの位置を調整

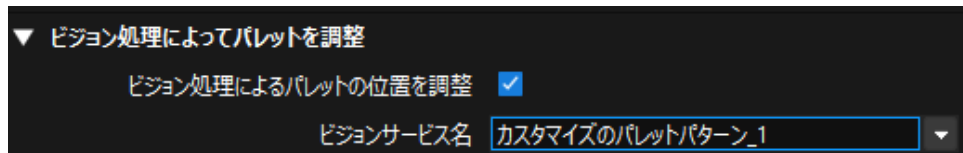
「ビジョン処理による認識」ステップを使用してパレットの位置を調整します。

ビジョン処理を使用してパレット位置を調整

デフォルトでチェックが外れています。チェックすると、動的にパレット位置を調整する必要がある場合にチェックします。「ビジョン処理による認識」ステップによって認識されたパレット位置を使用します。

ビジョンサービス名

パレット位置を認識するMech-Vision プロジェクト名（「ビジョン処理による認識」ステップ）を選択します。するとこのステップを実行するときに Mech-Vision プロジェクト名によって「ビジョン処理による認識」ステップを呼び出します。



パラメータ は外部設定（adapter）による調整も可能です。

5.4.5. ロボットハンド

5.4.5.1. ロボットハンドをチェック

機能

ロボットハンドを切り替えるために使用されます。このステップを使用して現在使っているロボットハンドは選択されたロボットハンドとは一致しているかを確認できます。一致している場合に左の「現在と一致」出口から実行し、ロボットハンド切り替えに関する移動・非移動ステップを実行する必要はありません。一致しない場合に右の「現在と不一致」出口からロボットハンド切り替えステップを実行します。

パラメータ

非移動ステップの共通パラメータ

実行をスキップ

スキップしない	初期値、このステップをスキップしません。
シミュレーション時	シミュレーションを実行する時にこのステップをスキップします。 出口は スキップ時の出口のインデックス によって決定されます。

いつも シミュレーション時も実際に実行中にも実行をスキップします。出口は **スキップ時の出口のインデックス** によって決定されます。

調整説明 **シミュレーション時** または **いつも** に設定すると、プロジェクト実行中にこのステップをスキップして後のステップを実行します。プロジェクトに「DIをチェック」ステップを配置したら、シミュレーション時に外部信号がなく、実行をスキップしなければプロジェクトの実行はここで停止することになります。このパラメータを設定すると、このステップを実行せずに次のステップを実行してシミュレーションを完了します。

スキップ時の出口のインデックス

実行をスキップ を **シミュレーション時** または **いつも** に設定すれば有効になります。このパラメータは、スキップしてから実行する出口を指定します。

ロボットハンドを設定

使用したいロボットハンドを選択して、現在使っているものとは一致しているかを確認します。

5.4.5.2. 関節角度を取得

機能

現在使っているロボットの関節角度を取得して Mech-Viz に同期させます。

使用シーン

セミティーチングの場合に、このステップを実行してロボットの関節角度を取得してロボットを制御することができます。

パラメータ

非移動ステップの共通パラメータ

実行をスキップ

スキップしない	初期値、このステップをスキップしません。
シミュレーション時	シミュレーションを実行する時にこのステップをスキップします。出口は スキップ時の出口のインデックス によって決定されます。
いつも	シミュレーション時も実際に実行中にも実行をスキップします。出口は スキップ時の出口のインデックス によって決定されます。

調整説明

シミュレーション時 または **いつも** に設定すると、プロジェクト実行中にこのステップをスキップして後のステップを実行します。プロジェクトに「DI をチェック」ステップを配置したら、シミュレーション時に外部信号がなく、実行をスキップしなければプロジェクトの実行はここで停止することになります。このパラメータを設定すると、このステップを実行せずに次のステップを実行してシミュレーションを完了します。

スキップ時の出口のインデックス

実行をスキップ を **シミュレーション時** または **いつも** に設定すれば有効になります。このパラメータは、スキップしてから実行する出口を指定します。

例

ビジョン処理による認識 は対応する Mech-Vision プロジェクトをトリガーする前に、**関節角度を取得** ステップを実行することでロボット実機の関節角度を取得し、Mech-Viz でロボット運動の開始位置とします。



5.4.5.3. 最大可搬質量設定

機能

ロボットの最大可搬質量を設定します。

パラメータ

非移動ステップの共通パラメータ

実行をスキップ

スキップしない	初期値、このステップをスキップしません。
シミュレーション時	シミュレーションを実行する時にこのステップをスキップします。 出口は スキップ時の出口のインデックス によって決定されます。
いつも	シミュレーション時も実際に実行中にも実行をスキップします。出口は スキップ時の出口のインデックス によって決定されます。
調整説明	シミュレーション時 または いつも に設定すると、プロジェクト実行中にこのステップをスキップして後のステップを実行します。プロジェクトに「DI をチェック」ステップを配置したら、シミュレーション時に外部信号がなく、実行をスキップしなければプロジェクトの実行はここで停止することになります。このパラメータを設定すると、このステップを実行せずに次のステップを実行してシミュレーションを完了します。

スキップ時の出口のインデックス

実行をスキップ を **シミュレーション時** または **いつも** に設定すれば有効になります。このパラメータは、スキップしてから実行する出口を指定します。

可搬質量 ID

複数の可搬質量がある場合に使用されます。可搬質量が一つだけある場合に初期値を使用すればいいです。

可搬質量

ロボットハンドと対象物も含む最大可搬質量(kg) を入力します。

フランジに対する荷重重心を設定

X/Y/Z: フランジに対する荷重重心を設定します（ロボットブランドによってはこのパラメータを設定する必要はない）。

5.4.5.4. 把持状態を設定

機能

把持の状態を設定します。普通は把持に失敗したあと、把持の状態を変更するために使用されます。

使用シーン

移動ステップの「把持/配置」パラメータの設定と組み合わせて把持の状態を更新します。

パラメータ

非移動ステップの共通パラメータ

実行をスキップ

- | | |
|------------------|--|
| スキップしない | 初期値、このステップをスキップしません。 |
| シミュレーション時 | シミュレーションを実行する時にこのステップをスキップします。
出口は スキップ時の出口のインデックス によって決定されます。 |
| いつも | シミュレーション時も実際に実行中にも実行をスキップします。出口は スキップ時の出口のインデックス によって決定されます。 |

調整説明

シミュレーション時 または **いつも** に設定すると、プロジェクト実行中にこのステップをスキップして後のステップを実行します。プロジェクトに「DI をチェック」ステップを配置したら、シミュレーション時に外部信号がなく、実行をスキップしなければプロジェクトの実行はここで停止することになります。このパラメータを設定すると、このステップを実行せずに次のステップを実行してシミュレーションを完了します。

スキップ時の出口のインデックス

実行をスキップ を **シミュレーション時** または **いつも** に設定すれば有効になります。このパラメータは、スキップしてから実行する出口を指定します。

把持状態

実際のプロジェクトに応じて「把持」または「把持無し」を選択します。

5.4.5.5. ロボットハンドを切り替え

機能

異なるロボットハンドを切り替える場合に使用されます。

パラメータ

非移動ステップの共通パラメータ

実行をスキップ

- | | |
|----------------|----------------------|
| スキップしない | 初期値、このステップをスキップしません。 |
|----------------|----------------------|

シミュレーション時	シミュレーションを実行する時にこのステップをスキップします。出口は スキップ時の出口のインデックス によって決定されます。
いつも	シミュレーション時も実際に実行中にも実行をスキップします。出口は スキップ時の出口のインデックス によって決定されます。
調整説明	シミュレーション時 または いつも に設定すると、プロジェクト実行中にこのステップをスキップして後のステップを実行します。プロジェクトに「DI をチェック」ステップを配置したら、シミュレーション時に外部信号がなく、実行をスキップしなければプロジェクトの実行はここで停止することになります。このパラメータを設定すると、このステップを実行せずに次のステップを実行してシミュレーションを完了します。

スキップ時の出口のインデックス

実行をスキップ を **シミュレーション時** または **いつも** に設定すれば有効になります。このパラメータは、スキップしてから実行する出口を指定します。

ロボットハンドを設定

ドロップメニューには追加されたロボットハンドが表示されます。使用するものを選択してください。

5.4.5.6. 制御を移転

機能

一時的に制御をロボットに移転し、ロボットの制御を続行するのを待ちます。

パラメータ

非移動ステップの共通パラメータ

実行をスキップ

スキップしない	初期値、このステップをスキップしません。
シミュレーション時	シミュレーションを実行する時にこのステップをスキップします。出口は スキップ時の出口のインデックス によって決定されます。
いつも	シミュレーション時も実際に実行中にも実行をスキップします。出口は スキップ時の出口のインデックス によって決定されます。

調整説明

シミュレーション時 または **いつも** に設定すると、プロジェクト実行中にこのステップをスキップして後のステップを実行します。プロジェクトに「DIをチェック」ステップを配置したら、シミュレーション時に外部信号がなく、実行をスキップしなければプロジェクトの実行はここで停止することになります。このパラメータを設定すると、このステップを実行せずに次のステップを実行してシミュレーションを完了します。

スキップ時の出口のインデックス

実行をスキップ を **シミュレーション時** または **いつも** に設定すれば有効になります。このパラメータは、スキップしてから実行する出口を指定します。

出口の数

出口の数を設定します。

関節角度をチェック

制御を移転する前にロボットの関節角度をチェックするかを指定します。

制御を移転する前の関節角度

関節角度をチェック にチェックを入れるとこのパラメータが表示されます。関節角度をチェックするために必要な関節角度を入力する必要があります。

5.4.6. サービス

5.4.6.1. 通知

機能

外部サービスヘカスタマイズの通知を送信します。

使用シーン

外部サービスと通信する必要があるときに使用します。

パラメータ

非移動ステップの共通パラメータ

実行をスキップ

スキップしない	初期値、このステップをスキップしません。
----------------	----------------------

シミュレーション時	シミュレーションを実行する時にこのステップをスキップします。出口は スキップ時の出口のインデックス によって決定されます。
いつも	シミュレーション時も実際に実行中にも実行をスキップします。出口は スキップ時の出口のインデックス によって決定されます。
調整説明	シミュレーション時 または いつも に設定すると、プロジェクト実行中にこのステップをスキップして後のステップを実行します。プロジェクトに「DI をチェック」ステップを配置したら、シミュレーション時に外部信号がなく、実行をスキップしなければプロジェクトの実行はここで停止することになります。このパラメータを設定すると、このステップを実行せずに次のステップを実行してシミュレーションを完了します。

スキップ時の出口のインデックス

実行をスキップ を **シミュレーション時** または **いつも** に設定すれば有効になります。このパラメータは、スキップしてから実行する出口を指定します。

Adapter 名

Adapter は **Adapter 名** によってこのステップの情報を取得します。プロジェクトには複数の **通知** ステップがある場合、すべての **通知** ステップに同じ **Adapter 名** を指定しなければなりません。

メッセージ

メッセージをカスタマイズします。Adapter はメッセージによって異なるステップを実行します。

失敗した時の動作

通知のメッセージを送信できなかった場合に実行する動作を選択します。

オプション：

- ワーニング：ワーニングします（初期値）。
- 再試行：再度試行します。
- 破棄：ステップを破棄します。

ロボット要停止

チェックすると、ロボットがこのステップで停止します。チェックを外すと、実行していると同時にメッセージを送信します。

タイムアウト

指定した時間内にメッセージを送信しない場合、ロボットが **失敗した時の動作** を実行します。

例

Adapter 名 を設定し、計画が失敗したメッセージを Adapter に送信します。失敗した時のメッセージを **finish** に設定します。



▼ 非移動タスクの基本パラメータ

実行をスキップ スキップしない

スキップ時の出口のインデックス 0

▶ 受信者

Adapter名

メッセージ finish

失敗した時の動作 ワーニング

ロボット要停止 ☒

タイムアウト 1000 ms

5.4.7. ツール

5.4.7.1. 分類

機能

「ビジョン処理による認識」ステップの前・後に実行します。Mech-Vision によって認識された対象物のラベルによって分岐を選択します。

使用シーン

Mech-Vision によって認識された対象物のラベルによって分類します。

- 「ビジョン処理による移動」ステップの前に：デフォルトの把持流れは、対象物を認識する
> ラベルによって異なる分岐を実行する > そのラベルの対象物を把持する。例：異なる部品
を把持するシーンでグリッパを交換する場合に使用します。
- 「ビジョン処理による移動」ステップの後に：デフォルトの把持流れは、対象物を認識する
> 対象物を把持する > 対象物の種類によって異なる分類操作を実行する。例：グリッパを変
更せず、対象物を分類してから異なるコンテナに配置する場合に使用します。

パラメータ

非移動ステップの共通パラメータ

実行をスキップ

スキップしない	初期値、このステップをスキップしません。
シミュレーション時	シミュレーションを実行する時にこのステップをスキップします。 出口は スキップ時の出口のインデックス によって決定されます。
いつも	シミュレーション時も実際に実行中にも実行をスキップします。出 口は スキップ時の出口のインデックス によって決定されます。
調整説明	シミュレーション時 または いつも に設定すると、プロジェクト実 行中にこのステップをスキップして後のステップを実行します。プ ロジェクトに「DI をチェック」ステップを配置したら、シミュレ ーション時に外部信号がなく、実行をスキップしなければプロジェ クトの実行はここで停止することになります。このパラメータを設 定すると、このステップを実行せずに次のステップを実行してシミ ュレーションを完了します。

スキップ時の出口のインデックス

実行をスキップ を **シミュレーション時** または **いつも** に設定すれば有効になります。このパラ
メータは、スキップしてから実行する出口を指定します。

「把持対象物を更新」からラベルを取得

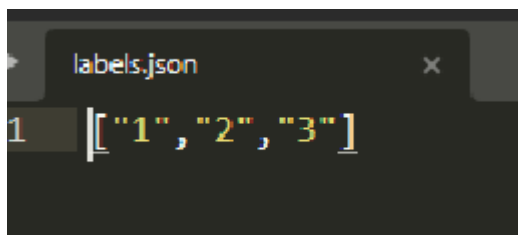
チェック	このステップの前には「把持済み対象物を更新」ステップであり、現在 の把持対象物のラベルの情報を取得してから分類の設定によって異なる 動作を実行します。対象物の積み重ねなどで対象物のラベルの情報を 取得できない場合、対象物を把持してから二回目の分類を実行し、ラベ ルを出力可能。
------	--

チェックしない 初期値。Mech-Vision で認識された対象物のラベルを使用する。**ビジョン処理による移動の前に分類** をチェックして対応する「ビジョン処理による移動」ステップを選択する必要があります。

ラベルを読み込む、ラベルを追加

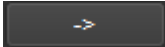
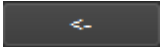


ラベルを読み込む 未使用のラベル リストに複数のラベルを追加します。 **ラベルを読み込む** をクリックし、下図に示すように複数のラベルを含む **.json** ファイルを選択します。それですべてのレベルファイルは **未使用のラベル** リストに読み込まれ、元のラベルが削除されます。



ラベルを追加 未使用のラベル リストに **unnamed_label** ラベルが表示されます。このラベルをダブルクリックするとラベル名を編集できます。

ラベルリスト

未使用のラベルと**使用されたラベル**の2つのリストがあります。ラベル名を選択してから  または  をクリックすると移動できます。

使用されたラベルは「分類」ステップの出口に表示されます。また、**その他** 出口もあり、**使用されたラベル** がないラベルに対応します。

5.4.7.2. カウンター

機能

現在のプロジェクトあるいは分岐が実行された回数を指定します。普通、[ステップをリセット](#)と組み合わせて使用します。

使用シーン

実行の回数をカウントする必要があるシーンに使用します。

パラメータ

非移動ステップの共通パラメータ

実行をスキップ

スキップしない

初期値、このステップをスキップしません。

シミュレーション時

シミュレーションを実行する時にこのステップをスキップします。出口は **スキップ時の出口のインデックス** によって決定されます。

いつも

シミュレーション時も実際に実行中にも実行をスキップします。出口は **スキップ時の出口のインデックス** によって決定されます。

調整説明

シミュレーション時 または **いつも** に設定すると、プロジェクト実行中にこのステップをスキップして後のステップを実行します。プロジェクトに「DIをチェック」ステップを配置したら、シミュレーション時に外部信号がなく、実行をスキップしなければプロジェクトの実行はここで停止することになります。このパラメータを設定すると、このステップを実行せずに次のステップを実行してシミュレーションを完了します。

スキップ時の出口のインデックス

実行をスキップ を **シミュレーション時** または **いつも** に設定すれば有効になります。このパラメータは、スキップしてから実行する出口を指定します。

計数回数

計数を実行する回数。指定した回数を超えたら **完成** 出口から実行します。

計数のタイプ

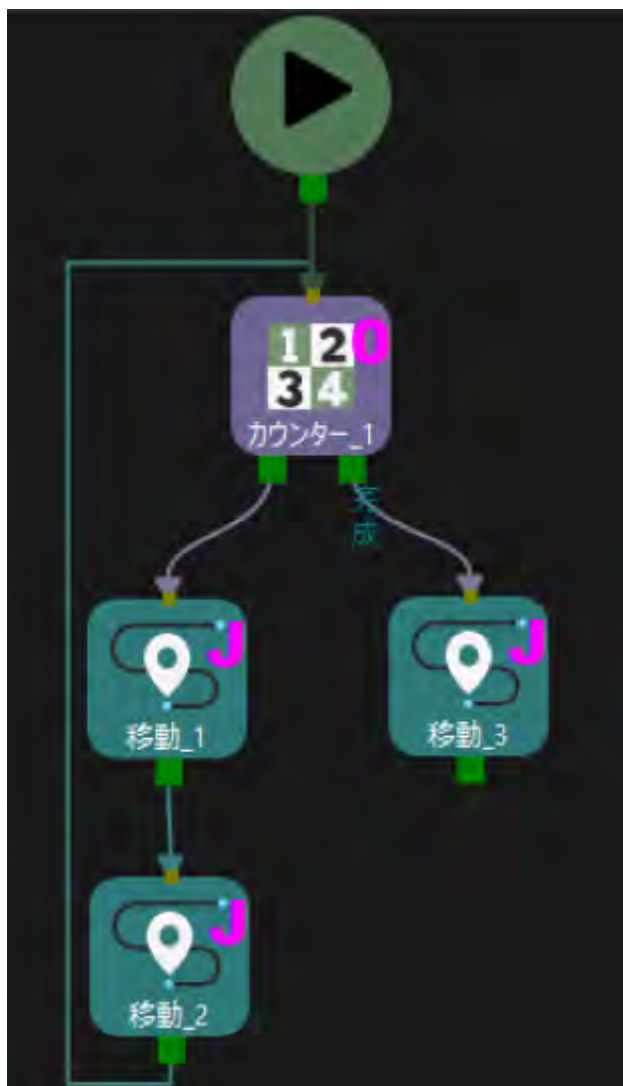
実行：カウンターに繋がれたステップの実行回数をカウントします。把持された対象物の数：ビジョン処理による移動 ステップを実行する回数をカウントします。

現在の計数

カウンターステップが実行された回数または「ビジョン処理による移動」ステップが実行された回数です。**計数のタイプ** の設定によります。

例

移動の経路を 2 回たどる必要があるシーン。



計数回数 を 2 に設定すると、2 回実行した後は **完成** 出口から実行します。

5.4.7.3. 完了確認

機能

インデックスが設定可能な移動ステップ（「グリッドによる移動」、「リストによる移動」など）の実行が完了したかを確認します。

使用シーン

普通、「グリッドによる移動」、「リストによる移動」と組み合わせて使用します。

一回実行が完了したら、**完成** 出口から実行し、完了していない場合は **未完成** 出口から実行します。

パラメータ

非移動ステップの共通パラメータ

実行をスキップ

スキップしない	初期値、このステップをスキップしません。
シミュレーション時	シミュレーションを実行する時にこのステップをスキップします。 出口は スキップ時の出口のインデックス によって決定されます。
いつも	シミュレーション時も実際に実行中にも実行をスキップします。出口は スキップ時の出口のインデックス によって決定されます。
調整説明	シミュレーション時 または いつも に設定すると、プロジェクト実行中にこのステップをスキップして後のステップを実行します。プロジェクトに「DI をチェック」ステップを配置したら、シミュレーション時に外部信号がなく、実行をスキップしなければプロジェクトの実行はここで停止することになります。このパラメータを設定すると、このステップを実行せずに次のステップを実行してシミュレーションを完了します。

スキップ時の出口のインデックス

実行をスキップ を **シミュレーション時** または **いつも** に設定すれば有効になります。このパラメータは、スキップしてから実行する出口を指定します。

移動ステップを選択

プロジェクトのワークフローから移動ステップを選択します。

例

「グリッドによる移動」ステップと組み合わせて使用し、実行が完成したかを判断します。



5.4.7.4. インデックスを変更

機能

現在のインデックス パラメータが設定できる移動ステップや「カウンター」と組み合わせて使用します。

実行するたびに、**ステップサイズ** 値に従って移動ステップの **現在のインデックス** あるいは「カウンター」ステップの **現在の計数** が変更します。

使用シーン

現在のインデックス パラメータが設定できる移動ステップや「カウンター」ステップと組み合わせて使用します。

パラメータ

非移動ステップの共通パラメータ

実行をスキップ

スキップしない	初期値、このステップをスキップしません。
シミュレーション時	シミュレーションを実行する時にこのステップをスキップします。出口は スキップ時の出口のインデックス によって決定されます。
いつも	シミュレーション時も実際に実行中にも実行をスキップします。出口は スキップ時の出口のインデックス によって決定されます。
調整説明	シミュレーション時 または いつも に設定すると、プロジェクト実行中にこのステップをスキップして後のステップを実行します。プロジェクトに「DIをチェック」ステップを配置したら、シミュレーション時に外部信号がなく、実行をスキップしなければプロジェクトの実行はここで停止することになります。このパラメータを設定すると、このステップを実行せずに次のステップを実行してシミュレーションを完了します。

スキップ時の出口のインデックス

実行をスキップ を **シミュレーション時** または **いつも** に設定すれば有効になります。このパラメータは、スキップしてから実行する出口を指定します。

ステップサイズ

初期値は 0 です。



移動ステップの **現在のインデックス** と「カウンター」ステップの **現在の計数** の値は 1 大きくなります。

ステップサイズ の値に従って移動ステップの **現在のインデックス** と「カウンター」ステップの **現在の計数** を変更します。

変更した **現在のインデックス** / **現在の計数** = 更新前の **現在のインデックス** / **現在の計数** + 1

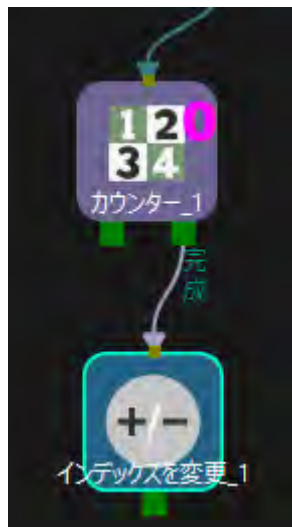
- 0：影響無し。
- -1：実行するたびに:移動ステップの **現在のインデックス** と「カウンター」ステップの **現在の計数** の値は 1 小さくなります。
- 1：移動ステップの **現在のインデックス** と「カウンター」ステップの **現在の計数** の値は 1 大きくなります。

インデックスパラメータが設定できる移動ステップ/カウンターを選択する

実際に応じて移動ステップの「現在のインデックス」と「カウンター」ステップを選択します。

例

「カウンター」ステップの後に接続します。**現在の計数**の値を変更することができます。



ステップサイズを -1 に設定すると、このステップを実行するたびに、「カウンター」ステップの**現在の計数**は1小さくなります。

5.4.7.5. ステップをリセット

機能

パラメータの設定通りに選択したステップをリセットします。

使用シーン

カウントや把持・配置の判断、ビジョン処理による結果の再使用などに使用します。

パラメータ

非移動ステップの共通パラメータ

実行をスキップ

スキップしない	初期値、このステップをスキップしません。
シミュレーション時	シミュレーションを実行する時にこのステップをスキップします。 出口は スキップ時の出口のインデックス によって決定されます。
いつも	シミュレーション時も実際に実行中にも実行をスキップします。出口は スキップ時の出口のインデックス によって決定されます。

調整説明

シミュレーション時 または **いつも** に設定すると、プロジェクト実行中にこのステップをスキップして後のステップを実行します。プロジェクトに「DI をチェック」ステップを配置したら、シミュレーション時に外部信号がなく、実行をスキップしなければプロジェクトの実行はここで停止することになります。このパラメータを設定すると、このステップを実行せずに次のステップを実行してシミュレーションを完了します。

スキップ時の出口のインデックス

実行をスキップ を **シミュレーション時** または **いつも** に設定すれば有効になります。このパラメータは、スキップしてから実行する出口を指定します。

一部をリセット

デフォルトではチェックが外れており、このステップの前の計数または、把持・配置の判断、ビジョン処理による結果の再使用などのステップを一つだけリセットします。**ステップ名を選択** でステップを選択します。

チェックすると、プロジェクトの計数、把持・配置の判断、ビジョン処理による結果の再使用などのステップが複数ある場合、**ステップ名を選択** でステップを選択したあと、インターフェイスの右下に「リセット」エリアにリセット可能なステップが表示されます。必要に応じてチェックしてください。

ステップを選択

リセットする必要があるステップを選択します。

5.4.7.6. 待つ

機能

このステップを実行するとき指定した時間待機します。ms を単位とします。

使用シーン

待ち時間 を指定する必要があるシーンに使用します。普通は「DO を設定」ステップや **ビジョン処理の結果をチェック** の **結果無し** 出口の後に接続します。

パラメータ

非移動ステップの共通パラメータ

実行をスキップ

スキップしない

初期値、このステップをスキップしません。

シミュレーション時	シミュレーションを実行する時にこのステップをスキップします。出口は スキップ時の出口のインデックス によって決定されます。
いつも	シミュレーション時も実際に実行中にも実行をスキップします。出口は スキップ時の出口のインデックス によって決定されます。
調整説明	シミュレーション時 または いつも に設定すると、プロジェクト実行中にこのステップをスキップして後のステップを実行します。プロジェクトに「DI をチェック」ステップを配置したら、シミュレーション時に外部信号がなく、実行をスキップしなければプロジェクトの実行はここで停止することになります。このパラメータを設定すると、このステップを実行せずに次のステップを実行してシミュレーションを完了します。

スキップ時の出口のインデックス

実行をスキップ を **シミュレーション時** または **いつも** に設定すれば有効になります。このパラメータは、スキップしてから実行する出口を指定します。

待ち時間

待機する時間を入力します。初期値は1000ms（1s）となります。

5.4.8. 経路

5.4.8.1. 経路ステップの組合せ

機能

経路点のテンプレート通りに移動します。 [ビジョン処理による移動](#) と組み合わせて使用します。

パラメータ

調整可能なパラメータはありません。

経路点の Json ファイルパス

事前に用意した経路点 Json ファイルを選択します。プロジェクトを実行する時に、ビジョンサービスから受信した点の数とテンプレートの点の数と一致しなければなりません。

5.4.9. ビジョン

5.4.9.1. ビジョン処理の結果をチェック

機能

ビジョン処理の結果をチェックし、それぞれに対応する出口から実行します。

このステップには5つの出口があります：



- 結果あり：ビジョン処理の結果が出力されたと確認したらこの出口を実行します。
- 結果無し：ビジョン処理の結果が出力されないと判断したらこの出口を実行します。
- 未撮影：複数の「ビジョン処理による認識」ステップが配置されたプロジェクトに使用されます。前のワークフローに現在のビジョンサービスの「ビジョン処理による認識」ステップがない場合にこの出口から実行します。このステップを **ビジョン処理による認識** ステップの後に繋いだ場合にこの出口に線を繋ぐ必要はありません。
- タイムアウト：ビジョン処理は設定した **待ち時間タイムアウト** の時間内に完了していない場合にこの出口から実行します。この出口は **ビジョン処理の結果をチェック** ステップの入口に繋いでステップを再実行することができます。また、**待つ** ステップに繋いでビジョン処理の結果をチェック ステップの入口に戻すこともできます。
- 点群無し：ROI 内に点群がない、つまり対象物を認識していない場合にこの出口から実行します。
 - 点群がなければロボットは初期位置に戻り、再度画像を取得してビジョン処理の結果をチェックして対象物があるかを判断します。対象物がない場合にロボットを停止して対象物が全部把持されたかをチェックする必要があります。

使用シーン

「ビジョン処理による認識」の結果に基づいて異なる分岐を実行します。

パラメータ

非移動ステップの共通パラメータ

実行をスキップ

スキップしない 初期値、このステップをスキップしません。

シミュレーション時	シミュレーションを実行する時にこのステップをスキップします。出口は スキップ時の出口のインデックス によって決定されます。
いつも	シミュレーション時も実際に実行中にも実行をスキップします。出口は スキップ時の出口のインデックス によって決定されます。
調整説明	シミュレーション時 または いつも に設定すると、プロジェクト実行中にこのステップをスキップして後のステップを実行します。プロジェクトに「DI をチェック」ステップを配置したら、シミュレーション時に外部信号がなく、実行をスキップしなければプロジェクトの実行はここで停止することになります。このパラメータを設定すると、このステップを実行せずに次のステップを実行してシミュレーションを完了します。

スキップ時の出口のインデックス

実行をスキップ を **シミュレーション時** または **いつも** に設定すれば有効になります。このパラメータは、スキップしてから実行する出口を指定します。

カスタマイズのキーをチェック

デフォルトではチェックが外れており、カスタマイズのキーをチェックしません。チェックを入れるとそれをチェックします。**チェックするキー** パラメータを設定して Adapter と連動します。

事前計画する出口のインデックス

プロジェクトをスムーズに実行するために、事前計画する出口を指定する必要があります。普通、可能性の高い分岐を指定します。

-1 事前計画する出口を指定しません。

0 **結果あり** 出口から実行します。

待ち時間タイムアウト

ビジョン処理の結果を待つ時間の上限を指定します。秒を単位とします。初期値は 5 秒です。出口 **タイムアウト** に適用されます。必要に応じてを設定してください。

ビジョンサービス名

ビジョンサービスを提供する Mech-Vision プロジェクトを指定します。

5.4.9.2. パレットの位置姿勢を更新

機能

ビジョンサービスを利用してパレットの位置姿勢を更新します。

使用シーン

パレタイジング完了後、新しいパレットが配置され、パレット位置姿勢も更新されます。ビジョンサービスを利用して指定されたパレタイジングステップのパレット位置姿勢を更新します。

パラメータ

非移動ステップの共通パラメータ


実行をスキップ

スキップしない	初期値、このステップをスキップしません。
シミュレーション時	シミュレーションを実行する時にこのステップをスキップします。出口は スキップ時の出口のインデックス によって決定されます。
いつも	シミュレーション時も実際に実行中にも実行をスキップします。出口は スキップ時の出口のインデックス によって決定されます。
調整説明	シミュレーション時 または いつも に設定すると、プロジェクト実行中にこのステップをスキップして後のステップを実行します。プロジェクトに「DIをチェック」ステップを配置したら、シミュレーション時に外部信号がなく、実行をスキップしなければプロジェクトの実行はここで停止することになります。このパラメータを設定すると、このステップを実行せずに次のステップを実行してシミュレーションを完了します。


スキップ時の出口のインデックス

実行をスキップ を **シミュレーション時** または **いつも** に設定すれば有効になります。このパラメータは、スキップしてから実行する出口を指定します。

パレット位置姿勢を保存

デフォルトではチェックが外れており、更新後のパレット位置姿勢を保存しません。チェックするとそれを保存します。**パレット位置姿勢を保存するファイルパス** の  をクリックして保存場所を指定します。

パレット位置姿勢を読み込む

デフォルトではチェックが外れており、パレット位置姿勢を読み込みません。チェックするとそれを読み込みます。**パレット位置姿勢を読み込むファイルパス** の  をクリックしてファイルを選択します。

パレタイジングステップを選択

パレット位置姿勢を更新する必要がある **パレタイジング ステップ** を選択します。

ビジョンサービス名

登録済みのビジョンサービスを選択し、このサービスから新しいパレットの位置姿勢を受信します。

例

ロボットがパレタイジングを実行後、新しいパレットが別の位置に配置されるのでパレット位置姿勢を更新する必要があります。[ビジョン処理による認識](#)と組み合わせて使用します。



5.4.9.3. 把持済み対象物を更新

機能

ビジョンサービスによって把持済み対象物の位置または寸法を更新して衝突を回避します。



対象物の位置姿勢はロボットが把持から配置までの対象物の位置姿勢です。

使用シーン

取得した対象物の寸法または位置姿勢が正確でない場合に使用します。

衝突を回避するために、把持対象物の正確な寸法と位置姿勢を再取得し、衝突モデルを改めて選択して衝突検出を実行する必要があります。

- 対象物は長方形あるいは垂直設置の円柱、水平設置の円柱です。
- このステップは移動ステップの計画を中断します。

パラメータ

非移動ステップの共通パラメータ

実行をスキップ

スキップしない	初期値、このステップをスキップしません。
シミュレーション時	シミュレーションを実行する時にこのステップをスキップします。 出口は スキップ時の出口のインデックス によって決定されます。
いつも	シミュレーション時も実際に実行中にも実行をスキップします。出口は スキップ時の出口のインデックス によって決定されます。

調整説明

シミュレーション時 または **いつも** に設定すると、プロジェクト実行中にこのステップをスキップして後のステップを実行します。プロジェクトに「DIをチェック」ステップを配置したら、シミュレーション時に外部信号がなく、実行をスキップしなければプロジェクトの実行はここで停止することになります。このパラメータを設定すると、このステップを実行せずに次のステップを実行してシミュレーションを完了します。

スキップ時の出口のインデックス

実行をスキップ を **シミュレーション時** または **いつも** に設定すれば有効になります。このパラメータは、スキップしてから実行する出口を指定します。

更新方法

「情報出所を修正」に **パラメータ設定** と **ビジョンサービス** が選択可能です。

パラメータ設定

把持対象物の寸法がない、またはロボットの把持はビジョン処理に依存しない場合に使用します。「**コンテンツを更新**」の「位置姿勢要更新」または「寸法要更新」にチェックを入れて値を設定することで対象物のモデルを作成します。それでこのモデルは3Dシミュレーションエリアに表示され、衝突検出に使用できます。

対象物のタイプ

長方形、垂直設置の円柱、水平設置の円柱、ブレーキディスク。


位置姿勢要更新

TCP 座標系にける位置姿勢を更新します。デフォルトではチェックが外れており、把持対象物の位置姿勢を更新しません。チェックすると把持対象物の位置姿勢を更新します。**位置姿勢を更新** に新しい位置姿勢の情報を入力します。

寸法要更新

把持対象物の寸法を更新します。デフォルトではチェックが外れており、把持対象物の寸法を更新しません。チェックすると把持対象物の位置姿勢を更新します。**寸法を更新** に新しい位置姿勢の情報を入力します。

ブレーキディスクのコンフィグファイルのパス

 をクリックしてブレーキディスクのコンフィグファイル **json** ファイルを選択します。寸法の調整にのみ対応可能。

ビジョンサービス

ロボットが対象物を把持して再度撮影することで対象物の寸法と TCP 座標系における位置姿勢を再度測定します。

コンテンツを更新

位置姿勢要更新

デフォルトではチェックが入っており、TCP 座標を自動的にプロジェクトに更新します。チェックを外すと把持済み対象物の位置姿勢を更新しません。

寸法要更新

デフォルトではチェックが入っており、把持済み対象物の寸法を自動的にプロジェクトに更新します。チェックを外すと更新しません。

運動事前計画

事前計画する出口のインデックス

スムーズな計画のために、事前計画する出口を指定する必要があります。普通、実行する可能性の高い出口を指定します。これにより再計画を回避してロボットの待機時間を低減します。

- 0：更新不要の出口、二回の撮影結果の偏差が指定したしきい値を更新より小さい場合に実行する
- -1：更新出口、更新に基づいて再計画する
- 異常出口、ビジョンサービスに異常が発生したときに実行する

しきい値を更新

精度要検査 デフォルトではチェックが入っており、**サイズしきい値**、**並進しきい値**、**回転しきい値**の値を調整可能。チェックを外すと精度のしきい値は設定されません。

例

分類 ステップと「混載パレットパターン」の一部の機能と組み合わせて使用できます。

分類 ステップと合わせて使用するとき、**分類**の「**把持対象物を更新**」から**ラベルを取得**にチェックを入れます。

混載パレットパターン ステップと組み合わせて使用するとき、「混載パレットパターン」ステップの**パレットパターン**・**落下配置**パラメータの**再撮影して箱の寸法を更新**にチェックを入れます。下図に示します。



5.4.9.4. シーンの物体を更新

機能

ビジョンサービスまたは指定したパラメータの値によってシーンの物体の寸法と位置姿勢を更新します。

使用シーン

Mech-Viz 実行中に、衝突の検出・回避のためにシーンの物体モデルの位置と寸法を調整する必要があります。

パラメータ

非移動ステップの共通パラメータ

実行をスキップ

スキップしない	初期値、このステップをスキップしません。
シミュレーション時	シミュレーションを実行する時にこのステップをスキップします。出口は スキップ時の出口のインデックス によって決定されます。
いつも	シミュレーション時も実際に実行中にも実行をスキップします。出口は スキップ時の出口のインデックス によって決定されます。
調整説明	シミュレーション時 または いつも に設定すると、プロジェクト実行中にこのステップをスキップして後のステップを実行します。プロジェクトに「DIをチェック」ステップを配置したら、シミュレーション時に外部信号がなく、実行をスキップしなければプロジェクトの実行はここで停止することになります。このパラメータを設定すると、このステップを実行せずに次のステップを実行してシミュレーションを完了します。

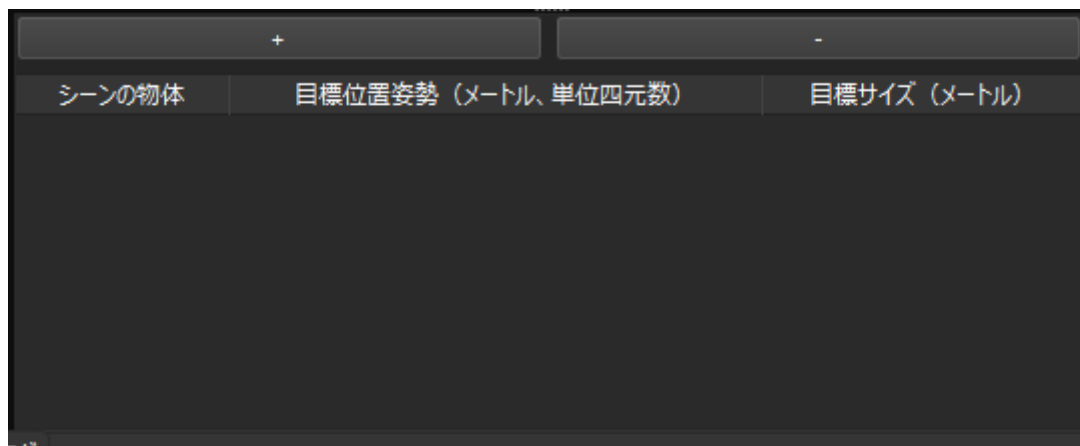
スキップ時の出口のインデックス

実行をスキップ を **シミュレーション時** または **いつも** に設定すれば有効になります。このパラメータは、スキップしてから実行する出口を指定します。

情報出所を更新

シーンの物体の位置姿勢と寸法の情報の出所を指定します。**パラメータ設定から** と **ビジョンサービスから** が選択できます。

パラメータ設定から



[+] : シーンの物体を追加します。

[-]：選択したシーンの物体を削除します。

実際に応じてシーンの物体を追加・削除します。リストの順序に沿って実行します。「+」をクリックしてから「シーンの物体」のドロップダウンメニューで追加したい物体の種類を選択します。それから **目標位置姿勢** と **目標サイズ** を指定します。目標位置姿勢を XYZ（メートルを単位とする）形式と四元数の 7 数字を入力してください。目標サイズは XYZ 形式の 3 数字（メートルを単位とする）を入力してください。

例えば、1.00, 0.00, 0.00, 1.00, 0.00, 0.00, 0.00 の目標位置姿勢、直方体の場合に 0.40, 0.40, 1.00（横、縦、高さ）の目標サイズ、円柱の場合に 0.50, 1.00, 0.00（半径、高さ、0）を入力することができます。

ビジョンサービスから

ビジョンサービス名

ビジョンサービスを提供する Mech-Vision プロジェクトを指定します。ただし、登録済みの Mech-Vision プロジェクトでなければなりません。

Mech-Vision プロジェクトを指定したあと、以下の手順に沿って Mech-Vision プロジェクトの「出力」ステップを設定してください。

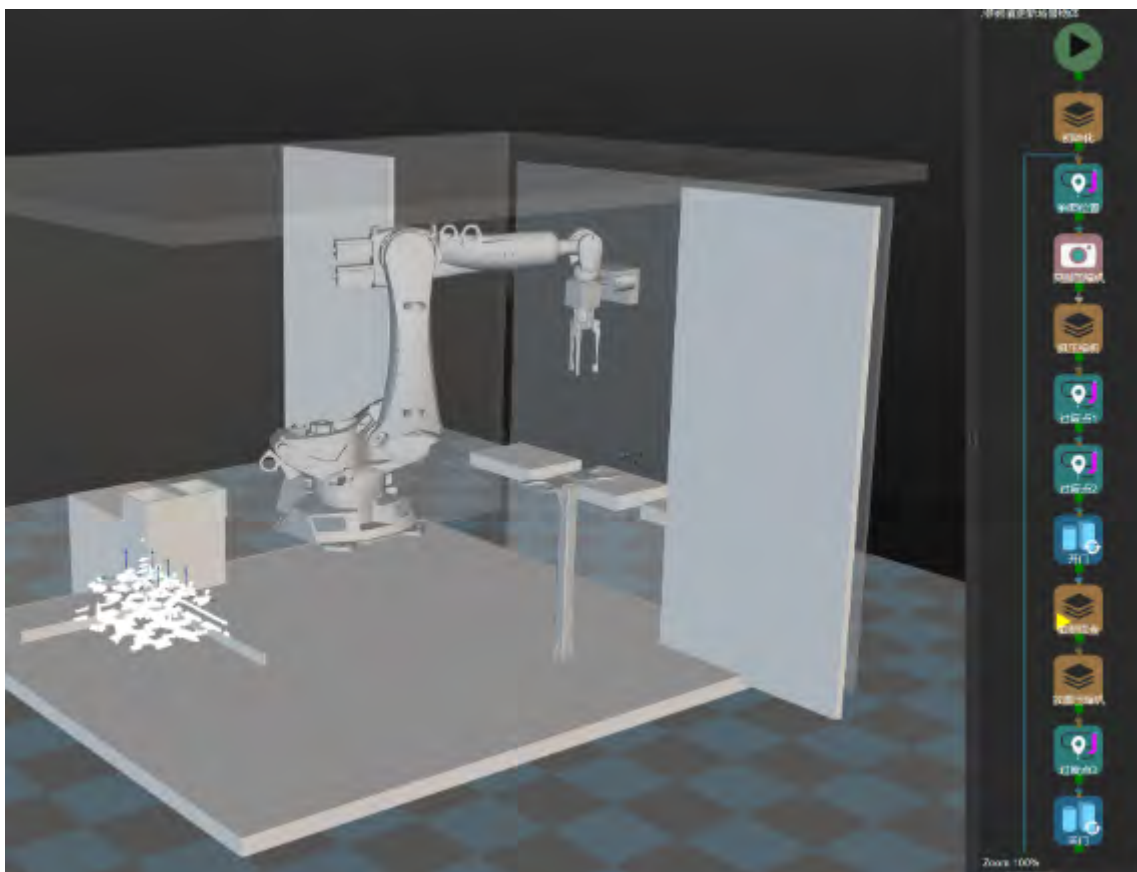
1. ステップの **ポートタイプ** に **カスタム** を選択します。
2. [外部通信への出力データタイプを設定] ボタンをクリックして出力設定アシスタントウィンドウを開きます。
3. [追加] をクリックして通信キーを追加できます。3つの通信キーを追加します。
4. 追加された通信キーをダブルクリックしてそれぞれ **sceneObjectNames**、**sceneObjectSizes**、**sceneObjectPoses** と名付けます。
5. [確認] をクリックして設定を保存します。
6. 出力 ステップの入力ポートを前のステップの対応する出力ポートに繋がめます。



