



# 비전 시스템 소프트웨어 매뉴얼

v1.7.4

# 목차

---

1. 환영합니다!	1
2. 업데이트 설명	3
2.1. 업그레이드 주의사항	3
2.2. Mech-Vision업데이트 설명	6
2.3. Mech-Viz업데이트 설명	15
2.4. Mech-Center업데이트 설명	26
2.5. 1.7.0 프로젝트 업그레이드 가이드	30
3. 소프트웨어 설치 가이드	36
4. Mech-Vision	42
4.1. Mech-Vision 소개	42
4.1.1. 용어 및 개념	42
4.1.2. 사용자 인터페이스	44
4.1.2.1. 메뉴 바	45
4.1.2.2. 툴 바	48
4.1.2.3. 프로젝트 리스트	49
4.1.2.4. 스텝 라이브러리	51
4.1.2.5. 프로젝트 툴 바	54
4.1.2.6. 프로젝트 편집 구역	54
4.1.2.7. 프로젝트 배치 구역	58
4.1.2.8. 로그바	64
4.1.3. 솔루션 구축 과정	65
4.2. 일반 사용 가이드	66
4.2.1. 솔루션 라이브러리 사용 가이드	67
4.2.1.1. 솔루션 라이브러리 열기	67
4.2.1.2. 솔루션 또는 프로젝트 필터링/찾기	69
4.2.1.3. 솔루션 또는 프로젝트 정보 보기	70
4.2.1.4. 솔루션 또는 프로젝트에 대한 사용자 안내서 보기	70
4.2.1.5. 솔루션 라이브러리에서 솔루션 생성	71
4.2.2. 솔루션 사용 가이드	72
4.2.2.1. 솔루션 생성 및 저장	72
4.2.2.2. 솔루션 열기	73
4.2.2.3. 솔루션 자동 로드	74
4.2.2.4. 프로젝트 이름 바꾸기	75
4.2.2.5. 프로젝트를 닫기	75
4.2.2.6. 솔루션 구조 이해하기	76
4.2.3. 프로젝트 사용 가이드	78
4.2.3.1. 프로젝트 생성 및 저장	78
4.2.3.2. 프로젝트 이름 바꾸기	80
4.2.3.3. 프로젝트 열기	80
4.2.3.4. 프로젝트 실행 및 디버그	81
4.2.3.5. 프로젝트를 닫기	82
4.2.3.6. 현재 프로젝트를 자동으로 로드하기	83
4.2.3.7. 프로젝트 번호 조정	83
4.2.3.8. 기존 프로젝트를 솔루션으로 옮기기	84

4.2.3.9. 프로젝트 파일 구조 이해하기.....	85
4.2.4. 로봇 및 인터페이스 구성 사용 가이드.....	86
4.2.4.1. Adapter 통신 구성.....	86
4.2.4.2. EtherNet/IP 통신 구성.....	87
4.2.4.3. Modbus TCP 통신 구성.....	88
4.2.4.4. PROFINET 통신 구성.....	89
4.2.4.5. 로봇의 모델을 선택하기.....	90
4.2.4.6. Siemens PLC Snap 7 통신 구성.....	91
4.2.4.7. TCP/IP 통신 구성.....	93
4.2.4.8. Mitsubishi MC 통신 구성.....	94
4.2.4.9. UDP 통신 구성.....	96
4.2.4.10. 복제 파일 가져오기.....	97
4.2.4.11. 로봇을 선택하기.....	97
4.2.4.12. 인터페이스 서비스 시작.....	99
4.2.4.13. 로봇 라이브러리 도구 사용 가이드.....	100
4.2.5. 스텝 사용 가이드.....	103
4.2.5.1. 스텝을 검색 & 추가 & 삭제하기.....	104
4.2.5.2. 스텝 간 연결 생성 및 삭제.....	105
4.2.5.3. 스텝의 입력 및 출력을 보기.....	107
4.2.5.4. 스텝의 다양한 상태 보기.....	108
4.2.5.5. 스텝 실행 및 시각화 출력 결과를 보기.....	108
4.2.5.6. 스텝의 비고를 추가&수정&확인하기.....	113
4.2.5.7. 프로시저의 기본 사용법.....	115
4.2.5.8. 프로시저 파라미터 수정.....	117
4.2.5.9. 프로시저 자체 정의.....	122
4.3. 스텝 사용 가이드.....	131
4.3.1. 2D 매칭.....	134
4.3.2. 3D 근사 매칭(멀티 모델).....	139
4.3.3. 모든 파라미터를 받아들이기.....	148
4.3.4. 포즈에 레이블을 추가하기.....	149
4.3.5. 카메라의 왜곡으로 인한 정확하지 않는 포즈를 조정하기.....	150
4.3.6. 옵셋에 따라 포즈를 조정하기.....	151
4.3.7. 기울임에 따라 포즈를 조정하기.....	151
4.3.8. 포즈를 포인트 클라우드의 표면으로 조정하기.....	151
4.3.9. 입력 포트 할당자.....	152
4.3.10. 마스크안에 대응하는 포인트 클라우드를 추출하기.....	153
4.3.11. 마스크안에 대응하는 포인트 클라우드를 추출하기.....	154
4.3.12. 부울 값 리스트 논리 연산.....	157
4.3.13. 절대값을 계산하기.....	157
4.3.14. 마스크 면적을 계산하기.....	158
4.3.15. 두 포즈가 지정된 방향에 따른 거리를 계산하기.....	159
4.3.16. 두 포즈의 지정된 축 사이의 각도를 계산하기.....	160
4.3.17. 포인트 클라우드의 법선 벡터를 계산하고 필터링하기.....	161
4.3.18. 평면 포인트 클라우드의 포즈와 크기를 계산하기.....	163
4.3.19. 직사각형의 사이즈와 포즈를 계산하기.....	166
4.3.20. 카메라에서 이미지를 캡처하기.....	168
4.3.21. 포인트 클라우드 처리(GPU).....	176

4.3.22. 3D ROI 내의 포즈를 획득하기	178
4.3.23. 수치 비교	179
4.3.24. 입력된 각 포즈의 일부를 추출하여 새로운 포즈를 생성하기	179
4.3.25. 데이터 리스트에서 지정된 차원의 요소 개수를 통계하기	181
4.3.26. 물체 치수 분해	181
4.3.27. 포즈를 사원수와 평행 이동 벡터로 분해하기	182
4.3.28. 라인 세그먼트 감지	183
4.3.29. 역치에 근거한 이진 분류	185
4.3.30. 포인트 클라우드 다운 샘플링	186
4.3.31. 레이블 리스트를 빠르게 만들기	187
4.3.32. 3D 방법을 사용하여 포인트 클라우드의 에지를 추정하기	188
4.3.33. 직육면체 내의 포인트 클라우드를 추출하기	189
4.3.34. 원기둥 범위 내의 포인트 클라우드를 추출하기	191
4.3.35. 3D ROI 내의 포인트 클라우드를 추출하기	193
4.3.36. 필터링	197
4.3.37. 포즈 좌표축의 방향을 뒤집기	199
4.3.38. 실제 치수를 픽셀 치수로 전환하기	201
4.3.39. Cloud (XYZ-Normal)를 Cloud (XYZ-RGB)로 전환하기	202
4.3.40. 뎀스 맵을 포인트 클라우드로 전환하기	204
4.3.41. PoseList를 PoseLists로 전환	208
4.3.42. Variant를 VariantList로 전환하기	209
4.3.43. 지정된 크기의 직사각형 에지 템플릿을 생성하기	210
4.3.44. 첫 번째 이미지를 획득하기	210
4.3.45. 가장 높은 층의 포인트 클라우드를 획득하기	211
4.3.46. 가장 높은 층의 포즈를 획득하기	214
4.3.47. 포즈의 역계산	215
4.3.48. 레이블 매핑	216
4.3.49. 여러 개의 픽 포인트에 매핑하기	218
4.3.50. 데이터를 합치기	219
4.3.51. 포인트 클라우드를 합치기	220
4.3.52. 형태학적 변환	221
4.3.53. 법선 방향을 계산하기	226
4.3.54. 수치 확대 및 축소	227
4.3.55. 수치 계산	228
4.3.56. 포즈의 축을 지정된 방향으로 가리키도록 회전하기	229
4.3.57. 포인트 클라우드 클러스터링	231
4.3.58. 포인트 클라우드를 필터링하기	234
4.3.59. 포인트 클라우드 모양 감지기	237
4.3.60. 포인트를 필터링하기	241
4.3.61. 포즈 필터링	244
4.3.62. 출력	245
4.3.63. 3D 포인트 클라우드를 2D 이미지로 투영하기	247
4.3.64. 물체 사이즈 읽기	249
4.3.65. 포인트 클라우드를 읽기	253
4.3.66. 파일에서 포즈 정보 획득	253
4.3.67. 2D ROI 내의 조정된 이미지를 복원하기	254
4.3.68. 겹친 물체를 제거하기	257

4.3.69. 겹치는 다각형을 제거하기	261
4.3.70. 마스크 외부에 있는 다각형을 제거하기	264
4.3.71. 데이터 반복 및 연결	265
4.3.72. 지정축과 기준 방향의 각도가 최소가 되도록 포즈를 조정하기	266
4.3.73. 지정한 축과 각도에 근거하여 포즈를 회전하기	269
4.3.74. 이미지를 저장하기	269
4.3.75. 결과를 파일로 저장하기	272
4.3.76. 스텝 파라미터를 파일로 저장하기	273
4.3.77. 2D ROI 내 이미지의 크기를 조정하기	273
4.3.78. 포인트 클라우드를 외부 서비스로 보내기	280
4.3.79. 포즈의 사원수를 설정하기	282
4.3.80. 포즈의 평행 이동의 벡터를 설정하기	283
4.3.81. 이미지를 표시하기	284
4.3.82. 포인트 클라우드 및 포즈를 표시하기	286
4.3.83. 포인트 클라우드의 순서를 배열하기	288
4.3.84. 역치에 근거하여 수치를 분류하기	289
4.3.85. 실수 뺄셈	292
4.3.86. 일부 영역에서 점수가 낮은 포즈를 제거하기(NMS)	292
4.3.87. 포인트 클라우드 변환	294
4.3.88. 포즈 변환	299
4.3.89. 지정한 방향으로 포즈를 이동하기	306
4.3.90. 부울 값에 의해 제어 흐름을 촉발하기	310
4.3.91. 출력 수 제한	311
4.3.92. 포즈의 개수를 제한하기	312
4.3.93. 포즈가 3D ROI에 있는지 확인하기	313
4.3.94. 필요한 레이블인지 판단하기	314
4.3.95. 포인트 클라우드가 요구에 부합한지 검증하기	316
4.3.96. 기준 방향과의 각도에 근거하여 포즈 유효성을 판단하기	317
4.4. 핸드-아이 캘리브레이션 사용 가이드	318
4.4.1. 로봇 핸드-아이 캘리브레이션 프로세스 선택	319
4.4.2. Eye to Hand 시나리오에서의 자동 캘리브레이션	320
4.4.3. Eye in Hand 시나리오에서의 자동 캘리브레이션	344
4.4.4. Eye to Hand 시나리오에서의 수동 캘리브레이션(캘리브레이션 보드의 여러 임의 포즈)	366
4.4.5. Eye in Hand 시나리오에서의 수동 캘리브레이션(캘리브레이션 보드의 여러 임의 포즈)	381
4.4.6. Eye to Hand 시나리오에서의 수동 캘리브레이션(TCP 끝단점 터치)	396
4.4.7. Eye in Hand 시나리오에서의 수동 캘리브레이션(TCP 끝단점 터치)	413
4.4.8. 트러스 로봇의 핸드-아이 캘리브레이션 완료	430
4.4.9. Eye to Eye 시나리오에서의 핸드-아이 캘리브레이션	447
4.4.10. 참조 정보	460
4.4.10.1. 캘리브레이션 관련 개념	460
4.4.10.2. 캘리브레이션 원리	461
4.4.10.3. 트러스 로봇 캘리브레이션 설명	467
4.4.10.4. 캘리브레이션 결과 체크 및 분석	470
4.4.10.5. 자주 나타나는 문제와 해결 방법	479
4.5. 측정 모드 사용 가이드	485
4.5.1. 측정 모드 시작하기	486
4.5.2. 주요 알고리즘 소개	490

4.6. 보조 도구 사용 가이드	490
4.6.1. 카메라 뷰어	490
4.6.2. 고정 배경을 설정하여 노이즈를 제거하기	492
4.6.3. ROI 설정	495
4.6.4. 시나리오 포인트 클라우드 설정	496
4.6.5. 매칭 모델 및 픽 포인트 편집기	498
4.6.6. 딥 러닝 서버	498
4.6.7. 딥 러닝 모델 패키지 관리 도구	500
4.6.8. 파라미터 레시피	505
4.6.9. 데이터 저장	519
4.6.10. 로그	523
5. Mech-Viz	525
5.1. Mech-Viz 소개	525
5.1.1. 기본 개념	525
5.1.2. 사용자 인터페이스	527
5.1.2.1. 메뉴 바	529
5.1.2.2. 툴 바	532
5.1.2.3. 프로젝트 리소스	532
5.1.2.4. 3D 시뮬레이션 공간	536
5.1.2.5. 작업 흐름	538
5.1.2.6. 로봇	541
5.1.2.7. 충돌 감지	541
5.1.2.8. 계획 기록	542
5.1.2.9. 기타	544
5.1.2.10. 로그	545
5.1.3. 프로젝트 구축 방안	546
5.2. 프로젝트 리소스 구성	546
5.2.1. 로봇	547
5.2.2. 말단장치	549
5.2.3. 작업물	554
5.2.3.1. 작업물의 회전 대칭성	555
5.2.3.2. 작업물 피킹 허용 편차	558
5.2.4. 바닥	559
5.2.5. 시나리오 물체	559
5.3. 작업 흐름 구축	562
5.3.1. 작업 흐름 구축	563
5.3.2. 시뮬레이션된 로봇을 움직이게 하는 방법	563
5.3.3. 실제 로봇을 움직이게 하는 방법	564
5.4. 스텝 라이브러리	565
5.4.1. 이동	566
5.4.1.1. 동적 이동	566
5.4.1.2. 이동	571
5.4.1.3. 배열대로 이동	577
5.4.1.4. 순서대로 이동	583
5.4.1.5. 외부 이동	588
5.4.1.6. 상대적인 이동	590
5.4.2. 논리적 토폴로지	598

5.4.2.1. 길표지 분기	598
5.4.2.2. 메시지 분기	599
5.4.2.3. 프로시저	600
5.4.2.4. 프로시저 아웃 포트	601
5.4.2.5. 길표지를 설정하기	602
5.4.3. 로봇 유틸리티	604
5.4.3.1. 말단장치를 체크하기	604
5.4.3.2. 관절 각도를 획득하기	605
5.4.3.3. 페이로드 설정	606
5.4.3.4. 피킹 상태 설정	607
5.4.3.5. 말단장치를 바꾸기	607
5.4.3.6. 제어권 이전	608
5.4.4. 서비스	609
5.4.4.1. 알림	609
5.4.5. 도구	611
5.4.5.1. 분류	611
5.4.5.2. 카운터	613
5.4.5.3. 완료 여부 확인	614
5.4.5.4. 인덱스 변경	616
5.4.5.5. 리셋	618
5.4.5.6. 기다리기	618
5.4.6. 경로	619
5.4.6.1. 경로 프로시저	619
5.4.7. 비전	619
5.4.7.1. 비전 결과 체크	620
5.4.7.2. 파렛트 포즈 업데이트	621
5.4.7.3. 피킹된 물체를 업데이트하기	623
5.4.7.4. 시나리오 물체 업데이트	626
5.4.7.5. 비전 결과를 모두 사용했음	629
5.4.7.6. 비전 인식	632
5.4.8. 기타	633
5.4.8.1. 작업 흐름 종료	633
5.4.9. 일반적인 파라미터	633
5.4.9.1. 이동 스텝 기본 파라미터	633
5.4.9.2. 피킹된 물체의 충돌 감지 모드	634
5.4.9.3. 비이동 스텝 기본 파라미터	635
5.4.9.4. 기본 이동 설정	635
5.4.9.5. 웨이포인트 유형	636
5.4.9.6. 관절 각도 제한 조건	637
5.5. 프로젝트 시뮬레이션 최적화	638
5.5.1. 충돌 감지	639
5.5.2. 계획 기록	642
5.5.3. 실제 로봇을 실행하기	644
5.5.4. 계산 설정	644
5.6. 도구 사용 가이드	645
5.6.1. 모델 편집기	645
5.6.1.1. 인터페이스 소개	652

5.6.1.2. STEP/STP 파일의 유효성 반정	655
5.6.2. 일반적인 빨판 구성기	655
5.6.3. 어레이 그리퍼 구성기	656
5.6.4. 파렛트 패턴 편집기	656
5.7. 부록	657
5.7.1. 로봇 모델의 제작 및 도입	657
5.7.1.1. 로봇 구성	664
5.7.1.2. [robot]_algo.json 파일의 파라미터 설명	666
5.7.1.3. [robot]_profile.json 파일의 파라미터 설명	667
5.7.2. 기타	668
5.7.2.1. 설정 - 옵션	668
5.7.2.2. 표시 - 디스플레이 설정	668
5.7.2.3. 툴 - 비전 기록 설정	670
6. Mech-Center	672
6.1. Mech-Center 퀵 가이드	672
6.1.1. 메뉴 바	673
6.1.2. 툴 바	674
6.1.3. 서비스 상태 표시줄	676
6.1.4. 프로젝트 상태 표시줄	676
6.1.5. 로그바	676
6.2. Mech-Center 시작하기	676
6.3. Mech-Interface	678
6.4. 표준 인터페이스	681
6.5. Adapter	683
6.5.1. Adapter 퀵 가이드	683
6.5.2. Adapter 생성기 매뉴얼	686
6.5.3. Adapter 프로그래밍 가이드	692
6.5.3.1. Adapter 프로그래밍 스타일 가이드라인	693
6.5.3.2. 추상 상위 클래스 인터페이스	695
6.5.3.3. Adapter util 패키지	711
6.5.3.4. 인터페이스를 획득하기	712
6.5.3.5. Mech-Vision 인터페이스	713
6.5.3.6. Mech-Viz 인터페이스	716
6.5.3.7. RobotService 인터페이스	724
6.5.3.8. 기타 인터페이스	728
6.5.4. Adapter 프로그래밍 샘플	730
6.5.4.1. Mech-Vision만 사용하여 비전 포즈를 전송하기	730

# 1. 환영합니다!

---

Mech-Mind 비전 소프트웨어 사용자 매뉴얼에 오신 것을 환영합니다.

Mech-Vision, Mech-Viz 및 Mech-Center 세 가지 소프트웨어로 구성된 Mech-Mind 비전 소프트웨어 시스템에서는 비전 처리, 경로 계획과 로봇 컨트롤 등 기능을 제공합니다.

## 업데이트 설명

### Mech-Vision 업데이트 설명

Mech-Vision 최신 버전의 기능 추가, 기능 최적화 및 문제 복원에 관한 설명입니다.

[더 보기](#)

### Mech-Viz 업데이트 설명

Mech-Viz 최신 버전의 기능 추가, 기능 최적화 및 문제 복원에 관한 설명입니다.

[더 보기](#)

### Mech-Center 업데이트 설명

Mech-Center 최신 버전의 기능 추가, 기능 최적화 및 문제 복원에 관한 설명입니다.

[더 보기](#)

### 업그레이드 주의사항

Mech-Vision, Mech-Viz 및 Mech-Center 소프트웨어를 업그레이드할 때의 주의사항입니다.

[더 보기](#)

### 소프트웨어 설치 가이드

소프트웨어의 다운로드, 설치, 업그레이드, 제거, 재설치 및 수정에 관한 소개입니다.

[더 보기](#)

### 프로젝트 업그레이드 가이드

소프트웨어 업그레이드에 필요한 프로젝트 마이그레이션입니다.

[더 보기](#)

## 소프트웨어 사용자 매뉴얼

### Mech-Vision

Mech-Vision 머신 비전 소프트웨어

[더 보기](#)

## **Mech-Viz**

Mech-Viz 로봇 프로그래밍 소프트웨어

[더 보기](#)

## **Mech-Center**

Mech-Center 통신 허브 및 제어 센터

[더 보기](#)

## 2. 업데이트 설명

이 부분에서는 Mech-Vision 1.7, Mech-Viz 1.7 및 Mech-Center 1.7 버전의 새로 추가된 기능, 최적화된 기능 및 복구된 문제에 대해 소개하겠습니다.



Mech-Vision, Mech-Viz 및 Mech-Center 소프트웨어를 업그레이드하기 전에 이 부분 내용을 자세히 읽어 주시기 바랍니다.

소프트웨어 업그레이드 주의사항 및 프로젝트 업그레이드 가이드는 [1.7.0 업그레이드 주의사항](#) 및 [1.7.0 프로젝트 업그레이드 가이드](#)에서 확인하실 수 있습니다.

Mech-Vision 1.7 버전의 새로 추가된 기능, 최적화된 기능 및 복원된 문제는 [Mech-Vision 1.7 업데이트 설명](#) 내용을 참조하십시오.

Mech-Viz 1.7 버전의 새로 추가된 기능, 최적화된 기능 및 복원된 문제는 [Mech-Viz 1.7 업데이트 설명](#) 내용을 참조하십시오.

Mech-Center 1.7 버전의 새로 추가된 기능, 최적화된 기능 및 복원된 문제는 [Mech-Center 1.7 업데이트 설명](#) 내용을 참조하십시오.

### 2.1. 업그레이드 주의사항

#### 1.7.2 업그레이드 주의사항

Mech-Vision 1.7.2 버전에 “딥 러닝 모델 패키지 CPU 추론” 및 “딥 러닝 모델 패키지 추론(Mech-DLK 2.2.0+)” 스텝을 대체하는 [딥 러닝 모델 패키지 추론](#) 스텝을 추가했습니다.

Mech-Vision 소프트웨어를 1.7.2 버전으로 업그레이드하면 구버전 프로젝트 중의 “딥 러닝 모델 패키지 CPU 추론” 및 “딥 러닝 모델 패키지 추론(Mech-DLK 2.2.0+)” 스텝은 자동으로 “딥 러닝 모델 패키지 추론” 스텝으로 대체됩니다. 새로운 추가된 스텝은 이전에 사용했던 두 개의 스텝과 같은 기능을 제공하므로 이전에 구축한 프로젝트의 정상적인 사용에 영향을 미치지 않습니다.

Mech-Vision 1.7.2 이전 버전에서 구버전 소프트웨어를 통해 최적화된 모델 패키지를 사용하는 경우 "딥 러닝 모델 패키지 추론" 스텝에서 모델 패키지의 첫 번째 실행 시간이 더 오래 걸릴 수 있습니다.

#### 1.7.0 업그레이드 주의사항

이 부분에서는 Mech-Vision 1.7.0, Mech-Viz 1.7.0 및 Mech-Center 1.7.0 버전의 업그레이드 주의사항을 소개하겠습니다.

##### Mech-Vision 1.7.0 업그레이드 주의사항

###### “솔루션”이 새로 추가된 후의 주의사항

Mech-Vision 1.7.0 버전에서는 로봇 모델 선택, 로봇 통신 구성, 비전 프로젝트 구축 및 로봇 경로 계획 등 비전 애플리케이션 배포를 하나의 솔루션으로 완료할 수 있는 "솔루션"을 추가합니다. 다음 사항에

유의하십시오:

- 표준 인터페이스 또는 Adapter를 통해 통신하는 경우, 로봇 프로그램이 호출하는 Mech-Vision 번호는 Mech-Center가 아닌 Mech-Vision의 프로젝트 리스트에서 획득됩니다.
- 소프트웨어를 업그레이드한 후 이전에 구축한 Mech-Vision 프로젝트를 이전해야 합니다. 상세한 설명은 [1.7.0 프로젝트 업그레이드 가이드](#) 내용을 참조하십시오.

#### “Mech-Center 인터페이스 서비스” 기능이 통합된 후의 주의사항

Mech-Vision 1.7.0 버전에서는 인터페이스 통신 서비스 기능이 이미 통합되었으므로 다음 사항에 유의해야 합니다:

- Mech-Vision 솔루션을 사용하고 표준 인터페이스 또는 Adapter를 통해 통신하는 경우 Mech-Center 소프트웨어를 사용할 필요가 없습니다.
  - Mech-Vision 소프트웨어의 툴 바에서 [ **로봇 및 인터페이스 구성** ] 버튼을 클릭하여 해당 로봇을 선택하고 통신 방식을 설정한 후 [ **응용** ] 버튼을 클릭하면 인터페이스 서비스가 자동으로 활성화됩니다.
  - 로봇 브랜드가 ABB, FANUC, YASKAWA, KAWASAKI, KUKA, UR, TM, ELITE, JAKA인 경우 기본적으로 선택된 통신 방식은 Mech-Mind에서 제공되는 로봇 측의 표준 인터페이스 샘플 프로그램에서 사용하는 통신 방식과 일치합니다. 다른 브랜드의 로봇인 경우, 기본적으로 TCP/IP 통신 방식을 선택합니다.
  - 인터페이스를 구성할 때 “솔루션을 열 때 인터페이스 서비스 자동 활성화”를 선택할 수 있습니다. 다음에 해당 솔루션을 열면 프로젝트가 로봇 측 프로그램과 직접 통신할 수 있습니다.
  - 통신과 관련된 로그는 Mech-Vision 로그바의 Console 탭에서 확인할 수 있습니다.
  - Adapter 인터페이스를 구성할 때 Adapter 프로젝트 폴더를 선택하면 폴더가 현재 프로젝트 폴더에 자동으로 복사됩니다.
  - 솔루션의 Adapter 프로그램을 수정하고 적용하면 Adapter 프로그램과 인터페이스 서비스를 재시작해야 합니다.
  - Mech-Center 소프트웨어의 “PC 부팅 시 Mech-Center 자동으로 실행하기” 기능을 비활성화해야 합니다.
- 통신 방식이 마스터 컨트롤 통신이거나 복잡한 작업을 달성하기 위해 Mech-Viz을 사용하는 경우, 여전히 Mech-Center 소프트웨어를 사용해야 합니다.
  - 사용자는 먼저 Mech-Center 소프트웨어를 연 다음 Mech-Vision 소프트웨어를 열어 이전 버전에서 설명한 대로 통신을 구성해야 합니다.
  - Mech-Center 소프트웨어의 “PC 부팅 시 Mech-Center 자동으로 실행하기” 기능을 활성화하여 Mech-Center 소프트웨어를 통해 Mech-Vision 소프트웨어를 시작하는 것을 권장합니다.

#### 캘리브레이션 주의사항

Mech-Vision 1.7.0 캘리브레이션 사전 설정에서는 로봇 선택, 프로그램 복제, 로봇 연결 등 기능이 추가되었으며, 캘리브레이션 사전 설정에서 로봇과의 연결을 완료할 수 있습니다. 프로젝트가 솔루션에 포함되지 않은 경우 캘리브레이션 기능을 사용할 수 없습니다.

#### 스텝과 관련 주의사항

Mech-Vision 1.7.0 버전에서 스텝을 사용할 때 다음과 같은 사항에 유의하십시오.

#### 삭제된 스텝

Mech-Vision 1.7.0 버전에서 아래와 같은 스텝들을 삭제했습니다.

클래스	삭제한 스텝
2D 특징 추출 - 기타	구석점과 사이즈에 근거하여 직사각형을 감지하기
2D 통용 처리 - 기타	다각형 배치하기
딥 러닝 - 임의의 물체 피킹	확률 히트맵에서 포즈를 계산하기, 픽 포인트를 예측하기(단일 물체 유형)(새로운 스텝 “픽 포인트를 예측하기 V2”를 사용하세요), 픽셀당 피킹 가능한 확률을 예측하기
측정 - 3D 길이/거리	점에서 두 평면의 교차점까지의 거리를 계산하기, 점에서 평면까지의 거리를 계산하기
구 버전	이미지에서 정보를 시각화하기(새로운 스텝 “이미지에서 정보를 시각화하기”를 사용하세요), 장애물에 근거하여 포즈를 조정하기
포즈 - 평행이동&방향 조정	장애물에 근거하여 포즈를 조정하기(V2)
기타	피킹 가능한 직사각형을 감지하기

구버전 딥 러닝 관련 스텝은 더 이상 유지되지 않습니다.

결함 검출, 에지 감지, 이미지 분류, 인스턴스 세그멘테이션, 물체 검출, 딥 러닝 추론(Mech-DLK 2.1.0/2.0.0) 등 스텝은 더 이상 사용되지 않고 나중에 새버전에서 삭제될 계획입니다.

- 하드웨어 유형이 GPU인 경우, 결함 검출, 에지 감지, 이미지 분류, 인스턴스 세그멘테이션, 물체 검출, 딥 러닝 추론(Mech-DLK 2.1.0/2.0.0) 등 스텝을 사용하려면 새버전 스텝 “딥 러닝 모델 패키지 추론(Mech-DLK 2.2.0+)”을 사용하십시오.
- 하드웨어 유형이 CPU인 경우, 이미지 분류, 인스턴스 세그멘테이션, 물체 검출, 딥 러닝 추론(Mech-DLK 2.1.0/2.0.0) 등 스텝을 사용하려면 새버전 스텝 “딥 러닝 모델 패키지 CPU 추론”을 사용하십시오.

### Mech-Viz 1.7.0 업그레이드 주의사항

Mech-Viz 1.7.0 버전에서는 로봇 네이밍이 표준화되었으며 많은 로봇 모델의 이름이 수정되었습니다. 사용자는 로봇 모델 라이브러리에서 로봇을 다시 선택해야 합니다.

Mech-Viz 1.7.0 버전에서는 일부 로봇의 DO 포트 상한 값이 수정되었고 일부 구버전 프로젝트를 열 때 팝업 프롬프트가 표시될 수 있습니다. 팝업 프롬프트를 없애려면 robot\_ability에 로봇과 해당하는 digital\_out\_range를 수동으로 수정해야 합니다.

Mech-Viz 1.7.0 버전에서 빈에 해당하는 포즈의 유효 범위를 수동으로 설정할 수 있습니다. 이전 버전을 통해 생성한 빈을 사용한 프로젝트인 경우, 빈 포즈의 유효 범위는 [빈 바닥, +∞) 에서 [빈 내부, 빈 상단에서 1m 이내]로 변경되었습니다.

"비전 이동" 스텝의 "디팔레타이징 빨판" 모드는 완전히 재구성되었습니다. 일부 원래 소프트웨어의 기능이 분할되어 새 버전의 기능에 직접 대응할 수 없으며 다수 피킹과 관련된 파라미터를 재설정해야 합니다.

"비전 이동" 스텝에서 "디팔레타이징 빨판" - "디팔레타이징 다수 피킹" 모드에서 상자를 조합한 후의 작업물 모델은 하나의 큰 상자에서 여러 작은 상자로 수정되었습니다. 이렇게 하면 작업물 모델은 실제 상황과 더욱 비슷하지만 작업물 모델의 변화로 인해 로봇 경로 계획 과정에서 불필요한 충돌이 발생할 수도 있습니다.

빨판의 구성 창이 말단장치의 구성 창으로 위치가 변경되고 빨판의 구성이 말단장치 구성에 바인딩됩니다. 이전에 빨판 구성을 사용한 프로젝트인 경우, 수동으로 빨판 구성을 재설정해야 합니다.

로봇 말단장치 충돌 모델은 obj 형식만 지원하며 더 이상 stl 형식을 지원하지 않습니다. 관련 충돌 구성 파라미터는 충돌 감지 구성 인터페이스에서 제거됩니다.

말단장치의 충돌 모델이 시각화 모델의 파일 이름 및 파일 유형과 완전히 같은 경우 소프트웨어는 시각화 모델을 충돌 모델로 대체합니다.

### Mech-Center 1.7.0 업그레이드 주의사항

Mech-Center 1.7.0은 반드시 Mech-Vision과 Mech-Viz 1.7.0 및 이상 버전과 결합해서 사용해야 합니다.

솔루션의 프로젝트에 "경로 계획" 스텝이 사용되는 경우, 로봇 측에서 인터페이스 프로그램을 다시 복제해야 하며 Mech-Vision에서 "로봇 및 인터페이스 구성"과 관련된 구성을 해야 합니다.

## 2.2. Mech-Vision 업데이트 설명

이 부분에는 Mech-Vision 1.7 버전의 추가된 기능, 최적화된 기능 및 복구된 문제에 대해 소개하겠습니다.

### Mech-Vision 1.7.4 업데이트 설명

이 부분에서는 Mech-Vision 1.7.4 버전의 최적화된 기능 및 복구된 문제에 대해 소개하겠습니다.

#### 기능 최적화

##### 동글 권한 부여에 소프트웨어 지역 제한이 추가됨

소프트웨어 판매 지역에 따라 동글 권한 부여 시 소프트웨어를 사용하는 지역에 대한 제한이 추가되었습니다.

Mech-Vision 1.7.4 버전에서 [도움](#) > [버전 정보](#) 소프트웨어 라이선스 및 지역 제한을 순서대로 클릭하면 소프트웨어 라이선스 및 지역 제한을 확인할 수 있습니다.

##### “시스템 언어” 옵션이 제거됨

Mech-Vision 1.7.4 버전부터 소프트웨어 언어 옵션 ([설정](#) > [옵션](#) > [일반 설정](#) > [언어](#))에서 “시스템 언어”를 제거하였습니다. 컴퓨터 운영 체제의 언어가 소프트웨어에서 지원하는 언어(영어, 간체 중국어, 한국어 또는 일본어)가 아닌 경우, 소프트웨어가 처음으로 설치될 때 기본 언어는 영어로 설정됩니다.

해당 최적화는 다음 두 가지 문제를 해결했습니다.

- 이전 버전에서는 현재 컴퓨터 운영 체제의 언어가 소프트웨어에서 지원한 시스템 언어 중 하나가 아닌 경우, 소프트웨어 내부 링크를 클릭한 후 해당 페이지에 들어갈 수 없으며 스텝 이름이 비정상적으로 표시되는 문제를 해결했습니다.
- 소프트웨어가 컴퓨터 언어에 따라 언어 패키지를 올바르게 설정할 수 없는 문제를 해결했습니다.

시나리오 포인트 클라우드 설정은 리스트 형식으로 표현된 포인트 클라우드를 지원합니다.

Mech-Vision 1.7.4부터 [프로젝트 도우미](#)에서 리스트 형식으로 표현된 포인트 클라우드를 참조용 시나리오 포인트 클라우드로 설정할 수 있습니다.

#### 일부 솔루션 제거

솔루션 라이브러리의 전반적인 품질과 사용 효과를 보장하기 위해 Mech-Vision 1.7.4는 솔루션 라이브러리에서 일부 솔루션을 일시적으로 제거하고 최적화합니다. 최적화 후 후속 버전에 다시 나열됩니다. 제거된 솔루션은 아래 표에서 확인하실 수 있습니다.

솔루션 유형	솔루션 명칭(굵게 표시된 솔루션은 빌트인 솔루션임)
부품 로드&언로드	<b>크랭크 축(대형), 완전한 파티션</b> , 얇은 빈, 알루미늄 인고트 벽돌, 들보, 크랭크 축(소형), 콘덴서, 등속 조인트의 외륜, 구리선, 커넥팅 로드, 체인 링크, 압축기, 4개의 측벽이 있는 큰 철제 빈(포인트 클라우드 품질 양호), 개스킷(소), 기어 샤프트, 바퀴통, 철볼, 철제 빈(4개의 기둥), 불안정하게 캡처된 빈, 철제 빈(양면 벽), 판스프링, 대형 링, 다중 파티션, 깔끔하게 정렬된 작은 작업물, PVC 패널, 회전자, 링 기어, 회전축, 무작위로 쌓인 작은 작업물(높은 반사), 무작위로 쌓인 작은 작업물(볼트), 무작위로 쌓인 작은 작업물(낮은 반사), 슬리브, 스틸 스페이서, 철근, 정사각형 벽돌, 트레이, 밸브 커넥터
디팔레타이징/팔레타이징	통
로케이팅/어셈블리	<b>휠 얼라인먼트</b> , 자동 충전 위치 지정(솔루션), 볼트 위치 지정, 자동차 프레임 위치 지정, 자동차 도어 프레임 위치 지정, 나사 구멍 위치 지정, 휠 밸브 코어 위치 지정
구즈 피킹	케이블, 약 상자



- 위의 솔루션을 이미 사용한 사용자의 경우 계속 사용하는 데 영향을 미치지 않습니다.
- 이전 버전의 소프트웨어를 사용하여 위의 온라인 솔루션 리소스를 획득한 경우 Mech-Vision 1.7.4로 업그레이드한 후 솔루션 카드가 여전히 존재할 수 있지만 솔루션을 사용할 수 없습니다.

## 문제 복구

Mech-Vision 1.7.4 버전에서 다음과 같은 문제들을 복구했습니다.

- "포인트 클라우드 클러스터링 후 합격한 포인트 클라우드 출력" 스텝은 잘못된 개수의 포인트 클라우드를 출력할 확률이 매우 낮습니다.
- 신경망에 보낸 이미지가 단일 채널 이미지인 경우 "딥 러닝 모델 패키지 추론" 스텝에서 "이미지에 결함 마스크 그리기" 기능이 활성화되고 프로젝트 실행 후 OpenCV 관련 오류가 나타납니다.
- "딥 러닝 모델 패키지 추론" 스텝을 사용하여 Mech-DLK 2.4.1에서 도출한 대상 감지 모델 패키지를 로드합니다. 모델 패키지를 도출할 때 "최대 인스턴스 수"는 1이며, 모델 패키지 추론 시의 "하드웨어 유형"이 "CPU"이면 모델 패키지 추론이 오래 걸립니다.
- "딥 러닝 모델 패키지 추론" 스텝을 사용하여 일부 입력한 이미지 데이터에 대해 인스턴스 세그먼테이션을 수행할 때 OpenCV 관련 오류가 발생할 확률이 매우 낮습니다.
- 트러스 로봇의 캘리브레이션 후의 외부 파라미터 파일은 "포즈 변환(트러스)" 및 "포인트 클라우드 변환(트러스)" 스텝과 함께 사용할 수 없습니다.
- "경로 계획" 스텝에서 "고정점 이동" 및 "상대적인 이동" 스텝의 "로봇으로 보내기" 파라미터를 "계획하지만 보내지 않음"으로 설정하면 위의 스텝은 여전히 웨이포인트를 로봇으로 보냅니다.

## Mech-Vision 1.7.2 업데이트 설명

이 부분에서는 Mech-Vision 1.7.2 버전의 새로 추가된 기능, 최적화된 기능 및 복구된 문제에 대해 소개하겠습니다.

### 기능 추가

“픽 포인트를 예측하기 V2” 스텝은 “약 상자” 응용 시나리오에 적용됨

픽 포인트를 예측하기 V2 스텝은 “약 상자” 응용 시나리오에 적용되며 무작위로 흩어진 약 상자를 분류하고 피킹할 수 있습니다.

“약 상자” 프로젝트(딥 러닝 모델이 포함됨)는 솔루션 라이브러리에서 획득하십시오.

### “딥 러닝 모델 패키지 관리 도구” 기능 추가

딥 러닝 모델 패키지 관리 도구는 “배치 크기”와 “정밀도” 두 가지 측면에서 모델 패키지의 추론 효율성을 구성할 수 있습니다(Mech-DLK 2.4.1 이후 버전을 통해 도출된 모델 패키지에만 적용됨).



Mech-DLK 2.4.1 및 이상 버전에서 내보낸 딥 러닝 모델 패키지는 Mech-Vision 1.7.2 및 이상 버전과 결합해서 사용하는 것을 권장합니다.

Mech-Vision 1.7.1 및 이상 버전에서도 Mech-DLK 2.4.1 버전에서 내보낸 딥 러닝 모델 패키지를 사용할 수 있지만 [호환성 설명](#)에 주의하여야 합니다.

### “딥 러닝 모델 패키지 추론” 스텝이 추가됨

Mech-Vision 1.7.2 버전부터 “딥 러닝 모델 패키지 CPU 추론”과 “딥 러닝 모델 패키지 추론(Mech-DLK 2.2.0+)” 스텝을 대체하는 [딥 러닝 모델 패키지 추론](#) 스텝을 추가했습니다.

Mech-Vision 1.7.2 버전을 통해 구버전 프로젝트를 연 후 “딥 러닝 모델 패키지 CPU 추론” 및 “딥 러닝 모델 패키지 추론(Mech-DLK 2.2.0+)” 스텝은 자동으로 “딥 러닝 모델 패키지 추론” 스텝으로 대체됩니다.

이 스텝은 Mech-DLK에서 내보낸 단일 또는 캐스케이딩 모델 패키지에 대해 추론을 수행하고 추론 결과를 출력할 수 있습니다. Mech-DLK 2.2.0 및 이상 버전에서 내보낸 모델 패키지만 지원합니다.



Mech-DLK 2.4.1 버전 이후 모델 패키지는 단일 및 캐스케이딩 모델 패키지 두 가지로 나뉩니다.

- 단일 모델 패키지: 모델 패키지에 단 하나의 딥 러닝 알고리즘 모듈이 있는 모델. 예: "인스턴스 세그멘테이션" 모듈.
- 캐스케이딩 모델 패키지: 모델 패키지에는 딥 러닝 알고리즘 모듈의 여러 모델이 직렬로 존재하며 이전 모델의 출력은 다음 모델의 입력입니다. 예: 모델 패키지에 “물체 검출”과 “인스턴스 세그멘테이션” 두 모듈이 있으면 추론 순서는 **물체 검출** > **인스턴스 세그멘테이션**이며 “물체 검출” 모듈의 출력 결과는 “인스턴스 세그멘테이션” 모듈의 입력이 됩니다.

이 스텝은 캐스케이딩 모델 패키지를 사용하여 추론을 수행할 때 “딥 러닝 결과 파서” 스텝을 통해 추론 결과에 대해 분석할 수 있습니다.

### “딥 러닝 결과 파서” 스텝이 추가됨

Mech-Vision 1.7.2 버전에서 [딥 러닝 결과 파서](#) 스텝이 새로 추가되며 “딥 러닝 모델 패키지 추론” 스텝에서 출력한 캐스케이딩 모델 패키지의 추론 결과를 분석할 수 있습니다.

### 솔루션 라이브러리에 새로운 솔루션 및 프로젝트가 추가됨

Mech-Vision 1.7.2 버전의 솔루션 라이브러리에 다음과 같은 새로운 솔루션 및 프로젝트가 추가되었습니다.

- 솔루션: 브레이크 디스크(단일 스테이션).
- 프로젝트: 약 상자, 일반 공작물 인식, 무작위로 쌓인 작은 작업물(볼트).



사전 설치된 솔루션이나 프로젝트를 사용해야 하는 경우 최신 버전의 소프트웨어를 사용하십시오.

## 기능 최적화

### 알고리즘 최적화

Mech-Vision 1.7.2 버전에서는 3D 매칭 스텝(“3D 상세 매칭”, “3D 근사 매칭” 등)을 최적화하여 스텝의 실행 속도를 향상시켰습니다.

## 스텝과 파라미터 이름 수정

Mech-Vision 1.7.2 버전에서는 아래 스텝과 파라미터의 이름을 수정했습니다.

	Mech-Vision 1.7.2 및 이전 버전	Mech-Vision 1.7.2
스텝 이름	2D 포즈를 사용하여 3D포즈를 조정하기	3D 포즈를 2D 포즈로 전환하기
파라미터 이름	지정축의 각도에 따라 후보 포즈를 필터링하기("3D 근사 매칭 V2", "3D 상세 매칭" 등 스텝)	모델 회전 각도로 포즈 필터링

## Mech-Vision 1.7.1 업데이트 설명

이 부분에서는 Mech-Vision 1.7.1 버전 복구된 문제에 대해 소개하겠습니다.

### 문제 복구

Mech-Vision 1.7.1 버전에서는 다음 문제를 복구했습니다:

- ".vis" 프로젝트 파일을 두 번 클릭하여 Mech-Vision을 열 때 "인터페이스 서비스"가 시작하지 못하는 문제를 복구했습니다.
- 네트워크 지연으로 인해 Mech-Vision이 온라인 사용자 매뉴얼로 이동하지 못하는 문제를 복구했습니다.
- "카메라에서 이미지를 캡처하기" 스텝을 사용하여 LSR, DEEP 시리즈 카메라를 연결하고 "덱스 맵 좌표계로 수정하기" 파라미터를 선택하지 않은 경우 컬러 포인트 클라우드의 표시에 오류가 있는 문제를 복구했습니다.
- "카메라에서 이미지를 캡처하기" 스텝의 이름을 변경한 후 가상 모드에서 "데이터 경로"를 선택하면 적용되지 않는 문제를 복구했습니다.
- "3D 공작물 인식 시각화 구성 도구"의 작업물 라이브러리에서 작업물을 선택한 후 선택을 취소하고 다시 작업물 라이브러리에 들어가면 해당 작업물이 여전히 선택된 상태로 유지되는 문제를 복구했습니다.
- "3D 공작물 인식 시각화 구성 도구"를 열고 프로젝트 실행 중에 직접 종료하면 소프트웨어가 충돌할 수 있는 문제를 복구했습니다.
- "3D 상세 매칭 (라이트 버전)" 스텝에서 면 매칭 결과에 오류가 있는 문제를 복구했습니다.
- "매칭 모델 및 픽 포인트 편집기" 도구에서 포즈의 매니플레이터 유형을 전환한 후 매니플레이터의 좌표축이 두꺼워지는 문제를 복구했습니다.
- 3D ROI 편집 창이 열려 있는 상태에서 Mech-Vision을 종료할 때 3D ROI 편집 창이 정상적으로 닫히지 않는 문제를 복구했습니다.
- 캘리브레이션 중에 카메라에서 프레임을 놓치거나 카메라 연결이 끊어지면 소프트웨어가 충돌하는 문제를 복구했습니다.
- 로봇이 연결된 상태에서 카메라 외부 파라미터를 캘리브레이션해야 하는 경우, 캘리브레이션 방법 및 로봇 제어 방법을 선택하기 전에 로봇 연결을 끊어야 하는 문제를 복구했습니다.

## Mech-Vision 1.7.0 업데이트 설명

이 부분에서는 Mech-Vision 1.7.0 버전의 추가된 기능, 최적화된 기능 및 복구된 문제에 대해 소개하겠습니다.

## 기능 추가

"솔루션" 추가, Mech-Vision을 통해 비전 응용 프로그램 배포를 일괄적으로 처리할 수 있음

Mech-Vision 1.7.0 버전에서 "솔루션" 기능이 추가되었습니다. 프로젝트는 로봇 및 통신, 시각 처리, 경로 계획 등 기능 구성 및 데이터 모음으로 비전 응용 프로그램을 구현하는 데 필요합니다.

솔루션은 하나 이상의 프로젝트로 구성되며, 프로젝트를 단독적으로 사용하는 것은 권장되지 않습니다. 프로젝트에 속해야 합니다.

솔루션 기능을 지원하면 사용자는 하나의 솔루션에서 로봇 선택, 로봇 통신 구성, 비전 프로젝트 구성 및 로봇 경로 계획을 일괄적으로 처리할 수 있습니다.



- 비전 결과 및 간단한 경로 계획 프로젝트 요구 사항의 경우 사용자는 Mech-Vision 소프트웨어만 사용하여 모든 응용 프로그램 배포를 완료할 수 있습니다.
- 표준 인터페이스 또는 Adapter를 통해 통신하는 경우, 로봇 프로그램이 호출하는 Mech-Vision 번호는 Mech-Center가 아닌 Mech-Vision의 프로젝트 리스트에서 획득됩니다.
- 소프트웨어를 업그레이드한 후 이전에 구축한 Mech-Vision 프로젝트를 이전해야 합니다. 상세한 설명은 [1.7.0 프로젝트 업그레이드 가이드](#) 내용을 참조하십시오.

### Mech-Vision은 Mech-Center의 통신 서비스 기능을 통합함

Mech-Vision 1.7.0 버전에서 "로봇 및 인터페이스 통신" 기능이 추가되어 로봇을 가져오고 선택할 수 있으며 Mech-Center 소프트웨어를 열 필요가 없습니다. Mech-Vision 소프트웨어 내에서 통신 구성을 설정한 후 Mech-Vision과 로봇 및 기타 외부 장치 간의 통신을 쉽게 할 수 있습니다.

Mech-Vision 1.7.0 버전에서 인터페이스 통신 서비스 기능이 통합되었으므로 다음 사항에 유의하십시오:

- Mech-Vision 솔루션을 사용하고 표준 인터페이스 또는 Adapter를 통해 통신하는 경우 Mech-Center 소프트웨어를 사용할 필요가 없습니다.
  - Mech-Vision 소프트웨어의 툴 바에서 [ **로봇 및 인터페이스 구성** ] 버튼을 클릭하여 해당 로봇을 선택하고 통신 방식을 설정한 후 “응용” 버튼을 클릭하면 인터페이스 서비스가 자동으로 활성화됩니다.
  - 로봇 브랜드가 ABB, FANUC, YASKAWA, KAWASAKI, KUKA, UR, TM, ELITE, JAKA인 경우 기본적으로 선택된 통신 방식은 Mech-Mind에서 제공되는 로봇 측의 표준 인터페이스 샘플 프로그램에서 사용하는 통신 방식과 일치합니다. 다른 브랜드의 로봇인 경우, 기본적으로 TCP/IP 통신 방식을 선택합니다.
  - 인터페이스를 구성할 때 “솔루션을 열 때 인터페이스 서비스 자동 활성화”를 선택할 수 있습니다. 다음에 해당 솔루션을 열면 프로젝트가 로봇 측 프로그램과 직접 통신할 수 있습니다.
  - 통신과 관련된 로그는 Mech-Vision 로그바의 Console 탭에서 확인할 수 있습니다.
  - Adapter 인터페이스를 구성할 때 Adapter 프로젝트 폴더를 선택하면 폴더가 현재 프로젝트 폴더에 자동으로 복사됩니다.
  - 솔루션의 Adapter 프로그램을 수정하고 적용하면 Adapter 프로그램과 인터페이스 서비스를 재시작해야 합니다.
  - Mech-Center 소프트웨어의 “PC 부팅 시 Mech-Center 자동으로 실행하기” 기능을 비활성화해야 합니다.
- 통신 방식이 마스터 컨트롤 통신이거나 “솔루션”을 사용하지 않을 때 Mech-Vision을 사용하는 경우, 여전히 Mech-Center 소프트웨어를 사용해야 합니다.
  - 사용자는 먼저 Mech-Center 소프트웨어를 연 다음 Mech-Vision 소프트웨어를 열어 이전 버전에서 설명한 대로 통신을 구성해야 합니다.
  - Mech-Center 소프트웨어의 “PC 부팅 시 Mech-Center 자동으로 실행하기” 기능을 활성화하여 Mech-Center 소프트웨어를 통해 Mech-Vision 소프트웨어를 시작하는 것을 권장합니다.

### "경로 계획" 고급 구성 요소 추가

Mech-Vision 1.7.0 버전에 "경로 계획" 스텝이 추가되어 비전 포인트를 입력하여 로봇 경로를 계획하고 충돌하지 않는 로봇 경로를 출력할 수 있습니다. 금속 부품의 머신 텐딩 시나리오에 적합하며, 비교적 간단한 경로 계획 요구 사항을 충족시킬 수 있습니다.

### "3D 공작물 인식" 고급 구성 요소 추가

Mech-Vision 1.7.0에 "3D 공작물 인식" 스텝이 추가되어 포인트 클라우드 사전 처리, 3D 매칭, 겹친 물체 제거 등 비전 처리 기능을 통해 빠르게 제품 인식을 완료할 수 있습니다. 금속 부품 머신 텐딩 시나리오에 적용되며, 다양한 형태 및 배치 방식의 작업물을 인식할 수 있습니다.

### "시작 화면" 추가

Mech-Vision 1.7.0에 "환영 화면"이 추가되어 소프트웨어 버전 정보와 도움말, 관련 작업에 대한 빠른 액세스를 제공합니다.

### "솔루션 라이브러리" 추가

Mech-Vision 1.7.0에 "솔루션 라이브러리"가 추가되어 공작물 로드, 팔레타이징& 디팔레타이징, 로케이팅& 어셈블리, 부품 피킹, 품질 검사 등 5개 산업 분야의 응용 프로그램 또는 엔지니어링 사례 및 예제 데이터가 포함되어 있습니다. 초보자는 예시 그림과 간단한 설명을 참조하여 일치하는 응용 프로젝트를 찾아 간단한 수정 후 직접 배포하여 사용할 수 있습니다.

Mech-Mind는 온라인 솔루션 라이브러리를 지속적으로 확장할 예정이며, 사용자는 최신 솔루션 라이브러리 데이터를 한번 클릭하여 필요에 따라 다운로드하여 사용할 수 있습니다.

### "일반 공작물 피킹" 솔루션 템플릿 추가

Mech-Vision 1.7.0에서 솔루션 라이브러리에 "일반 공작물 피킹" 슈퍼 솔루션 템플릿이 내장되어 있습니다. 이는 다른 모델 및 배치 방식의 작업물을 인식하고 로봇이 "제로 충돌" 제품 피킹을 완료할 수 있도록 지원합니다. 사용자는 4 단계만으로 3D 비전 솔루션을 구축할 수 있습니다. 기계 가공 머신 텐딩 및 핸들링 등 응용 시나리오에 적용됩니다.

### 알고리즘 최적화

Mech-Vision 1.7.0에서 알고리즘을 최적화했으며, 몇 가지 스텝이 추가되었습니다. 구체적으로 다음과 같습니다.

새로 추가한 스텝	설명
3D 상세 매칭 (라이트 버전)	이 스텝은 "3D 상세 매칭" 스텝의 사용하기 쉬운 버전으로 파라미터 조정이 보다 직관적이며 포인트 클라우드 템플릿을 시나리오 포인트 클라우드와 정확하게 매칭시켜 대상 물체 포즈를 출력할 수 있습니다.
겹친 물체를 제거하기 V2	이 스텝은 사용자가 지정한 규칙에 따라 겹친 물체의 비전 인식 결과를 제거할 수 있습니다. "겹친 물체를 제거하기" 스텝에 비해 해당 스텝은 투영 방식의 처리 속도를 높입니다.
대형 부품 측정과 관련된 프로시저	대형 부품 측정 산업을 위한 공통 기능 프로시저를 제공하여 현장에서 간단한 측정 프로젝트를 신속하게 구축할 수 있습니다.
픽 포인트를 예측하기 V2	이 스텝은 2D 맵과 덤스 맵을 통해 피킹된 물체를 식별할 수 있으며 해당 픽 포인트를 출력할 수 있다.
법선 방향 계산 및 포인트 클라우드의 가장자리 추출	이 스텝은 법선 방향을 계산하고 포인트 클라우드에서 물체 에지를 추정하여 에지 포인트 클라우드를 출력합니다.

새로 추가한 스텝	설명
포인트 클라우드 변환(트러스)	이 스텝은 입력한 포인트 클라우드를 카메라 또는 트러스 로봇 좌표계로 변환한 다음 출력할 수 있습니다.
포즈 변환(트러스)	이 스텝은 입력한 포즈를 카메라 또는 트러스 로봇 좌표계로 변환한 다음 출력할 수 있습니다.
피팅 서클	이 스텝은 측정 시나리오에 적용되며 일반적으로 입력된 2D 맵 속의 점을 원으로 피팅할 수 있습니다.
직선 피팅	이 스텝은 일반적으로 측정 시나리오에 적용되며 입력된 2D 맵 속의 포인트를 직선으로 피팅할 수 있습니다.
두 라인 세그먼트 사이의 교차점 계산	이 스텝은 두 라인 세그먼트 사이의 교차 위치를 계산합니다. 일반적으로 측정 시나리오에서 물체 특징점을 찾는 데 사용됩니다.
라인 세그먼트와 원 사이의 교차점 계산	이 스텝은 두 라인 세그먼트의 교점 위치를 계산합니다. 일반적으로 측정 시나리오에서 물체의 특징점을 찾는 데 사용됩니다.
2D 형상 처리	이 스텝은 이진 이미지를 입력하고 지정된 방법대로 이미지 중의 모양을 처리합니다. 일반적으로 다양한 계산을 위해 측정 시나리오에서 물체 윤곽을 처리하는 데 사용됩니다.
홀 채우기	이 스텝은 입력한 이진 이미지의 홀, 즉 0이 아닌 픽셀로 완전히 둘러싸인 영역을 채우는 데 사용됩니다. 이 스텝은 일반적으로 이미지 분할에 사용되며 대상 영역의 완전한 이미지를 획득하고 누락된 홀 이미지로 인한 간섭을 피합니다.
이미지 선명도 평가	이 스텝은 지정된 방법에 따라 입력한 이미지의 선명도를 계산합니다. 방법마다 선명도의 수치 표준과 척도가 다릅니다. 일반적으로 측정 시나리오에서 카메라 파라미터와 위치를 조정하는 데 도움이 됩니다.
컬러 정보 통계하기	컬러 맵을 입력하고 색공간을 선택하면 이 스텝은 평균값, 표준 편차, 최대값, 최소값을 포함하여 지정된 채널의 픽셀 값을 계산할 수 있습니다. 이 스텝은 일반적으로 측정 시나리오에서 이미지 색상을 평가하는 데 사용됩니다.
캘리퍼스 도구	이 스텝은 길고 좁은 ROI의 수직 방향을 따라 에지 포인트 또는 에지 포인트 쌍을 감지하고 에지 포인트 좌표와 포인트 쌍 사이의 거리를 출력하는 데 사용할 수 있습니다.
Shape2DList를 Pose2DList로 전환하기	이 스텝은 입력한 3개의 2D 모양 정보 리스트에서 X 값 리스트, Y 값 리스트, Theta 값 리스트(경사각)을 가져와 새로운 2D 포즈 리스트로 결합합니다.

### LNX 카메라 적용

“라인 스캐닝 레이저 카메라” 스텝에서 LNX 카메라 유형이 추가되어 LNX 카메라를 적용할 수 있습니다.

### “트러스 캘리브레이션” 프로세스 추가

Mech-Vision 1.7.0은 캘리브레이션 도구를 최적화하고, “트러스 캘리브레이션” 프로세스를 추가하여 트러스 로봇에 대한 특수 가이드 캘리브레이션 프로세스를 개발하였습니다. 복잡한 설정 없이도 트러스 로봇을 캘리브레이션할 수 있습니다.

동시에, “포인트 클라우드 변환(트러스)” 및 “포즈 변환(트러스)”와 같은 트러스 로봇 전용 스텝들이 추가되어 Mech-Vision 프로젝트에서 동적 외부 파라미터를 계산하는 데 사용됩니다.

### “딥 러닝 모델 패키지 관리 도구” 추가

Mech-Vision 1.7.0은 “딥 러닝 모델 패키지 추론 (Mech-DLK 2.2.0+)” 및 “딥 러닝 모델 패키지 CPU 추론” 스텝에서 사용되는 딥 러닝 모델 패키지를 최적화하고, 실행 모드, 하드웨어 유형 및 모델 패키지 상태를 관리할 수 있는 “딥 러닝 모델 패키지 관리 도구”를 추가하였습니다. 또한 이 도구는 IPC의 GPU 사용률을 모니터링할 수 있습니다.

### 기능 최적화

#### 알고리즘 최적화

Mech-Vision 1.7.0은 여러 스텝을 최적화하였습니다. 구체적으로는 다음과 같습니다.

최적화된 스텝	설명
딥 러닝 모델 패키지 추론(Mech-DLK2.2.0+)	<b>모델 패키지 설정 및 결합 판정 규칙 구성</b> (결함 세그멘테이션 응용 시나리오에 적용됨) 파라미터 그룹이 새로 추가되었습니다.
딥 러닝 모델 패키지 CPU 추론	<b>모델 패키지 설정</b> 파라미터 그룹이 새로 추가되었습니다.
통계 데이터를 통해 포즈의 반복 정밀도를 체크하기	<b>이상값 처리</b> 옵션 추가. 수치 편차가 설정한 역치를 초과할 경우 <b>이상치가 발견되면 즉시 오류 보고</b> 또는 <b>이상값 기록 및 표시</b> 두 가지 처리 방식을 선택할 수 있습니다. <b>이상값 기록 및 표시</b> 방식을 선택하면 출력된 텍스트 파일의 이상값이 빨간색으로 표시됩니다.
픽셀 단위 길이를 실제 길이로 변환하기	<b>캘리브레이션에 의한 계산</b> 옵션이 새로 추가되며 입력된 이미지를 사용하여 자동으로 캘리브레이션을 수행하고 획득한 카메라 파라미터를 기반으로 하여 실제 거리를 계산할 수 있습니다. 계산 결과는 캘리브레이션을 수행하지 않은 것보다 더 정확합니다.
이미지 역치 처리	<b>DualThreshold</b> 및 <b>DynamicThreshold</b> 유형이 새로 추가되었습니다.
배열 및 층 나누기	<b>각 층의 시작점</b> 파라미터가 새로 추가되며 층 간격과 정렬할 포즈의 배열을 입력하여 정렬된 배열과 인덱스를 반환합니다.

#### 삭제된 스텝

Mech-Vision 1.7.0 버전에서 아래와 같은 스텝들을 삭제했습니다.

클래스	삭제한 스텝
2D 특징 추출 - 기타	구석점과 사이즈에 근거하여 직사각형을 감지하기
2D 통용 처리 - 기타	다각형 배치하기
딥 러닝 - 임의의 물체 피킹	확률 히트맵에서 포즈를 계산하기, 픽 포인트를 예측하기(단일 물체 유형)(새로운 스텝 “픽 포인트를 예측하기 V2”를 사용하세요), 픽셀당 피킹 가능한 확률을 예측하기
측정 - 3D 길이/거리	점에서 두 평면의 교차점까지의 거리를 계산하기, 점에서 평면까지의 거리를 계산하기
구 버전	이미지에서 정보를 시각화하기(새로운 스텝 “이미지에서 정보를 시각화하기”를 사용하세요), 장애물에 근거하여 포즈를 조정하기
포즈 - 평행이동&방향 조정	장애물에 근거하여 목표점을 조정하기 V2

클래스	삭제한 스텝
기타	피킹 가능한 직사각형을 감지하기

### "매칭 모델 및 픽 포인트 편집기" 최적화

Mech-Vision 1.7.0 버전에서는 "매칭 모델 및 픽 포인트 편집기" 기능을 최적화했습니다.

- 주요 기능을 강조하여 메인 인터페이스를 최적화하고 사용 편의성을 향상시켰습니다.
- 툴 바의 레이아웃을 최적화하고 애니메이션 팁을 추가했습니다.
- 카메라로 포인트 클라운드를 캡처하는 프로세스를 최적화하고 엣지 포인트 클라우드 캡처 기능을 추가했습니다.
- "CAD 파일 도입"시 단위 선택 기능을 최적화했습니다.

### "모델 편집기(구버전)" 도구를 삭제했음

Mech-Vision 1.7.0에서 "모델 편집기(구버전)"가 삭제되었습니다. 포인트 클라우드 모델을 만들고 픽 포인트를 생성하려면 "매칭 모델 및 픽 포인트 편집기"를 사용하십시오.

### "6축 로봇 캘리브레이션 프로세스" 최적화

Mech-Vision 1.7.0 버전에서는 "6축 로봇 캘리브레이션 프로세스"를 최적화했습니다. 구체적으로 다음과 같습니다.

- 캘리브레이션 사전 설정에서는 로봇 선택, 프로그램 복제, 로봇 연결 등 기능이 추가되었으며, 캘리브레이션 사전 설정에서 로봇과의 연결을 완료할 수 있습니다.
- 로봇 시뮬레이션 인터페이스 (시나리오 뷰)가 추가되어 캘리브레이션 프로세스에서 설정한 캘리브레이션 경로와 로봇 위치를 실시간으로 볼 수 있으며 Mech-Viz 소프트웨어를 사용하지 않아도 됩니다.
- UI를 최적화하여 인터페이스 레이아웃이 더 합리적이며 사용자를 캘리브레이션하도록 더욱 명확하게 안내합니다.

### "파라미터의 레시피" 최적화

Mech-Vision 1.7.0은 "파라미터의 레시피" 기능을 최적화했습니다. [ **파라미터 업데이트** ] 버튼을 클릭하여 프로젝트의 파라미터를 현재 레시피에 한 번에 동기화할 수 있습니다.

### "프로젝트 편집 영역" 최적화

Mech-Vision 1.7.0은 프로젝트 편집 영역 상호 작용 디자인을 최적화하고 캔버스 그리드 및 정렬얼라인먼트 기능을 추가하여 사용자들은 자기 자신의 선호도에 따라 정렬 그리드가 표시되는지 여부를 설정할 수 있으며 참조선 및 스텝 드래그 앤 드롭 스냅을 조정할 수 있습니다.

### 카메라의 자체 정의한 이름을 표시할 수 있음

Mech-Vision에서 카메라에 연결한 후 마우스를 카메라 ID 위에 놓으면 카메라의 자체 정의한 이름과 IP 주소를 볼 수 있습니다.

### 전역 기본 길이 및 각도 단위 수정

전역 기본 길이 및 각도 단위가 "각 스텝 내장 단위 사용"에서 "mm" 및 "°"로 수정되었습니다.

## 온라인 및 오프라인으로 일본어와 한국어 언어 팩 업데이트 지원

Mech-Vision 1.7.0은 온라인으로 일본어와 한국어 UI 언어 팩을 자동으로 업데이트하는 기능을 지원합니다. 또한 Mech-Vision 1.7.0은 Mech-Mind 기술 지원에서 오프라인 일본어와 한국어 UI 언어 팩을 가져와 소프트웨어에 드래그하여 언어 팩을 업데이트 할 수 있습니다.

### 문제 복구

#### 스텝 "카메라에서 이미지를 캡처하기" 문제 복구

Mech-Vision 1.7.0은 하이크비전 2D 카메라에 연결할 수 없는 "카메라에서 이미지를 캡처하기" 스텝 문제를 복구했습니다.

#### 스텝 "Python을 통해 결과를 계산하기" 문제 복구

Mech-Vision 1.7.0은 "Python을 통해 결과를 계산하기" 스텝의 다음 문제를 해결했습니다.

- 프로젝트 실행 시이 스텝이 렉 걸리는 문제를 복구했습니다.
- PostList 데이터 유형 병합 시 출력 포즈의 사원수 순서가 잘못된 문제를 복구했습니다.

#### 스텝 "포즈 조정" 단계 문제 복구

Mech-Vision 1.7.0은 프로젝트 실행 시 "포즈 조정" 스텝으로 인해 발생하는 충돌 문제를 해결했습니다.

## 과거 버전 업데이트 설명

[Mech-Vision 1.6.2 업데이트 설명](#)

[Mech-Vision 1.6.1 업데이트 설명](#)

[Mech-Vision 1.6.0 업데이트 설명](#)

## 2.3. Mech-Viz 업데이트 설명

이 부분에는 Mech-Viz 1.7 버전의 추가된 기능, 최적화된 기능 및 복구된 문제에 대해 소개하겠습니다.

### Mech-Viz 1.7.4 업데이트 설명

이 부분에서는 Mech-Viz 1.7.4 버전 복구된 문제에 대해 소개하겠습니다.

#### 기능 최적화

##### 동글 권한 부여에 소프트웨어 지역 제한이 추가됨

소프트웨어 판매 지역에 따라 동글 권한 부여 시 소프트웨어를 사용하는 지역에 대한 제한이 추가되었습니다.

Mech-Viz 1.7.4 버전에서 [도움 > 버전 정보](#) 소프트웨어 라이선스 및 지역 제한을 순서대로 클릭하면 소프트웨어 라이선스 및 지역 제한을 확인할 수 있습니다.

##### “시스템 언어” 옵션이 제거됨

Mech-Viz 1.7.4 버전부터 소프트웨어 언어 옵션 ([설정 > 옵션 > 일반 설정 > 언어](#)) 에서 “시스템 언어”를

제거하였습니다. 컴퓨터 운영 체제의 언어가 소프트웨어에서 지원하는 언어(영어, 간체 중국어, 한국어 또는 일본어)가 아닌 경우, 소프트웨어가 처음으로 설치될 때 기본 언어는 영어로 설정됩니다.

해당 최적화는 다음 두 가지 문제를 해결했습니다.

- 이전 버전에서 소프트웨어가 현재 컴퓨터 운영 체제의 언어를 지원하지 않을 때, 소프트웨어 내부 링크를 클릭한 후 해당 페이지에 들어갈 수 없는 문제를 해결했습니다.
- 소프트웨어가 컴퓨터 언어에 따라 언어 패키지를 올바르게 설정할 수 없는 문제를 해결했습니다.

#### 기타

- “팔레타이징” 클래스에 속하는 스텝 파라미터의 “파렛트”는 “파렛트 위치”로 수정되었습니다.
- "비전 이동" 스텝과 관련된 로그 및 plan\_result에 상자 조합의 행 수 정보(box\_group\_row\_count)가 추가되었습니다.

#### 문제 복구

Mech-Viz 1.7.4 버전에서 다음과 같은 문제들을 복구했습니다.

- 시뮬레이션 공간 우클릭 메뉴 - 프로젝트 시뮬레이션 또는 실행 중에 물체 삭제 기능이 비활성화되지 않는 문제를 복구했습니다.
- 작업물 회전 대칭과 공구 회전 대칭을 동시에 사용할 때, 작업물 회전 대칭 시도가 잘못되는 문제를 복구했습니다.
- "메시지 분기" 스텝 이전에 이동 스텝이 없으면 실행을 건너뛸 때 해당 아웃포트를 통해 실행되지 않는 문제를 복구했습니다.
- 비전 결과를 재사용하면 비전 결과의 레이블이 누락되는 문제를 복구했습니다.
- 로그 항목을 드래그하여 드롭하면 소프트웨어가 종료될 수 있는 문제를 복구했습니다.
- "비전 이동" 스텝에서 "총수 제한"을 설정한 후 모든 가능한 조합 결과를 제공하지 않는 문제를 복구했습니다.
- "비전 이동" 스텝을 사용하여 계획된 진공 그리퍼의 DO가 잘못될 수 있는 문제를 복구했습니다.
- "비전 연속 팔레타이징" 스텝을 사용하여 파렛트 패턴을 일치시킬 때 방향이 잘못 매칭될 수 있는 문제를 복구했습니다.
- 작업물 모델로 사용자 정의 모델을 사용할 때 충돌 감지를 놓칠 수 있는 문제를 복구했습니다.

### Mech-Viz 1.7.2 업데이트 설명

이 부분에서는 Mech-Viz 1.7.2 버전의 새로 추가된 기능, 최적화된 기능 및 복구된 문제에 대해 소개하겠습니다.

#### 기능 최적화

##### 로봇은 모든 계획된 포인트를 시뮬레이션함

이전 버전에서는 이동 스텝의 "웨이포인트를 전송하기" 파라미터를 선택하지 않으면 이 웨이포인트는 경로 계획에만 참여하고 시뮬레이션 로봇은 이 웨이포인트까지 실행하지 않았습니다.

Mech-Viz 1.7.2 버전에서는 기본적으로 시뮬레이션 로봇은 모든 계획된 웨이포인트의 위치를 도달할 것입니다.

여전히 이전 버전의 상태를 유지해야 하는 경우 설정 > 옵션 > 기본 설정에서 조정하십시오.

### 로봇을 도출할 때 STL 및 DAE 소스 파일을 유지함

이전 버전에서는 로봇을 도출하면 해당 로봇의 STL 및 DAE 소스 파일이 자동으로 삭제되었습니다.

Mech-Viz 1.7.2 버전에서는 로봇을 도출할 때 STL 및 DAE 소스 파일을 유지합니다.

### “측면 빨판” 모드를 삭제했음

이전 버전에서는 "비전 이동" 스텝 및 "말단장치"에 "측면 빨판"을 구성하여 말단장치에 측면 빨판이 있는 디팔레타이징 애플리케이션을 해결할 수 있었습니다.

Mech-Viz 1.7.0 버전 이후 “디팔레타이징 빨판” 기능은 "측면 빨판" 모드의 모든 기능을 완전히 포함했습니다.

따라서 Mech-Viz 1.7.2 버전에서 "측면 빨판" 모드를 삭제했고 이전에 "측면 빨판" 모드를 구성한 사용자들은 “디팔레타이징 빨판” 기능을 사용하여 다시 구성해야 합니다.

### 18가지 유형의 오일러 각을 모두 보충했음

이전 버전에서는 자주 사용되는 5 가지 오일러 각 유형만 지원했습니다. Mech-Viz 1.7.2 버전에서는 나머지 13 가지 유형의 오일러 각을 모두 보충했습니다.

### “메시지 분기” 스텝의 파라미터 “계획 후 미리 실행”을 삭제했음

Mech-Viz 1.7.2 버전에서 “메시지 분기” 스텝의 파라미터 “계획 후 미리 실행”을 삭제했습니다.

### "Scara" 로봇 소프트 리미트 복구

이전 버전에서는 "Scara" 로봇의 소프트 리미트 알고리즘에 오류가 있어서 J3 소프트 리미트에 대한 설정 사항을 저장하지 못했습니다. Mech-Viz 1.7.2 버전에서 이 문제를 복구했지만 로봇 J3 소프트 리미트를 변경할 수 있으므로 수동으로 조정해야 합니다.

## 문제 복구

Mech-Viz 1.7.2 버전에서는 다음과 같은 문제들을 복구했습니다:

- 프로젝트 리소스에서 말단장치의 투명도 상태는 시뮬레이션 공간의 투명도 상태와 일치하지 않을 수 있는 문제를 복구했습니다.
- 시나리오에서 빈의 "포즈의 유효 범위를 설정하기" 기능이 적용되지 않는 문제를 복구했습니다.
- “스마트 장애물 회피” 기능을 사용할 때 소프트웨어가 충돌할 수 있는 문제를 복구했습니다.
- “배열대로 이동” 스텝을 사용하여 한번에 모든 웨이포인트를 실행할 때 웨이포인트를 놓치는 문제를 복구했습니다.
- "미리 설정된 파렛트 패턴" 스텝에서 상자의 크기를 조정 후, 파렛트 패턴의 변경에 따라 팔레타이징의 순서가 자동으로 조정되지 않는 문제를 복구했습니다.
- "혼합 팔레타이징" 스텝을 사용할 때 기존 상자의 위치에 계속 쌓이는 문제를 복구했습니다.
- "비전 연속 팔레타이징" 스텝에서 파렛트 소스를 이미 선택했지만 여전히 선택 항목이 없다는 알람이 표시되는 문제를 복구했습니다.
- 상자 디팔레타이징 알고리즘은 빨판의 간격도 유효한 빨판 영역으로 간주하는 문제를 복구했습니다.
- 개별 상자의 레이블은 상자 조합 이후의 비전 결과 출력에 출력되지 않는 문제를 복구했습니다.

## Mech-Viz 1.7.1 업데이트 설명

이 부분에서는 Mech-Viz 1.7.1 버전의 최적화된 기능 및 복구된 문제에 대해 소개하겠습니다.

## 기능 최적화

### “비전 이동” 스텝의 파라미터 “장애물 거리 역치”의 기본값을 40mm로 수정했음

이전 버전에서 “비전 이동” 스텝의 파라미터 “장애물 거리 역치”의 기본값은 20mm이었는데 이 설정 값은 일부 작업 장 응용 프로그램에 대해 너무 엄격하여 계획 중에 피킹할 수 있는 많은 픽 포인트를 포기하게 됩니다.

Mech-Viz 1.7.1 버전에서는 “비전 이동” 스텝의 파라미터 “장애물 거리 역치”의 기본값을 40mm로 수정했습니다.

## 문제 복구

Mech-Viz 1.7.1 버전에서는 다음 문제를 복구했습니다:

### 프로젝트 소스 - 로봇

- 로봇 소프트 리미트를 수정하면 소프트웨어가 충돌 할 수 있는 문제를 복구했습니다.
- FANUC R-2000iC / 270F 로봇 시각화 모델의 오류를 복구했습니다.
- 소프트웨어 실행 상태에서 로봇 패널에서 "TCP" 및 "관절 각도"를 전환 할 수 없는 문제를 복구했습니다.

### 프로젝트 소스 - 말단장치

- 말단장치 구성 창을 열 때 여전히 툴 바를 조작할 수 있는 문제를 복구했습니다.
- 말단장치 구성의 "TCP 캘리브레이션" 창에서 "포즈 추가"를 클릭 한 후 "로봇 플랜지 포즈를 입력하기" 창이 올바르게 표시되지 않는 문제를 복구했습니다.
- 말단장치 구성 인터페이스에서 취소를 클릭하면 수정된 말단장치 유형도 저장되는 문제를 복구했습니다.

### 프로젝트 소스 - 작업물

- 작업물은 X 축 및 Y 축 대칭성을 동시에 구성 할 수 있는 문제를 복구했습니다.
- 작업물 구성 창을 열 때 여전히 툴 바를 조작할 수 있는 문제를 복구했습니다.
- 기본적인 작업물 구성이 삭제 될 수 있는 문제를 복구했습니다.

### 프로젝트 소스 - 시나리오 물체

- 시나리오 물체를 구성하면 소프트웨어가 충돌 할 수 있는 문제를 복구했습니다.
- 시나리오 물체 구성 창을 열 때 여전히 툴 바를 조작할 수 있는 문제를 복구했습니다.
- 빈이 충돌 감지에 참여하지 않으면 비전 포즈가 빈 안에 없다고 인식되는 문제를 복구했습니다.

## 스텝

- 두 개의 "상대적인 이동" 스텝이 서로 의존하면 소프트웨어가 무한 루프에 빠지는 문제를 복구했습니다.
- "비전 연속 팔레타이징" 스텝이 실행할 수 없는 문제를 복구했습니다..
- "다수 피킹 팔레타이징" 스텝을 재설정하면 이미 쌓인 상자가 비워지지 않는 문제를 복구했습니다.
- 로봇이 움직이지 않을 때 "피킹된 물체를 업데이트하기" 스텝은 피킹된 작업물 모델을 생성할 수 없는 문제를 복구했습니다.
- "피킹된 물체를 업데이트하기" 스텝을 사용하면 소프트웨어가 렉 걸리는 문제를 복구했습니다.

- "분류"는 "비전 이동" 스텝 전에 있고 "비전 이동"의 "같은 포즈로 피킹하지 않기" 기능을 활성화할 때 소프트웨어가 무한 루프에 빠질 수 있는 문제를 복구했습니다.
- "비전 이동"은 실패 출구에 다른 이동 스텝이 연결되어 있어도 여전히 오류 알람이 표시되는 문제를 복구했습니다.

#### 피킹 계획

- 디팔레타이징 알고리즘에서 "안전한 행을 조합하기" 옵션을 선택할 때 Y 축 방향으로 조합되지 않는 문제를 복구했습니다.
- 디팔레타이징 알고리즘에서 작업물이 Y 축 방향으로 조합되면 빨판 옵션이 잘못되는 문제를 복구했습니다.
- "다수 피킹 팔레타이징" 스텝은 부동 소수점 오차로 인해 파렛트 패턴을 일치시키지 못하는 문제를 복구했습니다.
- "다수 피킹 팔레타이징" 스텝에서 솔루션을 선택 시 상자 수가 더 많은 솔루션을 우선적으로 시도하지 않는 문제를 복구했습니다.

#### 소프트웨어 빌트인 도구

- "사용자 정의 파렛트 패턴 편집기"를 사용하면 충돌이 발생하는 문제를 복구했습니다.
- "어레이 그리퍼 구성기"를 사용할 때 구성을 저장할 수 없는 문제를 복구했습니다.
- 비전 기록 인터페이스에서 비전 항목을 올바르게 로드할 수 없는 문제를 복구했습니다.
- "자체 정의한 파렛트 패턴" 스텝의 "파렛트 패턴 편집기"를 사용하여 기존 파렛트 패턴의 구성을 수정한 후 저장할 수 없는 문제를 복구했습니다.

#### 계획 기록

- 계획 기록 1 차 항목에 나열된 이동 스텝이 누락되는 문제를 복구했습니다.
- 로봇이 도달할 수 없어 계획이 실패 할 때, 말단장치 모델이 올바르게 표시되지 않는 문제를 복구했습니다.

#### 기타

- 로봇이 급정지하면 소프트웨어가 반복해서 오류 창을 표시하는 문제를 복구했습니다.

## Mech-Viz 1.7.0 업데이트 설명

이 부분에서는 Mech-Viz 1.7.0 버전의 추가된 기능, 최적화된 기능 및 복구된 문제에 대해 소개하겠습니다.

### 기능 추가

#### "시작 화면"이 새로 추가됨

소프트웨어를 시작하거나 프로젝트를 닫은 후 "시작 화면"이 나타납니다. "시작 화면"에서 프로젝트를 새로 만들거나 최근에 사용했던 프로젝트를 다시 열 수 있습니다.

#### "프로젝트 리소스" 영역이 새로 추가됨

Mech-Viz 1.7.0 버전에서는 이전 버전의 "시나리오 물체" 및 "공구와 작업물" 기능 패널을 제거하고 프로젝트에 필요한 리소스를 재구성하고 새로 추가된 프로젝트 리소스 관리 도구 "프로젝트 리소스"에 통합했습니다.

"프로젝트 리소스" 영역에 프로젝트, 로봇, 좌표계, 말단장치, 작업물, 바닥, 시나리오 물체와 모델 라이브러리 등 리소스가 포함되어 있습니다.

- 프로젝트: 프로젝트의 이름, 프로젝트에 대한 수정 사항, 해당 프로젝트가 자동으로 로드되는지를 표시합니다.
- 로봇: 현재 프로젝트에 사용된 로봇 모델을 표시합니다.
- 좌표계: 소프트웨어 시뮬레이션 공간에 있는 모든 좌표계를 표시합니다.
- 말단장치: 말단장치를 추가하거나 삭제하고 기존 말단장치를 관리합니다.
- 작업물: 작업물에 대한 구성을 설정하거나 삭제하고 기존 작업물을 관리합니다.
- 바닥: 시뮬레이션 공간에 있는 바닥의 높이를 조정합니다.
- 시나리오 물체: 시나리오 물체를 추가하거나 삭제하고 기존 시나리오 물체를 관리합니다.
- 모델 라이브러리: 프로젝트에 사용되는 모든 모델을 관리하며 모든 모델을 모델 라이브러리에 도입해야 사용될 수 있습니다.

#### "현재 작업 도구로 설정하기" 옵션이 새로 추가됨

소프트웨어에 “현재 작업 도구”라는 새로운 개념이 추가되었습니다. "현재 작업 도구"로 설정된 말단장치는 다음 시뮬레이션/실제로 실행할 때 계획 시작의 도구로 사용됩니다.

- 시뮬레이션 또는 프로젝트를 실행하지 않은 상태에서 프로젝트 리소스 패널에 말단장치의 이름을 마우스 오른쪽 버튼으로 클릭하고 팝업창에서 "현재 작업 도구로 설정하기"를 선택하면 "현재 작업 도구"를 수정할 수 있습니다.
- 시뮬레이션 또는 실제로 프로젝트를 실행할 때 "말단장치를 바꾸기" 스텝을 통해 “현재 작업 도구”를 수정할 수 있습니다.

#### "미리 설정된 파렛트 패턴" 스텝에 파렛트 패턴을 자동으로 생성하는 기능이 추가됨

"미리 설정된 파렛트 패턴" 스텝을 사용하면 파렛트 사이즈, 상자 치수, 간격 넓이, 파렛트 높이 등 제한에 따라 자동으로 계산하여 파렛트 수량이 가장 많고, 홀/짝수 총 패턴의 합리적이며 총수가 가장 많은 파렛트 패턴을 생성할 것입니다.

#### 기타

- 메뉴 바 "파일"에 "새로운 프로젝트" 버튼이 새로 추가되었습니다.
- ROKAE SR3, SR4 로봇 모델이 새로 추가되며 해당 특별한 로봇의 구성도 함께 개발되었습니다.
- 단축키 Ctrl + G를 누르면 한꺼번에 선택한 여러 스텝을 하나의 "프로시저"로 합칠 수 있으며 Ctrl + Shift + G를 누르면 "프로시저"를 신속히 분할할 수 있습니다.
- "메시지 분기" 스텝에 "타임아웃" 아웃 포트가 추가되고 이 옵션을 선택하면 제한 시간을 설정할 수 있습니다. 신호를 기다리는 데 이 설정된 제한 시간을 초과하면 프로젝트가 타임아웃 아웃포트를 통해 자동으로 실행됩니다.
- "혼합 팔레타이징" 스텝을 오프라인에서 사용할 때 미리 계산된 파렛트 패턴을 도입하고 저장할 수 있습니다.
- "혼합 팔레타이징" 스텝에 상자 팔레타이징 방향을 지정하는 기능이 새로 추가되었습니다.
- "미리 설정된 파렛트 패턴" 스텝에 windmill2 파렛트 패턴이 새로 추가되었습니다.
- "미리 설정된 파렛트 패턴" 스텝에 작업물 방향을 설정하는 기능이 새로 추가되었습니다.

#### 기능 최적화

##### 로봇 모델 라이브러리 확장

온라인 로봇 모델 라이브러리는 214종의 로봇에서 600종 이상으로 증가했으며 ABB, Denso, FANUC, Kawasaki, KUKA, Mitsubishi, Nachi, Universal Robots, Stäubli, Yaskawa 등 거의 모든 로봇의 수집을 완료했습니다.

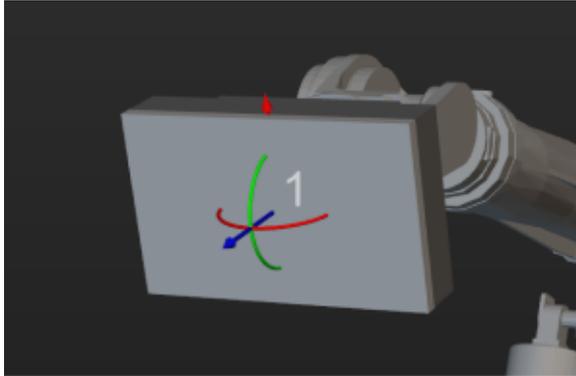
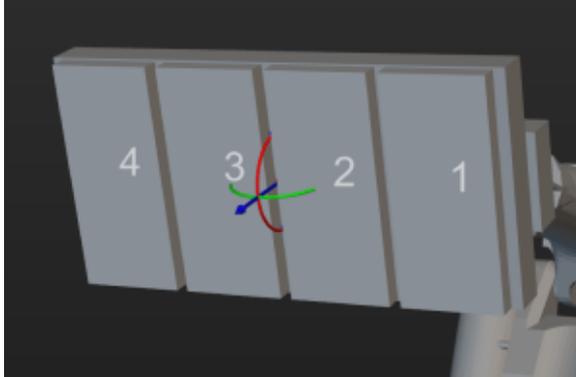
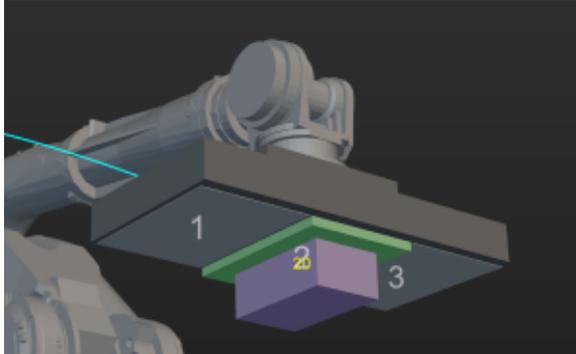
동시에 ABB, FANUC, Kawasaki, KUKA, Nachi, Universal Robots 및 Yaskawa의 로봇은 로봇 파라미터가 정확하고 신뢰할 수 있도록 생산 공정 중에 보정되었습니다.

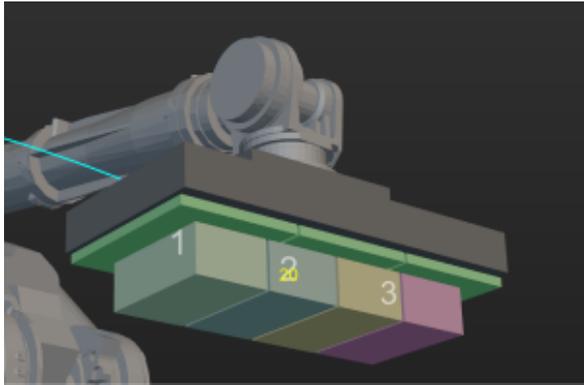
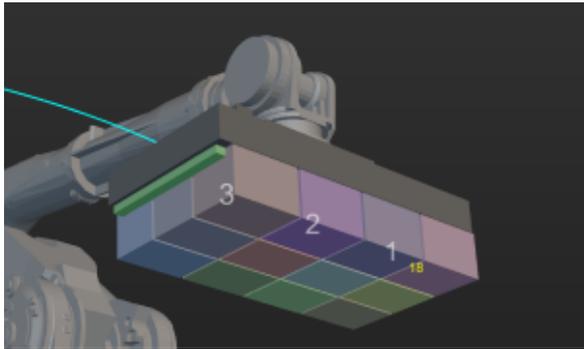
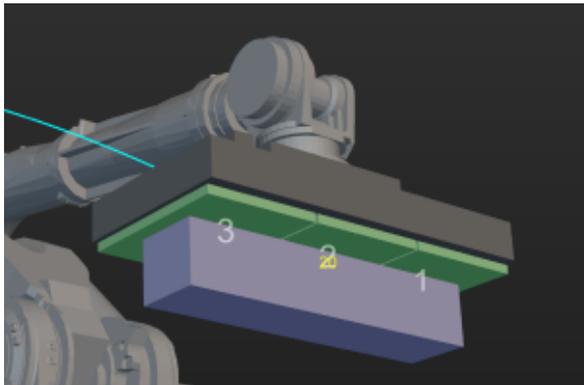
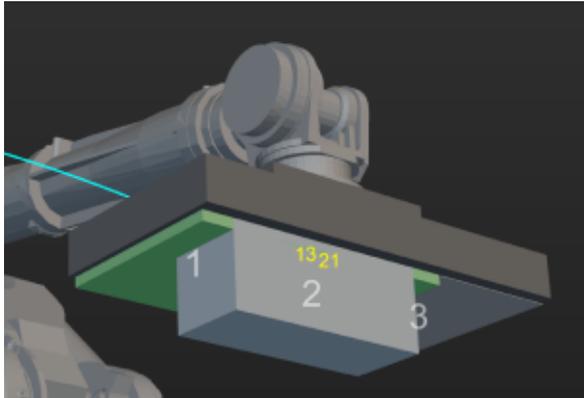
**STL 포맷의 말단장치 충돌 모델은 더 이상 지원되지 않음**

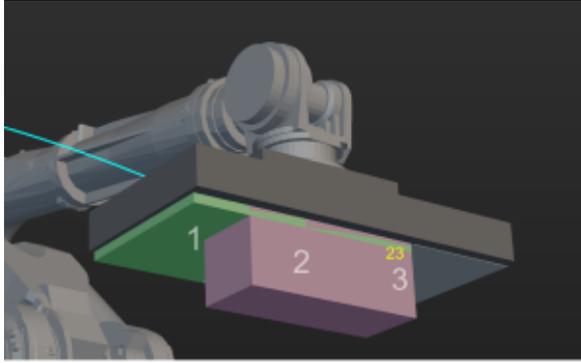
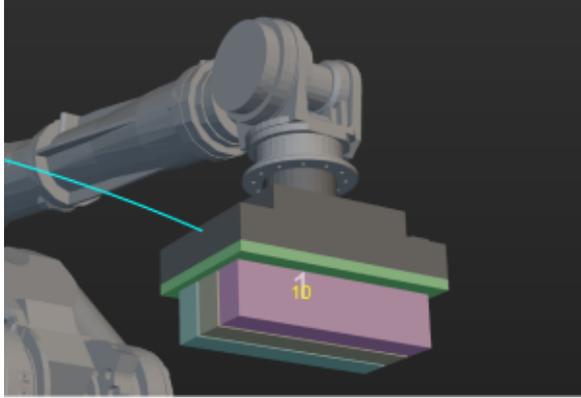
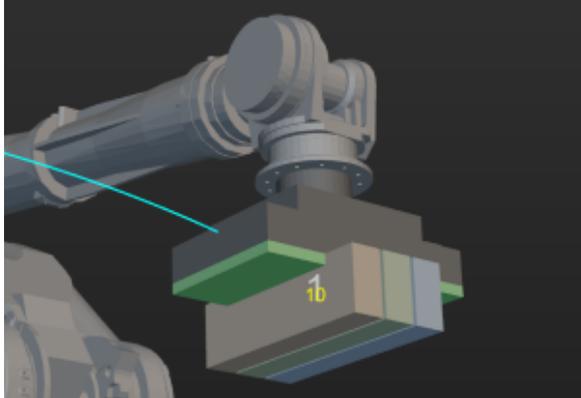
말단장치와 포인트 클라우드 사이의 충돌을 더 정확히 감지할 수 있도록 Mech-Viz 1.7.0 버전은 STL 포맷의 말단장치 충돌 모델을 지원하지 않습니다. 소프트웨어에서 지원하는 모델 포맷은 "프로젝트 리소스 모델에 관한 설명" 내용을 참조하십시오.

**"비전 이동" 스텝의 디팔레타이징 알고리즘 최적화**

Mech-Viz 1.7.0 버전에서는 "비전 이동" 스텝의 디팔레타이징 알고리즘을 최적화하고 재구성하여 다음과 같은 기능을 지원합니다.

단일 파티션 빨판	
멀티 파티션 빨판	
하나만 피킹하기	

<p>한 줄을 피킹하기</p>	
<p>여러 줄을 피킹하기</p>	
<p>빨판의 중심은 상자의 중심에 해당함</p>	
<p>빨판 측면의 중간점은 상자 측면의 중간점에 해당함</p>	

<p>빨판의 작은 상자의 각에 해당함</p>	
<p>빨판의 긴 쪽은 상자 조합의 긴 쪽과 평행함</p>	
<p>빨판의 긴 쪽은 상자 조합의 긴 쪽에 수직됨</p>	

또한 소프트웨어는 더 이상 TCP 좌표계의 X축이 빨판의 긴 쪽에 평행하거나 작업물 픽 포인트 좌표계의 X축이 상자의 긴 쪽에 평행하도록 제한하지 않습니다. 위의 모든 기능 간에는 결합 관계가 없으며 파라미터 다양한 피킹 전략을 구성할 수 있습니다.

#### 모델 편집기 최적화

- 모델 편집기에 "좌표계 설정" 기능이 새로 추가되었으며 이 기능을 사용하면 도출한 OBJ 모델이 로봇 플랜지 말단에 빠르고 정확하게 장착되도록 기존 모델의 좌표계 원점과 각 좌표축 방향을 설정할 수 있습니다.
- 모델 편집기의 프로젝트는 더 이상 파일을 생성하여 저장할 필요가 없이 .m3d 파일로 통합됩니다.
- "모델 편집기"에서 생성된 OBJ 모델은 Mech-Viz 1.6.0 이전의 소프트웨어 버전과 호환될 수 있습니다.

#### 오일러 각 표시 형식 최적화

로봇 브랜드마다 티치 펜던트에 오일러 각의 유형을 나타내는 기호가 다릅니다. 예를 들어 ABB는 EX, EY

및 EZ를 사용하고 KUKA는 A, B 및 C를 사용하고 Kawasaki는 O, A 및 T를 사용합니다.

Mech-Viz 1.7.0 버전에서는 일부 로봇 브랜드와 해당 오일러 각 유형 간의 대응 관계를 내장하며 오일러 각도를 나타내기 위해 이러한 브랜드에서 사용하는 특정 기호를 기록합니다. 사용자가 로봇을 선택한 후 소프트웨어는 프로젝트의 오일러 각 유형을 로봇에 해당하는 형식으로 자동으로 전환하고 브랜드에서 사용하는 기호로 인터페이스에 표시합니다.

#### 말단장치 구성 최적화

Mech-Viz 1.7.0 버전에서는 말단장치 구성 인터페이스의 기능을 통합하고 최적화했습니다. 새로운 말단장치 구성 인터페이스에는 말단장치의 유형 구성, 말단장치 제어 논리 구성 창이 포함되며 대칭성 구성 파라미터를 최적화했습니다.

#### 작업물 구성 최적화

Mech-Viz 1.7.0 버전의 새로운 작업물 구성 화면은 구 버전 소프트웨어에 산재해 있던 작업물 관련 구성을 통합하고 새로운 작업물 구성 화면에는 작업물 대칭성 구성, 피킹 허용 편차 구성, 솔루션 선택 전략 구성 등이 포함됩니다.

#### 시나리오 물체 구성 최적화

Mech-Viz 1.7.0 버전에서는 시나리오 물체를 새로 생성하는 화면과 시나리오 물체 구성 화면을 통합하며 빈 모델을 사용하는 경우 비전 포즈의 유효 범위를 설정하는 파라미터를 최적화했습니다.

#### 스텝 이름 수정

Mech-Viz 1.7.0 버전부터 이전의 “기능”과 “태스크 세트”는 각각 “스텝”과 “프로시저”로 이름이 변경되었습니다. 또한 Mech-Viz 1.7.0에서는 각 스텝의능을 보다 정확하게 나타내기 위해 아래 스텝이름을 수정하였습니다.

수정 전 이름	수정 후 이름
로봇 공구	로봇 유틸리티
TCP 체크	말단장치를 체크하기
TCP	말단장치를 바꾸기
비전 결과를 모두 써 버림	비전 결과를 모두 사용했음

기타

메뉴 바

- 메뉴 바 틀에 있는 "로봇에게 보낸 정보를 프린트 하지 않기" 옵션을 삭제했습니다. 즉 새버전 소프트웨어에서는 로봇에 보낸 정보를 항상 프린트합니다.
- 메뉴 바 틀에 있는 "디버그 파일을 생성하기(.dmp)" 옵션을 삭제하고 이 기능을 더 이상 지원하지 않습니다.
- 설정 > 옵션 에 있는 각도와 길이 단위의 기본값은 "각 스텝에 내장된 단위를 사용하기"에서 mm 및 °로 수정했습니다.

#### 프로젝트 리소스

- 툴 바에 있는 "프로젝트를 자동 로드하기" 옵션을 **프로젝트 리소스 > 프로젝트 > 마우스 오른쪽 버튼 클릭 메뉴 > 해당 프로젝트를 자동으로 로드하기로** 변경하였습니다.
- 빨판 구성은 말단장치 구성에 통합되어 별도의 빨판 구성 파일이 더 이상 보류되지 않습니다.
- 말단장치 구성, 작업물 구성 및 시나리오 물체 구성 화면이 열리면 메인 인터페이스의 시뮬레이션 공간의 시각 및 확대/축소를 동시에 조정할 수 있습니다.

#### 스텝

- 이동 스텝의 빠른 설정, 포즈 변환, 포즈 보정, 포즈 편집 버튼이 재정렬되었습니다.
- "DO 리스트 설정" 스텝의 포트 수 상한을 30000으로 설정했습니다.
- "비전 결과를 체크하기" 스텝의 "미완성" 아웃포트가 "타임아웃"으로 수정되었습니다.
- "비전 이동" 스텝 상자 다수 피킹 기능의 로그 내용을 최적화하고 상자 조합, 상자 움셋과 관련된 상세한 계획 내용을 추가했습니다.
- "시나리오 물체 업데이트" 스텝은 비전을 통해 업데이트할 때 "실패" 포트가 새로 추가됩니다.
- "비전 이동" 스텝을 통해 다수 피킹을 수행할 때 조합된 상자는 더 이상 하나의 큰 상자가 아니라 여러 개의 독립적인 작은 상자로 유지됩니다.
- "상대적인 이동"스텝에 있었던 "기준" 및 밑의 옵션들을 삭제했습니다.
- "RobotIQ 설정" 및 "로봇 함수 호출" 스텝을 삭제했습니다.
- "비전 이동" 스텝에 있었던 "실제 피킹한 포즈를 저장하기" 파라미터와 해당 기능을 삭제했습니다.

#### 로봇 패널

- 로봇 관절 각도 슬라이더를 드래그하면서 오른쪽에서 실시간으로 JPs 값을 표시됩니다.
- 로봇과의 일관성을 보장하기 위해 YASKAWA PL80, MPL80II, MPL100II 특수 정의 형식에 대한 로봇 관절 각도 표시를 최적화했습니다.
- 로봇 모델을 변경한 후 말단장치 리스트를 지우지 않습니다.
- 프로젝트에 필요한 로봇이 없는 경우 사용자가 로봇 라이브러리에서 수동으로 다른 로봇 모델을 선택할 수 있도록 지원합니다.

#### 충돌 모델

- 포인트 클라우드와의 충돌 감지의 ON/OFF에 관계없이 \"비전 이동\"으로 피킹된 작업물의 포인트 클라우드를 제거할 수 있습니다.
- 포인트 클라우드와의 충돌 감지의 ON/OFF에 관계없이 "비전 인식" 스텝을 리셋함으로써 포인트 클라우드를 리셋할 수 있습니다.

#### 소프트웨어 통신

- "기타" 패널에서 "비슷한 웨이포인트를 건너뛰기" 옵션이 추가되어 사용자가 이를 통해 두번째 웨이포인트를 전송할지 여부를 자체적으로 설정할 수 있습니다.

- 구버전 소프트웨어 "기타" 패널에 있었던 "TCP 포즈를 전송하기" 옵션을 삭제하고 항상 전송하는 것으로 설정됩니다.

## 문제 복구

Mech-Viz 1.7.0 버전에서는 다음 문제를 복구했습니다:

- 이전 프로젝트가 열릴 수 없을 때 일부 작업 흐름의 다이어그램이 여전히 로드되는 문제를 수정했습니다.
- 프로젝트가 잠긴 상태에서 비밀번호가 없어도 신규 사용자를 등록할 수 있는 문제를 수정했습니다.
- 가끔 프로젝트를 닫을 때 사용자에게 저장되지 않은 변경 사항에 대한 메시지가 표시되지 않는 문제를 수정했습니다.
- 시뮬레이션/실행 상태에서 좌표축을 드래그하여 시나리오 물체의 위치를 변경할 수 있는 문제를 수정했습니다.
- 시뮬레이션/실행 상태에서 로봇의 TCP 좌표축을 드래그하여 시뮬레이션 로봇을 드래그할 수 있는 문제를 수정했습니다.
- 로봇 설치 패키지 파일의 대문자로 인해 설치가 실패하는 문제를 수정했습니다.
- 3D 시뮬레이션 공간의 TCP 좌표계가 새로운 말단장치를 추가할 때 구성 변경을 따르지 않는 문제를 수정했습니다.
- 작업 흐름의 다이어그램이 자동으로 정렬될 때 스텝들이 겹칠 수 있는 문제를 수정했습니다.
- 작업 흐름 다이어그램의 검색 기능의 검색 대상이 검색 기능을 켜거나 끌 때 변경되는 문제를 수정했습니다.
- 비전 서비스가 "비전 인식" 스텝에 의해 트리거되지 않았을 때 "비전 이동" 오류 알림 메시지가 잘못되는 문제를 수정했습니다.
- "비전 이동" 스텝이 비전 결과를 재사용하고 계획이 실패할 때 오류 메시지가 명확하지 않은 문제를 수정했습니다.
- "비전 이동" 스텝에서 피킹한 수량이 설정된 값에 달할 때 알림 메시지가 잘못되는 문제를 수정했습니다.
- 비전을 통해 "피킹된 물체를 업데이트"할 때 물체 치수 정보가 파라미터에 설정된 값을 직접 사용하는 문제를 수정했습니다.
- 모델 유형이 사용자 정의 모델인 경우, 피킹 시 작업물 모델이 피킹되지 않는 문제를 수정했습니다.

## 과거 버전 업데이트 설명

[Mech-Viz 1.6.2 업데이트 설명](#)

[Mech-Viz 1.6.1 업데이트 설명](#)

[Mech-Viz 1.6.0 업데이트 설명](#)

## 2.4. Mech-Center 업데이트 설명

이 부분에는 Mech-Center 1.7 버전의 추가된 기능, 최적화된 기능 및 복구된 문제에 대해 소개하겠습니다.

## Mech-Center 1.7.4 업데이트 설명

이 부분에서는 Mech-Center 1.7.4 버전 복구된 문제에 대해 소개하겠습니다.

### 문제 복구

Mech-Center 1.7.4 버전은 210 명령어로 반환된 포즈 데이터에서 로봇 베이스의 높이가 제거되지 않는 문제를 수정하였습니다.

## Mech-Center 1.7.2 업데이트 설명

이 부분에는 Mech-Center 1.7.2 버전의 추가된 기능, 최적화된 기능 및 복구된 문제에 대해 소개하겠습니다.

### 기능 추가

#### 표준 인터페이스는 Mitsubishi(미쓰비시) MC 프로토콜을 지원함

표준 인터페이스는 Mitsubishi(미쓰비시) PLC와 MC 통신을 지원하며, 비전 시스템은 클라이언트로, Mitsubishi(미쓰비시) PLC 장치는 서버로 작동합니다. 사용자는 Mech-Vision의 "로봇 및 인터페이스 구성" 패널에서 PLC의 IP 주소 및 포트, 통신 프레임, D 레지스터 시작 주소를 설정할 수 있습니다.

#### 표준 인터페이스는 UDP 프로토콜을 지원함

표준 인터페이스는 UDP 프로토콜을 지원합니다. 비전 시스템은 서버로 작동하며, TCP 프로토콜과 유사한 방법 및 명령을 사용합니다. 적합한 로봇에서는 현대 로봇만 비전 시스템과 UDP를 사용하여 통신합니다.

#### 표준 인터페이스는 로케(ROKAE)(산업) 로봇, 현대 (산업) 로봇 및 나치 (산업) 로봇에 적용됨

표준 인터페이스는 로케(ROKAE)(산업) 로봇, 현대 (산업) 로봇 및 나치 (산업) 로봇에 적용되었습니다. Mech-Center 1.7.2 버전은 새로운 로봇 표준 인터페이스 프로그램을 추가했습니다.

#### URCap 1.5.0은 Polyscope 6 시스템을 지원함

Mech-Center 1.7.2 버전에는 URCap 1.5.0 플러그인 (1.5.0.urcap 및 1.5.0.urcapx)이 추가되었습니다.



UR Polyscope 6 시스템에서 URCap 플러그인의 확장자는 .urcapx입니다. Polyscope 6 시스템은 이전의 .urcap 파일과 호환되지 않습니다.

URCap 1.5.0 플러그인이 적용되는 Polyscope 버전 및 Mech-Mind 제품군 패키지의 버전은 아래 표에서 확인할 수 있습니다. 구체적인 로봇의 Polyscope 버전에 따라 해당하는 URCap 플러그인 파일을 선택하시기 바랍니다.

URCap 플러그인 파일	Polyscope 버전에 대한 요구사항	Mech-Mind 제품군 패키지 버전에 대한 요구사항
1.5.0.urcap	E 시리즈: 5.9 및 이상(6.0 이하) CB 시리즈: 3.14 및 이상	1.7.2
1.6.2	E 시리즈: 6.0 및 이상	1.7.2
1.7.0	E 시리즈: 5.3 및 이상(6.0 이하) CB 시리즈: 3.9 및 이상	1.6.1 및 이상

URCap 1.5.0 플러그인은 URCap 1.4.6 플러그인과 비교하여 다음과 같은 새로운 기능을 제공합니다:

Mech-Vision "경로 계획" 스텝을 사용하여 계획된 경로를 얻을 수 있습니다.

- 프로젝트 번호, 레시피 번호, 분기 번호 및 분기 아웃포트 번호를 변수 입력으로 지원합니다.
- Mech-Vision의 다양한 출력 결과를 수신할 수 있습니다: Basic(비전 포즈 및 레이블 수신), Custom(비전 포즈, 레이블 및 사용자 정의 포트 데이터 수신), Planned path(경로 계획 스텝에서 출력한 웨이포인트 및 레이블 수신).

### AB PLC와 EtherNet/IP 통신 가능

표준 인터페이스는 Allen-Bradley PLC(AB PLC)과 EtherNet/IP 통신이 가능하며, 비전 시스템은 슬레이브 장치로, AB PLC 장치는 마스터 장치로 작동합니다. 사용자는 Mech-Vision의 "로봇 및 인터페이스 구성" 패널에서 EtherNet/IP 통신을 설정할 수 있습니다.

### 기능 최적화

#### 이미지 캡처 완료 후 101 명령어(성공적으로 트리거됨) 반환을 지원함

이전 버전의 Eye In Hand 시나리오에서는 Mech-Vision 프로젝트가 실행되기를 기다리지 않고 로봇이 카메라 이미지 캡처를 완료한 후 바로 이동할 수 있도록 하기 위해 Mech-Vision 프로젝트의 "카메라에서 이미지를 캡처하기" 스텝 이후에 "알림" 스텝을 추가해야 했습니다.

새 버전에서는 "알림" 스텝을 추가할 필요가 없고 Mech-Vision의 "로봇 및 인터페이스 구성" 패널에서 "고급 설정" 버튼을 클릭 한 후 **캡처 후 1102: 성공적으로 트리거됨** 반환 버튼을 선택하기만 하면 됩니다. 로봇은 Mech-Vision 프로젝트를 트리거하고 "1102 : 성공적으로 트리거됨"을 수신 한 후 이동을 시작할 수 있으므로 사이클 타임을 단축 할 수 있습니다.

새 버전에서는 PROFINET 프로토콜의 표준 인터페이스를 사용하는 경우 다음 중 하나를 사용하여 카메라 이미지 캡처가 완료된 후 로봇을 이동시킬 수 있습니다:



- "카메라에서 이미지를 캡처하기" 스텝 후에 "알림" 스텝을 추가하고 Trigger Acknowledge 신호 값을 조건으로 사용하기만 하면 됩니다.
- **캡처 후 "1102: 성공적으로 트리거됨" 반환**을 선택하고 1102 상태 코드를 조건으로 사용하기만 하면 됩니다.

### 문제 복구

Mech-Center 1.7.2 버전에서는 다음 문제를 복구했습니다:

- 표준 인터페이스를 사용하여 자동 캘리브레이션을 수행 할 때 Mech-Vision 캘리브레이션 파라미터를 설정하는 동안 발생하는 통신 시간 초과 문제를 복구했습니다.
- Mech-Viz 프로젝트 명령을 트리거 한 후 즉시 Mech-Viz 분기를 설정하면 실패하는 문제를 복구했습니다.

### Mech-Center 1.7.1 업데이트 설명

이 부분에서는 Mech-Center 1.7.1 버전 복구된 문제에 대해 소개하겠습니다.

#### 문제 복구

Mech-Center 1.7.1 버전에서는 다음 문제를 복구했습니다:

- 210 명령이 출력하는 작업물의 방향 정보가 올바르지 않은 문제를 복구했습니다.
- 특별한 경우에는 새로운 IPC에서 "Mech-Center → 구성 설정 → → Mech-Interface → 목록에 있는 로봇"을 선택할 수 없는 문제를 복구했습니다.
- "어댑터 생성기 → 로봇 구성 Eye in Hand → 로봇 명칭" 파라미터의 경로가 잘못되는 문제를 복구했습니다.

- Mech-Center가 획득한 로봇 모델이 틀린 문제를 복구했습니다. 복구한 후 Mech-Vision 및 Mech-Viz가 선택한 로봇 모델이 다른 경우 Mech-Vision에서 인터페이스 서비스를 시작하면 Mech-Center는 Mech-Vision이 선택한 로봇 모델을 가져옵니다.
- 캘리브레이션 과정에서 Mech-Center를 종료할 수 없는 문제를 복구했습니다.
- Mech-Center와 Siemens PLC 연결 실패시 인터페이스 서비스를 직접 종료하는 문제를 복구했습니다. 복구한 후 Mech-Center는 인터페이스 서비스를 사용 가능한 상태로 유지하고 자동으로 다시 연결합니다.
- Mech-Center는 스텝로보틱스(STEP Robotics, 新时达机器人) 로봇의 오일러 각도 파라미터가 누락되는 문제를 복구했습니다.

## Mech-Center 1.7.0 업데이트 설명

이 부분에서는 Mech-Center 1.7.0버전의 추가된 기능, 최적화된 기능 및 복구된 문제에 대해 소개하겠습니다.

### 기능 추가

**표준 인터페이스 및 Adapter와의 통신은 더 이상 Mech-Center를 통해 구성할 필요가 없음**

Mech-Vision 1.7.0 버전에 이전 Mech-Center의 인터페이스 통신 서비스 기능이 통합되었습니다. 사용자는 Mech-Vision에서 직접 표준 인터페이스 및 Adapter와 통신을 완료할 수 있으며 Mech-Center 소프트웨어를 사용할 필요가 없습니다.



- 표준 인터페이스 통신 방식을 사용하는 경우 사용자는 표준 인터페이스 프로그램을 다시 복제해야 합니다.
- 마스터 컨트롤 통신 구성은 여전히 Mech-Center를 통해 구성해야 합니다. 사용자는 먼저 Mech-Center 소프트웨어를 연 다음 Mech-Vision 소프트웨어를 열어 이전 버전에서 설명한 대로 통신을 구성해야 합니다.

### 105 명령어 추가

Mech-Center 1.7.0 버전은 모든 표준 인터페이스 프로토콜(TCP/IP, PROFINET, EtherNet/IP, Modbus TCP, Siemens PLC Snap 7)에 대해 Mech-Vision에 새로 추가된 "경로 계획" 스텝에서 출력된 이동 경로를 수신하기 위해 105 명령어를 추가했습니다.

### 표준 인터페이스용 피킹 샘플 프로그램 추가

Mech-Center 1.7.0 버전에서는 ABB, FANUC, YASKAWA, KAWASAKI, KUKA, ELITE 및 JAKA 로봇의 표준 인터페이스에 대한 샘플 프로그램 "Mech-Vision 경로 계획스텝을 사용하여 계획된 경로를 획득하기"를 제공합니다.

Mech-Vision에서 경로 계획 기능이 제공해야 하는 솔루션을 사용하려면 사용자는 표준 인터페이스 프로그램과 샘플 로그를 다시 복제해야 합니다.

### JAKA 및 ELITE 표준 인터페이스 프로그램 추가

Mech-Center 1.7.0 버전에서는 JAKA 및 ELITE 표준 인터페이스 프로그램을 추가했습니다.

### 왼손 좌표계를 사용하는 로봇 지원

Mech-Center 1.7.0 버전에서는 왼손 좌표계를 사용하는 로봇을 지원합니다.

### 시멘스 PLC에서 110 명령 지원

Mech-Center 1.7.0 버전에서는 시멘스 PLC에 대해 110 명령어를 추가했습니다. 이 명령어는 시멘스 PLC가

Mech-Vision 사용자 정의 포트에서 출력된 데이터를 수신 할 수 있도록 합니다.

## 기능 최적화

### 101 및 201 명령 파라미터 수정

Mech-Center 1.7.0 버전은 101 및 201 명령의 **로봇 포즈 및 유형** 파라미터를 수정하여 로봇 측에서 사용자 정의 관절 각도를 전달 할 수 있습니다. ETH 시나리오에서 로봇은이 사전 정의된 관절 각도에서 경로를 계획하기 시작합니다.

### ESTUN 및 AUBO 마스터 컨트롤 통신 프로그램 다시 적용

Mech-Center 1.7.0 버전에서는 ESTUN 및 AUBO 마스터 컨트롤 통신 프로그램을 다시 적용합니다.

### Adapter 프로젝트 로드 최적화

Mech-Center 1.7.0 버전에서는 Adapter 프로젝트 소스 경로를 직접로드 할 수 있습니다. Adapter 프로젝트 파일을 수정한 후 Mech-Center 소프트웨어를 다시 시작 할 필요없이 인터페이스 서비스를 다시 시작하면 Adapter 프로젝트가 로드됩니다.

## 문제 복구

### EPSON SCARA 로봇 슬라이딩 막대 계산 문제 복구

Mech-Center 1.7.0 버전에서는 EPSON SCARA 로봇 슬라이딩 막대 계산 문제를 복구했습니다.

### FANUC SCARA 로봇 베이스 좌표계 문제 복구

Mech-Center 1.7.0 버전에서는 FANUC SCARA 로봇 베이스 좌표계 문제를 복구했습니다.

---

## 과거 버전 업데이트 설명

[Mech-Center 1.6.1 업데이트 설명](#)

[Mech-Center 1.6.0 업데이트 설명](#)

## 2.5. 1.7.0 프로젝트 업그레이드 가이드

Mech-Vision 1.7.0 버전에서 **솔루션** 기능이 추가되고 새로운 솔루션 파일 구조가 도입되었으므로 구버전 Mech-Vision 프로젝트는 이전된 후 정상적으로 사용될 수 있습니다.

소프트웨어를 1.7.0 버전으로 업그레이드한 후 이 부분 내용을 참조하여 프로젝트 마이그레이션을 완료하십시오.

## 개요

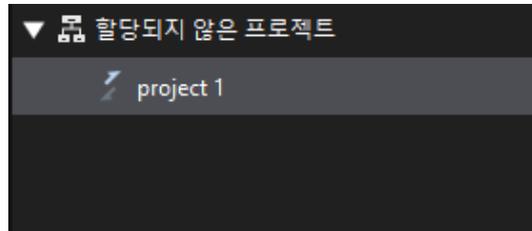
프로젝트 설명	이전 방식	업그레이드 후의 처리
표준 인터페이스 또는 Adapter를 통해 통신하고 Mech-Vision 프로젝트만으로 비전 포인트를 획득합니다.	Mech-Vision 프로젝트를 솔루션으로 이전하고 Mech-Vision에서 로봇 통신을 다시 구성합니다. Mech-Vision 소프트웨어를 별도로 시작해야 합니다.	Mech-Vision 프로젝트를 솔루션으로 이전하기 로봇 통신 설정 부팅 시 자동 시작 수정
Mech-Viz를 사용하여 간단한 경로 계획을 완료합니다.	Mech-Mind 기술 서포트팀에 문의하여 Mech-Vision 솔루션이 요구 사항을 충족하는지 평가합니다. 요구 사항을 충족하면 “경로 계획” 스텝을 사용하여 충돌이 없는 경로를 계획할 수 있는 솔루션을 구축합니다.	Mech-Vision 프로젝트를 솔루션으로 이전하기 로봇 통신 설정 부팅 시 자동 시작 수정 Mech-Vision 프로젝트에 “경로 계획” 스텝을 추가하기 로봇 인터페이스 프로그램을 복제하기 로봇 프로그램을 수정하기
Mech-Viz를 통해 로봇을 마스터 컨트롤합니다.	원래 소프트웨어 사용 방식(Mech-Center + Mech-Vision + Mech-Viz) 및 작동 습관을 유지하십시오. Mech-Center를 먼저 시작하고 소프트웨어에서 Mech-Vision 및 Mech-Viz를 시작합니다.	Mech-Vision 프로젝트를 솔루션으로 이전하기
Mech-Viz를 사용하여 혼합 팔레타이징, 다수 피킹&배치, 복잡한 경로 생성 등과 같은 까다로운 작업을 완료합니다.		

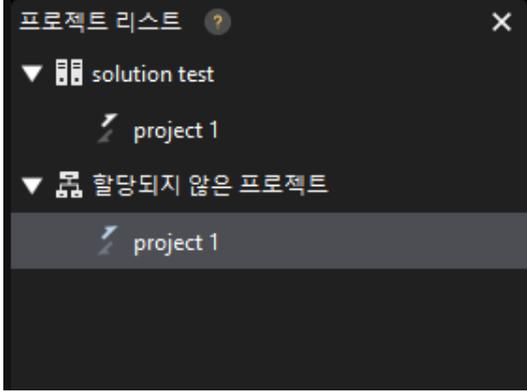
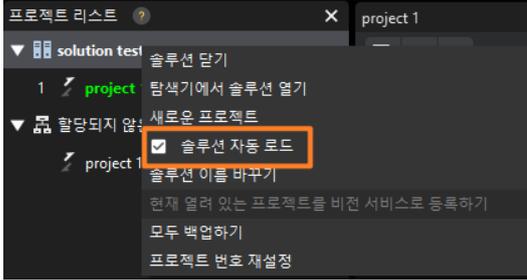
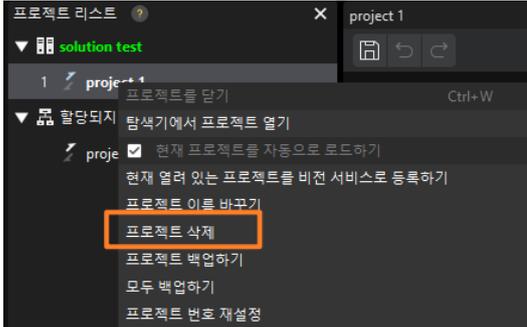
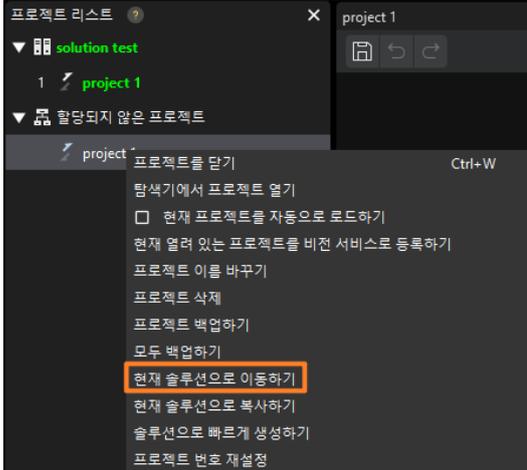
## 업그레이드 후의 처리

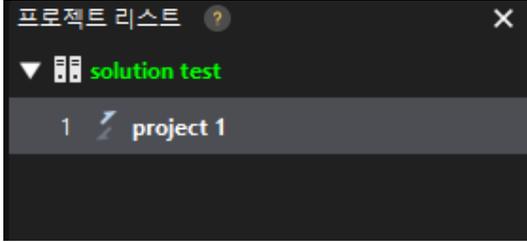
### Mech-Vision 프로젝트를 솔루션으로 이전하기

할당되지 않은 Mech-Vision 프로젝트를 번호별로 솔루션으로 이전합니다.

-  프로젝트가 솔루션에 포함되지 않은 경우 카메라 캘리브레이션 기능을 사용할 수 없습니다.

1	Mech-Vision에서 원래 프로젝트 번호에 따라 순서대로 프로젝트를 엽니다. 이때 프로젝트는 <b>할당되지 않은 프로젝트</b> 아래에 있습니다.	
---	--	--

<p>2</p>	<p><b>파일 &gt; 새로운 솔루션</b> 버튼을 클릭하십시오.  <b>파일 &gt; 솔루션 저장</b> 버튼을 클릭하여 솔루션을 저장할 경로를 선택하십시오.</p>	
<p>3</p>	<p>솔루션 이름을 마우스 오른쪽 버튼으로 클릭하고 <b>[솔루션 자동 로드]</b> 버튼을 클릭하십시오.</p>	
<p>4</p>	<p><b>프로젝트 1</b>를 마우스 오른쪽 버튼으로 클릭하고 메뉴에서 <b>[프로젝트 삭제]</b>를 클릭하십시오.</p>	
<p>5</p>	<p>활당되지 않은 프로젝트를 마우스 오른쪽 버튼으로 클릭하고 <b>[현재 솔루션으로 이동하기]</b> 버튼을 클릭하십시오.              활당되지 않은 모든 프로젝트에 대해 순서대로 이 작업을 수행하십시오.</p>	

6	현재 프로젝트 번호가 이전하기 전의 프로젝트 번호와 일치하는지 확인하십시오. <b>파일 &gt; 솔루션 저장</b> 버튼을 클릭하십시오.	
---	---	--

### 로봇 통신 설정

Mech-Vision에서 로봇 및 인터페이스 구성을 완료합니다. 다음 작업을 수행하기 전에 Mech-Center가 시작되지 않은 상태인지 확인하고 Mech-Vision을 별도로 시작하십시오.

1. Mech-Vision 툴 바에서 [ **로봇 및 인터페이스 구성** ]을 클릭합니다.
2. **로봇 선택**의 드롭다운 바에서 “목록에 있는 로봇”을 선택합니다.
3. [ **로봇 모델 선택** ] 버튼을 클릭하여 로봇 라이브러리에서 현재 사용하는 로봇 모델을 선택합니다.
4. [ **다음** ]을 클릭하여 통신 구성 인터페이스로 들어갑니다.
5. 현재 프로젝트의 실제 통신 구성에 따라 인터페이스 유형, 프로토콜 및 IP 주소를 설정합니다.
6. [ **응용** ] 버튼을 클릭하면 인터페이스 서비스가 시작됩니다. 이때 로그 표시줄에는 해당 로그가 출력되며 툴 바의 "인터페이스 서비스"가 활성화됩니다.

### 부팅 시 자동 시작 수정

Mech-Center의 자동 시작을 취소하고 Mech-Vision의 부팅 시 자동 시작을 설정합니다.

1. Mech-Center 소프트웨어를 시작하고 **구성 설정 > 선호 설정**을 순서대로 클릭하십시오.
2. [ **PC 부팅 시 Mech-Center 자동으로 실행하기** ] 버튼을 언체크하십시오.
3. [ **저장** ]을 클릭한 후 Mech-Center 소프트웨어를 닫으십시오.

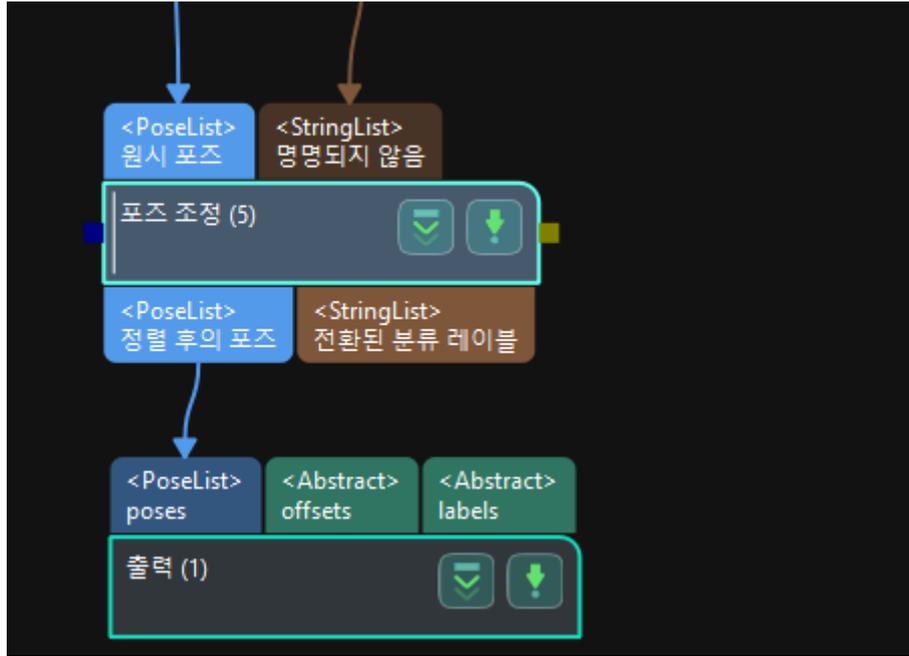
Mech-Vision의 부팅 시 자동 시작 기능을 활성화합니다.

1. Mech-Vision 소프트웨어를 시작하고 **설정 > 옵션 > 고급 설정**을 순서대로 클릭하십시오.
2. [ **시작할 때 Mech-Vision 실행** ] 버튼을 클릭하고 [ **확인** ] 버튼을 클릭하십시오.

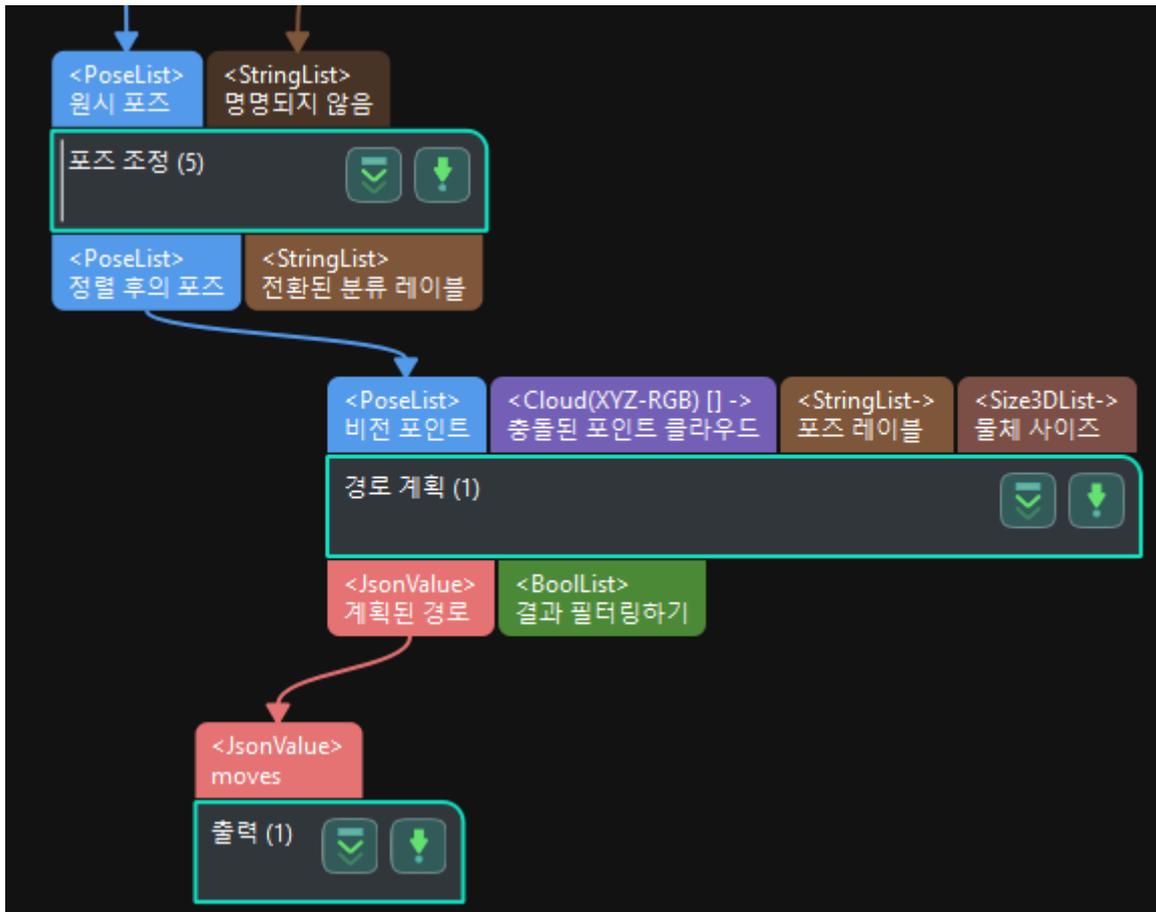
### Mech-Vision 프로젝트에 “경로 계획” 스텝을 추가하기

Mech-Vision 프로젝트에 “경로 계획” 스텝을 추가하고 로봇의 이동 경로를 계획합니다.

아래 그림에 표시된 프로젝트를 예로 들어 보겠습니다.



1. 스텝 라이브러리에서 “경로 계획” 스텝을 찾아 프로젝트 편집 영역으로 드래그합니다.
2. “출력” 스텝의 “포트 유형”을 “Predefined(robot path)”로 설정합니다.
3. 아래 그림을 참조하여 해당 포트의 연결을 완료합니다.



4. “경로 계획” 스텝의 파라미터 패널에서 [ 경로 계획 도구 열기 ] 기능을 클릭하십시오.

5. 경로 계획 도구를 참조하여 로봇의 이동 경로 계획을 완료하십시오.

### 로봇 인터페이스 프로그램을 복제하기

로봇 인터페이스 프로그램을 다시 복제합니다.

1. 로봇 및 인터페이스 구성에서 해당 브랜드의 로봇을 선택한 후 통신 구성 인터페이스에서 [ **프로그램 폴더 열기** ]를 클릭하여 이 폴더의 파일은 표준 인터페이스 프로그램 파일 및 피킹 샘플 프로그램입니다.
2. **표준 인터페이스 통신** 내용을 참조하여 프로그램 복제를 완료하십시오.

### 로봇 프로그램을 수정하기

Mech-Viz 프로젝트를 트리는 것에서 피킹 경로를 얻기 위해 Mech-Vision 프로젝트를 트리는 것으로 로봇 측의 프로그램을 수정합니다.

- call mm\_start\_vis으로 call mm\_start\_viz를 대체합니다.
- call MM\_Get\_VisPath로 call mm\_get\_vizdata를 대체합니다.

## 3. 소프트웨어 설치 가이드

이 부분에서는 Mech-Mind 비전 시리즈 소프트웨어의 다운로드, 설치, 업그레이드, 제거, 재설치 및 수정을 소개하겠습니다.

Mech-Mind 비전 시리즈 소프트웨어는 Mech-Mind에서 자체적으로 연구&개발한 Mech-Vision, Mech-Viz 및 Mech-Center 등 소프트웨어의 총칭입니다. Mech-Mind 비전 시리즈 소프트웨어는 간단하고 사용하기 편리한 설치 마법사를 제공하여 표준화된 설치, 업그레이드, 제거를 실현할 수 있고 중국어, 영어, 한국어, 일본어 등 다국어를 지원하며 다중 해상도 스케일링 비율 적용 지원합니다.

Mech-Mind 비전 시리즈 소프트웨어를 처음 사용하는 경우 [소프트웨어 설치](#)를 참조하여 소프트웨어를 설치할 수 있습니다.

Mech-Mind 비전 시리즈 소프트웨어를 이미 설치한 경우 [소프트웨어 업그레이드](#)를 참조하여 소프트웨어를 최신 버전으로 업그레이드할 수 있습니다.

### 시스템 요구 사항

Mech-Mind 비전 시리즈 소프트웨어는 사용하기 전에 IPC나 PC에 설치해야 합니다. 소프트웨어가 설치된 IPC 또는 PC는 다음과 같은 시스템 요구 사항(추천)을 충족해야 합니다:

- **운영 체제(OS)**: Windows 10 및 이상
- **CPU**: AVX2 명령어 세트가 지원되어야 하며 다음 조건 중 하나를 충족해야 합니다.
  - 개별 그래픽 카드가 없는 경우: Intel i5-12400 및 이상.
  - 개별 그래픽 카드가 있는 경우: Intel i7-6700 및 이상, 그래픽 카드는 GTX 1050 Ti보다 낮으면 안됩니다.
- **메모리**: 8GB 및 이상
- **GPU**: NAVDIA GTX 1050TI 및 이상(개별 그래픽 카드가 있는 경우)
- **하드 디스크**: 128GB SSD 및 이상



- Mech-Mind 비전 시리즈 소프트웨어는 다른 브랜드의 개별 그래픽 카드를 지원하지 않습니다.
- IPC 또는 PC에서 소프트웨어가 설치된 디스크의 공간이 10GB 이상인지 확인하십시오. 그렇지 않으면 설치가 실패할 수 있습니다.
- IPC나 PC에 동글 드라이버가 설치되어 있지 않은 경우 소프트웨어 설치 과정에서 설치 마법사가 자동으로 C 드라이브에 설치합니다. 드라이버를 제거하거나 설치 파일을 이동하지 마십시오. 그렇지 않으면 라이선스 인증에 실패하고 Mech-Mind 소프트웨어를 사용할 수 없습니다.

### 소프트웨어 설치 패키지 다운로드

비전 시스템에 필요한 소프트웨어 설치를 용이하고 신속하게 완료하기 위해 Mech-Mind에서는 모든 비전 시리즈 소프트웨어를 통합하는 소프트웨어 제품군(Mech-Mind Software Suite, 이하 "제품군"이라고 함)을 제공합니다. 또한 Mech-Mind에서는 비전 시리즈 소프트웨어마다 소프트웨어 설치 패키지(이하 "소프트웨어 패키지")를 제공합니다.



- 제품군에 Mech-Vision, Mech-Viz 및 Mech-Center 세 가지 소프트웨어가 통합됩니다.
- 제품군 패키지와 소프트웨어 패키지는 모두 동글에 적용하고 소프트웨어의 라이선스 승인을 완료하는 데 사용되는 동글 드라이버를 통합합니다.
- 제품군 패키지와 Mech-Center 소프트웨어 패키지에는 모두 Python 3.6.5 환경을 통합하여 환경 체크 도구를 포함하지 않습니다.  
Mech-Vision 소프트웨어 패키지에는 Python 3.6.8 환경을 통합합니다.

제품군 또는 비전 소프트웨어 패키지를 다운로드하려면 [Mech-Mind 온라인 커뮤니티](#)를 방문하거나 Mech-Mind 사전 판매 엔지니어 또는 영업 담당자에게 문의하십시오.

## 소프트웨어 설치

### 소프트웨어 설치 패키지 무결성 여부 확인

소프트웨어 설치 패키지는 전송 또는 다운로드 중에 손상될 수 있으므로 소프트웨어를 설치하기 전에 소프트웨어 설치 패키지의 무결성을 확인해야 합니다. 소프트웨어 설치 패키지의 무결성을 편리하게 검증하기 위해 소프트웨어 출시 시 설치 패키지의 CRC32 코드가 제공됩니다.

소프트웨어 패키지 무결성을 확인하려면 다음 단계를 수행하십시오:

1. 다운받은 소프트웨어 설치 패키지를 IPC나 PC의 지정된 디렉터리(예: "D:/")에 복사합니다.
2. 소프트웨어 설치 파일을 마우스 오른쪽 버튼을 클릭한 다음에 **CRC SHA > CRC32**를 선택하여 소프트웨어 설치 파일의 CRC32 체크 코드를 계산하십시오.



소프트웨어 설치 패키지가 압축 패키지인 경우 전체 압축 패키지와 압축 해제된 설치 파일에 대해 CRC32 코드를 계산하고 Content 폴더에 대해 CRC32 체크섬을 계산해야 합니다.

3. 계산된 CRC32 코드는 소프트웨어가 출시될 때 제공한 CRC32 코드와 일치하는지를 확인합니다.



CRC32 코드가 일치하지 않으면 소프트웨어의 설치 패키지를 다시 다운로드해야 합니다. 압축을 푼 후의 설치 파일의 코드 또는 Content 폴더의 체크섬이 일치하지 않으면 소프트웨어의 설치 패키지를 다시 압축을 푸십시오.

### 여러 개 비전 시리즈 소프트웨어의 빠른 설치

제품군을 사용하여 여러 개의 비전 시리즈 소프트웨어를 빠르게 설치하려면 다음 단계를 수행하십시오:

1. 획득한 패키지를 **D:/Mech-Mind**와 같은 IPC 또는 PC의 지정된 디렉토리에 복사하고 압축을 풉니다.



압축을 푼 후 소프트웨어의 설치 파일(Mech-Mind Software Suite Installer x.x.x.exe) 및 content 폴더의 위치를 변경하지 마십시오.

2. 소프트웨어 설치 파일을 더블 클릭하여 **Mech-Mind Software Suite 설치 마법사**를 엽니다. **환영합니다** 창이 표시됩니다.
3. **환영합니다** 창에서 소프트웨어와 관련된 설명 내용을 확인한 후 **[ 다음 ]** 버튼을 클릭하십시오.
4. **라이선스 계약** 창에서 라이선스 계약을 주의 깊게 읽고 **[ 라이선스 계약의 모든 조건에 동의합니다 ]** 확인란을 선택한 후 **[ 다음 ]**을 클릭합니다.
5. **제품 선택** 창에서 설치할 제품을 선택하고 필요에 따라 **[ 바탕 화면 바로 가기를 만들기 ]** 확인란을 선택한 후 **[ 다음 ]**을 클릭합니다.
6. **설치 경로 선택** 창에서 필요에 따라 설치 경로를 변경합니다(예: "D:/project"). 그다음 **[ 다음 ]** 버튼을 클릭합니다.



설치 경로를 변경하지 않으면 기본적으로 다음 경로에 소프트웨어가 설치됩니다:

- 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. **설치 전 확인** 창에서 설치할 제품이 맞는지 확인한 후 [ **설치** ]를 클릭합니다.
8. **설치** 창에서 소프트웨어 설치가 완료될 때까지 기다립니다.
9. 소프트웨어 설치가 완료되면 [ **완료** ] 창에서 [ **완료** ] 버튼을 클릭합니다.

## 지정된 비전 소프트웨어 설치

지정된 비전 소프트웨어를 설치하려면 다음 단계를 수행하십시오:

1. 획득한 소프트웨어의 설치 패키지를 **D:/Mech-Mind**와 같은 IPC 또는 PC의 지정된 디렉토리에 복사하고 필요가 있으면 압축을 풉니다.



압축을 푼 후 소프트웨어의 설치 파일(Xxxx-xxx Installer x.x.x.exe) 및 content 폴더의 위치를 변경하지 마십시오.

2. 소프트웨어 설치 파일을 더블 클릭하여 **Xxxx-xxx 설치 마법사**를 엽니다. **환영합니다** 창이 표시됩니다.
3. **환영합니다** 창에서 소프트웨어와 관련된 설명 내용을 확인한 후 [ **다음** ] 버튼을 클릭하십시오.
4. **라이선스 계약** 창에서 라이선스 계약을 주의 깊게 읽고 [ **라이선스 계약의 모든 조건에 동의합니다** ] 확인란을 선택한 후 **다음** 을 클릭합니다.
5. **제품 선택** 창에서 설치할 제품을 선택하고 필요에 따라 [ **바탕 화면 바로 가기를 만들기** ] 확인란을 선택한 후 [ **다음** ]을 클릭합니다.
6. **설치 경로 선택** 창에서 필요에 따라 설치 경로를 변경합니다(예: "D:/projectt" ). 그다음 [ **다음** ] 버튼을 클릭합니다.



설치 경로를 변경하지 않으면 기본적으로 다음 경로에 소프트웨어가 설치됩니다:

- 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. **설치 전 확인** 창에서 설치할 제품이 맞는지 확인한 후 [ **설치** ]를 클릭합니다.
8. **설치** 창에서 소프트웨어 설치가 완료될 때까지 기다립니다.
9. 소프트웨어 설치가 완료되면 **완료** 창에서 [ **완료** ] 버튼을 클릭합니다.



- 1.6.0부터 Mech-Viz 소프트웨어는 메인스트림 브랜드 로봇의 설치 패키지만 통합합니다.
- 다른 브랜드나 모델의 로봇 설치 패키지가 필요한 경우 [온라인 로봇 라이브러리](#)를 참조하여 다운로드하여 모델을 도입하십시오.

## 소프트웨어 업그레이드



- 1.6.0 및 이상 버전의 소프트웨어에서만 이 기능을 사용할 수 있습니다.
- 소프트웨어는 1.6.0 이전 버전에서 1.6.0 이상으로의 직접 업그레이드를 지원하지 않습니다. 이전 버전의 소프트웨어를 설치한 사용자의 경우 소프트웨어를 1.6.0 또는 이상 버전으로 업그레이드해야 하면, 먼저 구버전 소프트웨어를 제거한 다음 [새버전의 소프트웨어를 설치해야 합니다](#). 소프트웨어를 업그레이드하기 전에 [업그레이드 주의사항](#) 내용을 자세히 읽어 주시기 바랍니다.
- 이전에 소프트웨어 제품군 패키지를 사용하여 소프트웨어를 설치한 경우 업그레이드하기 전에 최신 버전의 소프트웨어 제품군 패키지를 획득하십시오. 단일 소프트웨어의 패키지를 사용하여 소프트웨어를 설치한 경우 업그레이드하기 전에 단일 소프트웨어 패키지의 새 버전을

획득하십시오.

소프트웨어를 업그레이드하려면 다음 단계를 수행하십시오:

1. 다운받은 새 버전의 소프트웨어 설치 패키지를 IPC나 PC의 지정된 디렉터리(예: “D:/Mech-Mind”)에 복사하고 압축을 풉니다(필요한 경우).
2. 소프트웨어 설치 파일을 더블 클릭하여 **Xxxx-xxx 설치 마법사**를 엽니다. **업그레이드** 창이 표시됩니다.
3. **업그레이드** 창에서 [ **업그레이드, 기존 버전 유지** ] (제품군 패키지에는 이 옵션을 지원하지 않음) 또는 [ **업그레이드, 기존 버전 삭제** ]를 선택하십시오.
4. 소프트웨어 업그레이드를 기다려 주십시오.

## 소프트웨어 제거

특정 비전 소프트웨어를 더 이상 사용할 필요가 없으면 다음 두 가지 방법으로 제거할 수 있습니다:

- 제거 마법사를 사용하여 제거(권장)
- 제어판을 사용하여 제거

### 제거 마법사를 사용하여 제거



이전에 소프트웨어 제품군 패키지를 사용하여 소프트웨어를 설치한 경우 소프트웨어 제품군 패키지를 통해 소프트웨어를 제거하십시오. 단일 소프트웨어의 패키지를 사용하여 소프트웨어를 설치한 경우 단일 소프트웨어의 패키지를 통해 소프트웨어를 제거하십시오.

제거 마법사를 사용하여 소프트웨어를 제거하려면 다음 단계를 따르십시오:

1. 소프트웨어 설치 파일을 더블 클릭하여 **Xxxx-xxx 설치 마법사**를 엽니다. **유지 보수** 창이 표시됩니다.
2. **유지 보수** 창에서 [ **제거** ] 버튼을 클릭하십시오.
3. **제거** 창에서 [ **사용자 구성 파일을 유지하기** ] 또는 [ **사용자 구성 파일을 유지하지 않기** ]를 선택하십시오.
4. 소프트웨어 제거를 기다려 주십시오.

### 제어판을 사용하여 제거

제어판을 사용하여 소프트웨어를 제거하려면 다음 단계를 따르십시오:

1. IPC 또는 PC에서 **제어판**을 엽니다.
2. **프로그램 > 프로그램 및 기능**을 선택합니다.
3. 제거하려는 소프트웨어를 마우스 오른쪽 버튼으로 클릭하여 [ **제거** ] 버튼을 클릭합니다.



이전에 소프트웨어 제품군 패키지를 사용하여 소프트웨어를 설치된 경우 설치된 모든 소프트웨어는 프로그램 리스트에서 전체적으로 “Mech-Mind Software Suite x.x.x”로 표시됩니다.

4. 소프트웨어 제거를 기다려 주십시오.

## 소프트웨어 복구

Mech-Mind 비전 시리즈 소프트웨어가 비정상적이어서 정상적으로 사용할 수 없는 경우 복구 기능을 통해

소프트웨어를 다시 설치할 수 있습니다.



- 1.6.0 및 이상 버전의 소프트웨어에서만 이 기능을 사용할 수 있습니다.
- 이전에 소프트웨어 제품군 패키지를 사용하여 소프트웨어를 설치한 경우 소프트웨어 제품군 패키지를 통해 소프트웨어를 복구하십시오. 단일 소프트웨어의 패키지를 사용하여 소프트웨어를 설치한 경우 단일 소프트웨어의 패키지를 통해 소프트웨어를 복구하십시오.

소프트웨어를 복구하려면 다음 단계를 수행하십시오:

1. 소프트웨어 설치 파일을 더블 클릭하여 **Xxxx-xxx 설치 마법사**를 엽니다. **유지 보수** 창이 표시됩니다.
2. **유지 보수** 창에서 **[복구]** 버튼을 클릭하십시오.
3. 소프트웨어 복구를 기다려 주십시오.

## 설치한 소프트웨어를 수정하기



1.6.0 및 이상 버전의 제품군 패키지에 이 기능을 사용할 수 있습니다.

설치한 소프트웨어를 수정하려면 다음 단계를 수행하십시오:

1. 소프트웨어 설치 파일을 더블 클릭하여 **Mech-Mind Mech-Mind Software Suite 설치 마법사**를 엽니다. **유지 보수** 창이 표시됩니다.
2. **유지 보수** 창에서 **[수정]** 버튼을 클릭하십시오.
3. **제품 선택** 창에서 새로 설치하려는 제품과 다른 구성 요소를 선택하거나 제거하려는 제품과 다른 구성 요소를 언체크합니다.
4. 설치가 완료될 때까지 후속 단계를 수행하십시오.

## 소프트웨어 라이선스 계약

Mech-Mind 비전 시리즈 소프트웨어의 라이선스 계약은 [최종 사용자 라이선스 계약](#) 내용을 참조하십시오.

## 설치 시 자주 발생하는 문제

### 설치 패키지가 정상적으로 열리지 못함

#### 문제 현상:

설치 패키지가 실행될 때 정상적으로 열리지 못하거나 창이 갑자기 닫혀버립니다.

#### 원인 추측:

시스템 드라이브 디스크 여유 공간이 부족합니다.

#### 솔루션:

시스템 드라이브 디스크의 여유 공간이 설치 패키지의 크기보다 큰지 확인하십시오.

- 여유 공간이 설치 패키지 크기보다 작으면 불필요한 파일을 삭제하여 소프트웨어 설치를 위한 충분한 공간을 확보한 후 다시 설치하십시오. 문제가 여전히 해결되지 못하면 Mech-Mind 서포트팀에게 문의하십시오.
- 여유 공간이 설치 패키지 크기보다 큰 경우에도 이 문제가 발생하면 Mech-Mind 서포트팀에게 문의하십시오.

## 설치 실패

### 문제 현상:

소프트웨어를 설치하는 동안 "설치 실패"라는 오류 메시지가 나타납니다.

### 원인 추측:

- 설치 패키지가 손상되었거나 필요한 파일이 없습니다.
- 해당 컴퓨터 사용자는 관리자 권한이 없습니다.
- 다른 프로그램이 설치 중이거나 Windows 시스템이 업그레이드 중입니다.
- 기타 원인.

### 솔루션:

1. 설치 패키지를 다시 획득하고 [소프트웨어 설치 패키지의 무결성을 확인](#)한 다음 설치를 다시 시도하십시오.
  - 문제가 해결되면 고장 처리가 끝났습니다.
  - 해결되지 못하면 단계2를 수행하십시오.

2. C:\ProgramData\Package Cache 路径下的 {180DA015-BC2A-4262-928A-D312B20D9977}\v7.40.4997.501 파일을 찾아 삭제하십시오.



해당 파일을 찾지 못하면 파일 탐색기의 **보기** 화면에서 **숨긴 파일 및 폴더**를 선택해서 확인하십시오.

- 문제가 해결되면 고장 처리가 끝났습니다.
  - 문제가 여전히 해결되지 못하면 단계3를 수행하십시오.
3. 설치 패키지를 마우스 오른쪽 버튼으로 클릭하고 "관리자 권한으로 실행"을 클릭하십시오.
    - 문제가 해결되면 고장 처리가 끝났습니다.
    - 문제가 여전히 해결되지 못하면 단계4를 수행하십시오.
  4. 다른 프로그램의 설치가 완료될 때까지 기다리거나 Windows 시스템 업그레이드가 완료된 후 설치를 다시 시도하십시오.
    - 문제가 해결되면 고장 처리가 끝났습니다.
    - 문제가 여전히 해결되지 못하면 단계5를 수행하십시오.
  5. IPC 혹은 PC를 리부팅한 후 소프트웨어를 다시 설치하십시오.
    - 문제가 해결되면 고장 처리가 끝났습니다.
    - 문제가 여전히 해결되지 못하면 **완료** 화면에 있는 "설치 로그" 를 클릭하여 설치 시의 로그 정보를 수집하여 기술 서포트팀에 문의하십시오.

## 4. Mech-Vision

Mech-Vision 사용자 매뉴얼에 오신 것을 환영합니다!

Mech-Vision은 Mech-Mind에서 연구 개발한 차세대 **머신 비전 소프트웨어**로써 완전한 그래픽 인터페이스를 통해 무질서한 물체 피킹, 고정밀 위치 지정, 조립, 산업 검사/측정, 자동 경로 생성과 같은 고급 머신 비전 애플리케이션을 코드 작성 없이 완료할 수 있습니다.

이 매뉴얼은 아래와 같이 구성되어 있습니다.

번호	제목	내용
1	<a href="#">Mech-Vision 소개</a>	소프트웨어 기능 소개, 머신 비전과 관련된 개념 및 용어 소개, 사용자 인터페이스 소개, Mech-Vision을 사용한 솔루션 작성의 일반적인 흐름
2	<a href="#">[vision-operation-guide:operation-guide:::operation-guide]</a>	Mech-Vision의 일반적인 사용 가이드 소개
3	<a href="#">[vision-steps:steps:::steps]</a>	스텝 및 프로시저의 참조 정보
4	<a href="#">[vision-calibration:calibration:::calib-guide]</a>	핸드-아이 캘리브레이션의 개념 및 사용 방법 소개
5	<a href="#">[vision-measure-mode:measure-mode:::measure-mode]</a>	측정 모드의 기본 소개 및 주요 알고리즘 소개
6	<a href="#">[vision-tools:tools:::tools-guide]</a>	Mech-Vision 일반 도구 사용 가이드

### 4.1. Mech-Vision 소개

#### 4.1.1. 용어 및 개념

이 부분에서는 머신 비전과 관련된 용어 및 개념을 소개합니다.

##### 솔루션 및 프로젝트

##### 솔루션 라이브러리

다양한 산업 분야의 예제 솔루션 또는 프로젝트가 포함된 리소스 라이브러리로 예제 데이터가 포함되어 있습니다.

##### 솔루션

솔루션은 비전 응용을 실현하는 데 필요한 로봇과 통신, 비전 처리 및 경로 계획 등 기능 구성과 데이터의 모음입니다.

## 프로젝트

프로젝트는 Mech-Vision 프로젝트를 말하며 하나 및 이상의 솔루션으로 구성되었습니다. 프로젝트는 단독으로 사용할 수 없으며 솔루션에 속해야 합니다.

## 스텝

스텝은 프로젝트 구성의 기초이며, 하나의 스텝은 한 알고리즘 처리 단위이며 서로 다른 스텝을 결합하여 서로 다른 알고리즘 처리 프로세스를 구성합니다.

## 프로시저

여러 스텝의 조합을 프로시저라고 합니다. 서로 다른 프로젝트에는 종종 일관되거나 유사한 알고리즘 처리 프로세스가 있습니다. 이러한 고정된 알고리즘 처리 스텝을 캡슐화하고 결합하여 쉽게 재사용할 수 있습니다.

## 파라미터의 레시피

파라미터 레시피는 프로젝트가 다양한 상황에서 실행될 때 조정해야 하는 파라미터 설정 모음입니다. 다양한 상황에 대해 서로 다른 파라미터 레시피를 구성함으로써 반복적인 프로젝트 구축을 피하고 프로젝트 적응성이 향상됨으로써 생산 효율성이 향상됩니다.

## 핸드-아이 캘리브레이션

### 핸드-아이 캘리브레이션

카메라 좌표계와 로봇 좌표계 사이의 대응 관계를 설정하고 비전 시스템에 의해 결정된 물체 포즈를 로봇 좌표계 아래의 포즈로 변환하여 로봇이 물체에 대한 작업을 완료하도록 가이드하는 프로세스를 말합니다.

## 내부 파라미터

카메라의 내부 파라미터는 카메라의 내부 파라미터에만 관련됩니다.

## 외부 파라미터

카메라 좌표계와 세계 좌표계 간의 좌표 변환 관계입니다.

## 오일러 각

3차원 공간에서 물체의 자세를 설명하는 방법으로 피치(원뿔)각, 편주각(yaw angle) 및 롤각을 사용하여 3차원 공간에서 물체의 회전을 표현합니다.

## TCP

TCP(Tool Center Point), 툴의 센터 포인트이며 말단 장치 끝점의 포인트입니다. 작업물 피킹 등 작업을 수행하기 위해 로봇이 공간에 있는 어떤 포인트로 이동하라고 할 때 사실 TCP를 해당 포인트로 이동시키는 것입니다.

## 비전 처리 프로세스

### 포인트 클라우드

데이터의 표면 특성을 포함하는 포인트 모음입니다.

### 포즈

로봇이 피킹할 때의 위치 및 방향 정보입니다.

### 마스크

선택한 이미지, 모양 또는 물체로 처리된 이미지(전체 또는 일부)를 가려 이미지 처리 영역 또는 프로세스를 제어합니다. 오버레이에 사용되는 특정 이미지 또는 물체를 마스크라고 합니다.

## ROI

머신 비전 및 이미지 처리에서 처리가 필요한 영역은 처리된 이미지에서 선택되며 이를 관심

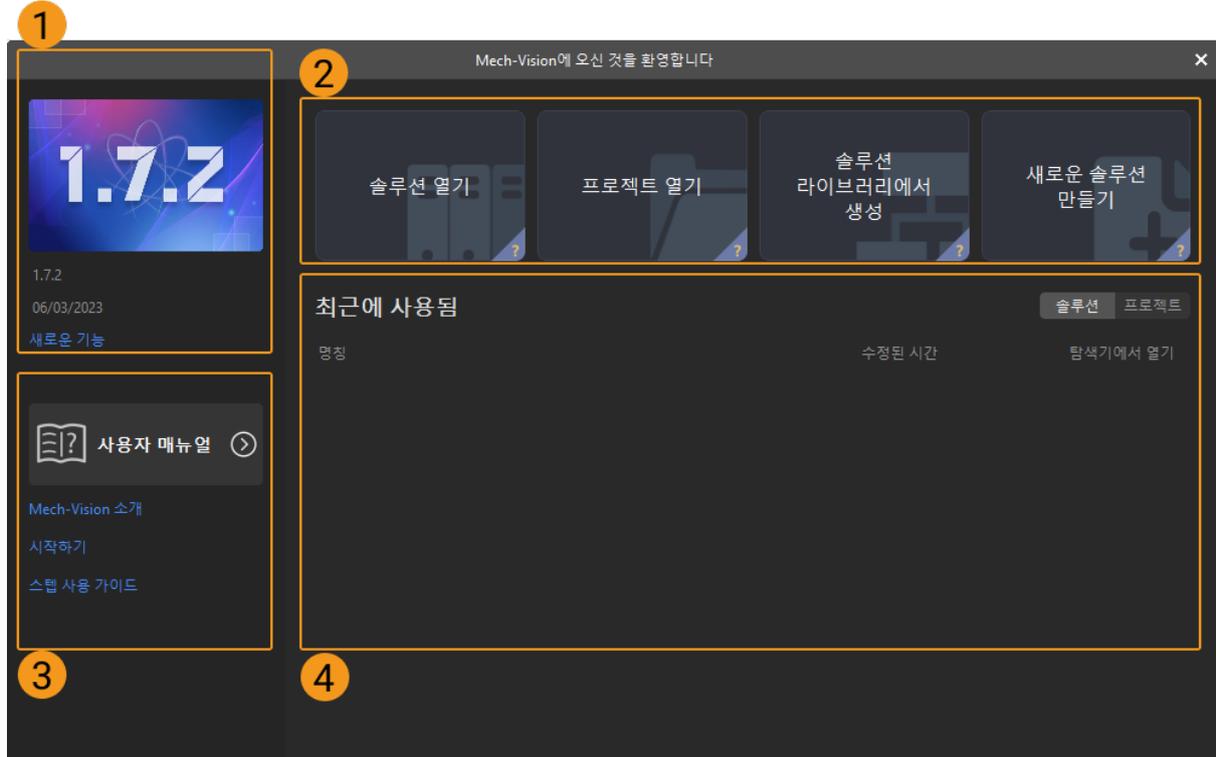
영역(ROI)이라고 합니다.

### 4.1.2. 사용자 인터페이스

이 부분에서는 Mech-Vision 소프트웨어의 시작 화면과 메인 인터페이스를 소개합니다.

#### 시작 화면

소프트웨어에 자동으로 로드된 계획이나 프로젝트가 없는 경우 Mech-Vision 소프트웨어를 연 후 아래 그림과 같이 시작 화면으로 들어갑니다.



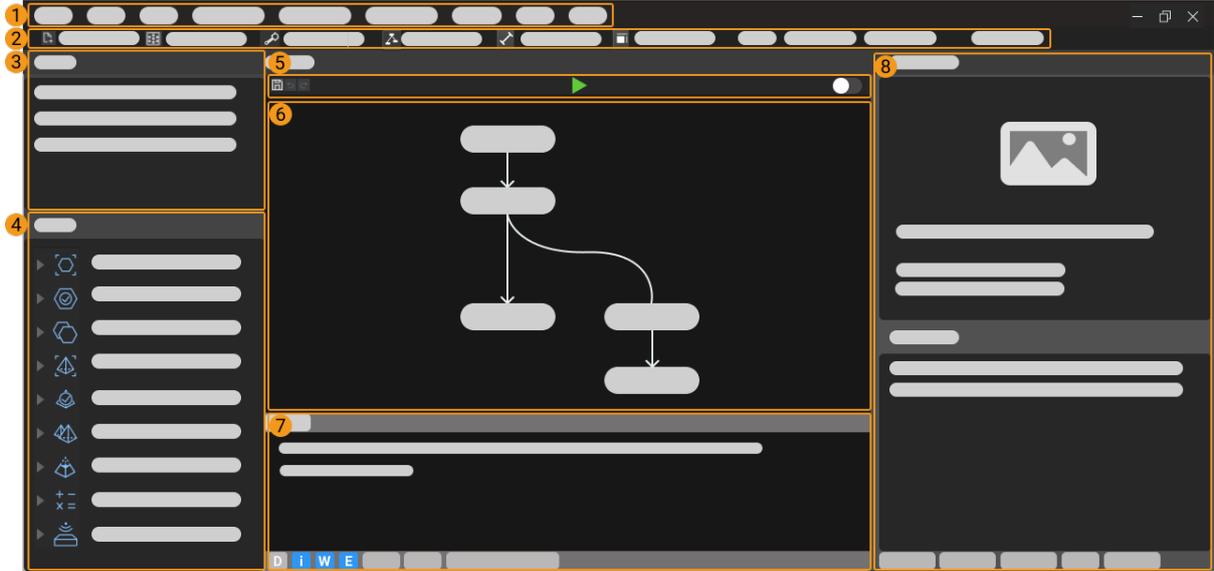
바로가기 항목에서 카드 우측 하단의 "?"에 마우스 포인터를 올리면 해당 카드에 대한 관련 소개를 볼 수 있습니다.

Mech-Vision 시작 화면은 다음 부분으로 구성됩니다.

일련 번호	화면	설명
1	버전 정보	버전 정보 및 소프트웨어 업데이트 설명을 볼 수 있습니다.
2	바로 가기	신속하게 솔루션, 프로젝트를 열고, 솔루션 라이브러리에서 새 솔루션 또는 프로젝트를 생성하고, 새로운 솔루션을 생성할 수 있습니다.
3	사용자 안내서	Mech-Vision 사용 설명서는 해당 버튼을 클릭하여 찾아볼 수 있습니다.
4	최근 사용	최근에 사용한 솔루션이나 프로젝트를 빠르게 열 수 있습니다.

## 메인 인터페이스

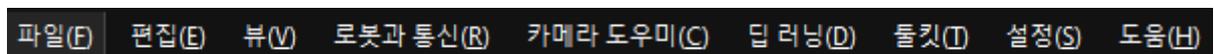
시작 화면을 종료하면 아래 그림과 같이 Mech-Vision의 메인 인터페이스로 들어갑니다.



Mech-Vision 메인 인터페이스는 다음 부분으로 구성됩니다.

번호	화면	설명
1	메뉴 바	파일, 편집, 뷰, 로봇과 통신, 카메라 도우미, 딥러닝, 툴킷, 설정, 도움 메뉴가 포함되어 있습니다.
2	툴 바	다양한 도구 및 기능에 대한 바로 가기 버튼이 포함되어 있습니다.
3	프로젝트 리스트	이미 열린 솔루션, 프로젝트 이름 및 통신 상태와 자동 로딩 상태를 포함한 상태를 표시할 수 있습니다.
4	스텝 라이브러리	프로젝트를 구축하는 데 사용할 수 있는 모든 스텝을 포함합니다.
5	프로젝트 툴 바	프로젝트 저장, 실행, 디버그 출력 등의 작업을 완료하는 데 사용됩니다.
6	프로젝트 편집 구역	비전 프로젝트 처리 흐름 또는 논리 흐름을 편집하는 데 사용됩니다.
7	프로젝트 배치 구역	디버그 출력, 스텝 파라미터, 프로젝트 도우미, 스텝 설명, 기록 및 스텝 비교를 포함합니다.
8	로그바	프로젝트가 실행될 때의 실시간 로그 정보를 표시하여 사용자가 특정 시간의 실행 기록을 쉽게 찾을 수 있습니다.

### 4.1.2.1. 메뉴 바



솔루션/프로젝트 작업과 관련된 기본 기능과 카메라, 딥러닝 등의 보조 도구를 제공합니다.

## 파일

솔루션 또는 프로젝트를 처리하는 데 사용됩니다.

옵션	설명	단축키
새로운 솔루션	새로운 솔루션 만들기	Ctrl+Shift+N
새로운 프로젝트	새로운 프로젝트를 만들기	Ctrl+N
솔루션 라이브러리에서 새로 만들기	솔루션 라이브러리에서 솔루션을 선택하고 새로 만들기	없음
솔루션 열기	기존 솔루션 열기	없음
프로젝트 열기	기존 프로젝트 폴더 열기	Ctrl+O
최근 사용했던 프로젝트를 열기	펼치면 최근에 사용한 프로젝트 및 솔루션이 표시되고, 클릭하면 바로 열수 있음	없음
실행 가능한 파일을 열기	소프트웨어가 위치한 디렉터리 열기	없음
솔루션 저장하기	현재 솔루션에 대한 변경 사항 저장하기	Ctrl+Shift+S
프로젝트를 저장하기	현재 프로젝트에 대한 변경 사항 저장하기	Ctrl+S
프로젝트를 JSON로 저장하기	프로젝트를 저장하고 프로젝트 폴더의 .vis 파일을 .json 파일로 저장하기	없음
저장 경로를 선택하기	프로젝트 콘텐츠를 지정된 위치에 저장하기	없음
프로젝트를 닫기	선택한 솔루션 닫기	없음
종료	닫고 종료하기 Mech-Vision	Ctrl+Q

## 편집

옵션	설명	단축키
취소	이전 작업을 취소하기	Ctrl+Z
다시 하기	이전 스텝 다시 실행하기	Ctrl+Y

## 뷰

인터페이스 디스플레이의 관련 설정을 변경하는 데 사용되며 체크 후 해당 탭이 소프트웨어 인터페이스에 표시됩니다.

옵션	설명
프로젝트 리스트	프로젝트 리스트를 열거나 닫습니다. 기본적으로 체크되어 있습니다.
스텝 라이브러리	스텝 라이브러리를 열거나 닫습니다. 기본적으로 체크되어 있습니다.
디버그 출력	디버그 결과 출력을 열거나 닫습니다. 기본적으로 활성화되어 있으며 비활성화할 수 없습니다.

옵션	설명
역사	역사 탭을 열거나 닫습니다. 기본적으로 체크되어 있습니다.
프로젝트 도우미	프로젝트 도우미 탭을 열거나 닫습니다. 기본적으로 체크되어 있습니다.
스텝 설명	스텝 설명 탭을 열거나 닫습니다. 기본적으로 체크되어 있습니다.
스텝 비교	스텝 비교 탭을 열거나 닫습니다. 기본적으로 체크되어 있습니다.
스텝 파라미터	스텝 파라미터 탭을 열거나 닫습니다. 기본적으로 체크되어 있습니다.
스텝 입력 포트 선택	스텝 입력 포트 선택을 열거나 닫습니다. 기본적으로 언체크되어 있습니다.
측정 결과 출력	측정 결과 출력을 열거나 닫습니다. 기본적으로 언체크되어 있습니다.
결과 보기	결과 보기(측정 모드 하에)를 열거나 닫습니다. 기본적으로 언체크되어 있습니다.
로그	로그 탭을 열거나 닫습니다. 기본적으로 체크되어 있습니다.
툴 바	툴 바를 열거나 닫습니다. 기본적으로 체크되어 있습니다.
기본적인 레이아웃	Mech-Vision 사용자 레이아웃을 기본 상태로 되돌립니다.
그리드 형식	그리드 형식 및 참조 라인 표시를 열거나 닫습니다. 기본적으로 언체크되어 있습니다.
정렬 유형	정렬 유형 선택(앵커, 수평, 수직)

### 로봇과 통신

로봇과 통신 관련 설정을 할 수 있습니다.

옵션	설명
로봇 및 인터페이스 구성	로봇과 해당 인터페이스를 구성할 수 있습니다.
로봇 라이브러리 도구	로봇을 도입, 도출할 수 있습니다.

### 카메라 도우미

이미지를 캡처하고 카메라를 캘리브레이션하고 파라미터를 분석할 수 있습니다.

옵션	설명	단축키
카메라 뷰어	이미지 획득 및 저장하기	Ctrl+Shift+V
카메라 캘리브레이션	표준: 일반 캘리브레이션 프로세스. 쿼: 기존 파라미터를 사용하여 신속하게 계산.	없음

## 딥 러닝

옵션	설명	단축키
딥 러닝 모델 패키지 관리 도구	딥 러닝 모델 패키지 관리 도구를 열기	없음
딥 러닝 서버	딥 러닝 서버를 열거나 닫기	Ctrl+Alt+D

## 툴킷

소프트웨어 보조 도구를 여는 데 사용됩니다.

옵션	설명
매칭 모델 및 픽 포인트 편집기	대상 물체의 포인트 클라우드 모델 생성, 포인트 클라우드 모델 편집 및 픽 포인트 추가

## 설정

소프트웨어의 공통 설정을 변경하는 데 사용됩니다.

옵션	설명	단축키
Mech-Center 주소 설정	Mech-Center 소프트웨어의 IP 주소를 설정하기	없음
프로젝트를 잠그기	현재 열려 있는 프로젝트를 잠그려면 비밀번호를 입력하세요.	없음
옵션	기본 소프트웨어 설정을 변경하는 데 사용됨(언어, 단위 등)	Ctrl + Shift + O
로그 레벨	저장할 로그 레벨 정보 선택: 결함 수정, 정상 정보, 경고 및 오류	없음

## 도움

소프트웨어의 현재 버전 정보, 업데이트 설명 및 사용자 안내서 등을 볼 수 있습니다.

옵션	설명
버전에 대하여	현재 버전 번호를 확인하는 데 사용됩니다.
업데이트 설명	업데이트 설명 페이지 열기
사용자 안내서	소프트웨어 사용자 안내서 보기
플러그인에 관하여	표시하려는 플러그인을 체크하여 각 플러그인의 명칭, 버전, 설명 및 화면 위치를 보기

### 4.1.2.2. 툴바

새로운 일반 프로젝트를 만들고 내장 도구를 사용하고 프로젝트 모드를 전환할 수 있습니다.

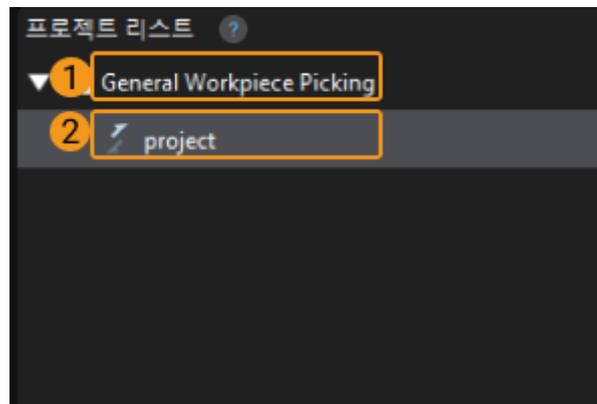


옵션	설명
솔루션 라이브러리에서 새로 만들기	신속하게 솔루션 생성
로봇 및 인터페이스 구성	로봇과 해당 인터페이스를 구성할 수 있음
카메라 캘리브레이션(표준)	카메라 캘리브레이션 표준 프로세스
매칭 모델 및 픽 포인트 편집기	대상 물체의 포인트 클라우드 모델 생성, 포인트 클라우드 모델 편집 및 픽 포인트 추가
표준 편집 모드	기본 모드
측정 모드	측정, 감지, 정렬, 식별 등에 사용됨
오퍼레이터 인터페이스(사용자 정의)	실제 필요에 따라 다양한 데이터를 표시할 수 있는 맞춤형 오퍼레이터 인터페이스
인터페이스 서비스	인터페이스 서비스 열기

#### 4.1.2.3. 프로젝트 리스트

프로젝트 리스트에는 현재 열려 있는 솔루션 및 프로젝트와 통신 상태 및 자동 로드 상태를 포함한 해당 상태가 표시됩니다.

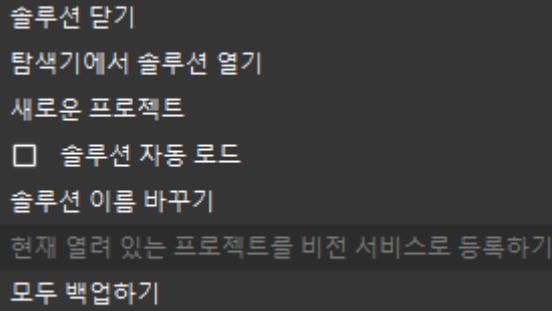
아래 그림과 같이 1은 솔루션 이름이고 2는 프로젝트 이름입니다.



- 위로 마우스 포인터를 놓으면 "솔루션"에 관한 소개를 볼 수 있습니다.
- "솔루션"과 관련된 도움말 문서를 보시려면 [?]을 클릭하십시오.

#### 솔루션 관련 작업

솔루션을 선택하고 마우스 오른쪽 버튼을 클릭하여 **솔루션 닫기**, **탐색기에서 솔루션 열기**, **솔루션 자동 로드**, **솔루션 이름 바꾸기** 등과 같은 작업을 수행할 수 있습니다.



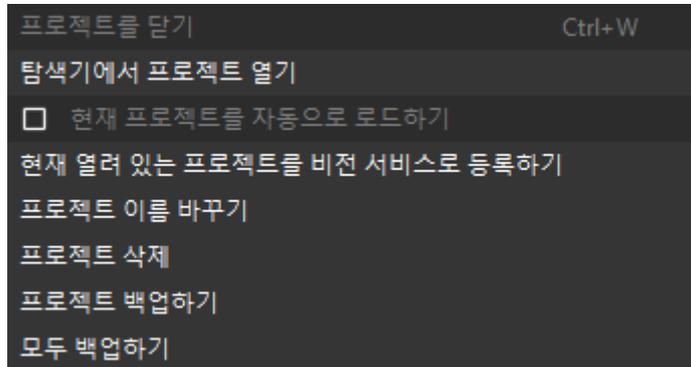
솔루션 닫기  
 탐색기에서 솔루션 열기  
 새로운 프로젝트  
 솔루션 자동 로드  
 솔루션 이름 바꾸기  
 현재 열려 있는 프로젝트를 비전 서비스로 등록하기  
 모두 백업하기

옵션 및 해당 설명은 아래 표와 같습니다.

옵션	설명
프로젝트를 닫기	현재 솔루션 닫기
탐색기에서 솔루션 열기	솔루션이 위치한 폴더를 열기
솔루션 자동 로드	솔루션의 모든 프로젝트가 자동으로 로드되도록 설정
프로젝트 이름 바꾸기	솔루션 이름에 대해 재명명하기
모두 백업하기	솔루션을 백업하기

#### 프로젝트 관련 작업

프로젝트를 선택하고 마우스 오른쪽 버튼을 클릭하여 **프로젝트를 닫기**, **탐색기에서 프로젝트 열기**, **현재 프로젝트를 자동으로 로드하기**, **프로젝트 이름 바꾸기** 등과 같은 작업을 수행할 수 있습니다.



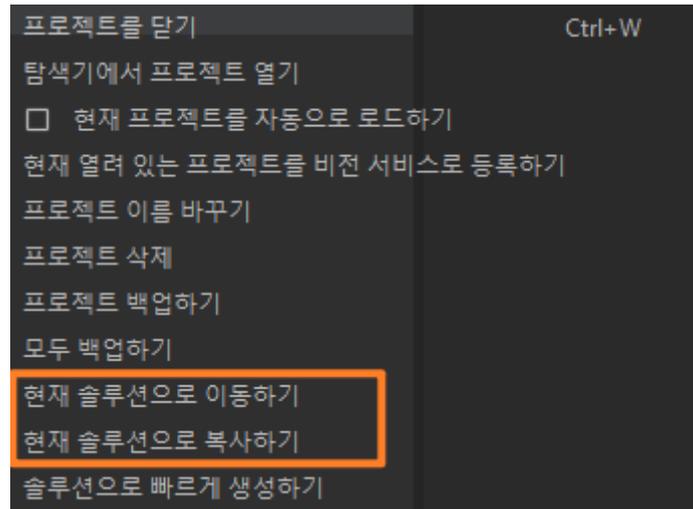
프로젝트를 닫기 Ctrl+W  
 탐색기에서 프로젝트 열기  
 현재 프로젝트를 자동으로 로드하기  
 현재 열려 있는 프로젝트를 비전 서비스로 등록하기  
 프로젝트 이름 바꾸기  
 프로젝트 삭제  
 프로젝트 백업하기  
 모두 백업하기

옵션 및 해당 설명은 아래 표와 같습니다.