ビジョンサービスからの結果	説明
sceneObjectNames	更新する Mech-Viz プロジェクトのシーンの物体名に対応する
sceneObjectSizes	XYZ の3つの値（メートルを単位とする）が必要
sceneObjectPoses	XYZ の3つの値（メートルを単位とする）と四元数の7数字が必要

例

下図は工程の衝突回避を実現するためのセンサドア開閉のシミュレーションです。パラメータを調整することでシーンの物体を更新します。



5.4.9.5. ビジョン処理の結果を使い切る

機能

指定した「ビジョン処理による移動」ステップのビジョン処理の結果が使い切れたかをチェックします。



「ビジョン処理による移動」ステップのパラメータ **ビジョン処理の結果を再使用** にチェックを入れなければ使用できない。

使用シーン

「ビジョン処理による移動」ステップのビジョン処理の結果に複数の使用可能なビジョン位置姿勢がある場合、それが全部使用されるまでロボットが順番に把持を実行します。

ビジョン処理の結果にある位置姿勢を使い切れたら画像を取得します。短いタクトタイムが求められる現場では、ビジョン処理の結果を再使用することでビジョン処理の時間を短縮できます。

パラメータ説明

非移動ステップの共通パラメータ

実行をスキップ

スキップしない	初期値、このステップをスキップしません。
シミュレーション時	シミュレーションを実行する時にこのステップをスキップします。出口は スキップ時の出口のインデックス によって決定されます。
いつも	シミュレーション時も実際に実行中にも実行をスキップします。出口は スキップ時の出口のインデックス によって決定されます。
調整説明	シミュレーション時 または いつも に設定すると、プロジェクト実行中にこのステップをスキップして後のステップを実行します。プロジェクトに「DIをチェック」ステップを配置したら、シミュレーション時に外部信号がなく、実行をスキップしなければプロジェクトの実行はここで停止することになります。このパラメータを設定すると、このステップを実行せずに次のステップを実行してシミュレーションを完了します。

スキップ時の出口のインデックス

実行をスキップ を **シミュレーション時** または **いつも** に設定すれば有効になります。このパラメータは、スキップしてから実行する出口を指定します。

事前計画する出口のインデックス

スムーズな計画のために、事前計画する出口を指定する必要があります。普通は実行する可能性の高い出口を指定し、再度計画を回避してロボットの待機時間を短縮します。

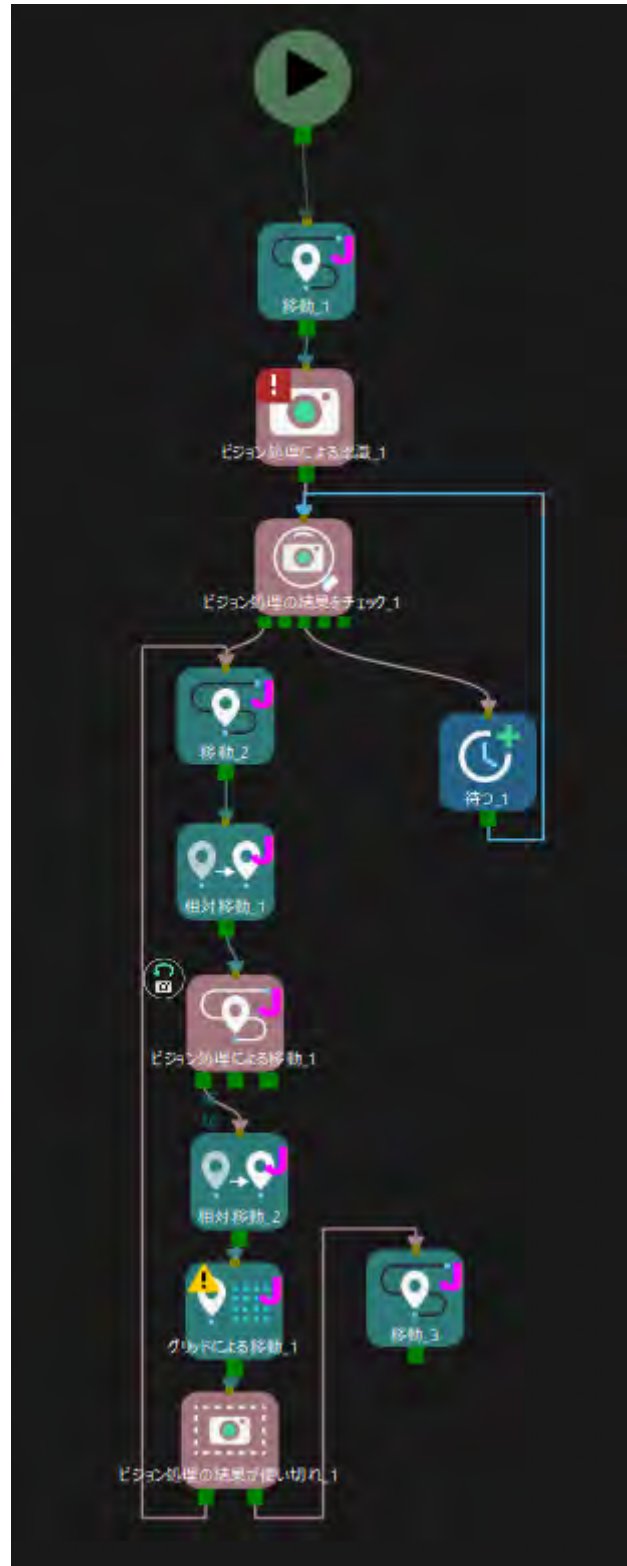
- 1 事前計画する出口を指定しません。
- 0 使い切っていない 出口から実行します。
- 1 使い切れ 出口から実行します。

「ビジョン処理による移動」を選択

チェックしたい「ビジョン処理による移動」ステップを選択します。

例

デパレタイジングのプロジェクトでは一回撮影してから段ごとの箱を把持します。**ビジョン処理の結果が使い切れ** ステップを使用して **ビジョン処理による移動** の結果が使い切れたかを判断します。



- ビジョン処理の結果が **使い切っていない**：把持を繰り返して実行します。「移動_2 →→ relative_move_1 →→ vision_move_1 →→ relative_move_3 →→ グリッドによる移動_1 →→ ビジョン処理の結果が使い切れ」を実行します。
- ビジョン処理の結果が **使い切れ**：使い切れ 出口から実行します。

5.4.9.6. ビジョン処理による認識

機能

Mech-Vision プロジェクトを実行し、ビジョン処理の結果を Mech-Vizz に出力します。

使用シーン

すべてのプロジェクトに適用します。普通は「ビジョン処理の結果をチェック」、「相対移動」、「ビジョン処理による移動」ステップと組み合わせて使用します。

パラメータ説明

非移動ステップの共通パラメータ

実行をスキップ

スキップしない	初期値、このステップをスキップしません。
シミュレーション時	シミュレーションを実行する時にこのステップをスキップします。出口は スキップ時の出口のインデックス によって決定されます。
いつも	シミュレーション時も実際に実行中にも実行をスキップします。出口は スキップ時の出口のインデックス によって決定されます。
調整説明	シミュレーション時 または いつも に設定すると、プロジェクト実行中にこのステップをスキップして後のステップを実行します。プロジェクトに「DI をチェック」ステップを配置したら、シミュレーション時に外部信号がなく、実行をスキップしなければプロジェクトの実行はここで停止することになります。このパラメータを設定すると、このステップを実行せずに次のステップを実行してシミュレーションを完了します。

スキップ時の出口のインデックス

実行をスキップ を **シミュレーション時** または **いつも** に設定すれば有効になります。このパラメータは、スキップしてから実行する出口を指定します。

移動の完全な到達を待つ

デフォルトではチェックが外れており、ロボットが指定位置に到達しなくてもこのステップを実行します。

ビジョン処理の結果の未使用を許容

デフォルトではチェックが外れており、ビジョン処理の結果を使用します。チェックを入れるとビジョン処理の結果を使用しません。

ビジョンサービス名

使用する Mech-Vision プロジェクトを選択します。

5.4.9.7. ビジョン処理による移動

機能

ビジョンサービスから取得した位置姿勢に従って動作します。

出口	説明
成功	経路の計画に成功した
計画失敗	経路の計画に失敗した
他の失敗	経路を計画していないとき、ビジョン処理による移動に必要なデータはない（Mech-Vision から位置姿勢が出力されない場合、Mech-Vision から位置姿勢が出力されましたが要件を満たしている結果はない場合を含むが、これに限定されるものではない）

使用シーン

普通、ロボット把持に使用されます。

パラメータの説明

本節では、☐はチェックが外れていること、☒はチェックが入っていることを表します。

移動ステップの共通パラメータ

移動目標点を送信

デフォルトではチェックが入っています。相手側（ロボットなど）に移動目標点の位置姿勢を送信します。チェックを外したら送信しませんが、この目標点位置姿勢は依然として経路計画の一部です。

移動コマンドの後の非移動コマンドをスムーズに実行することを試行

デフォルトではチェックが外れており、**移動ステップ**の間に「ビジョン処理による認識」、「DO を設定」、「DI をチェック」などの**非移動ステップ**を接続すると、ロボット移動経路の計画を中断し、ロボット実機動作中に一時中止することがあります。チェックを入れると現在のステップが完了しなくても続行することが可能です。これにより、ロボットがよりスムーズに動作します。ただし、ステップが途中で終了する可能性があります。



ステップが途中で終了する原因は何ですか？

Mech-Viz 実行中、ロボットに同時に複数の位置姿勢を送信しますが、最後の位置姿勢が現在のロボットの関節角度と一致しているかのみを判断します。一致していると判断した

ら、ロボットが最後の位置に到達していると見なします。例えば、10の移動ステップがある経路では、移動ステップ5の位置姿勢は最後のステップの位置姿勢とは同じとします。ロボットが低速に動作し、移動ステップ5の位置に到達したらその位置の関節角度をMech-Vizに送信します。経路では、移動ステップ5の位置姿勢は最後の移動ステップの位置姿勢とは同じなので、Mech-Vizはロボットの動作が完了したと判断して途中で実行を終了します。

配置された対象物との衝突を検出しない

デフォルトではチェックが外れており、配置された対象物との衝突を検出しません。チェックを入れると、ロボット本体・ロボットハンドと配置された対象物との衝突を検出します。

パレタイジングのシーンでは、以下の場合があります。

1. ロボットが段ボール箱を配置するときに配置済みの箱と軽く衝突する（箱の凹みや変形などが発生しない）ことがあります。Mech-Vizではこのような衝突を検出したら別の配置位置を計画するためパレットが満杯になりません。
2. 普通、吸盤を使用する場合、TCPを吸盤の表面でなく、モデル内部に設定するため、物体を吸着する時、吸盤と把持する箱のモデルと重なります（ソフトウェアでは吸盤と把持対象物との衝突を検出しない）。ロボットが箱を配置してから、把持された箱のモデルはシーンのモデルになり、吸盤がシーンの箱のモデルと衝突すると判断してメッセージを表示してパレタイジングが続行できなくなります。

これをチェックするとロボット本体やロボットハンドと配置済み対象物モデルとの衝突を検出しないので上記の問題を解決できます。

点群との衝突検出モード

現場の状況に応じて設定してください。普通、デフォルトの**自動**を使用します。ロボットが物体を把持する前の移動ステップは、**チェックしない**を選択し、把持した後の移動ステップを**チェック**を選択します。

自動 初期値。「ビジョン処理による移動」ステップと「ビジョン処理による移動」ステップに依存する「相対移動」の点群衝突だけを検出します。

チェックしない 移動ステップの点群衝突をいっさい検出しません。

チェック 全ての移動ステップの点群衝突を検出します。



衝突検出、**衝突検出設定**、**点群と他の対象物間の衝突を検出**をオンにすれば、Mech-Vizは経路計画を行うときにロボットモデル、ツールモデルと点群との衝突を検出します。普通、点群衝突の設定は、ロボットが把持を実行するときに把持対象物との衝突を検出するためです。空間内にノイズがあれば、ソフトウェアは対象物を把持する前の経路を計画するときに、ノイズがロボットモデル、ツールモデルと接触して点群との衝突が誤って判断されて経路計画の誤りが発生します。

対象物の対称性を使用しない

このパラメータは、**目標点タイプ**が**対象物位置姿勢**の移動ステップ（目標点タイプが物体の位置姿勢の移動ステップ、パレタイジングステップなど）に対してのみ有効です。目標点のタイプが関節角度、TCP位置姿勢の移動ステップには無効です。

- 無し** 無し：初期値。対称性を無効にしません。
- Z 軸** Z 軸の対称性のみを使用しません。
- XY 軸** X、Y 軸の対称性のみを使用しません。
- 全て** 全ての対称性を使用しません。

物体の対称性を使用しないように設定すると、ロボットが物体の位置姿勢に正確に到達して対象物を配置します。



対象物を把持できない場合に **リソース・対象物設定** の **回転対称** を設定します。対象物に対称性がある場合、複数の候補位置姿勢があります。Mech-Viz では対象物の把持を計画するときに、デフォルト位置姿勢が把持できない場合、候補位置姿勢を試行します。物体対称位置姿勢と Mech-Vision によって出力された元位置姿勢とは一致しなければ、ロボットによる対象物の配置位置姿勢は一致しないことがあります。

ビジョン処理の結果の全体的設定

一度に全ての目標点に移動

チェックを入れると、ロボットは一度に順番に全てのビジョン位置姿勢に到達します。普通、接着剤塗布など、ロボットが所定の経路に沿って移動し、動作中に DO 信号がないシーンに使用されます。他のパラメータと同時に調整することはできません。このパラメータにチェックを入れると、他のパラメータは非表示になります。

ビジョン処理の結果を再使用

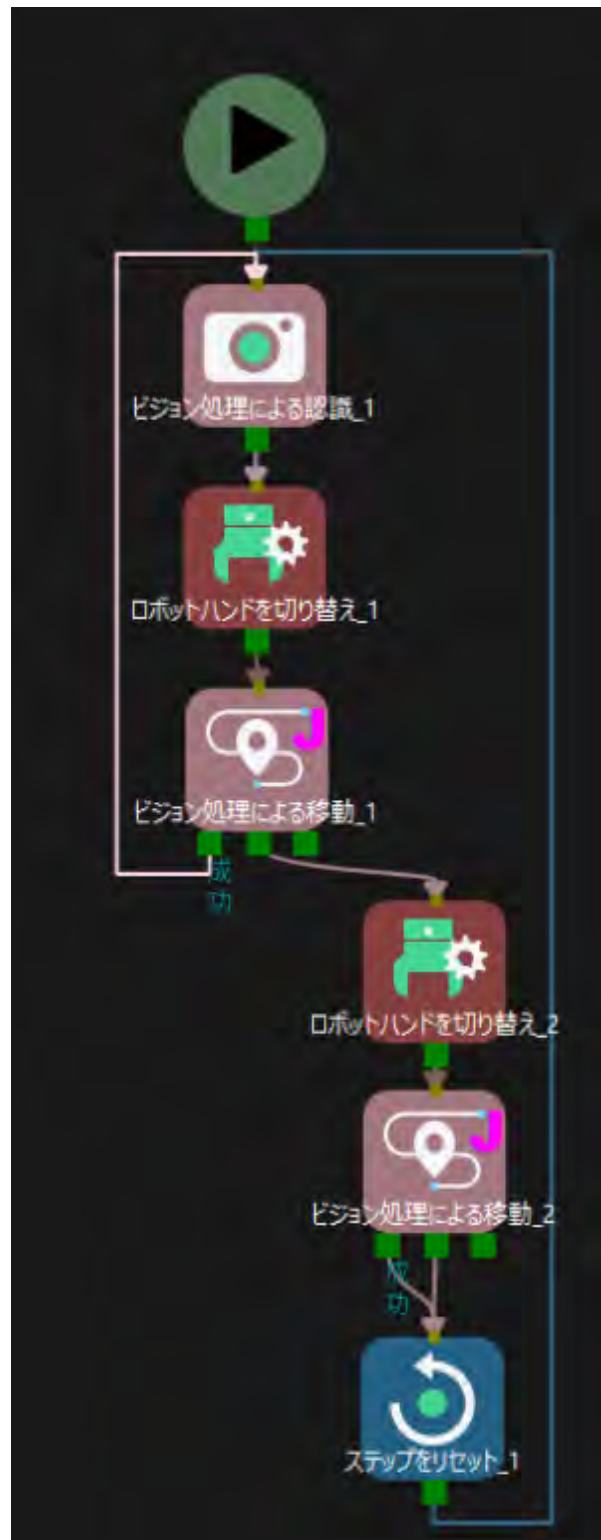
一つのビジョン処理の結果に複数の把持対象物があるとき、**物体を把持する時に他の物体の位置姿勢が変わらない限り**、このビジョン処理の結果を使い切るまで繰り返して使用することができます。

ビジョン処理の結果を共有

複数の **ビジョン処理による移動** ステップが一つのビジョン処理の結果を使用することをサポートします。ビジョンサービス名を指定する必要があります。

応用事例：把持しにくい対象物を扱うとき、多種類のロボットハンドを使用して何回も把持を試行することがあります。初回の把持に失敗したら、デフォルトではこのビジョン処理の結果が破棄されますが、この機能を使用すれば、画像を再取得せずに共有されたビジョン処理の結果を使用して再度把持を試行することになります。ここで注意すべきは、最後の「ビジョン処理による移動」ステップの後に「ステップをリセット」ステップを追加して共有されたビジョ

ン処理の結果をクリアする必要があります。そうしなければ、共有されたビジョン処理の結果を使い切らないうちに再度画像を取得しても古いビジョン処理の結果を使用することになり、さらにプロジェクトの正常な実行に影響が出てしまいます。



把持された直方体の点群を除去

把持された直方体の点群を除去します。これは点群衝突検出ではロボットハンドと対象外物体

の衝突だけを検出するためです。

追加除去された点群の X・Y 軸に沿う長さ

把持対象物の点群が衝突検出に関与しないように、把持対象物点群を衝突検出点群から除去します。それと同時に対象物外の点群も除去します。対象物の X と Y 軸に沿う距離の点群を除去するように指定します。

追加除去された点群の Z 軸に沿う長さ

把持対象物の点群が衝突検出に関与しないように、把持対象物点群を衝突検出点群から除去します。それと同時に対象物外の点群も除去します。対象物の Z 軸に沿う距離の点群を除去するように指定します。

実行モード

ロボットハンドの種類によって 3つのモードが選択されます。

一般的な ロボット ハンド	一般的なロボットハンド
デパレタイズ用吸盤	デパレタイズ用吸盤（複数把持）
	デパレタイズ用吸盤（単一把持）
配列タイプグリッパ	配列タイプグリッパ

ビジョンサービス名

普通、「ビジョン処理による認識」ステップと同じ Mech-Vision プロジェクトを選択します。

基本的な移動設定

基本的な移動設定は、移動ステップの主なパラメータを設定し、ロボットが目標点に到達する**速度**と**運動方式**を決定します。

把持と配置設定

未指定	初期値
把持	「ビジョン処理による移動」の前の移動ステップ。
放置	「ビジョン処理による移動」の後の移動ステップ。

調整説明 Mech-Viz プロジェクトのロジック検査に使用されます。現場の実行手順に応じて、把持をしてから配置するという基本的な原則に従って設定します。

運動タイプ

関節運動 ロボットが円弧に沿って走行します。スムーズに走行できるので特異点を回避することが可能。ロボットの移動範囲が広くて高い精度を求めないシーンに適用しています。

直線運動 ロボットが直線に沿って走行します。溶接や接着剤塗布、把持など高い精度が求められるシーンに適用しています。

速度&加速度

速度&加速度は、ロボット動作のスピードを決定します。普通、加速度の値を速度より小さく設定します。加速度の値を速度よりも大きくしたらロボット動作がスムーズでなくなります。



ロボットが安定に把持を実行するように、「ビジョン処理による移動」とその前後のステップの速度を低く設定してください。

ブレンド半径

初期値 50.00mm

調整説明 調整説明：通常、初期値を使用すればいいです。

ブレンド半径とは、回転する位置が目標点までの距離であり、大きいほどロボットの動きはスムーズになります。ロボットが狭いスペースで動作する場合、ブレンド半径を大きく設定する必要はありません。

ロボットがより広いスペースで動作し、障害物がなくて2つの経路の間に距離が遠い場合、ロボットがスムーズに動くように、ブレンド半径を大きく設定することができます。

以下のラベルの位置姿勢のみを使用

ビジョンサービスによって提供された視覚結果に異なるラベルを含む可能性があります。**ビジョン処理による移動**は選択されたラベルの位置姿勢のみを使用します。

ラベルを選択 ラベルを選択するには、ラベルを読み込む または ラベルを追加 をクリックする。ラベルを選択したら、それが右のドロップダウンメニューに表示される

ラベルを削除

ラベルを削除するには、 をクリックしてください

以下のラベルの対象物の設定を使用（計画用）

異なるワークに対して、異なる「対象物の設定」を使用し、また異なるラベルに対応します。

ビジョン処理の結果にラベルがないときに**デフォルト**を選択します。

ビジョン処理の結果に対応するレベルがあるときに対応するレベルを選択します。

5.4.9.7.1. 一般的なロボットハンド

同じ位置姿勢で把持することを回避

このパラメータは、把持に失敗した場合に、同じ場所で再び失敗することを回避するために使用されます。

優先度を減少する範囲の半径上限

初期値 0

調整説明 把持位置姿勢が前回の把持位置姿勢との直線距離がこの値より小さければ、前回の把持位置姿勢とは同じ位置姿勢と判断し、把持を実行する優先度を下げます。

破棄する範囲の半径上限

初期値 0

調整説明 把持位置姿勢が前回の把持位置姿勢との直線距離がこの値より小さければ、この位置姿勢を破棄します。

例 クランクシャフトを一つ把持するとき、もし初回の試行が失敗し、そのクランクシャフトを動かして位置が変わったら、次回の把持では成功する可能性があります。したがって、**優先度を減少する範囲の半径上限**を使用して優先度を下げますが、破棄はしません。もし試行が失敗し、かつそのクランクシャフトの位置が変わらなかったら、二回目の把持でも成功することはありません。この場合に無駄な動作を回避するために **破棄する範囲の半径上限** を使用してこの位置姿勢を破棄します。

回避リストの最大長さ

初期値 1

調整説明 把持を実行する時に回避する対象物（対象物の位置姿勢または対象物）の数の最大値。

応用事例 パラメータの値を2に設定し、Mech-Visionにより3つの位置姿勢を出力するとします。位置姿勢1の把持に失敗して記録します。位置姿勢2の把持に失敗して記録します。位置姿勢3の把持に失敗して記録します。このとき、位置姿勢1は破棄され、2と3の記録だけが残ります。

回避の対象

対象物の位置姿勢 把持対象物の位置姿勢を記録します。対象物に3つの把持点があり、その中に1つの把持点が試行されたと記録されても、残りの2つの位置姿勢は未試行とされます。

対象物 把持対象物を記録します。対象物のある把持点を試行したら、次回に試行するときにこの対象物の全ての把持点を試行したと見いだされてこの物を優先に把持しません。

把持数

計画総計数 対象物を把持する数の上限。

把持済み数 把持済み箱の総計数。

今回の把持数 今回把持する箱の数。

5.4.9.7.2. 配列タイプグリッパ

今、単列の吸盤のみをサポートしています。

マッチング条件

マッチング戦略

「位置のみをチェック」を選択したら、「距離しきい値」しか調整できません。「位置と方向をチェック」を選択したら、「距離しきい値」と「角度しきい値」を調整できます。

距離しきい値

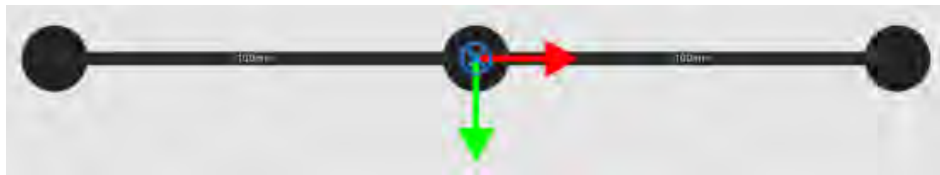
調整説明 位置姿勢を円心に、距離しきい値を半径をに円を描いて把持位置の範囲とします。

調整の例

1. 距離しきい値を 30mm に設定すれば、把持位置は下図に示します。



2. 三つの吸盤がある配列、吸盤同士の間隔は100mmです。下図に示します。



3. 把持の動作を計画する時に、ソフトウェアは三つの吸盤をそれぞれ把持の位置（下図の水色の円）に合わせます。



3つの吸盤を全部マッチングできない場合、2つの吸盤をマッチングします。

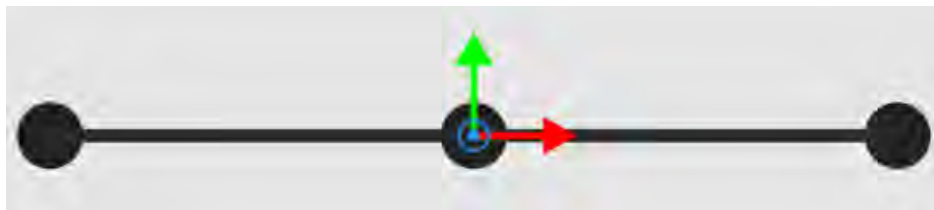
角度しきい値

調整説明

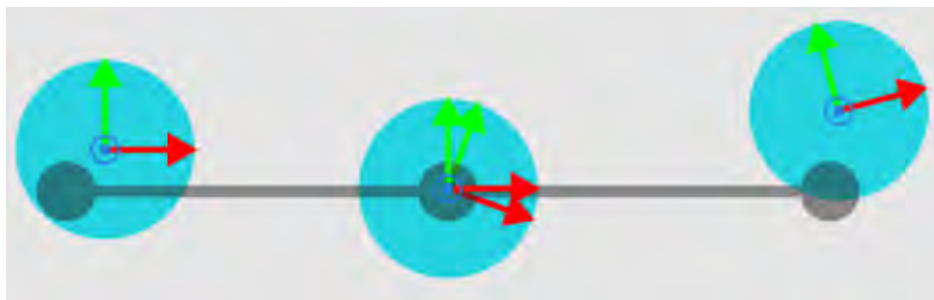
把持点と吸盤位置姿勢との角度を調整します。

調整の例

1. ソフトウェアは、TCP の座標を TCP の X 軸を中心に 180° 回転させて Z 軸を上向きにし、把持点の位置姿勢の向きと一致させます。



2. 把持点の位置姿勢と TCP の各軸との角度を確認します。角度が「角度しきい値」内であれば、把持に成功すると判断され、「角度しきい値」を超えたら失敗すると判断されます。



このモードでは、組合せに物体のロスが発生し、かつこの組合せは回転対称性がなくても構いません。例えば、4つの吸盤がある配列タイプグリッパの場合に、吸盤の番号をそれぞれ 0、1、2、3 とします。3つの物体があり、位置は OOXO であれば、吸盤 0、1、3 をオンにする必要があります。OOXO を 180° 回転させて OXOO にしたら、回転対称性はありません。

把持数

把持総計数制限	把持の総計数を制限します。
計画総計数	対象物を把持する数の上限。
把持済み数	把持済み箱の総計数。
今回の把持数	今回把持する箱の数。
一回の把持数設定	一回の把持の数を指定します。
一回の把持数の上限を設定	一回の把持数の最大数を指定します。

5.4.9.7.3. デパレタイズ用吸盤（単一把持）

「実行モード」「にデパレタイズ用吸盤（単一把持）」を選択した場合に以下のパラメータを調整できます。

吸盤オフセット - オフセット方法

吸盤により箱をデパレタイズする現場では、使用する吸盤の構造は変更できず交換もできませんが、寸法が異なる箱を処理します。対象外の箱を吸着することを回避するために、吸盤のオフセットが必要となります。

吸盤のオフセットは3つの方法があります：「デフォルト」、「カスタマイズ戦略優先度」、「エッジ/コーナーのシーケンスによって」。

デフォルト

優先的に吸盤のコーナーで箱を把持しようとしています。

箱カバー率下限

把持を実行する時に吸盤が箱上面をカバーする面積が箱上面面積に対する割合。カバー率がこの値より小さいとき、しっかりと箱を吸着していないと判断し把持を実行しません。

カスタマイズ戦略優先度

ユーザーが指定した優先度に従って把持します。

戦略の優先度

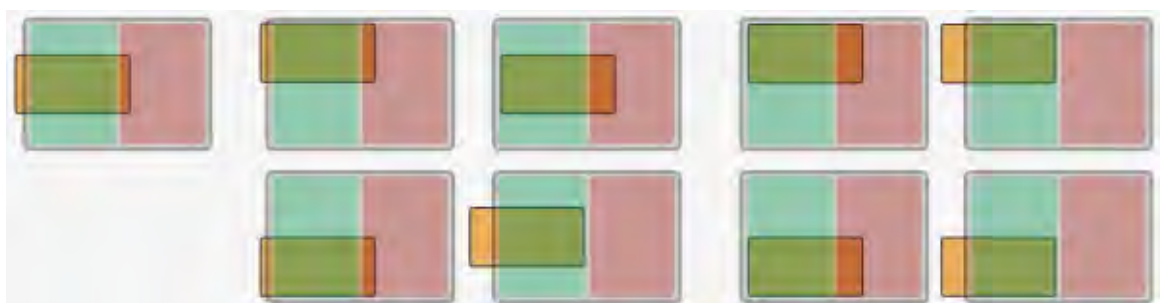
中心合わせ、辺の中心合わせ、コーナー合わせという3つの戦略を選択できます。

下図では、黄色い長方形は箱を、灰色の長方形は吸盤を表します。灰色の長方形の緑の部分はオンにしたブロックで、赤い部分はオフにしたブロックです。



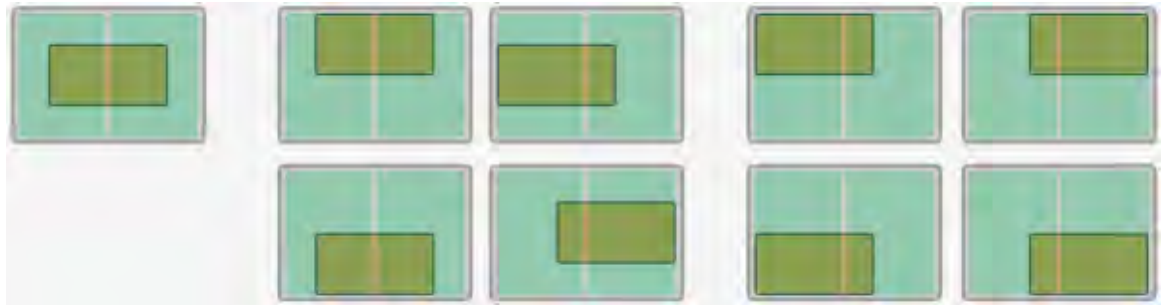
● 1つのブロックをオンにした場合

以下では、左のイメージ図は中心合わせ把持で、中のイメージ図は辺の中心合わせ把持で、右はコーナー合わせ把持です。



● 2つのブロックともオンにした場合

以下では、左のイメージ図は中心合わせ把持で、中のイメージ図は辺の中心合わせ把持で、右はコーナー合わせ把持です。



- より多くのブロックをオンにした場合はこれによって類推します。

箱カバース率下限

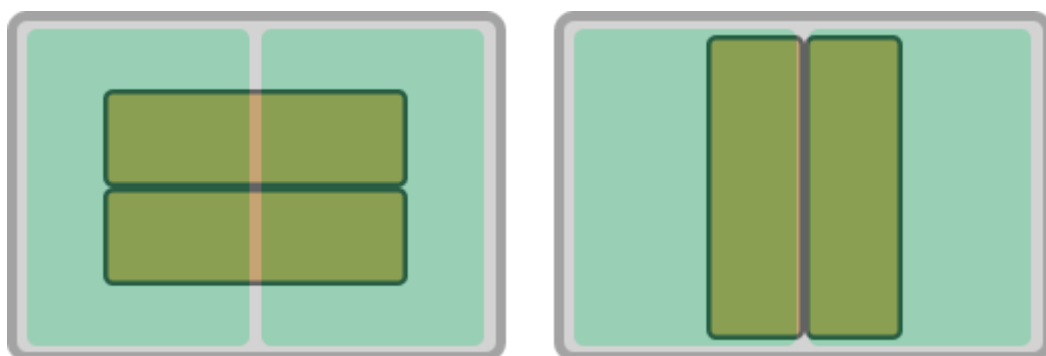
把持を実行する時に吸盤が箱上面をカバーする面積が箱上面面積に対する割合。カバー率がこの値より小さいとき、しっかりと箱を吸着していないと判断し把持を実行しません。

吸盤の向き

箱を把持する時に吸盤と箱の組合せの方向との相対向きを指定します。

「カスタマイズ」を選択すると、**吸盤の長辺の向き**と**方向の基準**を設定できます。これによって、4つの形式を指定できます。

- 吸盤の長辺の向きが箱の組合せの長辺に平行
- 吸盤の長辺の向きが箱の組合せの長辺に垂直
- 吸盤の長辺の向きが箱の組合せ方向に平行
- 吸盤の長辺の向きが箱の組合せ方向に垂直



吸盤の長辺の向きが箱の組合せの長辺に平行（左）、吸盤の長辺の向きが箱の組合せの長辺に垂直（右）

オフセット順序

「（TCP と）基準点との距離によってソート」をチェックすると、TCP と基準点との距離が小

さいほど優先的に実行します。以下のパラメータを設定できます。

基準点 X/Y

基準点を指定します。指定した基準点は仮想空間に現れます。

優先度の高いオフセットのみを保持

実行成功の確率が高い解のみを試行します。

計画解の最大数

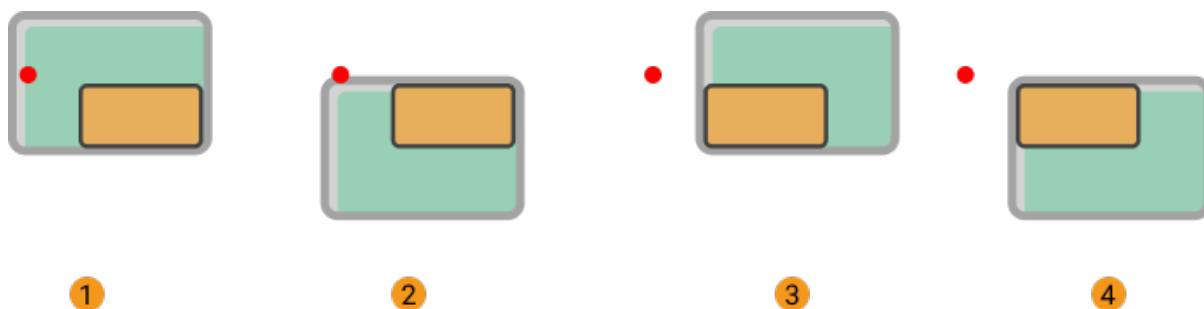
保持するオフセットの解の最大数。

例

下図では、赤いドットを基準点とします。



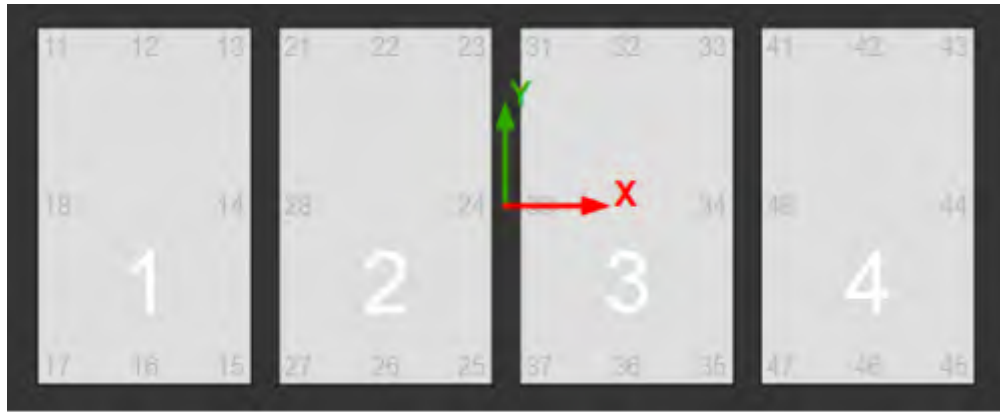
コーナー合わせ 戦略だけを使用すると、オフセットを試行する優先度は以下のようです。



エッジ/コーナーのシーケンスによって

エッジ/コーナーのシーケンスは吸盤コンフィギュレータで自動的に生成されます。

下図に示すように、吸盤の各ブロックのエッジ/コーナーにある番号はエッジ/コーナーのシーケンスです。



箱カバー率下限

把持を実行する時に吸盤が箱上面をカバーする面積が箱上面面積に対する割合。カバー率がこの値より小さいとき、しっかりと箱を吸着していないと判断し把持を実行しません。

エッジコーナーのラベル番号

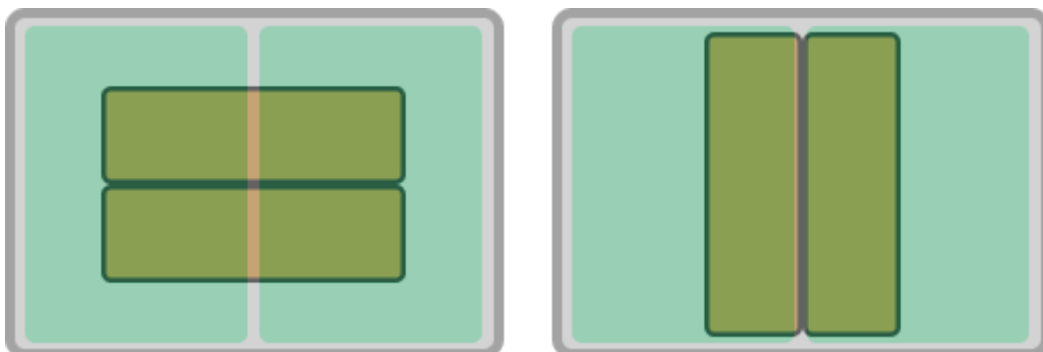
エッジ/コーナーのシーケンスを指定します。例えば、「11, 17, 21, 22」を入力すると、ソフトウェアはこれを順序として把持します。

吸盤の向き

箱を把持する時に吸盤と箱の組合せの方向との相対向きを指定します。

「カスタマイズ」を選択すると、**吸盤の長辺の向き**と**方向の基準**を設定できます。これによって、4つの形式を指定できます。

- 吸盤の長辺の向きが箱の組合せの長辺に平行
- 吸盤の長辺の向きが箱の組合せの長辺に垂直
- 吸盤の長辺の向きが箱の組合せ方向に平行
- 吸盤の長辺の向きが箱の組合せ方向に垂直



吸盤の長辺の向きが箱の組合せの長辺に平行（左）、吸盤の長辺の向きが箱の組合せの長辺に垂直（右）

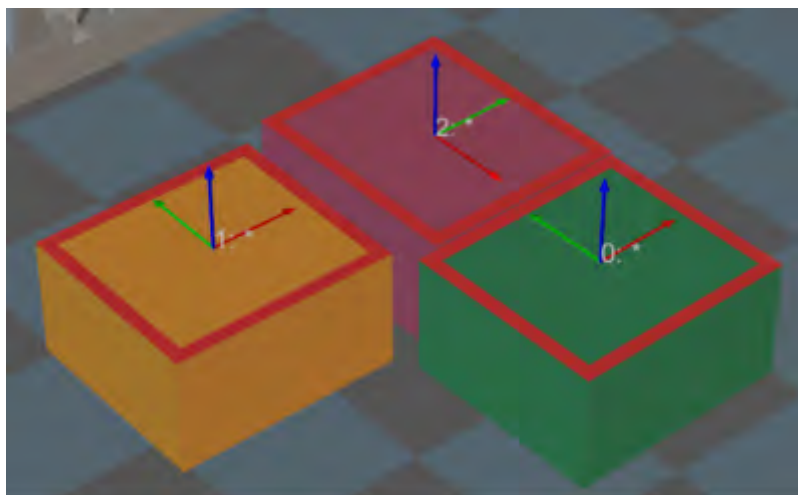
箱落下の検出

吸盤底部に取り付けられたセンサー（DI）を利用して箱の落下を検出します。

吸盤コンフィギュレータでDIの位置を指定すると、箱の把持を計画する時に箱をカバーしたセンサーを自動的に判別することで検出が必要なDI信号を自動的に計算します。

箱のエッジを取り除く距離

実際のデパレタイジングの現場では、センサーが箱のエッジ近くにある場合、箱同士の間隔が大きい、箱のエッジをしっかりと吸着していないなどで箱の落下を誤判定することがあります。**箱のエッジを取り除く距離**を設定することでこの問題を解決できます。センサーが指定した距離内にあれば落下検出に関与しません。下図では、赤い線の部分は箱落下を検出しない範囲を表します。



把持数

把持の合計数を設定します。

計画総計数 把持の合計数の上限。

把持済み数 把持した箱の合計数。

今回の把持数 今回把持する箱の総計数。

同じ位置姿勢で把持することを回避

把持に失敗した場合に使用し、失敗した位置で繰り返して把持することを回避します。

優先度を減少する範囲の半径上限

初期値 0

調整説明 前回把持を実行した位置姿勢との距離が指定した値より小さい位置姿勢に対し、前回の位置姿勢とは同じ位置姿勢として優先度を減少します。

破棄する範囲の半径上限

初期値 0

調整説明 前回把持を実行した位置姿勢との距離が指定した値より小さい位置姿勢に対し、直ちに破棄します。

例 クランクシャフトを把持するとします。最初の把持に失敗し、クランクシャフトがロボットハンドに触れられて位置が変わりました。次回の把持に成功する可能性があるので **優先度を減少する範囲の半径上限** を設定して破棄せずに優先度を下げます。初回の把持にクランクシャフトの位置が変わらない場合は、次回の把持でも成功する可能性がないので **破棄する範囲の半径上限** を設定して直ちに破棄します。

回避リストの最大長さ

初期値 1

調整説明 把持を回避する対象物の位置姿勢あるいは対象物の最大数。

例 「2」と設定した場合に、Mech-Vision から Mech-Vision3つの位置姿勢が出力されたとします。位置姿勢1での把持に失敗してこれを記録します。また位置姿勢2、位置姿勢3での把持に失敗して記録します。位置姿勢2と3しか記録しません。

回避のタイプ

対象物の位置姿勢 対象物の位置姿勢（把持点）を記録します。ある対象物に3つの把持点があるとして、1つの把持点で把持を実行したと記録したら、残りの2つの把持点はまだ試行していないと判断して把持します。

対象物 把持対象物を記録します。ある対象物に対し、その把持点で把持を実行したと記録したら、次回の把持を実行する際に、この対象物のほかの把持点も試行したと判断しこの対象物を優先的に把持しません。

5.4.9.7.4. デパレタイズ用吸盤（複数把持）

「実行モード」に「デパレタイズ用吸盤（複数把持）」を選択した場合に以下のパラメータを調整できます。

箱の組合せ-組み合わせ方式

デパレタイズ用吸盤（複数把持）モードでは、一回に複数の箱をデパレタイズするために、計画を行う前に、ソフトウェアは複数の箱を組み合わせで全体として処理します。

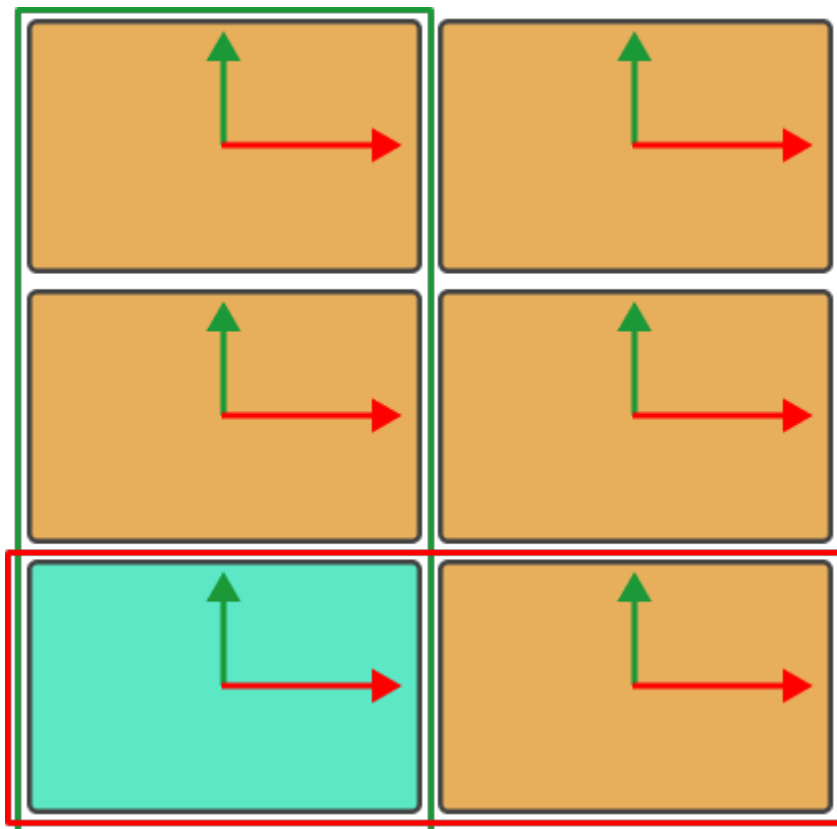
ソフトウェアでは3つの組合せ戦略が選択可能：「デフォルト」（初期設定）、「箱の座標系の軸に沿う」、「カスタマイズ座標系の軸に沿う」。

箱の座標系の軸に沿う

ある箱を組み合わせの出発点とし、この箱の座標系の軸の指向方向を組み合わせ方向として他の箱と組み合わせます。

組合せの制限

実際の状況に応じて組み合わせ方向を指定します。以下のような箱を把持する現場を例とします。左下（水色）の箱を出発点としてX軸（下図の赤い矢印）を指定する場合、赤枠に囲まれている2つ箱を組み合わせます。Y軸（下図の緑矢印）を指定する場合、緑枠に囲まれている2つ箱を組み合わせる可能性があります。



組合せの行数の上限

一回に数行の箱を把持する現場では、組み合わせの行数の上限を設定することができます。

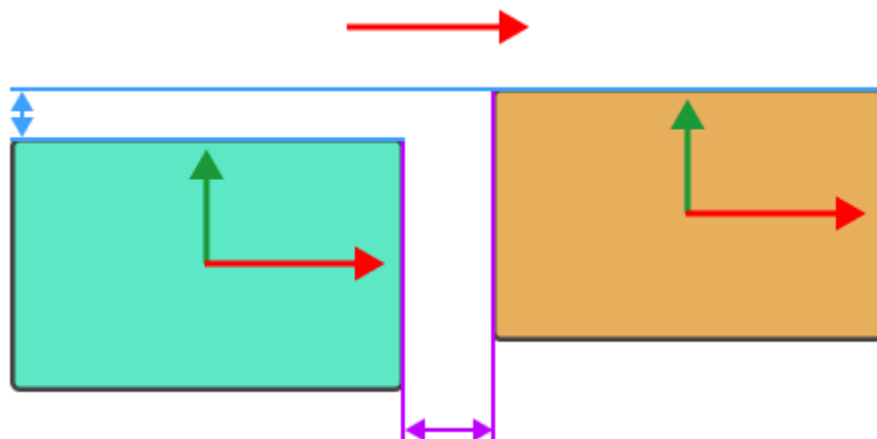
間隔の上限

組み合わせの方向で隣接する2つの箱同士の最大間隔。下図では紫の矢印は、間隔を表しています。隣接する箱の間隔が設定した値より小さい場合にこの2つの箱が組み合わせられると判断されます。

偏差上限

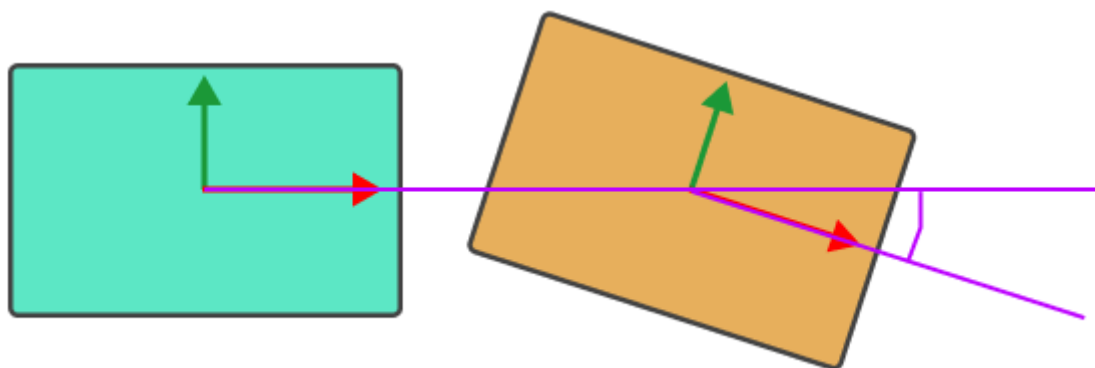
垂直方向に沿って組み合わせようとする2つの箱の最大偏差。下図では垂直方向（赤い矢印の指向方向）に沿って箱を組み合わせます。青い矢印は箱同士の偏差を表しています。偏差が設定

した値より小さい場合にこの2つの箱が組み合わせられると判断されます。



組合せ角度のしきい値

組み合わせようとする箱同士の回転角度。角度が設定した値より小さい場合にこの2つの箱が組み合わせられると判断されます。

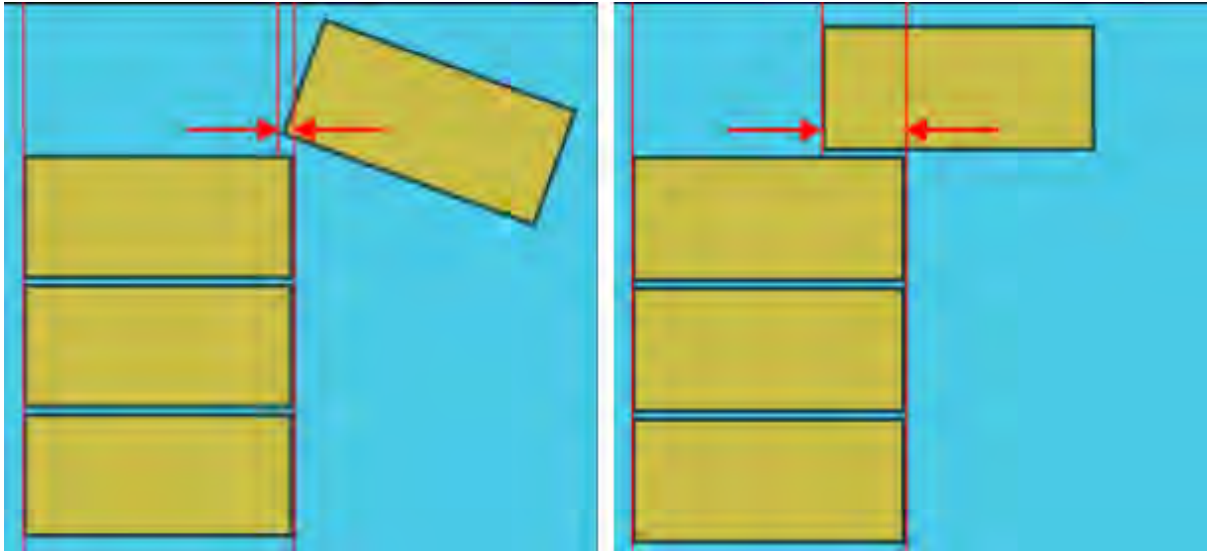


行ごと把持

行の端にある箱が干渉と判断されて行ごと把持することができなくなることがあります。許容干渉を設定することで行を干渉する箱を排除して行ごとを把持することができます。

許容干渉

箱の組み合わせ方向に垂直する方向に他の箱が干渉し、かつ干渉の距離が設定した値より小さい場合、干渉箱を把持対象外の箱と判断し、組み合わせられた箱のみを把持します。干渉距離が設定したあたりより大きい場合に行ごと把持できません。



カスタマイズ座標系の軸に沿う

以下では7つの箱を箱の座標系に沿って組み合わせると：

- X 軸の方向ならば：1 2、3 4、5、6、7。
- Y 軸の方向ならば：1 3、2 4、5 6 7。

1 2、3 4、5 6 7 というように組み合わせたい場合、赤い矢印の方向に沿って組み合わせます。この赤い矢印は、カスタマイズ座標系といいます。

カスタマイズ座標系の X/Y/Z 座標

カスタマイズ座標系の原点位置を調整します。

カスタマイズ座標系の回転角度

カスタマイズ座標系の向きを調整します。

組合せの行数の上限

一回に数行の箱を把持する現場では、組み合わせの行数の上限を設定することができます。

角度のしきい値（カスタマイズ座標系）

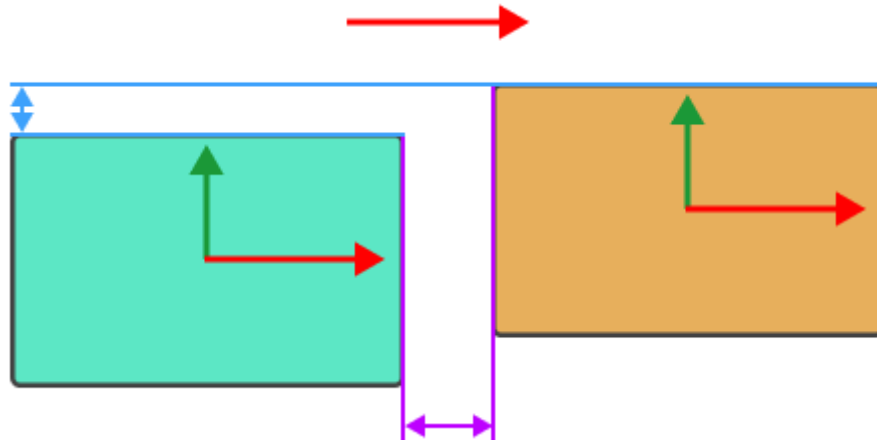
箱を組み合わせる前に、箱の座標系と選択した座標系との角度を確認します。角度が設定した値より大きい場合は、この箱を組み合わせません。

間隔の上限

組み合わせの方向で隣接する2つの箱同士の最大間隔。下図では紫の矢印は、間隔を表しています。隣接する箱の間隔が設定した値より小さい場合にこの2つの箱が組み合わせられると判断されます。

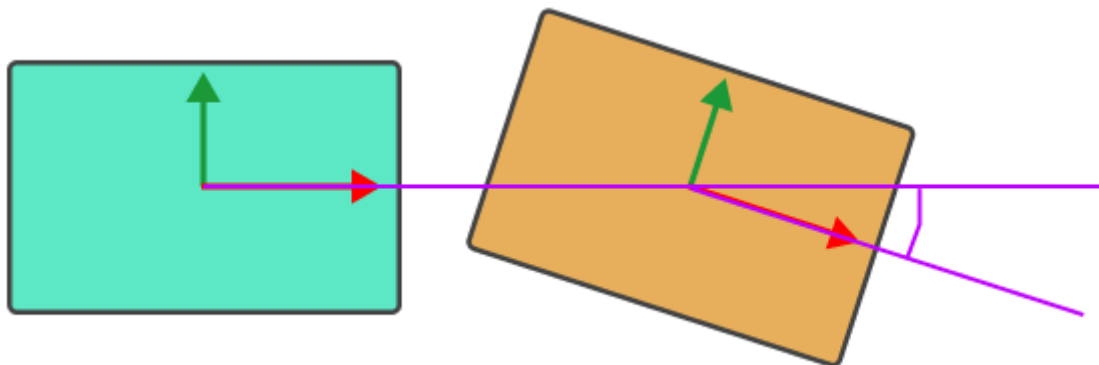
偏差上限

垂直方向に沿って組み合わせようとする2つの箱の最大偏差。下図では垂直方向（赤い矢印の指向方向）に沿って箱を組み合わせます。青い矢印は箱同士の偏差を表しています。偏差が設定した値より小さい場合にこの2つの箱が組み合わせられると判断されます。



組合せ角度のしきい値

組み合わせようとする箱同士の回転角度。角度が設定した値より小さい場合にこの2つの箱が組み合わせられると判断されます。

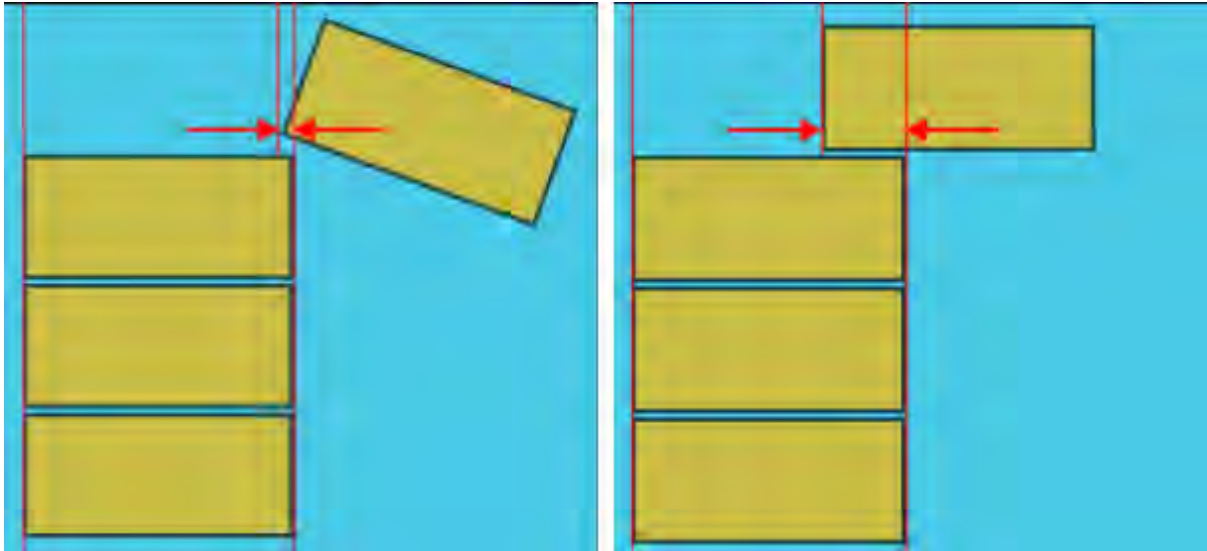


行ごと把持

行の端にある箱が干渉と判断されて行ごと把持することができなくなることがあります。許容干渉を設定することで行を干渉する箱を排除して行ごとを把持することができます。

許容干渉

箱の組合せ方向に垂直する方向に他の箱が干渉し、かつ干渉の距離が設定した値より小さい場合、干渉箱を把持対象外の箱と判断し、組み合わせられた箱のみを把持します。干渉距離が設定した値より大きい場合に行として見なしません。



デフォルト

実は「デフォルト」は特別な「箱の座標系の軸に沿う」戦略です。この戦略を選択すると、箱の2つの座標軸とも組み合わせ方向として組み合わせようとし、かつ一回により多くの箱を把持できる方向を実行します。

吸盤オフセット - オフセット方法

吸盤により箱をデパレタイズする現場では、使用する吸盤の構造は変更できず交換もできませんが、寸法が異なる箱を処理します。対象外の箱を吸着することを回避するために、吸盤のオフセットが必要となります。

吸盤のオフセットは3つの方法があります：「デフォルト」、「カスタマイズ戦略優先度」、「エッジ/コーナーのシーケンスによって」。

デフォルト

優先的に吸盤のコーナーで箱を把持しようとしています。

箱カバー率下限

把持を実行する時に吸盤が箱上面をカバーする面積が箱上面面積に対する割合。カバー率がこの値より小さいとき、しっかりと箱を吸着していないと判断し把持を実行しません。

カスタマイズ戦略優先度

ユーザーが指定した優先度に従って把持します。

戦略の優先度

中心合わせ、辺の中心合わせ、コーナー合わせという3つの戦略を選択できます。

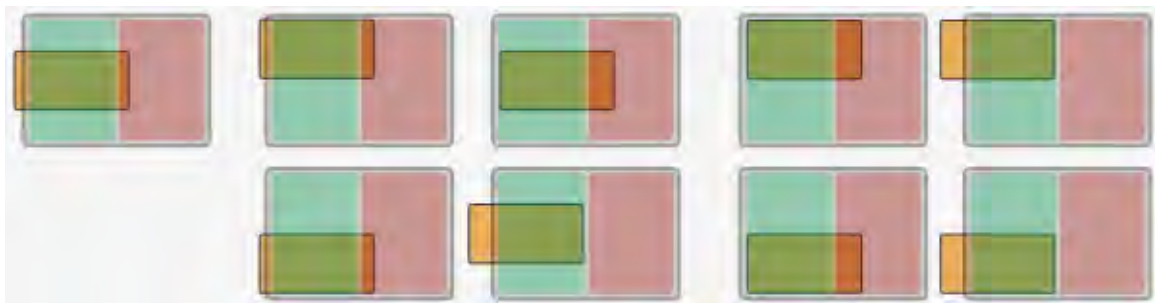
下図では、黄色い長方形は箱を、灰色の長方形は吸盤を表します。灰色の長方形の緑の部分は

オンにしたブロックで、赤い部分はオフにしたブロックです。



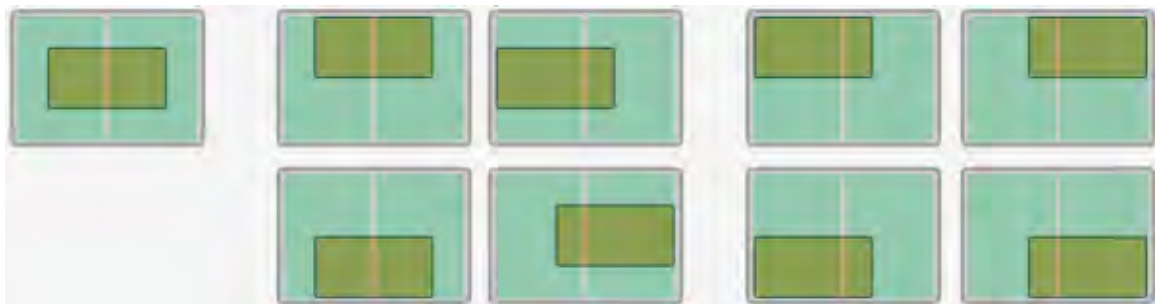
● 1つのブロックをオンにした場合

以下では、左のイメージ図は中心合わせ把持で、中のイメージ図は辺の中心合わせ把持で、右はコーナー合わせ把持です。



● 2つのブロックともオンにした場合

以下では、左のイメージ図は中心合わせ把持で、中のイメージ図は辺の中心合わせ把持で、右はコーナー合わせ把持です。



● より多くのブロックをオンにした場合はこれによって類推します。

箱カバー率下限

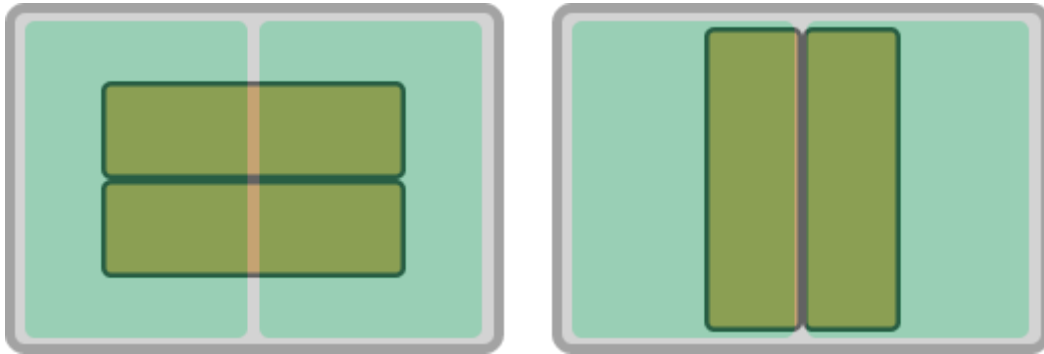
把持を実行する時に吸盤が箱上面をカバーする面積が箱上面面積に対する割合。カバー率がこの値より小さいとき、しっかりと箱を吸着していないと判断し把持を実行しません。

吸盤の向き

箱を把持する時に吸盤と箱の組合せの方向との相対向きを指定します。

「カスタマイズ」を選択すると、**吸盤の長辺の向き**と**方向の基準**を設定できます。これによって、4つの形式を指定できます

- 吸盤の長辺の向きが箱の組合せの長辺に平行
- 吸盤の長辺の向きが箱の組合せの長辺に垂直
- 吸盤の長辺の向きが箱の組合せ方向に平行
- 吸盤の長辺の向きが箱の組合せ方向に垂直



吸盤の長辺の向きが箱の組合せの長辺に平行（左）、吸盤の長辺の向きが箱の組合せの長辺に垂直（右）

オフセット順序

「（TCP と）基準点との距離によってソート」をチェックすると、TCP と基準点との距離が小さいほど優先的に実行します。以下のパラメータを設定できます。

基準点 X/Y

基準点を指定します。指定した基準点は仮想空間に現れます。

優先度の高いオフセットのみを保持

実行成功の確率が高い解のみを試行します。

計画解の最大数

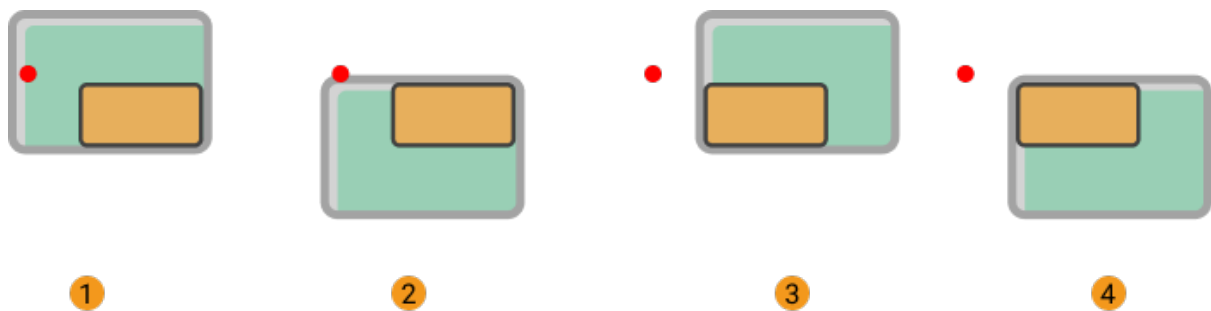
保持するオフセットの解の最大数。

例

下図では、赤いドットを基準点とします。



コーナー合わせ 戦略だけを使用すると、オフセットを試行する優先度は以下のようです。



エッジ/コーナーのシーケンスによって

ユーザーが指定したエッジ/コーナーのシーケンスに従って箱を把持します。

エッジ/コーナーのシーケンスは吸盤コンフィギュレータで自動的に生成されます。下図に示すように、吸盤の各ブロックのエッジ/コーナーにある番号はエッジ/コーナーのシーケンスです。



箱カバー率下限

把持を実行する時に吸盤が箱上面をカバーする面積が箱上面面積に対する割合。カバー率がこの値より小さいとき、しっかりと箱を吸着していないと判断し把持を実行しません。

エッジコーナーのラベル番号

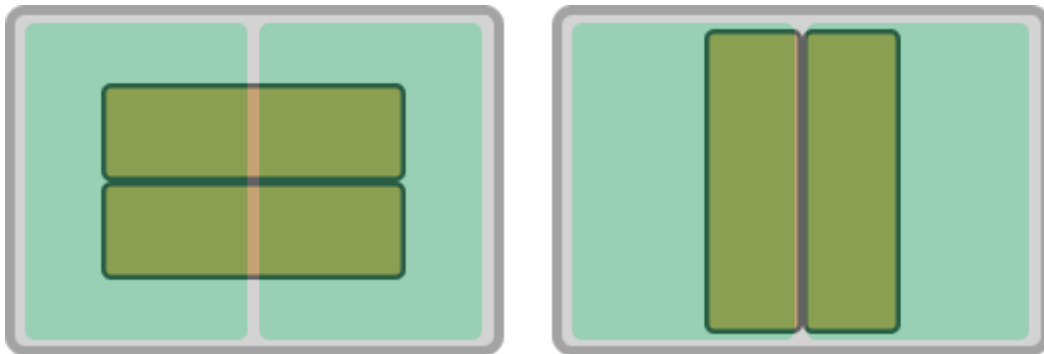
エッジ/コーナーのシーケンスを指定します。例えば、「11, 17, 21, 22」を入力すると、ソフトウェアはこれを順序として把持します。

吸盤の向き

箱を把持する時に吸盤と箱の組合せの方向との相対向きを指定します。

「カスタマイズ」を選択すると、**吸盤の長辺の向き**と**方向の基準**を設定できます。これによって、4つの形式を指定できます。

- 吸盤の長辺の向きが箱の組合せの長辺に平行
- 吸盤の長辺の向きが箱の組合せの長辺に垂直
- 吸盤の長辺の向きが箱の組合せ方向に平行
- 吸盤の長辺の向きが箱の組合せ方向に垂直



吸盤の長辺の向きが箱の組合せの長辺に平行（左）、吸盤の長辺の向きが箱の組合せの長辺に垂直（右）

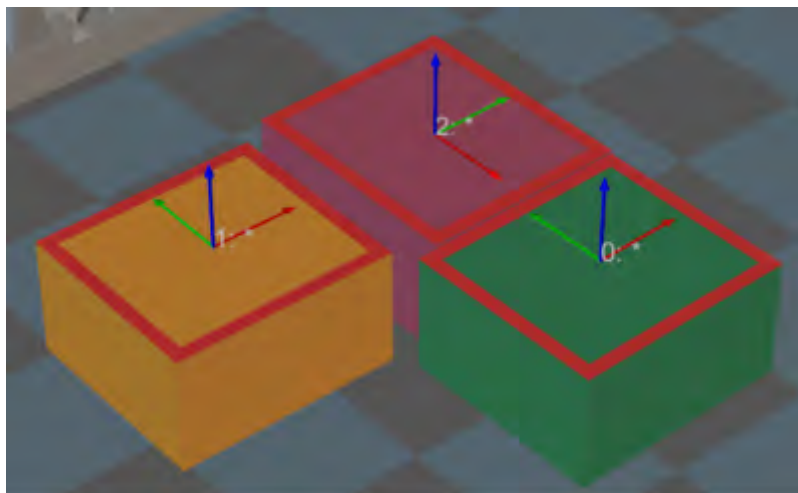
箱落下の検出

吸盤底部に取り付けられたセンサー（DI）を利用して箱の落下を検出します。

吸盤コンフィギュレータでDIの位置を指定すると、箱の把持を計画する時に箱をカバーしたセンサーを自動的に判別することで検出が必要なDI信号を自動的に計算します。

箱のエッジを取り除く距離

実際のデパレタイジングの現場では、センサーが箱のエッジ近くにある場合、箱同士の間隔が大きいか、箱のエッジをしっかりと吸着していないなどで箱の落下を誤判定することがあります。**箱のエッジを取り除く距離**を設定することでこの問題を解決できます。センサーが指定した距離内にあれば落下検出に関与しません。下図では、赤い線の部分は箱落下を検出しない範囲を表します。



把持数

把持総計数制限	把持の合計数を設定します。
計画総計数	把持の合計数の上限。
把持済み数	把持した箱の合計数。
今回の把持数	今回把持する箱の総計数。
一回の把持数設定	一回の把持数を設定します。
一回の把持数の上限を設定	一回の把持数の上限を設定します。

5.4.10. その他

5.4.10.1. 実行を中止

機能

複雑な現場に使用されます。例えば複数の階層がある **ステップの組合せ** 実行中にプロジェクトの実行を停止したい場合にワークフローにこのステップを接続できます。

パラメータ説明

調整可能なパラメータはありません。

停止原因

停止する原因を入力します。

5.4.11. 共通パラメータ

5.4.11.1. 移動ステップの共通パラメータ

移動目標点を送信

デフォルトではチェックが入っています。相手側（ロボットなど）に移動目標点の位置姿勢を送信します。チェックを外したら送信しませんが、この目標点位置姿勢は依然として経路計画の一部です。

移動コマンドの後の非移動コマンドをスムーズに実行することを試行

デフォルトではチェックが外れており、**移動ステップ**の間に「ビジョン処理による認識」、「DOを設定」、「DIをチェック」などの**非移動ステップ**を接続すると、ロボット移動経路の計画を中断し、ロボット実機動作中に一時中止することがあります。チェックを入れると現在のステップが完了しなくても続行することが可能です。これにより、ロボットがよりスムーズに動作します。ただし、ステップが途中で終了する可能性があります。



ステップが途中で終了する原因は何ですか？

Mech-Viz 実行中、ロボットに同時に複数の位置姿勢を送信しますが、最後の位置姿勢が現在のロボットの関節角度と一致しているかのみを判断します。一致していると判断したら、ロボットが最後の位置に到達していると思なします。例えば、10の移動ステップがある経路では、移動ステップ5の位置姿勢は最後のステップの位置姿勢とは同じとします。ロボットが低速に動作し、移動ステップ5の位置に到達したらその位置の関節角度をMech-Vizに送信します。経路では、移動ステップ5の位置姿勢は最後の移動ステップの位置姿勢とは同じなので、Mech-Vizはロボットの動作が完了したと判断して途中で実行を終了します。

配置された対象物との衝突を検出しない

デフォルトではチェックが外れており、配置された対象物との衝突を検出しません。チェックを入れると、ロボット本体・ロボットハンドと配置された対象物との衝突を検出します。

パレタイジングのシーンでは、以下場合があります。

1. ロボットが段ボール箱を配置するときに配置済みの箱と軽く衝突する（箱の凹みや変形などが発生しない）ことがあります。Mech-Vizではこのような衝突を検出したら別の配置位置を計画するためパレットが満杯になりません。
2. 普通、吸盤を使用する場合、TCPを吸盤の表面でなく、モデル内部に設定するため、物体を吸着する時、吸盤と把持する箱のモデルと重なります（ソフトウェアでは吸盤と把持対象物との衝突を検出しない）。ロボットが箱を配置してから、把持された箱のモデルはシーンのモデルになり、吸盤がシーンの箱のモデルと衝突すると判断してメッセージを表示してパレタイジングが続行できなくなります。

これをチェックするとロボット本体やロボットハンドと配置済み対象物モデルとの衝突を検出しないので上記の問題を解決できます。

点群との衝突検出モード

現場の状況に応じて設定してください。普通、デフォルトの **自動** を使用します。ロボットが物体を把持する前の移動ステップは、**チェックしない** を選択し、把持した後の移動ステップを **チェック** を選択します。

- 自動** 初期値。「ビジョン処理による移動」ステップと「ビジョン処理による移動」ステップに依存する「相対移動」の点群衝突だけを検出します。
- チェックしない** 移動ステップの点群衝突をいっさい検出しません。
- チェック** 全ての移動ステップの点群衝突を検出します。



衝突検出、**衝突検出設定**、**点群と他の対象物間の衝突を検出** をオンにすれば、Mech-Viz は経路計画を行うときにロボットモデル、ツールモデルと点群との衝突を検出します。普通、点群衝突の設定は、ロボットが把持を実行するときに把持対象物との衝突を検出するためです。空間内にノイズがあれば、ソフトウェアは対象物を把持する前の経路を計画するときに、ノイズがロボットモデル、ツールモデルと接触して点群との衝突が誤って判断されて経路計画の誤りが発生します。

対象物の対称性を使用しない

このパラメータは、**目標点タイプ** が **対象物位置姿勢** の移動ステップ（目標点タイプが物体の位置姿勢の移動ステップ、パレタイジングステップなど）に対してのみ有効です。目標点のタイプが関節角度、TCP位置姿勢の移動ステップには無効です。

- 無し** 無し：初期値。対称性を無効にしません。
- Z 軸** Z 軸の対称性のみを使用しません。
- XY 軸** X、Y 軸の対称性のみを使用しません。
- 全て** 全ての対称性を使用しません。

物体の対称性を使用しないように設定すると、ロボットが物体の位置姿勢に正確に到達して対象物を配置します。



対象物を把持できない場合に **リソース**、**対象物設定** の **回転対称** を設定します。対象物に対称性がある場合、複数の候補位置姿勢があります。Mech-Viz では対象物の把持を計画するときに、デフォルト位置姿勢が把持できない場合、候補位置姿勢を試行します。物体対称位置姿勢と Mech-Vision によって出力された元位置姿勢とは一致しなければ、ロボットによる対象物の配置位置姿勢は一致しないことがあります。

5.4.11.2. 把持された対象物との衝突検出モード

シーンの物体との衝突を検出しない

デフォルトではチェックが外れています。チェックすると把持された対象物とシーンの物体、ロボットとの衝突を検出せず、ソフトウェアの計画の速度を改善できます。普通、ロボットが

対象物を把持したあとの移動ステップに使用します。

衝突を検出しなければ、衝突が発生するリスクがあるので注意して使用してください。



衝突検出 ▶ **把持されている対象物の設定** ▶ **把持されている対象物とその他の物体との衝突を検出** をオンにすれば、把持されている対象物とシーンの物体のモデル・ロボットとの衝突を検出します。パレタイジングのシーンでは、ビジョン処理により取得した箱の寸法にはささやかな誤差があり、把持を実行する時に箱同士の摩擦が発生しますが衝突は発生しません。

パレタイズのシーンでは、「シーンの物体との衝突を検出しない」をチェックしても対象物と配置済みの箱との衝突検出に干渉しません。箱の山の下にシーンの物体がある場合、経路計画のためにこれをチェックしてください。

点群との衝突を検出しない

デフォルトではチェックが外れています。チェックすると **把持対象物のモデル** と **シーンの点群** が衝突するかを検出しないので、Mech-Viz での衝突検出計算量を低減し、計画の効率を向上させることができます。また、把持対象物と点群との衝突の誤検出も回避できます。



- **衝突検出** ▶ **把持されている対象物の設定** ▶ **把持されている対象物とその他の物体との衝突を検出** と **点群設定** ▶ **点群と他の対象物間の衝突を検出** をオンにすれば、把持されている対象物とシーンの物体のモデル・ロボットとの衝突を検出します。
- Mech-Vision が点群情報と対象物モデル情報を Mech-Viz に送信するときに、点群と対象物モデルは密着している状態です。ロボットが対象物を把持するときに、モデルはロボットの経路に沿って移動し、把持対象物モデルは点群と衝突します。
- 把持対象物は物体の点群と衝突する場合、この衝突をチェックすると Mech-Viz の計算量が増え、計画時間が長くなります。

5.4.11.3. 非移動ステップの共通パラメータ

実行をスキップ

スキップしない	初期値、このステップをスキップしません。
シミュレーション時	シミュレーションを実行する時にこのステップをスキップします。 出口は スキップ時の出口のインデックス によって決定されます。
いつも	シミュレーション時も実際に実行中にも実行をスキップします。出口は スキップ時の出口のインデックス によって決定されます。
調整説明	シミュレーション時 または いつも に設定すると、プロジェクト実行中にこのステップをスキップして後のステップを実行します。プロジェクトに「DI をチェック」ステップを配置したら、シミュレーション時に外部信号がなく、実行をスキップしなければプロジェクトの実行はここで停止することになります。このパラメータを設定すると、このステップを実行せずに次のステップを実行してシミュレーションを完了します。

スキップ時の出口のインデックス

実行をスキップをシミュレーション時またはいつもに設定すれば有効になります。このパラメータは、スキップしてから実行する出口を指定します。

5.4.11.4. 基本的な移動設定

基本的な移動設定は、移動ステップの主なパラメータを設定し、ロボットが目標点に到達する速度と運動方式を決定します。

把持と配置設定

未指定	初期値
把持	「ビジョン処理による移動」の前の移動ステップ。
放置	「ビジョン処理による移動」の後の移動ステップ。
調整説明	Mech-Viz プロジェクトのロジック検査に使用されます。現場の実行手順に応じて、把持をしてから配置するという基本的な原則に従って設定します。

運動タイプ

関節運動	ロボットが円弧に沿って走行します。スムーズに走行できるので特異点を回避することが可能。ロボットの移動範囲が広くて高い精度を求めないシーンに適用しています。
直線運動	ロボットが直線に沿って走行します。溶接や接着剤塗布、把持など高い精度が求められるシーンに適用しています。

速度&加速度

速度&加速度は、ロボット動作のスピードを決定します。普通、加速度の値を速度より小さく設定します。加速度の値を速度よりも大きくしたらロボット動作がスムーズでなくなります。



ロボットが安定に把持を実行するように、「ビジョン処理による移動」とその前後のステップの速度を低く設定してください。

ブレンド半径

初期値	50.00mm
-----	---------

調整説明 調整説明：通常、初期値を使用すればいいです。

ブレンド半径とは、回転する位置が目標点までの距離であり、大きいほどロボットの動きはスムーズになります。ロボットが狭いスペースで動作する場合、ブレンド半径を大きく設定する必要はありません。

ロボットがより広いスペースで動作し、障害物がなくて2つの経路の間に距離が遠い場合、ロボットがスムーズに動くように、ブレンド半径を大きく設定することができます。

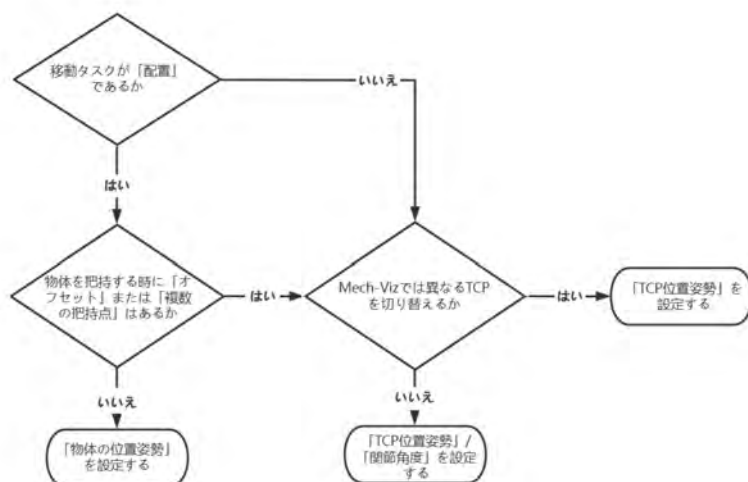
5.4.11.5. 目標点タイプ

TCP TCP 座標系の X、Y、Z 値、およびオイラー角または四元数によって目標点を表示します。

関節角度 ロボットの各関節の数値で目標点を表示します。

対象物位置姿勢 対象物座標系の X、Y、Z 値、およびオイラー角または四元数によって目標点を表示します。

下図のように目標点タイプを選択します。



位置姿勢を編集 位置姿勢を編集します。コピーと貼り付けができ、四元数とオイラー角の二つの形式に対応しています。

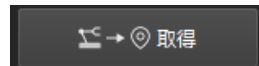
位置姿勢を変換 変換を定義することで現在の位置姿勢を新しい位置姿勢に変換します。

位置姿勢を校正

ABB ロボット三点法と類似しており、対象物の座標系を計算します。対象物の回転位置姿勢を確認できない場合に使用します。例えば、傾斜した直方体に対し、位置姿勢を校正することでその位置姿勢を計算し、ロボットが計算された位置姿勢に沿って動作させることが可能です。

関節角度を編集

位置姿勢を調整する方法と同じく、コピーと貼り付けができます。ラジアンと角度の二つの形に対応可能です。



仮想ロボットを指定した経路点に移動します。



ロボットを移動する目標位置姿勢を読み取り、経路点に設定します。



可能な関節角度の解を全部表示します。

5.4.11.6. 関節角度の制約条件

概念：

- 肩** 腕関節の中心と軸 1 との相対関係。軸1とは、ロボット1軸の回転中心軸です。
- 肘** 腕関節と前腕との相対関係。前腕とはロボットの 2、3 軸の回転中心軸です。
- 手首** ロボット 5 軸です。5 軸角度の正負は、腕の回転を表します。



オプション：

- 自動** 関節を制約しません。各軸の回転が最小の位置姿勢を最適解とします。
- 保持** ロボットの現在の状態を状態Aとしてロボットを制約します。Mech-Viz プロジェクトでは、次の移動の解を選択するとき、状態 A と一致する解を有効なものとしてします。ロボット 3 軸を例に、現在の状態では 3 軸は正方向に向いているので移動の解を求めるときに 3 軸だけが正方向の解を有効とします。
- その前** 腕関節中心が軸 1 の前にあります。
- その後** 腕関節中心が軸 1 の後ろにあります。

表示

をクリックするとこの位置姿勢が対応するすべての関節角度が表示されます。いずれをクリックしてシミュレーションエリアでロボットの姿勢を表示でき、異なる制約条件の下でその他の可能な関節角度を確認することができます。



1. 関節角度の制約条件は6軸ロボットにのみ有効です。4軸ロボットは、肩や肘、手首の回転はないとされます。
2. ソフトウェアでは、この機能はカスタマイズパレットパターンと事前計画したパレットパターンに対応できません。デフォルトでは「自動」に設定し、肩、肘、手首を変えないのでロボットが動作中に特異点を経過しません。

5.5. プロジェクトのシミュレーションと最適化

プロジェクトのリソース設定を完了し、プロジェクトを構築したあとシミュレーションを実行できます。

新規作成したプロジェクトを最適化するために以下の手順を実行してください。

深い箱から対象物を把持するシーンでは、ロボットハンドと箱などの物との衝突を防止することが非常に重要です。シミュレーションを実行する前に衝突検出を設定してください。衝突検出の設定を完了してから「ミュレート」をクリックしてシミュレーションを実行できます。

衝突検出

シミュレーションに衝突発生のお知らせや位置到達不能などの場合に、計画履歴とログで問題を確認して問題を突き止めて解決することができます。

計画履歴

ログ

プロジェクトを最適化したあと、ロボット実機を実行して検証します。

ロボット実機の実行

シミュレーションも実機の実行も問題がないと、計算設定を変更してソフトウェアの実行速度を改善することができます。

計算設定

5.5.1. 衝突検出

本節では衝突検出の設定について紹介します。以下の内容について説明します：

概要

部品ロード・アンロード、デパレタイジング・パレタイジングなどの応用では、プロジェクトをスムーズに実行するために、ロボットがコンテナなどの障害物との衝突を防ぐことが重要です。Mech-Viz は衝突検出機能によりロボット動作中の衝突を防ぐことができます。ロボット運動経路を計画するとき、Mech-Viz の仮想空間では衝突が発生する部分がハイライト表示され、プロジェクトを停止して現場の衝突を回避します。

計算設定

計算設定は、ソフトウェアにおける衝突の計算と記録を修正し、プロジェクトの実行速度を調整するために使用されます。

プロジェクトのデバッグ過程で [計算設定](#) のパラメータグループを **各解に対応する完全な衝突接触を計算+計画履歴に保存** に設定することを推奨します。これによって問題をすばやく突き止めて解決することができます。

衝突検出設定

衝突モデルを組み合わせることで衝突検出を実行します。衝突検出を実行する前に、リソースに衝突モデルを追加して設定します。

- ロボットハンドモデルの追加と設定は [ロボットハンド](#) をお読みください。
- シーンの物体のモデルの追加と設定については [シーンの物体](#) をお読みください。
- 対象物モデルは Mech-Vision により提供されたデータに基づいて自動的に生成されます。設定方法は以下のとおりです。

Mech-Viz では以下の物体間の衝突を検出します：

1	ロボット関節	ロボット関節
2	ロボット関節	シーンの物体
3	ロボット関節	ロボットハンド
4	シーンの物体	ロボットハンド

また、以下の物体間の衝突検出も設定できます。

衝突検出 パネルで、「衝突検出設定」をクリックして衝突検出設定画面を開きます。

点群と他の対象物間の衝突を検出

左の「点群設定」パレットで**点群と他の対象物間の衝突を検出**をチェックすると、点群とロボットハンドとの衝突を検出します。また、点群とロボット関節、把持されている対象物の衝突検出を設定することもできます。