옵션	설명	단축키
프로젝트를 닫기	현재 프로젝트 닫기	Ctrl+W
탐색기에서 프로젝트 열기	프로젝트가 위치한 폴더를 열기	없음
현재 프로젝트를 자동으로 로드하기	선택한 프로젝트를 자동으로 로드하도록 설정	없음
프로젝트 이름 바꾸기	선택한 프로젝트의 이름에 대해 재명명하기	없음
프로젝트 삭제	선택한 프로젝트를 삭제하기	없음

옵션	설명	단축키
프로젝트 백업하기	선택한 프로젝트를 백업하기	없음
모두 백업하기	모든 프로젝트를 백업하기	없음
프로젝트 번호 재설정	프로젝트의 번호를 재설정하기	없음

열려 있는 솔루션과 **할당되지 않은** 프로젝트가 있는 경우 해당 프로젝트를 **현재 솔루션으로 이동/복사** 할 수 있습니다.

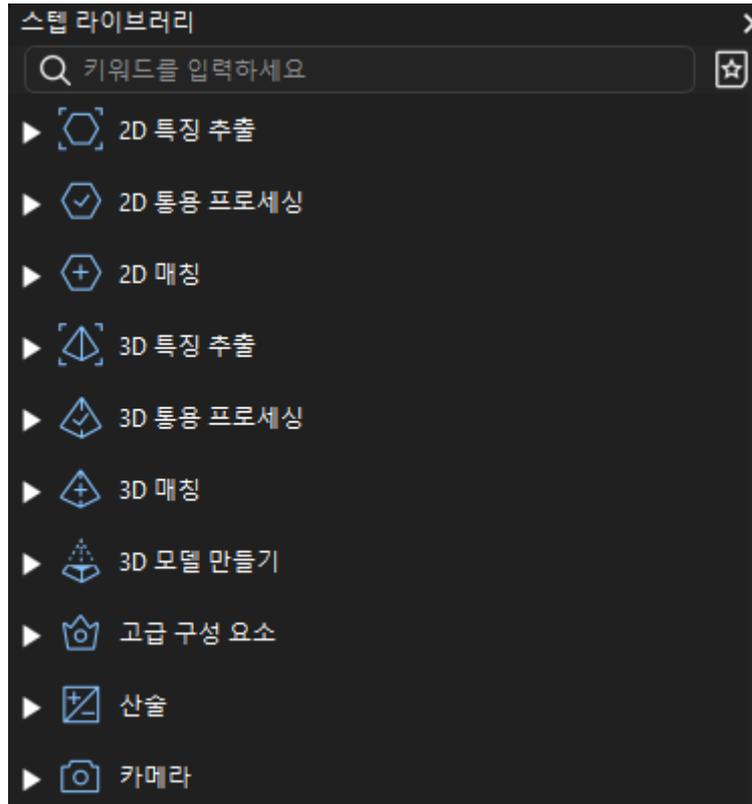


옵션 및 해당 설명은 아래 표와 같습니다.

옵션	설명
현재 솔루션으로 이동하기	선택한 프로젝트를 현재 열려 있는 프로젝트로 이동하기
현재 프로젝트에 복사하기	선택한 프로젝트를 현재 열려 있는 프로젝트에 복사하기

#### 4.1.2.4. 스텝 라이브러리

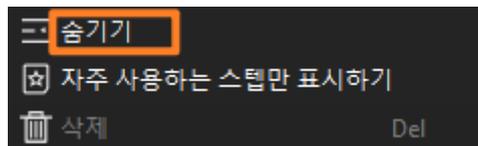
프로젝트를 구축하는 데 사용할 수 있는 모든 스텝은 아래 그림과 같이 스텝 라이브러리에 제공됩니다.



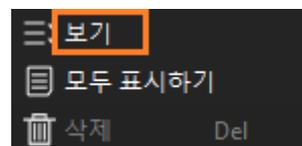
## 스텝 표시

### 숨기기/모두 표시하기

스텝 라이브러리 그룹이 완전히 펼쳐지면 스텝 라이브러리에서 마우스 오른쪽 버튼을 클릭한 다음 **숨기기**를 클릭하여 모든 스텝 라이브러리 그룹을 숨길 수 있습니다.

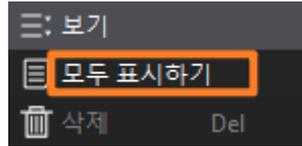


스텝 라이브러리 그룹이 숨겨진 상태일 때 스텝 라이브러리에서 마우스 오른쪽 버튼을 클릭한 다음 **모두 표시하기**를 클릭하여 스텝 라이브러리 그룹을 완전히 볼 수 있습니다.



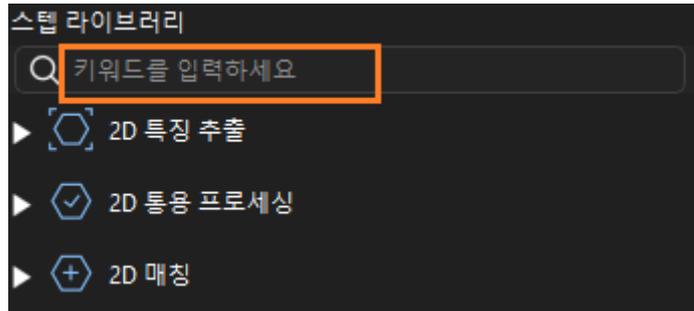
### 모두 표시하기/자주 사용하는 스텝만 표시하기

검색창 오른쪽에 있는 버튼이  이면 자주 사용하는 스텝이 스텝 라이브러리에 표시됩니다. 모든 스텝을 보려면 검색 표시줄 오른쪽에 있는  버튼을 클릭하여  모양으로 변경해야 모든 스텝이 표시됩니다. 또는 스텝 라이브러리를 마우스 오른쪽 버튼으로 클릭하고 [ **모두 표시하기** ]를 클릭합니다.

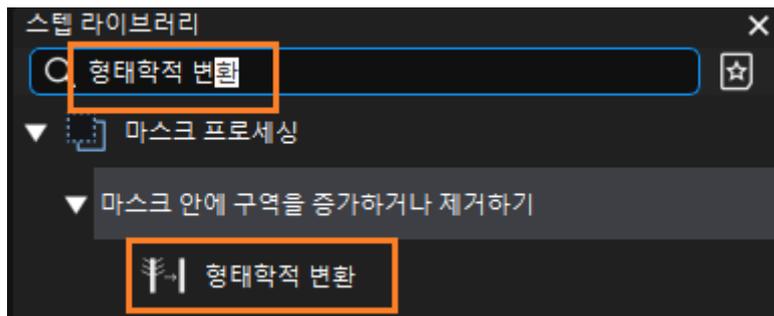


### 스텝 검색

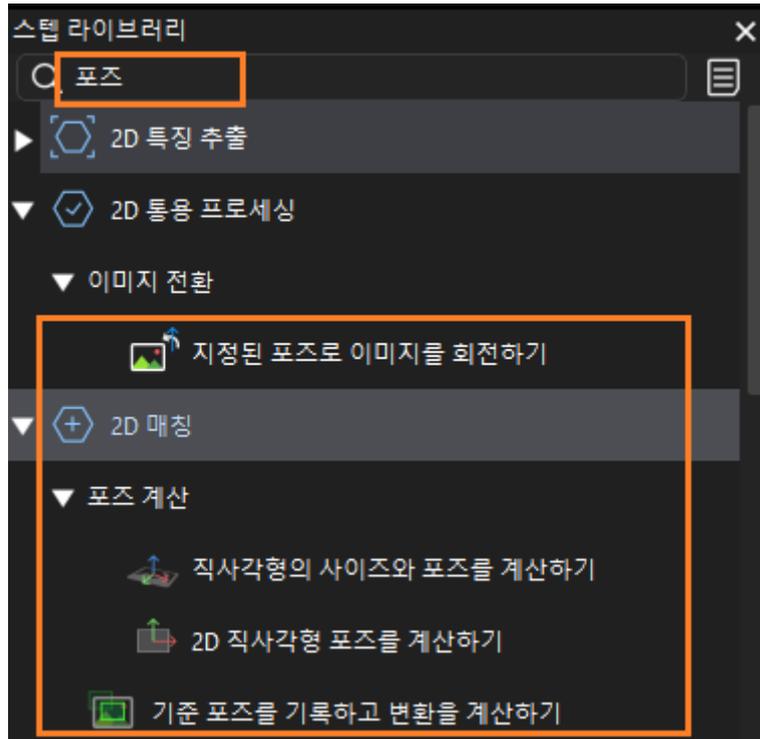
관련 스텝을 검색하려면 스텝 라이브러리 검색 상자에 스텝 이름 정보를 입력하십시오. 스텝 이름 정보를 입력할 때 스텝의 전체 이름을 입력하거나 스텝 이름의 주요 정보를 입력할 수 있습니다.



스텝의 전체 이름을 입력하면 이 스텝을 정확하게 검색할 수 있습니다. 아래 그림과 같이 "형태학적 변환"을 입력하는 것을 예로 들어 보겠습니다.

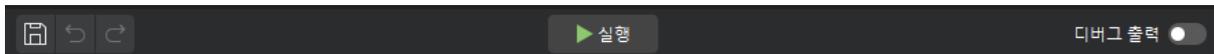


스텝의 주요 정보를 입력하면 해당 정보와 관련된 스텝이 검색됩니다. 아래 그림과 같이 "포즈"를 입력하는 것을 예로 들어 보겠습니다.



#### 4.1.2.5. 프로젝트 툴 바

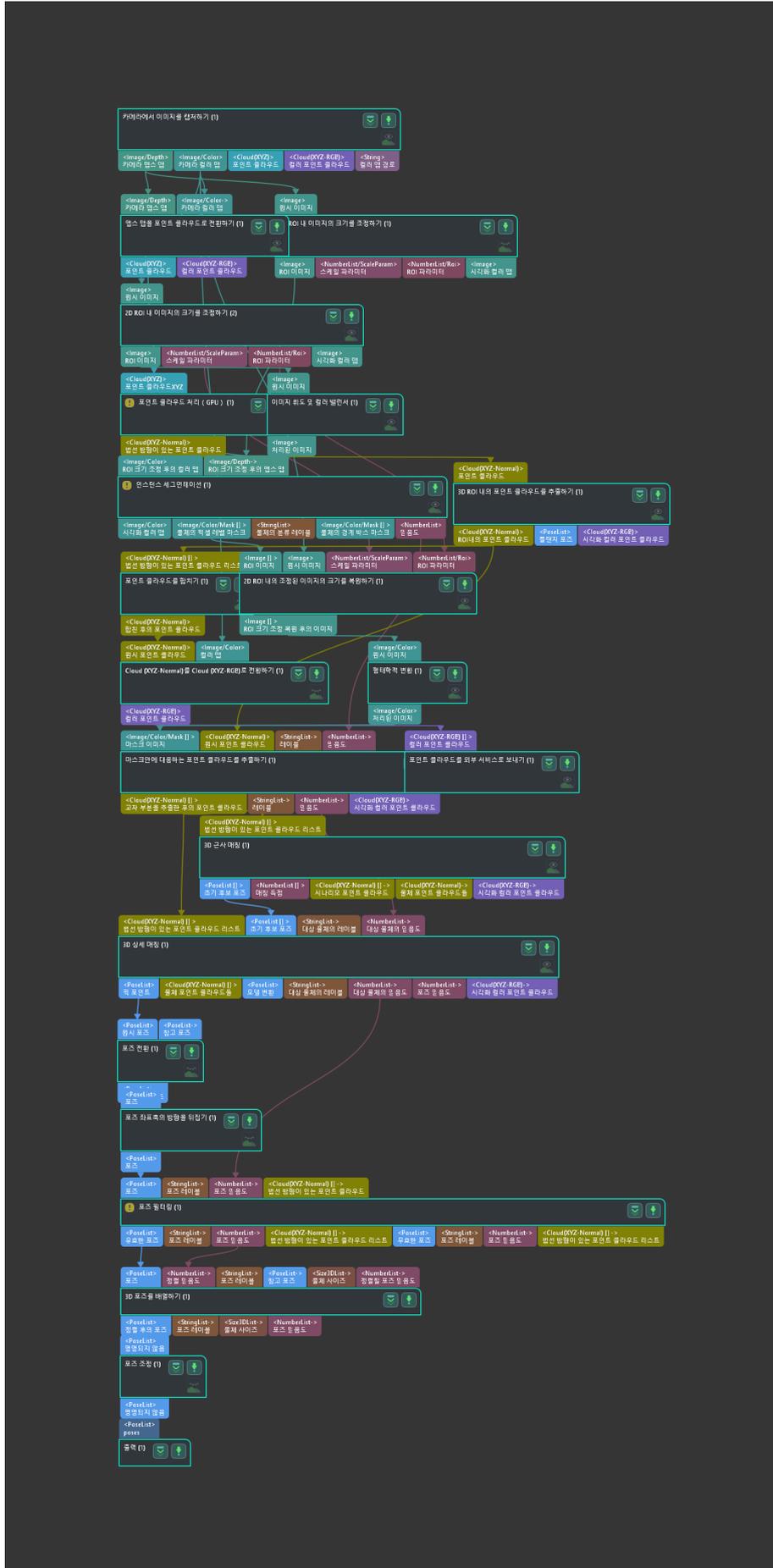
프로젝트 저장, 실행, 디버그 출력 등의 작업을 완료할 수 있습니다.



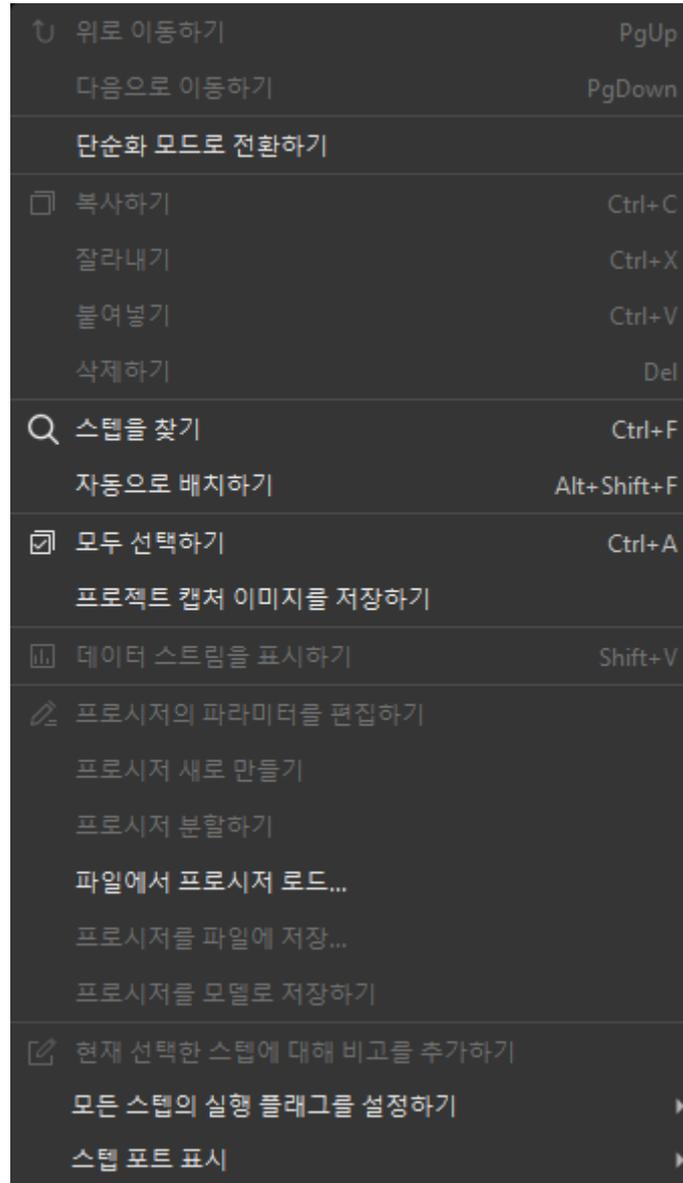
옵션	설명
저장	프로젝트를 저장하기
취소	작업을 취소하기
다시 하기	작업을 다시 하기
실행	프로젝트를 실행하기
디버그 출력	디버그 출력이 켜져 있으면 기타 디버그 정보가 표시됨

#### 4.1.2.6. 프로젝트 편집 구역

프로젝트 편집 구역은 소프트웨어 디버깅 프로세스에서 중요한 구역이며 주요 목적은 비전 프로젝트 처리 흐름 또는 논리 흐름을 편집하는 것입니다. 표준 모드에서 선은 데이터의 흐름을 나타내고 측정 모드에서는 선이 처리 흐름의 흐름을 나타냅니다.



프로젝트 편집 구역에서 마우스 오른쪽 버튼으로 클릭하면 다양한 기능 옵션이 나올 것이며 실제 수요에 근거하여 옵션을 선택하십시오.



각 옵션에 관한 설명은 아래와 같습니다.

옵션	설명	단축키
위로 이동하기	프로시저의 지난 화면으로 돌아갑니다.	PgUp
다음으로 이동하기	프로시저의 다음 화면으로 들어갑니다(마우스 오른쪽 버튼으로 프로시저를 클릭해야 이 옵션이 활성화될 수 있음).	PgDown
단순화 모드로 전환하기	현재 프로젝트에 사용된 모든 스텝을 단순화 모드로 전환합니다.	없음.

옵션	설명	단축키
복사하기	스텝을 복사합니다.	Ctrl+C
잘라내기	스텝을 잘라냅니다.	Ctrl+X
붙여넣기	스텝을 붙여넣습니다.	Ctrl+V
삭제하기	스텝 또는 연결선을 삭제합니다.	Del
스텝을 찾기	현재 프로젝트에서 원하는 스텝을 찾습니다.	Ctrl+F
자동으로 배치하기	현재 프로젝트에 사용된 모든 스텝을 자동으로 배치합니다.	Alt+Shift+F
모두 선택하기	현재 프로젝트에 있는 모든 스텝과 데이터 스트림을 선택합니다.	Ctrl+A
프로젝트 캡처 이미지를 저장하기	프로젝트 캡처 이미지를 지정한 경로로 저장합니다.	없음.
데이터 스트림을 표시하기	디버그 출력 창에서 입력&출력 데이터를 표시합니다(마우스 오른쪽 버튼으로 데이터 스트림을 클릭할 때 이 옵션이 활성화될 수 있음).	Shift+V
프로시저의 파라미터를 편집하기	프로시저의 파라미터를 편집합니다(마우스 오른쪽 버튼으로 프로시저를 클릭할 때 이 옵션이 활성화될 수 있음).	없음.
프로시저 새로 만들기	여러 스텝을 선택한 경우 여러 스텝을 프로시저로 생성할 수 있습니다.	없음.
프로시저 분할하기	하나의 프로시저를 선택하여 여러 스텝으로 분할합니다.	없음.
파일에서 프로시저 로드하기	기존의 .json 파일에서 프로시저를 로드합니다.	없음.
프로시저를 파일에 저장하기	프로시저를 .json 파일로 저장합니다.	없음.
프로시저를 모델로 저장하기	선택한 프로시저를 "사용자 지정 그룹화"에 저장합니다.	없음.
현재 선택한 스텝에 대해 비교를 추가하기	현재 선택한 스텝에 대해 비교를 추가합니다.	없음.
모든 스텝의 실행 플래그를 설정하기	텍스트 출력을 취소하기	없음.
	파일 다시 로드 설정/취소	없음.
	출력이 없을 경우 계속 실행하기 설정/취소	없음.

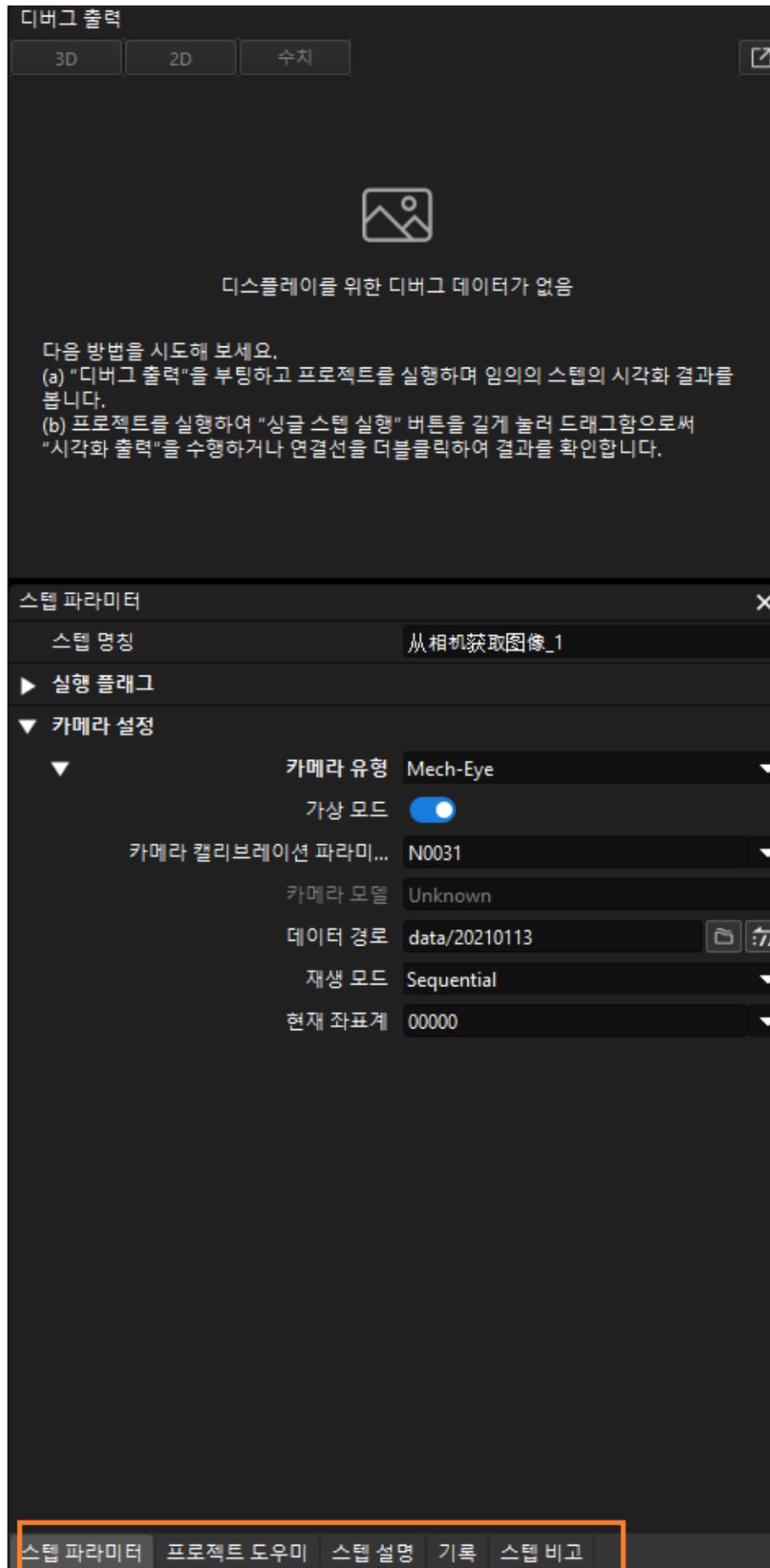
옵션	설명	단축키
스텝 포트 표시 설정	유형 및 이름을 표시하기	Ctrl+Alt+A
	이름만 표시하기	Ctrl+Alt+N
	유형만 표시하기	Ctrl+Alt+T

#### 4.1.2.7. 프로젝트 배치 구역

프로젝트 설정 구역에는 디버그 출력, 스텝 파라미터, 프로젝트 도우미, 스텝 설명, 기록 및 스텝 비교 등 내용이 포함되어 있습니다. 아래 그림과 같습니다.



프로젝트 설정 구역에 탭이 표시되지 않으면 **메뉴 바 > 뷰**에서 항목이 선택되었는지 확인할 수 있습니다.



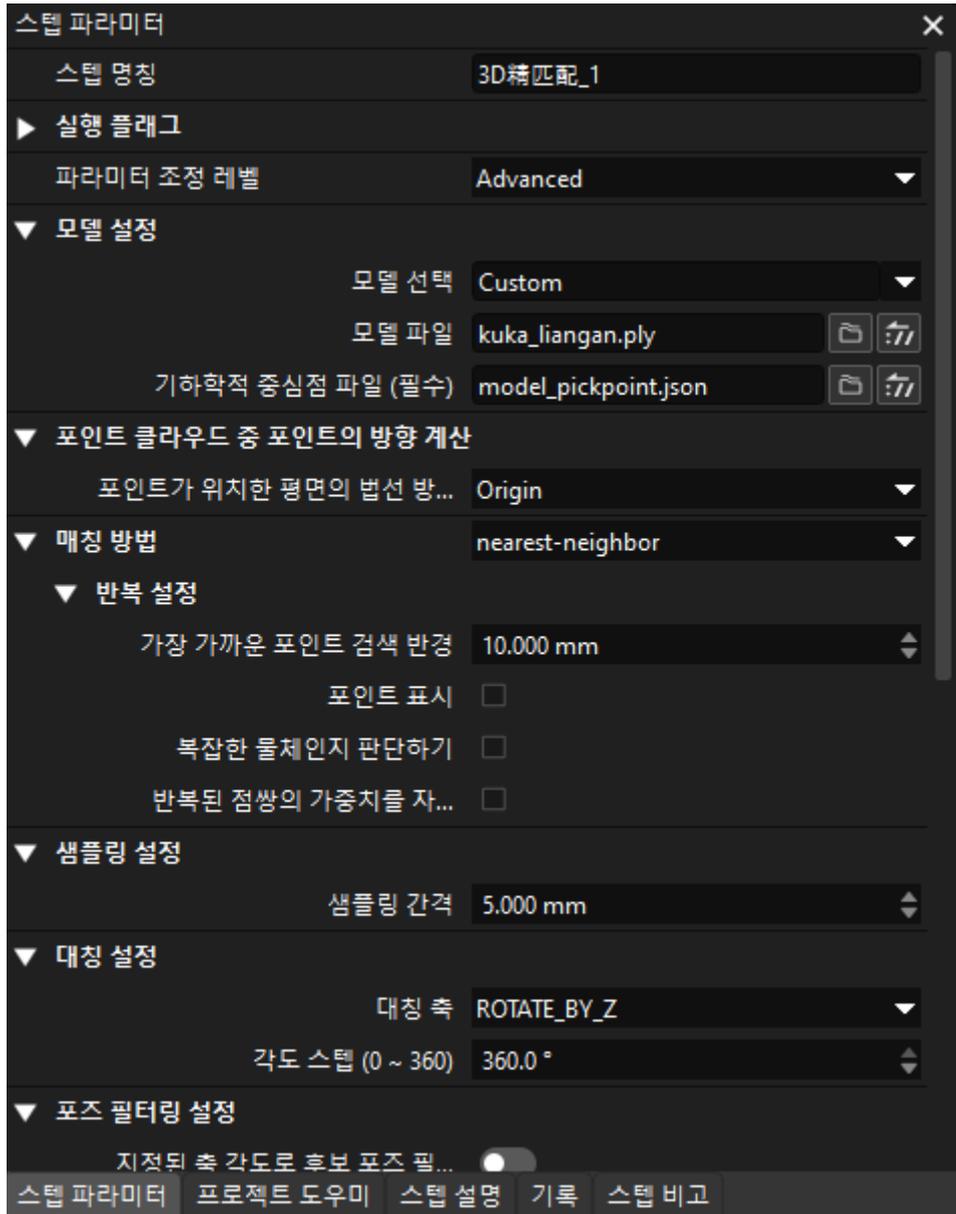
## 디버그 출력

디버그 출력을 열면 실행할 때 필요한 스텝의 출력 결과가 표시됩니다. 자세한 내용은 [스텝을 실행하고 시각화된 출력 결과 보기](#) 를 참고하세요.



## 스텝 파라미터

한 스텝을 선택하면 이 스텝의 파라미터를 조정할 수 있습니다.

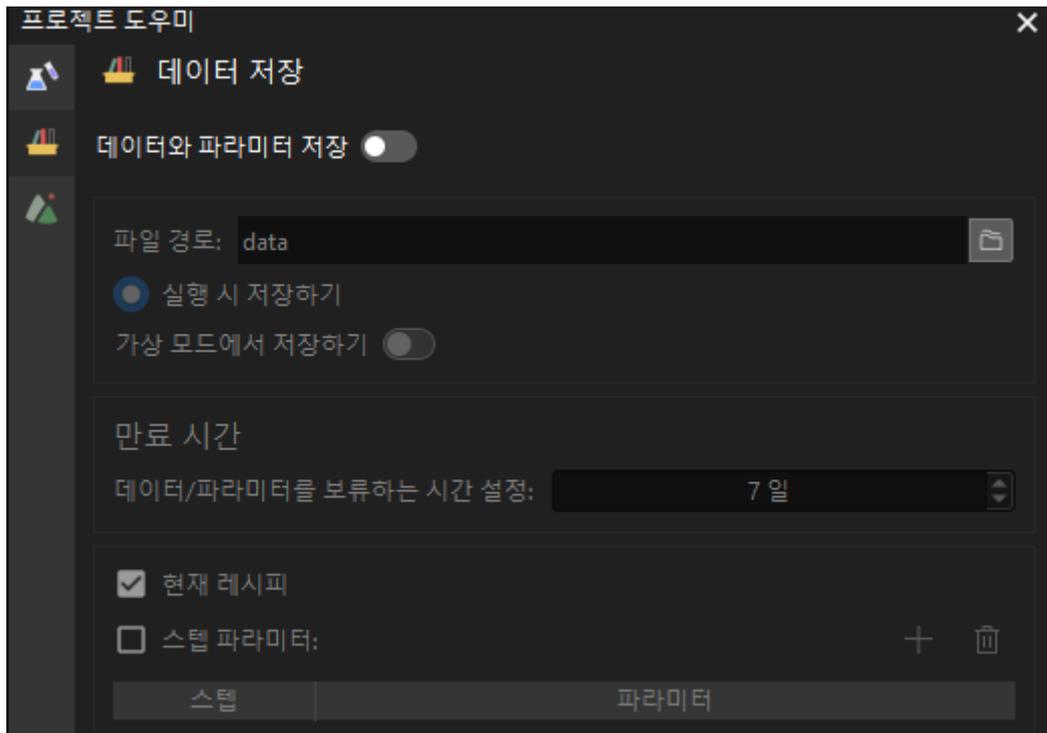


스텝 파라미터 옵션 탭에 기본 파라미터를 기본적으로 표시합니다. 모든 파라미터를 표시하려면 이 탭의 여백을 마우스 오른쪽 버튼으로 클릭한 다음 모든 파라미터를 표시하기를 클릭합니다.



#### 프로젝트 도우미

프로젝트 설정과 관련된 구성 내용을 표시합니다.

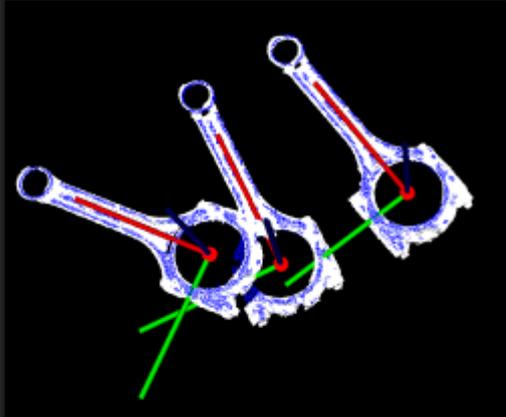


#### 스텝 설명

한 스텝을 선택하면 스텝의 기능, 사용 시나리오 및 입/출력에 대한 설명을 볼 수 있습니다. 다음 그림과 같습니다.

**스텝 설명** ✕

**기능 설명:**  
 포인트 클라우드 모델과 원시 포인트 클라우드와 상세 매칭을 하여 대상 물체의 정확한 포즈를 출력합니다.



**응용 시나리오:**  
 이 스텝은 "3D 근사 매칭"을 통해 출력된 초기 후보 포즈를 기반으로 하여 더 상세한 매칭을 실현해 대상 물체의 정확한 포즈를 출력합니다.

[상세 정보\(F1\)...](#)

**입력:**

- 이 포트에 입력한 포인트 클라우드는 매칭될 것입니다.(보통 스텝 "3D 근사 매칭"과 같은 포인트 클라우드를 사용함).
- 후보 포즈 리스트(보통 스텝 "3D 근사 매칭"에서 출력한 근사한 포즈).
- 보통 딥 러닝과 관련된 스텝에서 출력된 대상 물체의 레이블(선택 가능).
- 보통 딥 러닝과 관련된 스텝에서 출력된 대상 물체의 믿음도(선택 가능).

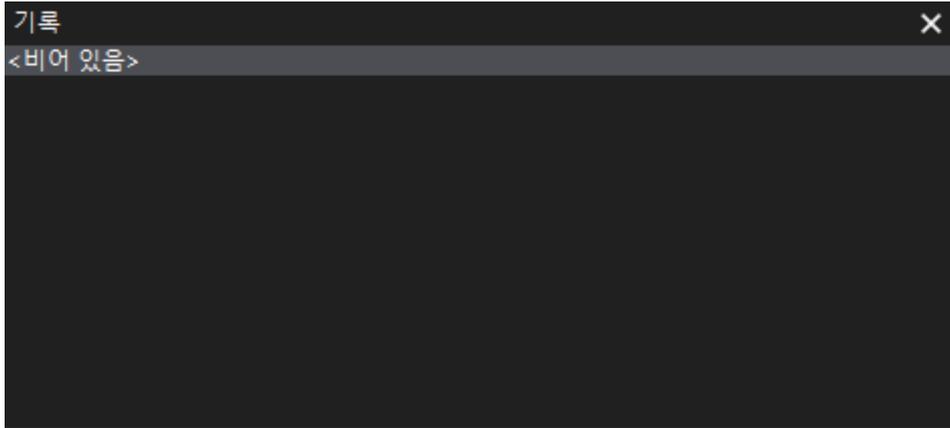
**출력:**

- 픽 포인트 리스트.
- 매칭 결과에 따라 생성된 모델 포인트 클라우드 리스트.
- 모델에서 시나리오까지의 변환 참조 포즈 리스트.
- 대상 물체의 레이블(딥 러닝 스텝 관련).
- 대상 물체의 믿음도(딥 러닝 스텝 관련).
- 물체 포즈에 해당하는 믿음도 리스트.

 F1 버튼을 클릭하면 해당 안내서 페이지로 이동할 수 있습니다.

## 역사

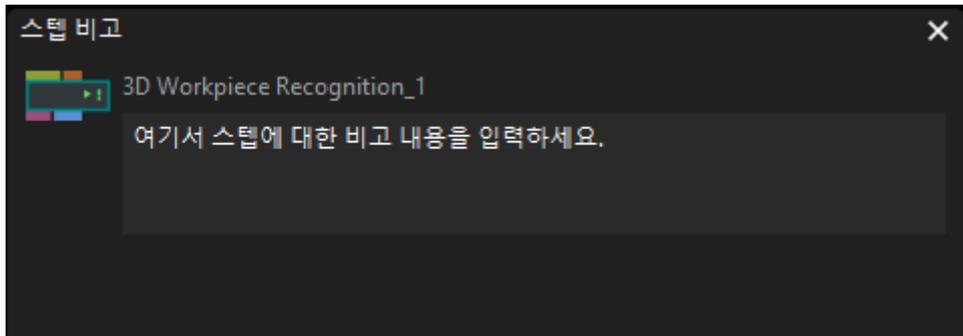
작업 역사를 기록하여 표시합니다. 항목을 클릭하면 이 항목 후의 모든 작업 스텝을 취소할 수 있습니다.



💡 측정 모드 입력이나 측정 결과 출력 창을 사용하려면 뷰에서 선택하세요.

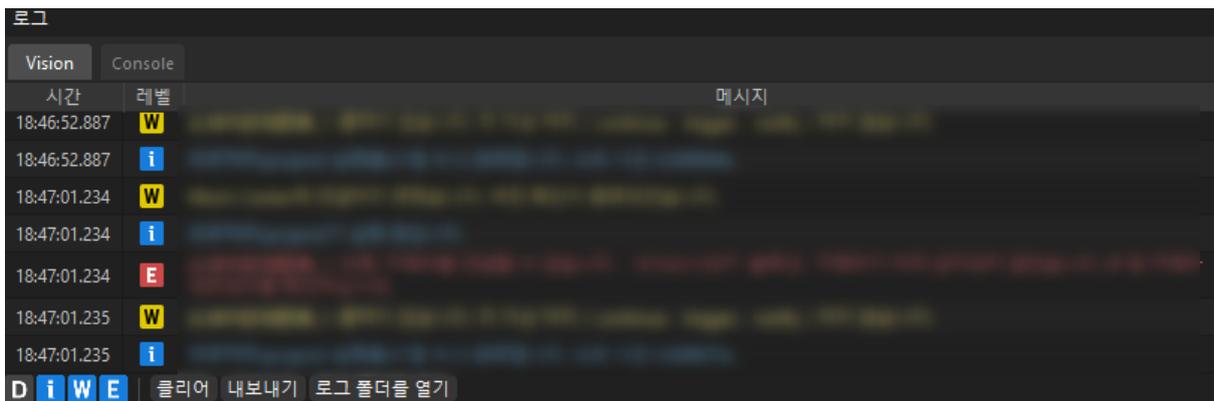
### 스텝 비교

스텝을 선택하거나 추가하거나 비교 내용을 확인합니다. 자세한 내용은 [스텝 추가](#), [수정](#), [비교 보기](#)를 참조하세요.



### 4.1.2.8. 로그바

로그바는 사용자들이 구체적인 시각의 솔루션/프로젝트 실행 기록을 쉽게 찾을 수 있도록 프로젝트 실행의 로그 정보를 실시간으로 표시합니다.



로그 레벨에 대한 설명은 아래와 같습니다.

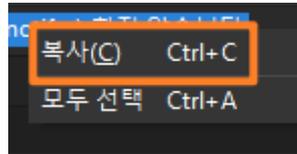
로그 레벨	설명
D	결함 수정(Debug)

로그 레벨	설명
i	정상 정보(Info)
W	경고(Warning)
E	오류(Error)

로그 표시줄의 왼쪽 하단에 있는 로그 레벨을 클릭하여 표시할 레벨 정보를 선택합니다. 로그 레벨 버튼의 색상이 파란색이면 표시되고 로그 레벨 버튼이 회색이면 표시되지 않습니다.



로그 내용을 더블 클릭하여 내용을 선택하고 마우스 왼쪽 버튼을 클릭하여 복사 버튼을 누르면 선택한 내용을 복사할 수 있습니다.

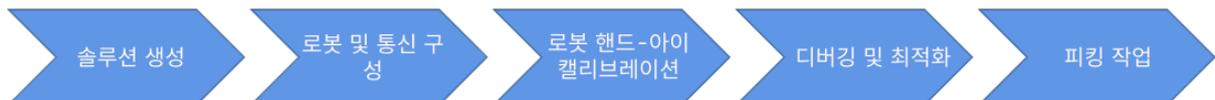


### 4.1.3. 솔루션 구축 과정

이 부분에서는 솔루션을 구축하는 방법에 대해 설명합니다.

솔루션은 비전 응용을 실현하는 데 필요한 로봇과 통신, 비전 처리 및 경로 계획 등 기능 구성과 데이터의 모음입니다.

솔루션 배포는 일반적으로 아래 그림과 같이 5개 단계로 나뉩니다.



#### 솔루션 생성

솔루션은 비전 응용을 실현하는 데 필요한 로봇과 통신, 비전 처리 및 경로 계획 등 기능 구성과 데이터의 모음입니다. 솔루션을 배포하기 전에 솔루션을 만들어야 합니다.

1. [\[vision-operation-guide:create-and-save-solution:::create-and-save-solution\]](#)
2. [\[vision-operation-guide:auto-load-solution:::autoload-solution\]](#)
3. 프로젝트 구축

#### 로봇 및 인터페이스 구성

솔루션 생성 후 로봇 및 통신 설정이 필요합니다.

통신은 비전 측과 로봇 측 간의 정보 교환 및 전달 과정을 의미합니다. 비전 측과 로봇 측이 정상적으로 통신할 수 있도록 비전 측 및 로봇 측의 통신 구성을 완료해야 합니다.

Mech-Mind 비전 시스템에서는 인터페이스 통신과 마스터 컨트롤 통신의 두 가지 통신 방식이 제공되며 인터페이스 통신에는 표준 인터페이스와 Adapter가 포함됩니다.

### 인터페이스 통신

인터페이스 통신 방식을 사용하는 경우 비전 측과 로봇 측에 각각 통신 구성을 설정하고 로봇 측과 비전 측 간의 통신이 정상적으로 이루어질 수 있는지 테스트해야 합니다.

로봇 측과 비전 측 간의 통신이 정상적으로 설정될 수 있는지 여부를 테스트하는 방법은 로봇 브랜드마다 다릅니다.

1. 비전 측과의 통신 설정
2. 로봇 측과의 통신 구성, 로봇 측과 비전 측 간의 통신이 정상적으로 설정될 수 있는지 테스트해야 합니다.

### 마스터 컨트롤 통신

마스터 컨트롤 통신 방식을 사용하는 경우 로봇 측에서 통신을 구성한 후 로봇 측과 비전 측 간의 통신이 정상적으로 설정될 수 있는지 테스트해야 합니다.

1. 로봇 측과의 통신 설정
2. 로봇 측과 비전 측 간의 통신이 정상적으로 설정될 수 있는지 테스트해야 합니다.

### 로봇 핸드-아이 캘리브레이션

로봇과 통신 구성이 완료되면 [로봇 핸드-아이 캘리브레이션](#)을 수행해야 합니다.

핸드-아이 캘리브레이션은 카메라 좌표계와 로봇 좌표계 사이의 대응 관계를 설정하는 프로세스를 말하며 비전 시스템에 의해 결정된 물체 포즈를 로봇 좌표계 아래의 포즈로 변환하여 로봇이 물체에 대한 작업을 완료하도록 가이드합니다.

로봇 핸드-아이 캘리브레이션은 매우 중요한 절차입니다. 캘리브레이션 결과의 정확성과 안정성은 로봇의 피킹 결과의 정확성에 영향을 미칩니다.

### 디버그 및 최적화

위의 작업이 완료되면 프로젝트를 실행하고 디버깅할 수 있습니다.

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

### 피킹 작업

프로젝트 운영 및 디버깅이 완료되면 실제 로봇을 피킹 작업에 사용할 수 있습니다.

- 인터페이스 통신을 사용하는 경우 표준 인터페이스를 통해 Mech-Vision 프로젝트를 트리거하고 Mech-Vision 프로젝트에서 출력되는 비전 포인트 또는 계획된 경로 포인트를 얻기 위해 로봇 피킹 프로그램을 작성해야 합니다.
- 마스터 컨트롤 통신을 사용하는 경우 Mech-Viz 프로젝트를 구축하여 경로 포인트를 계획해야 합니다.

## 4.2. 일반 사용 가이드

이 부분에서는 Mech-Vision 관련 사용 가이드에 대해 소개합니다.

다음과 같은 내용을 통해 **솔루션 라이브러리 사용 가이드**에 대해 알아보십시오.

[솔루션 라이브러리 사용 가이드](#)

---

다음과 같은 내용을 통해 **솔루션, 프로젝트 사용 가이드**에 대해 알아보십시오.

[솔루션 사용 가이드](#)

[프로젝트 사용 가이드](#)

---

다음과 같은 내용을 통해 **통신 관련 사용 가이드**에 대해 알아보십시오.

[로봇 및 인터페이스 구성 사용 가이드](#)

---

다음과 같은 내용을 통해 **스텝 관련 사용 가이드**에 대해 알아보십시오.

[스텝 사용 가이드](#)

## 4.2.1. 솔루션 라이브러리 사용 가이드

Mech-Vision 솔루션 라이브러리에서 다양한 산업 분야의 예제 솔루션 또는 프로젝트를 볼 수 있으며 예제 데이터가 포함되어 있습니다.

이 부분에서는 솔루션 라이브러리와 관련된 기본 작업에 대해 소개합니다.

[솔루션 라이브러리 열기](#)

[솔루션 또는 프로젝트 필터링/찾기](#)

[솔루션 또는 프로젝트 정보 보기](#)

[솔루션 또는 프로젝트에 대한 사용자 안내서 보기](#)

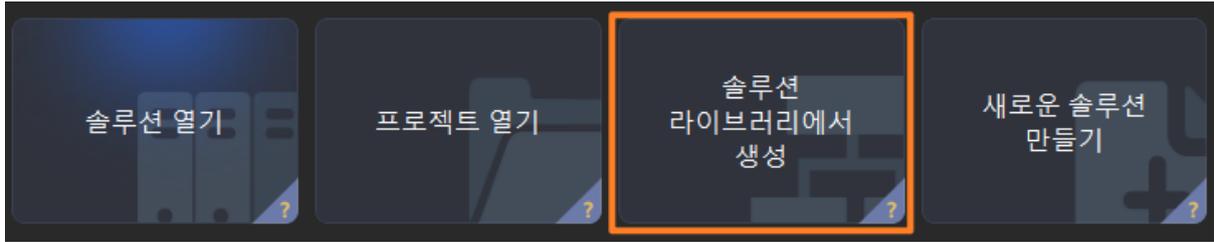
[솔루션 라이브러리에서 솔루션 생성](#)

### 4.2.1.1. 솔루션 라이브러리 열기

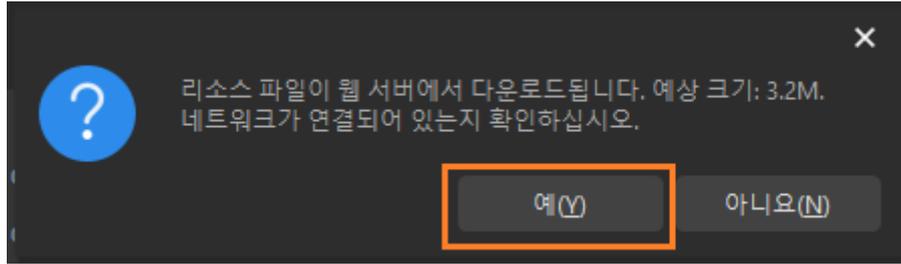
이 부분에서는 솔루션 라이브러리를 여는 방법을 소개합니다.

**시작 화면에서 솔루션 라이브러리를 엽니다.**

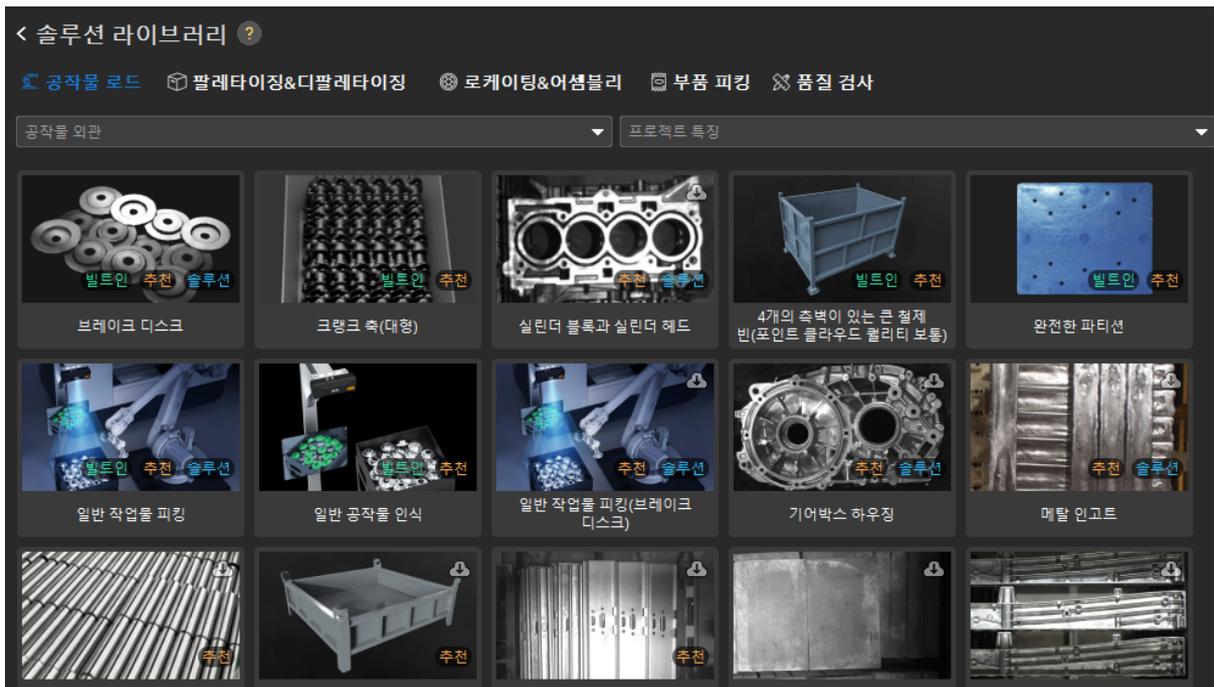
Mech-Vision 소프트웨어를 연 후 시작 화면에서 [ **솔루션 라이브러리에서 새로 만들기** ]를 클릭하여 솔루션 라이브러리를 엽니다.



솔루션 라이브러리에 미리 설치된 리소스가 있습니다. 더 많은 리소스를 얻으려면 솔루션 라이브러리 하단의 [더 많은 정보]를 클릭한 다음 팝업 창에서 [예(Y)]를 클릭하면 솔루션 라이브러리 리소스를 가져올 수 있습니다.



리소스가 로드된 후 아래 그림과 같이 솔루션 라이브러리에서 더 많은 리소스를 볼 수 있습니다.



- 솔루션: 비전 애플리케이션을 구현하는 데 필요한 로봇 및 통신, 비전 처리, 경로 계획 및 기타 기능 구성과 데이터의 모음입니다.
- 추천: 이 프로젝트 모델은 현장에서 여러 번 검증되었으며 적용 범위가 넓습니다.
- 빌트인: 프로젝트 모델이 소프트웨어 설치 패키지와 함께 로컬 디렉터리에 설치되어 다운로드 없이 사용할 수 있습니다.

#### 소프트웨어의 메인 인터페이스에서 솔루션 라이브러리 열기

소프트웨어의 메인 인터페이스에 이미 들어간 경우 다음 세 가지 방법에 따라 솔루션 라이브러리를 열 수 있습니다.

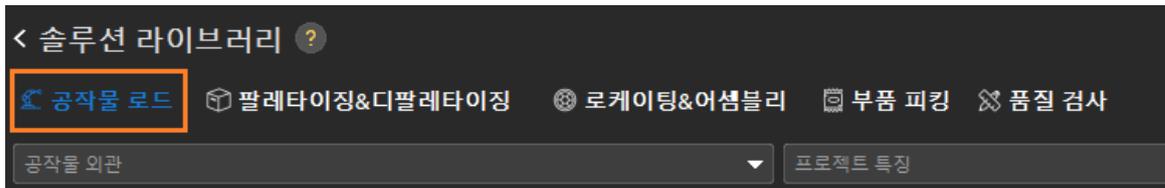
- 툴 바에서 [ **솔루션 라이브러리에서 새로 만들기** ]를 클릭합니다.
- 메뉴 바에서 **파일 > 솔루션 라이브러리에서 새로 만들기**를 클릭합니다.

#### 4.2.1.2. 솔루션 또는 프로젝트 필터링/찾기

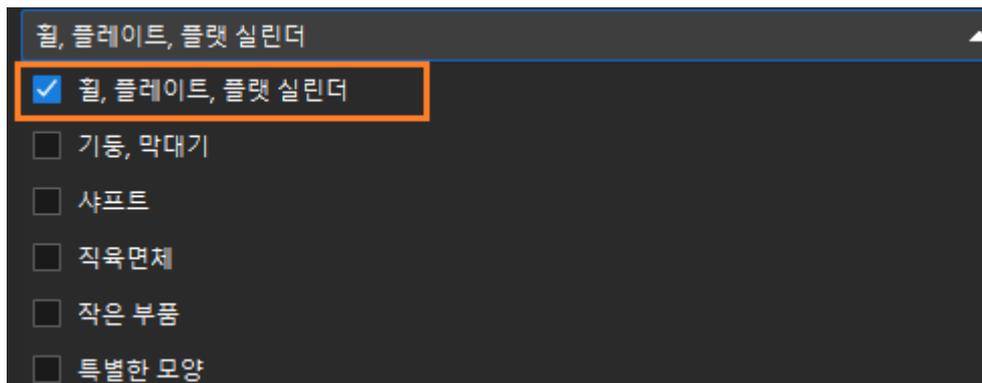
솔루션 라이브러리에서 **산업**에 따라 필요한 솔루션 또는 프로젝트를 찾을 수 있습니다. 그 중 공작물 로드 산업은 **공작물 외관** 및 **프로젝트 특징**을 통해 필요한 솔루션 또는 프로젝트를 필터링할 수 있습니다.

**공작물 로드** 산업에서 **혼합 물체 사양**이 있는 **브레이크 디스크** 프로젝트를 예로 들면 필터링 방법은 다음과 같습니다.

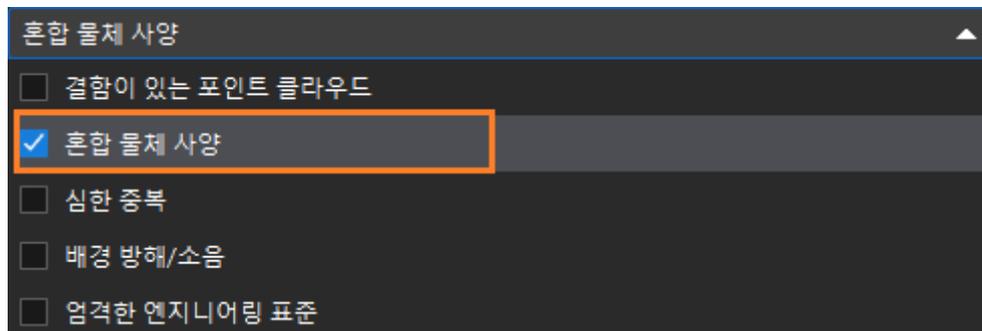
1. 산업을 필터링합니다. 솔루션 라이브러리 상단에서 [ **공작물 로드** ]를 클릭하여 공작물 로드 산업의 솔루션 또는 프로젝트를 필터링하십시오.



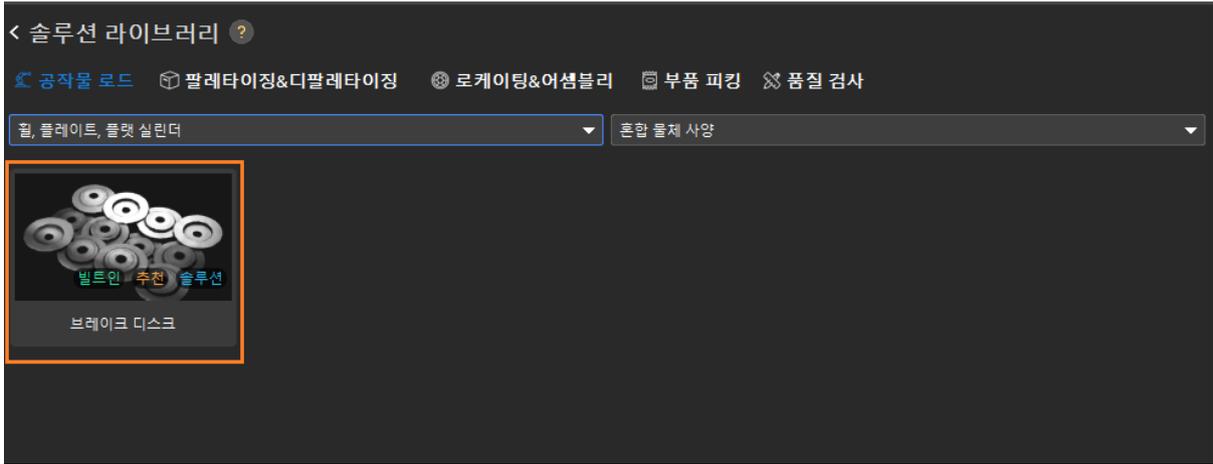
2. 공작물 외관을 필터링합니다. 공작물 외관의 드롭다운 열에서 **휠, 플레이트, 플랫 실린더**를 선택하여 공작물 로드 산업에서 관련된 솔루션 또는 프로젝트를 필터링하십시오.



3. 프로젝트 특징을 필터링합니다. 프로젝트 특징의 드롭다운 열에서 **혼합 물체 사양**을 선택하여 공작물 로드 산업에서 관련된 솔루션 또는 프로젝트를 필터링하십시오.



이로써 필터링 한 후의 솔루션 또는 프로젝트에서 브레이크 디스크 프로젝트를 얻을 수 있습니다.

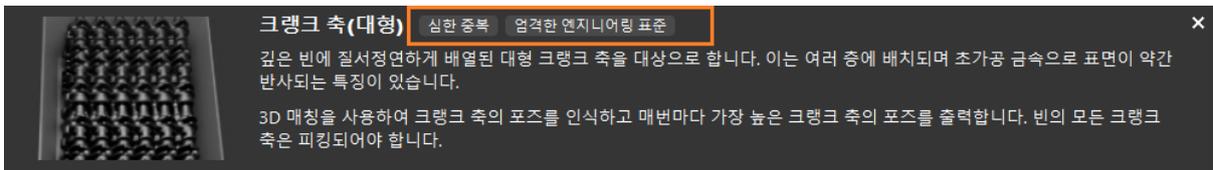


#### 4.2.1.3. 솔루션 또는 프로젝트 정보 보기

임의의 솔루션이나 프로젝트의 관련 정보를 보고 싶다면 솔루션 라이브러리에서 그 솔루션이나 프로젝트를 선택하여 솔루션 라이브러리 창 하단에서 관련 정보를 볼 수 있습니다.



그 중 프로젝트 이름 오른쪽은 아래 그림과 같이 **프로젝트 특징** 레이블입니다.



#### 4.2.1.4. 솔루션 또는 프로젝트에 대한 사용자 안내서 보기

솔루션 또는 프로젝트에 대한 도움말 문서에는 관련 솔루션 또는 프로젝트에 적용 가능한 시나리오, 기술 지표, 프로젝트 난이도, 비전 솔루션, 파라미터 조정 제안 등 내용이 포함됩니다.

솔루션 또는 프로젝트에 도움말 문서가 있는 경우 솔루션 또는 프로젝트를 클릭하면 솔루션 라이브러리 창

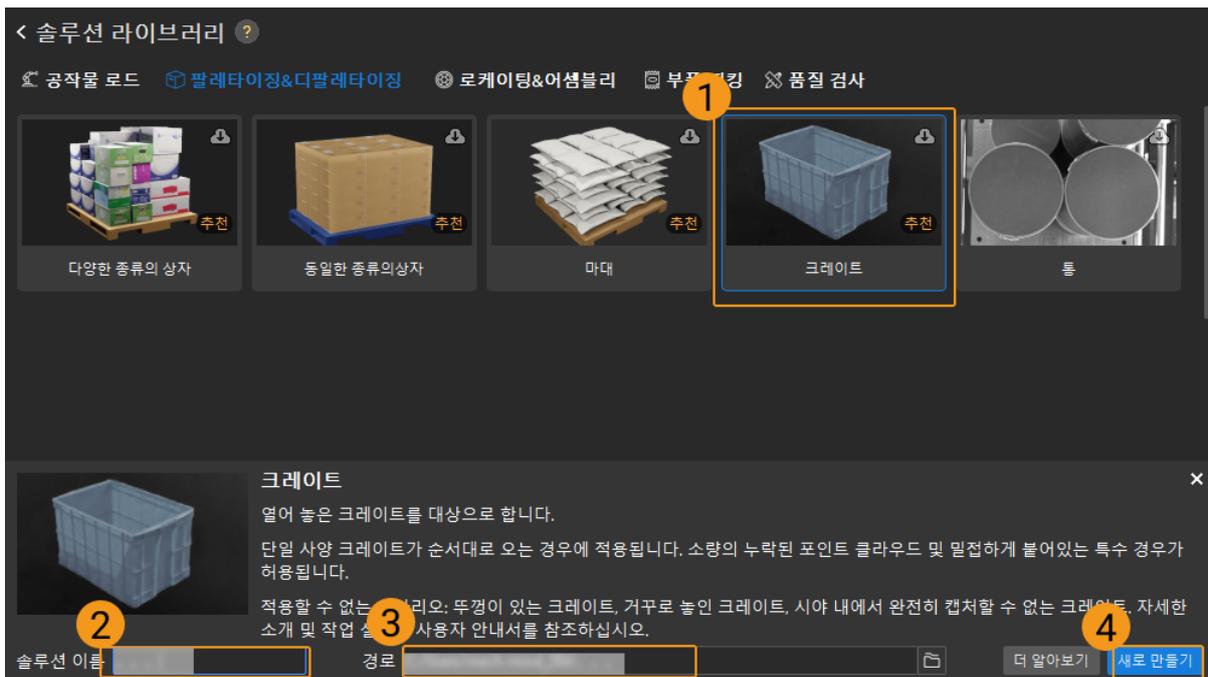
하단에 [ 더 많은 정보 ] 버튼이 표시됩니다.

솔루션 또는 프로젝트의 도움말 문서를 보려면 솔루션 라이브러리에서 솔루션 또는 프로젝트를 클릭하여 선택한 하단에 [ 더 많은 정보 ] 버튼을 클릭하면 팝업된 페이지에서 볼 수 있습니다.

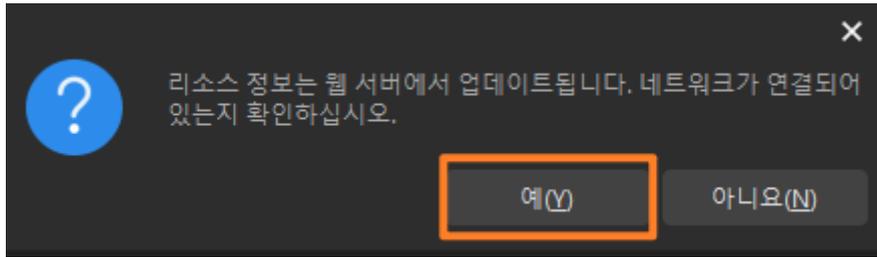


#### 4.2.1.5. 솔루션 라이브러리에서 솔루션 생성

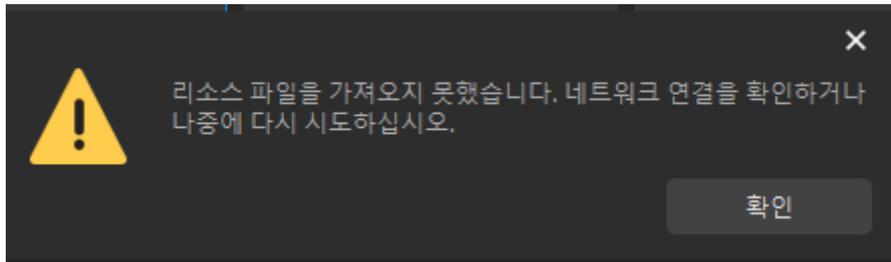
솔루션 라이브러리에서 솔루션을 생성하려면 [솔루션 라이브러리 열기](#)를 클릭하고 먼저 솔루션을 클릭하여 선택한 다음 **솔루션 이름** 및 **경로**를 입력해야 합니다. 마지막으로 솔루션 라이브러리 하단에서 [ **새로 만들기** ]를 클릭합니다.



팝업 창에서 [예(Y)]를 클릭하여 지정된 경로에 솔루션을 생성합니다.



아래와 같은 창이 뜨면 백그라운드에 해당 솔루션이 없다는 뜻입니다. 먼저 네트워크 연결을 확인하시고, 네트워크 연결이 정상인지 확인한 후 다시 시도해 주시고, 그래도 문제가 발생하면 Mech-Mind 엔지니어에게 문의하시기 바랍니다.



## 4.2.2. 솔루션 사용 가이드

솔루션은 비전 응용을 실현하는 데 필요한 로봇과 통신, 비전 처리 및 경로 계획 등 기능 구성과 데이터의 모음입니다.

이 부분에서는 솔루션과 관련된 기본 작업을 소개할 것입니다.

[솔루션 생성 및 저장](#)

[솔루션 열기](#)

[솔루션 자동 로드](#)

[프로젝트 이름 바꾸기](#)

[프로젝트를 닫기](#)

[솔루션 구조 이해하기](#)

### 4.2.2.1. 솔루션 생성 및 저장

이 부분에서는 솔루션을 새로 만들고 저장하는 방법을 소개합니다.

**새로운 솔루션**

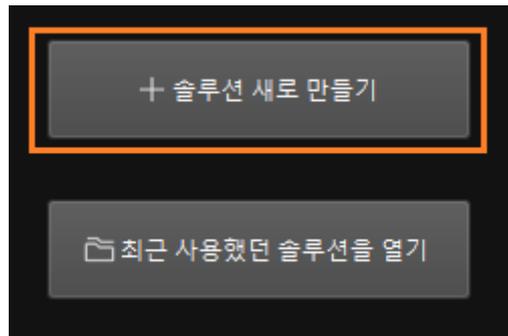
**새로운 솔루션 만들기**

새로운 솔루션을 만드는 방법은 다음과 같이 세 가지가 있습니다.

- 시작 화면에서 [ **새로운 솔루션 만들기** ]를 클릭하여 새로운 솔루션을 만듭니다.



- 메인 인터페이스의 메뉴 바에서 **파일 > 새로운 솔루션**을 선택하여 새로운 솔루션을 만듭니다.
- 소프트웨어가 솔루션이나 프로젝트를 열지 않으면 프로젝트 편집 구역에서 [**새로운 솔루션**]을 클릭하여 새로운 솔루션을 만들 수 있습니다.



#### 솔루션 라이브러리에서 솔루션 새로 만들기

솔루션 라이브러리에서 솔루션 또는 프로젝트 새로 만들기를 참조하십시오.

#### 솔루션 저장하기

솔루션이 생성되면 사용자는 솔루션을 제때 저장할 수 있도록 저장 경로를 설정하는 것이 좋습니다.

솔루션을 저장하는 두 가지 방법은 아래와 같습니다.

- 메뉴 바에서 **파일 > 프로젝트 저장**을 선택합니다.
- **Ctrl + Shift + S** 단축키를 눌러 새 솔루션을 만듭니다.



- 솔루션의 이름을 변경하려면 **솔루션 이름 바꾸기**를 참조하십시오.
- 다음에 소프트웨어를 시작할 때 자동으로 솔루션을 열려면 **솔루션을 자동으로 로드하기**를 참조하십시오.

#### 4.2.2.2. 솔루션 열기

이 부분에서는 솔루션을 여는 방법에 대해 설명합니다.

##### 기존 솔루션 열기

기존 솔루션을 여는 방법에는 두 가지가 있습니다.

- 시작 화면에서 [**솔루션 열기**]를 클릭한 다음 팝업한 파일 선택 창에서 열려는 솔루션 폴더를 선택합니다.
- 메인 인터페이스의 메뉴 바에서 **파일 > 솔루션 열기**를 클릭한 다음 열려는 솔루션 폴더를 선택합니다.

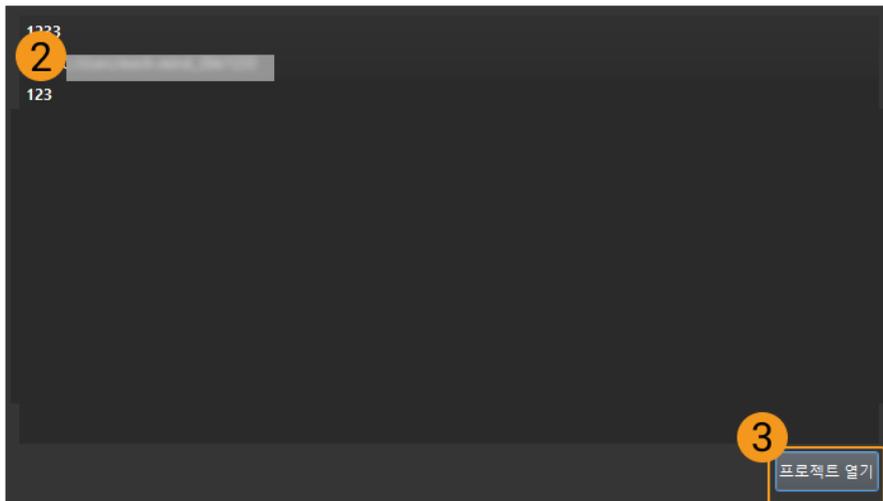
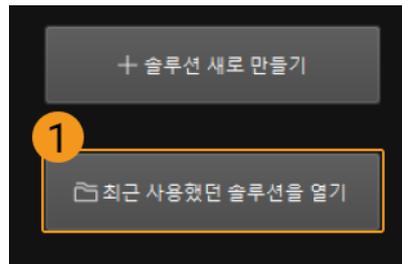
### 최근 사용했던 솔루션을 열기

다음 세 가지 방법을 사용하여 최근에 열었던 솔루션을 열 수 있습니다.

- 시작 화면의 **최근에 사용됨**에서 오른쪽에 있는 솔루션/프로젝트 전환 버튼을 **솔루션**으로 전환한 다음 열려는 솔루션을 선택하고 두 번 클릭합니다.



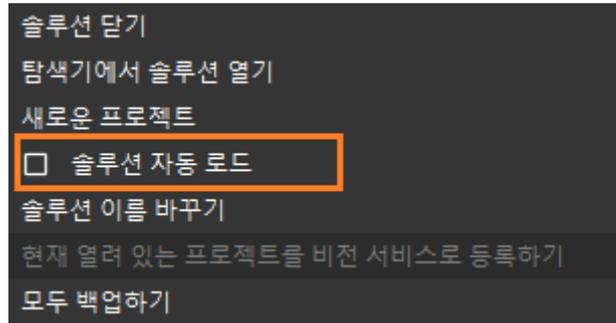
- 소프트웨어 메인 인터페이스의 메뉴 바를 클릭하고 **파일 > 최근에 사용했던 프로젝트를 열기**를 클릭하여 최근에 연 프로젝트를 찾아 바로 엽니다.
- 메인 인터페이스에 열려 있는 솔루션이 없을 때 프로젝트 편집 구역에서 [ **최근 사용했던 솔루션을 열기** ]를 클릭한 다음 팝업 창에서 열려는 솔루션을 선택합니다. 마지막으로  버튼을 클릭합니다.



#### 4.2.2.3. 솔루션 자동 로드

다음에 Mech-Vision 소프트웨어를 열어야 하는 경우 소프트웨어와 함께 임의의 솔루션이 열리므로 다음 작업을 수행해야 합니다.

프로젝트 리스트에서 솔루션을 선택하고 마우스 오른쪽 버튼을 클릭하여 아래 그림과 같이 **현재 프로젝트를 자동으로 로드하기**를 체크합니다.

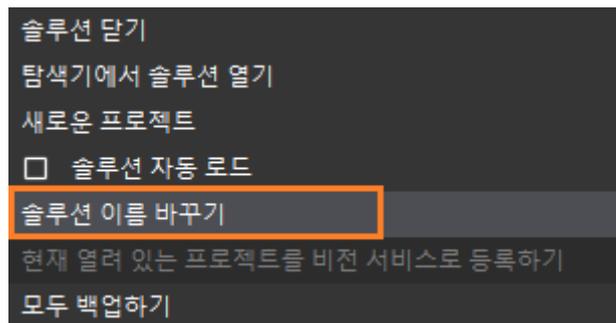


**현재 프로젝트를 자동으로 로드하기**를 체크한 후, 이 솔루션의 프로젝트는 자동으로 "자동 로딩" 모드로 설정되며 다음에 소프트웨어를 열 때 시작 화면 팝업창 없이 소프트웨어의 메인 인터페이스로 바로 들어갑니다.

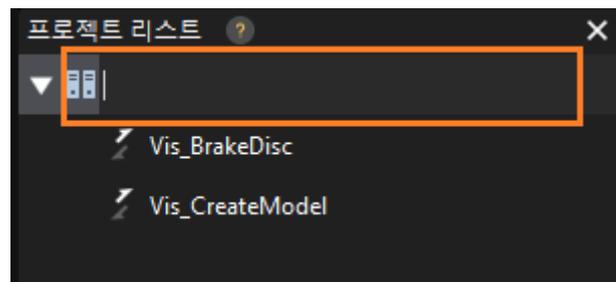
#### 4.2.2.4. 프로젝트 이름 바꾸기

이 부분에서는 프로젝트의 이름을 바꾸는 방법에 대해 설명합니다.

프로젝트의 이름을 바꾸려면 먼저 프로젝트 이름을 마우스 오른쪽 버튼으로 클릭한 다음 **프로젝트 이름 바꾸기**를 클릭하십시오.



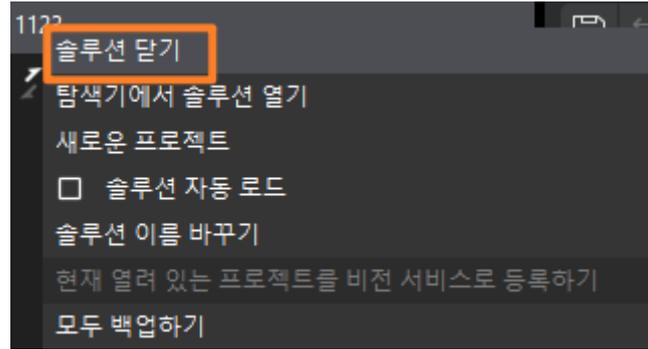
그 다음 프로젝트 이름에 프로젝트 이름을 다시 작성하고 키보드 **Enter** 키를 누릅니다.



#### 4.2.2.5. 프로젝트를 닫기

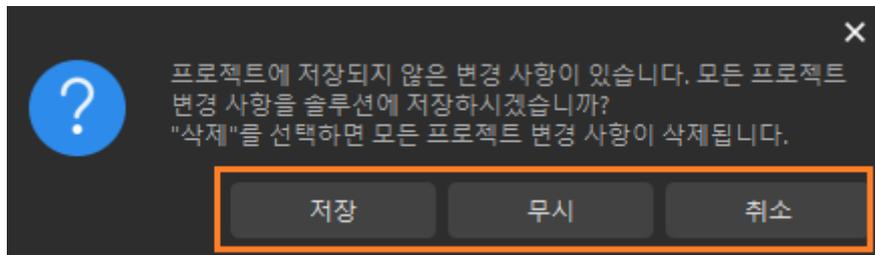
프로그램을 종료해야 하는 경우 다음 두 가지 방법으로 종료할 수 있습니다.

- 프로젝트 리스트에서 프로젝트를 마우스 오른쪽 버튼으로 클릭한 다음 **프로젝트를 닫기**를 클릭하여 아래 그림과 같이 현재 Mech-Vision 프로젝트를 닫습니다.



- 메뉴 바에서 **파일 > 솔루션 닫기**를 차례로 선택합니다.

솔루션 내의 프로젝트에 저장되지 않은 내용이 있는 경우 프롬프트 창이 나타나며 아래 그림과 같이 사용자는 실제 필요에 따라 선택할 수 있습니다.

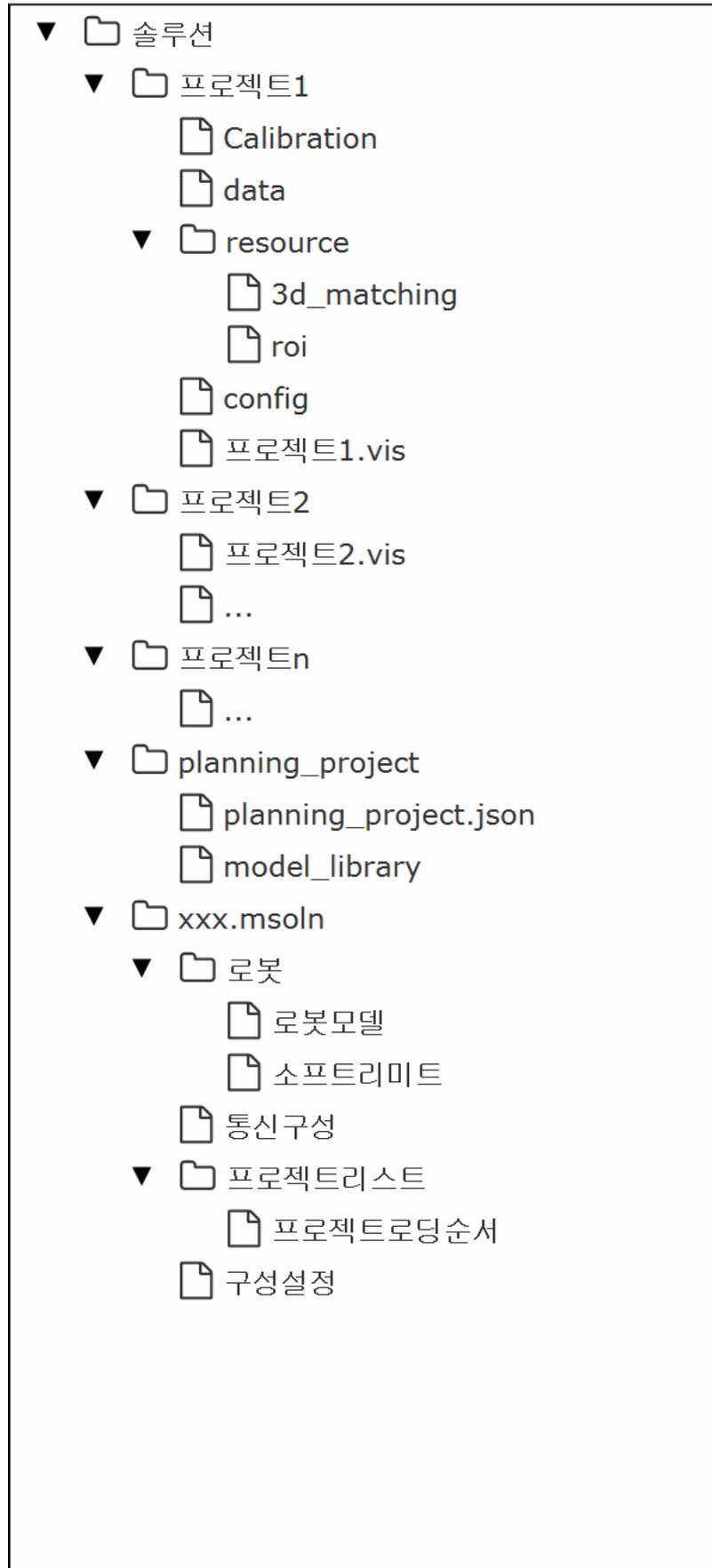


- 🔥 솔루션을 닫으면 이 솔루션 내의 모든 프로젝트도 종료됩니다.

#### 4.2.2.6. 솔루션 구조 이해하기

이 부분에서는 비전 응용을 실현하는 데 필요한 로봇과 통신, 비전 처리 및 경로 계획 등 기능 구성과 데이터의 모음을 포함하는 솔루션 구조를 소개합니다.

솔루션 구조는 아래 그림과 같습니다.



### 4.2.3. 프로젝트 사용 가이드

프로젝트는 Mech-Vision 프로젝트를 말하며 하나 및 이상의 솔루션으로 구성되었습니다. 프로젝트는 단독으로 사용할 수 없으며 솔루션에 속해야 합니다.

이 부분에서는 프로젝트와 관련된 기본 작업을 소개할 것입니다.

[프로젝트 생성 및 저장](#)

[프로젝트 이름 바꾸기](#)

[프로젝트 열기](#)

[프로젝트 실행 및 디버그](#)

[프로젝트를 닫기](#)

[현재 프로젝트를 자동으로 로드하기](#)

[프로젝트 번호 조정](#)

[기존 프로젝트를 솔루션으로 옮기기](#)

[프로젝트 파일 구조 이해하기](#)

#### 4.2.3.1. 프로젝트 생성 및 저장

이 부분에서는 Mech-Vision 프로젝트를 새로 만들고 저장하는 방법과 프로젝트의 명명 규칙을 소개합니다.

##### 새로운 프로젝트

##### 솔루션 라이브러리에서 새로 만들기

[솔루션 라이브러리에서 솔루션 또는 프로젝트 새로 만들기](#)를 참조하십시오.

##### 새로운 프로젝트를 만들기

새로운 프로젝트를 만드는 세 가지 방법은 다음과 같습니다.

- **새로운 솔루션**을 만든 후 이 솔루션에 새로운 프로젝트가 자동으로 생성됩니다.
- 솔루션에서 메뉴 바의 **파일 > 새로운 프로젝트**를 선택하여 비어 있는 새 프로젝트를 만듭니다.
- 솔루션에서 단축키 **Ctrl + N**을 직접 클릭합니다.



솔루션을 만든 후에만 새 프로젝트를 만들 수 있습니다.

##### 프로젝트를 저장하기

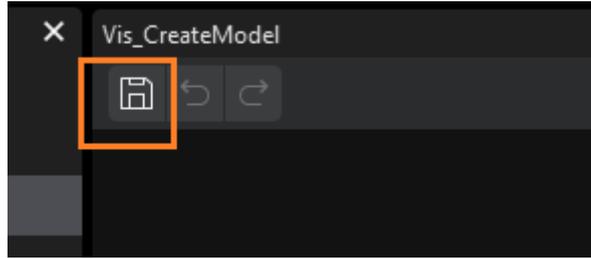
프로젝트를 저장할 때 솔루션에 단일 프로젝트만 저장하거나 솔루션에 있는 모든 프로젝트를 저장하도록 선택할 수 있습니다.

##### 솔루션 내의 단일 프로젝트 저장

솔루션 내의 단일 프로젝트를 저장하는 두 가지 방법은 아래와 같습니다.

- 메뉴 바에서 **파일 > 프로젝트를 저장하기**를 선택하여 새 프로젝트를 만듭니다.

- 프로젝트 툴 바 왼쪽에 있는  버튼을 클릭하여 프로젝트를 저장하거나 직접 단축키 **Ctrl + S** 를 눌러 프로젝트를 저장합니다.



#### 솔루션 내의 모든 프로젝트 저장

솔루션에서 솔루션에 있는 모든 프로젝트를 저장하려면 메뉴 바에서 **파일 > 솔루션 저장**을 차례로 클릭하여 솔루션에 있는 모든 프로젝트를 저장할 수 있습니다.

#### 프로젝트 명명 규칙

Mech-Vision 프로젝트 명칭 구조 및 설명은 아래 표와 같습니다.

이름 구조	설명
Vis	Mech-Vision 프로젝트와 Mech-Viz 프로젝트를 구분하는 데 사용(필수)
작업물 명칭 + 기능 시나리오	예시: 종이 상자 디팔레타이징/브레이크 디스크 텐딩/모터 어셈블리/스폰지 접착제 도포/도어 잠금 장치 피킹
워크스테이션 정보 번호/방향 정보	워크스테이션 정보 번호의 예시: 1# 결함 감지 1# 열처리 1# 방향 정보의 예시: 왼쪽 스테이션/왼쪽 카메라 (해당 정보가 없을 경우 비워둘 수 있습니다)
다른 기능/특수 기능	다른 기능 예시: 1종 작업물 인식/반사 작업물 인식/격판 인식/재료 바꾸니 인식/템플릿 만들기 특수 기능 예시: 포인트 클라우드 스티칭/듀얼 카메라 융합/카메라 보정(해당 정보가 없을 경우 공백으로 남겨둘 수 있음)

관련 명명 예시는 아래 표에 나와 있습니다.

시나리오 설명	예시
단일 워크스테이션 + 단일 카메라	Vis-종이 상자 디팔레타이징
	Vis-브레이크 디스크 텐딩
	Vis-모터 어셈블리
	Vis-스폰지 접착제 도포
	Vis-도어 잠금 장치 피킹
단일 워크스테이션 + 다수 카메라	Vis-금속 부품 텐딩-왼쪽 카메라
다수 워크스테이션 + 다수 카메라	Vis-크랭크 축 텐딩-결함 감지1#왼쪽 카메라
다수 워크스테이션 + 다수 카메라 + 다른 기능	Vis-크랭크 축 텐딩-결함 감지1#왼쪽 카메라-포인트 클라우드 스티칭

시나리오 설명	예시
단일 워크스테이션 프로젝트 조합	Vis-소형 비 평면 부품(등속 조인트의 외륜) 텐딩-5#왼쪽 카메라-1종 작업물 인식
	Vis-소형 비 평면 부품(등속 조인트의 외륜) 텐딩-5#왼쪽 카메라-2종 작업물 인식
	Vis-소형 비 평면 부품(등속 조인트의 외륜) 텐딩-5#왼쪽 카메라-재료 바꾸니 인식
	Vis-소형 비 평면 부품(등속 조인트의 외륜) 텐딩-5#왼쪽 카메라-템플릿 만들기
	Vis-소형 비 평면 부품(등속 조인트의 외륜) 텐딩-5#오른쪽 카메라-1종 작업물 인식
	Vis-소형 비 평면 부품(등속 조인트의 외륜) 텐딩-5#오른쪽 카메라-2종 작업물 인식

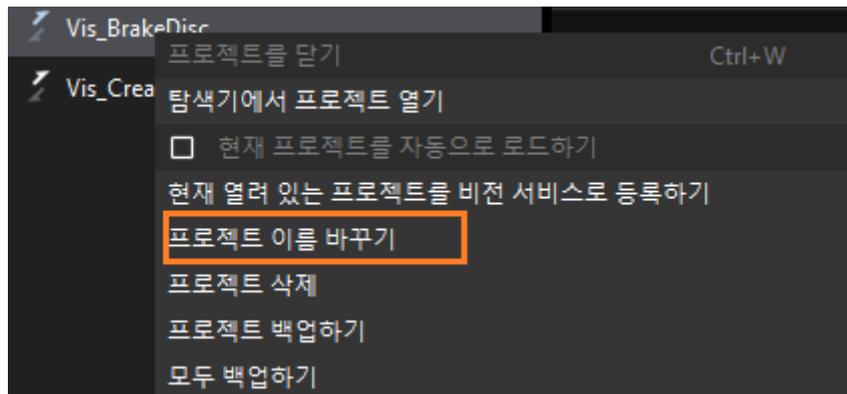
 영어로 이름을 지정할 때 다음 사항에 유의해야 합니다.

- 영어 약어가 있으면 영어 약어를 사용하십시오.
- 가능하면 중국어 병음을 사용하지 마십시오.
- 여러 영어 단어를 함께 사용할 경우 각 단어는 카멜 표기법으로 구분해야 합니다.

#### 4.2.3.2. 프로젝트 이름 바꾸기

프로젝트 이름을 변경해야 하는 경우 다음을 수행해야 합니다.

프로젝트 리스트에서 프로젝트를 선택하고 마우스 오른쪽 버튼을 클릭한 다음 [ **프로젝트 이름 바꾸기** ]를 클릭한 후 새 프로젝트 이름을 입력하고 **Enter** 키를 눌러 프로젝트 이름 변경을 완료합니다.



 위의 작업이 완료되면 프로젝트 폴더와 프로젝트 폴더에 있는 .vis 파일의 이름이 변경됩니다.

#### 4.2.3.3. 프로젝트 열기

이 부분에서는 Mech-Vision 솔루션을 여는 방법에 대해 설명합니다.

##### 기존 프로젝트 열기

다음 세 가지 방법을 사용하여 기존 프로젝트를 열 수 있습니다.

- 시작 화면에서 [ **프로젝트 열기** ]를 클릭한 다음 팝업한 파일 선택 창에서 열리는 프로젝트 폴더를 선택하면 되고 구체적인 .vis 파일을 선택할 필요는 없습니다.
- 메인 인터페이스의 메뉴 바에서 **파일 > 프로젝트 폴더 열기**(단축키 **Ctrl + O**)를 클릭한 다음 열리는 프로젝트 폴더를 선택하면 되고 구체적인 .vis 파일을 선택할 필요는 없습니다.
- 프로젝트 폴더에서 .vis 파일을 두 번 클릭하여 Mech-Vision 소프트웨어를 열고 프로젝트를 자동으로 로드합니다.

#### 최근 사용했던 프로젝트를 열기

다음 세 가지 방법을 사용하여 최근에 열었던 프로젝트를 열 수 있습니다.

- 시작 화면에서 오른쪽에 있는 솔루션/프로젝트 전환 버튼을 **프로젝트**로 전환한 다음 열리는 프로젝트를 두 번 클릭합니다.



- 소프트웨어 메인 인터페이스의 메뉴 바를 클릭하고 **파일 > 최근에 사용했던 프로젝트를 열기**를 클릭하여 최근에 연 프로젝트를 찾아 바로 엽니다.



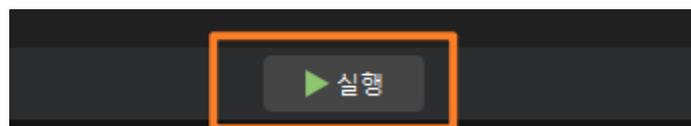
- Mech-Vision 1.4.0 및 이상 버전을 사용하여 구 버전의 프로젝트를 열면 프로젝트 저장 후 원본 .json 기본 파일이 자동으로 .vis로 대체되고 원본 파일은 .bak 파일로 백업됩니다.
- 메뉴 바에서 **파일 > 프로젝트를 JSON로 저장하기**를 클릭하여 프로젝트를 .json 형식으로 저장할 수 있습니다.

#### 4.2.3.4. 프로젝트 실행 및 디버그



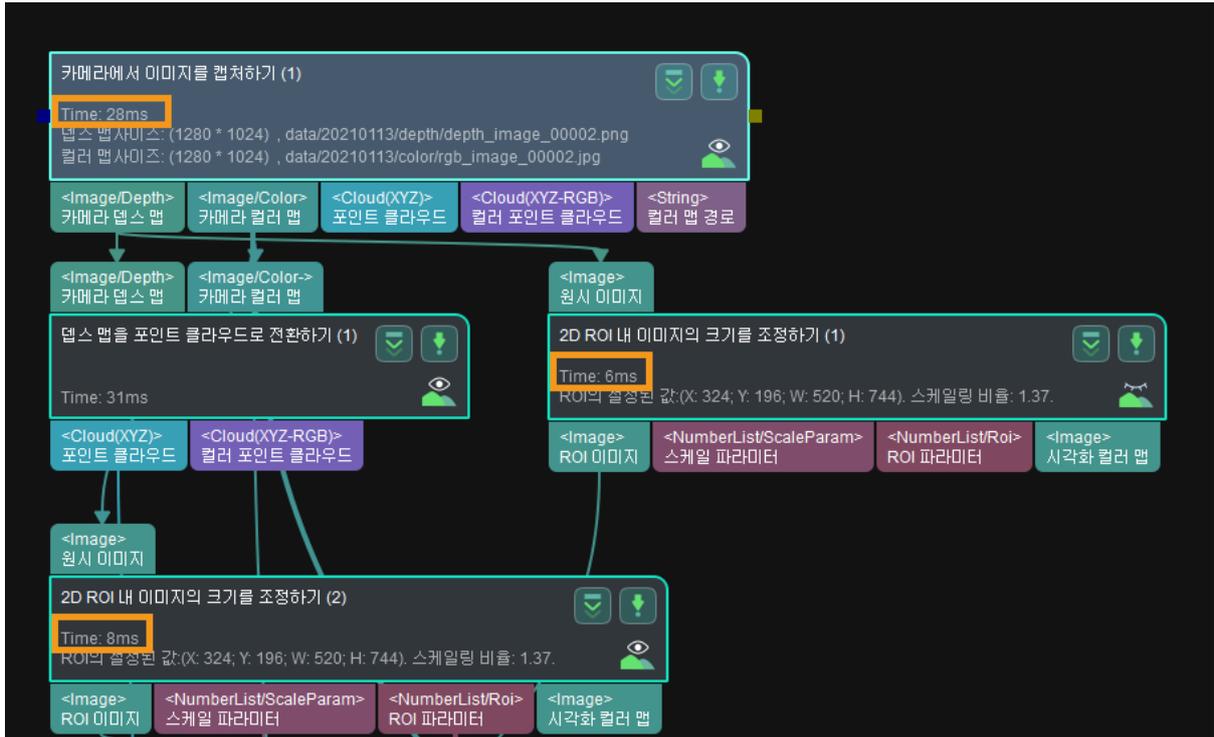
- Mech-Vision v1.4.0은 프로젝트 실행 시 버전 호환성 검사를 수행하므로 사용자는 v1.4.0 및 이상 버전에서 Mech-Viz와 Mech-Center를 사용하는 것을 권장합니다.
- 사용자가 사용하는 Mech-Viz 또는 Mech-Center 소프트웨어 버전이 v1.4.0 보다 낮은 경우 프로젝트 실행 시 위험 경고 창이 팝업됩니다.

프로젝트 톨 바에서 실행 버튼을 클릭하거나 단축키 **Ctrl + R**을 누르면 현재의 Mech-Vision 프로젝트를 실행할 수 있습니다. 다음 그림과 같습니다.



프로젝트 실행 결과가 기대에 미치지 못하면 프로젝트를 디버깅해야 합니다. 먼저 비정상 스텝을 찾아야 하며, 각 스텝을 하나씩 확인할 수 있습니다. 예상 결과와 일치하지 않는 스텝의 경우 파라미터 조정을 통해 최적화할 수 있습니다. 그중 완료된 각 스텝은 디스플레이 상자의 왼쪽 하단 모서리에 스텝을 실행하는 데

필요한 시간을 표시합니다. 다음 그림과 같습니다.

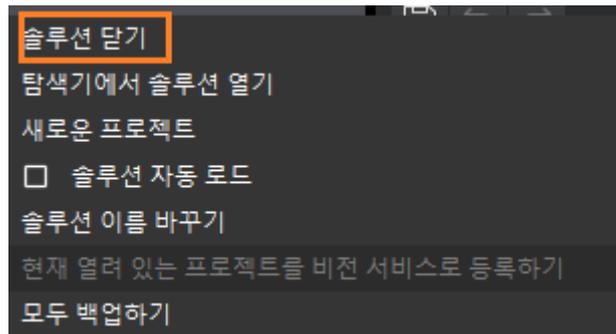


프로젝트 디버깅의 구체적인 내용은 [스텝을 실행하고 시각화된 출력 결과를 보기](#)를 참조하십시오.

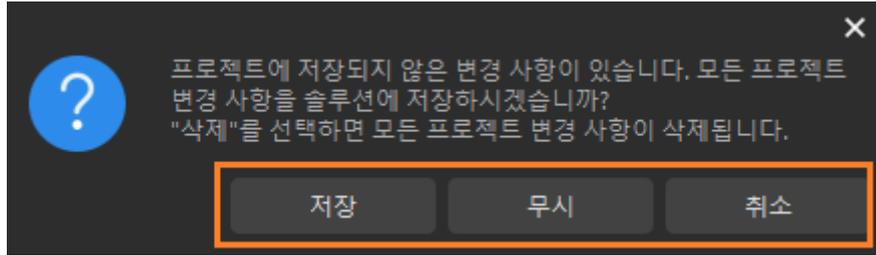
#### 4.2.3.5. 프로젝트를 닫기

솔루션에서 단일 프로젝트를 닫을 수 없지만 솔루션을 닫으면 솔루션의 모든 프로젝트를 닫을 수 있습니다. 구체적인 작업은 다음과 같습니다.

프로젝트 창에서 해당 솔루션을 마우스 오른쪽 버튼으로 클릭한 다음 **[솔루션 닫기]**를 클릭하여 아래 그림과 같이 솔루션과 관련된 모든 프로젝트를 닫습니다.



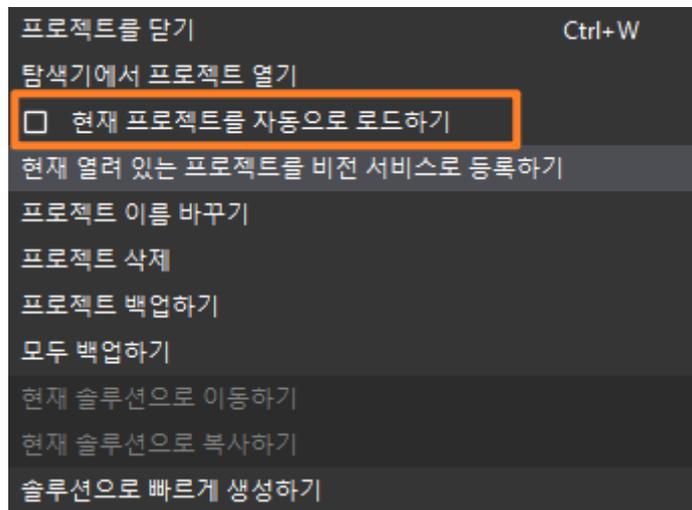
솔루션 내의 프로젝트에 저장되지 않은 내용이 있는 경우 프롬프트 창이 나타나며 아래 그림과 같이 사용자는 실제 필요에 따라 선택할 수 있습니다.



#### 4.2.3.6. 현재 프로젝트를 자동으로 로드하기

나중에 Mech-Vision 소프트웨어가 시작될 때 임의의 프로젝트도 함께 열리면 다음과 같이 설정해야 합니다.

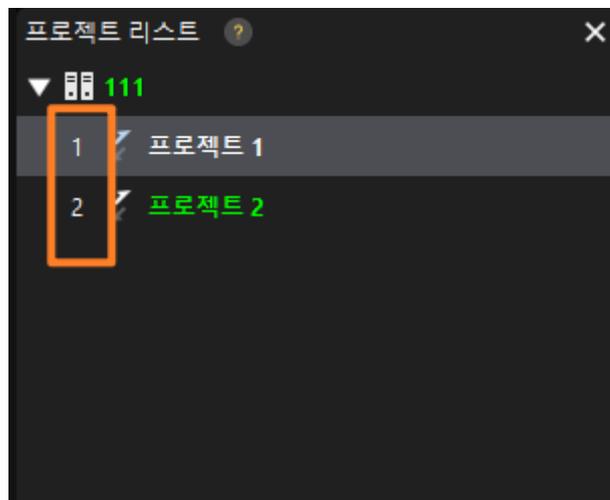
- 프로젝트가 할당되지 않은 경우 프로젝트 리스트에서 해당 프로젝트를 선택하고 마우스 오른쪽 버튼을 클릭한 다음 아래 그림과 같이 **현재 프로젝트를 자동으로 로드하기**를 체크합니다.



- 프로젝트가 솔루션에 있는 경우 **현재 프로젝트를 자동으로 로드하기**를 체크해야 합니다. 이 솔루션의 모든 프로젝트는 자동으로 "자동 로드" 모드로 설정됩니다.

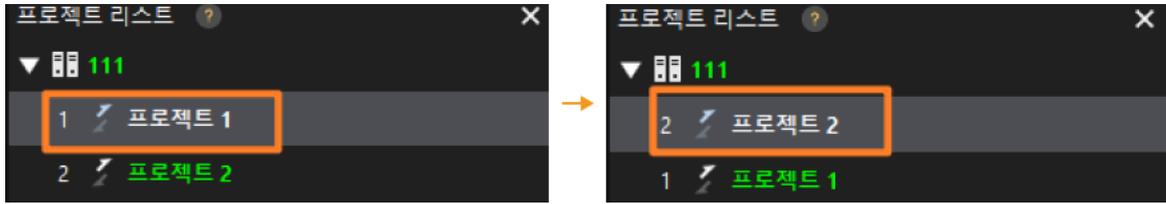
#### 4.2.3.7. 프로젝트 번호 조정

솔루션이 "자동 로드" 모드에 있으면 솔루션의 프로젝트도 "자동 로드"로 설정되고 각 프로젝트에는 아래 그림과 같이 프로젝트 번호가 할당됩니다.



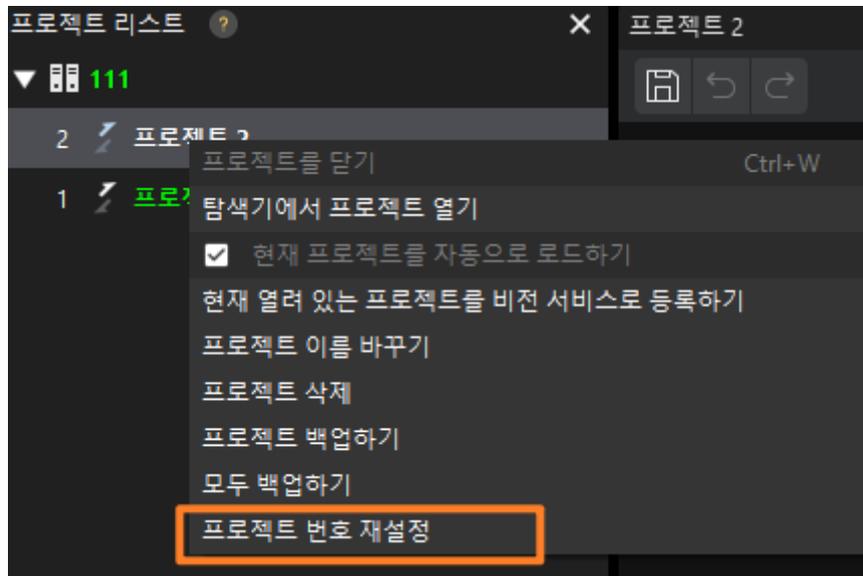
프로젝트 번호를 조정해야 하는 경우 다음 작업을 수행할 수 있습니다.

1. 드래그 앤 드롭하여 프로젝트 순서를 조정합니다. 프로젝트를 선택하고 위아래로 드래그하여 프로젝트 순서를 조정합니다. 아래 그림과 같이 왼쪽이 조정 전이고 오른쪽이 조정 후입니다.



이때 프로젝트 번호는 변경되지 않았지만 프로젝트 리스트에서 프로젝트의 위치는 조정되었습니다. "프로젝트 번호 재설정"도 필요합니다.

2. 프로젝트 번호를 재설정합니다. 솔루션 또는 프로젝트를 마우스 오른쪽 버튼으로 클릭한 다음 [ **프로젝트 번호 재설정** ]을 클릭하여 프로젝트 번호를 조정합니다.



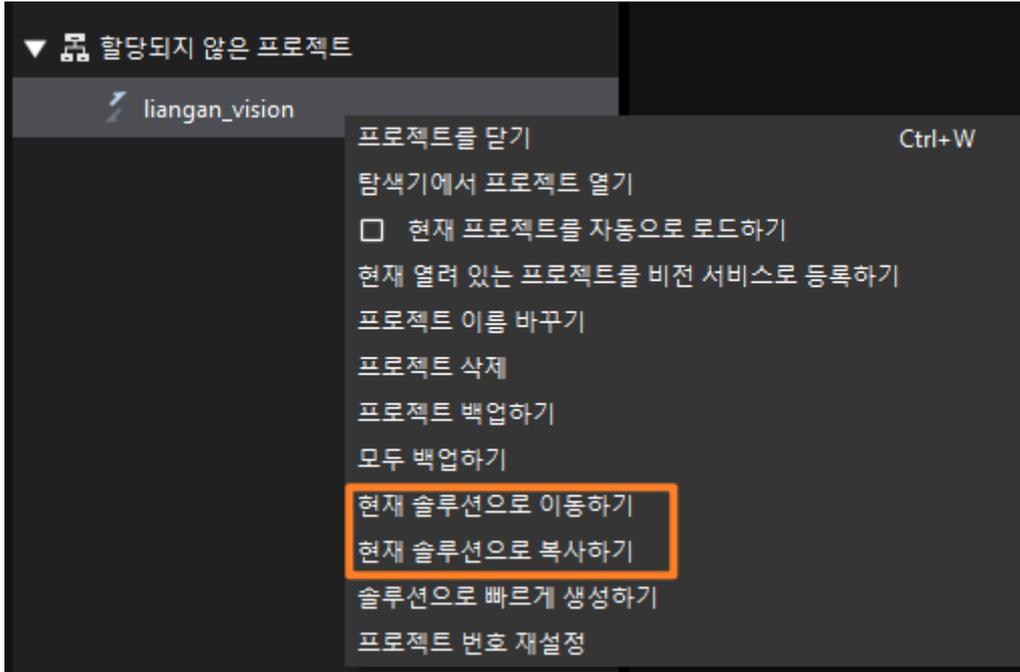
재설정 결과는 아래 그림과 같으며 왼쪽은 재설정 전, 오른쪽은 재설정 후입니다.



#### 4.2.3.8. 기존 프로젝트를 솔루션으로 옮기기

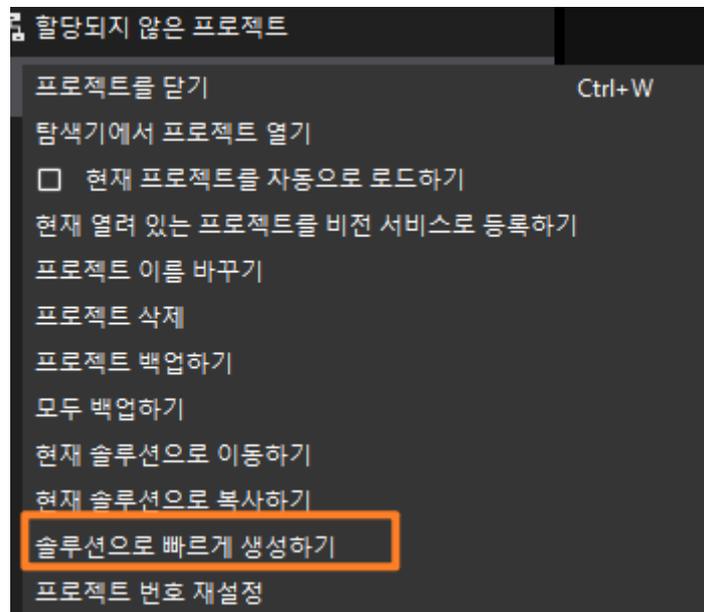
기존 프로젝트를 솔루션으로 옮겨야 하는 경우 다음 방법을 통해 수행할 수 있습니다.

- 새로운 솔루션을 생성하여 저장한 후 프로젝트 리스트에서 할당되지 않은 프로젝트를 선택하고 마우스 오른쪽 버튼을 클릭하여 [ **현재 솔루션으로 이동하기** ] 또는 [ **현재 솔루션으로 복사하기** ]를 선택합니다. 아래 그림과 같습니다.



옮겨간 프로젝트에서 해당 스텝의 파라미터가 절대 경로 또는 상대 경로를 사용하는 경우 옮긴 후의 프로젝트를 확인하여 옮김으로 인한 잘못된 경로를 수정하시기 바랍니다.

- 이 프로젝트를 새로운 솔루션으로 변환하려면 프로젝트 리스트에서 할당되지 않은 프로젝트를 선택하고 마우스 오른쪽 버튼을 클릭하여 [ **솔루션으로 빠르게 생성하기** ]를 선택한 다음 팝업창에서 [ **예** ]를 클릭하십시오. 프로젝트를 솔루션으로 변환할 수 있으며 프로젝트의 상위 폴더가 솔루션 폴더로 사용됩니다.



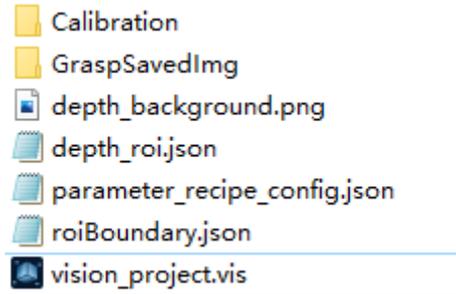
프로젝트를 새 솔루션으로 변환할 때 프로젝트의 상위 디렉터리는 디스크의 루트 디렉터리가 될 수 없습니다.

#### 4.2.3.9. 프로젝트 파일 구조 이해하기

비전 프로젝트의 파일 구조는 아래 그림과 같으며 주로 다음 파일로 구성됩니다.

- 알고리즘 프로세스 파일(그림의 vision\_project.vis) : 알고리즘 처리 과정을 저장하기 위해 사용합니다.
- 카메라 파라미터 파일(사진 내 Calibration): 카메라 내부/외부 파라미터 및 캘리브레이션 데이터를 저장하는 데 사용됩니다.
- 프로젝트 구성 파일(그림의 depth\_background.png, depth\_roi.json, roiBoundary.json 등): 작업 환경의 전체 조건을 설명하고 프로젝트에 제한을 추가하는 데 사용됩니다.

비전 프로젝트 파일은 해당 작업을 수행한 후 해당 프로젝트 경로 아래에 자동으로 생성되며, 수동으로 생성할 필요가 없습니다.



#### 4.2.4. 로봇 및 인터페이스 구성 사용 가이드

이 부분에서는 "로봇 및 인터페이스 구성"과 관련된 작업을 소개합니다.

구성하기 전에 [통신 개요](#)를 참조하여 통신 방법 및 통신 프로토콜을 확인하십시오.

- 통신 방법이 "표준 인터페이스 통신" 및 "Adapter 통신"인 경우 이 부분 내용을 참조하십시오.
  - [로봇을 선택하기](#)
  - 통신 구성:
    - [TCP/IP 통신 구성](#)
    - [Siemens PLC Snap 7 통신 구성](#)
    - [PROFINET 통신 구성](#)
    - [EtherNet/IP 통신 구성](#)
    - [Modbus TCP 통신 구성](#)
    - [Adapter 통신 구성](#)
    - [Mitsubishi MC 통신 구성](#)
    - [UDP 통신 구성](#)
  - [인터페이스 서비스 시작](#)
- 통신 방법이 마스터 컨트롤 통신인 경우 [마스터 컨트롤](#) 부분을 참조하십시오.

##### 4.2.4.1. Adapter 통신 구성

이 부분에서는 Adapter 통신을 구성하는 방법에 대해 설명합니다. 실행하기 전에 [로봇을 선택하기](#)를 확인하고 통신 구성 인터페이스로 들어가십시오.



1. 통신 옵션에서 인터페이스 유형을 **Adapter**로 선택합니다.
2. 를 클릭하여 Adapter 프로젝트 폴더를 선택합니다.



사용자는 사전에 Adapter 프로그램을 작성해야 합니다. 자세한 내용은 [Adapter 프로그래밍 가이드](#)를 참조하십시오.

3. IP 주소에서 호스트 IP 및 포트 번호를 입력합니다.



- 호스트 IP는 로봇 IP와 동일한 네트워크 세그먼트에 있어야 합니다.
- 호스트 포트 번호는 50000 및 이상으로 설정하는 것이 좋습니다.

4. (선택 사항) **솔루션을 열 때 인터페이스 서비스 자동 활성화**를 선택합니다. 다음에 솔루션을 열 때 **인터페이스 서비스 시작** 할 필요가 없습니다.
5. [응용]을 클릭합니다.
6. **인터페이스 서비스 시작** 합니다.

선택한 로봇에 따라 [표준 인터페이스 개발자 사용 가이드](#) 내용을 참조하여 로봇 프로그램을 작성한 후 프로그램 복제 작업을 수행합니다.

#### 4.2.4.2. EtherNet/IP 통신 구성

이 부분에서는 EtherNet/IP 통신을 구성하는 방법에 대해 설명합니다. 실행하기 전에 [로봇을 선택하기](#)를 확인하고 통신 구성 인터페이스로 들어가십시오.

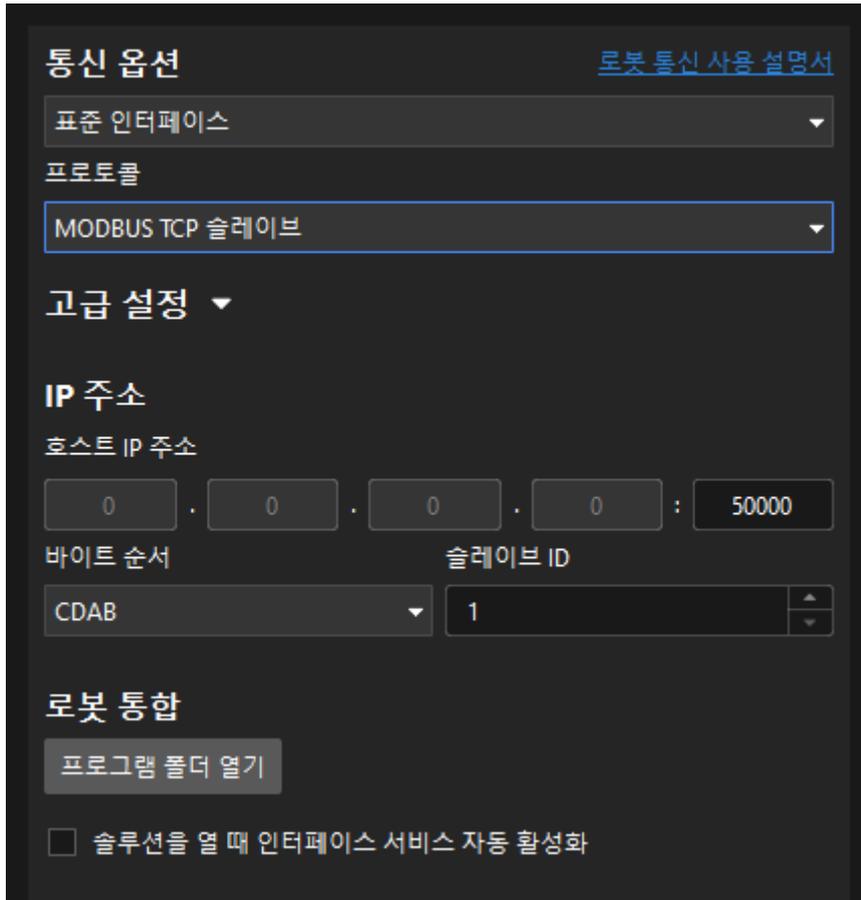


1. 통신 옵션에서 인터페이스 유형을 **표준 인터페이스**로 선택하고 프로토콜은 **ETHERNET IP**로 선택합니다.
2. (선택 사항) **솔루션을 열 때 인터페이스 서비스 자동 활성화**를 선택합니다. 다음에 솔루션을 열 때 **인터페이스 서비스 시작** 할 필요가 없습니다.
3. [응용]을 클릭합니다.
4. **인터페이스 서비스 시작** 합니다.

선택한 통신 대상에 따라 **표준 인터페이스 구성** 작업을 수행합니다.

#### 4.2.4.3. Modbus TCP 통신 구성

이 부분에서는 Modbus TCP 통신을 구성하는 방법에 대해 설명합니다. 실행하기 전에 **로봇을 선택하기**를 확인하고 통신 구성 인터페이스로 들어가십시오.



1. 통신 옵션에서 인터페이스 유형을 **표준 인터페이스**로 선택하고 프로토콜은 **MODBUS TCP SLAVE**로 선택합니다.
2. IP 주소 에서 다음과 같이 옵션을 설정합니다.
  - a. 호스트(슬레이브) 포트를 입력합니다. 포트 번호는 50000 및 이상으로 설정하는 것이 좋습니다.
  - b. 바이트 순서를 선택합니다. 마스터 스테이션의 부동 소수점 수의 바이트 순서에 따라 표준 빅 엔디안 데이터의 경우 DCBA를 선택하고 표준 리틀 엔디안 데이터의 경우 ABCD를 선택합니다.
  - c. 슬레이브 ID를 선택합니다.
3. (선택 사항) **솔루션을 열 때 인터페이스 서비스 자동 활성화**를 선택합니다. 다음에 솔루션을 열 때 **인터페이스 서비스 시작** 할 필요가 없습니다.
4. [응용]을 클릭합니다.
5. **인터페이스 서비스 시작** 합니다.

선택한 통신 대상에 따라 **표준 인터페이스 구성** 작업을 수행합니다.

#### 4.2.4.4. PROFINET 통신 구성

이 부분에서는 PROFINET 통신을 구성하는 방법에 대해 설명합니다. 실행하기 전에 **로봇을 선택하기**를 확인하고 통신 구성 인터페이스로 들어가십시오.



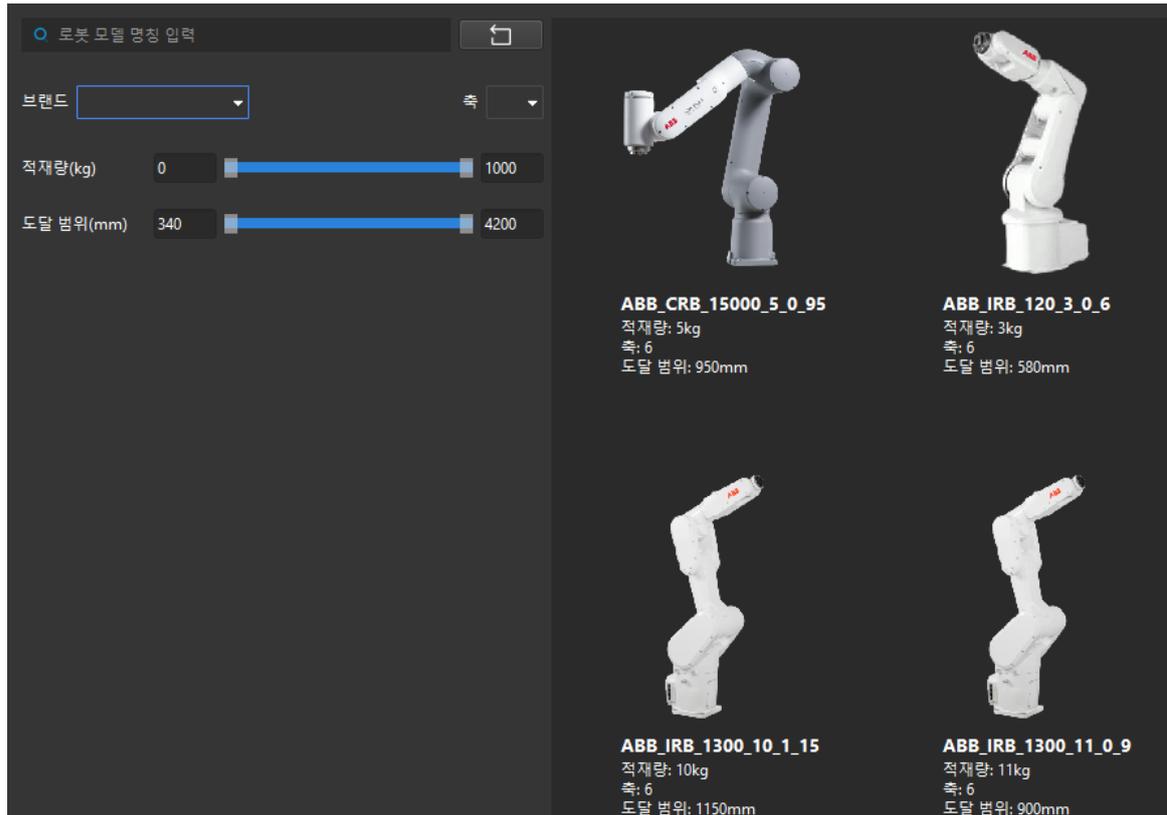
1. 통신 옵션에서 인터페이스 유형을 **표준 인터페이스**로 선택하고 프로토콜은 **PROFINET-IRT**로 선택합니다.
2. (선택 사항) **솔루션을 열 때 인터페이스 서비스 자동 활성화**를 선택합니다. 다음에 솔루션을 열 때 **인터페이스 서비스 시작** 할 필요가 없습니다.
3. [응용]을 클릭합니다.
4. **인터페이스 서비스 시작** 합니다.

선택한 통신 대상에 따라 **표준 인터페이스 구성** 작업을 수행합니다.

#### 4.2.4.5. 로봇의 모델을 선택하기

이 부분에서는 로봇 모델을 선택하는 방법에 대해 설명합니다.

1. **로봇 선택** 작업 단계에서 아래 그림과 같은 인터페이스로 들어갑니다.



3. 브랜드 뒤의 드롭다운 상자에서 로봇 브랜드를 선택하면 해당 브랜드의 모든 로봇이 오른쪽 창에 나타납니다.



- **축, 적재량, 도달 범위**를 선택하여 로봇의 범위를 더 좁힙니다.
- **검색 상자**에 로봇 브랜드 및 모델(영어로 대소문자 구분 안 함)을 입력하여 로봇을 필터링할 수 있습니다. 입력란의 내용을 지우려면  버튼을 클릭하십시오.

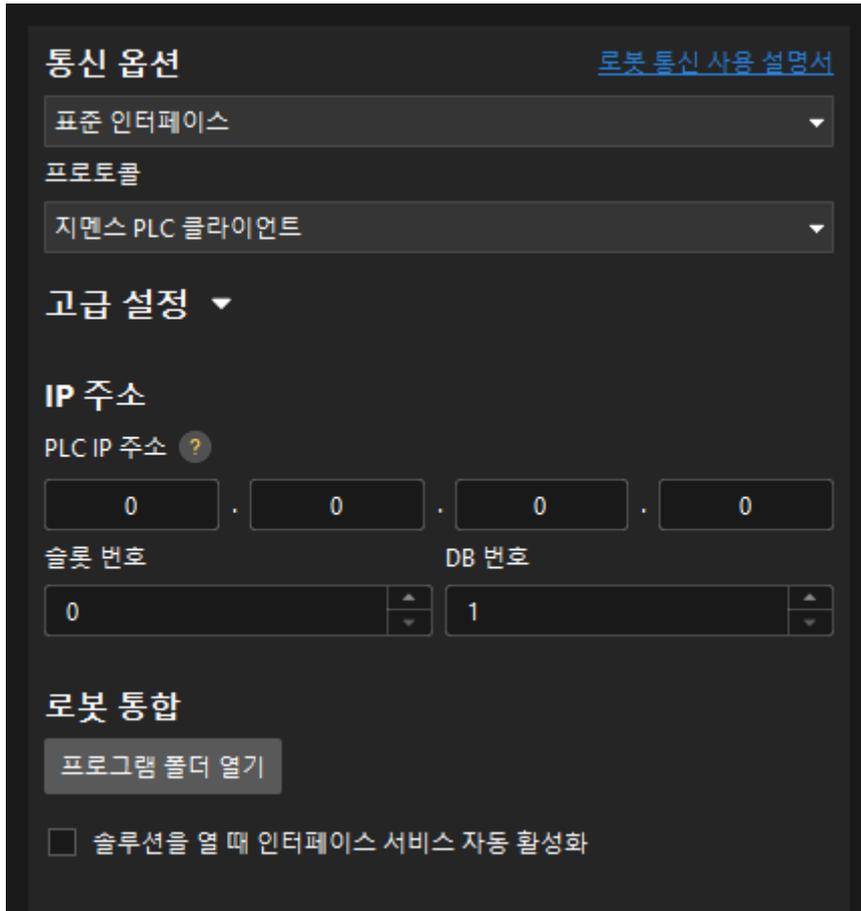


로봇 라이브러리에 필요한 로봇이 없으면 이 창을 닫고 《[로봇을 도입하기](#)》를 참조하여 작업한 후 여기서 작업을 수행합니다.

3. 오른쪽에서 선택할 로봇 위에 마우스를 올리면 회색 테두리와 [ **선택** ] 버튼이 표시되며 버튼을 클릭하고 창을 닫습니다.

#### 4.2.4.6. Siemens PLC Snap 7 통신 구성

이 부분에서는 Siemens PLC Snap 7 통신을 구성하는 방법에 대해 설명합니다. 실행하기 전에 [로봇을 선택하기](#)를 확인하고 통신 구성 인터페이스로 들어가십시오.



1. 통신 옵션에서 인터페이스 유형으로 **표준 인터페이스**를 선택하고 프로토콜을 **지멘스 PLC 클라이언트**로 선택합니다.
2. IP 주소 에서 다음과 같이 옵션을 설정합니다.
  - a. 호스트 IP 및 포트 번호를 입력합니다. PLC의 IP 주소는 IPC의 IP 주소와 동일한 네트워크 세그먼트에 있어야 합니다.
  - b. 슬롯 번호를 선택합니다. 아래 표를 참조하여 선택하십시오.

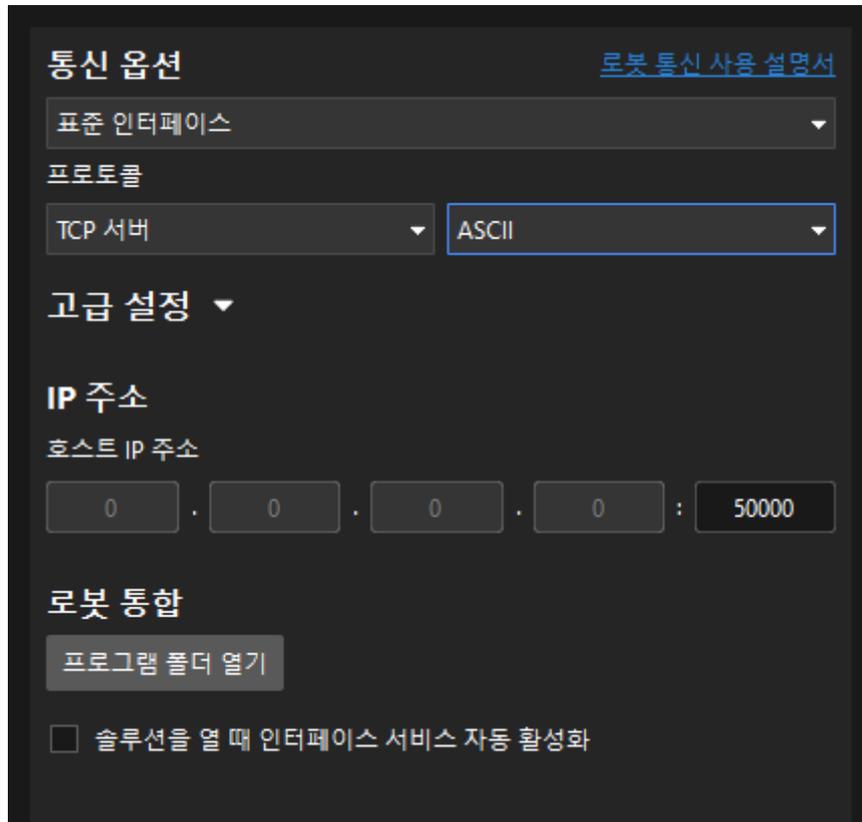
지멘스 S7 시리즈 모델	슬롯 번호
S7-300	2
S7-1200	0 또는 1
S7-1500	0 또는 1
기타	0

- a. DB 블록 번호를 선택합니다.
  1. (선택 사항) **솔루션을 열 때 인터페이스 서비스 자동 활성화**를 선택합니다. 다음에 솔루션을 열 때 **인터페이스 서비스 시작** 할 필요가 없습니다.
  2. [응용]을 클릭합니다.
  3. **인터페이스 서비스 시작** 합니다.

선택한 통신 대상에 따라 **표준 인터페이스 구성** 작업을 수행합니다.

#### 4.2.4.7. TCP/IP 통신 구성

이 부분에서는 TCP/IP 통신을 구성하는 방법에 대해 설명합니다. 실행하기 전에 먼저 [로봇을 선택하기](#)를 확인하고 통신 구성 인터페이스로 들어가십시오.



1. 통신 옵션에서 인터페이스 유형을 **표준 인터페이스**로 선택하고, 프로토콜을 **TCP 서버**로 선택하고 아래 표에 따라 통신 프로토콜 형식을 선택합니다.

로봇 유형	통신 프로토콜 형식
ABB	HEX(리틀 엔디안)
FANUC	HEX(빅 엔디안)
KUKA	HEX(리틀 엔디안)
Yaskawa	ASCII
Kawasaki	ASCII
UR	ASCII
TM	ASCII
ELITE	ASCII
JAKA	ASCII
ROKAE	ASCII
NACHI	ASCII

로봇 유형	통신 프로토콜 형식
기타	사용자는 로봇 측 통신 프로그램을 작성해야 하므로 로봇의 HEX 및 ASCII 지원에 따라 프로토콜 형식을 결정할 수 있습니다.

2. IP 주소에서 호스트 포트 번호를 입력합니다.

 호스트 포트 번호는 50000 및 이상으로 설정하는 것이 좋습니다.

3. (선택 사항) **솔루션을 열 때 인터페이스 서비스 자동 활성화**를 선택합니다. 다음에 솔루션을 열 때 **인터페이스 서비스 시작** 할 필요가 없습니다.

4. [응용]을 클릭합니다.

5. **인터페이스 서비스 시작** 합니다.

그런 다음 선택한 로봇에 따라 다음 해당 작업을 수행합니다.

- 선택한 로봇이 다음 브랜드의 로봇인 경우 아래 표의 해당 문서를 참조하여 해당 작업을 수행하십시오.

로봇 유형	관련된 문서
ABB	ABB 표준 인터페이스 구성
FANUC	FANUC 표준 인터페이스 구성
FANUC CRX	CRX 플러그인 설치 및 구성
Kawasaki	Kawasaki 표준 인터페이스 구성
KUKA	KUKA 표준 인터페이스 구성
YASKAWA	YASKAWA 표준 인터페이스 구성
UR	URCap 플러그인 설치 및 구성
TM	TM 표준 인터페이스 구성
LITE	ELITE 표준 인터페이스 구성
JAKA	JAKA 표준 인터페이스 구성

- 선택한 로봇이 상기 이외의 다른 브랜드 로봇인 경우 **표준 인터페이스 개발자 참조 설명서**를 참조하여 로봇 측 통신 프로그램을 작성하시기 바랍니다. 캘리브레이션 수행 시 해당 로봇의 **마스터 컨트롤 프로그램 복제**를 참고하여 **자동 캘리브레이션**을 완료합니다.
- 선택한 로봇이 다른 로봇인 경우 **표준 인터페이스 개발자 참조 설명서**를 참조하여 로봇 측 통신 프로그램을 작성합니다. 캘리브레이션을 수행할 때 관련 작업은 **수동 캘리브레이션**을 참조하십시오.

#### 4.2.4.8. Mitsubishi MC 통신 구성

이 부분에서는 Mitsubishi MC 통신을 구성하는 방법에 대해 설명합니다. 실행하기 전에 먼저 **로봇을 선택하기**를 확인하고 통신 구성 인터페이스로 들어가십시오.

## 통신 옵션

[로봇 통신 사용 설명서](#)

표준 인터페이스 ▼

프로토콜

미쓰비시 MC 클라이언트 ▼

### 고급 설정 ▼

### IP 주소

PLC IP 주소

0

.

0

.

0

.

0

:

50000

통신 프레임 ?                      레지스터(D) 기본 주소

3E ▼

4000

### 로봇 통합

프로그램 폴더 열기

솔루션을 열 때 인터페이스 서비스 자동 활성화

뒤로

응용

1. 통신 옵션에서 인터페이스 유형으로 **표준 인터페이스**를 선택하고 프로토콜을 **Mitsubishi MC 클라이언트**로 선택합니다.
  2. IP 주소 에서 다음과 같이 옵션을 설정합니다.
    - a. PLC IP 주소에서 PLC의 IP 주소를 입력합니다. 포트 번호는 50000 및 이상으로 설정하는 것이 좋습니다.
    - b. 통신 프레임에서 일반적으로 3E를 선택합니다.
    - c. 레지스터(D) 기본 주소를 설정합니다.
- 🔥
- 통신에 사용되는 구조 유형은 총 728개의 D 레지스터를 차지합니다.
3. (선택 사항) **솔루션을 열 때 인터페이스 서비스 자동 활성화**를 선택합니다. 다음에 솔루션을 열 때 **인터페이스 서비스 시작** 할 필요가 없습니다.

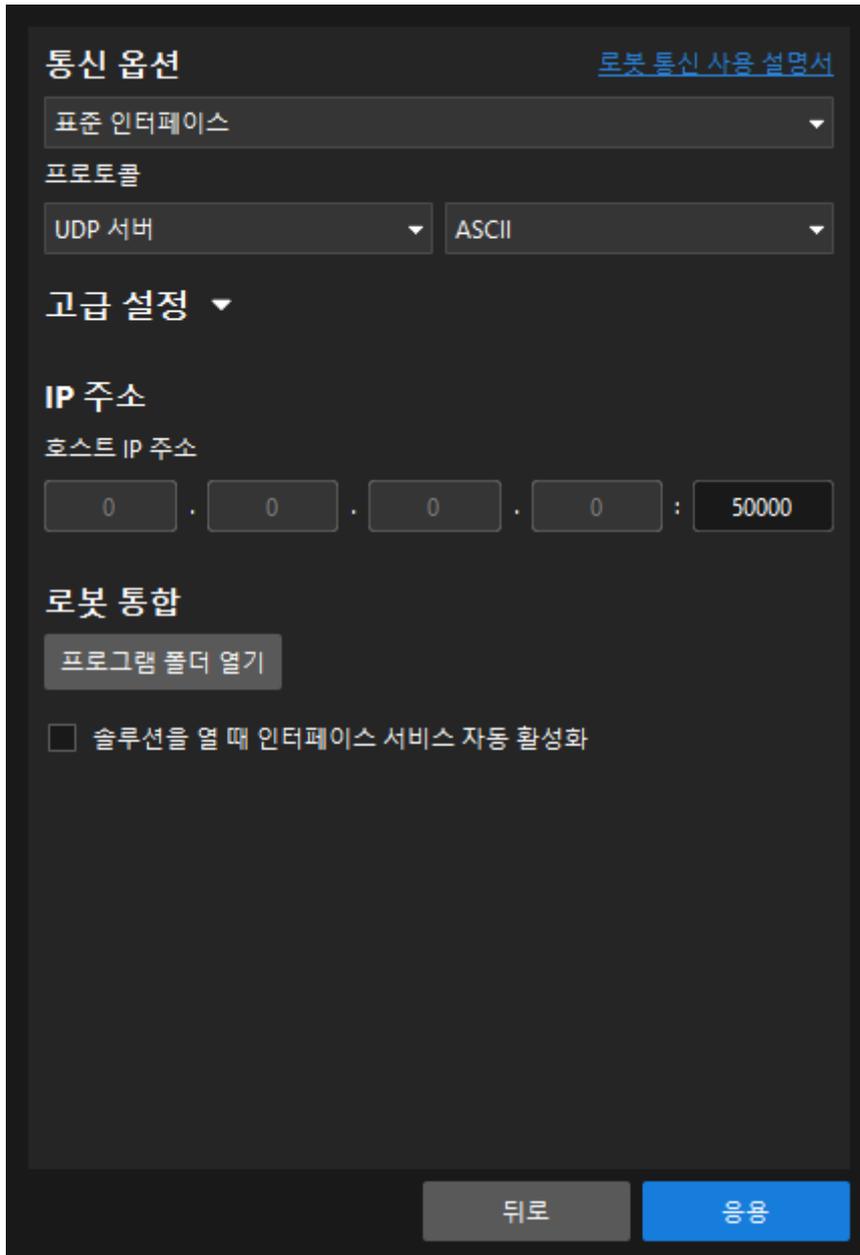
4. [응용]을 클릭합니다.

5. 인터페이스 서비스 시작 합니다.

선택한 통신 대상에 따라 표준 인터페이스 구성 작업을 수행합니다.

#### 4.2.4.9. UDP 통신 구성

이 부분에서는 UDP 통신을 구성하는 방법에 대해 설명합니다. 실행하기 전에 먼저 로봇을 선택하기를 확인하고 통신 구성 인터페이스로 들어가십시오.



1. 통신 옵션에서 인터페이스 유형을 표준 인터페이스로 선택하고, 프로토콜을 UDP 서버로 선택하고 HEX 및 ASCII에 대한 로봇의 지원에 따라 통신 프로토콜 형식을 선택할 수 있습니다.

2. IP 주소에서 호스트 포트 번호를 입력합니다.



호스트 포트 번호는 50000 및 이상으로 설정하는 것이 좋습니다.

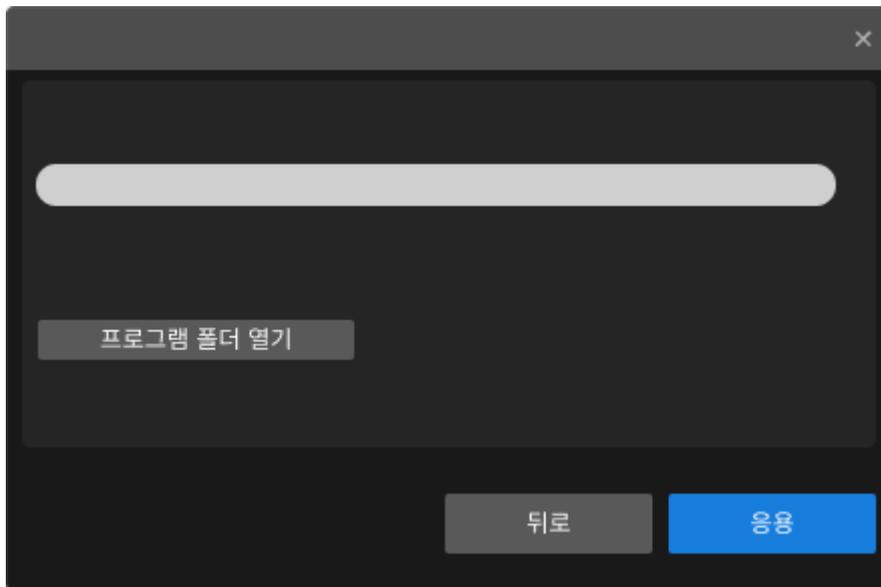
3. (선택 사항) 솔루션을 열 때 인터페이스 서비스 자동 활성화를 선택합니다. 다음에 솔루션을 열 때 인터페이스 서비스 시작 할 필요가 없습니다.
4. [응용]을 클릭합니다.
5. 인터페이스 서비스 시작 합니다.

선택한 통신 대상에 따라 표준 인터페이스 구성 작업을 수행합니다.

#### 4.2.4.10. 복제 파일 가져오기

이 부분에서는 복제 파일을 가져오는 방법에 대해 설명합니다.

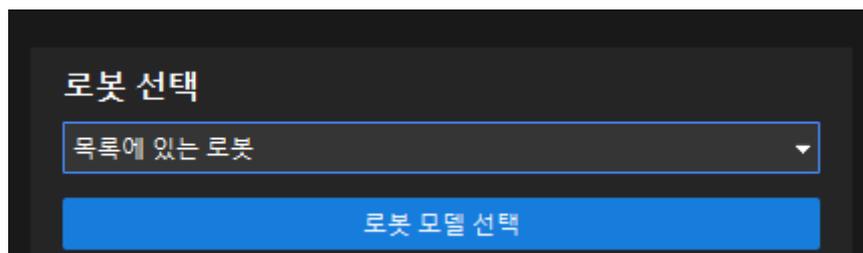
로봇을 선택하기 한 후 통신 구성 인터페이스로 들어가 [프로그램 폴더 열기]를 클릭하면 로봇 통신 프로그램과 샘플 프로그램이 저장된 폴더를 열수 있습니다.



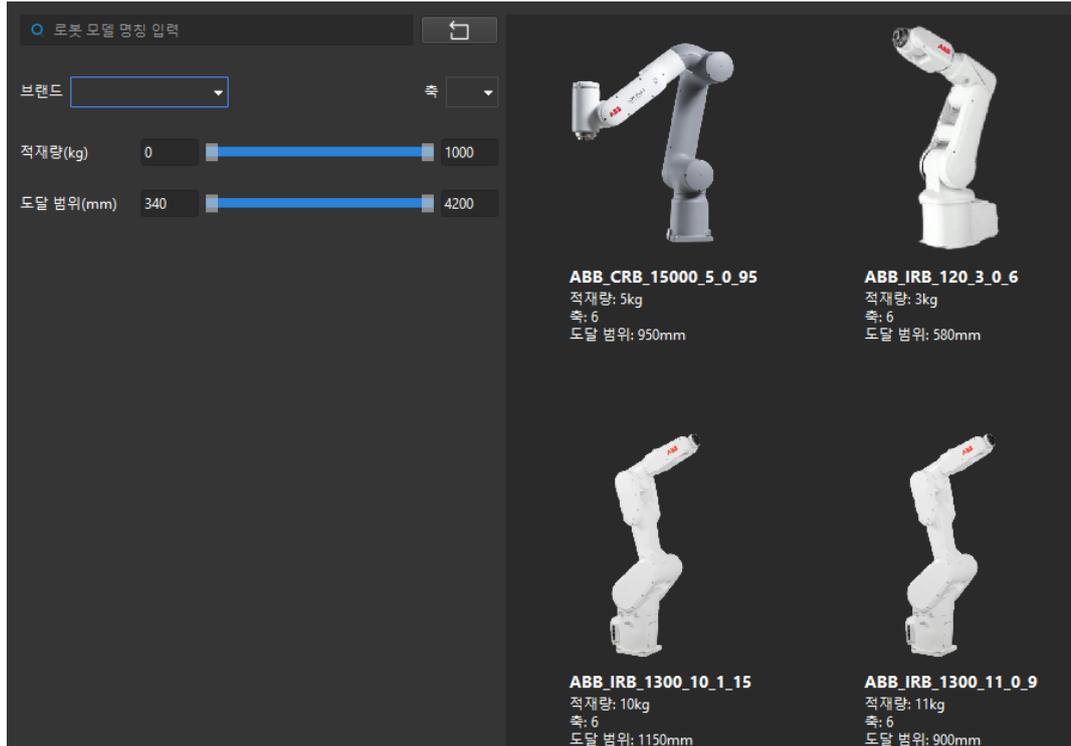
#### 4.2.4.11. 로봇을 선택하기

이 부분에서는 로봇을 선택하는 방법에 대해 설명합니다.

1. Mech-Vision 툴 바에서 [로봇 및 인터페이스 구성]을 클릭합니다.
2. 로봇을 선택하기.
  - 목록에 있는 로봇을 선택하기
    - 로봇 선택 드롭다운 상자를 클릭하고 목록에 있는 로봇을 선택한 다음 [로봇 모델 선택]을 클릭합니다.



- 브랜드 뒤의 드롭다운 상자에서 로봇 브랜드를 선택하면 해당 브랜드의 모든 로봇이 오른쪽 창에 나타납니다.

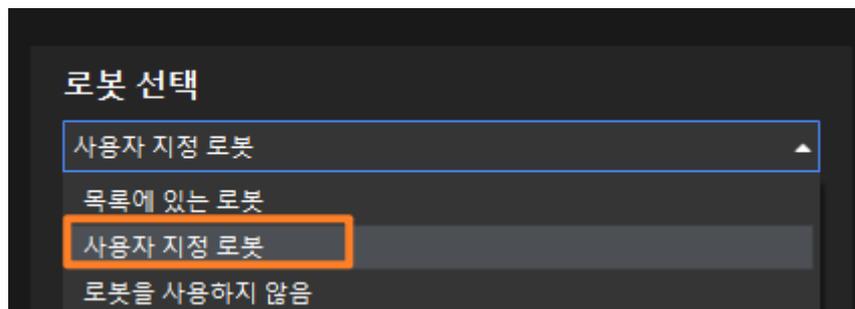


- **속, 적재량, 도달 범위**를 선택하여 로봇의 범위를 더 좁힙니다.
- **검색 상자에** 로봇 브랜드 및 모델(영어로 대소문자 구분 안 함)을 입력하여 로봇을 필터링할 수 있습니다. 입력란의 내용을 지우려면 버튼을 클릭하십시오.

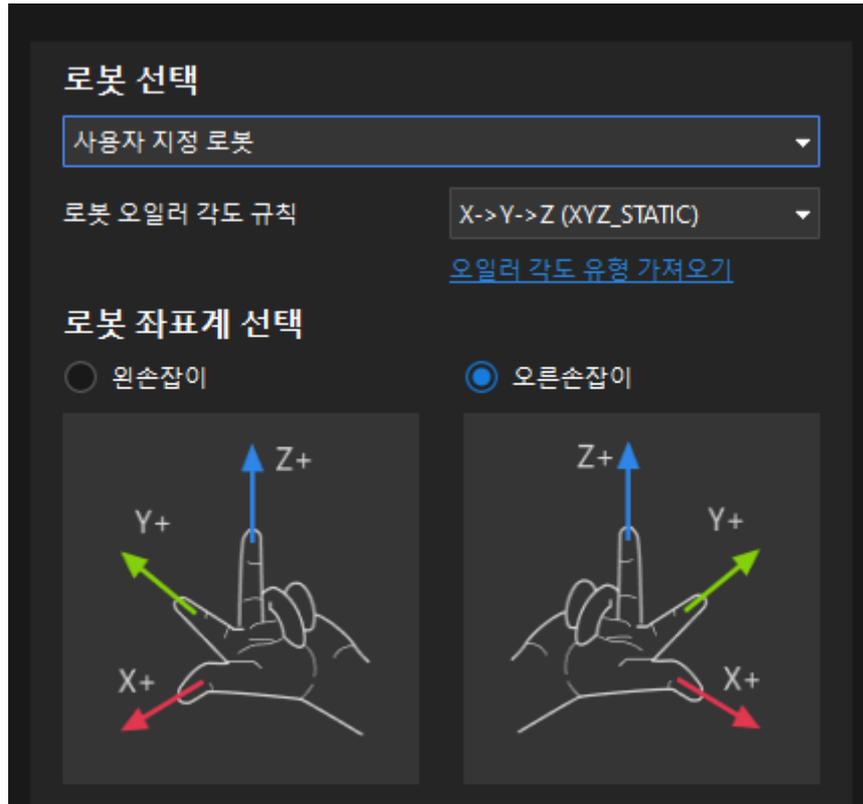


로봇 라이브러리에 필요한 로봇이 없으면 이 창을 닫고 《로봇을 도입하기》를 참조하여 작업한 후 여기서 작업을 수행합니다.

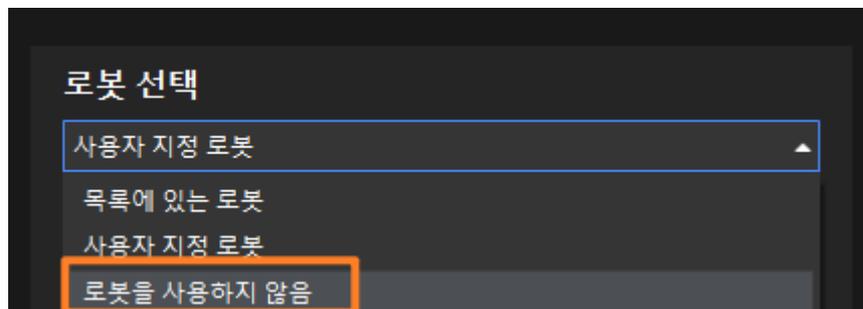
- 오른쪽에서 선택할 로봇 위에 마우스를 올리면 회색 테두리와 [ **선택** ] 버튼이 표시되며 버튼을 클릭하고 창을 닫습니다.
- [ **다음** ]을 클릭하여 통신 구성 인터페이스로 들어갑니다.
- 다른 로봇 선택(예: 트러스 로봇)
  - **로봇 선택** 드롭다운 상자를 클릭하고 **사용자 지정 로봇**을 선택합니다.



- 로봇의 오일러 각 유형을 선택합니다.
- 로봇 좌표계를 선택합니다.



- [다음]을 클릭하여 통신 구성 인터페이스로 들어갑니다.
- 로봇을 사용하지 않도록 선택(로봇 시나리오가 없음, 예를 들면 측량)
  - 로봇 선택 드롭다운 상자를 클릭하고 **로봇을 사용하지 않기를** 선택합니다.



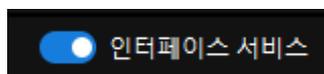
- [다음]을 클릭하여 Adapter 인터페이스 구성 화면으로 들어갑니다.

 로봇을 사용하지 않기를 선택한 후 통신 방식은 Adapter 통신만 선택할 수 있습니다.

#### 4.2.4.12. 인터페이스 서비스 시작

이 부분에서는 인터페이스 서비스를 시작하는 방법에 대해 설명합니다.

Mech-Vision 메인 인터페이스에서 아래 그림과 같이 툴 바의 맨 오른쪽에 있는 "인터페이스 서비스" 스위치를 클릭하면 인터페이스 서비스가 활성화됩니다.



시작에 성공하면 Mech-Vision 로그 창의 콘솔 탭 아래에 "XXXX 인터페이스 서비스 시작" 메시지가

표시됩니다.

시작에 실패하면 로봇 및 통신 구성이 올바른지 확인하십시오.

#### 4.2.4.13. 로봇 라이브러리 도구 사용 가이드

이 부분에서는 "로봇 라이브러리 도구"와 관련된 작업을 소개합니다.

##### 로봇을 도입하기

로봇을 도입하기 전에 [온라인 로봇 라이브러리](#)에서 필요한 로봇 모델 파일(파일 접미사는 .mrob)을 다운로드해야 합니다.

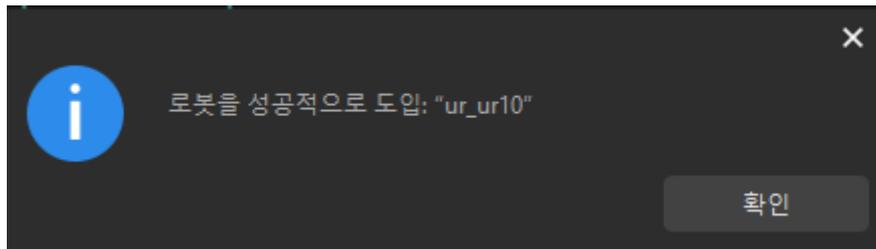


“온라인 로봇 라이브러리”에서 필요한 로봇을 찾을 수 없는 경우 로봇 모델을 만들어야 합니다. 그다음 Mech-Vision 소프트웨어로 도입합니다.

다음 방법 중 하나를 사용하여 로봇 모델을 도입할 수 있습니다.

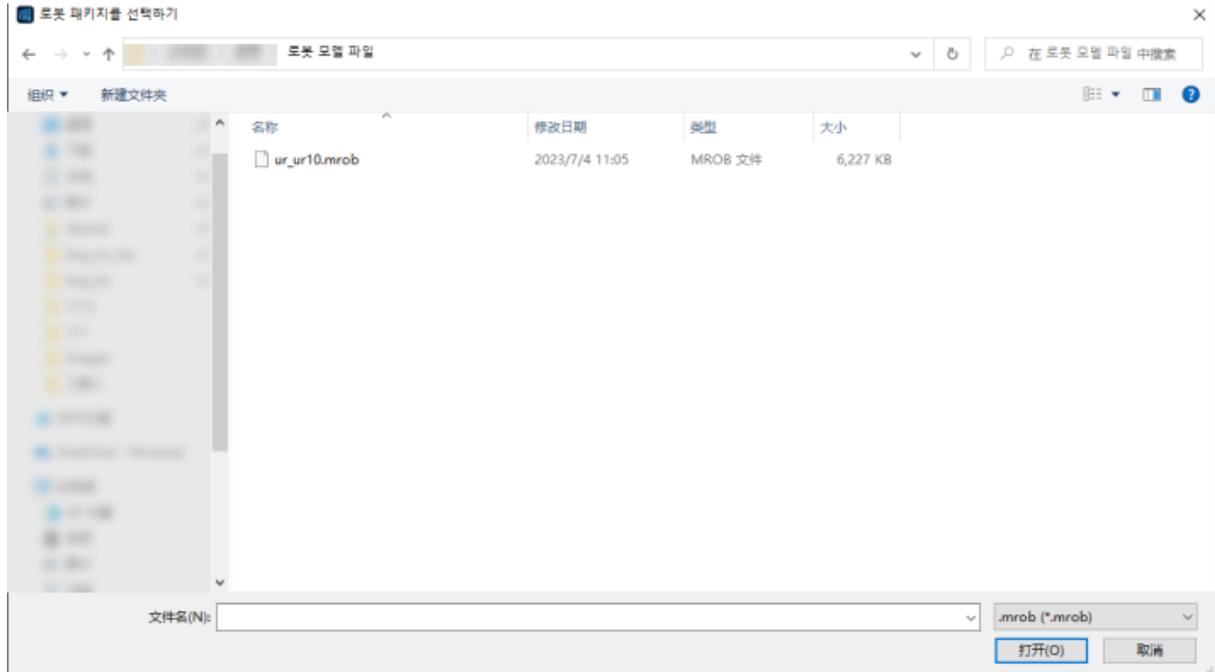
##### ● 방법1

- 마우스 왼쪽 버튼을 누르고 다운로드한 .mrob 파일을 Mech-Vision 소프트웨어 창으로 드래그한 다음 마우스 왼쪽 버튼을 놓습니다.
- "로봇 도입 완료" 프롬프트 창이 팝업되고 마지막으로 [ **확인** ]을 클릭합니다.

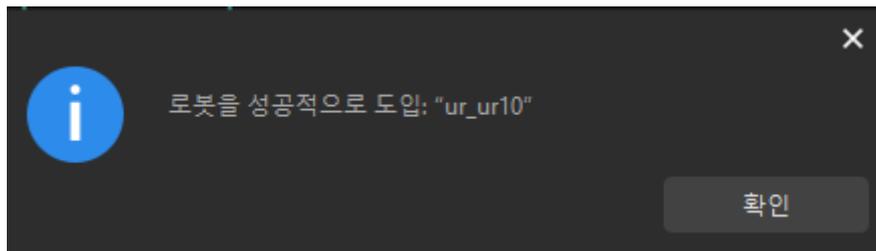


##### ● 방법2

- Mech-Vision 메뉴 바에서 **로봇과 통신 > 로봇 라이브러리 도구 > 로봇을 도입하기**를 클릭합니다.
- 팝업 창에서 다운로드한 .mrob 파일을 선택하고 [ **열기** ]를 클릭합니다.



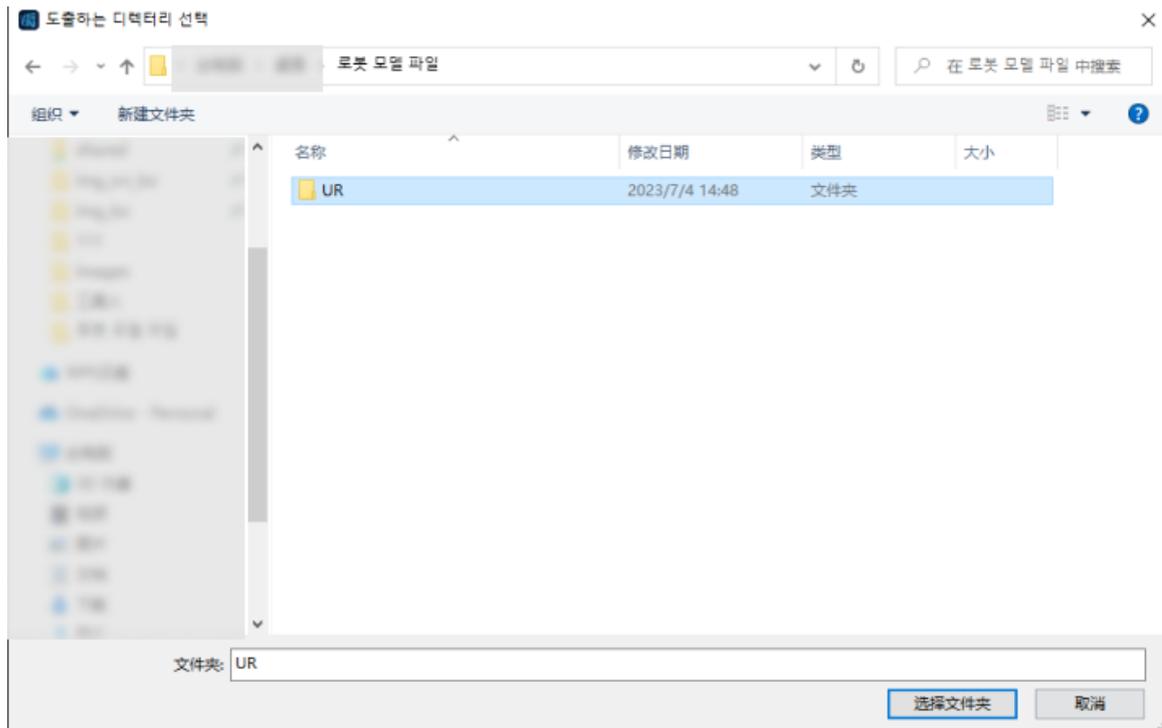
- "로봇 도입 완료" 프롬프트 창이 팝업되고 마지막으로 [ 확인 ]을 클릭합니다.



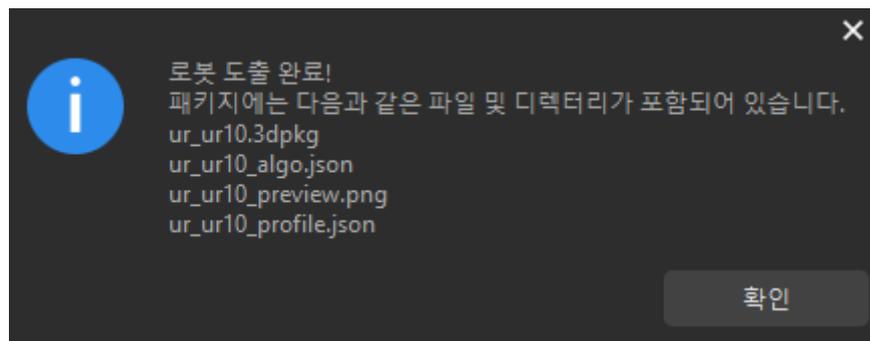
#### 현재 로봇을 도출하기

다음 방법을 사용하여 현재 로봇 모델을 도출할 수 있습니다.

1. Mech-Vision메뉴 바에서 **로봇과 통신** > **로봇 라이브러리 도구** > **현재 로봇을 도출하기**를 클릭합니다.
2. 팝업 창에서 도출할 폴더를 선택하고 [ **폴더 선택** ]을 클릭합니다.



3. “로봇 도출 완료” 프롬프트 창이 팝업되고 마지막으로 [ 확인 ]을 클릭합니다.

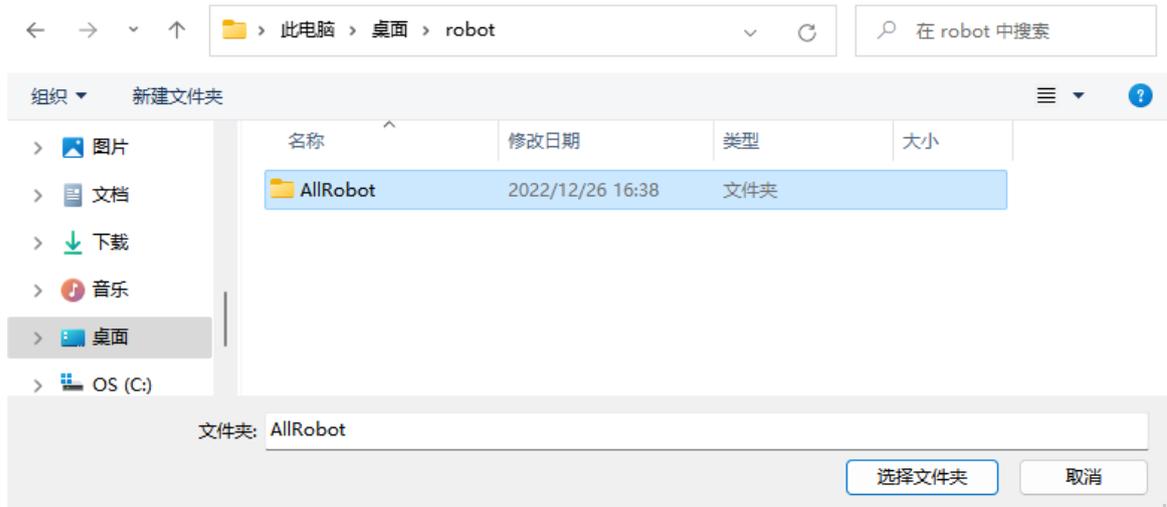


 도출이 완료되면 도출한 폴더에 .mrob 파일이 나타납니다. 이 파일은 도출한 로봇 모델입니다.

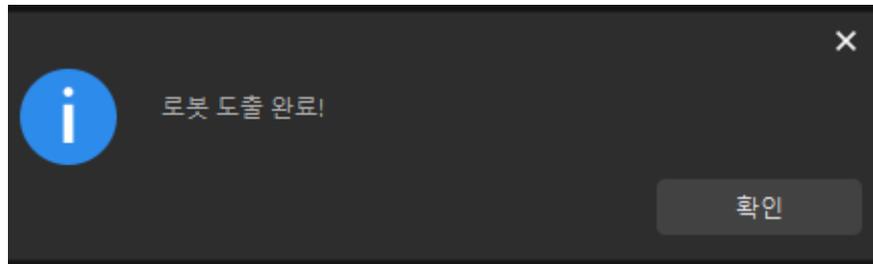
### 모든 로봇을 도출하기

다음 방법을 사용하여 Mech-Vision 소프트웨어에서 모든 로봇 모델을 도출할 수 있습니다.

1. Mech-Vision 메뉴 바에서 **로봇과 통신** > **로봇 라이브러리 도구** > **모든 로봇을 도출하기**를 클릭합니다.
2. 팝업 창에서 도출할 폴더를 선택하고 [ **폴더 선택** ]을 클릭합니다.



3. “로봇 도출 완료” 프롬프트 창이 팝업되고 마지막으로 [ 확인 ]을 클릭합니다.



도출이 완료되면 도출한 폴더에 .mrob 파일이 많이 나타납니다. 이 파일은 도출한 로봇 모델입니다.

#### 4.2.5. 스텝 사용 가이드

이 부분에서는 스텝에 대한 일반적인 작업을 소개할 것입니다.

[스텝을 검색 & 추가 & 삭제하기](#)

[스텝 간 연결 생성 및 삭제](#)

[스텝의 입력 및 출력을 보기](#)

[스텝의 다양한 상태 보기](#)

[스텝 실행 및 시각화 출력 결과를 보기](#)

[스텝의 비교를 추가&수정&확인하기](#)

여러 스텝의 조합을 프로시저라고 합니다. 서로 다른 프로젝트에는 종종 일관되거나 유사한 알고리즘 처리 프로세스가 있습니다. 이러한 고정된 알고리즘 처리 스텝을 캡슐화하고 결합하여 쉽게 재사용할 수 있습니다.

다음과 같은 내용을 통해 프로시저 관련 사용 가이드에 대해 알아보십시오.

[프로시저의 기본 사용법](#)

## 프로시저 파라미터 수정

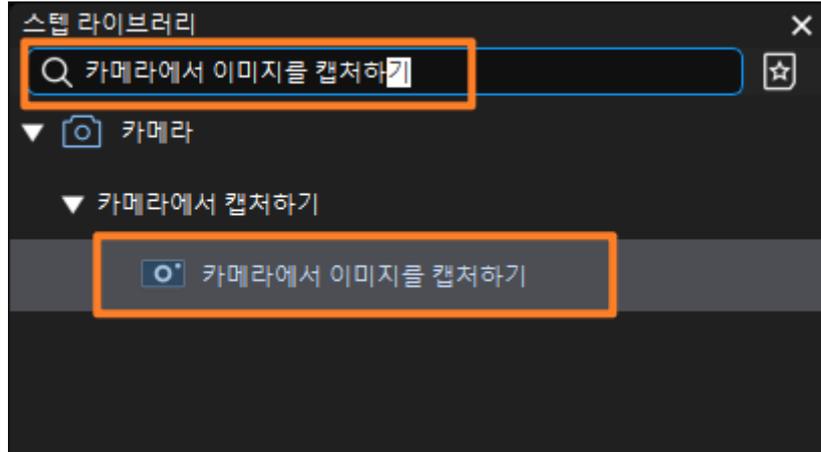
### 프로시저 자체 정의

#### 4.2.5.1. 스텝을 검색 & 추가 & 삭제하기

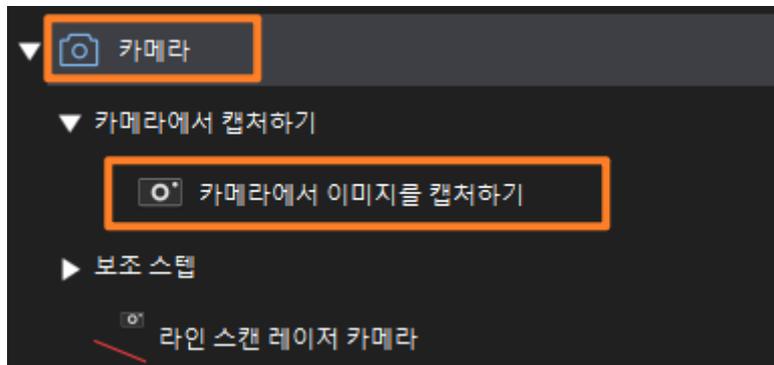
“카메라에서 이미지를 캡처하기” 스텝을 예로 들겠습니다.

##### 스텝을 검색하는 방법

- 스텝 라이브러리의 검색창에서 스텝 명칭(혹은 키워드만) 직접 입력합니다.

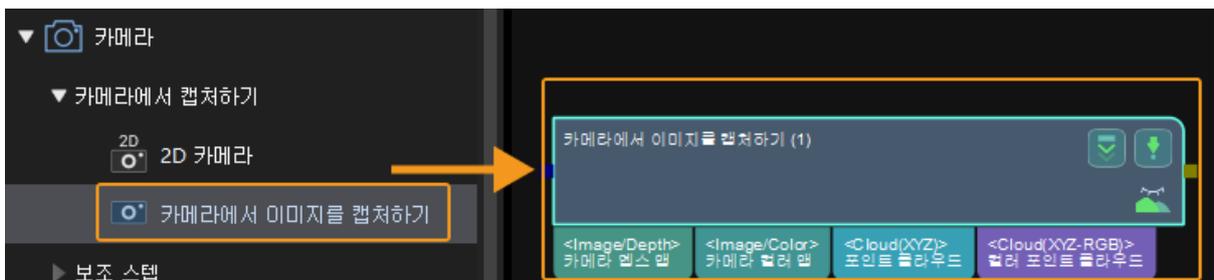


- 스텝 분류 클래스에서 검색합니다.



##### 스텝을 추가하는 방법

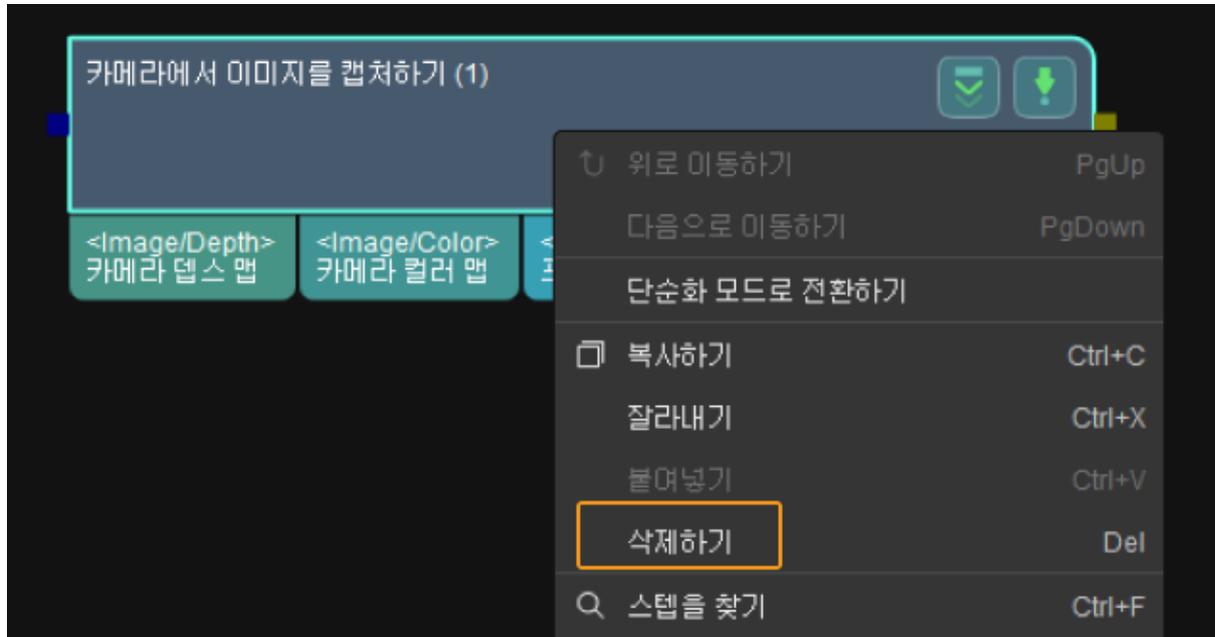
마우스 왼쪽 키로 원하는 스텝을 길게 눌러 프로젝트 편집 구역으로 드래그합니다.



스텝은 사용자가 새로운 프로젝트를 만들거나 기존 프로젝트를 열었을 때에만 추가할 수 있습니다.

#### 스텝을 삭제하는 방법

- 마우스 왼쪽 키로 스텝을 선택한 후 **Delete** 키를 누릅니다.
- 마우스 오른쪽 키로 스텝을 클릭하고 나서 팝업 창에서 [삭제] 버튼을 누릅니다.

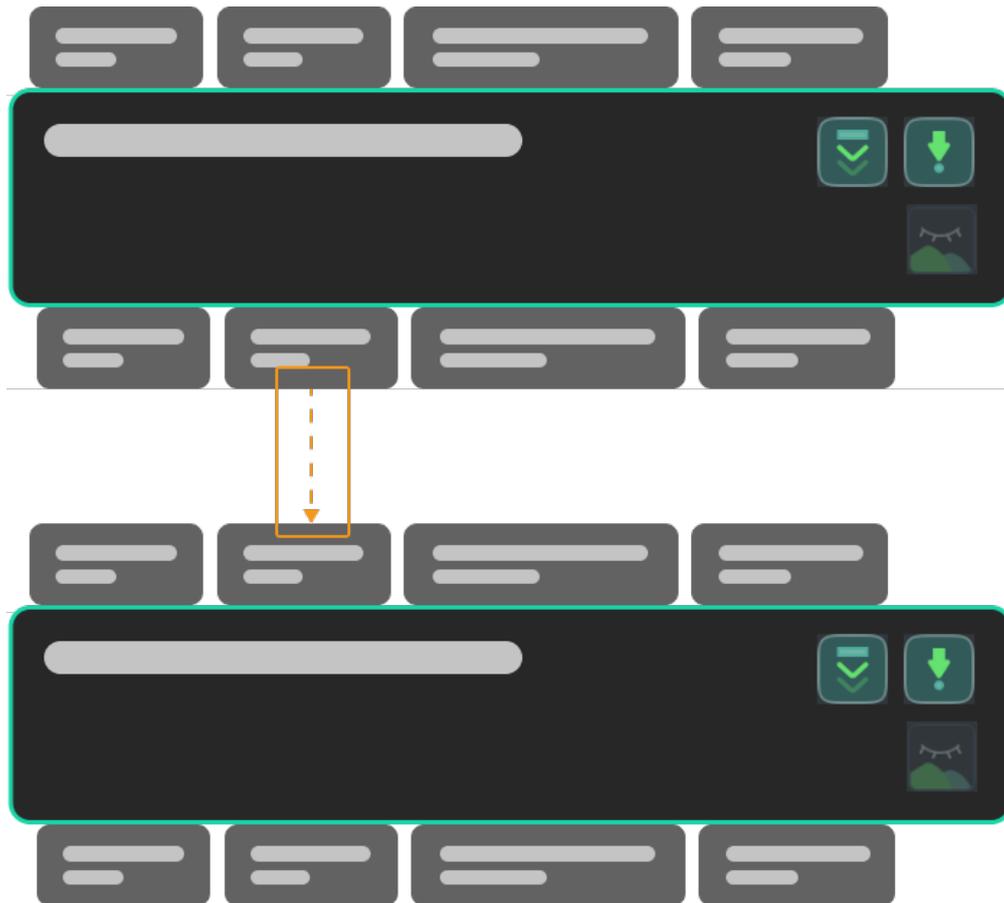


#### 4.2.5.2. 스텝 간 연결 생성 및 삭제

##### 연결을 만드는 방법

마우스를 "카메라에서 이미지를 캡처하기"의 출력 포트에 이동하고 왼쪽 마우스 버튼을 누른 상태에서 마우스를 "덱스 맵을 포인트 클라우드로 전환하기"의 입력 포트에 드래그한 다음 왼쪽 버튼을 놓아 연결을 만듭니다.

하나의 출력 포트는 여러 입력 포트에 연결할 수 있지만 입력 포트는 하나의 출력 포트에만 연결할 수 있습니다.



동일한 데이터 유형의 포트 간에만 연결을 생성할 수 있습니다. 예를 들어 **Cloud(XYZ-Normal)**의 출력 포트는 **Cloud(XYZ-Normal)**의 입력 포트와 연결해야 합니다. 다른 데이터 유형의 포트는 연결을 만들 수 없습니다.

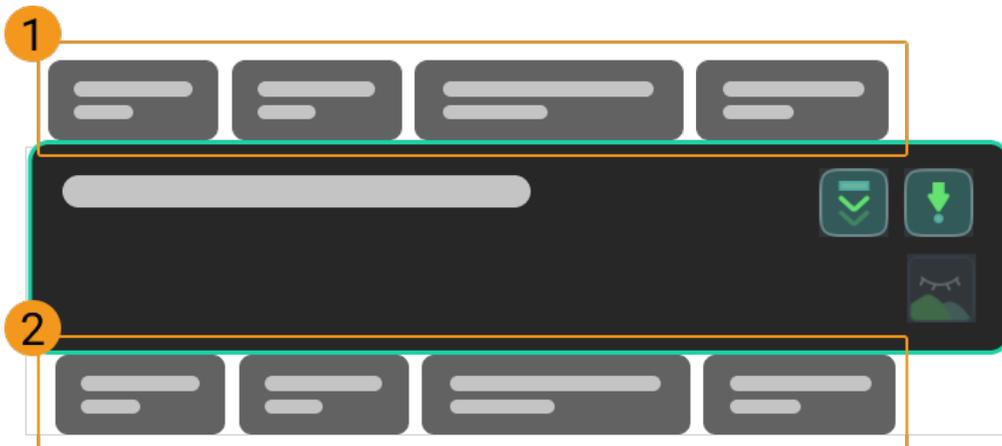
#### 연결을 삭제하는 방법

- 마우스 왼쪽 키로 연결선을 클릭하면 선택된 연결선이 밝아질 것이고 이 때 **Delete** 키를 누르면 삭제할 수 있습니다.
- 데이터 연결을 마우스 오른쪽 버튼으로 클릭하면 연결이 강조 표시되고 메뉴 표시줄이 나타나면 [**삭제**]를 클릭합니다.

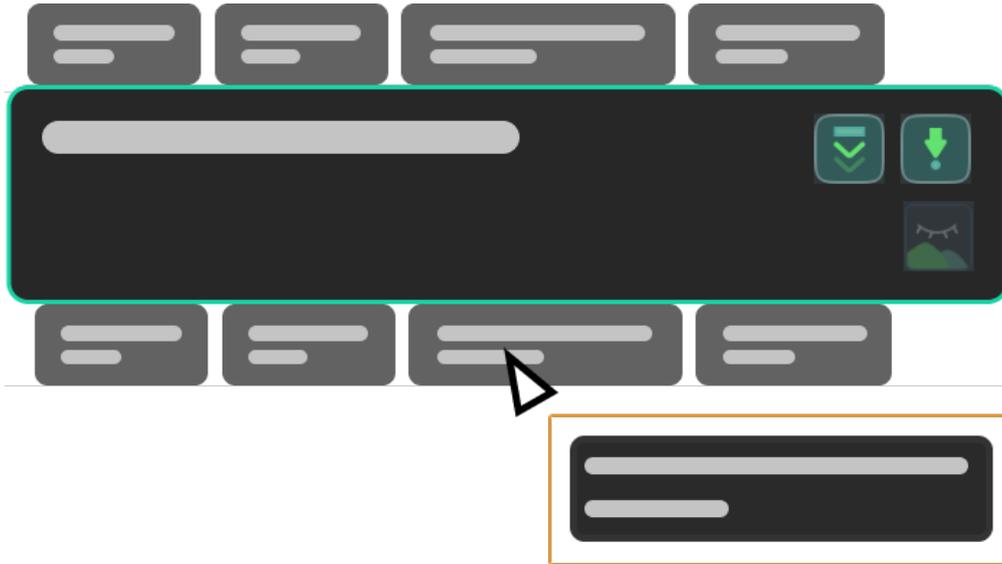


#### 4.2.5.3. 스텝의 입력 및 출력을 보기

각 스텝의 입력 및 출력 포트는 아래 그림과 같이 위쪽과 아래쪽에 배치되며 1은 입력 포트이고 2는 출력 포트입니다.