「ビジョン処理による移動」ステップと「ビジョン処理による移動」に依存する「相対移動」ステップにだけ点群衝突検出を実行し、その他の移動ステップは実行しません。

点群設定 - 点群立方体の辺の長さ

点群立方体は点群の点を中心に生成した立方体です。衝突を検出する時、立方体と他の物体との衝突は点群の衝突とされます。

衝突検出の精度と速度の要求を満たすようにパラメータを調整します。

ロボットハンド設定 - 衝突体積のしきい値

点群立方体とロボットハンド衝突モデルの衝突体積の最大許容値。衝突体積がこの値を超えると衝突が発生したと判断します。

実際に応じて設定してください。

ロボット関節設定 - ロボット関節

ロボット関節には、手首、前腕、上腕、ベースがあります。デフォルトでは関節と点群の衝突を検出しません。

実際に応じて設定してください。

ロボット関節設定 - 衝突体積のしきい値

点群立方体とロボット関節衝突モデルの衝突体積の最大許容値。衝突体積がこの値を超えると衝突が発生したと判断します。

実際に応じて設定してください。

把持されている対象物とその他の物体との衝突を検出

把持されている対象物とその他の物体との衝突を検出します。対象物を移動する過程で発生する衝突を検出します。

把持されている対象物のモデルタイプに応じて**直方体**または**外部モデル**を選択してください。

直方体

直方体モデルの追加方法

Mech-Viz は Mech-Vision によって提供された対象物の寸法に基づいて直方体の衝突モデルを自動的に生成します。そのため Mech-Vision で設定を行う必要があります。

1. Mech-Vision で「平面点群の位置姿勢とサイズを計算（複数種類の対象物を扱う現場

）」または「対象物の寸法を読み込む（単一種類の対象物を扱う現場）」を使用して対象物の寸法を取得します。

2. 寸法データを「出力」ステップで出力すると Mech-Viz では寸法を取得して衝突モデルが生成されます。

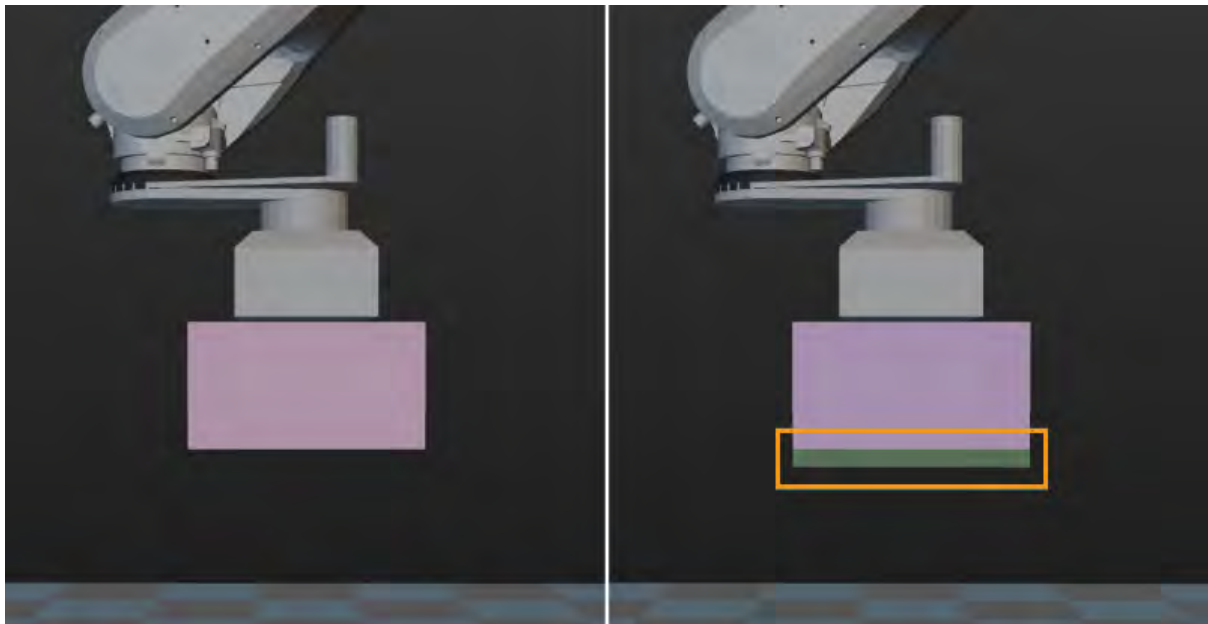


出力される「把持点位置姿勢」と「物体寸法」は一致しなければなりません。

直方体底部の安全距離設定

混載パレタイジングを実行する場合、直方体の対象物モデルに底部安全距離を設定する必要があります。底部安全距離を設定すると、対象物モデルの底部に相応の衝突検出範囲を設けます。この範囲に他の物体が干渉したら、それは対象物との衝突と判断されます。これにより、箱を把持してから移動する過程で配置済みの箱との衝突を回避できます。

直方体の対象物を扱う混載パレタイジングを実行するとき、把持と配置の過程で底部の安全距離内の衝突を検出をオンにして直方体底部の安全距離を設定します。



衝突体積のしきい値

点群立方体と把持されている対象物衝突モデルの衝突体積がこの値を超えると衝突が発生したと判断します。点群と他の対象物間の衝突を検出をオンにすると設定できます。

実際に応じて設定してください。

外部モデル

追加方法

1. 対象物の stl モデルと binvox モデル Mech-Viz のプロジェクトフォルダにある **collision_models** フォルダ（これがない場合に新規作成してください）に移動します。
2. Mech-Vision では相応のステップの「位置姿勢分類ラベル」ポートでラベルを確認し、

モデルファイルをラベル名に変更します。Mech-Vizのリソースにこの対象物を追加した場合、その名前もラベル名と一致するように設定してください。

3. Mech-Vision では [点群を外部サービスに送信](#) ステップの**対象物情報を送信**パラメータをチェックし、**すべての入力はカメラ座標系にある**のチェックを外してこのステップのすべての入力ポートをデータフローにつなぎます。



出力した点群と位置姿勢はいずれもロボット座標系にある必要があります。

衝突体積のしきい値

点群立方体と把持されている対象物衝突モデルの衝突体積がこの値を超えると衝突が発生したと判断します。**点群と他の対象物間の衝突を検出**をオンにすると設定できます。

実際に応じて設定してください。

5.5.2. 計画履歴

計画履歴の使用について紹介します。以下の内容について説明します。

概要

計画履歴の結果の構造はツリー構造であり、Mech-Viz の計画プロセスが詳細かつ完全に記録されます。

ログに比べて、計画履歴では、計画の各ノード（特に失敗したノード）をより詳細に表示できます。

- 一部のエラーはシリアルエラーです。1つの子ノードが計画に失敗した場合、全体的な計画は失敗します。例えば、ある関節角度が正しく設定されていない場合、ロボットは計画された位置に到達できません。
- 並列エラーもあります。1つの子ノードのみが正常に計画されている場合、全体的な計画は成功します。例えば、Mech-Vision によって計算された把持点には、ノードのいずれかが正常に実行された場合、それは、把持位置姿勢で対象物を把持できることを意味します。



プロジェクトを実行する過程で、失敗したノードに特に注意を払う必要があります。

衝突結果の可視化

衝突接触の計算と記録

衝突接触の計算と記録は普通、衝突結果の可視化のために使用されます。

[計算設定](#) の各解に対応する**完全な衝突接触を計算 + 計画履歴に保存**を使用して衝突接触を記録して計画履歴に保存することができます。

計画履歴の衝突結果のグラフィックスなプロンプト

メニューバーの「ディスプレイ」をクリックして、**計画中に衝突を表示**にチェックを入れます（デフォルトではチェックが入っています）。

シーンとの衝突	衝突が発生したステップの名前をクリックして、衝突したすべての対象物がハイライト表示される
	衝突した対象物をクリックして、それがハイライト表示される
点群との衝突	衝突が発生したステップの名前をクリックすると、衝突したすべての対象物がハイライト表示される
	衝突体積 をクリックして、衝突した点群がハイライト表示される
	点群と衝突したときに、 衝突接触を記録 しないと、別の衝突する対象物のみがハイライト表示される
把持対象物との衝突	把持対象物と衝突が発生した場合、それがハイライト表示される

よく発生する失敗と解決法

ロボットの自己衝突

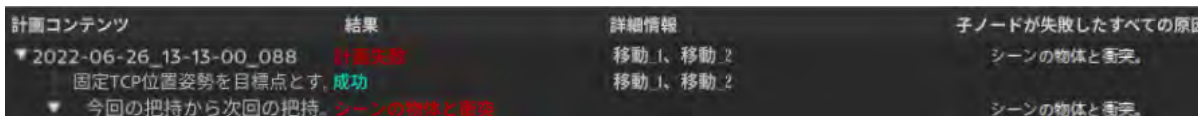


計画コンテンツ	結果	詳細情報	子ノードが失敗したすべての原因
▼ 2022-07-18_18-09-24_567	ロボット自己衝突	手首は前腕と衝突 関節角度: (3.40339, -1.5708, 3.21141, -1.5708, 0, ...)	子ノードが失敗したすべての原因 ロボット自己衝突。
現在のタスク	移動_2		
関節1	手首		
関節2	前腕		

問題 前腕とベースの衝突が発生しました。

解決法： ツールの位置姿勢を調整し、自己衝突を回避します。

シーンの物体との衝突



計画コンテンツ	結果	詳細情報	子ノードが失敗したすべての原因
▼ 2022-06-26_13-13-00_088	計画失敗	移動_1、移動_2	子ノードが失敗したすべての原因 シーンの物体と衝突。
固定TCP位置姿勢を目標点とす。	成功	移動_1、移動_2	
▼ 今回の把持から次回の把持。	シーンの物体と衝突		シーンの物体と衝突。

問題 ロボットハンドとシーン[1]の衝突が発生しました。

解決法： 衝突が発生したシーンの物体の位置をチェックします。それが変わっていない場合に現在のステップの相応の位置姿勢を調整します。

移動目標点に到達不能

計画コンテンツ	結果	詳細情報	子ノードが失敗したすべての原因
▼ 2022-07-18 16:56:09.450	移動目標点に到達不能	0/0/0 初期/ソフトマージン/最終値の数、(荷: 自動、対: 自動、手置: 自動)	移動目標点に到達不能。
現在のステップ	移動 1, ロボットハンド名...	TCP位置姿勢: (1.7, 2.54569, 2.14608, 0.5, -0.5, 0.5)	
搬送開始の基準ステップ		関節角度: 0	

問題 ロボットの移動範囲を超えたため、ロボットがこの位置に到達できません。

解決法: 現在のステップの相応の位置姿勢をロボットの到達可能な範囲に調整します。

衝突体積がしきい値超過

計画履歴	結果	詳細情報	子ノードが失敗したすべての原因
▶ 2022-07-18 18:45:30.838	計画成功	back to HOME	
▶ 2022-07-18 18:45:32.325	計画成功	move 1	
▼ 2022-07-18 18:45:50.483	衝突体積がしきい値超過	visual_move_1, relative_move_1, via_point, via_po...	移動目標点に到達不能。シーンの物体と衝突点群との衝突点の数が制限を...
▼ 1/22回の記録を試行	衝突体積がしきい値超過	visual_move_1, relative_move_1, via_point, via_po...	シーンの物体と衝突点群との衝突点の数が制限を超える。
▼ [visual_move_1] に対称性/把...	衝突体積がしきい値超過	visual_move_1, relative_move_1, via_point, via_po...	点群との衝突点の数が制限を超える。
▼ 現在のタスク	衝突体積がしきい値超過	visual_move_1, relative_move_1, via_point, via_po...	
▼ ロボットハンドの点群衝突を...	衝突体積がしきい値超過	visual_move_1, relative_move_1, via_point, via_po...	
▼ 現在のタスク	衝突体積がしきい値超過	visual_move_1, relative_move_1, via_point, via_po...	
▼ 開始1	衝突体積がしきい値超過	visual_move_1, relative_move_1, via_point, via_po...	
▼ ロボットハンド	衝突体積がしきい値超過	visual_move_1, relative_move_1, via_point, via_po...	
▼ 衝突点の数	84		

問題 ロボットハンドと点群の衝突体積が指定したしきい値を超えました。

解決法: 検出した衝突体積と現場の状況に応じて、衝突体積のしきい値を変更します。
あるいはTCP、把持点、ロボットハンド衝突モデルの位置姿勢を調整しま
す。あるいはTCP、把持点、ロボットハンド衝突モデルの位置姿勢を調整し
ます。

5.5.3. ロボット実機の実行

以下の手順を実行してロボット実機を操作します。

1. ロボットを接続する

Mech-Vizを使用してロボット実機を操作する前に、**ロボットの通信設定**を完成しなければなりません。

2. シミュレーションを中止する

Mech-Center > **設定** > **Mech-Viz** タブで**実行をシミュレート**のチェックを外します。

3. ロボットの稼働速度を下げる



事故を防止するために、ロボット実機を実行する前に必ずロボットの稼働速度を下げてください。

Mech-Vizのツールバーの**速度**と**加速度**を 5% **速度 5%** **加速度 5%** に調整します。

4. ロボットを操作する

Mech-Viz左上の[**実行**]をクリックするか、またはMech-Centerで[**実行**]をクリックする

と、ロボットは計画した経路に沿って動作します。



ロボットが動作しているとき、緊急時にはティーチペンダントの非常停止ボタンを押してください。


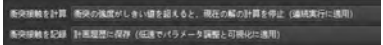
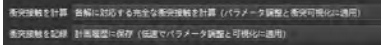
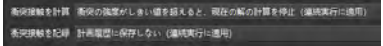
5. プロジェクトの実行を中止するには、以下のいずれかの手順を実行してください。

Mech-Vizのインターフェイスの左上の[**停止**]をクリックします。

ロボット実機を操作しているときにエラーが発生したら、[計画履歴](#) と [ログ](#) を使用してを実行してください。エラーが発生しなければ [計算設定](#) を修正して実行の速度を向上させることが可能です。

5.5.4. 計算設定

計算設定は、ソフトウェアにおける衝突の計算と記録を修正し、プロジェクトの実行速度を調整するために使用されます。

	プロジェクトのデバッグに使用され、計算速度が遅い
	項目をクリックして衝突を可視化できますが、衝突の一部のみが表示される
	完全な衝突接触は計画履歴に表示できますが、特定の衝突位置を可視化することはできない
	安定している生産に使用され、衝突接触を計画履歴に保存せず、高速な計算は実現可能

5.6. ツールの使用

5.6.1. モデルエディタ

本節ではモデルエディタとその使用について説明します。

概要

Mech-Viz で使用するロボットハンド衝突モデルは、凸多面体（凸包）で構成される obj でなければなりません。モデルエディタを使用してロボットハンドモデルの凸多面体を作成し、obj 形式に変換することができます。また、ロボットハンドモデルの座標系を再定義することやモデルを stl 形式としてエクスポートすることもできます。

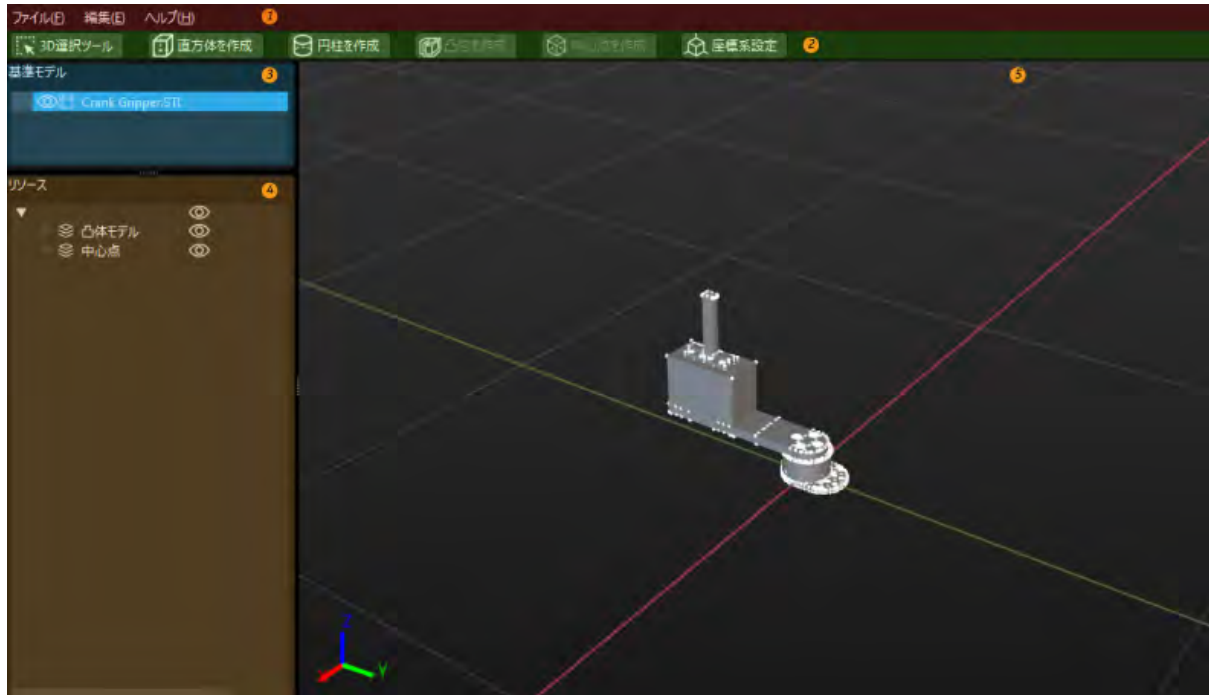
サポートするモデル形式

- インポートできる基準モデル形式：stp、step、stl、obj
- エクスポートできる基準モデル形式：stl

- エクスポートできる凸多面体モデル形式：obj

インターフェイス

モデルエディタのインターフェイスは五つの部分で構成されます：



詳しい説明はクリックしてお読みください。

メニューバー	モデルファイルをインポート・エクスポートする。操作を編集し、ユーザーズマニュアルを開く
ツールバー	よく使うツール
基準モデル	インポートした基準モデルファイルを表示する
リソース	基本的な幾何立体、凸包、中心点
3D 編集エリア	モデルを表示・編集する

凸多面体モデルを作成する

以下の手順を実行して、stl、stp、step、無効な obj モデルを凸多面体（凸包）で構成される obj モデルに変換します。

「座標系の設定」と「凸多面体の作成」を動画で説明します。

▶ <https://docs.mech-mind.net/download/ModelEditor/ModelEditorDemoen.mp4> (video)

基準モデルをインポートする

モデルエディタにインポートできる基準モデルの形式：stl、stp、step、obj。

以下のいずれかの手順を実行して基準モデルをインポートします。

- **ファイル** > **基準モデルをインポート**をクリックして基準モデルを選択します。
- 基準モデルをモデルエディタウィンドウにドラッグします。

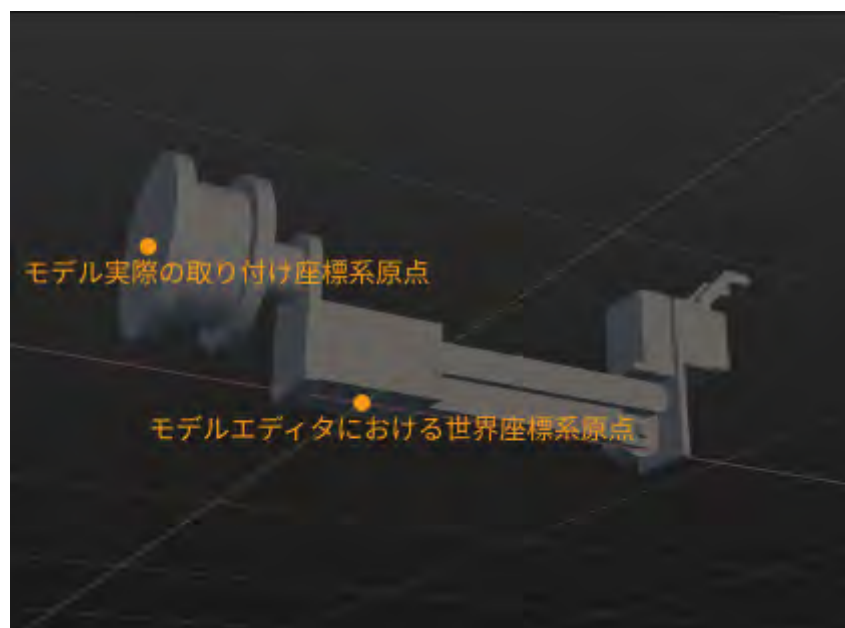
モデルの実際の寸法に基づいて「単位」を選択して[OK]をクリックします。



「モデルを読み込めませんでした。」というメッセージが表示されたら、[STEP/STP ファイル有効性の判断](#)を参考してモデルファイルをチェックしてください。

座標系を設定する

モデルエディタの世界座標系とモデルの実際の取り付け座標系とは一致しない場合、座標系を設定する必要があります。一致する場合はこの部分の説明を読まずに[凸多面体を作成する](#)から実行してください。



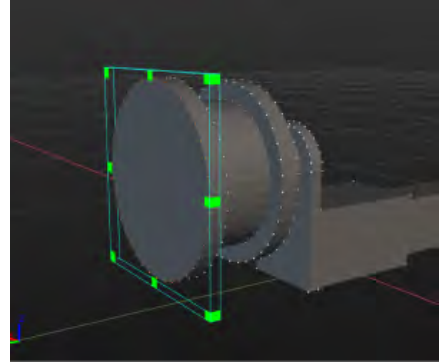
座標系を設定するには、新しい原点を設定し、または軸の方向を調整する必要があります。

新しい座標原点を設定する

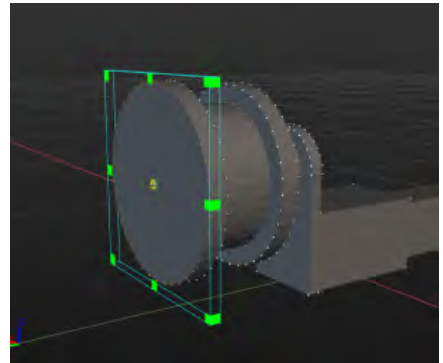
モデルの頂点を囲み、これらの頂点を基準に中心点を指定します。さらに、この中心点またはモデルの頂点を座標原点に設定します。

以下では、中心点の作成を例として説明します。

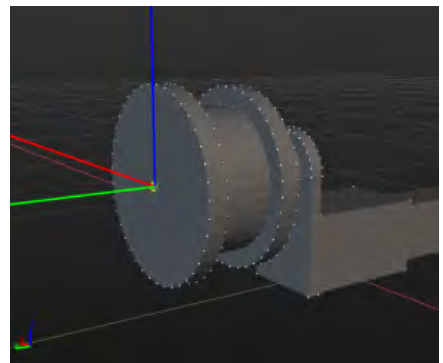
「基準モデル」パネルでモデルを選択する。
[3D 選択ツール]をクリックして 3D 選択直方体を調整し、フランジのエンドの頂点だけを囲む。3D 選択ツールの使用については、[3D 選択直方体の使用ガイド](#)をお読みください。
 モデルの頂点を基準に中心点を指定するので直方体を調整するとき必ず頂点を全部囲んでください。



[中心点を作成]をクリックする。



[座標系設定]をクリックする。
 右側のパラメータパネルで**中心点を選択**をチェックする。
座標系原点の右の**[選択]**をクリックします。
 3D 編集エリアで中心点をクリックし、この中心点を原点とする座標系を作成する。



お使いのモデルは頂点を座標系原点として使用できる場合、以下の手順を実行してください：

1. **[座標系設定]**をクリックします。
2. パラメータパネルで座標系原点の右の**[選択]**をクリックします。
3. 3D 編集エリアで中心点を選択します。

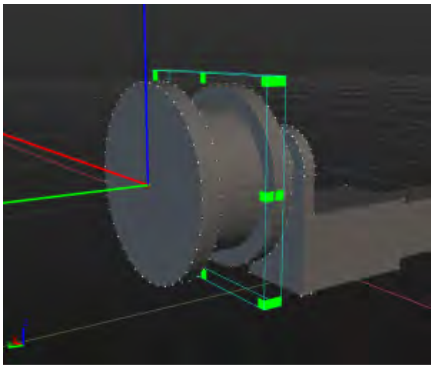
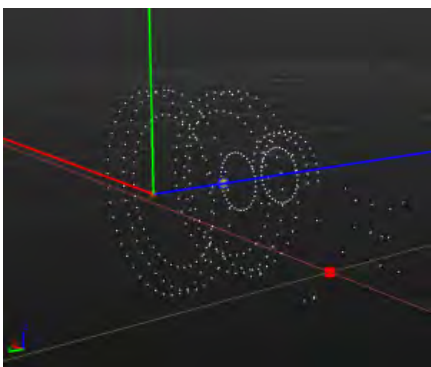
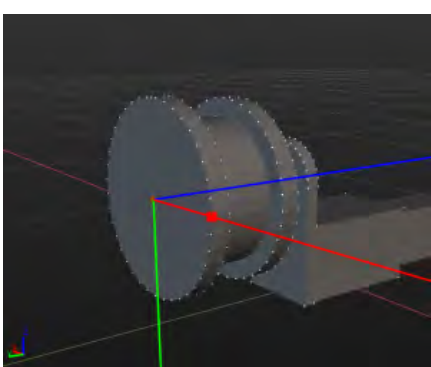



- **中心点を選択**をチェックしなければ、座標原点、X/Y/Z 軸はモデルの頂点しか使用できません。
- **中心点を選択**をチェックすると、座標原点、X/Y/Z 軸はモデルの中心点しか使用できません。
- 続けて軸の方向を調整する場合、**[確認]**をクリックしないでください。

軸の方向を調整する

前の手順を実行したあと座標系の軸の方向が正しい場合は調整する必要はありません。正しく

ない場合は、中心点またはモデル頂点を使用して調整することができます。以下の手順を実行します。

<p>中心点を選択のチェックを外す [3D 選択ツール]をクリックして直方体を調整し、エンドの頂点と並行する頂点を囲む [中心点を作成]をクリックして Z 軸の基準中心点を指定する</p>	
<p>パラメータパネルで中心点を選択をチェックする Z 軸の右の[選択]をクリックする 3D 編集エリアで Z 軸の基準中心点をクリックし、座標系 Z 軸の方向を変更する。</p>	
<p>中心点を選択のチェックを外す X 軸の右の[選択]をクリックする 3D 編集エリアで座標系原点の平面にある頂点をクリックし、X 軸の方向を変更する</p>	
<p>[OK]をクリックする モデル位置姿勢が変わり、モデルエディタの世界座標系原点とモデルの実際の取り付け座標系原点と一致する</p>	

凸多面体を作成する


Mech-Viz では、ロボットハンドの衝突モデルは凸多面体で構成されたものでなければなりません

ん。基準モデルの形に近づけて凸多面体を作成してください。

凸多面体を作成するとき、3D 選択ツールでモデル全体を囲んで作成したりしないでください。ロボットハンドをいくつかの部分に分けて各部分ずつ多面体を作成してください。下図に示すモデルを、11 の部分に分けて多面体を作成することができます。



詳しく説明すると：

1. 基準モデルパネルでモデルを選択してから [3D 選択ツール] をクリックします。
2. 3D 選択直方体を調整し、いずれかの部分の頂点を全部囲みます。[凸包を作成] をクリックします。
3. 11 の部分の凸包を全部作成するまでステップ 2 を繰り返します。
4. 基準モデル名の左の  をクリックして基準モデルを非表示し、凸多面体だけを表示します。



- 凸包または中心点はモデルの頂点に基づいて生成します。各部分の凸包を作成するときに必ずこの部分の頂点を全部囲んでください。
- 3D 選択ツールを使用しても凸包を作成できない部分に対して、[直方体/円柱を作成](#)することもできます。

プロジェクトのファイルを保存する

モデルを再編集するために、以下の手順を実行してモデルを m3d 形式として保存します。

1. **ファイル** > **保存** をクリックします。

2. 保存場所を選択してファイル名を入力します。
3. **[保存]**をクリックします。

編集後のモデルをエクスポートする

以下の手順を実行して編集後のモデルを obj 形式として保存します。

1. **ファイル** > **凸体モデルをエクスポート**をクリックします。
2. 保存場所を選択してファイル名を入力します。
3. **[保存]**をクリックします。

モデルをエクスポートしてからメインインターフェイスの**リソース** > **モデルライブラリ**にこのモデルを追加し、**リソース** > **ロボットハンド**で衝突モデルの設定を行うことができます。

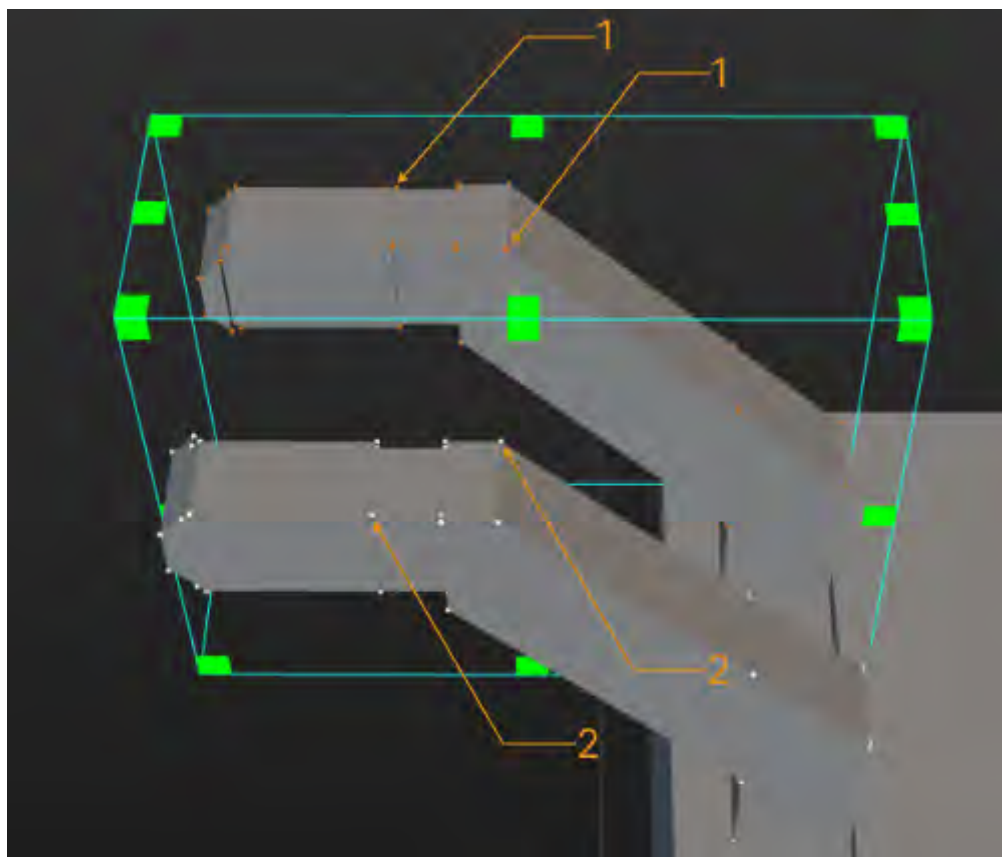
3D 選択直方体の使用ガイド



基準モデル名を選択してから左のをクリックするとモデル頂点でけが表示されます。

[3D 選択ツール]をクリックすると 3D 選択直方体が表示され、調整してモデル頂点を囲みます。以下のいずれかの方法で 3D 選択直方体の寸法と位置を調整できます。

- 3D 編集エリアで調整する：
 - **[Ctrl]**を押したまま 3D 選択直方体の頂点をクリックしてドラッグすると直方体の寸法が変更されます。
 - **[Ctrl]**を押したまま 3D 選択直方体の表面をクリックしてドラッグすると直方体の位置が変更されます。
- パラメータパネルで調整する：
 - **3D 選択ツール**のパラメータ値を調整して寸法を変更します。
 - **中心、回転**パラメータ値を調整して位置を変更します。



1 囲まれているモデル頂点。2 囲まれていないモデル頂点

直方体/円柱を作成する

3D 選択ツールを使用して凸包を作成するほか、直方体または円柱を作成してロボットハンドの一部を囲むこともできます。

直方体の作成を例として説明します（円柱の作成はほぼ同じです）。

1. [**直方体を作成**]をクリックします。
2. 表示されたウィンドウでカスタマイズの物体名を入力し、寸法を大まかに調整してから [**OK**] をクリックします。
3. 作成した直方体は3D 編集エリアの座標系原点にあります。 **Ctrl** を押したままドラッガーをクリックして直方体を適切な位置にドラッグします。
4. 直方体モデルをダブルクリックすると凸多面体の設定ウィンドウが表示されます。それが対象部分を覆うように寸法と位置姿勢を細かく調整します。



3D モデルファイルを stl 形式に変換する

stp、step 形式のモデルファイルを stl 形式に変換します。

1. 以下のいずれかの手順を実行して変換するモデルをインポートします。
 - **ファイル** > **基準モデルをインポート** をクリックして基準モデルを選択します。
 - 基準モデルをモデルエディタウィンドウにドラッグします。
2. menu:ファイル [基準モデルをエクスポート] をクリックします。
3. **モデルをエクスポート** ウィンドウで保存する場所を選択してファイル名を編集したあと [保存] をクリックします。

モデルをエクスポートしてから、メインインターフェイスの **リソース** > **モデルライブラリ** にこのモデルを追加し、**リソース** > **ロボットハンド** または **リソース** > **シーンの物体** で設定を行うことができます。

5.6.1.1. インターフェイス

本節ではモデルエディタのインターフェイスについて説明します。

メニューバー

ファイル

オプション	説明	ショートカット
新規作成	新しいモデルを作る	Ctrl + N
開く	.m3d ファイルを開く	Ctrl + O
保存	編集したものを .m3d ファイルとして保存する	Ctrl + S
名前を付けて保存	編集したものを m3d 形式として指定場所に保存する	Ctrl + Shift + S
基準モデルをインポート	モデルエディタに基準モデルをインポートする	
基準モデルをエクスポート	基準モデルを stl 形式ファイルとしてエクスポートする	
凸体モデルをエクスポート	凸体モデルを obj 形式ファイルとしてエクスポートする	

編集

オプション	説明	ショートカット
取り消し	操作を取り消す	Ctrl + Z
やり直し	操作をやり直す	Ctrl + Y

ヘルプ

オプション	説明
ユーザーズマニュアル	ブラウザでモデルエディタに関するユーザーズマニュアルを開く



ツールバー

オプション	説明	ショートカット
3D 選択ツール	3D 選択ツールを作成する	
直方体を作成	直方体を作成する	
円柱を作成	円柱を作成する	
凸包を作成	3D 選択直方体に囲まれた頂点を基準に凸包を作成する	Shift + C

オプション	説明	ショートカット
中心点を作成	3D 選択直方体に囲まれた頂点を基準に中心点を作成する	
座標系設定	中心点または頂点を基準に座標系を作成する	


基準モデル

インポートした基準モデルファイルを表示します。

- モデルファイル名をクリックすると、3D 編集エリアにモデルの頂点が表示されます。
- モデル名を選択したあと、 をクリックするとモデル本体が非表示され、モデルの頂点だけが表示されます。
- モデル名を選択しないまま、 をクリックするとモデル本体も頂点も非表示されます。

リソース

基本的な幾何立体、凸包、中心点を表示します。

 をクリックすると凸体または中心点が非表示されます。

基本的な幾何立体名を右クリックすると以下のオプションがあります：

オプション	説明	ショートカット
コピー	幾何立体をコピーする	Ctrl + C
貼り付け	幾何立体を貼り付ける	Ctrl + V
削除	幾何立体を削除する	Delete
名前を変更	基本的な幾何立体名を変更する	

凸包名を右クリックすると以下のオプションがあります：

オプション	説明	ショートカット
コピー	凸包をコピーする	Ctrl + C
貼り付け	凸包を貼り付ける	Ctrl + V
削除	凸包を削除する	Delete
名前を変更	凸包名を変更する	

オプション	説明	ショートカット
作成時の 3D 選択ツールを表示	凸包を作成時の 3D 選択ツールを表示する	

中心点名を右クリックすると以下のオプションがあります：

オプション	説明	ショートカット
コピー	中心点をコピーする	Ctrl + C
貼り付け	中心点を貼り付ける	Ctrl + V
削除	中心点を削除する	Delete
名前を変更	中心点名を変更する	
作成時の 3D 選択ツールを表示	中心点を作成時の 3D 選択ツールを表示する	
上部に固定	チェックするとモデル内部の中心点が表示される	

3D 編集エリア

モデルを表示・編集します。また、以下の操作を実行して視点を変えます。

- マウスホイールを回転させ 3D 編集エリアを拡大・縮小します。
- 左ボタンを押したままドラッグして視点を変えます。
- マウスホイールを押したままドラッグして視点を並進します。
- 右クリックしてドロップダウンメニューで視点を変えることができます。

5.6.1.2. STEP/STP ファイル有効性の判断

STEP/STP ファイルが以下の条件を満たさない場合、モデルを読み込む時に「モデルを読み込めませんでした」というメッセージが表示されます。

- STEP/STP ファイルは ASCII ファイルであること。
- STEP/STP ファイルが「ISO 10303-21」規格に準拠し、1 行目に「ISO 10303-21」が付いていること。
- ファイルのディスクリプションとデータを表示する HEADER と DATA もあること。

さらに、STEP/STP ファイルを開くと文字化けした場合、ファイルが暗号化されている可能性があります。ファイルの提供者にご連絡ください。

5.6.2. デパレタイズ用吸盤コンフィギュレータ

本節では、デパレタイズ用吸盤コンフィギュレータについて説明します。

概要

現在、長方形、単一の長方形吸盤、複数ブロックの吸盤、単列吸盤、各ブロックのサイズが一致している吸盤だけに対応しています。

吸盤設定の手順

1. 吸盤の構造を設定する

- 吸盤のブロックの数によって**ブロックの数**パラメータ値を設定します。複数のブロックが一つの DO 信号によって制御される場合、それらが一つのブロックと見なされ、一つの大きなブロックを作成すればいいです。
- ブロック間の間隔によって**間隔**を設定します。
- 実際の吸盤のブロックのサイズによって**ブロックの長さ/ブロックの幅**を設定します。
- スポンジの厚さ**を衝突検出に使用されるスポンジの厚さ、つまり TCP 平面の下のスポンジの厚さに設定します。


2. ロボットハンド設定ウィンドウで設定した TCP のパラメータ値によって**角度**を設定します。

3. 実際に応じて**DO/DI 有効信号**を設定します。

4. DI 検出点を設定する

- [+]をクリックして DI 検出点を追加します。デフォルトでは DI が 0 となり、吸盤の真ん中にあります。
- X と Y の値を変更するか、検出点をドラッグして検出点の位置を調整します。
- DI の値を変更して検出点の番号を変更します。



検出点を削除するには、検出点を選択してから  をクリックします。

- 吸盤の各ブロックのポート番号によって **DO** の値を入力します。
- ファイル** > **保存** をクリックして吸盤の設定を対応するロボットハンド設定に保存します。

吸盤設定ファイルをエクスポートする

ファイル > **エクスポート**をクリックして設定ファイルを指定したパスに保存します。

吸盤設定ファイルをインポートする

ファイル > **インポート**をクリックして吸盤設定ファイルをインポートします。

5.6.3. 配列タイプグリッパコンフィグレータ

本節では配列タイプグリッパコンフィグレータについて説明します。

設定手順

1. 実際の配列タイプグリッパのグリッパの数と間隔によって**合計数**と**間隔**パラメータ値を調整します。
2. 実際に応じて **DO 有効信号**パラメータ値を設定します。
3. ロボットハンド設定ウィンドウで設定した TCP のパラメータ値によって**配列の角度**を設定します。
4. **ファイル** > **保存**をクリックして吸盤の設定を対応するロボットハンド設定に保存します。

ツール配列設定ファイルをエクスポートする

ファイル > **エクスポート**をクリックして設定ファイルを指定したパスに保存します。

ツール配列設定ファイルをインポートする

ファイル > **インポート**をクリックして設定ファイルをインポートします。

5.6.4. パレットパターンエディタ

「カスタマイズのパレットパターン」ステップに使用され、ユーザーのカスタマイズパレットパターンを編集します。「カスタマイズのパレットパターン」ステップの[**パレットパターンエディタ**]をクリックしてウィンドウを開きます。

事前準備

パレットパターンを編集する前に、以下のことを確認してください：

- パレットの長さと幅
- パレットパターンの段数
- パレットパターンの段の高さ
- 各段のレイアウト
- 箱の長さと幅

パレットパターンを編集する

パレットパターンの編集は、ウィンドウの右の**段のレイアウト**と左の**パレットの段**を設定します。



ウィンドウの左上の**ファイル・ファイルから読み込む**をクリックすると編集済みのパレットパターンファイルを読み込みます。

段のレイアウト

パレットパターンの寸法、段のレイアウトの数、箱の数を設定します。

1. **パレットの長さ**と**パレットの幅**を設定します。パレットの長さは対象物を載せるパレットの長さで、幅はそのパレットの幅です。
2. **[位置を追加]**をクリックして箱を表示する黄色い長方形が表示されます。長方形を左クリックしてドラッグすると位置を調整できます。
3. **位置を設定**に箱の寸法や回転角、進入角度を設定します。
4. ステップ2と3を繰り返して段のレイアウト設定を完了します。
5. 複数の段のレイアウトを設定する場合、左側の編集エリアで**[段を追加]**をクリックして段数を増やします。

パレットの段

段数、段の高さ、各段のレイアウトを設定します。

1. **[段を追加]**をクリックして段数を増やします。
2. いずれかの段を選択して**段の高さ**を設定します。
3. **レイアウトを変更**に、選択された段に右の編集エリアで編集したレイアウトを選択します。
4. ステップ2と3を繰り返して各段の設定を完了します。

保存と使用

ファイル・保存 (Ctrl+S) をクリックすると今編集したパレットパターンを保存できます。ソフトウェアの仮想空間にパレットパターンが表示されます。

ファイル・ファイルに保存をクリックすると今設定したパレットパターンを **.json** ファイルとして保存します。「カスタマイズのパレットパターン」ステップの**[パターンを読み込む]**をクリックして保存したファイルを選択するとパレットパターンを読み込みます。

5.7. 付録

5.7.1. ロボットモデルの作成とインポート

ソフトウェアのロボットライブラリもオンラインロボットライブラリも使用するロボットがな

い場合、本節の説明を参考してロボットのモデルを作成してインポートしてください。

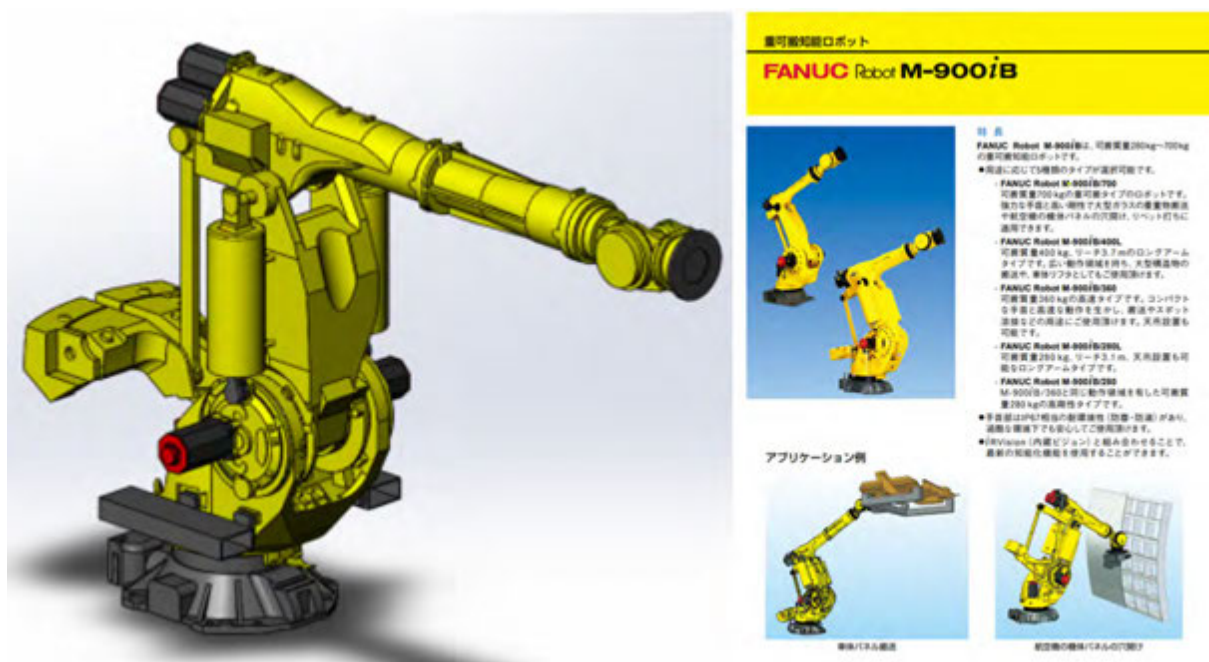
FANUC M-900iB/400L を例とします。

ロボット CAD ファイルと説明書を用意する

ロボットモデルを作成する前に、ロボットの CAD モデルを作成し、ロボットの DH パラメータや各関節リミットなどを明記する説明書ファイルを用意します。

ユーザーは、ロボットメーカーの公式サイトから使用するロボットのファイル入手することができます。

下図は、FANUC M-900iB/400L [CAD モデル](#) と [説明書](#) です。



一部のロボットメーカーのホームページに x_t フォーマットのモデルを入手できません。STEP フォーマットモデルに比べれば、x_t フォーマットを使用すればより高い成功率と速度が図れるので x_t フォーマットを優先使用することを推奨します。

[robot]_algo.json パラメータファイルを編集する

[robot]_algo.json ファイルモデル

[robot]_algo.json モデルファイルは Mech-Viz のインストールパスの [C:/Users/Administrator/AppData/Roaming/Mmind/robot/profile_example.json](#) に保存されます。ユーザーは、これに基づいて新たな _algo.json ファイルを作成することができます。



[robot]_algo.json ファイルの各パラメータについての説明は、[robot_profile.json ファイル属性](#) をご参照ください。

ロボットの構造タイプ (algo_type) を確認する

ロボットには、様々な構造タイプがあります。構造タイプによって座標系と DH パラメータ定義が異なります。

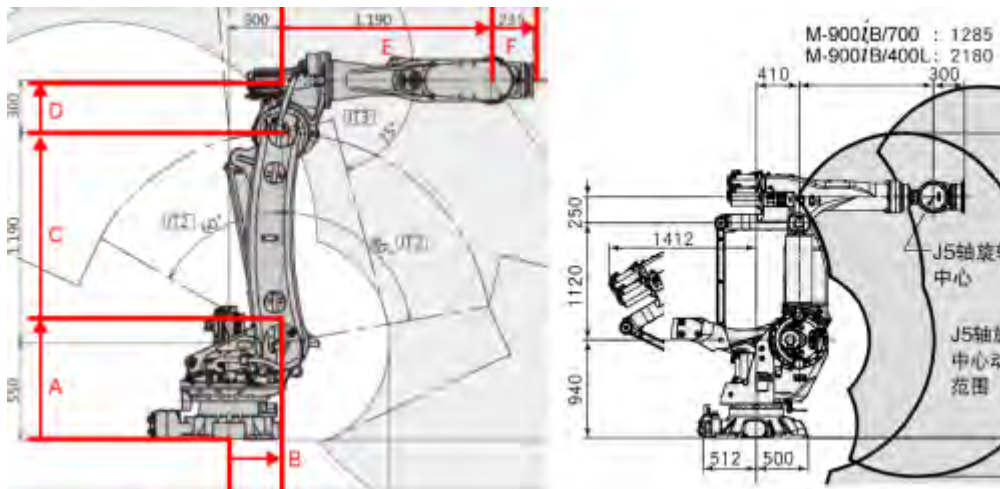
ロボット構造タイプについては [ロボット構造のタイプ](#) をご参照ください。

FANUC M-900iB/400L ロボットは、SphericalWrist_SixAxis タイプで一般的な 6 軸産業用ロボットです。

DH パラメータ (dh、dhPassive) を確認する

[ロボットの構造タイプ](#) に記載されているロボット分類の内容を参照して対応するロボット構造タイプ図を見つけて、ロボット説明書を使用して [robot]_algo.json の dh 値を確認します。

FANUC M-900iB/400L の dh パラメータは、 $a=0.940$ 、 $b=0.410$ 、 $c=1.120$ 、 $d=0.250$ 、 $e=2.180$ 、 $f=0.300$ です。



上図では、左は SphericalWrist_SixAxis ロボット構造タイプ図で、右は FANUC M-900iB/400L ロボット寸法図です。

ロボットの各軸のリミット (minlimits、maxlimits) を確認する

普通、ロボット説明書にはロボットの各軸の運動範囲が記載されますが、FANUC ロボットの場合、FANUC ロボットシミュレーションソフトウェア RoboGuide を起動してその情報を入手します。



その他のパラメータを確認する

一般的なロボットブランドに対し、すでに追加したロボットのデータを参照できます。参照になれるデータがなければ、現場ではエンジニアがロボット実機と合わせて以下のパラメータを確認する必要があります

：mastering_joints、axis_flip、base_z_offset、axis_flip、mastering_joints。

- json パラメータファイルの句読点（コンマなど）は、英文にしなければなりません。
- ロボットモデル作成後に、mastering_joints、axis_flip、base_z_offsetを **ロボットシミュレーションソフトウェア** と照合してください。
- axis_flip は minlimits、maxlimits と連動するので必要に応じて一つの関節の上・下限数値の正負を反転して位置を交換します。
- mastering_joints の最後の J6 はよく無視されます。これは **オイラー角** に影響を与えるのでチェックしてください。

[robot]_profile.json パラメータファイルを編集する

[robot]_profile.json ファイルモデルは Mech-Viz インストールパスにある [resource/robot/profile_example.json](#) に保存されます。ユーザーは、これに基づいて新たな _profile.json ファイルを作成することができます。

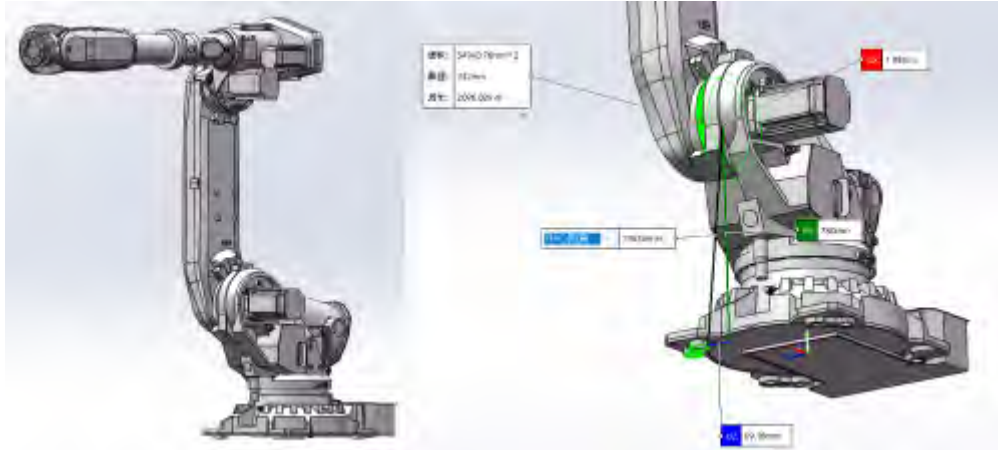


- [robot]_profile.json ファイルの各パラメータについての説明は [robot_profile_json_attribute](#) をご参照ください。
- このファイルは必須ではありません。

Solidworks を使用して完全な STL モデルを作成する

モデルを SolidWorks にインポートする

図に示すように、SolidWorks を使用して事前に作成したロボット CAD モデルを開きます。



上図では、左は 3D ソフトウェアで表示されたロボットモデルで、右は組立参考図です。

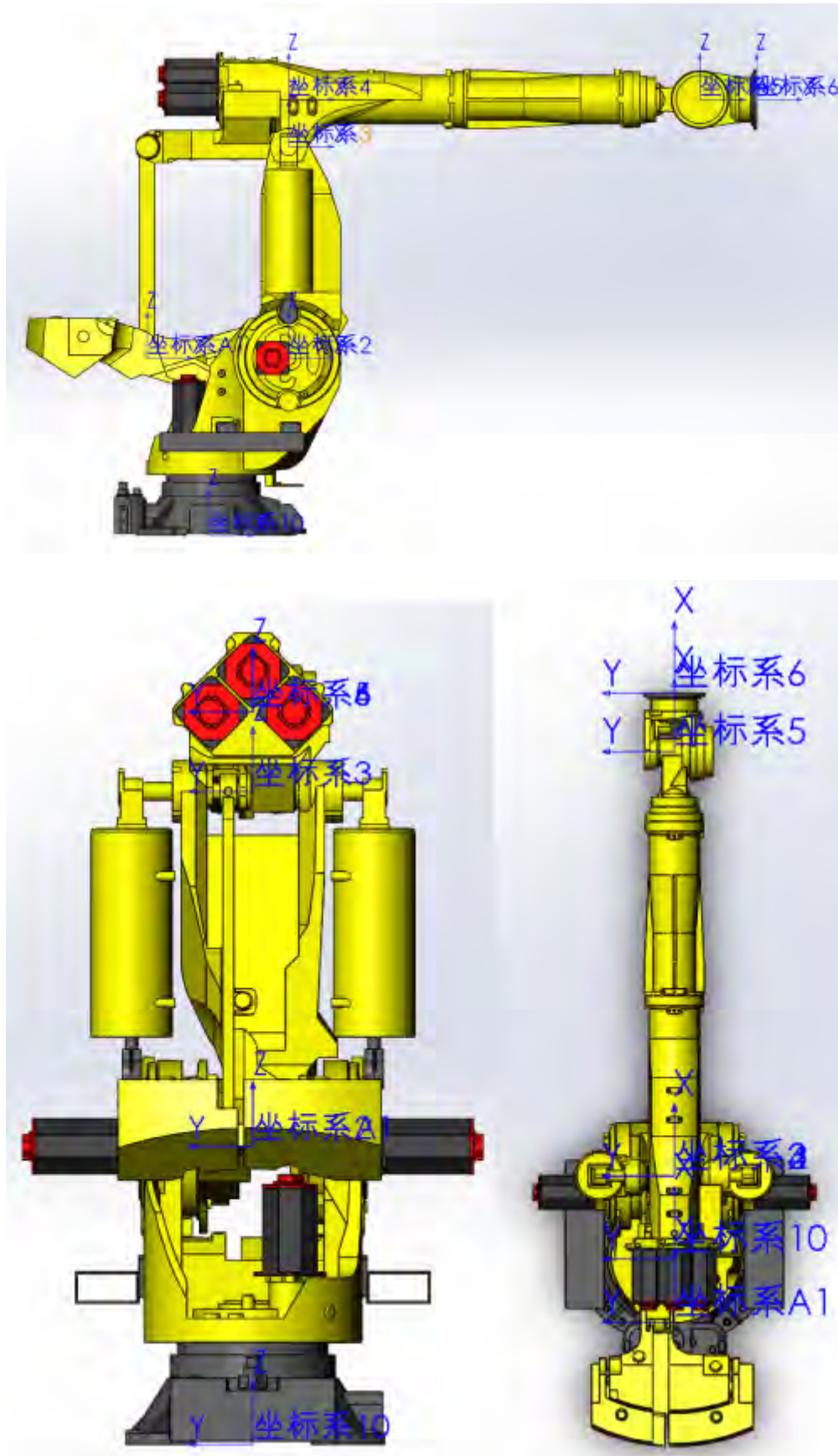


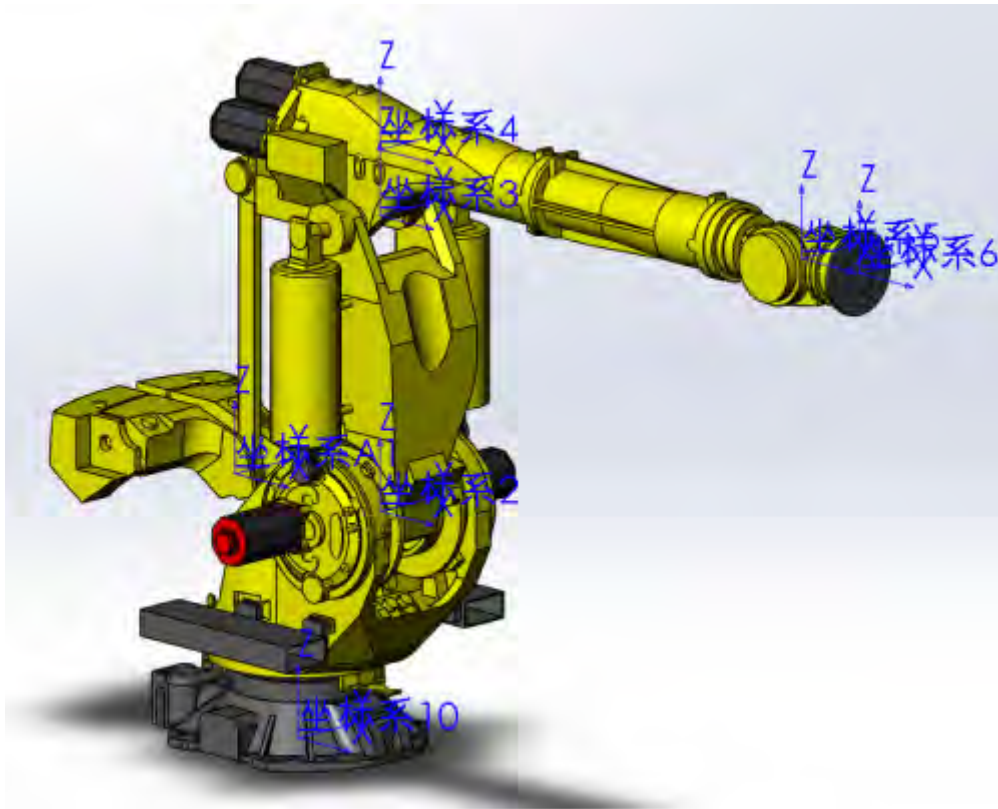
- ロボットモデルをダウンロードするとき、なるべくメーカーのホームページで予め組立済みの完全なモデルをダウンロードしてください。
- 完全なモデルでなく、パーツしか入手できない場合、パーツを組み立て、かつ dh パラメータを参照してチェックしてください。
- モデルの読み込みの速度を上げるために、衝突検出に影響しないモデルのディテールを適当に削除します。

座標系を構築する

[ロボット構造のタイプ](#) を参照して各軸の座標系を構築します。

FANUC M-900iB/400L ロボットは SphericalWrist_SixAxis タイプなので、正しい姿勢は：1 軸は正、2 軸は垂直で上向き、3、4 軸は水平前向き、5 軸は下向きでなく前向きです。姿勢を確認してから全てのパーツを固定にすることを推奨します。

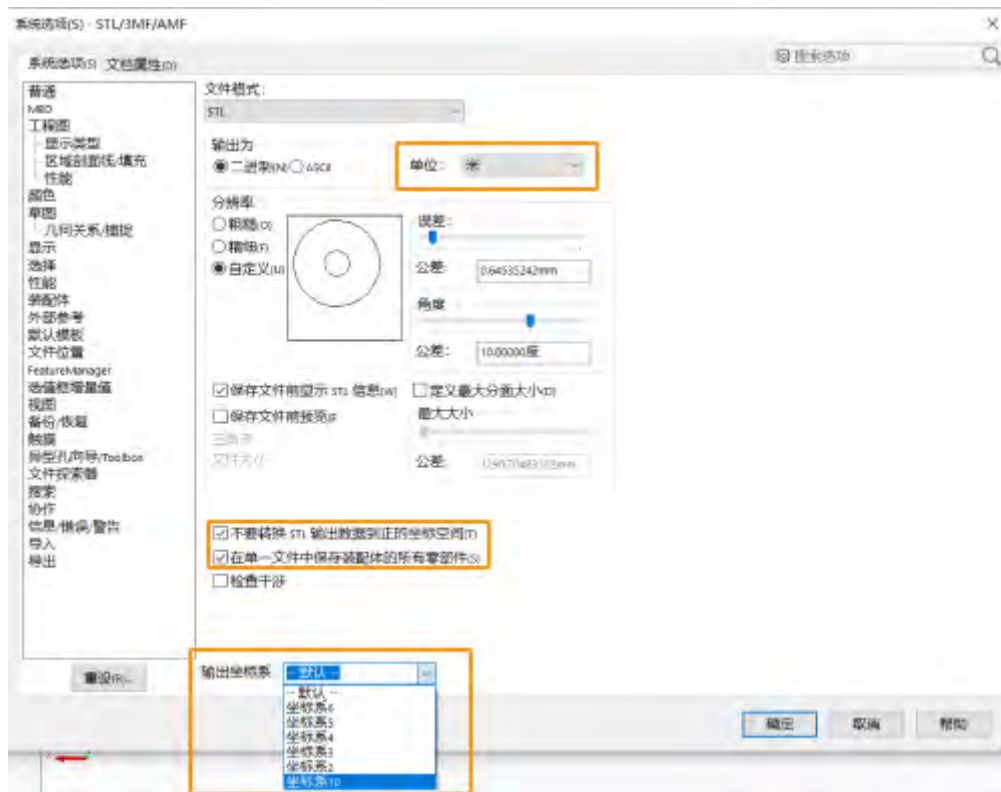




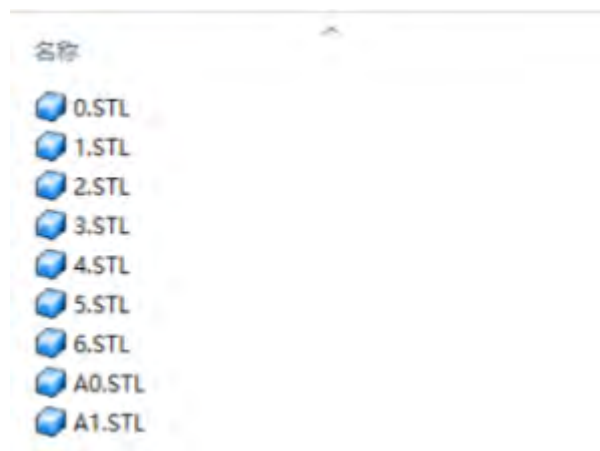
ロボットの各関節の STL モデルをエクスポートする

ロボットモデルをエクスポートする時、一つずつエクスポートするために、組み立て部品の中に他のパーツを非表示します。

モデルは stl フォーマットのファイルとしてエクスポートされ、出力がバイナリでメートルを単位とします。ファイルの命名と各部分に使用する座標系については、[ロボット構造のタイプ](#) をご参照ください。他のオプションについては下図をご参照ください。



下図は、FANUC M-900iB/400L ロボットモデルを保存した後の完全な STL モデルファイルです。



これでロボットモデルの作成が完了します。

モデルファイルを Mech-Viz にインポートする

ロボットライブラリは Mech-Viz インストールパスにある **resource/robot** フォルダに保存されます。同じブランドのロボットモデルファイルは、同じブランド名で名付けたフォルダに保存され、各ロボットのモデルファイルはそのブランドのフォルダの中に、ロボット名で名付けたフォルダに保存されます。

FANUC M900iB/400L ロボットを例に、そのモデルファイルは **resource/robot/fanuc_m900ib_400l** で、ファイルの構造は下図のようです。



- STL：モデルファイル
- fanuc_m900ib_400l_algo.json：ロボット運動学パラメータファイル
- fanuc_m900ib_400l_preview.png：ロボットの画像
- fanuc_m900ib_400l_profile.json：ロボットパラメータファイル

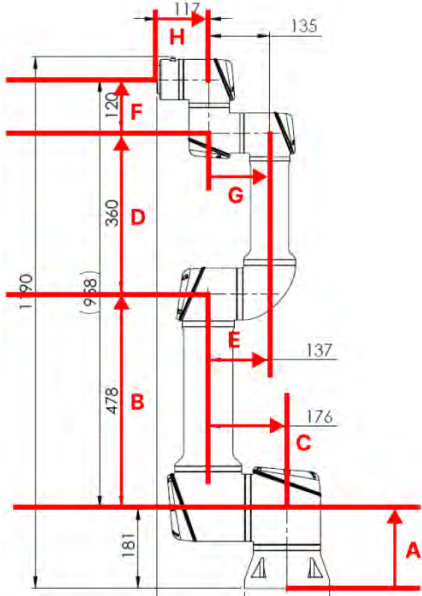
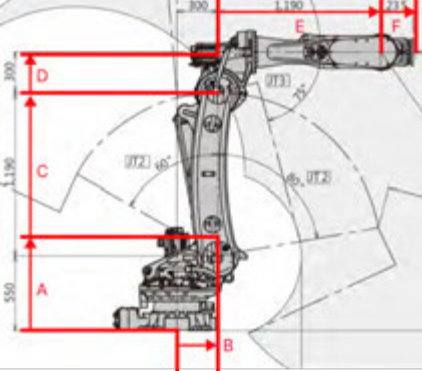

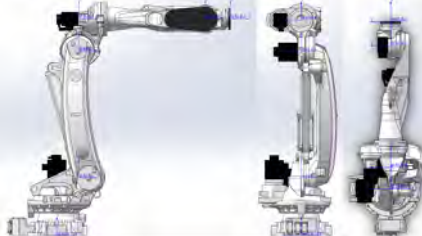


- ロボットファイル名は、いずれも小文字にしてください。
- STL、DAE、OBJ は、ロボットモデルファイルのフォーマットです。それぞれのフォルダに保存されます。
- STL モデルは 3D モデル・衝突モデルとして使用できます。
- DAE モデルは 3D モデルにしか使用できません。
- OBJ モデルは衝突モデルにしか使用できません。
- 完全なモデルは、3D モデルと衝突モデルがすくなくともそれぞれ一つなければなりません。

全てのファイルを対応するパスに保存すればロボットモデルのインポートが完了です。インポートされたロボットを使用するには、Mech-Viz のロボットライブラリで選択することができます。

5.7.1.1. ロボット構造のタイプ

軸数		6 軸	6 軸
構造	構造名	UR_UR5_Like	SphericalWrist_SixAxis
	説明	6 軸協働ロボット	一般的な産業用 6 軸ロボット

軸数	6 軸	6 軸
ロボット 構造のタ イプ		
	座標系 各軸名 	
リンク、座標系、対 応関係	0：座標系10、1：座標系10、2：座標系2、3：座標系3、4：座標系4 ；5：座標系5；6：座標系6	

軸数	6 軸	6 軸
[robot]_algo の例	<pre>{ "algo_type": "UR_UR5_Like", "robot_type": "UR_16E", "dh": [A, B, D, G, F, H], "shoulder_offset": C, "elbow_offset": E, "min_limits": [J1 min, J2 min, J3 min, J4 min, J5 min, J6 min], "max_limits": [J1 max, J2 max, J3 max, J4 max, J5 max, J6 max], "mastering_joints": [J1, J2, J3, J4, J5, J6], # Unnecessary "axis_flip": "J1J2J3J4J5J6" # Unnecessary }</pre>	<pre>{ "algo_type": "SphericalWrist_SixAxis", "robot_type": "KAWASAKI_CX110L", "dh": [A, B, C, D, E, F], "min_limits": [J1 min, J2 min, J3 min, J4 min, J5 min, J6 min], "max_limits": [J1 max, J2 max, J3 max, J4 max, J5 max, J6 max], "mastering_joints": [J1, J2, J3, J4, J5, J6], # Unnecessary "axis_flip": "J1J2J3J4J5J6" # Unnecessary }</pre>

5.7.1.2. [robot]_algo.json ファイル属性

[robot]_algo.json はロボット設定ファイルであり、ロボットのプロパティの情報が記載されており、ロボットの種類や各リンクの DH パラメータ、各関節の運動の上・下限、各関節の初期位置、各関節の回転方向などを決めます。

以下は [robot]_algo.json ファイルのコードの一部です。

```
{
  "algo_type": "UR_UR5_Like",
  "robot_type": "UR_16E",
  "dh": [ A, B, D, G, F, H ],
  "shoulder_offset": C,
  "elbow_offset": E,
  "min_limits": [ J1 min, J2 min, J3 min, J4 min, J5 min, J6 min ],
  "max_limits": [ J1 max, J2 max, J3 max, J4 max, J5 max, J6 max ],
  "mastering_joints": [ J1, J2, J3, J4, J5, J6 ],      # Unnecessary
  "axis_flip": "J1J2J3J4J5J6"                        # Unnecessary
}
```

algo_type

ロボットの構造タイプです。Mech-Viz では、ロボットは 13 類に分類されている。各タイプの詳細情報は [ロボット構造のタイプ](#) にアクセスしてください

robot_type	ロボットと名。ロボットモデルフォルダ名と一致させる必要がある。一致しないとロボットモデルが見つからなくなる
dh	ロボット関節の DH パラメータ
minlimits、maxlimits	<p>minlimits、maxlimits は、それぞれロボット各軸の最大・最小運動範囲を表す</p> <p>FANUC、Nachi 以外のロボットは、説明書からその情報を入力できる</p> <p>このパラメータは、各関節に作用し、上・下限を表すAxis_flip と連動する。必要に応じて一つの関節の上・下限数値の正負を反転して位置を交換する</p>
mastering_joints	ロボットの各軸が 0 度になった時の位置を表す。各パラメータが 0 となると、ロボット各軸の 0 度位置は、モデル作成時の姿勢と一致する。KUKA などのロボットは、J2、J3 が 0 になった姿勢は Mech-Viz ではデフォルト姿勢ではないので、それぞれ J2、J3 mastering_joints パラメータを調整する必要がある
axis_flip	<p>各軸の回転方向を決める。ロボットモデル作成後、Mech-Viz とロボットシミュレーションソフトウェアまたはロボット実機と合わせて各軸の回転方向が一致するかを確認する。一致しなければこのパラメータを調整してください</p> <p>このパラメータは、ロボット関節の運動上・下限 minlimits、maxlimits、動的制限 link3_dynamic_limits、link4_dynamic_limits のパラメータに影響を与える</p>

新しく作成したロボットモデルを使用するとき、ロボット実機で以下のパラメータを確認します。

axis_flip	Mech-Viz とロボット実機を照合して、各軸の回転方向が一致するかを確認する
------------------	--

dh	<p>以下の二つの方法で Mech-Viz の仮想ロボットとロボット実機の位置姿勢データが一致するかを確認する</p> <ul style="list-style-type: none"> ● 関節角度を同期してロボットハンドの位置姿勢を照合する ● ロボットハンドの位置姿勢を同期して関節角度を照合する <p>ロボット実機の DH パラメータがパラメータファイルにある理論値に近ければ近いほど、ロボットの精度は高くなる。普通、1mm 以下の誤差は許容される</p>
mastering_joints	<p>Mech-Viz の仮想ロボットとロボット実機の位置姿勢データが一致するかを確認する</p> <p>とくに、1、4、6 軸に注意深くチェックする</p>
min_limits/max_limit	<p>Mech-Viz の仮想ロボットとロボット実機の位置姿勢データが一致するかを確認する</p> <p>ソフトリミットがロボット実機のハードリミットを超えないようにしてください</p>

5.7.1.3. [robot]_algo.json ファイルのパラメータ説明

[robot]_profile.json は、ソフトウェアにおけるロボットの初期姿勢や可搬重量、稼働範囲、軸数などの情報が記載されます。

以下では [robot]_profile.json ファイルの一部を示します。

```
{
  "home_jps": [0,90,0,0,90,0],
  "max_tcp_vel": 7,
  "max_joint_vel": 500,
  "reach": 2.55,
  "payload": 40,
  "axes": 6
}
```

home_jps	Mech-Viz におけるロボットのデフォルト姿勢を決定する。 度 を単位とする
max_tcp_vel	最大 TCP 速度。 メートル/秒 を単位とし、特異点を除去するために使用される
max_joint_vel	最大関節速度。 度/秒 を単位とし、特異点を除去するために使用される

reach	ロボットの稼働範囲。 メートル を単位とし、ロボットライブラリに表示される
payload	ロボットの可搬重量。 キログラム を単位とし、ロボットライブラリに表示される
axes	ロボットの軸数。ロボットライブラリに表示される

5.7.2. その他

5.7.2.1. 設定-オプション

オプションウィンドウでは以下の設定内容があります。

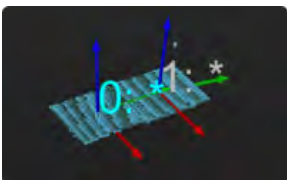
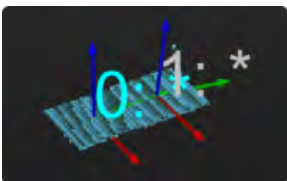
- 基本設定
 - カスタマイズのリソースパス
 - ログ
 - 仮想空間のパラメータ
 - メインウィンドウを閉じる
 - 言語、長さの単位、角度の単位
- 詳細設定
 - ロボット実機の一時停止機能を有効にする
 - 地面のタイプ

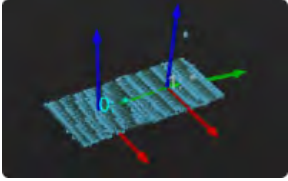
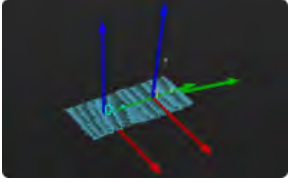
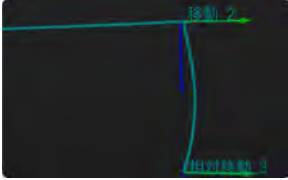
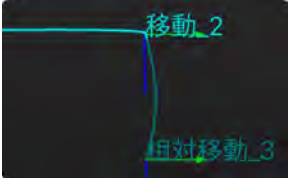
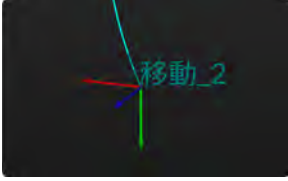



「*」符号が付いているオプションは、ソフトウェアを再起動してから有効になります。

5.7.2.2. ディスプレイ-ディスプレイ設定

ディスプレイ設定ウィンドウでディスプレイを詳細に設定します。

パラメータ	オプション	サンプル図
ビジョン位置姿勢のラベルサイズ	0.05	
	0.07	

パラメータ	オプション	サンプル図
ビジョン位置姿勢のドラッガーのサイズ	0.05	
	0.07	
経路のテキストサイズ	0.02	
	0.04	
経路のドラッガーのサイズ	0.05	
	0.1	

5.7.2.3. ツール-ビジョン処理の記録を設定

保存したビジョン処理の記録を使用してシミュレートまたは実際の生産に発生したトラブルを再現します。

Mech-Viz 実行中にトラブルが発生したとき、保存したビジョン処理の記録を使用することで対応する Mech-Vision プロジェクトを実行せずに Mech-Viz プロジェクトだけを実行してトラブルシューティングと解決が可能です。

ビジョン処理の記録を保存する

1. Mech-Viz メニューバーの **ツール** > **ビジョン処理の記録を設定** をクリックしてビジョン処理の記録を設定するウィンドウを開きます。
2. **位置姿勢情報**だけを保存すること、また**位置姿勢情報**と**点群情報**を同時に保存することができます。
 - 位置姿勢情報だけを保存：**位置姿勢情報** をチェックします。
 - 位置姿勢情報と点群情報を同時に保存：**位置姿勢情報**をチェックしてから**点群情報**をチェックします。
3. 実際に応じて**保存期間**を設定します。
4. **保存パス**を変更できます。**[選択]**をクリックしてパスを指定できます。vision_records フォルダでなければなりません。
5. **[OK]**をクリックします。
6. プロジェクトをシミュレート/実行します。



- ビジョン位置姿勢と点群のデータは vision_records フォルダ*対応する Mech-Vision プロジェクト名フォルダ*日付が付いたフォルダ に保存されます。
- 点群があるシーンでは、撮影するたびに点群ファイルが生成されて保存されます。点群ファイルのサイズが大きいのでテストを実行する場合にのみ **点群情報** をチェックすることを推奨します。プロジェクトが安定に実行するときは外してください。
vision_records フォルダ以外、ビジョン処理の記録を再現するために対応する Mech-Viz プロジェクトのファイルも保存してください。

ビジョン処理の記録を使用する

1. 対応する Mech-Viz プロジェクト、つまり実行中にトラブルが発生してビジョン処理の結果が保存された Mech-Viz プロジェクトを開きます。
2. Mech-Vision はまだ起動されていない、または Mech-Vision プロジェクトは Mech-Center に登録されていないことを確認します。Mech-Center に登録された場合、ソフトウェアは保存された視覚処理の記録を使用せず、優先的に Mech-Vision プロジェクトを呼び出します。
3. Mech-Viz プロジェクトのビジョン処理の記録ファイルをソフトウェアのルートディレクトリの下の vision_records フォルダに保存します (**ファイル** > **実行可能ファイルの場所を開く** をクリックしてソフトウェアのルートディレクトリを開くことができます)。
4. **ツール** の **保存されたビジョン処理の記録を使用 (ビジョンサービスが未登録の場合のみ)** をチェックします。
5. **[シミュレート]** をクリックすると、ソフトウェアは vision_records フォルダに保存されたビジョン処理の記録があるかをチェックし、**ビジョン処理の記録を設定** ウィンドウが表示されます。



6. 再使用するビジョン処理の記録を選択してから [OK] をクリックします。

自動的に次の記録を使用チェックを入れないと、選択した記録を使用します。チェックを入れると、選択した記録を再使用してから自動的に次の記録を使用します。最後の記録を使用したら停止します。



再使用した後、必ず **保存されたビジョン処理の記録を使用（ビジョンサービスが未登録の場合のみ）** のチェックを外してください。

6. Mech-Center

Mech-Centerのユーザーズマニュアルへようこそ！

Mech-Centerは、Mech-Mindが独自開発したソフトウェアシステムの中核およびコントロールセンターであり、関連するソフトウェアのグローバル設定、プロジェクト全体のバックアップと復元を行うことができます。また、Mech-Vision、Mech-Viz、Mech-Eye Viewer、ロボット、標準インターフェイス、およびアダプターの状態を直感的に確認できます。外部インターフェイスMech-Interfaceの起動および管理センターとしても機能します。

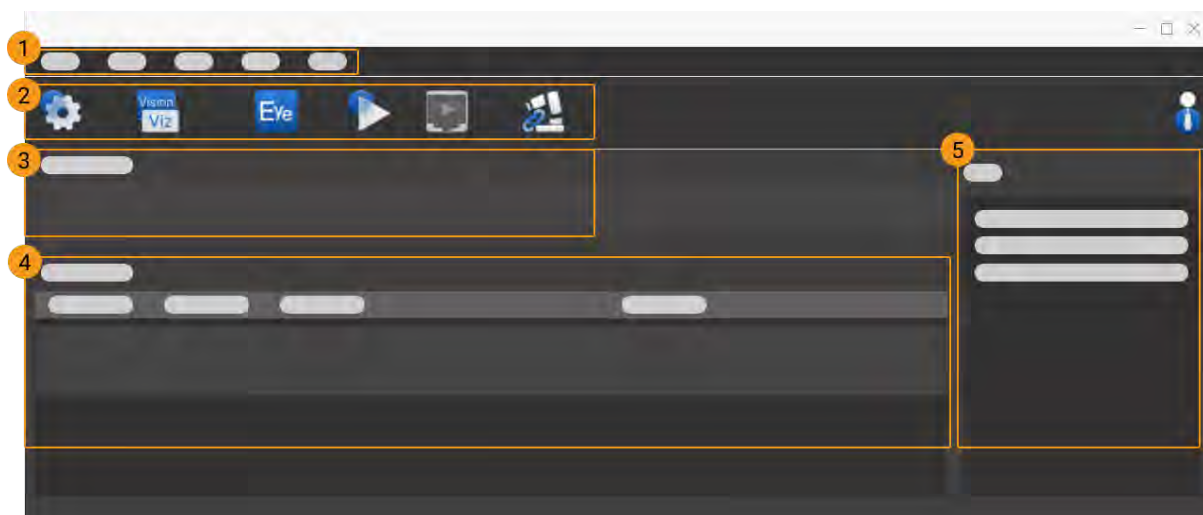
本マニュアルでは、以下の内容について説明します。

番号	内容	詳細
1	Mech-Centerの概要	画面紹介および機能紹介
2	Mech-Centerを使ってみる	基本的な使用流れ
3	Mech-Interface	外部インターフェースサービス
4	標準インターフェイス	標準インターフェースの説明
5	Adapter（アダプター）	Adapter使用説明

6.1. Mech-Centerの概要

Mech-Centerは、Mech-Mindが独自開発したソフトウェアシステムの中核およびコントロールセンターであり、関連するソフトウェアのグローバル設定、プロジェクト全体のバックアップと復元を行うことができます。また、Mech-Vision、Mech-Viz、Mech-Eye Viewer、ロボット、標準インターフェイス、およびアダプターの状態を直感的に確認できます。外部インターフェイスMech-Interfaceの起動および管理センターとしても機能します。

Mech-Centerのメイン画面は、主に5つの部分で構成されています。



1. **メニューバー**：プロジェクトへの操作、ユーザーおよびビューの管理、Adapter生成および情報確認などの機能が備わっています。
2. **ツールバー**：設定、ソフトウェアおよびインターフェースサービスの起動、ロボット接続、実行などのボタンがあります。
3. **サービスステータスバー**：登録されたソフトウェア、カメラおよびロボットなどが表示されます。
4. **プロジェクトのステータスバー**：Mech-Vision/Mech-Viz プロジェクトのステータス、実行時間および詳細な情報が表示されます。
5. **ログバー**：現在のプロジェクトおよびサービスのログ情報がリアルタイムに表示されます。

6.1.1. メニューバー

メニューバーには、ファイル、ツール、ユーザー、ビュー、ヘルプなど、操作に関連する基本機能を提供します。

ファイル

オプション	説明	ショートカット
プロジェクトをバックアップ	Mech-Mindソフトウェアシステムのプロジェクトバックアップを生成するために使用されます。自動的に読み込まれたMech-Viz/Mech-Visionプロジェクト、Mech-Centerのアダプタープロジェクト、およびMech-Centerの設定が保存されます。	Ctrl+B
プロジェクトを復元	選択したバックアップからプロジェクトと設定を復元するために使用されます。自動的に読み込まれたプロジェクトは復元後に置き換えられるので、プロジェクトをバックアップしておいてください。	Ctrl+R

オプション	説明	ショートカット
インポート	その他のパスからプロジェクトをインポートするときに使用されます。	Ctrl+I
終了	Mech-Mindのソフトウェアシステムを終了します。	Ctrl+Q



操作員モードでは、「バックアップ」と「終了」の2つのオプションのみがあります。

ツール

オプション	説明
デバッグデータをパッケージ化	Mech-Mindソフトウェアシステムのプロジェクトのデバッグデータをパッケージ化するために使用されます。期間、パッケージパスおよび保存オプションなどが選択可能です。
Adapterジェネレーター	このオプションは、カスタマイズのAdapterプログラムを生成するために使用されます。
ログビューア	Mech-Viz/Mech-Visionのログメッセージを表示します。手動でログファイルを選択してから、ログファイルを読み込む必要があります。このオプションは、ログレベルに応じて対応するログを表示することができます。また、キーワード（正規表現、大文字と小文字を区別、ワイルドカードが選択可能）を入力してログをフィルタリング・検索することもできます。さらに、タイムラインをスライドして、指定した時間以降のログを選択することもできます。
サービスステータスを表示	ソフトウェア、ロボットおよび各インターフェイスなどのサービスの接続ステータスを表示するために使用されます。



操作員モードで「Adapterジェネレーター」のオプションはありません。

ユーザー

「パスワードを変更」のオプションがあります。管理者モードでのみ、パスワードを変更できます。操作員モードではこの機能を使用できません。



Mech-CenterをIPCに初めてインストールする場合、管理者のデフォルトパスワードは「123456」です。また、パスワードを変更した後は、適切に保管してください。

ビュー

「ログ」にチェックを入れるかどうかを選択できます。チェックを入れれば、ログバーがメインインターフェイスで表示されます。

ヘルプ

オプション	説明	ショートカット
ユーザーマニュアル	インターフェイス使用のドキュメントディレクトリをすばやく開くために使用されます。	F1
更新説明	新たにリリースされたバージョンの新機能とエラー修正について説明します。	なし
バージョン情報	ソフトウェアのバージョンと著作権情報を表示します。	なし

6.1.2. ツールバー

ツールバーには、「設定」、「起動」、「Mech-Eye Viewerを起動」、「実行」、「インターフェイスサービスを起動」、「ロボットを制御」、「管理者」という7つのボタンがあります。

設定

これは、Mech-Centerの基本設定を実装し、各ソフトウェアのパスを設定し、自動的に読み込まれたプロジェクトパスを表示し、外部サービスを選択するために使用されます。

外観と動作

ユーザーが自分の好みに合わせて設定することができます。例えば、ソフトウェアのテーマ、ソフトウェアの表示言語を切り替えることができます（現在は、中国語、英語、日本語、韓国語の4つの言語に対応します）。また、ログ保存の関連設定などを変更できます。

- 「実行中にコンソールを非表示」にチェックを入れれば、Mech-Centerを起動したら、コンソールが非表示になります。
- 「PC起動時にMech-Centerを自動的に実行」にチェックを入れれば、コンピュータを起動すると、Mech-Centerは自動的に起動します。
- 「Mech-Vision/Mech-Viz/Mech-Interfaceを自動的に起動」（もしあれば）にチェックを入れれば、Mech-Centerを起動すると、Mech-Vision/Mech-Viz/Mech-Interfaceも自動的に起動します。
- 「Mech-Vision/Mech-Vizを起動するときにシステムトレイに最小化」にチェックを入れれば、起動したMech-Vision/Mech-Vizがデスクトップステータスバーのトレイに非表示になります。



変更を保存してMech-Centerを再起動してから、設定が有効になります。

Mech-Viz

これはMech-Vizのexeパスを設定し、自動的に読み込まれたプロジェクトのパスを表示するために使用されます。

- クリックして設定画面に入り、「Mech-Vizを使用」のチェックがデフォルトで入れられます。
- インストールされているMech-Vizのexeパスを設定します。[...]をクリックし、Mech-Vizインストールディレクトリの下.exeファイルを選択したら、パス選択は完了します。
- Mech-Vizで「自動的に読み込む」にチェックを入れれば、プロジェクトパスは自動的に読み込まれるので、手動で入力する必要はありません。
- 「実行をシミュレート」にチェックを入れれば、Mech-Vizはロボット実機をシミュレートするだけで、それを移動させることはません。

Mech-Vision、Mech-Eye Viewer

Mech-VisionとMech-Eye Viewerの設定は、Mech-Vizとほぼ同じです。

Robot Server

Mech-Vizによりロボットをティーチングまたは一時ティーチングする場合に使用されます。

- Robot Serverをクリックして設定画面に入ります。Mech-Vizによりロボットをティーチングまたは一時ティーチングする場合に、「Robot Serverを使用」にチェックを入れる必要があります。
- Robot Serverのパスは、デフォルトではMech-CenterインストールされているRobot Serverで、プロジェクトに特別なニーズがある場合は、現在のパスでmain.pyを置き換えることができます。
- Mech-Vizプロジェクトが正常に読み込まれた後、Mech-Vizプロジェクトに使用されるロボット型番は自動的に表示されるので、入力する必要はありません。
- 初回起動時にロボットに接続するためにロボット実機のIPアドレスを入力する必要があります。