포트로 마우스를 이동하면 아래 그림과 같이 데이터 유형에 대한 자세한 설명이 나타납니다.



#### 4.2.5.4. 스텝의 다양한 상태 보기

스텝을 실행한 후 스텝 테두리의 색상을 관찰하여 스텝의 다양한 상태를 볼 수 있습니다.

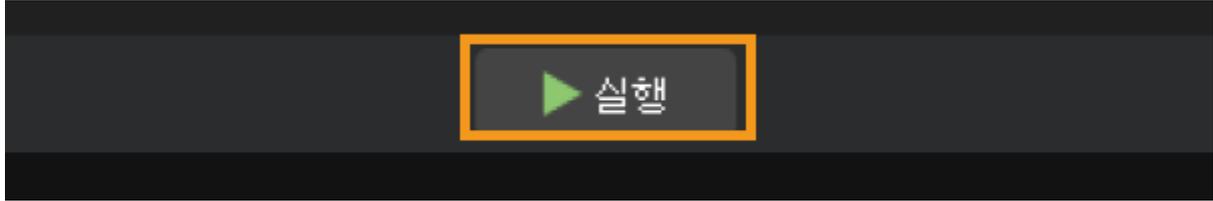
상태	예시 그림
정상	
선택됨	
경고	
잘못된 경우	

#### 4.2.5.5. 스텝 실행 및 시각화 출력 결과를 보기

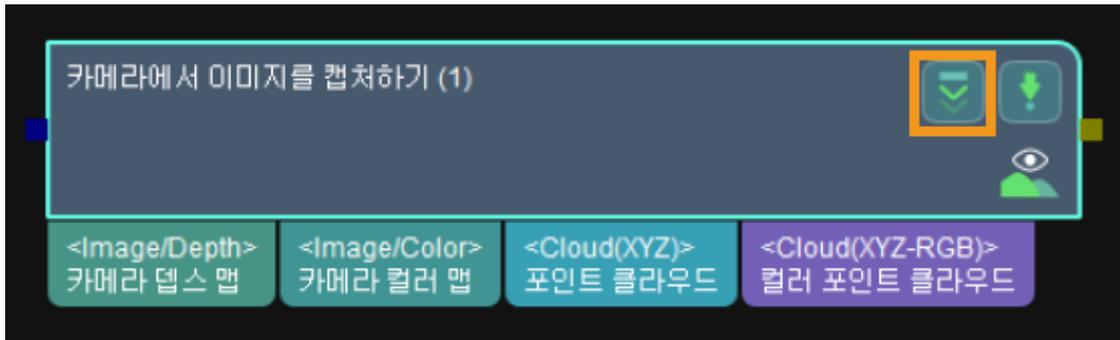
전체 프로젝트의 원활한 실행은 개별 스텝의 성공적인 실행에 달려 있습니다. 이 부분에서는 스텝을 실행하고 디버그 출력 기능을 사용하여 스텝의 실행 효과를 보는 방법을 설명합니다.

### 스텝 실행

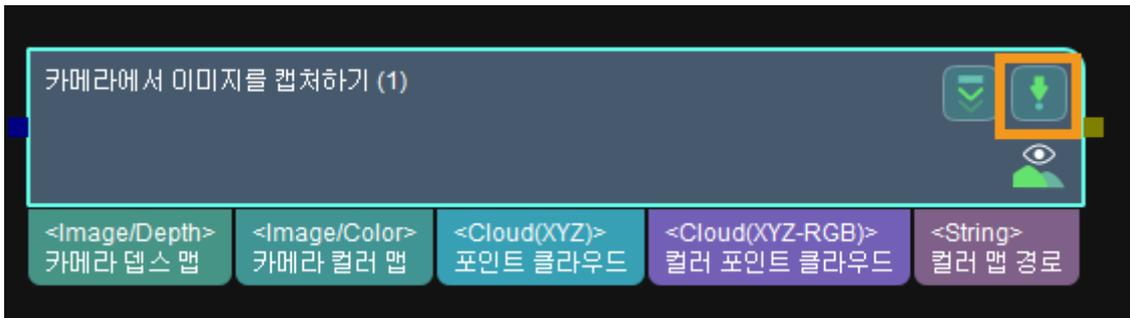
- 프로젝트 톨 바에서 [ 실행 ] 버튼을 클릭하여 전체 프로젝트를 실행합니다.



- 현재 스텝부터 아래로 프로젝트를 실행하려면 [ 실행 ] 버튼을 클릭하십시오.



- 단일 스텝을 실행하려면 [ 실행 ] 버튼을 클릭하십시오.



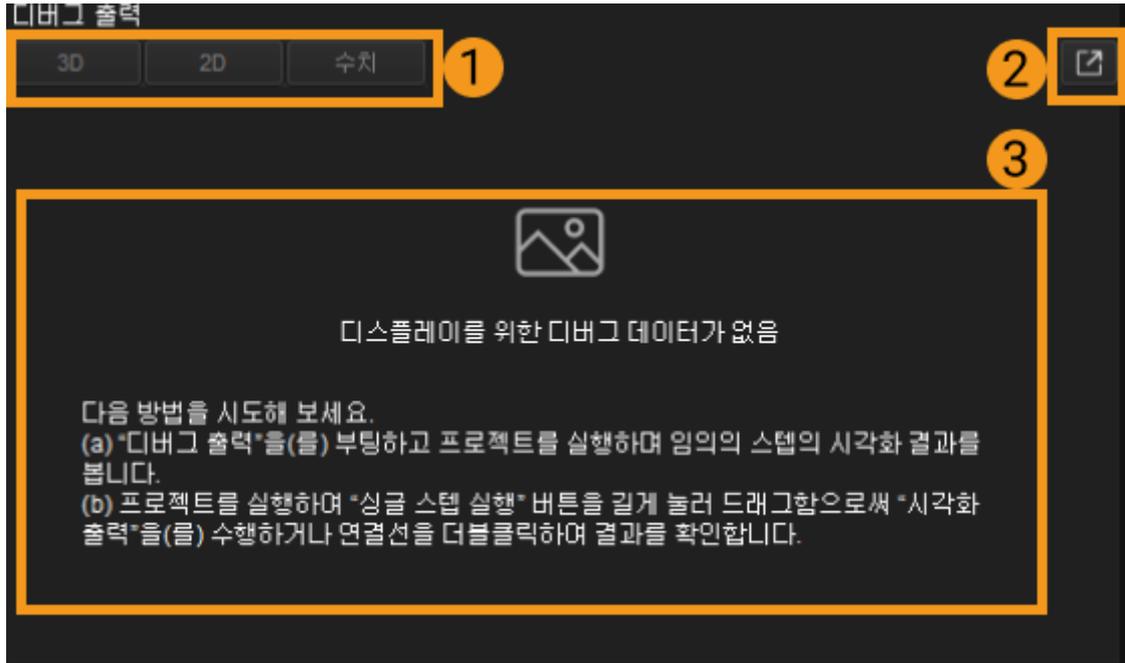
### 디버그 출력

[디버그 출력] 기능의 시각화 창은 프로젝트 결과를 표시하는 창이고 Mech-Vision 소프트웨어의 중요한 구성 요소입니다.

프로젝트를 디버그할 때 [디버그 출력] 기능을 사용하여 해당 스텝의 시각화 출력 결과를 직관적으로 볼 수 있습니다.

### 화면 소개

[디버그 출력] 화면은 Mech-Vision 소프트웨어 메인 인터페이스의 오른쪽 상단에 있습니다. 아래 그림과 같습니다.



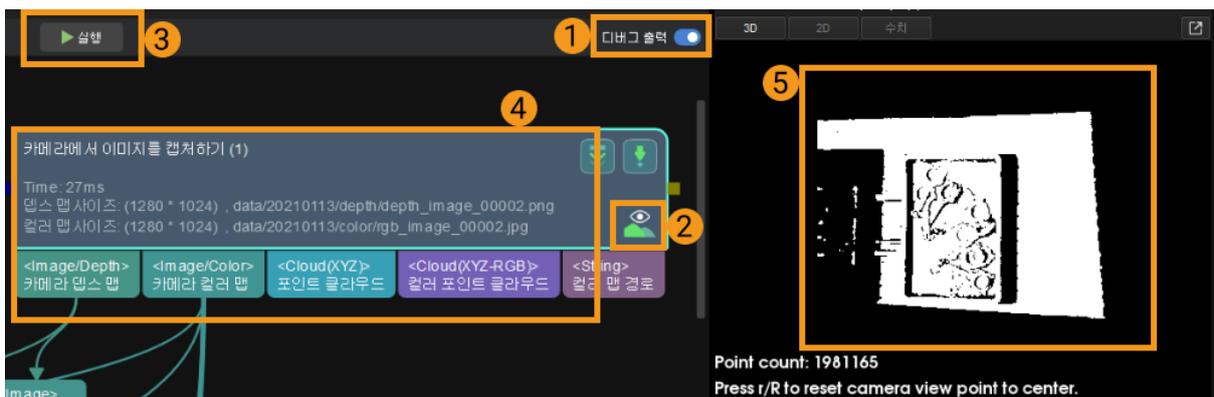
화면에 관한 설명은 아래와 같습니다.

- ①은 출력된 결과의 데이터 유형입니다.
- ②를 클릭하면 팝업창이 나타날 것이며 디버그 출력 결과를 단독으로 확인할 수 있습니다.
- ③은 시각화 출력 결과의 표시 구역입니다.

#### 사용 방법

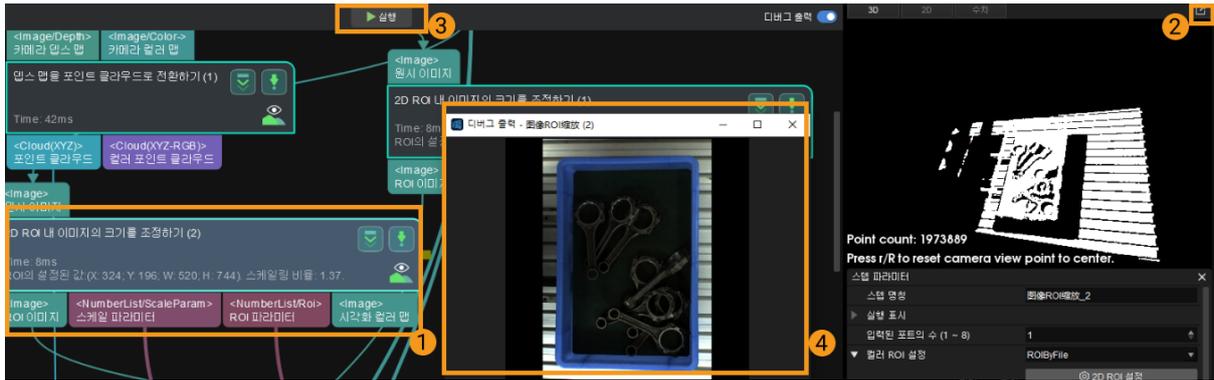
##### 디버그 모드에서 하나의 스텝의 시각화 출력 결과를 보기

프로젝트 툴 바에 있는 **디버그 출력** 버튼(①)을 클릭하고 오른쪽에 있는 시각화 버튼(단힘) 을 클릭하고 시각화 버튼을 켜 있는 상태 로 바꿉니다. 다음으로 [ **실행** ] 버튼을 클릭하며 시각화 출력 결과를 확인하려는 스텝을 클릭하여 바로 시각화 출력 창에서 해당 스텝의 출력 결과를 볼 수 있습니다.



##### 디버그 모드에서 하나의 스텝이 여러 번 실행된 후의 시각화 출력 결과를 보기

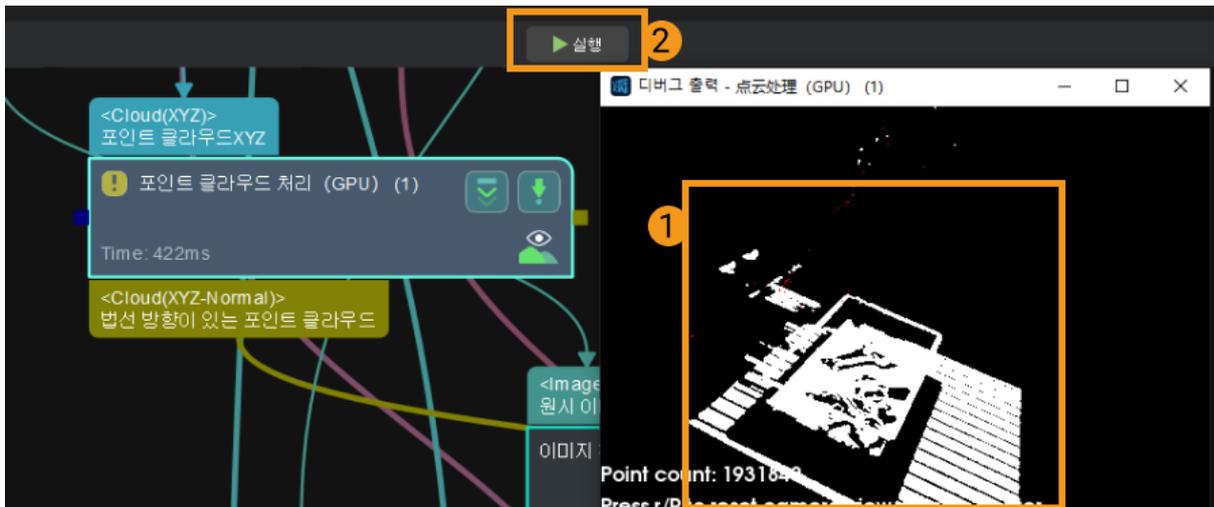
어떤 스텝의 여러 번 실행 결과를 보려면 스텝을 먼저 선택하고 을 클릭하여 디버그 출력 창이 팝업될 것입니다. 다음으로 프로젝트 툴 바에 있는 [ **실행** ] 버튼을 클릭하여 팝업된 디버그 출력 창(④)에서 해당 스텝의 여러번의 실행 결과를 볼 수 있습니다.



여러 개 스텝의 여러번 실행 결과를 확인하려면 위 방법을 통해 여러 개 디버그 출력 창을 단독으로 표시하도록 설정하면 됩니다.

### 디버그 모드에서 다른 시각에서 단일 스텝의 여러 시각화 출력 결과를 보기

다른 시각으로 스텝의 출력 결과를 보고 싶으시다면 단독으로 나온 디버그 출력 창에서 마우스를 드래그하여 시각을 조정할 수 있습니다. 다음으로 프로젝트 툴 바에 있는 [ 실행 ] 버튼을 눌러 해당 시각에서 스텝의 여러번 실행 결과를 볼 수 있습니다.



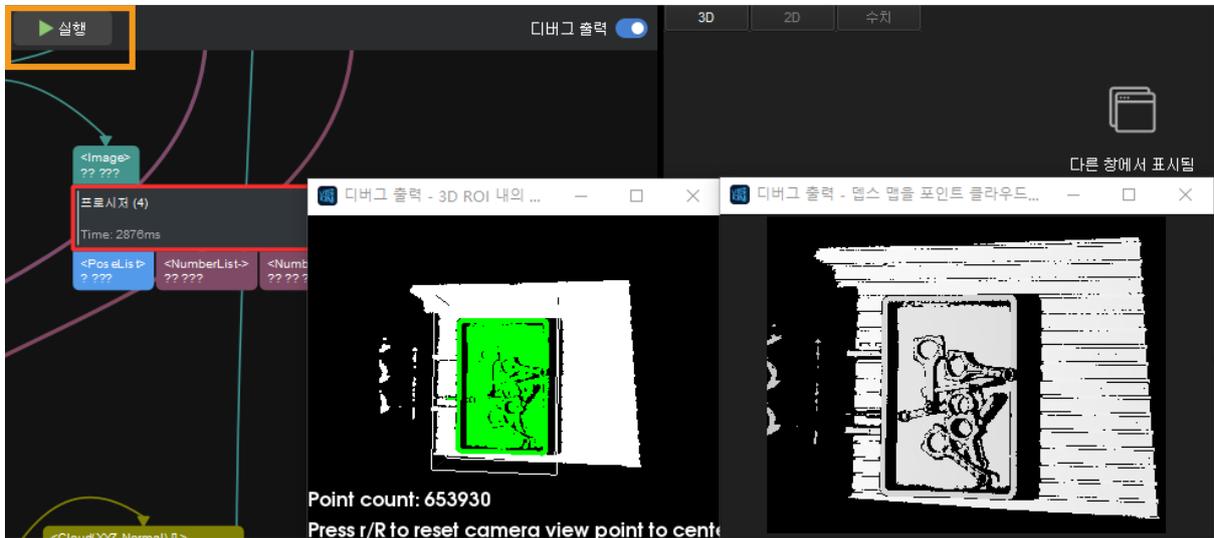
### 프로시저 내 스텝의 시각화 출력 결과를 전환해서 확인하기

한 프로젝트에 프로시저가 있고 프로시저 내 시각화 출력될 수 있는 스텝이 포함되며 기능이 활성화될 때 디버그 출력 기능을 통해 해당 프로시저의 스텝 결과를 확인할 수 있습니다.

프로시저를 먼저 선택하고 시각화 출력 창의 언더바(2)에서 시각화 출력 결과를 단독으로 표시하려는 스텝을 선택합니다.



다음으로 프로젝트 툴 바에 있는 [ 실행 ] 버튼을 눌러 프로시저 내 스텝이 여러번 실행되는 시각화 출력 결과를 볼 수 있습니다.



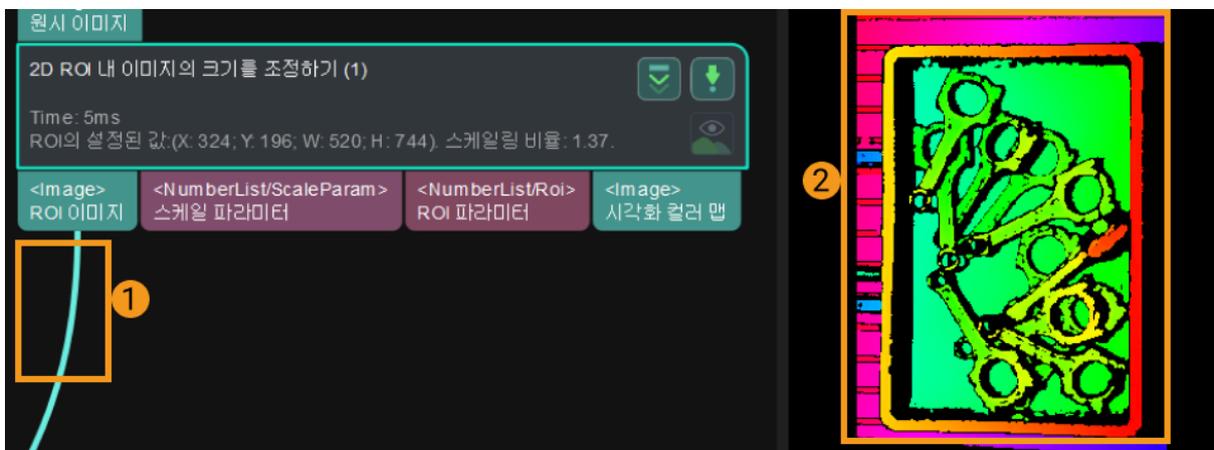
비 디버그 모드에서 스텝의 시각화 출력 결과를 보기

프로젝트 툴 바에 있는 **디버그 출력** 버튼이 닫힌 상태에서 스텝의 시각화 출력 결과를 보려면 다음과 같이 두 가지 방법을 시도할 수 있습니다.

-  을 마우스 왼쪽 버튼으로 길게 눌러 드래그하고 [ **시각화 출력** ]을 클릭하며 디버그 출력 창에서 스텝의 출력 결과를 볼 수 있습니다.



- 스텝 사이의 연결선을 더블클릭하여 디버그 출력 창에서 스텝의 시각화 출력 결과를 볼 수 있습니다.



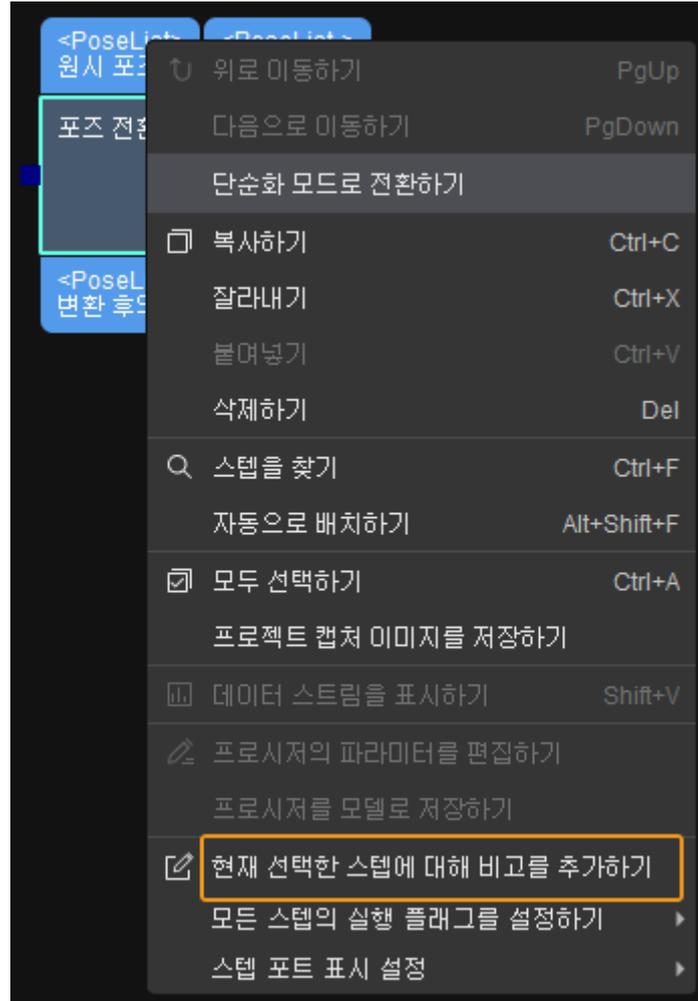
스텝의 시각화된 출력 결과를 볼 수 있는 전제는 프로젝트 또는 프로젝트에 있는 스텝이 이미 실행된 것입니다.

#### 4.2.5.6. 스텝의 비교를 추가&수정&확인하기

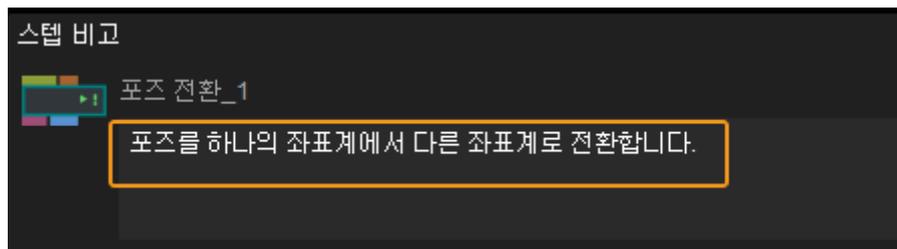
스텝의 비교 기능을 통해 프로젝트의 임의의 스텝 혹은 프로시저에 대해 해석이나 설명을 추가할 수 있습니다.

##### 비교를 추가하는 방법

1. 비교를 추가하려고 하는 스텝 혹은 프로시저를 마우스 오른쪽 키로 클릭한 다음에 **현재 선택한 스텝에 대해 비교를 추가하기**를 선택하면 됩니다.

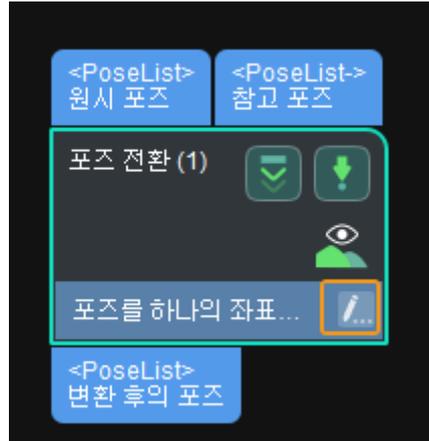


2. 팝업창에서 비교를 추가하고 창을 닫으면 됩니다.



#### 비교를 편집하는 방법

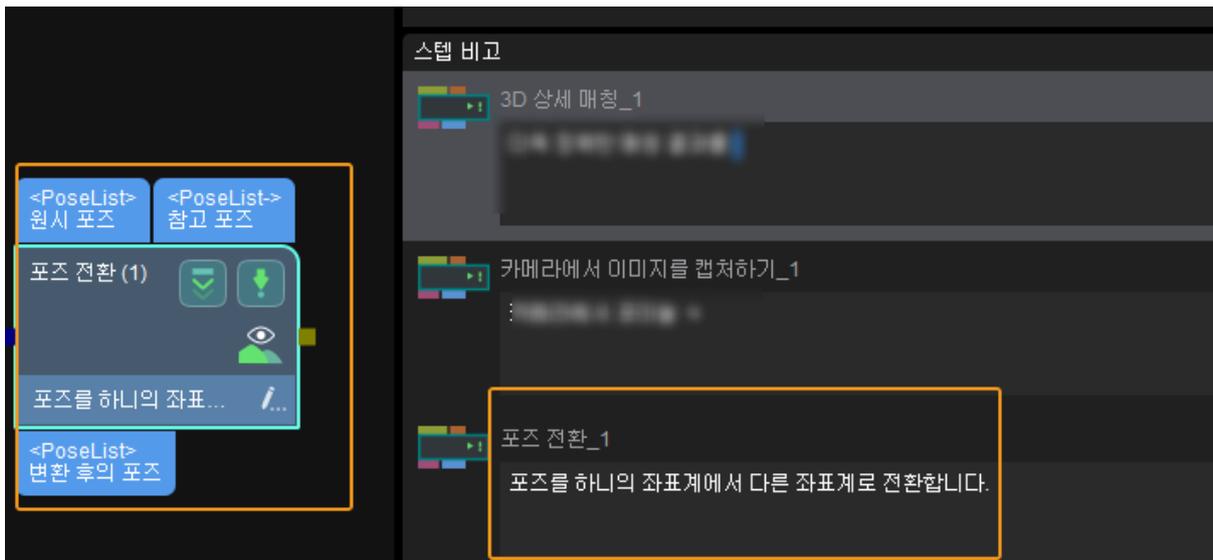
비교를 편집하려면 를 클릭하십시오.



#### 비교 내용을 확인하는 방법

**스텝 비교** 창을 엽니다. 오른쪽 프로젝트 구성 영역 하단의 리스트에 해당 탭이 없으면 메뉴 바 **뷰어 > 스텝 비교**가 체크되어 있는지 확인하십시오.

여기에서 이미 추가한 비교 내용을 볼 수 있으며 비교 내용을 클릭하면 프로젝트 편집 구역이 대응하는 스텝 혹은 프로시저로 이동하여 깜박이는 표시가 나타날 것입니다.



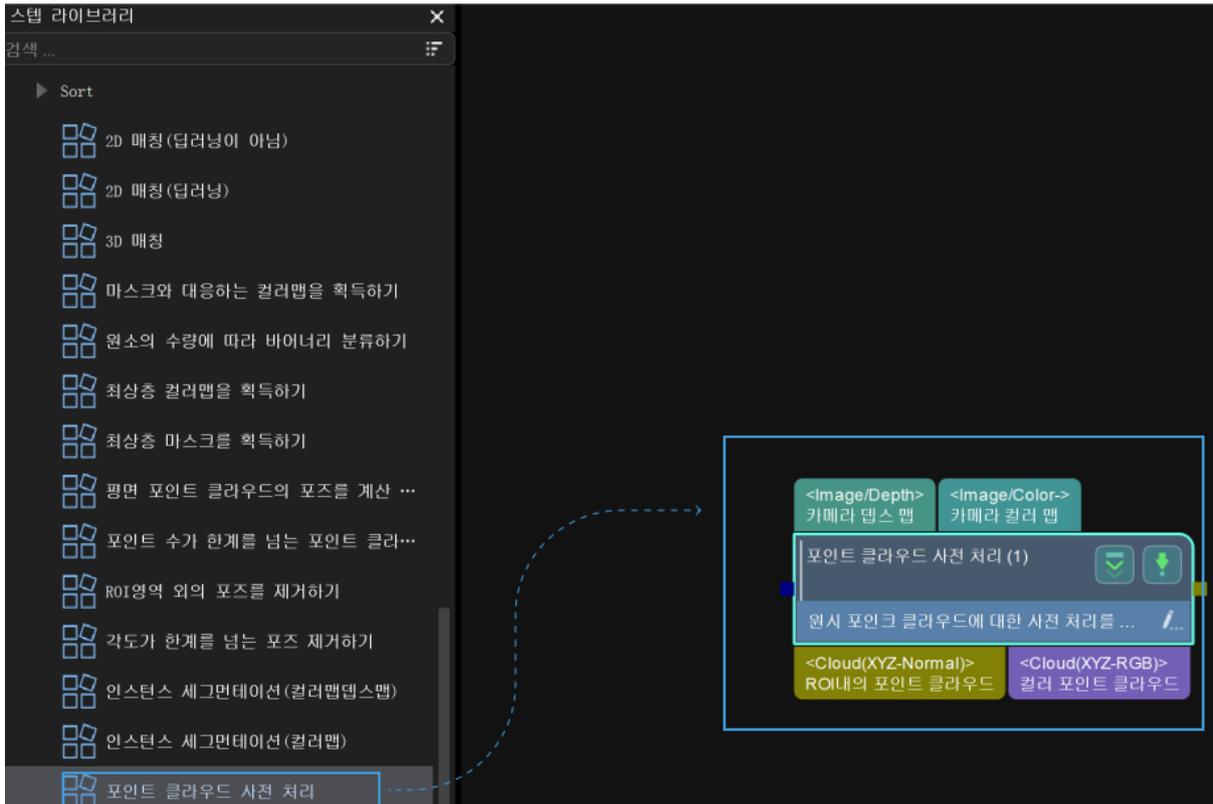
#### 4.2.5.7. 프로시저의 기본 사용법

프로시저를 알아보기 전에 먼저 **프로시저의 기본 사용법**을 살펴보십시오.

이 부분의 사용법은 **포인트 클라우드 사전 처리** 프로시저를 예로 사용합니다. 이 부분을 시작하기 전에 먼저 새로운 프로젝트를 만드십시오.

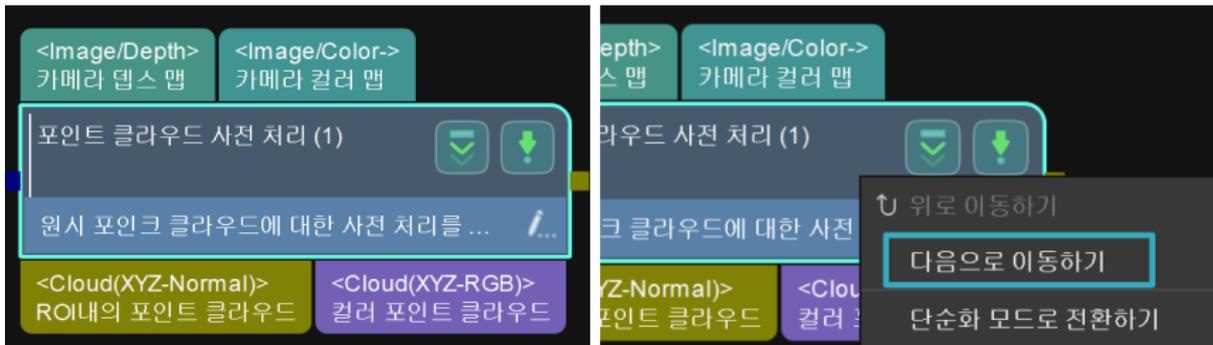
- 프로젝트에 프로시저를 추가하는 방법은 무엇입니까?

스텝 라이브러리에서 **자체 정의 > 포인트 클라우드의 사전 처리**를 찾아 프로젝트 편집 구역으로 드래그합니다.



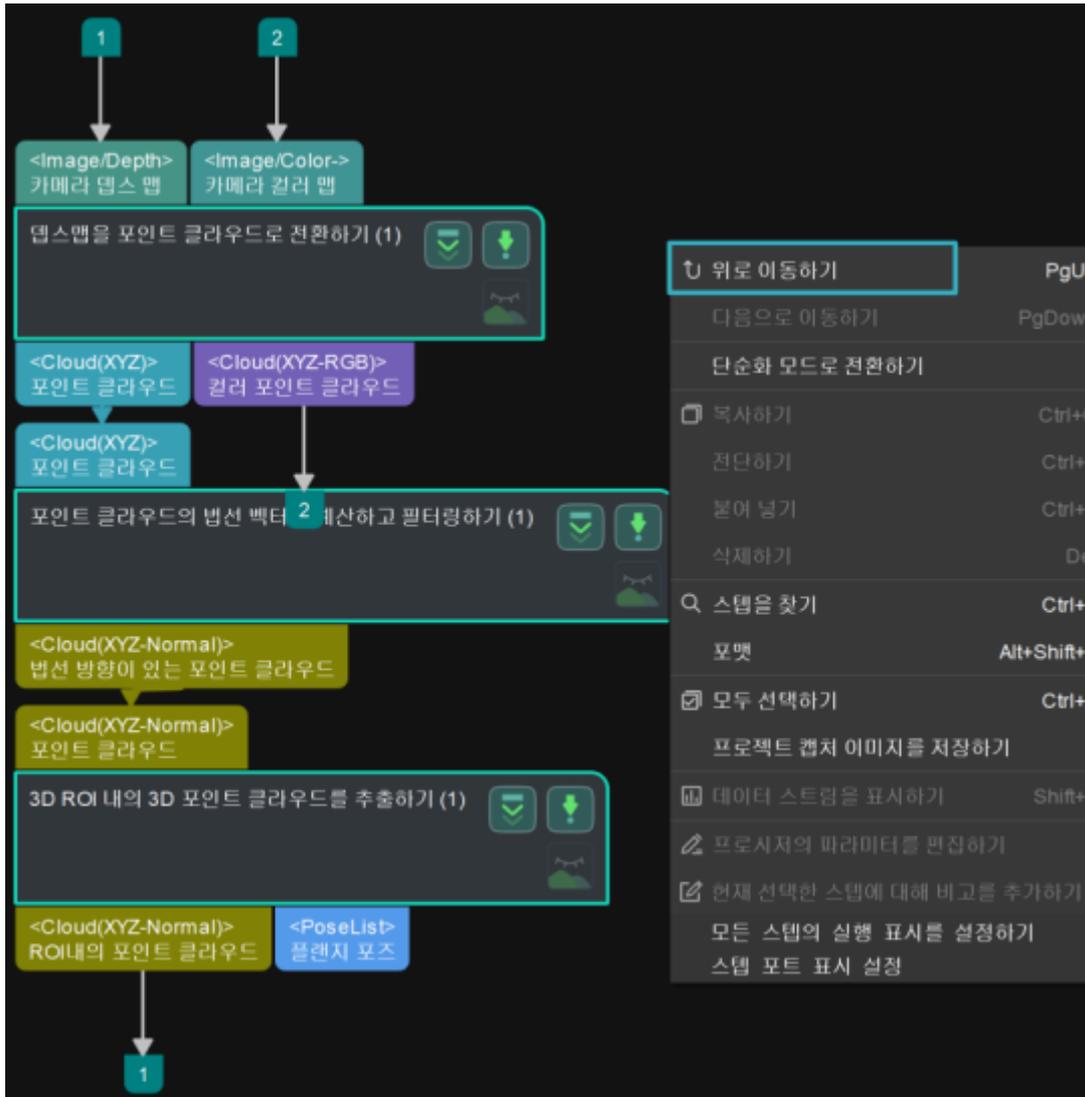
● 프로시저 화면에 들어가는 방법은 무엇입니까?

마우스 왼쪽 버튼으로 프로시저를 두 번 클릭하거나 프로시저를 마우스 오른쪽 버튼으로 클릭하고 팝업 메뉴 표시줄(단축키 **PgDown**)에서 [다음으로 이동하기]를 클릭합니다.



● 프로시저를 종료하는 방법은 무엇입니까?

프로젝트 편집 영역의 아무 곳에서 마우스 오른쪽 버튼으로 클릭해 팝업된 메뉴 바에서 [위로 이동하기](단축키 **PgUp**)를 클릭합니다.



#### 4.2.5.8. 프로시저 파라미터 수정

사용자는 다양한 응용 시나리오에 따라 프로시저의 파라미터를 수정할 수 있습니다. 이 부분에서는 **포인트 클라우드 사전 처리** 프로시저를 예로 들어 파라미터 수정의 관련 내용을 소개하며 주로 다음 두 부분을 포함합니다.

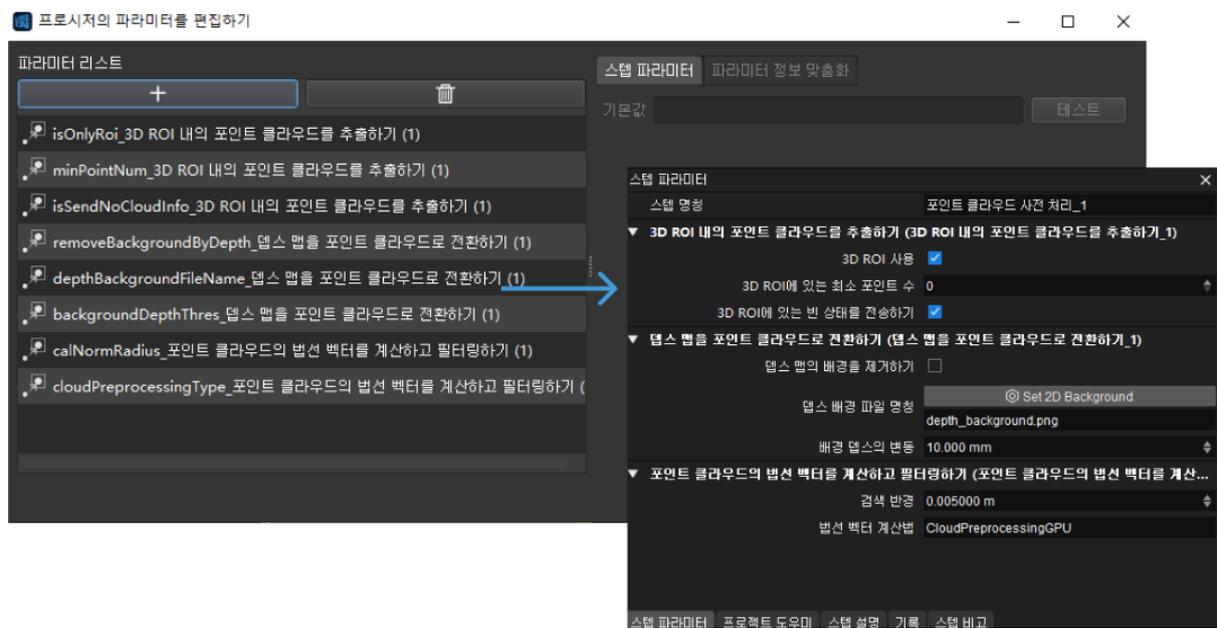
- 파라미터를 삭제, 추가하기
- 파라미터 표시 정보 자체 정의

**포인트 클라우드 사전 처리**를 마우스 오른쪽 버튼으로 클릭하고 팝업 메뉴에서 [ **프로시저의 파라미터를 편집하기** ]를 클릭합니다.



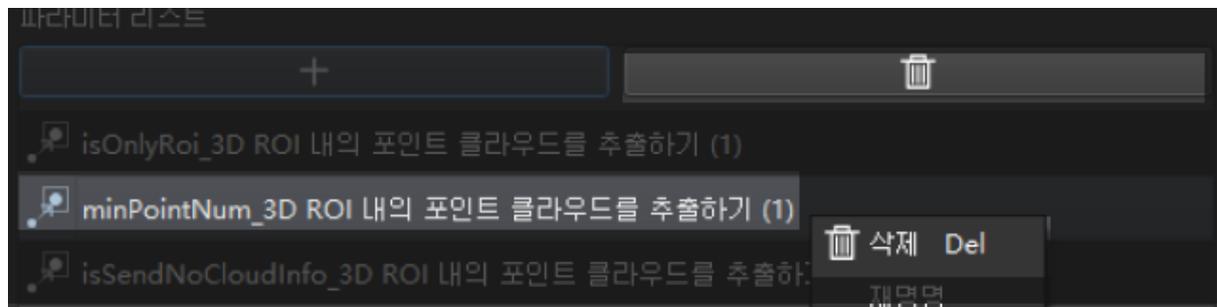
### 파라미터를 삭제, 추가하기

파라미터 리스트 중의 파라미터는 스텝 파라미터 열의 파라미터와 일대일로 대응합니다. 다음 그림과 같습니다.



### 삭제하기

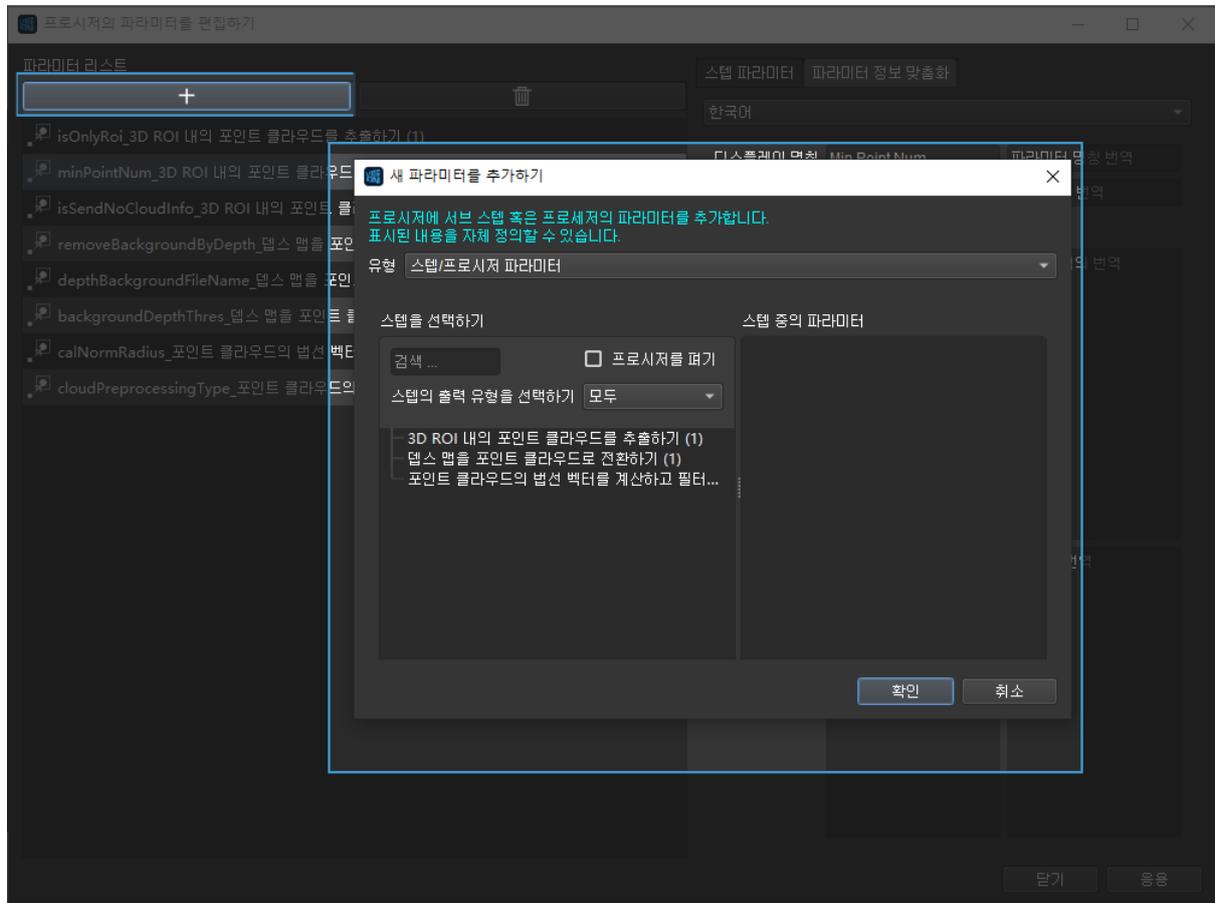
삭제할 파라미터를 마우스 왼쪽 버튼으로 클릭하고  를 클릭하거나 삭제할 파라미터를 마우스 오른쪽 버튼으로 클릭하고 팝업한 메뉴 바에서 [삭제]를 클릭하고 팝업 창에서 [예]를 클릭합니다.



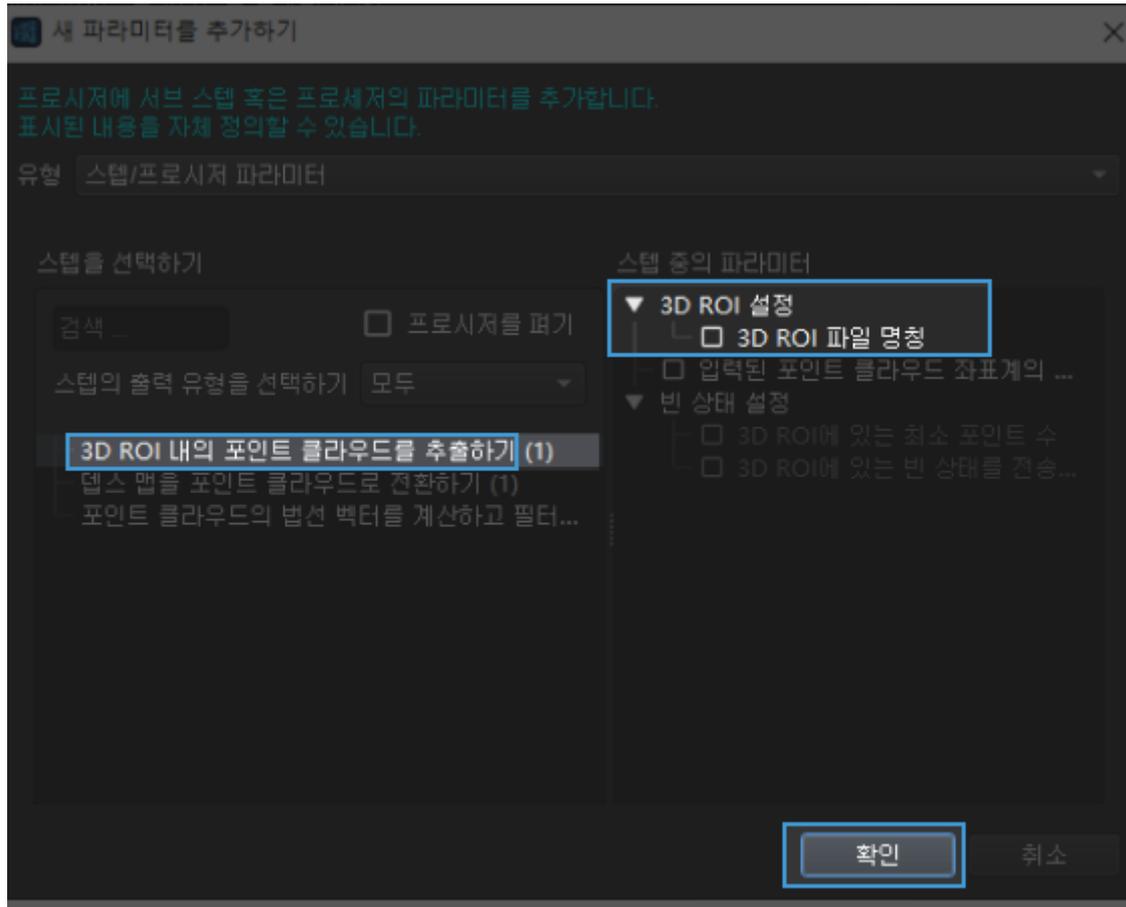
[응용]을 클릭하면 프로시저의 스텝 파라미터에서 해당 파라미터가 삭제된 것을 확인할 수 있습니다.

## 추가하기

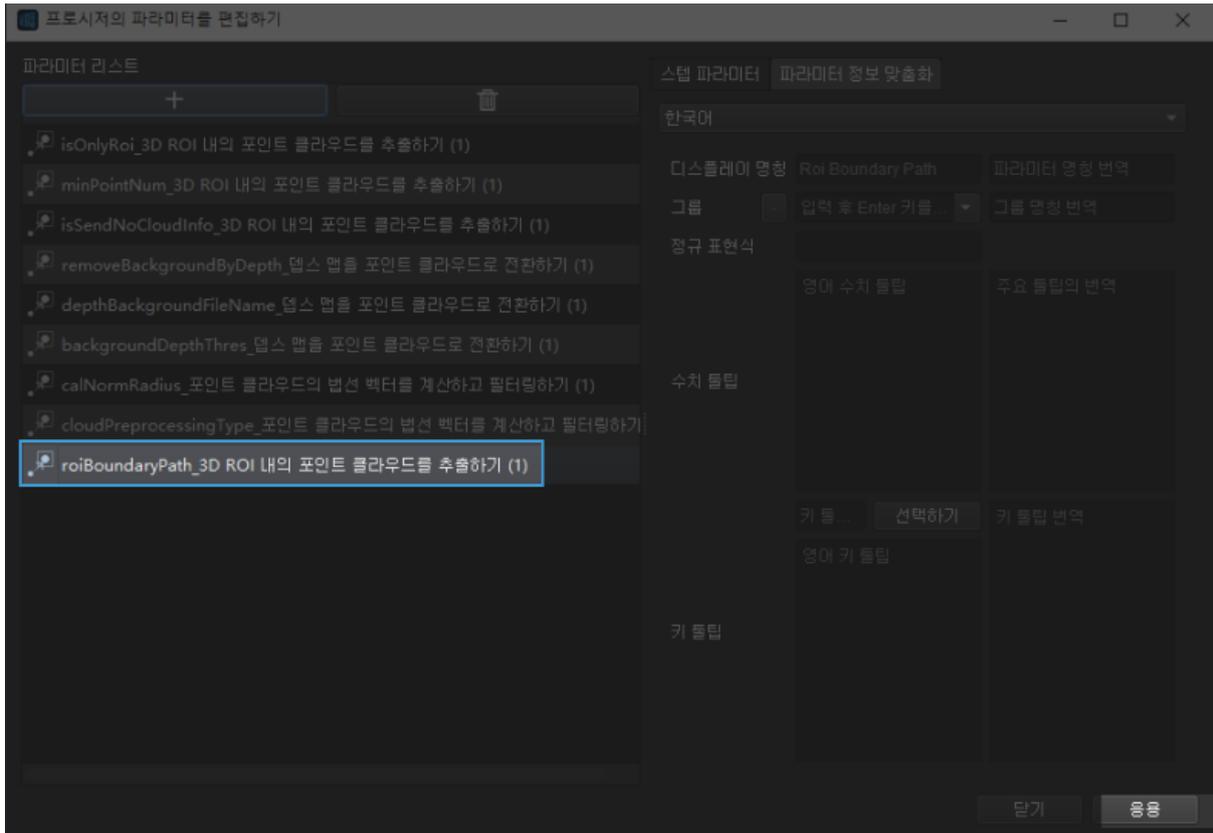
프로시저의 파라미터를 편집하기 팝업창이 나타나면 [ + ]를 클릭하여 새 파라미터를 추가하기 창으로 들어갑니다.



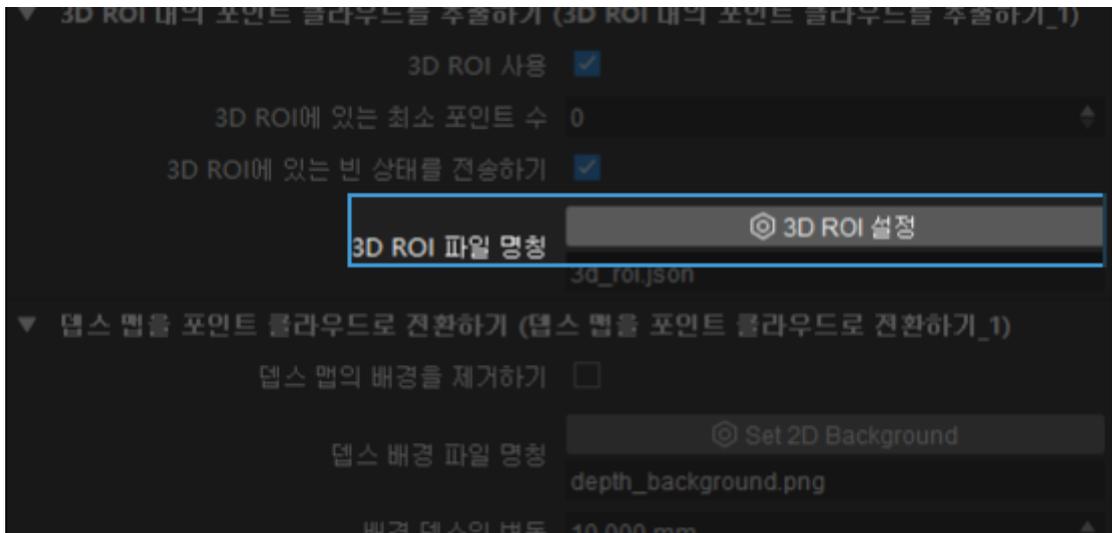
아래 그림과 같이 프로시저에서 스텝의 파라미터를 선택하고 "3D ROI 내의 포인트 클라우드를 추출하기" 스텝의 **3D ROI** 파라미터를 선택하고 [ OK ]를 클릭하여 설정을 저장합니다.



이때 파라미터 리스트에서 이전 단계에서 선택한 파라미터를 볼 수 있으며 [ 응용 ]을 클릭하여 작업을 적용합니다.



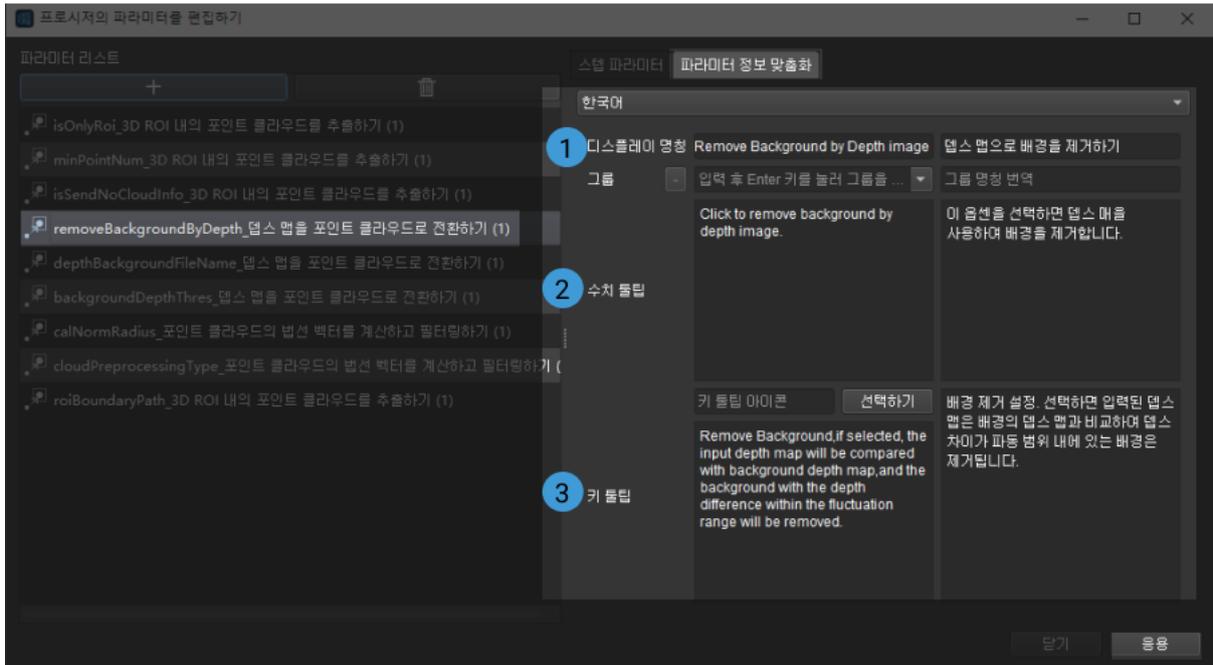
프로시저의 **스텝 파라미터**에서 새로 추가된 파라미터를 볼 수 있습니다.



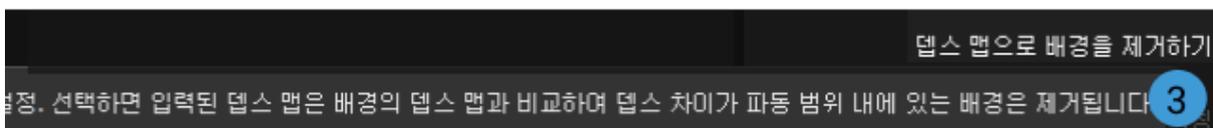
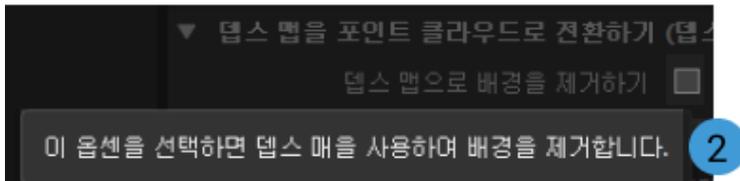
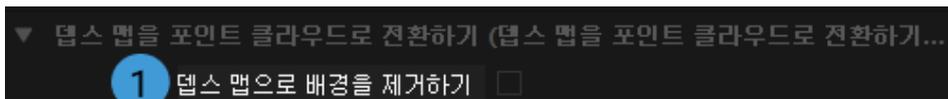
#### 파라미터 표시 정보 자체 정의

위와 같이 **프로시저의 파라미터를 편집하기** 창으로 진입하고 수정할 파라미터를 마우스 왼쪽 버튼으로 클릭하여 **파라미터 정보 사용자 정의** 페이지로 진입합니다.

디스플레이 명칭, 수치 툴팁, 키 툴팁의 내용을 아래 그림과 같이 수정한 후 [응용]을 클릭합니다.



다음 그림은 수정된 파라미터 표시 정보를 보여줍니다.



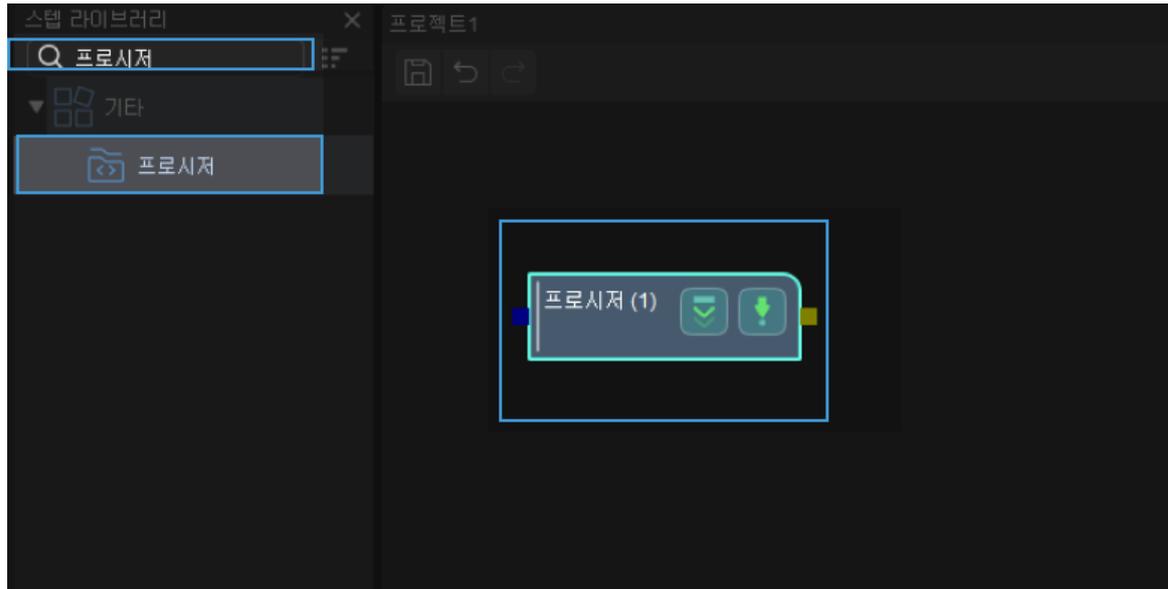
#### 4.2.5.9. 프로시저 자체 정의

소프트웨어에 내장되어 있는 일반적인 프로시저 이외에 사용자가 원하는 프로시저를 자체적으로 정의할 수 있습니다.

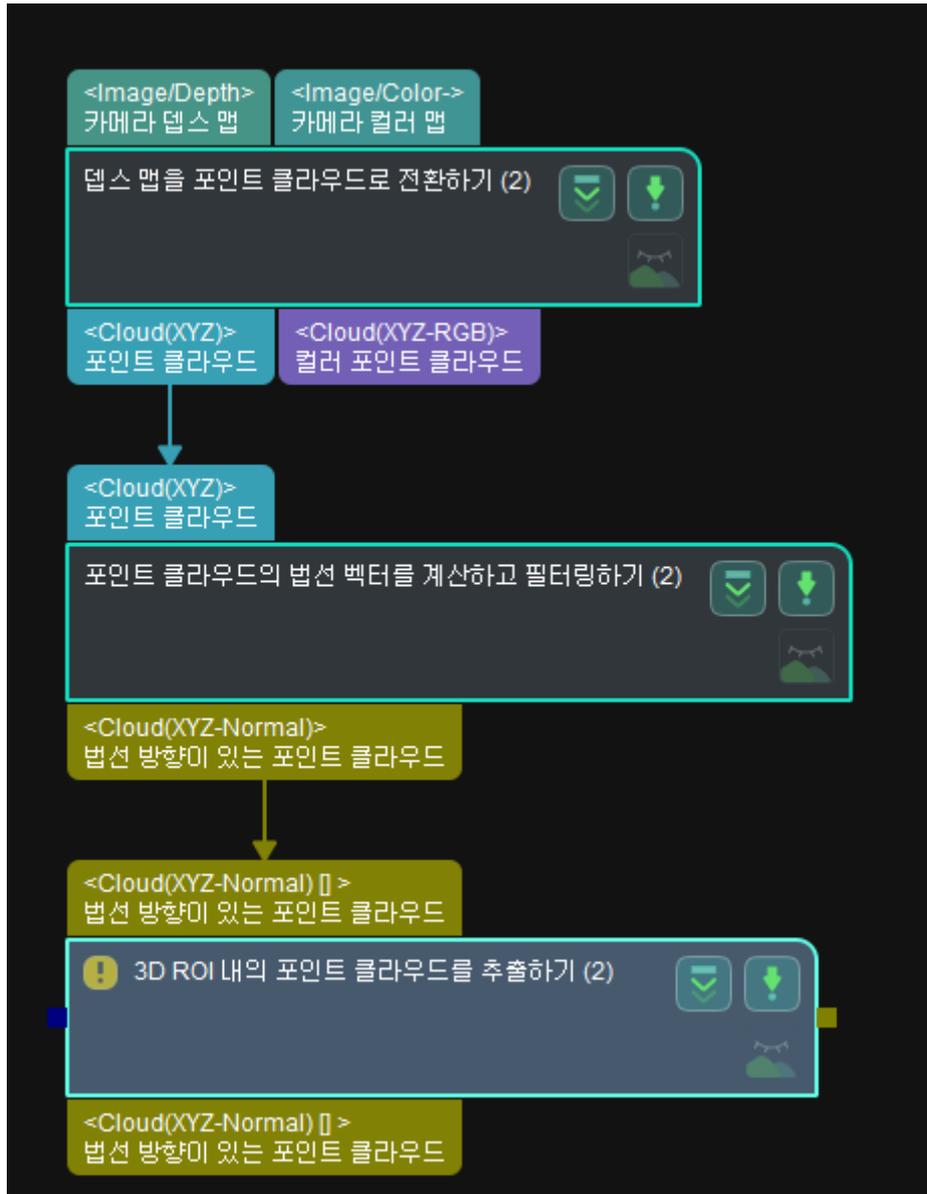
이 부분에서는 포인트 클라우드 처리를 캡슐화하는 프로시저를 예로 들어 설명합니다. 사용되는 스텝은 덱스 맵을 포인트 클라우드로 전환하기, 포인트 클라우드의 법선 벡터를 계산하고 필터링하기, 3D ROI 내의 포인트 클라우드를 추출하기입니다. 위의 스텝을 캡슐화하려면 "프로시저"를 사용해야 합니다.

##### 스텝 추가

1. 스텝 라이브러리에서 **프로시저**를 검색하여 프로젝트 편집 구역으로 드래그합니다. 다음 그림과 같습니다.



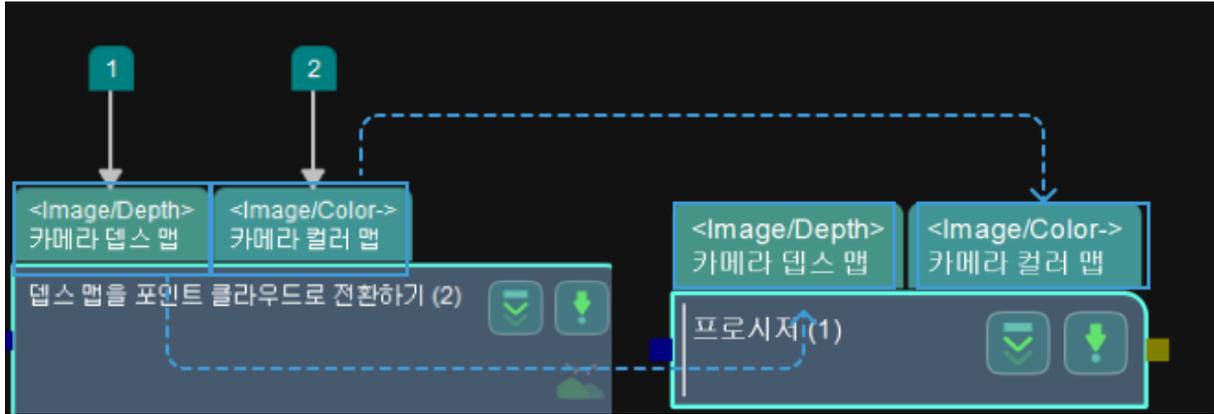
2. **프로시저**를 클릭하여 들어가 아래 그림을 참조하여 스텝을 추가하고 연결합니다.



#### “프로시저”의 입력 및 출력을 설정하기

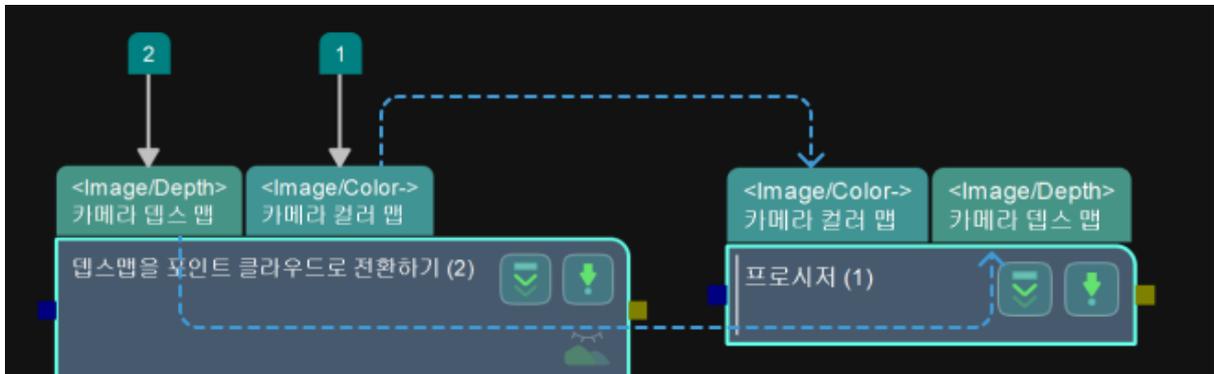
- 입력:

뎀스 맵을 포인트 클라우드로 전환하기의 입력 포트를 마우스 왼쪽 버튼으로 더블 클릭하면 번호가 매겨진 라인이 있는 포트가 프로시저의 입력 포트가 됩니다.



포트를 더블 클릭하는 순서에 따라 프로시저의 배열 순서가 결정됩니다.

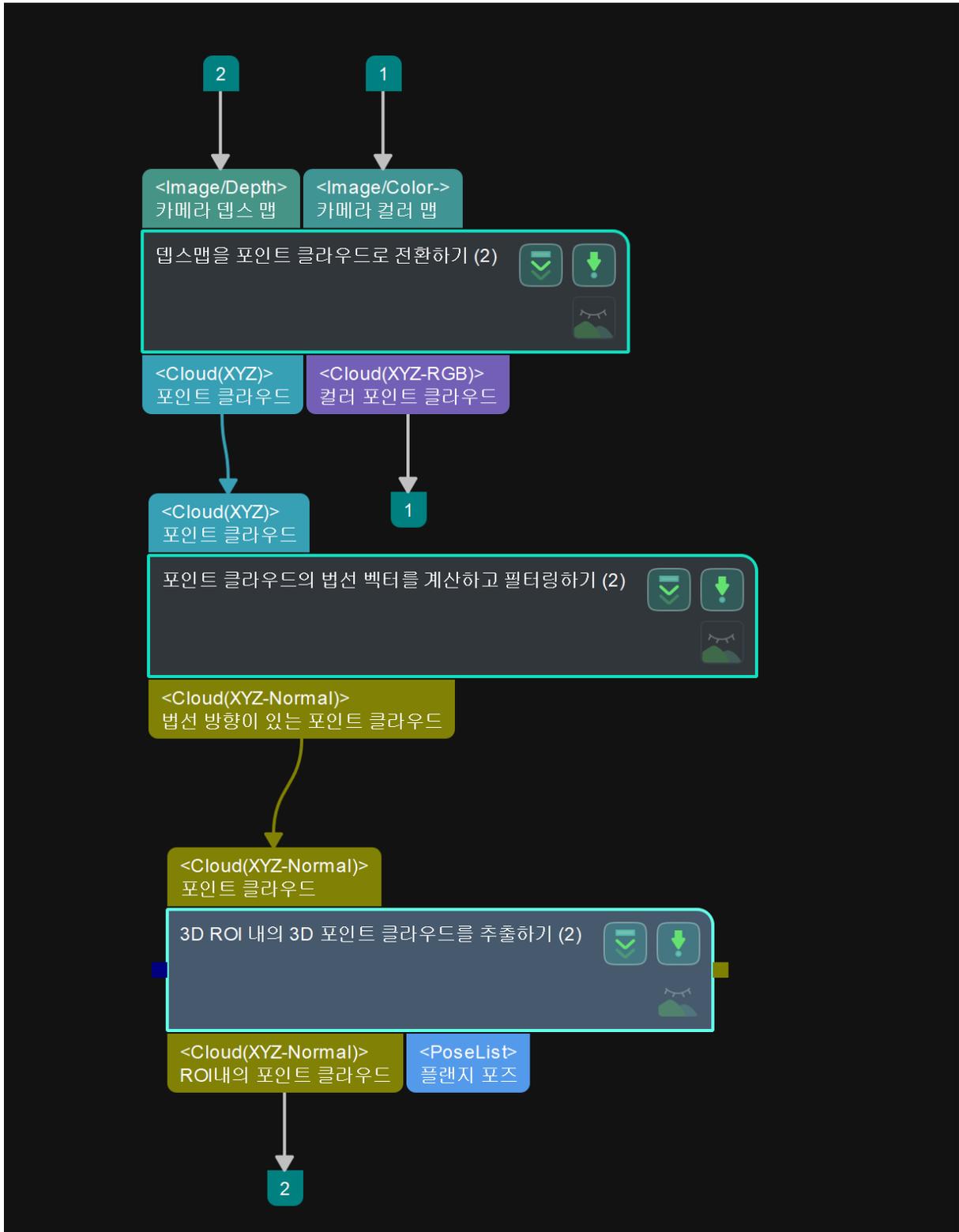
위의 그림은 **카메라 뎀스 맵** 포트를 먼저 더블 클릭한 후 **카메라 컬러 맵** 포트를 더블 클릭한 모습이고, 아래 그림은 **카메라 컬러 맵** 포트를 먼저 더블 클릭한 다음 **카메라 뎀스 맵** 포트를 두 번 클릭한 모습입니다.



마우스 왼쪽 버튼으로 클릭해 포트 위의 번호를 선택하여 **Delete** 키를 누르거나 마우스 오른쪽 버튼 메뉴에서 **[삭제]**를 클릭하여 포트를 삭제합니다.

- 출력:

출력 포트의 설정 방법은 입력 포트의 설정 방법과 동일합니다.



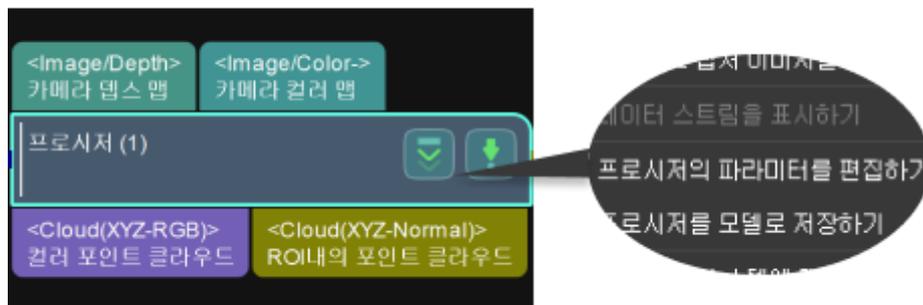
### 프로시저의 파라미터 설정

프로시저가 생성된 후에는 프로시저 파라미터의 명칭만 변경할 수 있습니다. 다음 그림과 같습니다.

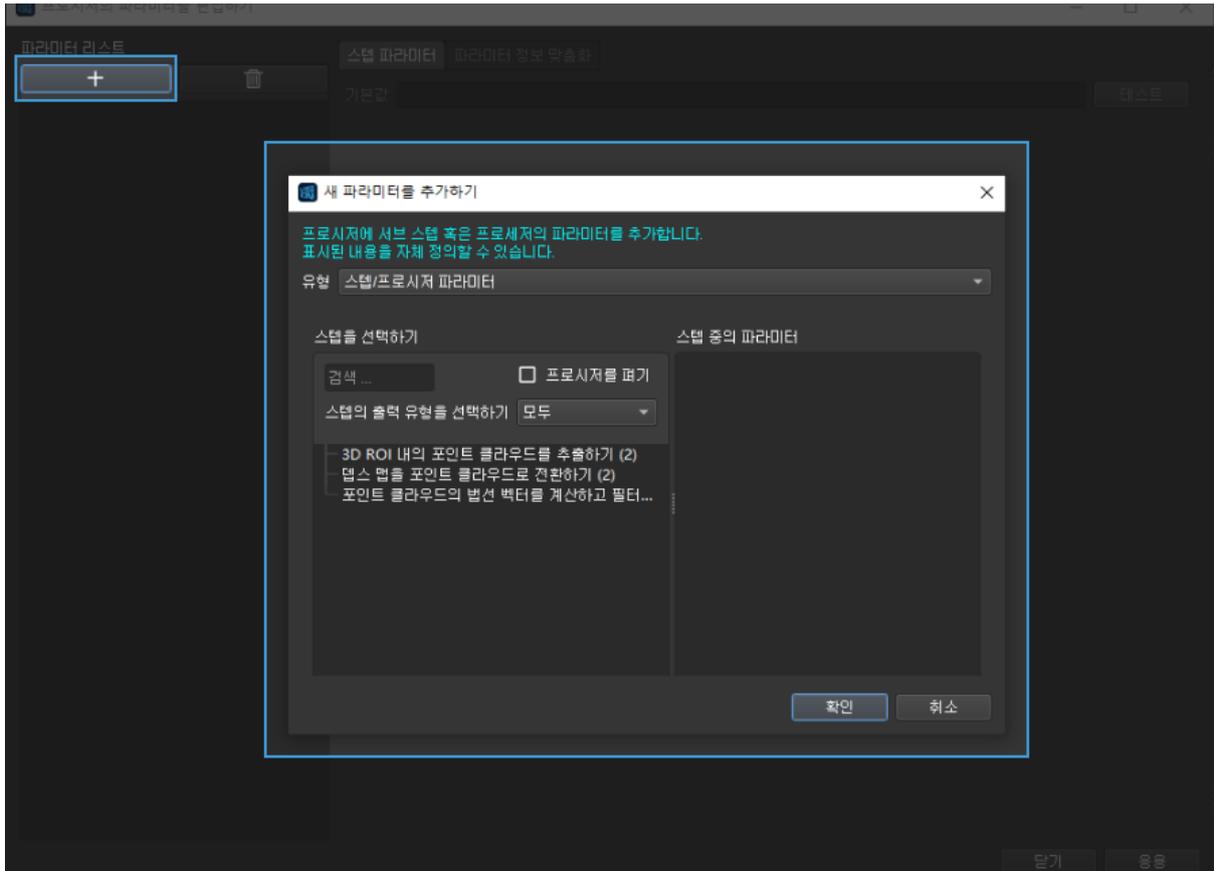


프로시저의 파라미터를 편리하게 조정하기 위해 프로시저의 각 스텝의 주요 파라미터를 프로시저의 파라미터로 설정할 수 있습니다.

1. 프로시저를 마우스 오른쪽 버튼으로 클릭하고 팝업한 메뉴 바에서 [ 프로시저의 파라미터를 편집하기 ]를 클릭합니다.



2. 프로시저의 파라미터를 편집하기 팝업창이 나타나면 [ + ]를 클릭하여 새 파라미터를 추가하기 창으로 들어갑니다.

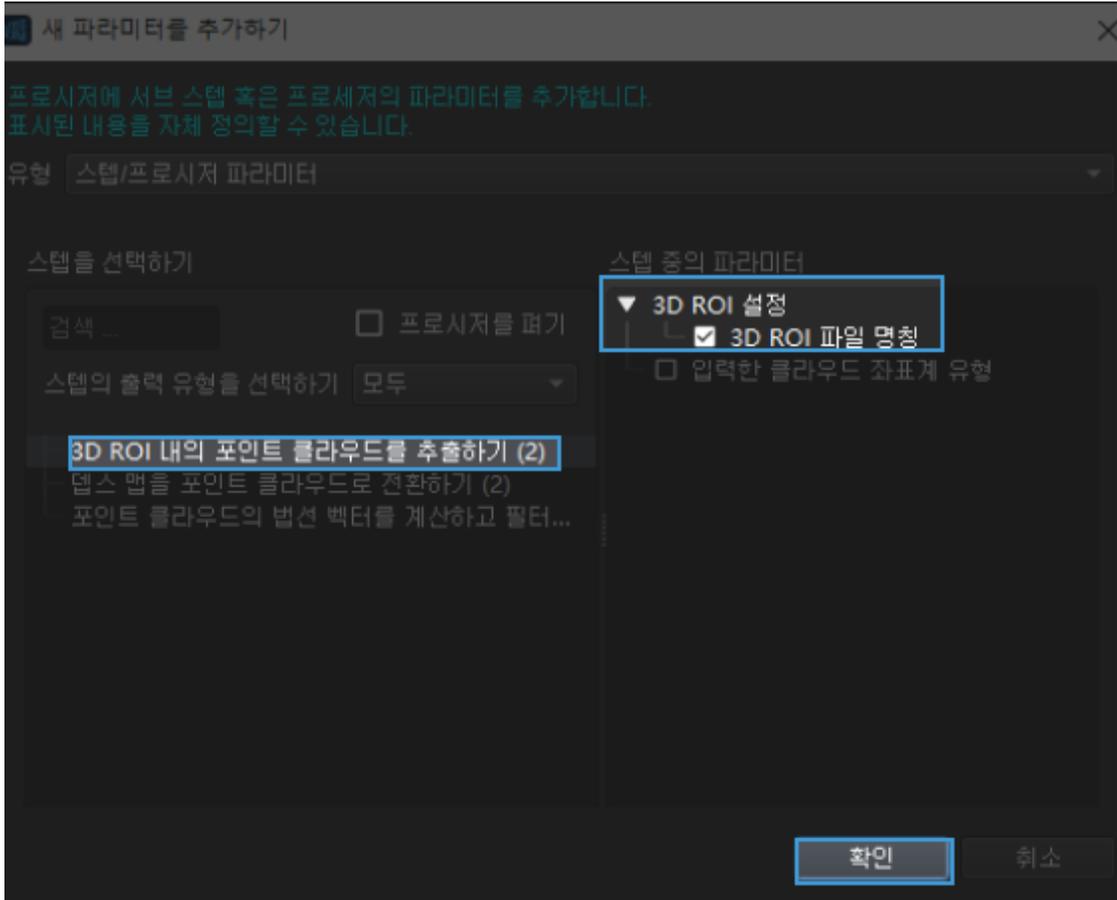


프로시저의 파라미터 설정에는 세 가지 유형이 있습니다.

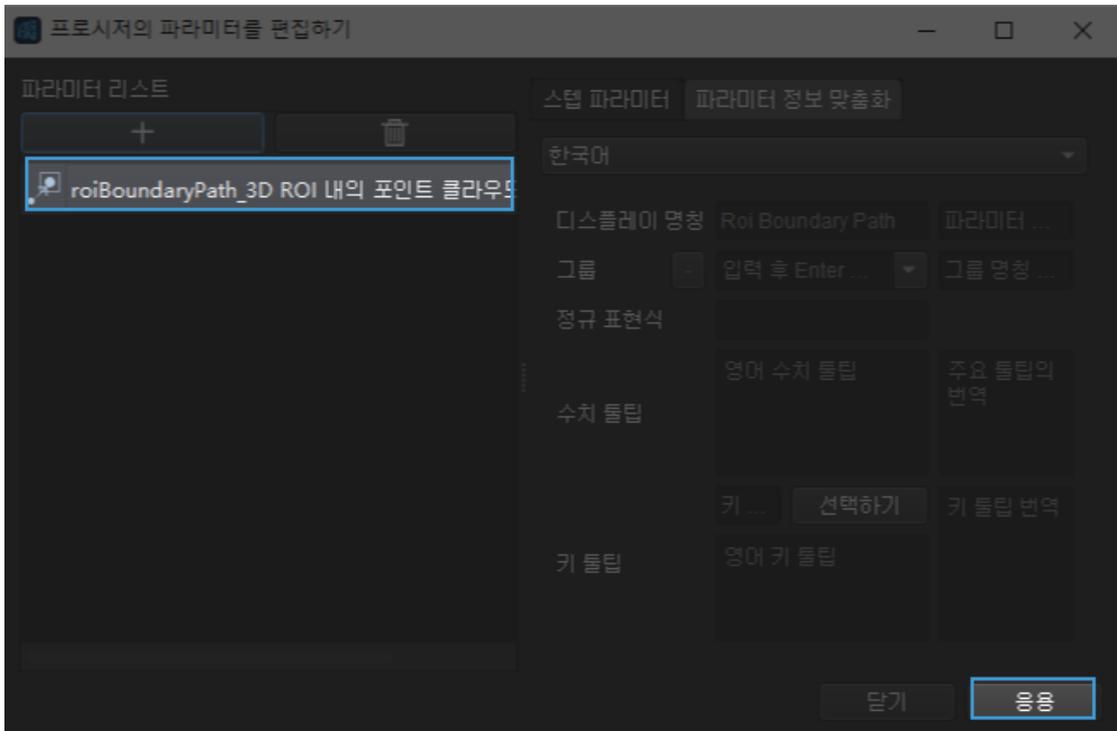
- 스텝/프로시저의 파라미터(이 부분은 이런 유형을 예시로 사용함)
- 매핑 파라미터 자체 정의
- 레시피 파라미터 자체 정의
  1. 아래 그림과 같이 프로시저에서 스텝의 파라미터를 선택하고 **3D ROI 내의 포인트 클라우드를 추출하기**의 **3D ROI** 파라미터를 선택하고 [OK]를 클릭하여 설정을 저장합니다.



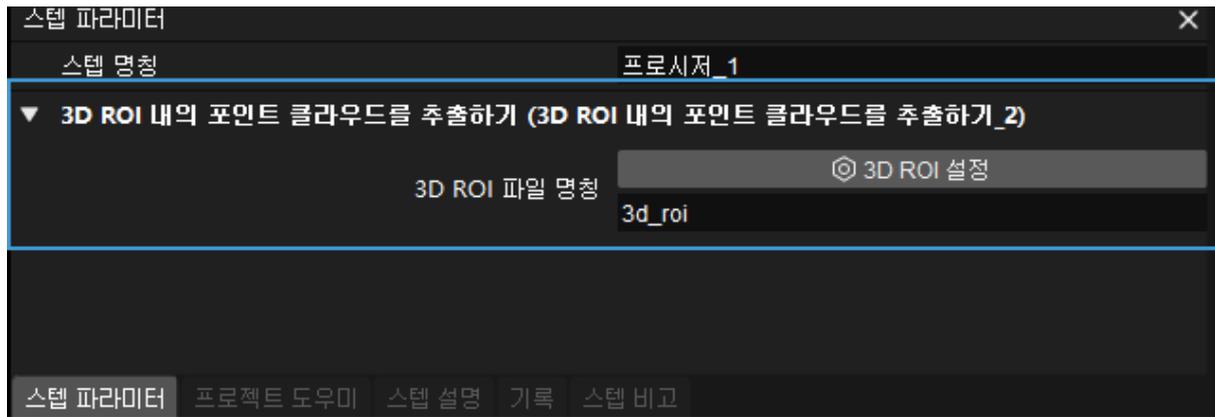
사용자는 실제 상황에 따라 여러 파라미터를 선택할 수 있습니다.



- 이때 파라미터 리스트에서 이전 단계에서 선택한 파라미터를 볼 수 있으며 [응용]을 클릭하여 작업을 적용합니다.



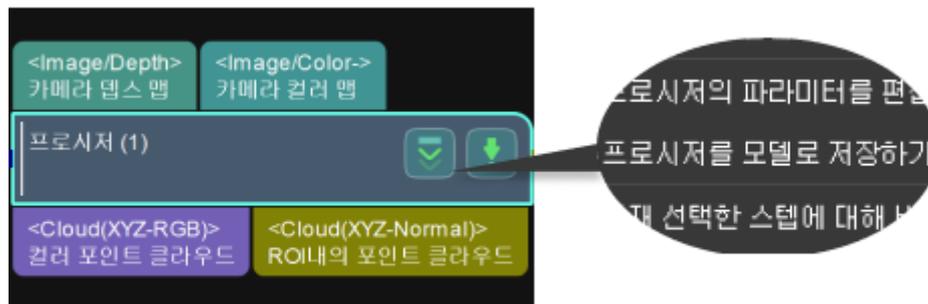
- 프로시저의 **스텝 파라미터**에서 파라미터가 성공적으로 설정되었음을 확인할 수 있습니다.



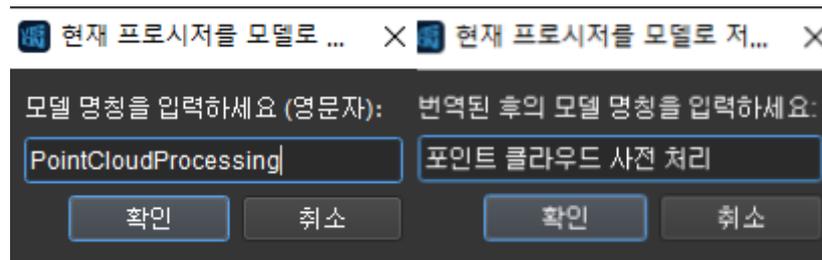
### 프로시저 저장

설정된 프로시저는 스텝 라이브러리의 자체 정의 목록에 추가할 수 있습니다.

1. 스텝을 마우스 오른쪽 버튼으로 클릭하고 팝업한 메뉴 바에서 [ **프로시저를 모델로 저장하기** ] (개발자 모드에만 이 버튼을 사용할 수 있음)를 클릭합니다.

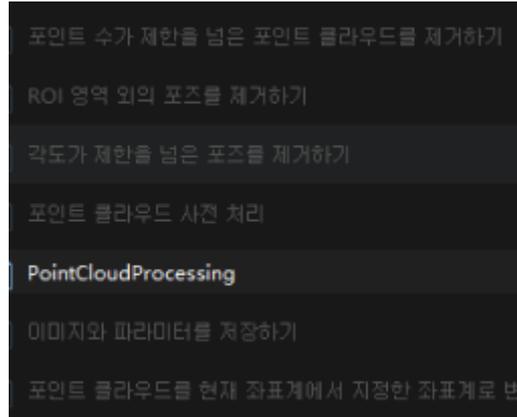


2. 소프트웨어에서 지시하는 대로 모델 명칭을 입력합니다.

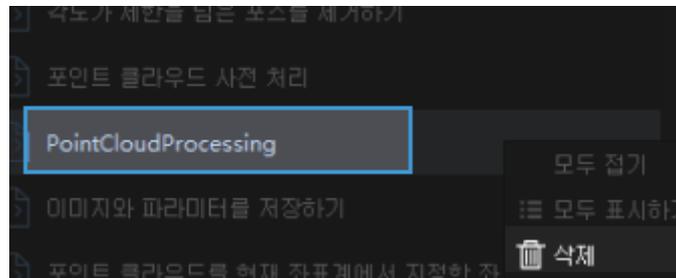


 모델 명칭은 스텝 라이브러리에 이미 있는 스텝 이름을 쓰지 마십시오.

3. 입력 후 [ **확인** ]을 클릭하여 저장하면 스텝 라이브러리의 사용자 정의 디렉터리에서 새로 추가된 프로시저를 찾을 수 있습니다.



사용자 자체 정의 프로시저를 마우스 왼쪽 버튼으로 클릭하고 **Delete** 키를 누르거나 사용자 자체 정의 프로시저를 마우스 오른쪽 버튼으로 클릭하고 팝업한 메뉴 바에서 [ **Delete** ]를 클릭하여 추가된 사용자 자체 정의를 삭제할 수 있습니다.



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

## 4.3. 스텝 사용 가이드

이 부분은 Mech-Vision 스텝과 프로시저에 관한 설명입니다.

### 스텝 사용 가이드

- 2D 매칭
- 3D 근사 매칭(멀티 모델)
- 모든 파라미터를 받아들이기
- 포즈에 레이블을 추가하기
- 카메라의 왜곡으로 인한 정확하지 않는 포즈를 조정하기
- 옵션에 따라 포즈를 조정하기
- 기울임에 따라 포즈를 조정하기
- 포즈를 포인트 클라우드의 표면으로 조정하기
- 입력 포트 할당자
- 마스크안에 대응하는 포인트 클라우드를 추출하기
- 마스크안에 대응하는 포인트 클라우드를 추출하기

- 부울 값 리스트 논리 연산
- 절대값을 계산하기
- 마스크 면적을 계산하기
- 두 포즈가 지정된 방향에 따른 거리를 계산하기
- 두 포즈의 지정된 축 사이의 각도를 계산하기
- 포인트 클라우드의 법선 벡터를 계산하고 필터링하기
- 평면 포인트 클라우드의 포즈와 크기를 계산하기
- 직사각형의 사이즈와 포즈를 계산하기
- 카메라에서 이미지를 캡처하기
- 포인트 클라우드 처리(GPU)
- 3D ROI 내의 포즈를 획득하기
- 수치 비교
- 입력된 각 포즈의 일부를 추출하여 새로운 포즈를 생성하기
- 데이터 리스트에서 지정된 차원의 요소 개수를 통계하기
- 물체 치수 분해
- 포즈를 사원수와 평행 이동 벡터로 분해하기
- 라인 세그먼트 감지
- 역치에 근거한 이진 분류
- 포인트 클라우드 다운 샘플링
- 레이블 리스트를 빠르게 만들기
- 3D 방법을 사용하여 포인트 클라우드의 에지를 추정하기
- 직육면체 내의 포인트 클라우드를 추출하기
- 원기둥 범위 내의 포인트 클라우드를 추출하기
- 3D ROI 내의 포인트 클라우드를 추출하기
- 필터링
- 포즈 좌표축의 방향을 뒤집기
- 실제 치수를 픽셀 치수로 전환하기
- Cloud (XYZ-Normal)를 Cloud (XYZ-RGB)로 전환하기
- 덤스 맵을 포인트 클라우드로 전환하기
- PoseList를 PoseLists로 전환
- Variant를 VariantList로 전환하기

- 지정된 크기의 직사각형 에지 템플릿을 생성하기
- 첫 번째 이미지를 획득하기
- 가장 높은 층의 포인트 클라우드를 획득하기
- 가장 높은 층의 포즈를 획득하기
- 포즈의 역계산
- 레이블 매핑
- 여러 개의 픽 포인트에 매핑하기
- 데이터를 합치기
- 포인트 클라우드를 합칩기
- 형태학적 변환
- 법선 방향을 계산하기
- 수치 확대 및 축소
- 수치 계산
- 포즈의 축을 지정된 방향으로 가리키도록 회전하기
- 포인트 클라우드 클러스터링
- 포인트 클라우드를 필터링하기
- 포인트 클라우드 모양 감지기
- 포인트를 필터링하기
- 포즈 필터링
- 출력
- 3D 포인트 클라우드를 2D 이미지로 투영하기
- 물체 사이즈 읽기
- 포인트 클라우드를 읽기
- 파일에서 포즈 정보 획득
- 2D ROI 내의 조정된 이미지를 복원하기
- 겹친 물체를 제거하기
- 겹치는 다각형을 제거하기
- 마스크 외부에 있는 다각형을 제거하기
- 데이터 반복 및 연결
- 지정축과 기준 방향의 각도가 최소가 되도록 포즈를 조정하기
- 지정한 축과 각도에 근거하여 포즈를 회전하기

- 이미지를 저장하기
- 결과를 파일로 저장하기
- 스텝 파라미터를 파일로 저장하기
- 2D ROI 내 이미지의 크기를 조정하기
- 포인트 클라우드를 외부 서비스로 보내기
- 포즈의 사원수를 설정하기
- 포즈의 평행 이동의 벡터를 설정하기
- 이미지를 표시하기
- 포인트 클라우드 및 포즈를 표시하기
- 포인트 클라우드의 순서를 배열하기
- 역치에 근거하여 수치를 분류하기
- 실수 빨셈
- 일부 영역에서 점수가 낮은 포즈를 제거하기(NMS)
- 포인트 클라우드 변환
- 포즈 변환
- 지정한 방향으로 포즈를 이동하기
- 부울 값에 의해 제어 흐름을 촉발하기
- 출력 수 제한
- 포즈의 개수를 제한하기
- 포즈가 3D ROI에 있는지 확인하기
- 필요한 레이블인지 판단하기
- 포인트 클라우드가 요구에 부합한지 검증하기
- 기준 방향과의 각도에 근거하여 포즈 유효성을 판단하기

## 자주 사용하는 프로시저 (작성 중)

### 4.3.1. 2D 매칭

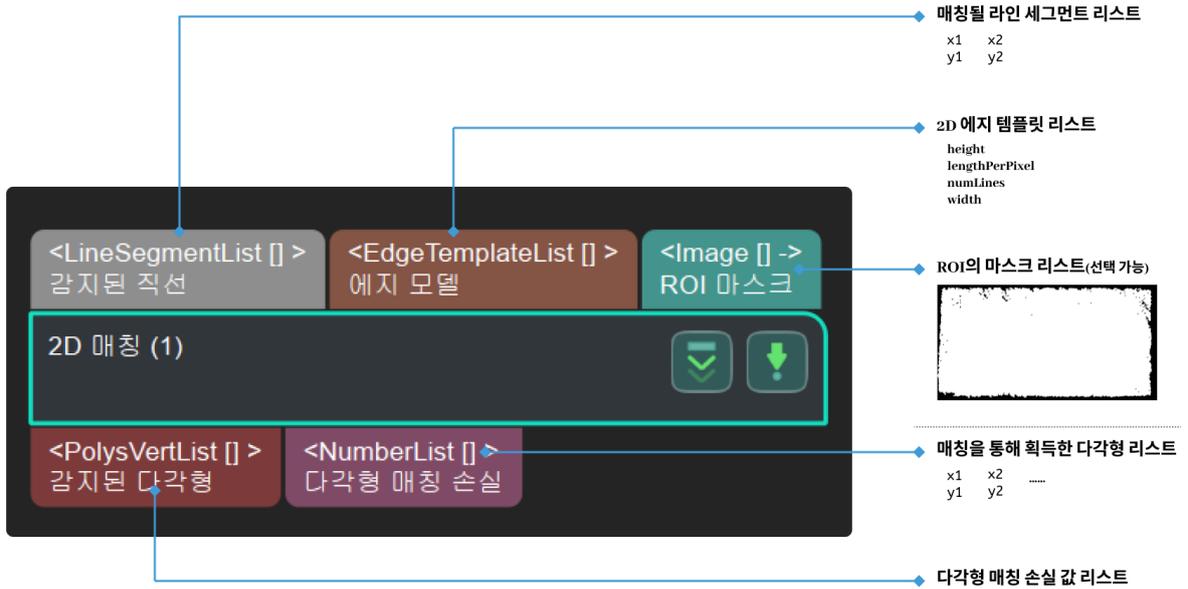
#### 기능 설명

2D 매칭 알고리즘을 통해 지정된 모양에 대해 템플릿 매칭을 수행합니다.

#### 응용 시나리오

2D 이미지에서 다각형 물체의 위치를 지정하고 후속 다각형의 3D 포즈를 계산하는 데 감지된 다각형 특징을 제공합니다. 일반적으로 **마스크 외부에 있는 다각형을 제거하기** 스텝과 함께 사용됩니다.

## 입력 및 출력



## 파라미터 설명

### 모델 변형

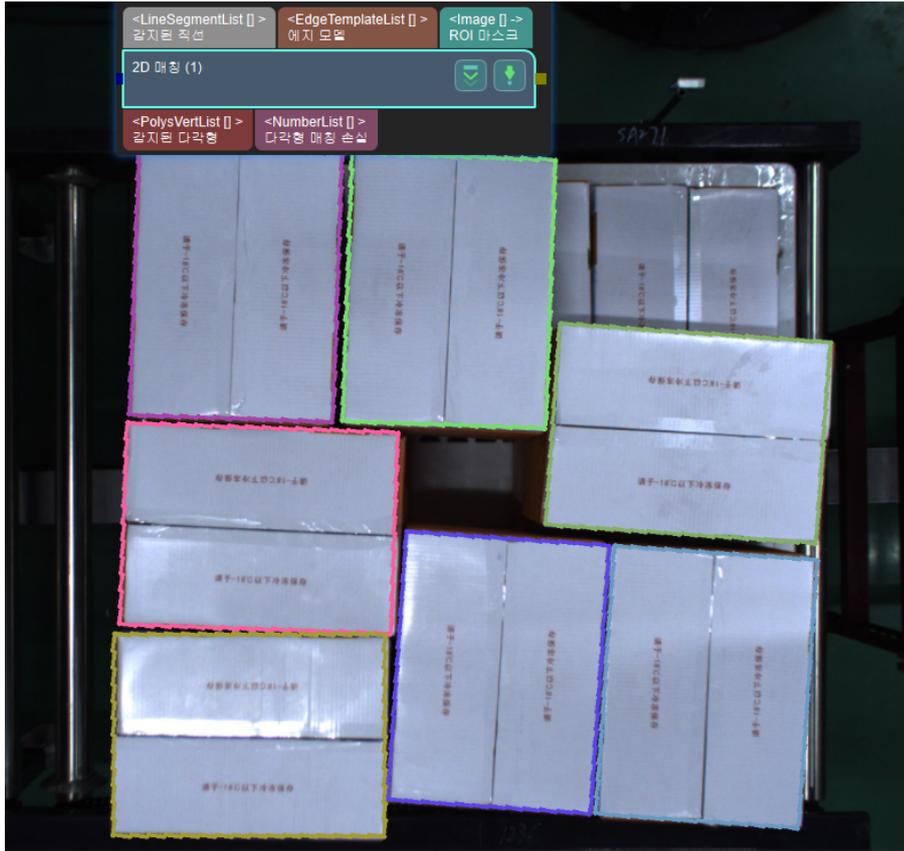
#### 윤곽선 각도 양자화(1~360)

기본값: 60

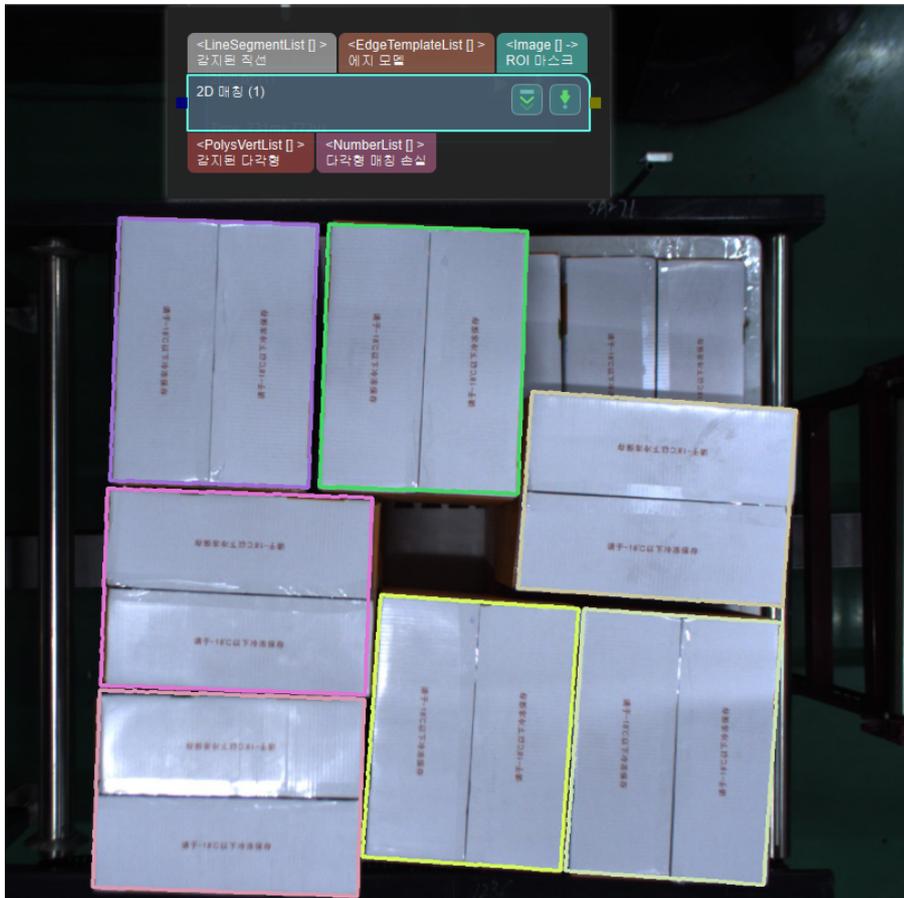
조절 설명: 윤곽선 방향을 계량화하고 분해합니다. 직사각형을 예로: 직사각형은 180도 대칭이기 때문에 180도를 계량화합니다. 입력할 파라미터는 계량화 부수입니다. 90으로 설정된다면  $180^\circ/90=2^\circ$ . 계량화 부수가 많을수록 테두리가 더욱 정교하게 보이지만 시간이 더 많이 걸립니다.

조절 예시:

파라미터를 ``30``으로 설정 시 스텝의 실행 시간과 계량화 결과는 아래 그림과 같습니다.



파라미터를 ``90``으로 설정 시 스텝의 실행 시간과 계량화 결과는 아래 그림과 같습니다.



## 각도 손실 가중치

기본값: 0.1

조절 설명: 이 수치가 높을수록 결과가 더욱 정확하지만 시간이 더 많이 걸립니다.

## 기초 스케일 비율

기본값: 1.1

조절 설명: 모델은 “기초 스케일 비율”, “비율 지수”를 통해 크기 조정을 합니다. “비율 지수”는 [“최소 스케일 지수”, “최대 스케일 지수”] 내의 정수입니다. 수치가 높을수록 정확하게 매칭하는 가능성이 더욱 크지만 시간이 더 많이 걸립니다.

## 최소 스케일 지수

기본값: 0

조절 설명: 참고치: -3, -2, -1, 0. 수치가 낮을수록 정확하게 매칭하는 가능성이 더욱 크지만 시간이 더 많이 걸립니다.

## 최대 스케일 지수

기본값: 0

조절 설명: 참고치: 0, 1, 2, 3. 수치가 높을수록 정확하게 매칭하는 가능성이 더욱 크지만 시간이 더 많이 걸립니다.

## 기본 종횡비

기본값: 1.1

조절 설명: 모델은 “기초 종횡비”, “종횡비 지수”를 통해 크기 조정을 합니다. “종횡비 지수”는 [“최소 종횡비”, “최대 종횡비”] 내의 정수입니다. 수치가 높을수록 정확하게 매칭하는 가능성이 더욱 크지만 시간이 더 많이 걸립니다.

## 최소 종횡비 지수

기본값: 0

조절 설명: 참고치: -3, -2, -1, 0. 수치가 낮을수록 정확하게 매칭하는 가능성이 더욱 크지만 시간이 더 많이 걸립니다.

## 최대 종횡비 지수

기본값: 0

조절 설명: 참고치: 0, 1, 2, 3. 수치가 높을수록 정확하게 매칭하는 가능성이 더욱 크지만 시간이 더 많이 걸립니다.

## 후보 검색 설정

### 물체의 대략적인 포즈(0~360)

기본 값: 0°

조절 설명: 시나리오에 있는 물체의 대략적인 방향을 뜻합니다. 단위는 도(°)입니다. 직사각형의 경우 수평 방향을 0° 방향으로, 수직 방향을 90° 방향으로 정의합니다.

### 물체 포즈 각도의 변화 범위(0~180)

기본 값: 90°

조절 설명: 시나리오에서 물체 포즈가 변화하는 범위를 뜻합니다. 단위는 도(°)입니다.

### 템플릿의 평행이동 보폭

기본값: 5.000 mm

조절 설명: 이 수치가 낮을수록 결과가 더욱 정확하지만 시간이 더 많이 걸립니다. 단위: 밀리미터(mm).

### 템플릿의 회전 이동 보폭(0~180)

기본값: 1.5000°

조절 설명: 이 수치가 낮을수록 결과가 더욱 정확하지만 시간이 더 많이 걸립니다. 단위는 도(°)입니다.

### 역치 설정

#### 평균 거리 오차에 대한 상한 역치

기본값: 8.000 mm

조절 설명: 매칭 템플릿의 최대 평균 거리 오차입니다. 단위: 밀리미터(mm). 수치가 높을수록 결과가 더욱 정확합니다.

조절 예시: 아래 그림과 같이 왼쪽은 기본 상태에서 출력한 결과이고 오른쪽은 파라미터가 0.002 시 출력한 결과입니다.



#### 겹치는 비율의 역치(0~1.0)

기본값: 0.5

조절 설명: 결과 중 겹치는 비율이 이 수치보다 높으면 삭제될 것입니다. 이 수치가 높을수록 더 많은 겹치는 결과를 출력할 것입니다.

조절 예시: 아래 그림과 같이 왼쪽은 파라미터가 0.2 시 출력한 결과이고 오른쪽은 파라미터가 0.5 시 출력한 결과입니다.



### 4.3.2. 3D 근사 매칭(멀티 모델)

#### 기능 설명

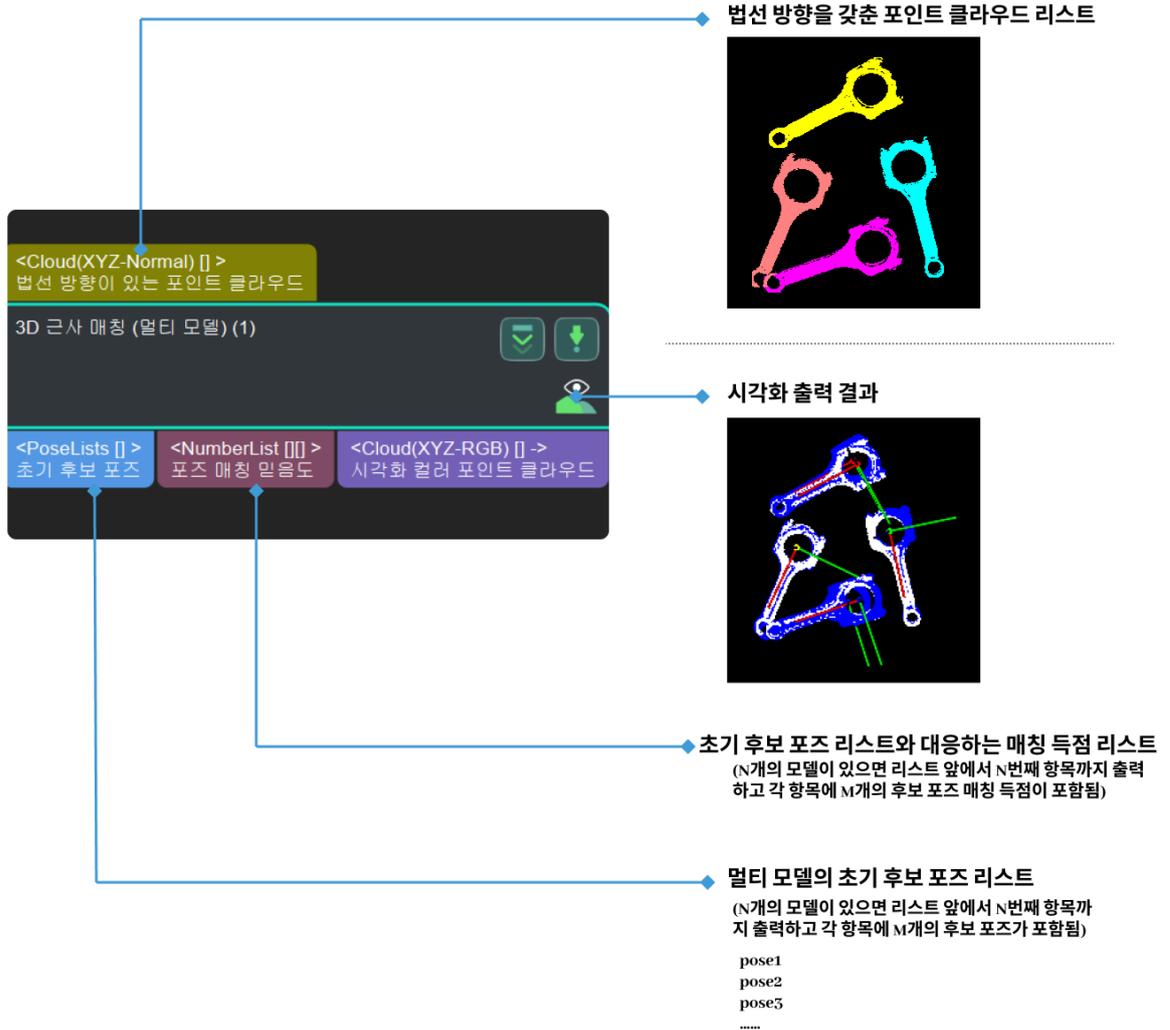
멀티 모델을 사용하여 시나리오에 있는 물체에 대해 근사 매칭을 진행하여 대상 물체의 초기 후보 포즈를 출력합니다.

#### 응용 시나리오

이 스텝은 멀티 모델을 사용하여 시나리오에 있는 초기 물체 포즈에 대해 계산합니다. 3D 근사 매칭 스텝의 확장 버전이고 파라미터 조절 방법도 이와 거의 일치합니다.

이 스텝은 멀티 모델 시나리오에 사용할 수 있고 또 다양한 유형의 작업물을 인식하거나 분류하는 데 사용됩니다. 매칭을 통해 더 정확한 포즈를 획득하기 위해 일반적으로 뒤에 3D 상세 매칭(멀티 모델) 스텝을 사용합니다.

#### 입력 및 출력



## 파라미터 설명

### 모델 및 픽 포인트 설정

모델 파일 및 픽 포인트 파일 경로 설정.

### 모델 파일(필수)

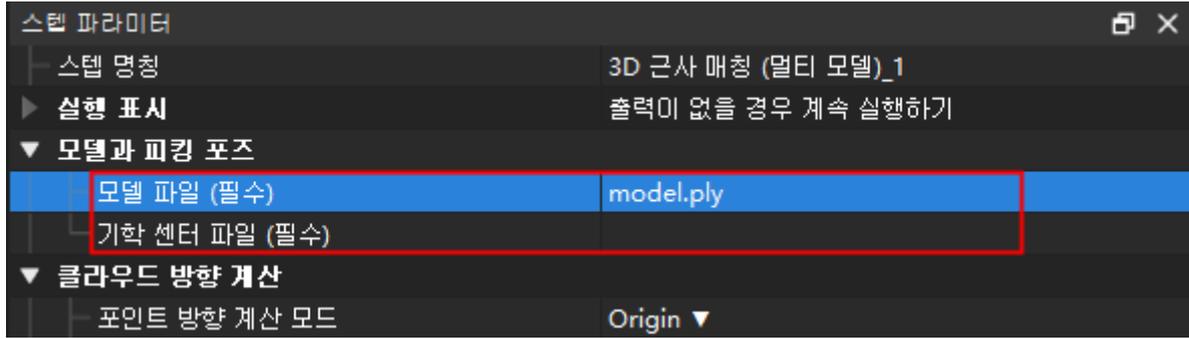
기본값: model.ply

설명: ply 형식의 모델 포인트 클라우드. 여러 파일 경로를 선택할 수 있으며 파일 사이는 쌍반점으로 구분합니다.

### 기하학적 중심 파일(필수)

설명: json 형식의 기하학적 중심 파일. 여러 파일 경로를 선택할 수 있으며 파일 사이는 쌍반점으로 구분합니다.

설명: 각 파라미터 아래에 입력된 파일의 순서가 일치한지 확인하십시오. 즉, **모델 파일**과 **기하학적 중심**의 파일 경로가 같은지 확인해야 합니다. 아래 그림과 같이 다른 파일은 `` ; `` 로 간격을 표시해야 합니다.



### 포인트 클라우드 중 포인트가 위치하는 평면의 법선 방향의 계산

#### 포인트가 위치한 평면의 법선 방향의 계산 방법

기본값: Origin

값 리스트	조절 설명
Origin	입력한 포인트 클라우드의 원시 법선 방향을 직접 사용합니다.
StandardMode	CPU를 사용하여 입력된 포인트 클라우드의 법선 방향을 재계산 하는 것은 모델에 법선 방향이 없는 경우에 사용하는 것을 권장합니다. 목표 포인트 근처에서 목표 포인트와 가장 가까운 k 개 포인트를 검색하고 PCA(주성분 분석 법) 를 사용해 최소 특징 벡터를 구하여 이 포인트의 법선 방향으로 간주합니다.
EdgeTangent	포인트의 접선 방향을 법선 방향으로 사용하여 입력한 에지 포인트 클라우드의 접선을 계산합니다. 외부 윤곽이 서로의 거울 이미지인 다른 물체를 구별할 수 있으며 평면 물체의 에지 포인트 클라우드를 매칭시킬 때 사용하는 것이 좋습니다.
EdgeNormal	입력한 에지 포인트 클라우드의 법선 방향을 계산하고, 점의 접선 방향을 법선 방향으로 사용하며 평면 물체의 에지 포인트 클라우드를 매칭할 때 사용하는 것을 권장합니다.



**EdgeTangent** 혹은 **EdgeNormal** 방법을 사용할 때 각 에지 포인트 클라우드에 여러 물체가 포함되어 있지 않은지, 즉 각 물체 포인트 클라우드가 분할되었는지 확인하십시오.

#### 선택된 인접점의 수

기본값: 10

설명: 이 파라미터는 포인트 방향의 인접점의 수, 즉 **StandardMode** 모드에서 k 값을 조정 계산하는데 사용됩니다.

#### 알고리즘 유형

기본값: SurfaceMatchingEasyMode

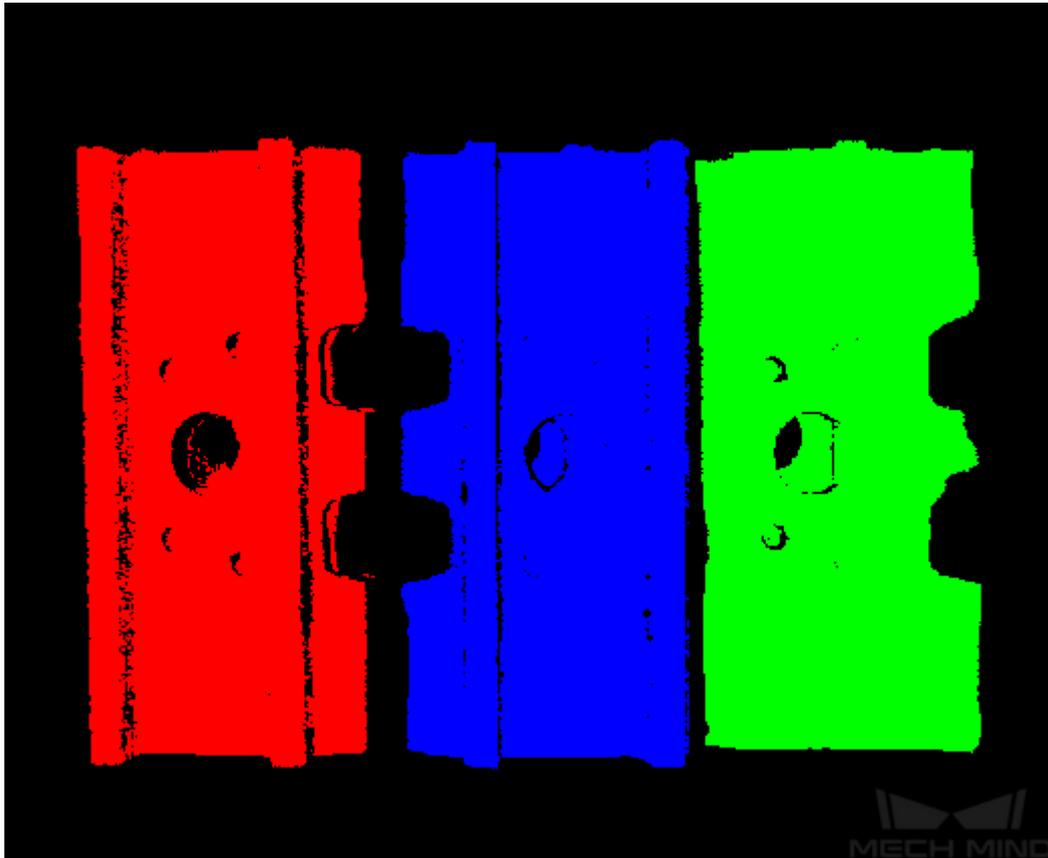
값 리스트: SurfaceMatchingEasyMode, SurfaceMatching

설명: 이 알고리즘은 두 가지 유형이 있습니다. 시각화 결과는 두 알고리즘이 다 조절할 수 있는 파라미터로

설정합니다. 알고리즘 유형의 파라미터 조절은 아래 그림의 입력한 포인트 클라우드를 예로 들어 정(正)반(反)모델의 각 매칭을 합니다. 먼저 **SurfaceMatchingEasyMode** 알고리즘에서 조절 가능한 파라미터를 소개합니다.

**SurfaceMatchingEasyMode** 알고리즘: 조절 가능한 파라미터 모듈은 속도 조절 파라미터 및 출력 설정입니다.

**SurfaceMatching** 알고리즘: 조절 가능한 파라미터 모듈은 샘플링 설정, 투표 설정, 클러스터링 설정 및 포즈 검증 설정입니다.



## SurfaceMatchingEasyMode 파라미터 설명

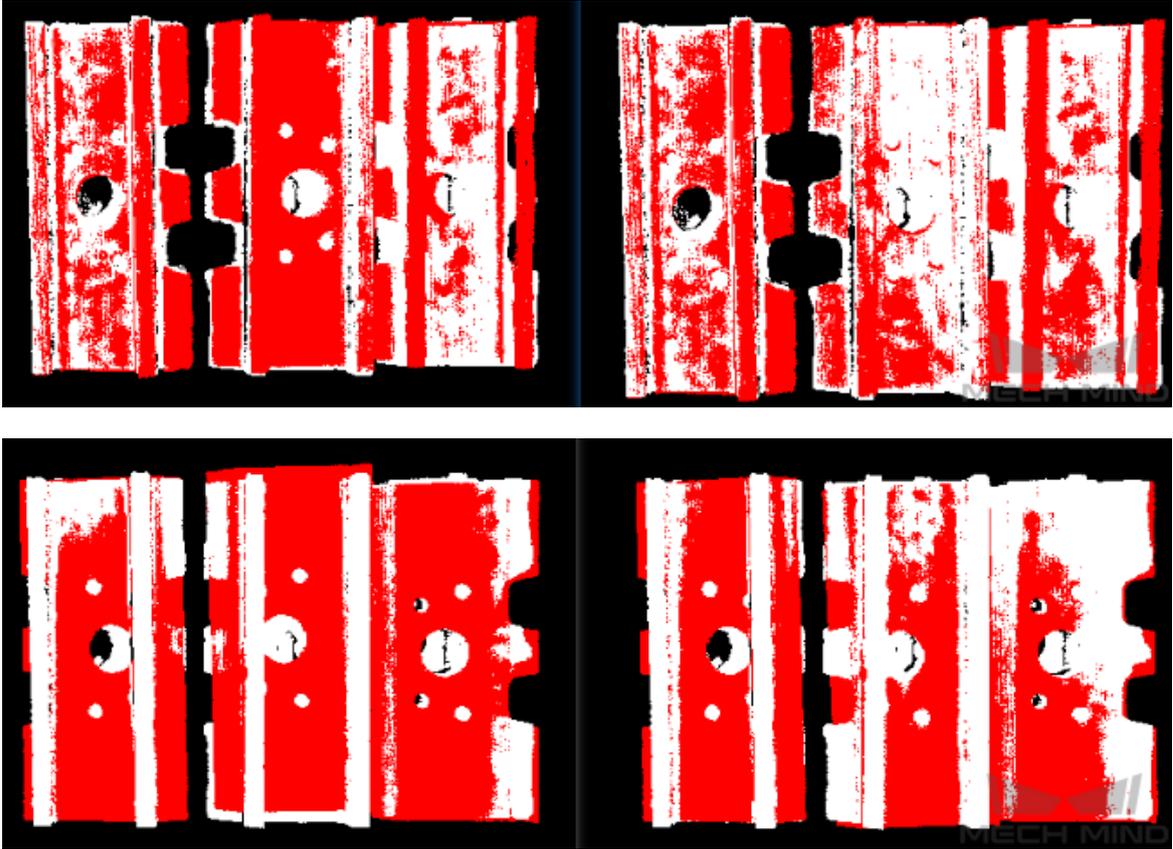
### 속도 조절 파라미터

#### 메인 속도 컨트롤러

기본값: 2

설명: 이 파라미터는 알고리즘 속도를 조절하는 데 사용됩니다. 값을 높이면 알고리즘 속도는 빨라지지만 매칭 정확도는 떨어집니다. 그 효과는 **보조 속도 컨트롤러** 보다 더 뚜렷합니다. 이 파라미터의 유효 범위는 1~6입니다.

조절 예시: 아래 그림과 같습니다. 왼쪽은 파라미터의 기본값이 2일 때의 결과이고 오른쪽은 이 파라미터를 6으로 조절한 결과입니다. 증가 후 매칭 정확도가 감소하는 것을 분명히 알 수 있습니다.

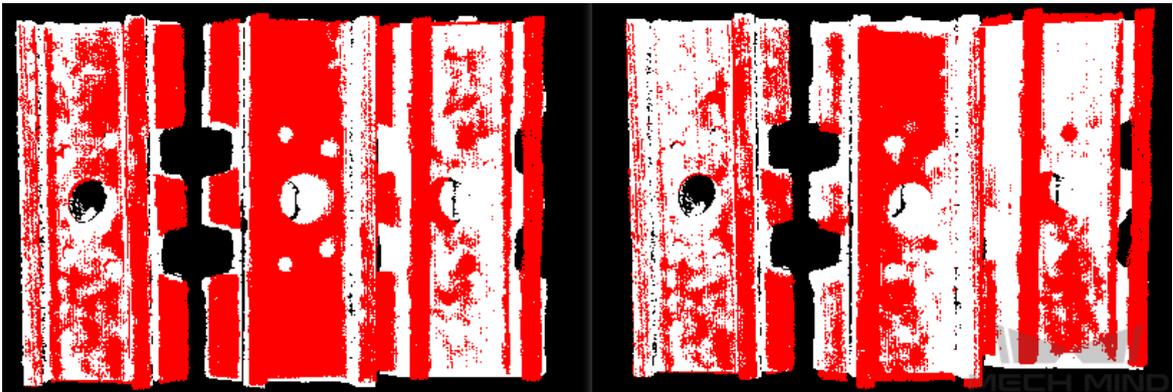


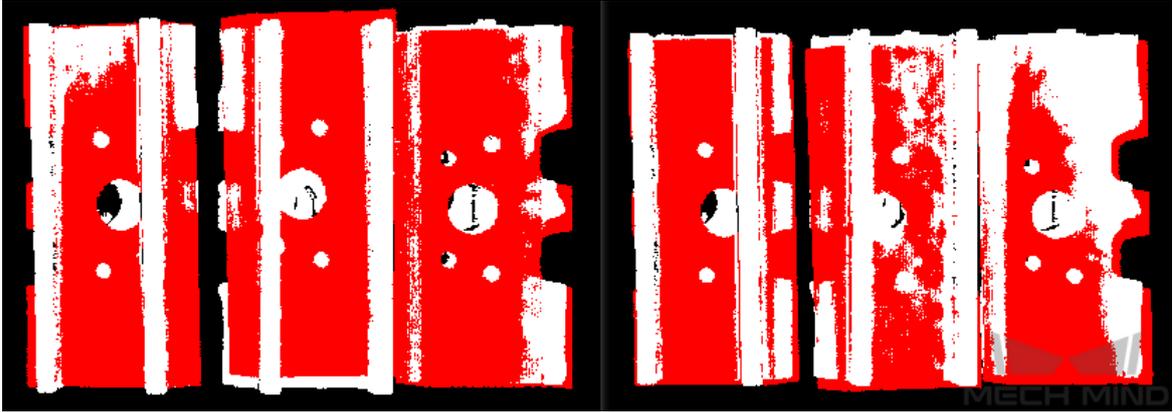
### 보조 속도 컨트롤러

기본값: 10

설명: 이 파라미터는 알고리즘 속도를 조절하는 데 사용됩니다. 값을 높이면 알고리즘 속도는 빨라지지만 매칭 정확도는 떨어집니다. 그 효과는 **메인 속도 컨트롤러** 보다 약합니다. 이 파라미터의 유효 범위는 1~20입니다.

조절 예시: 아래 그림과 같습니다. 왼쪽은 파라미터의 기본값이 10인 결과이고, 오른쪽은 파라미터를 15로 조절한 결과입니다. 조절한 후 매칭 정확도가 감소하지만 메인 속도 컨트롤러보다 영향이 작음을 알 수 있습니다.





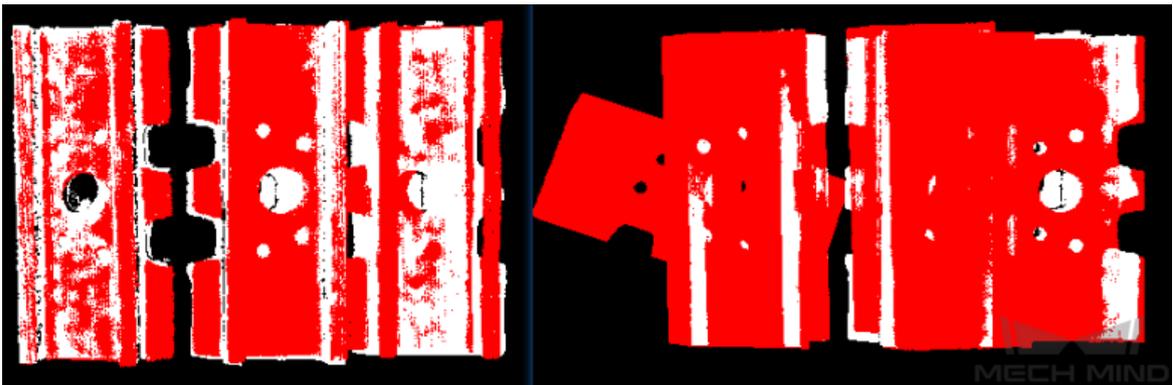
## 출력 설정

### 단일 포인트 클라우드에 대한 출력 결과 수

기본값: 3

설명: 이 파라미터는 각 포인트 클라우드 매칭에 대한 출력 수를 예상하는 데 사용됩니다. 이 값이 클수록 매칭하는 출력 결과가 더 많아집니다.

조절 예시: 비교 결과는 아래 그림과 같습니다. 왼쪽은 파라미터 값이 1일 때의 결과이고, 오른쪽은 파라미터를 3으로 조절한 결과입니다.



## SurfaceMatching 파라미터 설명

### 샘플링 설정

#### 자동 다운 샘플링

기본값: 선택함

설명: 이 파라미터는 자동 다운 샘플링을 사용할지를 결정하는 데 사용되며 이 파라미터를 선택하면 샘플링 후 모델의 희망 포인트 수에 따라 모델 포인트 클라우드 다운샘플링 간격 파라미터를 자동으로 조정합니다.

#### 샘플링 후 모델의 희망 포인트 수

기본값: 1000

설명: 이 파라미터는 샘플링된 포인트 클라우드의 포인트 수를 조정하는 데 사용되며 **자동 다운 샘플링**을 선택해야 이 파라미터를 설정할 수 있으며 희박한 포인트 클라우드의 포인트 수가 이 값에 가까울 때 적용됩니다. 값이 작을수록 샘플링된 포인트 클라우드에 있는 포인트 수가 적어 추정된

포즈의 정확도가 낮아집니다.

### 샘플링된 모델의 최대 포인트 수

기본값: 4000

설명: 이 파라미터는 모델 포인트 클라우드 다운 샘플링 후 포인트 클라우드의 최대 포인트 수를 설정하는 데 사용됩니다. 즉, 다운 샘플링 후 모델 포인트 클라우드의 포인트 수에 대한 상한을 설정하는 데 사용됩니다. 매칭 효과가 이상적이지 않으면 이 파라미터를 늘리는 것이 좋습니다. 매칭 속도에 대한 요구 사항이 더 높으면 이 파라미터를 줄이는 것이 좋습니다.

### 샘플링된 시나리오 모델의 최대 포인트 수

기본값: 3000

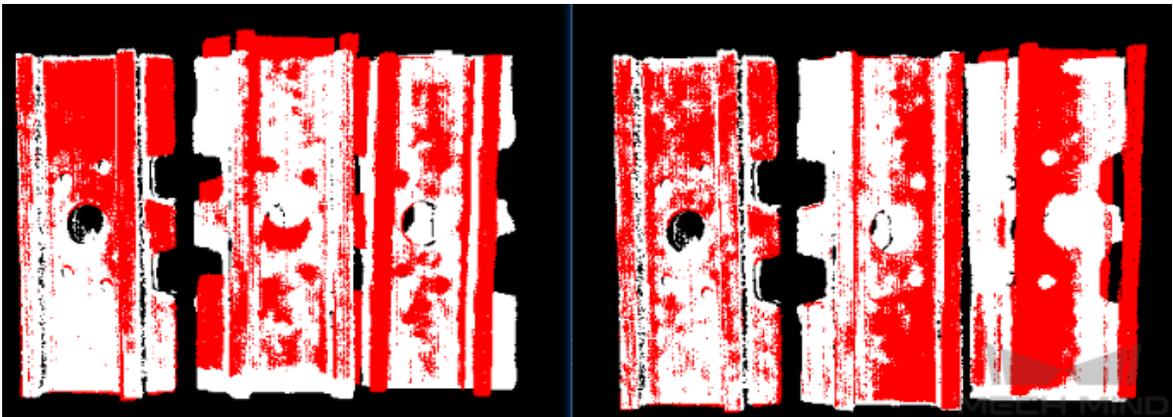
설명: 이 파라미터는 시나리오 포인트 클라우드 다운 샘플링 후 포인트 클라우드의 최대 포인트 수를 설정하는 데 사용됩니다. 즉, 다운 샘플링 후 시나리오 포인트 클라우드의 포인트 수에 대한 상한을 설정하는 데 사용됩니다. 매칭 효과가 이상적이지 않으면 이 파라미터를 늘리는 것이 좋습니다. 매칭 속도에 대한 요구 사항이 더 높으면 이 파라미터를 줄이는 것이 좋습니다.

### 모델 포인트 클라우드 다운 샘플링 간격

기본값: 10.000 mm

설명: 이 파라미터는 샘플링된 포인트 클라우드의 포인트 사이의 최대 거리(단위는 mm)를 조절하는 데 사용됩니다. 모델 포인트 클라우드 다운 샘플링 간격이 최소 샘플링 간격보다 작은 경우 최소 샘플링 간격을 실제 샘플링 간격으로 사용합니다. 이 값이 클수록 샘플링 후 계산에 사용되는 포인트 클라우드의 수가 적고 매칭 정확도가 낮아지며 알고리즘을 실행하는 데 걸리는 시간이 줄어듭니다.

조절 예시: 아래 그림과 같습니다. 왼쪽 그림은 기본값이 0.01일 때의 결과이고, 오른쪽 그림은 이 파라미터가 0.02로 조정된 결과입니다.



### 최소 샘플링 간격

기본값: 3.000 mm

설명: 이 파라미터는 샘플링 간격(단위:mm)을 계산하는 데 사용됩니다. **자동 다운 샘플링**을 선택한 후에만 이 파라미터를 설정할 수 있습니다. 계산된 샘플링 간격이 이 값보다 작으면 이 값이 실제 샘플링 간격으로 사용됩니다.

### 투표 설정

#### 거리 양자화

기본값: 1

설명: 이 파라미터는 포인트 사이의 거리를 양자화하는 데 사용됩니다.  $\text{거리 간격} = \text{거리 양자화} \times$

샘플링 간격이기 때문에 이 파라미터가 클수록 거리 간격이 커지고 매칭 정확도가 떨어집니다.

### 각도 양자화

기본값: 60

설명: 이 파라미터는 포인트 사이의 벡터 간의 협각 파라미터를 양자화하는 데 사용됩니다. 왜냐하면 **각도 간격 =  $2 \times 3.14$  / 각도 양자화**이기 때문에 이 파라미터가 클수록 각도 간격이 커지고 매칭 정확도가 낮아집니다.

### 최대 투표 비례

기본값: 0.8

설명: 이 파라미터는 최대 투표 수에 대한 투표 수 비율의 역치를 설정하는 데 사용됩니다. 이전 단계에서 각 포즈에 해당하는 투표 수를 구합니다. 최대 투표 수에 이 파라미터를 곱하여 역치를 구합니다. 한 포즈에 대한 투표 수가 역치보다 크면 해당 포즈가 보류된 클러스터링 작업을 진행합니다. 값이 작을수록 정확히 일치하는 항목을 찾을 가능성이 높아지지만 그에 따라 실행시간이 늘어납니다. 이 파라미터의 유효 범위는 0~1입니다.

### 기준점 샘플링 보폭

기본값: 5

설명: 이 파라미터는 기준점의 샘플링 보폭을 조정하는 데 사용됩니다. 포인트 클라우드에서 이 보폭을 간격으로 샘플링합니다. 이 값이 클수록 간격 샘플링 포인트가 적어지고 실행 속도가 빨라지지만 매칭 정확도가 떨어집니다.

### 비교점 샘플링 보폭

기본값: 1

설명: 이 파라미터는 비교점의 샘플링 보폭을 조정하는 데 사용됩니다. 포인트 클라우드에서 이 보폭을 간격으로 샘플링합니다. 이 값이 클수록 간격 샘플링 포인트가 적어지고 실행 속도가 빨라지지만 매칭 정확도가 떨어집니다.

- **기준점**과 **비교점**은 점쌍을 이루며, 샘플링 간격이 클수록 기준점과 비교점이 적어지고 점쌍이 적을수록 실행 속도가 빨라집니다.
- **기준점**은 매칭 모델에 있는 샘플 포인트입니다. **비교점**은 매칭 모델에 없는 샘플 포인트입니다.

### 클러스터링 설정

#### 클러스터링 비율

기본값: 0.1

설명: 이 파라미터는 클러스터링에 사용된 포즈 수와 계산된 전체 포즈의 비율을 조정하는 데 사용됩니다. 모든 포즈는 계산 과정에서 점수를 얻게 되며 모든 포즈는 점수의 크기에 따라 정렬됩니다. 이 파라미터는 클러스터링에 사용되는 포즈의 비례를 결정합니다. 기본값은 0.1, 즉 상위 10%를 취하여 클러스터링을 위한 포즈로 사용됩니다. 이 값이 클수록 정확히 매칭하는 항목을 찾을 가능성이 높아지지만 그에 따라 실행시간이 늘어납니다.

#### 각도 차이 역치

기본값: 15

설명: 이 파라미터는 클러스터링 과정에서 각도 증분 크기를 조정하는 데 사용됩니다. 최종 계산 결과 동일한 물체의 가능한 계산은 여러 포즈를 얻을 수 있으며, 매우 가까운 파라미터를 가진 포즈가 융합될 때 이 파라미터가 각도 파라미터의 증분을 결정합니다. 파라미터가 클수록 각도 차이가 큰 포즈도 최종 결과에 융합되어 매칭 정확도가 감소합니다.

## 거리 차이 역치

기본값: 0.02

설명: 이 파라미터는 클러스터링 과정에서 거리 증분 크기를 조정하는 데 사용됩니다. 최종 계산 결과 동일한 물체의 가능한 계산은 여러 포즈를 얻을 수 있습니다. 매우 가까운 파라미터를 가진 포즈가 융합될 때 이 파라미터는 거리 파라미터의 증분을 결정합니다. 파라미터가 클수록 거리가 커집니다. 크게 다른 포즈 또한 최종 결과에 융합되어 매칭 정확도가 감소합니다.

## 높은 점수를 가진 상위 N개의 클러스터링 결과를 출력하기

기본값: 5

설명: 이 파라미터는 클러스터링 후 얻은 여러 매칭 결과 중 가장 높은 점수를 가진 상위 N개의 결과를 최종 결과로 조정하는 데 사용됩니다.

## 포즈 검증 설정

### 포즈 검증 기능을 사용하기

기본값: 선택함

설명: 이 파라미터는 포즈 검증을 사용할지를 결정합니다. 이 파라미터를 선택하면 클러스터링의 모든 파라미터가 실효됩니다. 자세 검증 및 클러스터링은 최종 매칭 결과의 검증 및 선별의 두 가지 다른 방법이며 동시에 사용할 수 없습니다.

### 인접 포인트 검색 반경

기본값: 1

설명: 이 파라미터는 자세 검증 시 검증 영역의 크기를 컨트롤 하는데 사용되며, 단일 복셀이 단위입니다. 값이 증가하면 포즈 검증에 사용되는 표시 영역이 커지고 최종 결과 검증에 포함되는 해당 포인트가 증가하여 매칭 정확도가 감소합니다.

### 복셀 길이

기본값: 3

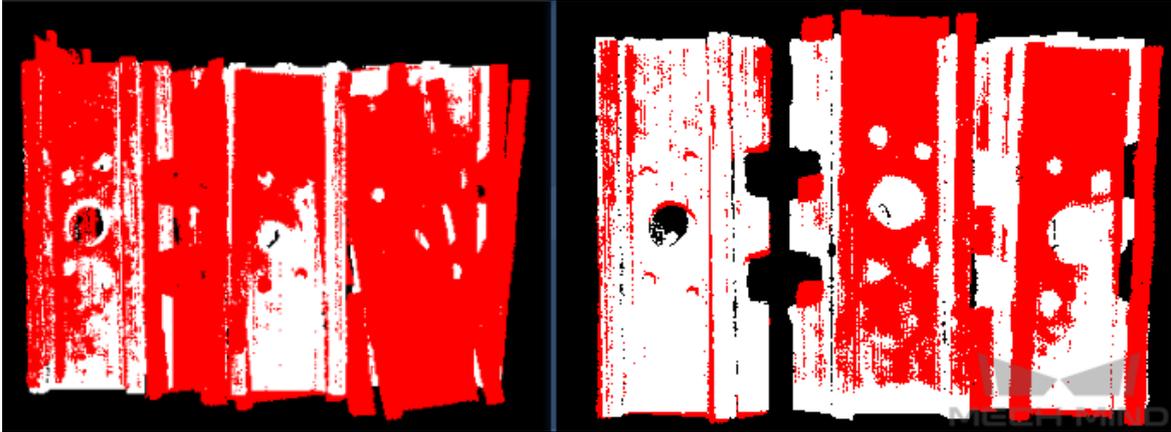
설명: 포인트 클라우드가 위치하는 공간은 3D 격자로 분할되는데, 이 파라미터는 3D 격자의 가장 작은 사이즈(단위:mm)입니다. 이 값을 높이면 프레임 선택 범위가 커지고 자세 검증을 위해 더 많은 포인트를 선택하게 되며 알고리즘 속도는 빨라지지만 매칭 정확도는 떨어집니다.

### 단일 포인트 클라우드에 대한 출력 결과 수

기본값: 3

설명: SurfaceMatching 알고리즘에서 이 파라미터 효과는 SurfaceMatchingEasyMode와 같습니다.

예시: 아래 그림과 같이 왼쪽이 이 파라미터 기본값이 3인 경우의 결과이고, 오른쪽이 이 파라미터를 1로 조정된 결과입니다.



## 결과 시각화

### 다운 샘플링된 포인트 클라우드 모델 표시

기본값: 선택하지 않음.

설명: 이 파라미터는 다운 샘플링된 모델 포인트 클라우드를 표시하는 데 사용됩니다.

### 다운 샘플링된 시나리오 포인트 클라우드 표시

기본값: 선택하지 않음.

설명: 이 파라미터는 다운 샘플링된 시나리오 포인트 클라우드를 표시하는 데 사용됩니다.

### 매칭 결과를 표시하기

기본값: 선택함

설명: 이 파라미터는 매칭 후의 모델과 시나리오 포인트 클라우드를 표시하는 데 사용됩니다.

## 4.3.3. 모든 파라미터를 받아들이기

### 기능 설명

이 스텝은 다른 모든 스텝의 출력을 받는 데 사용됩니다.

### 응용 시나리오

이 스텝은 디버깅 시 특정 포트의 데이터를 확인하는 데 자주 사용됩니다. 또한 실행 시간이 긴 스텝의 포트 데이터를 일시적으로 저장하고 디버깅 시간을 절약할 수도 있습니다.

### 입력 및 출력

모든 유형의 데이터



입력 데이터와 같은 데이터 유형



### 파라미터 설명

파라미터가 없음.

### 4.3.4. 포즈에 레이블을 추가하기

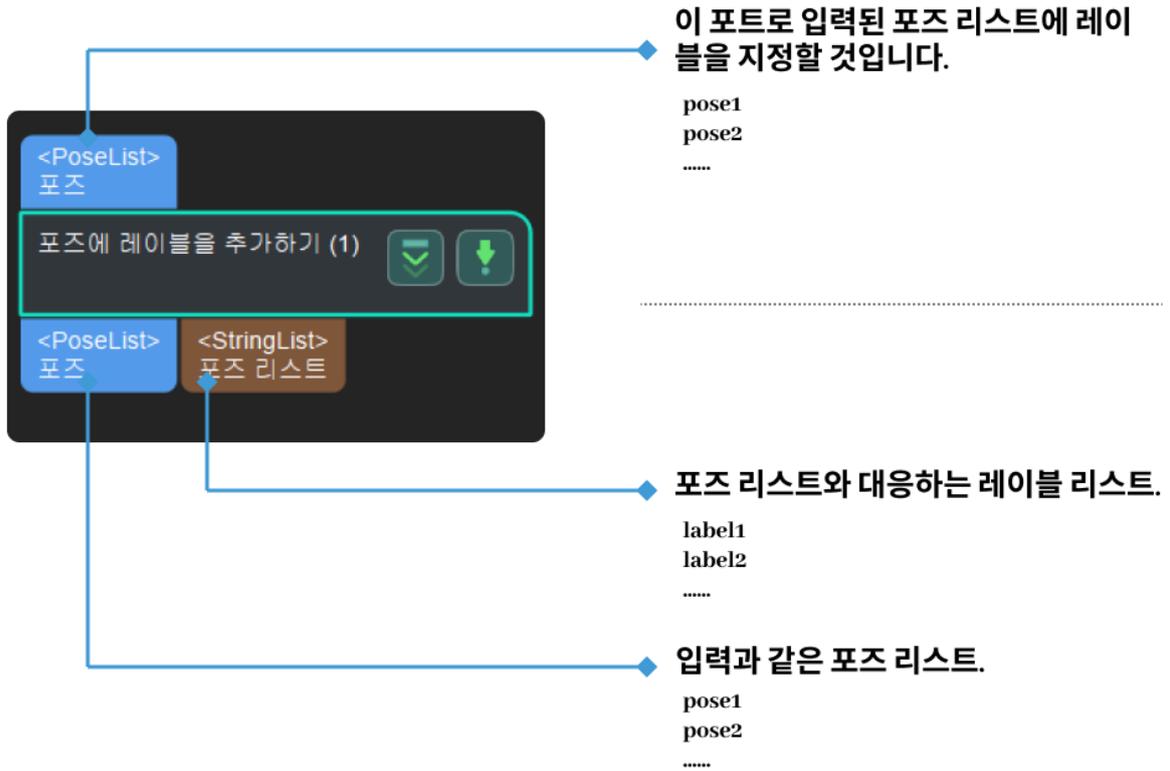
#### 기능 설명

입력한 각 포즈에 동일한 레이블을 추가합니다.

#### 응용 시나리오

특별한 제한 없이 대부분 시나리오에서 사용될 수 있는 포즈 처리 스텝입니다.

#### 입력 및 출력



#### 파라미터 설명

#### 파일 설정

#### 레이블 파일 경로

기본값: addLabels.json

설명: 레이블 파일의 절대 경로 혹은 상대 경로입니다. 파일에는 필요한 레이블 이름을 포함합니다.

### 4.3.5. 카메라의 왜곡으로 인한 정확하지 않는 포즈를 조정하기



이 스텝의 내용은 유지 보수 중입니다. 이 스텝에 대한 자세한 정보가 필요한 경우 ([docs@mech-mind.net](mailto:docs@mech-mind.net)) 로 문의하십시오.

#### 기능 설명

카메라 왜곡 계수(내부 파라미터에 포함됨)를 사용하여 포즈를 보정합니다.

#### 응용 시나리오

이 스텝은 특수 시나리오에 대한 사용자 지정 스텝이며 포즈에 대한 카메라 옵셋을 보정하는 데 사용됩니다.

#### 입력 및 출력

- **입력:** 이 포트에 입력한 포즈가 조정될 것입니다.
- **출력:** 조정된 포즈.

### 4.3.6. 옵셋에 따라 포즈를 조정하기



이 스텝의 내용은 유지 보수 중입니다. 이 스텝에 대한 자세한 정보가 필요한 경우 ([docs@mech-mind.net](mailto:docs@mech-mind.net))로 문의하십시오.

#### 기능 설명

이 스텝은 원기둥 포즈와 올바른 그리퍼 말단 포즈(즉, 픽 포인트) 사이의 포즈 옵셋을 계산하여 픽 포인트를 조정합니다. 출력은 원기둥의 실제 포즈와 포즈에서 픽 포인트까지의 옵셋입니다.

#### 응용 시나리오

이 스텝은 일반적으로 두 개의 원기둥 막대가 동시에 잡히지 않도록 끝에 꼭 맞는 원기둥형 막대를 잡는 시나리오에서 사용됩니다.

#### 입력 및 출력

##### ● 입력:

1. 끝과 끝이 맞는 원기둥형 막대가 포함된 포인트 클라우드.
2. 보정해야 하는 로봇 말단 그리퍼 포즈.
3. 단일 원기둥형 스톱 바의 길이.
4. 원기둥 모양 정보.

##### ● 출력:

1. 원기둥의 포즈.
2. 계산된 포즈에 상대적인 원기둥 픽 포인트의 옵셋.

### 4.3.7. 기울임에 따라 포즈를 조정하기



이 스텝의 내용은 유지 보수 중입니다. 이 스텝에 대한 자세한 정보가 필요한 경우 ([docs@mech-mind.net](mailto:docs@mech-mind.net))로 문의하십시오.

#### 기능 설명

이 스텝은 지정된 방향에 대한 포즈의 기울임에 따라 특정 거리만큼 포즈를 이동할 수 있습니다. 평행 이동 거리는 포즈에 수직인 방향으로 지정된 벡터의 투영 길이입니다.

#### 응용 시나리오

이 스텝은 특별한 시나리오에만 사용됩니다.

#### 입력 및 출력

- 입력: 이 포트에 입력한 포즈가 조정될 것입니다.
- 출력: 조정된 포즈.

### 4.3.8. 포즈를 포인트 클라우드의 표면으로 조정하기



이 스텝의 내용은 유지 보수 중입니다. 이 스텝에 대한 자세한 정보가 필요한 경우 ([docs@mech-mind.net](mailto:docs@mech-mind.net))로 문의하십시오.

## 기능 설명

이 스텝은 입력한 포즈를 지정된 방향을 따라 평행 이동할 수 있으므로 평행 이동한 후 포즈의 중심은 입력한 포인트 클라우드의 표면에 있습니다.

## 응용 시나리오

이 스텝은 일반적으로 픽 포인트를 보정하고 픽 포인트를 포인트 클라우드의 표면으로 보정하는 데 사용됩니다.

## 입력 및 출력

### ● 입력:

1. 이 포트에 입력한 포인트 클라우드는 포즈 조정을 위한 참조로 사용됩니다.
2. 이 포트에 입력한 포즈는 포인트 클라우드의 표면으로 조정될 것입니다.

### ● 출력:

1. 포인트 클라우드의 표면으로 조정된 포즈.
2. 시각화한 컬러 포인트 클라우드.

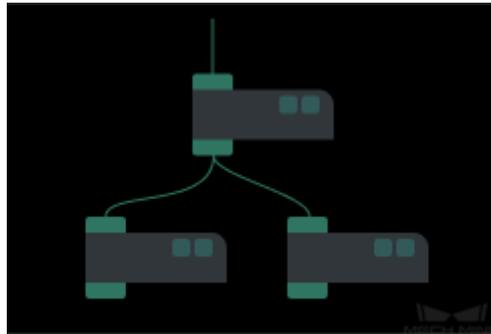
## 4.3.9. 입력 포트 할당자

## 기능 설명

프로시저에서 입력 포트에 대해 분배하는 데 사용됩니다.

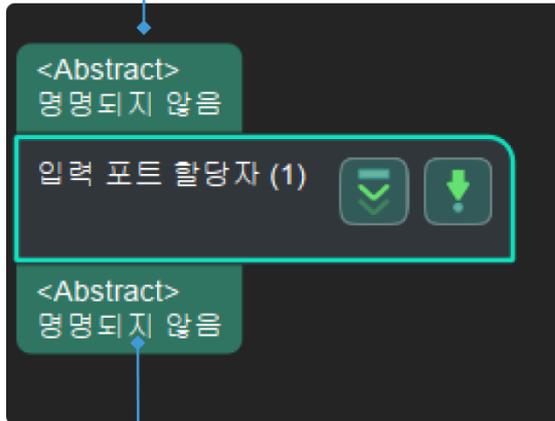
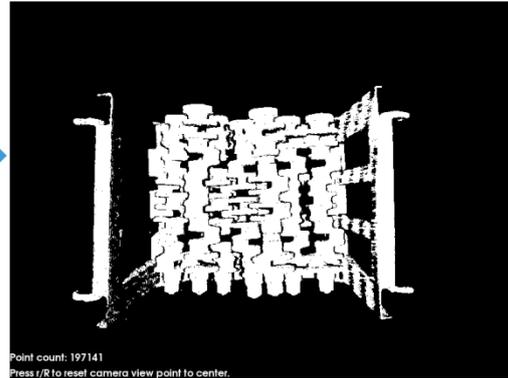
## 응용 시나리오

일반적으로 프로시저의 입력 포트 수를 줄이고 프로시저 외부에서 내부로 데이터를 전송하고 프로시저 내부에서 포트에 대해 분배합니다.

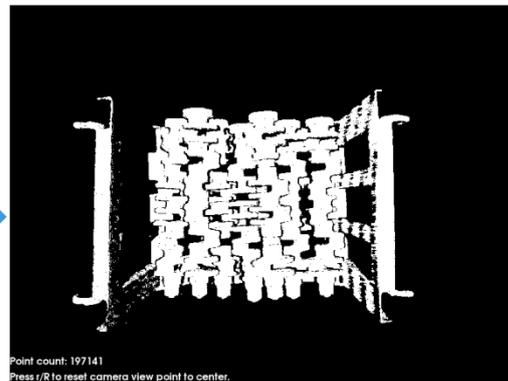


## 입력 및 출력

원시 포인트 클라우드 입력을 예시로



출력된 데이터 유형은 입력과 같습니다.



## 파라미터 설명

파라미터가 없음.

### 4.3.10. 마스크안에 대응하는 포인트 클라우드를 추출하기

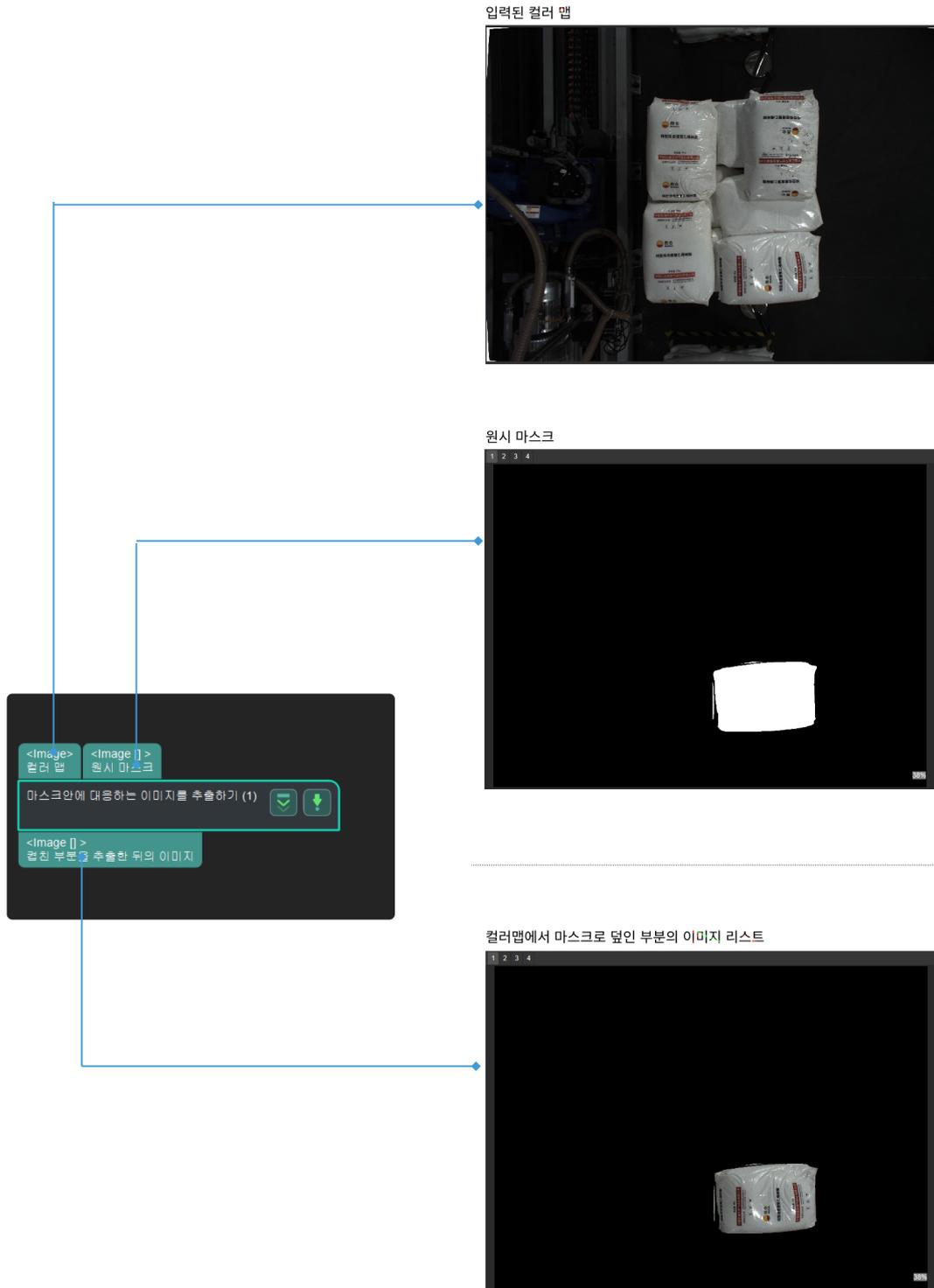
#### 기능 설명

이미지에 마스크를 적용하면 마스크로 가려진 부분이 추출되고 나머지 부분은 검은색 배경으로 채워집니다.

#### 응용 시나리오

일반적으로 가장 높은 층의 마스크를 획득하기 스텝과 함께 사용되며 시나리오에서 관련 없는 영역을 무시하고 간섭을 피하며 후속 딥 러닝 관련 스텝에 좋은 입력을 제공합니다.

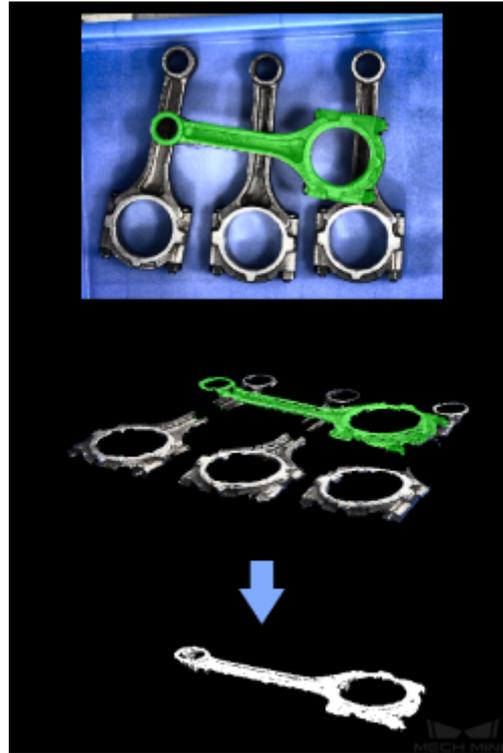
#### 입력 및 출력



### 4.3.11. 마스크안에 대응하는 포인트 클라우드를 추출하기

#### 기능 설명

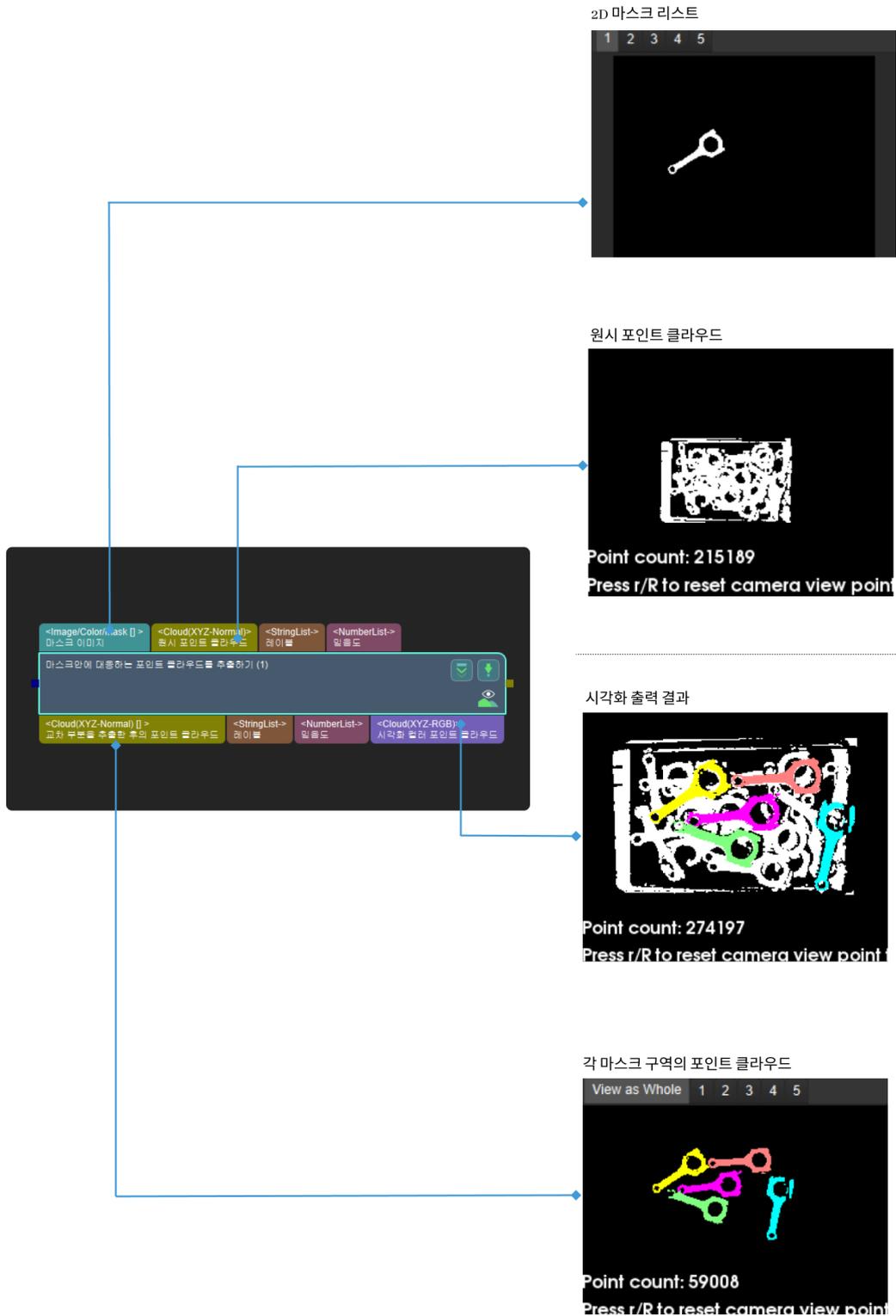
이 스텝은 포인트 클라우드에 마스크를 적용하는 데 사용됩니다. 마스크로 덮인 포인트 클라우드는 출력되고 마스크 외부의 포인트 클라우드는 삭제됩니다.



### 응용 시나리오

이 스텝은 마스크에 해당하는 3D 포인트 클라우드를 추출하는 데 사용됩니다.

### 입력 및 출력



## 파라미터 설명

파라미터가 없음.

### 4.3.12. 부울 값 리스트 논리 연산



이 스텝의 내용은 유지 보수 중입니다. 이 스텝에 대한 자세한 정보가 필요한 경우 ([docs@mech-mind.net](mailto:docs@mech-mind.net))로 문의하십시오.

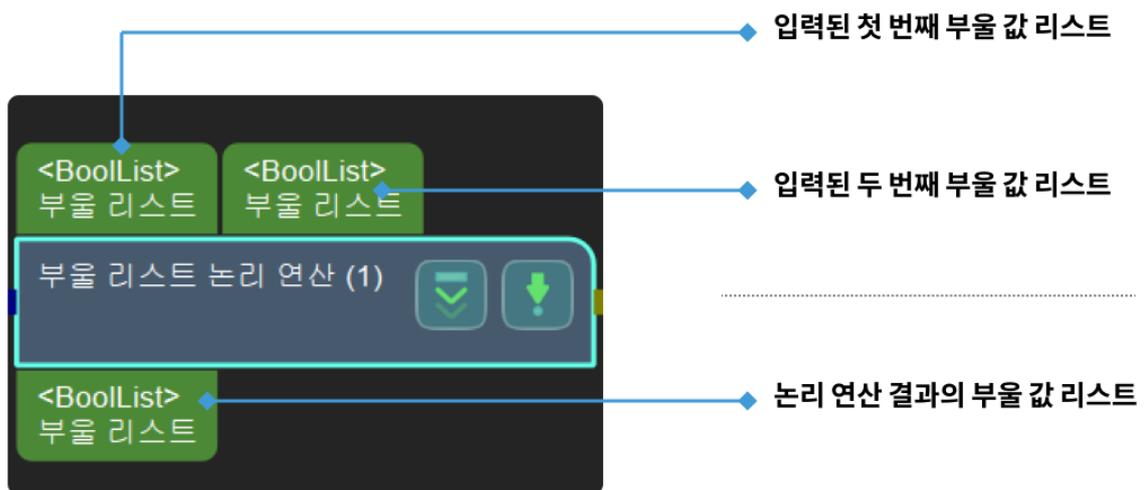
#### 기능 설명

설정된 논리 연산자에 근거하여 입력된 부울 값(True/False) 리스트에 대해 논리 연산을 합니다.

#### 응용 시나리오

특별한 제한 없이 대부분 시나리오에서 사용될 수 있는 논리 연산 스텝입니다.

#### 입력 및 출력



### 4.3.13. 절대값을 계산하기



이 스텝의 내용은 유지 보수가 중단되었습니다. 새 버전의 스텝 [수치 계산](#)을 사용하십시오. 이 스텝에 대한 자세한 정보가 필요한 경우 ([docs@mech-mind.net](mailto:docs@mech-mind.net))로 문의하십시오.

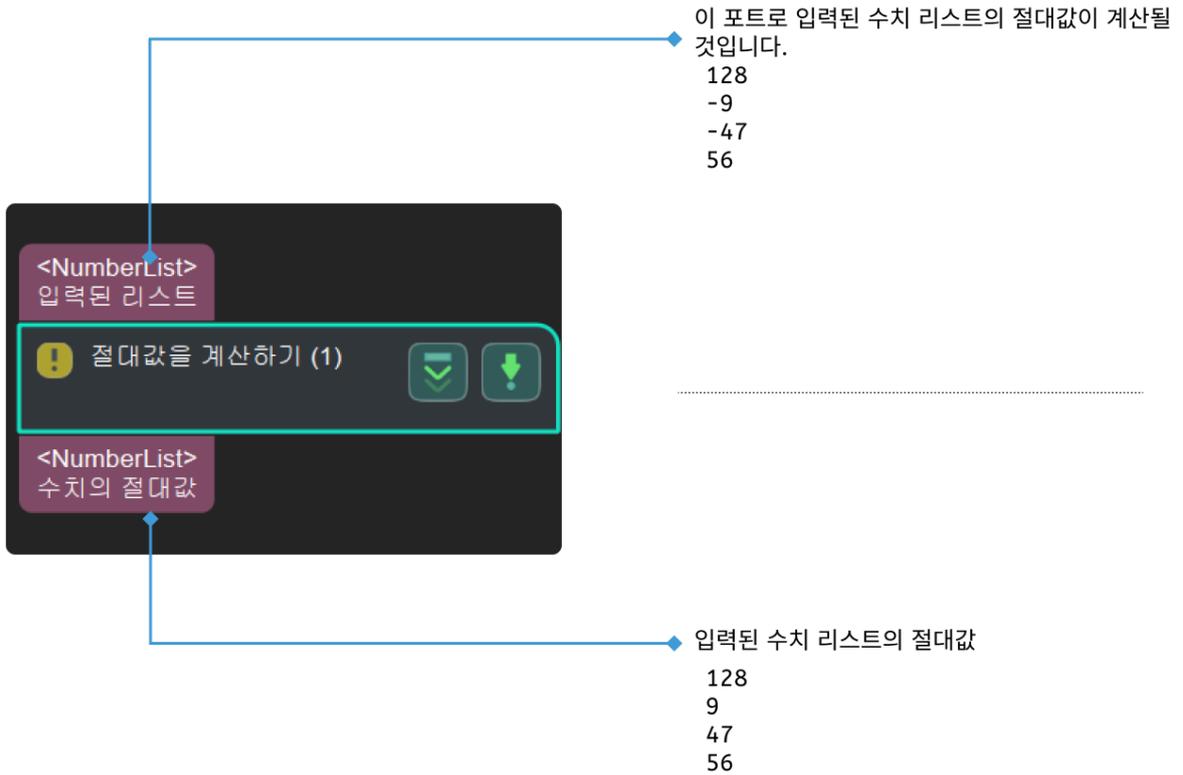
#### 기능 설명

입력된 수치 리스트의 절대값을 계산합니다.

#### 응용 시나리오

이 스텝은 구 버전이므로 보다 완전한 계산 기능을 갖춘 새 버전의 스텝 [수치 계산](#)을 사용하는 것이 좋습니다.

#### 입력 및 출력



#### 4.3.14. 마스크 면적을 계산하기



이 스텝의 내용은 유지 보수 중입니다. 이 스텝에 대한 자세한 정보가 필요한 경우 ([docs@mech-mind.net](mailto:docs@mech-mind.net))로 문의하십시오.

##### 기능 설명

여러 장 마스크 이미지의 면적을 계산합니다.

##### 응용 시나리오

이 스텝에서 계산된 면적 수치를 통해 후속 스텝의 순서 배열, 필터링 등 처리를 합니다.

##### 입력 및 출력



### 4.3.15. 두 포즈가 지정된 방향에 따른 거리를 계산하기



이 스텝의 내용은 유지 보수 중입니다. 이 스텝에 대한 자세한 정보가 필요한 경우 ([docs@mech-mind.net](mailto:docs@mech-mind.net))로 문의하십시오.

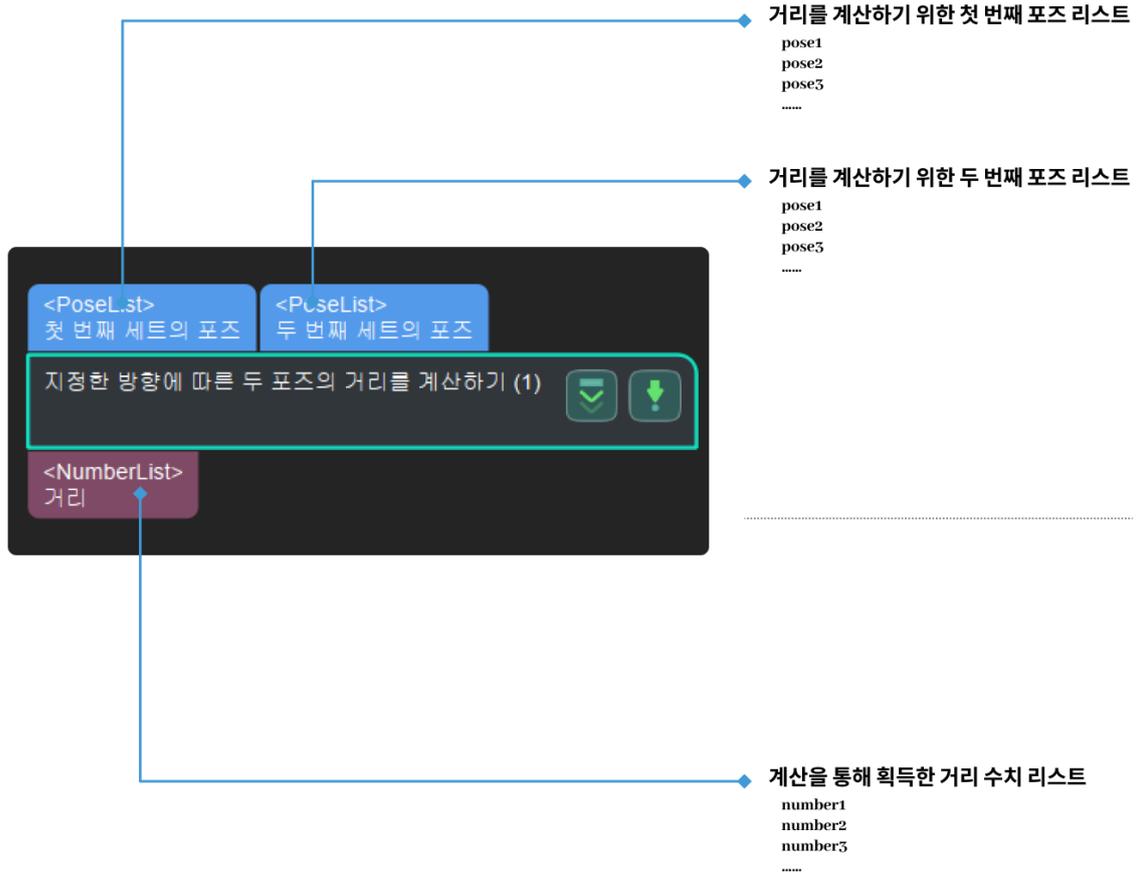
#### 기능 설명

두 포즈 중심으로 형성된 벡터가 지정된 축 또는 축의 조합에 투영되는 거리를 계산합니다.

#### 응용 시나리오

일반적인 포즈 거리 계산 스텝이며, 고정적인 응용 시나리오가 없습니다.

#### 입력 및 출력



### 4.3.16. 두 포즈의 지정된 축 사이의 각도를 계산하기



이 스텝의 내용은 유지 보수 중입니다. 이 스텝에 대한 자세한 정보가 필요한 경우 ([docs@mech-mind.net](mailto:docs@mech-mind.net))로 문의하십시오.

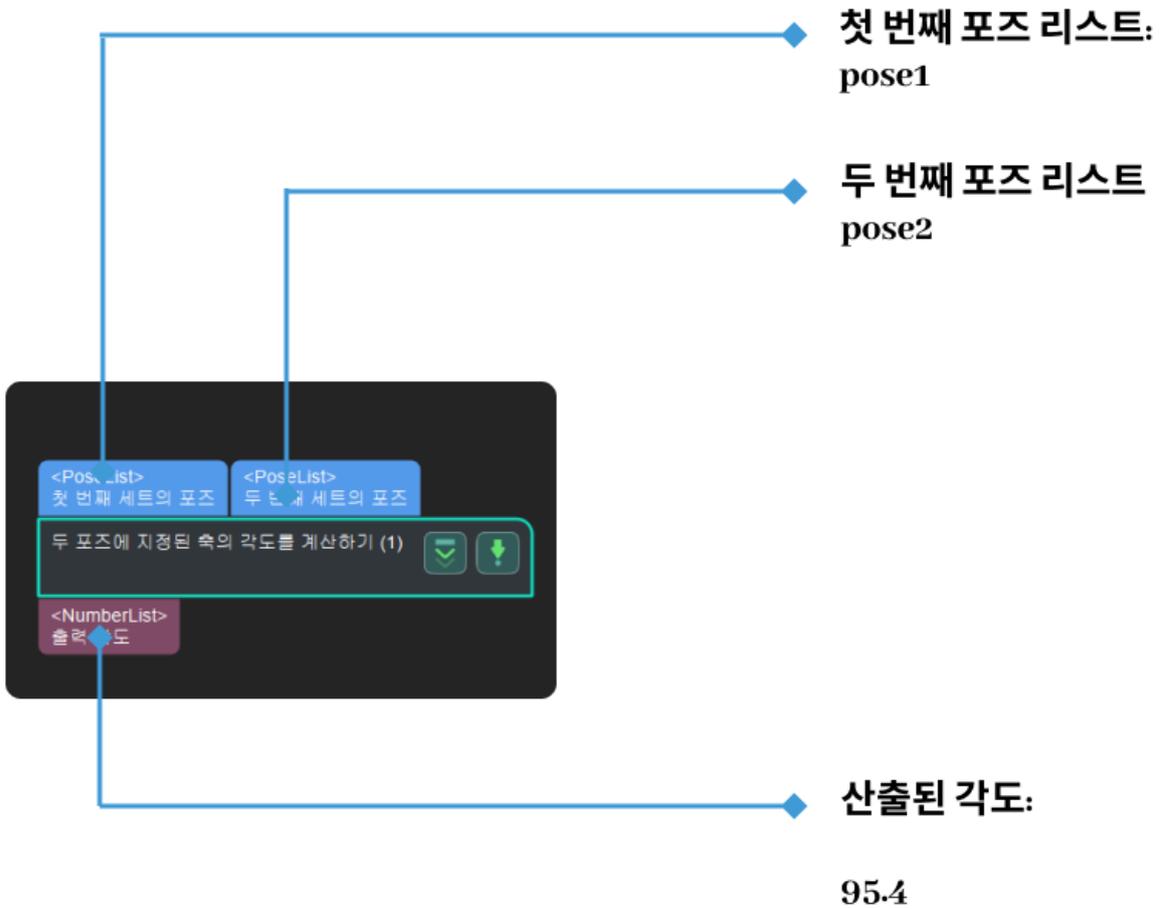
#### 기능 설명

두 포즈를 입력하고 지정된 축 사이의 각도를 계산합니다.