Mech-Interface

[Mech-Interface](#)はサードパーティとの通信に使用され、統一された外部インターフェイスです。

起動

Mech-VisionおよびMech-Vizをすばやく起動するために使用されます。正常に起動すると、ソフトウェアアイコンがサービスステータスバーに表示されます。



- プロジェクトの実行時に、Mech-Centerはバージョン互換性チェックを実行します。バージョンV1.4.0以降のMech-Viz、Mech-Vision、Mech-Centerを併用することを推奨しています。
- Mech-CenterがMech-VisionおよびMech-Vizソフトウェアバージョンが1.4.0よりも低いことを検出すると、バージョンは1.4.0より高くなければならないことを表すプロンプトが表示されます。[実行]をクリックすると、エラーメッセージボックスがポップアップ表示されます。

Mech-Eye Viewerを起動

Mech-Eye Viewerをすばやく起動するために使用されます。正常に起動すると、ソフトウェアアイコンがサービスステータスバーに表示されます。

実行

Mech-VizおよびMech-Visionで自動的に読み込まれたプロジェクトを実行します。

インターフェースサービスを起動

Mech-Interfaceを起動するために使用されます。

ロボットを制御

ロボットを制御するために使用されます。ロボット実機が正常に接続すると、ロボットのアイコンおよびその型番はサービスステータスバーに表示されます。

管理者

Mech-Centerには、「管理者」と「操作員」の2つのモードがあり、デフォルトでは「管理者」になります。「操作員」モードに設定すれば、ユーザーがプロジェクトの編集や設定変更を行うことができません。

- 切り替えたい場合、アイコンをクリックし、ダイアログボックスでユーザータイプを選択し、[サインイン]をクリックします。
- 「管理者」に切り替える場合、パスワードが必要です。パスワードを変更する場合はメニューバーで **ユーザー > パスワードを変更** を選択し、パスワードを変更します。

6.1.3. サービスステータスバー

Mech-Center実行中、サービスステータスバーに起動されたソフトウェア、カメラ、ロボット、およびインターフェースサービスを表示できます。



また、サービスステータスバーでソフトウェアアイコンをクリックして対応する画面に入ることができます。

6.1.4. プロジェクトのステータスバー

Mech-Viz/Mech-Visionプロジェクト名、ステータス、実行時間および詳細情報が表示されます。通常は右側にあるログバーと組み合わせて、プロジェクトの実行状況を確認します。

オプション	説明
プロジェクト名	Mech-Viz/Mech-Visionプロジェクト名を表示します。

オプション	説明
ステータス	プロジェクトの現在のステータス（アイドルまたは実行中）を表示します。
実行時間	プロジェクトが実行から終了するまでかかる時間を表示します。
詳細情報	プロジェクトの実行時間および対応するステータスを記録します。また、プロジェクトが正しく実行されない場合は、エラーメッセージがここに記録されます。

6.1.5. ログバー

実行中のプロジェクトとサービスのログ情報がリアルタイムに表示されます。

オプション	説明
D、i、W、E	4種類のログレベルです。その中、Dはデバッグ情報、Iは一般情報、Wはワーニング、Eはエラーを表します。レベルをクリックしてログメッセージをフィルタリングできます。また、レベルを同時に複数選択に対応します。
クリア	表示されるログをすべてクリアします。
フォルダを開く	Mech-Centerインストールパスの logs フォルダを開きます。



エラーメッセージのエラーコードをクリックすると、オンラインドキュメントにジャンプして詳細情報を確認できます。

6.2. Mech-Centerを使ってみる

ソフトウェアを設定

Mech-Centerを起動し、**ツールバー** での **[設定]** をクリックします。表示される画面で、必要に応じて設定を行い、設定後に **[保存]** をクリックします。詳細については、**設定** をご参照ください。



- 実際の状況に応じて設定を行ってください。
- Mech-Vision、Mech-VizおよびMech-Eye Viewerがインストールされた後、それらのパスが「設定」に自動的に読み込まれるので、手動で入力する必要はありません。

Mech-VisionとMech-Vizを起動

ツールバー の **[Mech-Vision/Mech-Vizを起動]** をクリックすれば、Mech-VisionおよびMech-Vizが起動されます。また、それらのアイコンはサービスステータスバーに表示されます。



Mech-VisionまたはMech-Vizのプロジェクトが自動読み込みに設定されている場合、Mech-

Centerは対応するプロジェクトを直接開きます。

Mech-Vizプロジェクトを開く

1. Mech-Centerを起動し、Mech-VizアイコンをクリックしてMech-Vizソフトウェアに入ります。
2. プロジェクトを新規作成するか、既存のファイルを開きます。
3. Mech-Vizのリソースパネルでプロジェクト名を右クリックし、**自動的に読み込む**にチェックを入れます。
4. 現在のプロジェクトの実行ステータスがプロジェクトステータスバーに確認できます。



自動的に読み込むにチェックを入れた後、Mech-VizプロジェクトパスがMech-Centerの**設定** > **Mech-Viz** > **プロジェクトパス**に自動的に記入されます。

Mech-Visionプロジェクトを開く

1. Mech-Centerを起動し、Mech-VisionアイコンをクリックしてMech-Visionソフトウェアに入ります。
2. ソリューションまたはプロジェクトを新規作成します。
3. ソリューションまたはプロジェクトを自動的に読み込みます。
4. 現在のプロジェクトの実行ステータスがプロジェクトステータスバーに確認できます。

カメラ設定を変更

実行中にカメラの設定を表示または調整したい場合は、**ツールバー**での**[Mech-Eye Viewer]**をクリックします。Mech-Eye Viewerでカメラの関連設定を行います。Mech-Eye Viewerが起動されると、そのアイコンはサービスステータスバーに表示されます。

ロボットを接続

- Vizティーチング通信を選択した場合は、選択したロボットに対応するVizティーチング通信マニュアルを参照し、**ツールバー**の**[ロボットを制御]**をクリックします。Mech-Centerがロボットに正常に接続すると、ロボットのアイコンはサービスステータスバーに表示されます。
- 標準インターフェースまたはAdapter通信を選択した場合、標準インターフェースまたはAdapter通信マニュアルを参照し、**ツールバー**の**[インターフェースサービスを起動]**をクリックします。インターフェースサービスが起動されると、対応するアイコンはサービスステータスバーに表示されます。

プロジェクトを実行



ロボット実機を動作させる場合、**設定** > **Mech-Viz**で**実行をシミュレート**のチェックを外してください。

ツールバーでの[実行]をクリックして確認ポップアップウィンドウが表示されます。その後、[Yes]をクリックして読み込まれたプロジェクトが実行されます。また、プロジェクトの実行情報がプロジェクトステータスバーに表示されます。



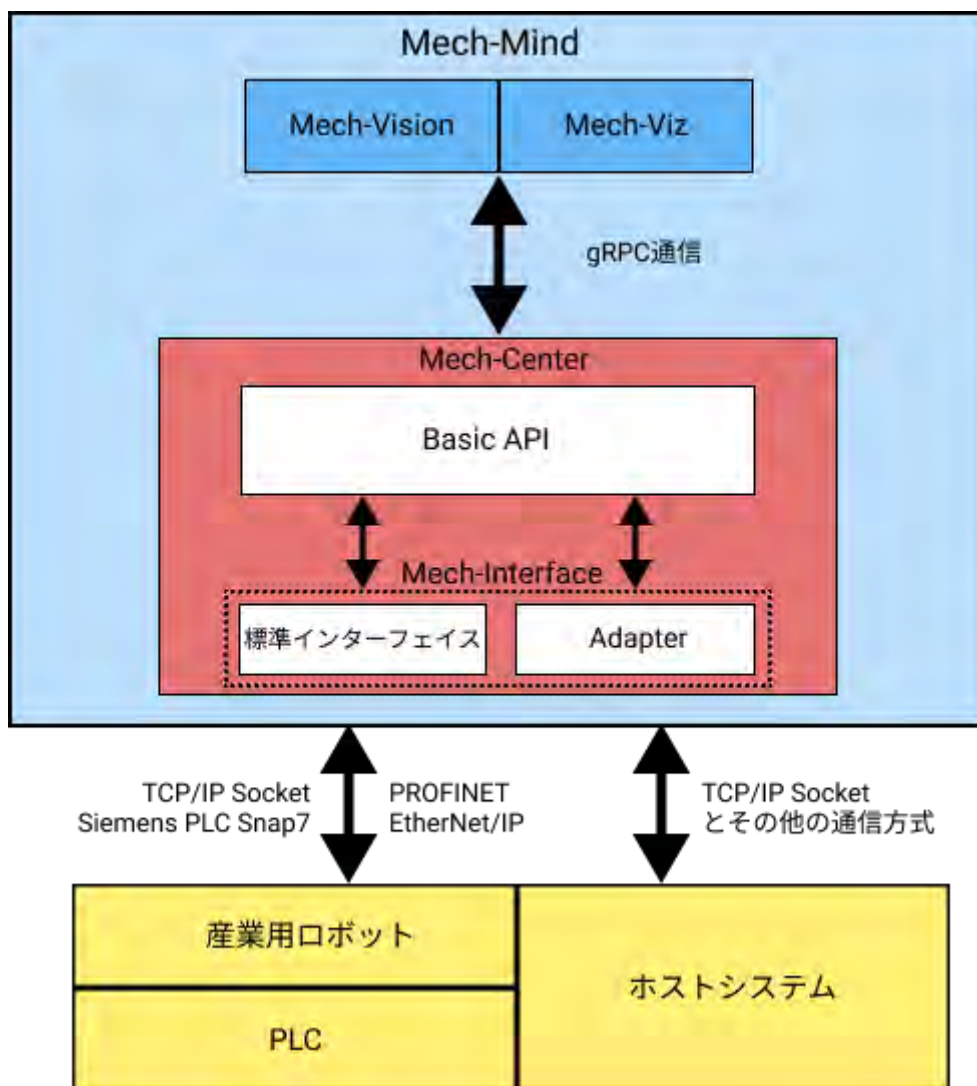
- Mech-Vizプロジェクトを開くときに **ソフトウェア互換性の説明** を処理しない場合は、Mech-Centerで[実行]をクリックすると、Mech-Centerのログに「Mech-Vizはプロジェクトを読み込んでいます。Mech-Vizプロジェクトを起動できません：XXX」というメッセージが表示されます。この時点で、ソフトウェア互換性プロンプトに対処する必要があります。つまり、ソフトウェア互換性プロンプトで[Yes]をクリックし、Mech-Center実行ボタンを復元して再度実行する必要があります。
- ロボットを操作する際は、人の安全を確保してください。緊急時には、ティーチペンドントの緊急停止ボタンを押してください！

6.3. Mech-Interface

Mech-InterfaceはMech-Mindソフトウェアシステムの外部インタフェースサービスで、外部との通信の架け橋としての役割を果たします。外部からの情報を受け取り、システム内部の情報を送信します。このインターフェースには、**標準的なインターフェース**や**Adapter**が含まれています。本節では、Mech-Interfaceの2種類の通信メカニズム、違い、および適用シーンについて説明します。

通信メカニズム

2種類の通信メカニズムは下図に示します。



- Adapter（アダプター）は外部通信デバイス（産業用ロボット、上位システム、PLCなど）と「Mech-Vision」および「Mech-Viz」ソフトウェアの間をつなぐPythonの適合プログラムです。内部ではMech-VisionおよびMech-Vizとの通信を行い、外部では外部デバイスとの通信を行います。
- 標準インターフェースはMech-Mindが提供する完全なセットのAdapterプログラムであり、さまざまな通信プロトコルをサポートし、強力な制御命令セットと異常警告システムを備えており、ほとんどのユーザーの要求を満たすことができます。

標準インターフェースとAdapterの違い

Adapterと標準インターフェースは、外部デバイスとMech-VisionおよびMech-Vizソフトウェアをつなぐための適合プログラムです。標準インターフェースはMech-Mindによって提供される固定のアダプタープログラムのセットであり、カスタマイズ開発に対応していません。

内部でBasic APIを呼び出してMech-VisionおよびMech-Vizと通信し、外部で特定の通信プロトコルを介して外部デバイスと通信します。

標準インターフェースとAdapterの主な違いは次の表に示します。

違い	標準インターフェース	Adapter（アダプター）
内部通信	両方とも、Basic APIの呼び出しによってMech-VisionおよびMech-Vizと通信します。	
外部通信プロトコル	標準インターフェースでは、次の通信プロトコルを使用した外部デバイスとの通信のみをサポートします： <ul style="list-style-type: none"> - TCP/IP Socket - UDP - Siemens PLC Snap7 - PROFINET - EtherNet/IP - Modbus TCP - 三菱 MC 	Pythonが実装できる任意の通信プロトコルを介して外部デバイスと通信できます。
機能	ビジョン結果の出力のみに対応します。	ビジョン関連の機能だけでなく、Pythonでサポートされている機能（例:GUI、データベース、注文システムなど）も提供することができます。
デプロイメントの柔軟性	簡単に使いやすいであるため、迅速にデプロイできます。	ユーザーによってプログラムが作成される必要があり、時間と人的コストがかかります。
拡張可能性	機能拡張はサポートされていません。	より多くの通信プロトコルと機能をサポートするように拡張できます。

適用シーン

実際の適用シーンでは、一般的には外部通信デバイス、使用する通信プロトコル、およびプロジェクトの要件に基づいて、どの種類のMech-Interfaceを使用するかを決定します。

標準インターフェースとAdapterがサポートする一般的な通信デバイスと通信プロトコルを次の表に示します。

通信先	通信プロトコル	Mech-Interfaceの種類	説明
ロボット	TCP/IP Socket	標準インターフェース	Mech-InterfaceはTCP/IP Socketのサーバーとして機能します。
	UDP		Mech-InterfaceはUDPのサーバーとして機能します。
	PROFINET		Mech-InterfaceはPROFINETのスレーブとして機能します。
	EtherNet/IP		Mech-InterfaceはEtherNet/IPのスレーブとして機能します。
	Modbus TCP		Mech-InterfaceはModbus TCPのスレーブとして機能します。
上位システム	HTTP	Adapter（アダプター）	統合プロジェクトに適しており、Vizティーチング通信を採用しています。
	WebSocket		
	TCP/IP Socket	標準インターフェース	Mech-InterfaceはTCP/IP Socketのサーバーとして機能します。
PLC	TCP/IP Socket	標準インターフェース	Mech-InterfaceはTCP/IP Socketのサーバーとして機能します。
	Siemens PLC Snap7		Mech-InterfaceはSiemens PLC Snap7のクライアントとして機能します。
	PROFINET		Mech-InterfaceはPROFINETのスレーブとして機能します。
	EtherNet/IP		Mech-InterfaceはEtherNet/IPのスレーブとして機能します。
	Modbus TCP		Mech-InterfaceはModbus TCPのスレーブとして機能します。
	三菱MC		Mech-InterfaceはMCのクライアントとして機能します。



- プロジェクトの要件に標準インターフェースが対応できる場合は、標準インターフェースを使用することを推奨します。しかし、標準インターフェースがプロジェクトの要件を満たせない場合（例えば、外部デバイスが標準インターフェースでサポートされていない通信プロトコルを使用している、プロジェクトが標準インターフェースでサポートされていない機能を必要としている場合など）、その場合にはAdapterを使用することが適切です。
- 標準インターフェースがサポートする機能一覧については、[標準インターフェース開](#)

[発者向けマニュアル](#)をご参照ください。

- Adapterがサポートする機能一覧については、[Adapter機能](#)をご参照ください。

標準インターフェースとAdapterの詳細については、以下の内容をお読みください。

- [標準インターフェース](#)
- [Adapter \(アダプター\)](#)

6.4. 標準インターフェース

位置姿勢のみを必要とし、Mech-Mindソフトウェアシステムでロボットの動作を制御する必要がない場合、標準インターフェースを使用してデータを転送することができます。標準インターフェースは、位置姿勢およびタスクデータの送信のような、最も基本的なインターフェース機能のみを実現します。より多くのインターフェース機能が必要な場合、Adapterを使用してカスタマイズすることができます。標準インターフェイスがMech-Centerに統合されるので、生成する必要はありません。使用するには、ソフトウェアで標準インターフェースを起動する必要があります。

標準インターフェースを起動

設定 ▶ **Mech-Interface** で **Mech-Interfaceを使用** にチェックを入れたら、インターフェースサービスのタイプを **標準インターフェース** に設定します。

インターフェースオプションとホストアドレス

必要に応じて外部サービスタイプを選択することができます。

Siemens PLC Client

Siemens PLC Clientを選択する場合、ホストIPアドレス、**PLCスロット番号**、**DBブロック番号** の設定が必要です。

TCP Server

TCP Serverを選択する場合、ASCIIか、HEXか、プロトコル形式を選択する必要があります。HEXのプロトコル形式を使用すれば、**ビッグエンディアン**と**リトルエンディアン** を選択する必要があります。

また、実際の状況に応じてホストアドレスのポート番号を設定する必要があります。デフォルトのポート番号は50000です。

ロボット側に応じて、使用可能な通信形式を選択してインターフェースプログラムを作成することができます。

ロボットタイプ		標準インターフェースのサンプル例
産業用 ロボッ ト	ABB	HEXパッケージ
	FANUC	HEXパッケージ
	KUKA	HEXパッケージ
	YASKAWA	ASCIIパッケージ
	KAWASAKI	ASCIIパッケージ
	ROKAE	ASCIIパッケージ
	NACHI	ASCIIパッケージ
協働ロ ボット	FANUC CRX	Pluginプラグイン（HEX）
	UR	URCapプラグイン（ASCII）
	TM	Plug and Playプラグイン（ASCII）
	JAKA	ASCIIパッケージ（Addonプラグイン搭載）
	ELITE	ASCIIパッケージ
その他		ユーザーはロボット側のプログラムを作成する必要があり、Mech-Centerはサンプルを提供していません



ロボットインターフェースのサンプルプログラムパッケージと操作ドキュメントは、ソフトウェアインストールディレクトリ **Mech-Center/Robot_Interface** フォルダに格納されています。

PROFINET

PROFINETを使用する場合、ロボットの型番を設定する必要があります。

EtherNet/IP

EtherNet/IPを使用する場合、ロボットの型番を設定する必要があります。

MODBUS TCP SLAVE

Modbus TCP SLAVEは、スレーブIP、ポート番号、デバイス アドレス、およびバイト順を設定する必要があります（ユーザーは、マッチングするバイト順を選択するためにテストできます）。

ロボットの選択

現場に応じて、使用するロボットのブランドと型番を選択します。

設定が完了したら、「保存」をクリックしてMech-Centerを再起動します。再起動後、**インタ**

一フェースサービスを起動 をクリックしたら標準インターフェースが起動されます。

詳細設定

毎回送信する位置姿勢の最大数、Mech-Vizからデータ送信を待つタイムアウト時間、Mech-Visionからデータ送信を待つタイムアウト時間が設定可能です。

標準インターフェースを使用したMech-Vizサンプルプロジェクト

Mech-Mindソフトウェアシステムのインストールディレクトリの **Mech-Center/tool/viz_project** には、標準インターフェースを使用したMech-Vizサンプルプロジェクトは4つあります。

check_collision

このサンプルプロジェクトは、ビジョンシステムによるワーク把持のための簡単な経路計画と衝突検出を行うために使用されます。

outer_move

このサンプルプロジェクトは、Mech-Vizの仮想ロボットが外部クライアントから受信された位置に移動する必要がある場合に使用されます。

suction_zone

このサンプルプロジェクトは、Mech-Vizのために、パーティション制御された吸盤を使用して複数の段ボール箱を把持する際に、吸盤のパーティション制御信号を出力するために使用されます。

vision_result_reuse

このサンプルプロジェクトは、Mech-Vizは同じ撮影結果を使用してビジョンシステムにより複数のワーク把持を実行する場合、経路計画と衝突検出を行うために使用されます。

6.5. Adapter（アダプター）

標準インターフェースの機能がプロジェクトの要求を満たせない場合、ユーザーはAdapterプログラムを作成して複雑なプロセス制御を実現することができます。

本節では、以下の内容が含まれています。

- [Adapterの概要](#)
- [Adapterジェネレーター](#)
- [Adapterプログラミングのガイド](#)
- [Adapterプログラミングの例](#)



Adapterプロジェクトで追加のPythonライブラリを使用する必要がある場合は、それ

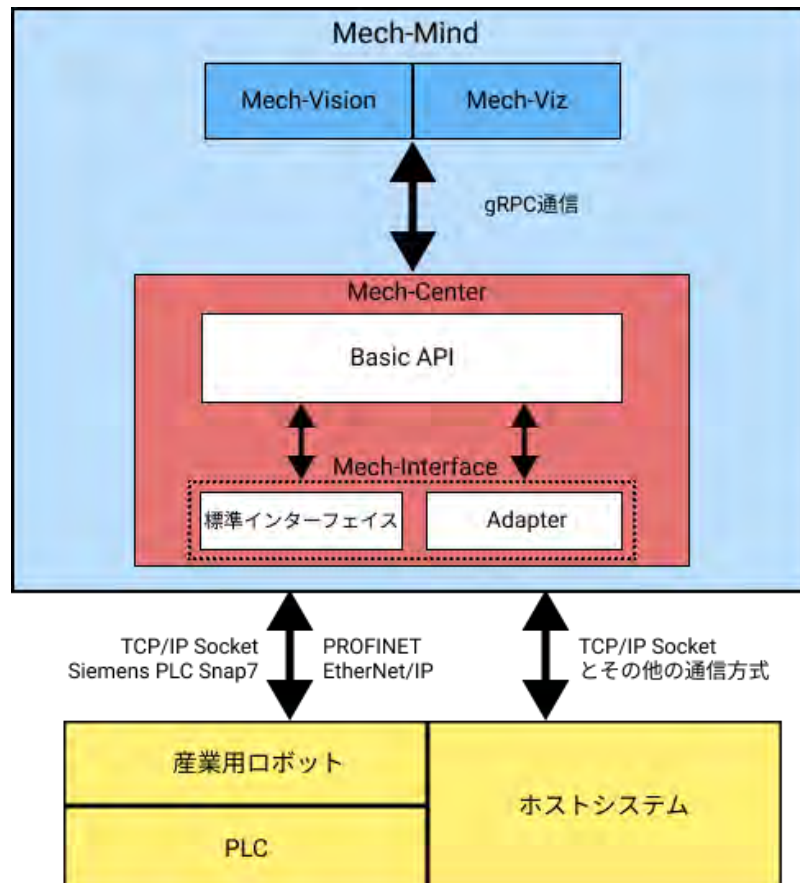
をMech-Mindソフトウェアシステムの **Mech-Center/python** ディレクトリにインストールしてください。Pythonライブラリのインストール方法は次のとおりです。

1. 「コマンドプロンプト」または「PowerShell」プログラムを起動します。
2. Mech-Mindソフトウェアシステムの **Mech-Center/python** ディレクトリに切り替えます。
3. 「./python -m pip install *library_name*」コマンドを実行します。

6.5.1. Adapterの概要

Adapterについて

AdapterはMech-Centerソフトウェアの通信コンポーネントです。Basic APIインターフェースを介してMech-VisionおよびMech-VizとのgRPC通信を行います。また、外部デバイスとの間では、TCP/IP Socket、HTTPおよびPLCプロトコル（三菱 PLC MC プロトコルなど）の様々な一般的な産業用通信方式を使用して通信を行います。



Adapterの機能

Adapterは次の機能に対応します。

- 内部でMech-VisionおよびMech-Vizの制御を実現しています。

タイプ	機能
Mech-Vision関連	Mech-Visionプロジェクトを実行してビジョン結果を取得
	Mech-Visionステップパラメータの設定
	Mech-Visionステップパラメータの読み取り
	Mech-Visionパラメータレシピの切替
Mech-Viz関連	Mech-Vizプロジェクトの起動
	Mech-Vizプロジェクトの停止
	Mech-Vizステップパラメータの設定
	Mech-Vizステップパラメータの読み取り
	ロボットハンドの番号の設定
	ロボットの動作速度の設定
	点群衝突検出のパラメータの設定
	Mech-Vizの実行ステータスを取得して返す
その他	詳細については、 Adapterプログラミングのガイド をご参照ください。

- 外部で、ユーザーインターフェース、データベース、ファイルの読み書き、Webシステムとの通信などのビジョン以外の機能を実現しています。

外部の機能を実現するためには、Pythonプログラミングを使用して開発する必要があります。

Adapterの開発

TCP/IP Socket通信方式に対して、Mech-Centerは **Adapterジェネレーター** 機能を提供しており、Adapterに初めて使用するユーザーが迅速にAdapterプログラムを生成し、Adapterプロジェクトを構築することに役立ちます。詳細については、[Adapterジェネレーターのマニュアル](#)をご参照ください。

ユーザーは生成されたAdapterを基にしてプログラムをカスタイズすることもできますし、ゼロからAdapterプログラムを作成することもできます。詳細については、[Adapterプログラミングのガイド](#)と [Adapterのプログラミング例](#)をご参照ください。

Adapterプロジェクトの設定

Adapterプロジェクトを作成したら、以下の手順に従ってAdapterプロジェクトを設定できます。

1. Mech-Centerを起動し、ツールバーの[**設定**]をクリックします。
2. **設定** 画面の左側で **Mech-Interface** をクリックし、 **Mech-Interfaceを使用** にチェックを入れて、 **インターフェイスサービスのタイプ** を **Adapter** に設定します。
3. **Adapterプロジェクト・ディレクトリ** を、Adapterプログラムが保存されているディレクトリに設定します。
4. 実際の状況に応じて **ホストアドレス** を設定します。ポートが通信先と一致する必要があります。
 - 通信先がサーバーの場合、 **ホストアドレス** は相手側のIPアドレスに設定する必要があります。
 - 通信先がクライアントの場合、 **ホストアドレス** は「0.0.0.0」に設定する必要があります。
5. [**保存**] をクリックしその画面を閉じます。その後、Mech-Centerを再起動します。
6. ツールバーの[**インターフェイスサービスを起動**] をクリックしてAdapterサービスを起動します。

インターフェイスサービスを起動 が **インターフェイスサービスを停止** になって、サービスステータス内に起動されたAdapterプロジェクトが表示されるとAdapterサービスが起動されている状態になります。

Adapterを簡単に理解したら、 [Adapterジェネレーターのマニュアル](#) を参照して、最初のAdapterプロジェクトをすばやく生成できます。

6.5.2. Adapterジェネレーター

本節では、Adapterジェネレーターと、Adapterジェネレーターを使用してAdapterプログラムをすばやく生成する方法について説明します。

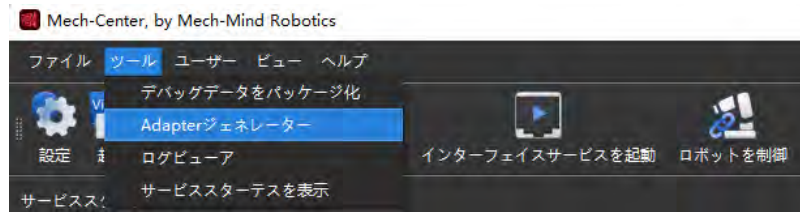
Adapterジェネレーターの紹介

Adapterジェネレーターは、Mech-Centerに統合されたコンポーネントです。Adapterジェネレーターは、TCP/IP Socket通信を使用し、Mech-Visionのみを使用してビジョンポイントを提供する場合にのみ適しています。

Adapterジェネレーターは、以下のような場合に役立ちます。

- Adapterプロジェクトをすばやく生成する場合。
- 複雑なニーズを満たすために、Adapterプログラミングの開発をすばやく学ぶ場合。

ツール ▶ **AAdapterジェネレーター** をクリックしてAdapterジェネレーターを起動して設定を行います。



Adapterジェネレーターは、管理者モードでのみ使用できます。

Adapterジェネレーターを使用してAdapterプログラムをすばやく生成



コンポーネントにマウスを置くと詳細な説明が表示されます。

ネットワーク設定 - サーバー/クライアント

この手順で **Adapter名**、**Adapterの役割**、**通信形式** を設定します。



パラメータ説明：

- **Adapter名**：Adapterプロジェクトの名前を設定します。
- **Adapterの役割**：AdapterをTCP/IP Socket通信用のサーバー (Server) またはクライアント (Client) として設定します。Adapterがクライアントとして機能し、サーバーがクライアントに対してポート制限を持っている場合は、**ポートバインド** にチェックを入れる必要があります。



通信先と正常に通信するために、Adapterプロジェクトを設定するときはホストIPアドレスとポートを正しく設定する必要があります。詳細については、[Adapterプロジェクトの設定](#) をご参照ください。

- **形式**：通信時のデータ送信フォーマットを設定し、ASCII文字列またはHEX（16進数）をサポートします。「HEX」に設定されている場合、バイトオーダー（エンディアン）も「ビッグエンディアン」または「リトルエンディアン」で指定する必要があります。

Mech-Viz/Mech-Vision設定－位置姿勢と関節角度



- この設定を行う前に、Mech-Visionプロジェクトと衝突検出のためのMech-Vizプロジェクトを開いて自動読み込みに設定し、プロジェクトがMech-Centerに読み込まれていることを確認します。
- Mech-Centerは、衝突検出のためのMech-Vizサンプルプロジェクト「check_collision」を提供しています。それは、Mech-Centerのインストールパスの「tool\\viz_project」フォルダに格納されています。



「check_collision」サンプルプロジェクトに示すように、プロジェクトは「ビジョン処理による認識」によってトリガーされて撮影し、非移動ステップを含める必要があります。また、notify_1、notify_2およびvisual_look_1などのステップ名を変更することはできません。

この手順でMech-VizおよびMech-Visionプロジェクトのパラメータを設定します。



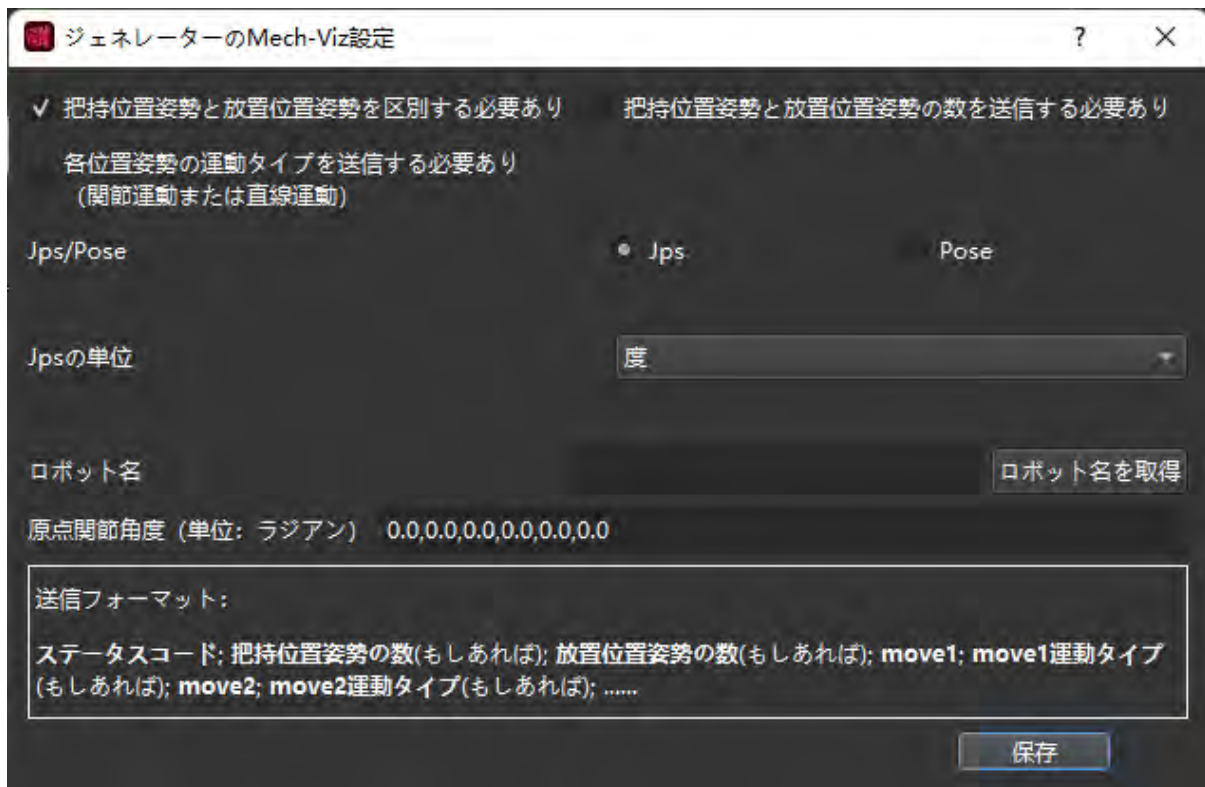
Mech-Vision関連のパラメータ

- **Mech-Visionプロジェクト名**：Mech-Visionプロジェクト名を選択します。ドロップダウンリストからAdapterと通信するためのMech-Visionプロジェクトを選択します。
- **位置姿勢の数**：通信先に送信する位置姿勢の数を設定します。
- **位置姿勢の形式**：四元数またはオイラー角を選択可能です。
- **位置姿勢の単位**：一般的に、ミリメートルおよび度が使用されます。
カメラ：カメラの取り付け方法を選択します。ETHとEIHの2つの方法があります。

- **位置姿勢の基準座標系**：送信する位置姿勢が基づく座標系を選択します。一般に、位置姿勢はロボットのベース座標系に基づいています。位置姿勢は、EIHモードでロボットがロボット先端の位置姿勢を提供できない場合にのみ、ツール座標系に基づきます。
- **小数点以下の桁数を保持**：送信する位置姿勢の小数点以下の桁数を設定します。最大の桁数は10です。

Mech-Viz関連のパラメータ

- **Mech-Vizを使って衝突を検出**：チェックを入れたら、ビジョンポイントがMech-Vizによって検出・計算されます。それにより、計画に失敗した位置姿勢は除去され、衝突のない把持位置姿勢が選出されます。
- **Mech-Viz設定**：Mech-Vizを使って衝突を検出 にチェックを入れると有効になります。クリックすれば **ジェネレーターのMech-Viz設定** 画面が表示されます。設定が完了したら、右下隅にある **[保存]** をクリックします。

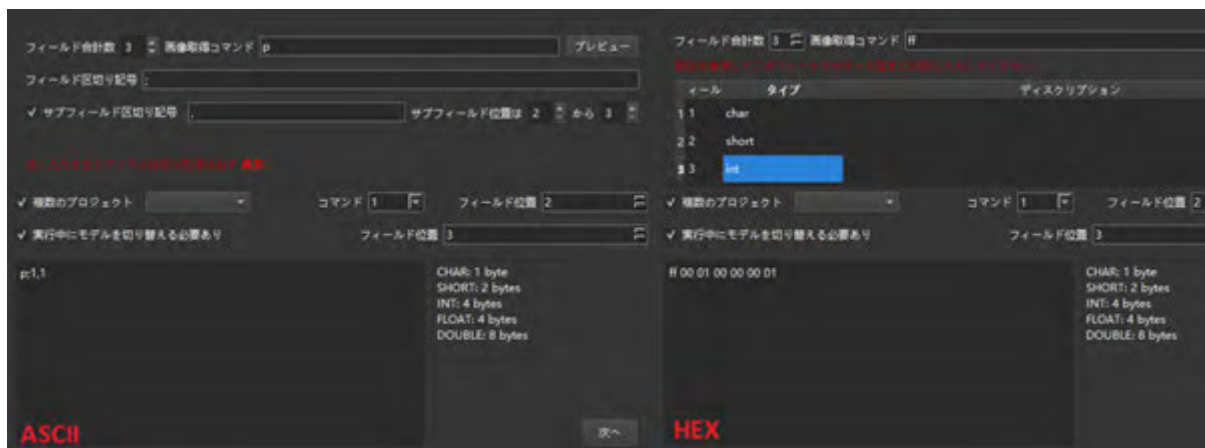


- **把持位置姿勢と配置位置姿勢を区別する必要あり**：把持位置姿勢は「ビジョン処理による移動」の前のすべての位置姿勢（ビジョン処理による移動を含む）であり、配置位置姿勢は「ビジョン処理による移動」の後のすべての位置姿勢です。場合によっては、ロボットがステップによって把持動作と配置動作を区別する必要があります。
- **把持位置姿勢と配置位置姿勢の数を送信する必要あり**：把持位置姿勢の数が多い場合、数量フィールドをつけることができます。チェックを入れた後、デフォルトの位置姿勢の数が1以上であればこのフィールドをつけます。
- **各位置姿勢の運動タイプを送信する必要あり**：Mech-Vizでの移動ステップは、関節運動または直線運動と2種類があります。

- **コードをアップデート**：デフォルトで、関節運動の対応コードは1で、直線運動の対応コードは2です。コードはカスタマイズ可能で、それを変更して[**コードをアップデート**]をクリックすると有効になります。
- **Jps/位置姿勢**：位置姿勢の送信形式で、デフォルトでJpsを使用します。
- **Jpsの単位／位置姿勢の単位**：通常、送信する位置姿勢の形式は**Jps**の場合、Jpsの単位が度に設定されます。送信する位置姿勢の形式は**Pose**の場合、位置姿勢の単位はメートルに設定されます。
- **ロボット名**：ロボットサービスの名前を設定します。Mech-Vizでロボットの動作をシミュレートするには、ロボット実機のサービスが必要です。このサービスは、生成されたAdapterによってシミュレートされます。ロボット名はこのサービスの名前であり、Mech-Vizでのロボット名と一致している必要があります。「ロボット名を取得」をクリックすれば、ロボット名が正常に追加されます。
- **原点関節角度（ラジアン単位）**：これは、Mech-Vizでロボット動作の基準原点を指します。単位はラジアンで、コンマで区切られます。Mech-Vizの「移動」ステップを編集して原点とし、原点の関節位置をコピーすることができます。

ストリームの受信設定 - 受信するフィールド

この手順で受信フィールドのフォーマットを設定する必要があります。画像取得コマンド、複数のプロジェクト、フィールド合計数、フィールドタイプ、フィールド区切り記号とサブフィールド区切り記号を設定する必要があります。設定が完了したら、右下隅にある[次へ]をクリックします。



- **フィールド合計数**：設定するパラメータの数によって異なります。その範囲を「1~10」に設定できます。また、フィールドに画像取得コマンドがなければなりません。
- **画像取得コマンド**：外部からMech-Mindソフトウェアシステムに画像取得コマンドを送信し、カメラで撮影します。ASCIIの場合、pのような文字を入力し、フィールド位置をデフォルトの1にすることをお勧めします。HEXの場合、0xffまたはffなどの16進数形式の整数を使用する必要があります。
- **フィールド区切り記号とサブフィールド区切り記号**：ASCIIの場合は設定が必要です。2つ以上のフィールドがある場合は、フィールド区切り記号を入力する必要があります。追加情

報で別のフィールド区切り記号が必要な場合は、サブフィールド区切り記号も必要です。また、サブフィールド区切り記号の開始範囲と終了範囲を指定できます。

- **フィールドタイプとディスクリプション**：HEXの場合は設定が必要です。CHAR、SHORT、INT、FLOATおよびDOUBLEが選択可能です。**ディスクリプション**はフィールドの機能または意味を説明するために使用されます。
- **複数のプロジェクト**：この設定は選択可能です。1つの項目に複数のMech-Visionプロジェクトがある場合、外部からのコマンドによって異なるMech-Visionプロジェクトを呼び出す必要があります。その中、コマンドを設定することができます。



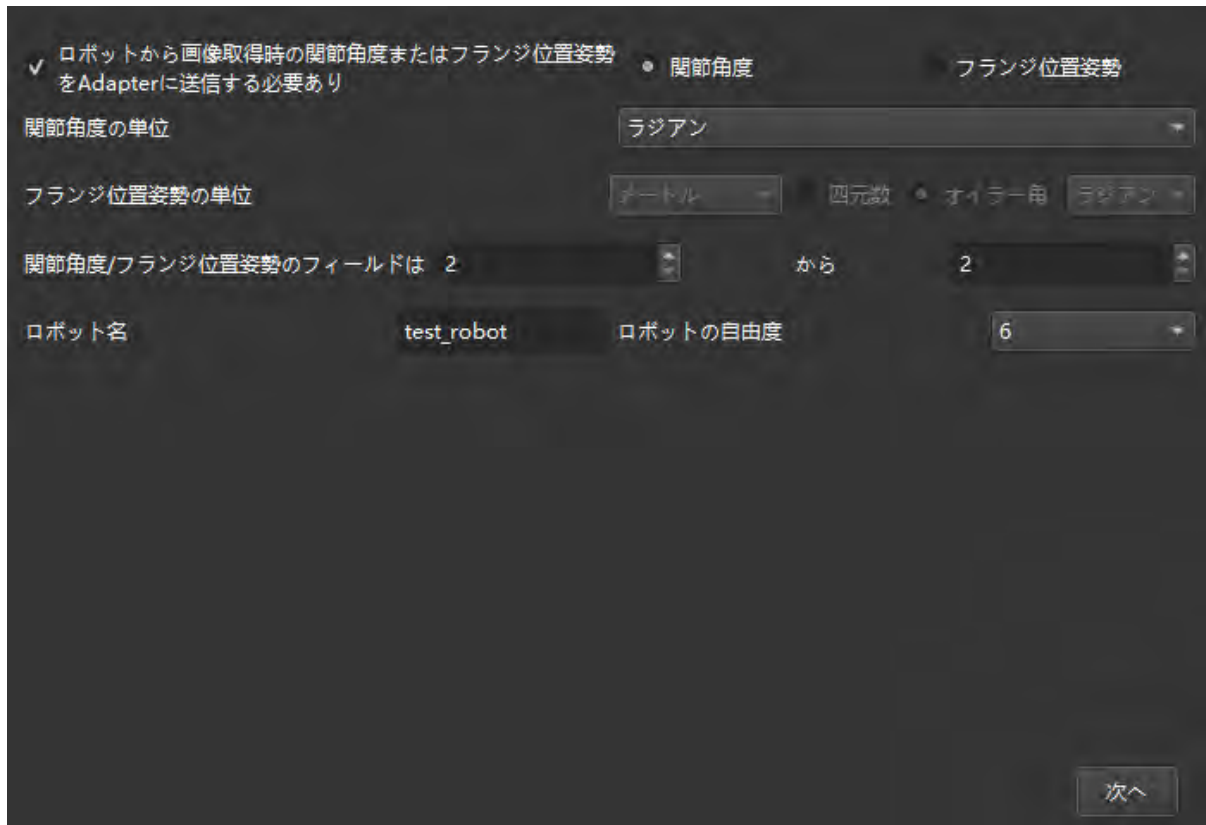
プロジェクトごとに唯一のコマンドが対応します。また、フィールドの位置も唯一で、他のフィールドと重複してはいけません。

ロボットの設定 - Eye in Hand



[Mech-Viz/Mech-Vision設定一位置姿勢と関節角度](#)の手順で **カメラ** が **Eye In Hand** に設定されている場合に使用できます。

この手順で、画像撮影時のロボットの位置姿勢の形式を設定してから、[次へ]をクリックします。



パラメータ説明：

- **ロボットから画像取得時の関節角度またはフランジ位置姿勢をAdapterに送信する必要あり**：通信先はロボットベース座標系に基づいた位置姿勢を提供する必要がある場合、画像取得時のロボットからの関節角度またはフランジ姿勢が必要であるため、これにチェックを入れ

てください。チェックを入れたら、位置姿勢を関節角度またはオイラー角に設定することができます。

- **関節角度の単位**：ラジアンまたは度を選択可能です。
- **フランジ位置姿勢の単位**：フランジ位置姿勢の単位（四元数またはオイラー角）を設定します。四元数の場合はメートルとミリメートルを選択可能で、オイラー角の場合は度とラジアンを選択可能です。
- **関節角度/フランジ位置姿勢のフィールド位置**：全体フィールド内の関節角度または位置姿勢の開始フィールドと終了フィールドの位置を設定します。