#### 응용 시나리오

일반적인 포즈 협각 계산 스텝이며, 고정적인 응용 시나리오가 없습니다.

#### 입력 및 출력



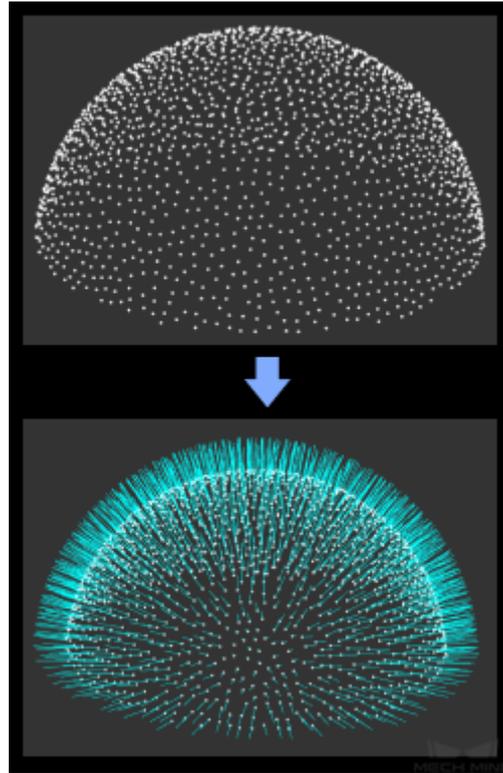
#### 4.3.17. 포인트 클라우드의 법선 벡터를 계산하고 필터링하기



이 스텝의 내용은 유지 보수 중입니다. 이 스텝에 대한 자세한 정보가 필요한 경우 ([docs@mech-mind.net](mailto:docs@mech-mind.net)) 로 문의하십시오.

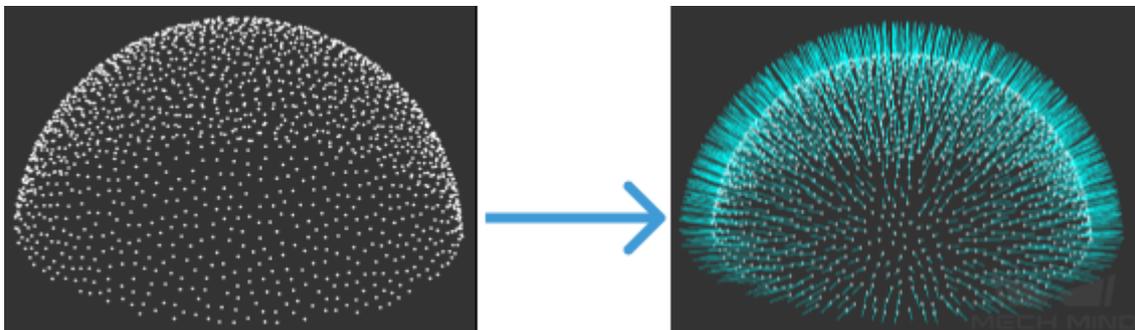
##### 기능 설명

이 스텝은 포인트 클라우드의 법선 방향을 계산하고 노이즈와 이상치를 제거합니다.

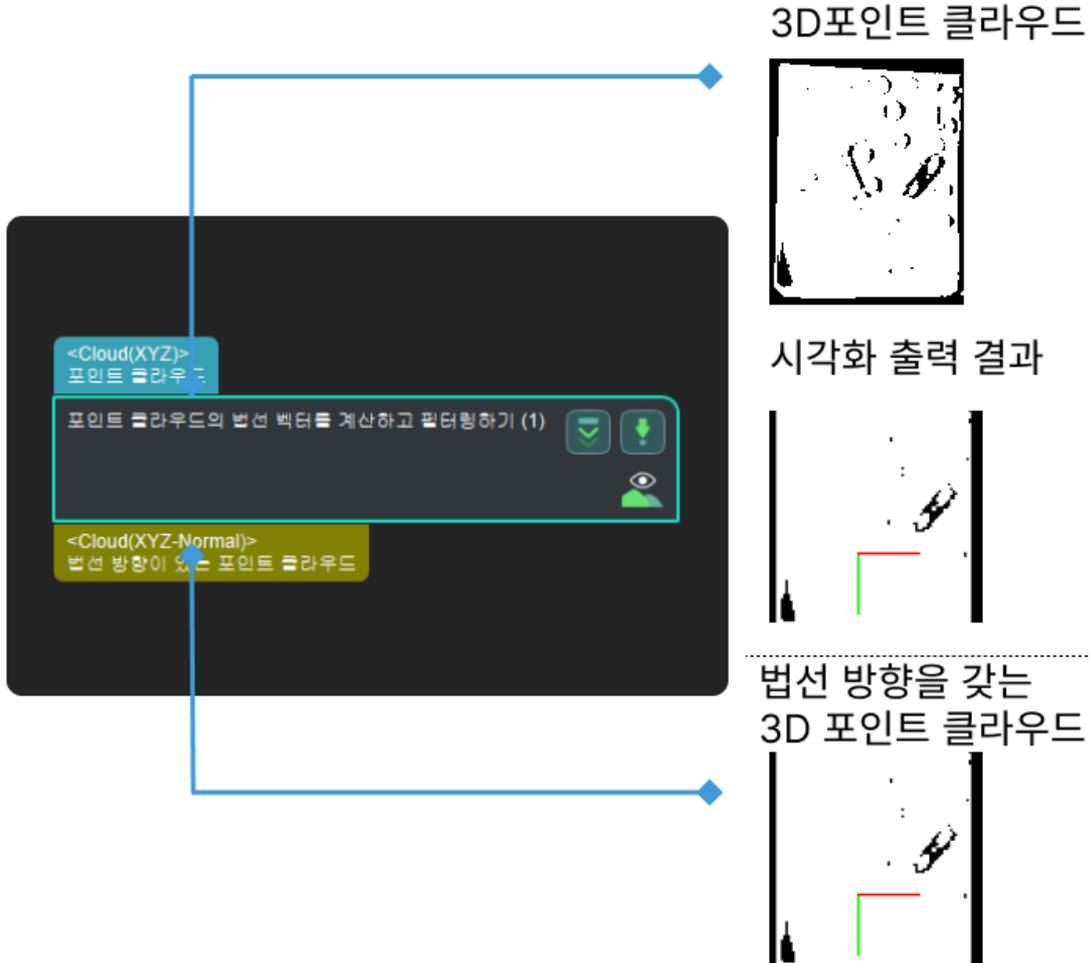


### 응용 시나리오

카메라에서 이미지를 캡처하기, **덱스 맵을 포인트 클라우드로 전환하기** 스텝 뒤에 사용되고 원시 포인트 클라우드에 법선 방향을 추가하고 노이즈와 이상치를 제거합니다.



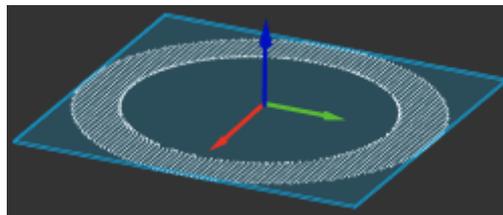
### 입력 및 출력



#### 4.3.18. 평면 포인트 클라우드의 포즈와 크기를 계산하기

##### 기능 설명

이 스텝은 평면 포인트 클라우드의 포즈 및 경계 박스의 크기를 계산합니다.



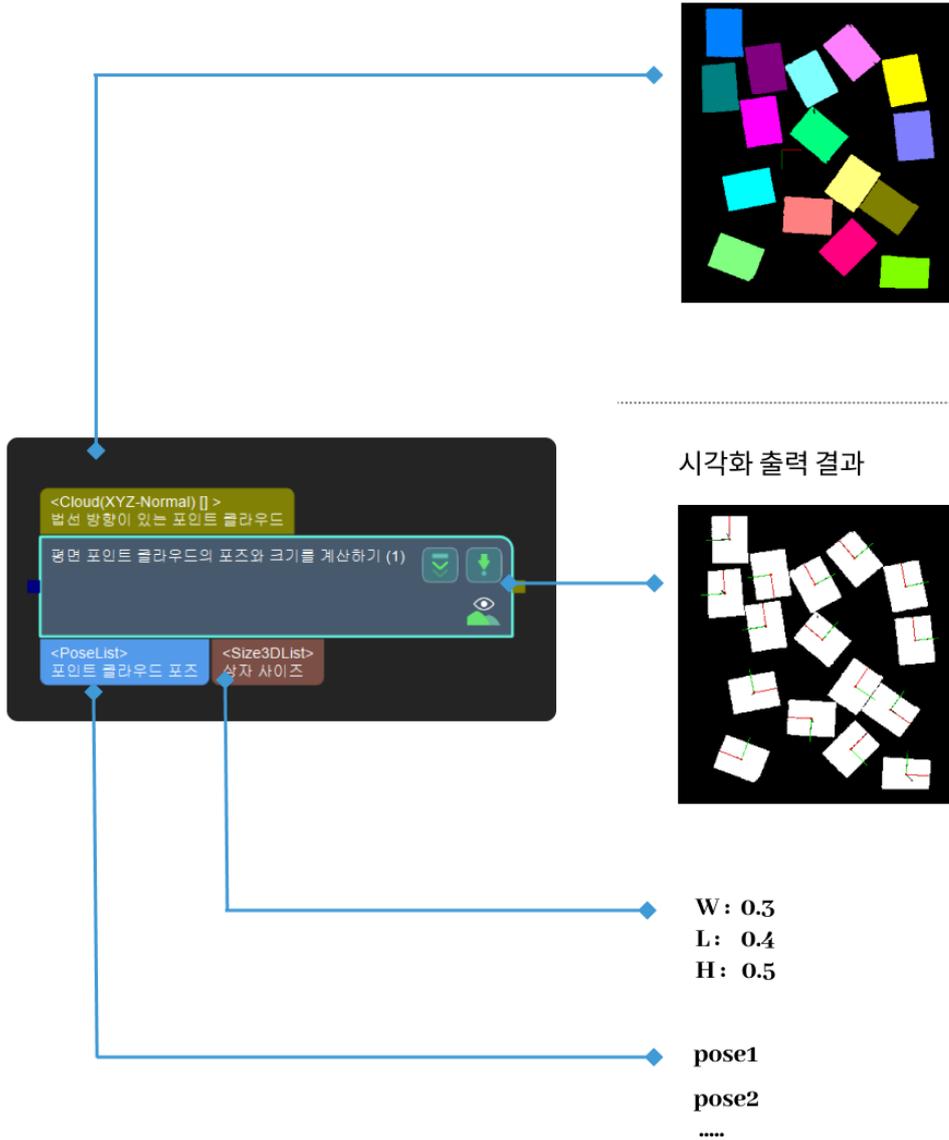
##### 응용 시나리오

이 스텝은 보통 상자, 마대 등 평면 포인트 클라우드의 픽 포인트 및 사이즈를 계산합니다.



이 스텝을 사용하기 전에 노이즈가 결과에 미칠 영향을 줄이기 위해 포인트 클라우드에 대해 필터링, 클러스터링 스텝을 먼저 진행해야 합니다.

##### 입력 및 출력



## 일반적인 파라미터 조절

### 포즈 계산 설정



포즈 중심 계산법과 포즈 자세 계산법은 1대1 대응해야 합니다.

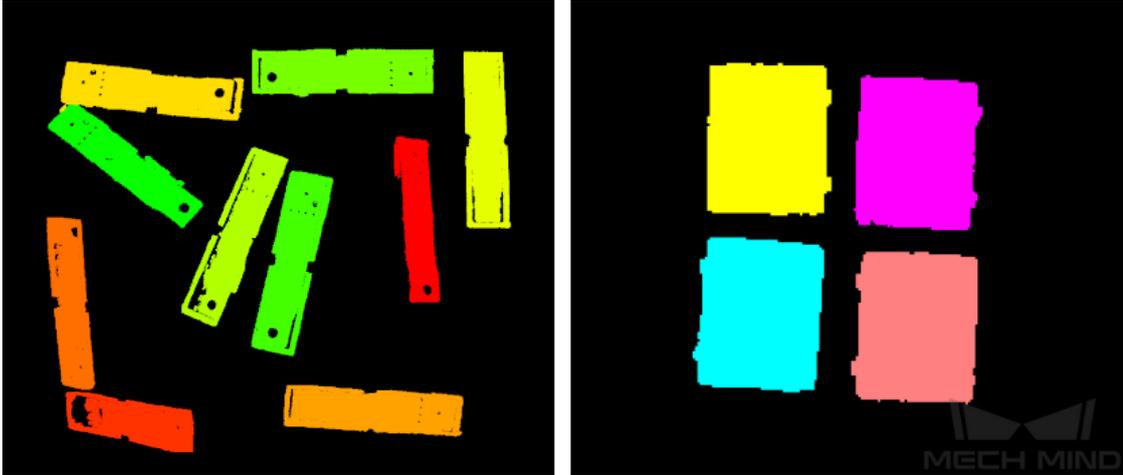
- BoundingRectCenter는 BoundingRectOrientation과 대응합니다.
- CloudCenter는 CloudOrientation과 대응합니다.

### 포즈 중심 계산법

파라미터 설명: 이 파라미터는 포즈 위치의 중심을 지정하는 데 사용됩니다.

값 리스트:

- BoundingRectCenter(포인트 클라우드 경계 박스 중심): 기본값. 아래 그림의 오른쪽과 같이 물체의 포인트 클라우드 분포가 비교적 균일할 때 사용합니다.
- CloudCenter(포인트 클라우드 중심): 아래 그림의 왼쪽과 같이 물체의 포인트 클라우드 분포가 고르지 않을 때 사용합니다.



조정 제안: 이 파라미터는 실제 필요에 따라 설정해야 합니다.

### 포즈 자세 계산법

파라미터 설명: 이 파라미터는 포즈의 방향을 지정하는 데 사용됩니다.

값 리스트:

- BoundingRectOrientation(포인트 클라우드 경계 박스 자세): 위 그림의 오른쪽과 같이 물체의 포인트 클라우드 분포가 비교적 균일할 때 사용합니다.
- CloudOrientation(포인트 클라우드 중심 자세): 아래 그림의 왼쪽과 같이 물체의 포인트 클라우드 분포가 고르지 않을 때 사용합니다.

조정 제안: 이 파라미터는 실제 필요에 따라 설정해야 합니다.

### 물체의 윤곽선과 정렬된 외접 직사각형을 사용하기

파라미터 설명: 이 파라미터는 외접 직사각형의 측면을 물체의 윤곽선과 최대한 평행하게 만들지 여부를 결정하는 데 사용됩니다. 선택하지 않으면 최소 외접 직사각형이 직접 사용됩니다.

기본값: 선택하지 않음.

조정 제안: 이 파라미터는 실제 필요에 따라 설정해야 합니다.

### 검색 반경 조정

파라미터 설명: 이 파라미터는 외접 직사각형이 물체의 윤곽점과 정렬되는지 여부를 판단하는 데 사용되며 단위는 픽셀(px)입니다. 윤곽점과 외접 직사각형 사이의 거리가 이 값보다 작으면 윤곽점이 외접 직사각형과 정렬된 것으로 간주됩니다.

기본값: 3 px



이 파라미터는 **물체의 윤곽선과 정렬된 외접 직사각형을 사용하기**를 선택한 후에만 설정할 수 있습니다.

### 상자 사이즈 계산 설정



이 파라미터 그룹은 상자에만 적용되며 마대를 계산하는 데 사용되지 않습니다.

### 상자 높이를 계산하기

파라미터 설명: 이 파라미터는 상자의 Z 방향 높이를 계산하는 데 사용됩니다.

기본값: 선택하지 않음.

조정 제안: 상자의 Z 방향 높이를 계산할 때만 이 파라미터를 선택하십시오.

#### 고정된 상자 하단 높이

파라미터 설명: 상자의 Z 방향 높이 계산을 용이하게 하기 위해 이 파라미터는 상자가 놓이는 베이스의 높이를 설정하는 데 사용됩니다. 이 값은 일반적으로 로봇 좌표계에서의 컨베이어 라인 또는 적재 플랫폼의 높이입니다.

기본값: 0.000 mm

조정 제안: 이 파라미터는 실제 필요에 따라 설정해야 합니다.



이 파라미터는 **상자 높이를 계산하기**를 선택한 후에만 설정할 수 있습니다.

### 4.3.19. 직사각형의 사이즈와 포즈를 계산하기

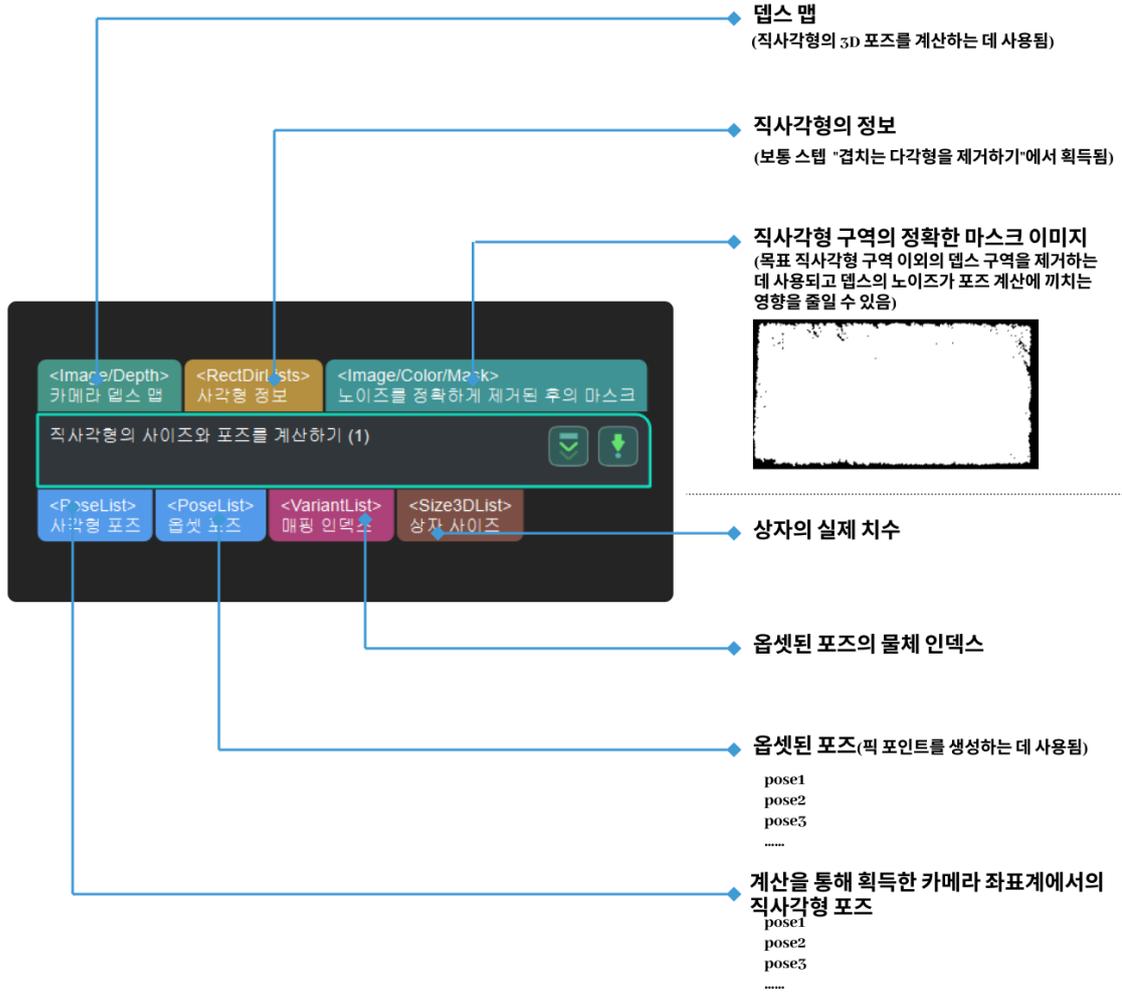
#### 기능 설명

덱스 맵과 직사각형의 정보를 통해 3D 공간에서 직사각형의 포즈를 계산합니다.

#### 응용 시나리오

고정적으로 **겹치는 다각형을 제거하기** 스텝과 함께 사용하며 **2D 매칭**을 통해 얻은 직사각형 물체의 카메라 좌표계에서의 포즈를 계산합니다.

#### 입력 및 출력



## 파라미터 설명

### 입력 유형

설명: 이 파라미터로 입력 유형을 선택할 수 있습니다.

기본값: Depth

값 리스트: Depth, VertexAndSize

조정 제안: 실제 상황에 근거하여 설정합니다.

### 포즈 계산의 설정

#### 포즈 Z 값 계산을 위한 검색 반경

조절 설명: 이 파라미터는 이미지의 특정 위치에 대한 뎀스 검색 반경을 조절하는 데 사용됩니다. 단위는 픽셀입니다. 이 파라미터는 입력 유형이 **Depth**인 경우에만 유효합니다.

기본값: 20

권장값: 20

유효 범위 :  $[0, +\infty)$

## 진공 그리퍼 크기 설정

### X축에 따른 진공 그리퍼의 길이

조절 설명: 이 파라미터는 진공 그리퍼로 덮인 구역의 X방향 사이즈를 조절하는 데 사용되며 진공 그리퍼로 덮인 구역의 크기를 조절할 수 있습니다. 단위는 밀리미터(mm)입니다. 진공 그리퍼가 목표 상자보다 크면 옆 상자를 피킹하지 않도록 작게 조정해야 합니다.

기본값: 0

권장값: 진공 그리퍼가 목표 상자보다 작은 경우 0으로 설정하고 진공 그리퍼가 목표 상자보다 큰 경우 진공 그리퍼의 실제 사이즈에 따라 설정하는 것을 추천합니다.

유효 범위 : [0,+∞)

### Y축에 따른 진공 그리퍼의 길이

조절 설명: 이 파라미터는 진공 그리퍼로 덮인 구역의 Y방향 사이즈를 조절하는 데 사용되며 진공 그리퍼로 덮인 구역의 크기를 조절할 수 있습니다. 단위는 밀리미터(mm)입니다. 진공 그리퍼가 목표 상자보다 크면 옆 상자를 피킹하지 않도록 작게 조정해야 합니다.

기본값: 0

권장값: 진공 그리퍼가 목표 상자보다 작은 경우 0으로 설정하고 진공 그리퍼가 목표 상자보다 큰 경우 진공 그리퍼의 실제 사이즈에 따라 설정하는 것을 추천합니다.

유효 범위 : [0,+∞)

### 진공 그리퍼와 다른 상자가 겹치는 비율의 역치

조절 설명: 진공 그리퍼가 작동 중 인접한 상자나 박스를 빨아들이는 것을 방지하기 위해 이 값을 현재 흡입 작동 실행 여부를 결정하는 기준으로 삼으십시오. 대상이 아닌 상자와의 겹침 비율이 이 값보다 작으면 피킹 작업을 수행하고, 이 값보다 크면 피킹 작업을 수행하지 않습니다.

기본값: 0.3

권장값: 실제 상황에 근거하여 설정합니다.

## 포즈 출력 방법

### 포즈 수량

파라미터 설명: 이 파라미터는 출력할 상자 포즈 수를 설정하는 데 사용됩니다.

기본값: One

값 리스트: One, Multiple

조정 제안: 이 파라미터는 실제 필요에 따라 설정해야 합니다.

## 4.3.20. 카메라에서 이미지를 캡처하기

### 기능 설명

이 스텝은 실제 혹은 가상 카메라를 통해 컬러 맵, 덤스 맵과 포인트 클라우드를 획득하는 데 사용됩니다.

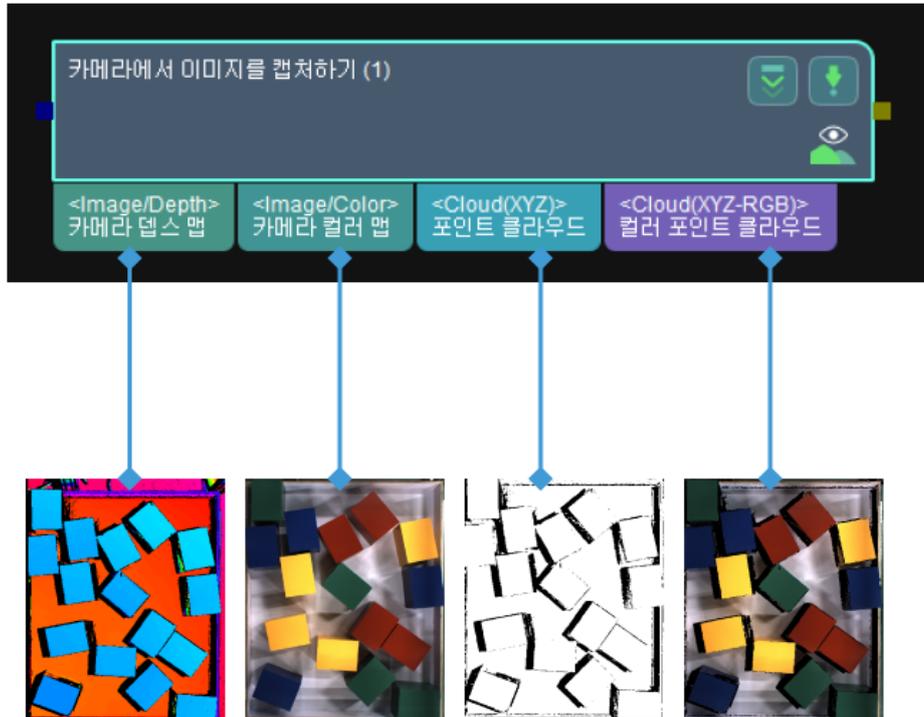


이 스텝을 통해 DEEP(V4) 및 LSR(V4) 시리즈 카메라와 연결하는 경우 이 스텝에서 출력된 컬러 맵은 2D맵(덤스 소스)입니다. 2D맵(덤스 소스)을 조절하려면 Mech-Eye Viewer에 있는 **2D맵(덤스 소스) 노출 모드** 및 **2D맵(덤스 소스) 노출 시간**을 조절하십시오.

## 응용 시나리오

이 시스템은 일반적으로 Mech-Vision 프로젝트의 첫 입력으로 사용됩니다. Mech-Eye 산업용 3D 카메라 또는 타사 카메라를 통해 데이터를 캡처하거나 가상 모드에서 로컬에 저장된 이미지 데이터를 읽습니다.

## 입력 및 출력



## 파라미터 조절 설명

### 카메라 설정

#### 카메라 유형

파라미터 설명: 이 파라미터는 사용할 카메라 유형을 선택하는 데 사용됩니다.

값 리스트:

- External2D: 외부 2D 카메라.
  - LMI: LMI 카메라.
  - Mech-Eye: 기본값, Mech-Eye 산업용 3D 카메라.
  - Mech-EyeTOF: Mech-Eye 산업용 3D 맞춤형 카메라.
- 
  - Mech-Eye 산업용 3D 카메라 펌웨어 버전을 2.0.0으로 업그레이드한 후, Mech-Vision 1.6.2 및 이상 버전을 사용하는 것을 권장합니다. Mech-Vision 1.6.1 및 이전 버전을 사용하면 **External2D EBUSCamera**에 있는 Mech-Eye 산업용 3D 카메라와 연결할 때 소프트웨어 충돌이 발생할 수 있습니다.
  - Mech-Eye만 **가상 모드**를 지원합니다.

조절 설명: 서로 다른 유형의 카메라는 다른 파라미터와 대응하여 실제 상황에 따라 조절하십시오.

## Mech-Eye

가상 모드가 꺼지면 실제 카메라를 사용하고, 가상 모드가 켜지면 로컬 데이터를 사용합니다. 가상 모드는 기본적으로 꺼져 있습니다.

### 실제 카메라를 사용하는 경우

#### 카메라 ID

파라미터 설명: 이 파라미터는 카메라와 연결하고 카메라 캘리브레이션 파라미터 그룹을 선택하는 데 사용됩니다.

설정 방법:

1. [ 카메라 선택 ] 버튼을 클릭하여 사용할 카메라 및 캘리브레이션 파라미터 그룹 선택 창을 엽니다.
2. 연결할 카메라를 선택하여 를 클릭합니다.
  -  카메라가 성공적으로 연결되면  버튼이 가 됩니다.
3. 카메라가 성공적으로 연결된 후 파라미터 그룹을 선택하기 목록 상자에서 캘리브레이션 파라미터 그룹을 선택하여 [ 확인 ]을 클릭합니다.



캘리브레이션 파라미터 그룹을 선택한 다음에 카메라 ID, 카메라 캘리브레이션 파라미터 그룹, Ip 및 Port 파라미터를 자동으로 채워집니다.

#### 카메라 캘리브레이션 파라미터 그룹

파라미터 설명: 이 파라미터는 카메라 캘리브레이션 파라미터 그룹을 지정하는 데 사용됩니다.

#### IP

파라미터 설명: 이 파라미터는 카메라 IP 주소를 표시하므로 변경하지 마십시오.

#### Port

파라미터 설명: 이 파라미터는 카메라 포트 번호를 표시하여 변경하지 마십시오.



IP 및 포트 번호는 표시에만 사용됩니다. 이 두 파라미터를 변경하면 카메라 연결이 끊어질 수 있습니다.

#### 타임아웃

파라미터 설명: 이 파라미터는 카메라가 데이터를 수신한 후 클라이언트에 데이터를 전송할 때까지의 최대 응답 시간을 설정하는 데 사용하며 단위는 초(s)입니다. 다음과 같은 두 가지 경우가 있습니다.

- 카메라는 최대 시간에 성공적으로 연결하지 않습니다.
- 카메라는 최대 시간에 성공적으로 데이터를 캡처하지 않습니다.

기본값: 20s

#### 카메라 재연결 횟수

파라미터 설명: 이 파라미터는 연결 시 오류가 발생한 경우, 카메라에 다시 연결할 수 있는 최대 횟수를 설정하는 데 사용됩니다.

기본값: 3

#### 파라미터 그룹 명칭

파라미터 설명: 이 파라미터는 카메라 파라미터 그룹을 선택하는 데 사용되며 카메라는 선택된 파라미터에 따라 데이터를 캡처합니다.



Mech-Eye Viewer에서 파라미터 그룹을 추가/삭제할 수 있습니다.

## 카메라 모델

파라미터 설명: 이 파라미터는 현재 연결된 카메라의 모델을 표시하는 데 사용하며 읽기 전용 파라미터입니다.

### 2D 맵 유형

파라미터 설명: 이 파라미터는 카메라에서 출력된 2D 맵 유형을 설정하는 데 사용됩니다.

값 리스트:

- Internal IR Image: 기본값. 2D 맵(덱스 소스), 덱스 맵 좌표계와 동일하여 수정할 필요가 없습니다.
- External Color Image: 2D 맵(텍스처), 수정해야 사용할 수 있습니다. 이 파라미터를 사용하려면 **덱스 맵 좌표계로 수정하기**를 선택해야 합니다.  
 LSR(V4) 시리즈와 DEEP(V4) 시리즈의 컬러 카메라에만 적용됩니다.

### 덱스 맵 좌표계로 수정하기

파라미터 설명: 이 파라미터는 외부 컬러 맵(External Color Image) 좌표계를 덱스 맵 좌표계로 수정하는 데 사용합니다.

기본값: 선택함.

 2D 맵 유형은 External Color Image인 경우 이 파라미터를 설정할 수 있습니다.

## 덱스 배경 제거 설정

### 덱스 맵으로 배경을 제거하기

파라미터 설명: 이 파라미터는 덱스 맵의 배경 정보를 제거하는 데 사용됩니다.

기본 값: 선택하지 않음.

### 덱스 맵 배경 이미지

파라미터 설명: 이 파라미터는 배경의 덱스 맵을 캡처하거나 읽는 데 사용됩니다.

기본값: depth\_background.png

설정 방법:

1. [ **배경 덱스 맵 파일 선택** ] 버튼을 클릭하여 배경 설정 화면으로 들어갑니다.
  - 로컬에 사용할 수 있는 배경 이미지가 있으면 2단계를 참조하십시오.
  - 로컬에 사용할 수 있는 배경 이미지가 없으면 3단계를 참조하십시오.
2. [ ... ]를 클릭하여 이미지를 선택하며 [ **확인** ]을 클릭하면 덱스 맵 배경 이미지가 선택되었습니다.
3. 배경 설정 화면 오른쪽에 있는 **감지된 카메라**에서 카메라를 선택하고 를 클릭하여 카메라와 연결합니다.



카메라가 성공적으로 연결되면 왼쪽의 **이미지 뷰어**에서는 카메라에서 캡처된 덱스 맵을 표시합니다.

4. 덱스 맵을 업데이트하려면 [ **한번 캡처** ]나 [ **연속 캡처** ]를 클릭하면 됩니다.
5. 배경 이미지 파일 명칭을 변경하려면 **배경 이미지**에서 파일 명칭을 입력하면 됩니다.
6. [ **확인** ]을 클릭하면 덱스 맵 배경 이미지가 선택되었습니다.

## 배경 뎁스의 변동

파라미터 설명: 이 파라미터는 물체에서 포인트의 뎁스와 배경 이미지의 뎁스를 비교하는 데 사용되며 단위는 밀리미터(mm)입니다. 포인트의 뎁스와 배경 이미지 뎁스의 값은 이 값보다 작을 때 포인트는 배경의 포인트로 간주되어 제거됩니다.

기본 값: 10mm

## 재캡처 횟수

파라미터 설명: 이 파라미터는 카메라 수집 시간 초과로 인해 카메라가 데이터 캡처에 실패한 경우 카메라가 데이터를 캡처하는 총 횟수를 지정하는 데 사용됩니다.

기본값: 3

권장값: 3

## Mech-Center에 있는 로봇 서비스 명칭

파라미터 설명: 이 파라미터는 로봇 서비스 명칭을 입력하는 데 사용되고 Mech-Center에서 연결된 로봇의 명칭과 동일해야 합니다.

기본값: 비어 있음.

## 가상 카메라를 사용하는 경우

가상 모드가 켜질 때 설정할 파라미터는 다음과 같습니다.

### 카메라 캘리브레이션 파라미터 그룹

조절 설명: \*데이터 경로가 선택된 다음에 **카메라 캘리브레이션 파라미터 그룹**을 자동으로 입력합니다. **데이터 경로** 아래에 여러 카메라 캘리브레이션 파라미터 그룹이 있는 경우 이 항목의 드롭다운 목록에서 사용하려는 것을 선택할 수 있습니다.

### 데이터 경로

파라미터 설명: 이 파라미터는 이미지, 내부 파라미터 및 외부 파라미터가 저장된 폴더를 선택하는 데 사용됩니다.

설정 방법: 오른쪽에 있는 를 클릭하여 이미지, 내부 파라미터 및 외부 파라미터가 저장된 폴더를 선택합니다.



가상 모드에서 이미지 파일 명칭과 파일의 형식은 다음과 같은 요구사항을 준수해야 하며 컬러 맵과 뎁스 맵의 이미지 번호는 일대일로 일치해야 합니다.

- 컬러 맵의 명칭은 rgb\_image\_xxxx.jpg로 명명해야 합니다.
- 뎁스 맵의 명칭은 depth\_image\_xxxx.png로 명명해야 합니다.

카메라 설치 방식이 Eye-in-Hand인 경우 이미지 번호와 플랜지 포즈는 일대일로 일치해야 합니다.

## 가상 카메라 도우미:

- 다음과 같은 경우에는 **가상 카메라 도우미**가 트리거되어 사용자가 이미지 데이터를 선택하는 데 도움을 줍니다.
  - **데이터 경로**에서 경로를 입력하지 않습니다.
  - 선택한 데이터 폴더에 여러 세트의 데이터가 있습니다.
  - 선택한 데이터 폴더에 컬러 맵 데이터, 뎁스 맵 데이터 또는 내부 및 외부 파라미터 데이터가 없습니다.
- 가상 카메라 도우미를 사용하는 경우 데이터를 읽는 단계는 다음과 같습니다.

- 가상 카메라 도우미에서 를 클릭하여 데이터 경로를 선택합니다.



경로를 선택한 다음에 **카메라 ID** 및 **파라미터 그룹**을 자동으로 입력합니다.

- **[검증]**을 클릭하여 **카메라 파라미터 그룹이 업데이트되었습니다.**라는 말이 나옵니다.
- 컬러 맵, 뎀스 맵, 플랜지 포즈 우측의  버튼을 차례로 클릭하여 컬러 맵, 뎀스 맵, 플랜지 포즈가 위치한 폴더를 선택합니다. 선택한 후 **[확인]**을 클릭합니다.



카메라 설치 방식이 Eye-in-Hand인 경우만 플랜지 포즈를 추가해야 합니다.

- 마지막에 나오는 **설정 완료** 창에서 **[확인]**을 클릭합니다.



가상 모드에서 데이터 경로의 이미지 데이터가 변경되면 데이터 경로를 다시 선택해야 하며, 그렇지 않으면 변경된 이미지 데이터를 읽을 수 없습니다.

## 재생 모드

파라미터 설명: 이 파라미터는 이미지의 읽는 순서를 지정하는 데 사용됩니다.

### 값 리스트

- 순차: 기본값. 폴더에 있는 이미지 순서대로 이미지를 읽습니다.
- 한 번 반복: 현재 이미지를 반복해서 읽습니다.
- 모두 반복: 폴더에 있는 이미지 순서대로 모든 이미지를 읽은 후 처음부터 읽습니다.
- 랜덤: 이미지를 랜덤으로 읽습니다.

조절 설명: 실제 수요에 따라 이미지 데이터를 읽는 모드를 선택하십시오.

## 현재 프레임

파라미터 설명: 이 파라미터는 현재 읽고 있는 이미지의 번호와 시간을 표시하는 데 사용됩니다.

### 이미지 명칭 유형

파라미터 설명: 이 파라미터는 **컬러 맵 경로** 출력 포트에서 출력된 이미지 명칭 유형을 선택하는 데 사용됩니다.

값 리스트: 완전한 경로, 파일 완전한 명칭, 파일 명칭.

기본값: 완전한 경로.

## 뎀스 배경 제거 설정

### 뎀스 맵으로 배경을 제거하기

파라미터 설명: 이 파라미터는 뎀스 맵의 배경 정보를 제거하는 데 사용됩니다.

기본 값: 선택하지 않음.

### 뎀스 맵 배경 이미지

파라미터 설명: 이 파라미터는 배경의 뎀스 맵을 캡처하거나 읽는 데 사용됩니다.

기본값: depth\_background.png

설정 방법:

1. **[배경 뎀스 맵 파일 선택]** 버튼을 클릭하여 배경 설정 화면으로 들어갑니다.

- 로컬에 사용할 수 있는 배경 이미지가 있으면 2단계를 참조하십시오.
  - 로컬에 사용할 수 있는 배경 이미지가 없으면 3단계를 참조하십시오.
2. [ ... ]를 클릭하여 이미지를 선택하며 [ 확인 ]을 클릭하면 덤스 맵 배경 이미지가 선택되었습니다.
  3. 배경 설정 화면 오른쪽에 있는 **감지된 카메라**에서 카메라를 선택하고 를 클릭하여 카메라와 연결합니다.



카메라가 성공적으로 연결되면 왼쪽의 **이미지 뷰어**에서는 카메라에서 캡처된 덤스 맵을 표시합니다.

4. 덤스 맵을 업데이트하려면 [ **한번 캡처** ]나 [ **연속 캡처** ]를 클릭하면 됩니다.
5. 배경 이미지 파일 명칭을 변경하려면 **배경 이미지**에서 파일 명칭을 입력하면 됩니다.
6. [ **확인** ]을 클릭하면 덤스 맵 배경 이미지가 선택되었습니다.

### 배경 덤스의 변동

파라미터 설명: 이 파라미터는 물체에서 포인트의 덤스와 배경 이미지의 덤스를 비교하는 데 사용되며 단위는 밀리미터(mm)입니다. 포인트의 덤스와 배경 이미지 덤스의 값은 이 값보다 작을 때 포인트는 배경의 포인트로 간주되어 제거됩니다.

기본 값: 10mm

### External2D

#### 카메라 ID

파라미터 설명: 이 파라미터는 카메라와 연결하고 카메라 캘리브레이션 파라미터 그룹을 선택하는 데 사용됩니다.

설정 방법:

1. [ **카메라 선택** ] 버튼을 클릭하여 **사용할 카메라 및 캘리브레이션 파라미터 그룹 선택** 창을 엽니다.
2. 연결할 카메라를 선택하여 를 클릭합니다.



카메라가 성공적으로 연결되면  버튼이 가 됩니다.

3. 카메라가 성공적으로 연결된 후 **파라미터 그룹을 선택하기** 목록 상자에서 캘리브레이션 파라미터 그룹을 선택하여 [ **확인** ]을 클릭합니다.



캘리브레이션 파라미터 그룹을 선택한 다음에 카메라 ID, **카메라 캘리브레이션 파라미터 그룹**, **Ip** 및 **Port** 파라미터를 자동으로 채워집니다.

#### 카메라 캘리브레이션 파라미터 그룹

파라미터 설명: 이 파라미터는 카메라 캘리브레이션 파라미터 그룹을 지정하는 데 사용됩니다.

#### IP

파라미터 설명: 이 파라미터는 카메라 IP 주소를 표시하므로 변경하지 마십시오.

#### Port

파라미터 설명: 이 파라미터는 카메라 포트 번호를 읽는 데 사용됩니다.



**IP** 및 **포트 번호**는 읽기에만 사용됩니다. 이 두 파라미터를 변경하면 카메라 연결이 끊어질 수 있습니다.

#### 타임아웃

파라미터 설명: 이 파라미터는 카메라가 데이터를 수신한 후 클라이언트에 데이터를 전송할 때까지의

최대 응답 시간을 설정하는 데 사용하며 단위는 초(s)입니다. 다음과 같은 두 가지 경우가 있습니다.

- 카메라는 최대 시간에 성공적으로 연결하지 않습니다.
- 카메라는 최대 시간에 성공적으로 데이터를 캡처하지 않습니다.

기본값: 20s

### 카메라 재연결 횟수

파라미터 설명: 이 파라미터는 연결 시 오류가 발생한 경우, 카메라에 다시 연결할 수 있는 최대 횟수를 설정하는 데 사용됩니다.

기본값: 3

### 자동 노출

파라미터 설명: 이 파라미터는 자동 노출 모델을 설정하는 데 사용됩니다.

기본값: 선택함

### 평균 그레이스케일 값

파라미터 설명: 이 파라미터는 이미지의 그레이스케일 기대값을 설정하는 데 사용됩니다. 이 값이 클수록 이미지가 밝아지고 작을수록 이미지가 어두워집니다.

기본값: 100

### 노출 시간

파라미터 설명: 이 파라미터는 이미지 휘도에 영향을 미칩니다. 이 값이 클수록 이미지가 밝아지고 작을수록 이미지가 어두워집니다.

기본값: 10

### 재캡처 횟수

파라미터 설명: 이 파라미터는 카메라 수집 시간 초과로 인해 카메라가 데이터 캡처에 실패한 경우 카메라가 데이터를 캡처하는 총 횟수를 지정하는 데 사용됩니다.

기본값: 3

권장값: 3

### Mech-Center에 있는 로봇 서비스 명칭

파라미터 설명: 이 파라미터는 로봇 서비스 명칭을 입력하는 데 사용되고 Mech-Center에서 연결된 로봇의 명칭과 동일해야 합니다.

기본값: 비어 있음.

## LMI

### 카메라 ID

파라미터 설명: 이 파라미터는 카메라와 연결하고 카메라 캘리브레이션 파라미터 그룹을 선택하는 데 사용됩니다.

설정 방법:

1. [카메라 선택] 버튼을 클릭하여 사용할 카메라 및 캘리브레이션 파라미터 그룹 선택 창을 엽니다.
2. 연결할 카메라를 선택하여 를 클릭합니다.



카메라가 성공적으로 연결되면  버튼이 가 됩니다.

3. 카메라가 성공적으로 연결된 후 **파라미터 그룹을 선택하기** 목록 상자에서 캘리브레이션 파라미터 그룹을 선택하여 [ **확인** ]을 클릭합니다.



캘리브레이션 파라미터 그룹을 선택한 다음에 카메라 ID, **카메라 캘리브레이션 파라미터 그룹**, **Ip** 및 **Port** 파라미터를 자동으로 채워집니다.

#### 카메라 캘리브레이션 파라미터 그룹

파라미터 설명: 이 파라미터는 카메라 캘리브레이션 파라미터 그룹을 지정하는 데 사용됩니다.

#### IP

파라미터 설명: 이 파라미터는 카메라 IP 주소를 표시하므로 변경하지 마십시오.

#### 재캡처 횟수

파라미터 설명: 이 파라미터는 카메라 수집 시간 초과로 인해 카메라가 데이터 캡처에 실패한 경우 카메라가 데이터를 캡처하는 총 횟수를 지정하는 데 사용됩니다.

기본값: 3

권장값: 3

#### Mech-Center에 있는 로봇 서비스 명칭

파라미터 설명: 이 파라미터는 로봇 서비스 명칭을 입력하는 데 사용되고 Mech-Center에서 연결된 로봇의 명칭과 동일해야 합니다.

기본값: 비어 있음.

#### Mech-EyeTOF

이 모드를 사용할 수요가 있으면 기술 서포트팀에게 문의하십시오.

### 4.3.21. 포인트 클라우드 처리(GPU)



이 스텝의 내용은 오래되어 유지 보수가 중단된 상태입니다. 새 버전 스텝 [포인트 클라우드의 법선 벡터를 계산하고 필터링하기](#)를 사용하십시오. 만약 긴급하게 이 스텝에 대한 추가 정보가 필요한 경우 ([docs@mech-mind.net](mailto:docs@mech-mind.net)) 로 문의하십시오.

#### 기능 설명

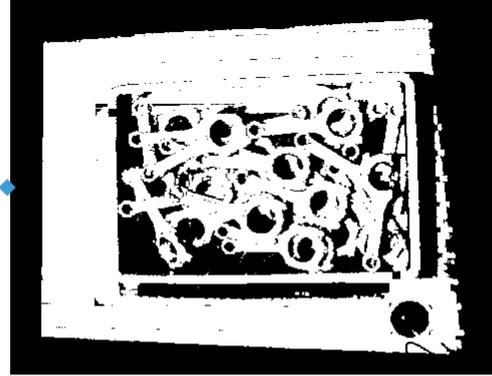
포인트 클라우드의 법선 벡터를 계산하고 노이즈와 이상치를 제거합니다.

#### 응용 시나리오

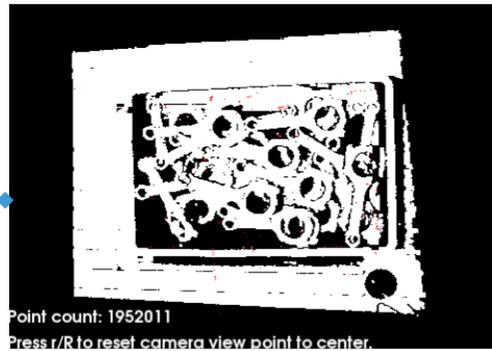
[카메라에서 이미지를 캡처하기](#), [덱스 맵을 포인트 클라우드로 전환하기](#) 스텝 뒤에 사용되고 원시 포인트 클라우드에 법선 벡터를 추가하고 노이즈와 이상치를 제거합니다.

#### 입력 및 출력

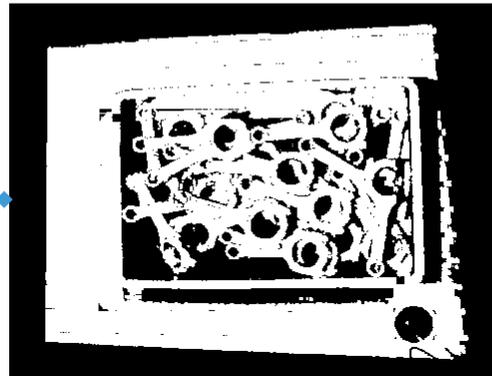
3D 포인트 클라우드



시각화 출력 결과



법선 방향이 있는 3D 포인트 클라우드



## 파라미터 설명

### Calculation Settings

#### 일괄 처리

파라미터 설명: 이 파라미터는 포인트 클라우드를 일괄 처리할지 여부를 결정하는 데 사용됩니다.

기본값: 선택하지 않음

조절 제안: 포인트 클라우드 포인트가 너무 많고 컴퓨터 그래픽 카드 메모리가 너무 작은 경우 이 파라미터를 선택하는 것이 좋습니다.

### 법선 방향을 계산할 때의 반경

파라미터 설명: 이 파라미터는 법선 벡터를 계산할 때 검색 반경을 설정하는 데 사용됩니다.

기본값: 5.000 mm

### Removal of Sparse Points

#### 이상치 제거

파라미터 설명: 이 파라미터는 포인트 클라우드의 이상치를 제거할지 여부를 결정하는 데 사용됩니다.

기본값: 선택하지 않음

조절 설명: 이 파라미터를 선택한 후 이상치는 설정된 이상치 역치에 따라 제거될 수 있습니다.

## 4.3.22. 3D ROI 내의 포즈를 획득하기



이 스텝의 내용은 유지 보수가 중단되었습니다. 새 버전의 스텝 [포즈가 3D ROI에 있는지 확인하기](#)를 사용하십시오. 이 스텝에 대한 자세한 정보가 필요한 경우 ([docs@mech-mind.net](mailto:docs@mech-mind.net))로 문의하십시오.

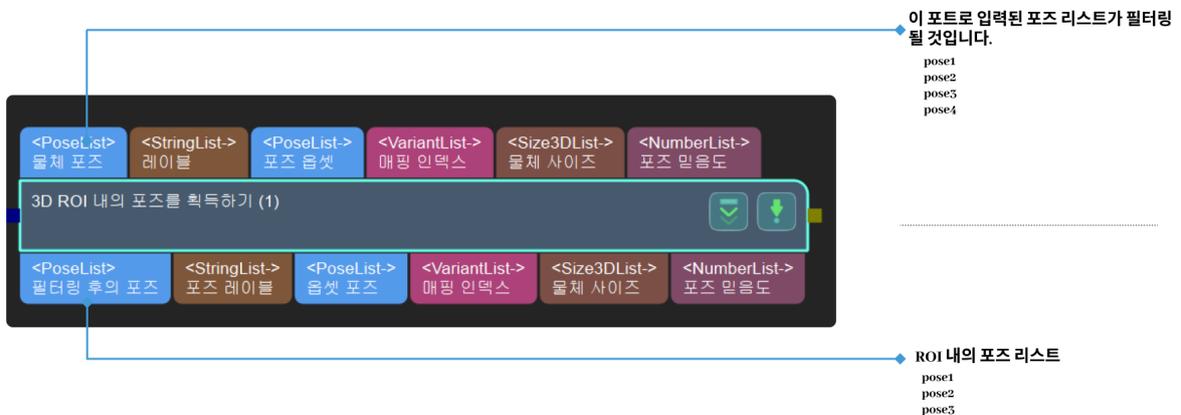
### 기능 설명

설정된 3D ROI를 통해 입력된 포즈 리스트에 대해 필터링합니다. ROI 내에 있는 포즈는 보류되며 외부에 있는 포즈는 제거됩니다.

### 응용 시나리오

일반적인 포즈 필터링 스텝입니다.

### 입력 및 출력



### 파라미터 설명

#### 3D ROI 설정

#### 3D ROI 설정

파라미터 설명: 이 파라미터는 3D ROI를 설정하는 데 사용됩니다.

조절 설명: [ **3D ROI 설정** ] 버튼을 클릭하여 나온 화면에서 **3D ROI**를 설정합니다.

#### 입력된 포즈가 있는 좌표계의 유형

파라미터 설명: 이 파라미터는 포즈가 있는 좌표계의 유형을 선택하는 데 사용됩니다.

값 리스트: 로봇 좌표계, 카메라 좌표계.

기본값: 로봇 좌표계.

### 4.3.23. 수치 비교



이 스텝의 내용은 유지 보수 중입니다. 이 스텝에 대한 자세한 정보가 필요한 경우 ([docs@mech-mind.net](mailto:docs@mech-mind.net))로 문의하십시오.

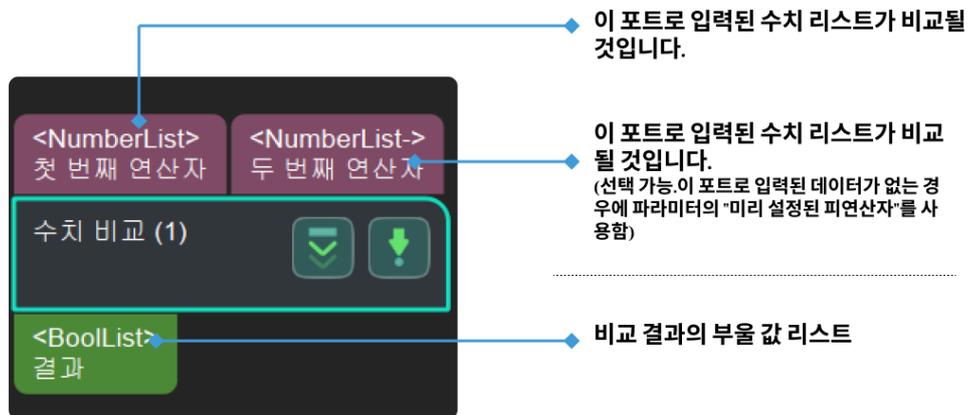
#### 기능 설명

설정된 처리 유형에 근거하여 입력된 두 수치 리스트를 비교하고 비교 결과의 부울 값(True/False) 리스트를 출력합니다.

#### 응용 시나리오

특별한 제한 없이 대부분 시나리오에서 사용될 수 있는 수치 비교 스텝입니다.

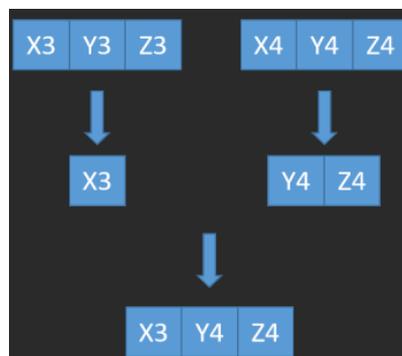
#### 입력 및 출력

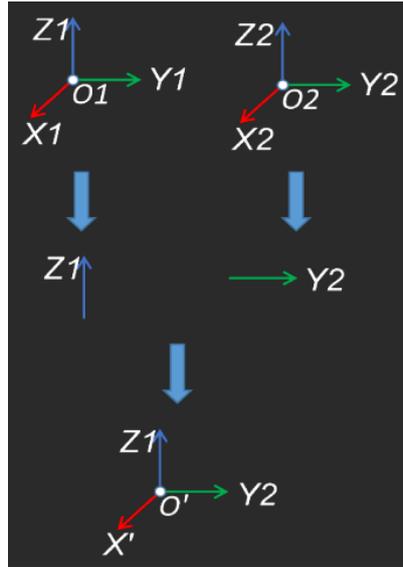


### 4.3.24. 입력된 각 포즈의 일부를 추출하여 새로운 포즈를 생성하기

#### 기능 설명

입력된 포즈의 일정한 부분을 새로운 포즈로 결합합니다.

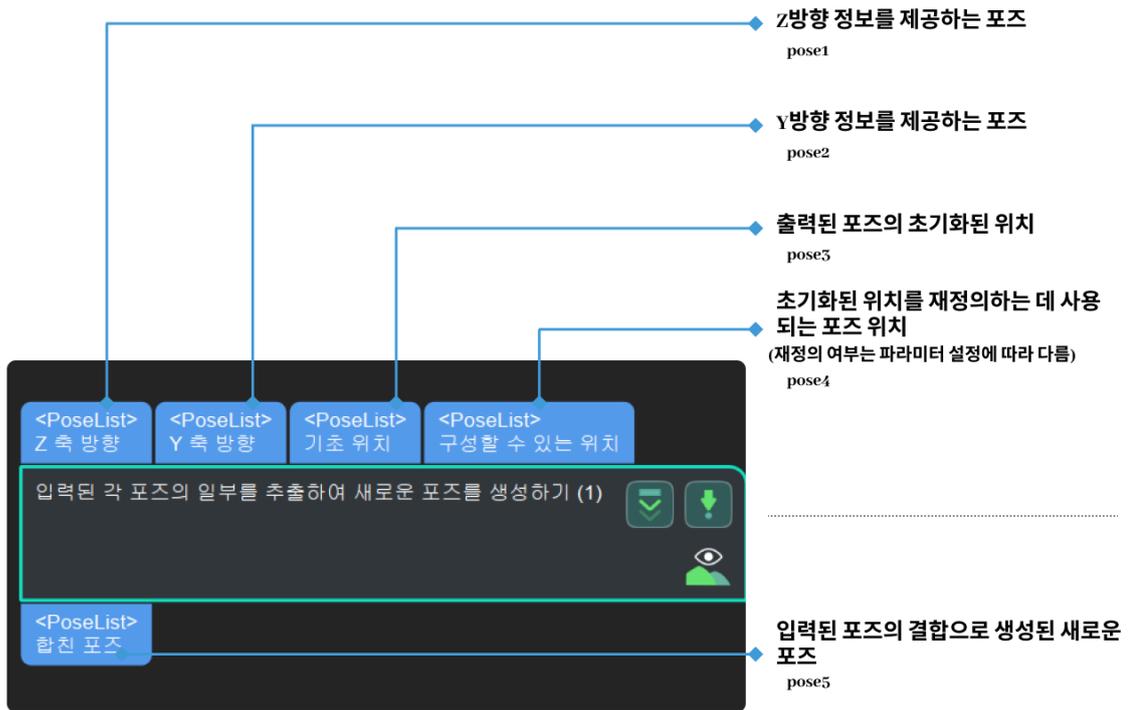




**응용 시나리오**

특별한 제한 없이 대부분 시나리오에서 사용될 수 있는 포즈 계산 스텝입니다.

**입력 및 출력**



**파라미터 설명**

**시각화 설정**

**포즈 시각화 설정**

기본값: 출력한 포즈만 표시하기.

값 리스트: 입력한 포즈만 표시하기, 출력한 포즈만 표시하기, 입력 및 출력한 포즈를 모두 표시하기.

설명: 처리 후의 포즈만 표시하기(기본값), 처리 전의 포즈만 표시하기, 처리하기 전/후의 포즈를 모두 표시하기.

#### 네 번째 입력에 사용되는 축

기본값: X

값 리스트: X, Y, Z, XY, XZ, YZ, XYZ, NONE

설명: 평행이동 파라미터를 선택하세요. 예를 들어 X를 선택하는 경우 세 번째 입력의 X값(X3)과 네 번째 입력의 Y,Z값(Y4, Z4)이 사용될 것입니다.

### 4.3.25. 데이터 리스트에서 지정된 차원의 요소 개수를 통계하기



이 스텝의 내용은 유지 보수 중입니다. 이 스텝에 대한 자세한 정보가 필요한 경우 ([docs@mech-mind.net](mailto:docs@mech-mind.net))로 문의하십시오.

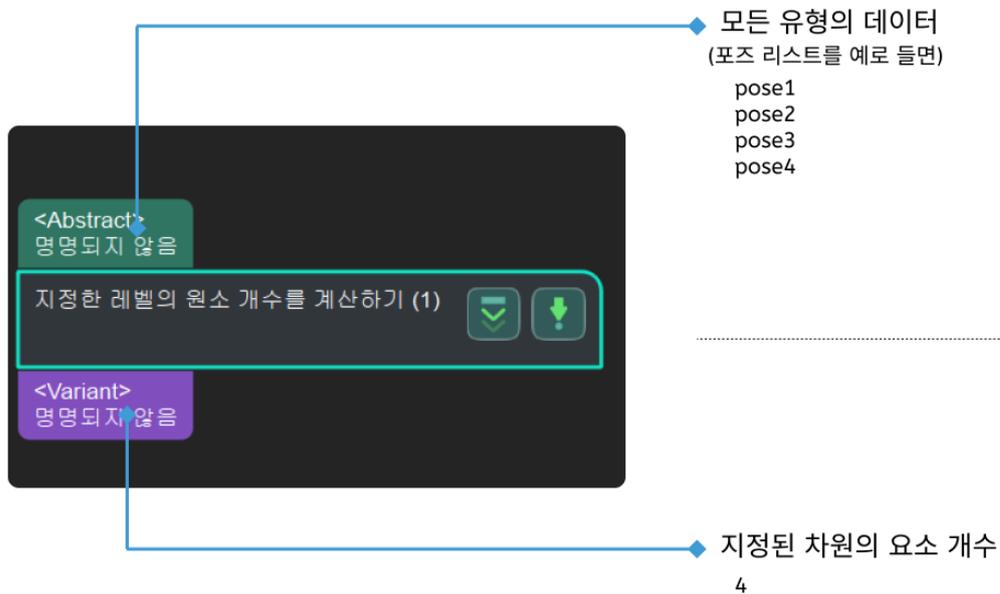
#### 기능 설명

입력된 데이터가 지정된 차원에서의 요소 개수를 통계합니다.

#### 응용 시나리오

일반적인 요소 개수 통계 스텝으로 "데이터 반복 및 연결" 스텝과 같이 사용될 수 있으며 후속 스텝의 각 입력 포트로 입력된 데이터 길이를 일치하게 만들 수 있습니다.

#### 입력 및 출력



### 4.3.26. 물체 치수 분해



이 스텝의 내용은 유지 보수 중입니다. 이 스텝에 대한 자세한 정보가 필요한 경우 ([docs@mech-mind.net](mailto:docs@mech-mind.net))로 문의하십시오.

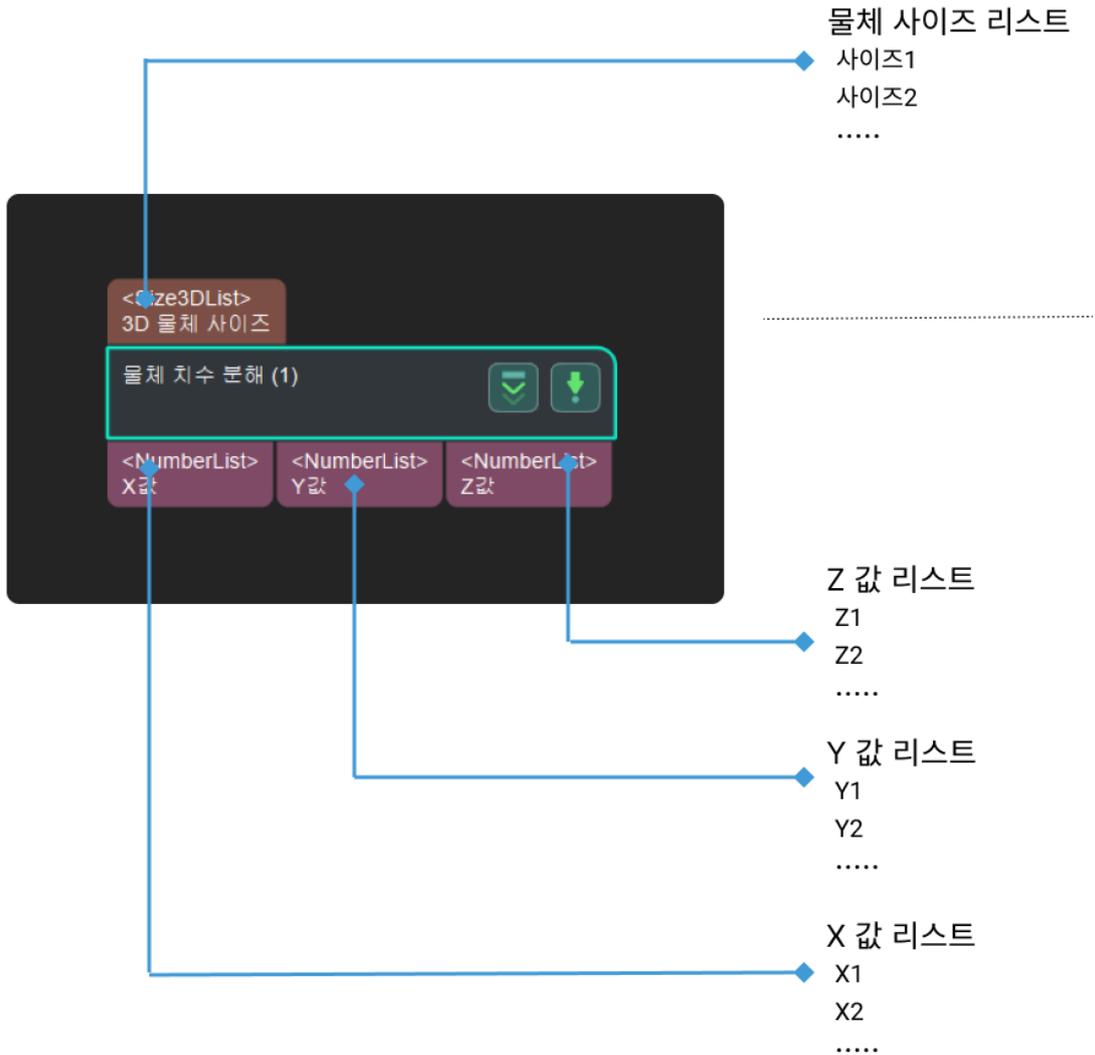
### 기능 설명

리스트에 있는 물체의 길이, 너비 및 높이 치수(X, Y, Z)를 세 개의 개별 값 목록으로 분할합니다.

### 응용 시나리오

일반적으로 물체의 길이, 너비, 높이를 추출한 후 각 항목을 별도로 처리합니다.

### 입력 및 출력



### 4.3.27. 포즈를 사원수와 평행 이동 벡터로 분해하기



이 스텝의 내용은 유지 보수 중입니다. 이 스텝에 대한 자세한 정보가 필요한 경우 ([docs@mech-mind.net](mailto:docs@mech-mind.net))로 문의하십시오.

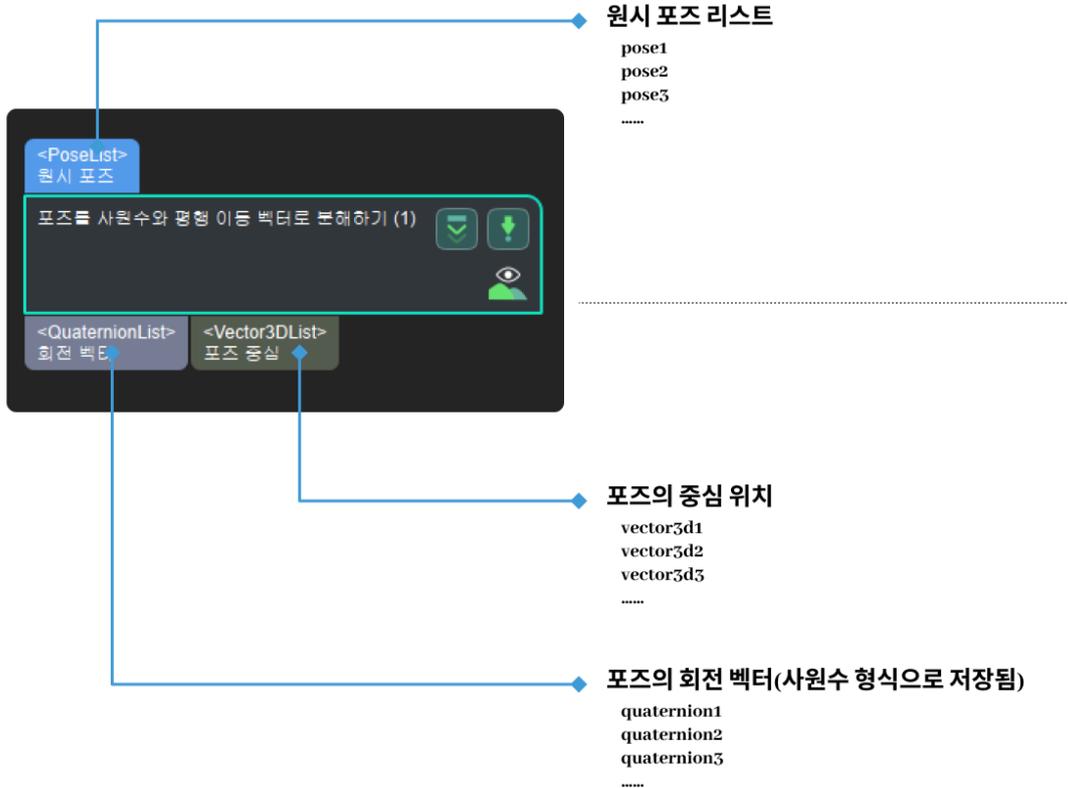
### 기능 설명

포즈를 사원수와 평행 이동 벡터로 분해합니다.

## 응용 시나리오

일반적으로 사용되는 포즈 분해 스텝으로 **회전 벡터** 및 **포즈 중심** 데이터 유형을 입력해야 하는 스텝에 데이터 소스를 제공합니다.

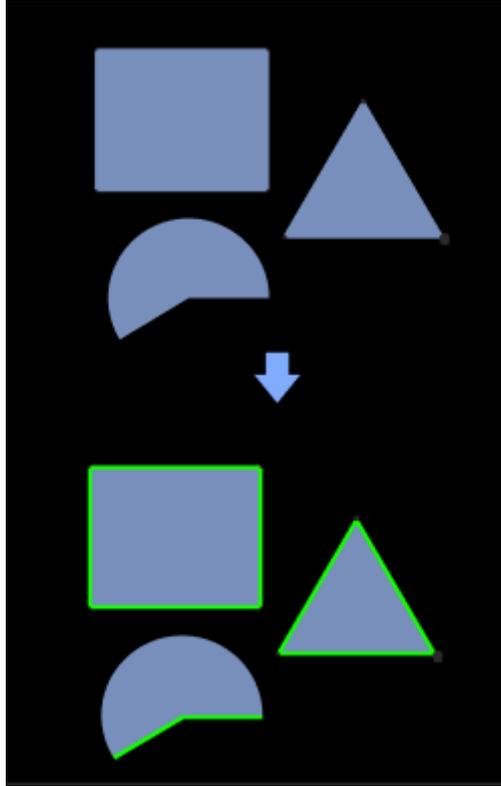
## 입력 및 출력



## 4.3.28. 라인 세그먼트 감지

### 기능 설명

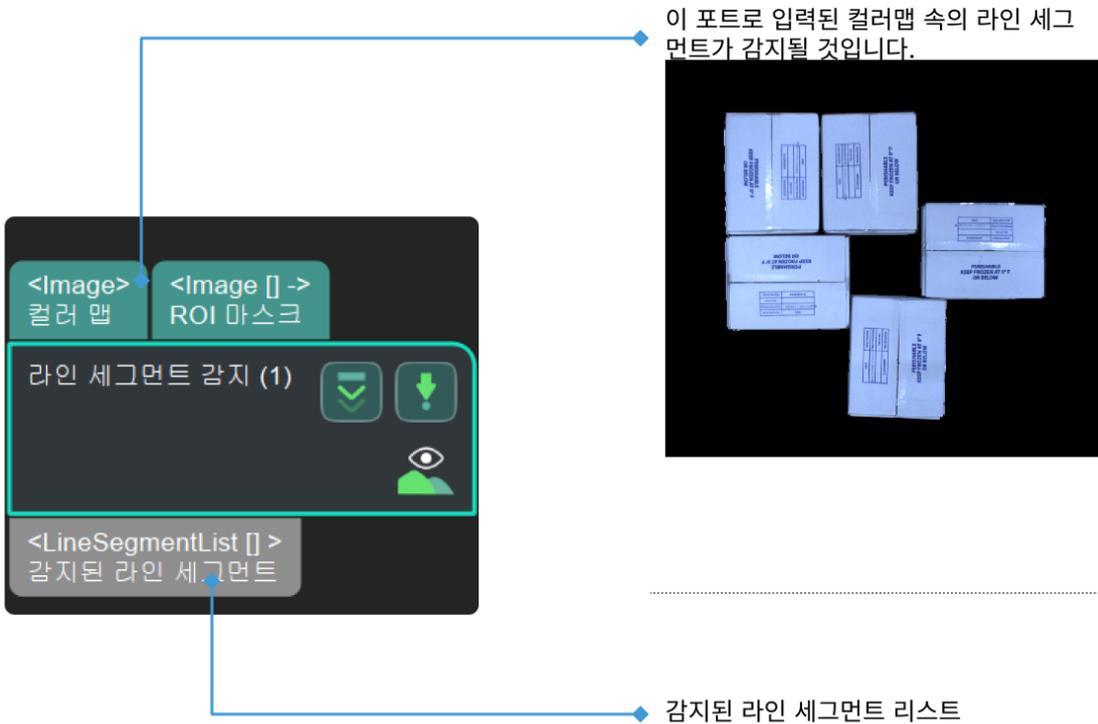
이 스텝은 이미지 속의 라인 세그먼트를 감지합니다.



### 응용 시나리오

이 스텝은 2D 맵에 있는 물체의 라인 세그먼트 특징을 추출하여 후속 매칭에 사용합니다. 2D 매칭 스텝과 함께 사용될 수 있습니다.

### 입력 및 출력



## 자주 사용되는 파라미터 설명

### 라인 측정기 설정

#### 라인 측정기 유형

설명: 이 파라미터는 라인 측정기의 유형을 지정하는 데 사용됩니다.

값 리스트:

- LineSegmentDetector
- FastLineDetector
- EdgeDrawingDetector

기본값: LineSegmentDetector

### 라인 속성

#### 라인 세그먼트 길이 하한 역치

설명: 이 파라미터는 감지된 라인 세그먼트의 최소 길이를 설정하는 데 사용되며 단위는 픽셀(px)입니다. 감지된 라인 세그먼트의 길이가 이 값보다 작으면 라인 세그먼트는 필터링됩니다.

기본값: 10px

권장값: 10px, 20px, 30px

#### 마스크 외의 라인 세그먼트 필터링 여부

설명: 이 파라미터는 마스크 외에 감지된 라인 세그먼트를 필터링할지를 결정하는 데 사용됩니다.

기본값: 선택하지 않음.

조정 제안: 실제 상황에 따라 이 파라미터를 설정해야 합니다.

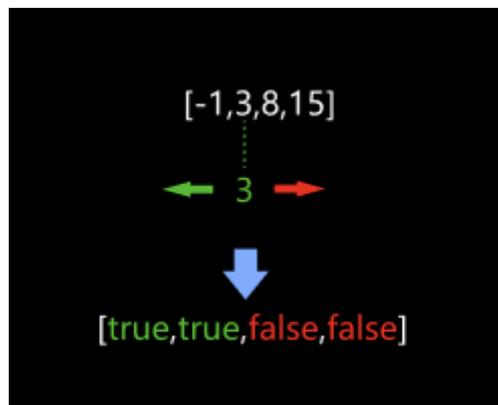
## 4.3.29. 역치에 근거한 이진 분류



이 스텝의 내용은 유지 보수 중입니다. 이 스텝에 대한 자세한 정보가 필요한 경우 ([docs@mech-mind.net](mailto:docs@mech-mind.net))로 문의하십시오.

### 기능 설명

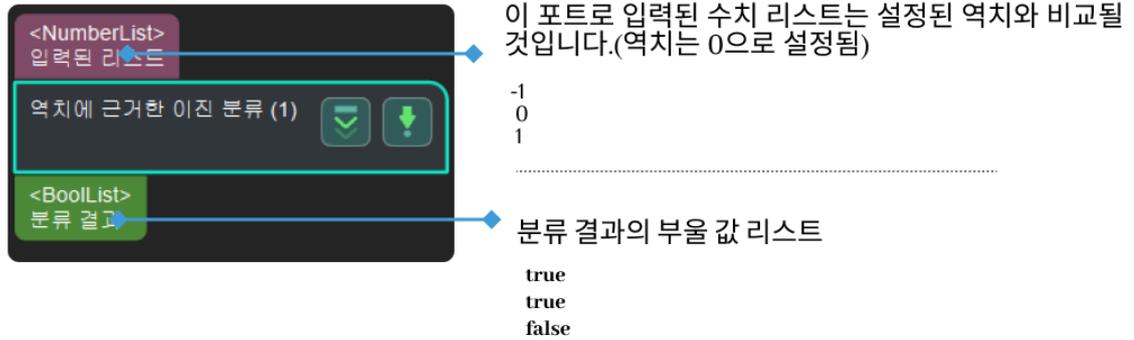
이 스텝은 설정된 역치에 따라 입력한 리스트의 각 숫자 값의 크기를 판단할 수 있습니다.



## 응용 시나리오

값 목록의 값이 해당 역치보다 작거나 같으면 출력한 목록의 해당 항목은 True이고, 그렇지 않으면 False입니다. 이 스텝은 일반적으로 **필터링** 스텝과 함께 사용됩니다.

## 입력 및 출력



### 4.3.30. 포인트 클라우드 다운 샘플링



이 스텝의 내용은 유지 보수 중입니다. 이 스텝에 대한 자세한 정보가 필요한 경우 ([docs@mech-mind.net](mailto:docs@mech-mind.net))로 문의하십시오.

## 기능 설명

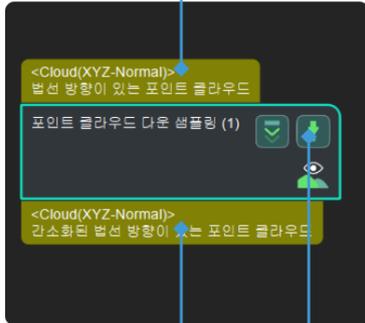
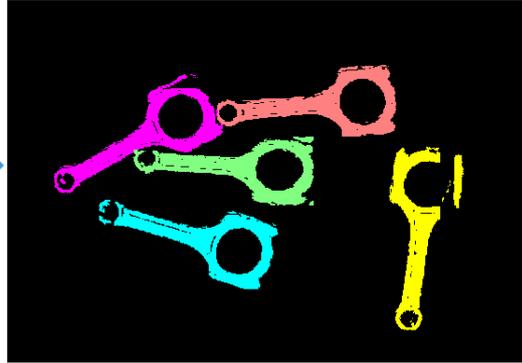
포인트 클라우드에 대해 다운 샘플링하여 포인트의 수를 줄입니다.

## 응용 시나리오

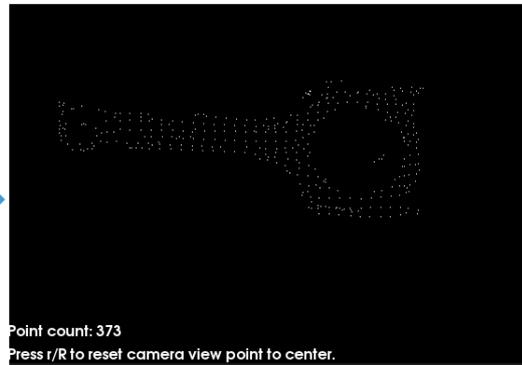
주로 포인트 클라우드 사전 처리 단계에서 사용되며 처리해야 할 포인트 수가 매우 많은 경우에 사용되고 실행 속도를 향상시킬 수 있습니다.

## 입력 및 출력

다운 샘플링 처리될 포인트 클라우드



시각화 출력 결과



다운 샘플링 처리 후의 포인트 클라우드



### 4.3.31. 레이블 리스트를 빠르게 만들기



이 스텝의 내용은 유지 보수 중입니다. 이 스텝에 대한 자세한 정보가 필요한 경우 ([docs@mech-mind.net](mailto:docs@mech-mind.net))로 문의하십시오.

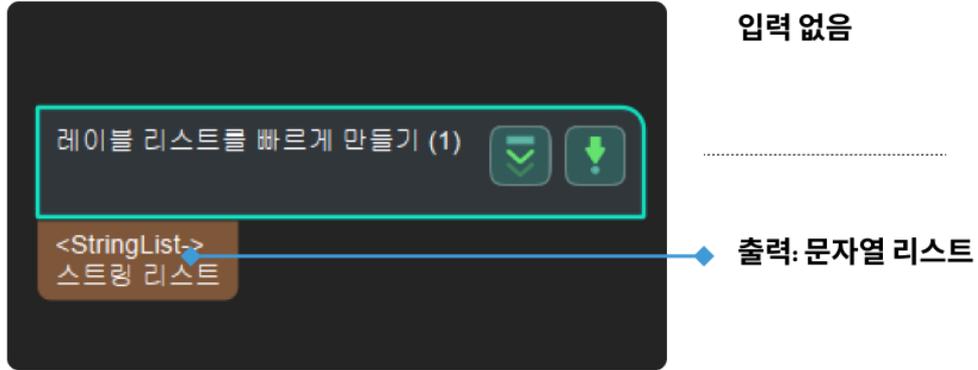
#### 기능 설명

파라미터에 입력된 문자열 리스트를 통해 레이블 리스트를 만듭니다.

### 응용 시나리오

일반적인 레이블 리스트를 만드는 스텝입니다.

### 입력 및 출력



입력 없음

레이블 리스트를 빠르게 만들기 (1)

<StringList>  
스트링 리스트

출력: 문자열 리스트

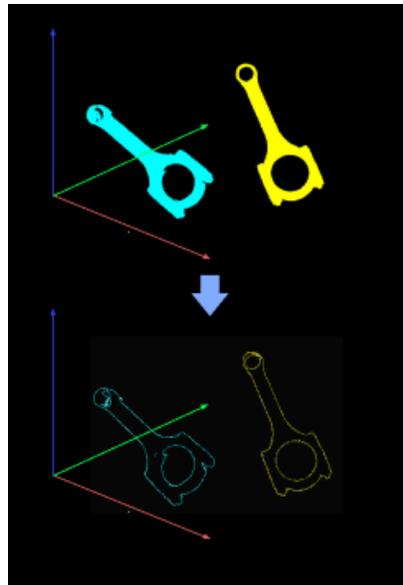
### 4.3.32. 3D 방법을 사용하여 포인트 클라우드의 에지를 추정하기



이 스텝의 내용은 유지 보수 중입니다. 이 스텝에 대한 자세한 정보가 필요한 경우 ([docs@mech-mind.net](mailto:docs@mech-mind.net))로 문의하십시오.

### 기능 설명

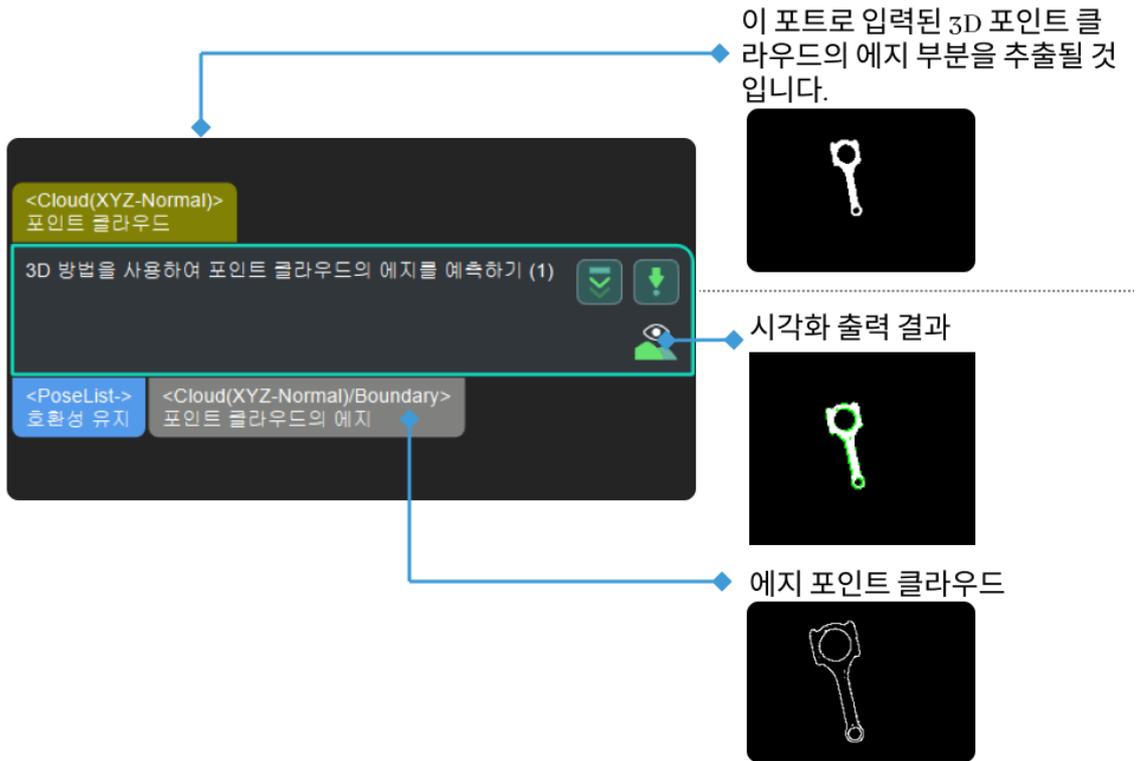
3D 포인트 클라우드의 에지를 추출합니다.



### 응용 시나리오

일반적으로 포인트 클라우드의 에지를 통해 매칭하는 데 사용되고 모든 포인트를 사용하는 것보다 매칭 속도가 더 빠릅니다.

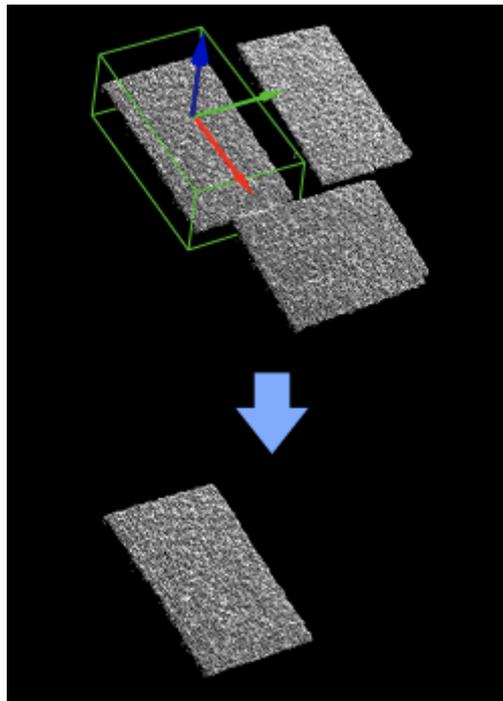
### 입력 및 출력



### 4.3.33. 직육면체 내의 포인트 클라우드를 추출하기

#### 기능 설명

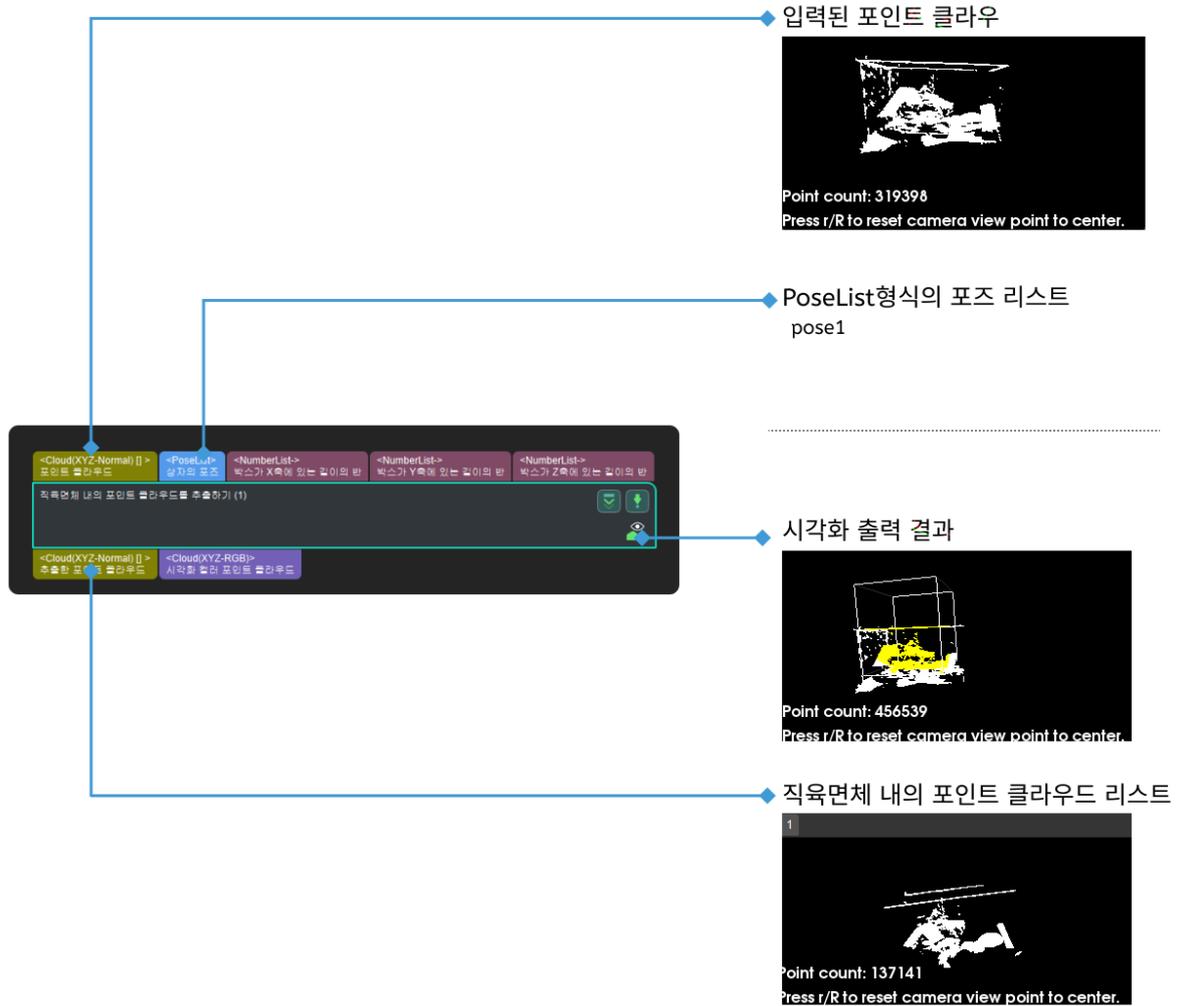
이 스텝은 지정된 직육면체 내의 포인트 클라우드를 추출하고 직육면체 외부의 포인트 클라우드를 제거합니다.



## 응용 시나리오

이 스텝은 특별한 제한 없이 대부분 시나리오에서 사용될 수 있는 포인트 클라우드 추출 스텝입니다. 직육면체의 치수는 포트를 통해 입력하거나 파라미터에서 설정할 수 있습니다.

## 입력 및 출력



## 파라미터 설명

### 상자 크기 설정

#### X축에 있는 상자의 절반 길이

기본값: 100.000

조절 설명: 이 파라미터는 세계 좌표계의 X축에 있는 상자 길이의 절반이며 아래 그림의 **halfX**에 해당합니다. 단위: 밀리미터(mm).

#### Y축에 있는 상자의 절반 길이

기본값: 100.000

조절 설명: 이 파라미터는 세계 좌표계의 Y축에 있는 상자 길이의 절반이며 아래 그림의 **halfY**에 해당합니다. 단위: 밀리미터(mm).

### Z축에 있는 상자의 절반 길이

기본값: 100.000

조절 설명: 이 파라미터는 세계 좌표계의 Z축에 있는 상자 길이의 절반이며 아래 그림의 **halfZ**에 해당합니다. 단위: 밀리미터(mm).

[extract 3d points in cuboid schematic diagram] | *extract-3d-points-in-cuboid-schematic\_diagram.png*

### 입력된 포인트 클라우드의 사용 방법

기본값: OneToOne

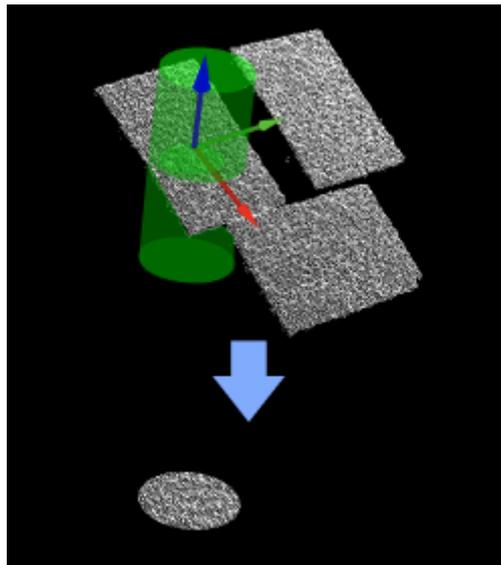
값 리스트: OneToOne, FirstToAll, AllToFirst

조절 설명: 다른 파라미터를 선택하여 입력된 포인트 클라우드의 다른 용법을 채택합니다.

## 4.3.34. 원기둥 범위 내의 포인트 클라우드를 추출하기

### 기능 설명

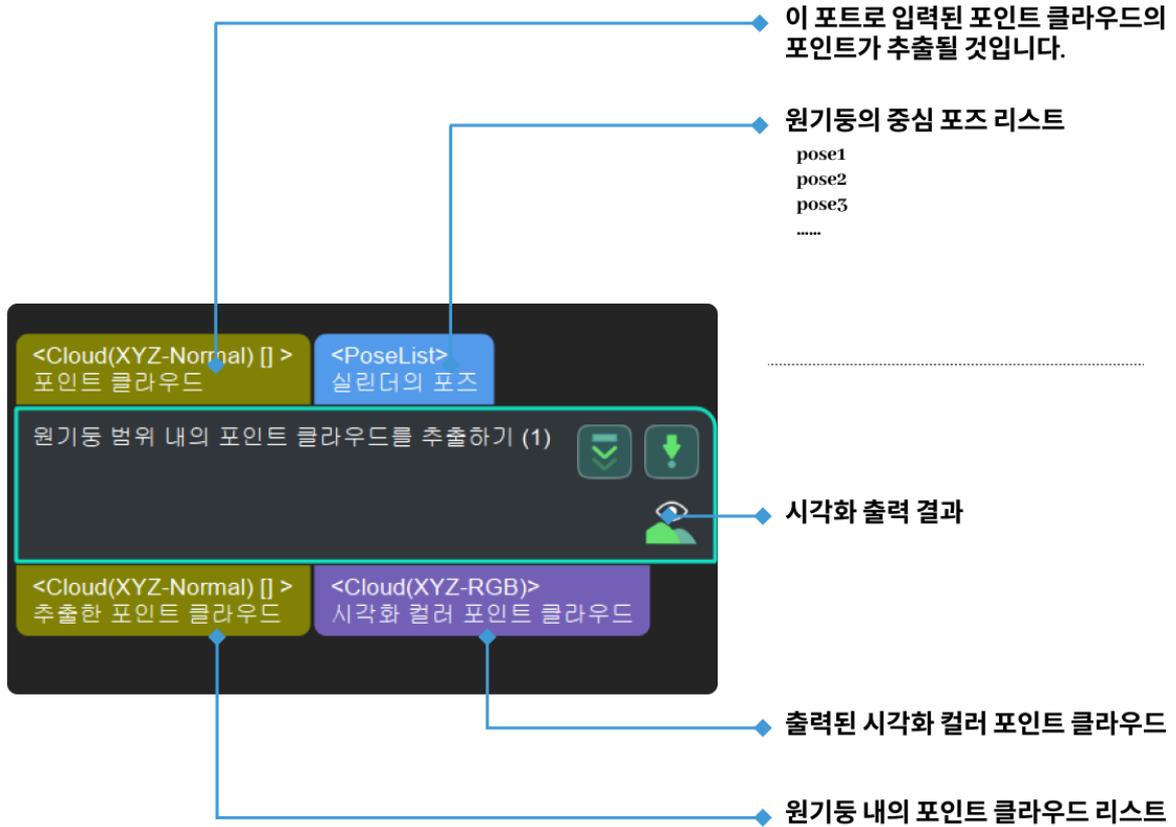
이 스텝은 지정된 원기둥 내의 포인트 클라우드를 추출하고 원기둥 외부의 포인트 클라우드를 버립니다.



### 응용 시나리오

특별한 제한 없이 대부분 시나리오에서 사용될 수 있는 포인트 클라우드 추출 스텝입니다.

### 입력 및 출력



## 파라미터 설명

### 형태 유형

기본값: 속이 빈 원기둥.

값 리스트: 원기둥, 링, 섹터.

조절 설명: 다양한 시나리오의 필요에 따라 다양한 모양의 포인트 클라우드를 추출할 수 있습니다. 모양을 선택하면 지정된 모양 내의 3D 포인트 클라우드가 추출됩니다.

### Z 축 파라미터

#### Z 값 하한

기본값: -100.000mm

조절 설명: 이 하한은 Z축의 최소 좌표이며 아래 그림의 **Z**-에 해당합니다. 단위는 밀리미터(mm)입니다.

#### Z 값 상한

기본값: 100.000mm

조절 설명: 이 상한은 Z축의 최소 좌표이며 아래 그림의 **Z**-에 해당합니다. 단위는 밀리미터(mm)입니다.

[extract 3d points in cylinder schematic diagram] | *extract-3d-points-in-cylinder-schematic\_diagram.png*

### 링과 부채꼴 반경 파라미터

### 내부 원 반경

기본값: 50.000mm

조절 설명: XY 평면의 최소 거리는 위 그림의 **R1**에 해당합니다. 단위는 밀리미터(mm)입니다. 이 파라미터를 **0**으로 설정하면 원기둥을 얻을 수 있습니다.

### 외부 원 반경

기본값: 100.000mm

조절 설명: XY 평면의 최대 거리는 위 그림의 **R2**에 해당합니다. 단위는 밀리미터(mm)입니다.

### 부채꼴 각도 파라미터

**형태 유형**에서 **Sector**를 선택한 경우에 다음 두 가지 파라미터를 통해 부채꼴의 형태를 설정할 수 있습니다.

#### 시작 각도

기본값: -180°

조절 설명: 이 파라미터는 부채꼴이 XY 평면에 있는 시작 각도입니다.

#### 끝 각도

기본값: 180°

조절 설명: 이 파라미터는 부채꼴이 XY 평면에 있는 끝 각도입니다.

### 입력된 포인트 클라우드의 용법 설정

#### 포인트 클라우드의 중심을 기준으로 원기둥 내부에 물체가 있는지를 판단하기

기본값: 선택하지 않음.

설명: 선택하면 대상 포인트 클라우드의 중심을 사용하여 물체가 원기둥 내에 있는지를 확인하며 선택하지 않으면 확인하지 않을 것입니다.

#### 입력된 포인트 클라우드의 용법

기본값: All

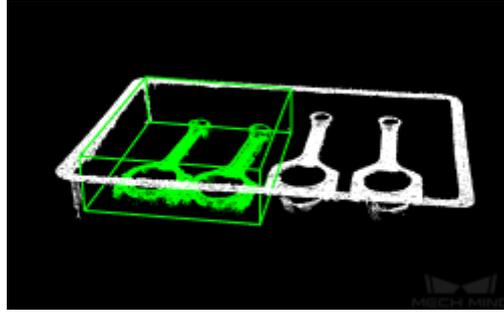
값 리스트: All, FirstOnly

설명: 다른 파라미터 수치를 설정함으로써 입력된 포인트 클라우드의 다양한 용법을 바꿀 수 있습니다.

## 4.3.35. 3D ROI 내의 포인트 클라우드를 추출하기

### 기능 설명

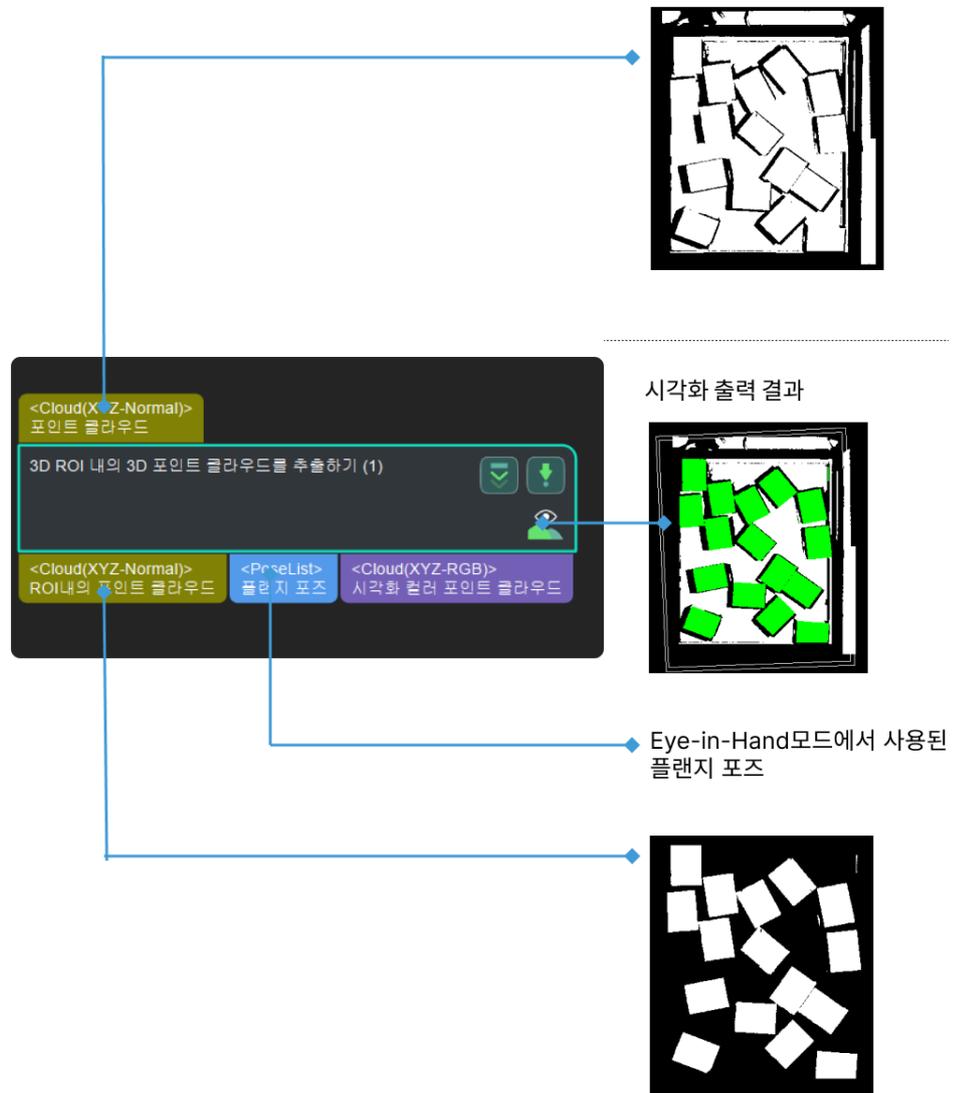
3D 공간에서 ROI를 설정하고 ROI 내부에 있는 포인트 클라우드는 유지되고 외부에 있는 포인트 클라우드는 제거될 것입니다.



### 응용 시나리오

이 스텝을 통해 대상 물체의 포인트 클라우드에만 집중하고 배경 포인트 클라우드 및 이상치의 간섭을 받지 않습니다.

### 입력 및 출력

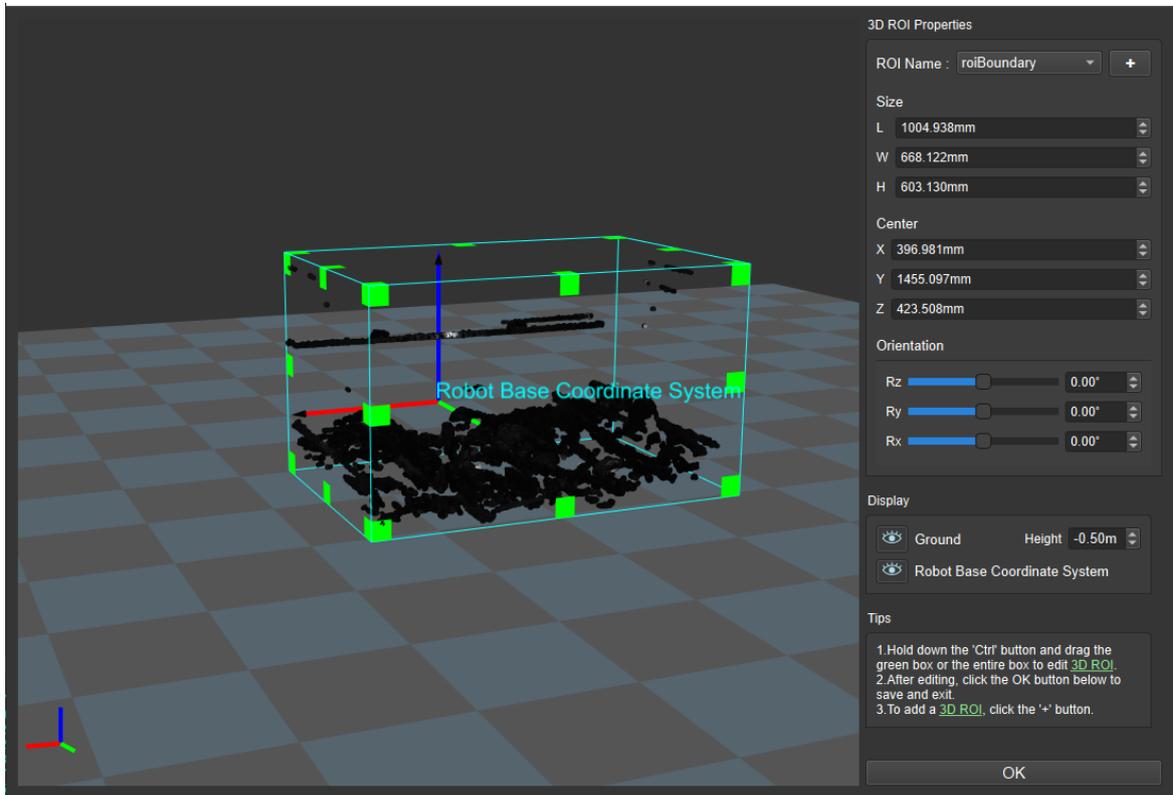


## 파라미터 설명

### 3D ROI 설정

#### 3D ROI

조정 설명: [ 3D 관심 영역(ROI) 설정 ]을 클릭하여 아래 그림과 같이 대상 영역 설정 창으로 들어갑니다. 자세한 설정은 3D ROI 설정 프로세스를 참고하십시오.

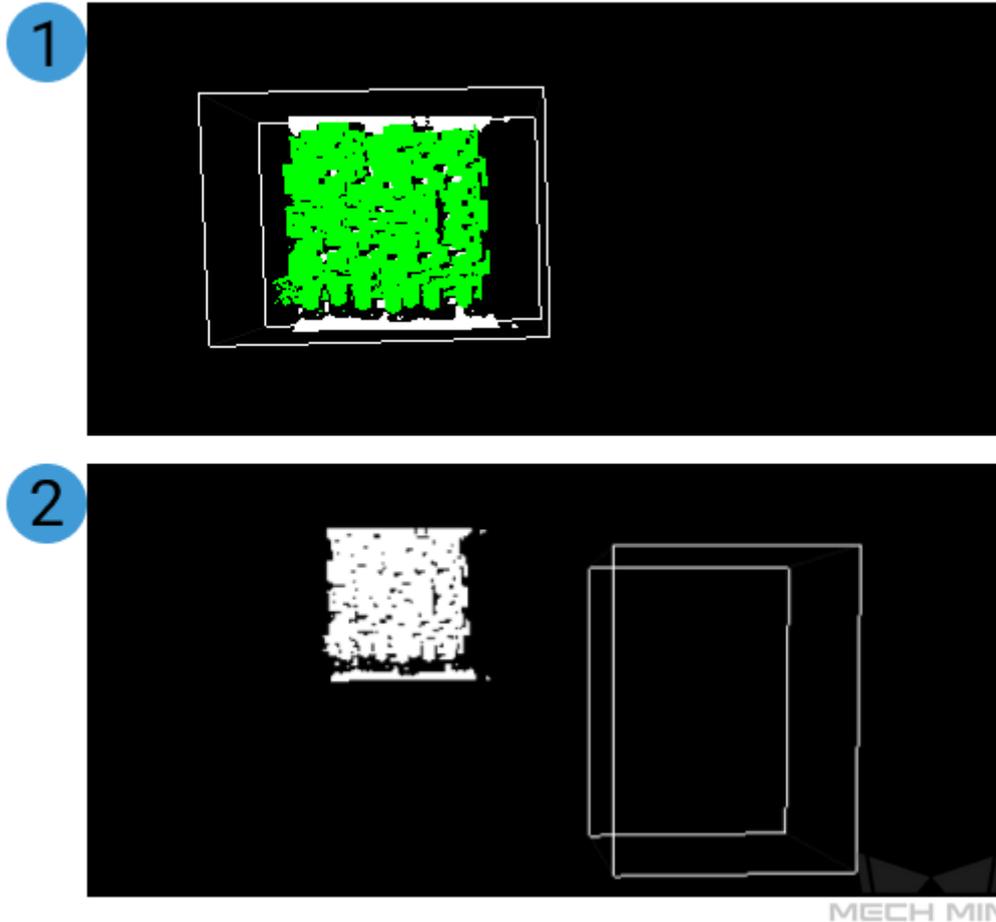


#### 입력된 포인트 클라우드 좌표계의 유형

기본값: 카메라 좌표계

값 리스트: 카메라 좌표계, 로봇 좌표계

조정 설명: 입력된 포인트 클라우드가 있는 좌표계를 변경합니다. 아래 그림과 같이 그림 1은 카메라 좌표계에서 입력한 포인트 클라우드이며 그림 2는 로봇 좌표계에서 입력한 포인트 클라우드입니다.



## 빈 상태 설정

### 3D ROI에 있는 최소 포인트 수

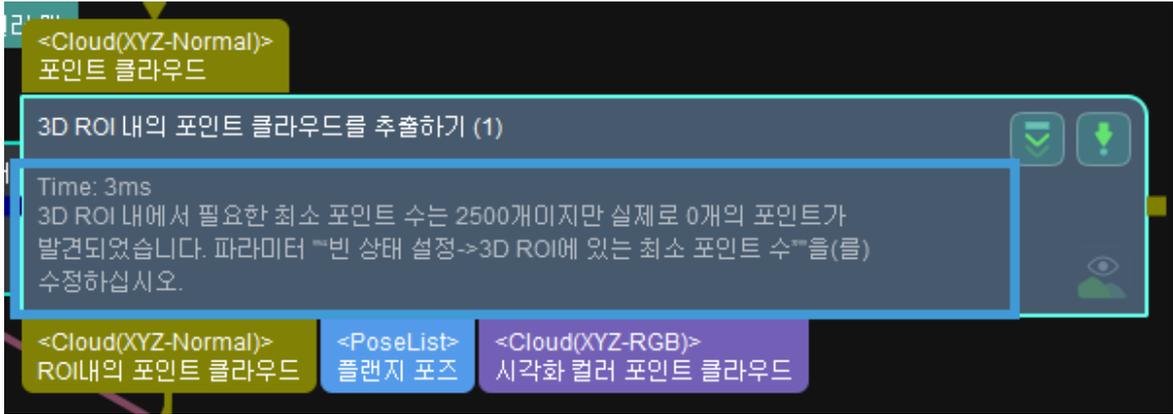
기본값: 0

조정 설명: 이 파라미터는 3D ROI에서 추출할 수 있는 3D 포인트 수의 최소값을 조정합니다. 3D ROI에서 추출한 3D 포인트의 수가 이 수치보다 작으면 선택한 3D ROI 내의 포인트의 수가 최소 포인트 수를 충족하지 못했음을 나타내며 이때 포인트 클라우드를 반환하지 않습니다.

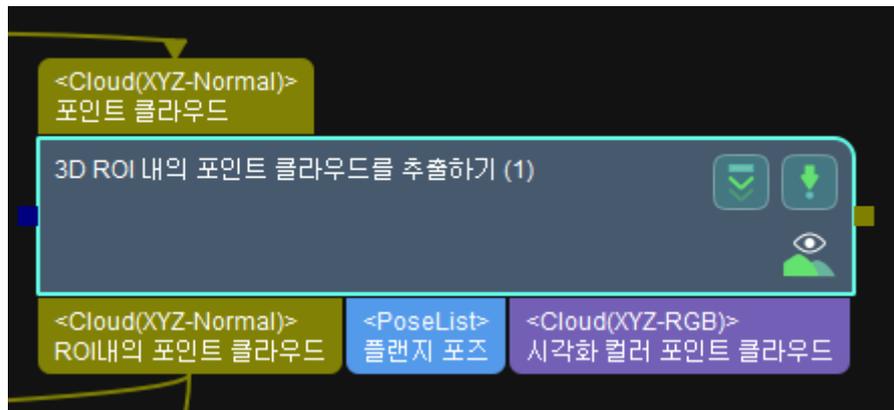
### 3D ROI에 있는 포인트가 비어 있는지를 전송하기

기본값: 선택함

조정 설명: 이 옵션을 선택하면 3D ROI 안에 있는 포인트가 비어 있는지를 전송합니다. 기본적인 상태에서 3D ROI 내의 실제 포인트 수가 “3D ROI 내의 최소 포인트 수”보다 작으면 아래 그림과 같은 알람이 나타날 것입니다.



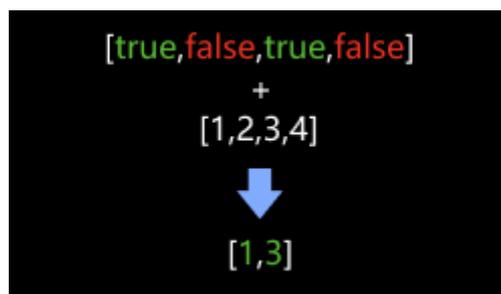
이 옵션을 선택하지 않으면 알림이 나오지 않습니다. 아래 그림과 같습니다.



### 4.3.36. 필터링

#### 기능 설명

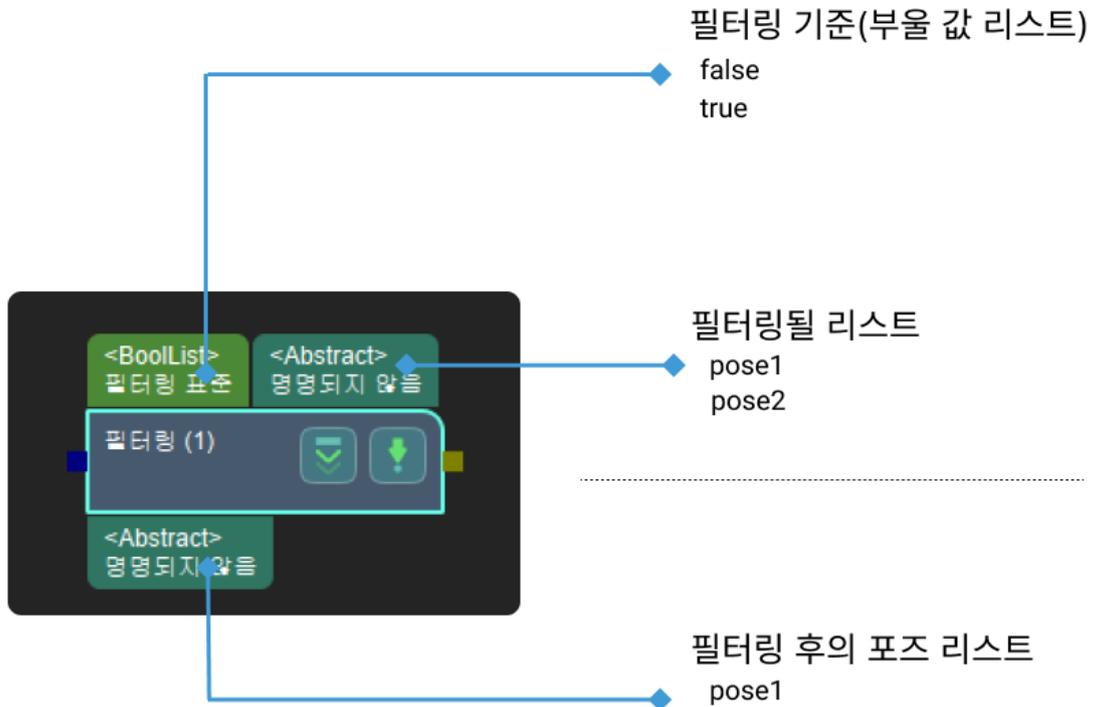
이 스텝은 일반적인 배열 필터입니다. 입력된 배열을 True/False 리스트와 각각 대응하여 True와 대응하는 데이터를 출력합니다.



#### 응용 시나리오

부울 값 리스트를 사용하여 입력한 데이터를 필터링하며, 일반적으로 이 스텝은 역치에 근거한 이진 분류, 필요한 레이블인지 판단하기, 기준 방향과의 각도에 근거하여 포즈 유효성을 판단하기, 포인트 클라우드가 요구에 맞출 수 있는지 확인하기 등 부울 값을 출력하는 스텝과 함께 사용됩니다.

#### 입력 및 출력



## 파라미터 설명

### 포트 설정

#### 포트 수(1~15)

기본값: 1

조절 범위: 1~15

설명: 사용자가 자체 정의한 포트 수량.

### 작업층

기본값: 0

조절 범위: 0~14

설명: 작업층이 0이면 리스트에 있는 모든 요소에 대한 작업을 수행합니다.

### 부울 값을 되돌리기

기본값: 선택하지 않음.

조절 설명:

- 선택하지 않음: 입력한 부울 값 리스트에서 부울 값이 false인 항목을 제거할 것입니다.
- 선택함: 입력한 부울 값 리스트에서 부울 값이 true인 항목을 제거할 것입니다.

조절 예시:

부울 값 및 입력 정보는 다음 그림과 같습니다.

```

BoolList:
Size:3
false,
true,
true
    
```



부울 값을 되돌리기 옵션을 선택하지 않을 때 입력한 부울 값 리스트에서 부울 값이 false인 항목이 제거되었습니다.



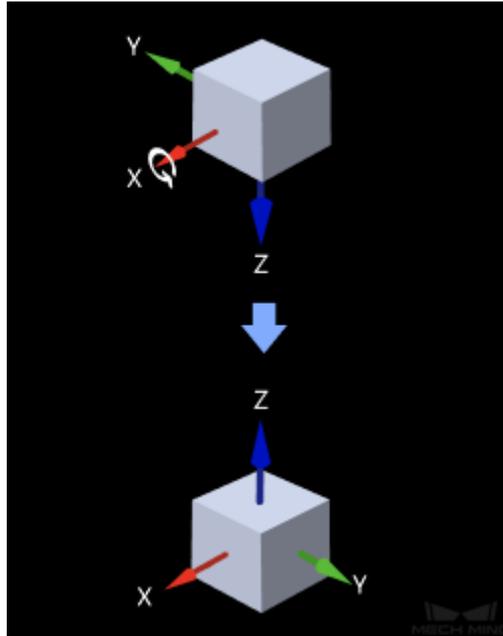
부울 값을 되돌리기 옵션을 선택할 때 입력한 부울 값 리스트에서 부울 값이 true인 항목이 제거되었습니다.



### 4.3.37. 포즈 좌표축의 방향을 뒤집기

#### 기능 설명

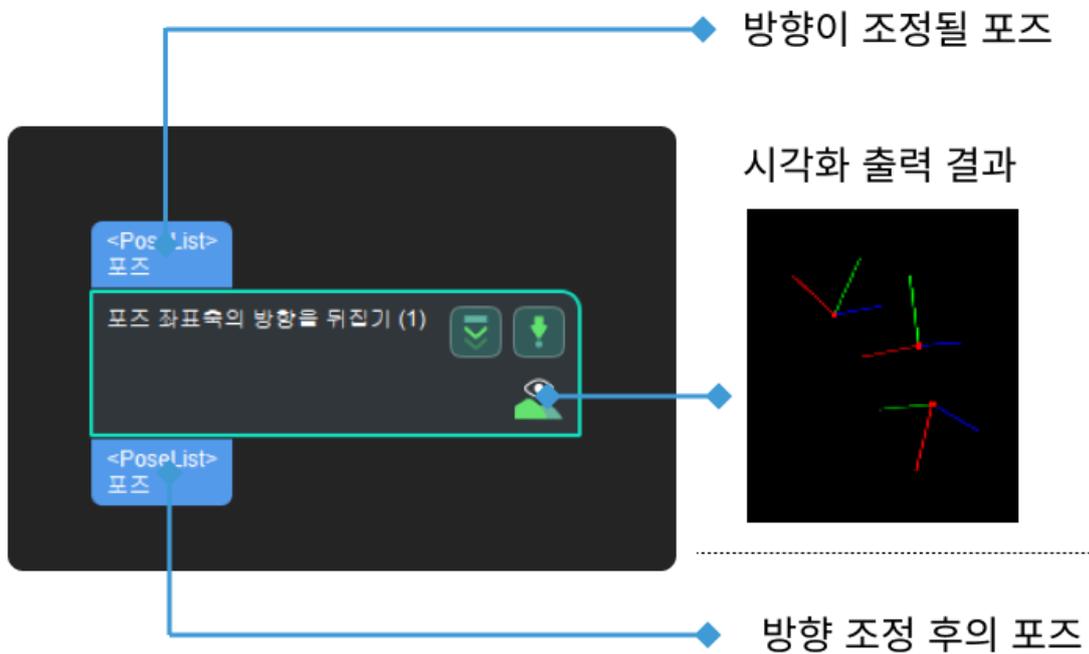
방향을 정하고 이 스텝을 통해 입력된 포즈의 축 방향을 조정합니다.



### 응용 시나리오

이 스텝은 일반적으로 포즈 후속 처리 단계에 사용되고 모든 포즈의 어떤 좌표축 지향을 동일하게 만듭니다.

### 입력 및 출력



### 파라미터 설명

#### 시각화 설정

#### 포즈 시각화 설정

기본값: 출력한 포즈만 표시하기.

값 리스트: 입력한 포즈만 표시하기, 출력한 포즈만 표시하기, 입력 및 출력한 포즈를 모두 표시하기.

설명: 처리 후의 포즈만 표시하기(기본값), 처리 전의 포즈만 표시하기, 처리하기 전/후의 포즈를 모두 표시하기.

## 축 설정

### 뒤집히게 될 축

기본값: Z

값 리스트: X, Y, Z

설명: 뒤집히게 될 좌표축 유형을 선택합니다.

### 방향의 유형

기본값: 부방향

값 리스트: 부방향(포즈의 축을 세계 좌표계의 부방향으로 조정하는 경우, 부방향과의 협각이 90도 보다 큰 축이 조정될 것임). 정방향(포즈의 축을 세계 좌표계의 정방향으로 조정하는 경우, 정방향과의 협각이 90도 보다 큰 축이 조정될 것임).

설명: 뒤집기 대기 중인 축의 방향. 예: 파라미터 **뒤집히게 될 축**의 유형이 **Z**인 경우 방향의 유형은 **정방향**이며 이때 세계 좌표계 정방향과의 협각이 90도보다 작은 포즈의 Z 축이 조정되지 않으며 90도 보다 큰 포즈의 Z 축은 기준축을 중심으로 180도로 조정될 것입니다.

### 회전할 때의 기준 축

기본값: X

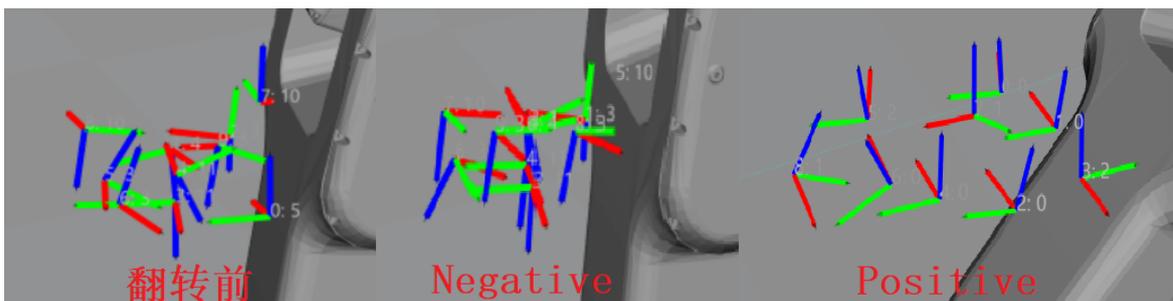
값 리스트: X, Y, Z

설명: 어느 축을 중심으로 회전하는지 선택합니다.



뒤집어야 하는 축과 회전 기준 축은 동일한 축이 될 수 없습니다.

예시: 여기서 Z축을 뒤집히게 될 축으로 선택하고, X축을 회전축으로 선택하는 경우는 다음 그림과 같습니다.



## 4.3.38. 실제 치수를 픽셀 치수로 전환하기

### 기능 설명

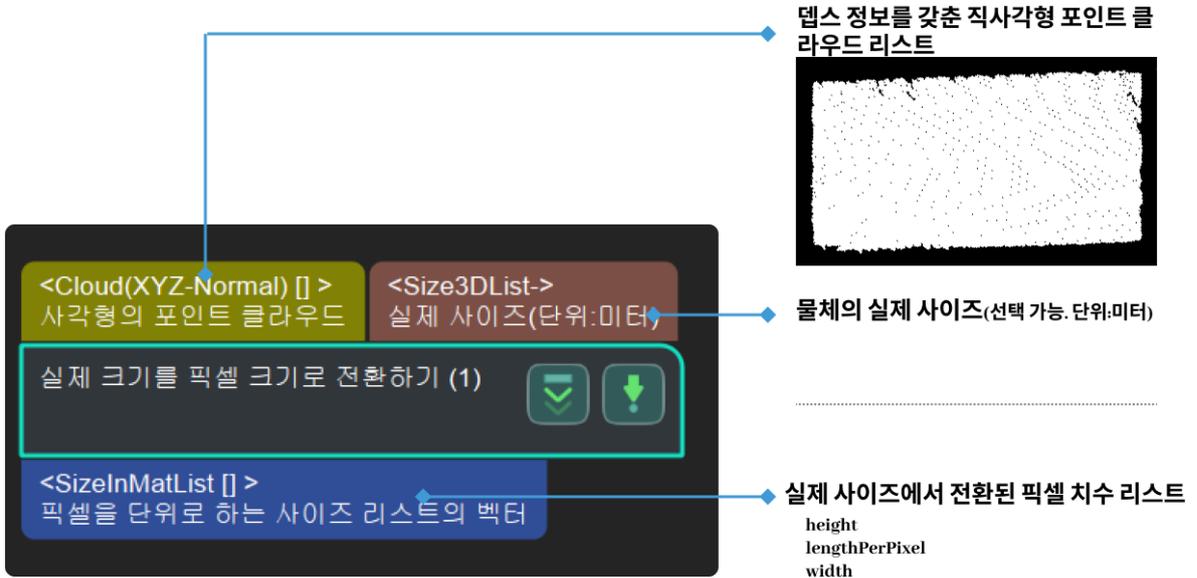
직사각형 포인트 클라우드의 뎀스 정보와 실제 사이즈를 제공하여 대응하는 물체의 픽셀을 단위로 하는 치수를 계산합니다.

### 응용 시나리오

일반적으로 스텝 **지정된 크기의 직사각형 에지 템플릿을 생성하기**와 함께 사용하여 후속 2D 매칭에 에지

템플릿을 제공합니다.

### 입력 및 출력



### 파라미터 설명

#### 해당 설정

#### 해당 유형

파라미터 설명: 이 파라미터는 입력된 치수와 실제 포인트 클라우드 치수의 해당 유형을 선택하는 데 사용됩니다.

값 리스트: OneActualSizeWithAllPointCloud, OneActualSizeWithOnePointCloud

- OneActualSizeWithAllPointCloud : 입력된 첫 번째 실제 치수는 모든 포인트 클라우드에 적용할 수 있습니다.
- OneActualSizeWithOnePointCloud : 입력된 치수는 실제 포인트 클라우드와 일대일로 대응합니다.

기본값: OneActualSizeWithAllPointCloud

조절 제안: 이 파라미터는 실제 필요에 따라 설정해야 합니다.

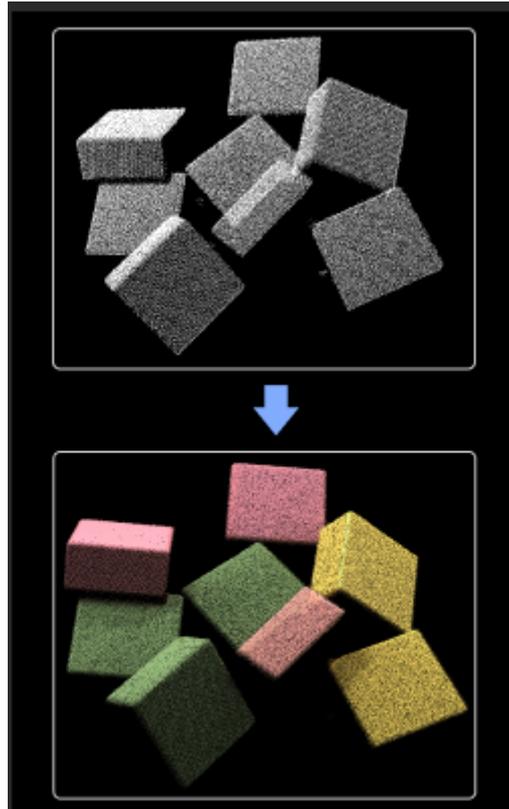
### 4.3.39. Cloud (XYZ-Normal)를 Cloud (XYZ-RGB)로 전환하기



이 스텝의 내용은 유지 보수 중입니다. 이 스텝에 대한 자세한 정보가 필요한 경우 ([docs@mech-mind.net](mailto:docs@mech-mind.net))로 문의하십시오.

#### 기능 설명

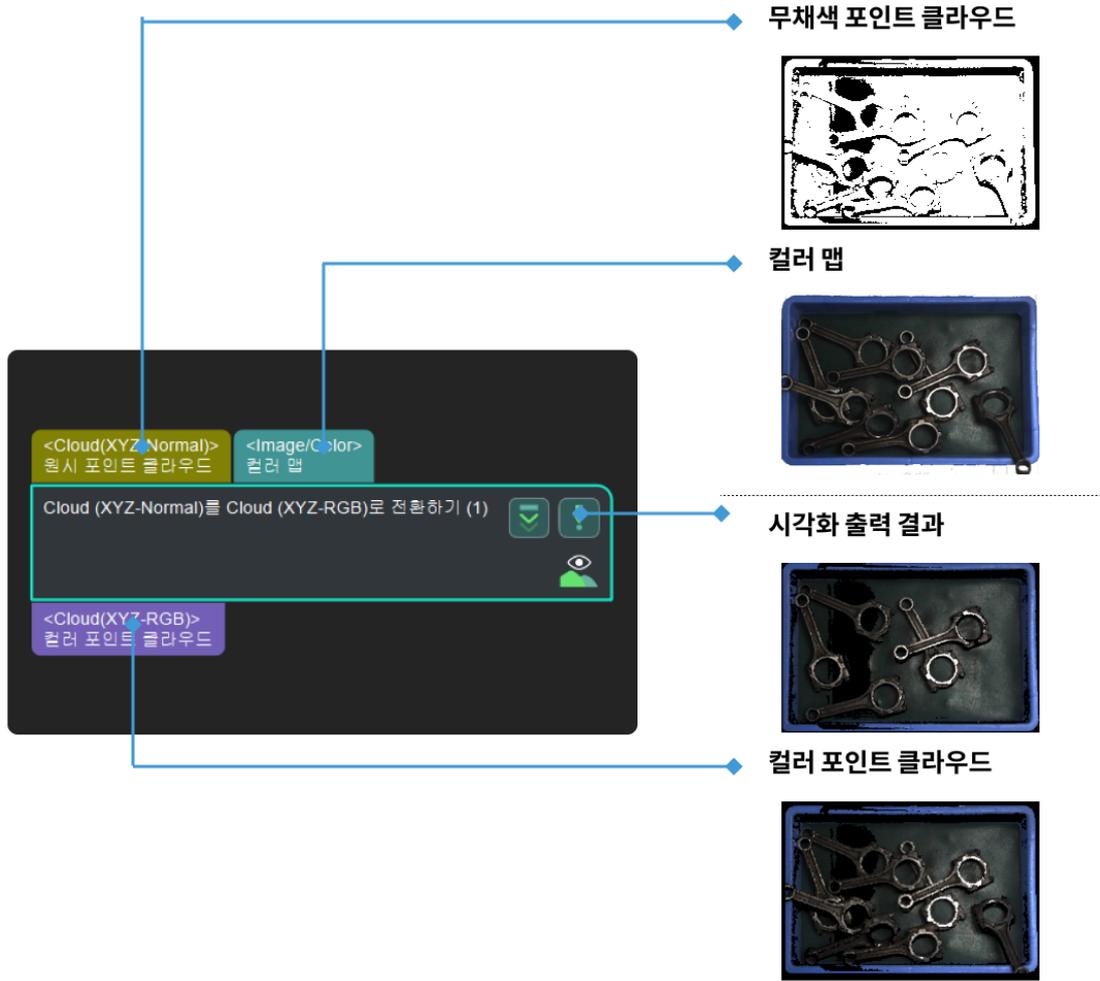
컬러 맵을 사용하여 법선 방향이 있는 무채색 포인트 클라우드를 컬러 포인트 클라우드로 전환합니다.



### 응용 시나리오

일반적으로 시뮬레이션 시나리오에서 컬러 포인트 클라우드의 효과를 보거나 컬러 포인트 클라우드 시각화가 필요한 다른 사용 시나리오에 사용됩니다.

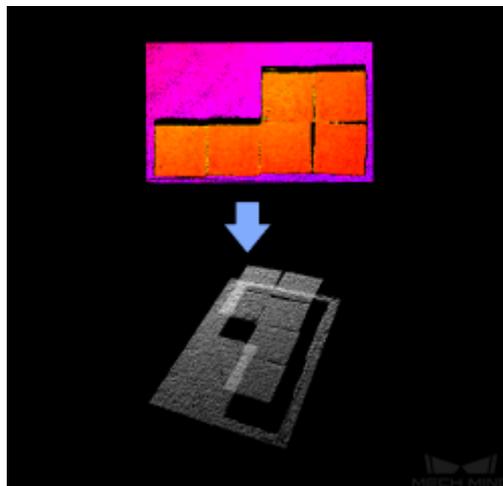
### 입력 및 출력



### 4.3.40. 뎀스 맵을 포인트 클라우드로 전환하기

#### 기능 설명

이 스텝은 뎀스 맵과 컬러 맵을 통해 포인트 클라우드를 생성할 수 있습니다.



유사한 스텝 비교: 카메라에서 이미지를 캡처하기도 직접 포인트 클라우드와 컬러 포인트 클라우드를

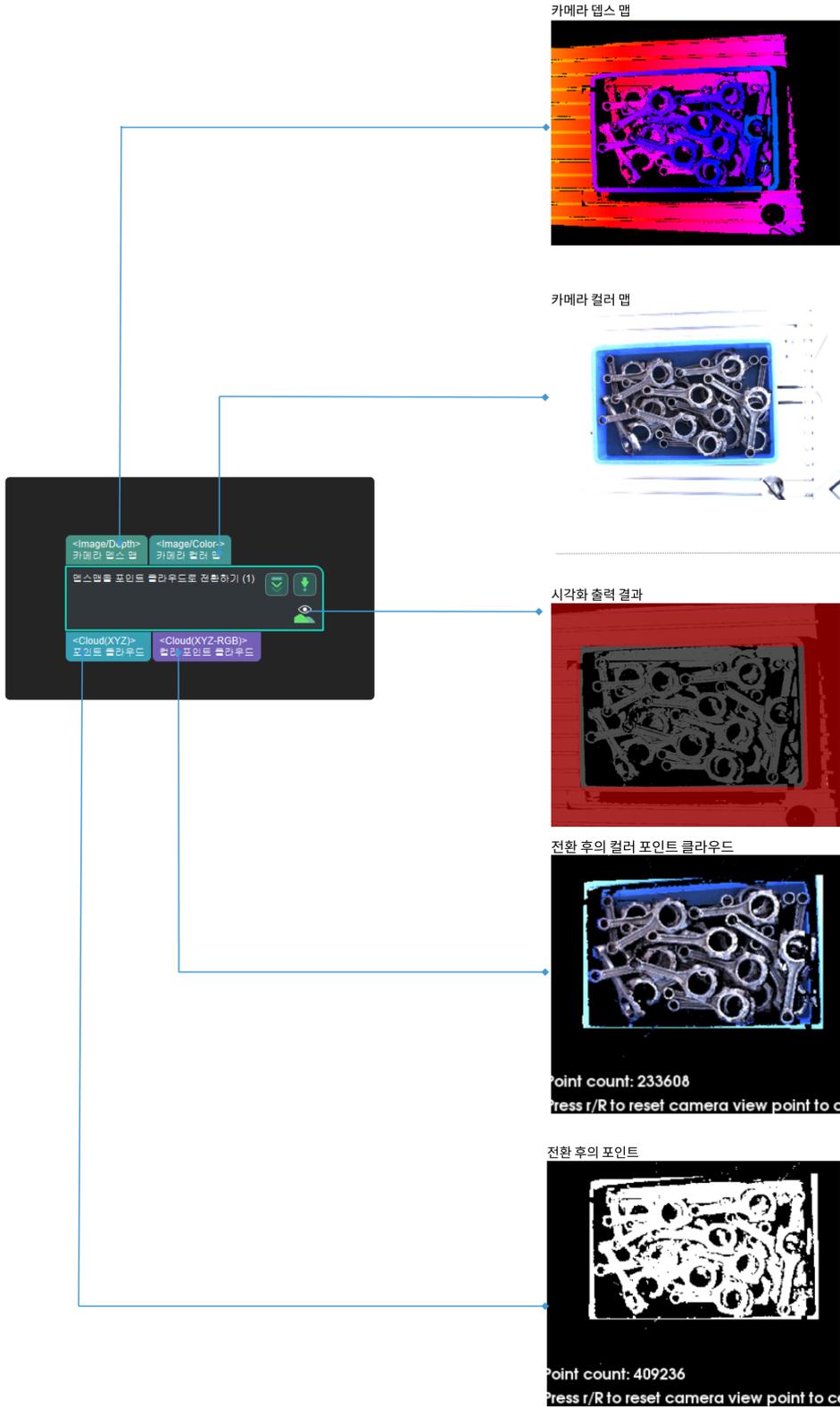
출력할 수 있습니다. 그러나 프로젝트 실행 속도를 향상시키기 위해 일반적으로 **덱스 맵을 포인트 클라우드로 전환하기** 스텝을 포인트 클라우드 및 컬러 포인트 클라우드를 획득하는 방법으로 사용합니다.

**덱스 맵을 포인트 클라우드로 전환하기** 스텝을 사용해 ROI를 선택하여 불필요한 포인트 클라우드 수를 줄임으로써 실행 속도를 높입니다.

## 응용 시나리오

보통 “카메라에서 이미지를 캡처하기” 스텝 뒤에 사용되고 카메라에서 캡처된 덱스 맵을 포인트 클라우드로 전환하며 ROI를 통해 불필요한 포인트 클라우드를 제거할 수 있어서 실행 속도를 높일 수 있습니다.

## 입력 및 출력

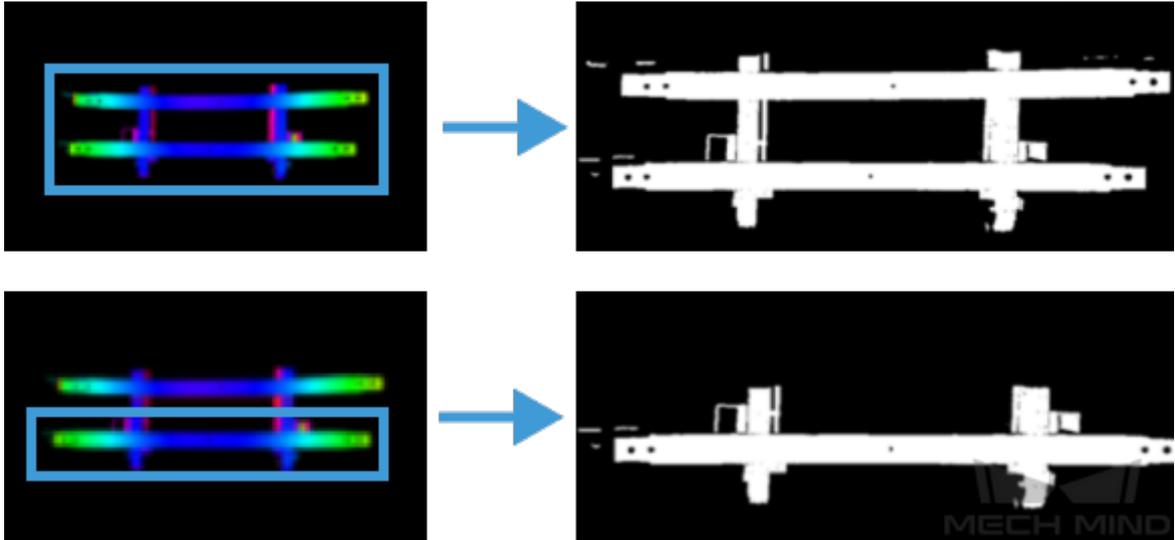


**파라미터 설명**

**뎀스 맵 ROI 파일**

조정 설명: 뎀스 맵의 ROI를 선택합니다.

조정 예시: 다양한 ROI를 선택하고 비교합니다.



위 그림과 같이 왼쪽 이미지는 뎀스 맵의 ROI이며 오른쪽 이미지는 출력된 포인트 클라우드입니다.

조정 스텝:

1. 뎀스 맵 ROI 파일 오른쪽에 있는 [ 2D 관심 영역 설정(ROI) ] 버튼을 클릭하여 ROI를 설정하기 화면으로 들어갑니다.
2. 화면에서 ROI를 선택합니다.
3. [ 확인 ]을 클릭하여 설정을 저장합니다. 동시에 프로젝트의 루트 디렉터리 아래에서 depth\_image\_roi.json 파일이 생성됩니다.

## 배경 제거 설정

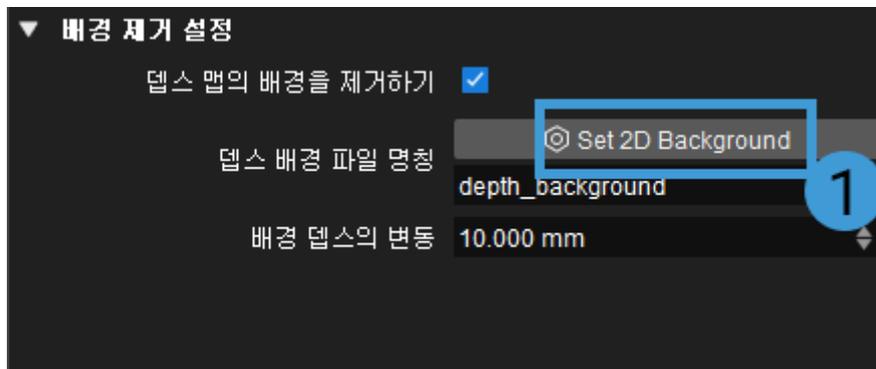
### 뎀스 맵으로 배경을 제거하기

기본값: 선택하지 않음. 설명: 뎀스 맵의 배경을 제거하기를 선택할 때만 파라미터를 설정할 수 있습니다.

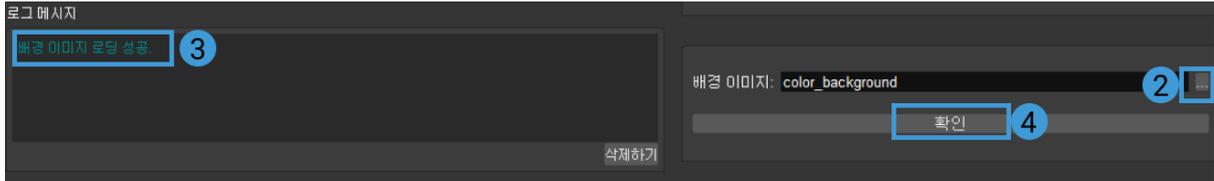
### 배경 뎀스 맵

스텝: 배경 뎀스 맵을 설정하는 스텝은 다음과 같습니다.

배경 뎀스 맵 우측의 [ 2D 배경 설정 ] 버튼을 클릭하여 배경 설정 창으로 이동합니다.



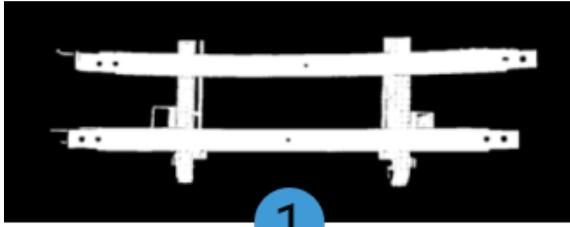
[ ... ]를 클릭하여 이미 준비한 배경 이미지를 선택하십시오. 로그 메시지에서 **배경 이미지 로딩 성공**이란 알림이 나타나면 [ 확인 ]을 클릭하십시오.



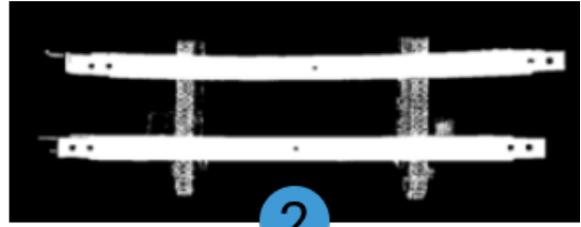
실제 카메라를 연결하여 작업 현장의 실제 배경 이미지를 캡처할 수도 있습니다.

### 배경 뎀스의 변동

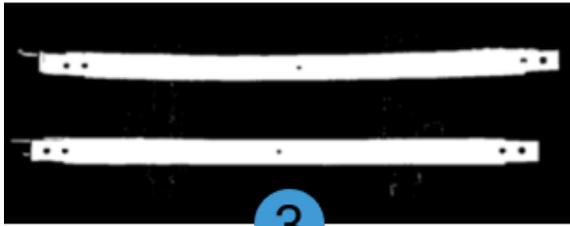
기본값: 10.000 mm 설명: 입력된 뎀스 맵과 배경 뎀스 맵을 비교하면, 뎀스 차이가 변동 범위 내에 있는 배경이 제거될 것입니다. 예시: 아래 그림에서 그림1은 배경 제거를 설정하지 않은 것이고 그림 2, 3, 4가 각각 배경 제거로 설정되어 있고 배경 뎀스 변동은 1mm, 10mm, 30mm로 설정되어 있을 때의 조정 효과입니다.



1



2



3



4

MECH MIND

알 수 있듯이, 배경 뎀스의 변동 값이 너무 작으면 배경이 완전히 제거되지 않을 수 있습니다. 배경 뎀스의 변동 값이 너무 크면, 작업물 포인트 클라우드도 제거될 수 있습니다. 그러므로 배경 뎀스의 변동은 현장 상황을 근거로 하여 적합한 수치 설정이 필요합니다. 일반적으로 10mm를 추천합니다.

### 4.3.41. PoseList를 PoseLists로 전환

#### 기능 설명

포즈 리스트의 각 포즈를 한 요소의 포즈 리스트로 전환합니다.

#### 응용 시나리오

일반적인 데이터 구조 전환 스텝입니다. 예를 들어 스텝 3D 근사 매칭과 3D 상세 매칭 사이, 혹은 평면 포인트 클라우드의 포즈와 크기를 계산하기와 필터링 사이 전환하는 데 사용됩니다.

#### 입력 및 출력



#### 4.3.42. Variant를 VariantList로 전환하기

##### 기능 설명

입력을 Variant에서 VariantList로 전환합니다.

##### 응용 시나리오

일반적인 데이터 유형 변환 스텝입니다. 예를 들어 [데이터 리스트에서 지정된 차원의 요소 개수를 통계하기](#) 및 [VariantList를 NumberList로 전환하기](#) 스텝 사이에 사용되어 요소 개수를 수치 리스트의 형식으로 전환하여 출력하는 목적을 달성합니다.

##### 입력 및 출력



### 4.3.43. 지정된 크기의 직사각형 에지 템플릿을 생성하기

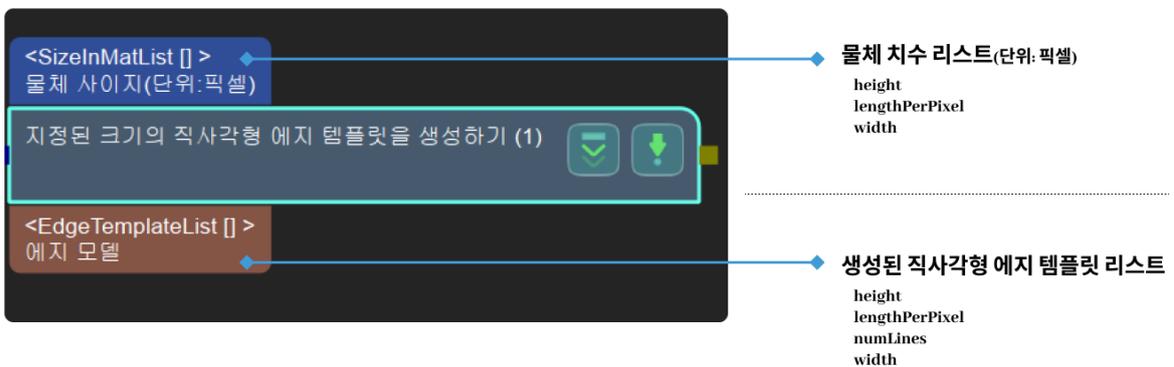
#### 기능 설명

물체의 치수(단위: 픽셀)를 통해 직사각형 에지 모델 리스트를 생성합니다. 일반적으로 스텝 2D 매칭 앞에 사용됩니다.

#### 응용 시나리오

고정적으로 스텝 실제 크기를 픽셀 크기로 전환하기와 함께 사용되어 스텝 2D 매칭을 위해 에지 템플릿을 생성합니다.

#### 입력 및 출력



#### 파라미터 설명

##### 템플릿 속성

##### 템플릿 대각선 양자화의 수

파라미터 설명: 이 파라미터는 템플릿의 대각선 양자화 수를 컨트롤하는 데 사용되며, 직사각형 템플릿의 대각선 길이에서 템플릿 대각선 양자화의 수를 나누어 윤곽선의 양자화 픽셀 수를 획득합니다.

기본값: 10

권장값: [10, 15]

유효 범위 : [0,+∞)

### 4.3.44. 첫 번째 이미지를 획득하기



이 스텝의 내용은 유지 보수가 중단되었습니다. 새 버전의 스텝 출력 수 제한, 데이터 압축 풀기 조합을 사용하십시오. 이 스텝에 대한 자세한 정보가 필요한 경우 ([docs@mech-mind.net](mailto:docs@mech-mind.net))로 문의하십시오.

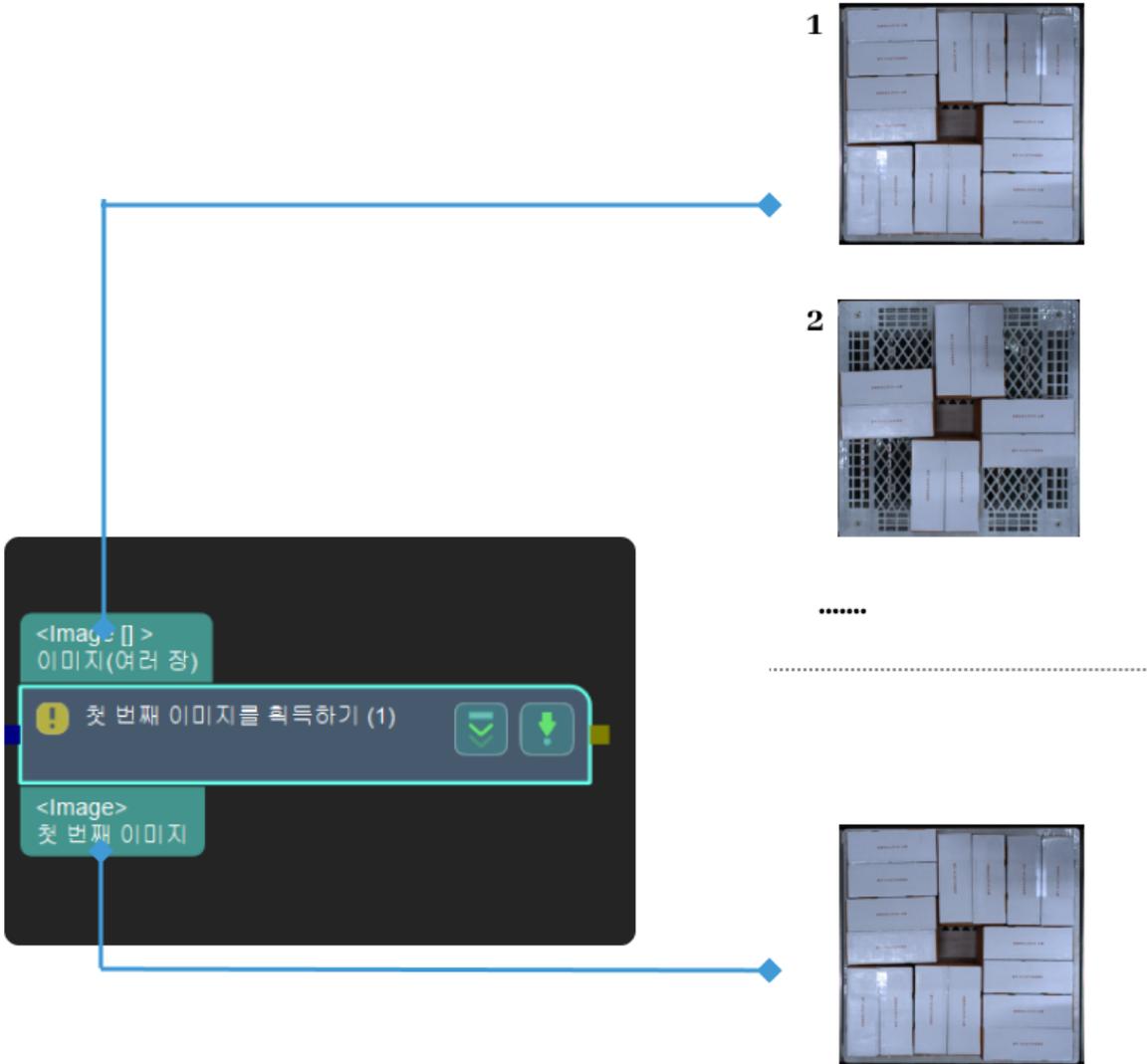
#### 기능 설명

이미지 리스트에서 첫 번째 이미지를 가져옵니다.

**응용 시나리오**

이 스텝은 구 버전입니다. 위의 새 버전 스텝을 사용하는 것이 좋습니다.

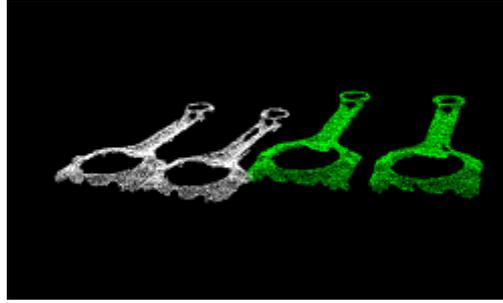
**입력 및 출력**



**4.3.45. 가장 높은 층의 포인트 클라우드를 획득하기**

**기능 설명**

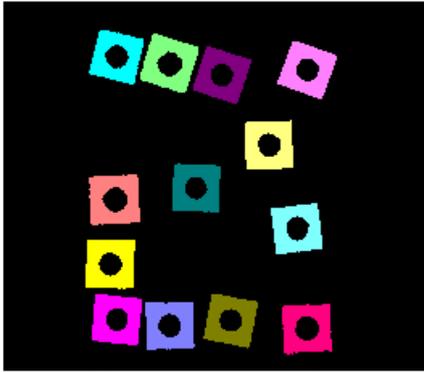
각 포인트 클라우드를 지정된 방향을 따라 내림차순으로 정렬한 다음 가장 높은 층의 포인트 클라우드를 획득합니다.



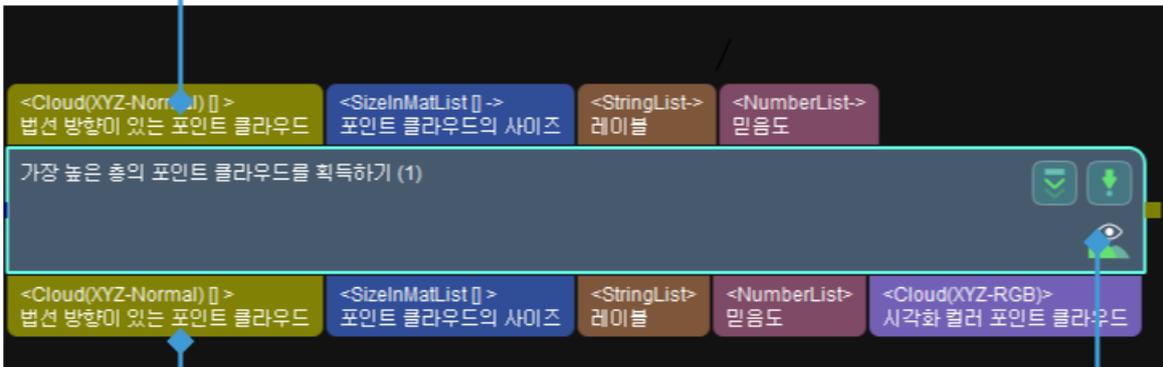
### 응용 시나리오

지정된 참조 방향에서 총 높이 범위 내의 포인트 클라우드가 보존됩니다. 가장 높은 층의 포인트를 획득하기와 달리 이 스텝은 포인트 클라우드를 직접 처리하며 일반적으로 포인트 클라우드 클러스터링 및 마스크안에 대응하는 포인트 클라우드를 추출하기 등 스텝 후에 사용됩니다.

### 입력 및 출력



입력한 포인트 클라우드



처리한 후의 포인트 클라우드



시각화 출력 결과



### 파라미터 설명

#### 기준 방향

X, Y, Z

기본값: X = 0, Y = 0, Z = 1.0

조정 설명: X, Y, Z의 값을 설정함으로써 기준 방향을 설정합니다.

#### 총 설정

### 가장 높은 층의 결과만 출력

기본값: 선택하지 않음

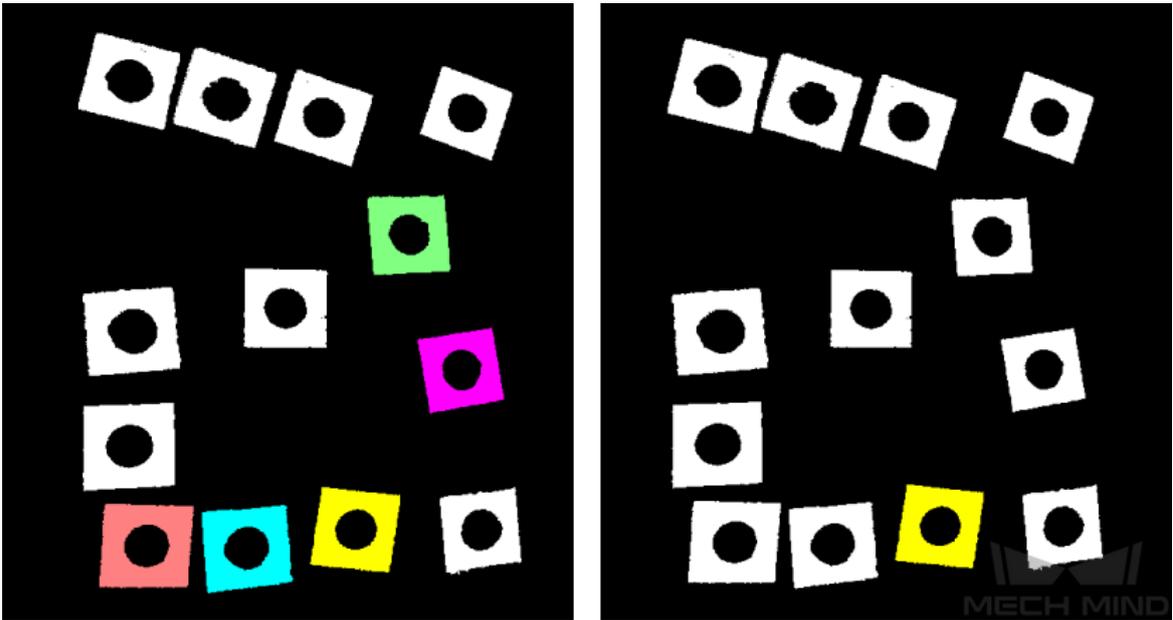
조정 설명: 이 옵션을 선택하면 지정한 방향에 따라 가장 높은 위치의 단일한 포인트 클라우드를 획득하고, 반대로 가장 높은 층의 포인트 클라우드를 획득합니다.

### 층 높이

파라미터 설명: 이 파라미터는 가장 높은 층 포인트 클라우드의 레이어 높이 범위를 설정하는 데 사용되며 레이어 높이 범위 내의 포인트 클라우드는 보존됩니다.

기본값: 100.000 mm

조정 예시: 층 높이가 100.000 mm 및 35 mm로 설정된 경우의 출력 효과는 각각 아래 왼쪽 그림, 오른쪽 그림과 같습니다.



### 층 허용 편차

기본값: 1.0000

조정 설명: 이 파라미터는 가장 높은 층과 나머지 층 사이의 높이 차이 상한을 설정하는 데 사용됩니다. 물체와 가장 높은 위치에 있는 물체 간의 거리가 층 높이 \* 층 허용 편차보다 작은 경우, 이 물체는 가장 높은 층에 위치하는 것입니다.

### 포인트 클라우드 설정

#### 포인트 수가 가장 많은 포인트 클라우드 법선 방향을 사용하기

기본값: 선택하지 않음.

설명: 이 옵션을 선택하면 사이즈가 가장 큰 포인트 클라우드의 법선 벡터를 기준 방향으로 사용하며 기준 방향 파라미터는 무효로 됩니다.

### 4.3.46. 가장 높은 층의 포즈를 획득하기



이 스텝의 내용은 유지 보수 중입니다. 이 스텝에 대한 자세한 정보가 필요한 경우 ([docs@mech-mind.net](mailto:docs@mech-mind.net))로 문의하십시오.

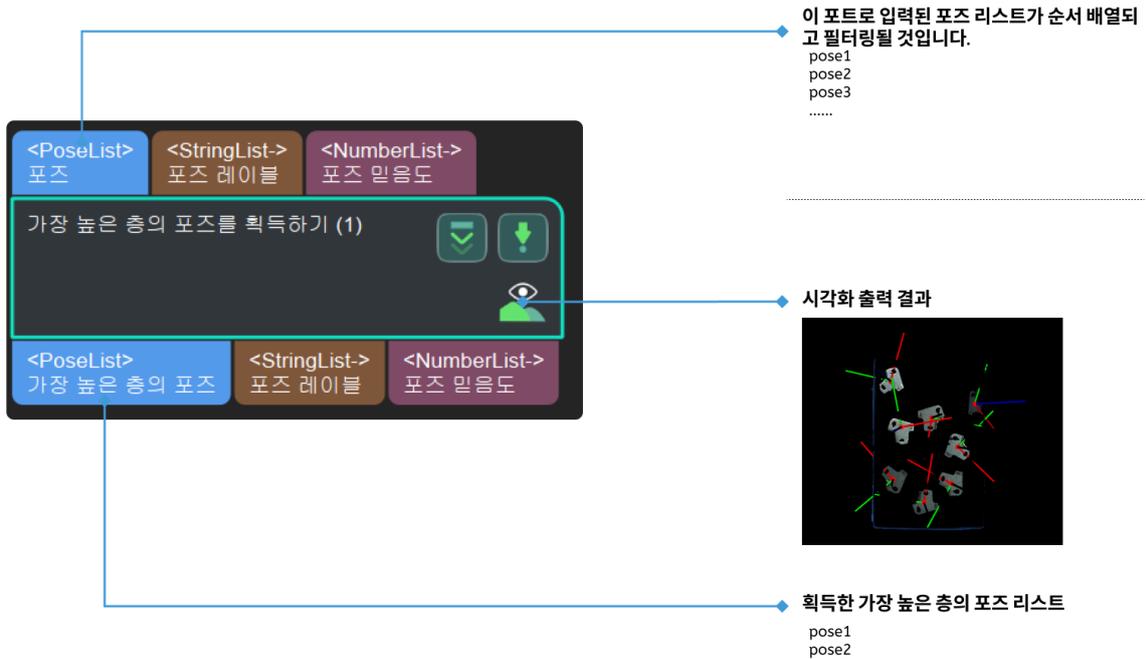
### 기능 설명

지정된 방향을 따라 포즈를 내림차순으로 배열하고 가장 높은 층의 포즈를 출력합니다.

### 응용 시나리오

일반적인 포즈 필터링 스텝으로 설정된 파라미터에 근거하여 요구에 부합한 포즈 리스트를 출력합니다.

### 입력 및 출력



## 4.3.47. 포즈의 역계산

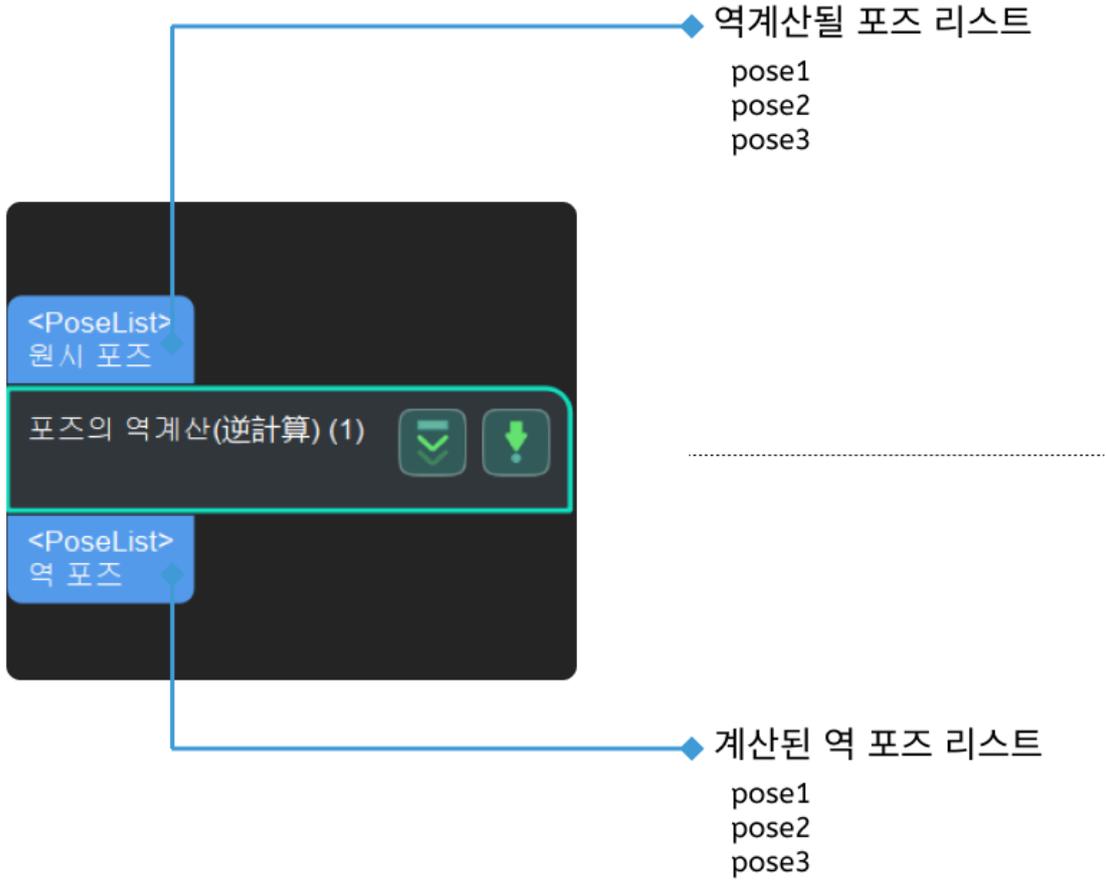
### 기능 설명

입력된 리스트의 포즈에 대한 역계산(逆計算)입니다.

### 응용 시나리오

일반적인 포즈 계산 스텝입니다. 참조 좌표계 간 변환에 사용할 수 있습니다.

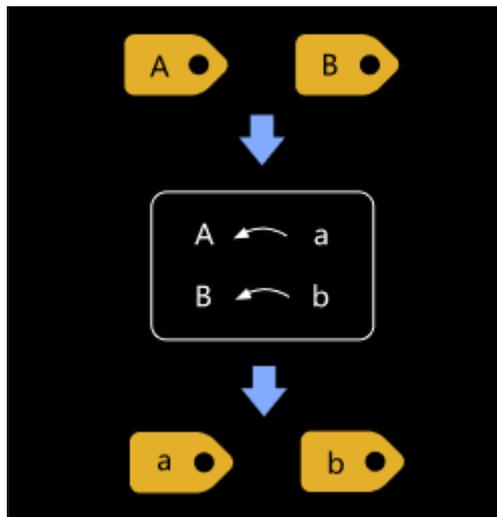
### 입력 및 출력



### 4.3.48. 레이블 매핑

#### 기능 설명

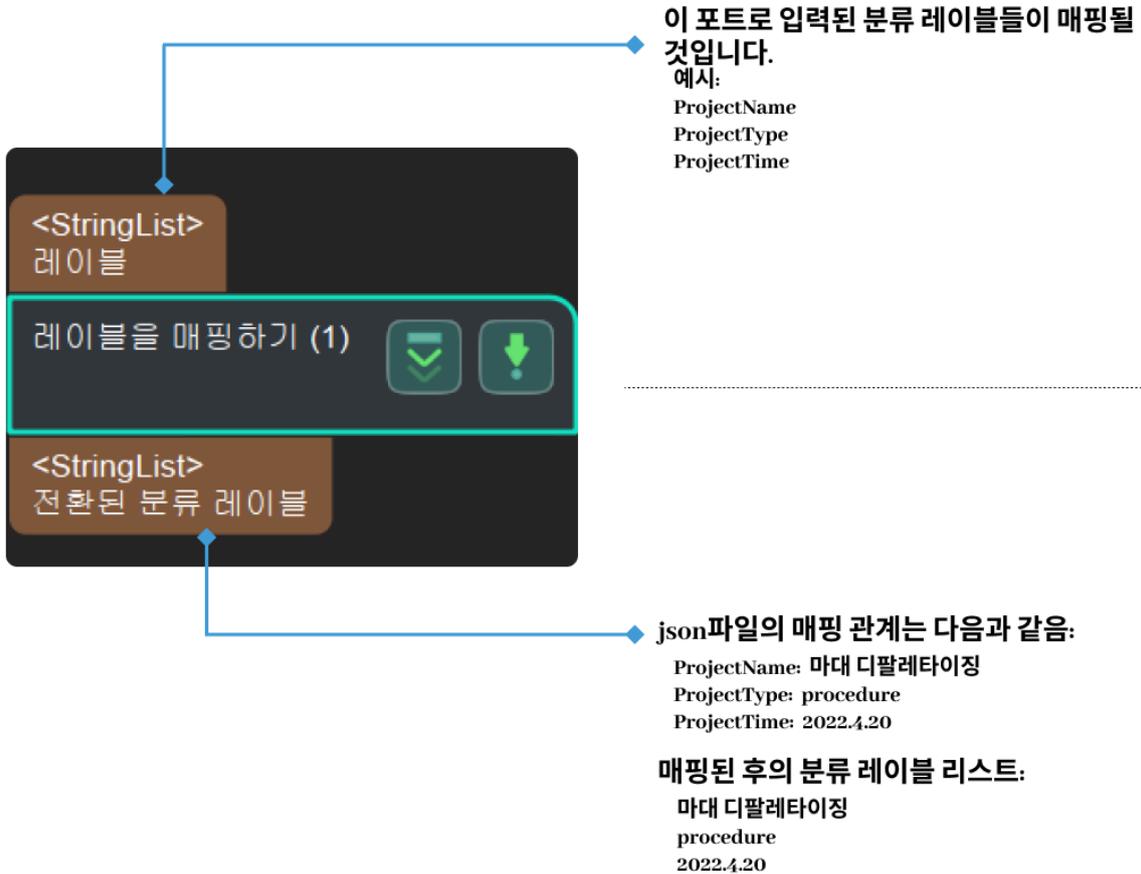
이 스텝에서는 입력한 레이블의 일대일 매핑을 통해 레이블 이름을 수정합니다.



## 응용 시나리오

이 스텝은 레이블 이름을 수정하는 데 사용됩니다.

## 입력 및 출력



## 파라미터 조절 설명

### 포트 설정

#### 입력 포트 수(1~8)

설명: 이 파라미터는 이 스텝의 입력 포트 수를 설정하는 데 사용됩니다.

기본값: 1

값 범위: 1~8

### 파일 설정



레이블 매핑은 입력 포트 수가 1일 때만 설정할 수 있습니다.

### 레이블 파일

설명: 이 파라미터는 매핑 후 레이블 파일을 설정하는 데 사용됩니다.

기본값: labelmap.json

설정 방법:

1. [레이블 매핑 설정]을 클릭하여 JSON 편집기 페이지로 이동합니다.
2. **+**를 클릭하고 **전환 전의 레이블**에 변경할 레이블 이름을 입력하고 [확인]을 클릭한 후 **\*전환 후의 레이블**에 수정된 레이블 이름을 입력하고 [확인]을 클릭합니다. 단일 레이블의 이름이 성공적으로 수정되었으며 해당 관계가 매핑 열에 나타납니다.



레이블 이름에 소수점 또는 특수 기호를 입력하면 레이블이 추가되지 않고 리스트 아래에 입력이 잘못되었음을 나타내는 프롬프트가 표시됩니다.

3. 편집해야 하는 모든 레이블을 추가할 때까지 필요에 따라 2단계를 반복합니다.
4. **폴더 경로**는 현재 프로젝트 경로입니다. 오른쪽의 [⋯]을 클릭하여 프로젝트 위치를 변경할 수 있으며, 필요에 따라 설정하십시오.
5. **JSON 파일 명칭**에 파일 명칭을 입력한 후 [저장하기]를 클릭하여 레이블이 해당 파일에 저장되도록 합니다.
6. [확인]을 클릭하여 레이블 추가를 완료합니다.

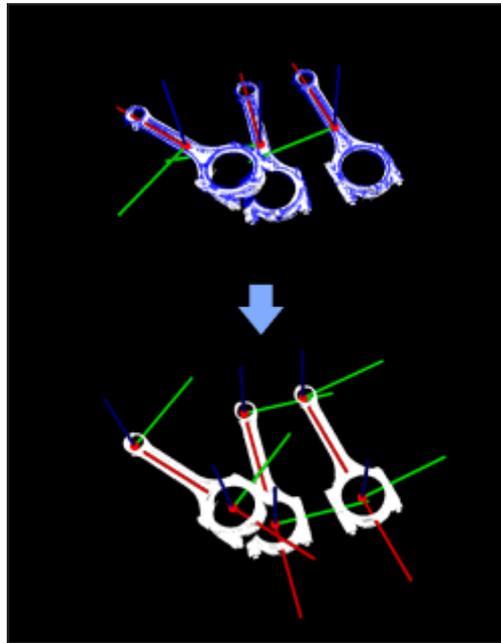


이미 미리 설정된 레이블 파일이 있는 경우 **JSON 파일 명칭** 옆의 [⋯]을 직접 클릭하여 파일을 선택하고, 매핑 관계를 읽어올 수 있습니다.

### 4.3.49. 여러 개의 픽 포인트에 매핑하기

#### 기능 설명

물체의 기하학적 중심점과 설정된 매핑 관계를 통해 다른 픽 포인트를 계산합니다.