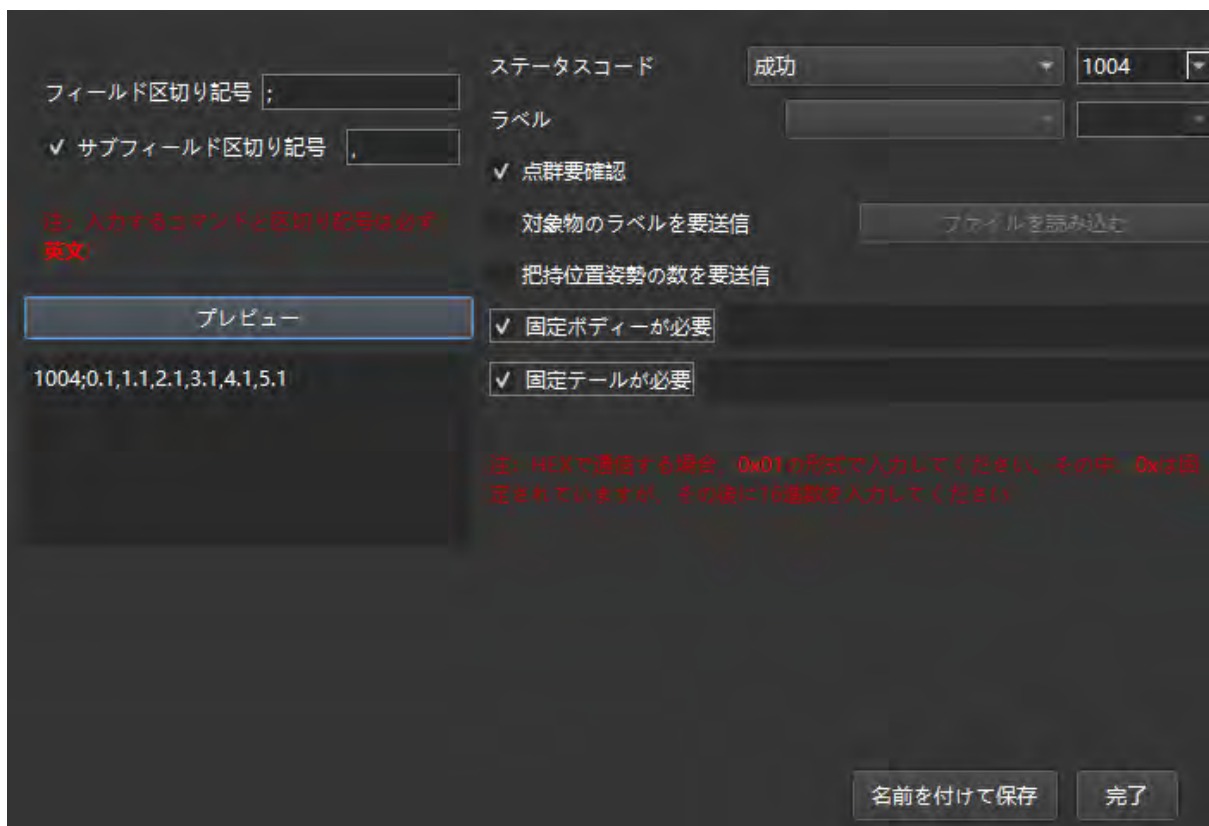


インデックス位置は1からカウントされ、他のフィールドと重複することはできません。例えば、インデックス位置「1」は画像取得コマンドです。

- **ロボット名**：ロボットサービスをマークする名で、Mech-Vizに使用されているロボットの名と一致させる必要があります。
- **ロボットの自由度**：現在は4軸と6軸のロボットに対応しています。実際の状況に応じて、対応するロボットの自由度を選択してください。

送信するストリームの設定-フィールド送信

下図に示すように、この手順で送信する位置姿勢の形式を設定してから、[次へ]をクリックします。



パラメータ説明：

- **フィールド区切り記号 と サブフィールド区切り記号**：区切り記号の形式を設定します。ASCIIの場合は設定が必要です。2つ以上のフィールドがある場合は、フィールド区切り記号を入力する必要があります。追加情報で別のフィールド区切り記号が必要な場合は、サブフィールド区切り記号も必要です。また、サブフィールド区切り記号の開始範囲と終了範囲を指定できます。
- **ステータスコード**：送信形式を設定します。各ステータスが唯一のステータスコードに対応します。
- **点群要確認**：チェックを入れれば、点群がAdapterによって確認されます。点群がない場合は、対応するステータスコードが出力されます。
- **対象物のラベルを要送信**：対象物のラベルの送信は、Mech-Visionで認識されたラベルを通信先に送信することです。各ラベルは位置姿勢の後に付きます。通信先にとってラベルの文字列を解析するのが不便な場合は、対応のラベルのコードを指定することもできます。その中、すべてのラベルの文字列を含むラベルファイルを読み込む必要があります。ラベルファイルの形式はjson配列形式である必要があります。
- **把持位置姿勢の数を要送信**：今回送信する位置姿勢の数を送信します。
- **固定ボディーが必要**：ビジョン認識に失敗した場合、通信先にメッセージ（エラーコード後のメッセージ）を送信します。
- **固定テールが必要**：チェックを入れれば、送信するデータの後にテールを固定する標識が追加されます。



通信形式を16進数（HEX）に設定する場合、ステータスコード、位置姿勢の数、位置姿勢のタイプを設定する必要があります。

上記の設定をすべて完了したら、「完了」または「名前を付けて保存」をクリックしてAdapterプロジェクトを保存します。Adapterプロジェクトを保存した後、[Adapterプロジェクトの設定](#)を参照してそれを実装することができます。

Adapterプログラミング開発に関する詳細情報については、以下の内容をお読みください。

- [Adapterプログラミングのガイド](#)
- [Adapterプログラミングの例](#)

6.5.3. Adapterプログラミングのガイド

本節では、Adapterのプログラミングスタイル規則とプログラミング文法の知識をについて説明していきます。

Adapter開発者は、以下のことを事前に把握しておく必要があります。

- Pythonの基本文法
- JSONデータ形式

最初に **Adapterプログラミングスタイル規則** を把握しておく必要があります。

- [Adapterプログラミングスタイル規則](#)

次に、Adapterプログラミングに関連する **抽象親クラスインターフェイス** と **Util** パッケージを学び、Adapterの最も基本的な親クラスと一般的な関数を理解します。

- [抽象親クラスインターフェイス](#)
- [Adapter utilパッケージ](#)

最後に、要件に基づいて抽象親クラスインターフェイスを実装し、それによって内部（Mech-VisionおよびMech-Viz）と外部デバイスとの通信を実現します。

- [インターフェースの取得](#)
- [Mech-Visionインターフェース](#)
- [Mech-Vizインターフェース](#)
- [RobotServiceインターフェース](#)
- [その他のインターフェース](#)

6.5.3.1. Adapterプログラミングスタイル規則

AdapterはPython言語で記述されているため、Adapter開発者は [Pythonプログラミング規則](#) を厳守する必要があります。

また、Adapterプログラミングのメンテナンス性と読みやすさを向上させるために、Adapterは本節で記述される規則も厳守する必要があります。

命名規則

1. クラス名はキャメルケースの形式で命名されます。例：

```
class AdapterWidget
```

2. クラスのメンバー変数とメンバー関数は、アンダースコアで結合された小文字の単語を使用します。次に例を示します。例：

```
self.pick_count # Note that variables are generally nouns
def start_viz(self): # Note that the function name is generally a verb + object
    pass
```

3. クラスのメンバー変数とメンバー関数がクラス内でのみ使用される場合は、名前の前にアンダースコアを追加して、その名前をクラス外で使用することは推奨されていませんが、クラス外で使用できます。次に例を示します。

```
self._socket = socket() # Indicates that the _socket variable is only used inside the class
def _init_widget(self): # Indicates that the _init_widget function is only used inside the
class
    pass
```

4. 定数はアンダースコアで区切られた大文字の単語を使用します。次に例を示します。

```
ADAPTER_DIR = "D:/adapter_for_communication"
```

コメントの規則

コメントは適切な場所や必要な箇所にのみ追加する必要があります。

- 一行コメント

一行コメントは#で始めます。

- 複数行コメント

複数行コメントは"""と"""の間に記述します。

- 関数、クラス、またはパッケージ紹介のコメント

コメントは、関数の下、クラスの下、またはパッケージの先頭に配置されます。次に例を示します。

```
def viz_is_running(self):

    """
    Will be called when find viz is running during starting viz.
    """

class Adapter(QObject):

    """
    Base class which encapsulates Viz/Vision/Hub/Robserver interfaces.
    """

    """
    Service base classes.
    """
```



```
import logging
```

ログ規則

ログは、トラブルシューティングや問題の分析に役立つ場合があります。ログには、プログラムの実行中に発生した情報、警告、エラーなどが記録されます。ログ情報を適切なタイミングで出力することは重要です。

Adapterは2つのログ出力方法をサポートしています。

- **print**：この方法は、ログをコンソールに出力するだけです。実行時のリアルタイム監視には便利ですが、プログラムはエラーが発生するとメッセージが失われます。したがって、実際の生産で使用することはお勧めしません。
- **logging**：この方法は、ログ表示形式をフォーマットすることをサポートし、ログをログファイルに保存することもサポートします（オプション機能）。したがって、この方法を使用することをお勧めします。

6.5.3.2. 抽象親クラスインターフェース

抽象親クラスインターフェースとは、子クラスが親クラスを継承する際に、実際のニーズに応じて書き換えることができる関数を指します。本節では、次の抽象親クラスについて説明します。

Communication

通信関連のクラスのソースファイルは、Mech-Mindソフトウェアシステムのインストールディレクトリの **Mech-Center/src/interface/communication.py** ファイルに格納されています。

Communicationクラス

Communicationクラスは、通信に関する基本的な機能を提供するクラスであり、一連のインターフェースを提供します。サーバーまたはクライアントは、このクラスのインターフェース関数を書き換える必要があります。

クラス関数	説明
is_connected()	現在の接続が確立されているかどうかを確認します。
set_recv_size()	受信データの長さを設定します（デフォルトは1024バイト）。
send()	データを送信するためのインターフェース関数です。
recv()	データを受信するためのインターフェース関数です。
close()	接続を切断するためのインターフェース関数です。
before_recv()	インターフェース関数で、データを受信する前に実際の状況に応じてロジックを追加できます（この関数を書き換えることができます）。

クラス関数	説明
after_recv()	インターフェース関数で、データを受信した後に実際の状況に応じてロジックを追加できます（この関数を書き換えることができます）。
after_handle()	インターフェース関数で、データを処理した後に実際の状況に応じてロジックを追加できます（この関数を書き換えることができます）。

TcpServerクラス

TcpServerクラスは、TCP/IP Socketのサーバーをパッケージ化します。

クラス関数	説明
bind_and_listen()	ポートをバインドします。
local_socket()	ネイティブソケット情報を提供します。
remote_socket()	リモートソケット情報を提供します。
accept()	クライアント接続を受け入れます。
send()	データを送信します。
recv()	データを受信します。
close()	Socket接続を終了します。
close_client()	クライアント接続を閉じます。

TcpClinetクラス

TcpClientクラスは、TCP/IP Socketのクライアントをパッケージ化します。

クラスのプロパティ	説明
is_bind_port	ポートをバインドするかどうか（接続されたクライアントのポートがサーバーによって制限されている場合、この変数はTrueに設定する必要があります）

クラス関数	説明
send()	データを送信します。
recv()	データを受信します。
close()	接続を閉じます。
set_timeout()	タイムアウトを設定します（単位は秒）。

クラス関数	説明
reconnect_server()	サーバーを再接続します。
after_connect_server()	インターフェース関数で、サーバーへの最初の接続が成功した後の操作です。
after_reconnect_server()	インターフェース関数で、サーバーへの再接続が成功した後の操作です。
after_timeout()	インターフェース関数で、タイムアウト後の操作です。

Adapter (アダプター)

Adapter関連のクラスのソースファイルは、Mech-Mindソフトウェアシステムのインストールディレクトリの [Mech-Center/src/interface/adapter.py](#) ファイルに格納されています。

Adapterクラス

Adapterクラスは、Mech-Viz、Mech-Vision、Mech-CenterおよびRobserverに関連する呼び出しをパッケージ化します。その中には、Mech-Vizの起動、Mech-Vizの停止、Mech-VisionまたはMech-Vizステップパラメータの設定、Mech-Vision認識の実行などの関数が含まれます。AdapterプログラムがMech-VizまたはMech-Visionを呼び出す限り、Adapterクラスを継承する必要があります。

Adapterクラスのプロパティを次の表に示します。

クラスのプロパティ	説明
viz_project_dir	現在のMech-Vizプロジェクトパス
vision_project_name	現在のMech-Visionプロジェクト名
is_simulate	Mech-Vizをシミュレーションで実行するかどうか
is_keep_viz_state	Mech-Vizを最後に停止した時の状態を保持するかどうか
is_save_executor_data	Mech-Vizアクチュエータのデータを保存するかどうか
is_force_simulate	Mech-Vizを強制的にシミュレーションで実行するかどうか
is_force_real_run	Mech-Vizを強制的に実行するかどうか
code_signal	Mech-CenterのメインインターフェースでAdapterメッセージを表示するための信号（エラーコード付きのメッセージ）
msg_signal	Mech-CenterのメインインターフェースでAdapterメッセージを表示するための信号（エラーコードのないメッセージ）

クラスのプロパティ	説明
i_code_signal	Mech-CenterのメインインターフェースでMech-Interfaceメッセージを表示するための信号（エラーコード付きのメッセージ）
i_msg_signal	Mech-CenterのメインインターフェースでMech-Interfaceメッセージを表示するための信号（エラーコードのないメッセージ）
viz_finished_signal	Mech-Viz実行終了信号（正常終了または異常終了）
connect_robot_signal	ロボット接続/切断の信号
start_adapter_signal	Adapterの起動信号
service_name_changed	Mech-CenterメインインターフェースでMech-VizおよびMech-Visionステータを表示するための信号
setting_infos	Mech-Centerの設定情報
service_name	登録サービス名

Adapterクラスのプロパティを次の表に示します。

クラス関数	説明
on_exec_status_changed()	Mech-VizおよびMech-Visionのステータス情報を受信します。
register_self_service()	Adapterサービスを登録します。
vision_project_dirs(self):	Mech-Visionプロジェクトのフォルダパスを探します。
vision_project_names()	すべてのMech-Visionプロジェクト名を探します。
vision_project_names_in_center()	Mech-Centerに登録されたすべてのMech-Visionプロジェクト名を探します。
is_viz_registered()	Mech-Vizプロジェクトが登録されているかどうかを確認します。
is_viz_in_running()	Mech-Vizが実行しているかどうかを確認します。
is_vision_started()	Mech-Visionプロジェクトが登録されているかどうかを確認します。
find_services()	サービスを探します。
before_start_viz()	Mech-Vizの起動前に呼び出される関数です。
after_start_viz()	Mech-Vizの起動後に呼び出される関数です。

クラス関数	説明
<code>viz_not_registerd()</code>	Mech-Vizの起動後、Mech-Vizプロジェクトが登録されていない場合はこの関数が呼び出されます。
<code>viz_is_running()</code>	Mech-Vizの起動後、Mech-Vizが実行している場合はこの関数が呼び出されます。
<code>viz_run_error()</code>	Mech-Vizの起動後、Mech-Viz実行中にエラーが発生した場合はこの関数が呼び出されます。
<code>viz_run_finished()</code>	Mech-Vizの実行が終了したときに呼び出される関数です。
<code>viz_plan_failed()</code>	Mech-Vizの計画が失敗したときに呼び出される関数です。
<code>viz_no_targets()</code>	Mech-Vizの計画に移動点がない場合に呼び出される関数です。
<code>viz_unreachable_targets()</code>	Mech-Vizの計画に到達不能な移動点がある場合に呼び出される関数です。
<code>viz_collision_checked()</code>	Mech-Vizの計画に衝突が検出された場合に呼び出される関数です。
<code>parse_viz_reply()</code>	Mech-Vizの返信を解析します。
<code>wait_viz_result()</code>	Mech-Vizの返信を待ちます。
<code>start_viz()</code>	Mech-Vizを起動します。
<code>stop_viz()</code>	Mech-Vizを停止します。
<code>pause_viz()</code>	Mech-Vizを一時停止します。
<code>find_vision_pose()</code>	Mech-Visionプロジェクトをトリガーして撮影します。
<code>async_call_vision_run()</code>	Mech-Visionプロジェクトを非同期にトリガーして撮影します。
<code>async_get_vision_callback()</code>	Mech-Visionからの結果を非同期的に受信します。
<code>deal_vision_result()</code>	Mech-Visionからの結果を処理します。
<code>set_step_property()</code>	Mech-Visionのステップパラメータを設定します。
<code>read_step_property()</code>	Mech-Visionのステップパラメータを読み取ります。
<code>select_parameter_group()</code>	Mech-Visionプロジェクトのパラメータレシピを選択します。
<code>set_task_property()</code>	Mech-Vizのステップパラメータを設定します。
<code>read_task_property()</code>	Mech-Vizのステップパラメータを読み取ります。

クラス関数	説明
get_digital_in()	DIを取得します。
set_digital_out()	DOを設定します。
before_start_adapter()	Adapterの起動前に呼び出される関数です。
start()	Adapterを起動します。
close()	Adapterを終了します。
handle_command()	受信した外部コマンドを処理します。

TcpServerAdapterクラス

TcpServerAdapterクラスはAdapterを継承し、TcpServerの機能をパッケージ化します。詳細は次の通りです。

```
class TcpServerAdapter(Adapter):
    def __init__(self, host_address, server=TcpServer):
        super(TcpServerAdapter, self).__init__()
        self.init_server(host_address, server)

    def init_server(self, host_address, server=TcpServer):
        self._server = server(host_address)

    def set_recv_size(self, size):
        self._server.set_recv_size(size)

    def send(self, msg, is_logging=True):
        return self._server.send(msg, is_logging)

    def recv(self):
        return self._server.recv()

    def start(self):
        self.before_start_adapter()
        while not self.is_stop_adapter:
            try:
                self._server.before_recv()
                cmds = self._server.recv()
                logging.info("Received raw data from client:{}".format(cmds))
                if not cmds:
                    logging.warning("Adapter client is disconnected!")
                    self.code_signal.emit(logging.WARNING, CENTER_CLIENT_DISCONNECTED)
                    self._server.close_client()
```

```

        self.accept()
        continue
    self._server.after_recv()
except socket.error:
    logging.warning("Adapter client is closed!")
    self.code_signal.emit(logging.WARNING, CENTER_CLIENT_DISCONNECTED)
    self._server.close_client()
    self.accept()
except Exception as e:
    logging.exception("Exception occurred when receiving data from client: {}".format(e))
else:
    try:
        self.handle_command(cmds)
        self._server.after_handle()
    except Exception as e:
        self.msg_signal.emit(logging.ERROR, _translate("messages", "Handle command exception: {}".format(e)))
        logging.exception("Adapter exception in handle_command(): {}".format(e))

def close(self):
    super().close()
    self._server.close()

def before_start_adapter(self):
    super().before_start_adapter()
    self.accept()

def accept(self):
    if self.is_stop_adapter:
        return
    self.code_signal.emit(logging.INFO, CENTER_WAIT_FOR_CLIENT)
    self._server.accept()
    if self._server.is_connected():
        self.code_signal.emit(logging.INFO, CENTER_CLIENT_CONNECTED)
        self.msg_signal.emit(logging.INFO, _translate("messages", "Client address is") + "{}".format(self._server.remote_socket()[1]))

```

TcpClientAdapterクラス

TcpClientAdapterクラスはAdapterを継承し、TcpClientの機能をパッケージ化します。詳細は次の通りです。

```

class TcpClientAdapter(Adapter):

    def __init__(self, host_address):
        super().__init__()

```

```

        self.init_client(host_address)

    def init_client(self, host_address, client=TcpClient):
        self._client = client(host_address)

    def set_bind_port(self, is_bind=True):
        self._client.is_bind_port = is_bind

    def set_rcv_size(self, size):
        self._client.set_rcv_size(size)

    def send(self, msg, is_logging=True):
        self._client.send(msg, is_logging)

    def rcv(self):
        return self._client.rcv()

    def start(self):
        self.reconnect_server(False)
        while not self.is_stop_adapter:
            try:
                self._client.before_rcv()
                cmds = self._client.rcv()
                if not cmds:
                    self.reconnect_server()
                    continue
                logging.info("Received command from server:{}".format(cmds))
                self._client.after_rcv()
            except socket.timeout:
                logging.warning("Socket timeout")
                self._client.after_timeout()
            except socket.error:
                sleep(5)
                self.reconnect_server()
            except Exception as e:
                logging.exception("Exception occurred when receiving from server: {}".format(
e))
            else:
                try:
                    self.handle_command(cmds)
                except Exception as e:
                    self.msg_signal.emit(logging.ERROR, _translate("messages", "Handle command
exception: {}".format(e)))
                    logging.exception("Adapter exception in handle_command(): {}".format(e))

    def close(self):
        super().close()
        self._client.close()

```



```
def reconnect_server(self, is_reconnect=True):
    self._client.reconnect_server()
    if self.is_stop_adapter:
        return
    if self._client.is_connected():
        self.code_signal.emit(logging.INFO, CENTER_CONNECT_TO_SERVER)
    else:
        self.code_signal.emit(logging.WARNING, CENTER_SERVER_DISCONNECTED)
```

TcpMultiplexingServerAdapterクラス

TcpMultiplexingServerAdapterクラスはAdapterを継承しており、主に複数のクライアントの接続に使用されます。詳細は次の通りです。

```
class TcpMultiThreadingServerAdapter(Adapter):
    def __init__(self, address):
        super().__init__()
        self._servers = {}
        self.add_server(address)
        self.sockets = {}
        self.clients_ip = {}
        self.thread_pool = ThreadPoolExecutor(max_workers=4, thread_name_prefix
        ="tcp_multi_server_thread")
        self.thread_id_socket_dict = {}
        self.set_recv_size()

    def set_recv_size(self, size=1024):
        self.recv_size = size

    def _find_client_ip(self, sock):
        for k, v in self.sockets.items():
            if v == sock:
                return k

    def _find_server(self, sock):
        for k, v in self._servers.items():
            if v == sock:
                return k

    def add_server(self, host_address):
        server = TcpServer(host_address)
        server.bind_and_listen()
        self._servers[server] = server.local_socket()

    def set_clients_ip(self, clients_ip):
        """
```

```

        Must be called before start().
        `clients_ip` is a dict(key is client ip, value is client description).
    """
    self.clients_ip = clients_ip

    def add_connection(self, ip_port, sock):
        self.sockets[ip_port] = sock
        logging.info("Add {}, connections: {}".format(ip_port, self.sockets))
        self.msg_signal.emit(logging.INFO, _translate("messages", "The client {} gets
online.").format(ip_port))

    def del_connection(self, ip):
        logging.info("Del {}, connections: {}".format(ip, self.sockets))
        if self.client_connection(ip):
            self.client_connection(ip).close()
            self.sockets.pop(ip)
            self.msg_signal.emit(logging.WARNING, _translate("messages", "The client {} gets
offline.").format(ip))

    def client_connection(self, client_ip):
        return self.sockets.get(client_ip)

    def check_read_events(self, rs):
        for s in rs:
            if s in self._servers.values(): # recv connection
                server = self._find_server(s)
                if self.is_stop_adapter:
                    return
                server.accept()
                client_socket, client_addr = server.remote_socket()
                ip_port = "{}:{}".format(str(client_addr[0]), str(client_addr[1]))
                self.add_connection(ip_port, client_socket)
            elif s in self.sockets.values(): # recv data
                client_ip = self._find_client_ip(s)
                if not client_ip:
                    continue
                msg = self.recv_by_s(s)
                if not msg:
                    self.del_connection(client_ip)
                    return
                try:
                    future = self.thread_pool.submit(self.handle_command_thread, s, msg)
                except Exception as e:
                    logging.exception("Adapter exception in handle_command(): {}".format(e))

    def handle_command_thread(self, s, msg):
        thread_id = threading.get_ident()
        self.thread_id_socket_dict[thread_id] = s

```

```

self.handle_command(msg)
# del self.thread_id_socket_dict[thread_id]

def send(self, msg, is_logging=True):
    thread_id = threading.get_ident()
    sock = self.thread_id_socket_dict.get(thread_id)
    len_total = len(msg)
    while msg:
        if sock:
            len_sent = sock.send(msg)
        else:
            for v in self.sockets.values():
                try:
                    len_sent = v.send(msg)
                except Exception as e:
                    logging.warning(e)
            if not len_sent:
                logging.warning("Connection lost, close the client connection.")
                return len_sent
            if is_logging:
                logging.info("Server send: {}, len_sent: {}".format(msg, len_sent))
            msg = msg[len_sent:]
    return len_total

def recv(self):
    thread_id = threading.get_ident()
    sock = self.thread_id_socket_dict.get(thread_id)
    return self.recv_by_s(sock)

def recv_by_s(self, sock):
    msg = b""
    try:
        msg = sock.recv(self.recv_size)
    except socket.error:
        logging.error("The client is closed!")
    if msg:
        logging.info("Received message: {}".format(msg))
    return msg

def check_task(self):
    """
    Interface.
    """

def close(self):
    super().close()
    for server in self._servers.keys():
        server.close()

```

```

for client_ip in self.sockets.keys():
    try:
        self.client_connection(client_ip).close()
        logging.info("Close socket :{}".format(client_ip))
    except Exception as e:
        logging.warning("Close socket error:{}, exception:{}".format(client_ip, e))
self.sockets = {}

def start(self):
    self.before_start_adapter()
    while not self.is_stop_adapter:
        avalible_sockets = list(self.sockets.values()) + list(self._servers.values())
        rs, _, _ = select(avalible_sockets, [], [], 0.1)
        self.check_read_events(rs)
        try:
            self.check_task()
        except Exception as e:
            self.msg_signal.emit(logging.ERROR,
                                _translate("messages", "Handle command exception: {}".
                                .format(e)))
            logging.exception("Exception when check task:{}".format(e))
            sleep(5)

```

IOAdapterクラス

IOAdapterクラスはAdapterを継承し、DIを周期的に取得する操作をパッケージ化します。詳細は次の通りです。

```

class IOAdapter(Adapter):
    robot_name = None
    check_rate = 0.5

    def __init__(self, host_address):
        super().__init__()
        self.last_gi = 0

    def get_digital_in(self, timeout=None):
        return super().get_digital_in(self.robot_name, timeout)

    def set_digital_out(self, port, value, timeout=None):
        super().set_digital_out(self.robot_name, port, value, timeout)

    def _check_gi(self):
        gi_js = self.get_digital_in()
        gi = int(json.loads(gi_js.decode())["value"])
        if self.last_gi != gi:
            self.last_gi = gi

```

```

        logging.info("Check GI signal status: {}".format(gi))
        self.handle_gi(gi)

    def start(self):
        self.before_start_adapter()
        while not self.is_stop_adapter:
            try:
                self._check_gi()
            except Exception as e:
                logging.exception(e)
                self.check_gi_failed()
                sleep(self.check_rate)

    def handle_gi(self, gi):
        """
        Interface.
        """

    def check_gi_failed(self):
        """
        Interface.
        """

```

AdapterWidgetクラス

AdapterWidgetクラスは、Adapter UIをカスタマイズするための親クラスであり、UIをカスタマイズする機能はそれから継承する必要があります。詳細は次の通りです。

```

class AdapterWidget(QWidget):

    def set_adapter(self, adapter):
        self.adapter = adapter
        self.after_set_adapter()

    def after_set_adapter(self):
        """
        Interface.
        """

    def close(self):
        super().close()
        """
        Interface.
        """

```

Service

サービス関連のクラスのソースファイルは、Mech-Mindソフトウェアシステムのインストールディレクトリ `Mech-Center/src/interface/services.py` ファイルに格納されています。

NotifyServiceクラス

NotifyServiceクラスは次の通りです。

```
class NotifyService(JsonService):
    service_type = "notify"
    service_name = "adapter"

    def handle_message(self, msg):
        """
        Interface.
        """

    def notify(self, request, _):
        msg = request["notify_message"]
        logging.info("notify message:{}".format(msg))
        return self.handle_message(msg)
```

デフォルトのサービス名はadapterです。プロジェクトで複数の通知サービスが必要な場合は、子クラスでservice_nameを書き換えて、異なるサービスを区別できます。クラス関数の説明を次の表に示します。

クラス関数	説明
handle_message()	インターフェース関数で、この関数にロジックを実装できるように子クラスはこの関数を書き換えることができます。
notify()	メッセージの解析を提供します（通常、子クラスは書き換える必要はありません）。

VisionResultSelectedAtServiceクラス

VisionResultSelectedAtServiceクラスは次の通りです。

```
class VisionResultSelectedAtService(JsonService):
    service_type = "vision_watcher"
    service_name = "vision_watcher_adapter"

    def __init__(self):
        self.poses = None

    def poses_found(self, result):
```

```

"""
    Interface.
"""

def posesFound(self, request, _):
    logging.info("{} result:{}".format(jk.mech_vision, request))
    self.poses_found(request)

def poses_planned(self, result):
    """
        Interface.
    """

def posesPlanned(self, request, _):
    logging.info("Plan result:{}".format(request))
    self.poses_planned(request)

def multiPickCombination(self, request, _):
    logging.info("multiPickCombination:{}".format(request))

```

デフォルトのサービスタイプはvision_watcherで、タイプを変更することはできません。デフォルトの名前はvision_watcher_adapterです。プロジェクトで複数のvision_watcherサービスが必要な場合は、子クラスでservice_nameを書き換えて、異なるサービスを区別できます。クラス関数の説明を次の表に示します。

クラス関数	説明
poses_found()	インターフェース関数で、パラメータはMech-Visionからの認識結果です（この関数にロジックを実装できるように子クラスはこの関数を書き換えることができます）。
posesFound()	Mech-Visionによって認識されたメッセージを解析します（通常、子クラスを書き換える必要はありません）。
poses_planned()	インターフェース関数で、パラメータはMech-Viz計画によって選択されたビジョンポイントです。
posesPlanned()	Mech-Vizからの計画メッセージの解析を提供します。

RobotServiceクラス

RobotServiceクラスは次の通りです。

```

class RobotService(JsonService):
    service_type = "robot"
    service_name = "robot"
    jps = [0, 0, 0, 0, 0, 0]

```



```
pose = [0, 0, 0, 1, 0, 0, 0]

def getJ(self, * _):
    return {"joint_positions": self.jps}

def setJ(self, jps):
    logging.info("setJ:{}".format(jps))
    self.jps = jps

def getL(self, * _):
    return {"tcp_pose": self.pose}

def getFL(self, * _):
    return {"flange_pose": self.pose}

def setL(self, pose):
    logging.info("setL:{}".format(pose))
    self.pose = pose

def moveXs(self, params, _):
    pass

def stop(self, * _):
    pass

def setTcp(self, * _):
    pass

def setDigitalOut(self, params, _):
    pass

def getDigitalIn(self, * _):
    pass

def switchPauseContinue(self, * _):
    pass
```

デフォルトのサービスタイプはrobotで、タイプを変更することはできません。デフォルトの名前はrobotで、子クラスに対応するロボット名に変更する必要があります。また、Mech-Vizの実行中に1つの位置姿勢を固定するために、子クラスにjpsまたはpose値を設定する必要があります。この位置姿勢が経路全体でシーンと衝突しないようにすることに注意してください。クラス関数の説明を次の表に示します。

クラス関数	説明
getJ()	Mech-Viz/Mech-Visionへ関節角度を返します。

クラス関数	説明
setJ()	外部で関節角度を設定します（ラジアン単位）。
getL()	Mech-Viz/Mech-Visionへツール位置姿勢を返します。
getFL()	Mech-Viz/Mech-Visionへフランジ位置姿勢を返します。
setL()	外部でフランジ位置姿勢（四元数の形式、メートル単位）を設定します。
moveXs()	Mech-Vizが経路を計画した後にこの関数が呼び出され、パラメータには移動点のプロパティが含まれます（Mech-Vizプロジェクトに「DIをチェック」、分岐関連のステップなど、事前計画を中断するステップがある場合は、Mech-Vizはこの関数を複数回呼び出します）。
stop()	ロボットを停止します（通常は使用されません）。
setTcp()	TCPを設定します（通常は使用されません）。
setDigitalOut()	DOを設定します（通常は使用されません）。
getDigitalIn()	DIを取得します（通常は使用されません）。
switchPauseContinue()	ロボットを一時停止/再開します（通常は使用されません）。

OuterMoveServiceクラス

OuterMoveServiceクラスは次の通りです。

```
class OuterMoveService(JsonService):
    service_type = "outer_move"
    service_name = "outer_move"
    move_target_type = TCP_POSE
    velocity = 0.25
    acceleration = 0.25
    blend_radius = 0.05
    motion_type = MOVEJ
    is_tcp_pose = False
    pick_or_place = 0

    def __init__(self):
        self.targets = []

    def gather_targets(self, di, jps, flange_pose):
        """
        Interface.
```

```

        Please add targets to `self.targets` here if needed.
        """

    def add_target(self, move_target_type, target):
        self.targets.append({"move_target_type": move_target_type, "target": target})

    def getMoveTargets(self, params, *_):
        """
        @return: targets(move_target_type 0:jps, 1:tcp_pose, 2:obj_pose)
                velocity(default 0.25)
                acceleration(default 0.25)
                blend_radius(default 0.05)
                motion_type(default moveJ 'J':moveJ, 'L':moveL)
                is_tcp_pose(default False)
        """
        di = params["di"]
        jps = params["joint_positions"]
        flange_pose = params["pose"]
        logging.info("getMoveTargets: di={}, jps={}, flange_pose={}".format(di, jps,
        flange_pose))

        self.gather_targets(di, jps, flange_pose)
        targets = self.targets[:]
        self.targets.clear()
        logging.info("Targets: {}".format(targets))
        return {"targets": targets, "velocity": self.velocity, "acceleration": self
        .acceleration, "blend_radius": self.blend_radius,
                "motion_type": self.motion_type, "is_tcp_pose": self.is_tcp_pose,
        "pick_or_place": self.pick_or_place}

```

デフォルトのサービスタイプと名前はouter_moveです。プロジェクトで複数のouter_moveサービスが必要な場合は、子クラスでservice_nameを書き換えて、異なるサービスを区別できます。クラス関数の説明を次の表に示します。

クラス関数	説明
move_target_type()	移動点のタイプ (0:jps、1:tcp_pose、2:obj_pose)。
velocity()	移動点の速度（初期値は0.25）。
acceleration()	移動点の加速度（初期値は0.25）。
blend_radius()	移動点のブレンド半径（初期値は0.05m）。
motion_type()	移動点の運動タイプ（'J':moveJ、'L':moveL）。
is_tcp_pose()	移動点はTCPであるかどうか。

クラス関数	説明
gather_targets()	インターフェース関数で、すべての移動点を取得します。パラメータは現時点でのロボットの関節角度、フランチ位置姿勢およびDI値で、必要に応じて子クラスを変更できます。
add_target()	単一の移動点を追加します（子クラスでこの関数を呼び出して移動点を追加できます）。
getMoveTargets()	Mech-Vizは外部移動ステップに実行すると、この関数が呼び出されます（パラメータには現時点でのロボットの関節角度、フランチ位置姿勢、およびDI値が含まれます）。

サービスを登録

上記4つのカテゴリに対応するサービスは、登録後のみ使用できます。登録サービス関数は次の通りです。

```
def register_service(hub_caller, service, other_info=None):
    server, port = start_server(service)
    if service.service_type == "robot":
        other_info["from_adapter"] = True
        other_info["simulate"] = False
    hub_caller.register_service(service.service_type, service.service_name, port, other_info)
    return server, port
```

6.5.3.3. Adapter utilパッケージ

Adapter utilパッケージは、Mech-Centerソフトウェアのインストールパスの `/src/util` フォルダに格納されています。このパッケージには、さまざまなモジュールが含まれており、一般的な関数を提供しています。これらの関数は、Adapterの開発者がプログラムを効率的に作成するのに役立ちます。プログラミングの過程で、まずutilパッケージに必要な機能が既に実装されているかを確認します。もし実装されている場合は、それを直接使用することができます。もし実装されていない場合で、かつ一般的な機能である場合は、それを小さな関数として抽象化し、utilパッケージに追加することができます。

以下では、各モジュールについて簡単な紹介をします。

databaseモジュール

databaseモジュールは、データベースの操作を提供します。Mech-Centerは実行時にデフォルトでmechmind.dbというデータベースファイルを作成します。このファイルは実行中のログを保存するために使用されます。databaseモジュールには、SQL文の実行やレコードの一部またはすべてをクエリするための関数が提供されています。

json_keysモジュール

json_keysモジュールには、Mech-Centerで使用されるJSONキー/値文字列が保存されています。他のモジュールでは、直接インポートして使用することができます。

message_boxモジュール

message_boxモジュールは、ポップアッププロンプトの機能を提供します。ポップアッププロンプトのタイプにはinformation（一般情報）、warning（ワーニング）およびcritical（重大エラー）が含まれます。

timestampモジュール

timestampモジュールは、現在のタイムスタンプを返す関数を提供します。

transformsモジュール

transformsモジュールは、オイラー角から四元数に変換する、四元数からオイラー角に変換する、位置姿勢乗算、対象物の位置姿勢からTCPに変換する、TCPから対象物の位置姿勢に変換する、対象物回転計算などの関数を提供します。サードパーティのライブラリtransforms3dもオイラー角から四元数に変換する、四元数からオイラー角に変換する機能を提供していますが、実際の使用では、transforms3dによって変換された値が間違っている場合があります。実際の計算では、最初にtransforms3dライブラリを使用できますが、結果が正しくない場合は、transformsモジュールが提供するカスタマイズの変換関数を使用できます。

util_fileモジュール

util_fileモジュールは、一般的に使用されるjsonファイルなど、読み取りおよび書き込み関数を提供します。

timerモジュール

timerモジュールは、便利なTimerクラスを提供します。タイミング関数が必要な場合は、Timerオブジェクトを生成し、コールバック関数を渡し、start()を呼び出すことができます。使用後、Timerオブジェクトを破棄する必要はありません。プログラムの終了時に自動的に破棄されます。

poseモジュール

poseモジュールは、Mech-Vizの位置姿勢を同じ形式で表すためのクラスを提供しており、並進（メートル単位）と回転（四元数形式）を含んでいます。このクラスでは、逆行列を取得したり、行列同士の乗算を行ったりすることができます。また、listからの変換やlistへの変換も可能です。さらに、poseモジュールでは、位置姿勢に関連するいくつかの単位変換関数も提供されています。それには、ミリメートルからメートルへの変換、メートルからミリメートルへの変換、ラジアンから度への変換、度からラジアンへの変換、四元数からオイラー角への変換、オイラー角から四元数への変換が含まれています。

6.5.3.4. インターフェースの取得

現在のMech-Vizプロジェクトに使用されているステップを取得

現在のMech-Vizプロジェクトに使用されているステップを取得するための関数は、次のとおりです。

```
def get_viz_task_names(self, msg={}, timeout=None):
    result = self.call_viz("getAllTaskNames", msg, timeout)
    logging.info("Property result: {}".format(json.loads(result)))
    return result
```

get_viz_task_names()を呼び出した後、json形式の文字列を返します。これは、取得したすべてのステップを示します。

Mech-VizまたはMech-Visionプロジェクトのパラメータを取得

Mech-VizまたはMech-Visionプロジェクトのパラメータを取得するための関数は、次のとおりです。

```
def get_property_info(self, msg={}, get_viz=True, timeout=None):
    result = (self.call_viz if get_viz else self.call_vision)("getPropertyInfo", msg, timeout)
    logging.info("{0} property result: {1}".format("Viz" if get_viz else "Vision", json.loads(result)))
    return result
```

msgパラメータに「type」を指定せずに呼び出した場合、すべてのパラメータを取得することを意味します。指定すると、対応するパラメータのみが取得されます。例えば、get_property_info(msg={"type": "move"})を呼び出した後、json形式の文字列を返します。これは、取得された移動ステップパラメータを示します。

6.5.3.5. Mech-Visionインターフェース

本節では、Mech-Visionを使用するインターフェースについて説明します。次のインターフェースが含まれています。

ビジョン目標点の取得

以下に示すように、adapter.pyのfind_vision_pose()関数でMech-Visionのビジョン結果を取得します。

```
def find_vision_pose(self, project_name=None, timeout=default_vision_timeout):
    vision_result = self.call_vision("findPoses", project_name=project_name, timeout=timeout)
    logging.info("Find vision result: {}".format(vision_result))
    return vision_result
```

Mech-Visionのビジョンポイントは通常、四元数の形式で表示される対象物の位置姿勢（obj_pose）で出力されます。ツール位置姿勢を出力することもできますが、Mech-Visionプロジェクトで変換する必要があります。

サンプル

Adapterは対象物の位置姿勢をツール位置姿勢（tcp_pose）に変換する関数を構築して、ロボット側に送信します。場合によっては四元数をオイラー角、ラジアンを度などに変換する必要があります（変換することは、ロボット側と一致するかどうかによって決定されます）。さらに、AdapterはMech-Visionによって出力された次のようなビジョンポイントを確認することもできます。

```
def check_vision_result(self, vision_result):
    if vision_result["noCloudInRoi"]:
        # Determine whether it is an empty
        bin
        logging.info("Layer has no objects")
        self.send(pack('>2B6i', CODE_NO_CLOUD, vision_num, *EMPTY_PLACEHOLDER))
        return
    poses = vision_result.get("poses", [])
    if len(poses) == 0:
        # Get pose
        # Determine whether there is a vision
        point
        logging.warning("No pose from vision")
        self.send(pack('>2B6i', CODE_NO_POSE, vision_num, *EMPTY_PLACEHOLDER))
        return
    self.send(pack_pose(poses[0], vision_num))
    # Send after format conversion
```

以下は、find_vision_pose()からvision_resultを取得するための呼び出し例です。

```
self.check_vision_result(json.loads(self.find_vision_pose().decode()))
```

vision_resultを関数に渡す際に、プロジェクトにROIが設定された場合、まず空の箱かどうかを確認し、それからビジョンポイントを取得します。ビジョンポイントが正常であれば、その結果を変換関数（pack_pose()）に渡して送信します。

ステップパラメータの設定

通常、Mech-Visionステップパラメータを動的に設定する場合は、adapter.py内のset_step_property()を呼び出すだけで十分です。

```
def set_step_property(self, msg, project_name=None, timeout=None):
    return self.call_vision("setStepProperties", msg, project_name, timeout)
```

msgは特定のステップ名と設定が必要なパラメータを決定します。

サンプル

Mech-Visionのマッチングモデルをワークの種類に応じて動的に設定する必要がある場合は、次の関数を作成してmsgを設定できます。

```
def _step_matching_model_cell(step_name, model_type):
    msg = {"name": step_name,
          "values":
            {"modelFile": model_type["ply"],
             "pickPointFilePath": model_type["json"]}}
    return msg
```

ここで、step_nameは設定する必要があるMech-Visionステップの名前であり、model_typeはワークの種類に対応するファイルパスです。「ply」と「json」を使用して、それぞれ「.ply」と「.json」で終わるモデルファイルのパスを取得し、Mech-Visionの対応するステップパラメータにそれぞれ入力します。

step_nameが「Local Matching」の場合、呼び出し例は以下のようになります。

```
msg = _step_matching_model_cell("Local Matching", model_type)
self.set_step_property(msg)
```

ステップパラメータの読み取り

Mech-Visionの特定のステップのパラメータを取得するには、adapter.py内のread_step_property()を呼び出すことができます。

```
def read_step_property(self, msg):
    result = self.call_vision("readStepProperties", msg)
    logging.info("Property result: {}".format(result))
    return result
```

msgは取得するステップ名とパラメータ値を決定します。関数を作成してmsgを書き換えることができます。

サンプル

カメラIPを取得する場合、呼び出し例は以下のようになります。

```
def read_camera_property(self):
    msg = {"type": "Camera",
          "properties": ["MechEye"]}
    property_results = json.loads(self.read_step_property(msg).decode())
    camera_ip = property_results["MechEye"]["NetCamIp"]
```

Mech-Visionプロジェクトにカメラが1つしかない場合、タイプ（type）のみに基づいてステッ

プを見つけることができます。Mech-Visionプロジェクトに同じステップが複数あり、特定のステップパラメータのみを取得または設定する場合は、名前（name）でステップを見つけることができます。以下に示すように、`read_step_property()`を呼び出してjson形式に変換することにより、カメラのすべてのパラメータ値（json形式）を取得します。

```
Property result:
{
  "MechEye": {
    "NetCamIp": "127.0.0.1",
    "TimeOut": "10",
    "configGroup": "",
  }
}
```