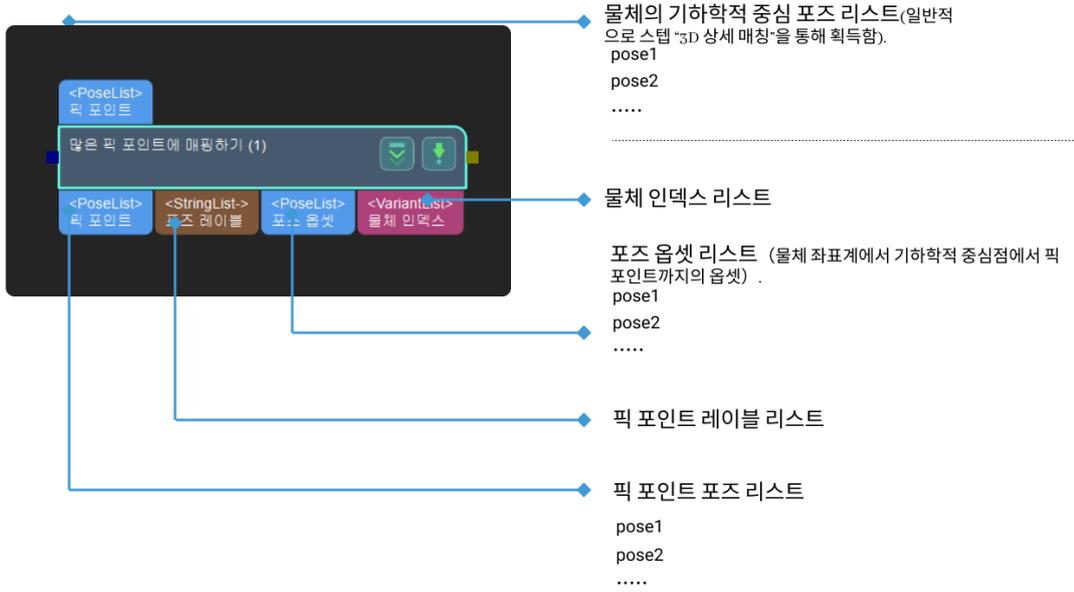
#### 응용 시나리오

픽 포인트와 기하학적 중심점이 다르거나 픽 포인트가 여러 개 있을 때 이 스텝을 사용해야 합니다. 다음과 같은 방법을 통해 시나리오에 있는 물체에 로봇 컨트롤 소프트웨어로 보낼 수 있는 픽 포인트를 추가하세요.

이 스텝에서 출력된 "픽 포인트"가 스텝 **포즈 전환**의 "참고 포즈" 포트와 연결되어야 합니다.

이 스텝에서 출력된 "포즈 옵션"이 스텝 **포즈 전환**의 "원시 포즈" 포트와 연결되어야 합니다.

## 입력 및 출력



## 파라미터 설명

### 포즈 파일의 경로

### 기하학적 중심점 파일

템플릿에서 물체 기하학적 중심점의 포즈

조절 설명: 기하학적 중심점 파일을 저장하는 절대 경로나 상대 경로.

### 배치 포인트 포즈 파일

물체 배치 포인트의 포즈

조절 설명: 배치 포인트 파일을 저장하는 절대 경로나 상대 경로.

### 픽 포인트 포즈 파일

템플릿에서 물체 픽 포인트의 포즈

조절 설명: 픽 포인트 파일을 저장하는 절대 경로나 상대 경로.

### 레이블 가져오기(선택 사항)

### 포즈 레이블 파일

픽 포인트 파일과 대응하는 레이블 파일

조절 설명: 레이블 파일을 저장하는 절대 경로나 상대 경로.

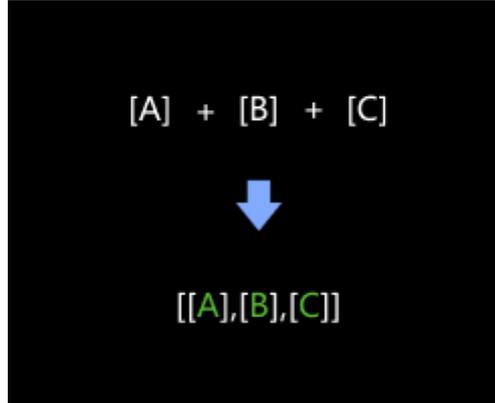
## 4.3.50. 데이터를 합치기



이 스텝의 내용은 유지 보수 중입니다. 이 스텝에 대한 자세한 정보가 필요한 경우 ([docs@mech-mind.net](mailto:docs@mech-mind.net))로 문의하십시오.

## 기능 설명

일반적인 데이터 합치기 스텝입니다.

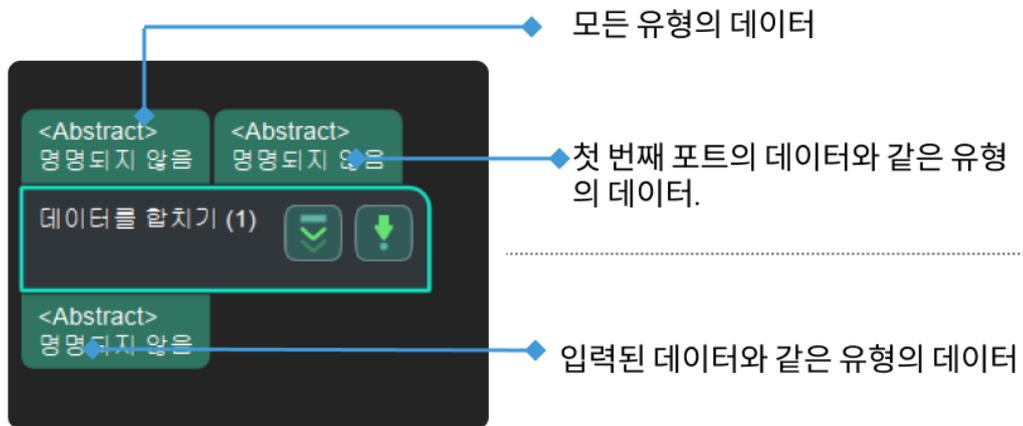


## 응용 시나리오

포즈 및 값과 같은 데이터를 병합하는 데 사용됩니다.

이 스텝은 새 버전이므로 구 버전 스텝인 [포즈 리스트를 합치기](#), [포인트 클라우드를 합치기](#)를 이 스텝으로 교체하는 것이 좋습니다.

## 입력 및 출력



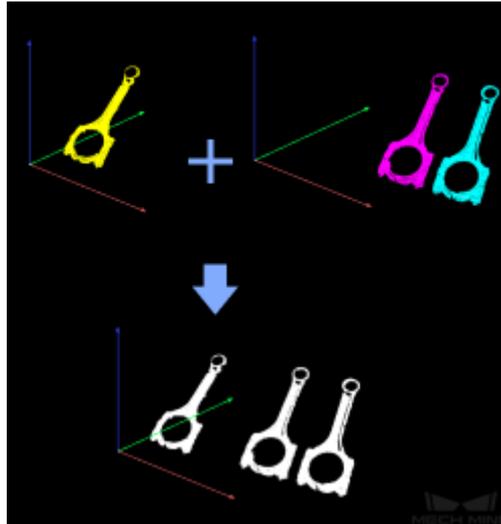
### 4.3.51. 포인트 클라우드를 합치기



이 스텝의 내용은 유지 보수 중입니다. 이 스텝에 대한 자세한 정보가 필요한 경우 ([docs@mech-mind.net](mailto:docs@mech-mind.net)) 로 문의하십시오.

## 기능 설명

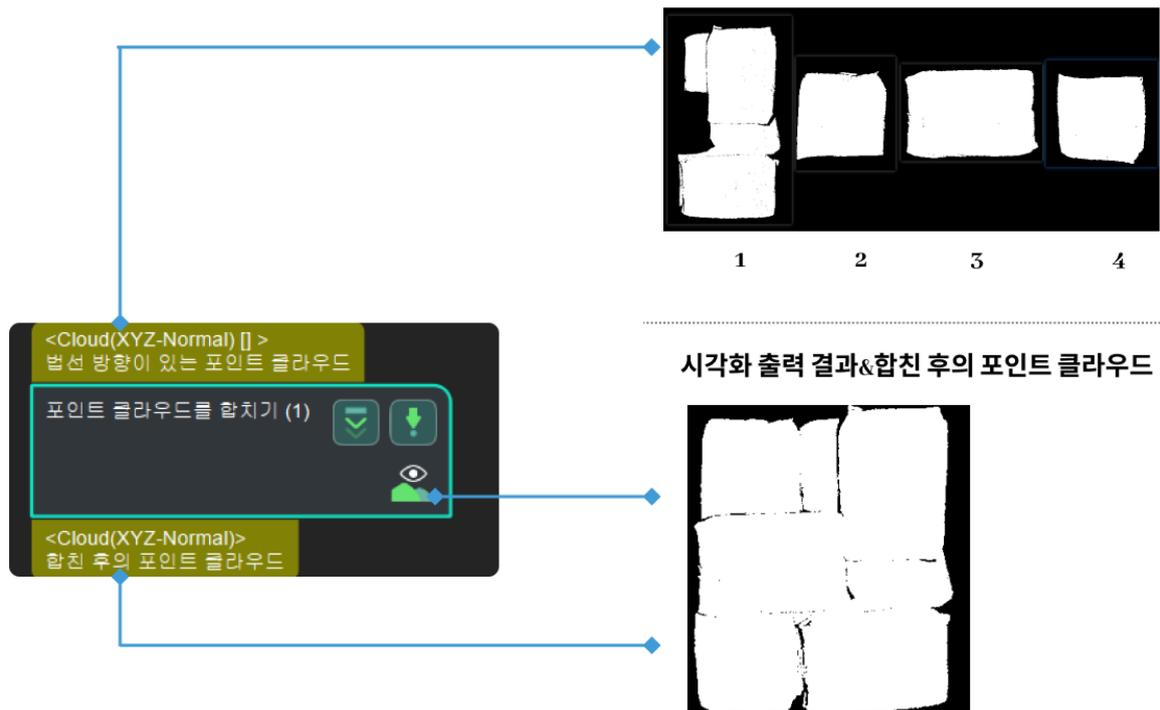
여러 개의 포인트 클라우드를 하나로 합칩니다.



### 응용 시나리오

보통 포인트 클라우드를 클러스터링, 마스크 안에 대응하는 포인트 클라우드를 추출하기 등 스텝 후에 사용되고 포인트 클라우드 리스트에 있는 여러 개 포인트 클라우드를 하나로 합치고 나중에 포인트 클라우드를 한 전체로 처리하는 데 편리를 줍니다.

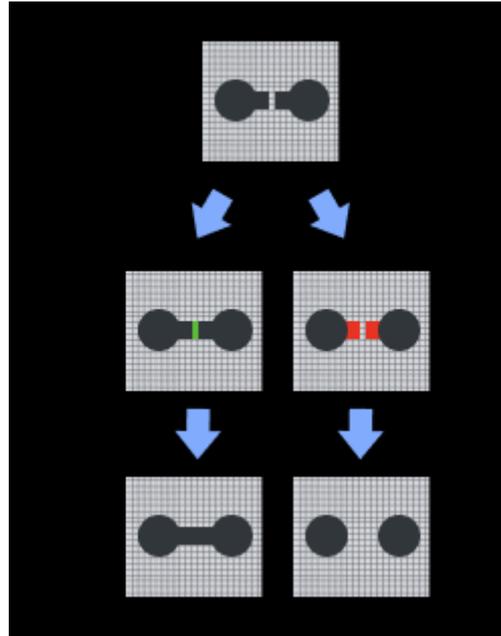
### 입력 및 출력



## 4.3.52. 형태학적 변환

### 기능 설명

이 스텝은 형태학적 연산을 통해 이미지를 처리합니다.

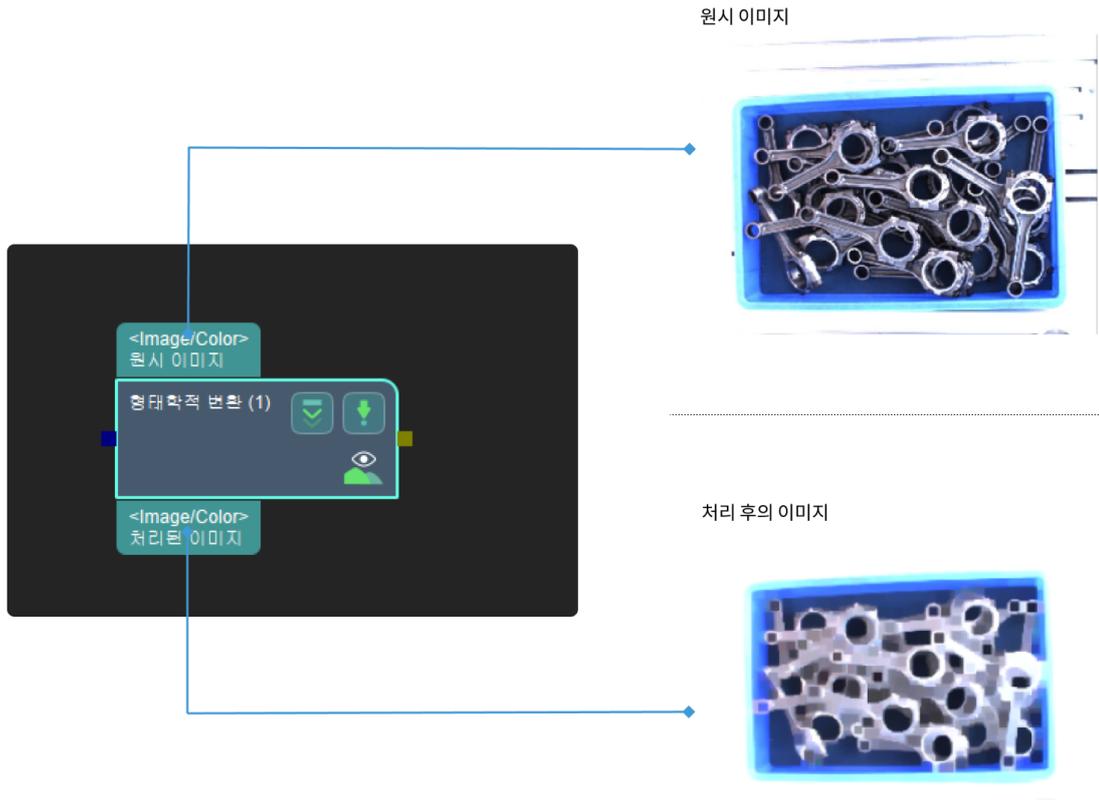


### 응용 시나리오

이 스텝은 일반적으로 이미지 사전 처리 단계에서 사용됩니다. 이미지의 효과가 좋지 않거나 계속 조절해야 할 경우에 다른 수요에 따라 팽창, 침식, 오픈닝(opening)과 닫힘(closing) 연산을 사용하여 이미지에 대해 노이즈를 제거하거나 채웁니다.

### 입력 및 출력

닫힘(closing) 을 예로 들면, 형태학적 변환 입/출력 변환의 효과는 다음 그림과 같습니다.



## 파라미터 설명

### 형태학적 처리 유형

파라미터 설명: 이 파라미터는 형태학적 연산 유형을 선택하는 데 사용됩니다.

기본값: DilateOperator

값 리스트: ClosingOperator(닫힘), DilateOperator(팽창), ErodeOperator(침식), OpeningOperator(오픈닝), Skeleton(스켈레톤)

조정 설명:

1. ClosingOperator : 침식 후 팽창의 또 다른 이름입니다. 이 기능은 전경 물체 내부의 작은 구멍이나 물체의 작은 검은 포인트를 닫는 데에 도움이 됩니다.

### 커널(kernel) 사이즈

기본값: 21 pixel

조정 설명: 실제 상황에 따라 설정하십시오. 커널(kernel) 조작이 커질수록 팽창 후 침식 효과가 강해집니다. 특정 효과에 대한 내용은 [조정 예시](#)를 참조하십시오.

### 커널(kernel) 형태

기본값: 직사각형(MORPH\_RECT)

값 리스트: 직사각형(MORPH\_RECT), 십자형(MORPH\_CROSS), 타원형(MORPH\_ELLIPSE)

설명: 기본값으로 선택하는 것을 권장합니다. 만약 특별한 상황이 있는 경우 실제 수요에 맞게 설정하십시오.

- 이미지 디테일 복원 정도 MORPH\_RECT < MORPH\_CROSS < MORPH\_ELLIPSE
- 처리 소요시간 MORPH\_RECT < MORPH\_CROSS < MORPH\_ELLIPSE

2. DilateOperator : 팽창. 물체의 면적을 늘리고 물체의 훼손 부분을 연결합니다.

#### 커널(kernel) 사이즈

기본값: 3 pixel

조정 설명: 커널(kernel) 조작이 커질수록 팽창 효과가 강해지며, 감지 요구에 따라 설정합니다. 특정 효과에 대한 내용은 [조정 예시](#)를 참조하십시오.

#### 커널(kernel) 형태

기본값: 직사각형(MORPH\_RECT)

값 리스트: 직사각형(MORPH\_RECT), 십자형(MORPH\_CROSS), 타원형(MORPH\_ELLIPSE)

설명: 기본값으로 선택하는 것을 권장합니다. 만약 특별한 상황이 있는 경우 실제 수요에 맞게 설정하십시오.

- 이미지 디테일 복원 정도 MORPH\_RECT < MORPH\_CROSS < MORPH\_ELLIPSE
- 처리 소요시간 MORPH\_RECT < MORPH\_CROSS < MORPH\_ELLIPSE

3. ErodeOperator : 침식. 작은 하얀 노이즈를 제거하고 연결된 두 물체를 분리합니다.

#### 커널(kernel) 사이즈

기본값: 3 pixel

조정 설명: 커널(kernel) 조작이 커질수록 팽창 효과가 강해지며, 감지 요구에 따라 설정합니다. 특정 효과에 대한 내용은 [조정 예시](#)를 참조하십시오.

#### 커널(kernel) 형태

기본값: 직사각형(MORPH\_RECT)

값 리스트: 직사각형(MORPH\_RECT), 십자형(MORPH\_CROSS), 타원형(MORPH\_ELLIPSE)

설명: 기본값으로 선택하는 것을 권장합니다. 만약 특별한 상황이 있는 경우 실제 수요에 맞게 설정하십시오.

- 이미지 디테일 복원 정도 MORPH\_RECT < MORPH\_CROSS < MORPH\_ELLIPSE
- 처리 소요시간 MORPH\_RECT < MORPH\_CROSS < MORPH\_ELLIPSE

4. OpeningOperator : 침식 후 팽창의 또 다른 이름이며 노이즈를 제거하는 데에 도움이 됩니다.

#### 커널(kernel) 사이즈

기본값: 21 pixel

조정 설명: 실제 상황에 따라 설정하십시오. 특정 효과에 대한 내용은 [조정 예시](#)를 참조하십시오.

#### 커널(kernel) 형태

기본값: 직사각형(MORPH\_RECT)

값 리스트: 직사각형(MORPH\_RECT), 십자형(MORPH\_CROSS), 타원형(MORPH\_ELLIPSE)

설명: 기본값으로 선택하는 것을 권장합니다. 만약 특별한 상황이 있는 경우 실제 수요에 맞게 설정하십시오.

- 이미지 디테일 복원 정도  $MORPH\_RECT < MORPH\_CROSS < MORPH\_ELLIPSE$
- 처리 소요시간  $MORPH\_RECT < MORPH\_CROSS < MORPH\_ELLIPSE$

5. Skeleton : 물체의 윤곽선을 추출합니다. 특정 효과에 대한 내용은 [조정 예시](#)를 참조하십시오.

## 조정 예시

### ClosingOperator 작업

**ClosingOperator** 작업을 선택한 경우 커널 크기가 커질 때 이미지 효과는 아래 그림과 같습니다(그림에서 커널 크기가 1에서 3으로 점차 증가함).



### DilateOperator 작업

**DilateOperator** 작업을 선택한 경우 커널 크기가 커질 때 이미지 효과는 아래 그림과 같습니다(그림에서 커널 크기가 1에서 3으로 점차 증가함).



### ErodeOperator 작업

**ErodeOperator** 작업을 선택한 경우 커널 크기가 커질 때 이미지 효과는 아래 그림과 같습니다(그림에서 커널 크기가 1에서 3으로 점차 증가함).



### OpeningOperator 작업

**OpeningOperator** 작업이 선택되었을 때, 커널 크기가 커질 때의 이미지 효과는 아래 그림과

같습니다(그림에서 커널 크기는 1에서 3까지 점차 증가합니다).



#### Skeleton 작업

이 작업 처리 전후의 이미지 효과는 다음 그림과 같습니다(1은 처리 전, 2는 처리 후).



### 4.3.53. 법선 방향을 계산하기



이 스텝의 내용은 유지 보수가 중단되었습니다. 새 버전 스텝 **포인트 클라우드의 법선 벡터를 계산하고 필터링하기**에서 **NormalEstimation**의 법선 방향 계산 방법을 사용하십시오. 이 스텝에 대한 자세한 정보가 필요한 경우 ([docs@mech-mind.net](mailto:docs@mech-mind.net))로 문의하십시오.

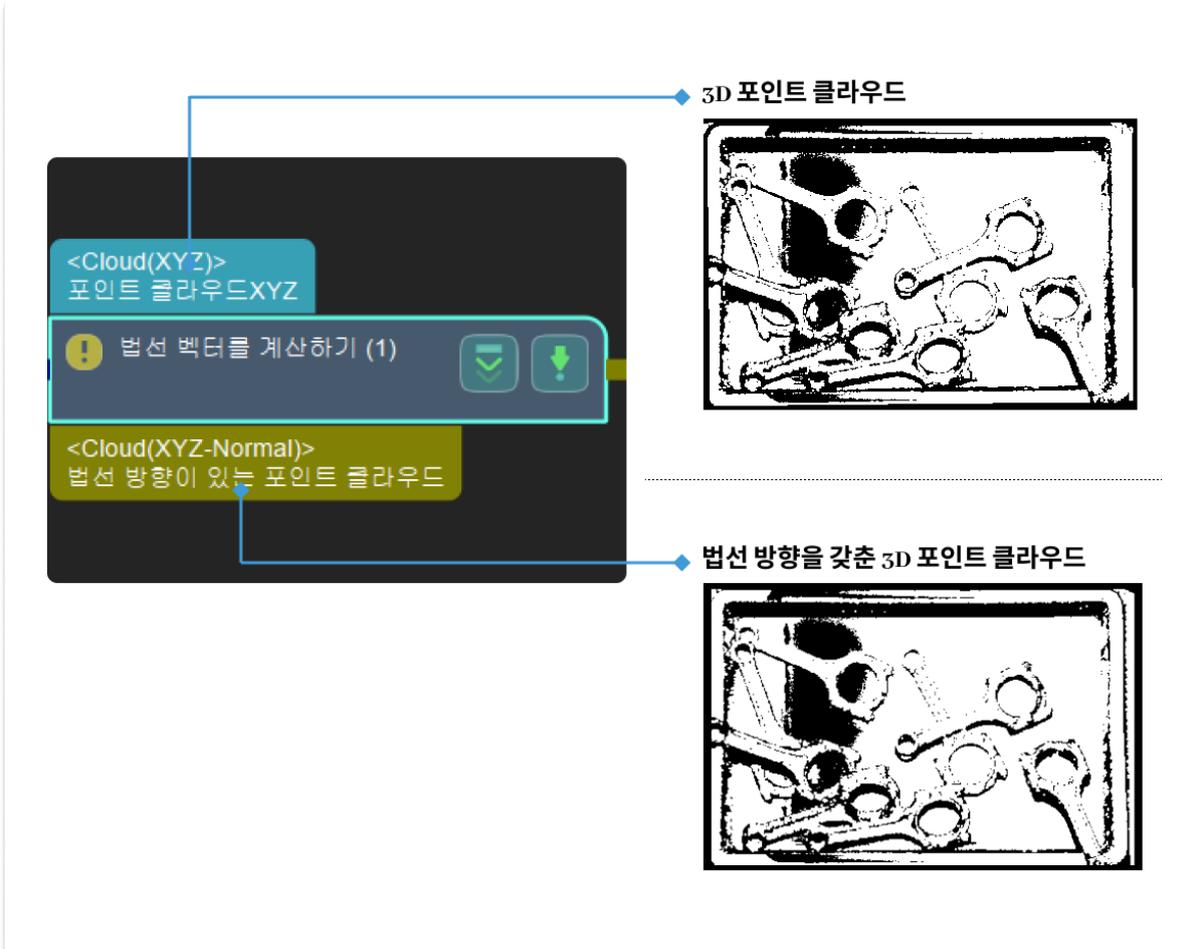
#### 기능 설명

포인트 클라우드의 법선 벡터를 계산합니다.

#### 응용 시나리오

이 스텝은 구 버전입니다. 새 버전으로 교체하십시오.

#### 입력 및 출력



### 4.3.54. 수치 확대 및 축소



이 스텝의 내용은 유지 보수가 중단되었습니다. 새 버전의 스텝 **수치 계산**을 사용하십시오. 이 스텝에 대한 자세한 정보가 필요한 경우 ([docs@mech-mind.net](mailto:docs@mech-mind.net))로 문의하십시오.

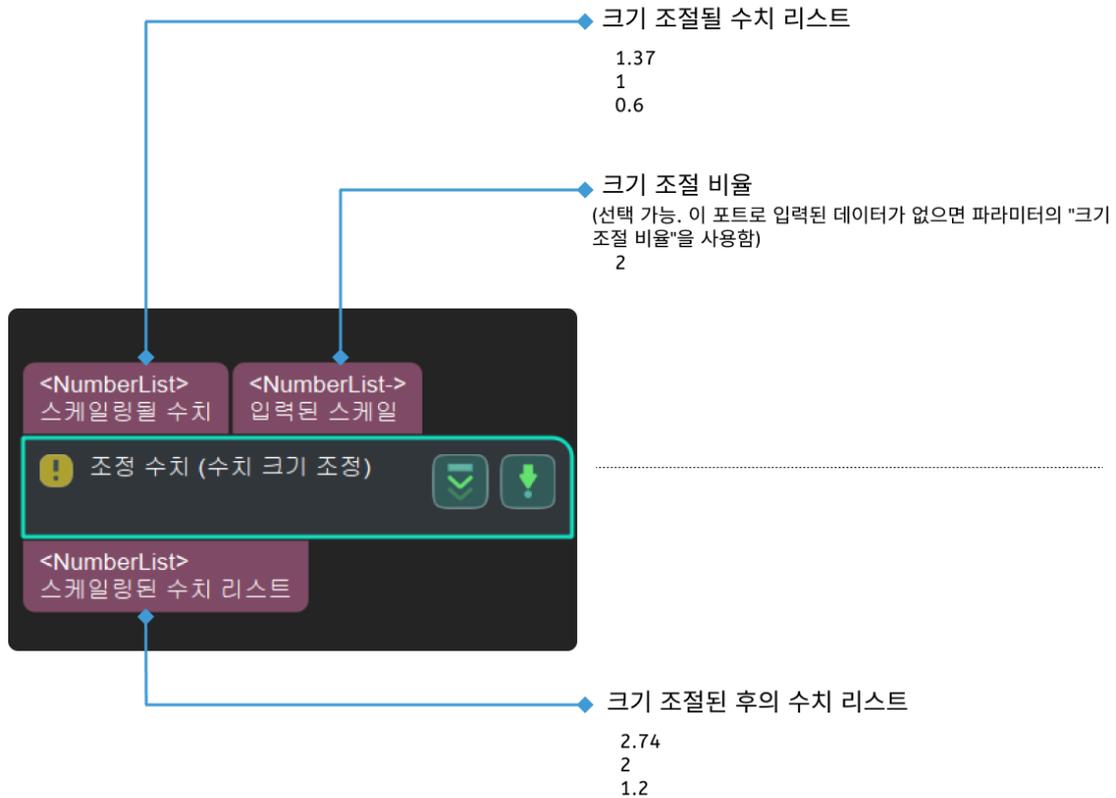
#### 기능 설명

첫 번째 포트에 입력된 수치 리스트의 크기를 조절합니다. 두 번째 포트의 입력을 사용하거나 파라미터에서 크기 조절 비율을 설정할 수 있습니다.

#### 응용 시나리오

이 스텝은 구 버전입니다. 더 많은 기능을 갖춘 새 버전으로 교체하십시오.

#### 입력 및 출력



### 4.3.55. 수치 계산



이 스텝의 내용은 유지 보수 중입니다. 이 스텝에 대한 자세한 정보가 필요한 경우 ([docs@mech-mind.net](mailto:docs@mech-mind.net))로 문의하십시오.

#### 기능 설명

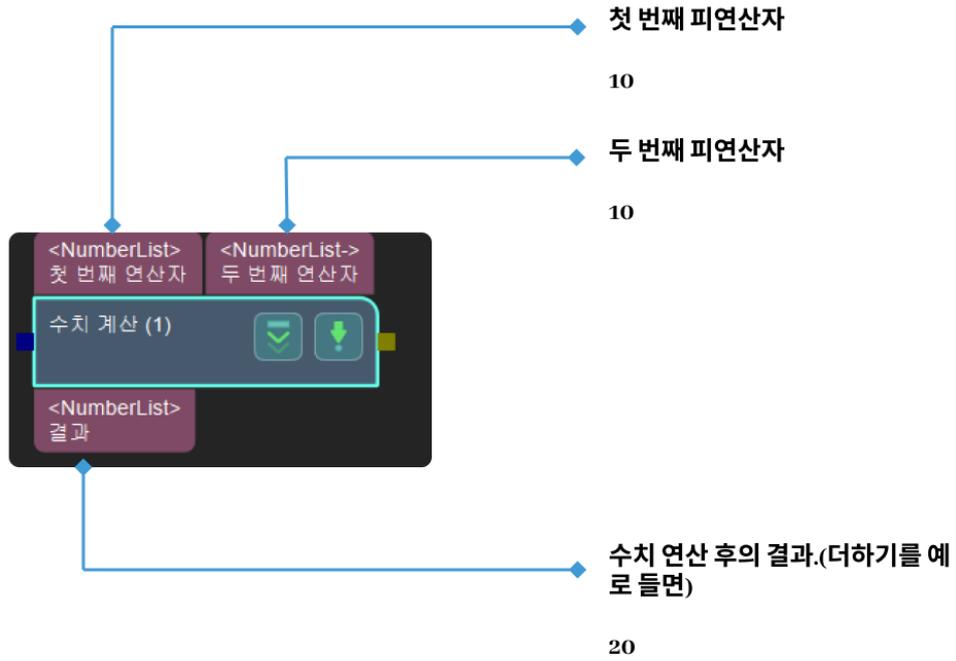
입력된 수치에 대해 수치 연산을 합니다. 현재 가능한 연산 유형은 더하기, 빼기, 곱하기와 나누기 등입니다.

#### 응용 시나리오

일반적으로 사용되는 연산 스텝이고 첫 번째 피연산자를 두 번째 피연산자로 더하거나 빼거나 곱하거나 나누게 될 것입니다.

복잡한 수치 연산 수요가 있는 사용자는 스텝 [JavaScript 엔진을 통한 결과 평가](#)를 사용할 수 있습니다.

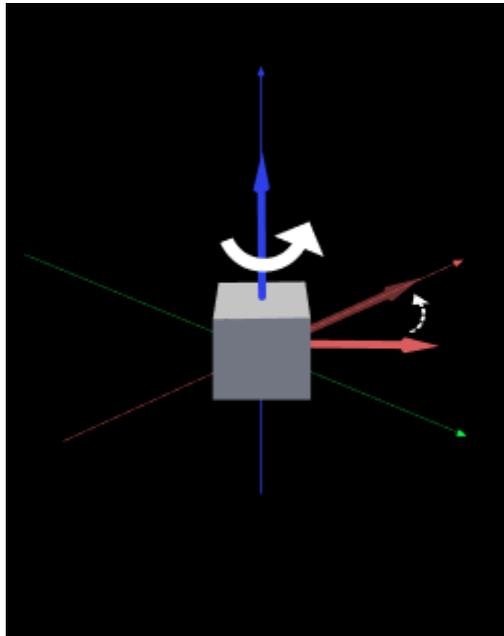
#### 입력 및 출력



### 4.3.56. 포즈의 축을 지정된 방향으로 가리키도록 회전하기

#### 기능 설명

설정된 기준 방향을 가리키도록 포즈의 축을 조정합니다.

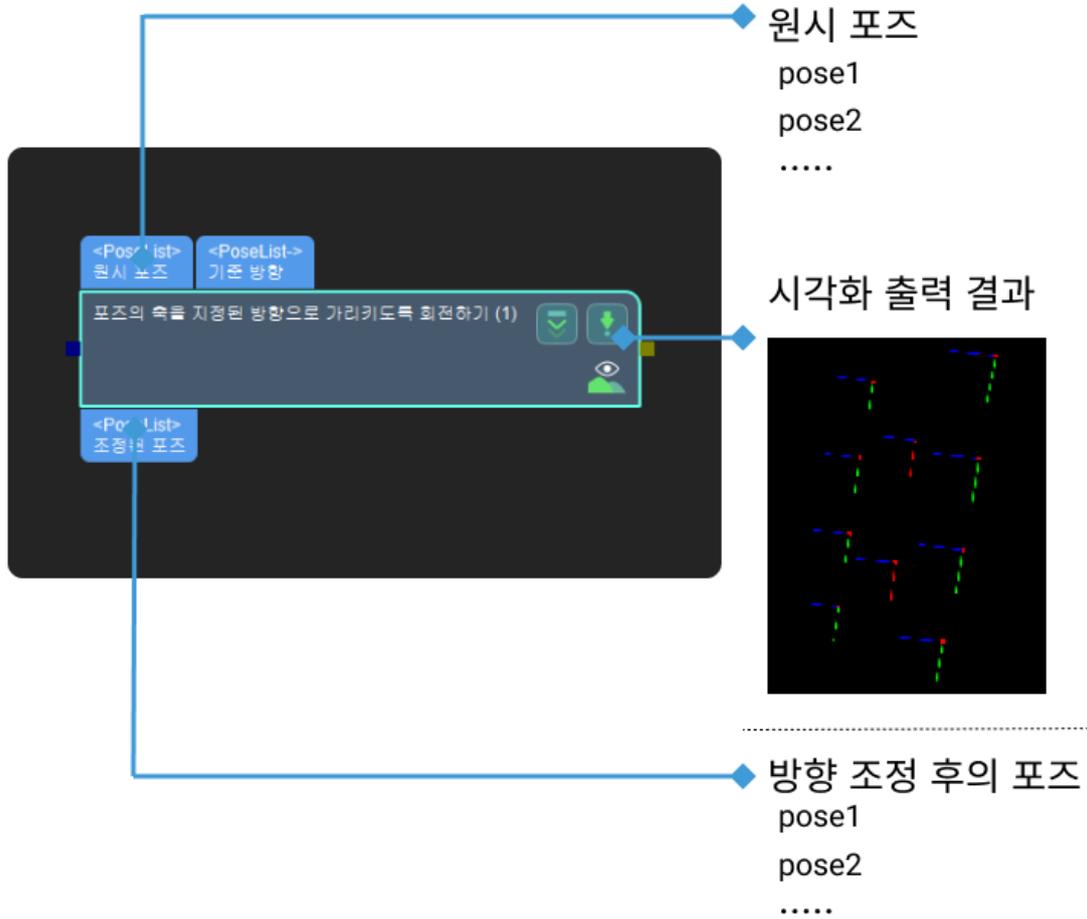


#### 응용 시나리오

일반적으로 포즈 방향을 조정해야 하는 다양한 시나리오에서 사용됩니다.

이 스텝은 구 버전입니다. 물체의 대칭성에 따라 포즈를 회전해야 하는 경우(이 스텝에서 "보정 방법" 파라미터를 "ROTATION"으로 설정하는 것에 해당) 새 버전의 스텝을 사용하여 회전하십시오. 새 버전 스텝 **지정된 방향으로 포즈를 회전하기(대칭성 제약)**과 스텝 **3D 벡터를 빠르게 만들기**를 함께 사용합니다.

입력 및 출력



파라미터 설명

시각화 설정

포즈 시각화 설정

기본값: 출력 포즈만 표시하기

값 리스트: 입력 포즈만 표시하기, 출력 포즈만 표시하기, 입력 및 출력 포즈를 모두 표시하기

설명: 처리 후의 포즈만 표시하기(기본값), 처리 전의 포즈만 표시하기, 처리하기 전/후의 포즈를 모두 표시하기

방법 선택

보정 방법

HARD\_RECTIFY : 기본값 두 번째 입력(기준 포즈)이 있고 useFirstRefPose가 체크되어 있으면 첫 번째 기준 포즈에 해당하는 축 유형의 방향이 기준 방향이 되고, 그렇지 않으면 모든 기준 포즈에 해당하는 축 유형이 기준 방향으로 사용됩니다. 기준 포즈 입력이 없으면 reference Dir이 기준 방향을 결정합니다.

ROTATION: 물체의 대칭성에 따라 보정할 회전축의 방향이 지정된 기준축의 방향과 대체로 일치하도록 대칭 각도를 설정합니다.

## 축 설정

### 조정될 축

설명: 조정할 회전축입니다.

값 리스트: X ; Y ; Z(기본값)

### 기준 방향

#### X 축

기본값: 0

설명: 기준 축 X의 방향 벡터

#### Y 축

기본값: 0

설명: 기준 축 Y의 방향 벡터

#### Z 축

기본값: 1

설명: 기준 축 Z의 방향 벡터

## 4.3.57. 포인트 클라우드 클러스터링

### 기능 설명

지정된 규칙에 따라 포인트 클라우드를 클러스터링하고 보통 대상 물체를 분할하는 데 사용됩니다.



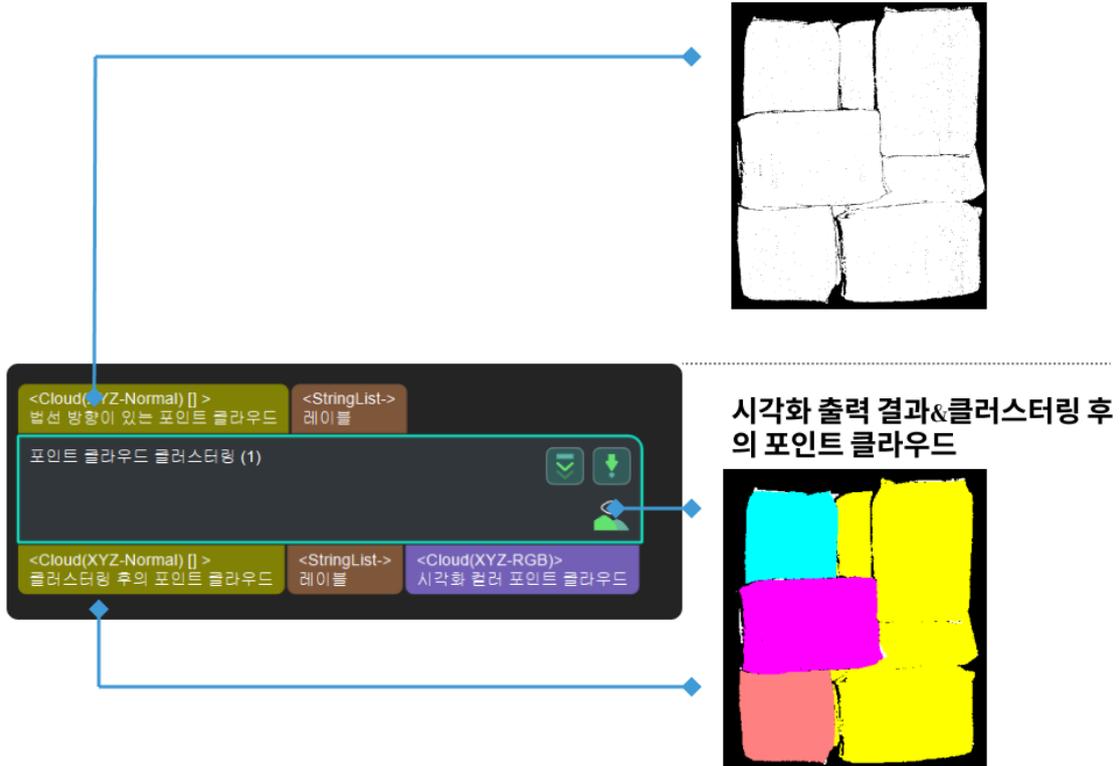
### 응용 시나리오

일반적으로 포인트 클라우드의 사전 처리 단계에 사용되며 간섭을 줄 수 있는 포인트 클라우드를 제거합니다.

유클리드 거리(EuclideanCluster) 방법과 영역 성장 세그멘테이션(RegionGrowingSeg) 방법을 통하여 포인트 클라우드에 대해 클러스터링을 할 수 있습니다.

포인트 클라우드가 공간적으로 서로 분리되는 경우에 전자(EuclideanCluster)를 사용하고, 공간적으로 연속되지만 연결점 위치에서 곡률 변동이 큰 경우에 후자(RegionGrowingSeg)를 사용하는 것을 권장합니다.

### 입력 및 출력



## 파라미터 설명

### 클러스터링 알고리즘

기본값: EuclideanCluster

값 리스트: EuclideanCluster, RegionGrowingSeg

조정 설명: 이 파라미터는 클러스터링 방법을 선택하는 데 사용됩니다. 총 2개의 클러스터링 방법이 있습니다. 실제 작업 현장에서 EuclideanCluster 알고리즘을 사용하는 것을 권장하므로 우선 이 알고리즘을 소개하겠습니다.

- **EuclideanCluster**: 거리에 따라서 어떤 범주에 속하는지를 판단합니다.
- **RegionGrowingSeg**: 법선 방향과 곡률에 따라 어떤 범주에 속하는지를 판단합니다.

### EuclideanCluster

#### 출력된 클러스터 중에 인접된 포인트 간의 최대 거리

기본값: 3.000 mm  
 파라미터 설명: 이 파라미터는 클러스터링 공차를 설정하는 데 사용되며 단위는 밀리미터 (mm)입니다.

조정 설명: 이 파라미터는 클러스터의 허용 오차입니다. 이 수치를 높이면 멀리 떨어져 있는 포인트들은 같은 클러스터로 분류되고 수치를 작게 설정하면 간격이 가까운 포인트들은 다른 클러스터로 분류될 것입니다.

조절 예시: 아래 그림과 같습니다. 아래 그림과 같이 왼쪽 그림은 기본값이 3.000 mm일 때의 결과이고, 오른쪽 그림은 이 파라미터가 5.000 mm로 조정된 결과입니다. 조정 후 중간 녹색 포인트 클라우드가 다른 유형으로 나뉘어져 있음을 알 수 있습니다.

[point cloud clustering Adjacent point distance] | *point-cloud-clustering-*

## Adjacent\_point\_distance.png

### 코어

기본값: 4

조정 설명: 이 파라미터는 영역 성장 과정에서 인접된 포인트의 법선 방향 각도 차이 역치를 조절하는 데 사용되며 이 수치를 높이면 인접된 임의의 포인트의 법선 방향 각도 차이에 대한 허용 오차가 커지며 각도 차이가 큰 포인트들은 여전히 같은 클러스터로 분류될 것입니다.

### RegionGrowingSeg

#### 인접된 포인트 수

기본값: 30

조정 설명: 이 파라미터는 영역 성장 과정의 포인트의 곡률 역치의 상한을 조정하는 데 사용됩니다. 이 값을 크게 조정하면 검색되는 픽셀 수가 증가하여 영역 성장이 빨라지며 이로 인해 얻어지는 클러스터 수가 줄어듭니다.

#### 평활도 역치

기본값: 4

조정 설명: 이 파라미터는 영역 성장 과정의 인접 포인트의 법선 방향 각도 차이 역치를 조정하는 데 사용됩니다. 이 값을 증가시키면 인접한 픽셀들의 법선 방향 각도에 대한 허용 오차가 커지며, 각도 차이가 큰 점들도 여전히 같은 범주로 분류됩니다.

#### 곡률 역치

기본값: 1

설명: 이 파라미터는 영역 성장 과정의 포인트의 곡률 역치의 상한을 조정하는 데 사용됩니다.

#### 클러스터 안의 최소 포인트 수

기본값: 800

조정 설명: 이 파라미터는 클러스터링 후 결과를 필터링하는 데 사용되며 클러스터 안의 최소 포인트 수보다 큰 클래스가 출력됩니다. 이 파라미터를 증가시키면 최종 출력되는 클래스의 수가 줄어들고, 이 파라미터를 감소시키면 최종 출력되는 클래스의 수가 늘어납니다.

#### 클러스터 안의 최대 포인트 수

기본값: 3000000

조정 설명: 클러스터링 클래스를 필터링하기 위한 이 파라미터의 결과는 클래스의 최대 포인트 수보다 작아야 결과가 출력됩니다. 이 파라미터를 증가시키면 최종 출력되는 클래스의 수가 증가하며, 너무 큰 경우 큰 영향을 미치지 않습니다. 이 파라미터를 감소시키면 최종 출력되는 클래스의 수가 줄어듭니다.

조정 예시: 포인트 클라우드 클러스터링 클래스가 5 개 범주 중에서 나온다고 가정하면 포인트는 각각 10000, 20000, 30000, 40000 및 50000입니다. 클러스터 안의 최대 포인트 수가 45000으로 설정되고 최소 포인트 수가 15000으로 설정되면 10000과 50000의 클래스가 필터링되고 출력 포인트 수는 20000, 30000 및 40000입니다.

#### GPU 사용

기본값: 선택하지 않음

파라미터 설명: 이 파라미터는 GPU를 사용하여 계산 속도를 높일지를 결정합니다.

### 4.3.58. 포인트 클라우드를 필터링하기



이 스텝의 내용은 유지 보수가 중단된 상태입니다. 새 버전 스텝 **포인트 클라우드가 요구에 부합한지 검증하기**를 사용하십시오. 이 스텝에 대한 자세한 정보가 필요한 경우 ([docs@mech-mind.net](mailto:docs@mech-mind.net))로 문의하십시오.

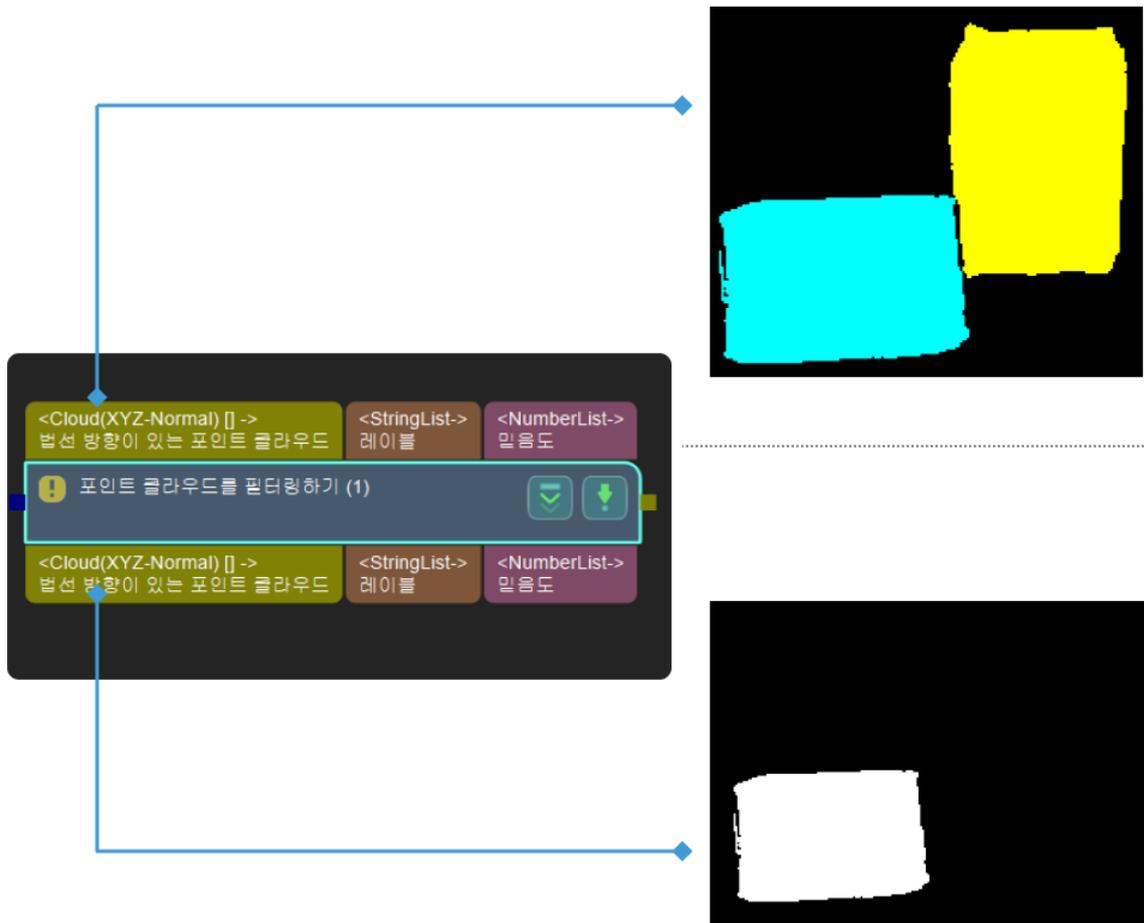
#### 기능 설명

설정된 규칙에 따라 포인트 클라우드를 필터링합니다.

#### 응용 시나리오

일반적으로 포인트 클라우드의 포인트는 필터링에 사용됩니다("필터 유형" 파라미터의 "CloudCapacityFilter"에 해당).

#### 입력 및 출력



#### 파라미터 설명

##### 필터 유형

설명: 이 파라미터는 필터 유형을 선택하는 데 사용됩니다. 네 가지의 유형을 선택할 수 있습니다. 원형 포인트 클라우드 필터(CircleCloudsFilter), 포인트 클라우드 용량 필터(CloudCapacityFilter), 사각형 포인트 클라우드 필터(RecCloudsFilter) 및 솔리드 포인트 클라우드 필터(SolidCloudsFilter)입니다.

기본값: CircleCloudsFilter

권장값: 실제 수요에 따라 설정합니다.

값 리스트: CircleCloudsFilter, CloudCapacityFilter, RecCloudsFilter, SolidCloudsFilter

### 원형 포인트 클라우드 필터(CircleCloudsFilter)

원형 포인트 클라우드 필터: 3D 포인트 클라우드를 2D 마스크로 투영하고 마스크의 외부 윤곽을 획득하여 원형도 및 둘레 길이를 계산해 역치 범위 이외의 포인트 클라우드를 제거합니다.

#### 윤곽 검색 모드

설명: 이 파라미터는 윤곽 검색 모드를 선택하는 데 사용됩니다. RETR\_EXTERNAL 및 RETR\_HULL 두 가지 유형이 있습니다.

기본값: RECT\_EXTERNAL

권장값: 실제 감지 수요에 따라 설정합니다.

#### 최소 면적 역치

설명: 이 파라미터는 원형 포인트 클라우드의 최소 면적을 설정하는 데 사용됩니다.

기본값: 100

권장값: 실제 상황에 근거하여 설정합니다.

#### 최대 면적 역치

설명: 이 파라미터는 원형 포인트 클라우드의 최대 면적을 설정하는 데 사용됩니다.

기본값: 1000

권장값: 실제 상황에 근거하여 설정합니다.

#### 최소 진원도

설명: 이 파라미터는 원형 포인트 클라우드의 최소 진원도를 설정하는 데 사용됩니다.

기본값: 0.5

권장값: 실제 상황에 근거하여 설정합니다.

#### 최대 진원도

설명: 이 파라미터는 원형 포인트 클라우드의 최대 진원도를 설정하는 데 사용됩니다.

기본값: 1

권장값: 실제 상황에 근거하여 설정합니다.

#### 최소 불룩한 정도

설명: 이 파라미터는 원형 포인트 클라우드의 최소 불룩한 정도를 설정하는 데 사용됩니다.

기본값: 0.8

권장값: 실제 상황에 근거하여 설정합니다.

#### 최대 불룩한 정도

설명: 이 파라미터는 원형 포인트 클라우드의 최대 불룩한 정도를 설정하는 데 사용됩니다.

기본값: 1

권장값: 실제 상황에 근거하여 설정합니다.

#### 최소 관성

설명: 이 파라미터는 장축 길이에 대한 단축 길이의 최소 비율 역치를 설정하는 데 사용됩니다.

기본값: 0.6

권장값: 실제 상황에 근거하여 설정합니다.

#### 최대 관성

설명: 이 파라미터는 장축 길이에 대한 단축 길이의 최대 비율 역치를 설정하는 데 사용됩니다.

기본값: 1

권장값: 실제 상황에 근거하여 설정합니다.

#### 포인트 클라우드 용량 필터(CloudCapacityFilter)

포인트 클라우드 용량 필터: 포인트 클라우드 리스트에 포인트 수가 역치 범위를 초과한 포인트 클라우드를 제거합니다.

##### 최소 포인트 수 역치

설명: 이 파라미터는 최소 포인트 수를 설정하는 데 사용됩니다.

기본값: 6000

권장값: 실제 상황에 근거하여 설정합니다.

유효 범위: (0, maxPointsNum)

##### 최대 포인트 수 역치

설명: 이 파라미터는 최대 포인트 수를 설정하는 데 사용됩니다.

기본값: 360000

권장값: 실제 상황에 근거하여 설정합니다.

유효 범위: (minPointsNum, +∞)

#### 사각형 포인트 클라우드 필터(RecCloudsFilter)

##### 최소 직각도 역치

설명: 이 파라미터는 최소 직각도의 크기를 조절하는 데 사용됩니다. 직사각형은 포인트 클라우드의 실제 면적과 최소 외접 직사각형의 면적의 비율과 같습니다.

기본값: 0.9000

권장값: 실제 상황에 근거하여 설정합니다.

#### 솔리드 포인트 클라우드 필터(SolidCloudsFilter)

##### 비율의 역치

설명: 이 파라미터는 포인트 클라우드와 매우게 된 실제 물체 사이즈의 최소 비율입니다.

기본값: 0.9000

권장값: 실제 상황에 근거하여 설정합니다.

유효 범위: (0, 1]

#### 완전히 채워진 포인트 클라우드 제거

설명: 이 파라미터는 솔리드 포인트 클라우드를 제거할지 여부를 결정하는 데 사용됩니다. 이 파라미터를 확인한 후 솔리드 포인트 클라우드가 제거되고, 그렇지 않으면 빈 포인트 클라우드가 제거됩니다.

기본값: 선택함.

권장값: 실제 상황에 근거하여 설정합니다.

### 4.3.59. 포인트 클라우드 모양 감지기

#### 기능 설명

평면 혹은 원기둥 모양의 포인트 클라우드를 감지하고 출력합니다. 파라미터에서 필요한 모양을 선택할 수 있습니다.

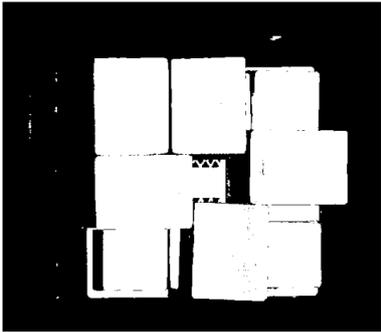
#### 응용 시나리오

일반적으로 평면 혹은 원기둥 모양의 물체를 감지하는 데 사용됩니다.

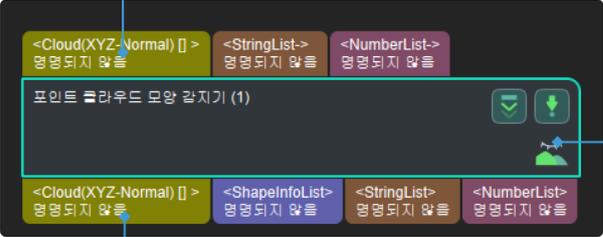
원기둥 포인트 클라우드의 픽 포인트를 계산할 수요가 있으면 이 스텝 뒤에 [원기둥의 픽 포인트를 계산하기](#) 혹은 [옵셋에 따라 포즈를 조정하기](#) 스텝을 연결할 수 있습니다.

#### 입력 및 출력

이 포트로 입력된 포인트 클라우드들의 모양이 감지될 것입니다.



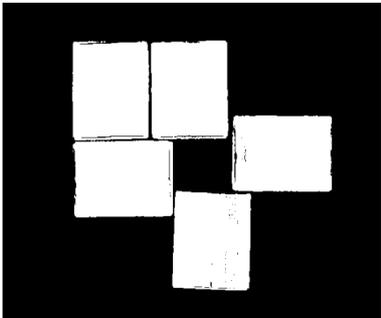

---



시각화 출력 결과



감지된 해당 모양의 포인트 클라우드 리스트



## 파라미터 설명

### 감지할 모양

조절 설명: 이 파라미터는 감지 대상의 모양 유형을 선택하는 데 사용되며 평면(plane), 원기둥(cylinder) 및 구체(sphere)의 세 가지 옵션이 있습니다.

기본값: 평면

권장값: 실제 상황에 따라 결정합니다.

### 감지 설정

#### 탐지기

조절 설명: 이 파라미터는 감지 과정에서 사용하는 알고리즘을 선택하는 데 사용되며 Method 1 및

Method 2 이 두 가지 알고리즘이 있습니다. (나중에 최적화될 예정이며 Method 1 알고리즘만 존재합니다.)

기본값: Method 1

권장값: Method 2

### 법선 방향 편차 최대 역치

조절 설명: 이 파라미터는 실제 이미지에서 포인트의 법선 방향과 투영 후 이론적으로 해당하는 표준 모양의 포인트에 해당하는 법선 방향 사이의 각도 편차의 상한을 나타내며 단위는 °로 실제 이미지의 차이를 나타냅니다. 이는 법선 방향과 이론상 법선 방향 사이의 오차 범위의 상한선입니다.

기본값: 30.0000

권장값: 실제 상황에 따라 설정합니다.

### 허용 가능한 포인트 클라우드 변동

조절 설명: 이 파라미터는 포인트 클라우드 파동 오차의 허용 범위를 정의합니다. 단위는 밀리미터(mm)입니다.

기본값: 5.000 mm

권장값: [3.000, 5.000].

### 최소 클러스터 간격

조절 설명: 이 파라미터는 포인트 클라우드의 상호 연결성을 결정하는 간격의 최소값을 조절하는 데 사용됩니다. 단위는 밀리미터(mm)입니다. 값을 크게 조절하면 멀리 있는 포인트도 동일한 범주로 분류되고, 값을 작게 조절하면 가까운 포인트가 여전히 다른 범주로 분류됩니다.

기본값: 5.000 mm

권장값: 실제 상황에 따라 설정합니다.

### 포인트 클라우드 양자화 해상도

조절 설명: 이 파라미터는 포인트의 개수와 밀도를 조절하는 데 사용됩니다. 단위는 밀리미터(mm)입니다. 파라미터를 크게 조절하면 포인트 클라우드가 밀접해지고 계산 정확도가 향상되지만 계산 시간이 길어지고, 파라미터가 작게 조절하면 포인트 클라우드가 희박해지며 계산 정확도가 떨어질 수 있지만 계산 시간이 절약됩니다.

기본값: 5.000 mm

권장값: 계산 효과와 시간의 길이를 동시에 확보해야 할 경우에는 “클러스터링 최소 간격”을 2배 정도 취하는 것이 좋습니다.

### 출력된 결과의 개수

조절 설명: 이 파라미터는 각 입력된 포인트 클라우드의 최종 출력된 결과 수를 조절하는 데 사용됩니다.

기본값: 1

권장값: 실제 상황에 따라 설정합니다.

### 포인트 클라우드 수로 정렬

조절 설명: 이 파라미터는 포인트 클라우드의 수에 따라 정렬할지를 결정하는 데 사용됩니다.

기본값: 선택함

권장값: 실제 상황에 따라 설정합니다.

### 성공 확률(0~1.0)

조절 설명: 이 파라미터는 해당 모양을 감지하기에 충분한 성공률이 있는지 확인하기 위해 감지 성공률의 크기를 조정하는 데 사용됩니다.

기본값: 0.9900

권장값: 실제 상황에 따라 설정합니다.

### 후보 필터

#### 최소 커버 비율(0~1.0)

조절 설명: 감지된 모양이 있는 포인트의 수량이 총 수에 대한 비율입니다. 이 파라미터 설정된 값보다 작은 모양의 포인트는 필터링됩니다.

기본값: 0.1000

권장값: 실제 상황에 따라 설정합니다.

#### 최소 평균 거리 오차(0~1.0)

조절 설명: 실제 포인트와 감지된 이론적 모델에 있는 해당 포인트 사이의 평균 거리 오류입니다. 단위는 밀리미터(mm)입니다. 값을 크게 조절하면 오류가 큰 포인트 클라우드는 필터링되지 않고, 값을 작게 조절하면 오류가 작은 포인트 클라우드로도 필터링됩니다.

기본값: 10.000 mm

권장값: 실제 상황에 따라 설정합니다.

#### 최소 평균 각도 오차(0~90)

조절 설명: 실제 포인트와 감지된 이론적 모델에 있는 해당 포인트 사이의 평균 각도 오류입니다. 단위는 °입니다. 값을 크게 조절하면 오류가 큰 포인트 클라우드는 필터링되지 않고, 값을 작게 조절하면 오류가 작은 포인트 클라우드로도 필터링됩니다.

기본값: 45.0000

권장값: 실제 상황에 따라 설정합니다.

### 원기둥 속성

#### 최소 반경

조절 설명: 이 파라미터는 원기둥 반경의 최소값을 설정하는 데 사용됩니다. 단위는 밀리미터(mm)입니다. 반경이 이 값보다 작은 원기둥이 필터링될 것입니다.

기본값: 10.000mm

권장값: 실제 상황에 따라 설정합니다.

#### 최대 반경

조절 설명: 이 파라미터는 원기둥 반경의 최대값을 설정하는 데 사용됩니다. 단위는 밀리미터(mm)입니다. 반경이 이 값보다 큰 원기둥이 필터링될 것입니다.

기본값: 100.000 mm

권장값: 실제 상황에 따라 설정합니다.

### 원기둥 형태의 파라미터를 최적화함

조절 설명: 이 파라미터는 원기둥 모양의 파라미터(중심점 및 축)의 최적화 여부를 결정합니다.

기본값: 선택하지 않음.

권장값: 선택함.

### 4.3.60. 포인트를 필터링하기

#### 기능 설명

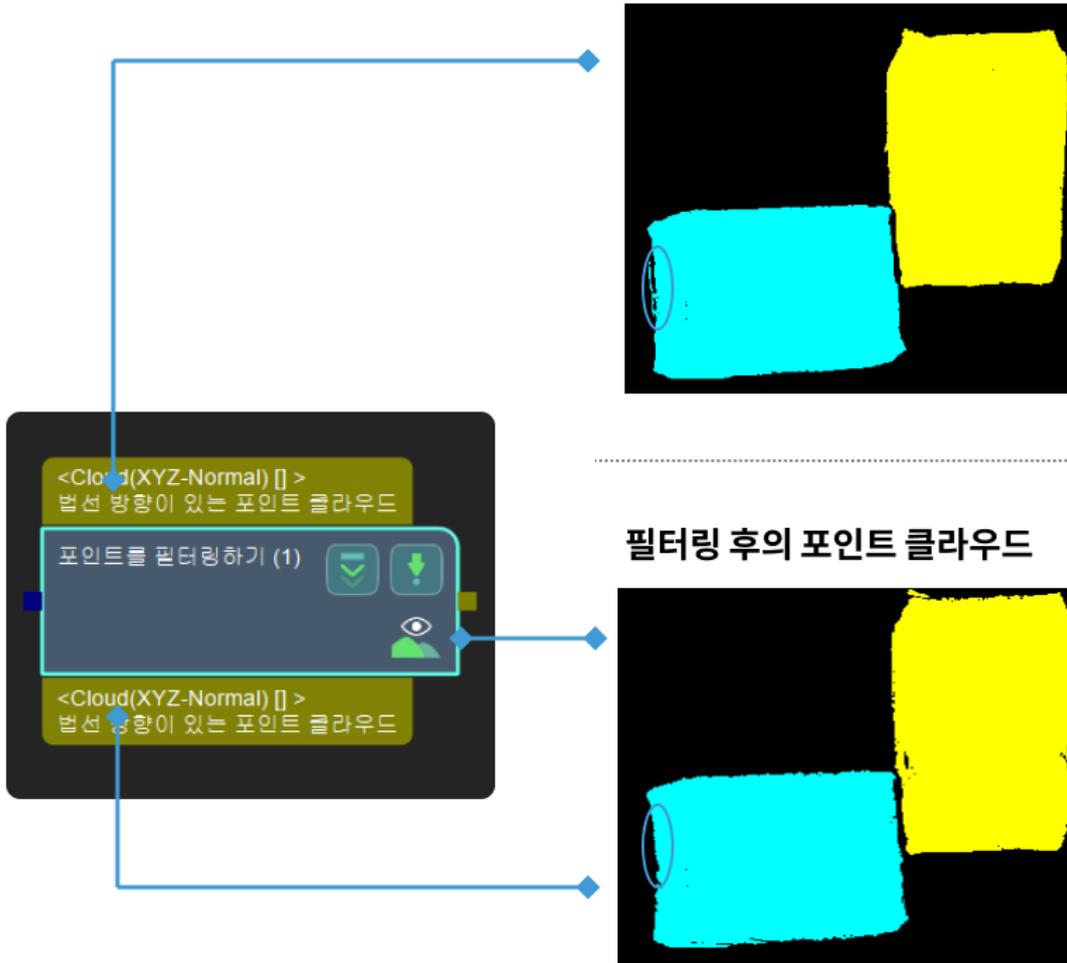
사용자가 지정한 규칙에 따라 포인트 클라우드에 대해 필터링하고 요구에 부합하지 않은 포인트들은 제거될 것입니다.



#### 응용 시나리오

이 스텝은 보통 포인트 클라우드의 사전 처리에 사용되고 심각한 이상치 및 불필요한 부분을 필터링하고 포인트 클라우드 클러스터링 등 후속 처리에 편리를 줍니다.

#### 입력 및 출력

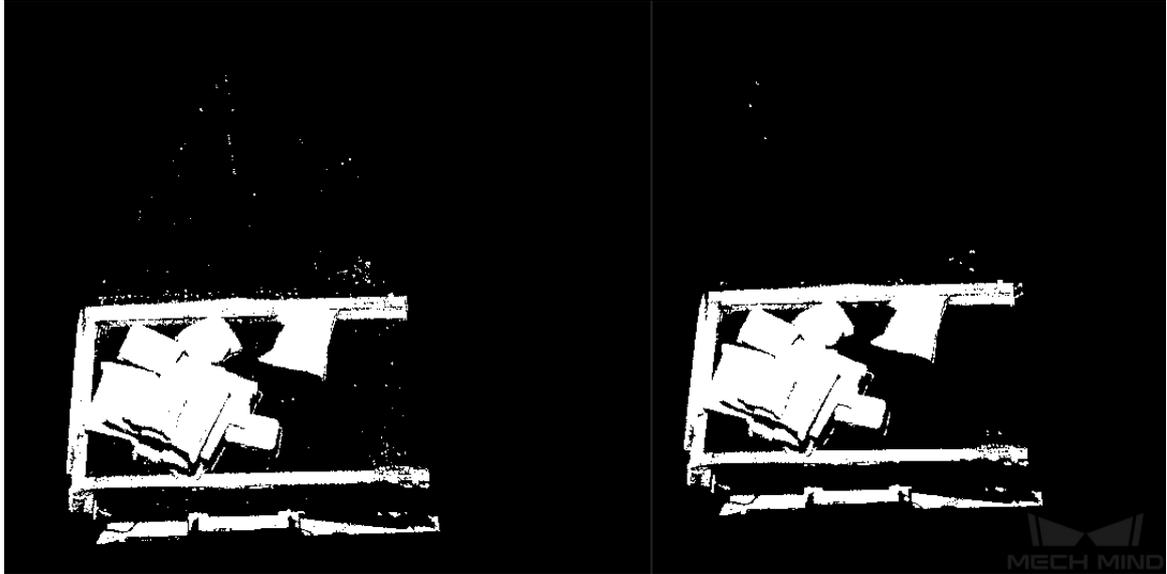


## 파라미터 설명

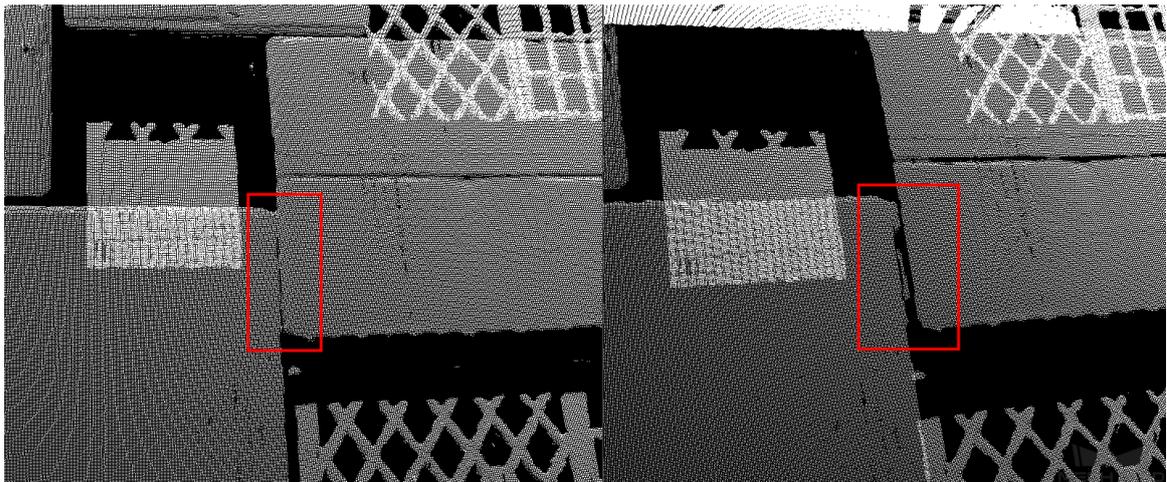
### 포인트 필터 유형

설명: 포인트 클라우드 중의 포인트를 필터링하는 방법은 2가지 있는데 실제 수요에 따라 선택하십시오.

- **StatisticalOutlierFilter**: 통계 필터는 주어진 파라미터에 따라 지정된 범위를 넘은 포인트를 제거합니다. 아래 그림과 같이 포인트 클라우드에 뚜렷한 이상치가 있을 때 사용하는 것이 좋습니다.



- **NormalsFilter:** 법선 방향 필터는 주어진 기준 방향을 통해 포인트 클라우드의 각 포인트와 기준 방향 사이의 각도를 계산하고 지정된 범위를 넘은 포인트를 필터링합니다. 아래 그림의 빨간 박스와 같이 점착을 피하기 위해 포인트 클라우드를 레이어링해야 할 때 사용하는 것이 좋습니다.



### StatisticalOutlierFilter

#### 가장 가까운 포인트의 수

기본값: 30

조절 설명: 이 파라미터는 각 검색에 대한 가장 가까운 포인트의 수를 제어하는 데 사용됩니다. 이 파라미터가 클수록 평균 거리 값을 계산하는 데 사용되는 이웃 포인트의 수가 많아지고 적용 범위가 넓어집니다.

#### 거리 역치 표준 편차 승수

기본값: 2

조절 설명: 이 파라미터는 역치 기준에서 표준 편차의 승수 크기를 제어하는 데 사용됩니다. 이 파라미터가 작을수록 역치가 낮아지고 더 적은 포인트가 유지됩니다. 그러나 포인트 수가 너무 적고 가져오는 정보가 줄어듭니다.

## NormalsFilter

### Reference Direction

#### 로봇 Z축을 사용하기

기본값: 선택하지 않음.

조절 설명: 이 파라미터는 카메라 좌표에 로봇의 Z축을 기준 방향으로 사용할지를 결정합니다. 이 옵션을 선택되면 **기준 방향 X/Y/Z 값** 이 모두 무효로 됩니다.

#### 기준 방향 X 값

기본값: 0

조절 설명: 이 파라미터는 법선 방향 필터링을 위한 기준 축을 설정하는 데 사용됩니다. 이것은 3개의 파라미터로 설정되며, 3개의 파라미터 X, Y 및 Z는 기준 축의 방향을 설정하는 데 사용되는 벡터를 형성합니다. **카메라 좌표에 로봇 Z축을 사용하기** 옵션을 선택하는 경우 위의 3개의 파라미터가 무효로 되고 기준 축을 로봇 Z축으로 설정됩니다.

#### 기준 방향 Y 값

기본값: 0

조절 설명: **기준 방향 X값**과 같습니다.

#### 기준 방향 Z 값

기본값: -1

조절 설명: **기준 방향 X값**과 같습니다.

## Angle Thresholding Settings

### 최소 각도 차이

기본값: 0

조절 설명: 이 파라미터는 포인트를 필터링하는 과정에서 각도 차이의 하한을 제어하는 데 사용되며 단위는 °입니다. 법선 벡터와 참조 벡터 간의 각도 차이가 이 역치보다 낮으면 포인트가 필터링됩니다.

### 최대 각도 차이

기본값: 90

조절 설명: 이 파라미터는 포인트를 필터링하는 과정에서 각도 차이의 상한을 제어하는 데 사용되며 단위는 °입니다. 법선 벡터와 참조 벡터 간의 각도 차이가 이 역치보다 높으면 포인트가 필터링됩니다.

## 4.3.61. 포즈 필터링



이 스텝의 내용은 유지 보수가 중단되었습니다. 새 버전의 스텝 **기준 방향과의 각도에 근거하여 포즈 유효성을 판단하기**를 사용하십시오. 이 스텝에 대한 자세한 정보가 필요한 경우 ([docs@mech-mind.net](mailto:docs@mech-mind.net))로 문의하십시오.

### 기능 설명

입력된 포즈의 Z 방향과 기준 방향 사이의 각도를 계산하여 설정된 각도 역치보다 작은 포즈는 보류되며 역치보다 큰 포즈는 제거될 것입니다.

## 응용 시나리오

이 스텝은 구버전이고 새버전 **기준 방향과의 각도에 근거하여 포즈 유효성을 판단하기로** 바뀌어 사용하세요. 구 버전 스텝을 통하면 Z축과 기준 방향 사이의 각도만 계산할 수 있는데 새 버전에는 모든 축과 기준 방향 사이의 각도를 계산하여 포즈 필터링을 실현할 수 있습니다.

## 입력 및 출력



### 4.3.62. 출력

#### 기능 설명

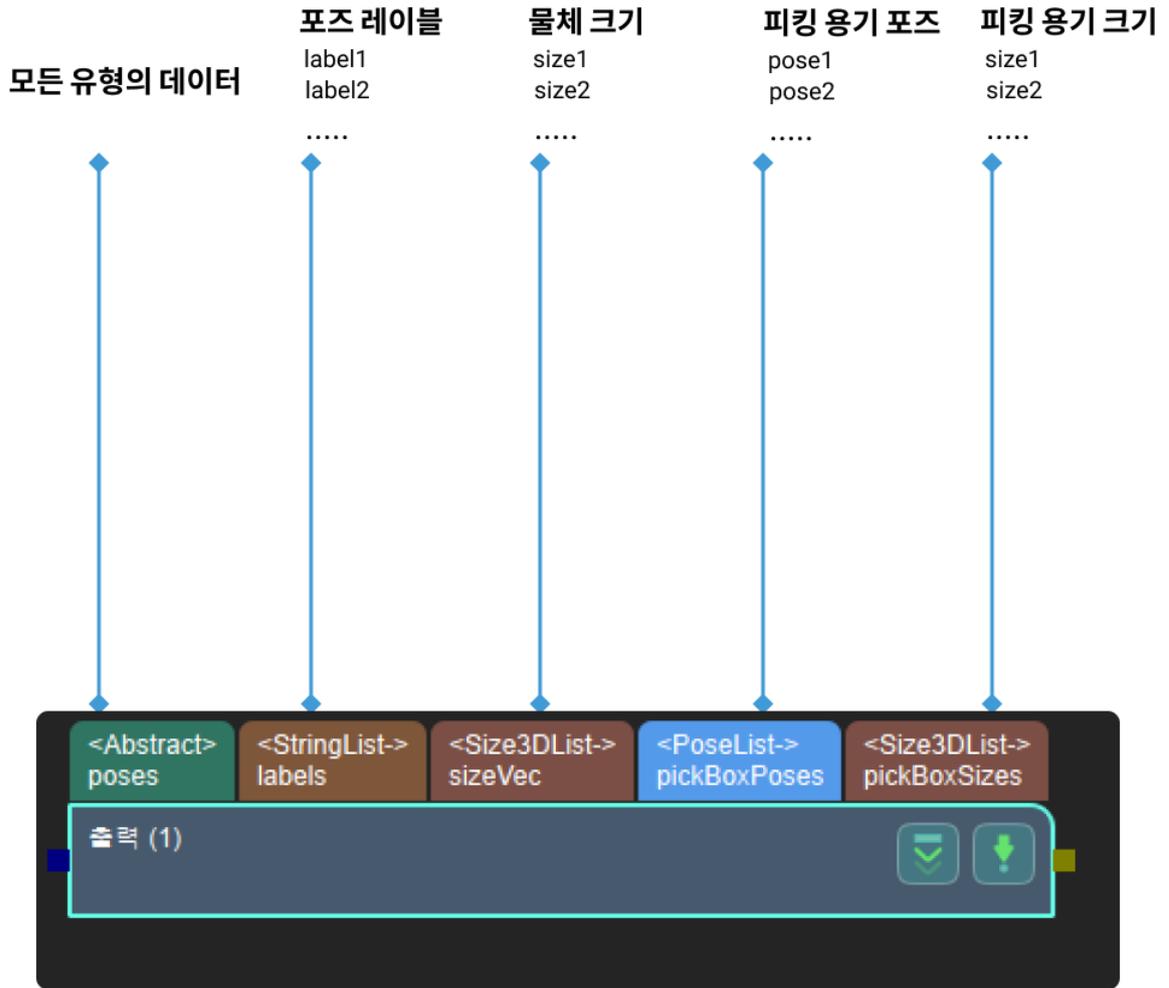
이 스텝은 현재 프로젝트의 출력 결과를 백그라운드 서비스로 보낼 수 있습니다.

#### 응용 시나리오

이 스텝은 일반적으로 다양한 비전 프로젝트 결과를 Mech-Viz 또는 Mech-Center로 보내는 데 사용됩니다.

#### 입력 및 출력

프로젝트에 경로 계획 관련 스텝이 포함되어 있지 않으면 입력 및 출력은 아래 그림과 같습니다.



## 파라미터 설명

### 포트 유형

입력 포트의 유형을 선택하는 데 사용됩니다.

값 리스트:

- 자체 정의: 구체적인 수요에 근거하여 설정해 주십시오.
- 미리 정의 (비전 결과): 기본값, 프로젝트에 경로 계획 관련 스텝이 포함되지 않은 경우에 사용됩니다.
- 미리 정의 (로봇 경로): 프로젝트에 경로 계획 관련 스텝이 포함된 경우에 사용됩니다.

### 자체 정의

파라미터 설명: 외부 통신 장치에 출력하는 데이터 유형을 설정하는 데 사용됩니다.

작업 프로세스:

1. [ 외부 통신 출력 데이터 유형 설정 ]을 클릭하면 출력 설정 도우미로 이동합니다.
2. 통신 키를 미리 설정 또는 통신 키를 자체 정의에 포트를 추가하십시오.
  - 통신 키를 미리 설정: 필요한 포트의 확인란을 선택하십시오.

- 통신 키를 자체 정의: 이 창의 빈 영역을 클릭한 후 [추가]를 클릭하여 포트 이름을 입력하십시오.



자체 정의한 통신 키는 이름을 변경해야만 포트가 작동합니다.

3. 다음으로 [확인] 버튼을 클릭하면 설정이 완료됩니다.

### 4.3.63. 3D 포인트 클라우드를 2D 이미지로 투영하기

#### 기능 설명

이 스텝은 지정된 투영 유형에 의해 3D 포인트 클라우드를 2D 이미지로 전환하는 데 사용됩니다.

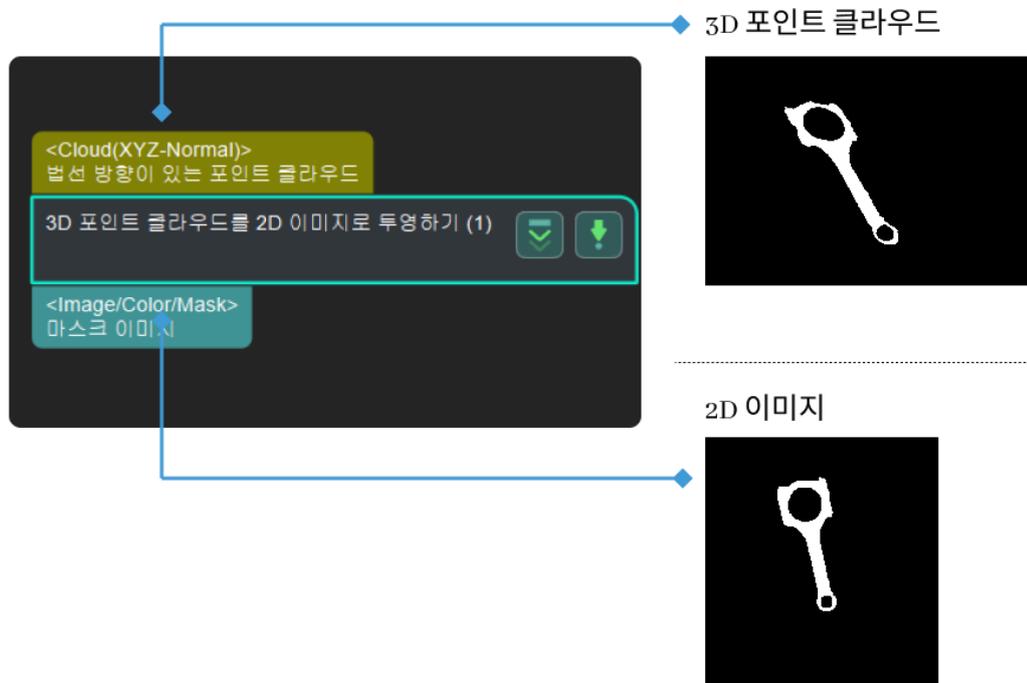
#### 응용 시나리오

이 스텝은 포인트 클라우드 사전 처리 단계에 사용됩니다. 먼저 3D 포인트 클라우드를 2D 이미지에 투영하고 2D 이미지 처리(예:2D형태학 처리)를 한 다음에 스텝 [마스크 안에 대응하는 포인트 클라우드를 추출하기](#)를 통해 3D 포인트 클라우드로 전환합니다.

피킹 시나리오에서 투영 유형은 **PerspectiveProjection**을 사용하는 것을 추천합니다.

측정 시나리오에서 **OrthographicProjection**을 사용하는 것을 추천하며 스텝 [직교 투영](#)을 바꾸는 것을 추천합니다.

#### 입력 및 출력



#### 파라미터 설명

## 투영 유형

파라미터 설명: 이 파라미터는 포인트 클라우드를 2D 이미지로 변환하는 투영 유형을 지정하는 데 사용됩니다.

값 리스트:

- **PerspectiveProjection**: 투시 투영, 핑킹 시나리오에 적용합니다.
- **OrthographicProjection**: 직교 투영, 측정 시나리오에 적용합니다.

기본값: PerspectiveProjection

## PerspectiveProjection

### 이미지 유형

파라미터 설명: 이 파라미터는 카메라에서 출력된 2D 맵 유형을 지정하는 데 사용됩니다.

값 리스트:

- **Color**: 컬러 맵.
- **Depth**: 뎀스 맵.

기본값: Color

### 팽창 커널 사이즈

파라미터 설명: 이 파라미터는 팽창 커널의 사이즈를 설정하는 데 사용되며 단위는 픽셀(px)입니다. 팽창 커널은 물체의 면적을 늘리고 물체의 훼손 부분을 연결할 수 있습니다. 이 값이 클수록 팽딩하는 부분이 많아지고 작을수록 팽딩하는 부분이 적어집니다.

기본값: 21px

이미지 설명: 커널(kernel) 사이즈가 증가하는 이미지 효과는 다음 그림과 같습니다(그림에서 1부터 3까지 커널의 사이즈가 점차 커짐).



### 침식 커널 사이즈

파라미터 설명: 이 파라미터는 침식 커널의 사이즈를 설정하는 데 사용되며 단위는 픽셀(px)입니다. 침식 커널은 작은 소음을 제거하고 연결된 물체를 분리하는 데 사용됩니다. 이 값이 클수록 제거하는 부분이 많아지고 작을수록 제거하는 부분이 적어집니다.

기본값: 21px

이미지 설명: 커널 사이즈가 증가하는 이미지 효과는 다음 그림과 같습니다(그림에서 1부터 3까지 커널의 사이즈가 점차 커짐).



## OrthographicProjection

### 투영 설정

#### 물체의 Z 축 방향에 따라 투영하기

파라미터 설명: 이 파라미터는 포인트 클라우드가 물체의 Z 축 방향에 따라 투영하도록 하는 데 사용됩니다.

기본값: 선택함

#### 스케일 비율

파라미터 설명: 이 파라미터는 투영 후 이미지의 스케일 비율을 설정하는 데 사용됩니다.

기본값: 1000(한 픽셀은 1mm)

#### 닫힘 연산 커널 사이즈

파라미터 설명: 이 파라미터는 닫힘 연산 커널의 사이즈를 설정하는 데 사용되며 단위는 픽셀(px)입니다. 물체 내부의 작은 구멍을 채우고, 인접한 물체와 연결하고, 경계를 매끄럽게 만드면서 면적을 크게 변경하지 않습니다. 이 값이 클수록 효과가 더 뚜렷해집니다.

기본값: 7px

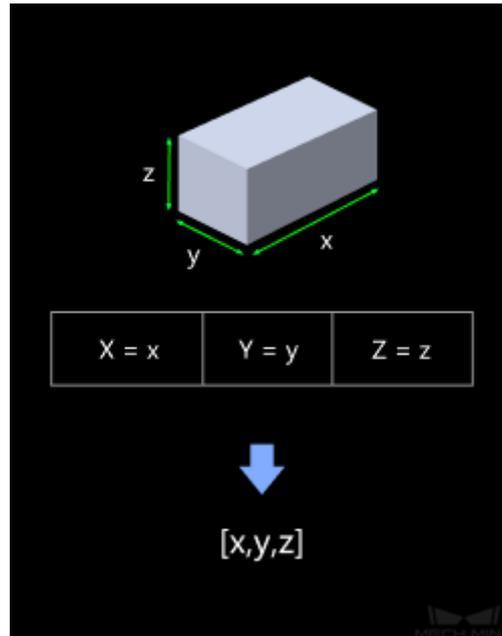
이미지 설명: 커널 사이즈가 증가하는 이미지 효과는 다음 그림과 같습니다(그림에서 1부터 3까지 커널의 사이즈가 점차 커짐).



## 4.3.64. 물체 사이즈 읽기

### 기능 설명

이 스텝은 물체 사이즈 리스트를 읽어냅니다.



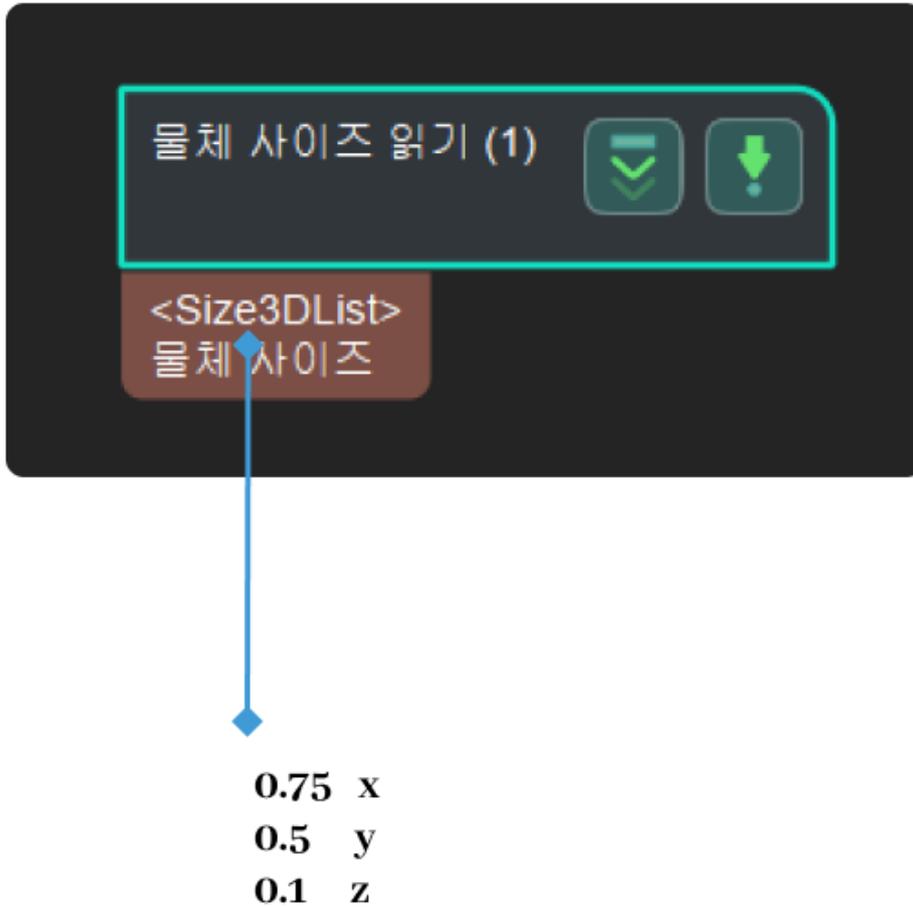
### 응용 시나리오

이 스텝에서 물체의 크기를 입력하는 방식은 두 가지가 있습니다.

1. 물체 크기 JSON 파일.
2. 파라미터.

이 스텝은 일반적으로 **디팔레타이징 프로젝트**에서 상자의 크기를 동적으로 설정할 때 혹은 물체 크기에 의해 후속 포인트 클라우드를 분류할 때 사용됩니다.

### 입력 및 출력



#### 파라미터 설명

#### 읽기 설정

#### 파라미터에서 사이즈를 읽기

기본값: 선택함.

조절 설명: 이 파라미터를 선택하면 **상자 크기 설정**에서 X, Y, Z축에 상자의 실제 길이를 입력할 수 있으며, 이 스텝을 실행할 때 파라미터에서 채워진 크기 값을 읽습니다. **False**로 설정하면 아래 그림과 같이 지정된 파일 "boxSizesList.json"의 파라미터를 통해 물체의 크기 정보를 읽어오며 파일 내용은 [0.4, 0.3, 0.2]이고 해당 상자의 길이,너비와 높이는 0.4m, 0.3m, 0.2m입니다.

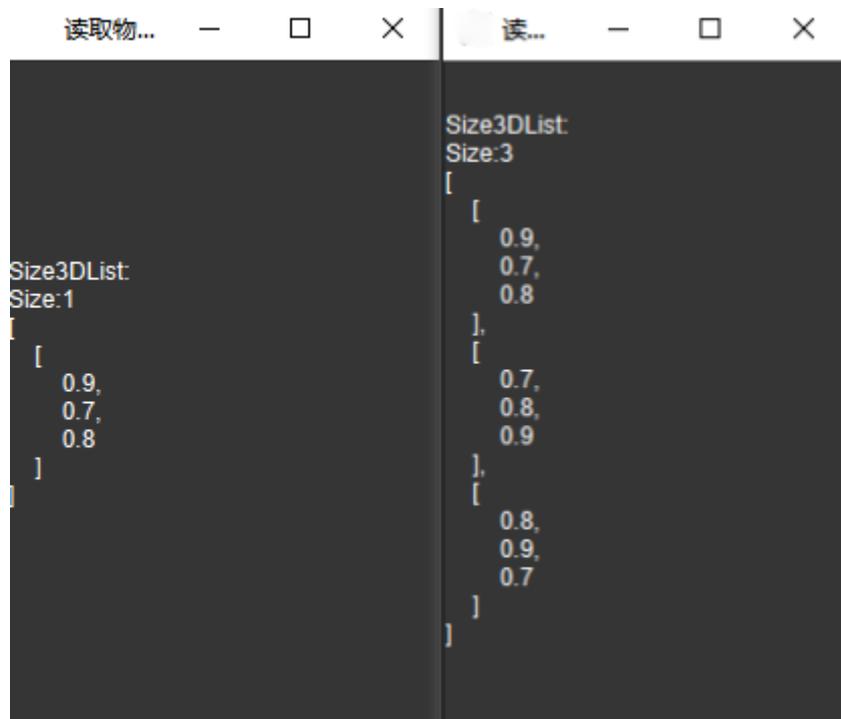


### 상자 사이즈 설정

#### 자동으로 상자 사이즈를 생성하기

기본값: 선택하지 않음.

조절 설명: 이 파라미터는 다중 템플릿 매칭에 사용되며 일반적으로 사용되지 않습니다. 기본적으로 아래 그림의 왼쪽과 같이 사이즈 값 세트가 출력되는데, 이 파라미터를 체크하면 사이즈 값 3세트가 출력되며 원래의 x/y/z 값이 출력됩니다. 파라미터에서는 그림 오른쪽과 같이 xyz/yzx/zxy 형태로 출력됩니다.



#### X축에서의 길이

기본값: 100.000mm

조절 설명: 실제로 물체가 X축에서의 길이를 입력하는 데입니다. 단위는 mm이고 유효 범위는 (0, +∞) 입니다.

### Y축에서의 길이

기본값: 100.000mm

조절 설명: 실제로 물체가 Y축에서의 길이를 입력하는 데입니다. 단위는 mm이고 유효 범위는 (0, +∞) 입니다.

### Z축에서의 길이

기본값: 100.000mm

설명: 실제로 물체가 Z축에서의 길이를 입력하는 데입니다. 단위는 mm이고 유효 범위는 (0, +∞) 입니다.

## 4.3.65. 포인트 클라우드를 읽기



이 스텝의 내용은 유지 보수가 중단되었습니다. 새 버전의 스텝 **포인트 클라우드를 읽기(V2)**를 사용하십시오. 이 스텝에 대한 자세한 정보가 필요한 경우 ([docs@mech-mind.net](mailto:docs@mech-mind.net))로 문의하십시오.

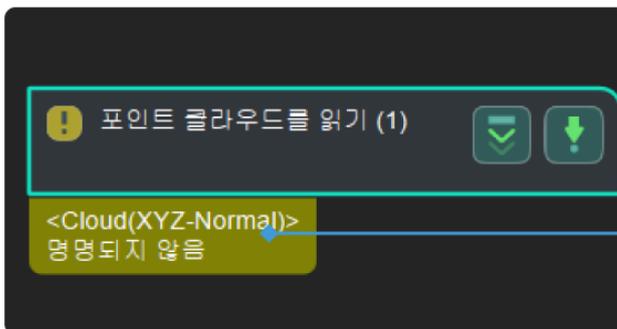
### 기능 설명

파일 혹은 폴더에서 포인트 클라우드를 읽습니다.

### 응용 시나리오

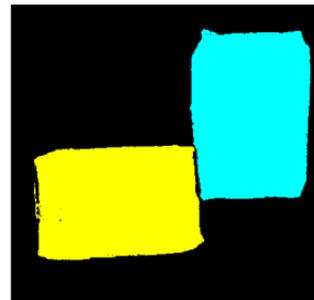
경로를 지정하여 PLY 형식의 포인트 클라우드를 읽어옵니다.

### 입력 및 출력



입력 없음

법선 방향을 갖춘 3D 포인트 클라우드



## 4.3.66. 파일에서 포즈 정보 획득

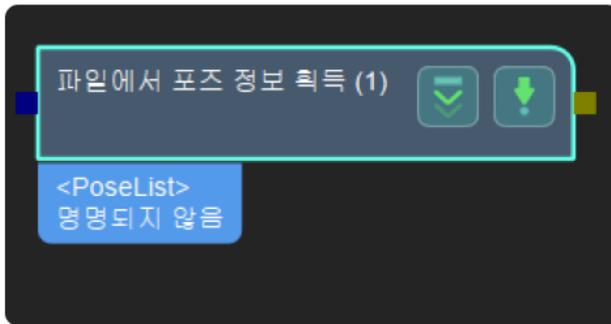
### 기능 설명

로컬 파일에서 포즈 리스트를 읽습니다.

### 응용 시나리오

이 스텝은 참조 포즈의 주요 소스입니다. 현재 프로젝트에 읽을 특정 포즈 파일이 없으면 **포즈를 빠르게 만들기** 스텝을 사용하는 것이 좋습니다.

### 입력 및 출력



입력: 없음

PoseList 형식의 포즈 리스트

pose1  
psoe2  
.....

### 파라미터 설명

#### 포즈 파일 경로

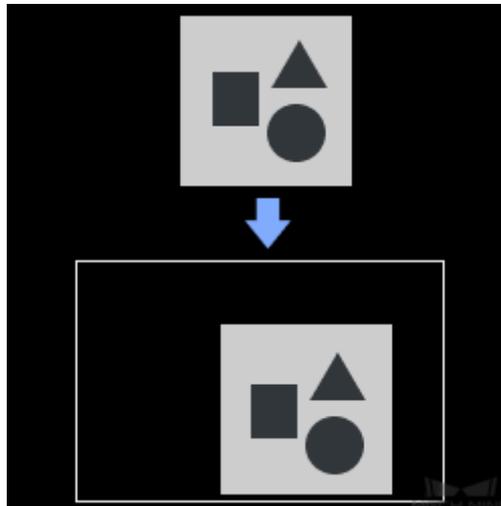
파라미터 해석: 이 파라미터는 포즈 파일의 경로를 선택하는 데 사용됩니다.

조절 설명:  을 클릭하고 원하는 파일을 선택합니다.

## 4.3.67. 2D ROI 내의 조정된 이미지를 복원하기

### 기능 설명

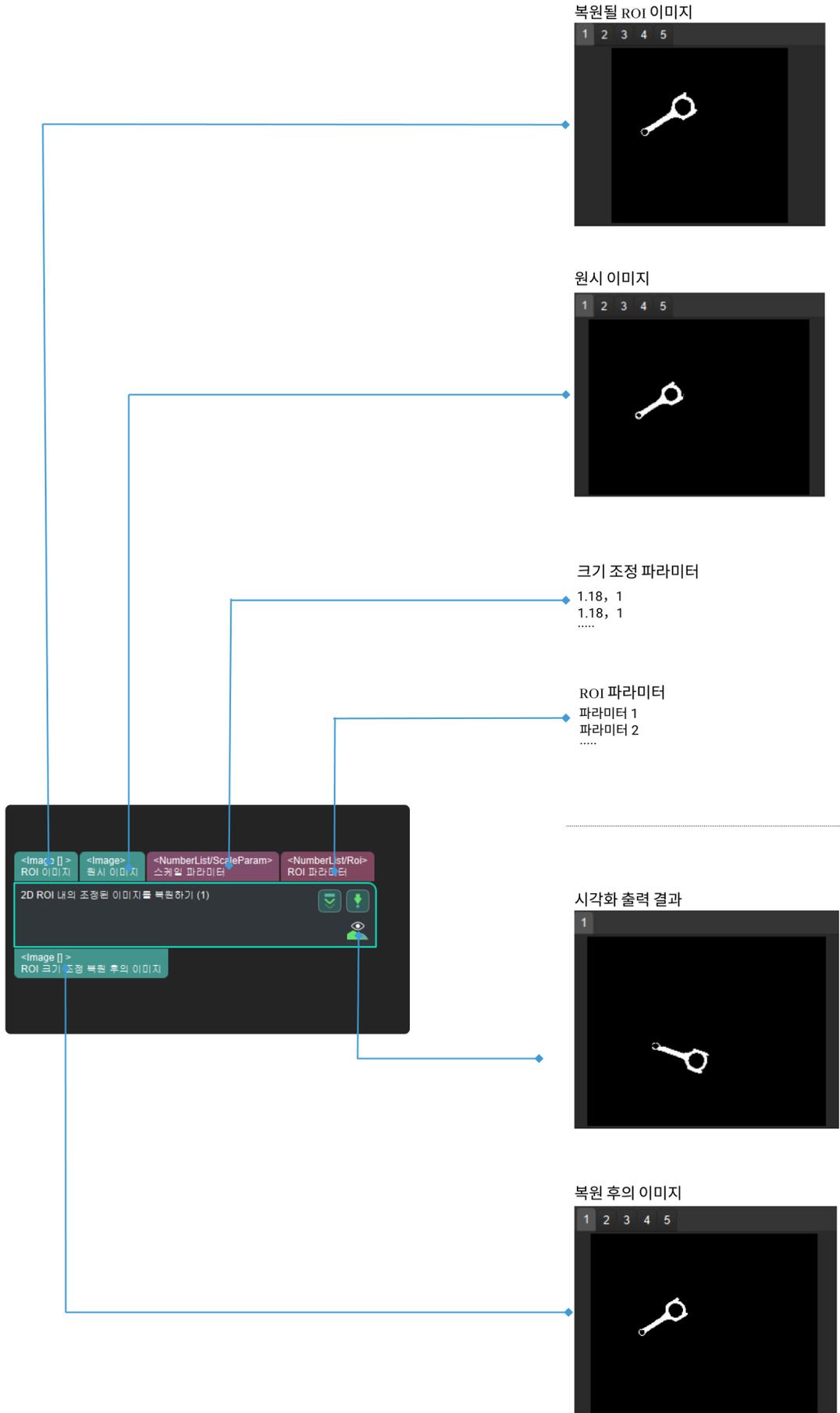
이 스텝을 사용하여 입력된 이미지의 크기를 스케일 비율에 따라 복원하고 지정된 크기의 검은색 배경에 넣어 둡니다.



## 응용 시나리오

이 스텝은 일반적으로 2D ROI 내의 이미지를 조정하기 스텝 및 딥 러닝과 관련된 스텝과 함께 사용됩니다.

## 입력 및 출력



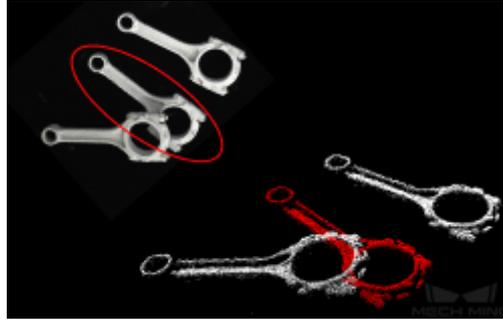
## 파라미터 설명

파라미터가 없음.

### 4.3.68. 겹친 물체를 제거하기

#### 기능 설명

이 스텝은 사용자가 자체 정의한 규칙에 따라 겹친 물체의 포즈를 제거합니다. 아래 그림과 같이 오른쪽 하단의 빨간색 물체는 겹친 물체입니다.



#### 응용 시나리오

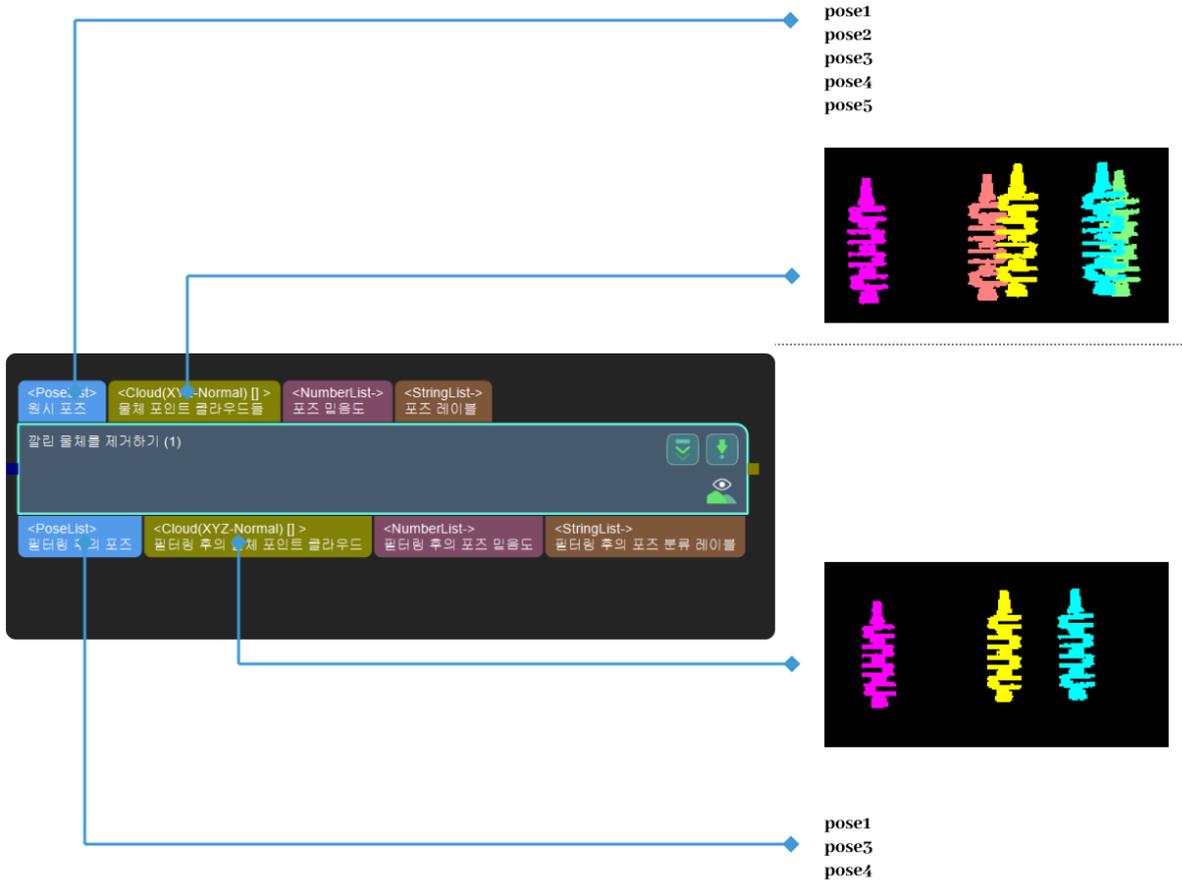
이 스텝은 보통 스텝 [3D상세 매칭](#) 뒤에 사용되고 구체적인 수요에 따라 매칭된 포즈에 대해 필터링하여 겹친 물체의 포즈를 제거합니다.



[겹친 물체를 제거하기V2](#) 스텝은 “프로젝터(2D)” 메소드에 대해 최적화하고 이 메소드 사용하려면 [겹친 물체를 제거하기V2](#) 스텝을 사용하는 것이 좋습니다.

[겹친 물체를 제거하기V2](#) 스텝은 “경계 박스(3D)” 메소드에 대해 수정하지 않았습니다. “경계 박스(3D)” 메소드를 사용하려면 “겹친 물체를 제거하기” 및 [겹친 물체를 제거하기V2](#) 스텝을 모두 사용할 수 있습니다.

#### 입력 및 출력



## 파라미터 조절 설명

알고리즘 유형이 두 가지 유형을 포함해서 선택된 유형에 따라 파라미터를 조절하십시오.

### BoundingBoxOfObjectIn3DBased (경계 박스 메소드)

이 알고리즘은 물체의 포인트 클라우드의 경계 박스를 작은 입방체로 나누고, 물체와 겹친 물체가 겹치는 작은 입방체의 수를 계산하여 물체와 겹친 물체의 겹침 비율을 계산합니다.

### 역치 설정

#### 겹친 비율의 역치(0~1.0)

파라미터 설명: 겹침 비율은 물체의 경계 박스와 겹친 물체의 경계 박스가 겹치는 부분의 부피 비율을 의미합니다. 어떤 물체의 포인트 클라우드는 겹친 물체의 포인트 클라우드와의 겹친 부분의 비율이 이 값을 초과하면 이 물체의 포인트 클라우드를 제거합니다.

값 범위: 0~1.0

기본값: 0.60

조절 제안: 실제 수요에 근거하여 보폭 0.01에 따라 **겹친 비율의 역치**를 조정합니다.

### 포인트 클라우드 해상도 설정

#### 포인트 클라우드 경계 박스의 해상도

조절 설명: 이 파라미터는 포인트 클라우드 경계 박스의 치수, 즉, 물체의 포인트 클라우드 박스를 나눌 때의 포인트 클라우드 경계 박스의 치수를 설정하는 데 사용됩니다. 그 다음에 포인트 클라우드 경계

박스의 수를 계산을 통해 물체와 겹친 물체 간의 겹친 비율을 계산합니다.

기본 값: 3mm.

권장값: 실제 상황에 따라 설정합니다.

#### \*물체 높이 설정

##### 물체 높이의 계산법

파라미터 설명: 이 파라미터는 물체 높이를 계산하는 방법을 선택하는 데 사용됩니다.

값 리스트:

- UsingInputCloudHeight: 기본값, 입력된 포인트 클라우드에 의해 높이를 계산합니다.
- UsingSpecifiedHeight: 물체 높이를 지정하고 포인트 클라우드가 평평하거나 물체의 완전한 모양을 반영할 수 없는 경우에 사용됩니다.

##### 특정 높이

파라미터 설명: 이 파라미터는 물체의 높이를 지정하는 데 사용되며 단위는 밀리미터(mm)입니다. **물체 높이의 계산법은 특정 높이인 경우 이 파라미터를 설정해야 합니다.**

기본값: 100mm

조절 제안: 이 파라미터는 실제 필요에 따라 설정해야 합니다.

#### 확장 설정

##### 경계 박스 X축에 따른 확장률

파라미터 설명: 경계 박스를 물체 포즈의 X축에 따라 확장하면 겹침을 더 민감하게 감지할 수 있습니다.

기본값: 1.0000

조절 제안: 이 파라미터는 실제 필요에 따라 설정합니다.

##### 경계 박스 Y축에 따른 확장률

파라미터 설명: 경계 박스를 물체 포즈의 Y축에 따라 확장하면 겹침을 더 민감하게 감지할 수 있습니다.

기본값: 1.0000

조절 제안: 이 파라미터는 실제 필요에 따라 설정합니다.

##### 경계 박스 Z축에 따른 확장률

파라미터 설명: 경계 박스를 물체 포즈의 Z축에 따라 확장하면 겹침을 더 민감하게 감지할 수 있습니다. **물체 높이의 계산법은 특정 높이인 경우 이 파라미터를 설정할 필요가 없습니다.**

기본값: 3.0000

조절 제안: 이 파라미터는 실제 필요에 따라 설정합니다.

#### ProjectionOfObjectIn2DBased(프로젝터 메소드)

이 알고리즘은 물체가 2D 평면에 투영한 겹친 영역의 비율을 계산하여 물체가 겹치었는지 여부를 판단합니다.

#### 역치 설정

### 겹친 비율의 역치(0~1.0)

파라미터 설명: 겹친 비율은 물체의 경계 박스와 겹친 물체의 경계 박스가 겹치는 부분의 부피 비율을 의미합니다. 어떤 물체의 포인트 클라우드를 겹친 물체의 포인트 클라우드와의 겹친 부분의 비율이 이 값을 초과하면 이 물체의 포인트 클라우드를 제거합니다.

값 범위: 0~1.0

기본값: 0.60

조절 제안: 실제 수요에 근거하여 보폭 0.01에 따라 **겹친 비율의 역치**를 조정합니다.

### 물체 높이 설정

#### 물체 높이의 계산법

파라미터 설명: 이 파라미터는 물체 높이를 계산하는 방법을 선택하는 데 사용됩니다.

값 리스트:

- UsingInputCloudHeight: 기본값, 입력된 포인트 클라우드에 의해 높이를 계산합니다.
- UsingSpecifiedHeight: 물체 높이를 지정하고 포인트 클라우드가 평평하거나 물체의 완전한 모양을 반영할 수 없는 경우에 사용됩니다.

#### 특정 높이

파라미터 설명: 이 파라미터는 물체의 높이를 지정하는 데 사용되며 단위는 밀리미터(mm)입니다. **물체 높이의 계산법은 특정 높이인 경우 이 파라미터를 설정해야 합니다.**

기본값: 100mm

조절 제안: 이 파라미터는 실제 필요에 따라 설정합니다.

### 고급 계산 설정

#### 마스킹 클로징 커널의 사이즈

조절 설명: 이 파라미터는 마스크 클로징 커널 사이즈를 조절하는 데 사용됩니다.

기본값: 3

권장값: 실제 상황에 따라 설정합니다.

#### 덱스 맵 팽창 커널 사이즈

조절 설명: 이 파라미터는 덱스 맵 팽창 커널 사이즈를 조절하는 데 사용됩니다.

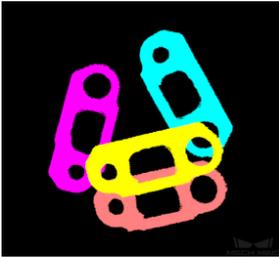
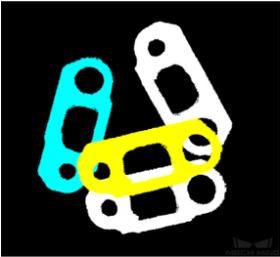
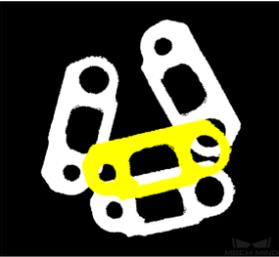
기본값: 4

권장값: 실제 상황에 따라 설정합니다.

### 조절 예시

#### 겹친 비율의 역치

**겹친 비율의 역치**가 각각 0.60, 0.10, 0.03으로 설정된 경우 이 스텝의 시각화 출력 결과는 아래 표와 같습니다. 그 중에 제거된 물체의 포인트 클라우드는 흰색으로 표시됩니다.

역치 값 범위	0.60	0.10	0.03
시각화 출력 결과			

### 4.3.69. 겹치는 다각형을 제거하기

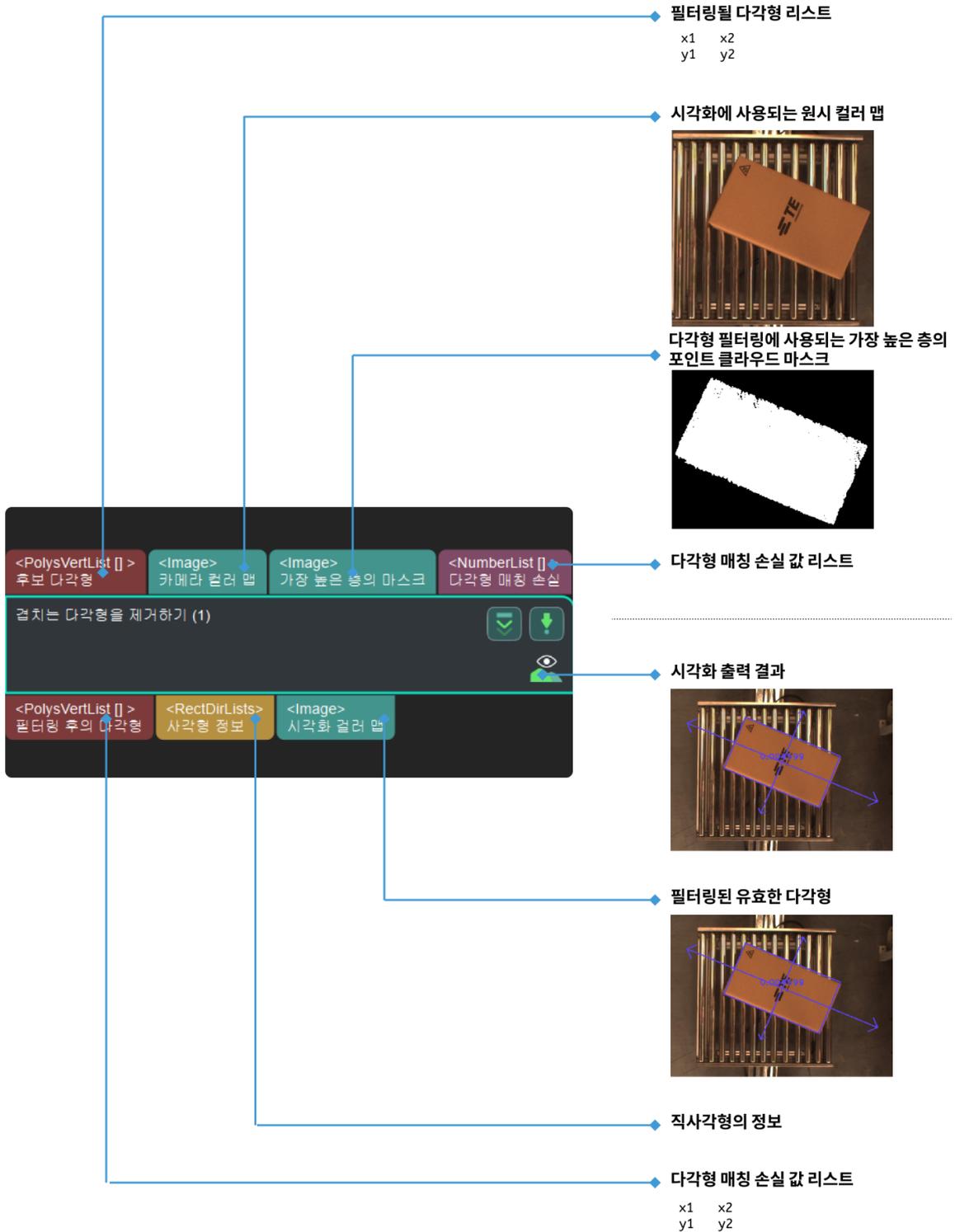
#### 기능 설명

겹치거나 마스크 외부에 있는 무효한 다각형을 제거합니다.

#### 응용 시나리오

고정적으로 **마스크 외부에 있는 다각형을 제거하기** 및 **직사각형의 사이즈와 포즈를 계산하기** 스텝과 함께 사용합니다. 실제 물체의 윤곽을 반영하지 않는 다각형을 필터링하는 데 사용됩니다.

#### 입력 및 출력



파라미터 설명

마스크 설정

입력된 마스크의 사용법

기본값 : SingleMask

값 리스트: SingleMask(하나의 마스크를 모든 다각형에 적용함), CorrespondingMasks(다각형마다 다른 마스크를 사용함)

조절 설명: 마스크 사용법을 설정하는 것입니다. 하나의 마스크를 모든 다각형에 적용하거나 다각형마다 다른 마스크를 사용하는 것을 선택할 수 있습니다.

## 검사 기준

### 에지를 초과하는 비율의 최소 역치

기본값: 0.4500

조절 설명: 상자의 마스크를 긴 축과 짧은 축의 양수 방향과 음수 방향으로 각각 길이의 절반과 너비의 절반으로 평행이동하고 평행이동된 마스크 부분에서 최상위 레이어 마스크를 벗어나는 부분의 면적이 마스크 자체 면적의 비율을 계산하고, 이 비율이 역치를 초과하면 상자가 최상위 레이어의 경계에 있다고 판단합니다. 일반적으로 직사각형 상자 마스크가 최상위 레이어 마스크의 모서리에 있는 경우 마스크가 최상위 레이어 마스크에서 벗어나는 방향으로 길이 또는 너비의 절반만큼 평행이동한 비율 값이 0.5이므로 최소 역치 기본값은 0.45입니다. 일반적으로 값을 0.5 이상으로 늘리면 모든 다각형이 필터링됩니다. 이 스텝에서는 모서리에 위치한 상자의 마스크를 우선 처리하므로 경계에 있는 상자 마스크 영역을 먼저 확인해야 합니다.

### 에지를 초과하는 비율의 최대 역치

기본값: 0.6000

조절 설명: 상자의 마스크를 긴 축과 짧은 축의 양수 방향과 음수 방향으로 각각 길이의 절반과 너비의 절반으로 평행이동하고 평행이동된 마스크 부분에서 최상위 레이어 마스크를 벗어나는 부분의 면적이 마스크 자체 면적의 비율을 계산하고, 이 비율이 역치보다 작으면 상자가 최상위 레이어의 경계에 있다고 판단합니다. 일반적으로 직사각형 상자 마스크가 최상위 레이어 마스크의 모서리에 있는 경우 마스크가 최상위 레이어 마스크에서 벗어나는 방향으로 길이 또는 너비의 절반만큼 평행이동한 비율 값이 0.5이므로 최대 역치 기본값은 0.6입니다. 이 스텝에서는 모서리에 위치한 상자의 마스크를 우선 처리하므로 경계에 있는 상자 마스크 영역을 먼저 확인해야 합니다.

### 다각형 겹치는 최대 비율(0~1)

기본값: 0.2000

설명: 두 다각형이 겹치는 비율이 이 역치보다 크면 하나를 제거해야 합니다.

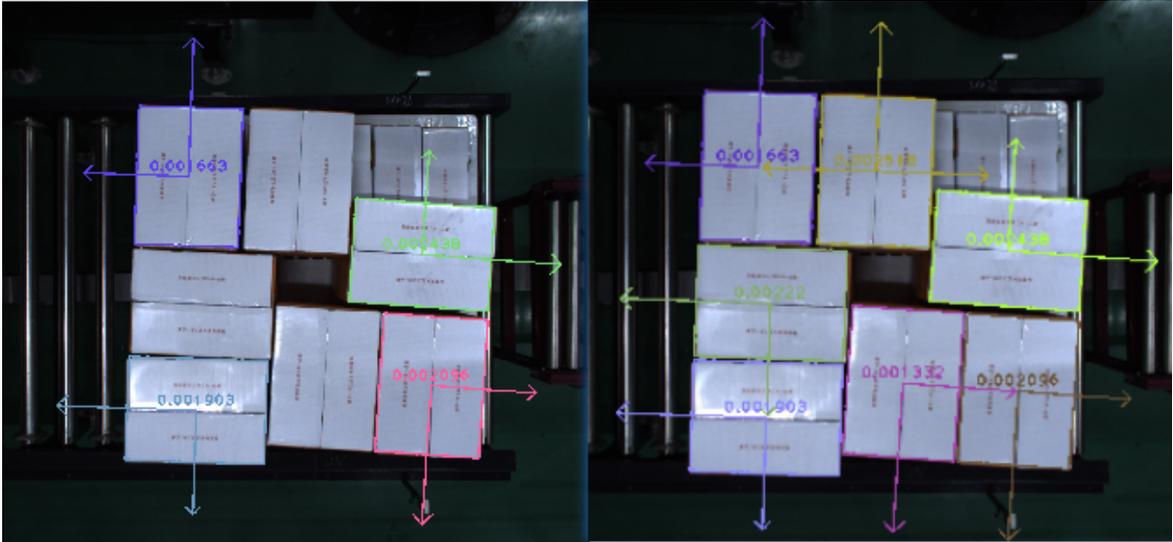
## 출력 전략

### 위치 제한

기본값: AtCorners

값 리스트: AtCorners, None

조절 설명: 아래 그림과 같이 왼쪽은 기본 설정으로 출력된 이미지이고, 오른쪽은 **None**으로 설정 시 출력된 이미지입니다.



### 4.3.70. 마스크 외부에 있는 다각형을 제거하기

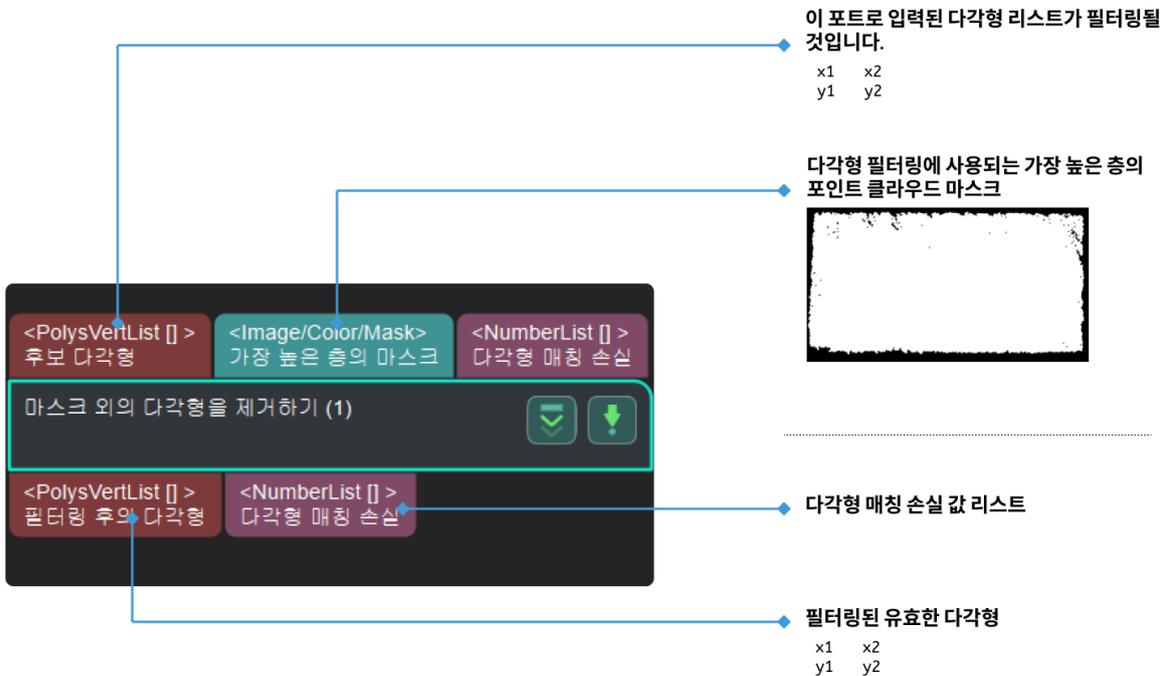
#### 기능 설명

마스크와 겹친 구역의 면적이 설정된 값보다 작은 다각형(간섭을 줄 수 있음)을 제거합니다.

#### 응용 시나리오

고정적으로 2D 매칭 및 겹치는 다각형을 제거하기 스텝과 함께 사용하며 마스크를 통해 매칭하여 얻은 간섭 다각형을 필터링합니다.

#### 입력 및 출력



## 파라미터 설명

### 입력된 마스크의 사용법

기본값: SingleMask

값 리스트: SingleMask(하나의 마스크를 모든 다각형에 적용함), CoorespondinMasks(다각형마다 다른 마스크를 사용함)

조절 설명: 마스크 사용법을 설정하는 것입니다. 하나의 마스크를 모든 다각형에 적용하거나 다각형마다 다른 마스크를 사용하는 것을 선택할 수 있습니다.

### 마스크에 대한 다각형의 겹친 면적 비율의 최소값(0~1.0)

기본값: 0.8000

조절 설명: 이 파라미터로 마스크와 겹친 면적이 차지하는 비율의 역치를 조절할 수 있습니다. 마스크에 대한 다각형의 겹친 면적 비율이 이 역치보다 크면 해당 다각형은 보류되고 이 역치보다 작으면 마스크 외부의 다각형은 제거됩니다.

## 4.3.71. 데이터 반복 및 연결



이 스텝의 내용은 유지 보수 중입니다. 이 스텝에 대한 자세한 정보가 필요한 경우 ([docs@mech-mind.net](mailto:docs@mech-mind.net))로 문의하십시오.

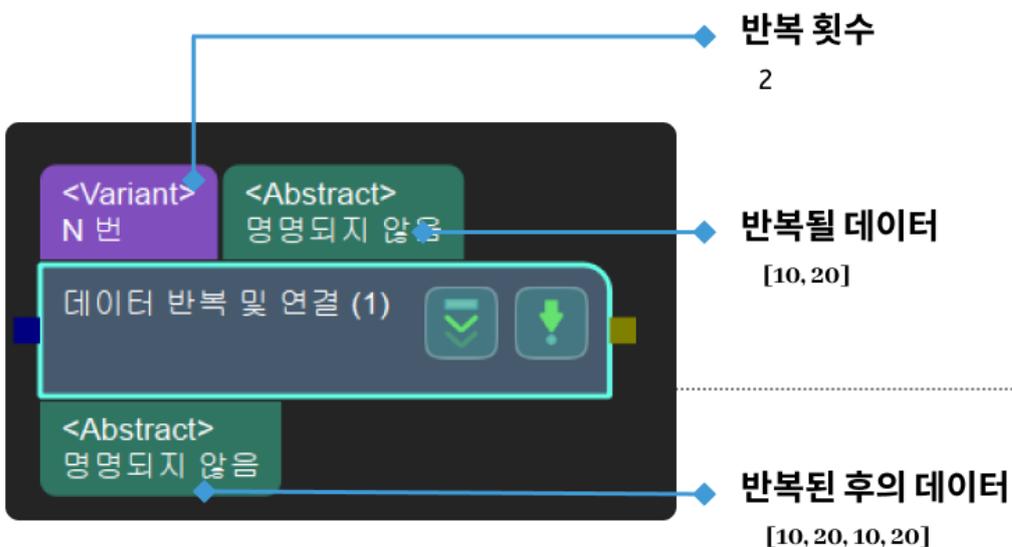
### 기능 설명

입력된 데이터를 설정된 횟수대로 복사하고 앞뒤 연결하여 출력합니다. 예를 들어 원시 데이터 리스트가 [10, 20]이고 두 번 반복해야 하는 경우 출력 결과는 [10, 20, 10, 20]입니다.

### 응용 시나리오

일반적인 데이터 복사 스텝으로 자주 스텝 데이터 리스트에서 지정된 차원의 요소 개수를 통계하기와 함께 사용되어 후속 스텝의 입력된 데이터 리스트 안의 요소 개수를 일치하게 만듭니다.

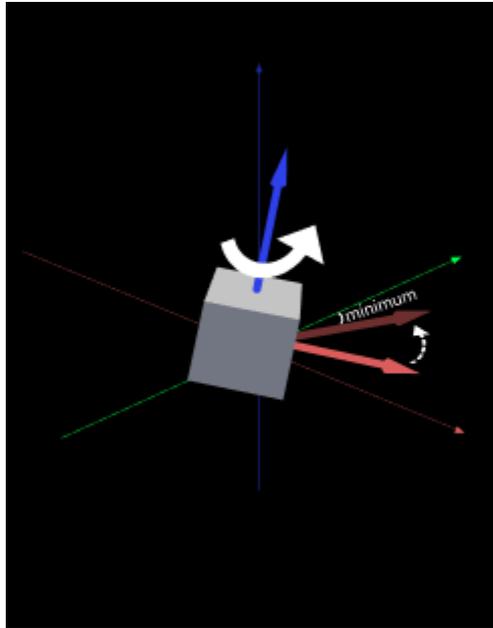
### 입력 및 출력



### 4.3.72. 지정축과 기준 방향의 각도가 최소가 되도록 포즈를 조정하기

#### 기능 설명

이 스텝은 포즈의 특정 방향 축을 중심으로 회전하고 다른 축은 기준 방향과 가장 작은 각도를 가진 방향으로 회전할 수 있습니다.

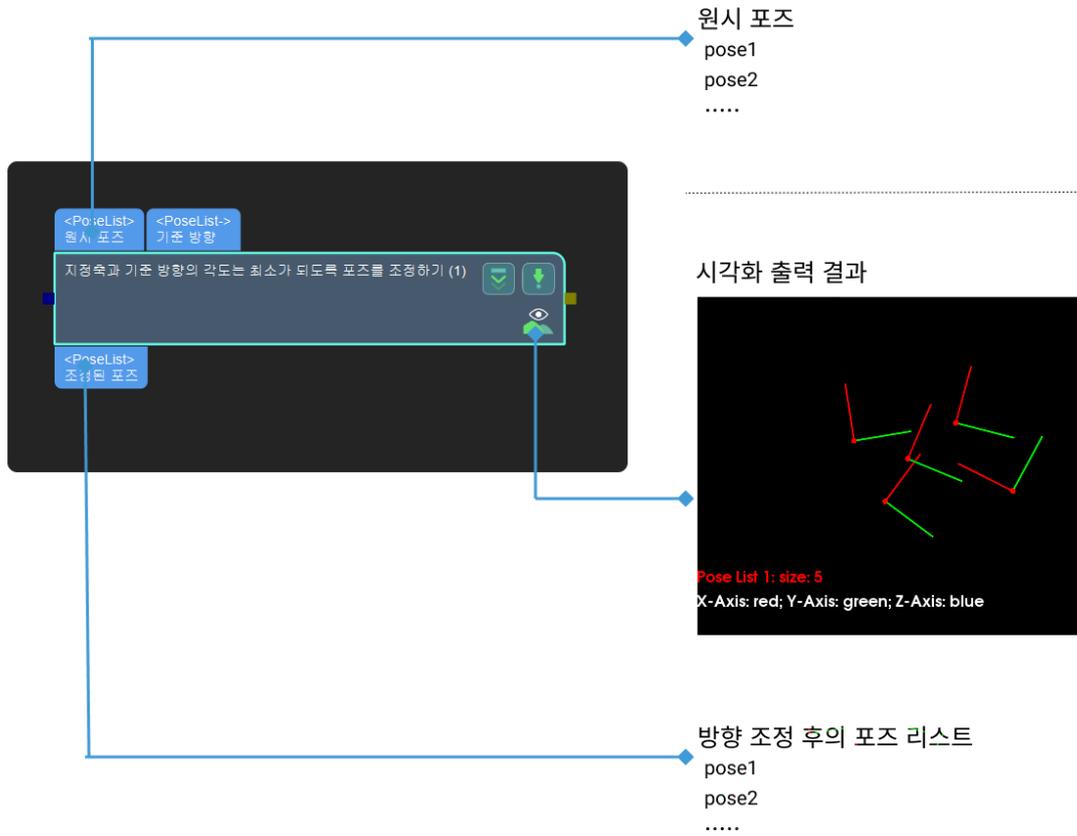


#### 응용 시나리오

이 스텝은 일반적으로 포즈 방향을 조정해야 하는 다양한 시나리오에서 사용됩니다.

이 스텝은 구 버전이며 더 풍부한 기능을 가진 새 버전 스텝 **목표 방향으로 포즈를 회전하기(제약 없음)**로 대체해야 하며 **3D 벡터를 빠르게 만들기** 스텝과 함께 사용됩니다.

#### 입력 및 출력



## 파라미터 설명

### 시각화 설정

#### 포즈 시각화 설정

파라미터 설명: 이 파라미터는 표시할 포즈 유형을 지정하는 데 사용됩니다.

값 리스트:

- 입력 포즈만 표시하기
  - 출력 포즈만 표시하기
  - 입력과 출력 포즈를 함께 표시하기
- 기본값: 출력 포즈만 표시하기

#### 기준점을 사용하기

파라미터 설명: 이 파라미터는 기준 방향을 계산하기 위해 기준점을 사용할지 여부를 선택하는 데 사용됩니다. 이 항목을 체크한 후 **기준점을 사용하기** 파라미터 그룹을 설정해야 합니다.

기본값: 선택하지 않음.

조절 설명: 기준점을 사용하여 기준 방향을 계산할 때만 선택하십시오. 구성된 방향은 입력한 포즈에서 기준점을 가리키는 방향입니다.

#### 기준점 설정

### 상대 Z값을 사용하기

파라미터 설명: 이 파라미터는 기준점의 Z 값을 재정의할지 여부를 선택하는 데 사용됩니다. 기준점 Z 값 = 원시 포즈 Z 값 + 상대 값

기본값: 선택하지 않음.

#### X

파라미터 설명: 이 파라미터는 기준점의 X 값 좌표를 나타냅니다.

기본값: 0.0000

#### Y

파라미터 설명: 이 파라미터는 기준점의 Y 값 좌표를 나타냅니다.

기본값: 0.0000

#### Z

파라미터 설명: 이 파라미터는 기준점의 Z 값 좌표를 나타냅니다.

기본값: 0.0000

### 상대 Z

파라미터 설명: 이 파라미터는 Z 방향의 평행 이동 벡터를 나타냅니다. **상대 Z값을 사용하기**를 선택한 경우 이 파라미터를 설정해야 합니다.

기본값: 0.0000

### 축의 설정

#### 고정축의 방향

파라미터 설명: 이 파라미터는 포즈가 회전할 때 고정 축을 지정하는 데 사용됩니다.

값 리스트: X, Y, Z

기본값: Y

#### 회전해야 할 축의 방향

파라미터 설명: 이 파라미터는 지정된 방향으로 회전하는 축, 즉 회전해야 하는 축을 선택하는 데 사용됩니다.

값 리스트: X, Y, Z

기본값: Z

#### 기준 방향 설정

#### X

파라미터 설명: 이 파라미터는 기준점의 X 방향 벡터를 나타냅니다.

기본값: 0.0000

#### Y

파라미터 설명: 이 파라미터는 기준점의 Y 방향 벡터를 나타냅니다.

기본값: 0.0000

## Z

파라미터 설명: 이 파라미터는 기준점의 Z 방향 벡터를 나타냅니다.

기본값: 1.0000

### 4.3.73. 지정한 축과 각도에 근거하여 포즈를 회전하기



이 스텝의 내용은 유지 보수 중입니다. 이 스텝에 대한 자세한 정보가 필요한 경우 ([docs@mech-mind.net](mailto:docs@mech-mind.net))로 문의하십시오.

#### 기능 설명

지정된 축을 중심으로 지정된 각도만큼 입력한 포즈를 회전합니다.

#### 응용 시나리오

일반적으로 사용되는 포즈 회전 스텝입니다. 고정 용법이 없습니다.

#### 입력 및 출력

##### ● 입력:

1. 이 포트에 입력한 포즈가 회전될 것입니다.
2. 포즈는 이 포트에 의해 입력된 축을 중심으로 회전됩니다.
3. 포즈의 회전 각도입니다.

##### ● 출력:

1. 회전된 후의 포즈.

### 4.3.74. 이미지를 저장하기

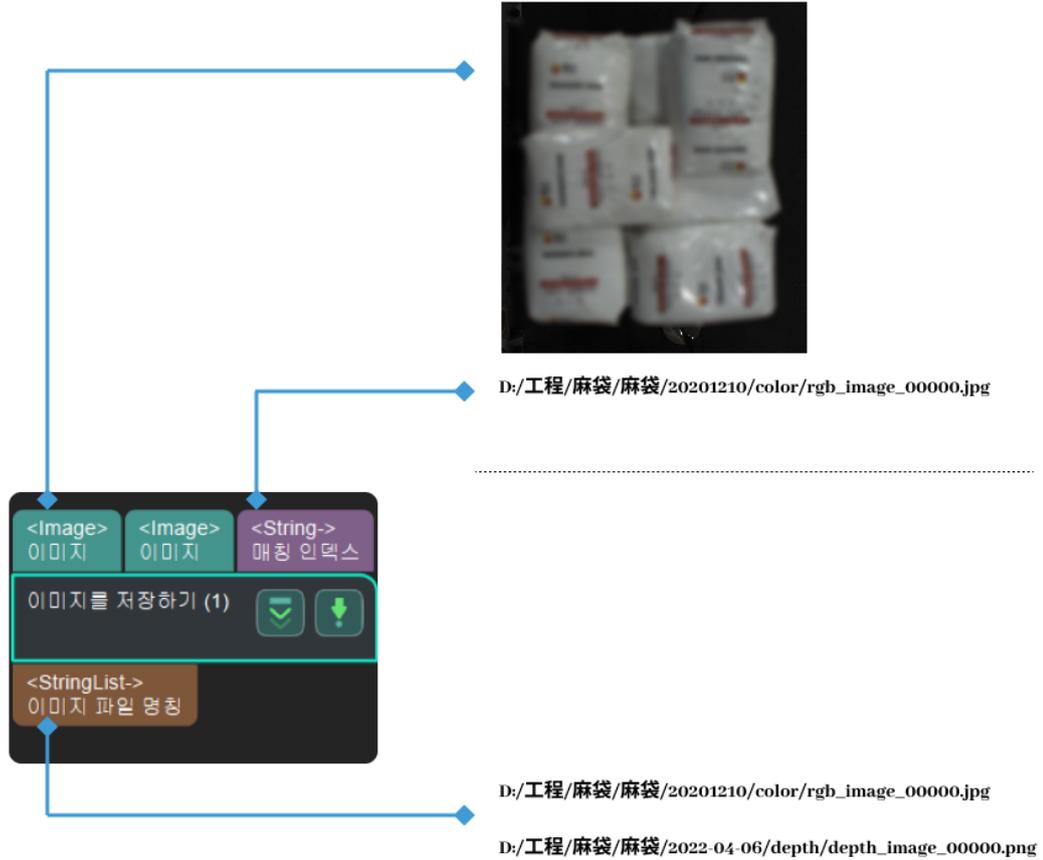
#### 기능 설명

이 스텝은 컬러 맵과 덤스 맵을 지정된 경로로 저장합니다.

#### 응용 시나리오

이 스텝은 카메라의 원시 컬러 맵과 덤스 맵을 저장하는 데 자주 사용되며 인스턴스 세그먼테이션 그래픽 등을 저장하는 데에도 사용할 수 있습니다. 사용자는 필요에 따라 **저장 경로** 및 **하위 폴더의 명칭**과 같은 파라미터를 변경할 수 있습니다.

#### 입력 및 출력



## 파라미터 설명

### 디렉터리 설정

### 저장 경로

파라미터 해석: 이 파라미터는 이미지의 저장 경로를 선택하는 데 사용됩니다.

조정 설명: 실제 필요에 따라 이미지를 저장하는 경로를 선택해야 합니다.

### 목표 명칭

파라미터 설명: 이 파라미터는 대상 물체의 이름을 딴 하위 폴더를 만드는 데 사용됩니다.

기본값: 비어 있음

### 하위 폴더의 명칭

파라미터 설명: 이 파라미터는 사진 폴더를 다른 형식으로 설정하는 데 사용되며 폴더는 ";"로 구분됩니다.

형식: XXXX;XXXX;XXXX;XXXX

기본값: depth, color

### 날짜 관련 설정

#### 날짜별 하위 폴더 생성

파라미터 설명: 이 파라미터는 현재 날짜로 이름 지어진 하위 폴더를 생성할지 여부를 선택하는 데 사용됩니다.

기본값: 선택함



**날짜별 하위 폴더 생성** 체크를 해제하면 **하위 폴더의 최대 수** 및 **만료된 하위 폴더를 삭제하기**를 설정할 필요가 없습니다.

### 하위 폴더의 최대 수

파라미터 설명: 이 파라미터는 하위 폴더를 저장할 최대 일수를 설정하는 데 사용됩니다.

기본값: 7 day

### 만료된 하위 폴더를 삭제하기

파라미터 설명: 이 파라미터는 **하위 폴더의 최대 수**(가장 긴 저장 일수)를 초과하는 폴더를 삭제할지 여부를 설정하는 데 사용됩니다.

기본값: 선택함

### 저장 내용 설정

#### 카메라 파라미터를 저장하기

파라미터 설명 :이 파라미터는 카메라 파라미터를 폴더에 저장할지 여부를 선택하는 데 사용됩니다. 선택하면 이미지를 저장할 때 카메라 파라미터를 저장합니다.

기본값: 선택함

조정 설명: 기본값을 선택하는 것이 좋습니다.

#### 플랜지 포즈를 저장하기

파라미터 설명 :이 파라미터는 플랜지 포즈를 폴더에 저장할지 여부를 선택하는 데 사용됩니다. 선택하면 이미지를 저장할 때 플랜지 포즈를 저장합니다.

기본값: 선택함

조정 설명: 기본값을 선택하는 것이 좋습니다.

#### 가상 카메라를 사용하는 경우 저장하지 않기

파라미터 설명 : 이 파라미터는 가상 카메라의 데이터를 폴더에 저장할지 여부를 선택하는 데 사용됩니다. 선택한 후 가상 카메라가 프로젝트에 사용되면 입력 된 이미지가 저장되지 않습니다.

기본값: 선택함

조정 설명: 기본값을 선택하는 것이 좋습니다.

### 이미지 인덱스 설정

#### 명칭이 같은 이미지를 덮어쓰기

파라미터 설명 : 이 파라미터는 명칭이 같은 이미지를 덮어쓸지를 설정하는 데 사용됩니다. 선택하면 명칭이 같은 이미지를 자동으로 덮어씁니다. 선택하지 않으면 이미지를 덮어쓸지 여부를 나타내는 팝업 창이 표시됩니다.

기본값: 선택함

#### 최대 이미지 수

파라미터 설명: 이 파라미터는 저장된 최대 이미지 수를 설정하는 데 사용됩니다. 이미지 수가 최대 값을 초과하면 다음 사진 일련 번호가 0으로 재설정됩니다.

기본값: 1000

값 범위: 1과 같거나 1보다 큼니다.

#### 사용자 자체 정의한 시작 인덱스

파라미터 설명: 이 파라미터는 이미지를 저장할 때의 시작 인덱스를 설정하는 데 사용됩니다.

기본값: -1

#### 파일 명칭 설정

##### PNG 포맷으로 컬러 맵을 저장하기

파라미터 설명: 이 파라미터는 컬러 맵이 .png 포맷으로 저장되는지 여부를 설정하는 데 사용됩니다.

기본값: 선택하지 않음

##### 이미지 파일의 접두사(자체 정의)

파라미터 설명: 이 파라미터는 이미지 파일의 접두사를 설정하는 데 사용됩니다.

기본값: 비어 있음

### 4.3.75. 결과를 파일로 저장하기



이 스텝의 내용은 유지 보수 중입니다. 이 스텝에 대한 자세한 정보가 필요한 경우 ([docs@mech-mind.net](mailto:docs@mech-mind.net))로 문의하십시오.

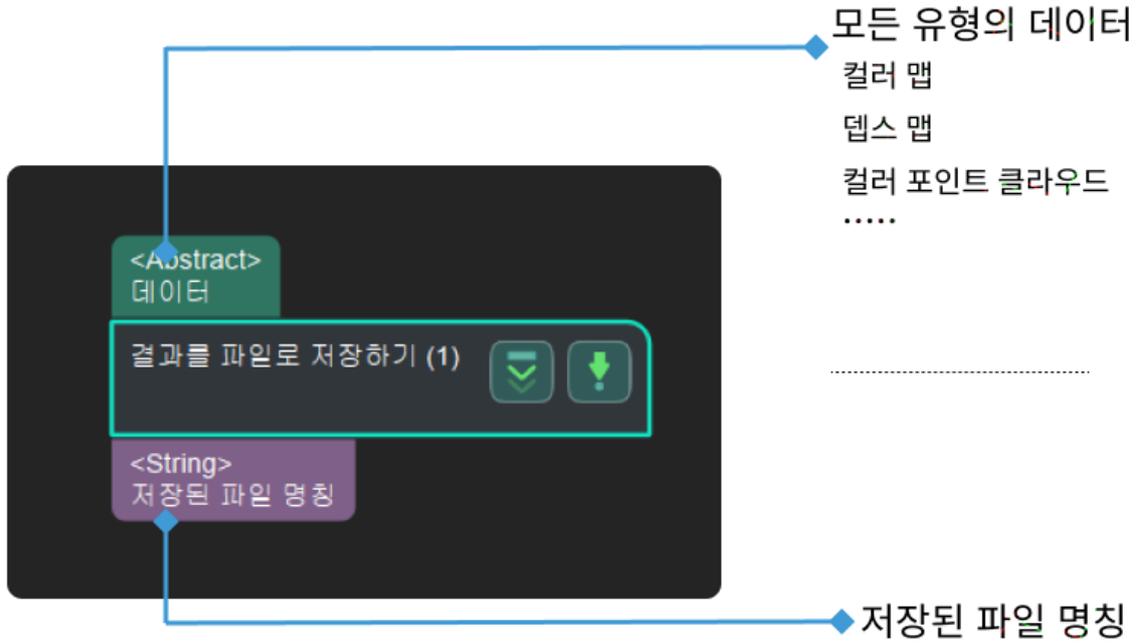
#### 기능 설명

프로젝트 데이터를 로컬 파일로 저장합니다.

#### 응용 시나리오

이 스텝에서는 모든 유형의 데이터를 저장할 수 있지만, 2D 맵을 저장해야 하는 경우 [이미지를 저장하기](#) 스텝을 사용하여 저장하는 것이 좋습니다.

#### 입력 및 출력



### 4.3.76. 스텝 파라미터를 파일로 저장하기

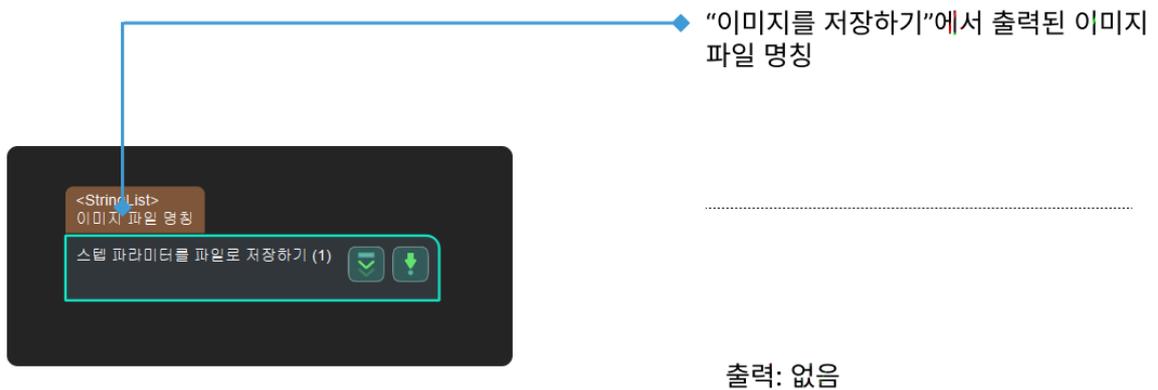
#### 기능 설명

지정된 스텝의 파라미터를 로컬 파일로 저장합니다.

#### 응용 시나리오

이 스텝은 다른 스텝의 파라미터를 도출하는 데 사용되며 앞에 [이미지를 저장하기](#) 스텝을 연결해야 합니다.

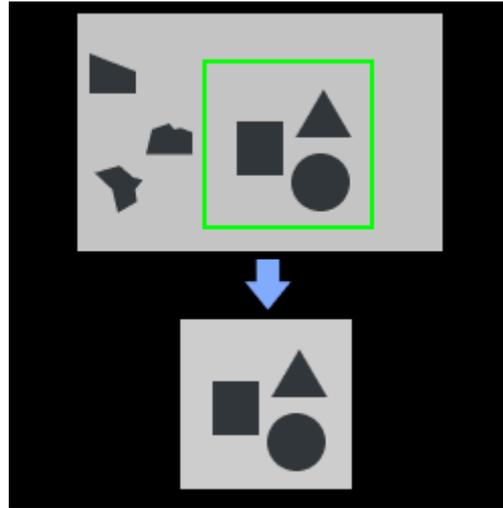
#### 입력 및 출력



### 4.3.77. 2D ROI 내 이미지의 크기를 조정하기

#### 기능 설명

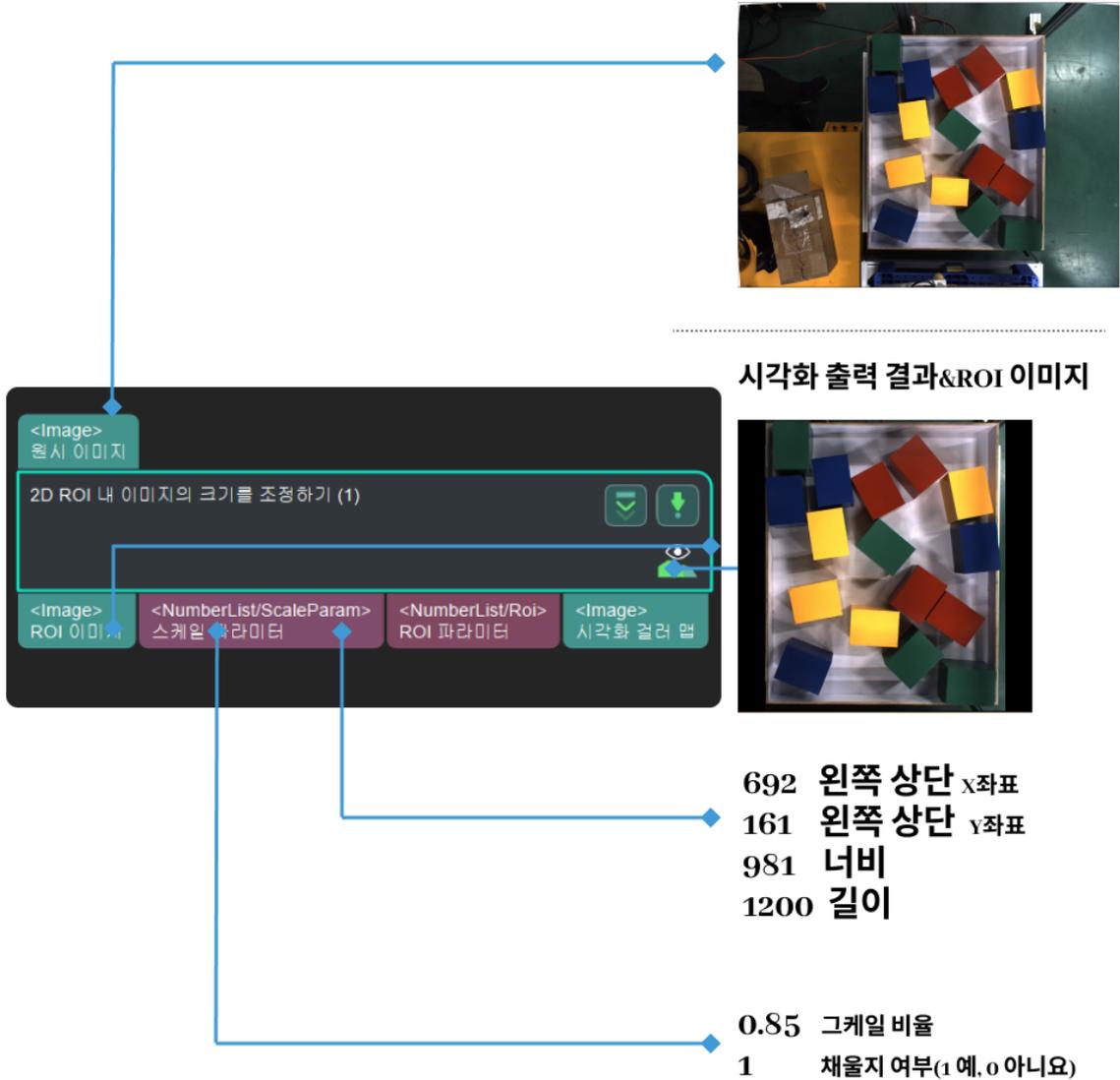
이 스텝은 이미지의 ROI를 지정된 크기로 조정할 수 있습니다.



### 응용 시나리오

이 스텝은 딥 러닝과 관련된 스텝의 이미지 사전 처리에 사용되고 일반적으로 딥 러닝 관련 스텝 혹은 [2D ROI](#) 내의 조정된 이미지를 복원하기와 함께 사용됩니다.

### 입력 및 출력



**시각화 출력 결과&ROI 이미지**

692 왼쪽 상단 X좌표  
161 왼쪽 상단 Y좌표  
981 너비  
1200 길이

0.85 스케일 비율  
1 채울지 여부(1 예, 0 아니요)

## 사용 방법

이 스텝은 ROI 내 이미지를 지정된 크기로 조정하는 데 사용됩니다. 컬러 ROI 스케일링 및 이상적인 대상 해상도 조절을 통해 원하는 크기를 출력합니다.

### ROI 이미지를 자동으로 스케일링하기

ROI 이미지를 자동으로 스케일링하는 방법은 다음과 같습니다.

1. 컬러 ROI 스케일링에 있는 **자동으로 스케일링하기**를 선택합니다.
2. 그 다음에 **이상적인 대상 해상도** 중의 **입력된 컬러 ROI 및 스킵 스케일링과 일치하게 만들기**를 선택합니다.

### 원시 ROI 이미지를 출력하기

원시 ROI 이미지를 출력하는 방법은 다음과 같습니다.

1. 컬러 ROI 스케일링에 있는 **\*\*자동으로 스케일링하기\***를 선택 취소합니다.
2. 사용자 자체 정의 스케일을 1로 설정합니다.

3. **이상적인 대상 해상도** 아래의 **\*\*입력된 컬러 ROI 및 스킵 스케일링과 일치하게 만들기\***를 선택합니다.

#### 지정된 배수로 ROI 이미지를 출력하기

지정된 배수로 ROI 이미지를 출력하는 방법은 다음과 같습니다.

1. 컬러 ROI 스케일링에 있는 **자동으로 스케일링하기**를 선택 취소합니다.
2. 수요에 따라 **사용자 자체 정의 스케일**을 설정합니다. 이 파라미터가 1보다 큰 경우 이미지가 확대되고 파라미터가 1보다 작은 경우 이미지가 축소됩니다.
3. **이상적인 대상 해상도** 중의 **입력된 컬러 ROI 및 스킵 스케일링과 일치하게 만들기**를 선택합니다.

#### 지정된 길이 또는 너비로 ROI 이미지를 출력하기

지정된 길이 또는 너비로 ROI 이미지를 출력하는 방법은 다음과 같습니다.

1. 그 다음에 **이상적인 대상 해상도** 중의 선택된 **입력된 컬러 ROI 및 스킵 스케일링과 일치하게 만들기**를 취소합니다.
2. 수요에 따라 **너비 및 높이**를 설정합니다.
3. 그 다음에 수요에 따라 **패딩**을 선택하는지 결정합니다.



각 파라미터의 구체적인 설명은 파라미터 설명을 참조하십시오.

### 파라미터 설명

#### 컬러 ROI 설정

파라미터 설정: 이 파라미터는 컬러 ROI를 설정하는 방법을 지정하는 데 사용됩니다.

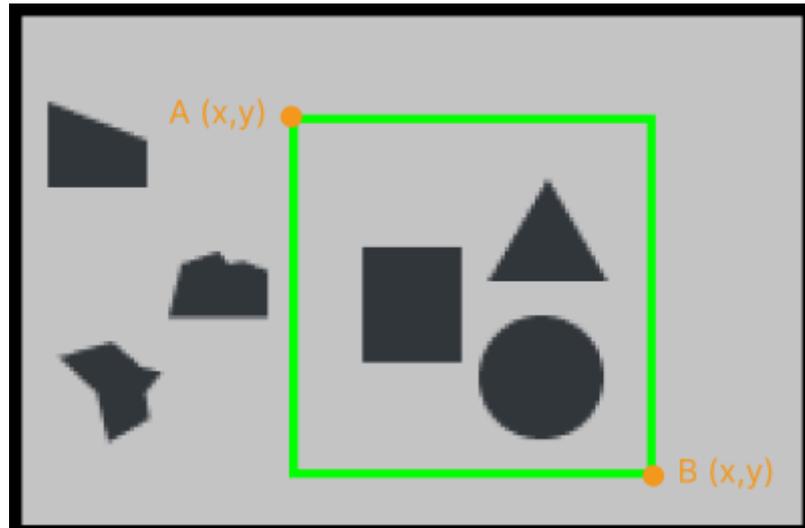
값 리스트:

- ROIByFile: 기본값. ROI를 시각화 설정하고 왼쪽 상단의 꼭짓점과 오른쪽 하단의 꼭짓점의 좌표를 통해 ROI를 확인합니다.
- ROIByParam: 적절적으로 파라미터를 입력하여 ROI를 확인합니다. 파라미터는 왼쪽 상단의 꼭짓점과 ROI의 너비 및 높이입니다.

#### ROIByFile

##### 컬러 ROI 파일

파라미터 설명: 이 파라미터는 컬러 맵의 ROI를 설정하는 데 사용됩니다. [2D ROI 설정](#)을 참조하십시오. 다음의 그림에서 A는 왼쪽 상단의 꼭짓점이고 B는 오른쪽 하단의 꼭짓점입니다.



 원시 이미지 왼쪽 상단의 좌표는 (0, 0)입니다.

## ROIByParam

### Start X

파라미터 설명: 이 파라미터는 ROI 왼쪽 상단의 꼭짓점 X축의 좌표를 표시합니다.

기본값: 0

조절 설명:  $\text{Start X} + \text{Width} < \text{원시 이미지의 너비}$

### Start Y

파라미터 설명: 이 파라미터는 ROI 왼쪽 상단의 꼭짓점 Y축의 좌표를 표시합니다.

기본값: 0

조절 설명:  $\text{Start y} + \text{Height} < \text{원시 이미지의 높이}$

### Width

파라미터 설명: 이 파라미터는 ROI의 너비를 표시합니다.

기본값: 0

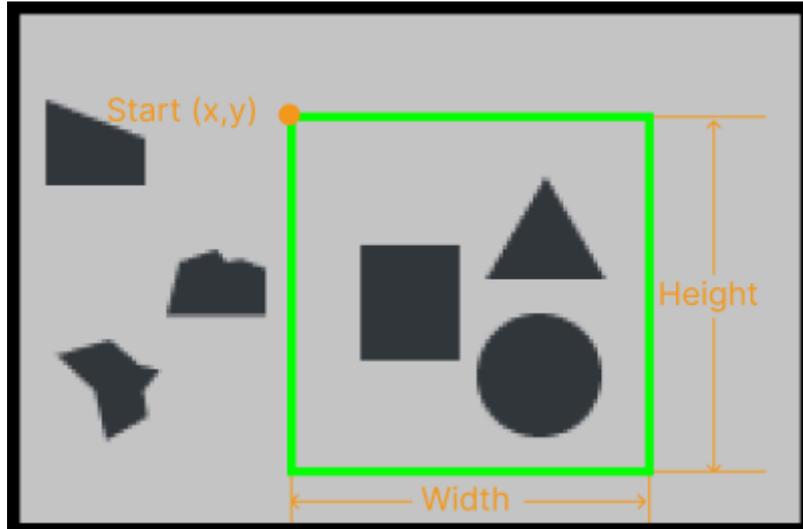
조절 설명:  $\text{Start X} + \text{Width} < \text{원시 이미지의 너비}$

### Height

파라미터 설명: 이 파라미터는 ROI의 높이를 표시합니다.

기본값: 0

조절 설명:  $\text{Start y} + \text{Height} < \text{원시 이미지의 높이}$



 원시 이미지 왼쪽 상단의 좌표는 (0, 0)입니다.

### 컬러 맵 ROI 영역 업데이트 방법

파라미터 설명: 이 파라미터는 ROI 이미지의 형식을 지정하는 데 사용됩니다.

값 리스트:

**\*Origin:** ROI의 원시 컬러 맵.

**\*AdaptiveModifiedRoiOriginP:** 이미지 틀을 사용하여 컬러 맵 ROI을 조정하고 이미지 비0 구역의 중심을 ROI 중심으로 자동으로 조정합니다.

**\*BoundRect:** 입력된 이미지의 경계 사각형 영역을 추출하고 이미지 비0의 직사각형을 사용하여 오리지널 이미지를 교체하며 스케일 비율을 계산합니다.

### 컬러 ROI 스케일링

#### 자동으로 스케일링하기

파라미터 설명: 이 파라미터는 ROI 이미지의 크기를 자동으로 스케일링하는 데 사용됩니다.

기본값: 선택함

#### 사용자 자체 정의 스케일

파라미터 설명: 이 파라미터는 ROI의 스케일을 지정하는 데 사용됩니다.

스케일링한 후 이미지 크기 = ROI 원시 크기 × 사용자 자체 정의 스케일. 다음의 그림과 같이 왼쪽의 그림은 원시 크기이며 사용자 자체 정의 스케일의 계수는 0.5입니다.

 자동으로 스케일링하기를 선택하지 않는 경우 이 파라미터를 설정할 수 있습니다.

### 이상적인 대상 해상도

#### 입력된 컬러 ROI 및 스킵 스케일링과 일치하게 만들기

파라미터 설명: 이 파라미터는 출력된 이미지와 컬러 ROI 스케일링한 후의 이미지의 크기를 일치하는지 설정하는 데 사용됩니다.

기본값: 선택하지 않음

 이 파라미터를 선택하지 않으면 출력된 이미지의 크기는 스케일링한 후의 크기입니다.

## 너비/높이

파라미터 설명: 이 파라미터는 ROI 이미지의 너비/높이를 지정하는 데 사용되며 단위는 픽셀(px)입니다.

기본 값: 1024px

조절 설명: 실제 상황에 따라 설정하십시오.



이상적인 대상 해상도를 선택하지 않는 경우 이 파라미터를 설정할 수 있습니다.

## 패딩 컬러



- ROI 이미지의 너비/높이를 지정하는 경우에만 패딩 기능을 사용합니다.
- 패딩을 선택해야 R값, G값 및 B값을 설정할 수 있습니다.

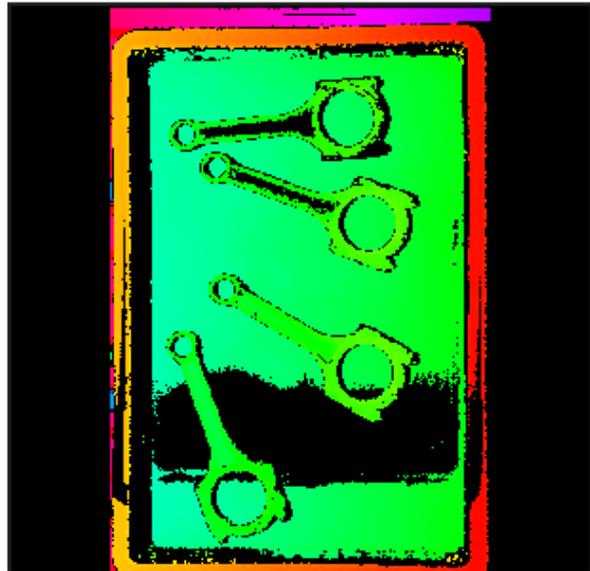
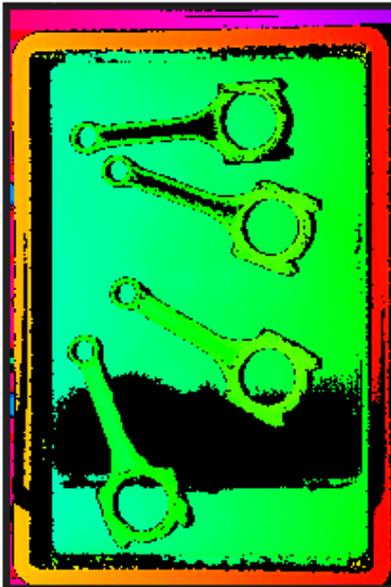
## 패딩

파라미터 설명: ROI 이미지가 지정된 길이/너비로 확대된 후 다른 구역이 패딩 기능으로 채워집니다.

기본 값: 선택함

조절 설명: 검정색(0,0,0) 또는 회색(128,128,128)을 권장합니다.

조절 예시: 패딩 기능을 사용하지 않은 경우 ROI 이미지를 다음 그림의 왼쪽과 같이 지정된 길이/너비로 확대합니다. 패딩 기능을 사용하는 경우 ROI 이미지는 지정된 크기여야 하며, 아래 그림의 오른쪽과 같이 다른 구역은 패딩 색상으로 채워집니다.



## R / G / B 값

파라미터 설명: 컬러 맵은 R, G, B 셋 가지 색상으로 구성됩니다. R/G/B 값을 다르게 설정하여 다양한 색상이 채워질 수 있습니다.

기본 값: 0

값 범위: 0~255.

### 4.3.78. 포인트 클라우드를 외부 서비스로 보내기

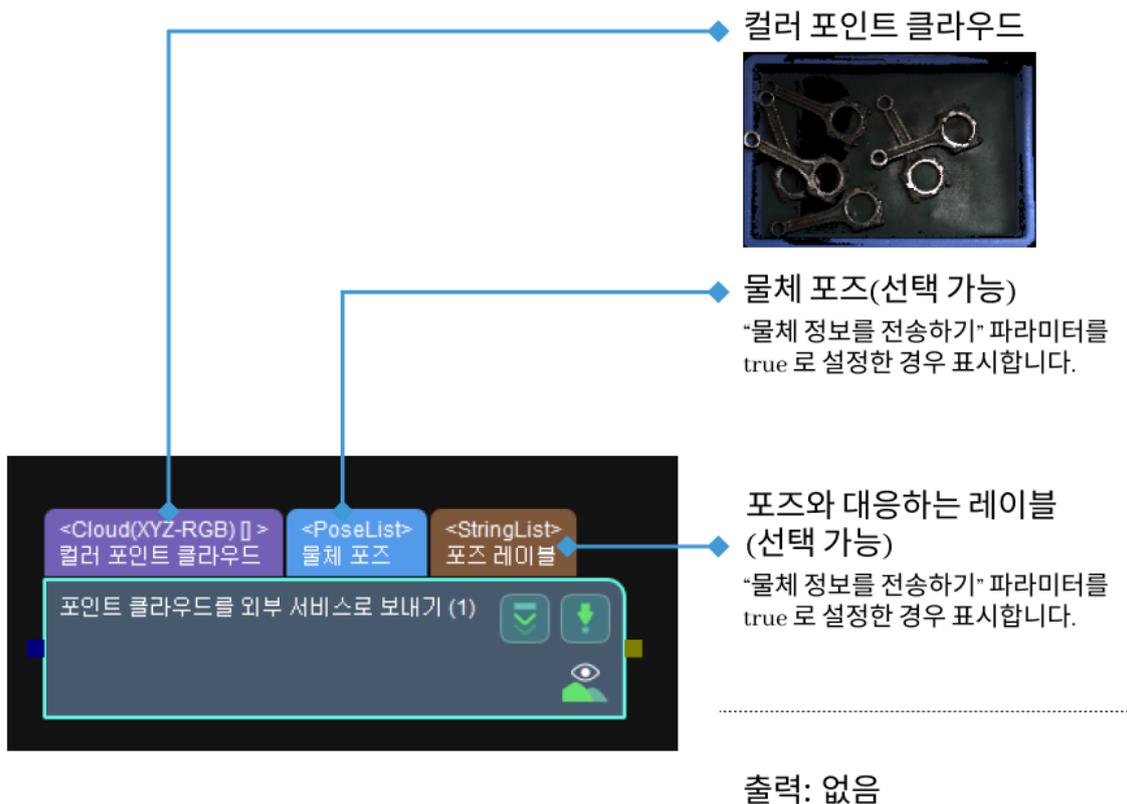
#### 기능 설명

이 스텝은 포인트 클라우드를 Mech-Viz로 보낼 수 있습니다.

#### 응용 시나리오

이 스텝은 보통 프로젝트 디버그 또는 프로젝트 실제 실행 효과를 확인하는 데 사용됩니다.

#### 입력 및 출력



#### 파라미터 설명

##### 포인트 클라우드 설정

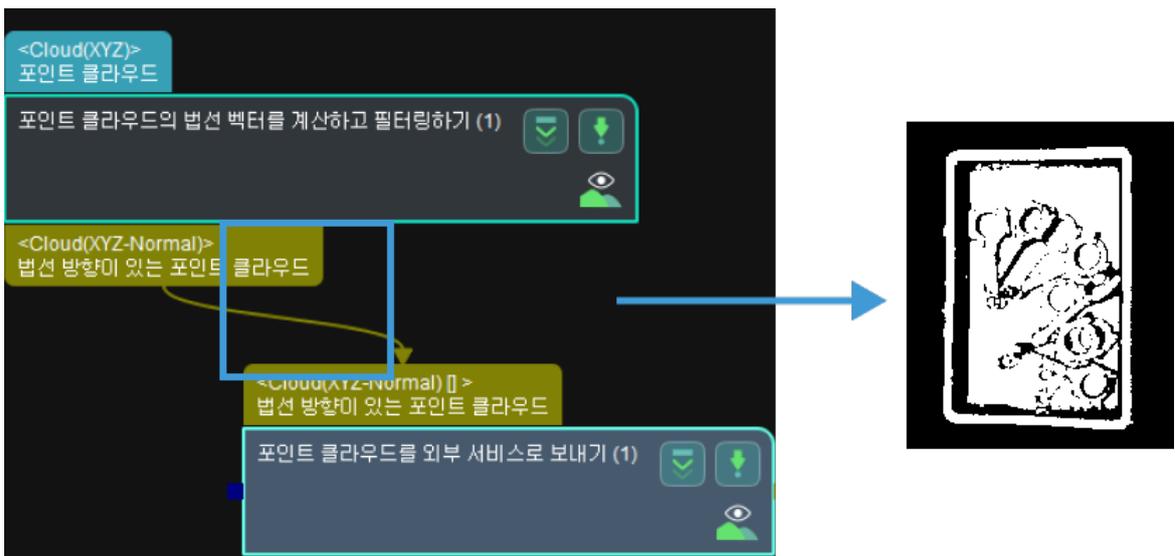
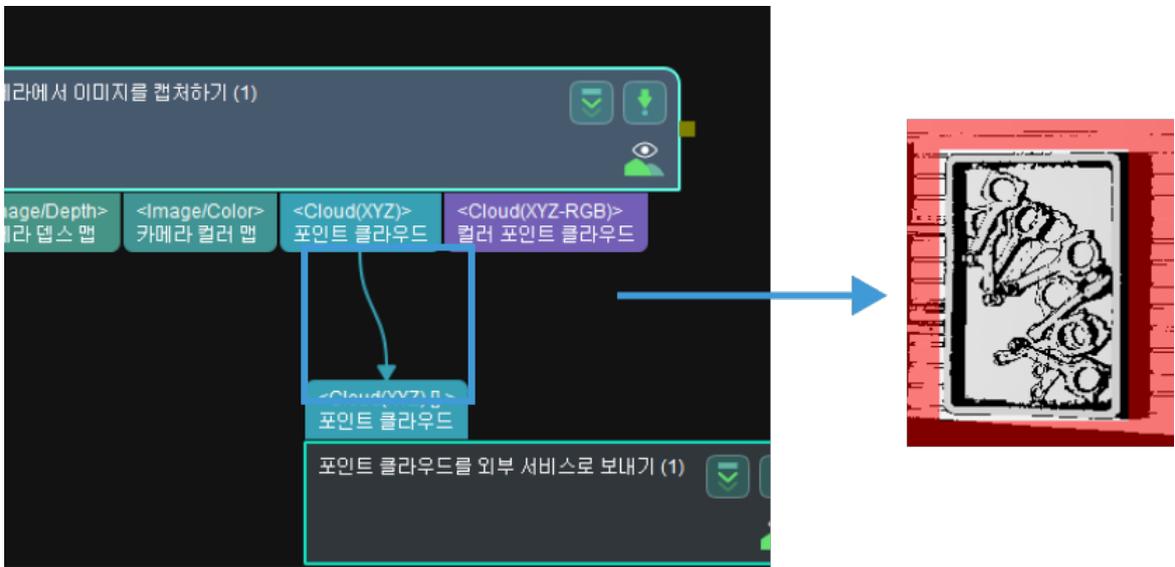
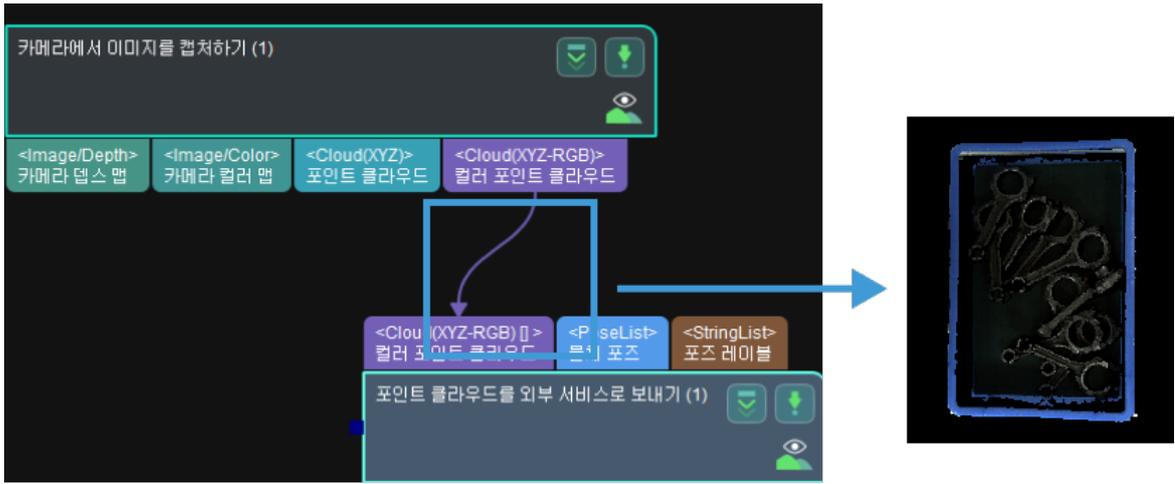
##### 입력된 포인트 클라우드의 유형

기본값: CloudXYZRGB(컬러 포인트 클라우드)

값 리스트: CloudXYZRGB(컬러 포인트 클라우드), CloudXYZ(포인트 클라우드), CloudNORMAL(법선 방향을 갖춘 포인트 클라우드).

설명: 실제 수요에 따라 보낼 포인트 클라우드 유형을 선택하십시오.

예시: CloudXYZRGB, CloudXYZ, CloudNORMAL를 선택했을 때 입력한 포인트 클라우드의 유형은 각각 아래 그림과 같습니다.



### 물체 정보 설정

### 물체 정보를 전송하기

기본값: 선택하지 않음

설명: 선택하지 않으면 **물체 포즈** 및 **포즈 레이블** 포트가 닫히고 선택하면 물체 포즈 및 포즈 분류 레이블 포트가 열립니다. 실제 수요에 따라 **물체 포즈** 또는 **포즈 레이블**을 보낼지 선택하십시오. 선택 전과 후의 효과는 아래 그림과 같으며 그림 1은 선택 전, 그림 2는 선택 후입니다.

[send point cloud to external service send object information] | *send-point-cloud-to-external-service-send\_object\_information.png*

### 입력된 포인트 클라우드가 카메라 좌표계에 있는지

기본값: 선택함.

설명: 포인트 클라우드와 포즈가 카메라 좌표계에 있을 때 이 옵션을 선택하면 입력한 포인트 클라우드 및 포즈 정보는 로봇 좌표계로 전환되어 Mech-Viz로 보내게 됩니다. 선택하지 않으면 입력한 포인트 클라우드는 직접 Mech-Viz로 보내게 됩니다.

### 4.3.79. 포즈의 사원수를 설정하기



이 스텝의 내용은 유지 보수 중입니다. 이 스텝에 대한 자세한 정보가 필요한 경우 ([docs@mech-mind.net](mailto:docs@mech-mind.net))로 문의하십시오.

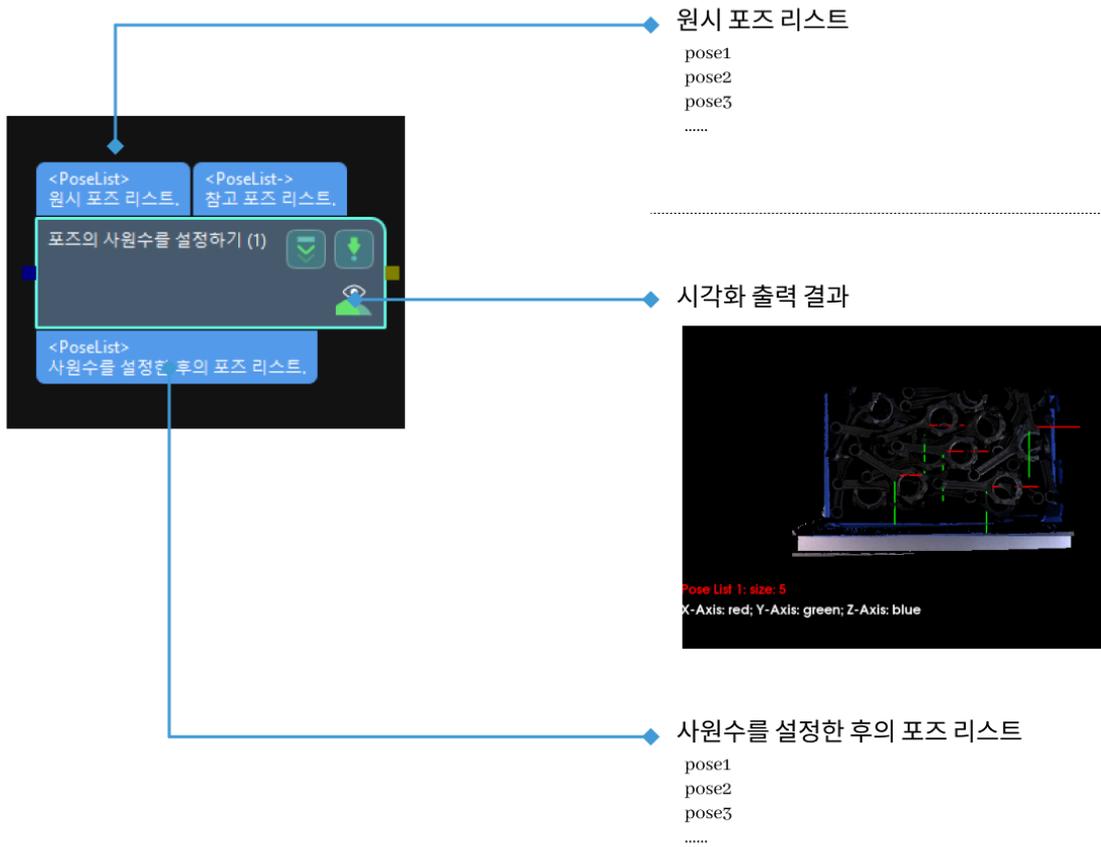
#### 기능 설명

두 번째 포트로 입력된 참고 포즈 혹은 파라미터에서 설정된 사원수를 사용하여 포즈의 방향(사원수로 표시됨)을 설정합니다.

#### 응용 시나리오

일반적인 포즈 사원수 설정 스텝으로 참고 포즈 리스트에 여러 개 포즈가 존재하는 경우에 첫 번째 포즈의 사원수만 사용하여 설정합니다.

#### 입력 및 출력



### 4.3.80. 포즈의 평행 이동의 벡터를 설정하기



이 스텝의 내용은 유지 보수 중입니다. 이 스텝에 대한 자세한 정보가 필요한 경우 ([docs@mech-mind.net](mailto:docs@mech-mind.net))로 문의하십시오.

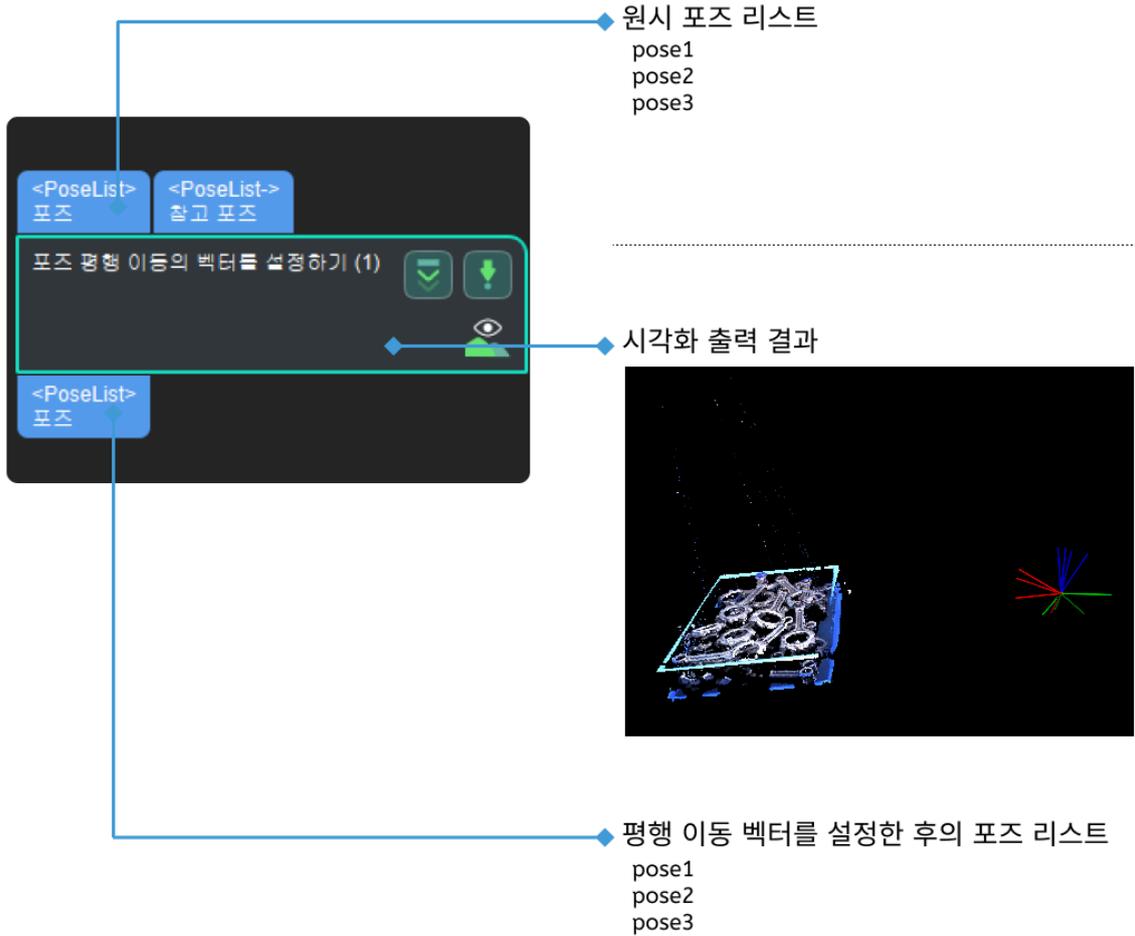
#### 기능 설명

원시 포즈의 평행 이동 벡터를 설정합니다. 참고 포즈나 파라미터에서 설정된 기준 방향을 사용할 수 있습니다.

#### 응용 시나리오

일반적인 포즈 평행 이동 벡터 설정 스텝입니다. 참고 포즈 리스트에 여러 개의 포즈가 존재하는 경우 첫 번째 포즈의 평행 이동 벡터만 사용될 것입니다.

#### 입력 및 출력



### 4.3.81. 이미지를 표시하기

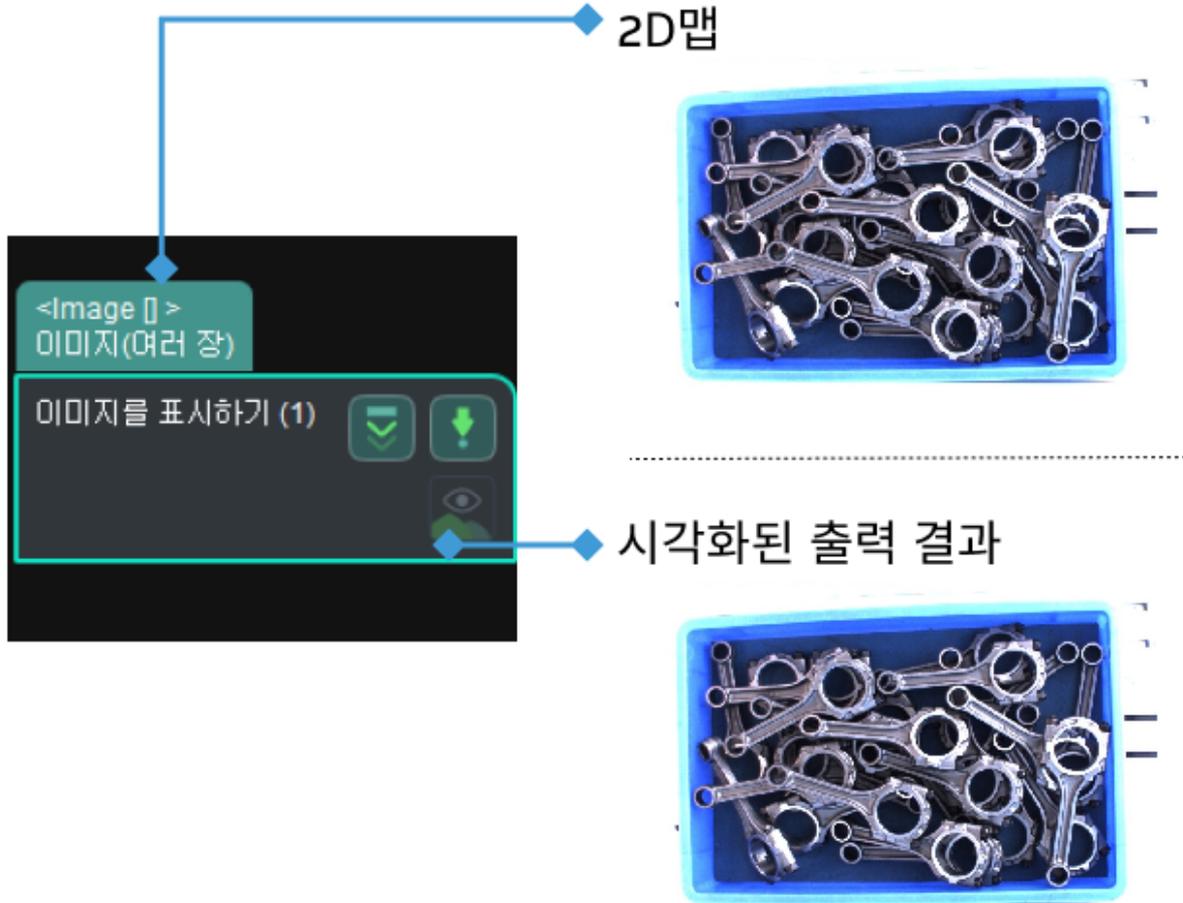
#### 기능 설명

다른 스텝에서 출력된 2D 맵을 시각화합니다. 이 스텝은 2D 매칭, 2D 특징 추출, 2D 일반 처리와 같은 다른 스텝에서 출력된 2D 맵을 표시하는 데 사용됩니다.

#### 응용 시나리오

이 스텝은 2D 맵의 표시 효과를 통해 이미지가 요구에 부합할 때까지 대응하는 입력한 스텝의 파라미터를 조절합니다.

#### 입력 및 출력



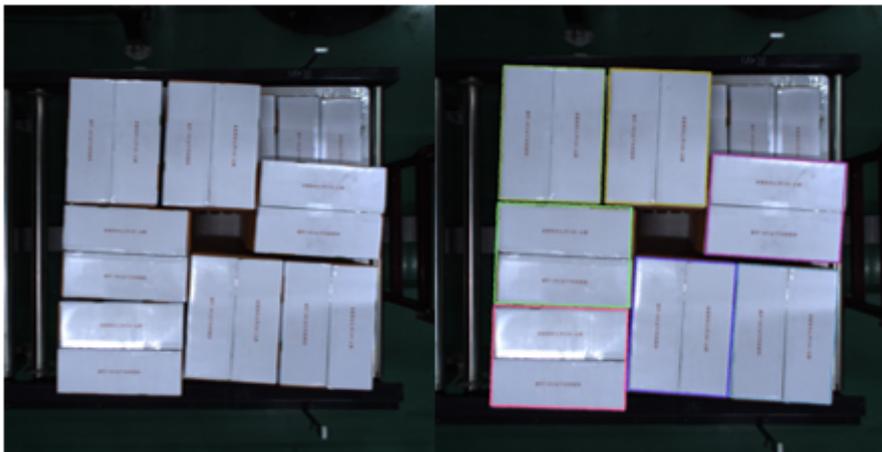
파라미터 설명

표시 유형

기본값: Multimages

값 리스트: Multimages, ImgAndPolyVerts

설명: **Multimages**를 선택하면 아래 그림의 왼쪽과 같이 이미지만 표시됩니다.  
**ImgAndPolyVerts**를 선택하면 아래 그림의 오른쪽과 같이 이미지와 후보 다각형이 표시됩니다.



### 4.3.82. 포인트 클라우드 및 포즈를 표시하기

#### 기능 설명

이 스텝에서는 포인트 클라우드와 포즈를 시각화할 수 있습니다.



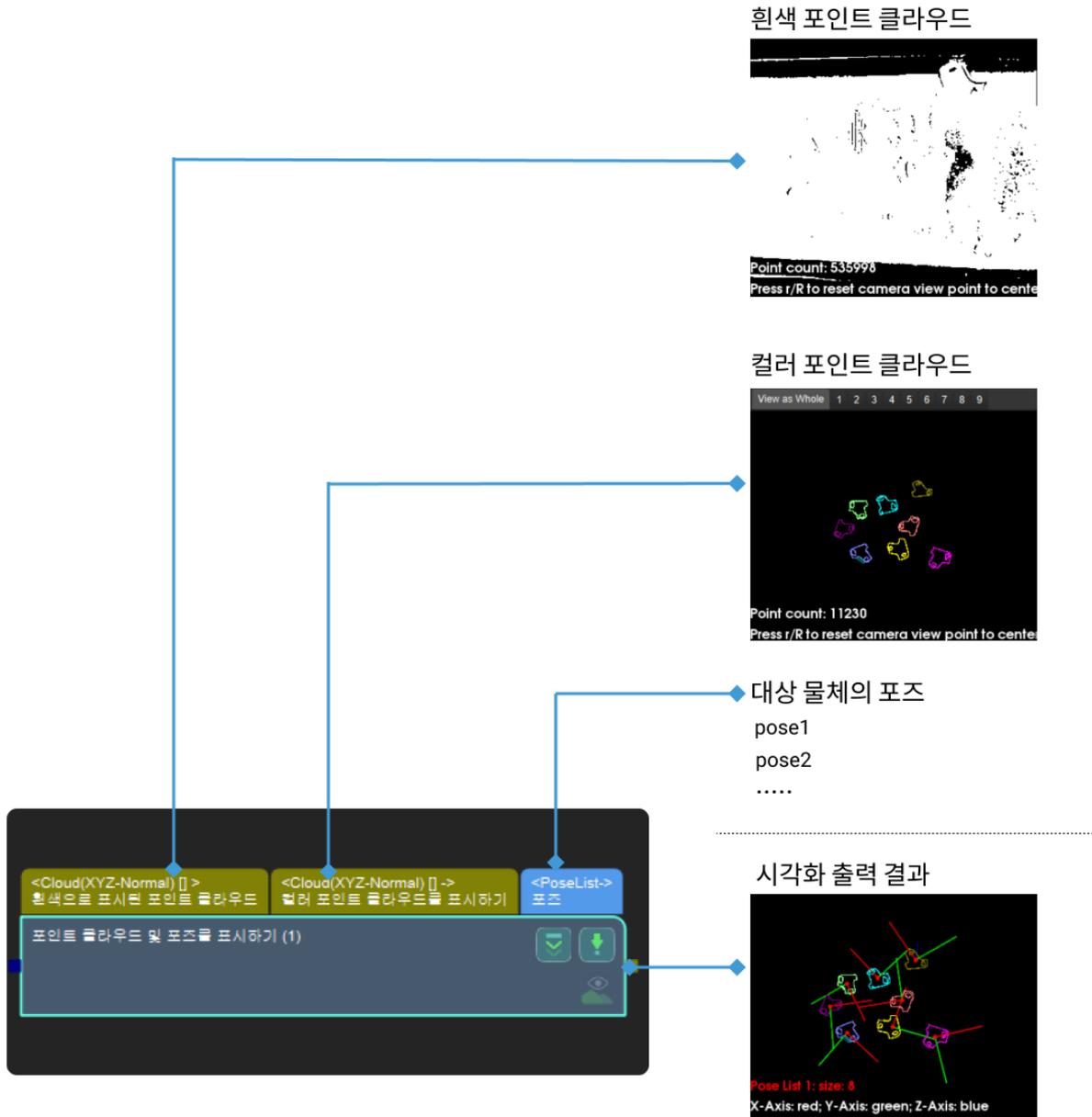
비슷한 스텝 비교:

**포인트 클라우드 및 포즈를 표시하기** 스텝은 포인트 클라우드 및 포즈를 표시하는 데 사용됩니다.  
**이미지를 표시하기** 스텝은 컬러 맵 혹은 마스크 이미지를 표시하는 데 사용됩니다.

#### 응용 시나리오

이 스텝은 포즈와 포인트 클라우드의 상대적 위치를 표시하는 데 사용됩니다. 예를 들어 물체 피킹 시나리오에서 포즈가 맞는지 체크하고 보통 디버깅 단계에 사용됩니다.

#### 입력 및 출력



## 파라미터 설명

### 법선 벡터 시각화 설정

#### 법선 벡터 표시하기

파라미터 설명 :이 파라미터는 포인트 클라우드의 법선 벡터가 시각화 출력 결과에 표시되는지 여부를 결정합니다.

기본 값: 선택하지 않음.

조정 설명: 이 항목을 선택한 후 포인트 클라우드 법선 벡터가 시각화 출력 결과에 표시됩니다. 특정 효과에 대한 내용은 [조정 예시](#)를 참조하십시오.

#### 법선 벡터 표시 간격

파라미터 설명: 이 파라미터의 값은 법선 벡터의 밀도를 결정합니다(이 항목은 **법선 벡터 표시하기**를 선택한 후 표시됩니다).

기본값: 20

조정 설명: 이 파라미터는 법선 벡터 사이의 거리를 설정하는 데 사용되며 단위는 mm입니다. 이 값이 클수록 법선 벡터를 드물게 표시합니다. 특정 효과에 대한 내용은 [조정 예시](#)를 참조하십시오.

## Z값 시각화 설정

### 시각화 포인트 클라우드 Z값

파라미터 설명 :이 파라미터는 그레이스케일이 다른 포인트 클라우드를 표시하는 데 사용됩니다.

기본 값: 선택하지 않음.

조정 설명: 이 옵션을 선택하면 Z값에 따라 다양한 그레이스케일로 포인트 클라우드를 표시할 것입니다.

#### 상한

파라미터 설명 : 색칠이 필요한 Z 방향 최대값 (**Z 값 시각화 설정**이 선택된 경우에만 유효합니다).

기본값: 0

조정 설명: 이 값은 색칠이 필요한 Z 방향 최대 값을 나타냅니다. 이 값이 클수록 이미지가 점점 어두운 회색으로 변하며 덜 뚜렷해집니다.

#### 하한

파라미터 설명 : 색칠이 필요한 Z 방향 최소값 (**Z 값 시각화 설정**이 선택된 경우에만 유효합니다).

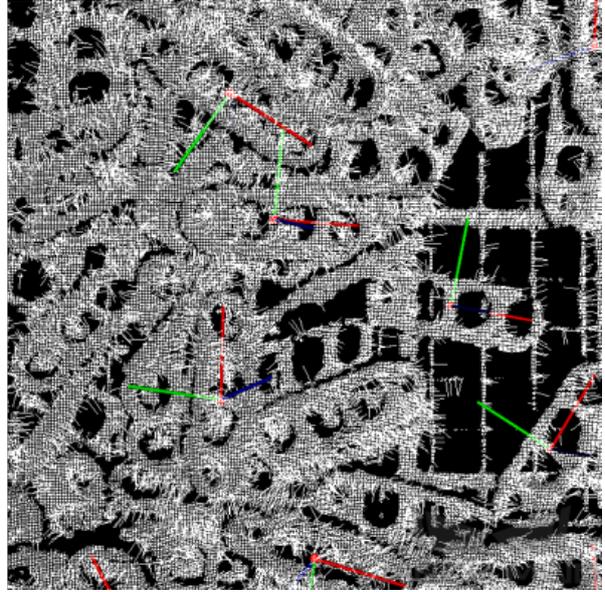
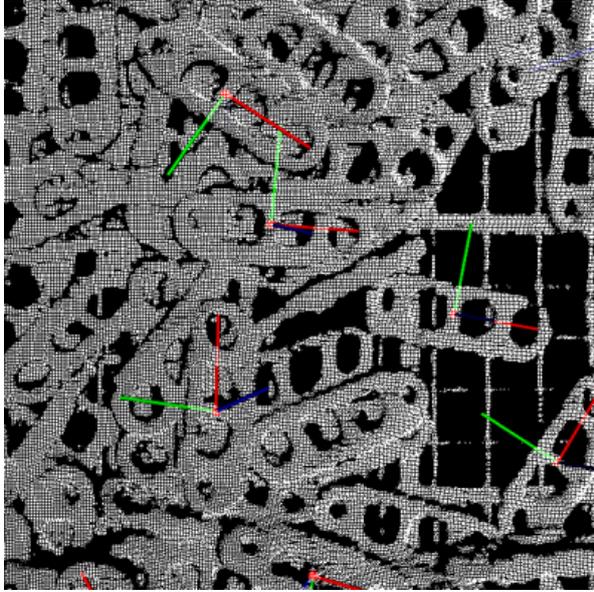
기본값: 0

조정 설명: 이 값은 색칠이 필요한 Z 방향 최소 값을 나타냅니다. 이 값이 작을수록 이미지가 더 뚜렷해집니다.

## 조정 예시

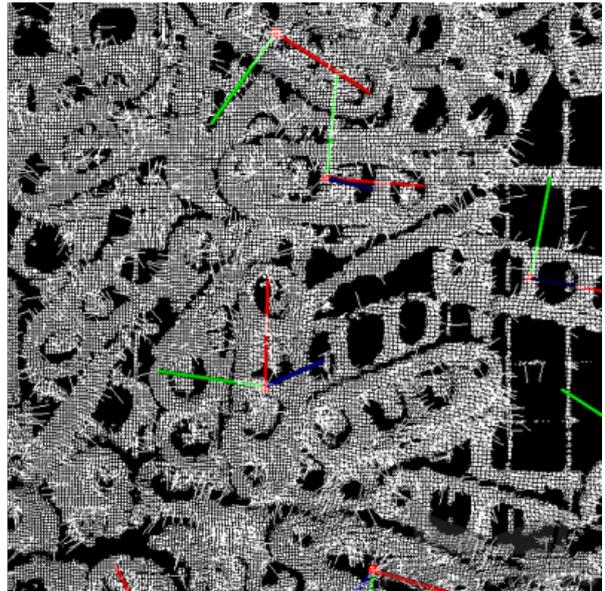
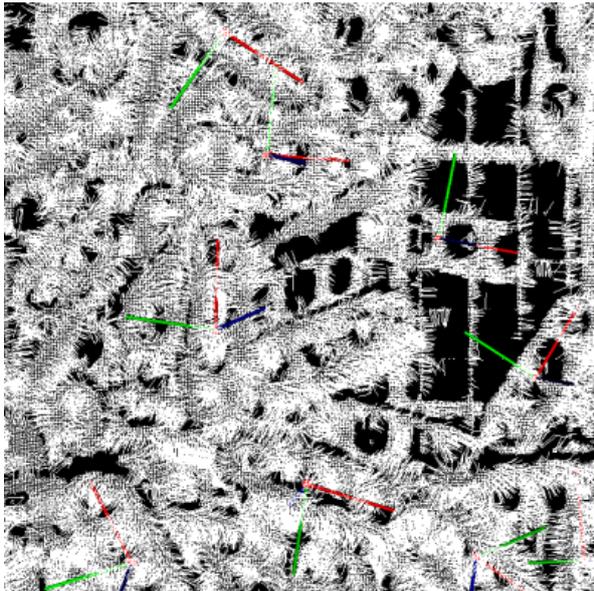
### 포인트 클라우드의 법선 벡터 표시

아래 그림과 같이 왼쪽은 법선 벡터를 표시하지 않고 오른쪽은 20mm 간격으로 법선 벡터를 표시한 것입니다.



#### 법선 벡터 표시 간격 변경

아래 그림과 같이, 왼쪽 그림의 법선 벡터 표시 간격은 5mm이며, 법선 벡터가 밀집되어 있습니다. 오른쪽 그림의 법선 벡터 표시 간격은 40mm이며, 법선 벡터 표시 간격이 더 희박합니다.



#### 4.3.83. 포인트 클라우드의 순서를 배열하기



이 스텝의 내용은 유지 보수가 중단된 상태입니다. 새 버전 스텝 [지정한 포인트 클라우드의 파라미터를 계산하기](#)를 사용하십시오. 이 스텝에 대한 자세한 정보가 필요한 경우 ([docs@mech-mind.net](mailto:docs@mech-mind.net))로 문의하십시오.

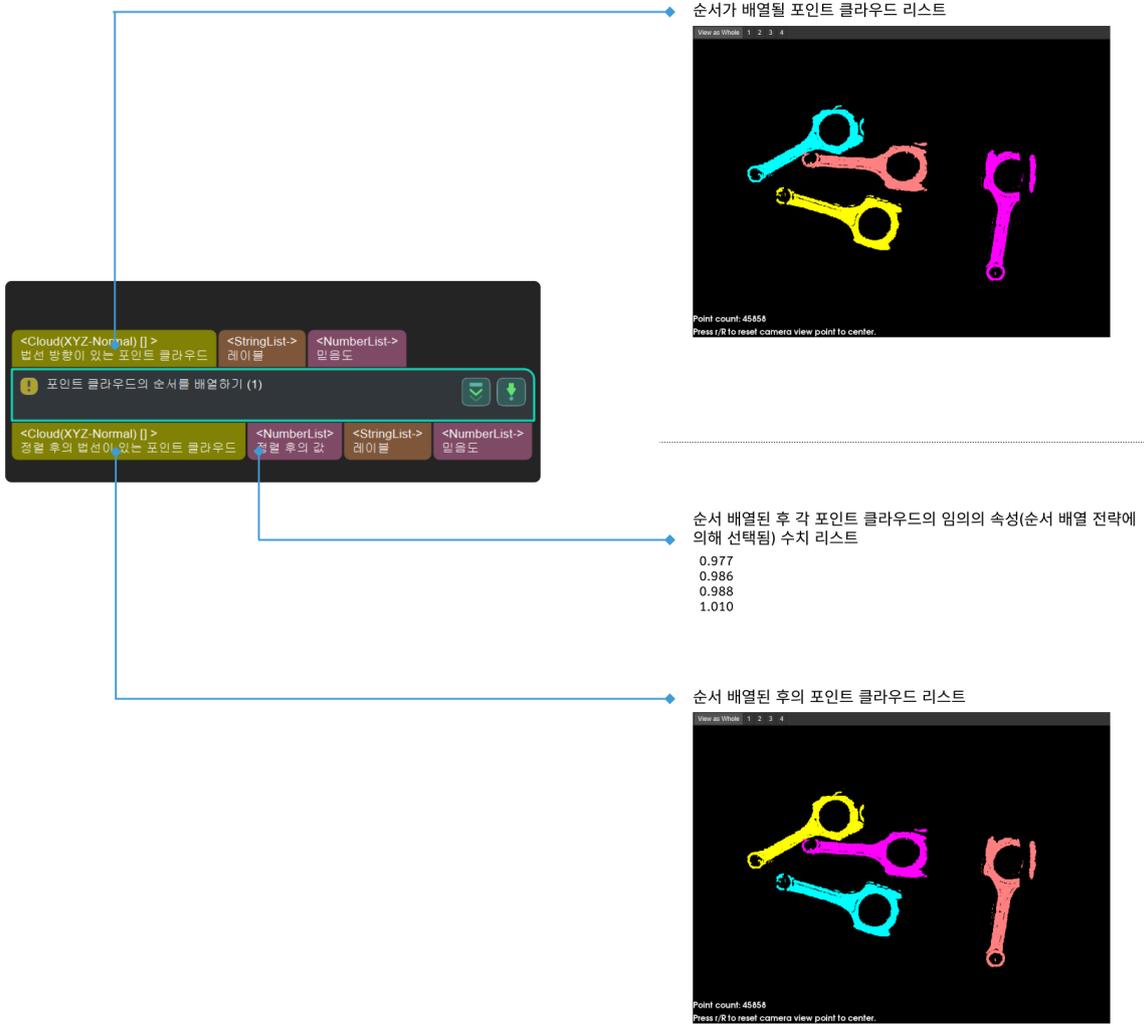
#### 기능 설명

지정된 규칙에 근거하여 포인트 클라우드의 순서를 배열하는 동시에 대응하는 레이블과 믿음도를 출력합니다.

### 응용 시나리오

피킹 순서를 정하는 데 사용됩니다.

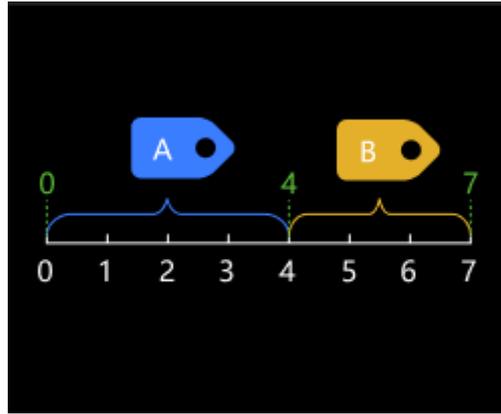
### 입력 및 출력



### 4.3.84. 역치에 근거하여 수치를 분류하기

#### 기능 설명

이 스텝은 설정된 역치에 따라 입력한 값을 분류하고 분류 레이블을 출력합니다.

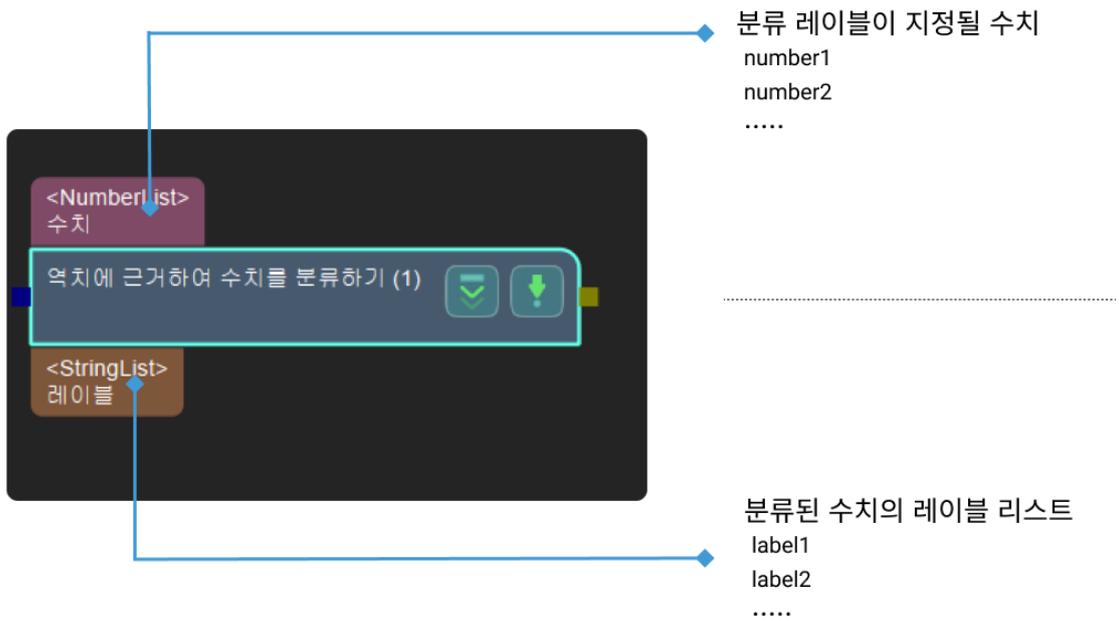


### 응용 시나리오

이 스텝은 값을 크기별로 분류하고 레이블을 지정하는 데 사용됩니다. 일반적으로 값 리스트를 값 범위별로 필터링하기 위해 **필요한 레이블인지 판단하기** 스텝과 함께 사용됩니다.

값 리스트를 역치에 따라 두 부분으로 나누고 한 부분을 필터링해야 하는 경우 **역치에 근거한 이진 분류** 스텝을 사용하고 **필터링** 스텝과 함께 사용합니다.

### 입력 및 출력



### 파라미터 설명

#### 파일 설정

#### 역치 파일 명칭

파라미터 설명: 이 파라미터는 역치를 설정하는 데 사용되며 입력한 값은 역치에 따라 분류되며 역치는 JSON 파일에 저장됩니다.

값 유형: 숫자



다른 유형의 데이터는 유효하지 않은 데이터로 간주되며 성공적으로 입력 할 수 없습니다.

조정 제안: 하나 또는 여러 개의 역치를 설정할 수 있으며 역치는 작은 것에서 큰 것으로 정렬되어야 합니다.

설정 방법:

1. [역치를 설정하기]를 클릭하여 JSON 편집기 페이지로 이동합니다.
2. **+**를 클릭하고 팝업 페이지에 값을 입력한 다음 [확인]을 클릭하면 단일 역치가 성공적으로 추가됩니다.
3. 여러 역치를 설정해야 하는 경우 단계 2를 반복합니다.



역치는 작은 것부터 커지는 순서로 정렬해야 합니다. 추가된 역치 순서가 요구 사항을 충족하지 않는 경우 **▲** 또는 **▼**를 클릭하여 역치 순서를 조정합니다.

4. **폴더 경로**에서 [···]을 클릭하여 JSON 파일을 저장할 경로를 선택합니다. 기본 경로는 프로젝트가 있는 경로입니다.
5. **JSON 파일 명칭**에 파일 명칭을 입력한 후 [저장]을 클릭하여 역치가 해당 파일에 저장되도록 합니다.
6. [확인]을 클릭하여 역치 추가를 완료합니다.



이미 미리 설정된 역치 파일이 있는 경우 **JSON 파일 명칭** 옆의 [···]을 직접 클릭하여 파일을 선택하고, 역치를 읽어올 수 있습니다.

### 레이블 파일 명칭

파라미터 설명: 이 파라미터는 데이터 분류 후 레이블을 지정하는 데 사용되며 레이블은 JSON 파일에 저장됩니다.

데이터 유형: 문자열 (특수 문자를 포함할 수 없음)



입력한 데이터가 요구 사항을 충족하지 않으면 데이터는 유효하지 않은 데이터로 간주되며 성공적으로 입력할 수 없습니다.

조정 설명: 레이블 수 = 역치 개수 + 1. 역치와 레이블 간의 해당 관계는 역치와 레이블의 정렬에 의해 결정되며 레이블 이름과 관련이 없습니다.



설정 방법:

1. [레이블을 설정하기]를 클릭하여 JSON 편집기 페이지로 이동합니다.
2. **+**를 클릭하고 팝업 페이지에 문자열을 입력한 다음 [확인]을 클릭하면 단일 레이블이 성공적으로 추가됩니다.
3. 2단계를 반복하여 레이블을 추가하고 레이블 수는 역치 개수에 1을 더한 값입니다.



추가된 레이블 순서가 요구 사항을 충족하지 않는 경우 **▲** 또는 **▼**를 클릭하여 레이블 순서를 조정합니다.

4. \*폴더 경로\*는 현재 프로젝트 경로입니다. 오른쪽의 [···]을 클릭하여 프로젝트 위치를 변경할 수

있으며, 필요에 따라 설정하십시오.

5. **JSON 파일 명칭**에 파일 명칭을 입력한 후 [**저장**]을 클릭하여 레이블이 해당 파일에 저장되도록 합니다.
6. [**확인**]을 클릭하여 레이블 추가를 완료합니다.



이미 미리 설정된 레이블 파일이 있는 경우 **JSON 파일 명칭** 옆의 [...]을 직접 클릭하여 파일을 선택하고, 레이블을 읽어올 수 있습니다.

#### 4.3.85. 실수 뺄셈



이 스텝의 내용은 유지 보수가 중단되었습니다. 새 버전의 스텝 **수치 계산**을 사용하십시오. 이 스텝에 대한 자세한 정보가 필요한 경우 ([docs@mech-mind.net](mailto:docs@mech-mind.net))로 문의하십시오.

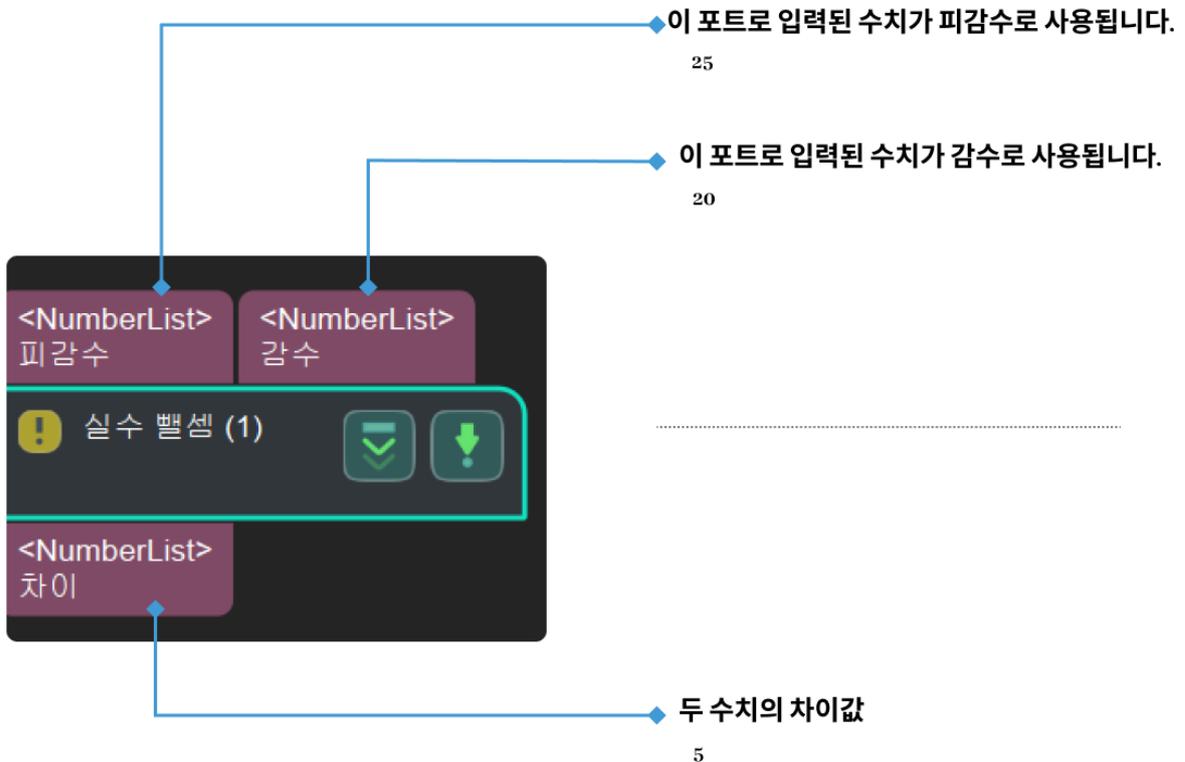
##### 기능 설명

두 입력 리스트의 요소 간의 차이를 일대일로 출력하기 위해 첫 번째 포트로 입력된 수치에서 두 번째 포트로 입력된 수치를 뺍니다.

##### 응용 시나리오

이 스텝은 구 버전이므로 보다 완전한 계산 기능을 갖춘 새 버전의 스텝 **수치 계산**을 사용하는 것이 좋습니다.

##### 입력 및 출력



#### 4.3.86. 일부 영역에서 점수가 낮은 포즈를 제거하기(NMS)



이 스텝의 내용은 유지 보수가 중단되었습니다. 새 버전의 스텝 **3D 근사 매칭**을 사용하십시오. 이 스텝에 대한 자세한 정보가 필요한 경우 ([docs@mech-mind.net](mailto:docs@mech-mind.net))로 문의하십시오.

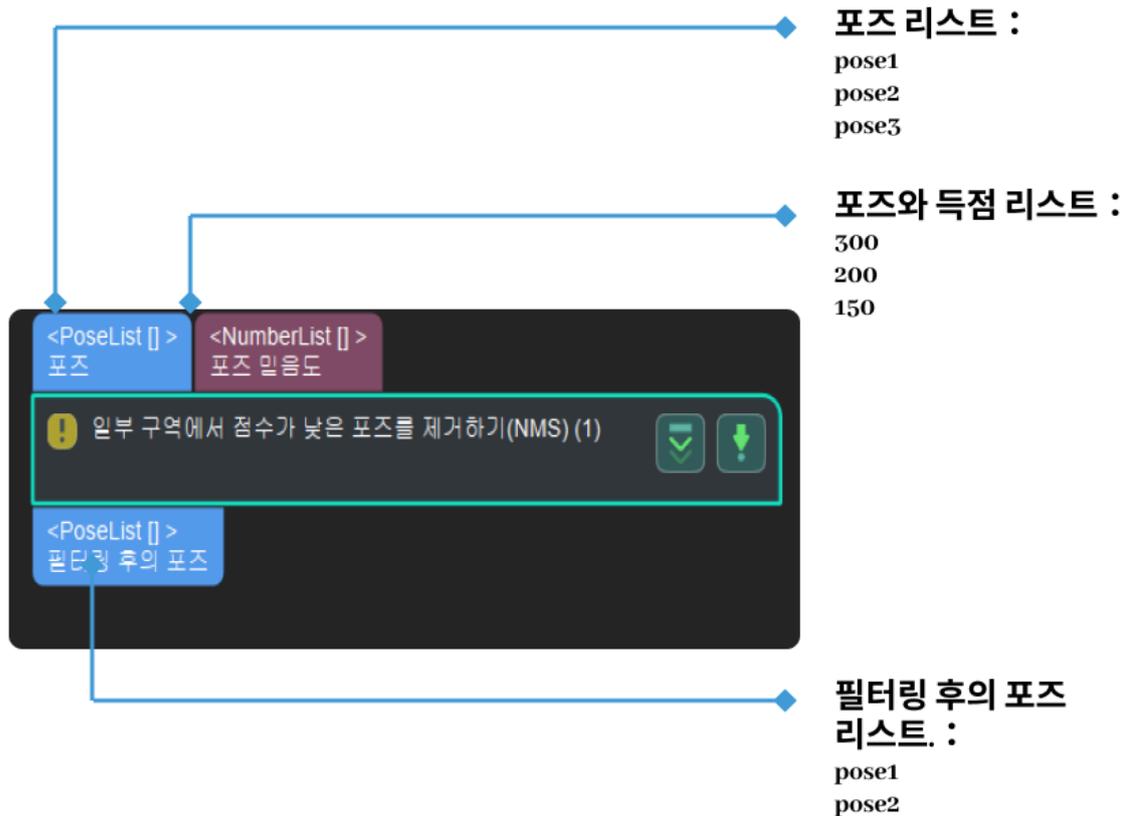
## 기능 설명

설정된 범위에서 특징이 가장 높은 포즈를 찾아 설정된 역치에 근거하여 근처에 특징이 낮은 포즈를 제거합니다.

## 응용 시나리오

포즈 필터링 방법 중 하나입니다.

## 입력 및 출력



## 파라미터 설명

### 이웃 포즈 검색 설정

기본값: 50.000 mm

조절 설명: 일부 영역에서 검색 범위의 크기를 밀리미터(mm) 단위로 정의합니다. 크게 조절하면 더 많은 포즈가 계산에 들어갈 것이고 후속 계산 시간이 더 늘어질 수 있습니다.

### 역치 설정

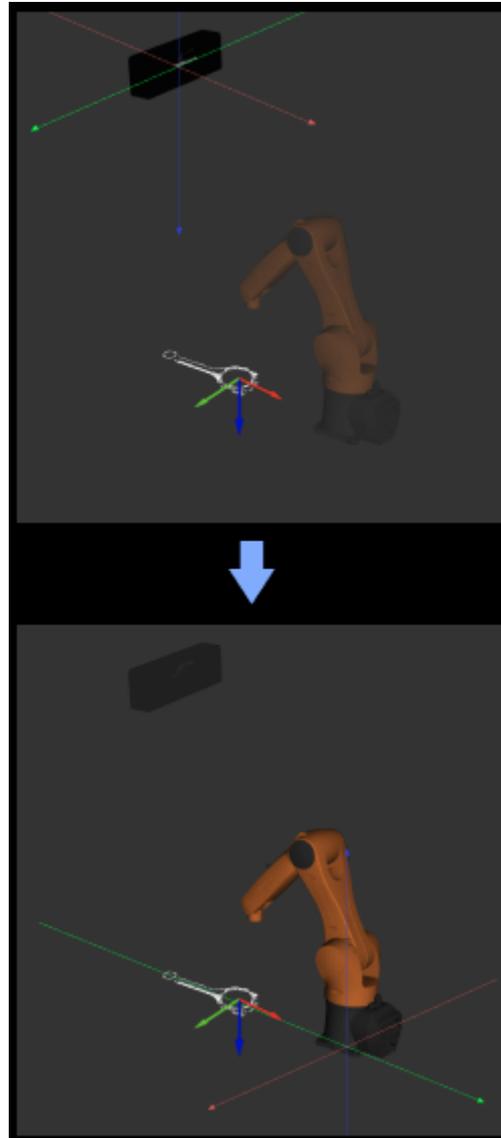
기본값: 0.02

조절 설명: 포즈의 점수가 역치보다 적으면 점수를 고려하지 않으며 점수가 늘이는 것은 포즈 점수에 대한 요구가 더 높은 것을 의미합니다.

### 4.3.87. 포인트 클라우드 변환

#### 기능 설명

이 스텝은 포인트 클라우드를 하나의 좌표계에서 다른 좌표계로 전환하거나 기준 포즈를 기반으로 하여 좌표계 내에서 변환할 수 있습니다.

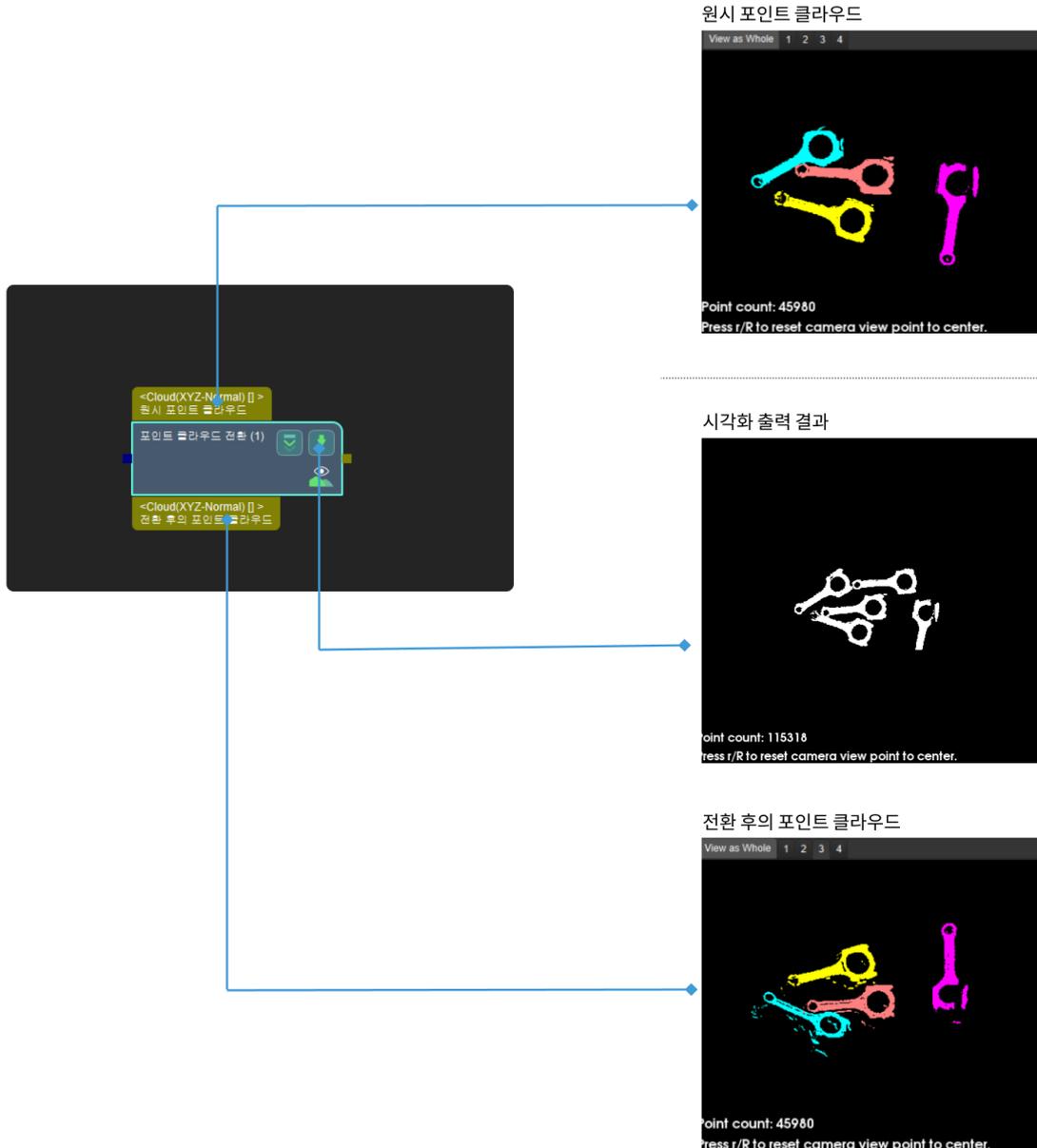


#### 응용 시나리오

이 스텝은 보통 카메라 좌표계 및 로봇 좌표계에서의 포인트 클라우드의 상호 전환에 사용됩니다.

프로젝트에서 트러스 로봇을 사용하는 경우 **포인트 클라우드 변환(트러스)** 스텝을 사용하도록 선택해야 합니다.

#### 입력 및 출력



## 파라미터 설명

### 변환 설정

### 변환 유형

파라미터 설명: 이 파라미터는 포인트 클라우드의 변환 유형을 설정하는 데 사용됩니다.

기본값: CameraToRobot

값 리스트: CameraToRobot, RobotToCamera, AllWithFirst, FirstWithAll, UseCorrespondenceInput, AffineTransform

조정 설명:

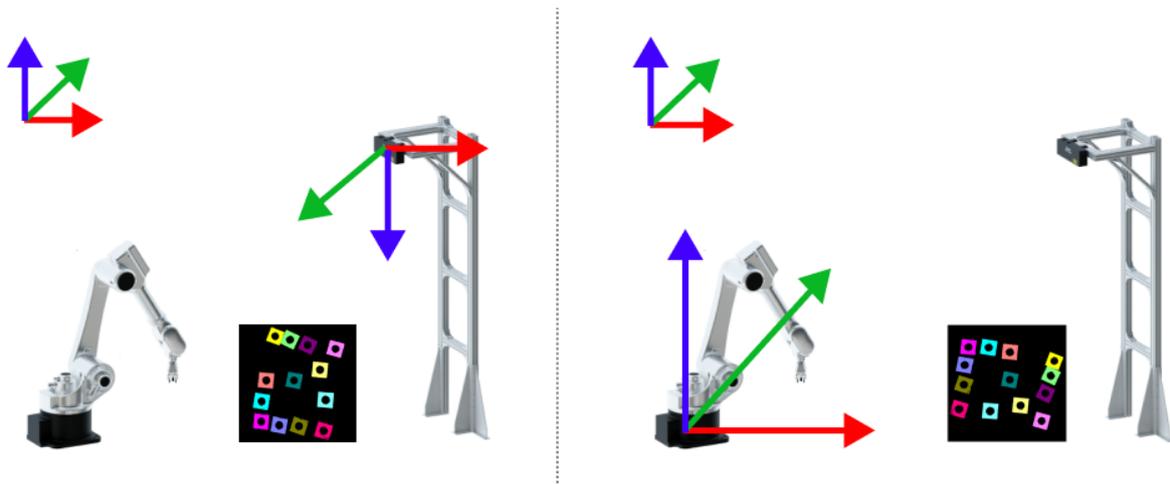
- CameraToRobot: 포인트 클라우드를 카메라 좌표계에서 로봇 좌표계로 전환합니다. 특정 효과에 대한 내용은 [조정 예시](#)를 참조하십시오.

- RobotToCamera: 포인트 클라우드를 로봇 좌표계에서 카메라 좌표계로 전환합니다. 특정 효과에 대한 내용은 [조정 예시](#)를 참조하십시오.
- AllWithFirst: 첫 번째 기준 포즈를 사용하여 포인트 클라우드를 변환합니다. 이 파라미터를 선택하면 기준 포즈 리스트를 입력해야 합니다. 특정 효과에 대한 내용은 [조정 예시](#)를 참조하십시오.
- FirstWithAll: 입력한 모든 기준 포즈를 사용하여 첫 번째 포인트 클라우드를 변환하려면 기준 포즈 리스트를 입력해야 합니다. 특정 효과에 대한 내용은 [조정 예시](#)를 참조하십시오.
- UseCorrespondenceInput: 원시 포인트 클라우드를 일대일로 기준 포즈가 있는 좌표계로 변환합니다. 이 파라미터를 선택하면 기준 포즈 리스트를 입력해야 하며 입력된 포즈의 수가 기준 포즈의 수와 같아야 합니다. 특정 효과에 대한 내용은 [조정 예시](#)를 참조하십시오.
- AffineTransform: 변환 매트릭스를 사용하여 포인트 클라우드를 변환합니다. 스텝 "PoseList를 Matrix4D로 전환하기"를 사용하여 postlist를 matrix4D로 전환하여 이 스텝의 Matrix4D 입력 포트에 입력할 수 있습니다. 특정 효과에 대한 내용은 [조정 예시](#)를 참조하십시오.

## 조정 예시

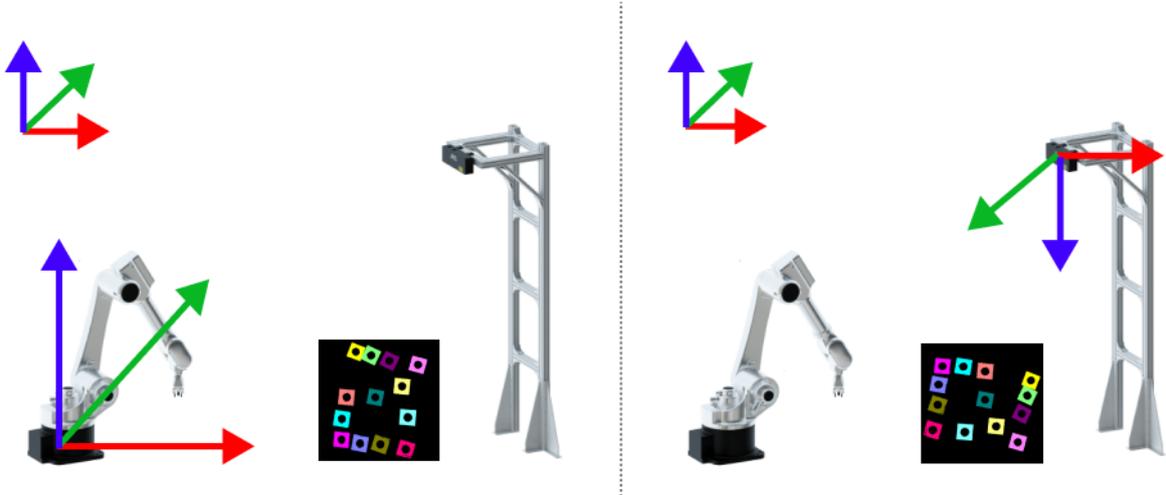
### CameraToRobot 변환 유형

**CameraToRobot** 포인트 클라우드 변환 효과는 아래 그림과 같습니다. 왼쪽은 변환하기 전이고 오른쪽은 변환한 후입니다.



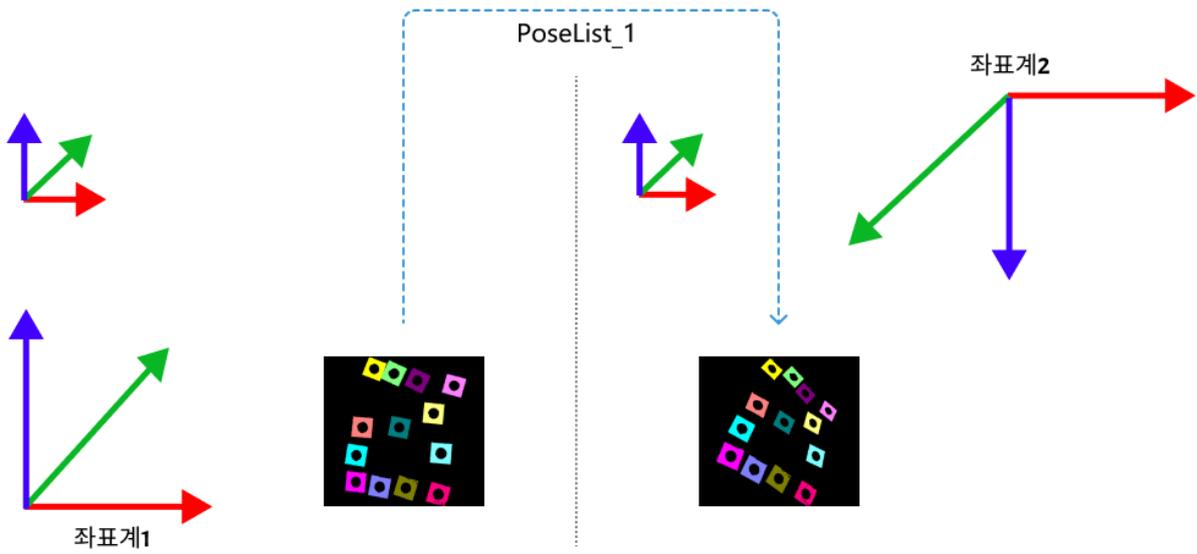
### RobotToCamera 변환 유형

**RobotToCamera** 포인트 클라우드 변환 효과는 아래 그림과 같습니다. 왼쪽은 변환하기 전이고 오른쪽은 변환한 후입니다.



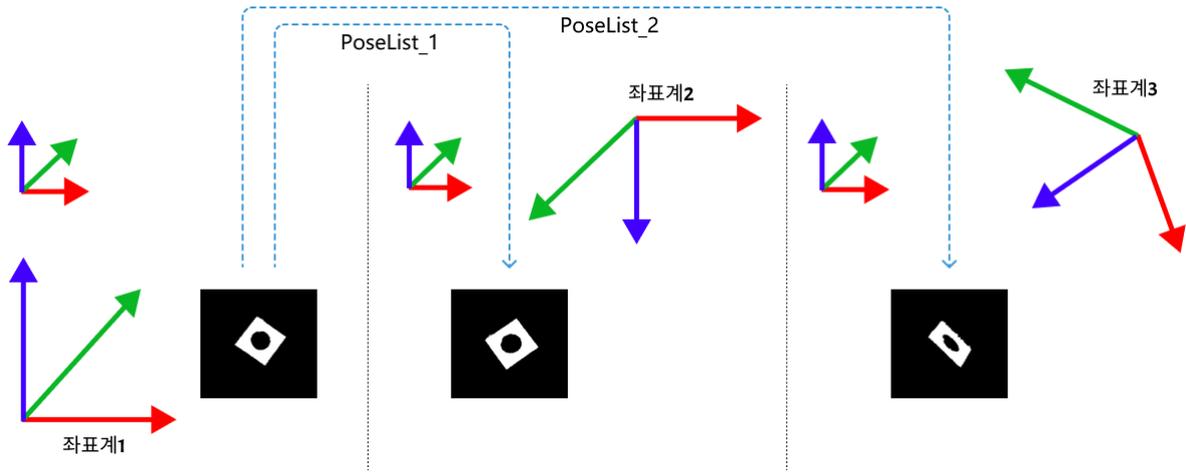
**AllWithFirst 변환 유형**

**AllWithFirst** 포인트 클라우드 변환 효과는 아래 그림과 같습니다. 왼쪽은 변환하기 전이고 오른쪽은 변환한 후입니다.



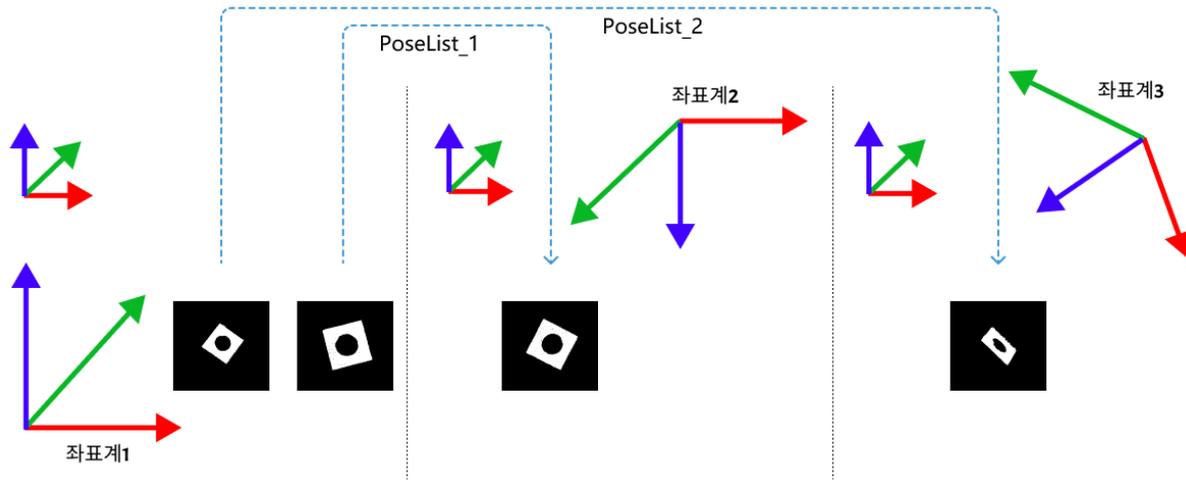
**FirstWithAll 변환 유형**

**FirstWithAll** 포인트 클라우드 변환 효과는 아래 그림과 같습니다.



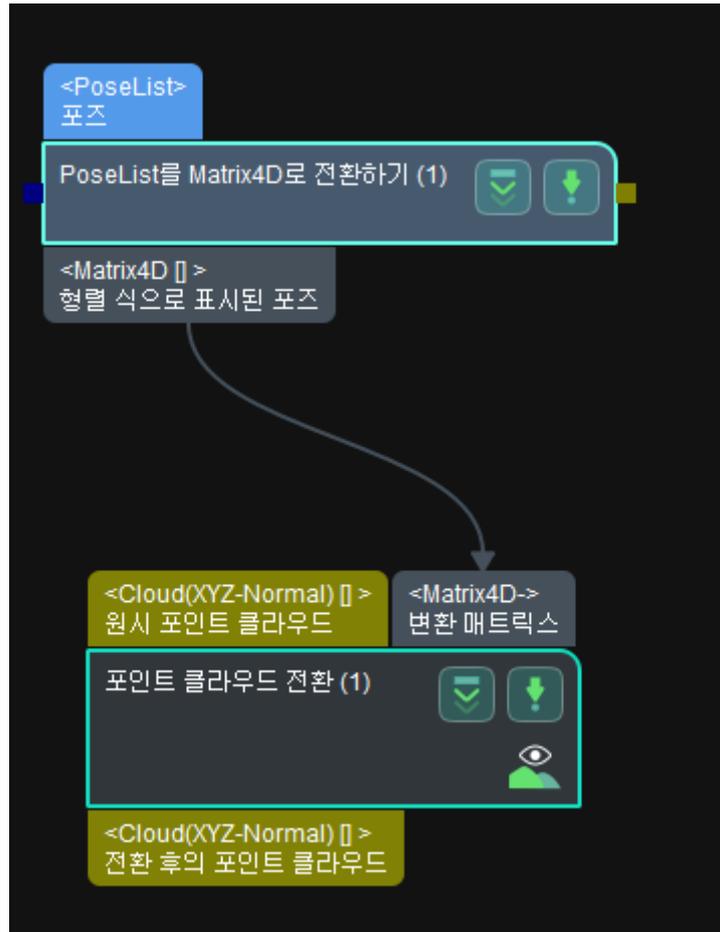
### UseCorrespondenceInput 변환 유형

UseCorrespondenceInput 포인트 클라우드 변환 효과는 아래 그림과 같습니다.



### AffineTransform 변환 유형

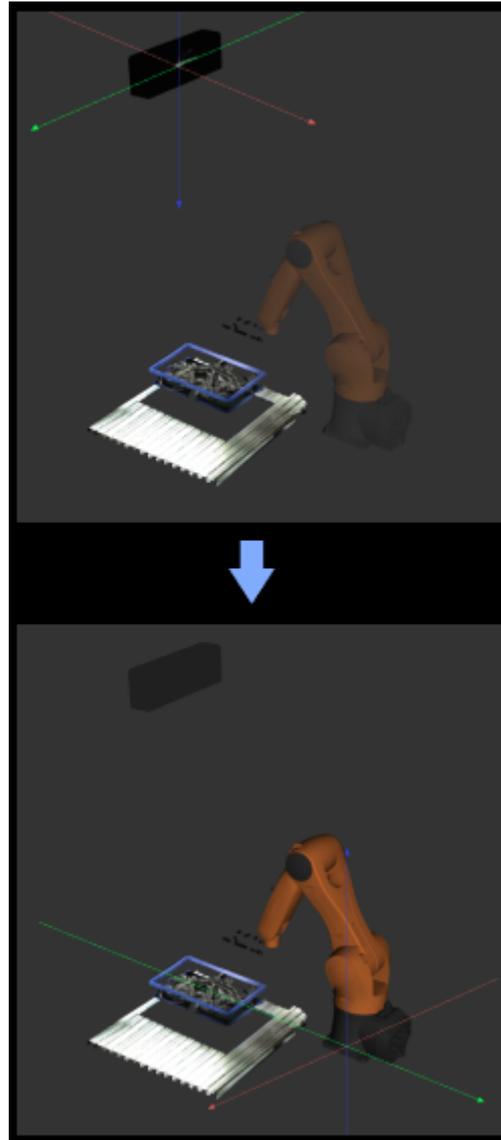
AffineTransform 포인트 클라우드 변환 효과는 아래 그림과 같습니다.



### 4.3.88. 포즈 변환

#### 기능 설명

이 스텝은 포즈를 하나의 좌표계에서 다른 좌표계로 전환하거나 기준 포즈를 기반으로 하여 좌표계 내에서 변환할 수 있습니다.

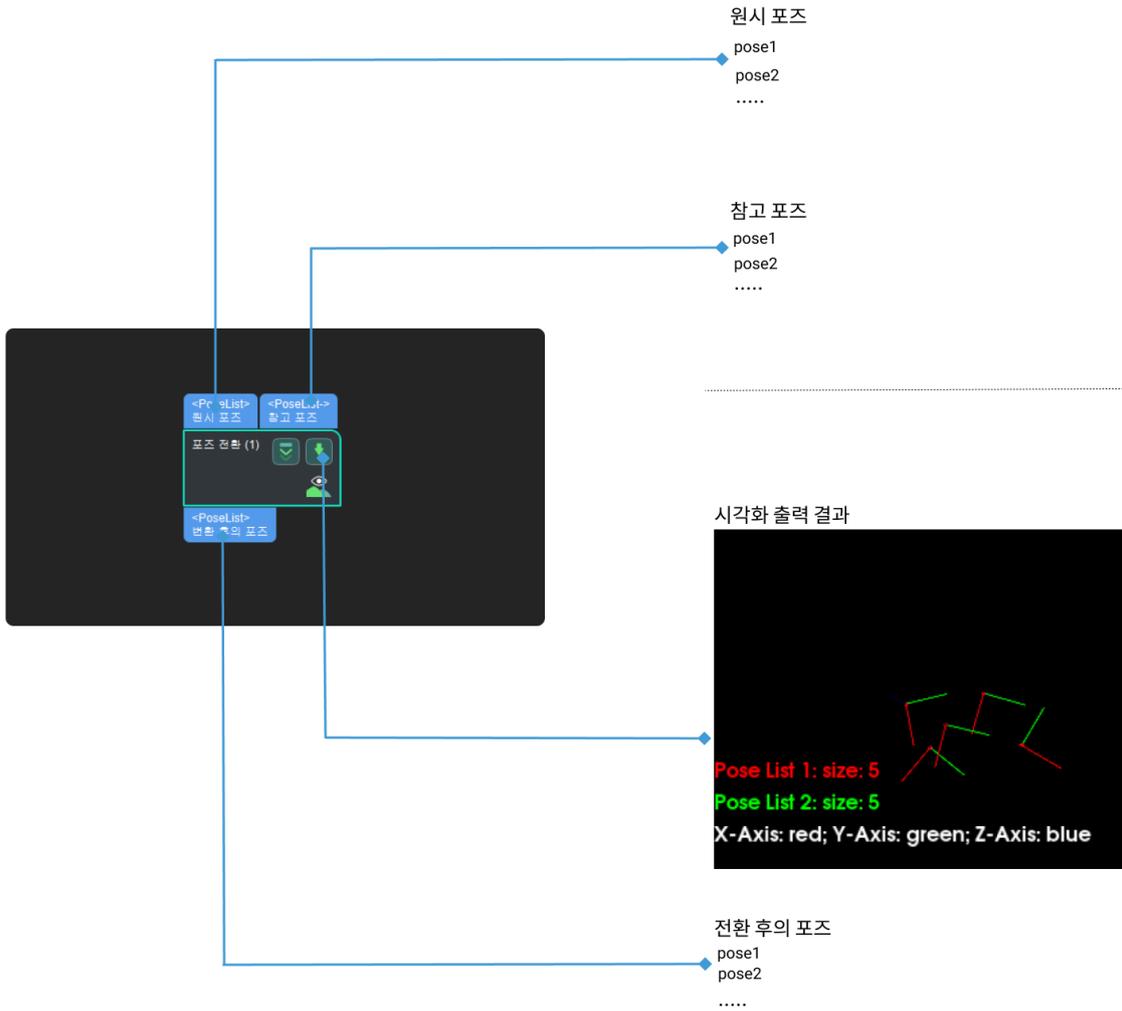


### 응용 시나리오

이 스텝은 보통 포즈가 카메라 기준 좌표계 및 로봇 기준 좌표계 간의 상호 전환에 사용됩니다.

프로젝트에서 트러스 로봇을 사용하는 경우 [포즈 변환\(트러스\)](#) 스텝을 사용하도록 선택해야 합니다.

### 입력 및 출력



## 파라미터 설명

### 시각화 설정

#### 선택한 좌표계에서 포인트 클라우드를 표시하기

파라미터 설명: 로봇 또는 카메라 좌표계에서 포인트 클라우드를 표시합니다.

기본값: 로봇 좌표계

값 리스트: 로봇 좌표계, 카메라 좌표계

설명: **카메라 좌표계**를 선택할 때 디버그 출력 창에서 카메라 좌표계에서의 시각화 출력 결과를 표시합니다. **로봇 좌표계**를 선택할 때 디버그 출력 창에서 로봇 좌표계에서의 시각화 출력 결과를 표시합니다. 특정 효과에 대한 내용은 [조정 예시](#)를 참조하십시오.



프로젝트에 시나리오 포인트 클라우드를 설정한 경우에만 이 파라미터를 설정할 수 있습니다.



- **로봇 좌표계**를 선택하는 경우 디버그 출력 창은 검은색이고 시각을 조정해야 포인트 클라우드 및 좌표를 볼 수 있습니다.
- **카메라 좌표계**를 선택하는 경우 디버그 출력 창에는 포인트 클라우드만 표시하고 포인트 클라우드 방향을 조정해야 좌표를 볼 수 있습니다.

## 포즈 시각화 설정

파라미터 설명: 이 파라미터는 포즈의 시각화 형식을 설정하는 데 사용됩니다.

기본값: 출력한 포즈만 표시하기

값 리스트: 출력한 포즈만 표시하기, 입력한 포즈만 표시하기, 출력한 포즈와 입력한 포즈를 모두 표시하기

조정 설명: 실제 필요에 따라 드롭다운 목록에서 적절한 포즈 표시 설정을 선택합니다. 특정 효과에 대한 내용은 [조정 예시](#)를 참조하십시오.

## 변환 설정

### 변환 유형

파라미터 설명: 이 파라미터는 포즈의 변환 유형을 설정하는 데 사용됩니다.

기본값: CameraToRobot

값 리스트: CameraToRobot, RobotToCamera, AllWithFirst, FirstWithAll, UseCorrespondenceInput

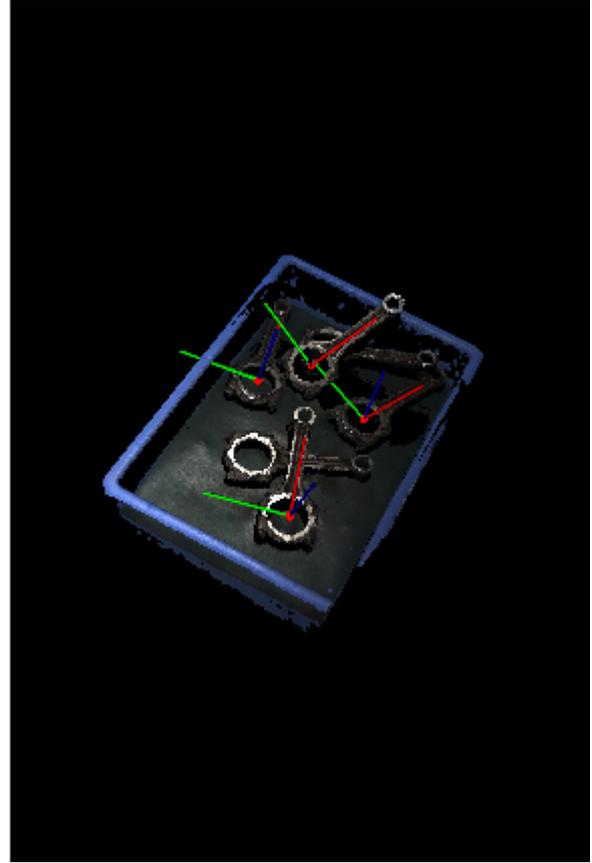
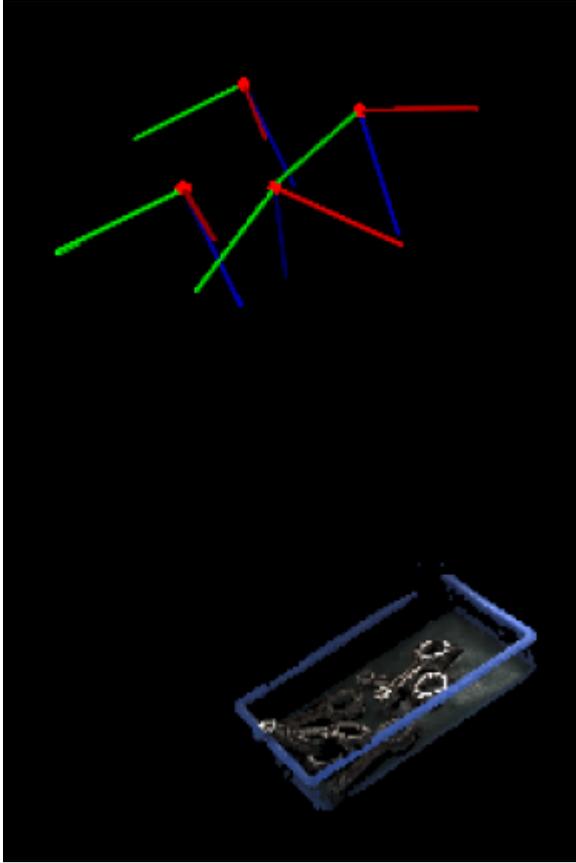
조정 설명:

- CameraToRobot : 카메라 좌표계에서의 물체의 포즈 Pose1이 알려진 경우 변환 유형을 **CameraToRobot**으로 설정하면 로봇 좌표계에서의 물체의 포즈 Pose2를 출력할 수 있습니다. 특정 효과에 대한 내용은 [조정 예시](#)를 참조하십시오.
- RobotToCamera : 로봇 좌표계에서의 물체의 포즈 Pose1이 알려진 경우 변환 유형을 **RobotToCamera**로 선택하면 카메라 좌표계에서의 물체의 포즈 Pose2를 출력할 수 있습니다. 특정 효과에 대한 내용은 [조정 예시](#)를 참조하십시오.
- AllWithFirst : 좌표계 1에서의 물체 A, B의 포즈 Pose1A, Pose1B가 알려진 경우 변환 유형을 **AllWithFirst**로 선택할 때(기준 포즈를 입력해야 함) 좌표계 2에서의 물체 A, B의 포즈 Pose2A, Pose2B를 출력할 수 있습니다. 변환하는 동안 첫 번째 기준 포즈를 사용하여 변환할 모든 포즈를 변환해야 합니다. 특정 효과에 대한 내용은 [조정 예시](#)를 참조하십시오.
- FirstWithAll : 좌표계 1에서의 물체 A, B의 포즈 Pose1A, Pose1B가 알려진 경우 변환 유형을 **FirstWithAll**로 선택할 때(기준 포즈를 입력해야 함) 좌표계 2 및 좌표계 3에서의 물체 A의 포즈 Pose2A, Pose3A를 출력할 수 있습니다. 변환하는 동안 모든 기준 포즈를 사용하여 첫 번째 변환할 포즈를 변환해야 합니다. 특정 효과에 대한 내용은 [조정 예시](#)를 참조하십시오.
- UseCorrespondenceInput : 좌표계 1에서의 물체 A, B의 포즈 Pose1A, Pose1B가 알려진 경우 변환 유형을 **UseCorrespondenceInput**로 선택할 때(기준 포즈를 입력해야 함) 좌표계 2에서의 물체 A의 포즈 Pose2A와 좌표계 3에서의 물체 B의 포즈 Pose3B를 출력할 수 있습니다. 변환하는 동안 기준 포즈를 사용해서 1대1로 변환할 포즈를 변환해야 합니다. 특정 효과에 대한 내용은 [조정 예시](#)를 참조하십시오.

## 조정 예시

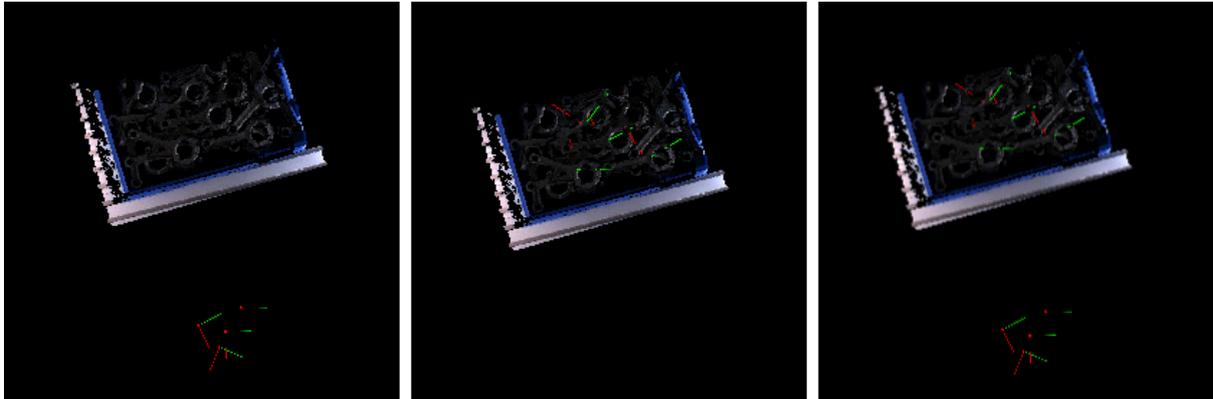
### 선택한 좌표계에서 포인트 클라우드를 표시하기

아래 그림과 같이 왼쪽은 카메라 좌표계의 포인트 클라우드이며 오른쪽은 로봇 좌표계의 포인트 클라우드입니다.



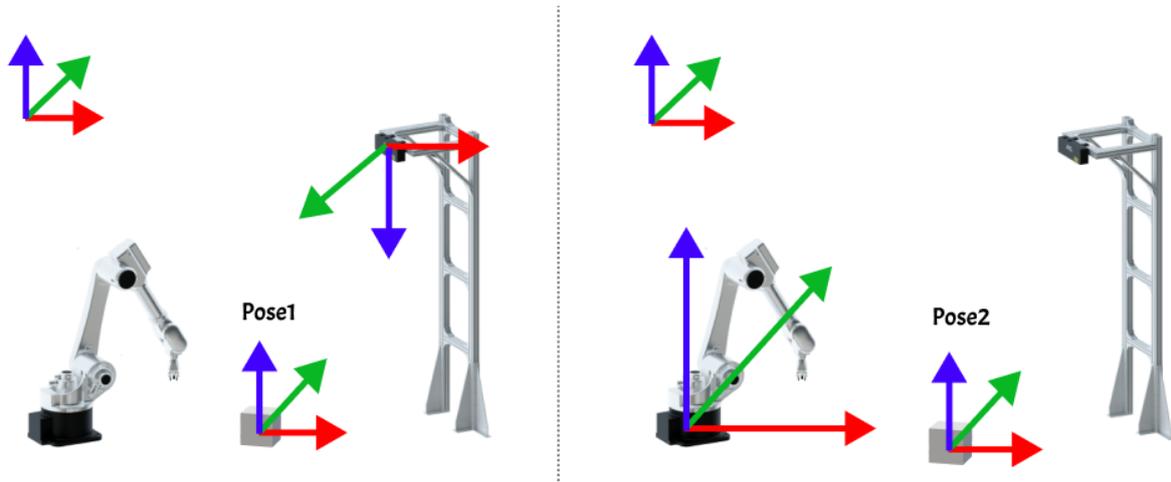
### 포즈 시각화 설정

설명: 카메라 좌표계를 예시로 설명하자면 **출력된 포즈만 표시하기**, **입력한 포즈만 표시하기**, **입력과 출력 포즈를 함께 표시하기**를 선택할 때 각각의 포즈 표시 효과는 아래 그림과 같습니다.



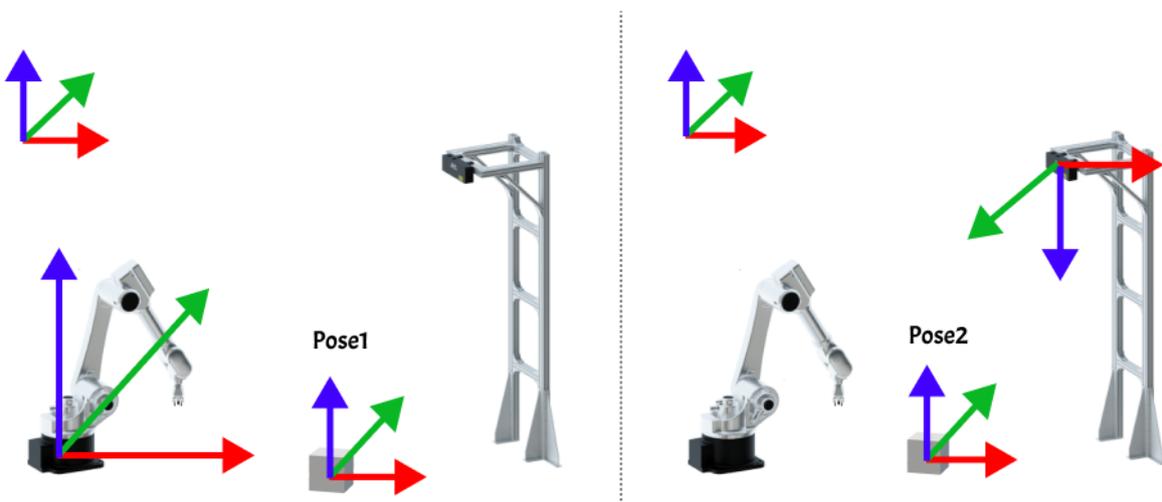
### CameraToRobot 변환 유형

**CameraToRobot** 포즈 변환 효과는 아래 그림과 같습니다. 왼쪽은 변환하기 전이고 오른쪽은 변환한 후입니다.



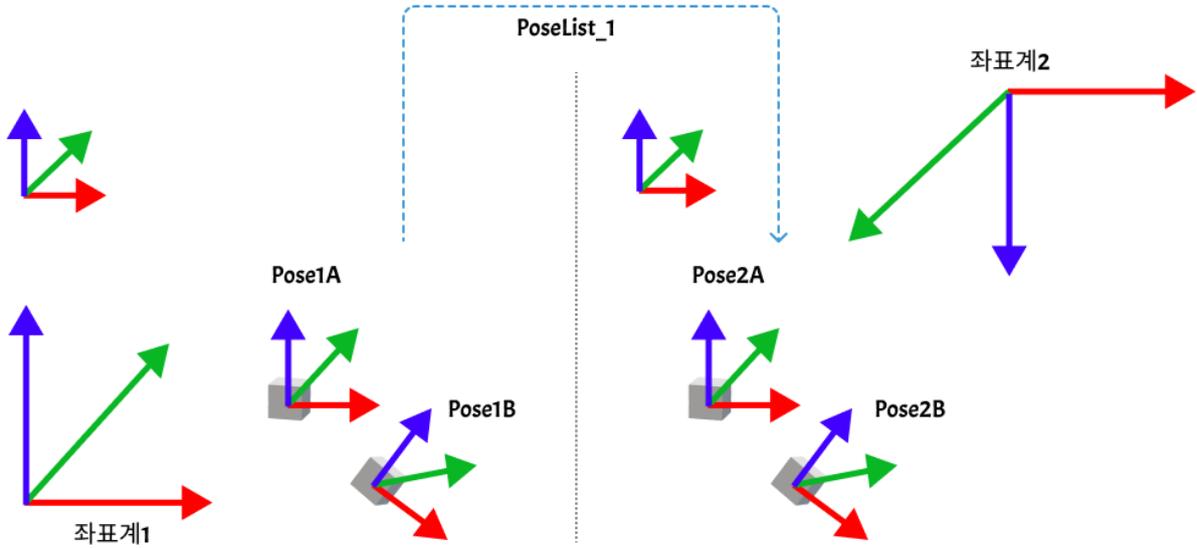
### RobotToCamera 변환 유형

**RobotToCamera** 포즈 변환 효과는 아래 그림과 같습니다. 왼쪽은 변환하기 전이고 오른쪽은 변환한 후입니다.



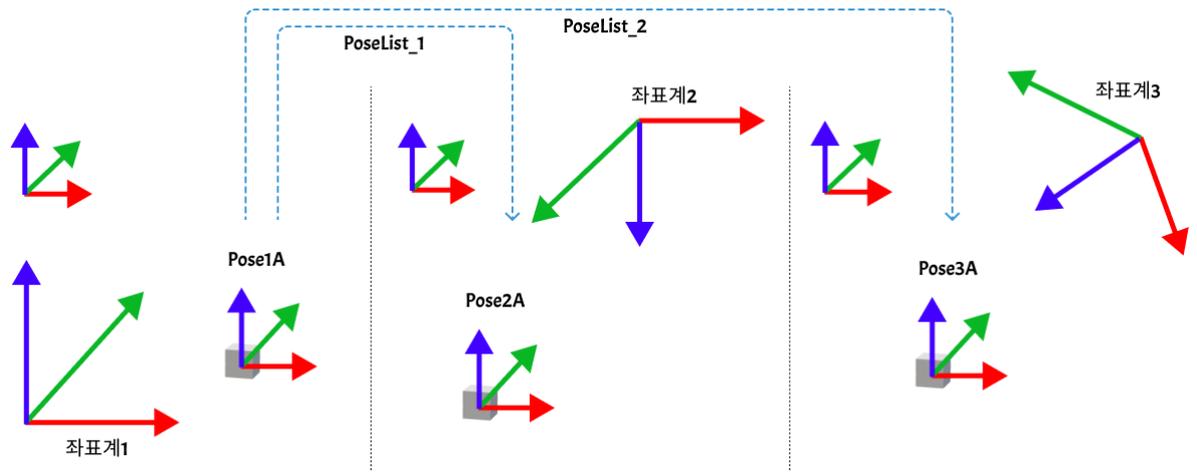
### AllWithFirst 변환 유형

**AllWithFirst** 포즈 변환 효과는 아래 그림과 같습니다. 왼쪽은 변환하기 전이고 오른쪽은 변환한 후입니다.



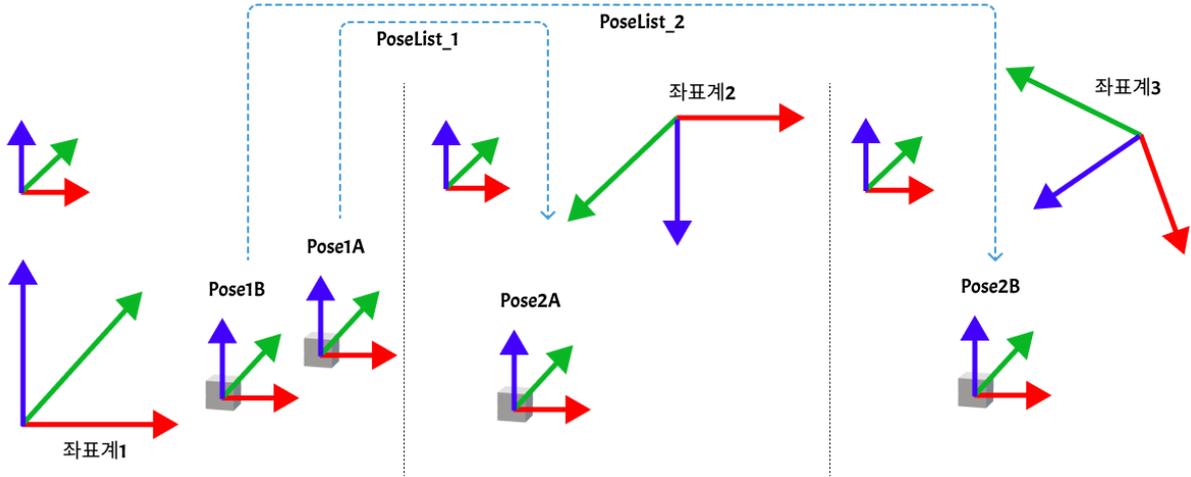
### FirstWithAll 변환 유형

FirstWithAll 포즈 변환 효과는 아래 그림과 같습니다.



### UseCorrespondenceInput 변환 유형

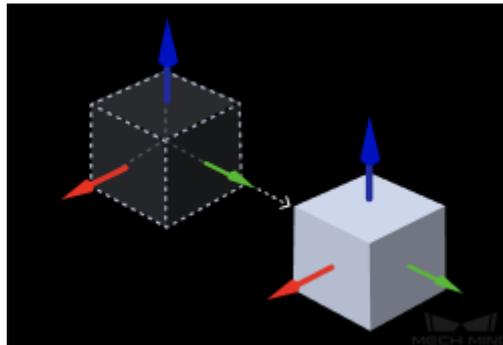
UseCorrespondenceInput 포즈 변환 효과는 아래 그림과 같습니다.



### 4.3.89. 지정한 방향으로 포즈를 이동하기

#### 기능 설명

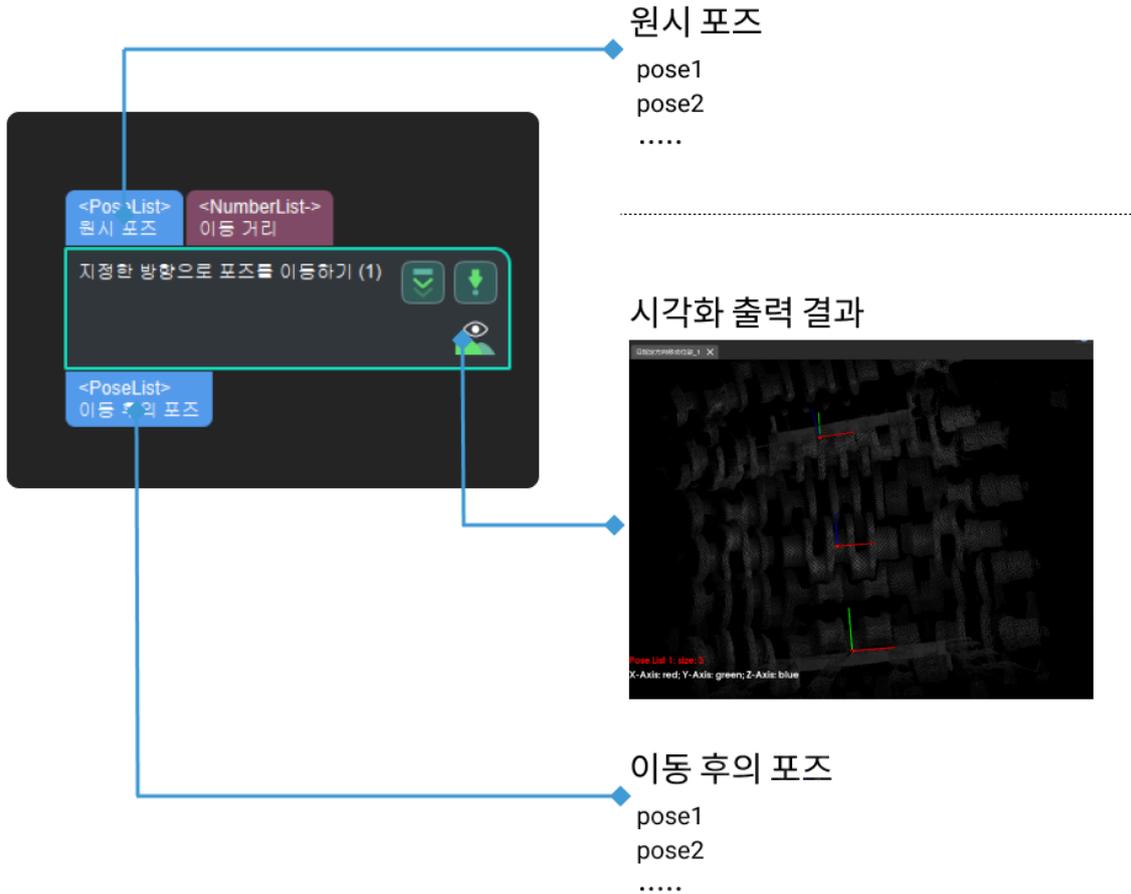
입력된 포즈를 평행 이동합니다.



#### 응용 시나리오

사용자가 지정한 방향 및 거리대로 포즈를 평행이동합니다.

#### 입력 및 출력



**파라미터 설명**

**시각화 설정**

**선택한 좌표계에서 포인트 클라우드를 표시하기**

기본값: 로봇 좌표계

값 리스트: 로봇 좌표계, 카메라 좌표계

조정 설명: [선택한 좌표계에서 포인트 클라우드를 표시하기](#)를 참조하십시오.

**포즈 시각화 설정**

기본값: 출력한 포즈만 표시하기

값 리스트: 출력한 포즈만 표시하기, 입력한 포즈만 표시하기, 출력한 포즈와 입력한 포즈를 모두 표시하기

조정 설명: [선택한 좌표계에서 포인트 클라우드를 표시하기](#)를 참조하십시오.

**평행이동 거리 설정**

**평행이동 거리**

기본값: 0

조정 설명: 실제 상황에 따라 설정하십시오.

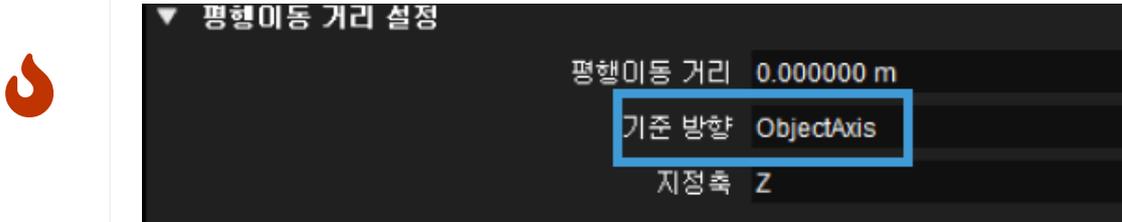
## 기준 방향

기본값: CustomizedDirection(자체 정의한 방향에 따라 평행이동합니다)

값 리스트: CustomizedDirection(자체 정의한 방향에 따라 평행이동합니다), ObjectAxis(물체의 지정된 축에 따라 평행이동합니다)

조정 설명: 실제 상황에 따라 선택하십시오.

기준 방향을 ObjectAxis로 선택할 때 실제 상황에 따라 지정축을 선택할 수 있습니다.



**자체 정의한 방향**(기준 방향을 CustomizedDirection로 선택할 때만 이 옵션을 볼 수 있습니다)

X

기본값: 0

Y

기본값: 0

Z

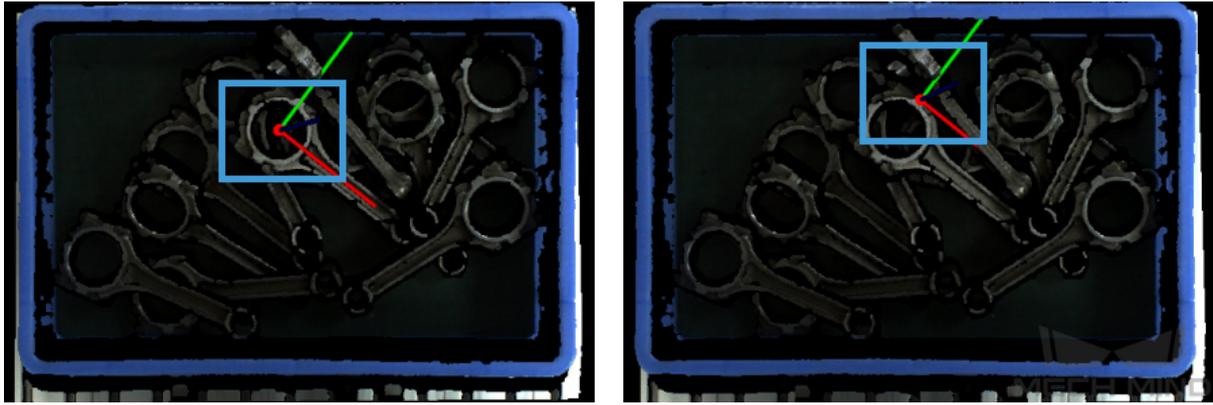
기본값: 1

## 조정 예시

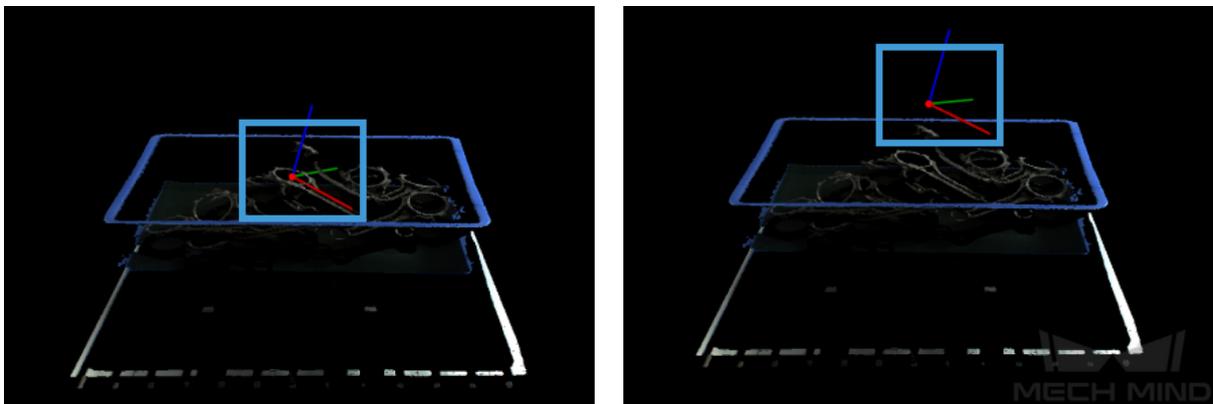
- 기준 방향은 ObjectAxis, 지정축은 X축, 이동 거리는 각각 0 및 0.03일 때 포즈의 이동 효과는 아래 그림과 같습니다. 그중에 왼쪽 그림은 이동 거리가 0인 경우이고, 오른쪽 그림은 왼쪽 그림에 있는 포즈가 X축 방향에 따라 0.03m 이동한 후의 결과입니다.



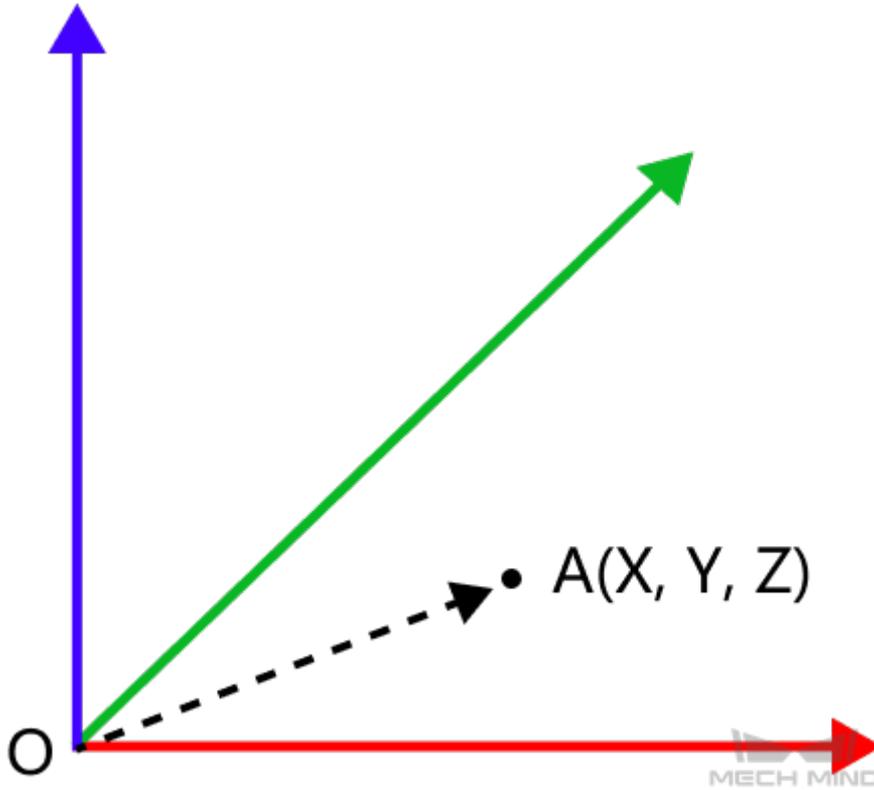
- 기준 방향은 ObjectAxis, 지정축은 Y축, 이동 거리는 각각 0 및 0.03일 때 포즈의 이동 효과는 아래 그림과 같습니다. 그중에 왼쪽 그림은 이동 거리가 0인 경우이고, 오른쪽 그림은 왼쪽 그림에 있는 포즈가 Y축 방향에 따라 0.03m 이동한 후의 결과입니다.



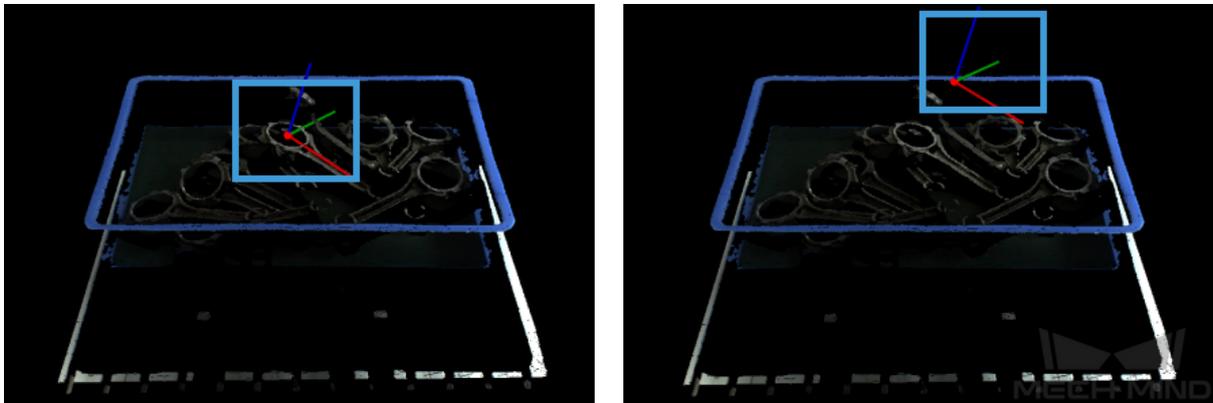
- 기준 방향은 ObjectAxis, 지정축은 Z축, 이동 거리는 각각 0 및 0.08일 때 포즈의 이동 효과는 아래 그림과 같습니다. 그중에 왼쪽 그림은 이동 거리가 0인 경우이고, 오른쪽 그림은 왼쪽 그림에 있는 포즈가 Z축 방향에 따라 0.08m 이동한 후의 결과입니다.



- 기준 방향이 CustomizedDirection 일 때 자체 정의한 방향에서 X, Y, Z의 값을 설정해야 합니다. 아래 그림과 같이 원점 O에서 A(X, Y, Z)로 향하는 방향이 포즈 평행 이동의 방향입니다.



X, Y, Z를 5로 설정하고 이동 거리를 0.08mm로 설정했을 때 포즈 평행 이동에 따른 효과는 아래 그림과 같습니다. 왼쪽 이미지는 움직이기 전의 포즈이고, 오른쪽 이미지는 왼쪽 이미지를 자체 정의한 방향으로 0.08mm 이동한 결과입니다.



#### 4.3.90. 부울 값에 의해 제어 흐름을 촉발하기



이 스텝의 내용은 유지 보수 중입니다. 이 스텝에 대한 자세한 정보가 필요한 경우 ([docs@mech-mind.net](mailto:docs@mech-mind.net))로 문의하십시오.

##### 기능 설명

입력된 부울 값(True/False) 및 설정된 규칙에 근거하여 제어 흐름을 촉발합니다.

##### 응용 시나리오

프로젝트의 분기 구조에서 사용되며 다양한 규칙에 따라 다양한 후속 스텝을 트리거 합니다. 일반적으로 **역치에 근거한 이진 분류** 스텝과 같이 부울 값을 출력하는 스텝과 함께 사용됩니다.

입력 및 출력

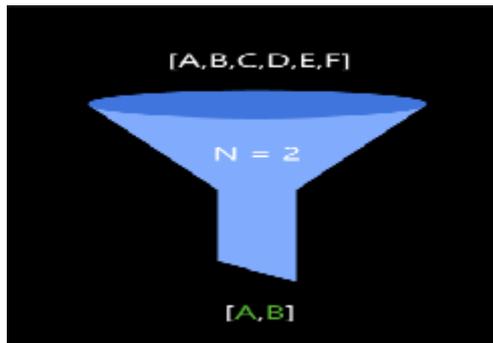


출력 없음

4.3.91. 출력 수 제한

기능 설명

이 스텝은 데이터 리스트의 상위 N개 항목을 유지하고 출력하는 데 사용되며 N번째 항목 이후 항목은 폐기됩니다.

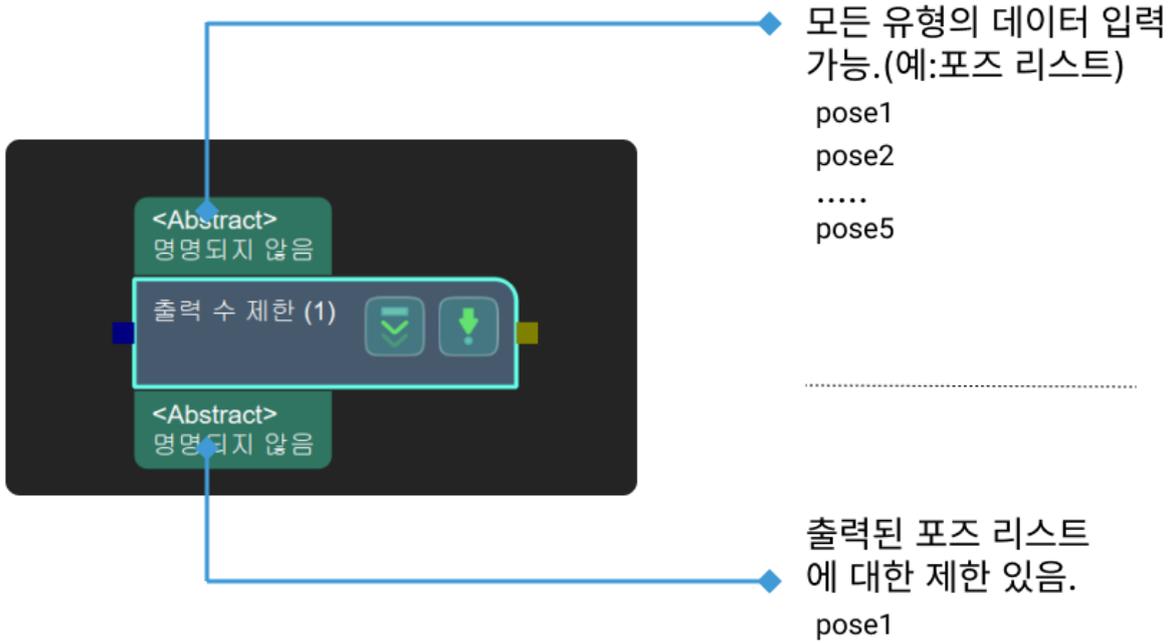


응용 시나리오

이 스텝은 리스트에서 상위 N개 항목을 추출하는 데 사용되며 N 값은 요구 사항에 따라 파라미터에서 설정할 수 있습니다.

입력 및 출력

이 스텝에서 입력과 출력은 동일한 데이터 유형입니다. 예를 들어 입력이 포즈 데이터인 경우 이 스텝의 입력과 출력은 다음과 같습니다.



## 파라미터 설명

### 포트 설정

#### 포트 수 (1~15)

파라미터 설명: 이 파라미터는 입력 포트와 출력 포트의 수를 설정하는 데 사용됩니다.

기본값: 1

값 리스트: 1~15

#### 작업 중인 레이어

파라미터 설명: 이 파라미터는 리스트에서 작업을 수행해야 하는 요소의 레이어를 지정하는 데 사용됩니다. 이 파라미터의 값이 0이면 리스트의 모든 요소에 대해 작업이 수행됩니다. 그렇지 않으면 지정된 레이어의 요소에 대해서만 작업이 수행됩니다.

기본값: 0

#### 출력 결과 수

파라미터 설명: 이 파라미터는 출력 결과의 수를 설정하는 데 사용됩니다.

기본값: 1

조절 제안: 이 파라미터는 실제 필요에 따라 설정합니다.

## 4.3.92. 포즈의 개수를 제한하기



이 스텝의 내용은 유지 보수가 중단되었습니다. 새 버전의 스텝 [출력 수 제한](#)을 사용하십시오. 이 스텝에 대한 자세한 정보가 필요한 경우 ([docs@mech-mind.net](mailto:docs@mech-mind.net))로 문의하십시오.

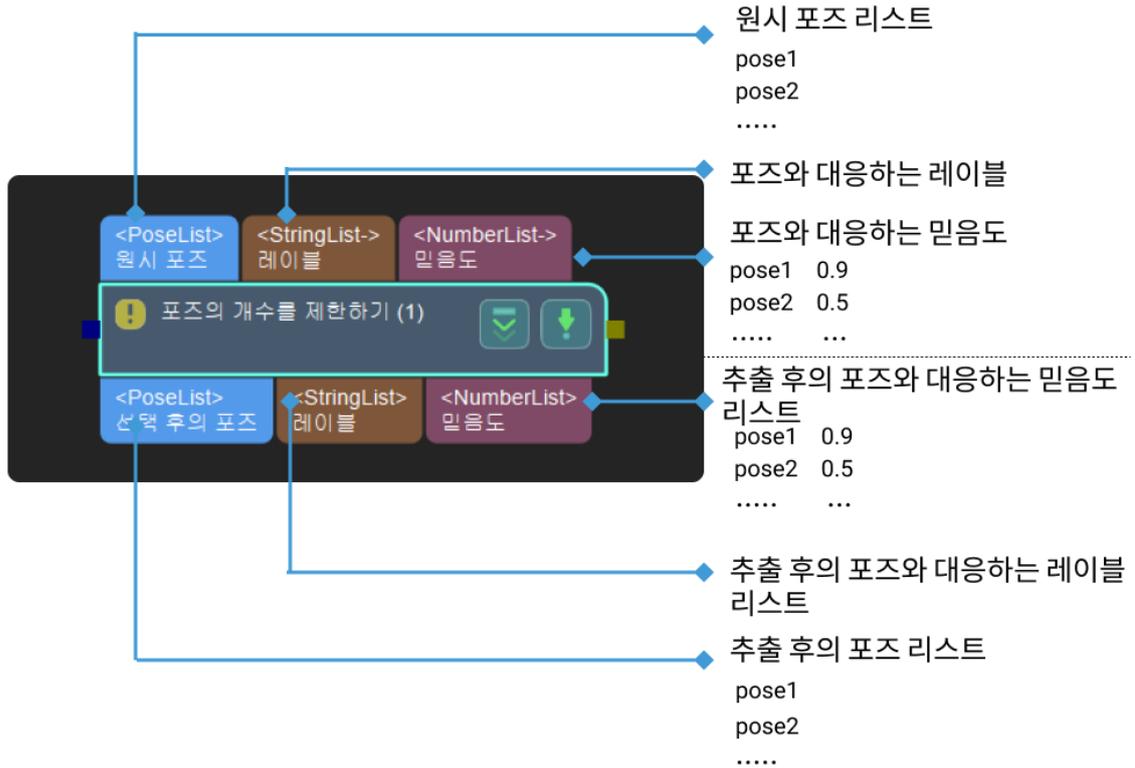
## 기능 설명

포즈 리스트의 상위 N개 항목은 유지되어 출력되며, N번째 항목 이후의 항목은 폐기됩니다.

## 응용 시나리오

포즈 리스트에서 상위 N번째 포즈까지 추출합니다. N의 수치는 수요에 따라 파라미터에서 설정할 수 있습니다.

## 입력 및 출력



## 파라미터 설명

### 출력할 포즈의 수량 제한

기본값: 1

설명: 출력할 포즈의 개수를 정합니다.

### 4.3.93. 포즈가 3D ROI에 있는지 확인하기



이 스텝의 내용은 유지 보수 중입니다. 이 스텝에 대한 자세한 정보가 필요한 경우 ([docs@mech-mind.net](mailto:docs@mech-mind.net))로 문의하십시오.

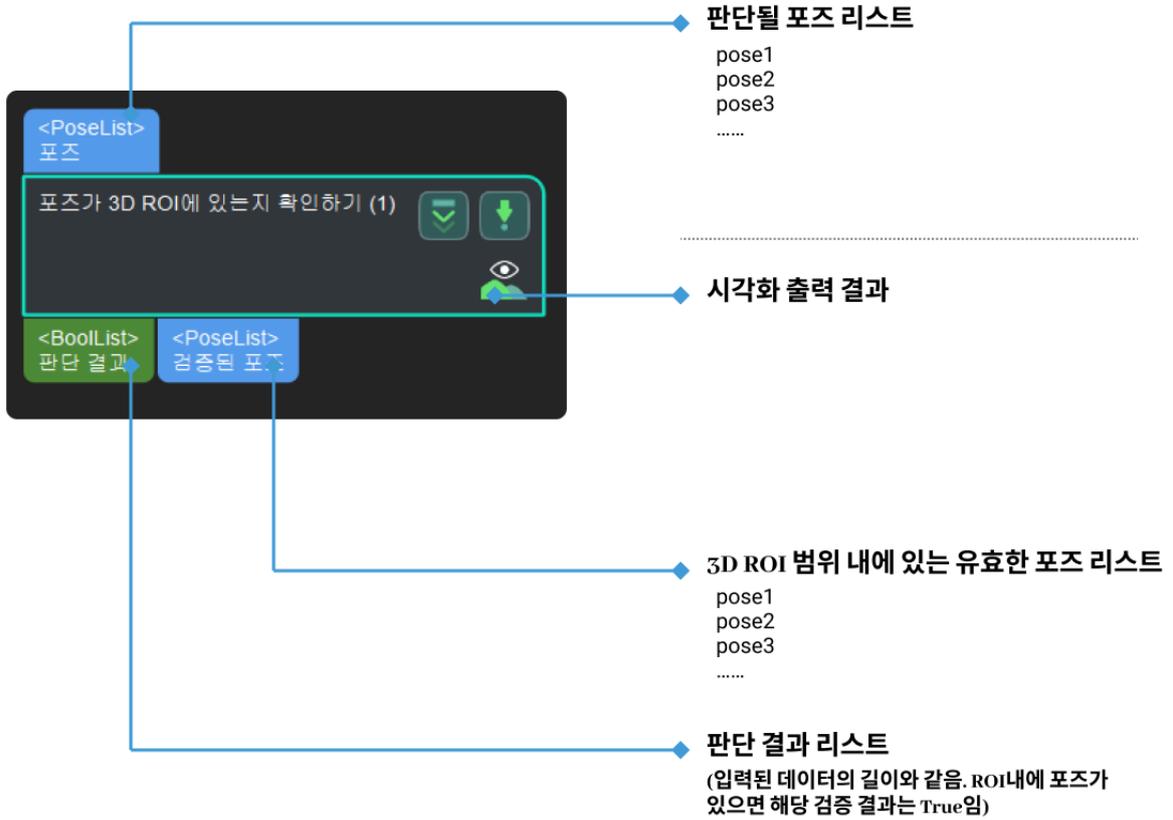
## 기능 설명

입력된 포즈 리스트가 설정된 3D ROI 범위 내에 있는지 판단하고 판단 결과 및 3D ROI 범위 내에 있는 포즈를 출력합니다.

### 응용 시나리오

일반적인 포즈 필터링 스텝으로 3D ROI를 통해 필터링하고 결과를 출력합니다.

### 입력 및 출력

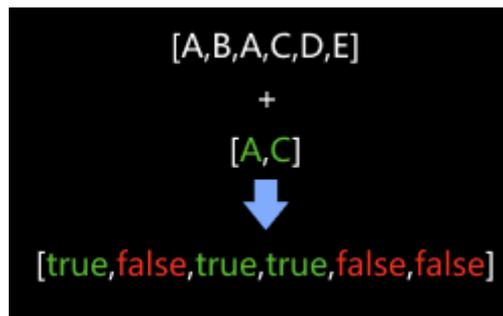


### 4.3.94. 필요한 레이블인지 판단하기

#### 기능 설명

이 스텝은 입력된 원시 레이블을 참조 레이블 리스트와 대조하여 부울 값 리스트 형식으로 결과를 출력합니다.

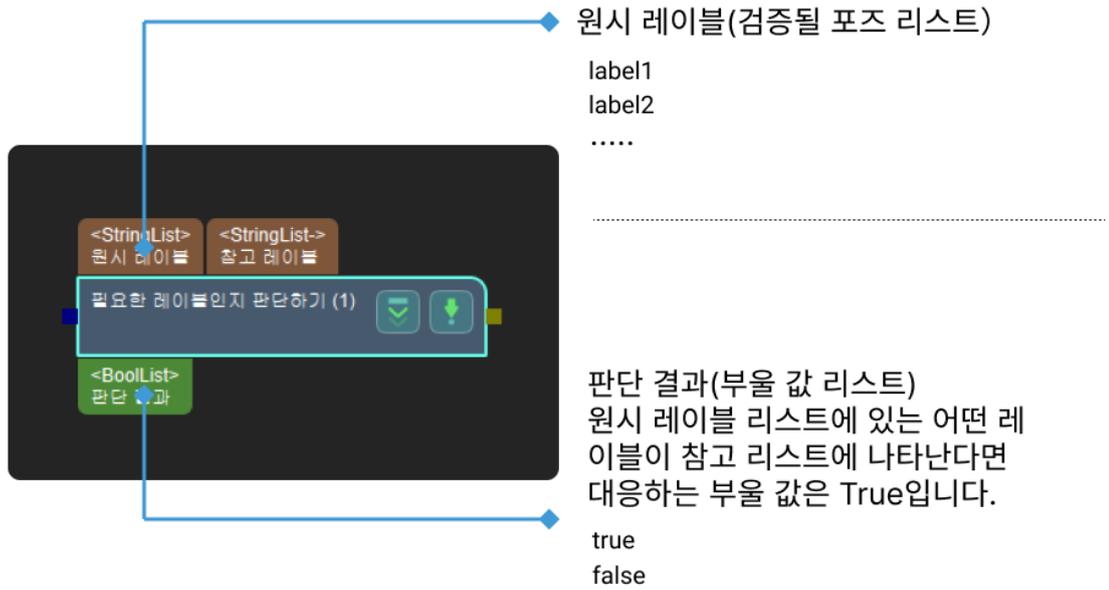
- 만약 입력된 원시 레이블이 참조 레이블 리스트에 나타난다면 해당 출력은 True입니다.
- 만약 입력된 원시 레이블이 참조 레이블 리스트에 나타나지 않는다면 해당 출력은 False입니다.



## 응용 시나리오

보통 필터링 스텝과 함께 사용하고 이 스텝에서 출력된 부울 값 리스트는 배열 필터링의 근거가 될 것입니다.

## 입력 및 출력



## 파라미터 설명

### 레이블 파일 명칭

파라미터 설명: 이 파라미터는 참조 레이블 파일을 설정하는 데 사용됩니다.

기본값: labelFilter.json

데이터 유형: 문자열 (특수 문자를 포함할 수 없음)



입력한 데이터가 요구 사항에 맞지 않으면 해당 데이터는 유효하지 않은 데이터로 간주되며, 성공적으로 입력되지 않습니다.

설정 방법:

1. [ **기준 레이블을 설정하기** ]를 클릭하여 JSON 편집기 페이지로 이동합니다.
2. **+**를 클릭하고, **아이템 추가**에서 참조 레이블 이름을 입력한 후 [ **확인** ]을 클릭하면 개별 참조 레이블을 성공적으로 추가합니다.



레이블 이름에 소수점 또는 특수 기호를 입력하면 레이블이 추가되지 않고 리스트 아래에 입력이 잘못되었음을 나타내는 프롬프트가 표시됩니다.

1. 여러 참조 레이블을 설정해야 하는 경우 단계 2를 반복합니다.
2. **폴더 경로**는 현재 프로젝트 경로입니다. 오른쪽의 [ **...** ]을 클릭하여 프로젝트 위치를 변경할 수

있으며, 필요에 따라 설정하십시오.

3. **JSON 파일 명칭**에 파일 명칭을 입력한 후 btn:[]을 클릭하여 레이블이 해당 파일에 저장되도록 합니다.
4. **[확인]**을 클릭하여 참조 레이블 파일 설정을 완료합니다.



이미 미리 설정된 참조 레이블 파일이 있는 경우 **JSON 파일 명칭** 옆의 [...]을 직접 클릭하여 파일을 선택하고, 참조 파일을 읽어올 수 있습니다.

### 4.3.95. 포인트 클라우드가 요구에 부합한지 검증하기



이 스텝의 내용은 유지 보수 중입니다. 이 스텝에 대한 자세한 정보가 필요한 경우 ([docs@mech-mind.net](mailto:docs@mech-mind.net))로 문의하십시오.

#### 기능 설명

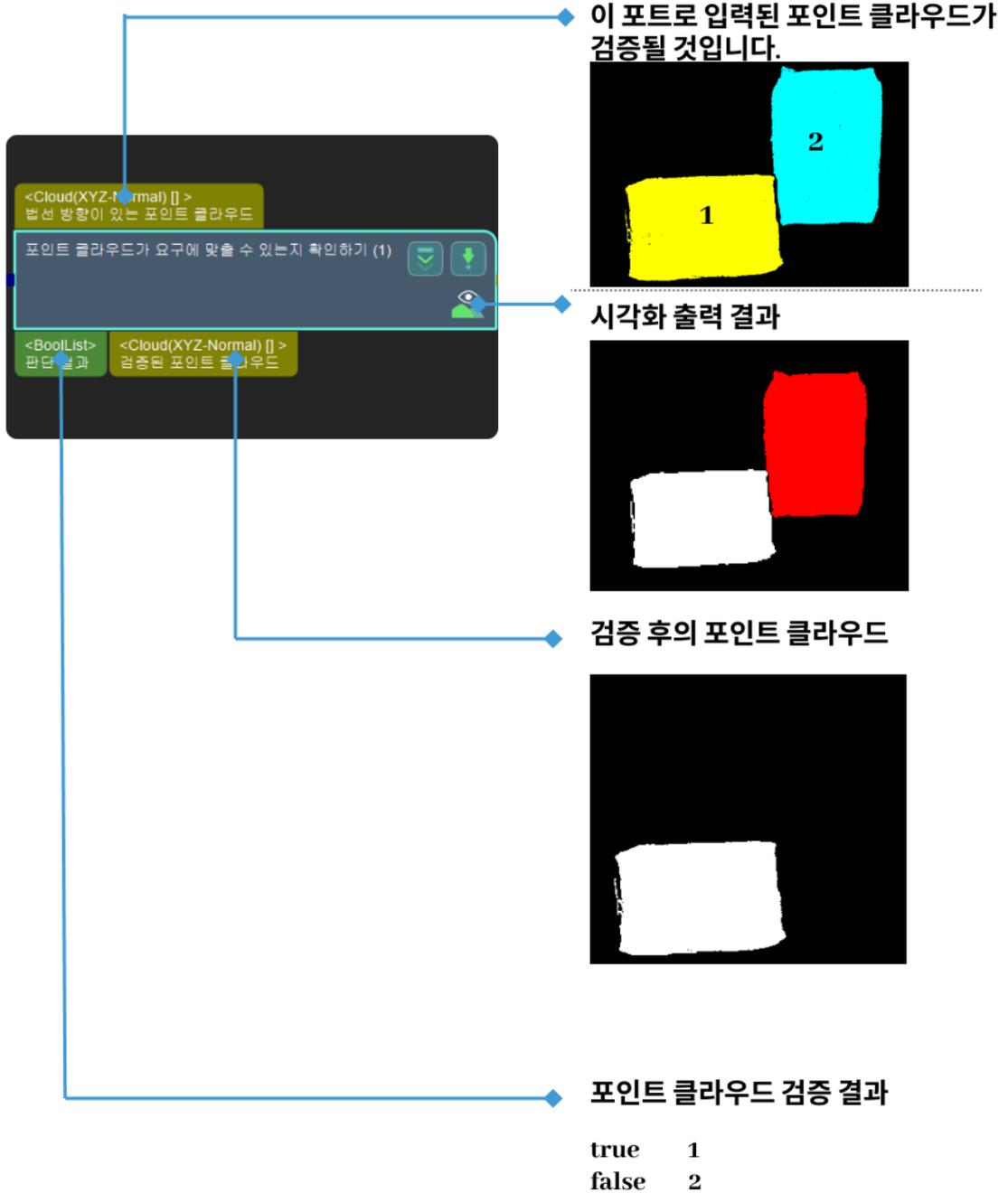
이 스텝은 설정된 규칙에 근거하여 포인트 클라우드가 합격한지를 판단합니다.

#### 응용 시나리오

이 스텝에서는 일반적으로 포인트 클라우드 포인트 수를 기반으로 포인트 클라우드가 원하는 것인지 여부를 판단합니다(파라미터 **필터링 유형**을 **CloudCapacityFilter**로 설정하는 것을 통해 실현합니다).

검증 후 **필터링** 스텝을 사용하여 검증 결과에 따라 요구 사항을 충족하지 않는 포인트 클라우드를 필터링할 수 있습니다.

#### 입력 및 출력



### 4.3.96. 기준 방향과의 각도에 근거하여 포즈 유효성을 판단하기



이 스텝의 내용은 유지 보수 중입니다. 이 스텝에 대한 자세한 정보가 필요한 경우 ([docs@mech-mind.net](mailto:docs@mech-mind.net))로 문의하십시오.

#### 기능 설명

포즈의 지정된 축과 기준 방향 사이의 각도를 계산하여 각도의 값이 설정된 최대 각도 차이보다 크지 않은 포즈는 보류되고 결과는 True로 판단되며 설정된 최대 각도 차이보다 큰 포즈는 제거되고 결과는 False로 판단됩니다.

## 응용 시나리오

이 스텝은 특별한 제한 없이 대부분 시나리오에서 사용될 수 있는 포즈 필터링 스텝입니다.

## 입력 및 출력



## 4.4. 핸드-아이 캘리브레이션 사용 가이드

이 부분은 로봇의 핸드-아이 캘리브레이션을 하는 방법에 대해 소개합니다.

다음 내용을 통해 본인에게 적합한 로봇 핸드-아이 캘리브레이션을 선택하십시오.

[로봇 핸드-아이 캘리브레이션 프로세스 선택](#)

다음 내용에 따라 로봇 핸드-아이 캘리브레이션 프로세스를 완료하십시오.

- [Eye to Hand 시나리오에서 자동 캘리브레이션 완료](#)
- [Eye in Hand 시나리오에서 자동 캘리브레이션 완료](#)
- [Eye to Hand 시나리오에서 수동 캘리브레이션 완료\(캘리브레이션 보드의 여러 임의 포즈\)](#)
- [Eye in Hand 시나리오에서 수동 캘리브레이션 완료\(캘리브레이션 보드의 여러 임의 포즈\)](#)
- [Eye to Hand 시나리오에서 수동 캘리브레이션 완료\(TCP 점점 터치\)](#)
- [Eye in Hand 시나리오에서 수동 캘리브레이션 완료\(TCP 점점 터치\)](#)
- [트러스 로봇의 핸드-아이 캘리브레이션 완료](#)
- [Eye to Hand 시나리오에서 핸드-아이 캘리브레이션 완료](#)

로봇 핸드-아이 캘리브레이션에 대해 알아보시려면 다음 내용을 살펴보십시오.

- [캘리브레이션 관련 개념](#)
- [캘리브레이션 원리](#)
- [트러스 로봇 캘리브레이션 설명](#)
- [캘리브레이션 결과 체크 및 분석](#)
- [자주 나타나는 문제와 해결 방법](#)

#### 4.4.1. 로봇 핸드-아이 캘리브레이션 프로세스 선택

핸드-아이 캘리브레이션은 \*카메라 좌표계와 로봇 좌표계 간의 해당 관계를 구축하는 과정\*을 말하며, 비전 시스템에 의해 제공된 물체 포즈를 로봇 좌표계 아래의 포즈로 전환하여 로봇이 피킹 작업을 완료하도록 가이드합니다. 핸드-아이 캘리브레이션 결과의 정확도는 로봇의 피킹 정확도에 영향을 미치는 중요한 요소 중 하나입니다.

Mech-Mind 비전 시스템을 구축한 후 로봇의 핸드-아이 캘리브레이션을 수행할 수 있습니다. 또한 다음과 같은 경우 로봇의 핸드-아이 캘리브레이션을 다시 실행해야 합니다.

- 카메라의 위치는 로봇 베이스(카메라 Eye to Hand 설치 방식) 또는 플랜지(카메라 Eye in Hand 설치 방식)를 기준으로 변경됩니다.
- 카메라를 교체합니다.
- 로봇을 교체합니다.

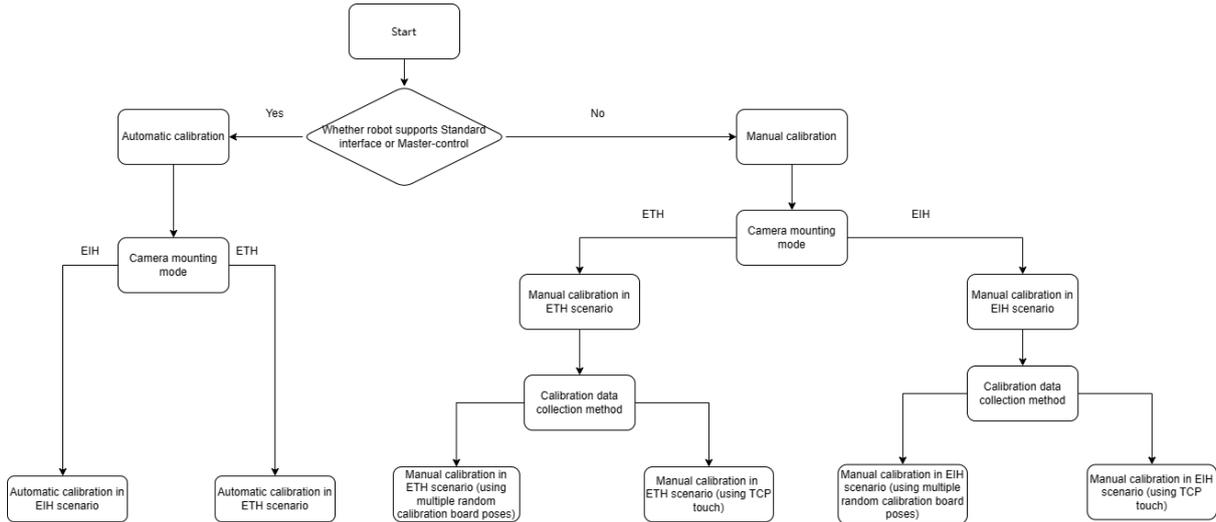
#### 적합한 캘리브레이션 프로세스 선택

로봇의 통신 방식, 카메라의 설치 방식, 캘리브레이션 방식 및 캘리브레이션 데이터 수집 방법은 모두 구체적인 캘리브레이션 프로세스에 영향을 미칩니다. Mech-Vision은 다양한 시나리오에 대해 표준화된 캘리브레이션 프로세스를 제공하여 카메라의 내부 및 외부 파라미터를 빠르고 정확하게 캘리브레이션할 수 있습니다.



로봇의 통신 방식, 카메라의 설치 방식, 캘리브레이션 방식 및 캘리브레이션 데이터 수집 방법은 [캘리브레이션 관련 개념](#) 내용을 참조하십시오.

적절한 캘리브레이션 프로세스를 선택하려면 이 부분의 내용을 참조하십시오.



1. 로봇이 표준 인터페이스 또는 마스터 컨트롤 통신을 지원하는지 확인합니다.
  - 지원되는 경우 자동 캘리브레이션을 선택하고 단계 2로 이동합니다.
  - 지원되는 않는 경우 자동 캘리브레이션을 선택하고 단계 3으로 이동합니다.
2. 자동 캘리브레이션 시의 카메라 설치 방식을 확인합니다.
  - 카메라 설치 방식이 ETH(Eye to Hand)인 경우 [ETH 시나리오에서의 자동 캘리브레이션 프로세스](#)를 선택합니다.
  - 카메라 설치 방식이 EIH(Eye in Hand)인 경우 [EIH 시나리오에서의 자동 캘리브레이션 프로세스](#)를 선택합니다.
3. 수동 캘리브레이션 시의 카메라 설치 방식을 확인합니다.
  - 카메라 설치 방식이 ETH인 경우 캘리브레이션 데이터 수집 방법을 확인합니다.
    - 캘리브레이션 데이터를 수집하는 방법이 캘리브레이션 보드의 여러 임의 포즈인 경우 [ETH 시나리오에서의 수동 캘리브레이션\(캘리브레이션 보드의 여러 임의 포즈\) 프로세스](#)를 선택합니다.
    - 캘리브레이션 데이터 수집 방법이 TCP 끝단점 터치인 경우 [ETH 시나리오에서의 수동 캘리브레이션\(TCP 끝단점 터치\) 프로세스](#)를 선택합니다.
  - 카메라 설치 방식이 EIH인 경우 캘리브레이션 데이터 수집 방법을 확인합니다.
    - 캘리브레이션 데이터 수집 방법이 캘리브레이션 보드의 여러 임의 포즈인 경우 [EIH 시나리오에서의 수동 캘리브레이션\(캘리브레이션 보드의 여러 임의 포즈\) 프로세스](#)를 선택합니다.
    - 캘리브레이션 데이터 수집 방법이 TCP 끝단점 터치인 경우 [EIH 시나리오에서의 수동 캘리브레이션\(TCP 끝단점 터치\) 프로세스](#)를 선택합니다.

프로젝트에서 듀얼 카메라를 사용하는 경우 [Eye to Eye 시나리오에서의 수동 캘리브레이션 프로세스](#)를 참조하여 캘리브레이션을 진행하십시오. 프로젝트에서 트러스 로봇을 사용하는 경우 [트러스 로봇의 핸드-아이 캘리브레이션 프로세스](#)를 참조하여 캘리브레이션을 진행하십시오.

#### 4.4.2. Eye to Hand 시나리오에서의 자동 캘리브레이션

이 부분에서는 ETH(Eye to Hand) 시나리오에서 자동 캘리브레이션을 완료하는 방법에 대해 설명합니다.

##### 캘리브레이션 사전 준비

핸드-아이 캘리브레이션을 수행하기 전에 다음 준비를 완료해야 합니다.

- Mech-Mind 비전 시스템 구축을 완료합니다.
- 캘리브레이션에 필요한 재료를 준비합니다.
- 캘리브레이션 보드 포인트 클라우드 이미징을 조정합니다.
- 로봇 및 인터페이스 구성을 완료합니다.

### 비전 시스템 전체 구축 완료

Mech-Mind 비전 시스템 구축을 완료하려면 비전 시스템 구축 부분 내용을 참조하십시오.

핸드-아이 캘리브레이션에는 Mech-Eye Viewer, Mech-Center, Mech-Vision 및 Mech-Viz 소프트웨어가 필요합니다. 위의 소프트웨어가 설치되고 최신 버전으로 업데이트되었는지 확인하십시오.

### 캘리브레이션에 필요한 재료를 준비

ETH 시나리오에서 자동 캘리브레이션을 하려면 캘리브레이션 보드를 사용해야 합니다.

다음 요구 사항에 따라 캘리브레이션 보드를 준비하십시오.

- 캘리브레이션 보드에서 도트가 잘 보이고 눈에 띄는 굵은 흔적이 없으며 캘리브레이션 보드가 뒤틀리지 않아야 합니다.
- ETH 시나리오에서 먼저 로봇의 말단 플랜지에 캘리브레이션 보드 커넥터를 설치한 다음 캘리브레이션 보드를 커넥터에 설치합니다. 캘리브레이션 보드가 단단히 설치되어 있고 캘리브레이션 보드가 로봇 말단의 XY 평면과 평행한지 확인하십시오.



로봇 플랜지에 분리 불가능한 그리퍼를 장착한 경우에 캘리브레이션 보드를 그리퍼에 직접 고정하면 됩니다.

또한 캘리브레이션 시작 전 로봇을 캘리브레이션 시작 포즈로 이동합니다. ETH 시나리오에서의 자동 캘리브레이션인 경우 로봇의 캘리브레이션 초기 위치는 카메라 시야의 최하부 중앙에 있습니다 (캘리브레이션 보드는 로봇에 따라 하층에서 상층으로 캘리브레이션함).

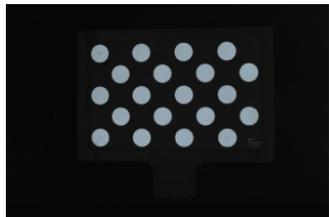
### 캘리브레이션 보드 포인트 클라우드 이미징을 조정하기

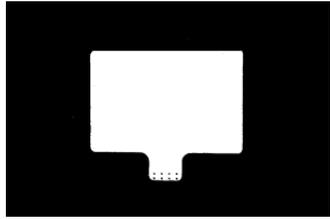
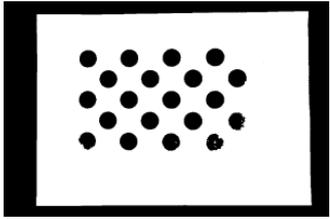
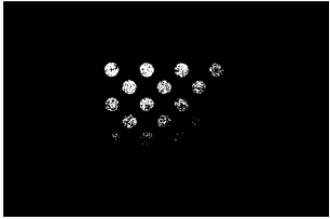
1. Mech-Eye Viewer 소프트웨어를 열고 카메라 파라미터를 조정합니다.
2. 2D 파라미터를 조정함으로써 2D 맵의 캘리브레이션 보드가 뚜렷하고 노출 과다나 노출 부족 등 현상이 없도록 합니다.
3. 3D 파라미터를 조정함으로써 캘리브레이션 보드의 원의 포인트 클라우드를 완전하게 하고 **포인트 클라우드 후처리에서 포인트 클라우드 평활화 및 노이즈 제거를 일반으로 변경하는 것이 좋으며** 포인트 클라우드 변동 범위를 줄입니다.



현장에 환경광이 복잡한 경우, 환경광이 2D 맵과 포인트 클라우드에 주는 영향을 줄이기 위해 광원을 차단하거나 보충하기 바랍니다.

1. 위의 단계를 완료하여 캘리브레이션 보드의 포인트 클라우드 이미징 품질이 요구 사항을 충족하는지 확인하십시오.

	정상	노출 과다	노출 부족
2D 맵			

	정상	노출 과다	노출 부족
포인트 클라우 드			

### 로봇 및 인터페이스 구성 완료

로봇이 표준 인터페이스를 사용해 비전 측과 통신하는 경우 로봇의 표준 인터페이스 통신 구성을 완료하십시오. 자세한 내용은 [표준 인터페이스 통신](#) 부분의 관련 내용을 참조하십시오.

로봇이 표준 인터페이스를 사용해 비전 측과 통신하는 경우 로봇의 마스터 컨트롤 통신 구성을 완료하십시오. 자세한 내용은 [마스터 컨트롤 통신](#) 부분의 해당 내용을 참조하십시오.

### 캘리브레이션 사전 구성

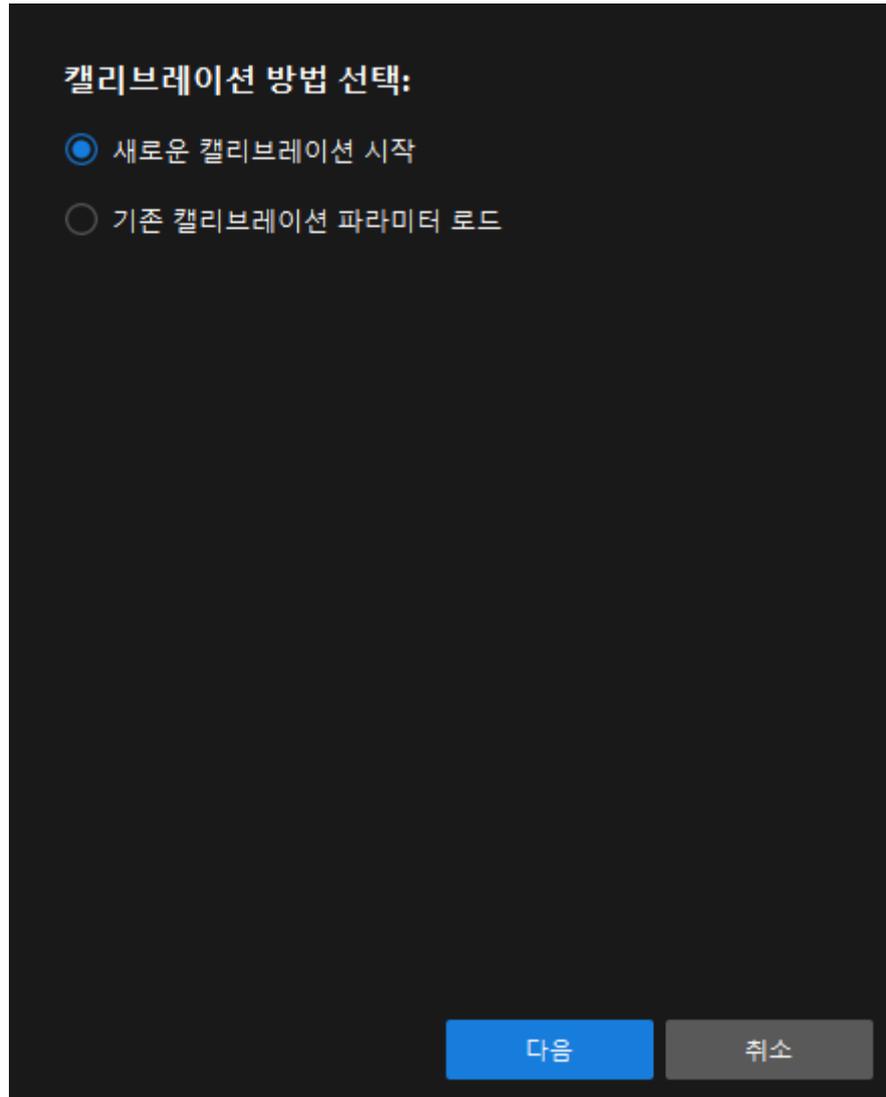
로봇이 표준 인터페이스를 사용해 비전 측과 통신하는 경우 [캘리브레이션 사전 구성\(표준 인터페이스\)](#)을 완료하십시오.

로봇이 마스터 컨트롤 방식을 사용해 비전 측과 통신하는 경우 [캘리브레이션 사전 구성\(마스터 컨트롤\)](#)을 완료하십시오.

#### 캘리브레이션 사전 구성(표준 인터페이스)

표준 인터페이스를 사용하여 로봇과 통신하는 경우 다음 단계를 수행하십시오.

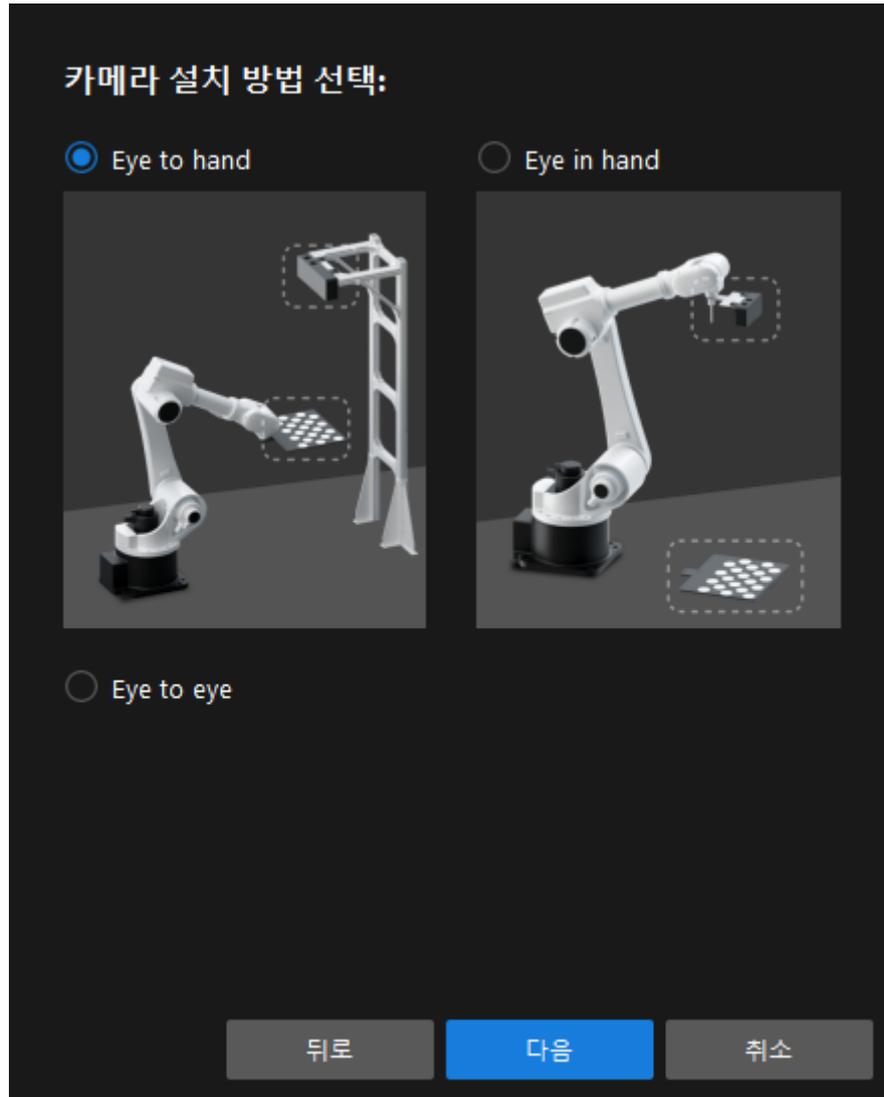
1. Mech-Vision 소프트웨어에서 톨 바의 [ [카메라 캘리브레이션\(표준\)](#) ] 버튼을 클릭합니다. [캘리브레이션 사전 구성](#) 창이 팝업됩니다.
2. [캘리브레이션 방법 선택](#) 창에서 [새로운 캘리브레이션 시작](#) 라디오 버튼을 선택한 후 [ [다음](#) ] 버튼을 클릭합니다.



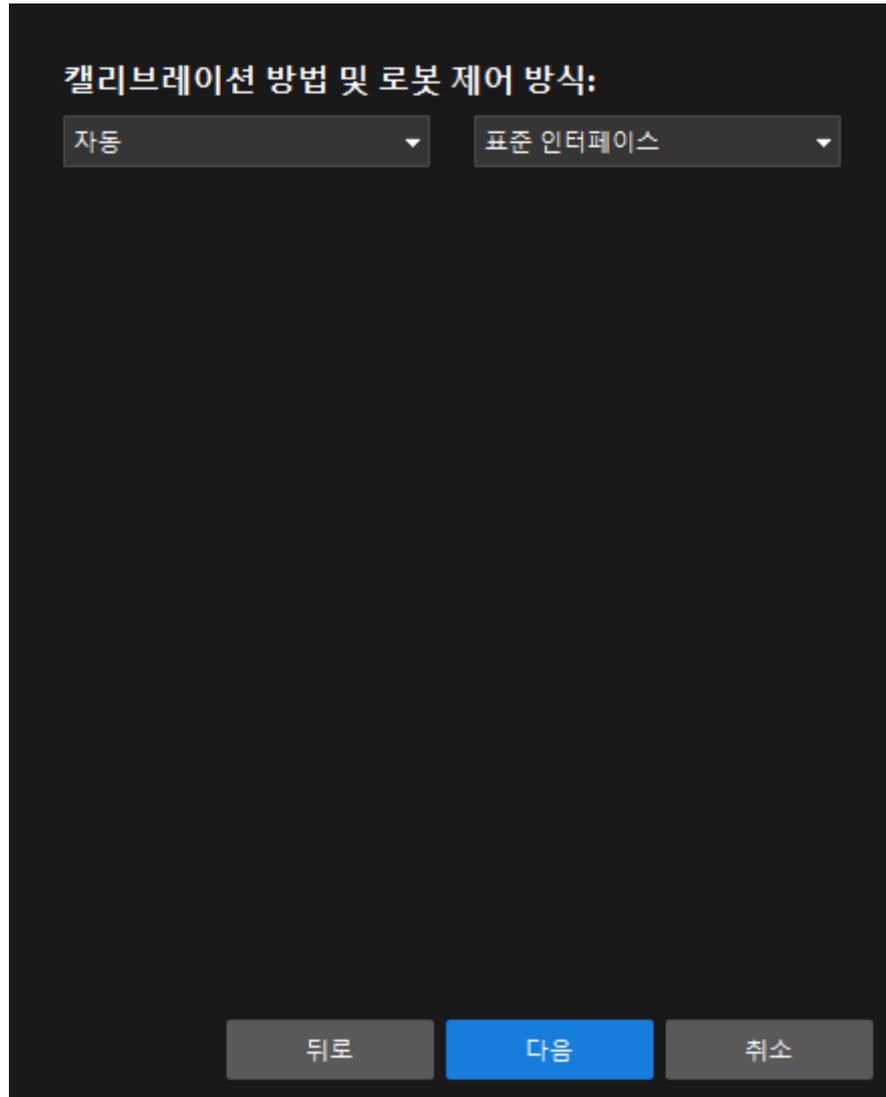
3. **캘리브레이션 작업 선택** 창의 드롭다운 목록 상자에서 **목록에 있는 로봇을 위한 핸드-아이 캘리브레이션**을 선택하고 **[로봇 모델 선택]** 버튼을 클릭하여 프로젝트에서 사용하는 로봇 모델을 선택한 후 **[다음]** 버튼을 클릭합니다.



4. 카메라 설치 방법 선택 창에서 **Eye to hand** 라디오 버튼을 선택한 후 [ 다음 ] 버튼을 클릭합니다.



5. 캘리브레이션 방법 및 로봇 제어 방식 창에서 자동 캘리브레이션 및 표준 인터페이스를 선택한 후 [ 다음 ] 버튼을 클릭합니다.



6. 통신 설정 창에서 파라미터 **프로토콜** 및 **호스트 IP 주소**를 설정한 다음 [ **인터페이스 서비스 시작** ] 버튼을 클릭합니다. 버튼이 **로봇이 연결되기를 기다리는 중...**으로 변경됩니다.



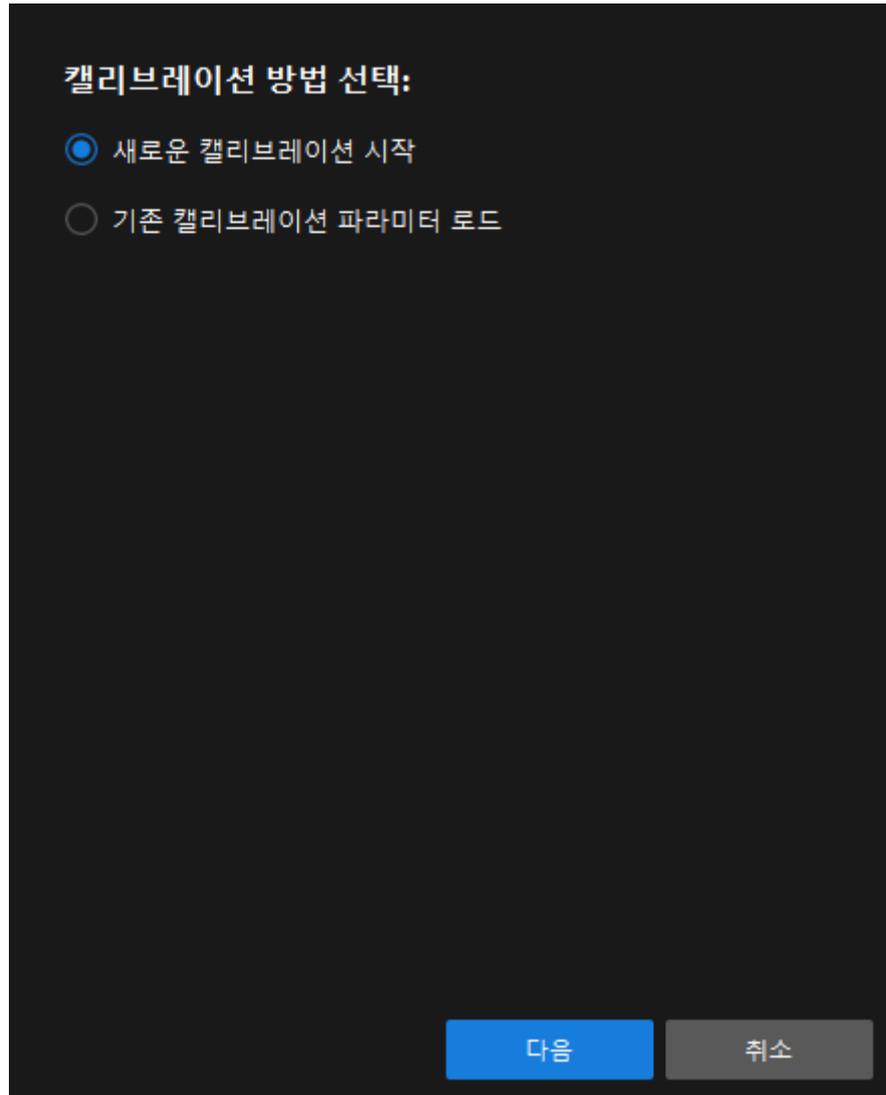
위 그림에서 **TCP 서버**와 **Hex(Little-endian)**은 ABB 로봇을 위한 **프로토콜** 구성의 예시일 뿐입니다. 기타 로봇의 프로토콜 설정이 다르기 때문에 소프트웨어가 자동으로 로봇의 기본 구성을 추가하므로 기본 구성을 유지하십시오.

7. 로봇의 티치 펜던트에서 **자동 캘리브레이션 절차를 선택하고 티치 펜던트를 통해 캘리브레이션 시작점을 설정한 다음 캘리브레이션 절차를 실행**합니다. 다른 로봇의 캘리브레이션 작업 프로세스에 대해서는 **표준 인터페이스 통신** 부분을 참조하십시오. 프로그램이 성공적으로 시작된 후 로그 바에서 "캘리브레이션 프로세스로 들어갑니다. Mech-Vision에서 캘리브레이션을 시작하십시오"라는 로그가 로그 열에 인쇄됩니다.
8. Mech-Vision 소프트웨어로 돌아가 **로봇 연결** 영역에 "연결됨" 상태 메시지가 표시될 때까지 기다린 다음 [ **캘리브레이션 시작** ] 버튼을 클릭합니다. **캘리브레이션(Eye to Hand)** 창이 나타납니다.

#### 캘리브레이션 사전 구성(마스터 컨트롤)

마스터 컨트롤을 사용하여 로봇과 통신하는 경우 다음 단계를 수행하십시오.

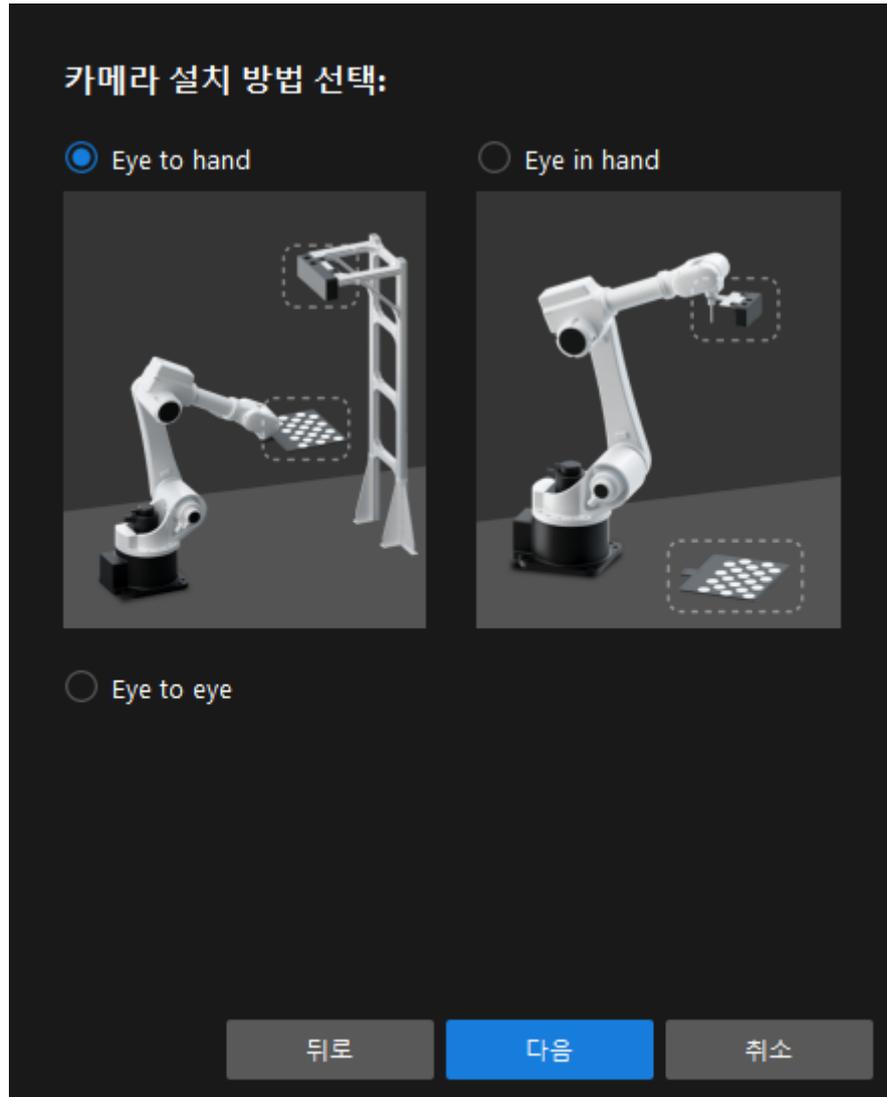
1. Mech-Vision 소프트웨어에서 톨 바의 [ **카메라 캘리브레이션(표준)** ] 버튼을 클릭합니다. **캘리브레이션 사전 구성** 창이 팝업됩니다.
2. **캘리브레이션 방법 선택** 창에서 **새로운 캘리브레이션 시작** 라디오 버튼을 선택한 후 [ **다음** ] 버튼을 클릭합니다.



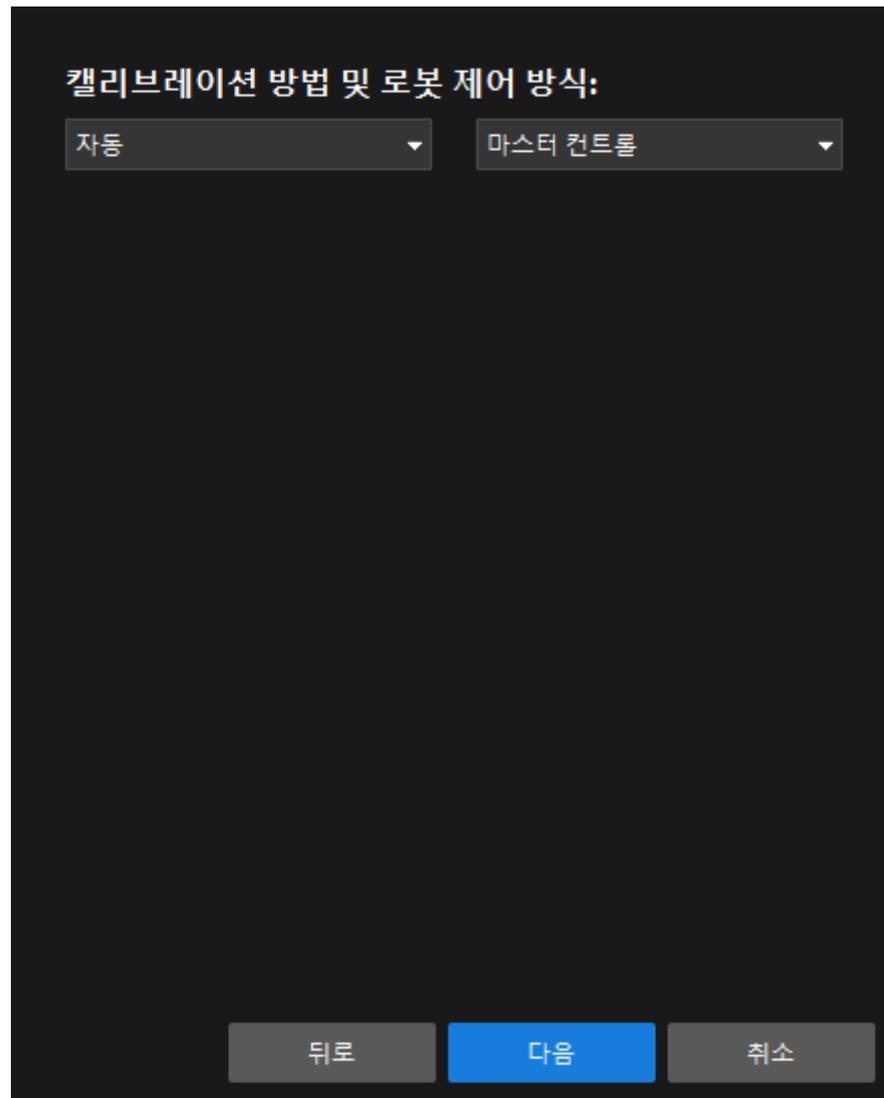
3. **캘리브레이션 작업 선택** 창의 드롭다운 목록 상자에서 **목록에 있는 로봇을 위한 핸드-아이 캘리브레이션**을 선택하고 **[로봇 모델 선택]** 버튼을 클릭하여 프로젝트에서 사용하는 로봇 모델을 선택한 후 **[다음]** 버튼을 클릭합니다.



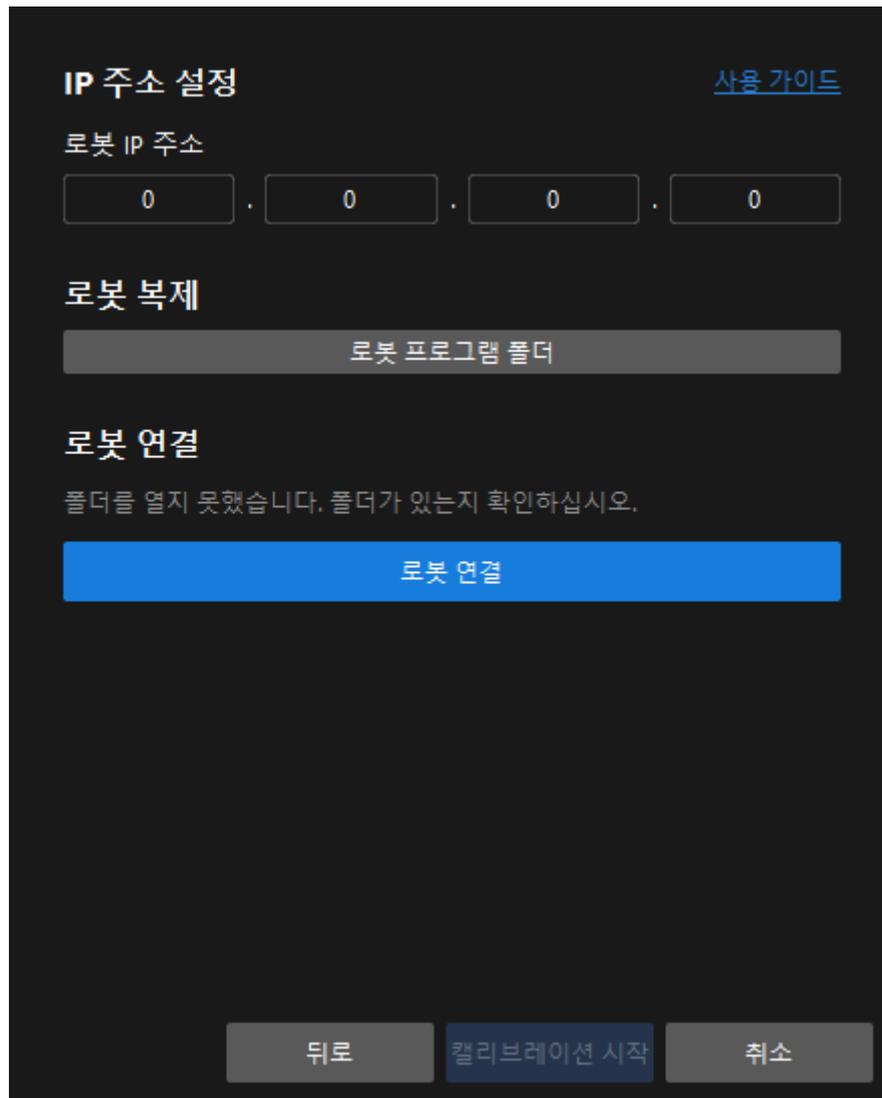
4. 카메라 설치 방법 선택 창에서 **Eye to hand** 라디오 버튼을 선택한 후 [ 다음 ] 버튼을 클릭합니다.



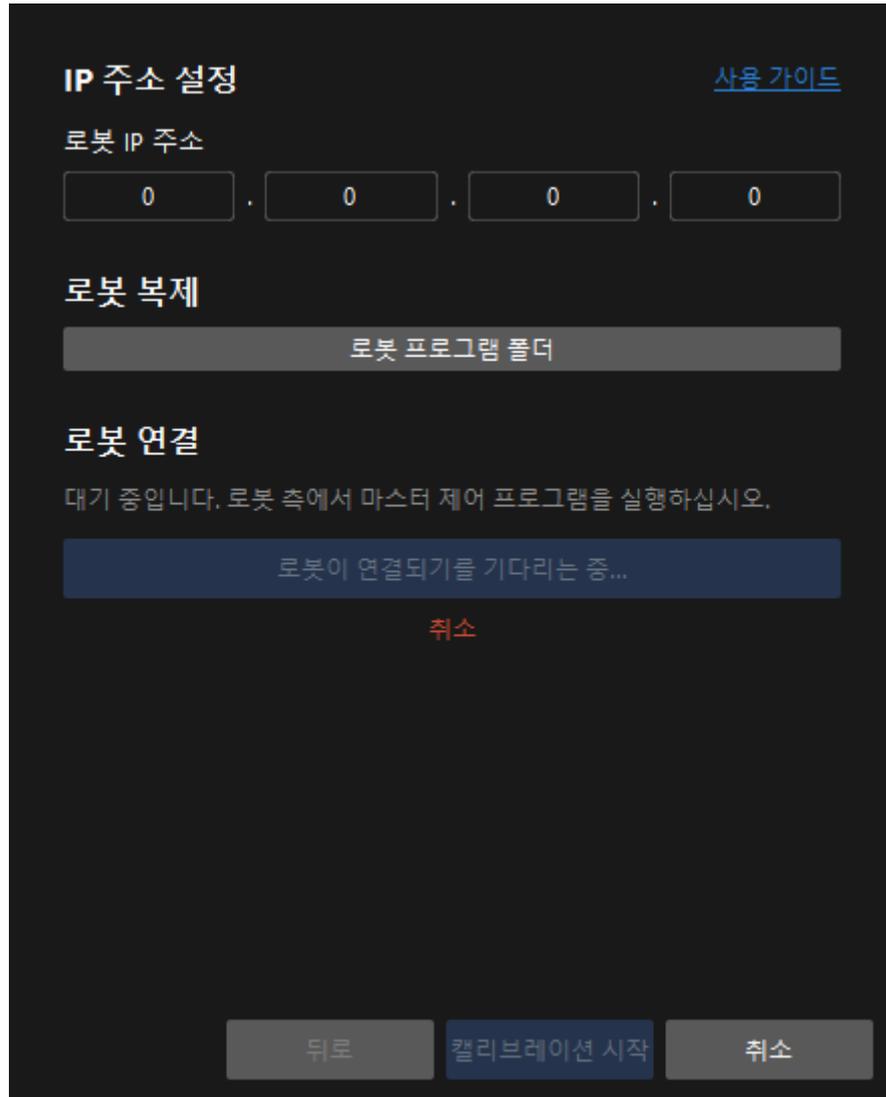
5. 캘리브레이션 방법 및 로봇 제어 방식 창에서 자동 캘리브레이션 및 마스터 컨트롤을 선택한 후 [ 다음 ] 버튼을 클릭합니다.



6. 새 창에서 **로봇 IP 주소** 파라미터를 설정합니다.



7. 로봇의 티치 펜던트에서 **마스터 컨트롤 프로그램을 선택하고 실행**합니다. 다른 로봇의 사용 방법은 **마스터 컨트롤 통신** 부분을 참조하십시오.
8. Mech-Vision 소프트웨어로 돌아가서 **로봇 연결** 영역에서 [ **로봇 연결** ] 버튼을 클릭합니다. 버튼이 **로봇이 연결되기를 기다리는 중...**으로 변경됩니다.



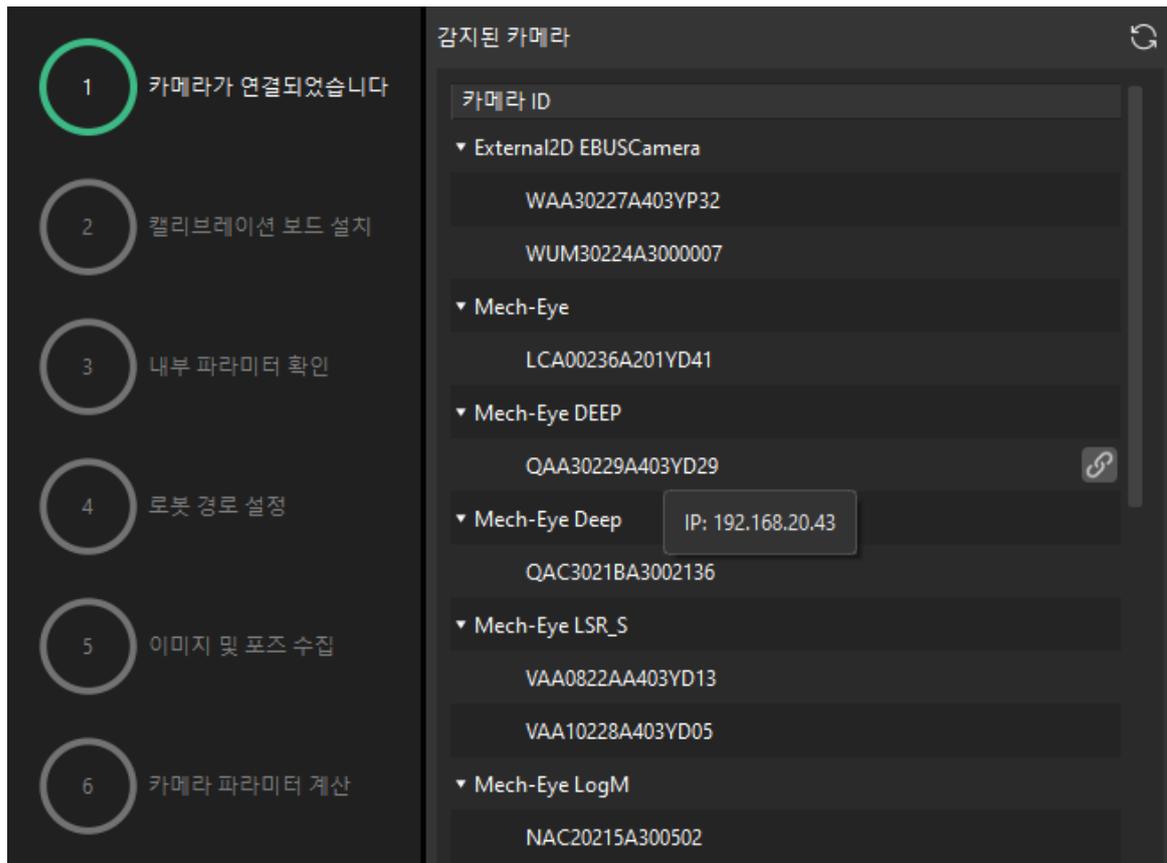
9. **로봇 연결** 영역에 "연결됨" 상태 메시지가 표시될 때까지 기다린 다음 [ **캘리브레이션 시작** ] 버튼을 클릭합니다. **캘리브레이션(Eye to Hand)** 창이 나타납니다.

여기까지 캘리브레이션 사전 구성이 완료되었으며 정식 캘리브레이션 프로세스로 들어갑니다.