本例では、特定のパラメータフィールド（「MechEye」および「NetCamIp」）に従って、カメラIP（127.0.0.1）を取得します。

パラメータレシピの切替

Mech-Visionを使用する際に、ワークフローが同じであるが、一部のパラメータが異なる場合、パラメータレシピを使用することで、異なるプロジェクトに対応するパラメータを切り替えることができます。この機能は、`adapter.py`の`select_parameter_group()`の呼び出しによって実現されます。

```
def select_parameter_group(self, project_name, group_index, timeout=None):
    msg = {"parameter_group_idx": group_index}
    result = self.call_vision("selectParameterGroup", msg, project_name, timeout)
    logging.info("selectParameterGroup result: {}".format(result))
    return result
```

`project_name`はMech-Visionのプロジェクト名で、`group_index`はレシピ番号です。

サンプル

パラメータレシピの切替が必要な場合は、次のサンプルコードを使用し、`select_parameter_group()`関数を呼び出し、例外を処理します。

```
try:
    result = self.select_parameter_group(self.vision_project_name, model_code-1)
    if result:
        result = result.decode()
        if result.startswith("CV-E0401"):
            return -1
        elif result.startswith("CV-E0403"):
            return -1
```

```

        raise RuntimeError(result)
    except Exception as e:
        logging.exception('Exception when switch model: {}'.format(e))
    return -1
return 0

```

self.vision_project_nameは受信されたMech-Visionのプロジェクト名で、model_code-1は受信されたレシピ番号です。

6.5.3.6. Mech-Vizインターフェース

本節では、Mech-Vizを使用するインターフェースについて説明します。次のインターフェースが含まれています。

Mech-Vizプロジェクトの起動

Mech-Vizを起動する関数は、adapter.pyファイルのAdapterクラスで定義されているため、コード内でstart_viz()を直接呼び出すことができます。また、プロジェクトの要件に応じ、self.before_start_viz()とself.after_start_viz()を書き換えることで、Mech-Viz起動前後のカスタマイズの動作を実現できます。

関数定義

```

def start_viz(self, in_new_thread=True, timeout=None):
    if not self.is_viz_registered():
        logging.error("{} has not registered in {}".format(jk.mech_viz, jk.mech_center))
        self.code_signal.emit(ERROR, VIZ_NOT_REGISTERED)
        self.viz_finished_signal.emit(True)
        self.viz_not_registerd()
        return False
    if self.is_viz_in_running():
        logging.info("{} is already running.".format(jk.mech_viz))
        self.code_signal.emit(WARNING, VIZ_IS_RUNNING)
        self.viz_finished_signal.emit(False)
        self.viz_is_running()
        return False
    self._read_viz_settings()
    if not self.viz_project_dir:
        self.msg_signal.emit(ERROR, _translate("messages", "The project of {0} is not
registered. Please make sure Autoload Project is selected in {0}.").format(jk.mech_viz))
        self.viz_finished_signal.emit(True)
        return False
    msg = {"simulate": self.is_simulate, "project_dir": self.viz_project_dir}
    if self.is_keep_viz_state:
        msg["keep_exec_state"] = self.is_keep_viz_state
    if self.is_save_executor_data:
        msg["save_executor_data"] = self.is_save_executor_data

```

```
self.before_start_viz()
self.viz_finished_signal.emit(False)
if in_new_thread:
    threading.Thread(target=self.wait_viz_result, args=(msg, timeout)).start()
else:
    self.wait_viz_result(msg, timeout)
self.after_start_viz()
return True
```

start_viz()は、デフォルトで新しいスレッドでMech-Vizが実行を終了するのを待ちます。この目的は、Mech-Vizの起動以外の他の操作に影響を与えないようにすることです。

次にステップパラメータを動的に設定することを例として、self.before_start_viz()を書き換える方法を示します。

```
def before_start_viz(self):
    self.set_move_offset(x, y, z)
```

Mech-Vizを起動する前に、読み取んだデータに従って、ある移動点のx、y、z方向のオフセットを設定します。

Mech-Vizプロジェクトの停止

Mech-Vizを停止する関数は、adapter.pyファイルのAdapterクラスで定義されているため、コード内でstop_viz()を直接呼び出すことができます。

関数定義

```
def stop_viz(self, timeout=None):
    if not self.is_viz_registered():
        self.code_signal.emit(WARNING, VIZ_NOT_REGISTERED)
        return False
    self.call_viz("stop", timeout=timeout)
    self.code_signal.emit(INFO, VIZ_STOP_OK)
    return True
```

Mech-Vizの一時停止と再開

Mech-Vizを一時停止・再開する関数は、adapter.pyファイルのAdapterクラスで定義されています。Mech-Vizソフトウェアの停止ボタンと同じように機能し、シミュレーションのみに使用できます。

関数定義

```
def pause_viz(self, msg, timeout=None):
```

```
if not self.is_viz_registered():
    self.code_signal.emit(WARNING, ADAPTER_CANCEL_PAUSE)
    return
self.call_viz("switchPauseContinue", msg, timeout)
self.code_signal.emit(INFO, ADAPTER_PAUSE_VIZ if msg.get(
    "to_pause") else ADAPTER_CONTINUE_VIZ)
```

ステップパラメータの設定

通常、Adapterクラスのset_task_property()を呼び出して、Mech-Vizでステップパラメータを動的に設定します。

関数定義

```
def set_task_property(self, msg, timeout=None):
    return self.call_viz("setTaskProperties", msg, timeout)
```

msgは様々なステップに対して様々なパラメータを設定することを決定します。

移動

Mech-Vizの実行中に、移動ステップでX、Y、Zのオフセットを微調整する必要がある場合があります。Mech-Vizを制御するメインプログラムには、次の関数を記述できます。

サンプル

```
def set_move_offset(self, name, x_offset, y_offset, z_offset):
    msg = {"name": name,
          "values": {"xOffset": x_offset / UNIT_PER_METER,
                    "yOffset": y_offset / UNIT_PER_METER,
                    "zOffset": z_offset / UNIT_PER_METER}}
    self.set_task_property(msg)
```

nameは移動ステップの名前を表します。UNIT_PER_METER=1000。通常、x_offset、y_offsetおよびz_offsetのデータ単位はmmで、Mech-Vizでのデータ単位はmであるため、単位変換にはUNIT_PER_METERを使用します。

以下の方法でset_move_offset()関数を呼び出すと、Mech-Vizの対応する「移動_1」ステップのX、Y、Zオフセットがそれに応じて変更されます。

```
self.set_move_offset("move_1", 100, 200, 300)
```

リストによる移動/グリッドによる移動

通常、事前にMech-Vizで編集する必要があります。その後、Adapterがロジックに従ってインデ

ックスを変更します。使用方法は、[移動](#)や [パレタイジング](#) ステップと同じです。

外部移動

制御を計画するために複数の外部目標位置姿勢をMech-Vizに送信する必要がある場合、それらは外部移動ステップによって実現できます。外部移動ステップは、JPs、TCP、対象物の位置姿勢の設定に対応しています。使用方法は次の通りです。

サンプル

```
class CustomOuterMoveService(OuterMoveService):
    def gather_targets(self, di, jps, flange_pose):
        self.add_target(self.move_target_type, [0.189430,-0.455540,0.529460,-0.079367,0.294292,-0.952178,0.021236])
```

Mech-Vizは外部移動ステップに実行すると、getMoveTargets()が呼び出されます。様々な外部移動ステップは、サービス名によって区別されます。

```
def _register_service(self):
    self.outer_move_service = CustomOuterMoveService()
    self._outer_move_server, port = register_service(outer_move_service, port)
```



外部移動ステップの前に、把持することを示すステップが必要です。そうしないと、「- 対象物が把持されてない場合、対象物の位置姿勢は無効です。」というエラーが報告されます。

パレタイジング

Mech-Vizの実行中に、様々なパレタイジングステップに応じて様々なパラメータを設定する必要がある場合があります。変更が必要なステップは、パレタイジングステップの名前で見つけることができます。Mech-Vizのプロジェクト編集エリアでステップを選択すると、パラメータバーに表示されるすべてのパラメータを変更できます。

サンプル

例えば、カスタマイズのパレットパターンの場合、一般的に変更が必要なパラメータは **開始インデックス** および **ファイル名**（**動的に読み込む** にチェックを入れる場合のみに表示）です。したがって、メインプログラムで次の関数を定義できます。

```
def set_stack_pallet(self, name, startIndex, fileName):
    msg = {
        "name": name,
        "values": {
            "startIndex": startIndex,
            "fileName": fileName,
        }
    }
```

```
}
self.set_task_property(msg)
```

その中で、「startIndex」は **開始インデックス** を示し、「fileName」は **ファイル名** に示します。「curIndex」および「filename」のパラメータ名は、Mech-Vizで定義されています。

以下の方法を使用してset_stack_pallet()関数を呼び出します。

```
self.set_stack_pallet("common_pallet_1", 2, "re.json")
```

事前計画パレットパターンの場合、通常、変更が必要なパラメータは **開始インデックス**、**パレットパターン**、**箱の長さ**、**箱の幅**、**箱の高さ**、**行の数**、**列の数** および **段数** です。したがって、メインプログラムで次の関数を定義できます。

```
def set_stack_pallet(self, name, startIndex, stack_type):
    pallet_info = self.box_data_info[stack_type]
    """
        pallet_info: Length(mm),Width(mm),Height(mm),pallet type,rows,columns,layers
    """
    msg = {
        "function": "setTaskProperties",
        "name": name,
        "values": {
            "startIndex": startIndex,
            "palletType": pallet_info[3],
            "cartonLength": pallet_info[0] / UNIT_PER_METER,
            "cartonWidth": pallet_info[1] / UNIT_PER_METER,
            "cartonHeight": pallet_info[2] / UNIT_PER_METER,
            "cylinderRows": pallet_info[4],
            "cylinderCols": pallet_info[5],
            "layerNum": pallet_info[6]
        }
    }
    self.set_task_property(msg)
```

設定するパラメータが多いため、通常はパラメータをexcelファイルに書き込み、excelファイルのデータを読み取って、self.box_data_infoに記録します。これは、後でstack_typeの値によってインデックス化できます。「startIndex」、「palletType」および「cartonLength」などの名パラメータはMech-Vizで固定されています。

カスタマイズのパレットパターン クラスと **事前計画パレットパターン** クラスのmsg値を比較すると、違いが「values」の値にあることがわかります。ステップパラメータを設定するには、対応するパラメータ名と値を、「values」で追加する必要があります。他のパレットパターンのステップパラメータ設定については、上記の例をご参照ください。

メッセージによって異なる分岐を実行

Mech-Vizは分岐ステップに実行すると、外部信号（Adapter）が出口を指定するのを待ちます。分岐ステップの場合、分岐制御を行うために次の関数を定義できます。

サンプル

```
def set_branch(self, name, area):
    time.sleep(1) # The delay of 1s here is to wait for the {product-viz} executor to fully
    start
    try:
        info = {"out_port": area, "other_info": []}
        msg = {"name": name,
              "values": {"info": json.dumps(info)}}
        self.set_task_property(msg)
    except Exception as e:
        logging.exception(e)
```

その中で、nameは分岐ステップの名前を示し、areaは出口を示します。出口には、左から右に0、1、2、…の番号が付けられています。このステップが左端の出口から実行する場合、area=0になります。Mech-Vizを起動せずに分岐ステップを直接呼び出すと、Mech-Vizは「エグゼキュータはありません」というエラーが報告されます。

カウンター

カウンターステップを使用する場合、通常、計数の合计数と現在の計数を設定する必要があります。このステップを定義して設定するコードは次の通りです。

サンプル

```
def set_counter_property(self, name, count, curCount):
    msg = {"name": name,
          "values": {"count": count, "currentCount": curCount}}
    self.set_task_property(msg)
```

この関数の呼び出し例を以下に示します。

```
self.set_counter_property("counter_1", 5, self.success_stack_num)
```

self.success_stack_numは正常にパレタイジングされた対象物の数を示します。パレタイジング中に箱が下した場合、人間の介入により、Mech-Vizは停止します。この時点で、Mech-Vizの「カウンター_1」というカウンターステップは、「currentCount」の値を保存しません。Mech-Vizを再起動した後、カウンターステップの現在の計数は、self.success_stack_numによってリセットできます。

ステップパラメータの読み取り

Mech-Vizの実行中に、ステップパラメータを読み取る必要がある場合は、メインプログラムで次の関数を定義できます。

サンプル

```
def read_move_pallet(self, name):  
    msg = {"name": name,  
           "properties": ["xOffset", "yOffset", "zOffset", ]}  
    return read_task_property(msg)
```

「name」によって読み取るステップ名を見つけることができます。「properties」の後のリストの値は、必要に応じてパラメータを追加または削除できます。本例では、「xOffset」、「yOffset」、および「zOffset」の値が読み取られるので、3つの値を「properties」に追加します。呼び出す方法を以下に示します。

```
self.read_move_pallet("move_3")
```

呼び出した後、次の結果が得られます。

```
{'zOffset': -0.23, 'xOffset': -0.12, 'yOffset': -0.15}
```

また、Mech-Vizのステップから得られるパラメータ値は、計画進行時の値であり、実際の実行時の値ではないことにご注意ください。計画は、実行前に実行します（つまり、Mech-Vizは以後の動作を実際実行より前に計画します）。ここで、パレタイジングステップの開始インデックスを例として説明します。メインプログラムで以下のような関数を定義できます。

```
def read_pallet_current_index(self, name):  
    msg = {"name": name,  
           "properties": ["curIndex"]}  
    return read_task_property(msg)
```

開始インデックスの値を呼び出す方法は以下の通りです。

```
self.read_pallet_current_index("common_pallet_1")
```

呼び出すと以下の結果が得られます。

```
{'curIndex': 5}
```

ここで表示される「5」は、5回の計画が実行されたことを意味します。例えば、箱配置のプロジェクトでは、5は計画実行中に5つの箱が配置されましたが、実際にロボットは5箱より少ない箱しか配置しない可能性があります。したがって、このパラメータは実際実行時の値ではなく、計画された値を示します。

TCPの設定

TCPの設定は、Mech-Vizのロボットハンドリストで対応するインデックスを指定するだけです。Mech-Vizのロボットハンドのインデックスは、上から下まで0、1、2、3、...です。範囲を超えないように注意してください。TCP関数の設定は次の通りです。

サンプル

```
def set_tcp(self, index):
    msg = {"function": "setTcp", "index": index}
    self.call("executor", msg)
```

実行中のグローバル速度を設定

Mech-Vizの実行中に、ロボットの実行速度を動的に調整する必要がある場合は、メインプログラムに次の関数を定義できます。

サンプル

```
def set_vel(self, vel_scale):
    msg = {"function": "setConfig",
          "velScale": vel_scale / 100, "accScale": vel_scale / 100}
    self.call("executor", msg)
```

速度が80%に設定されている場合、関数は次のように呼び出すことができます。

```
self.set_vel(80)
```



この関数は、Mech-Vizの起動後に呼び出す必要があります。そうしないと、エラーが報告されます。つまり、このモジュールの呼び出し条件は分岐ステップと一致しています。したがって、分岐ステップでset_vel()を呼び出し、新しいスレッドでset_vel()を呼び出さないようにします。例を以下に示します。

```
def set_branch(self, name, area):
    time.sleep(1)
    if self.box_data_info[int(self.pallet_info)][7] <= 10:
        self.set_vel(100)
    else:
        self.set_vel(80)
    try:
```

```

info = {"out_port": area, "other_info": []}
msg = {"function": "setTaskProperties",
      "name": name,
      "values": {"info": json.dumps(info)}}
self.set_task("executor", msg)
except Exception as e:
    logging.exception(e)

```

点群衝突検出のパラメータを設定

点群衝突パラメータを設定します。Mech-Vizに対応するインターフェースはsetConfig()であり、これは実行時にグローバル速度を設定するインターフェースと同じですが、設定する情報が異なります。例を以下に示します。

サンプル

```

msg = {}
msg["function"] = "setConfig"
msg["check_pcl_collision"] = True
msg["collided_point_thre"] = 5000
msg["collide_area_thre"] = 20
msg["pcl_resolution_mm"] = 2
self.call("executor", msg)

```

Mech-Vizの返された値

Mech-Vizの返された値を次の表に示します。

返された値	意味
Finished	正常に実行しました（Mech-Vizプロジェクトに「ビジョン処理による移動」ステップの右の分岐がステップにつなぐ場合は正常に戻ります）
Command Stop	停止ボタンを押すか、stop_viz()関数を呼び出します
No targets	ビジョンポイントはありません（Mech-Visionからのビジョン結果は空であることを意味します）
No proper vision poses	ビジョンポイントに到達できません（ロボット自体が現在認識された対象物に到達できない、またはそれと衝突したことを意味します）
PlanFail	Mech-Vizの動作計画に失敗しました
SceneCollision	衝突が発生しました

Mech-Vizの返された値に応じて、Adapter基本クラスは対応するインターフェース関数を提供します。

6.5.3.7. RobotServiceインターフェース

本節では、RobotServiceを使用するインターフェースについて説明します。次のインターフェースが含まれています。

モックRobotServerサービス

モックRobotServerサービスは通常、Vizティーチングの場合に使用されますが、ロボットをRobotServerを介して制御することはできず、その代わりにRobotServerの役割を模擬するためにRobotServiceを作成します。RobotServiceがMech-Centerに登録された後、Mech-Vizはロボット実機と同様の方法で「RobotService」と情報のやり取りを行います。

モックRobotServerサービスは、以下の場合に使用されます。

- Mech-Visionに計算されるロボットの関節角度を取得する場合（Eye In Hand）。
- ロボット実機から位置姿勢を取得し、それをRobotServiceサービスに転送する場合。

サンプル

以下は、Adapterの子クラス内での呼び出し方法の例です。

```
def _register_service(self):
    """
    register_service
    :return:
    """
    if self.robot_service:
        return

    self.robot_service = RobotService(self)
    other_info = {'robot_type': self.robot_service.service_name}
    self.server, _ = register_service(self.hub_caller, self.robot_service, other_info)

    self.robot_service.setJ([0.0, 0.0, 0.0, 0.0, 0.0, 0.0])
```

以下は、RobotServiceクラスの例です。

```
class RobotService(JsonService):
    service_type = "robot"
    service_name = "robot"
    jps = [0, 0, 0, 0, 0, 0]
    pose = [0, 0, 0, 1, 0, 0, 0]

    def getJ(self, *_):
        return {"joint_positions": self.jps}
```

```
def setJ(self, jps):
    logging.info("setJ:{}".format(jps))
    self.jps = jps

def getL(self, *_):
    return {"tcp_pose": self.pose}

def getFL(self, *_):
    return {"flange_pose": self.pose}

def setL(self, pose):
    logging.info("setL:{}".format(pose))
    self.pose = pose

def moveXs(self, params, _):
    pass

def stop(self, *_):
    pass

def setTcp(self, *_):
    pass

def setDigitalOut(self, params, _):
    pass

def getDigitalIn(self, *_):
    pass

def switchPauseContinue(self, *_):
    pass
```

getJ

getJ()は、Mech-VizおよびMech-Visionで現在の関節角度を取得するために使用されます。通常の場合、関節角度を設定するためにはsetJ()を使用し、その後getJ()を呼び出すことが一般的です。以下に例を示します。

サンプル

```
def getJ(self, *_):
    pose = {"joint_positions": self._jps}
    return pose
```

1. Eye In Hand：ロボットから送信された関節角度をsetJ()に書き込みます。

```
def setJ(self, jps):
    assert len(jps) == 6
    for i in range(6):
        jps[i] = deg2rad(float(jps[i]))
    self._jps = jps
    logging.info("SetJ:{}".format(self._jps))
```

その中、jpsはロボットから送信された関節角度データで、getJ()によって呼び出されるようにself._jpsに割り当てられます。関節角度データを取得する際に、適切な単位変換を行ってラジアン形式のデータを取得する必要があります。

2. Eye to Handの場合、現在のロボットの関節角度に基づいて目標点を設定する必要はありませんが、ロボット実機の状態をシミュレーションする場合、安全な目標点を設定する必要があります（通常は初期位置）。そうしないと、ランダムな値が割り当てられてしまい、エラーが発生しやすくなります。

```
def getJ(self, * _):
    return {"joint_positions": [1.246689, -0.525868, -0.789761, -1.330814, 0.922581, 4.364021]}
```

getFL

getFL()は、ロボットの画像撮影時におけるフランジ位置姿勢を提供するために使用されます。Eye In Hand方式では、フランジとカメラの位置関係をキャリブレーションしますが、最終的に使用するデータは基準座標系であるため、画像撮影時のロボットのフランジ位置姿勢が必要です。

```
def getFL(self, * _):
    return {"flange_pose": self.pose}
```

Eye In Hand方式でビジョンポイントを提供する際には、以下の点に注意する必要があります。

1. ロボットが画像撮影時の関節角度（JPs）を返す場合、RobotServiceでsetJ()を直接呼び出すことができます（ラジアン単位）。この関数は[]を返します。
2. ロボットがフランジ位置姿勢を返す場合、次の操作を行う必要があります。
 - Mech-Visionプロジェクトの「extri_param.json」外部パラメータファイルの「is_eye_in_hand」がtrueに設定されていることを確認します。
 - RobotServiceのsetFL()を呼び出します（メートル単位、四元数形式の位置姿勢）。

moveXs

Adapterによって作成されたRobotServiceサービスは、moveXs()を使用して、Mech-Vizによって計画された経路点データを受け取ります。以下に例を示します。

関数定義

```
def moveXs(self, params, _):
    with self.lock:
        for move in params["moves"]:
            self.targets.append(move)
    return {"finished": True}
```

その中で、paramsはMech-Vizプロジェクトのすべてのパラメータが渡されており、params["moves"]を使用すると、ビジョン移動や相対移動を含むすべての移動点の位置姿勢情報を取得できます。位置姿勢情報はデフォルトで関節角度の形式で返されます。その位置姿勢をself.targetsに渡すことで、後続の呼び出しで使えるようになります。

サンプル

通常、この関数は通知と共に使用されます。Adapterが通知メッセージを受信した場合、self.targetsを関数に送信して変換し、パッケージ化してロボットに送信します。以下に例を示します。

```
def notify(self, request, _):
    msg = request["notify_message"]
    logging.info("{} notify message:{}".format(self.service_name, msg))
    if msg == "started":
        with self.lock:
            self.move_targets.clear()
    elif msg == "finished":
        with self.lock:
            targets = self.move_targets[:]
            self.move_targets.clear()
            self.send_moves(targets)
```

通知メッセージが「started」の場合、目標点リストをクリアします（Mech-Vizのエラー中断により、前後の2つの移動点が重なってしまうのを防ぐため）。通知メッセージが「finished」の場合、目標点リストをpack_move関数に渡します（データ統合送信のため）。

```
def pack_move(self, move_param):
    move_list = []
    for i, move in enumerate(move_param):
        target = move["target"]
        move_list.append(target)
    logging.info("move list num:{}".format(len(move_list)))
    logging.info("move list:{}".format(*move_list))
    motion_cmd = pack('>24f', *move_list)
    self.send(motion_cmd)
```

現場の要件に応じて、Mech-Vizのすべての移動点を送信するか、indexに従っていくつかの点を選択して送信することができます。pack_move()は通常、ロボットの各ブランドが必要とするフォーマットに従ってデータを統合します（通常、通信プロトコルで事前に設定される）。

6.5.3.8. その他のインターフェース

本節では、Adapterの他のインターフェースについて説明します。具体的には以下のインターフェースが含まれます。

通知サービス

Mech-Vizプロジェクトが特定の分岐または特定のステップに実行されたときに、Adapterプログラムの対応する関数を呼び出したい場合は、Mech-Vizに通知ステップを追加できます。

サンプル

たとえば、Adapterに把持された対象物の数を1ずつ増やす関数を記述した場合、デパレタイジングプロセスの最後のステップの後に通知ステップを追加できます。ここで到達すると、Adapterをトリガーして、対応する関数を呼び出すことができます。この機能の実装例を以下に示します。

1. NotifyServiceを継承したクラスを作成します。

```
from interface.services import NotifyService, register_service

class NotifyService(NotifyService):
    service_type = "notify"
    service_name = "FANUC_M410IC_185_COMPACT"

    def __init__(self, update_success_num, update_fail_num):
        self.update_success_num = update_success_num
        self.update_fail_num = update_fail_num

    def handle_message(self, msg):
        if msg == "Success":
            self.update_success_num()
        elif msg == "Fail":
            self.update_fail_num()
```

この通知は、次の機能を実装できます。Mech-Vizは「Success」を送信した通知ステップに実行すると、Adapterはupdate_success_num()関数を呼び出します。「Fail」を送信した通知ステップに実行すると、Adapterはupdate_fail_num()関数を呼び出します。

2. Mech-Vizメインプログラムを制御するクラスにNotifyServiceクラスをインスタンス化してこのサービスを登録します。

```
class MyClient(TcpClientAdapter):

    def __init__(self, host_address):
        super().__init__(host_address)
        self._register_service()

    def _register_service(self):
        self.robot_service = NotifyService(self.update_success_num, self.update_fail_num)
        self.server, port = register_service(self.hub_caller, self.robot_service)

    def update_success_num(self):
        # the num of unstack successfully plus 1
        self.success_num += 1

    def update_fail_num(self):
        # the num of unstack fiplus 1
        self.fail_num += 1
```

3. 必要に応じて、Mech-Vizプロジェクトに対応する通知ステップを追加します。

通知ステップで **Adapter名** および **メッセージ** を入力する必要があります。これら2つの値は、上記のNotifyServiceクラスのservice_nameおよびmsgの値と一致する必要があります。

プログラムが実行されると、service_typeとservice_nameがMech-Centerに表示されれば、通知サービスが正常に登録されました。

VisionWatcherサービス

Mech-Visionの実行が終了すると、いくつかの結果が出力されます。例えば：vision result:{‘noCloudInRoi’: False, ‘function’: ‘posesFound’, ‘vision_name’: ‘TJTvion-3’}。一部の異常な状況では、AdapterはVisionWatcherサービスを介してエラー情報を送信することができます。

サンプル

1. VisionResultSelectedAtServiceを継承したクラスを作成します。

```
from interface.services import VisionResultSelectedAtService, register_service

class VisionWatcher(VisionResultSelectedAtService):
    def __init__(self, send_err_no_cloud):
        super().__init__()
        self.send_err_no_cloud = send_err_no_cloud

    def poses_found(self, result):
```

```
has_cloud_in_roi = not result.get("noCloudInRoi", False)

if not has_cloud_in_roi:
    time.sleep(2)
    self.send_err_no_cloud()
```

子クラスVisionWatcherは、親クラスのposes_found()関数を書き変える必要があるため、Adapterでsend_err_no_cloud()(エラー情報を送信する関数)を呼び出し、poses_found()で書き換えられます。実行中に、Mech-Visionによって返されたビジョンポイントの値は、poses_found()の結果パラメータに渡されます。

2. Mech-Vizメインプログラムを制御するクラスにVisionWatcherクラスをインスタンス化します。

```
class MyClient(TcpClientAdapter):
    def __init__(self, host_address):
        super().__init__(host_address)
        self._register_service()
    def _register_service(self):
        self.robot_service = VisionWatcher(self.send_err_no_cloud)
        self.server, port = register_service(self.hub_caller, self.robot_service)

    def send_err_no_cloud(self):
        # send no cloud error message
        self.send("12,NoCloudErr,done".encode())
```

VisionWatcherクラスのインスタンス中に、send_err_no_cloud()関数をパラメータとしてVisionWatcher()に渡します。点群のない場合は、poses_found()のロジックに従ってエラーメッセージを送信する関数が呼び出されます。

プログラムが実行されると、登録されたサービスがMech-Centerに表示されれば、VisionWatcherサービスがAdapterによって正常に登録されました。

6.5.4. Adapterプログラミングの例

Adapterのプログラミング文法とロジックを理解したら、本節で提供されているサンプルプログラムを参考にしてAdapterプログラムを作成できます。

- [Mech-Visionのみを使用してビジョンポイントを送信](#)

6.5.4.1. Mech-Visionのみを使用してビジョンポイントを送信

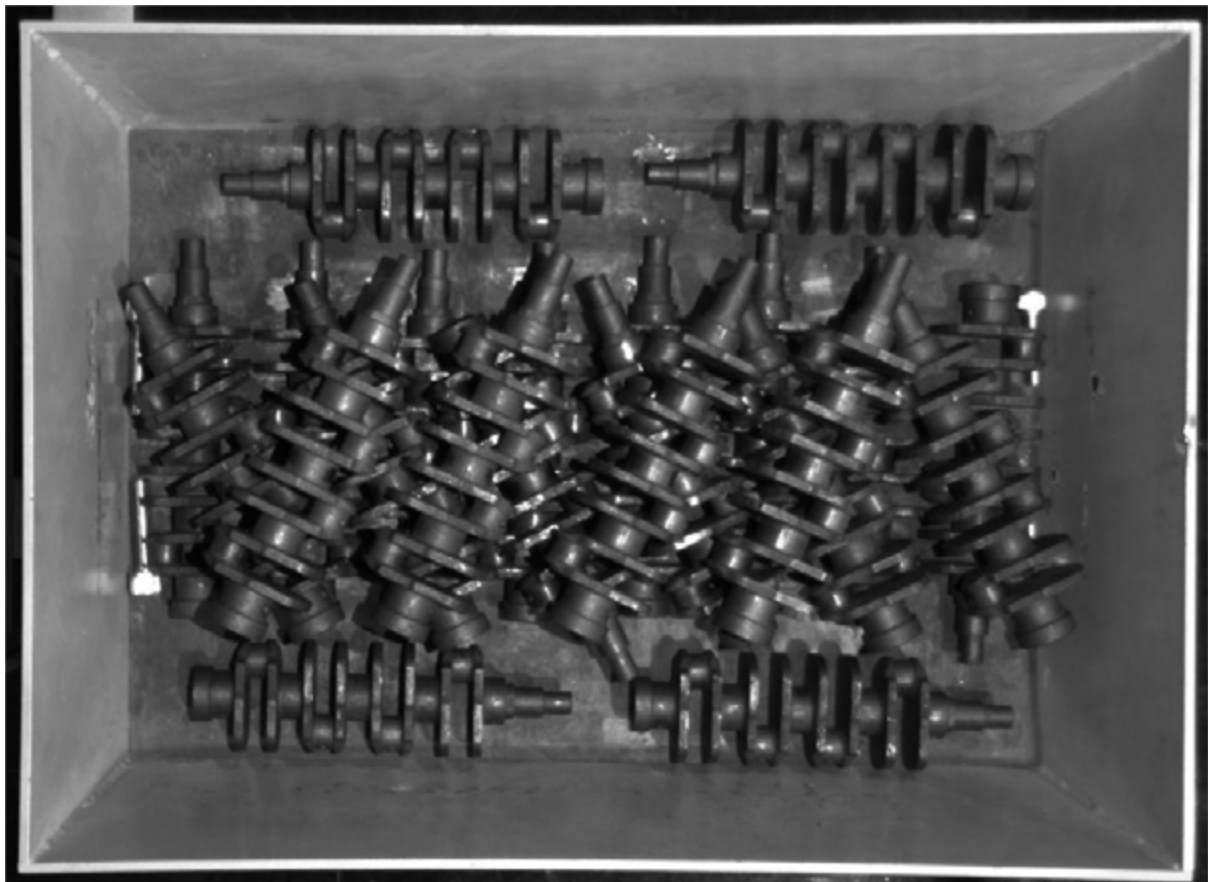
本節では、Mech-Visionのみを使用してビジョンポイントを送信するAdapterサンプルプログラムについて詳しく説明します。

背景

本例は、クランクシャフト供給の応用シーンに適しています。カメラは箱の上方のスタンドに取り付けられ、Mech-Visionは画像を撮影して把持可能なワーク座標をロボット側に出力します。

本例では、Mech-Visionに組み込まれているサンプルプロジェクトを使用して説明します。

このプロジェクトは3Dモデルマッチングのアルゴリズムを使用し、異なるワークに対して異なるモデルファイルと把持位置姿勢を設定する必要があります。したがって、Mech-Visionでパラメータレシピを設定する必要があり、ロボット側から画像撮影コマンドを送信する際にレシピ番号（ワーク番号）を設定します。

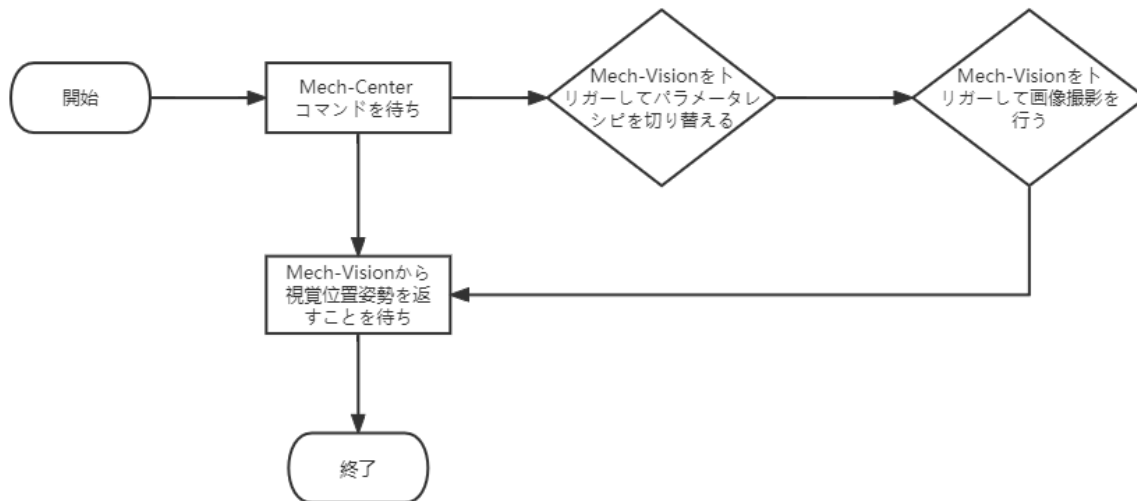


通信ソリューション

ロボットは、TCP/IP Socketプロトコルを使用してMech-MindソフトウェアシステムがインストールされたIPCと通信します。通信フォーマットはASCII文字列で、英語のコンマ（,）をデータ区切りとして使用されます。

その中、ビジョンシステムはサーバーとして使用され、ロボットはクライアントとして使用されます。

通信のプロセスは下図に示します。



通信プロセスは、次のように詳細に説明されています。

1. Mech-Centerは、ロボットが画像撮影コマンド **P** とレシピ番号を送信するのを待ちます。
2. Mech-Centerは、Mech-Visionをトリガーしてパラメータレシピを切り替えます。
3. Mech-Centerは、Mech-Visionをトリガーして撮影します。
4. Mech-Visionが画像を撮影して認識に成功すると、ステータスコードとビジョンポイントをMech-Centerに返します。
5. Mech-Centerは、ステータスコードとビジョンポイントをロボットに返します。



ロボットの把持を容易にするために、Mech-Centerは把持するワークの位置姿勢をロボットのTCP位置姿勢に変換します。

通信メッセージのフォーマット

詳細は下記の通りです。

	リクエストコマンド	ワーク番号
送信（ロボット → IPC）	P	整数、「1~100」の範囲
	ステータスコード	ビジョンポイント（TCP座標）
受信（IPC → ロボット）	「0~4」範囲内にある整数です。0-正常認識；1-エラーのコマンドコード；2-Mech-Visionプロジェクトが登録されない；3-ビジョンポイントなし；4-点群なし	カンマ「,」で区切られた6つの浮動小数点データ (x,y,z,a,b,cの形式)



応答メッセージの長さは固定です。応答メッセージのステータスコードが異常コード（1~4）の場合、ビジョンポイントのデータは0で埋める必要があります。

通信メッセージの例

リクエストメッセージ

P,1

通常応答時のメッセージ

0,1994.9217,-192.198,506.4646,-23.5336,-0.2311,173.6517



本例では、Mech-Visionは正常に認識し、「1994.9217,-192.198,506.4646,-23.5336,-0.2311,173.6517」のTCP座標を返します。

異常応答時のメッセージ：エラーのコマンドコード

1,0,0,0,0,0,0

異常応答時のメッセージ：Mech-Visionプロジェクトが登録されない

2,0,0,0,0,0,0

異常応答時のメッセージ：ビジョンポイントなし

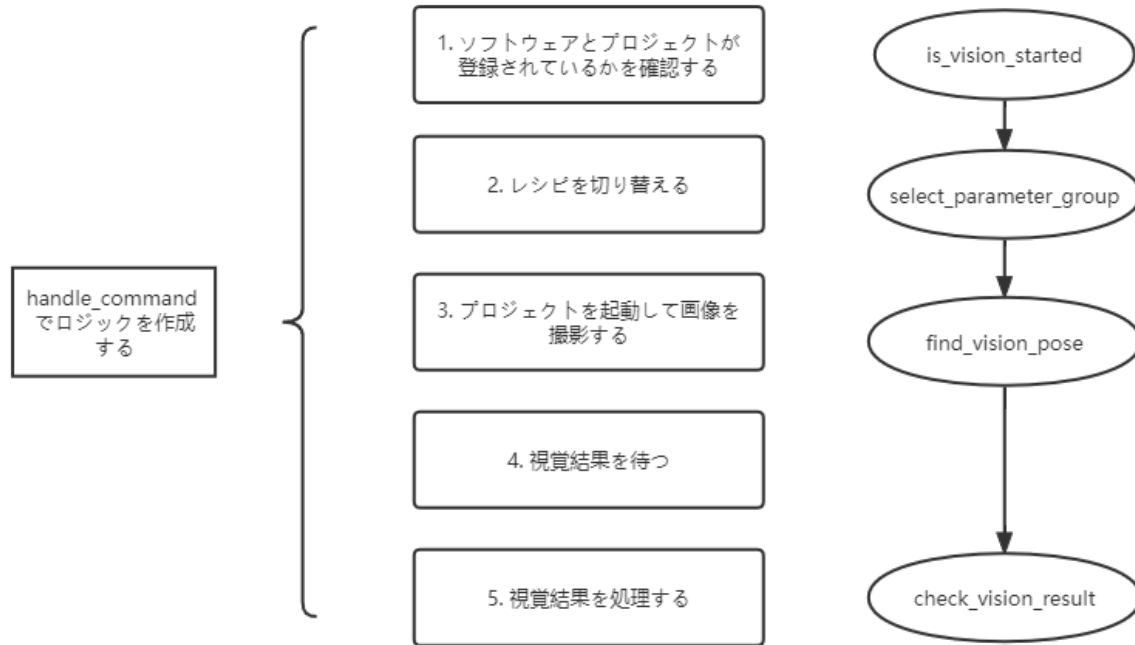
3,0,0,0,0,0,0

異常応答時のメッセージ：点群なし

4,0,0,0,0,0,0

プログラミング手順

プロジェクトの要件を満たすために、本例では、下図に示すような手順に従ってAdapterプログラムを作成します。



上記の図は、サンプルプログラムのメッセージ処理ロジックのみを示しています。以下では、サンプルプログラムについて詳しく説明します。

サンプルプログラムの詳細な説明



クリックして [Adapterサンプルプログラム](#) をダウンロードします。

Pythonパッケージのインポート

Adapterプログラムが依存するすべてのモジュールをインポートします。

```

import json
import logging
import math
import sys
from time import sleep
import os

sys.path.append(os.path.abspath(os.path.join(os.path.dirname(__file__), "..", "..")))
from transforms3d import euler
from interface.adapter import TcpServerAdapter, TcpClientAdapter
from util.transforms import object2tcp
  
```

クラスの定義

「TcpServerAdapter」親クラスを継承した「TestAdapter」サブクラスを定義します。

```
class TestAdapter(TcpServerAdapter):
    vision_project_name = "Large_Non_Planar_Workpieces"
    # vision_project_name = 'Vis-2StationR7-WorkobjectRecognition-L1'
    is_force_real_run = True
    service_name = "test Adapter"

    def __init__(self, address):
        super().__init__(address)
        self.robot_service = None
        self.set_recv_size(1024)
```



本例では、AdapterプログラムをTCP/IP Socket通信用のサーバーとして定義します。

受信コマンドと処理ロジックの設定

リクエスト（画像撮影コマンド、パラメータレシピ含む）を受け取る処理ロジックを設定します。

```
# Receive command _create_received_section
def handle_command(self, cmds):
    photo_cmd, *extra_cmds = cmds.decode().split(',')
    recipe = extra_cmds[0]
    # Check command validity _check_cmd_validity_section
    if photo_cmd != 'P':
        self.msg_signal.emit(logging.ERROR, 'Illegal command: {}'.format(photo_cmd))
        self.send(('1' + ' ' + ' ').encode())
        return
    # Check whether vision is registered _check_vision_service_section
    if not self.is_vision_started():
        self.msg_signal.emit(logging.ERROR, 'Vision not registered: {}'.format(self
.vision_project_name))
        self.send(('2' + ' ' + ' ').encode())
        return
    # Change TODO parameter "extra_cmds" according to actual conditions
    sleep(0.1) # wait for a cycle of getting in Vision
    # _check_vision_result_function_section

    try:
        result = self.select_parameter_group(self.vision_project_name, int(recipe) - 1)
        if result:
            result = result.decode()
            if result.startswith("CV-E0401"):
                self.send(('5' + ' ' + ' ').encode())
                return
            elif result.startswith("CV-E0403"):
                self.send(('5' + ' ' + ' ').encode())
                return
```

```

        raise RuntimeError(result)
    except Exception as e:
        logging.exception('Exception happened when switching model: {}'.format(e))
        self.send(('5' + ' ' + ' ').encode())
        return
    self.show_custom_message(logging.INFO, "Switched model for project successfully")

    self.msg_signal.emit(logging.WARNING, 'Started capturing image')
    try:
        self.check_vision_result(json.loads(self.find_vision_pose().decode()))

    except Exception as e:
        self.msg_signal.emit(logging.ERROR, 'Calling project timed out. Please check whether
the project is correct: {}'.format(e))
        self.send(('2' + ' ' + ' ').encode())

```



「handle_command」関数は、TCP/IP Socketサーバーがメッセージを受信するための処理エントリとして使用されます。

Mech-Visionからのビジョン結果のチェックを定義

Mech-Visionによって出力されたビジョン結果を確認するようにAdapterを設定します。

```

# Check vision results
def check_vision_result(self, vision_result, at=None):

    noCloudInRoi = vision_result.get('noCloudInRoi', True)
    if noCloudInRoi:
        self.msg_signal.emit(logging.ERROR, 'No point clouds')
        self.send(('4' + ' ' + ' ').encode())
        return

    poses = vision_result.get('poses')
    labels = vision_result.get('labels')
    if not poses or not poses[0]:
        self.msg_signal.emit(logging.ERROR, 'No visual points')
        self.send(('3' + ' ' + ' ').encode())
        return

    self.send(self.pack_pose(poses, labels).encode())
    self.msg_signal.emit(logging.INFO, 'Sent TCP successfully')

```

ビジョンポイントの出力形式を設定

ビジョンポイントが外部への出力形式を設定します。

```

# Pack pose _pack_pose_section

```

```
def pack_pose(self, poses, labels, at=None):

    pack_count = min(len(poses), 1)
    msg_body = ''
    for i in range(pack_count):
        pose = poses[i]
        object2tcp(pose)
        t = [p * 1000 for p in pose[:3]]
        r = [math.degrees(p) for p in euler.quat2euler(pose[3:], 'rzyx')]
        p = t + r
        self.msg_signal.emit(logging.INFO, 'Sent pose: {}'.format(p))
        msg_body += ('{:.4f},' * (len(p) - 1) + '{:.4f}').format(*p)

    if i != (pack_count - 1):
        msg_body += ','
    return '{}'.format(0) + msg_body + ''
```

Adapterの閉じる操作を定義

Adapterを閉じる方法を定義します。

```
def close(self):
    super().close()
```