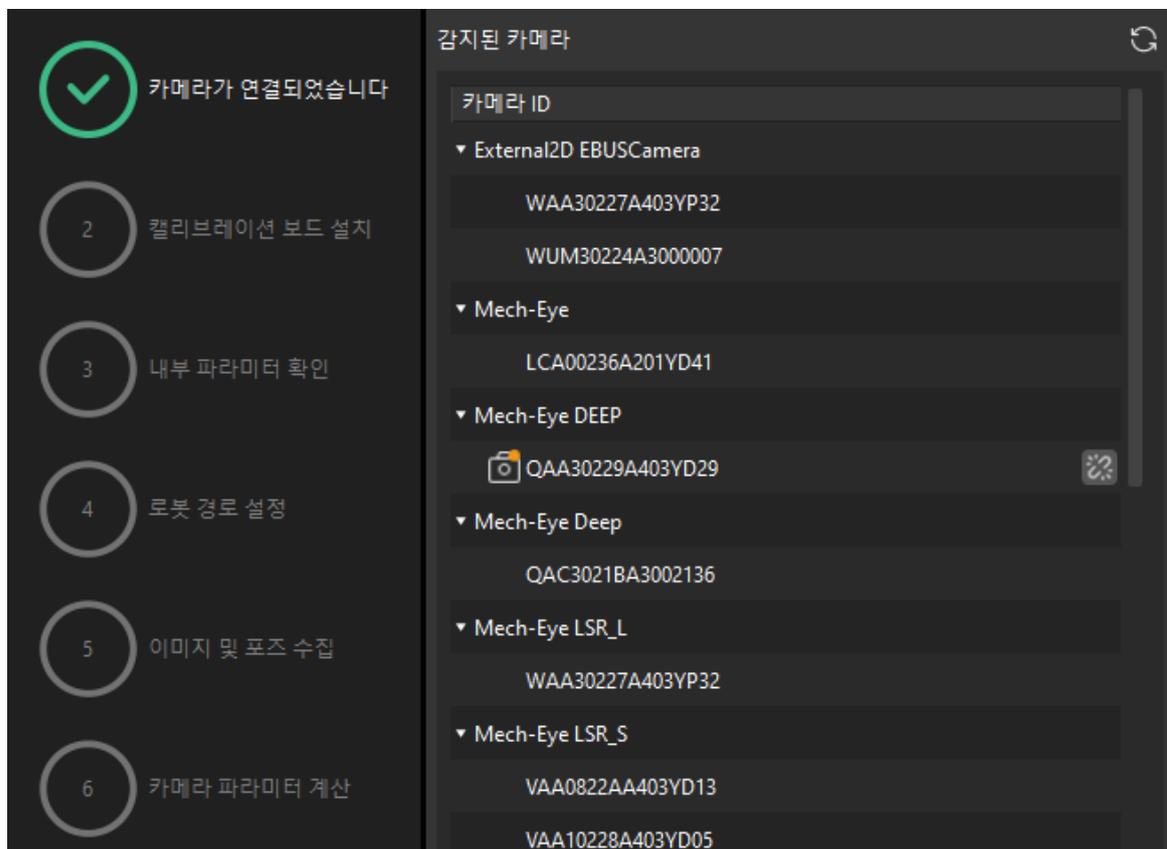
### 캘리브레이션 프로세스

#### 카메라 연결

1. **카메라 연결** 단계에서 **감지된 카메라** 리스트에서 연결할 카메라를 찾은 다음  을 클릭하면 카메라가 연결되며 또는 직접 연결할 카메라를 더블클릭합니다.



2. 카메라를 연결한 후 [연속 캡처] 또는 [한번 캡처] 버튼을 선택할 수 있습니다.



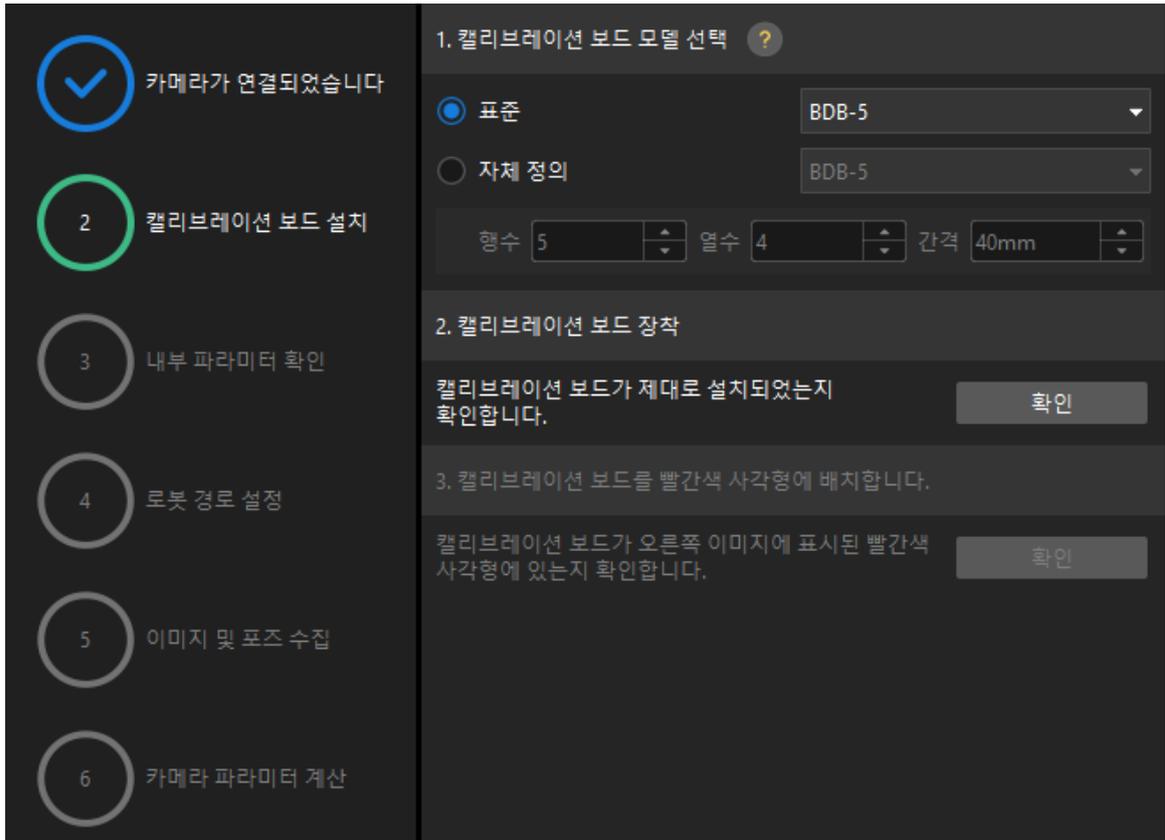
3. 이미지 뷰어에서 카메라가 캡처한 2D 맵과 덤스 맵이 캘리브레이션 요구 사항을 충족하는지 확인한 다음 하단 표시줄에서 [ 다음 ] 버튼을 클릭합니다.



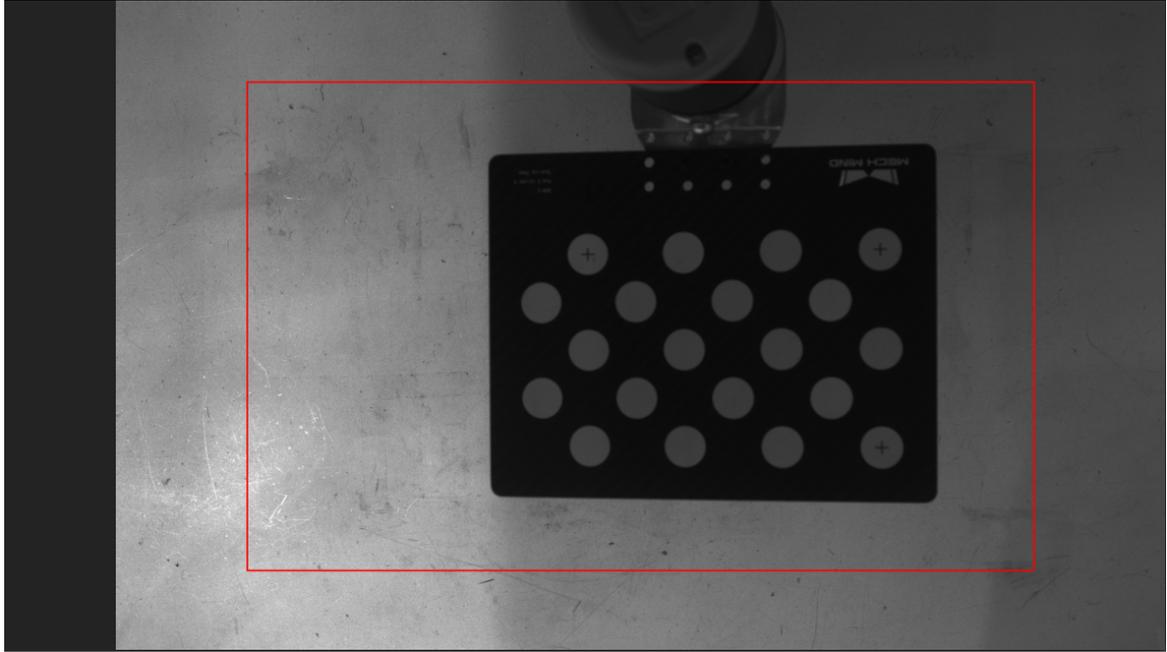
캡처한 이미지가 캘리브레이션 요구 사항을 충족하지 않으면 Mech-Eye Viewer 소프트웨어를 열어 카메라의 2D 및 3D 노출 파라미터를 조정하고 다시 캡처해야 합니다.

#### 캘리브레이션 보드 설치

1. 캘리브레이션 보드 설치 단계의 1. 캘리브레이션 보드 유형 선택 영역에서 표준 라디오 버튼을 선택한 후 캘리브레이션 보드 모델 레이블에 따라 해당 캘리브레이션 보드 유형을 선택합니다.



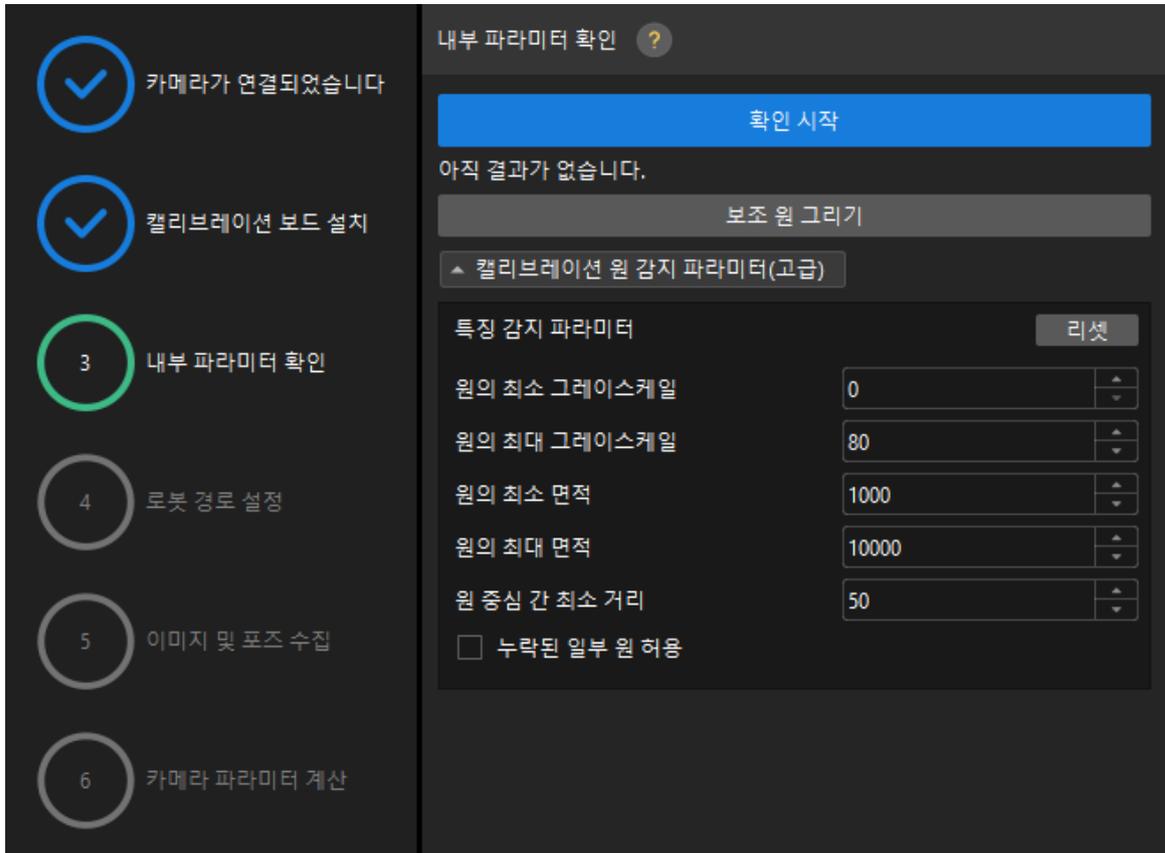
2. 캘리브레이션 보드가 작업 평면 중앙에 놓여 있는지 확인한 후 2. 캘리브레이션 보드 설치 영역에서 [ 확인 ] 버튼을 클릭합니다.
3. 캘리브레이션 보드가 카메라 시야의 중앙(빨간색 사각형 내)에 있는지 확인한 다음 3. 캘리브레이션 보드를 빨간색 사각형에 배치 영역에서 [ 확인 ] 버튼을 클릭합니다.



4. 캘리브레이션 보드 관련 작업을 모두 완료한 후 하단 표시줄에서 [ 다음 ] 버튼을 클릭합니다.

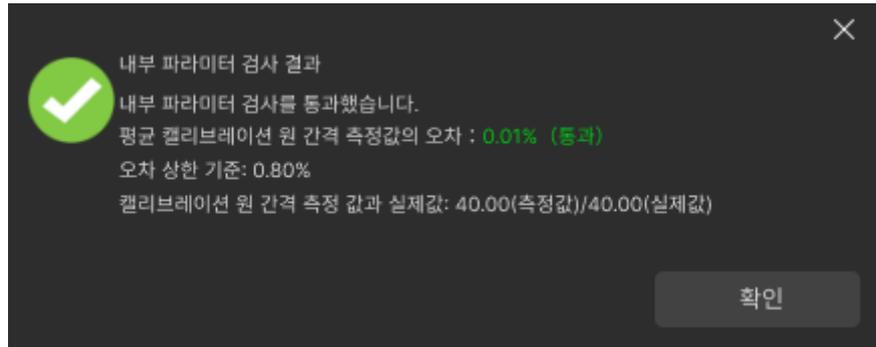
#### 카메라 내부 파라미터를 검사하기

1. 내부 파라미터 검사 단계에서 [ 검사 시작 ] 버튼을 클릭합니다.



2. 카메라의 내부 파라미터 검사 결과를 확인합니다.

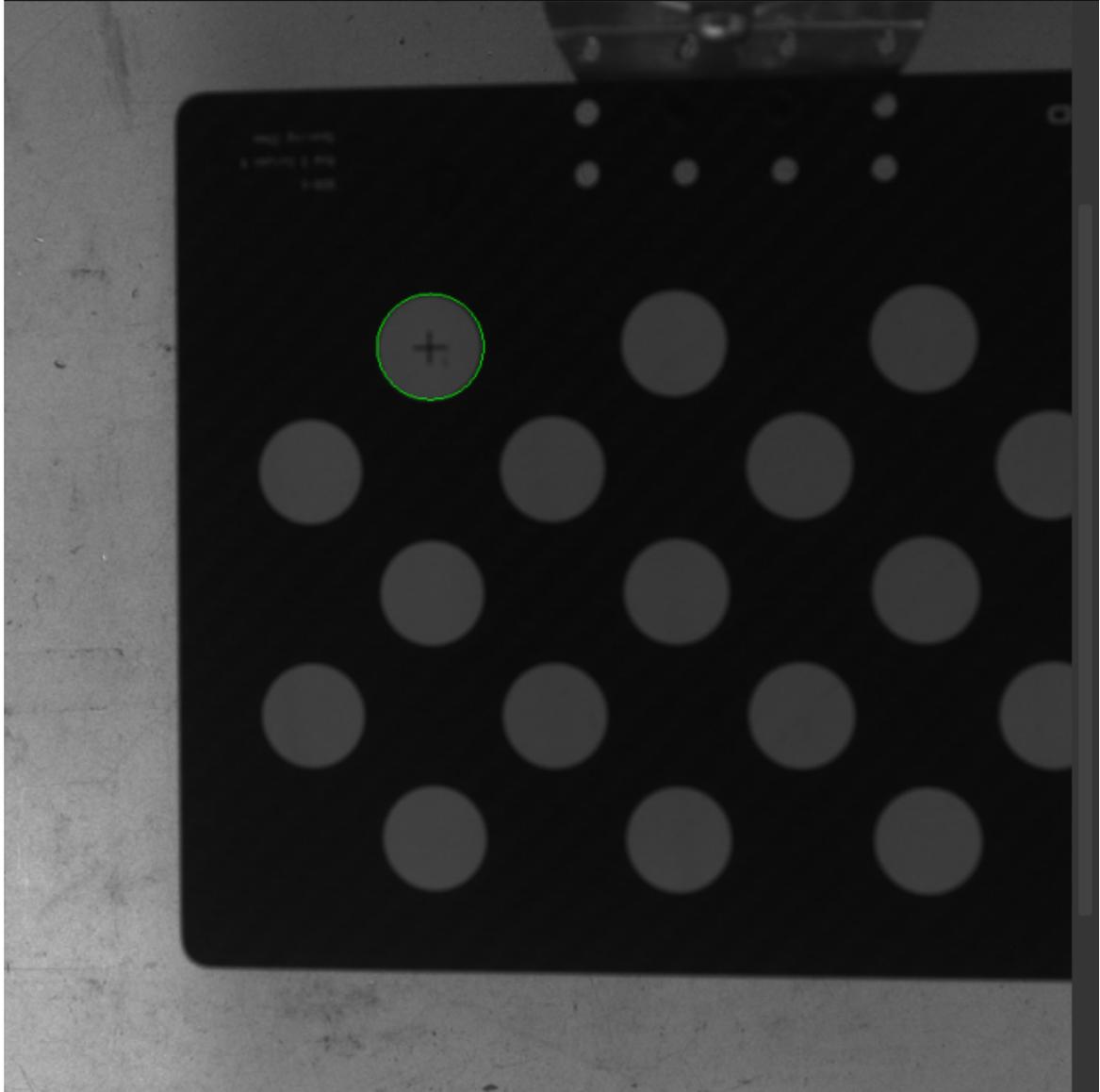
- 카메라의 내부 파라미터 검사에 성공하면 [ 확인 ]을 클릭한 후 [ 다음 ] 버튼을 클릭하십시오.



- 카메라의 내부 파라미터 검사에 실패하면 보조원을 그리거나 캘리브레이션 원을 수동으로 편집하여 파라미터를 감지한 다음 [ 다시 검사 ] 버튼을 클릭하십시오.

#### 보조원을 그리기

1. 보조원을 그리기로 선택한 경우 [ 보조원 그리기 ] 버튼을 클릭하십시오.
2. 오른쪽 이미지 뷰어 패널에서 마우스 오른쪽으로 캘리브레이션 보드 이미지를 클릭하고 해당 창 확인란을 언체크 한 다음 **Ctrl** 키를 누른 상태에서 롤러를 드래그하여 이미지를 적절한 크기로 조정하십시오.
3. 마우스 포인터를 캘리브레이션 원의 중간 줄 지점으로 이동하고 왼쪽 마우스 버튼을 누른 다음 보조원에 캘리브레이션 원을 완전히 포함시킨 다음 해제하십시오.



4. [ 다시 확인 ] 버튼을 클릭하고 카메라 내부 파라미터 검사가 통과되었는지 확인하십시오.

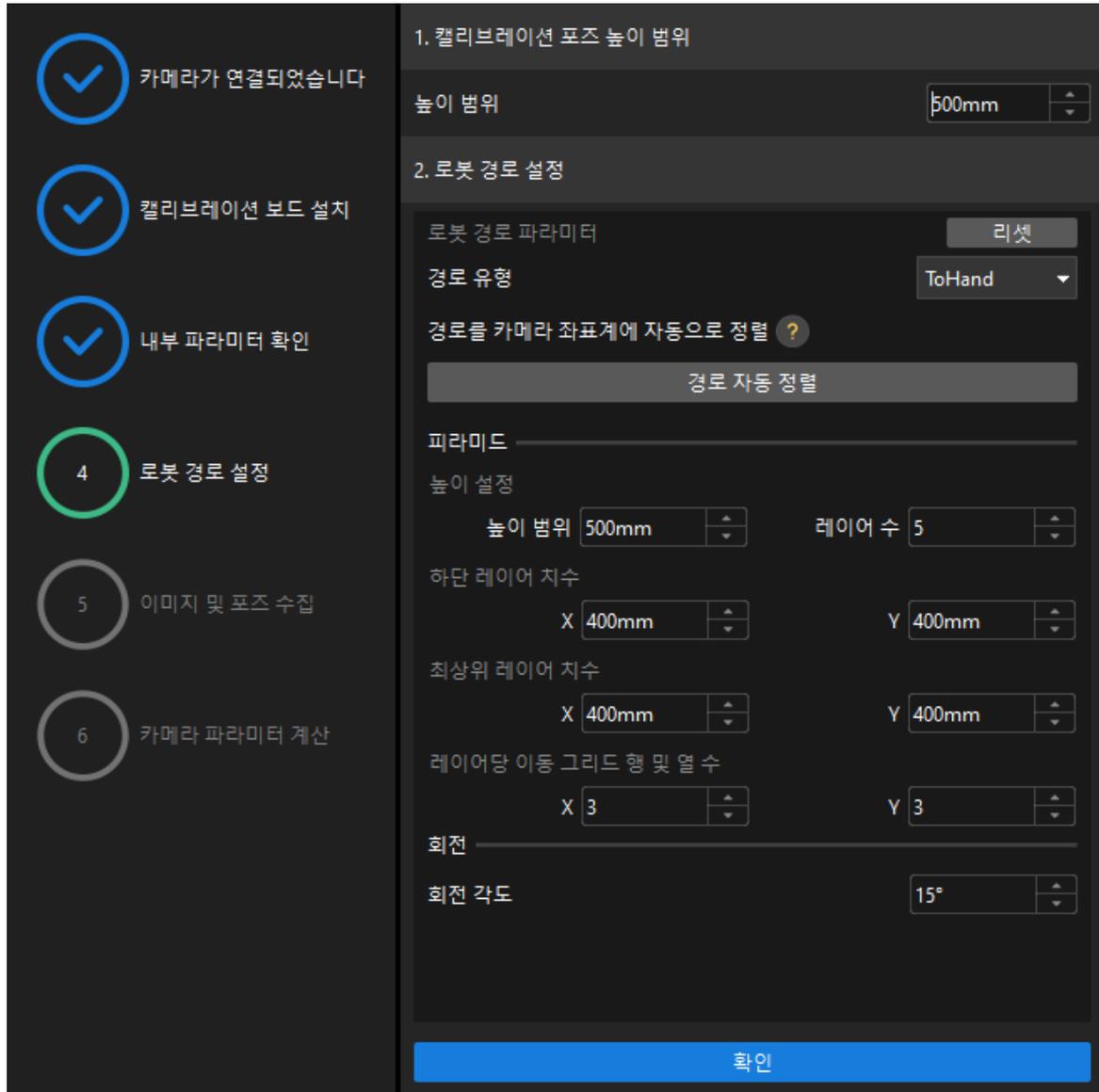
#### 감지 파라미터 수동으로 조정

감지 파라미터를 수동으로 조정할 경우 [ 캘리브레이션 원 감지 파라미터(고급) ]을 클릭하여 감지 파라미터의 값을 변경하면 됩니다.

캘리브레이션 원이 여전히 감지되지 않은 경우 실제 작업 조건에 따라 카메라 관련 파라미터를 조정해야 하며 카메라 파라미터는 [파라미터 조정](#) 내용을 참조하십시오.

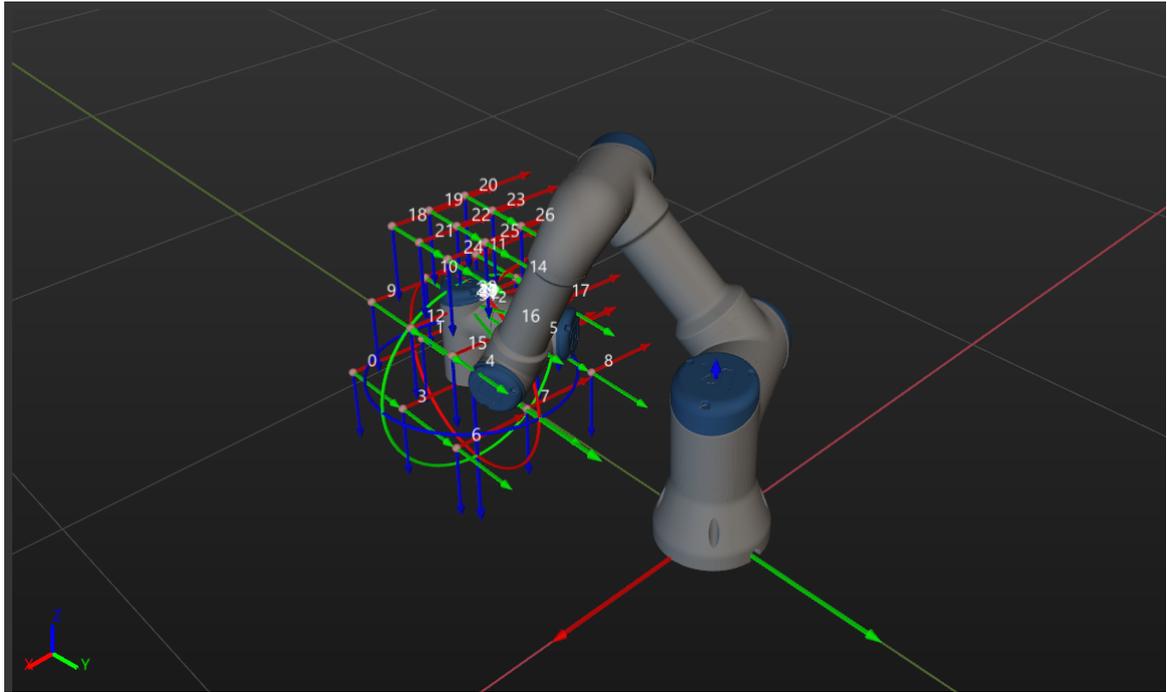
#### 로봇 경로 설정

1. 로봇 경로 설정 단계에서 카메라 뎁스 방향 캘리브레이션 범위 파라미터를 설정합니다.



카메라 뎁스 방향 캘리브레이션 범위 파라미터는 카메라의 권장 작동 거리 범위와 로봇의 작업 공간에 따라 설정해야 합니다.

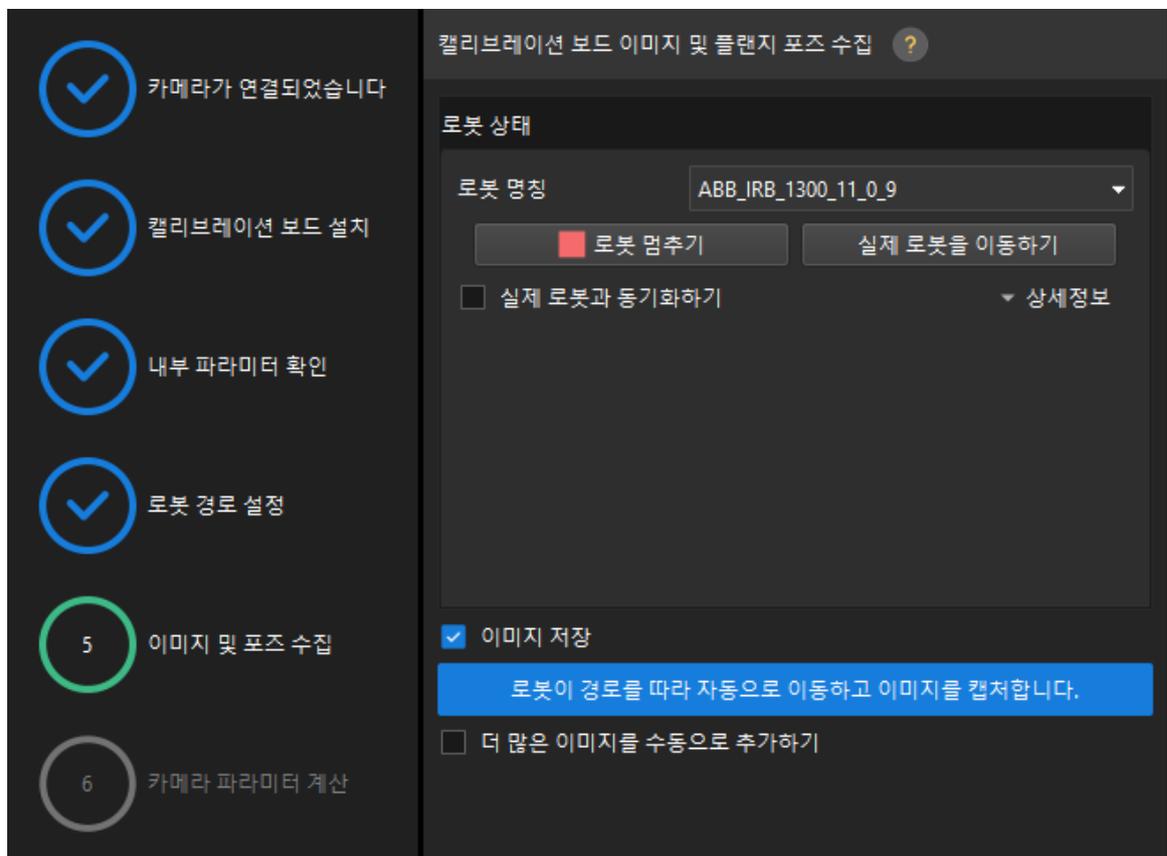
2. 카메라의 z 축과 세계 좌표계의 z 축이 평행하지 않은 경우 [ 경로 자동 정렬 ] 버튼을 클릭하십시오.
3. 필요에 따라 **경로 유형** 파라미터를 ToHand로 설정하고 피라미드 파라미터 **높이 범위**, **층 수**, **하단 크기 X/Y**, **상단 크기 X/Y** 및 **행당 이동 그리드의 행 및 열 수**, 회전 파라미터 **회전 각도**를 지정한 다음 [ **확인** ] 버튼을 클릭합니다.
4. 우측 시나리오 뷰 패널에서 각 웨이포인트의 위치가 합리적이고 주변 환경과 충돌하지 않는지 확인합니다.



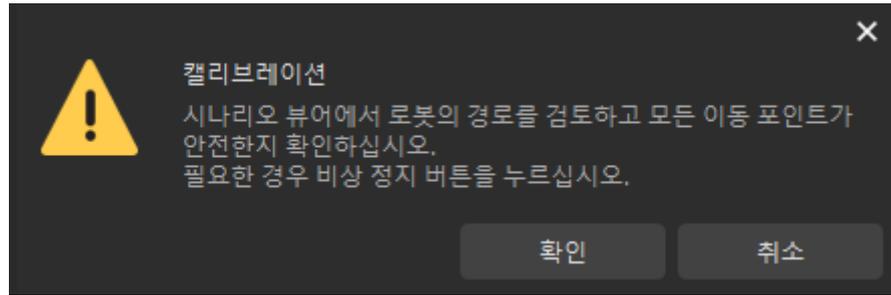
5. 하단에서 [ 다음 ] 버튼을 클릭하십시오.

#### 이미지 및 포즈 가져오기

1. 이미지 및 포즈 가져오기 단계에서 이미지를 저장하기를 선택한 다음 [ 로봇이 경로를 따라 움직이고 이미지를 캡처하게 하기 ] 버튼을 클릭합니다.



2. 로봇 이동 안전 수칙을 주의 깊게 읽은 다음 [확인] 버튼을 클릭합니다.

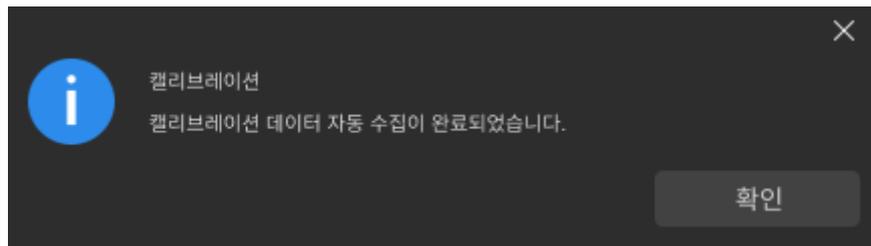


3. 설정된 경로에 따라 로봇이 움직이고 카메라가 각 웨이포인트의 이미지 캡처를 완료할 때까지 기다립니다. 오른쪽의 **이미지 뷰어** 패널에 캡처한 이미지가 표시됩니다.



- 로봇이 움직일 때 로봇이 사람을 다치게 하지 않도록 로봇 작업 구역에서 멀리 떨어져 계십시오.
- 로봇 멈추기를 클릭하여 캘리브레이션을 종료할 수 있지만 로봇은 즉시 중지되지 않으며 현재 웨이포인트를 이동 한 후에 중지됩니다. 비상 사태가 발생하면 로봇 터치 펜던트에서 비상 정지 버튼을 누르십시오 (비상 정지 버튼을 누른 후 로봇을 다시 연결해야 함).

4. 이미지를 캡처 한 후 팝업창에서 [확인] 버튼을 클릭하십시오.



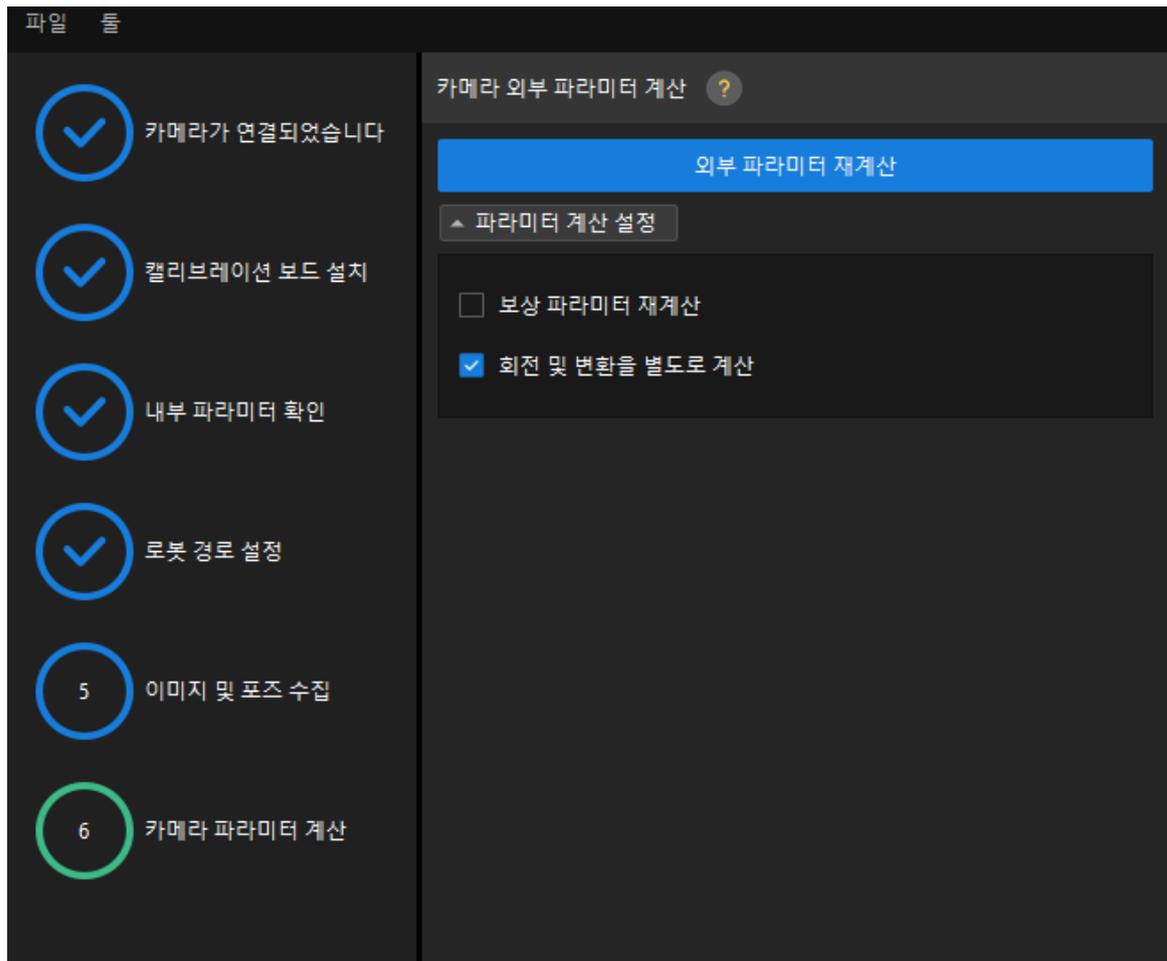
5. 현재 인식한 캘리브레이션 포인트가 요구 사항을 충족하는지 확인한 다음 하단 열의 [다음] 버튼을 클릭하십시오.



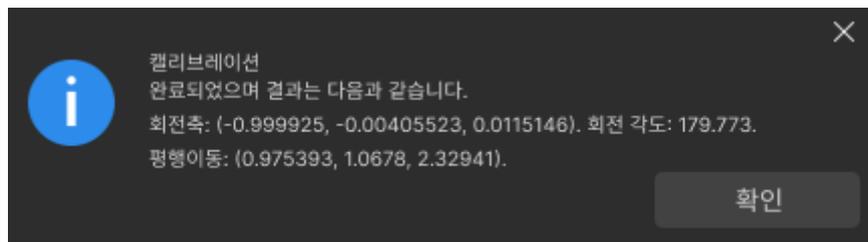
만족하지 않는 경우, 로봇을 수동으로 이동한 후 (로봇 터치 펜던트로 이동하거나 Mech-Viz를 통해 이동하는 것을 선택할 수 있음), [더 많은 이미지를 수동으로 추가하기]를 선택하고 [이미지를 추가하고 플랜지 포즈를 기록하기]를 클릭하여 캘리브레이션 보드 이미지를 추가하고 로봇의 플랜지 포즈를 입력합니다.

#### 카메라 파라미터를 계산하기

1. 카메라 파라미터를 계산하기 단계에서 **회전 및 변환을 별도로 계산** 파라미터를 선택한 다음 [카메라 외부 파라미터 계산] 버튼을 클릭하십시오.



2. 팝업한 캘리브레이션 성공 다이얼로그 박스에서 **확인** 버튼을 클릭합니다.



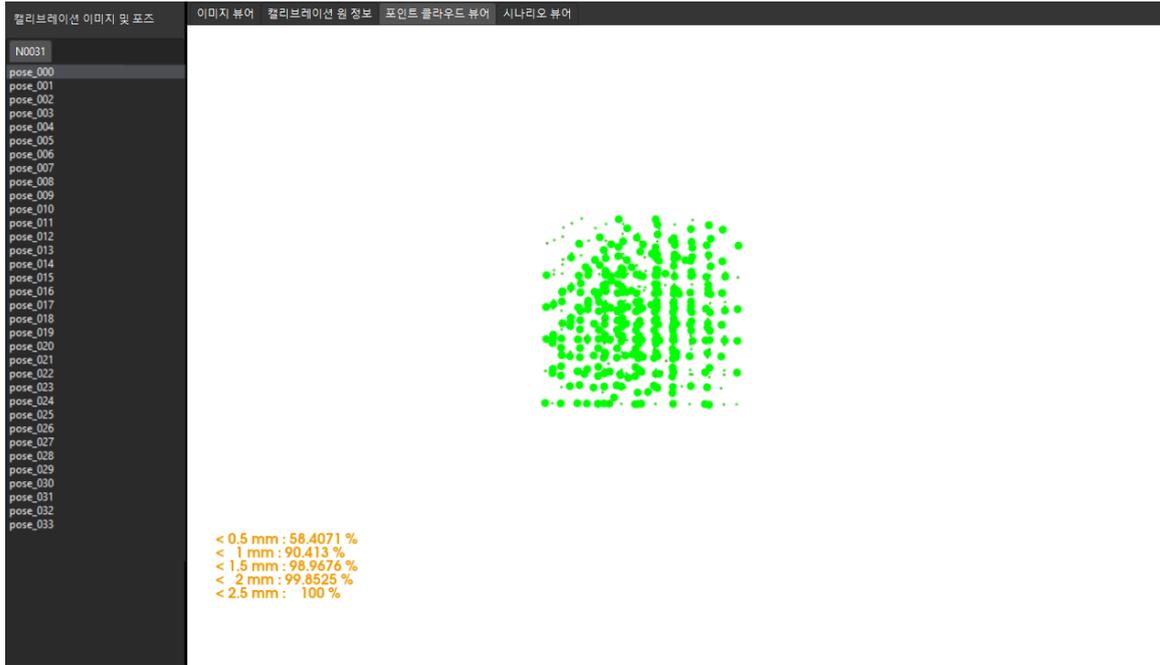
1. 오른쪽 **포인트 클라우드 뷰어** 패널에서 캘리브레이션 오차 포인트 클라우드를 확인합니다.



오차 포인트 클라우드는 각 캘리브레이션 포즈의 캘리브레이션 보드 원의 진릿값과 계산된 값 사이의 편차를 표시하는 데 사용됩니다.

2. 캘리브레이션 정확도가 예상 요구 사항을 충족하는지 확인합니다.

캘리브레이션의 정확도를 결정하기 위해 100%를 차지하는 오차 값을 찾으십시오. 예를 들어, 아래 그림의 정확도는 2.5mm 미만입니다.



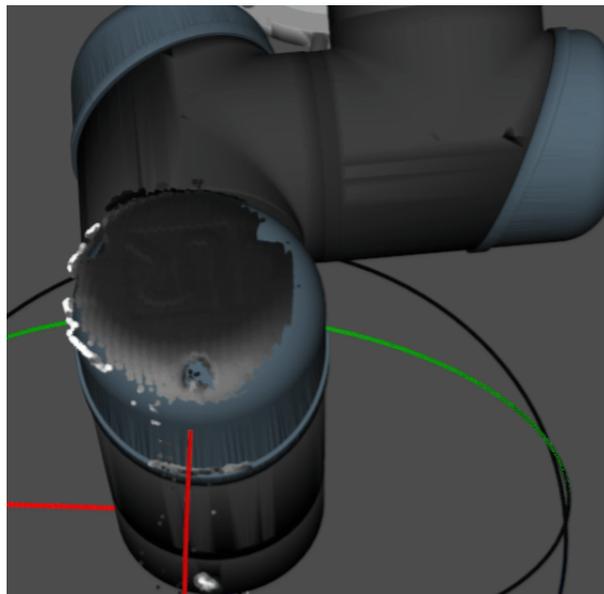
캘리브레이션 정확도를 개선해야 하는 경우 [캘리브레이션 결과 분석](#) 내용을 참조하십시오.

### 캘리브레이션 결과 검증 및 저장

카메라 외부 파라미터를 계산한 후 시나리오 뷰어에서 로봇 포인트 클라우드와 로봇 모델의 매칭 정도를 보고 캘리브레이션 결과를 대략적으로 판단할 수 있습니다. 구체적인 작업은 다음과 같습니다.

1. 로봇 팔을 카메라의 시야 내로 이동합니다.
2. 카메라 파라미터 계산 단계에서 [ 카메라 외부 파라미터 계산 ] 버튼을 클릭합니다. 이 작업은 카메라가 사진을 캡처하도록 트리거합니다.
3. 시나리오 뷰어를 클릭하여 로봇 포인트 클라우드와 로봇 모델 간의 매칭 정도를 확인합니다.

로봇의 포인트 클라우드가 로봇 모델과 대략적으로 매칭하면 캘리브레이션 결과를 사용할 수 있음을 의미합니다.





1. 로봇 모델은 실제 로봇의 모양과 완전히 매칭하지 않을 수 있으며 외부 파라미터를 미세 조정하기 위한 기준으로 사용할 수 없습니다.
2. **시나리오 뷰어** 외에도 Mech-Viz 소프트웨어에서 로봇의 포인트 클라우드와 로봇 모델 간의 매칭 정도를 확인할 수 있습니다.

4. 하단 표시줄에서 [저장] 버튼을 클릭하고 팝업한 **캘리브레이션 파일 저장** 다이얼로그 박스에서 [확인] 버튼을 클릭합니다. 카메라 캘리브레이션 결과는 프로젝트의 "calibration" 디렉터리에 자동으로 저장됩니다.

이로써 캘리브레이션 프로세스가 완료됩니다.

### 4.4.3. Eye in Hand 시나리오에서의 자동 캘리브레이션

이 부분에서는 EIH(Eye in Hand) 시나리오에서 자동 캘리브레이션을 완료하는 방법에 대해 설명합니다.

#### 캘리브레이션 사전 준비

핸드-아이 캘리브레이션을 수행하기 전에 다음 준비를 완료해야 합니다.

- Mech-Mind 비전 시스템 구축을 완료합니다.
- 캘리브레이션에 필요한 재료를 준비합니다.
- 캘리브레이션 보드 포인트 클라우드 이미징을 조정합니다.
- 로봇 및 인터페이스 구성을 완료합니다.

#### 비전 시스템 전체 구축 완료

Mech-Mind 비전 시스템 구축을 완료하려면 비전 시스템 구축 부분 내용을 참조하십시오.

핸드-아이 캘리브레이션에는 Mech-Eye Viewer, Mech-Center, Mech-Vision 및 Mech-Viz 소프트웨어가 필요합니다. 위의 소프트웨어가 설치되고 최신 버전으로 업데이트되었는지 확인하십시오.

#### 캘리브레이션에 필요한 재료를 준비

EIH 시나리오에서 자동 캘리브레이션을 하려면 캘리브레이션 보드를 사용해야 합니다.

다음 요구 사항에 따라 캘리브레이션 보드를 준비하십시오.

- 캘리브레이션 보드에서 도트가 잘 보이고 눈에 띄는 굵은 흔적이 없으며 캘리브레이션 보드가 뒤틀리지 않아야 합니다.
- EIH 시나리오에서 캘리브레이션 보드를 작업대의 중앙에 평평하게 놓으십시오.

또한 캘리브레이션 시작 전 로봇을 캘리브레이션 시작 위치로 이동합니다. EIH 시나리오에서의 자동 캘리브레이션인 경우 로봇의 캘리브레이션 초기 위치는 카메라의 작업 위치입니다(카메라는 로봇에 따라 하층에서 상층으로 캘리브레이션함).

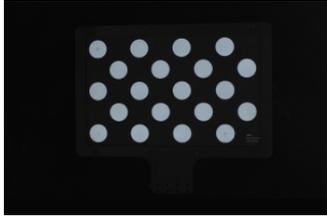
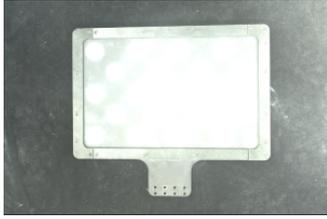
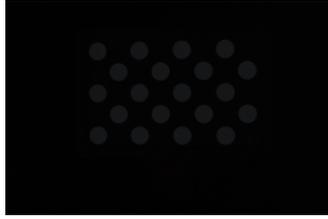
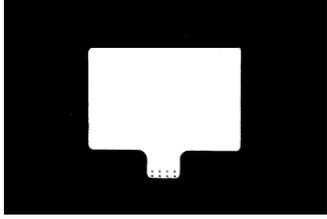
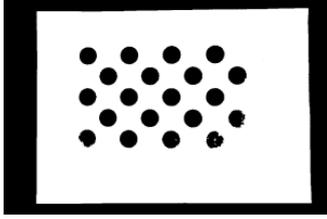
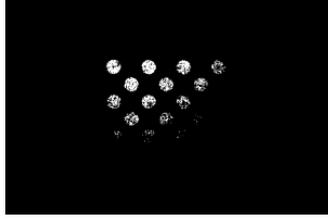
#### 캘리브레이션 보드 포인트 클라우드 이미징을 조정하기

1. Mech-Eye Viewer 소프트웨어를 열고 카메라 파라미터를 조정합니다.
2. 2D 파라미터를 조정함으로써 2D 맵의 캘리브레이션 보드가 뚜렷하고 노출 과다나 노출 부족 등 현상이 없도록 합니다.
3. 3D 파라미터를 조정함으로써 캘리브레이션 보드의 원의 포인트 클라우드를 완전하게 하고 **포인트 클라우드 후처리**에서 **포인트 클라우드 평활화** 및 **노이즈 제거**를 **일반**으로 변경하는 것이 좋으며 포인트 클라우드 변동 범위를 줄입니다.



현장에 환경광이 복잡한 경우, 환경광이 2D 맵과 포인트 클라우드에 주는 영향을 줄이기 위해 광원을 차단하거나 보충하기 바랍니다.

4. 위의 단계를 완료하여 캘리브레이션 보드의 포인트 클라우드 이미징 품질이 요구 사항을 충족하는지 확인하십시오.

	정상	노출 과다	노출 부족
2D 맵			
포인트 클라우드			

#### 로봇 및 인터페이스 구성 완료

로봇이 표준 인터페이스를 사용해 비전 측과 통신하는 경우 로봇의 표준 인터페이스 통신 구성을 완료하십시오. 자세한 내용은 [표준 인터페이스 통신](#) 부분의 관련 내용을 참조하십시오.

로봇이 표준 인터페이스를 사용해 비전 측과 통신하는 경우 로봇의 마스터 컨트롤 통신 구성을 완료하십시오. 자세한 내용은 [마스터 컨트롤 통신](#) 부분의 해당 내용을 참조하십시오.

#### 캘리브레이션 사전 구성

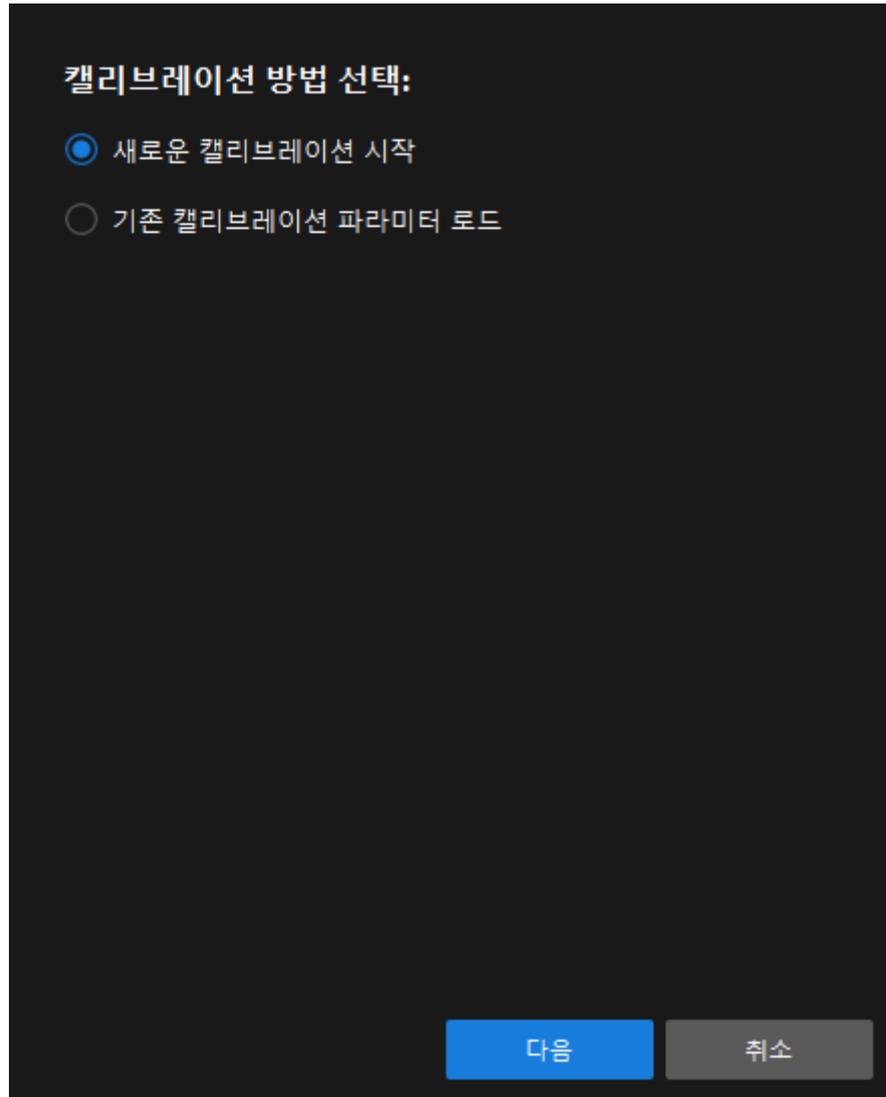
로봇이 표준 인터페이스를 사용해 비전 측과 통신하는 경우 [캘리브레이션 사전 구성\(표준 인터페이스\)](#)을 완료하십시오.

로봇이 마스터 컨트롤 방식을 사용해 비전 측과 통신하는 경우 [캘리브레이션 사전 구성\(마스터 컨트롤\)](#)을 완료하십시오.

#### 캘리브레이션 사전 구성(표준 인터페이스)

표준 인터페이스를 사용하여 로봇과 통신하는 경우 다음 단계를 수행하십시오.

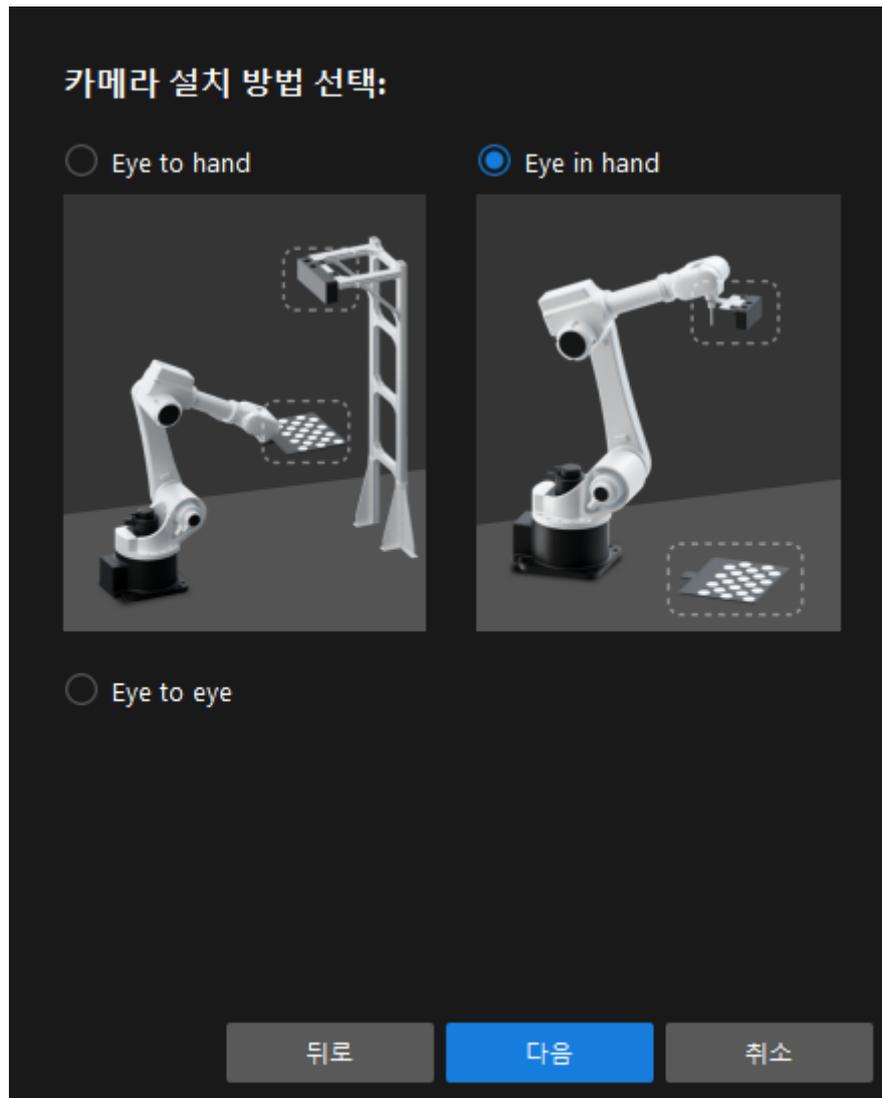
1. Mech-Vision 소프트웨어에서 톨 바의 [ [카메라 캘리브레이션\(표준\)](#) ] 버튼을 클릭합니다. [캘리브레이션 사전 구성](#) 창이 팝업됩니다.
2. [캘리브레이션 방법 선택](#) 창에서 [새로운 캘리브레이션 시작](#) 라디오 버튼을 선택한 후 [ [다음](#) ] 버튼을 클릭합니다.



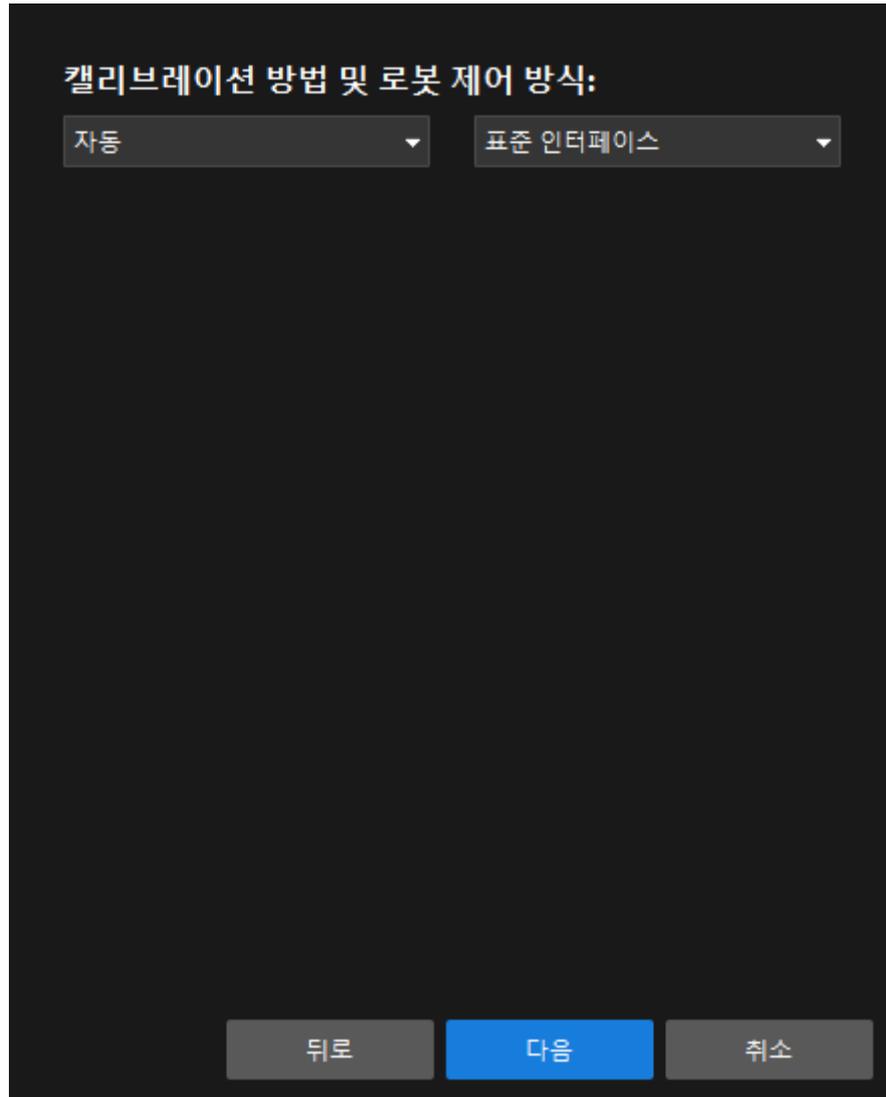
3. **캘리브레이션 작업 선택** 창의 드롭다운 목록 상자에서 **목록에 있는 로봇을 위한 핸드-아이 캘리브레이션**을 선택하고 **[로봇 모델 선택]** 버튼을 클릭하여 프로젝트에서 사용하는 로봇 모델을 선택한 후 **[다음]** 버튼을 클릭합니다.



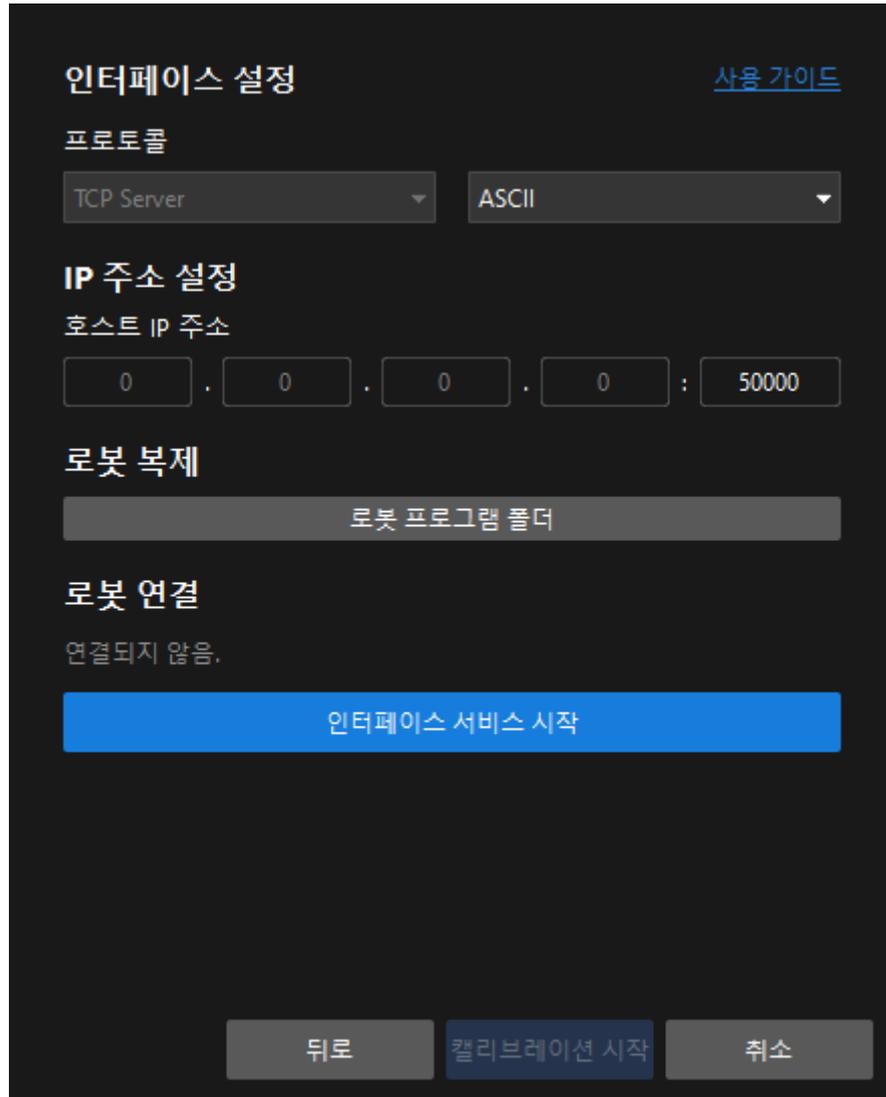
4. 카메라 설치 방법 선택 창에서 **Eye in hand** 라디오 버튼을 선택한 후 [ 다음 ] 버튼을 클릭합니다.



5. 캘리브레이션 방법 및 로봇 제어 방식 창에서 자동 캘리브레이션 및 표준 인터페이스를 선택한 후 [ 다음 ] 버튼을 클릭합니다.



6. 통신 설정 창에서 파라미터 **프로토콜** 및 **호스트 IP 주소**를 설정한 다음 [ **인터페이스 서비스 시작** ] 버튼을 클릭합니다. 버튼이 **로봇이 연결되기를 기다리는 중...**으로 변경됩니다.



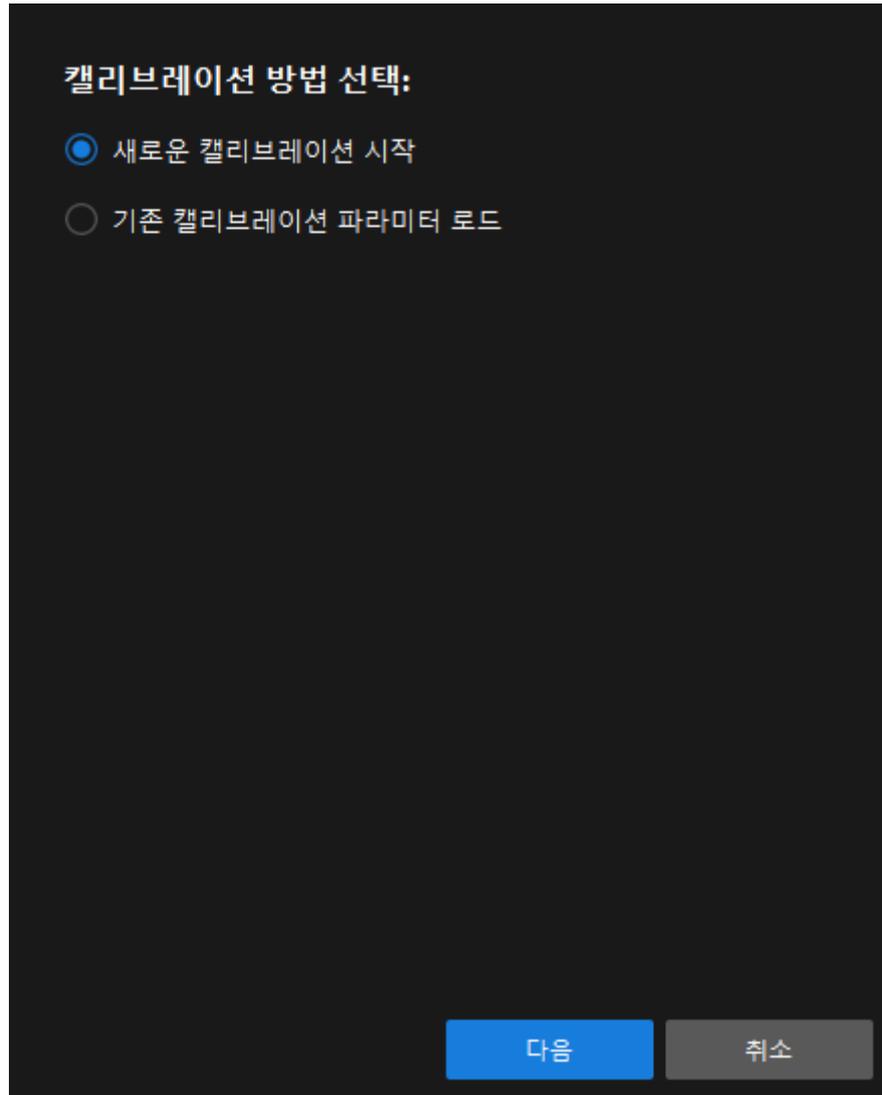
위 그림에서 **TCP 서버**와 **Hex(Little-endian)**은 ABB 로봇을 위한 **프로토콜** 구성의 예시일 뿐입니다. 기타 로봇의 프로토콜 설정이 다르기 때문에 소프트웨어가 자동으로 로봇의 기본 구성을 추가하므로 기본 구성을 유지하십시오.

7. 로봇의 티치 펜던트에서 **자동 캘리브레이션 절차를 선택하고 티치 펜던트를 통해 캘리브레이션 시작점을 설정한 다음 캘리브레이션 절차를 실행**합니다. 다른 로봇의 캘리브레이션 작업 프로세스에 대해서는 **표준 인터페이스 통신** 부분을 참조하십시오. 프로그램이 성공적으로 시작된 후 로그 바에서 "캘리브레이션 프로세스로 들어갑니다. Mech-Vision에서 캘리브레이션을 시작하십시오"라는 로그가 로그 열에 인쇄됩니다.
8. Mech-Vision 소프트웨어로 돌아가 **로봇 연결** 영역에 "연결됨" 상태 메시지가 표시될 때까지 기다린 다음 [ **캘리브레이션 시작** ] 버튼을 클릭합니다. **캘리브레이션(Eye in Hand) 창**이 나타납니다.

#### 캘리브레이션 사전 구성(마스터 컨트롤)

마스터 컨트롤 방식을 사용하여 로봇과 통신하는 경우 다음 단계를 수행하십시오.

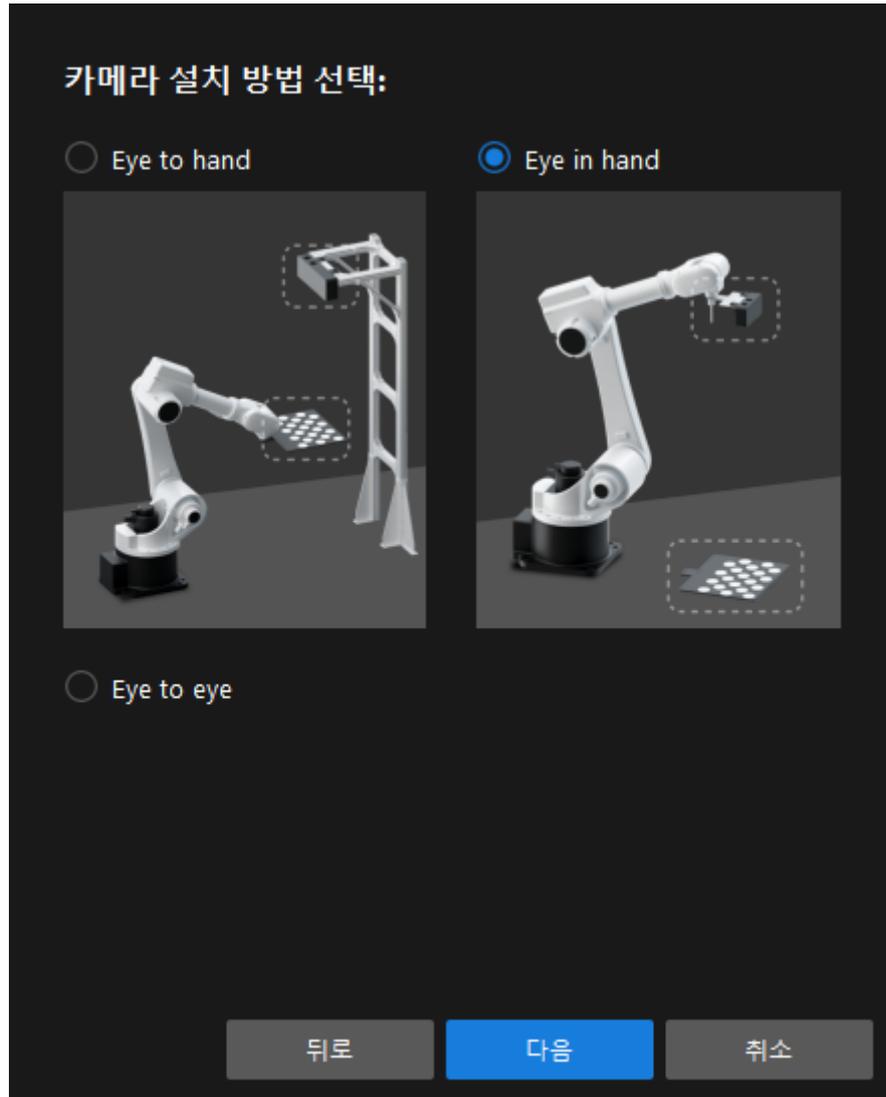
1. Mech-Vision 소프트웨어에서 톨 바의 [ **카메라 캘리브레이션(표준)** ] 버튼을 클릭합니다. **캘리브레이션 사전 구성** 창이 팝업됩니다.
2. **캘리브레이션 방법 선택** 창에서 **새로운 캘리브레이션 시작** 라디오 버튼을 선택한 후 [ **다음** ] 버튼을 클릭합니다.



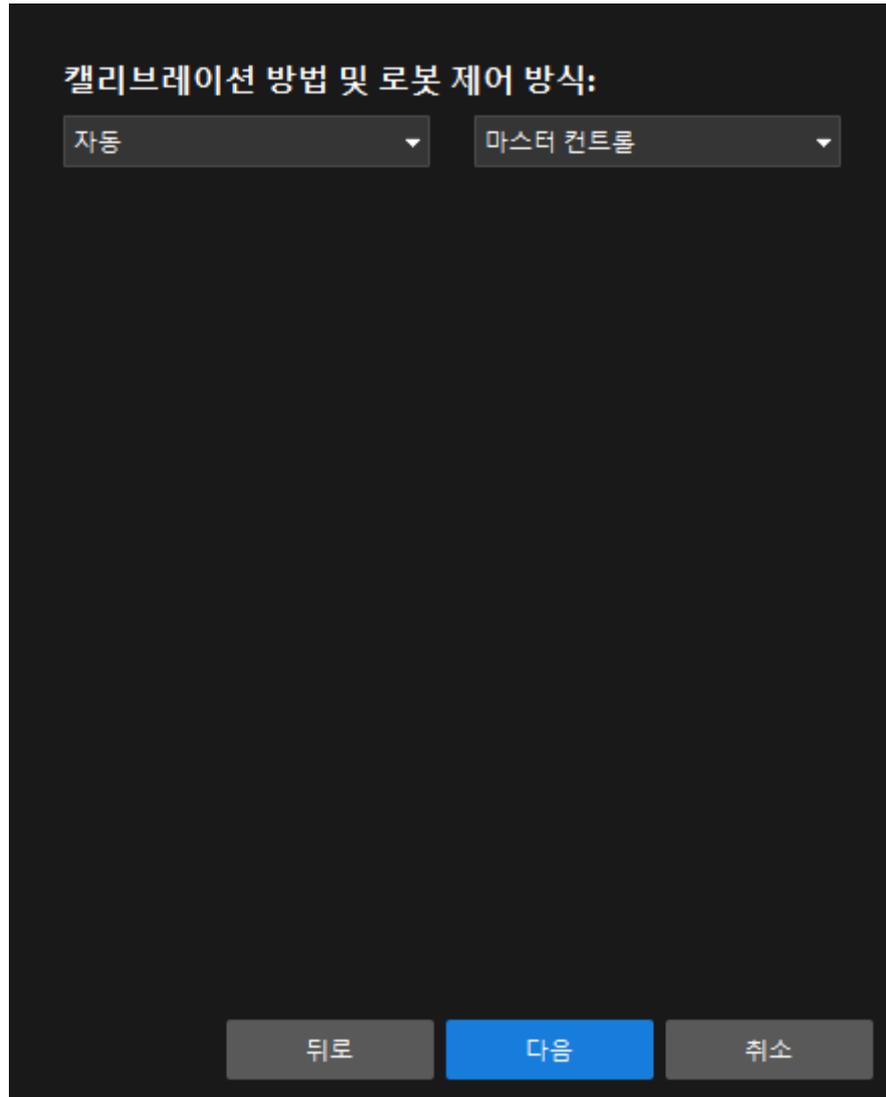
3. **캘리브레이션 작업 선택** 창의 드롭다운 목록 상자에서 **목록에 있는 로봇을 위한 핸드-아이 캘리브레이션**을 선택하고 **[로봇 모델 선택]** 버튼을 클릭하여 프로젝트에서 사용하는 로봇 모델을 선택한 후 **[다음]** 버튼을 클릭합니다.



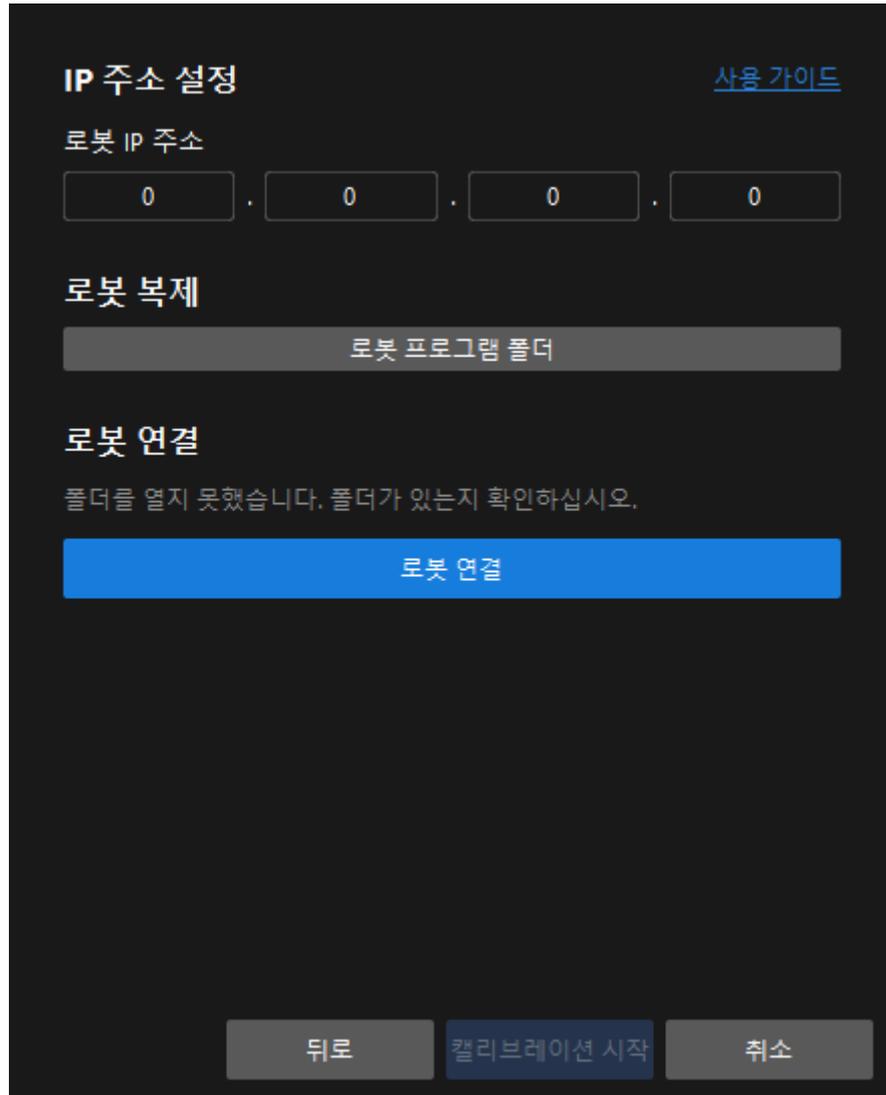
4. 카메라 설치 방법 선택 창에서 **Eye in hand** 라디오 버튼을 선택한 후 [ 다음 ] 버튼을 클릭합니다.



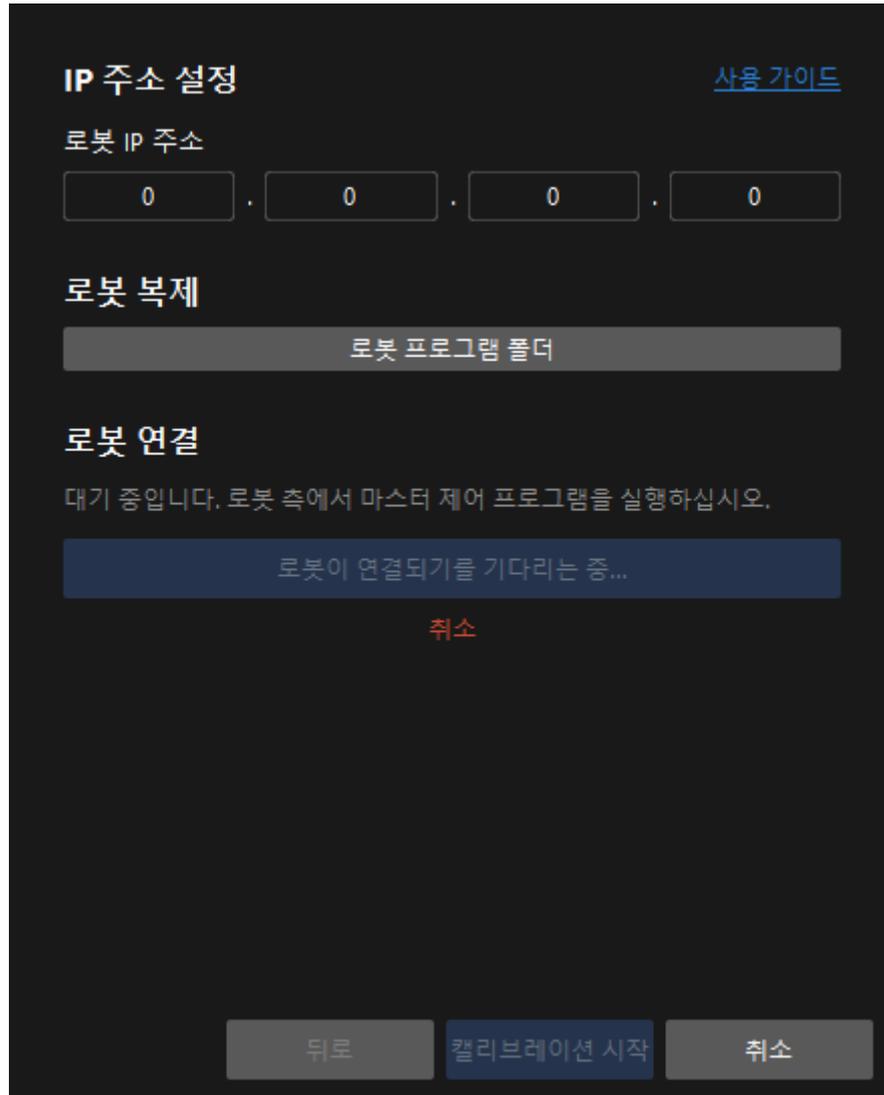
5. 캘리브레이션 방법 및 로봇 제어 방식 창에서 자동 및 마스터 컨트롤을 선택한 후 [다음] 버튼을 클릭합니다.



6. 새 창에서 **로봇 IP 주소** 파라미터를 설정합니다.



7. 로봇의 티치 펜던트에서 **마스터 컨트롤 프로그램을 선택하고 실행**합니다. 다른 로봇의 사용 방법은 **마스터 컨트롤 통신** 부분을 참조하십시오.
8. Mech-Vision 소프트웨어로 돌아가서 **로봇 연결** 영역에서 [ **로봇 연결** ] 버튼을 클릭합니다. 버튼이 **로봇이 연결되기를 기다리는 중...**으로 변경됩니다.



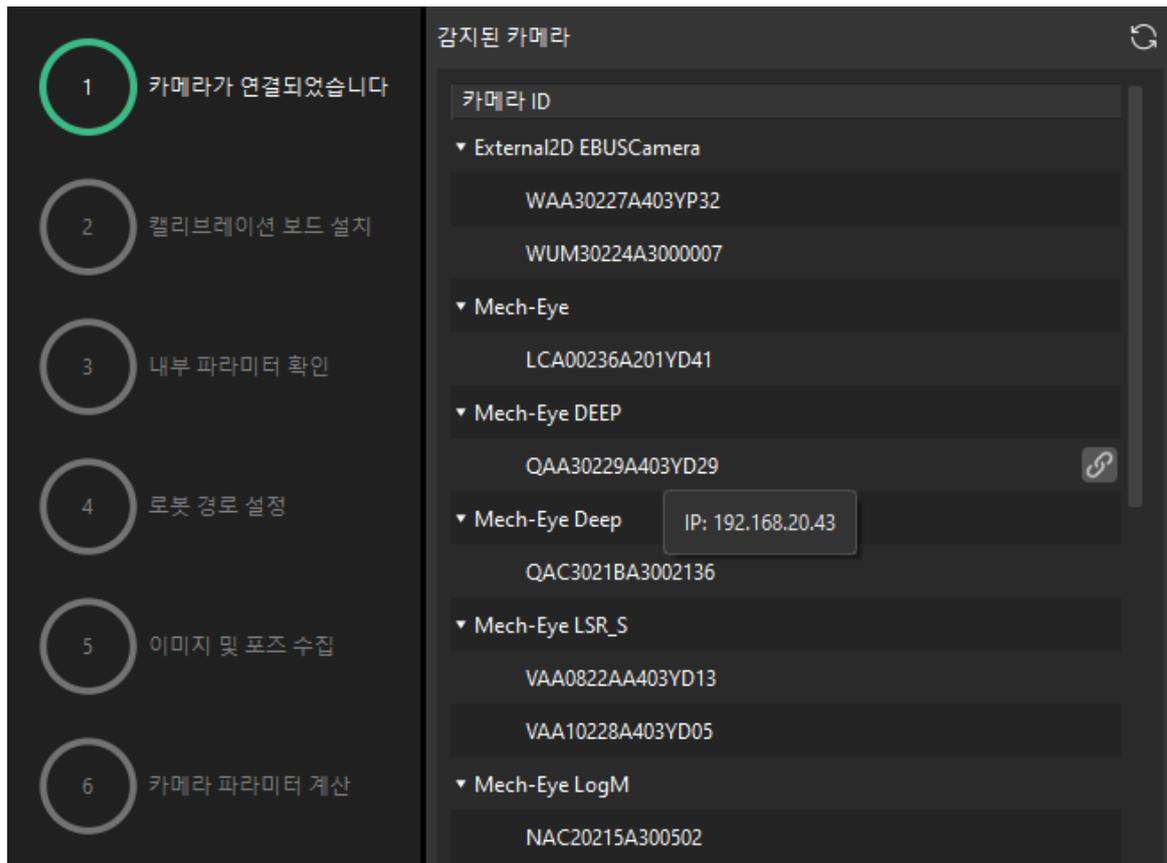
9. **로봇 연결** 영역에 "연결됨" 상태 메시지가 표시될 때까지 기다린 다음 [ **캘리브레이션 시작** ] 버튼을 클릭합니다. **캘리브레이션(Eye in Hand)** 창이 나타납니다.

여기까지 캘리브레이션 사전 구성이 완료되었으며 정식 캘리브레이션 프로세스로 들어갑니다.

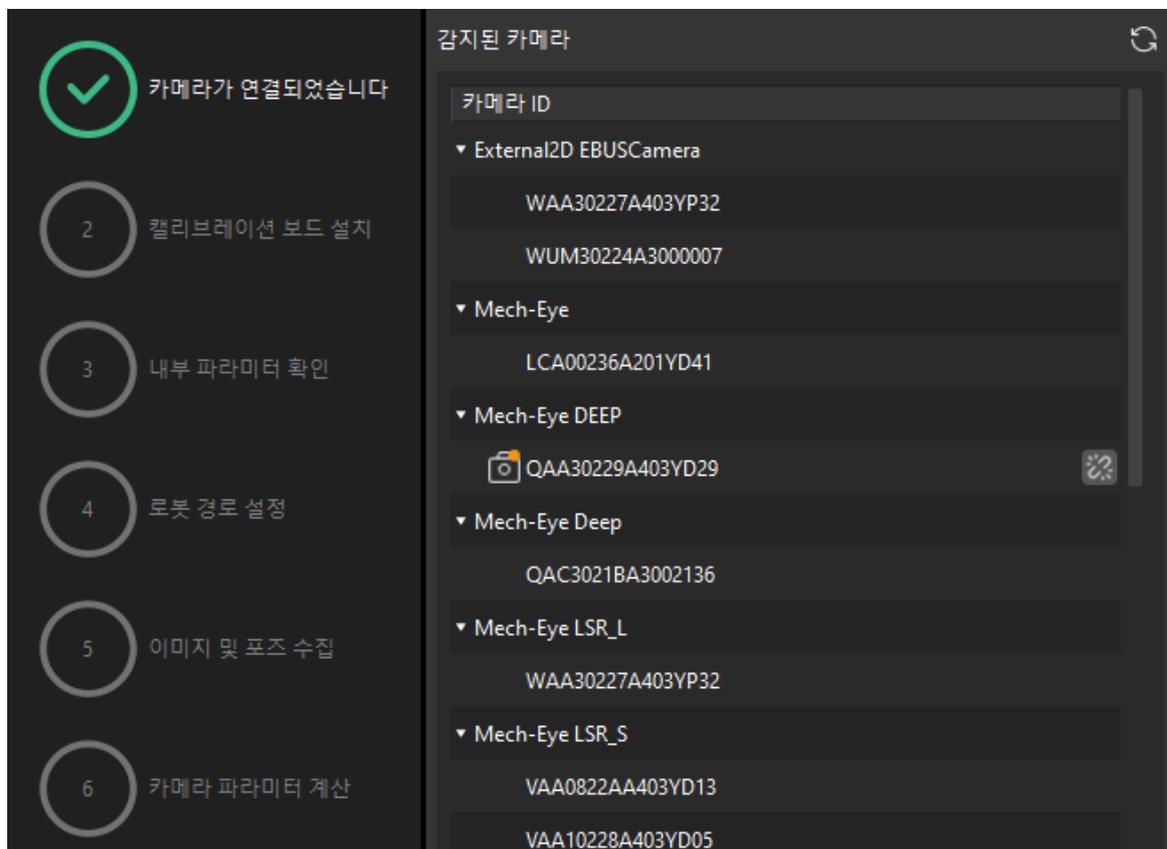
### 캘리브레이션 프로세스

#### 카메라 연결

1. **카메라 연결** 단계에서 **감지된 카메라** 리스트에서 연결할 카메라를 찾은 다음  을 클릭하면 카메라가 연결되며 또는 직접 연결할 카메라를 더블클릭합니다.



2. 카메라를 연결한 후 [연속 캡처] 또는 [한번 캡처] 버튼을 선택할 수 있습니다.



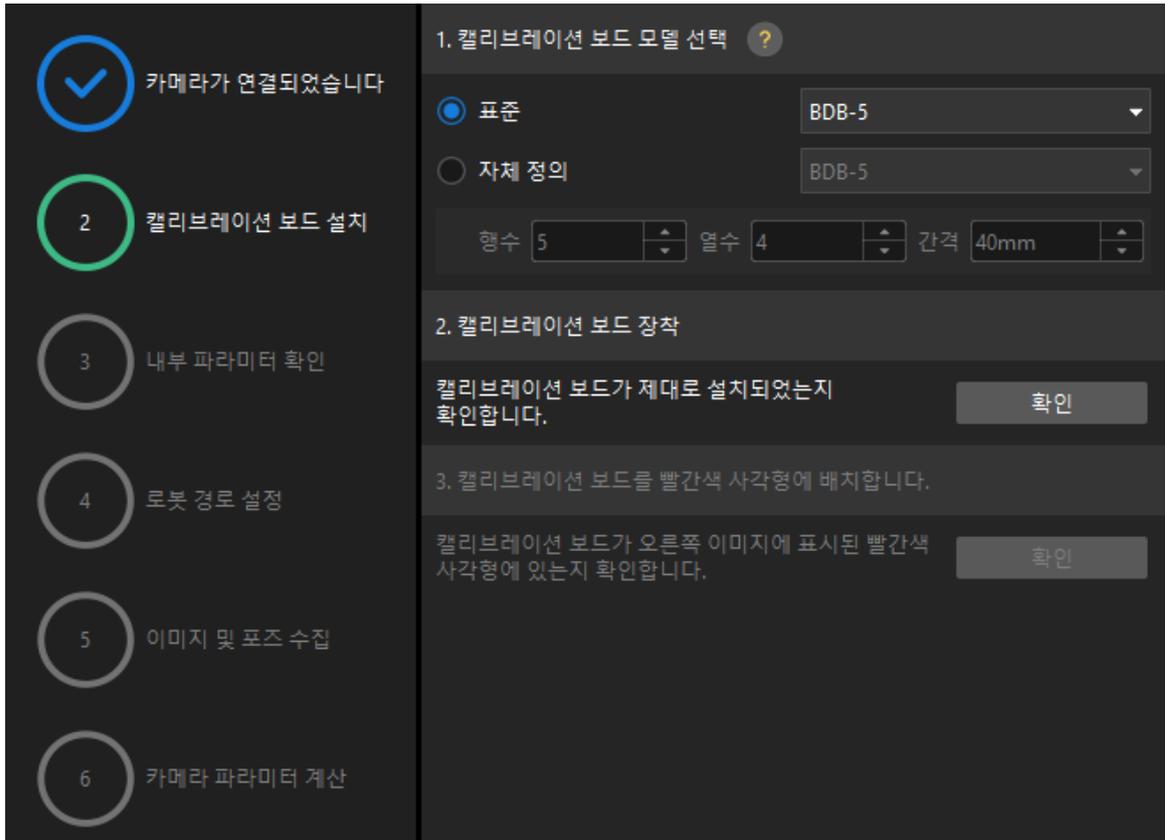
3. 이미지 뷰어에서 카메라가 캡처한 2D 맵과 덤스 맵이 캘리브레이션 요구 사항을 충족하는지 확인한 다음 하단 표시줄에서 [ 다음 ] 버튼을 클릭합니다.



캡처한 이미지가 캘리브레이션 요구 사항을 충족하지 않으면 Mech-Eye Viewer 소프트웨어를 열어 **카메라의 2D 및 3D 노출 파라미터를 조정**하고 다시 캡처해야 합니다.

#### 캘리브레이션 보드 설치

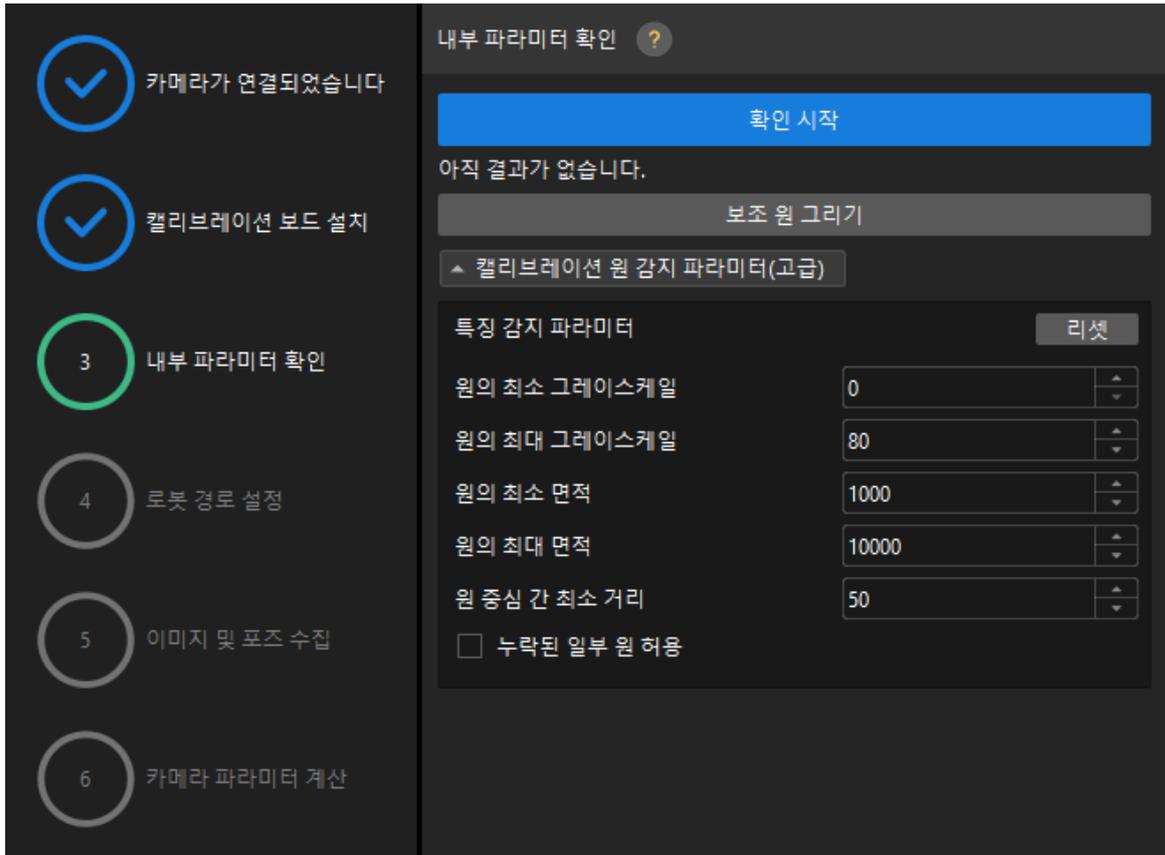
1. 캘리브레이션 보드 설치 단계의 **1. 캘리브레이션 보드 유형 선택** 영역에서 **표준** 라디오 버튼을 선택한 후 캘리브레이션 보드 모델 레이블에 따라 해당 캘리브레이션 유형을 선택합니다.



2. 캘리브레이션 보드가 작업 평면 중앙에 놓여 있는지 확인한 후 **2. 캘리브레이션 보드 설치** 영역에서 [ 확인 ] 버튼을 클릭합니다.
3. 캘리브레이션 보드가 카메라 시야의 중앙(빨간색 사각형 내)에 있는지 확인한 다음 **3. 캘리브레이션 보드를 빨간색 사각형에 배치** 영역에서 [ 확인 ] 버튼을 클릭합니다.
4. 캘리브레이션 보드 관련 작업을 모두 완료한 후 하단 표시줄에서 [ 다음 ] 버튼을 클릭합니다.

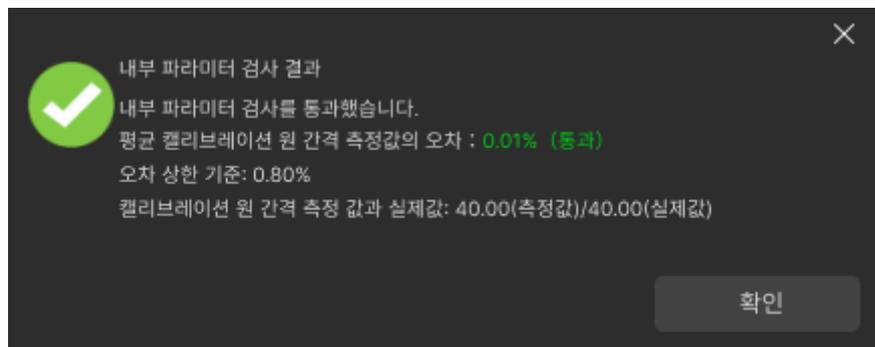
#### 카메라 내부 파라미터를 검사하기

1. 내부 파라미터 검사 단계에서 [ 검사 시작 ] 버튼을 클릭합니다.



## 2. 카메라의 내부 파라미터 검사 결과를 확인합니다.

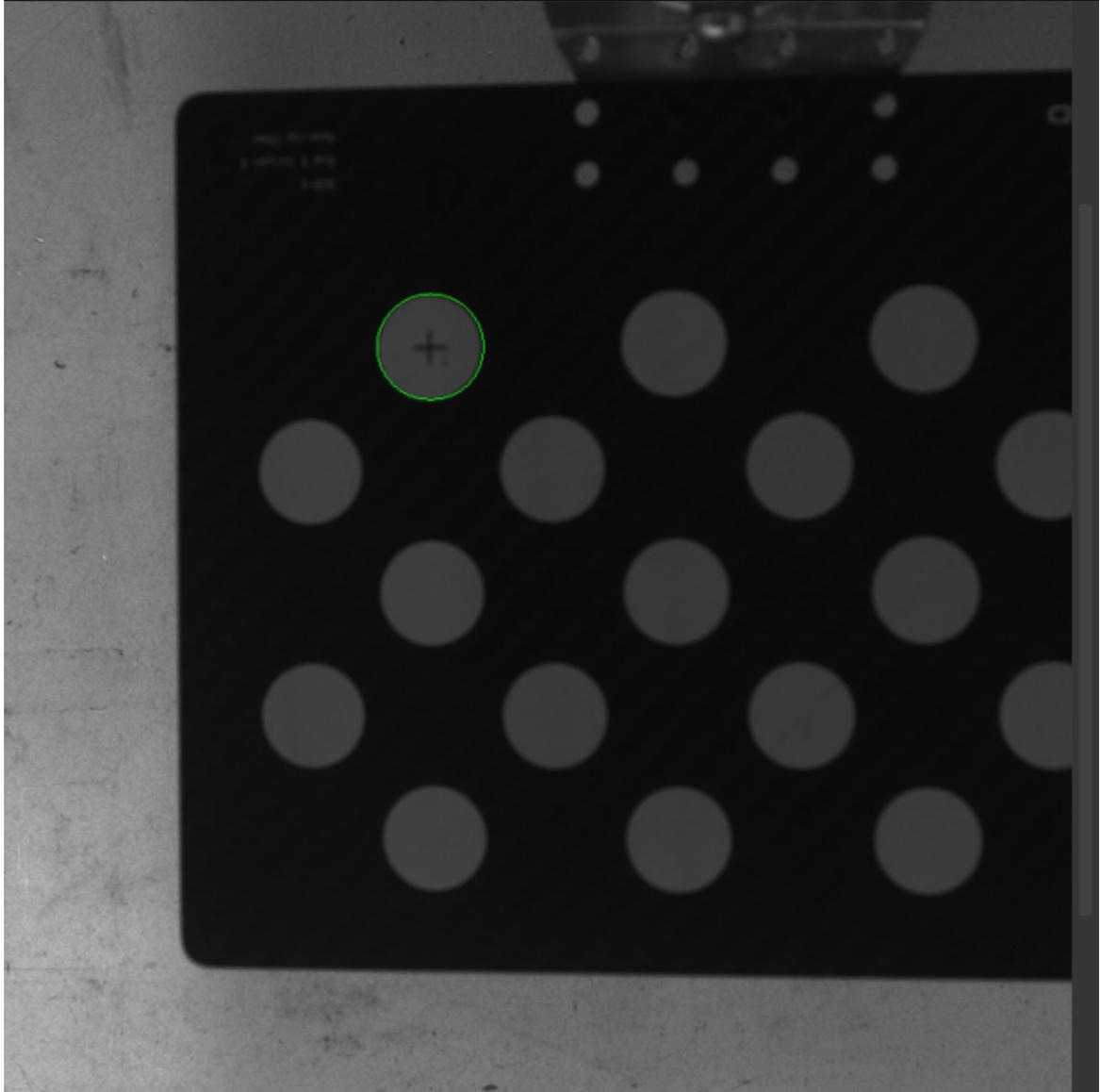
- 카메라의 내부 파라미터 검사에 성공하면 [확인]을 클릭한 후 [다음] 버튼을 클릭하십시오.



- 카메라의 내부 파라미터 검사에 실패하면 보조원을 그리거나 캘리브레이션 원을 수동으로 편집하여 파라미터를 감지한 다음 [다시 검사] 버튼을 클릭하십시오.

### 보조원을 그리기

1. 보조원을 그리기로 선택한 경우 [보조원 그리기] 버튼을 클릭하십시오.
2. 오른쪽 이미지 뷰어 패널에서 마우스 오른쪽으로 캘리브레이션 보드 이미지를 클릭하고 해당 창 확인란을 언체크 한 다음 **Ctrl** 키를 누른 상태에서 롤러를 드래그하여 이미지를 적절한 크기로 조정하십시오.
3. 마우스 포인터를 캘리브레이션 원의 중간 줄 지점으로 이동하고 왼쪽 마우스 버튼을 누른 다음 보조원에 캘리브레이션 원을 완전히 포함시킨 다음 해제하십시오.



4. [ 다시 확인 ] 버튼을 클릭하고 카메라 내부 파라미터 검사가 통과되었는지 확인하십시오.

#### 감지 파라미터 수동으로 조정

감지 파라미터를 수동으로 조정할 경우 [ 캘리브레이션 원 감지 파라미터(고급) ]을 클릭하여 감지 파라미터의 값을 변경하면 됩니다.

캘리브레이션 원이 여전히 감지되지 않은 경우 실제 작업 조건에 따라 카메라 관련 파라미터를 조정해야 하며 카메라 파라미터는 [파라미터 조정](#) 내용을 참조하십시오.

#### 로봇 경로 설정

1. 로봇 경로 설정 단계에서 카메라 댄스 방향 캘리브레이션 범위 파라미터를 설정합니다.

- ✓ 카메라가 연결되었습니다
- ✓ 캘리브레이션 보드 설치
- ✓ 내부 파라미터 확인
- 4 **로봇 경로 설정**
- 5 이미지 및 포즈 수집
- 6 카메라 파라미터 계산

### 1. 캘리브레이션 포즈 높이 범위

높이 범위

### 2. 로봇 경로 설정

로봇 경로 파라미터 리셋

경로 유형 InHand ▾

경로를 카메라 좌표계에 자동으로 정렬 ?

경로 자동 정렬

피라미드

높이 설정

높이 범위  레이어 수

하단 레이어 치수

X  Y

최상위 레이어 치수

X  Y

레이어당 이동 그리드 행 및 열 수

X  Y

위성

스텝 수

중심에서 캘리브레이션 보드까지의 높이

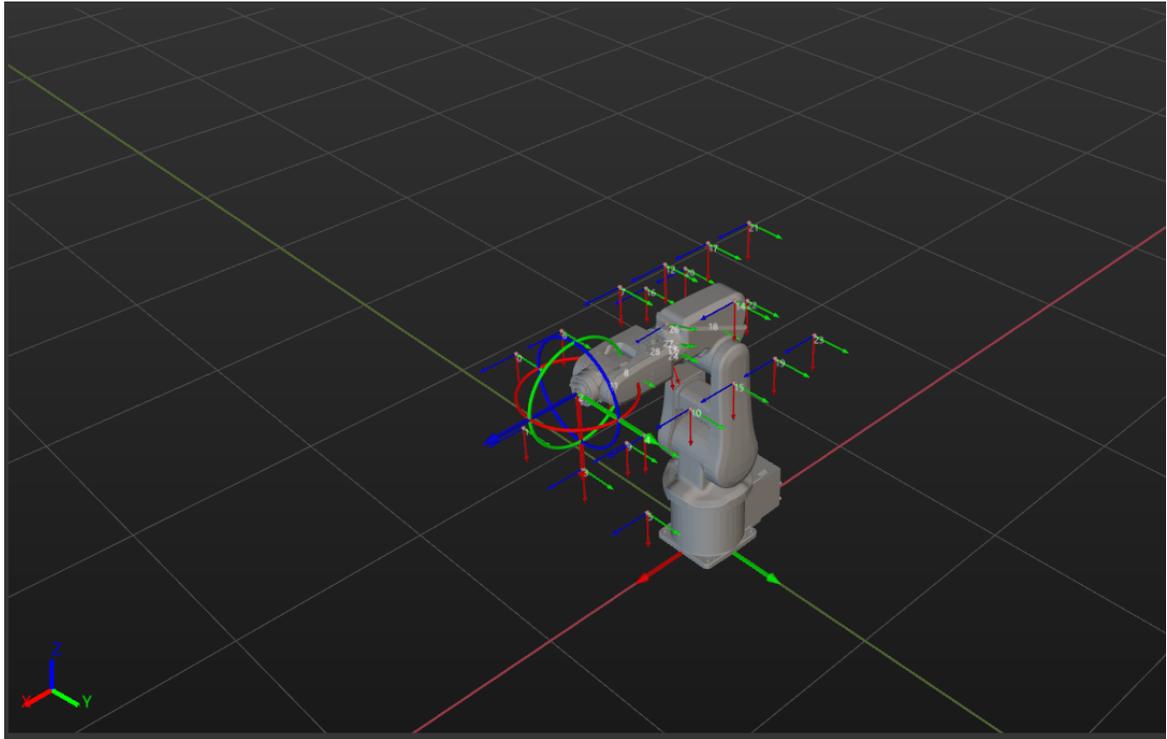
반경

확인



카메라 뎀스 방향 캘리브레이션 범위 파라미터는 카메라의 권장 작동 거리 범위와 로봇의 작업 공간에 따라 설정해야 합니다.

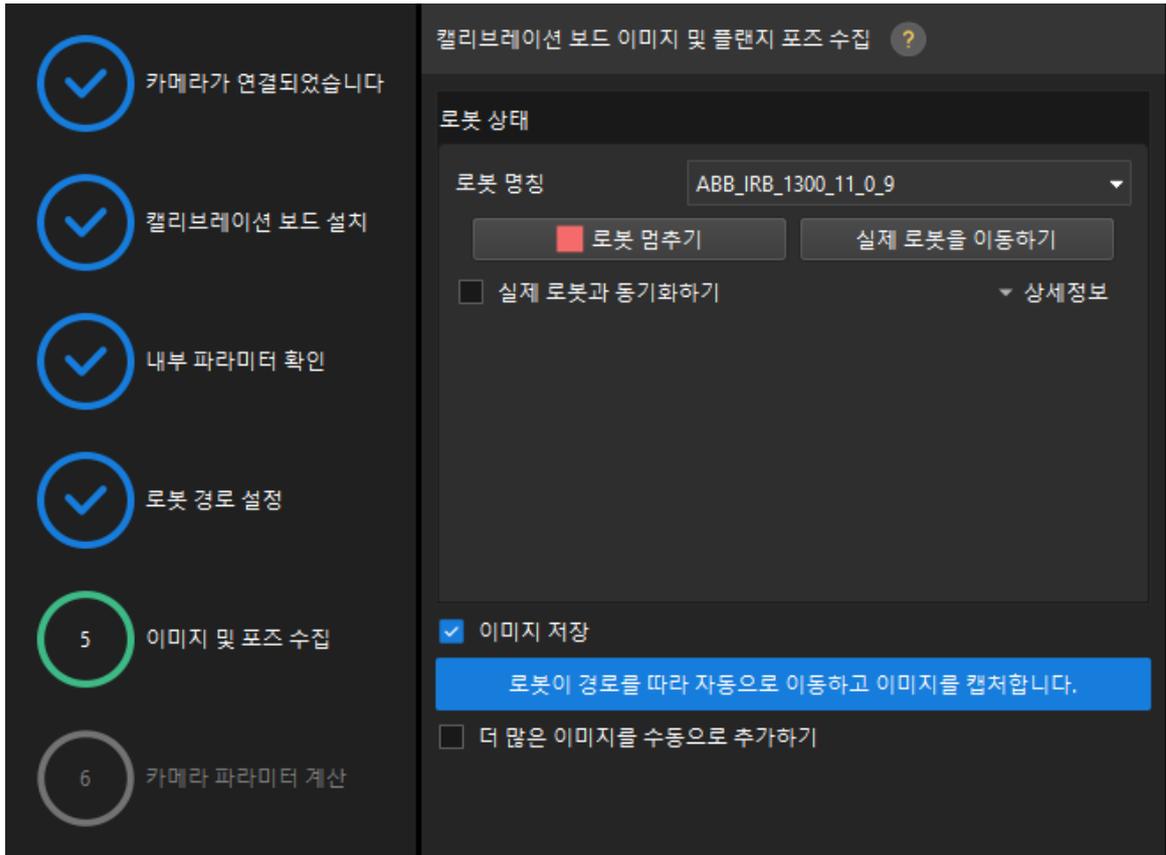
2. 카메라의 z 축과 로봇의 플랜지 좌표계의 z 축이 평행하지 않은 경우 [ 경로 자동 정렬 ] 버튼을 클릭하십시오.
3. 필요에 따라 경로 유형 파라미터를 InHand로 설정하고 피라미드 파라미터 높이 범위, 층 수, 하단 크기 X/Y, 상단 크기 X/Y 및 행당 이동 그리드의 행 및 열 수, 위성 파라미터 스텝 수, 중심에서 캘리브레이션 보드까지의 높이, 반경을 지정한 다음 [ 확인 ] 버튼을 클릭합니다.
4. 우측 시나리오 뷰 패널에서 각 웨이포인트의 위치가 합리적이고 주변 환경과 충돌하지 않는지 확인합니다.



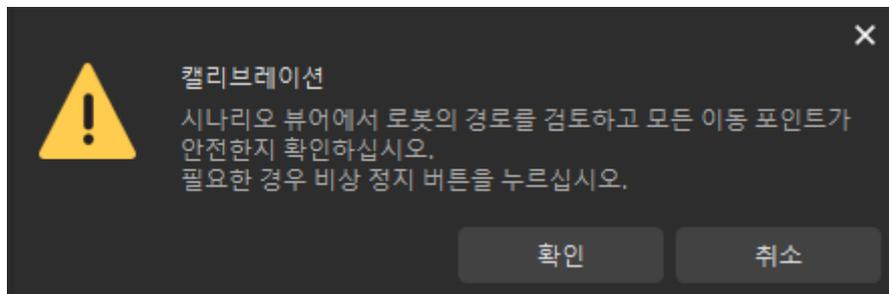
5. 하단에서 [ 다음 ] 버튼을 클릭하십시오.

#### 이미지 및 포즈 가져오기

1. 이미지 및 포즈 가져오기 스텝에서 이미지를 저장하기를 선택한 다음 [ 로봇이 경로를 따라 움직이고 이미지를 캡처하게 하기 ] 버튼을 클릭합니다.



2. 로봇 이동 안전 수칙을 주의 깊게 읽은 다음 [확인] 버튼을 클릭합니다.

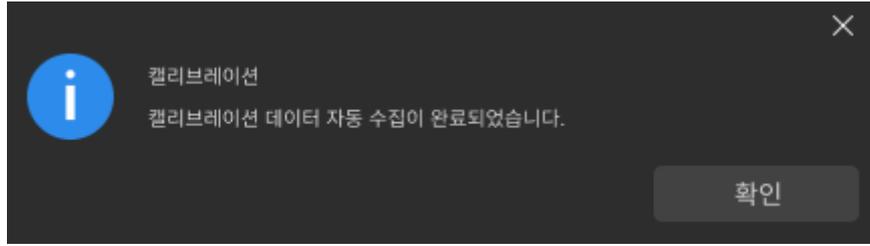


3. 설정된 경로에 따라 로봇이 움직이고 카메라가 각 웨이포인트의 이미지 캡처를 완료할 때까지 기다립니다. 오른쪽의 **이미지 뷰어** 패널에 캡처한 이미지가 표시됩니다.



- 로봇이 움직일 때 로봇이 사람을 다치게 하지 않도록 로봇 작업 구역에서 멀리 떨어져 계십시오.
- 로봇 멈추기를 클릭하여 캘리브레이션을 종료할 수 있지만 로봇은 즉시 중지되지 않으며 현재 웨이포인트를 이동 한 후에 중지됩니다. 비상 사태가 발생하면 로봇 터치 펜던트에서 비상 정지 버튼을 누르십시오 (비상 정지 버튼을 누른 후 로봇을 다시 연결해야 함).

4. 이미지를 캡처 한 후 팝업창에서 [확인] 버튼을 클릭하십시오.



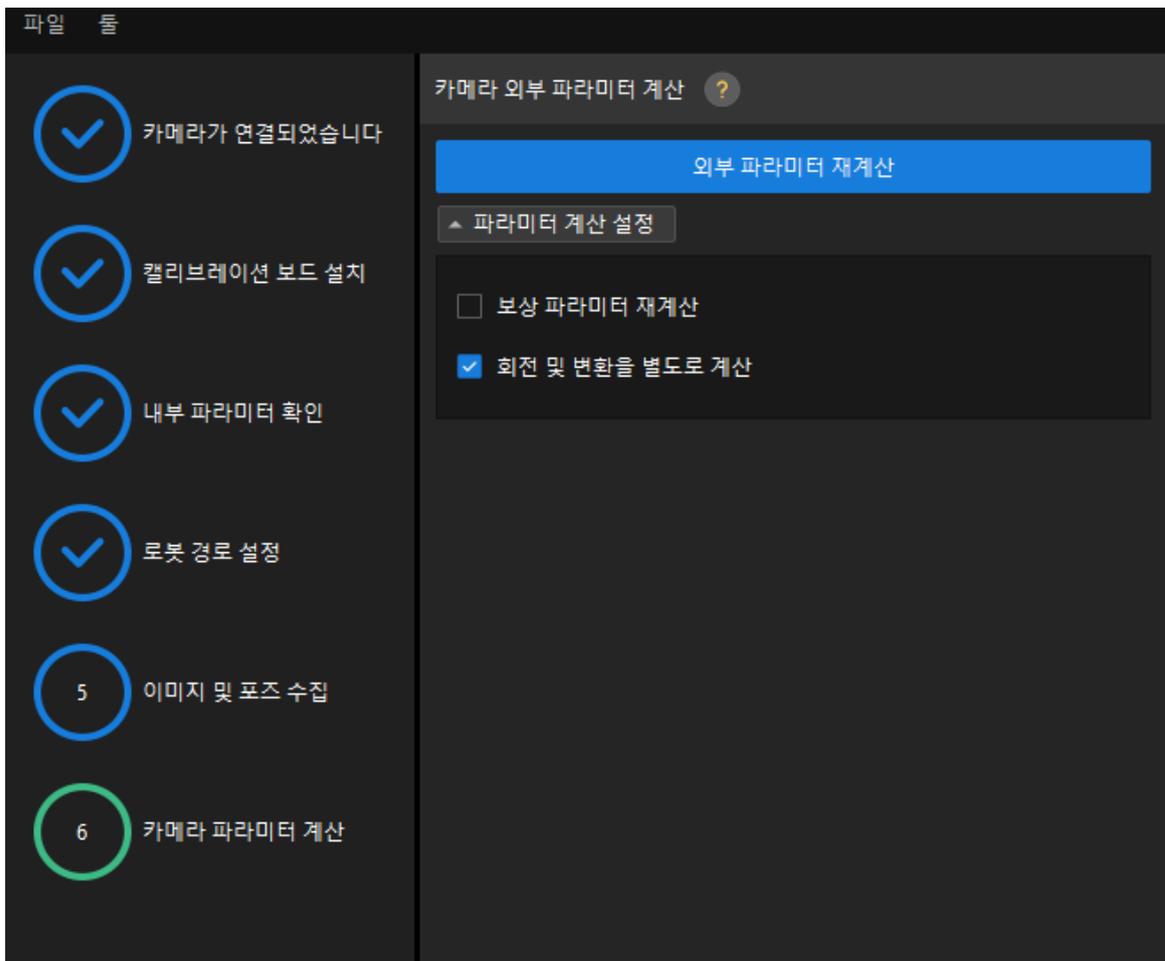
5. 현재 인식한 캘리브레이션 포인트가 요구 사항을 충족하는지 확인한 다음 하단 열의 [ 다음 ] 버튼을 클릭하십시오.



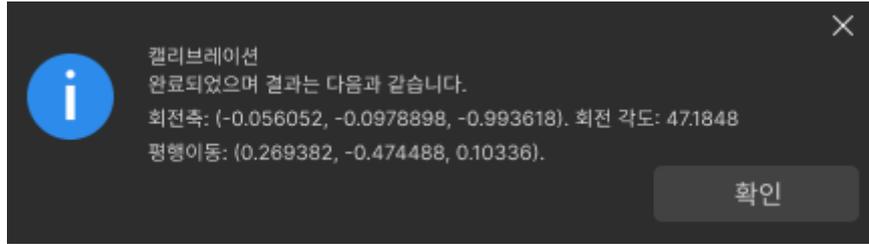
만족하지 않는 경우, 로봇을 수동으로 이동한 후 (로봇 터치 펜던트로 이동하거나 Mech-Viz를 통해 이동하는 것을 선택할 수 있음), **더 많은 이미지를 수동으로 추가하기**를 선택하고 [ **이미지를 추가하고 플랜지 포즈를 기록하기** ] 버튼을 클릭하여 캘리브레이션 보드 이미지를 추가하고 로봇의 플랜지 포즈를 입력합니다.

### 카메라 파라미터를 계산하기

1. 카메라 파라미터를 계산하기 단계에서 **회전 및 변환을 별도로 계산** 파라미터를 선택한 다음 [ **카메라 외부 파라미터 계산** ] 버튼을 클릭하십시오.



2. 팝업한 캘리브레이션 성공 다이얼로그 박스에서 [ **확인** ] 버튼을 클릭합니다.



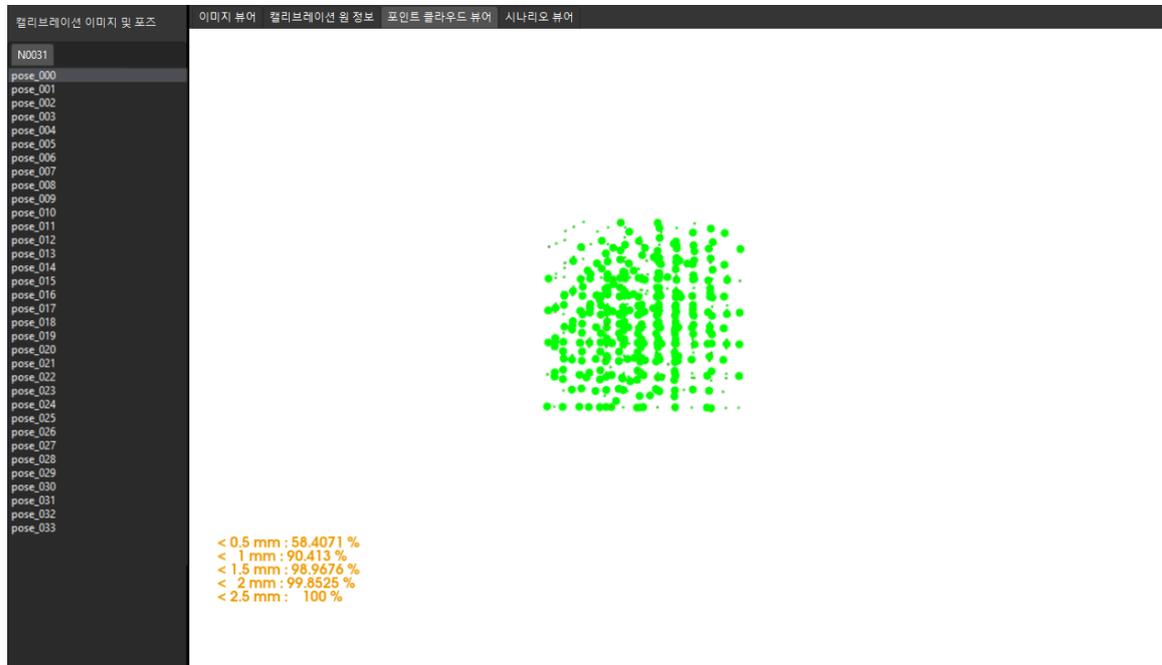
- 오른쪽 **포인트 클라우드 뷰어** 패널에서 캘리브레이션 오차 포인트 클라우드를 확인합니다.



오차 포인트 클라우드는 각 캘리브레이션 포즈의 캘리브레이션 보드 원의 진릿값과 계산된 값 사이의 편차를 표시하는 데 사용됩니다.

- 캘리브레이션 정확도가 예상 요구 사항을 충족하는지 확인합니다.

캘리브레이션의 정확도를 대략적으로 확인하기 위해 100%를 차지하는 오차 값을 찾으십시오. 예를 들어, 아래 그림의 정확도는 2.5mm 미만입니다.



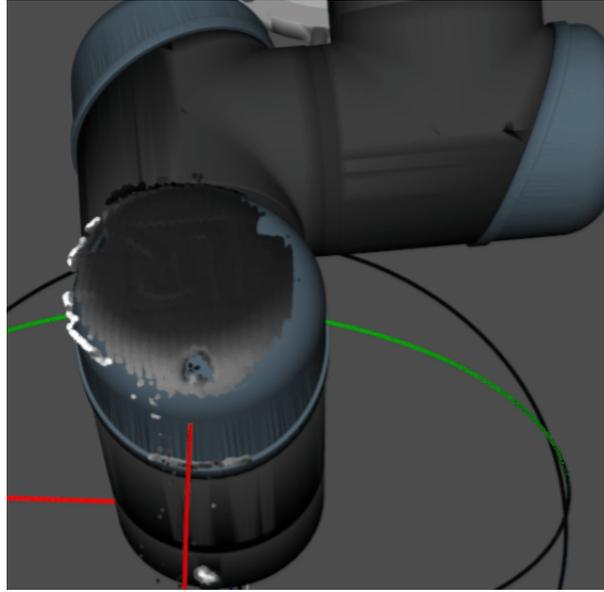
캘리브레이션 정확도를 개선해야 하는 경우 [캘리브레이션 결과 분석](#) 내용을 참조하십시오.

### 캘리브레이션 결과 검증 및 저장

카메라 외부 파라미터를 계산한 후 시나리오 뷰어에서 로봇 포인트 클라우드와 로봇 모델의 매칭 정도를 보고 캘리브레이션 결과를 대략적으로 판단할 수 있습니다. 구체적인 작업은 다음과 같습니다.

- 로봇 팔을 카메라의 시야 내로 이동합니다.
- 카메라 파라미터 계산 단계에서 [ **카메라 외부 파라미터 계산** ] 버튼을 클릭합니다. 이 작업은 카메라가 사진을 캡처하도록 트리거합니다.
- 시나리오 뷰어를 클릭하여 로봇 포인트 클라우드와 로봇 모델 간의 매칭 정도를 확인합니다.

로봇의 포인트 클라우드가 로봇 모델과 대략적으로 매칭하면 캘리브레이션 결과를 사용할 수 있음을 의미합니다.



1. 로봇 모델은 실제 로봇의 모양과 완전히 매칭하지 않을 수 있으며 외부 파라미터를 미세 조정하기 위한 기준으로 사용할 수 없습니다.
2. 시나리오 뷰어 외에도 Mech-Viz 소프트웨어에서 로봇의 포인트 클라우드와 로봇 모델 간의 매칭 정도를 확인할 수 있습니다.

4. 하단 표시줄에서 [ 저장 ] 버튼을 클릭하고 팝업한 **캘리브레이션 파일 저장** 다이얼로그 박스에서 [ 확인 ] 버튼을 클릭합니다. 카메라 캘리브레이션 결과는 프로젝트의 "calibration" 디렉터리에 자동으로 저장됩니다.

이로써 캘리브레이션 프로세스가 완료됩니다.

#### 4.4.4. Eye to Hand 시나리오에서의 수동 캘리브레이션(캘리브레이션 보드의 여러 임의 포즈)

이 부분에서는 ETH(Eye to Hand) 시나리오에서 수동 캘리브레이션(캘리브레이션 보드의 여러 임의 포즈)을 완료하는 방법에 대해 설명합니다.

##### 캘리브레이션 사전 준비

핸드-아이 캘리브레이션을 수행하기 전에 다음 준비를 완료해야 합니다.

- Mech-Mind 비전 시스템 구축을 완료합니다.
- 캘리브레이션에 필요한 재료를 준비합니다.
- 캘리브레이션 보드 포인트 클라우드 이미징을 조정합니다.
- 로봇 및 인터페이스 구성을 완료합니다.

##### 비전 시스템 전체 구축 완료

Mech-Mind 비전 시스템 구축을 완료하려면 비전 시스템 구축 부분 내용을 참조하십시오.

핸드-아이 캘리브레이션에는 Mech-Eye Viewer, Mech-Center, Mech-Vision 및 Mech-Viz 소프트웨어가 필요합니다. 위의 소프트웨어가 설치되고 최신 버전으로 업데이트되었는지 확인하십시오.

### 캘리브레이션에 필요한 재료를 준비

ETH 시나리오에서 수동 캘리브레이션을 하려면 캘리브레이션 보드를 사용해야 합니다.

다음 요구 사항에 따라 캘리브레이션 보드를 준비하십시오.

- 캘리브레이션 보드에서 도트가 잘 보이고 눈에 띄는 굵은 흔적이 없으며 캘리브레이션 보드가 뒤틀리지 않아야 합니다.
- ETH 시나리오에서 먼저 로봇의 말단 플랜지에 캘리브레이션 보드 커넥터를 설치한 다음 캘리브레이션 보드를 커넥터에 설치합니다. 캘리브레이션 보드가 단단히 설치되어 있고 캘리브레이션 보드가 로봇 말단의 XY 평면과 평행한지 확인하십시오.



로봇 플랜지에 분리 불가능한 그리퍼를 장착한 경우에 캘리브레이션 보드를 그리퍼에 직접 고정하면 됩니다.

또한 캘리브레이션 시작 전 로봇을 캘리브레이션 시작 포즈로 이동합니다.

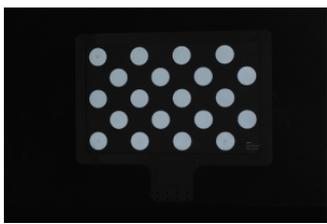
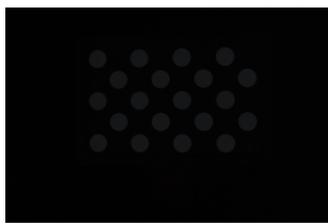
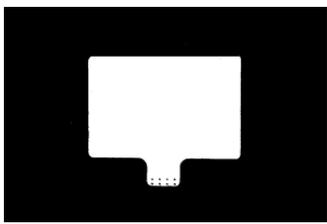
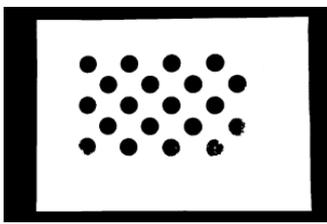
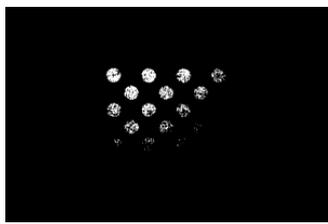
### 캘리브레이션 보드 포인트 클라우드 이미징을 조정하기

1. Mech-Eye Viewer 소프트웨어를 열고 카메라 파라미터를 조정합니다.
2. 2D 파라미터를 조정함으로써 2D 맵의 캘리브레이션 보드가 뚜렷하고 노출 과다나 노출 부족 등 현상이 없도록 합니다.
3. 3D 파라미터를 조정함으로써 캘리브레이션 보드의 원의 포인트 클라우드를 완전하게 하고 **포인트 클라우드 후처리에서 포인트 클라우드 평활화 및 노이즈 제거를 일반으로 변경하는 것이 좋으며** 포인트 클라우드 변동 범위를 줄입니다.



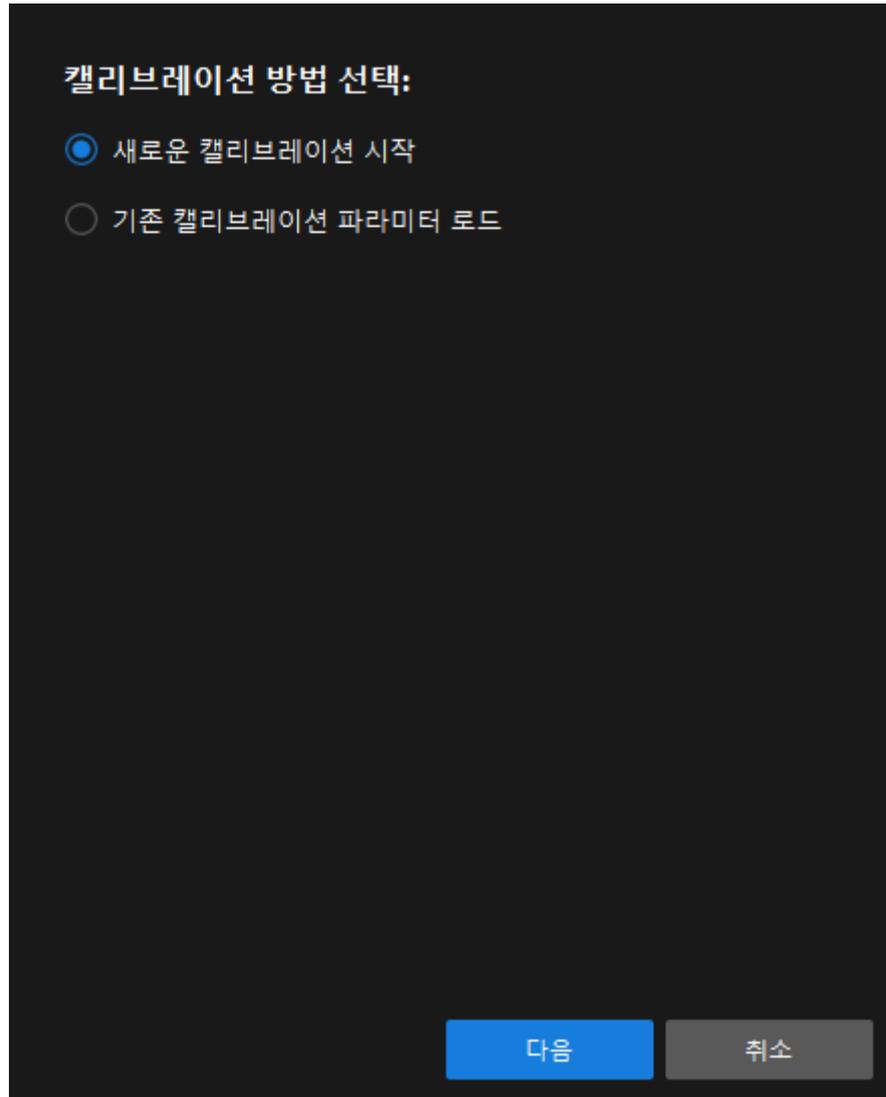
현장에 환경광이 복잡한 경우, 환경광이 2D 맵과 포인트 클라우드에 주는 영향을 줄이기 위해 광원을 차단하거나 보충하기 바랍니다.

4. 위의 단계를 완료하여 캘리브레이션 보드의 포인트 클라우드 이미징 품질이 요구 사항을 충족하는지 확인하십시오.

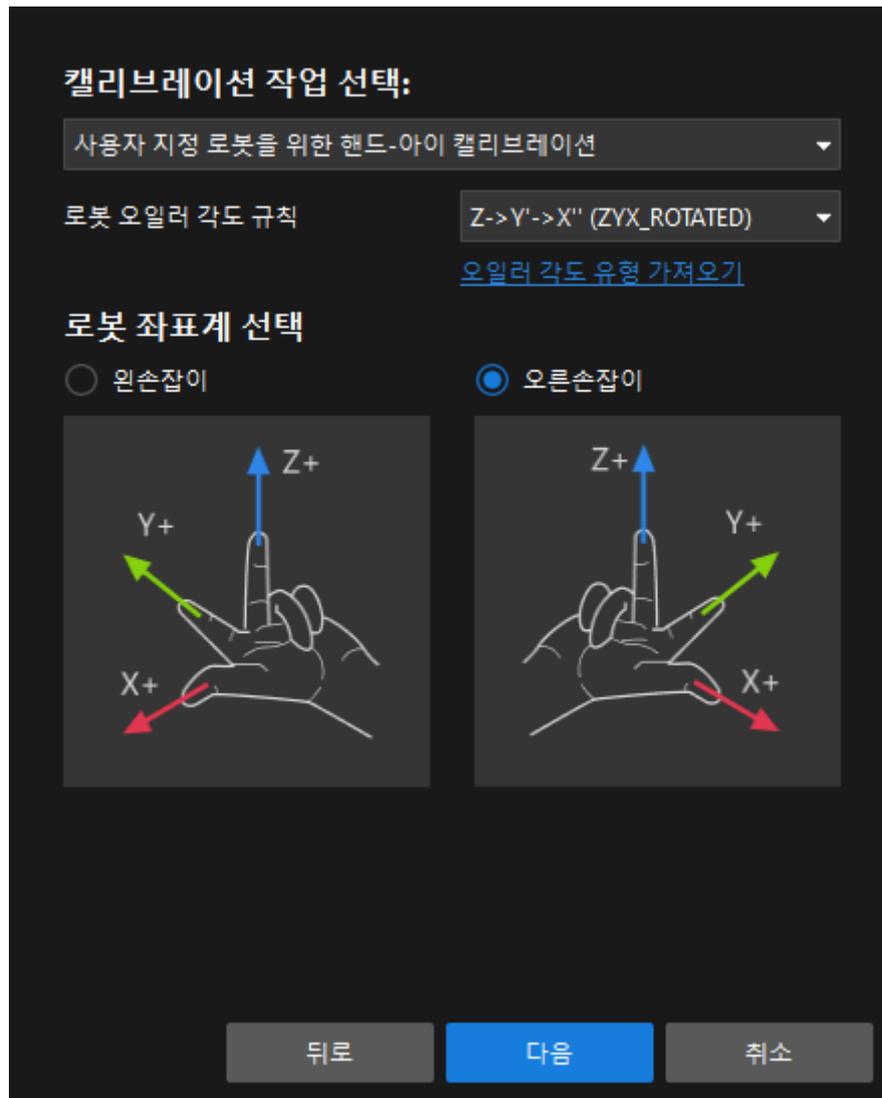
	정상	노출 과다	노출 부족
2D 맵			
포인트 클라우드			

### 캘리브레이션 사전 구성

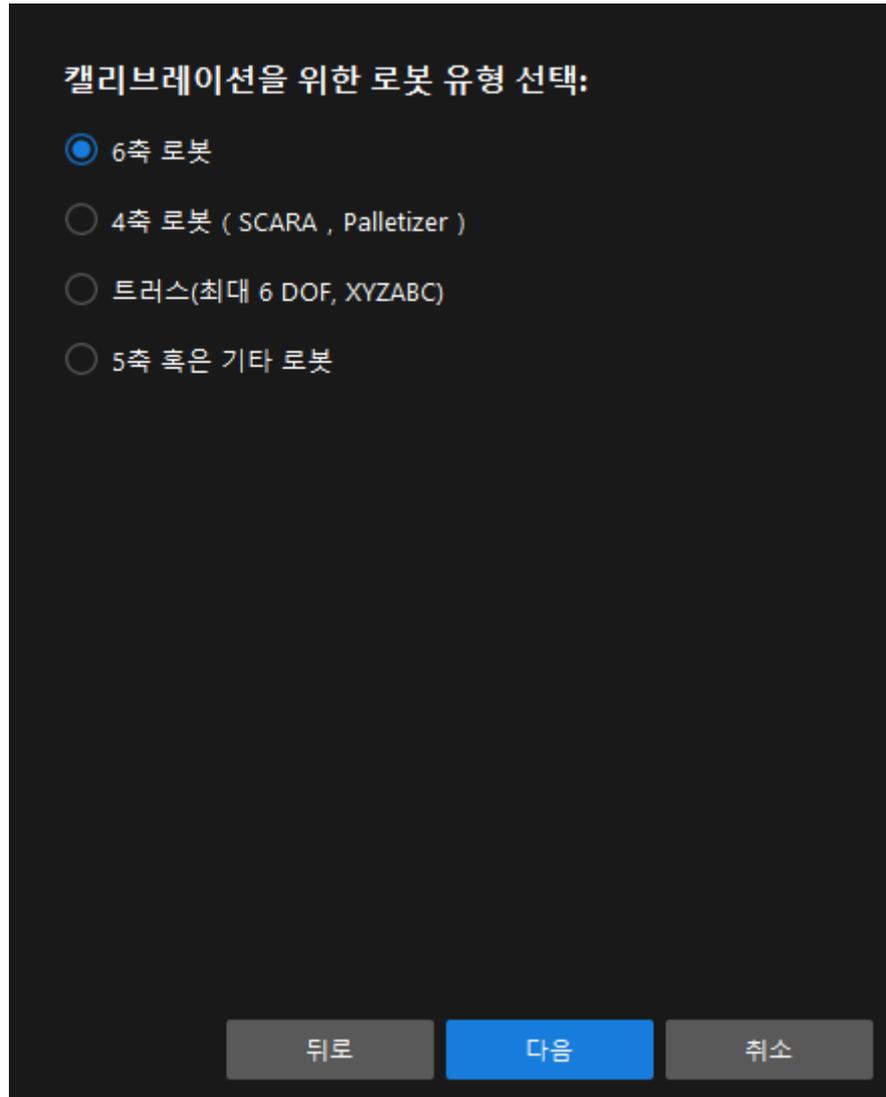
1. Mech-Vision 소프트웨어에서 톨 바의 [ 카메라 캘리브레이션(표준) ] 버튼을 클릭합니다. **캘리브레이션 사전 구성** 창이 팝업됩니다.
2. **캘리브레이션 방법 선택** 창에서 **새로운 캘리브레이션 시작** 라디오 버튼을 선택한 후 [ 다음 ] 버튼을 클릭합니다.



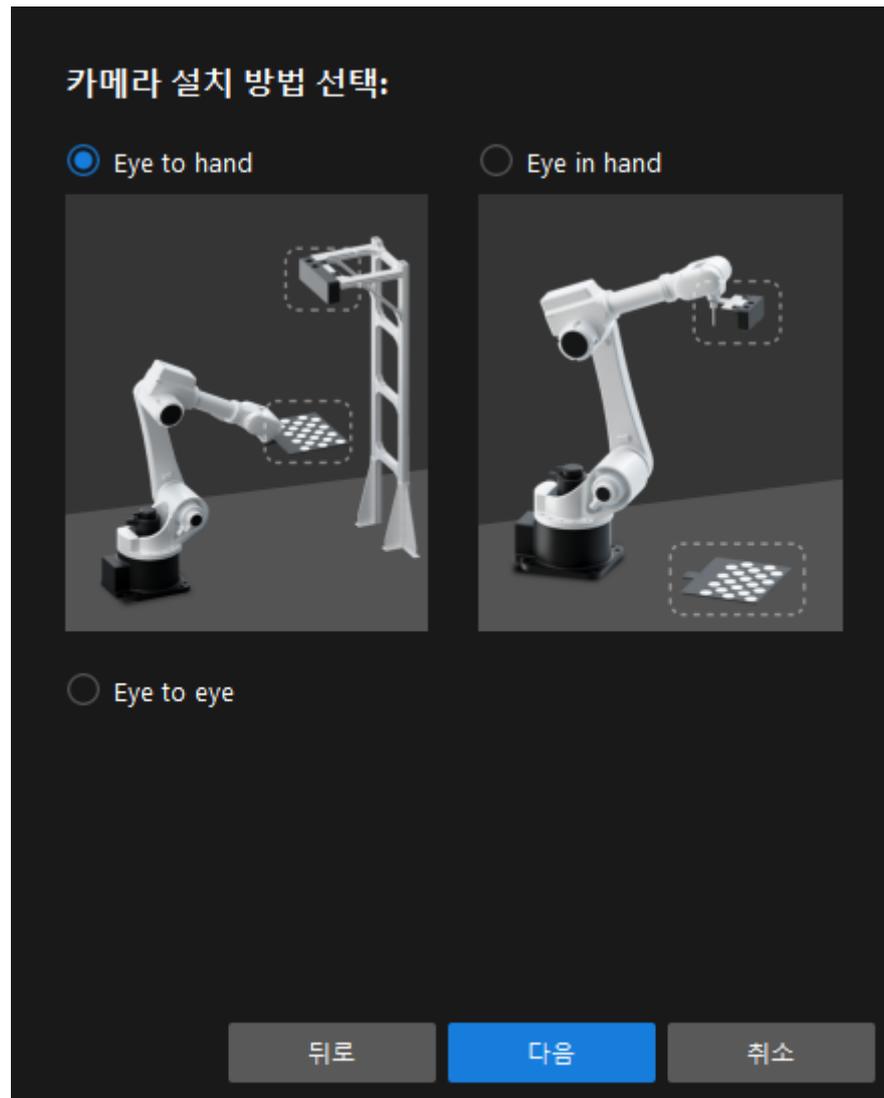
3. **캘리브레이션 작업 선택** 창의 드롭다운 목록 상자에서 **사용자 지정 로봇을 위한 핸드-아이 캘리브레이션**을 선택하고 **로봇 오일러 각도 유형** 파라미터를 지정하여 로봇 좌표계 유형을 선택한 후 [ **다음** ] 버튼을 클릭합니다.



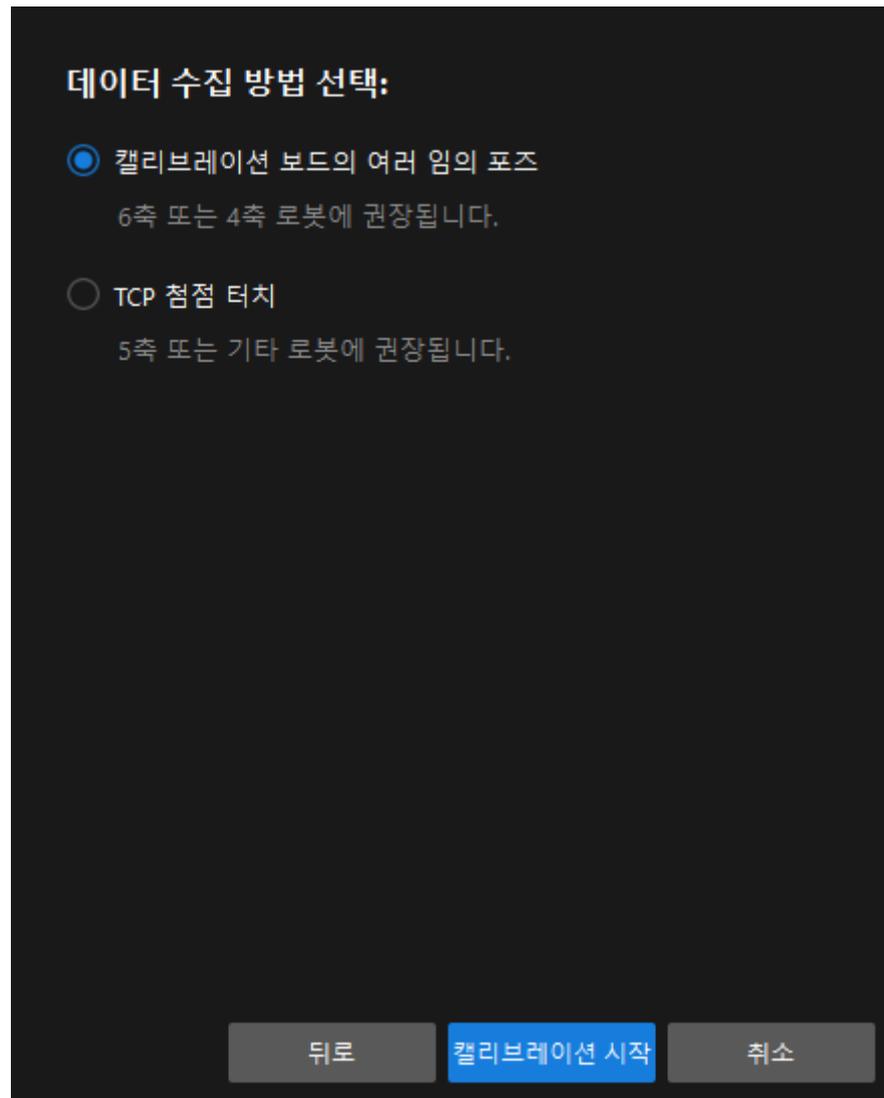
4. 캘리브레이션을 위한 로봇 유형 선택 창에서 로봇 유형에 따라 6축 로봇, 4축 로봇 (SCARA, Palletizer) 또는 5축 혹은 기타 로봇 라디오 상자를 선택한 후 [다음] 버튼을 클릭합니다.



5. 카메라 설치 방법 선택 창에서 **Eye to hand** 라디오 버튼을 선택한 후 [ 다음 ] 버튼을 클릭합니다.



6. 데이터 수집 방법 선택 창에서 캘리브레이션 보드의 여러 임의 포즈 라디오 버튼을 선택한 다음 [캘리브레이션 시작] 버튼을 클릭합니다. 캘리브레이션(Eye to Hand) 창이 나타납니다.

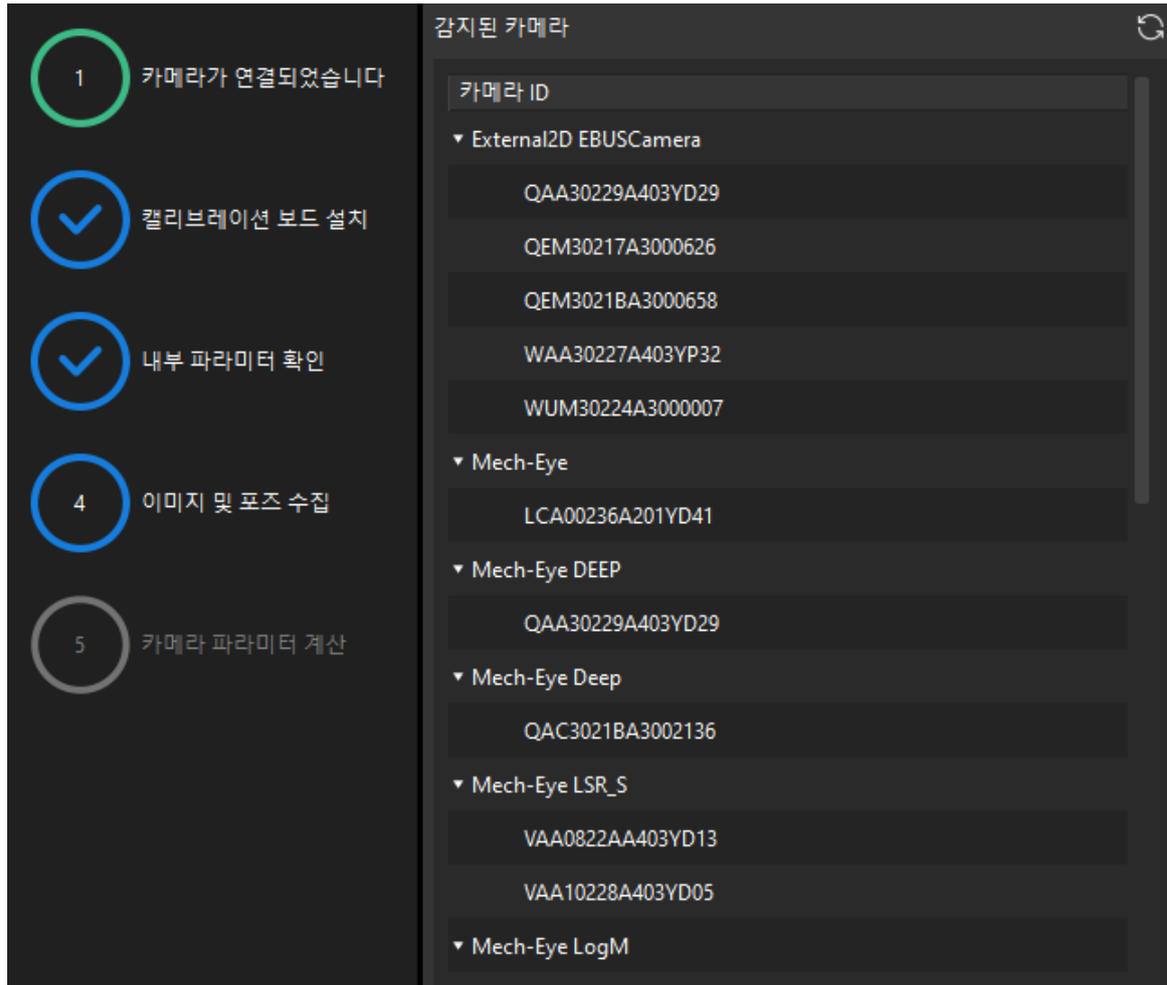


여기까지 캘리브레이션 사전 구성이 완료되었으며 정식 캘리브레이션 프로세스로 들어갑니다.

## 캘리브레이션 프로세스

### 카메라 연결

1. **카메라 연결** 단계에서 **감지된 카메라** 리스트에서 연결할 카메라를 선택한 다음  을 클릭하면 카메라가 연결되며 또는 직접 연결할 카메라를 더블클릭합니다.



2. 카메라를 연결한 후 [연속 캡처] 또는 [한번 캡처] 버튼을 선택할 수 있습니다.



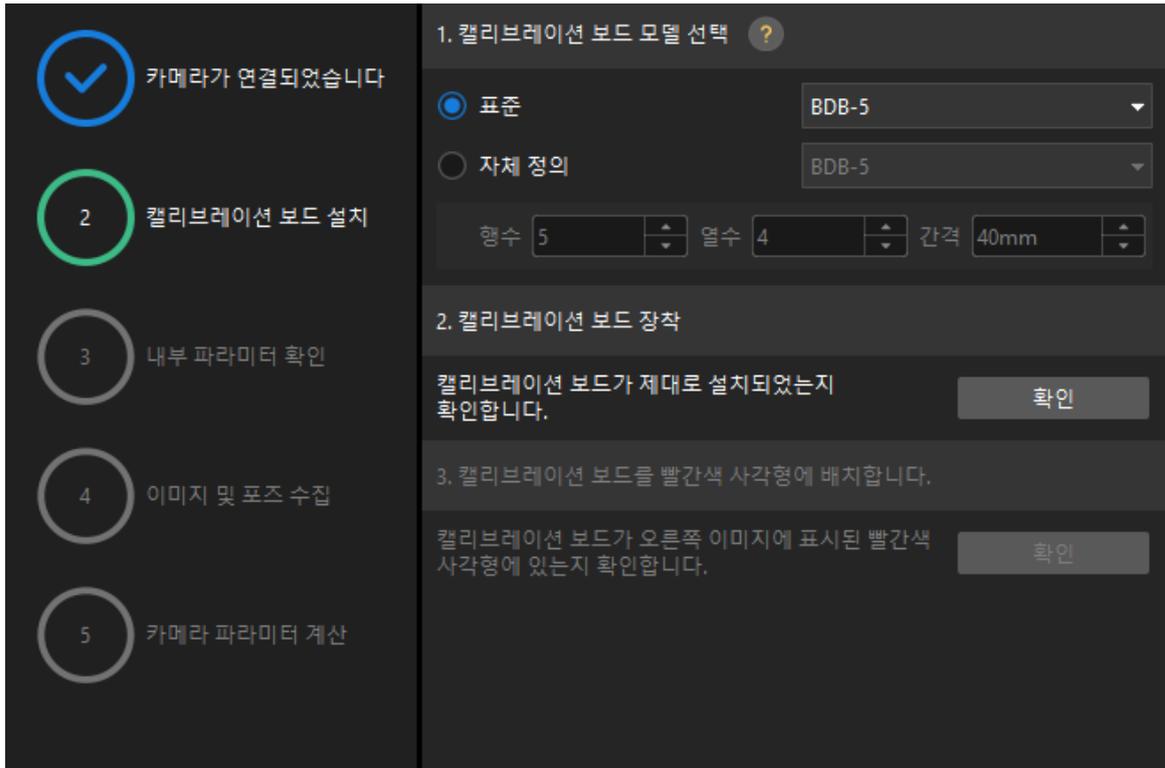
3. 이미지 뷰어에서 카메라가 캡처한 2D 맵과 뎀스 맵이 캘리브레이션 요구 사항을 충족하는지 확인한 다음 하단 표시줄에서 [ 다음 ] 버튼을 클릭합니다.



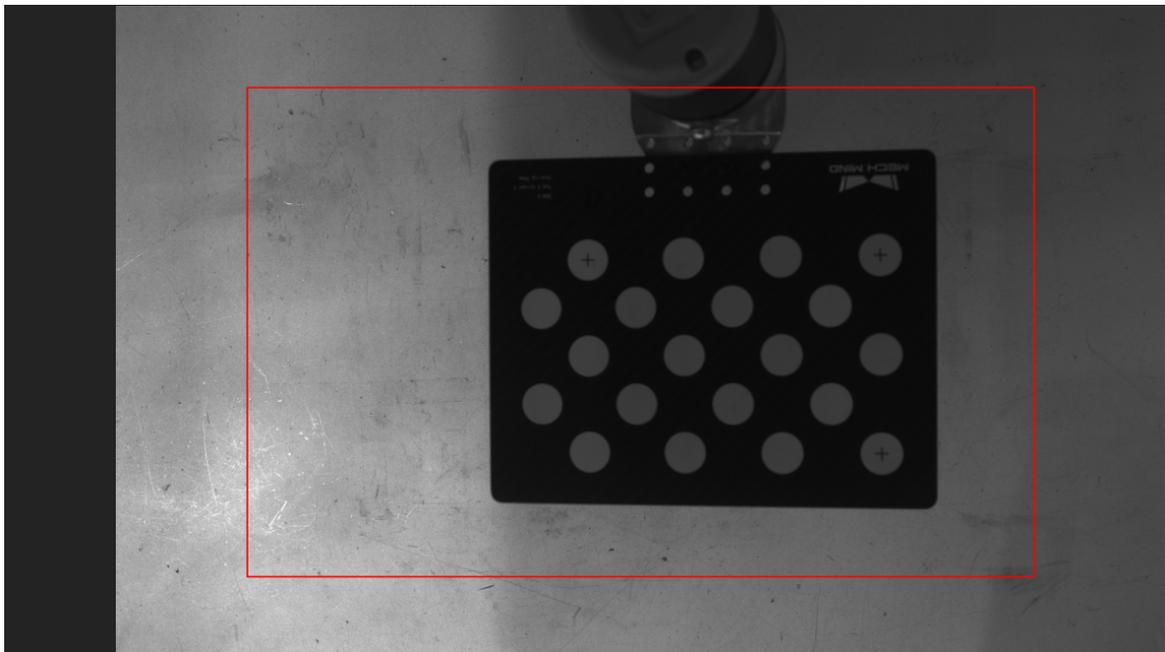
캡처한 이미지가 캘리브레이션 요구 사항을 충족하지 않으면 Mech-Eye Viewer 소프트웨어를 열어 **카메라의 2D 및 3D 노출 파라미터를 조정**하고 다시 캡처해야 합니다.

#### 캘리브레이션 보드 설치

1. 캘리브레이션 보드 설치 단계의 **1. 캘리브레이션 보드 유형 선택** 영역에서 **표준** 라디오 버튼을 선택한 후 캘리브레이션 보드 모델 레이블에 따라 해당 캘리브레이션 유형을 선택합니다.



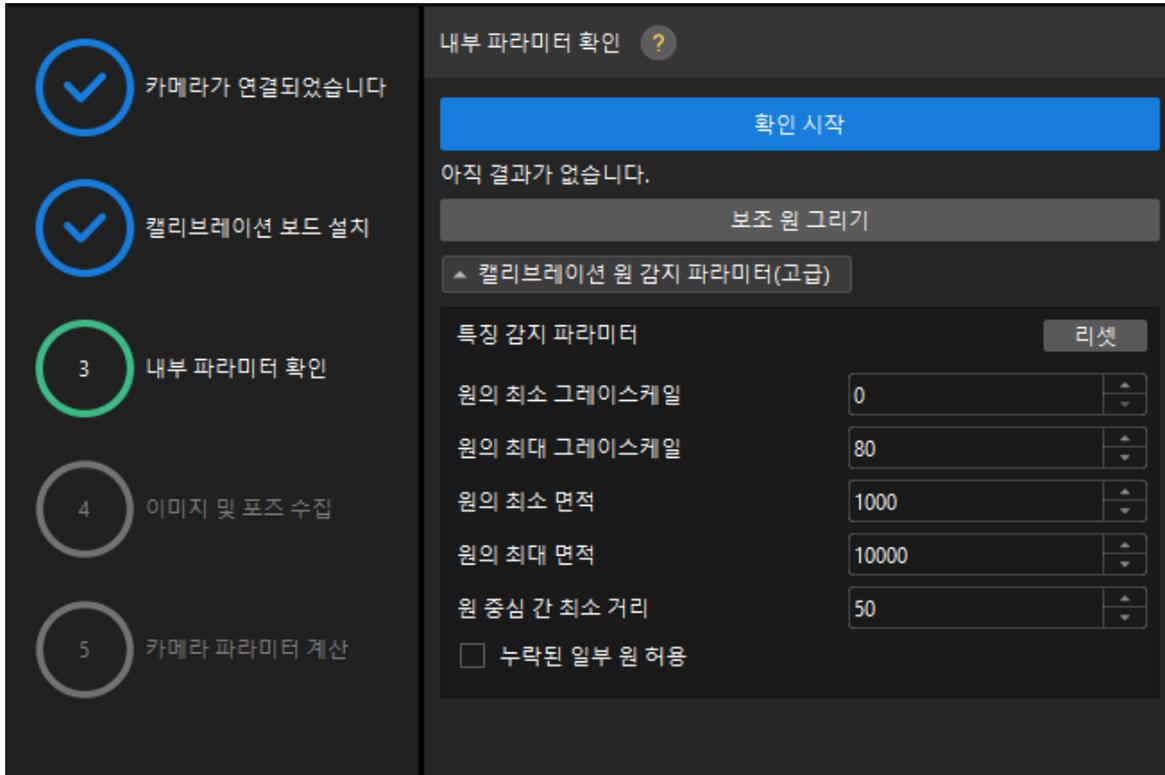
2. 캘리브레이션 보드가 작업 평면 중앙에 놓여 있는지 확인한 후 **2. 캘리브레이션 보드 설치** 영역에서 [ **확인** ] 버튼을 클릭합니다.
3. 캘리브레이션 보드가 카메라 시야의 중앙(빨간색 사각형 내)에 있는지 확인한 다음 **3.캘리브레이션 보드를 빨간색 사각형에 배치** 영역에서 [ **확인** ] 버튼을 클릭합니다.



4. 캘리브레이션 보드 관련 작업을 모두 완료한 후 하단 표시줄에서 [ **다음** ] 버튼을 클릭합니다.

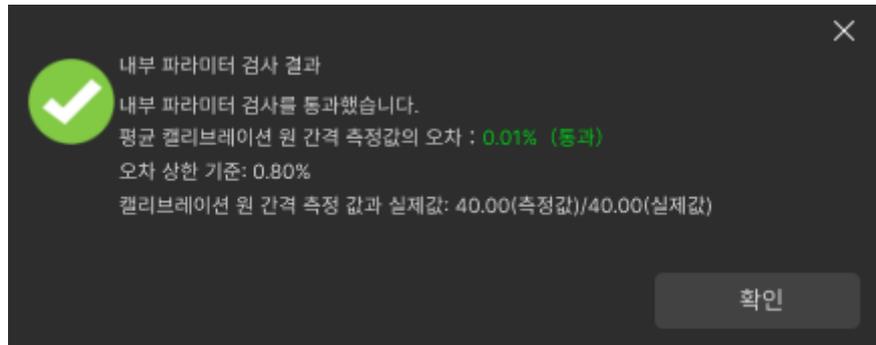
## 카메라 내부 파라미터를 검사하기

1. 내부 파라미터 검사 단계에서 [ 검사 시작 ] 버튼을 클릭합니다.



2. 카메라의 내부 파라미터 검사 결과를 확인합니다.

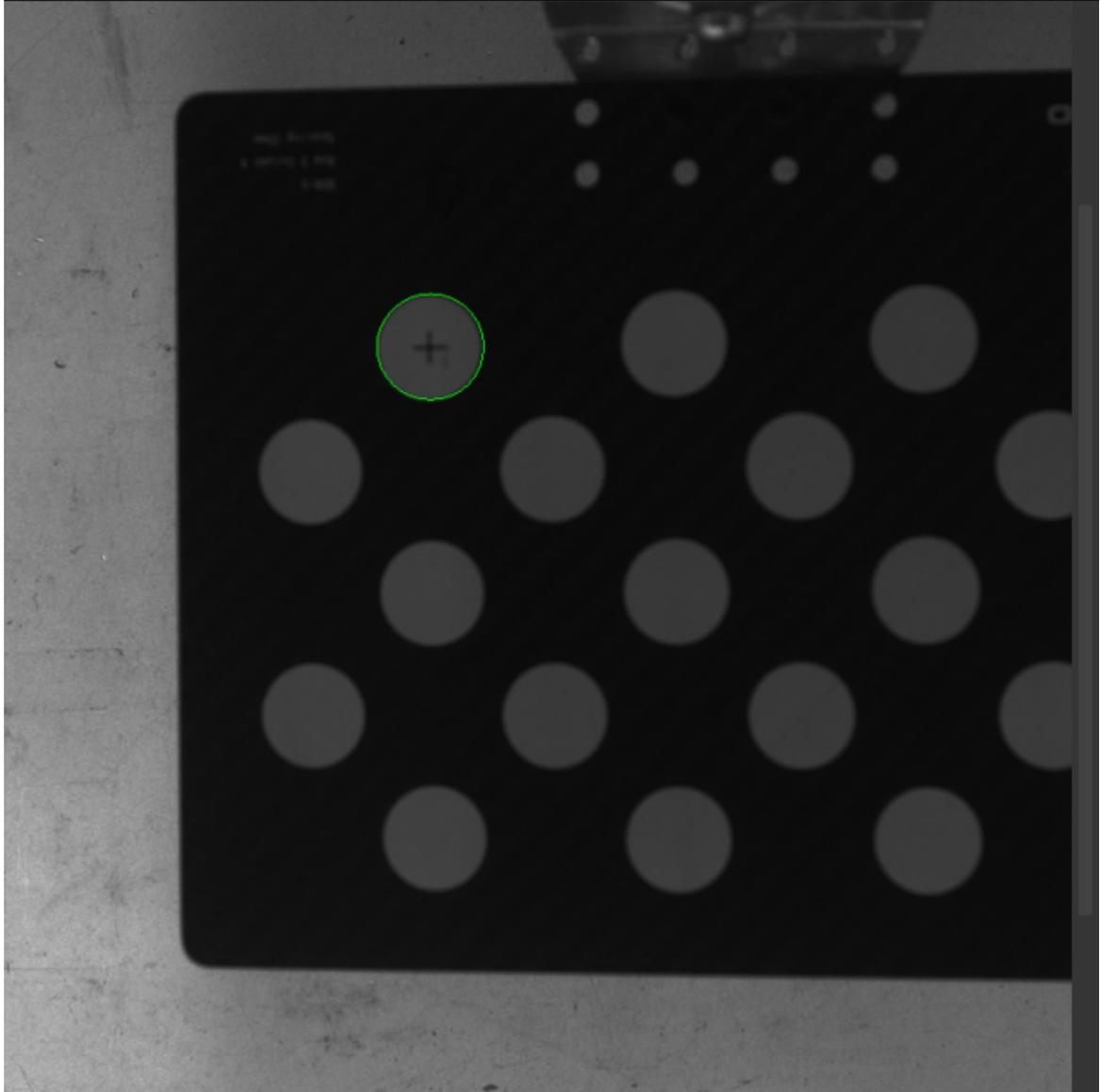
- 카메라의 내부 파라미터 검사에 성공하면 [ 확인 ]을 클릭한 후 [ 다음 ] 버튼을 클릭하십시오.



- 카메라의 내부 파라미터 검사에 실패하면 보조원을 그리거나 캘리브레이션 원을 수동으로 편집하여 파라미터를 감지 한 다음 [ 다시 검사 ] 버튼을 클릭하십시오.

## 보조원을 그리기

1. 보조원을 그리기로 선택한 경우 [ 보조원 그리기 ] 버튼을 클릭하십시오.
2. 오른쪽 이미지 뷰어 패널에서 마우스 오른쪽으로 캘리브레이션 보드 이미지를 클릭하고 해당 창 확인란을 언체크 한 다음 **Ctrl** 키를 누른 상태에서 롤러를 드래그하여 이미지를 적절한 크기로 조정하십시오.
3. 마우스 포인터를 캘리브레이션 원의 중간 줄 지점으로 이동하고 왼쪽 마우스 버튼을 누른 다음 보조원에 캘리브레이션 원을 완전히 포함시킨 다음 해제하십시오.



4. [ 다시 검사 ] 버튼을 클릭하고 카메라 내부 파라미터 검사가 통과되었는지 확인하십시오.

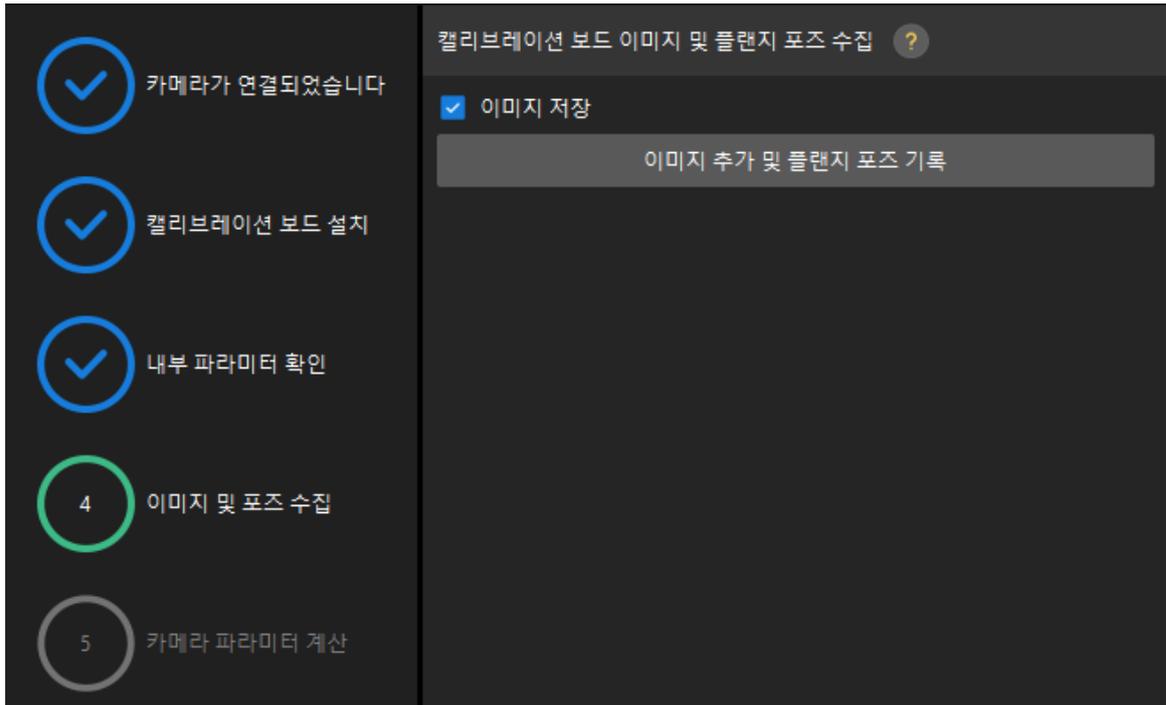
#### 감지 파라미터 수동으로 조정

감지 파라미터를 수동으로 조정할 경우 [ 캘리브레이션 원 감지 파라미터(고급) ]을 클릭하여 감지 파라미터의 값을 변경하면 됩니다.

캘리브레이션 원이 여전히 감지되지 않은 경우 실제 작업 조건에 따라 카메라 관련 파라미터를 조정해야 하며 카메라 파라미터는 [파라미터 조정](#) 내용을 참조하십시오.

#### 이미지 및 포즈 가져오기

1. 이미지 및 포즈 가져오기 단계에서 [ 이미지를 저장하기btn:[ ] ] 체크박스를 선택합니다.



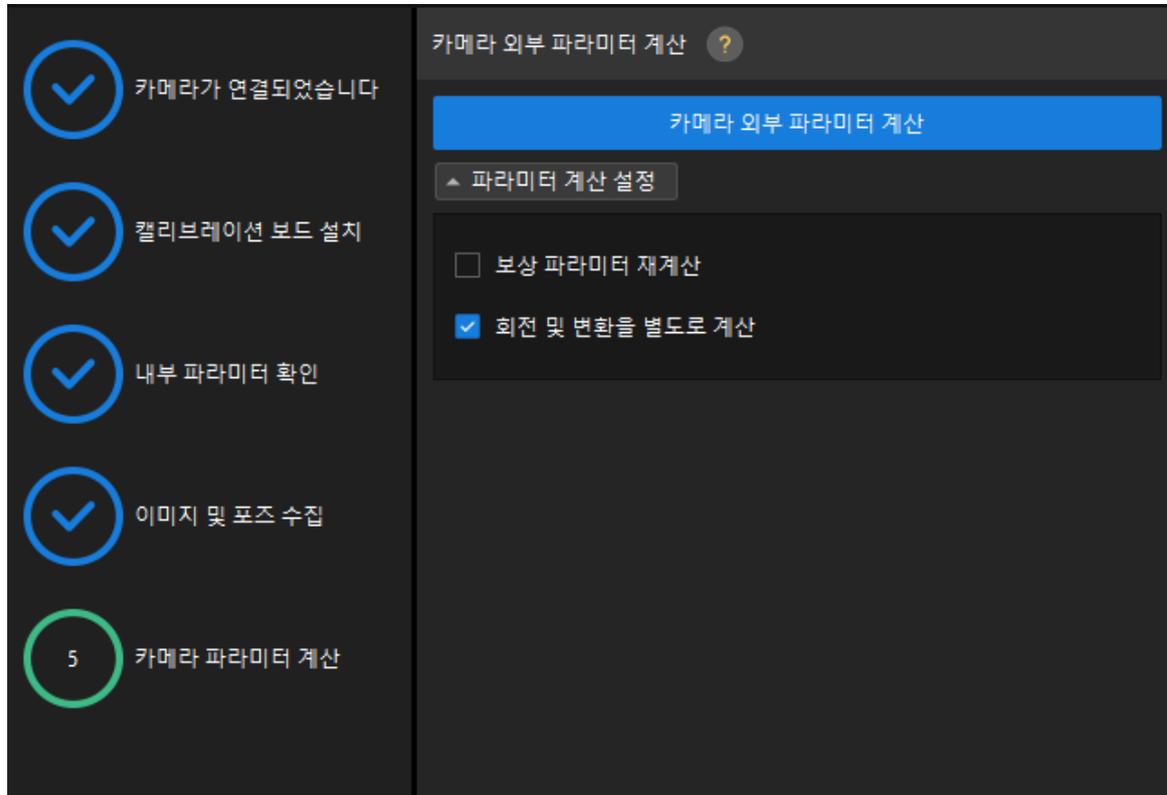
2. 다른 캘리브레이션 포인트로 이동하도록 로봇을 제어한 다음 [ 이미지 추가 및 플랜지 포즈 기록 ] 버튼을 클릭합니다.
3. 팝업창에서 로봇 플랜지의 포즈를 입력합니다.



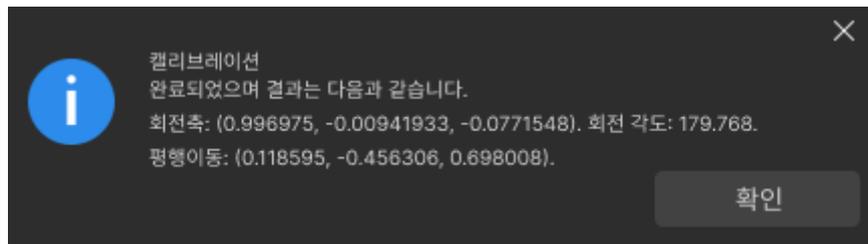
4. 현재 인식한 캘리브레이션 포인트가 요구 사항을 충족하는지 확인한 다음 하단 열의 [ 다음 ] 버튼을 클릭하십시오.

## 카메라 파라미터를 계산하기

1. 카메라 파라미터를 계산하기 단계에서 **회전 및 변환을 별도로 계산** 파라미터를 선택한 다음 [ **카메라 외부 파라미터 계산** ] 버튼을 클릭하십시오.



2. 팝업한 캘리브레이션 성공 다이얼로그 박스에서 [ **확인** ] 버튼을 클릭합니다.



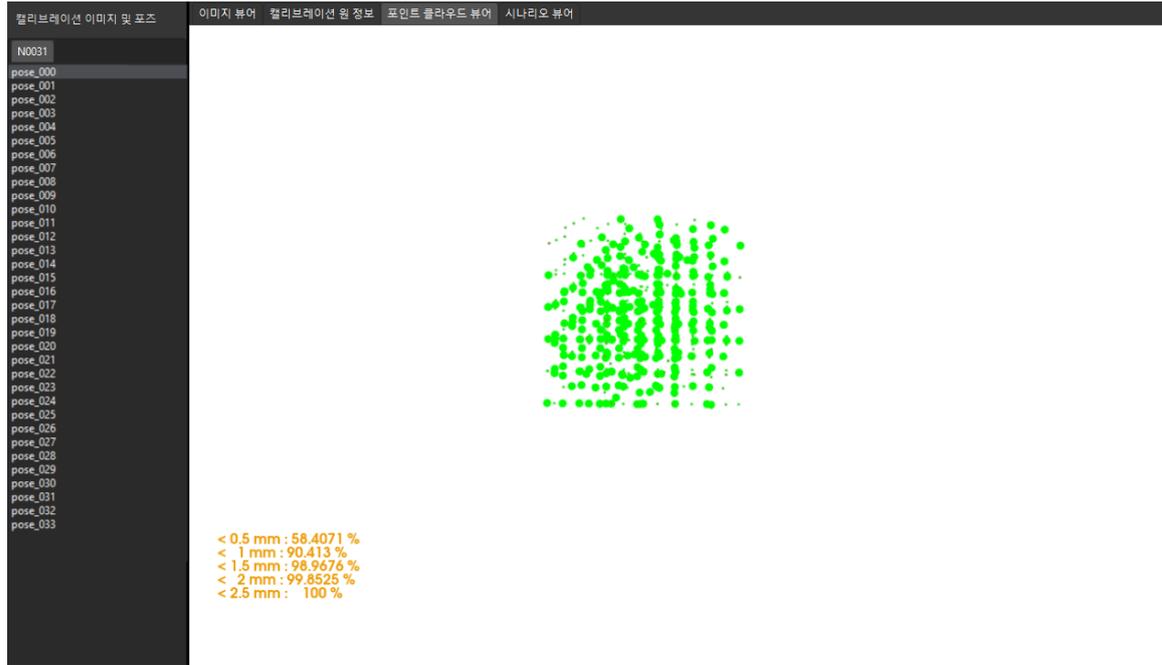
3. 오른쪽 **포인트 클라우드 뷰어** 패널에서 캘리브레이션 오차 포인트 클라우드를 확인합니다.



오차 포인트 클라우드는 각 캘리브레이션 포즈의 캘리브레이션 보드 원의 진릿값과 계산된 값 사이의 편차를 표시하는 데 사용됩니다.

4. 캘리브레이션 정확도가 예상 요구 사항을 충족하는지 확인합니다.

캘리브레이션의 정확도를 결정하기 위해 100%를 차지하는 오차 값을 찾으십시오. 예를 들어, 아래 그림의 정확도는 2.5mm 미만입니다.



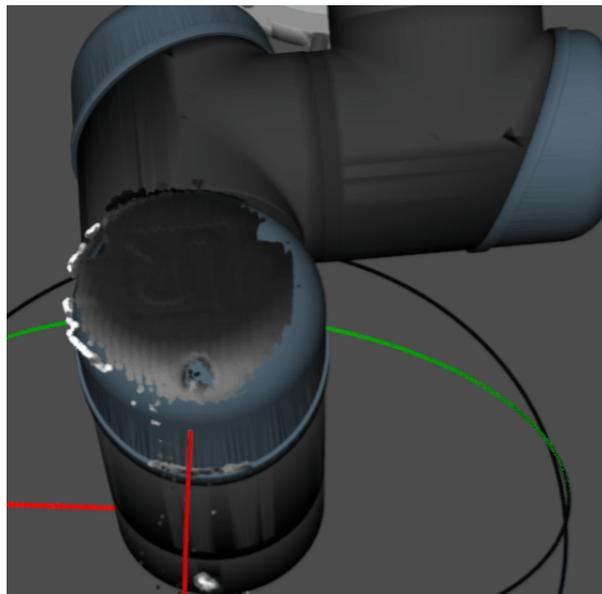
캘리브레이션 정확도를 개선해야 하는 경우 [캘리브레이션 결과 분석](#) 내용을 참조하십시오.

### 캘리브레이션 결과 검증 및 저장

카메라 외부 파라미터를 계산한 후 시나리오 뷰어에서 로봇 포인트 클라우드와 로봇 모델의 매칭 정도를 보고 캘리브레이션 결과를 대략적으로 판단할 수 있습니다. 구체적인 작업은 다음과 같습니다.

1. 로봇 팔을 카메라의 시야 내로 이동합니다.
2. 카메라 파라미터 계산 단계에서 [ 카메라 외부 파라미터 계산 ] 버튼을 클릭합니다. 이 작업은 카메라가 사진을 캡처하도록 트리거합니다.
3. 시나리오 뷰어를 클릭하여 로봇 포인트 클라우드와 로봇 모델 간의 매칭 정도를 확인합니다.

로봇의 포인트 클라우드가 로봇 모델과 대략적으로 매칭하면 캘리브레이션 결과를 사용할 수 있음을 의미합니다.





1. 로봇 모델은 실제 로봇의 모양과 완전히 매칭하지 않을 수 있으며 외부 파라미터를 미세 조정하기 위한 기준으로 사용할 수 없습니다.
2. 시나리오 뷰어 외에도 Mech-Viz 소프트웨어에서 로봇의 포인트 클라우드와 로봇 모델 간의 매칭 정도를 확인할 수 있습니다.

4. 하단 표시줄에서 [저장] 버튼을 클릭하고 팝업한 **캘리브레이션 파일 저장** 다이얼로그 박스에서 [확인] 버튼을 클릭합니다. 카메라 캘리브레이션 결과는 프로젝트의 "calibration" 디렉터리에 자동으로 저장됩니다.

이로써 캘리브레이션 프로세스가 완료됩니다.

#### 4.4.5. Eye in Hand 시나리오에서의 수동 캘리브레이션(캘리브레이션 보드의 여러 임의 포즈)

이 부분에서는 EIH(Eye in Hand) 시나리오에서 수동 캘리브레이션(캘리브레이션 보드의 여러 임의 포즈)을 완료하는 방법에 대해 설명합니다.

##### 캘리브레이션 사전 준비

핸드-아이 캘리브레이션을 수행하기 전에 다음 준비를 완료해야 합니다.

- Mech-Mind 비전 시스템 구축을 완료합니다.
- 캘리브레이션에 필요한 재료를 준비합니다.
- 캘리브레이션 보드 포인트 클라우드 이미징을 조정합니다.
- 로봇 및 인터페이스 구성을 완료합니다.

##### 비전 시스템 전체 구축 완료

Mech-Mind 비전 시스템 구축을 완료하려면 비전 시스템 구축 부분 내용을 참조하십시오.

핸드-아이 캘리브레이션에는 Mech-Eye Viewer, Mech-Center, Mech-Vision 및 Mech-Viz 소프트웨어가 필요합니다. 위의 소프트웨어가 설치되고 최신 버전으로 업데이트되었는지 확인하십시오.

##### 캘리브레이션에 필요한 재료를 준비

EIH 시나리오에서 수동 캘리브레이션을 하려면 캘리브레이션 보드를 사용해야 합니다.

다음 요구 사항에 따라 캘리브레이션 보드를 준비하십시오.

- 캘리브레이션 보드에서 도트가 잘 보이고 눈에 띄는 굵은 흔적이 없으며 캘리브레이션 보드가 뒤틀리지 않아야 합니다.
- EIH 시나리오에서 캘리브레이션 보드를 작업대의 중앙에 평평하게 놓으십시오.

또한 캘리브레이션 시작 전 로봇을 캘리브레이션 시작 포즈로 이동합니다.

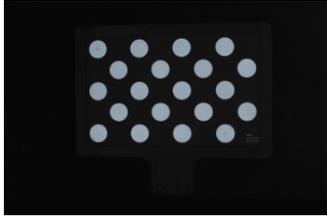
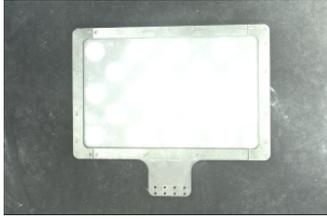
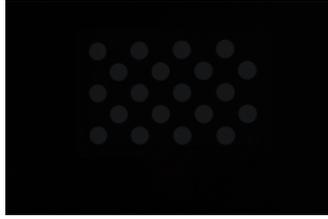
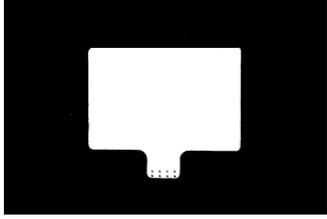
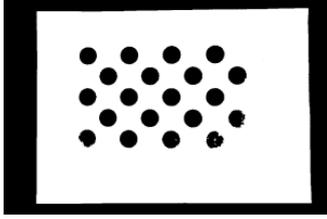
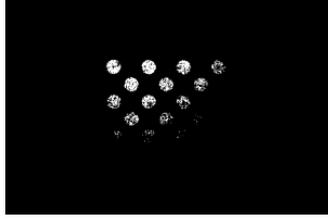
##### 캘리브레이션 보드 포인트 클라우드 이미징을 조정하기

1. Mech-Eye Viewer 소프트웨어를 열고 카메라 파라미터를 조정합니다.
2. 2D 파라미터를 조정함으로써 2D 맵의 캘리브레이션 보드가 뚜렷하고 노출 과다나 노출 부족 등 현상이 없도록 합니다.
3. 3D 파라미터를 조정함으로써 캘리브레이션 보드의 원의 포인트 클라우드를 완전하게 하고 **포인트 클라우드 후처리에서 포인트 클라우드 평활화 및 노이즈 제거를 일반으로 변경하는 것이 좋으며** 포인트 클라우드 변동 범위를 줄입니다.



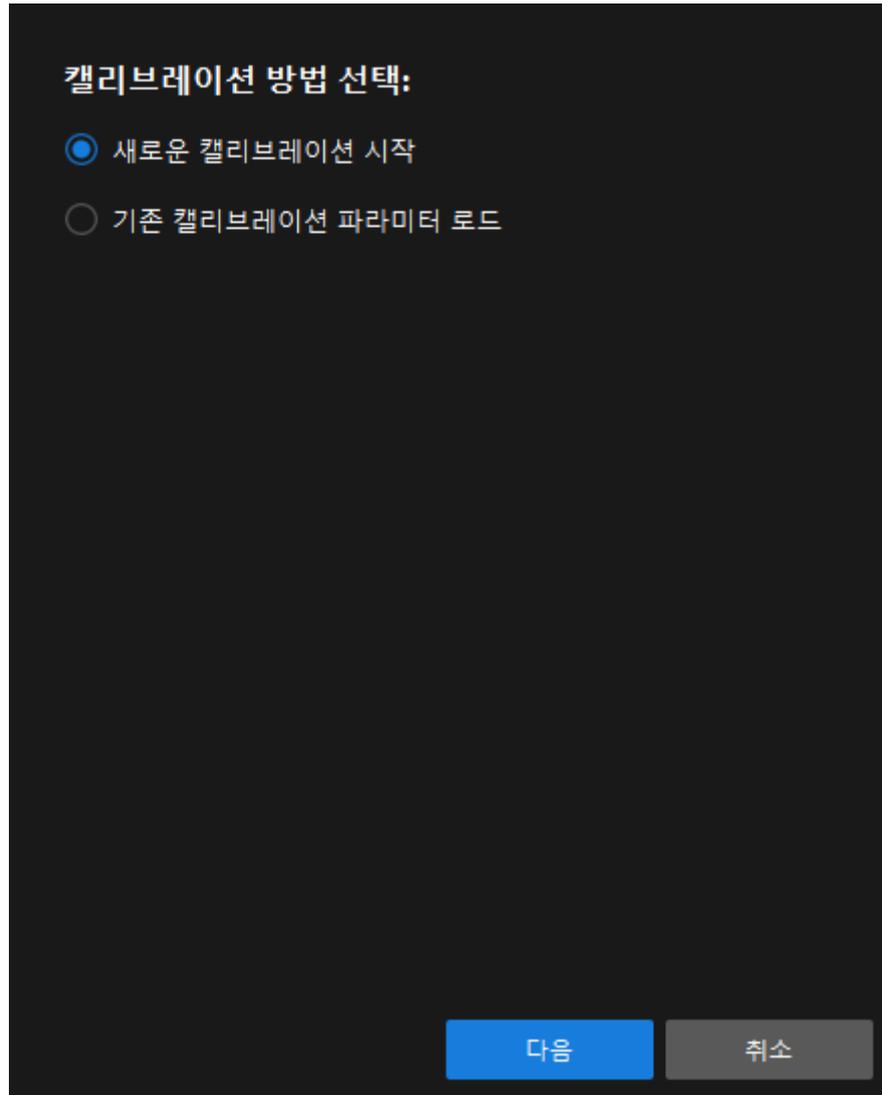
현장에 환경광이 복잡한 경우, 환경광이 2D 맵과 포인트 클라우드에 주는 영향을 줄이기 위해 광원을 차단하거나 보충하기 바랍니다.

4. 위의 단계를 완료하여 캘리브레이션 보드의 포인트 클라우드 이미징 품질이 요구 사항을 충족하는지 확인하십시오.

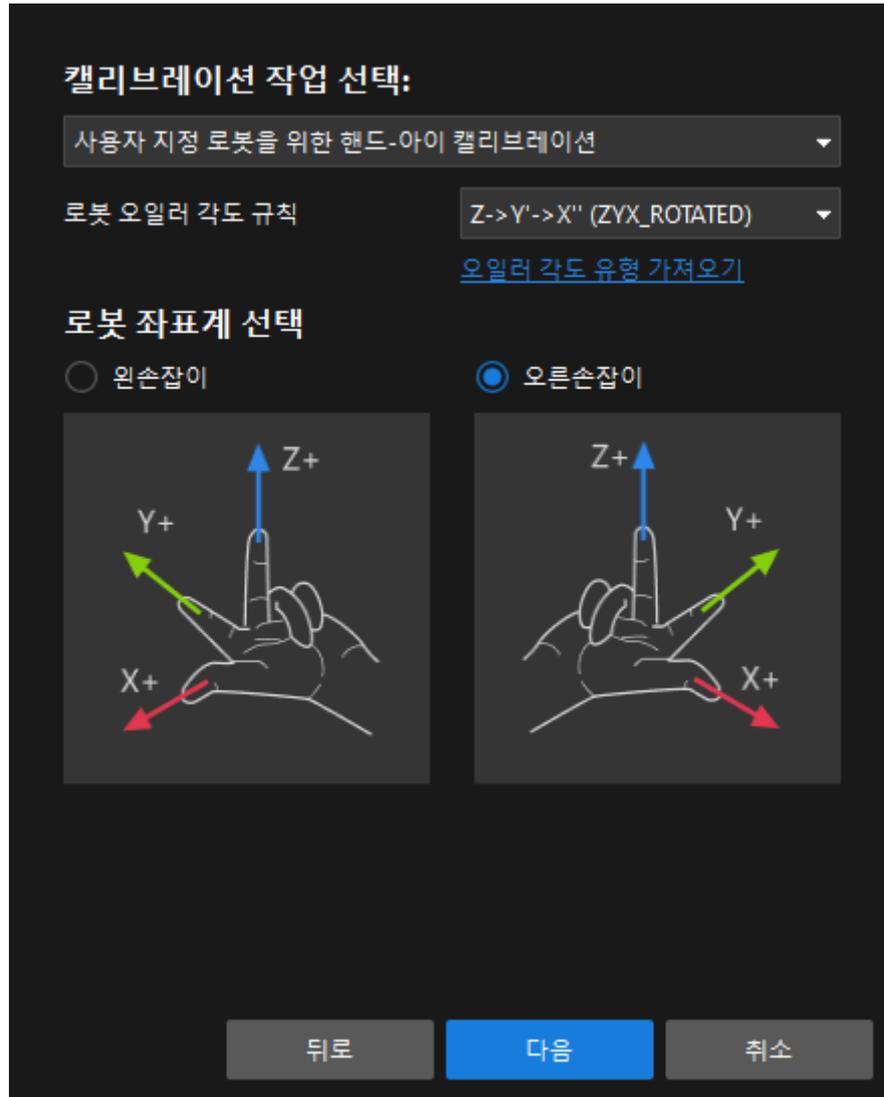
	정상	노출 과다	노출 부족
2D 맵			
포인트 클라우드			

#### 캘리브레이션 사전 구성

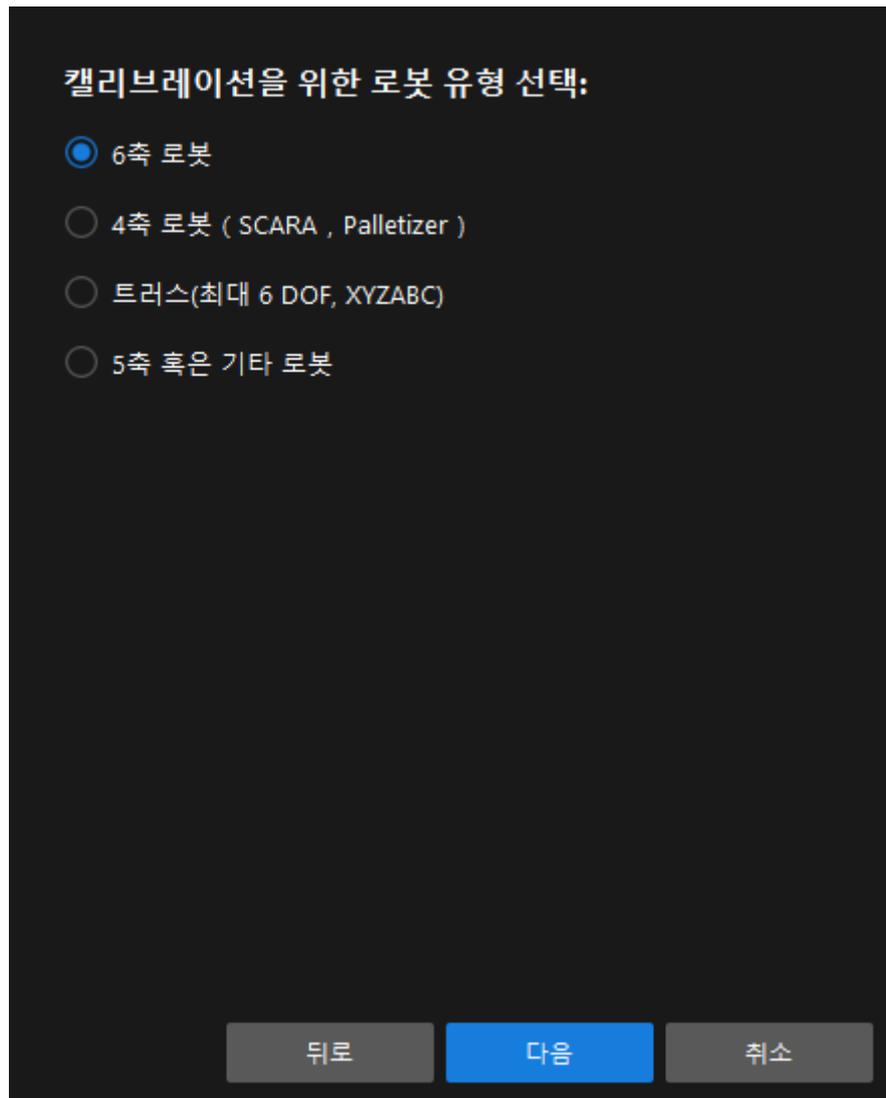
1. Mech-Vision 소프트웨어에서 톨 바의 [ 카메라 캘리브레이션(표준) ] 버튼을 클릭합니다.  
캘리브레이션 사전 구성 창이 팝업됩니다.
2. 캘리브레이션 방법 선택 창에서 새로운 캘리브레이션 시작 라디오 버튼을 선택한 후 [ 다음 ] 버튼을 클릭합니다.



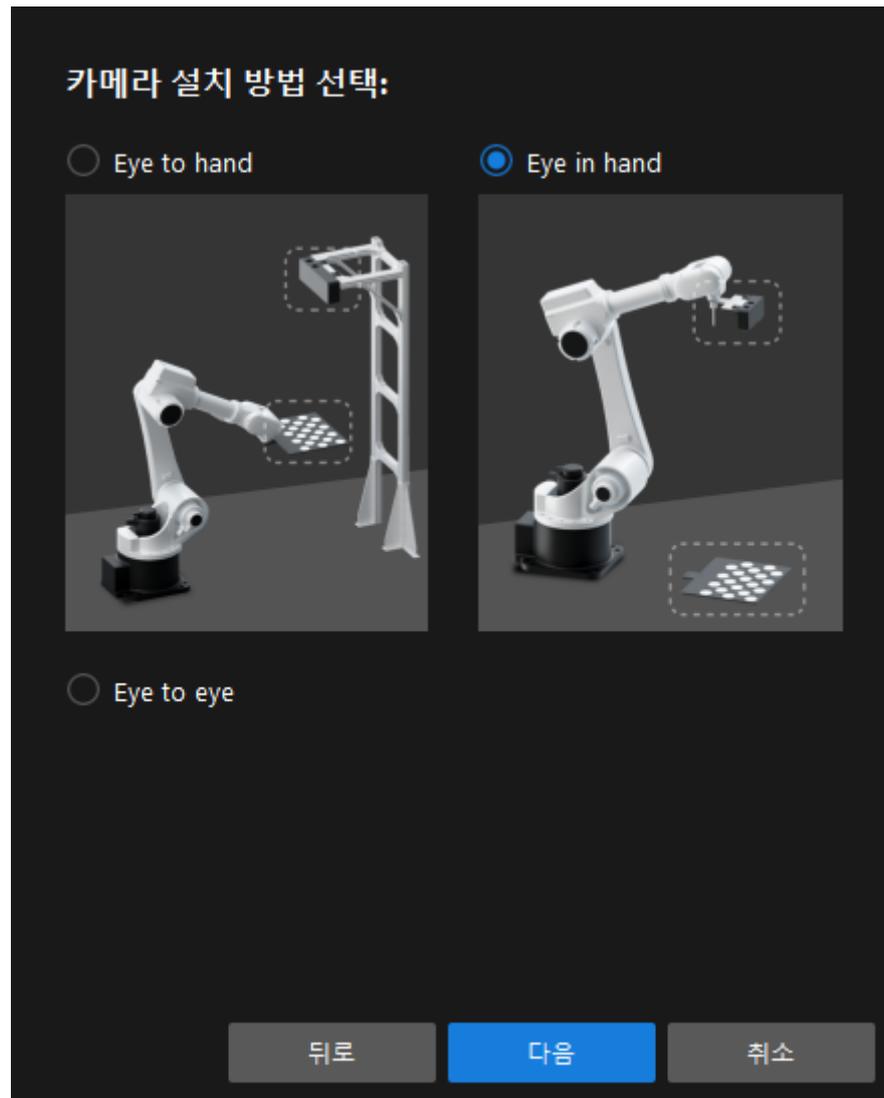
3. **캘리브레이션 작업 선택** 창의 드롭다운 목록 상자에서 **사용자 지정 로봇을 위한 핸드-아이 캘리브레이션**을 선택하고 **로봇 오일러 각도 유형** 파라미터를 지정하여 로봇 좌표계 유형을 선택한 후 [ **다음** ] 버튼을 클릭합니다.



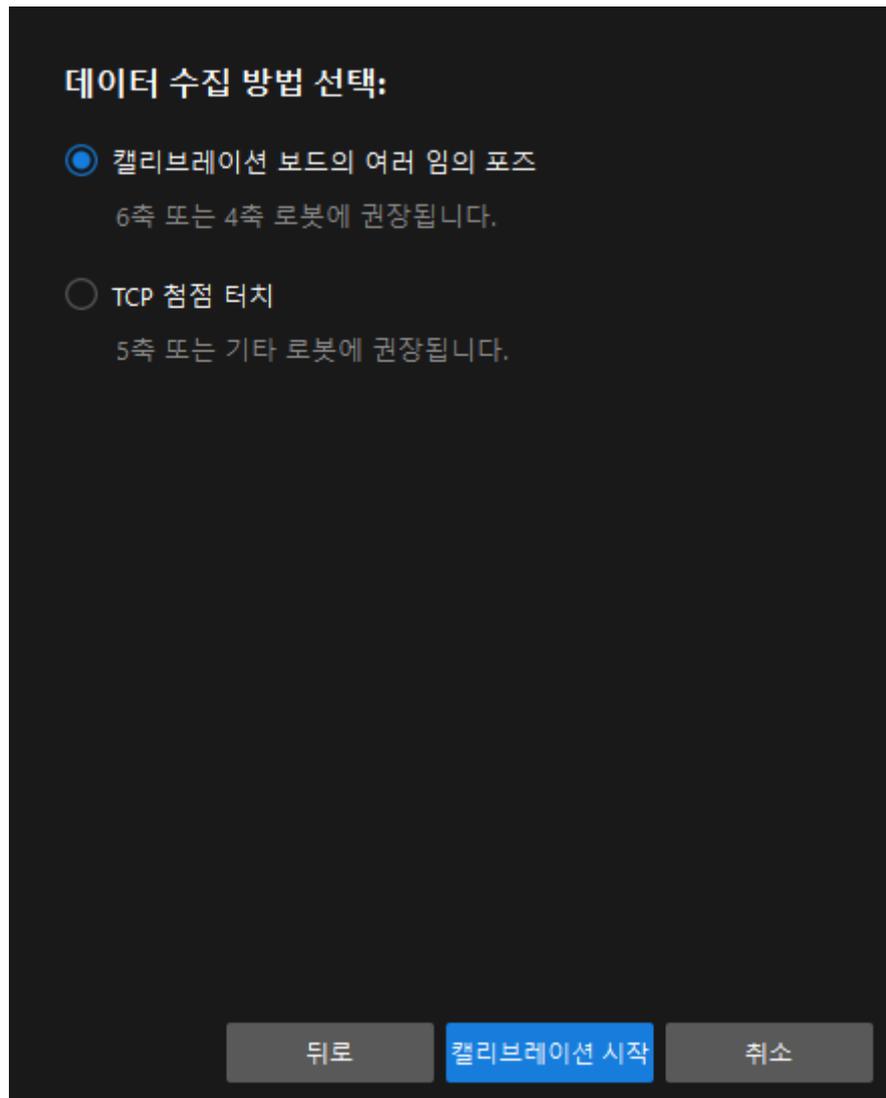
4. 캘리브레이션을 위한 로봇 유형 선택 창에서 로봇 유형에 따라 6축 로봇, 4축 로봇 (SCARA, Palletizer) 또는 5축 혹은 기타 로봇 라디오 상자를 선택한 후 [다음] 버튼을 클릭합니다.



5. 카메라 설치 방법 선택 창에서 **Eye in hand** 라디오 버튼을 선택한 후 [ 다음 ] 버튼을 클릭합니다.



6. 데이터 수집 방법 선택 창에서 캘리브레이션 보드의 여러 임의 포즈 라디오 버튼을 선택한 다음 [캘리브레이션 시작] 버튼을 클릭합니다. 캘리브레이션(Eye in Hand) 창이 나타납니다.

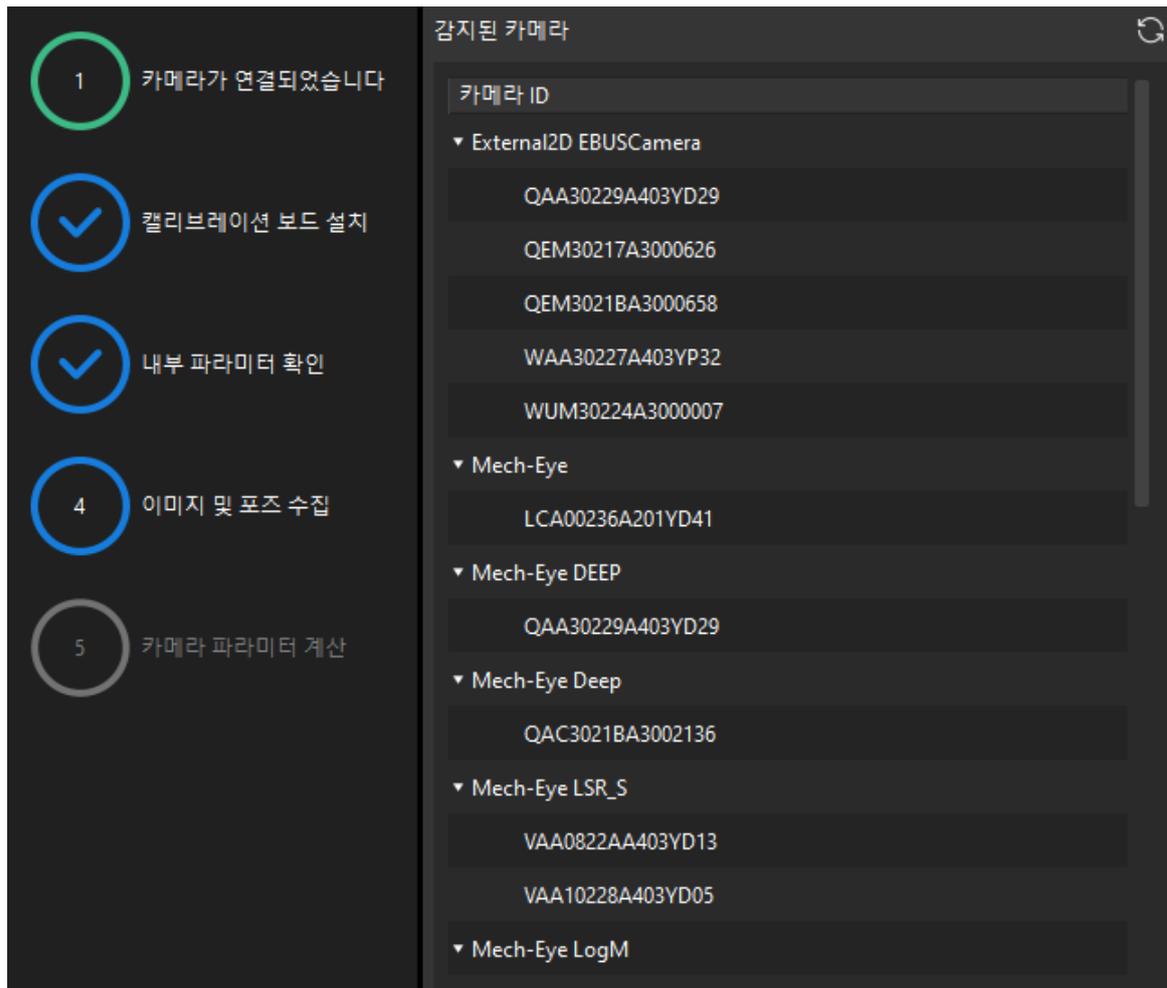


여기까지 캘리브레이션 사전 구성이 완료되었으며 정식 캘리브레이션 프로세스로 들어갑니다.

## 캘리브레이션 프로세스

### 카메라 연결

- 1. 카메라 연결** 단계에서 **감지된 카메라** 리스트에서 연결할 카메라를 선택한 다음  을 클릭하면 카메라가 연결되며 또는 직접 연결할 카메라를 더블클릭합니다.



2. 카메라를 연결한 후 [연속 캡처] 또는 [한번 캡처] 버튼을 선택할 수 있습니다.



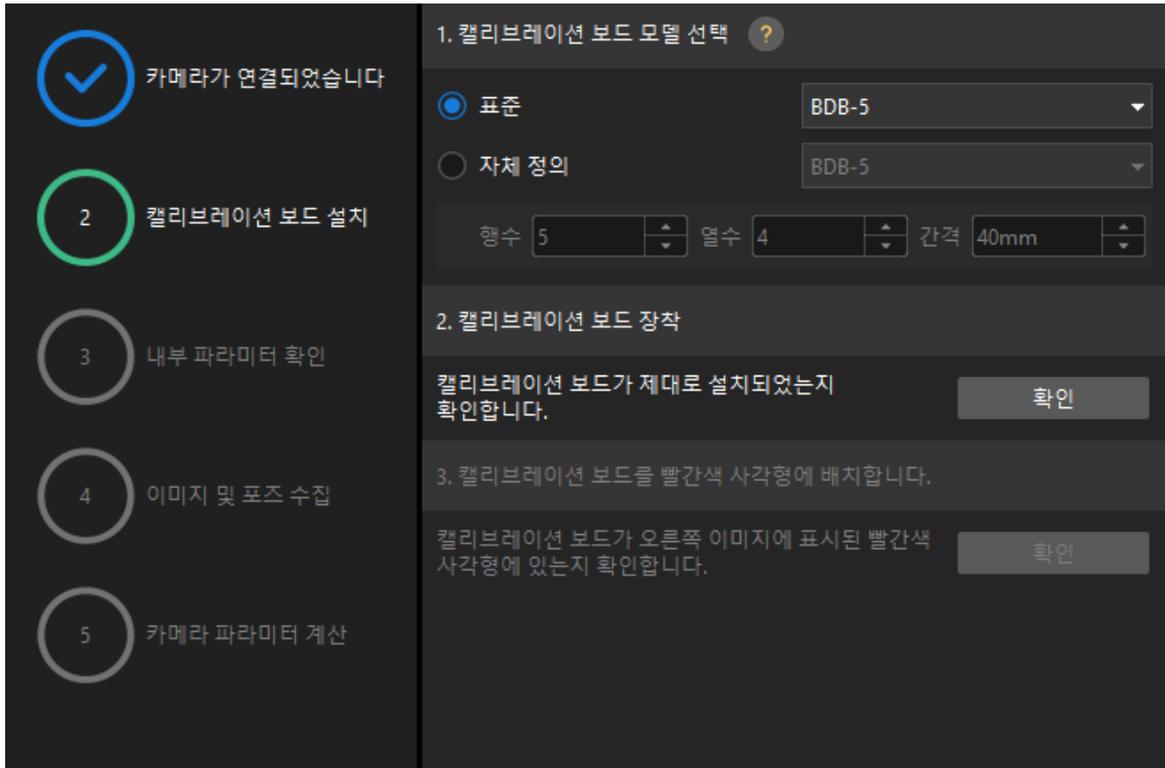
3. 이미지 뷰어에서 카메라가 캡처한 2D 맵과 뎀스 맵이 캘리브레이션 요구 사항을 충족하는지 확인한 다음 하단 표시줄에서 [ 다음 ] 버튼을 클릭합니다.



캡처한 이미지가 캘리브레이션 요구 사항을 충족하지 않으면 Mech-Eye Viewer 소프트웨어를 열어 **카메라의 2D 및 3D 노출 파라미터를 조정**하고 다시 캡처해야 합니다.

#### 캘리브레이션 보드 설치

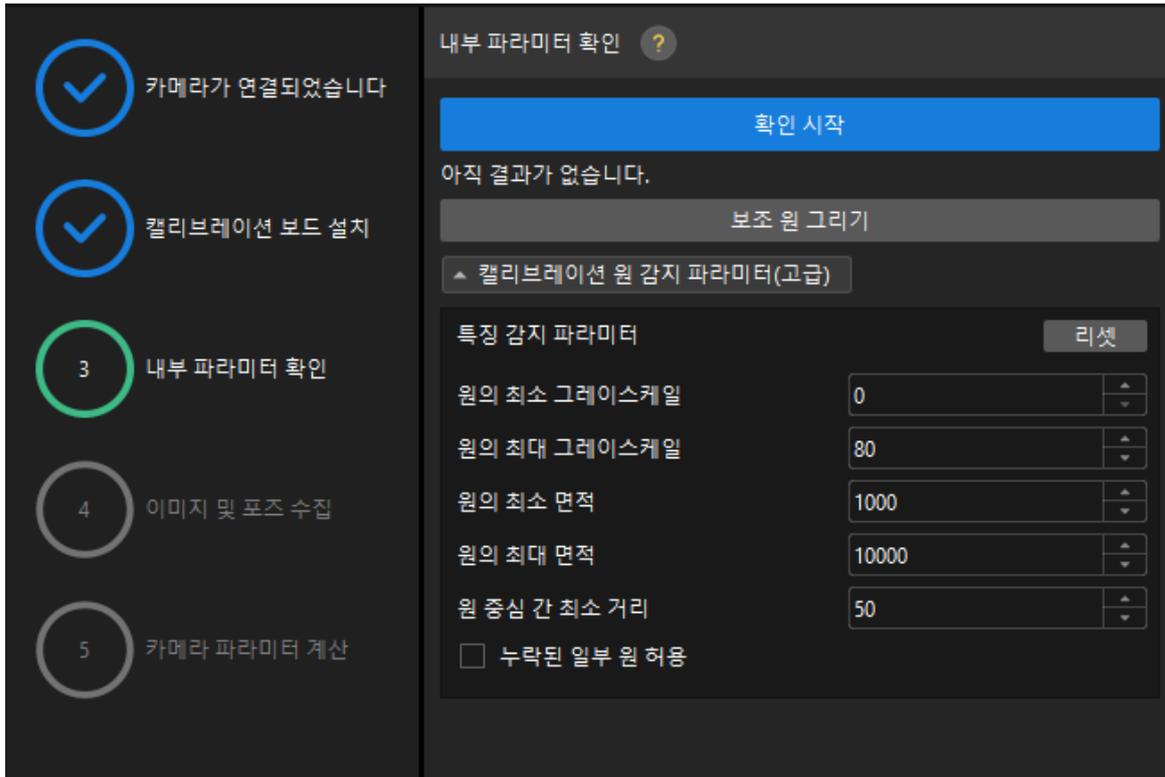
1. 캘리브레이션 보드 설치 단계의 **1. 캘리브레이션 보드 유형 선택** 영역에서 **표준** 라디오 버튼을 선택한 후 캘리브레이션 보드 모델 레이블에 따라 해당 캘리브레이션 유형을 선택합니다.



2. 캘리브레이션 보드가 작업 평면 중앙에 놓여 있는지 확인한 후 **2. 캘리브레이션 보드 설치** 영역에서 [ **확인** ] 버튼을 클릭합니다.
3. 캘리브레이션 보드가 카메라 시야의 중앙(빨간색 사각형 내)에 있는지 확인한 다음 **3.캘리브레이션 보드를 빨간색 사각형에 배치**영역에서 [ **확인** ] 버튼을 클릭합니다.
4. 캘리브레이션 보드 관련 작업을 모두 완료한 후 하단 표시줄에서 [ **다음** ] 버튼을 클릭합니다.

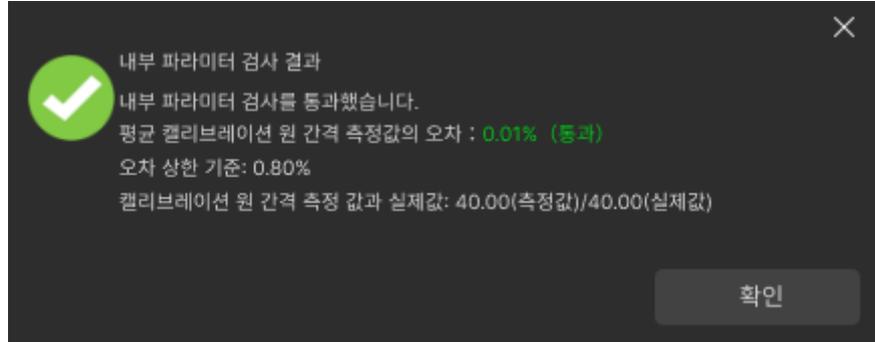
#### 카메라 내부 파라미터를 검사하기

1. 내부 파라미터 검사 단계에서 [ **검사 시작** ] 버튼을 클릭합니다.



## 2. 카메라의 내부 파라미터 검사 결과를 확인합니다.

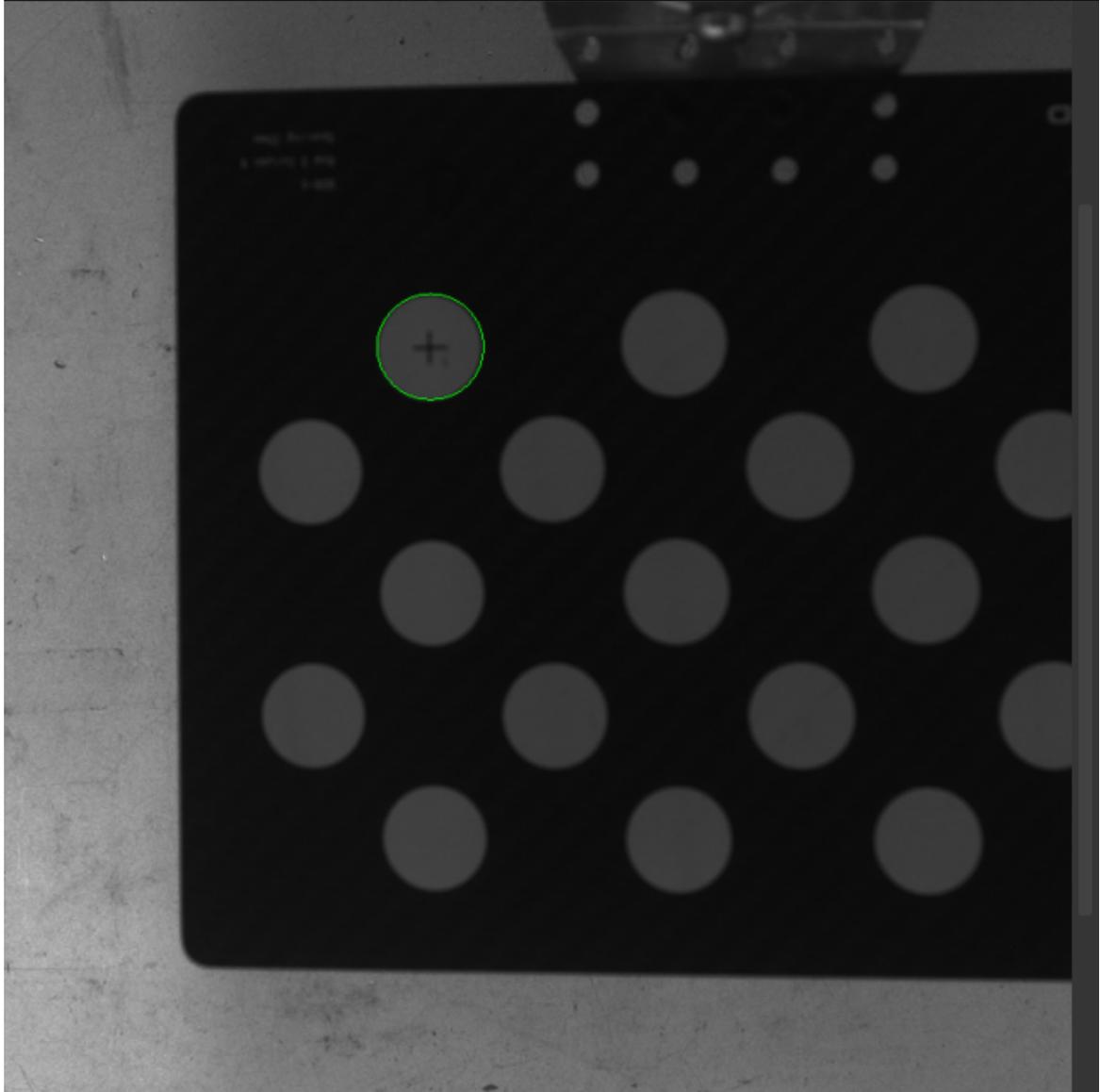
- 카메라의 내부 파라미터 검사에 성공하면 [확인]을 클릭한 후 [다음] 버튼을 클릭하십시오.



- 카메라의 내부 파라미터 검사에 실패하면 보조원을 그리거나 캘리브레이션 원을 수동으로 편집하여 파라미터를 감지 한 다음 [다시 검사] 버튼을 클릭하십시오.

### 보조원을 그리기

1. 보조원을 그리기로 선택한 경우 [보조원 그리기] 버튼을 클릭하십시오.
2. 오른쪽 이미지 뷰어 패널에서 마우스 오른쪽으로 캘리브레이션 보드 이미지를 클릭하고 해당 창 확인란을 언체크 한 다음 **Ctrl** 키를 누른 상태에서 롤러를 드래그하여 이미지를 적절한 크기로 조정하십시오.
3. 마우스 포인터를 캘리브레이션 원의 중간 줄 지점으로 이동하고 왼쪽 마우스 버튼을 누른 다음 보조원에 캘리브레이션 원을 완전히 포함시킨 다음 해제하십시오.



4. [ 다시 검사 ] 버튼을 클릭하고 카메라 내부 파라미터 검사가 통과되었는지 확인하십시오.

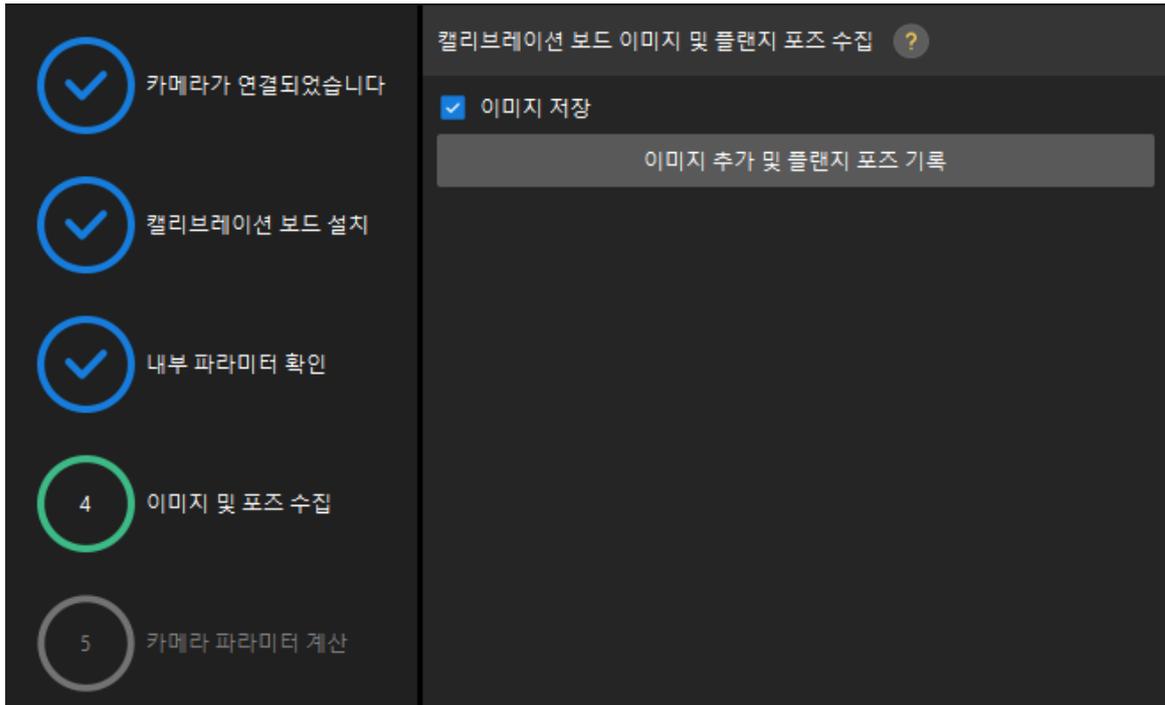
#### 감지 파라미터 수동으로 조정

감지 파라미터를 수동으로 조정할 경우 [ 캘리브레이션 원 감지 파라미터(고급) ]을 클릭하여 감지 파라미터의 값을 변경하면 됩니다.

캘리브레이션 원이 여전히 감지되지 않은 경우 실제 작업 조건에 따라 카메라 관련 파라미터를 조정해야 하며 카메라 파라미터는 [파라미터 조정](#) 내용을 참조하십시오.

#### 이미지 및 포즈 가져오기

1. 이미지 및 포즈 가져오기 단계에서 이미지를 저장하기 체크박스를 선택합니다.



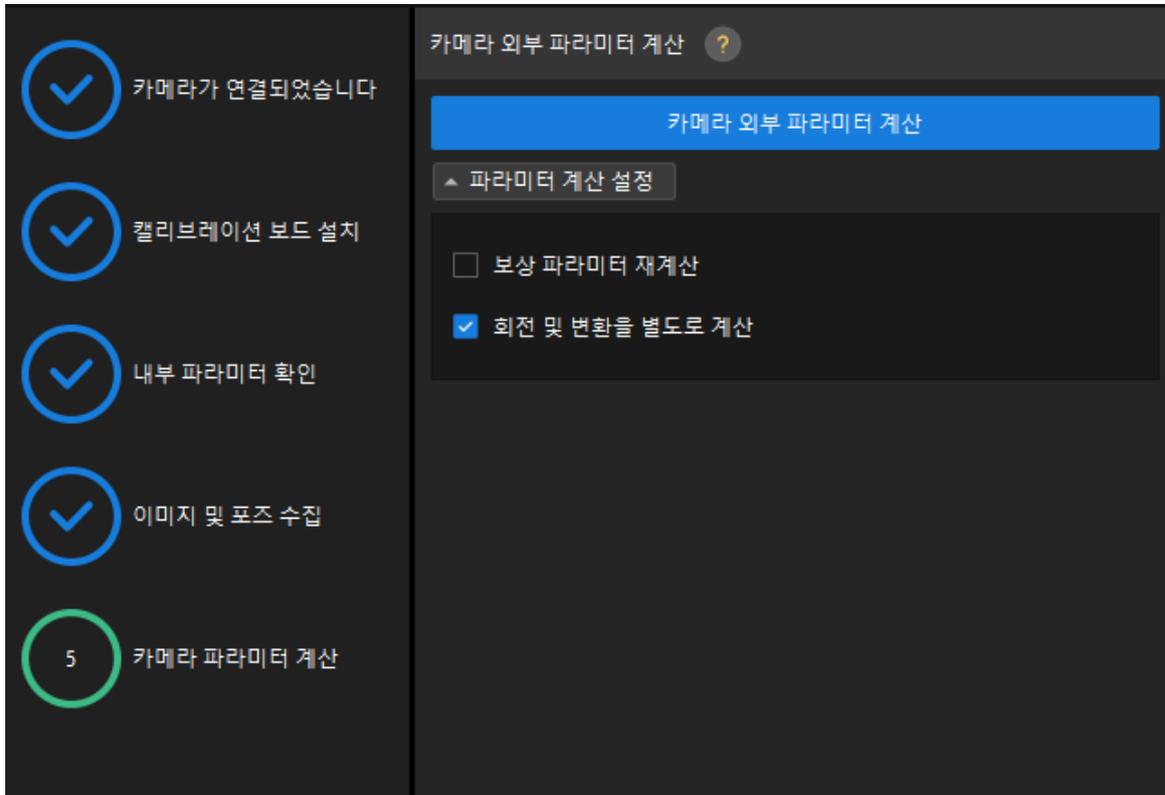
2. 다른 캘리브레이션 포인트로 이동하도록 로봇을 제어한 다음 [ 이미지 추가 및 플랜지 포즈 기록 ] 버튼을 클릭합니다.
3. 팝업창에서 로봇 플랜지의 포즈를 입력합니다.



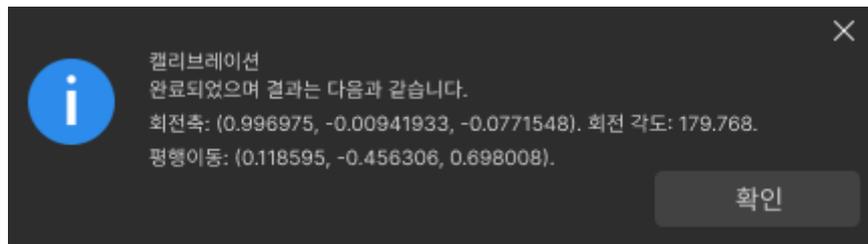
4. 현재 인식한 캘리브레이션 포인트가 요구 사항을 충족하는지 확인한 다음 하단 열의 [ 다음 ] 버튼을 클릭하십시오.

## 카메라 파라미터를 계산하기

1. 카메라 파라미터를 계산하기 단계에서 **회전 및 변환을 별도로 계산** 파라미터를 선택한 다음 [ **카메라 외부 파라미터 계산** ] 버튼을 클릭하십시오.



2. 팝업한 캘리브레이션 성공 다이얼로그 박스에서 [ **확인** ] 버튼을 클릭합니다.



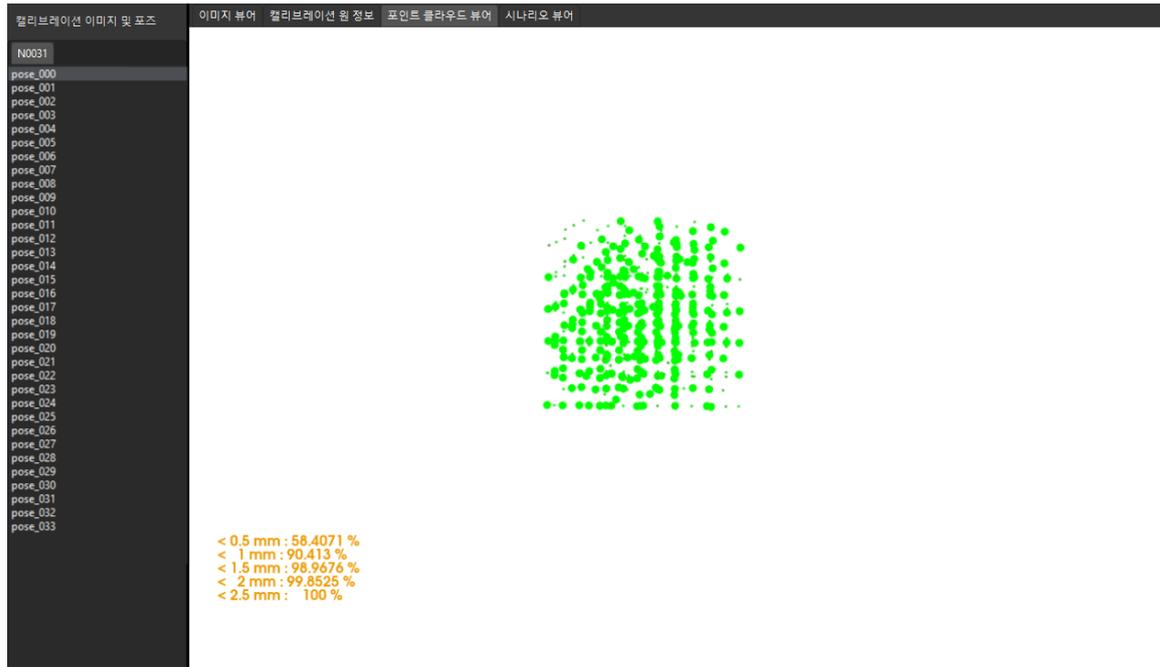
3. 오른쪽 **포인트 클라우드 뷰어** 패널에서 캘리브레이션 오차 포인트 클라우드를 확인합니다.



오차 포인트 클라우드는 각 캘리브레이션 포즈의 캘리브레이션 보드 원의 진릿값과 계산된 값 사이의 편차를 표시하는 데 사용됩니다.

4. 캘리브레이션 정확도가 예상 요구 사항을 충족하는지 확인합니다.

캘리브레이션의 정확도를 대략적으로 확인하기 위해 100%를 차지하는 오차 값을 찾으십시오. 예를 들어, 아래 그림의 정확도는 2.5mm 미만입니다.



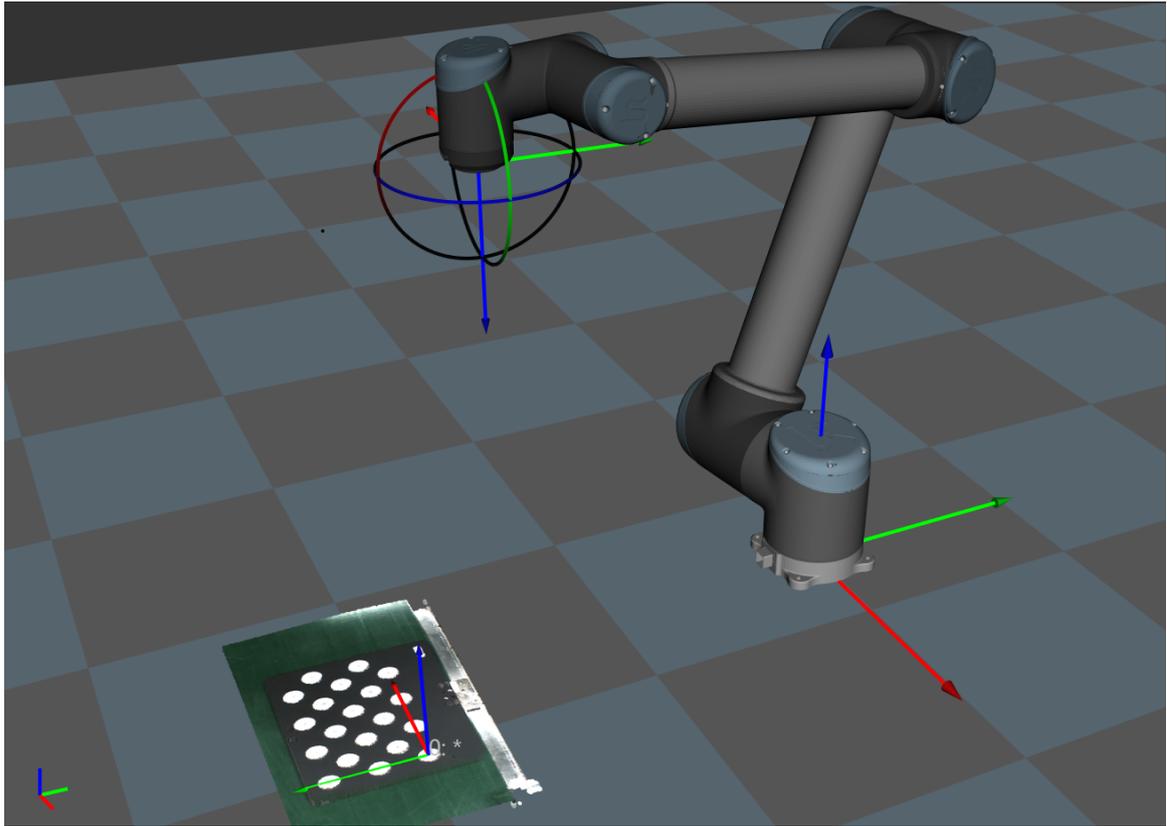
캘리브레이션 정확도를 개선해야 하는 경우 [캘리브레이션 결과 분석](#) 내용을 참조하십시오.

### 캘리브레이션 결과 검증 및 저장

카메라 외부 파라미터를 계산한 후 시나리오 뷰어에서 고정점을 기준으로 하는 캘리브레이션 보드 포인트 클라우드(캘리브레이션 원 십자의 중심점)의 움직임을 보고 캘리브레이션 결과를 대략적으로 판단할 수 있습니다. 구체적인 작업은 다음과 같습니다.

1. 캘리브레이션 보드를 고정된 위치에 놓습니다.
2. Mech-Viz 소프트웨어를 열고 고정점을 추가하고 캘리브레이션 보드에 있는 캘리브레이션 원 중 하나의 교차 중심점과 겹치도록 합니다.
  - a. 스텝 라이브러리에서 "이동" 스텝을 찾아 프로젝트 편집 영역으로 드래그합니다.
  - b. 이 스텝을 선택한 다음 파라미터 편집 영역에서 **이동 목표 유형** 파라미터를 "작업물 포즈"로 설정하고 포즈의 X/Y/Z 좌표를 조정하여 포즈가 캘리브레이션 보드의 캘리브레이션 원 중 하나의 교차 중심점과 겹치도록 합니다.
3. 로봇을 제어하여 카메라 포즈를 여러 번 변경하고 캘리브레이션 프로세스의 **카메라 파라미터를 계산하기** 스텝에서 [ **외부 파라미터 재계산** ] 버튼을 클릭합니다. 이 작업은 카메라가 사진을 캡처하도록 트리거합니다.
4. 시나리오 뷰어에서 캘리브레이션 보드의 포인트 클라우드가 고정점을 기준으로 분명히 움직이는지 여부를 관찰합니다.

캘리브레이션 보드의 포인트 클라우드에 고정점에 대한 명확한 움직임이 없으면 캘리브레이션 결과를 사용할 수 있음을 의미합니다.



1. EIH 시나리오에서 로봇이 연결되지 않은 상태에서 실시간 로봇 포즈를 획득할 수 없습니다. 로봇 현재의 촬영 포인트 포즈를 입력해야 Mech-Viz에서 표시된 포인트 클라우드를 확인할 수 있습니다.
2. 시나리오 뷰어 외에도 Mech-Viz 소프트웨어에서 캘리브레이션 보드의 포인트 클라우드가 고정점에 대해 명확한 움셋을 가지고 있는지 여부를 확인할 수 있습니다.

1. 하단 표시줄에서 [ 저장 ] 버튼을 클릭하고 팝업한 **캘리브레이션 파일 저장** 다이얼로그 박스에서 [ 확인 ] 버튼을 클릭합니다. 카메라 캘리브레이션 결과는 프로젝트의 "calibration" 디렉터리에 자동으로 저장됩니다.

이로써 캘리브레이션 프로세스가 완료됩니다.

#### 4.4.6. Eye to Hand 시나리오에서의 수동 캘리브레이션(TCP 끝단점 터치)

이 부분에서는 ETH(Eye to Hand) 시나리오에서 수동 캘리브레이션(TCP 끝단점 터치)을 완료하는 방법에 대해 설명합니다.

##### 캘리브레이션 사전 준비

핸드-아이 캘리브레이션을 수행하기 전에 다음 준비를 완료해야 합니다.

- Mech-Mind 비전 시스템 구축을 완료합니다.
- 캘리브레이션에 필요한 재료를 준비합니다.
- 캘리브레이션 보드 포인트 클라우드 이미징을 조정합니다.
- 로봇 및 인터페이스 구성을 완료합니다.

## 비전 시스템 전체 구축 완료

Mech-Mind 비전 시스템 구축을 완료하려면 비전 시스템 구축 부분 내용을 참조하십시오.

핸드-아이 캘리브레이션에는 Mech-Eye Viewer, Mech-Center, Mech-Vision 및 Mech-Viz 소프트웨어가 필요합니다. 위의 소프트웨어가 설치되고 최신 버전으로 업데이트되었는지 확인하십시오.

### 캘리브레이션에 필요한 재료를 준비

ETH 시나리오에서 수동 캘리브레이션을 하려면 캘리브레이션 보드와 TCP 끝단점을 사용해야 합니다.

다음 요구 사항에 따라 캘리브레이션 보드를 준비하십시오.

- 캘리브레이션 보드에서 도트가 잘 보이고 눈에 띄는 굵은 흔적이 없으며 캘리브레이션 보드가 뒤틀리지 않아야 합니다.
- ETH 시나리오에서 TCP 끝단점 터치 방식으로 캘리브레이션 데이터를 수집하는 경우 캘리브레이션 보드를 작업 평면 중앙에 평평하게 놓으십시오.

TCP 끝단점은 로봇 플랜지 중앙에 설치해야 합니다. 로봇 플랜지에 분리 불가능한 그리퍼를 장착한 경우에 TCP 끝단점을 그리퍼에 직접 고정하면 됩니다.

또한 캘리브레이션 시작 전 로봇을 캘리브레이션 시작 포즈로 이동하여 카메라가 캘리브레이션 보드 이미지를 캡처하는 데 방해가 되지 않도록 하십시오.

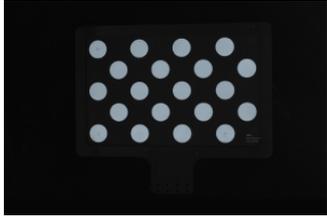
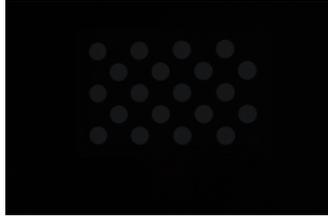
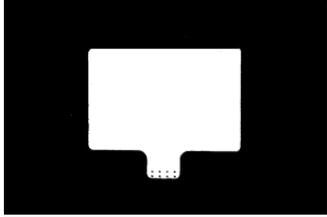
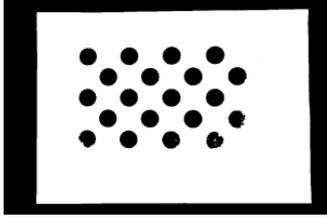
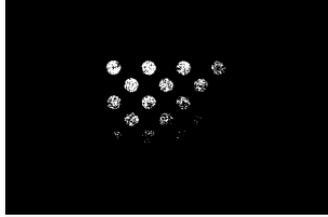
### 캘리브레이션 보드 포인트 클라우드 이미징을 조정하기

1. Mech-Eye Viewer 소프트웨어를 열고 카메라 파라미터를 조정합니다.
2. 2D 파라미터를 조정함으로써 2D 이미지의 캘리브레이션 보드가 뚜렷하고 노출 과다나 노출 부족 등 현상이 없도록 합니다.
3. 3D 파라미터를 조정함으로써 캘리브레이션 보드의 원의 포인트 클라우드를 완전하게 하고 **포인트 클라우드 후처리에서 포인트 클라우드 평활화 및 노이즈 제거를 일반적으로 변경하는 것이 좋으며** 포인트 클라우드 변동 범위를 줄입니다.



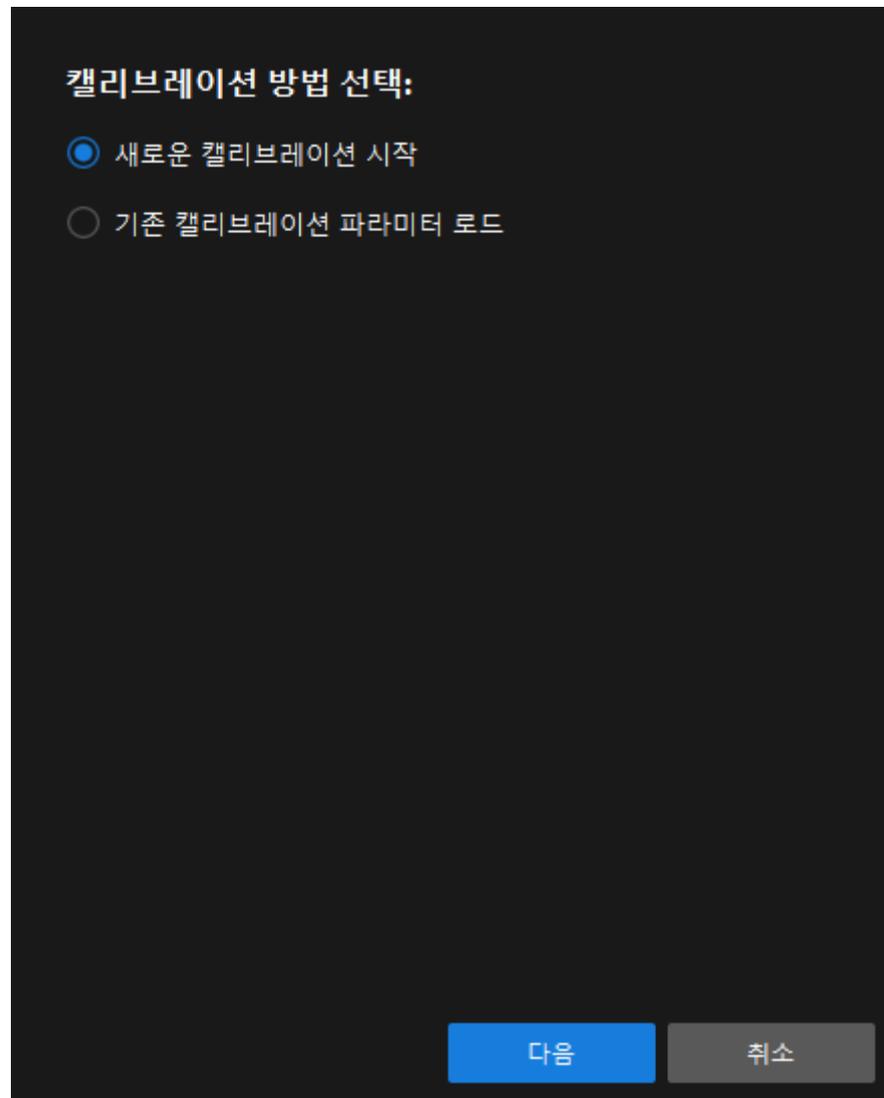
현장에 환경광이 복잡한 경우, 환경광이 2D 맵과 포인트 클라우드에 주는 영향을 줄이기 위해 광원을 차단하거나 보충하기 바랍니다.

4. 위의 단계를 완료하여 캘리브레이션 보드의 포인트 클라우드 이미징 품질이 요구 사항을 충족하는지 확인하십시오.

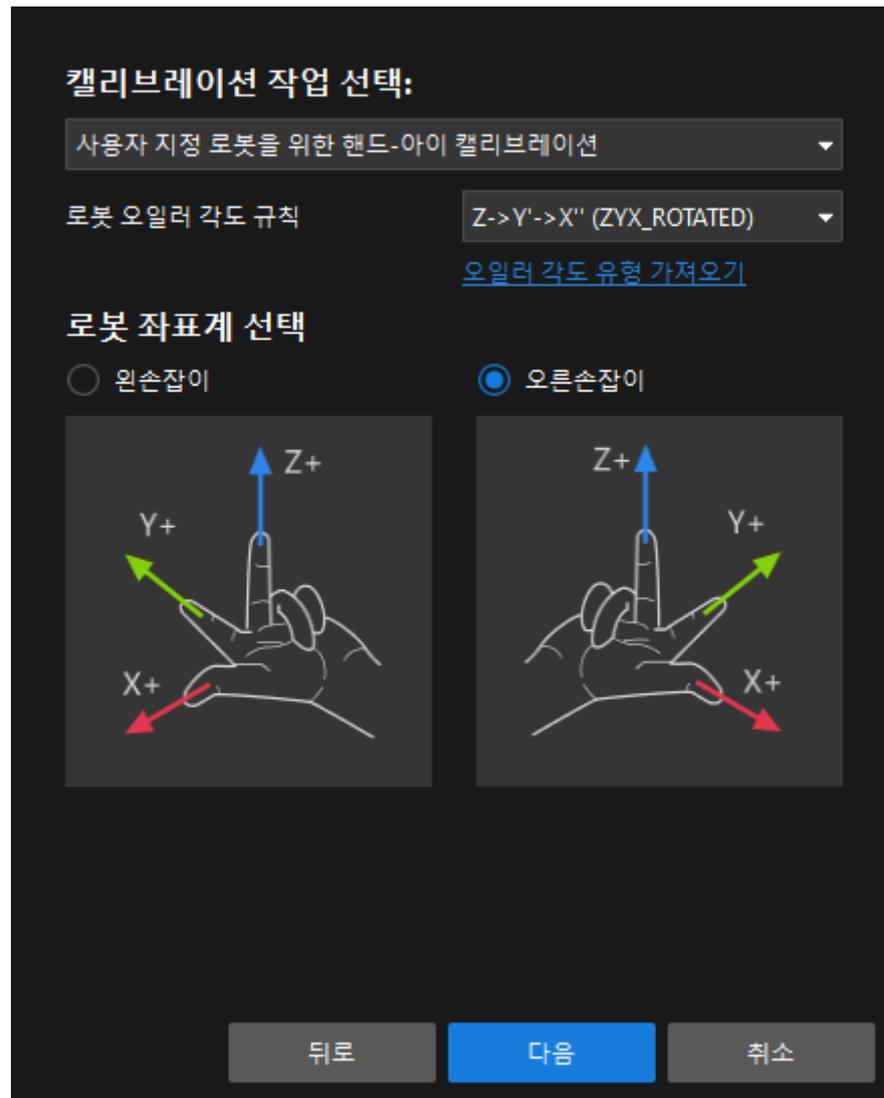
	정상	노출 과다	노출 부족
2D 맵			
포인트 클라우드			

### 캘리브레이션 사전 구성

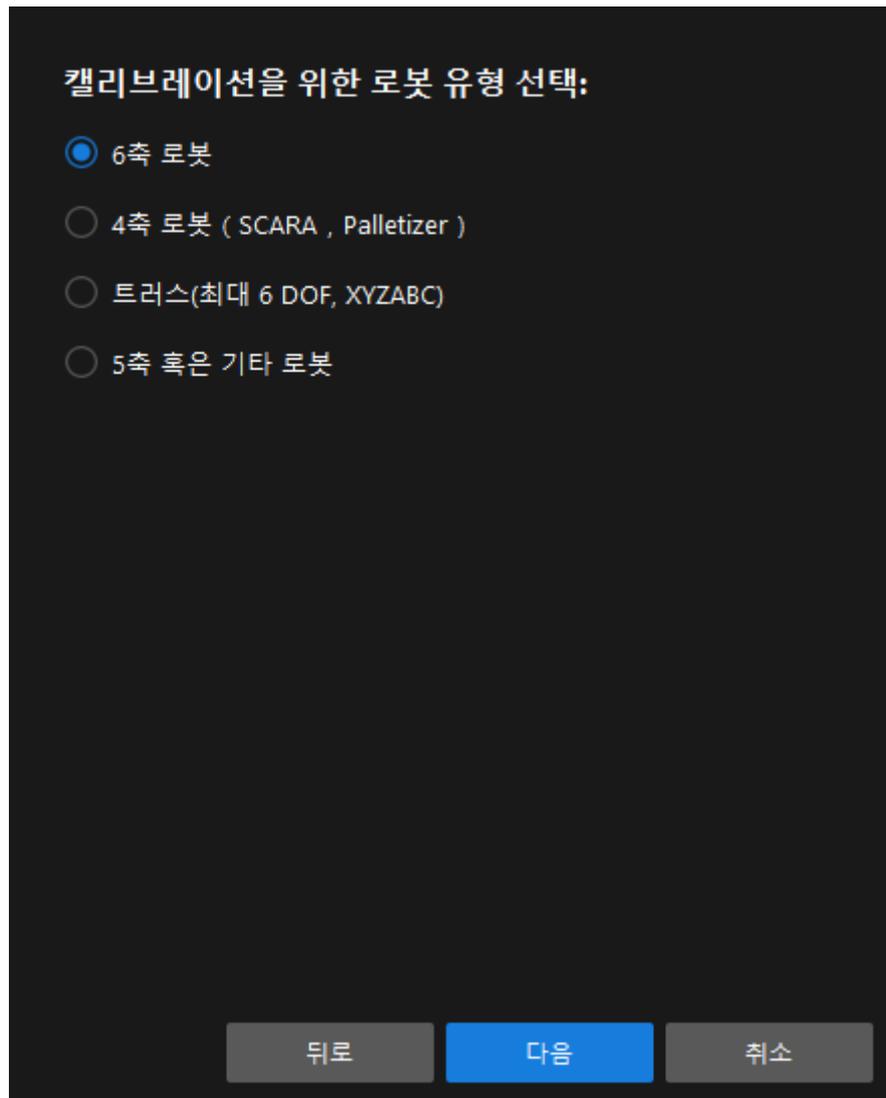
1. Mech-Vision 소프트웨어에서 톨 바의 [ 카메라 캘리브레이션(표준) ] 버튼을 클릭하면 **캘리브레이션 사전 구성** 창이 팝업합니다.
2. **캘리브레이션 방법 선택** 창에서 **새로운 캘리브레이션 시작** 라디오 버튼을 선택한 후 [ 다음 ] 버튼을 클릭합니다.



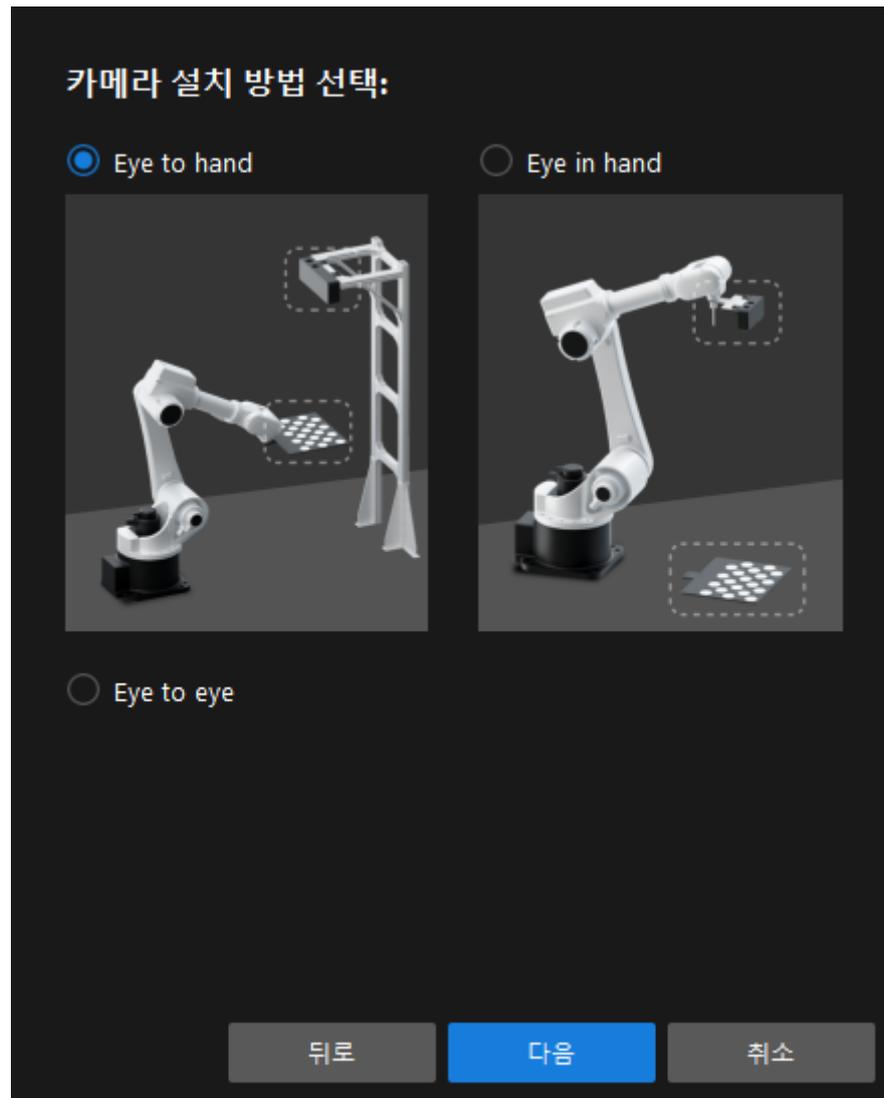
3. **캘리브레이션 작업 선택** 창의 드롭다운 목록 상자에서 **사용자 지정 로봇을 위한 핸드-아이 캘리브레이션**을 선택하고 **로봇 오일러 각도 유형** 파라미터를 지정하여 로봇 좌표계 유형을 선택한 후 [ 다음 ] 버튼을 클릭합니다.



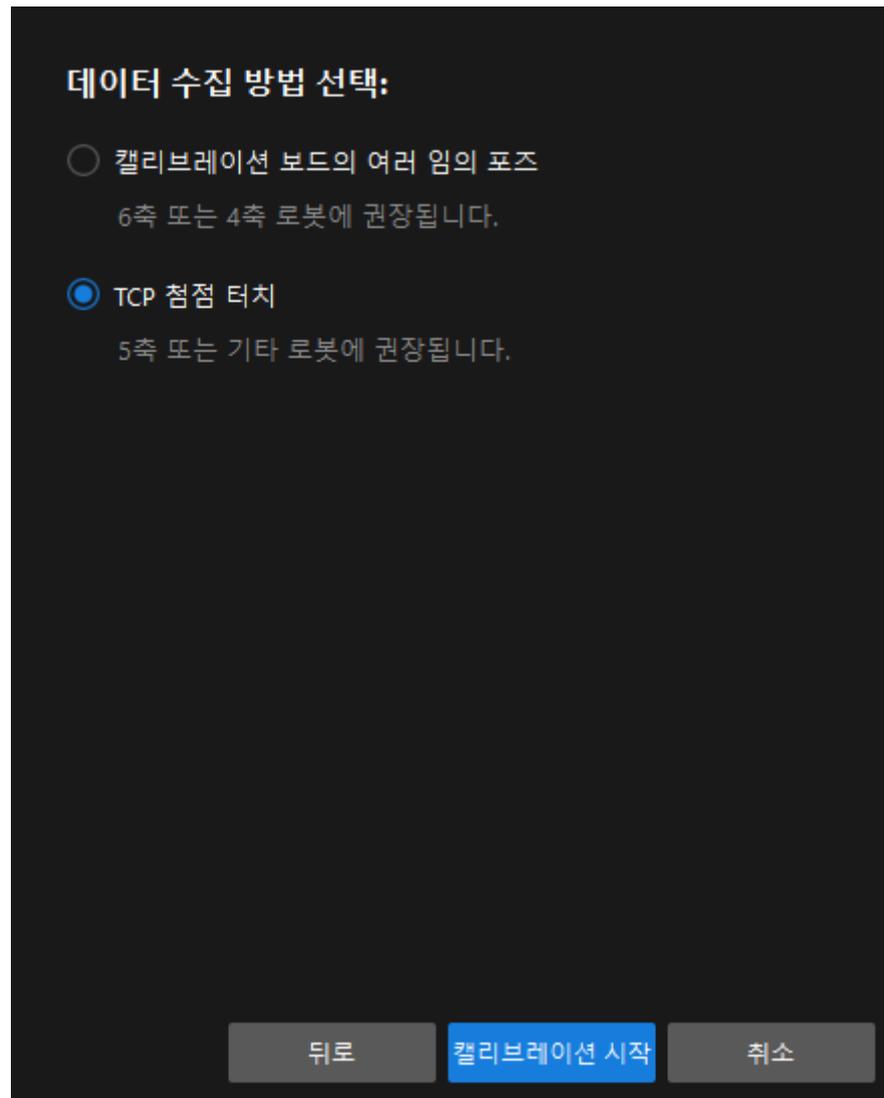
4. 캘리브레이션을 위한 로봇 유형 선택 창에서 로봇 유형에 따라 6축 로봇, 4축 로봇 (SCARA, Palletizer) 또는 5축 혹은 기타 로봇 라디오 상자를 선택한 후 [다음] 버튼을 클릭합니다.



5. 카메라 설치 방법 선택 창에서 **Eye to hand** 라디오 버튼을 선택한 후 [ 다음 ] 버튼을 클릭합니다.



6. 데이터 수집 방법 선택 창에서 TCP 끝단점 터치 라디오 버튼을 선택한 다음 [ 캘리브레이션 시작 ] 버튼을 클릭합니다. 캘리브레이션(Eye to Hand) 창이 나타납니다.

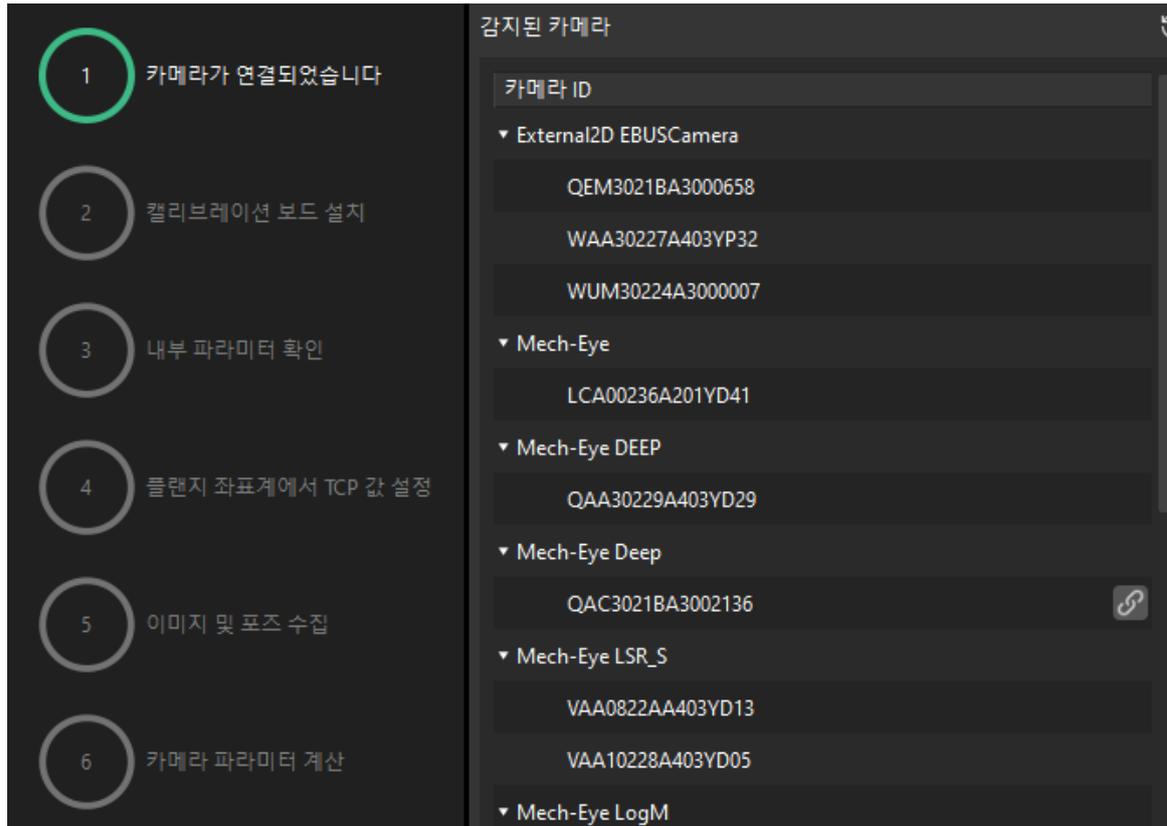


여기까지 캘리브레이션 사전 구성이 완료되었으며 정식 캘리브레이션 프로세스로 들어갑니다.

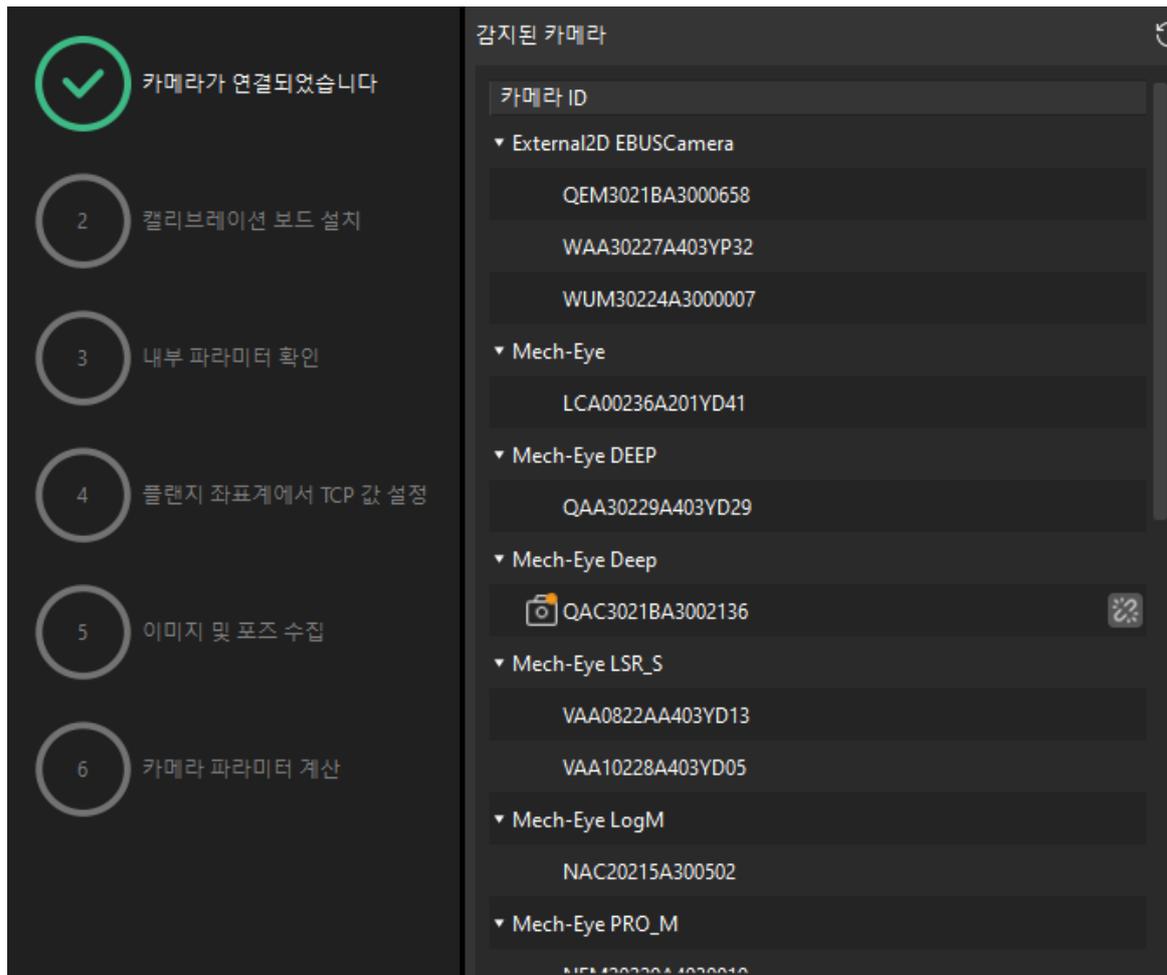
## 캘리브레이션 프로세스

### 카메라 연결

1. **카메라 연결** 단계에서 **감지된 카메라** 리스트에서 연결할 카메라를 선택한 다음  을 클릭하면 카메라가 연결되며 또는 직접 연결할 카메라를 더블클릭합니다.



2. 카메라를 연결한 후 [연속 캡처] 또는 [한번 캡처] 버튼을 선택할 수 있습니다.



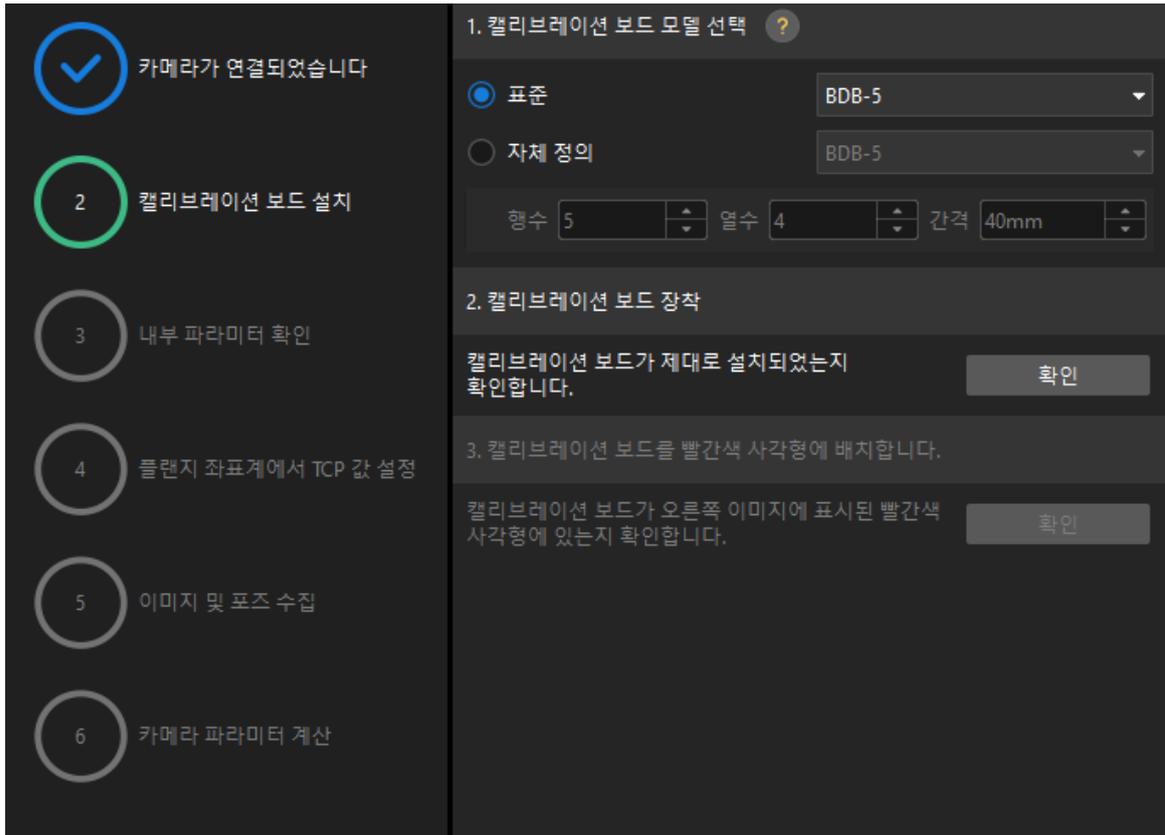
3. 이미지 뷰어에서 카메라가 캡처한 2D 맵과 덤스 맵이 캘리브레이션 요구 사항을 충족하는지 확인한 다음 하단 표시줄에서 [ 다음 ] 버튼을 클릭합니다.



캡처한 이미지가 캘리브레이션 요구 사항을 충족하지 않으면 Mech-Eye Viewer 소프트웨어를 열어 **카메라의 2D 및 3D 노출 파라미터를 조정**하고 다시 캡처해야 합니다.

#### 캘리브레이션 보드 설치

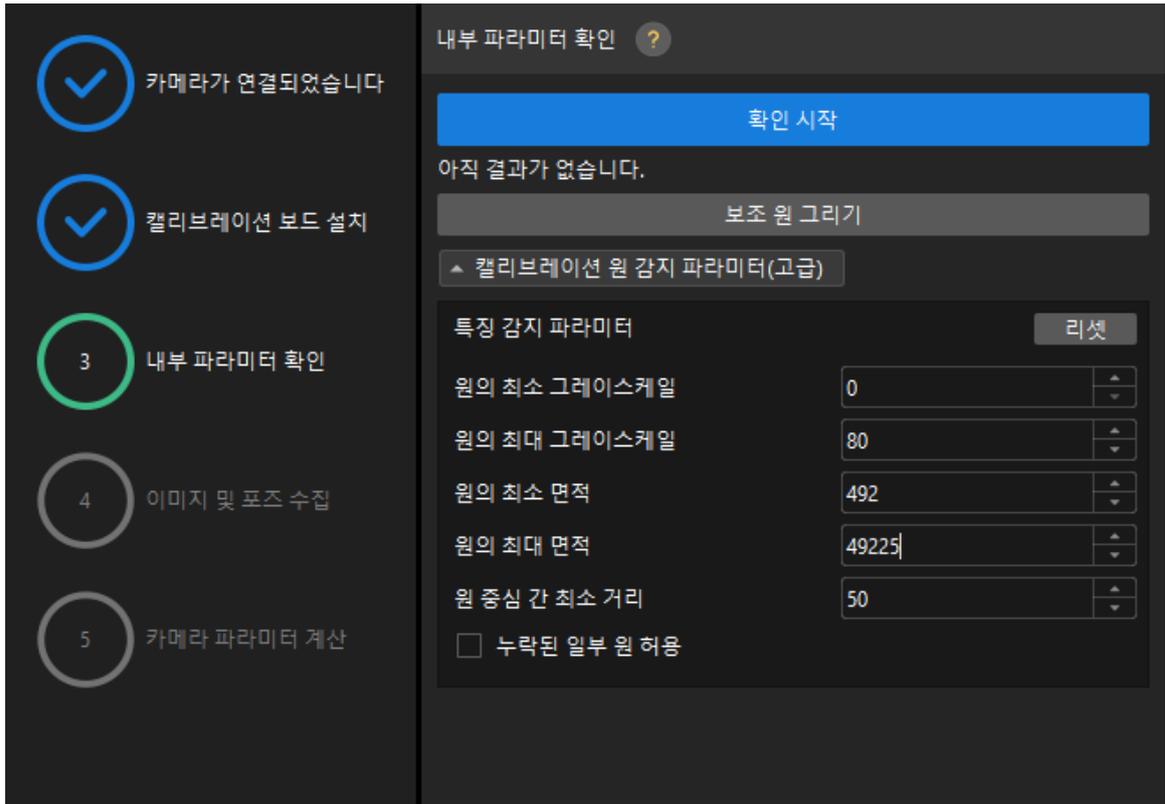
1. 캘리브레이션 보드 설치 단계의 **1. 캘리브레이션 보드 유형 선택** 영역에서 **표준** 라디오 버튼을 선택한 후 캘리브레이션 보드 모델 레이블에 따라 해당 캘리브레이션 유형을 선택합니다.



2. 캘리브레이션 보드가 작업 평면 중앙에 놓여 있는지 확인한 후 **2. 캘리브레이션 보드 설치** 영역에서 [ **확인** ] 버튼을 클릭합니다.
3. 캘리브레이션 보드가 카메라 시야의 중앙(빨간색 사각형 내)에 있는지 확인한 다음 **3.캘리브레이션 보드를 빨간색 사각형에 배치** 영역에서 [ **확인** ] 버튼을 클릭합니다.
4. 캘리브레이션 보드 관련 작업을 모두 완료한 후 하단 표시줄에서 [ **다음** ] 버튼을 클릭합니다.

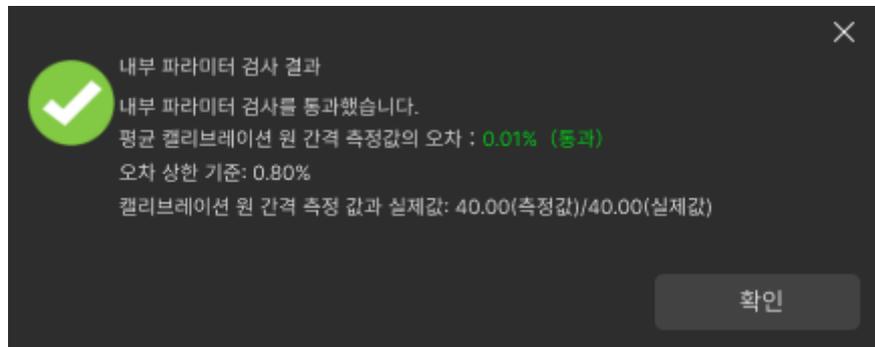
#### 카메라 내부 파라미터를 검사하기

1. 내부 파라미터 검사 단계에서 [ **검사 시작** ] 버튼을 클릭합니다.



## 2. 카메라의 내부 파라미터 검사 결과를 확인합니다.

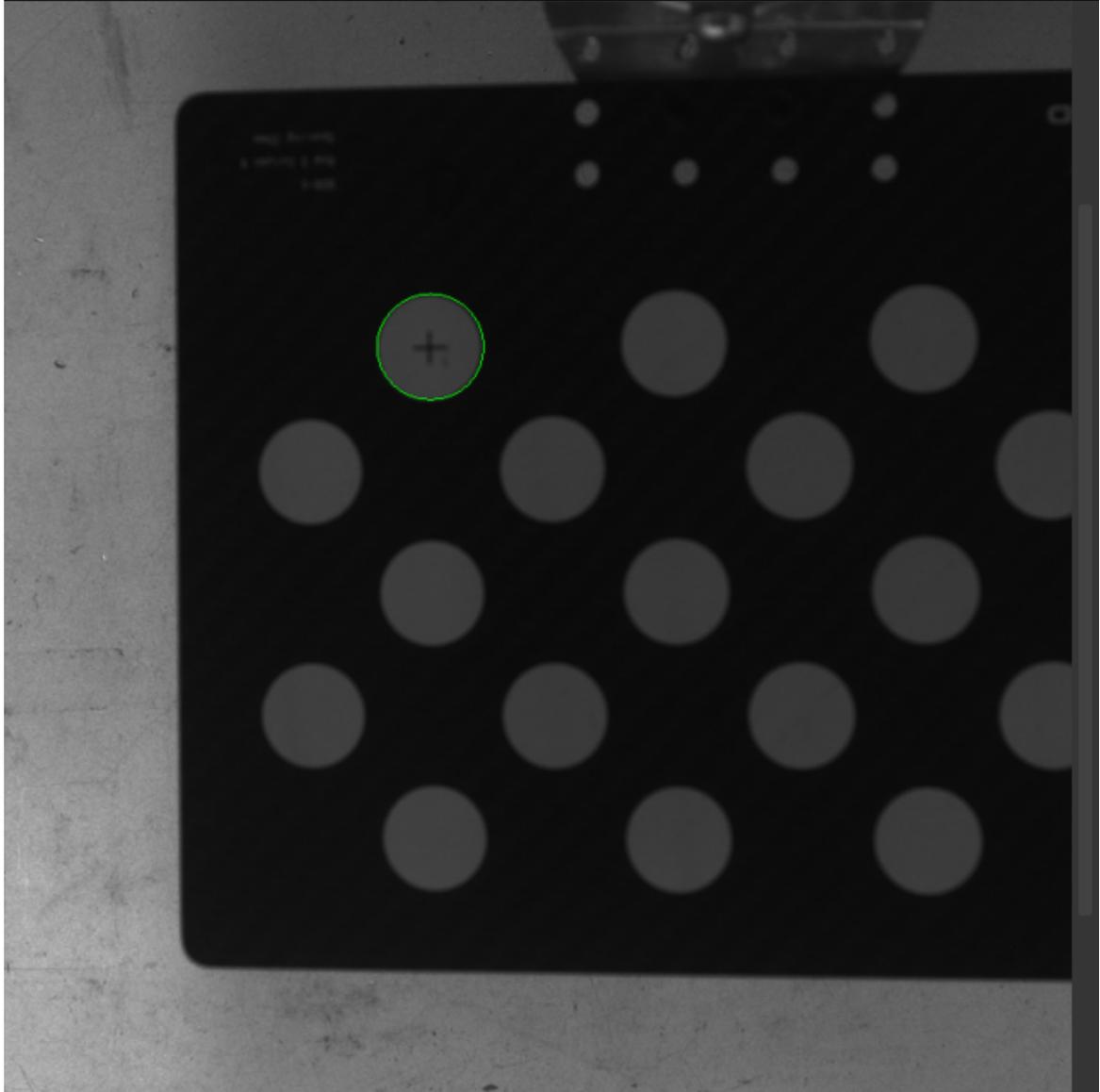
- 카메라의 내부 파라미터 검사에 성공하면 [확인]을 클릭한 후 [다음] 버튼을 클릭하십시오.



- 카메라의 내부 파라미터 검사에 실패하면 보조원을 그리거나 캘리브레이션 원을 수동으로 편집하여 파라미터를 감지 한 다음 [다시 검사] 버튼을 클릭하십시오.

### 보조원을 그리기

1. 보조원을 그리기로 선택한 경우 [보조원 그리기] 버튼을 클릭하십시오.
2. 오른쪽 이미지 뷰어 패널에서 마우스 오른쪽으로 캘리브레이션 보드 이미지를 클릭하고 해당 창 확인란을 언체크 한 다음 **Ctrl** 키를 누른 상태에서 롤러를 드래그하여 이미지를 적절한 크기로 조정하십시오.
3. 마우스 포인터를 캘리브레이션 원의 중간 줄 지점으로 이동하고 왼쪽 마우스 버튼을 누른 다음 보조원에 캘리브레이션 원을 완전히 포함시킨 다음 해제하십시오.



4. [ 다시 검사 ] 버튼을 클릭하고 카메라 내부 파라미터 검사가 통과되었는지 확인하십시오.

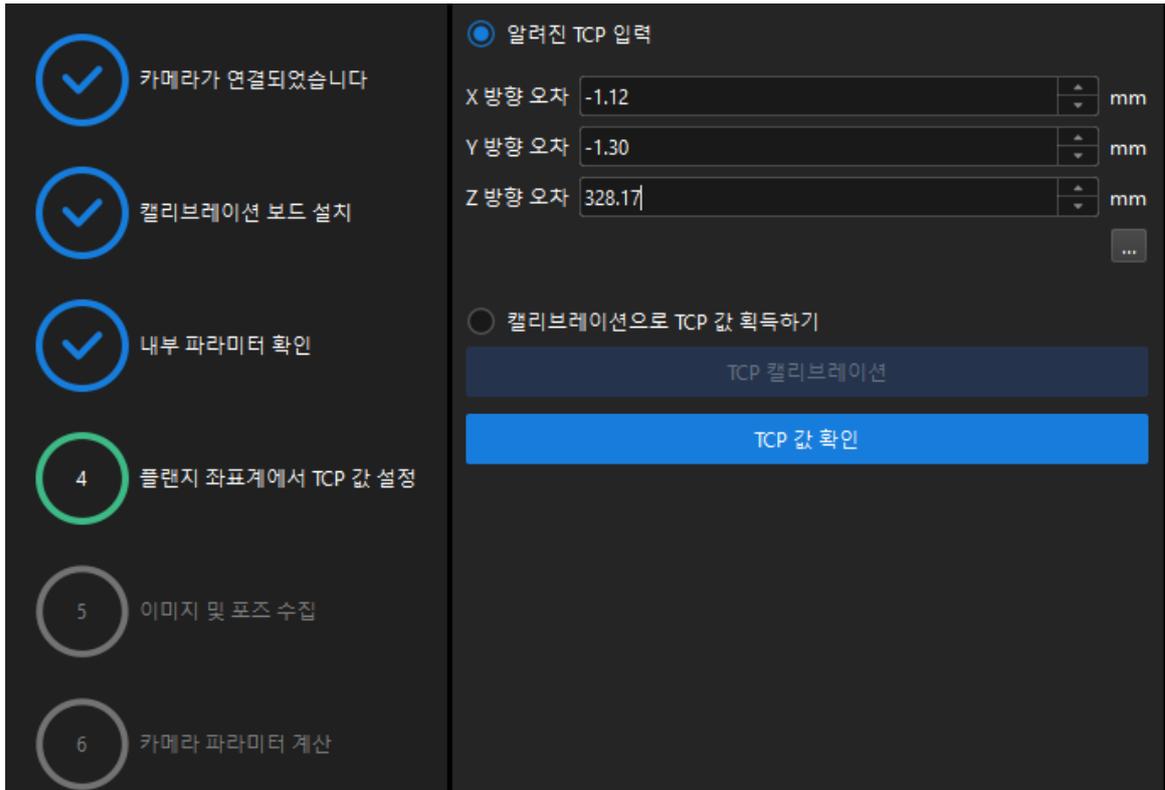
#### 감지 파라미터 수동으로 조정

감지 파라미터를 수동으로 조정할 경우 [ 캘리브레이션 원 감지 파라미터(고급) ]을 클릭하여 감지 파라미터의 값을 변경하면 됩니다.

캘리브레이션 원이 여전히 감지되지 않은 경우 실제 작업 조건에 따라 카메라 관련 파라미터를 조정해야 하며 카메라 파라미터는 [파라미터 조정](#) 내용을 참조하십시오.

#### 플랜지 좌표계에서 TCP 값 설정

1. 플랜지 좌표계에서 TCP 값 설정 단계에서 알려진 TCP 입력 라디오 상자를 선택합니다.



2. [ **TCP 값 확인** ]을 클릭합니다.
3. 하단에서 [ **다음** ] 버튼을 클릭하십시오.



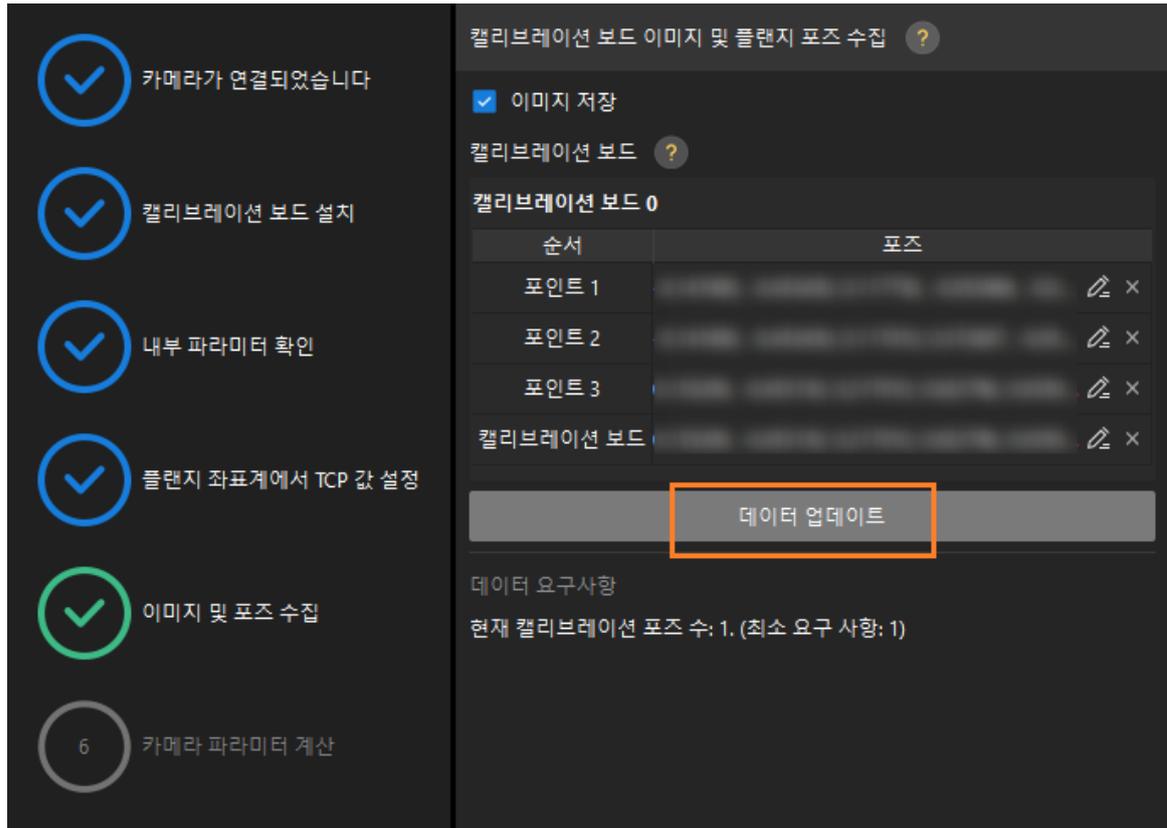
TCP 값을 알 수 없는 경우 **캘리브레이션으로 TCP 값 획득하기** 라디오 상자를 선택하고 [ **TCP 캘리브레이션** ] 버튼을 클릭합니다. TCP 캘리브레이션 도구를 사용하여 TCP 값을 계산할 수 있습니다.

#### 이미지 및 포즈 가져오기

1. **이미지 및 포즈 가져오기** 단계에서 **이미지를 저장하기** 체크박스를 선택합니다.
2. 끝점이 캘리브레이션 보드의 포인트 1의 교차 중심점에 닿도록 로봇의 움직임을 제어하고 티치 펜던트에서 로봇 플랜지의 포즈를 기록합니다.
3. 포인트 1의  버튼을 클릭하고 팝업된 **로봇 플랜지 포즈를 입력하기** 다이얼로그 박스에서 로봇의 플랜지 포즈를 입력한 후 [ **확인** ] 버튼을 클릭합니다.



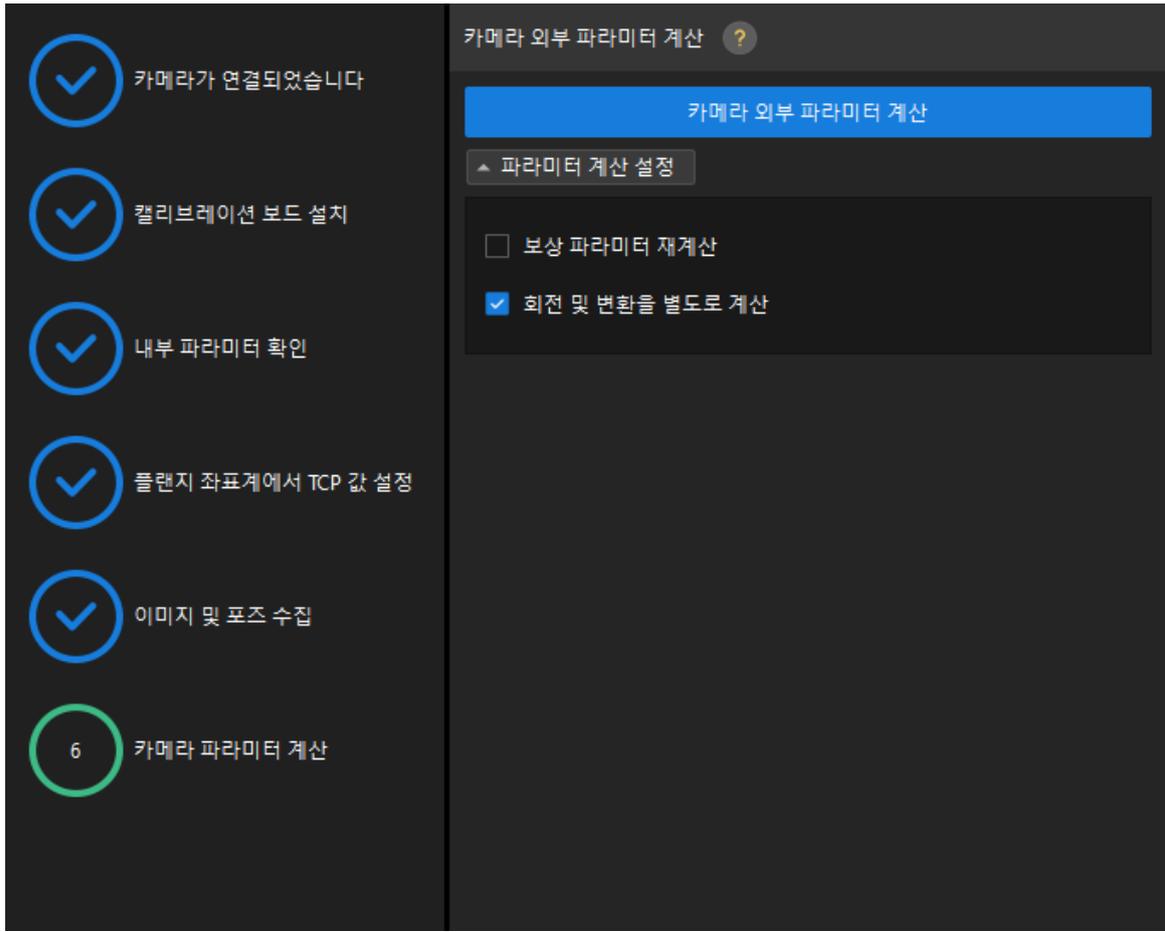
4. 위의 단계를 반복하여 TCP 끝단점이 포인트 2와 포인트 3의 교차 중심점에 닿도록 하고 로봇 플랜지의 포즈를 입력합니다.
5. 현재 캘리브레이션 보드의  버튼을 클릭하여 캘리브레이션 보드의 이미지를 캡처합니다.
6. [ 데이터 업데이트 ] 버튼을 클릭한 후 하단 표시줄에서 [ 다음 ] 버튼을 클릭합니다.



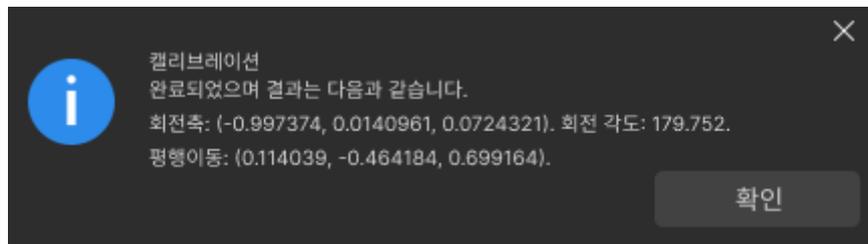
캘리브레이션 절차에서는 외부 파라미터를 계산하기 위해 최소 3개의 비동일선 지점을 터치해야 합니다.

### 카메라 파라미터를 계산하기

1. 카메라 파라미터를 계산하기 단계에서 **회전 및 변환을 별도로 계산** 파라미터를 선택한 다음 [ **카메라 외부 파라미터 계산** ] 버튼을 클릭하십시오.



2. 팝업한 캘리브레이션 성공 다이얼로그 박스에서 [ 확인 ] 버튼을 클릭합니다.



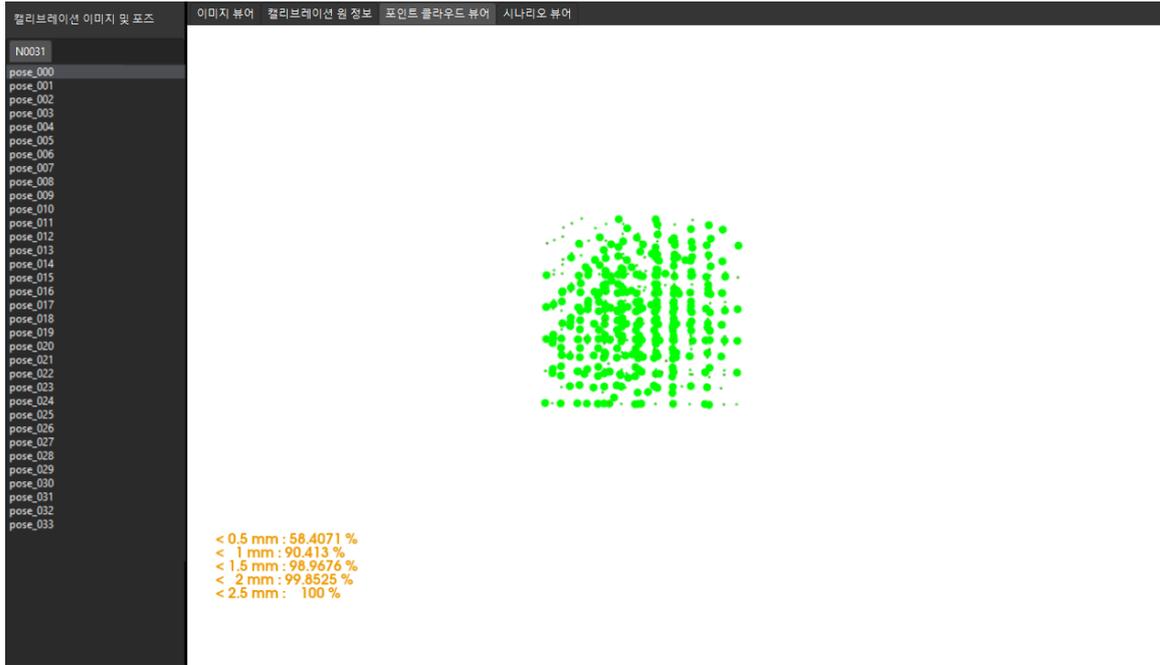
3. 오른쪽 **포인트 클라우드 뷰어** 패널에서 캘리브레이션 오차 포인트 클라우드를 확인합니다.



오차 포인트 클라우드는 각 캘리브레이션 포즈의 캘리브레이션 보드 원의 진릿값과 계산된 값 사이의 편차를 표시하는 데 사용됩니다.

4. 캘리브레이션 정확도가 예상 요구 사항을 충족하는지 확인합니다.

캘리브레이션의 정확도를 결정하기 위해 100%를 차지하는 오차 값을 찾으십시오. 예를 들어, 아래 그림의 정확도는 2.5mm 미만입니다.



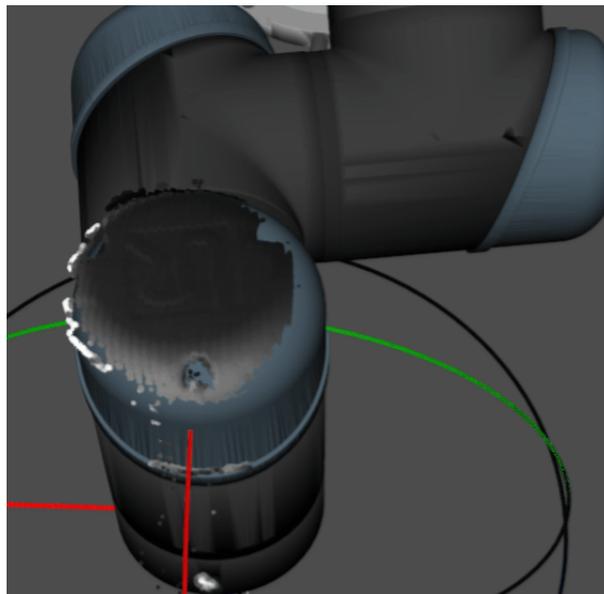
캘리브레이션 정확도를 개선해야 하는 경우 [캘리브레이션 결과 분석](#) 내용을 참조하십시오.

### 캘리브레이션 결과 검증 및 저장

카메라 외부 파라미터를 계산한 후 시나리오 뷰어에서 로봇 포인트 클라우드와 로봇 모델의 매칭 정도를 보고 캘리브레이션 결과를 대략적으로 판단할 수 있습니다. 구체적인 작업은 다음과 같습니다.

1. 로봇 팔을 카메라의 시야 내로 이동합니다.
2. 카메라 파라미터 계산 단계에서 [ 카메라 외부 파라미터 계산 ] 버튼을 클릭합니다. 이 작업은 카메라가 사진을 캡처하도록 트리거합니다.
3. 시나리오 뷰어를 클릭하여 로봇 포인트 클라우드와 로봇 모델 간의 매칭 정도를 확인합니다.

로봇의 포인트 클라우드가 로봇 모델과 대략적으로 매칭하면 캘리브레이션 결과를 사용할 수 있음을 의미합니다.





1. 로봇 모델은 실제 로봇의 모양과 완전히 매칭하지 않을 수 있으며 외부 파라미터를 미세 조정하기 위한 기준으로 사용할 수 없습니다.
2. 시나리오 뷰어 외에도 Mech-Viz 소프트웨어에서 로봇의 포인트 클라우드와 로봇 모델 간의 매칭 정도를 확인할 수 있습니다.

4. 하단 표시줄에서 [ 저장 ] 버튼을 클릭하고 팝업한 **캘리브레이션 파일 저장** 다이얼로그 박스에서 [ 확인 ] 버튼을 클릭합니다. 카메라 캘리브레이션 결과는 프로젝트의 "calibration" 디렉터리에 자동으로 저장됩니다.

이로써 캘리브레이션 프로세스가 완료됩니다.

#### 4.4.7. Eye in Hand 시나리오에서의 수동 캘리브레이션(TCP 끝단점 터치)

이 부분에서는 EIH(Eye in Hand) 시나리오에서 수동 캘리브레이션(TCP 끝단점 터치)을 완료하는 방법에 대해 설명합니다.

##### 캘리브레이션 사전 준비

핸드-아이 캘리브레이션을 수행하기 전에 다음 준비를 완료해야 합니다.

- Mech-Mind 비전 시스템 구축을 완료합니다.
- 캘리브레이션에 필요한 재료를 준비합니다.
- 캘리브레이션 보드 포인트 클라우드 이미징을 조정합니다.
- 로봇 및 인터페이스 구성을 완료합니다.

##### 비전 시스템 전체 구축 완료

Mech-Mind 비전 시스템 구축을 완료하려면 비전 시스템 구축 부분 내용을 참조하십시오.

핸드-아이 캘리브레이션에는 Mech-Eye Viewer, Mech-Center, Mech-Vision 및 Mech-Viz 소프트웨어가 필요합니다. 위의 소프트웨어가 설치되고 최신 버전으로 업데이트되었는지 확인하십시오.

##### 캘리브레이션에 필요한 재료를 준비

EIH 시나리오에서 수동 캘리브레이션을 하려면 캘리브레이션 보드와 TCP 끝단점을 사용해야 합니다.

다음 요구 사항에 따라 캘리브레이션 보드를 준비하십시오.

- 캘리브레이션 보드에서 도트가 잘 보이고 눈에 띄는 굵은 흔적이 없으며 캘리브레이션 보드가 뒤틀리지 않아야 합니다.
- EIH 시나리오에서 캘리브레이션 보드를 작업대의 중앙에 평평하게 놓으십시오.

TCP 끝단점은 로봇 플랜지 중앙에 설치해야 합니다. 로봇 플랜지에 분리 불가능한 그리퍼를 장착한 경우에 TCP 끝단점을 그리퍼에 직접 고정하면 됩니다.

또한 캘리브레이션 시작 전 로봇을 캘리브레이션 시작 포즈로 이동합니다.

##### 캘리브레이션 보드 포인트 클라우드 이미징을 조정하기

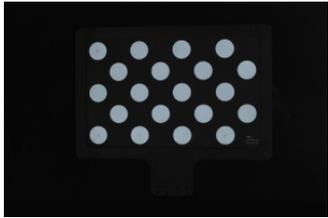
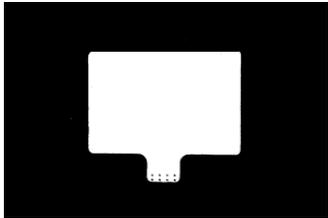
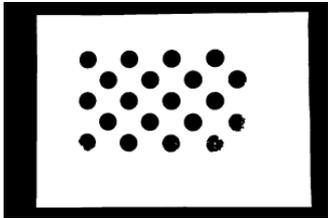
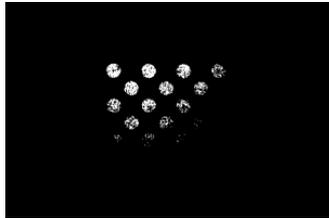
1. Mech-Eye Viewer 소프트웨어를 열고 카메라 파라미터를 조정합니다.
2. 2D 파라미터를 조정함으로써 2D 이미지의 캘리브레이션 보드가 뚜렷하고 노출 과다나 노출 부족 등 현상이 없도록 합니다.
3. 3D 파라미터를 조정함으로써 캘리브레이션 보드의 원의 포인트 클라우드를 완전하게 하고 **포인트 클라우드 후처리**에서 **포인트 클라우드 평활화** 및 **노이즈 제거**를 **일반**으로 변경하는 것이 좋으며

포인트 클라우드 변동 범위를 줄입니다.



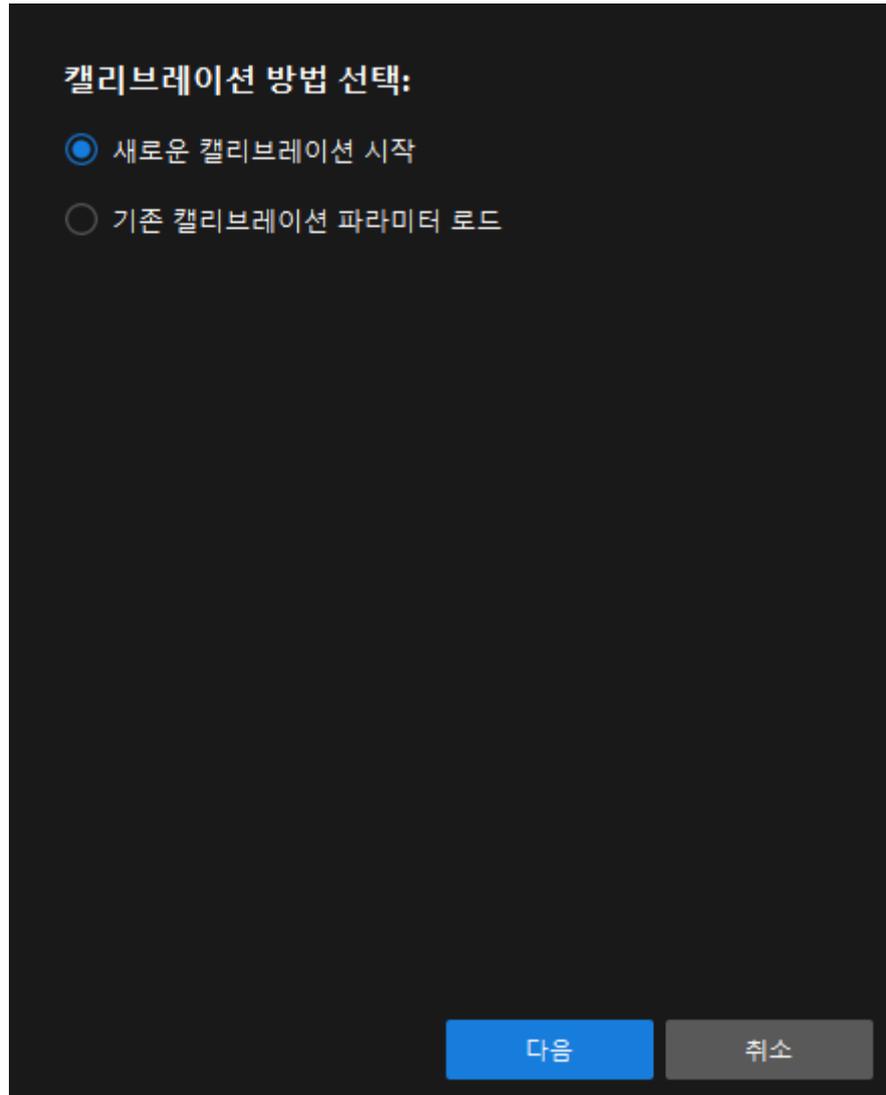
현장에 환경광이 복잡한 경우, 환경광이 2D 맵과 포인트 클라우드에 주는 영향을 줄이기 위해 광원을 차단하거나 보충하기 바랍니다.

- 위의 단계를 완료하여 캘리브레이션 보드의 포인트 클라우드 이미징 품질이 요구 사항을 충족하는지 확인하십시오.

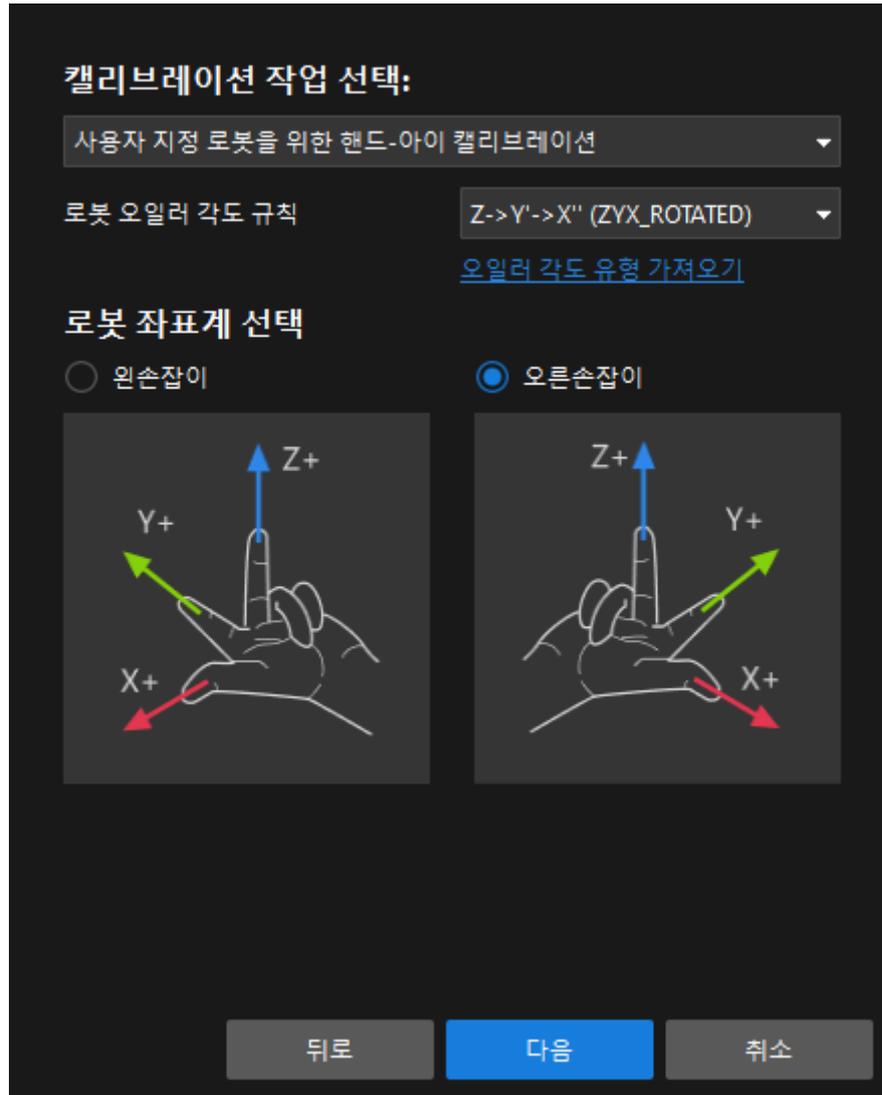
	정상	노출 과다	노출 부족
2D 맵			
포인트 클라우드			

#### 캘리브레이션 사전 구성

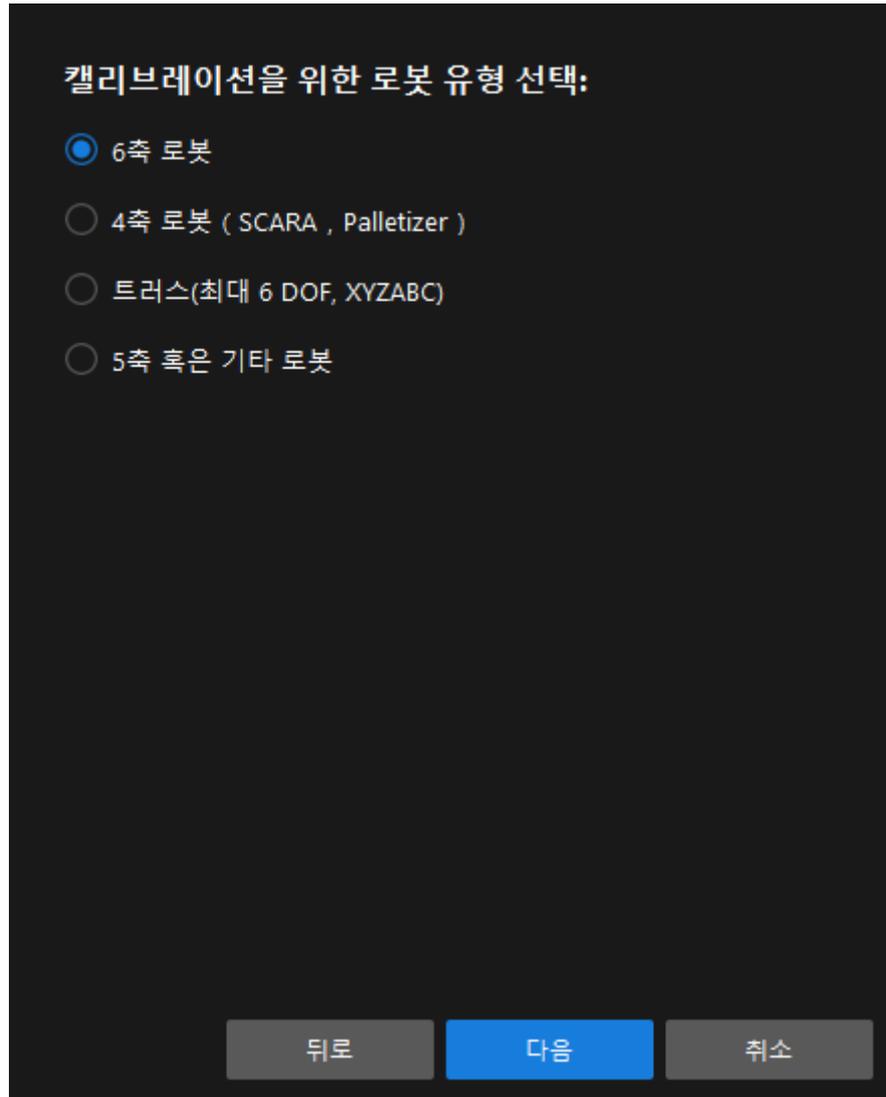
- Mech-Vision 소프트웨어에서 톨 바의 [ 카메라 캘리브레이션(표준) ] 버튼을 클릭합니다. 캘리브레이션 사전 구성 창이 팝업됩니다.
- 캘리브레이션 방법 선택 창에서 새로운 캘리브레이션 시작 라디오 버튼을 선택한 후 [ 다음 ] 버튼을 클릭합니다.



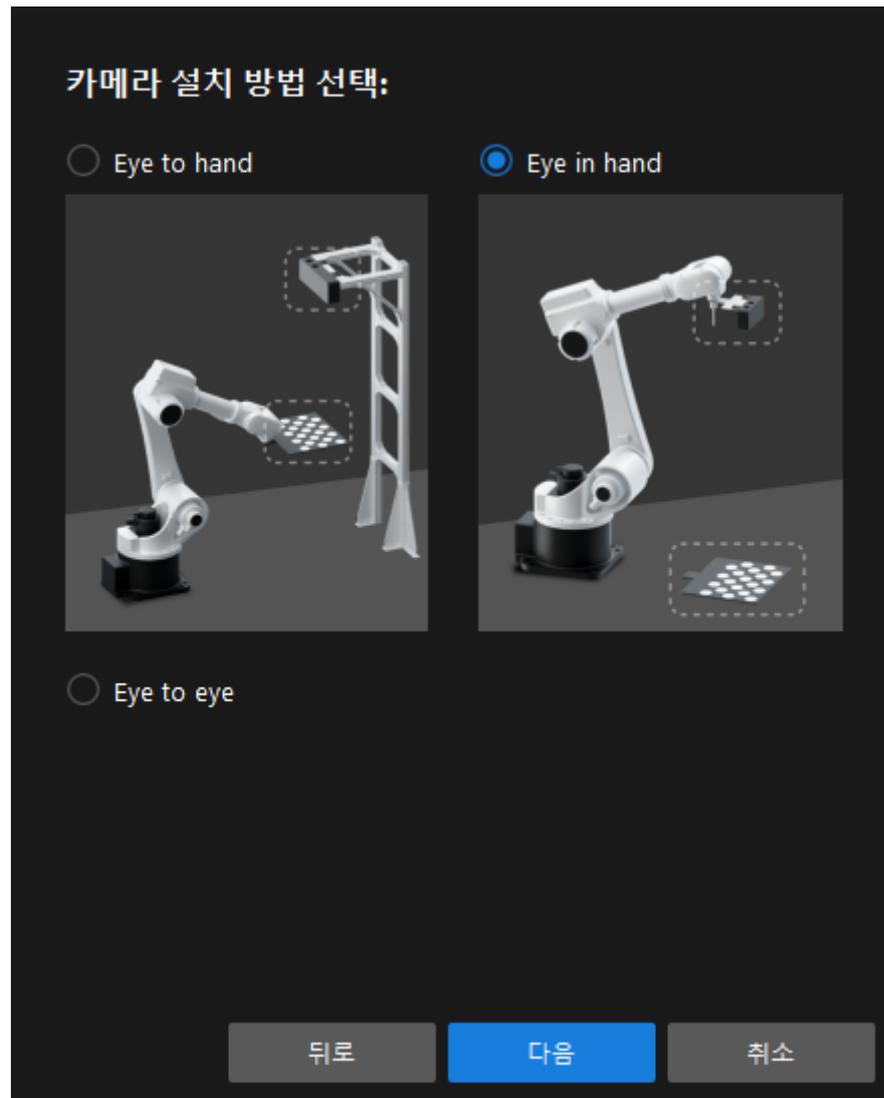
3. 캘리브레이션 작업 선택 창의 드롭다운 목록 상자에서 사용자 지정 로봇을 위한 핸드-아이 캘리브레이션을 선택하고 로봇 오일러 각도 유형 파라미터를 지정하여 로봇 좌표계 유형을 선택한 후 [ 다음 ] 버튼을 클릭합니다.



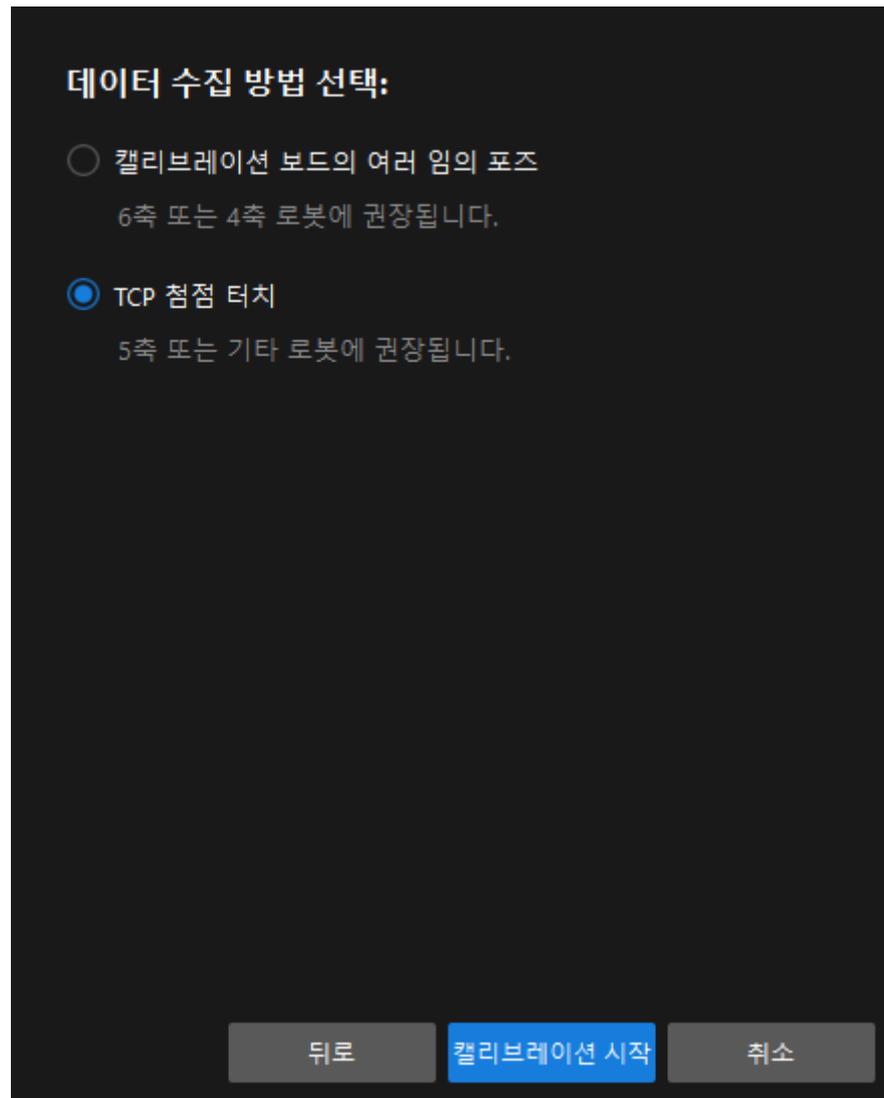
4. 캘리브레이션을 위한 로봇 유형 선택 창에서 로봇 유형에 따라 6축 로봇, 4축 로봇 (SCARA, Palletizer) 또는 5축 혹은 기타 로봇 라디오 상자를 선택한 후 [다음] 버튼을 클릭합니다.



5. 카메라 설치 방법 선택 창에서 **Eye in hand** 라디오 버튼을 선택한 후 [ 다음 ] 버튼을 클릭합니다.



6. 데이터 수집 방법 선택 창에서 TCP 끝단점 터치 라디오 버튼을 선택한 다음 [ 캘리브레이션 시작 ] 버튼을 클릭합니다. 캘리브레이션(Eye in Hand) 창이 나타납니다.

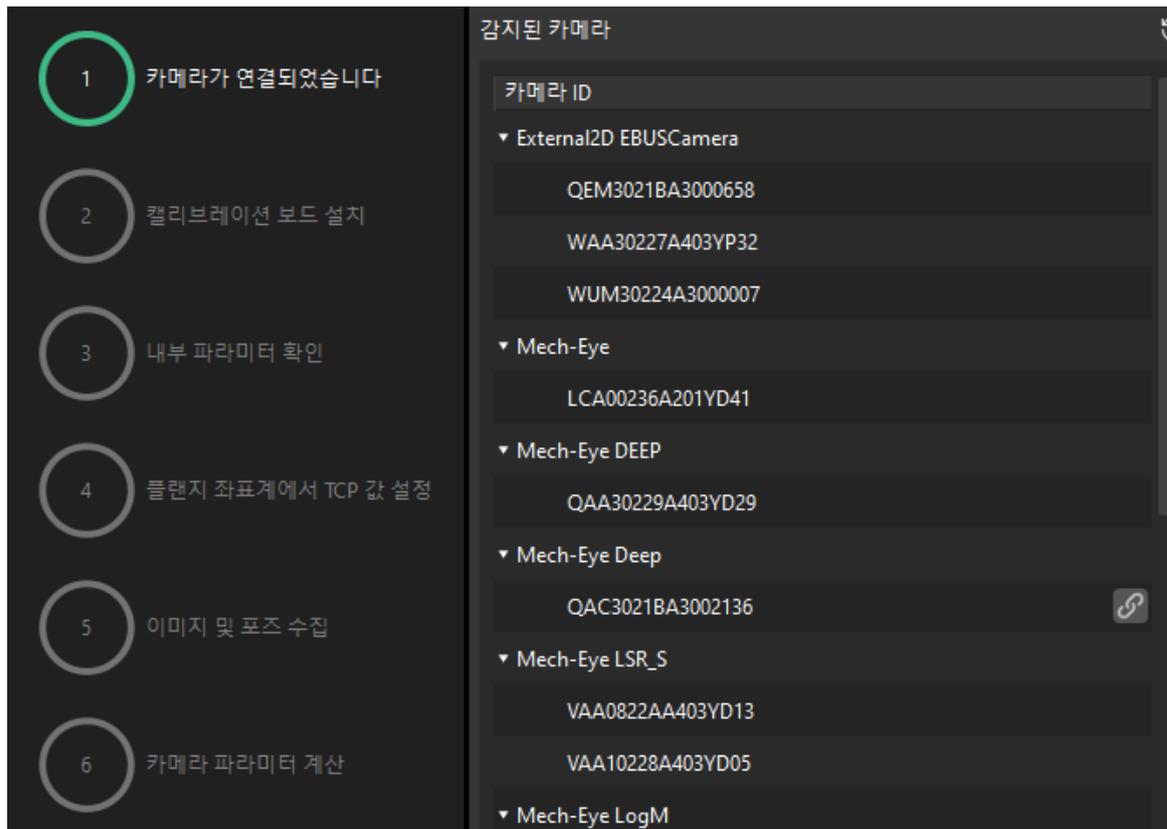


여기까지 캘리브레이션 사전 구성이 완료되었으며 정식 캘리브레이션 프로세스로 들어갑니다.

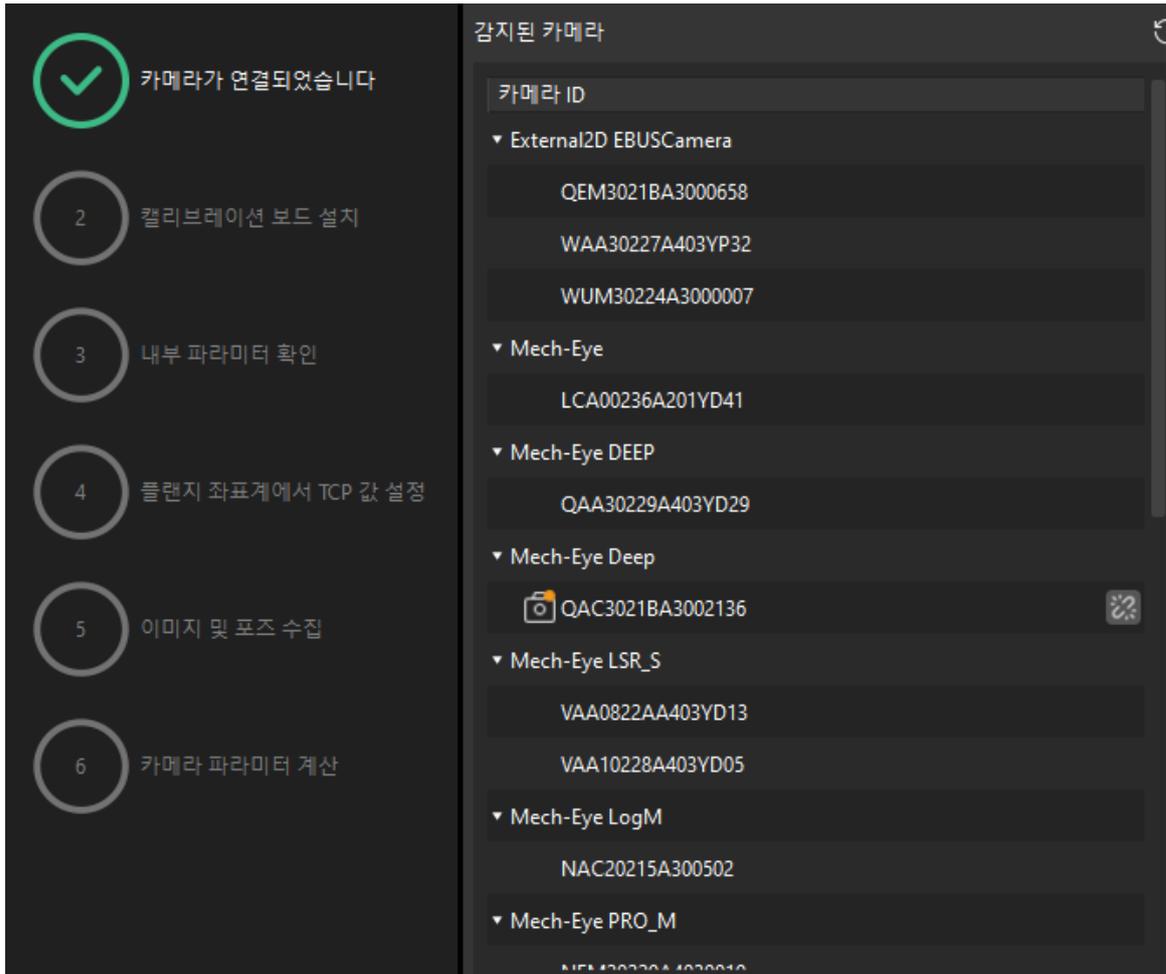
## 캘리브레이션 프로세스

### 카메라 연결

1. **카메라 연결** 단계에서 **감지된 카메라** 리스트에서 연결할 카메라를 선택한 다음  을 클릭하면 카메라가 연결되며 또는 직접 연결할 카메라를 더블클릭합니다.



2. 카메라를 연결한 후 [연속 캡처] 또는 [한번 캡처] 버튼을 선택할 수 있습니다.



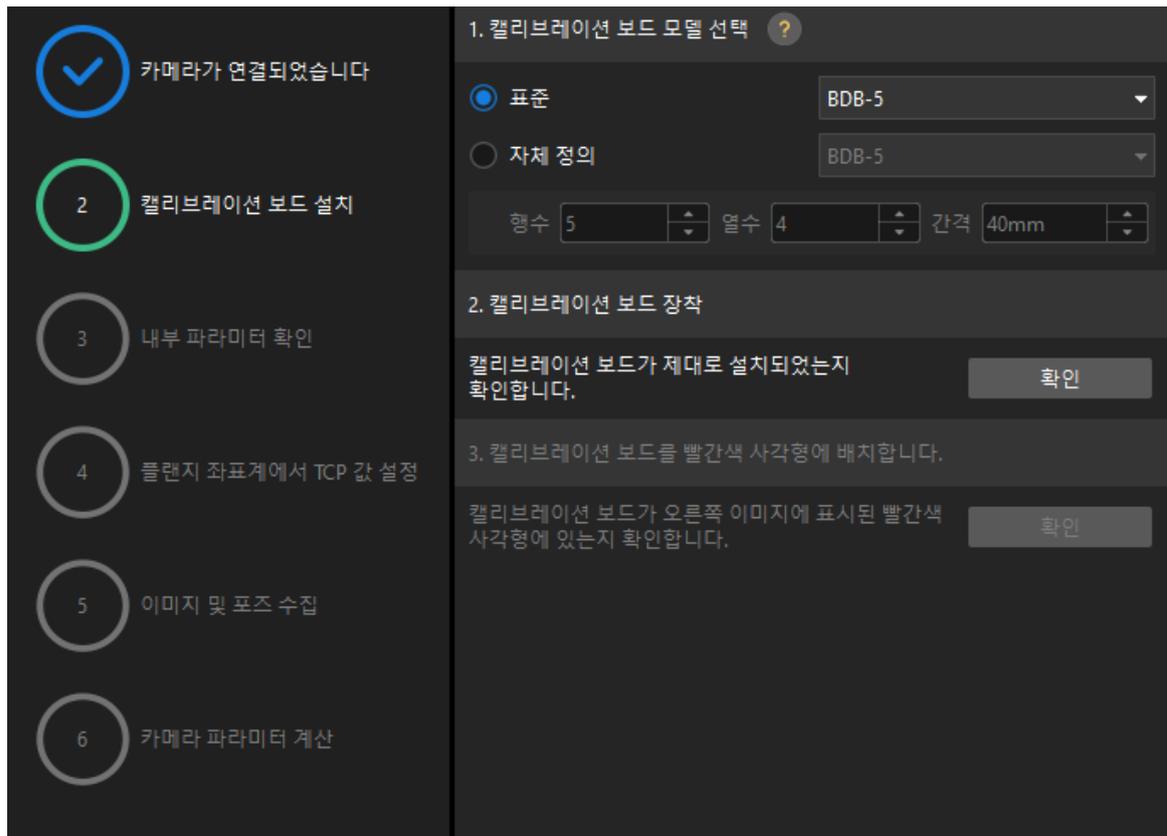
3. 이미지 뷰어에서 카메라가 캡처한 2D 맵과 덤스 맵이 캘리브레이션 요구 사항을 충족하는지 확인한 다음 하단 표시줄에서 [ 다음 ] 버튼을 클릭합니다.



캡처한 이미지가 캘리브레이션 요구 사항을 충족하지 않으면 Mech-Eye Viewer 소프트웨어를 열어 **카메라의 2D 및 3D 노출 파라미터를 조정**하고 다시 캡처해야 합니다.

#### 캘리브레이션 보드 설치

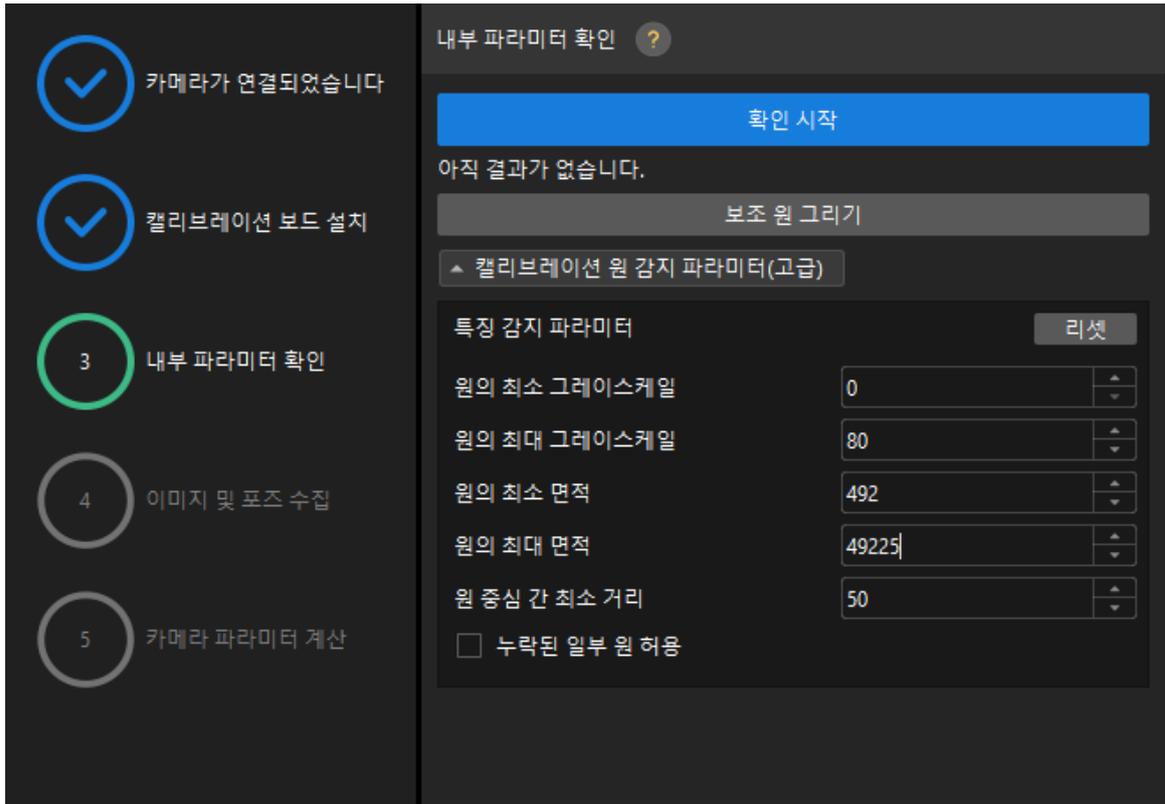
1. 캘리브레이션 보드 설치 단계의 **1. 캘리브레이션 보드 유형 선택** 영역에서 **표준** 라디오 버튼을 선택한 후 캘리브레이션 보드 레이블에 따라 해당 캘리브레이션 유형을 선택합니다.



2. 캘리브레이션 보드가 작업 평면 중앙에 평평하게 놓여 있는지 확인한 후 **2. 캘리브레이션 보드 설치** 영역에서 [ **확인** ] 버튼을 클릭합니다.
3. 캘리브레이션 보드가 카메라 시야의 중앙(빨간색 사각형 내)에 있는지 확인한 다음 **3.캘리브레이션 보드를 빨간색 사각형에 배치** 영역에서 [ **확인** ] 버튼을 클릭합니다.
4. 캘리브레이션 보드 관련 작업을 모두 완료한 후 하단 표시줄에서 [ **다음** ] 버튼을 클릭합니다.

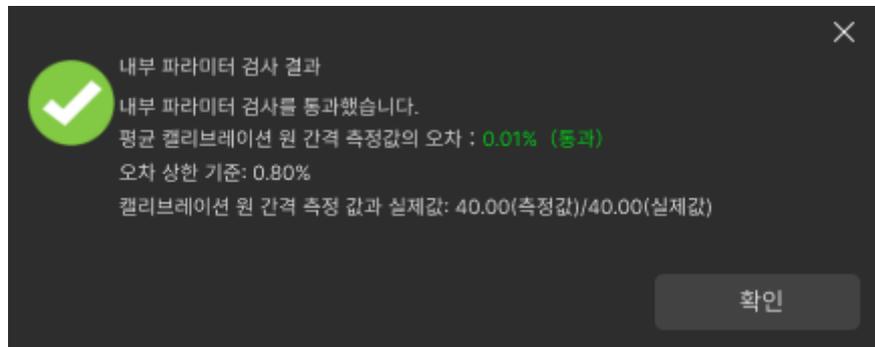
#### 카메라 내부 파라미터를 검사하기

1. 내부 파라미터 검사 단계에서 [ **검사 시작** ] 버튼을 클릭합니다.



## 2. 카메라의 내부 파라미터 검사 결과를 확인합니다.

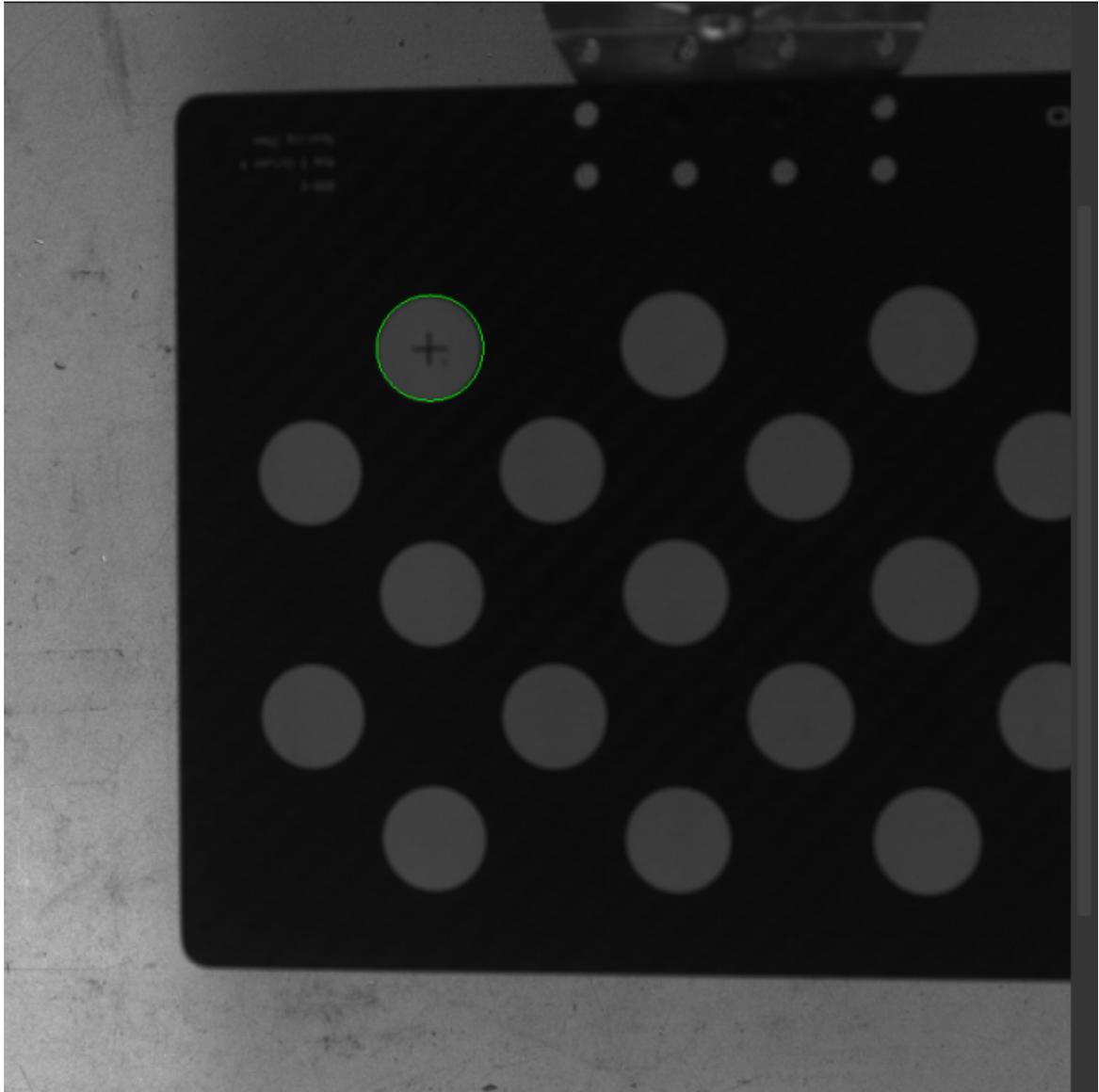
- 카메라의 내부 파라미터 검사에 성공하면 [확인]을 클릭한 후 [다음] 버튼을 클릭하십시오.



- 카메라의 내부 파라미터 검사에 실패하면 보조원을 그리거나 캘리브레이션 원을 수동으로 편집하여 파라미터를 감지 한 다음 [다시 검사] 버튼을 클릭하십시오.

### 보조원을 그리기

1. 보조원을 그리기로 선택한 경우 [보조원 그리기] 버튼을 클릭하십시오.
2. 오른쪽 이미지 뷰어 패널에서 마우스 오른쪽으로 캘리브레이션 보드 이미지를 클릭하고 해당 창 확인란을 언체크 한 다음 **Ctrl** 키를 누른 상태에서 롤러를 드래그하여 이미지를 적절한 크기로 조정하십시오.
3. 마우스 포인터를 캘리브레이션 원의 중간 줄 지점으로 이동하고 왼쪽 마우스 버튼을 누른 다음 보조원에 캘리브레이션 원을 완전히 포함시킨 다음 해제하십시오.



4. [ 다시 검사 ] 버튼을 클릭하고 카메라 내부 파라미터 검사가 통과되었는지 확인하십시오.

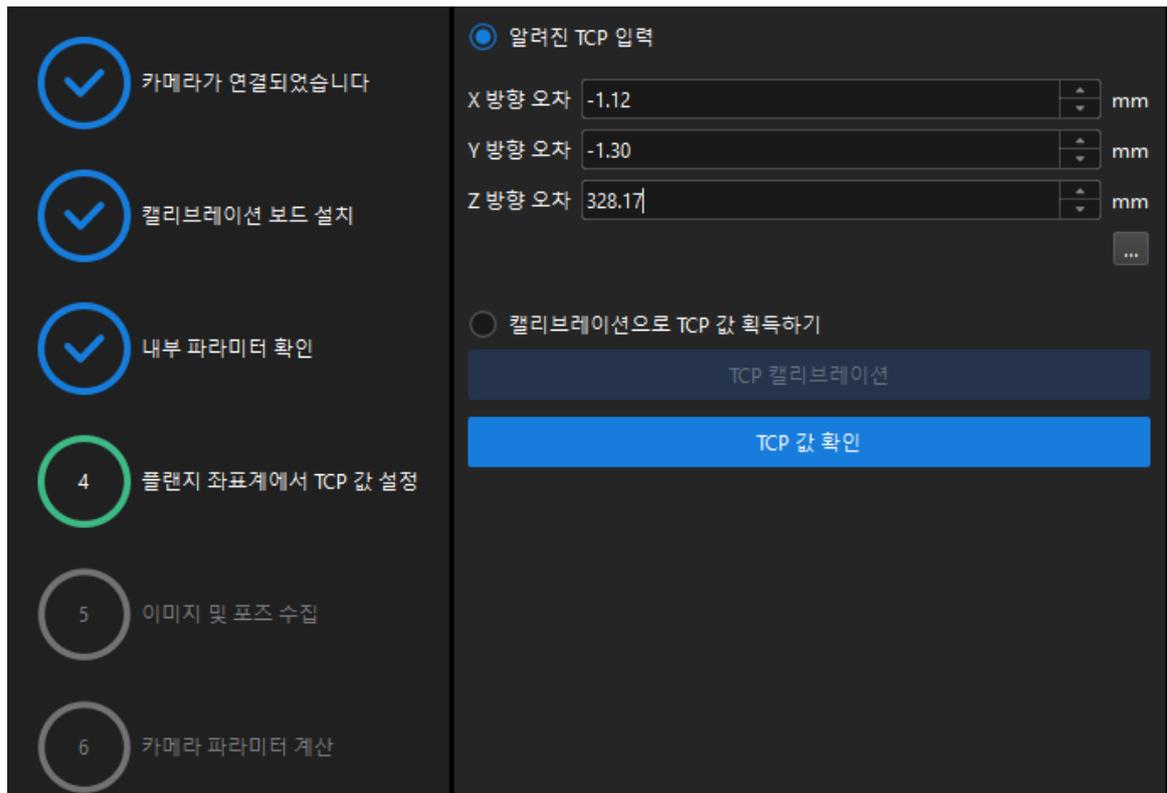
#### 감지 파라미터 수동으로 조정

감지 파라미터를 수동으로 조정할 경우 [ 캘리브레이션 원 감지 파라미터(고급) ]을 클릭하여 감지 파라미터의 값을 변경하면 됩니다.

캘리브레이션 원이 여전히 감지되지 않은 경우 실제 작업 조건에 따라 카메라 관련 파라미터를 조정해야 하며 카메라 파라미터는 [파라미터 조절](#) 내용을 참조하십시오.

#### 플랜지 좌표계에서 TCP 값 설정

1. 플랜지 좌표계에서 TCP 값 설정 단계에서 알려진 TCP 입력 라디오 상자를 선택합니다.



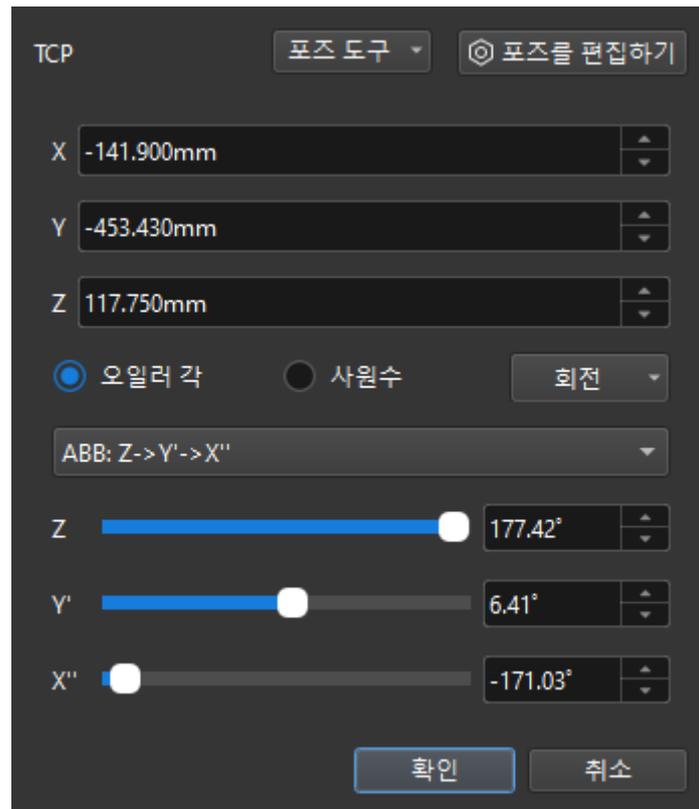
2. [ **TCP 값 확인** ]을 클릭합니다.
3. 하단에서 [ **다음** ] 버튼을 클릭하십시오.



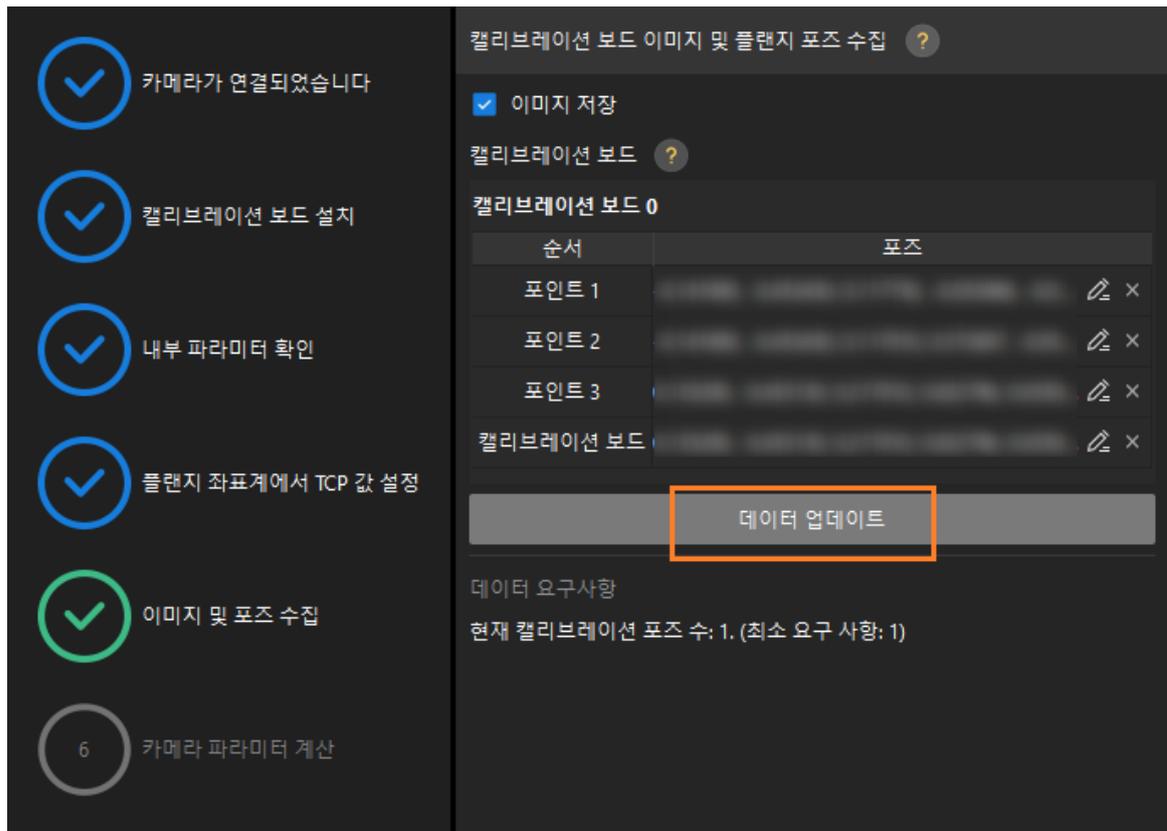
TCP 값을 알 수 없는 경우 **캘리브레이션으로 TCP 값 획득하기** 라디오 상자를 선택하고 [ **TCP 캘리브레이션** ] 버튼을 클릭합니다. TCP 캘리브레이션 도구를 사용하여 TCP 값을 계산할 수 있습니다.

#### 이미지 및 포즈 가져오기

1. **이미지 및 포즈 가져오기** 단계에서 이미지를 저장하기 체크박스를 선택합니다.
2. 끝점이 캘리브레이션 보드의 포인트 1의 교차 중심점에 닿도록 로봇의 움직임을 제어하고 티치 펜던트에서 로봇 플랜지의 포즈를 기록합니다.
3. 포인트 1의  버튼을 클릭하고 팝업된 **로봇 플랜지 포즈를 입력하기** 다이얼로그 박스에서 로봇의 플랜지 포즈를 입력한 후 [ **확인** ] 버튼을 클릭합니다.



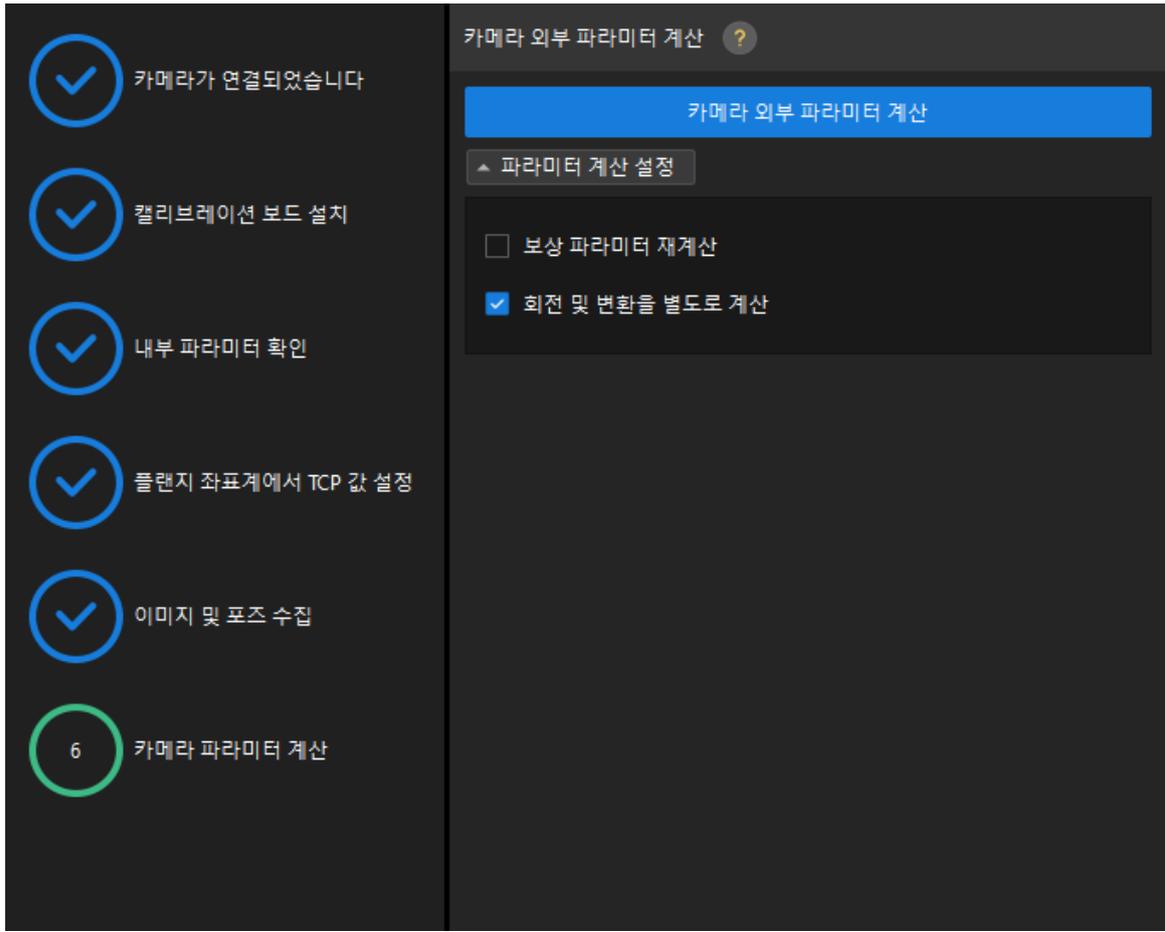
4. 위의 단계를 반복하여 TCP 끝단점이 포인트 2와 포인트 3의 교차 중심점에 닿도록 하고 로봇 플랜지의 포즈를 입력합니다.
5. 현재 캘리브레이션 보드의  버튼을 클릭하여 캘리브레이션 보드의 이미지를 캡처합니다.
6. [ 데이터 업데이트 ] 버튼을 클릭한 후 하단 표시줄에서 [ 다음 ] 버튼을 클릭합니다.



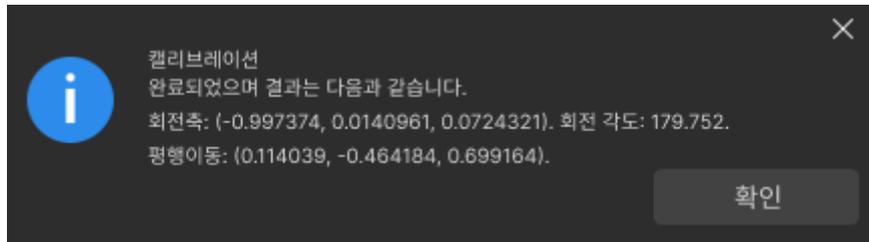
캘리브레이션 절차에서는 외부 파라미터를 계산하기 위해 최소 3개의 비동일선 지점을 터치해야 합니다.

### 카메라 파라미터를 계산하기

1. 카메라 파라미터를 계산하기 단계에서 **회전 및 변환을 별도로 계산** 파라미터를 선택한 다음 [ **카메라 외부 파라미터 계산** ] 버튼을 클릭하십시오.



2. 팝업한 캘리브레이션 성공 다이얼로그 박스에서 [ 확인 ] 버튼을 클릭합니다.



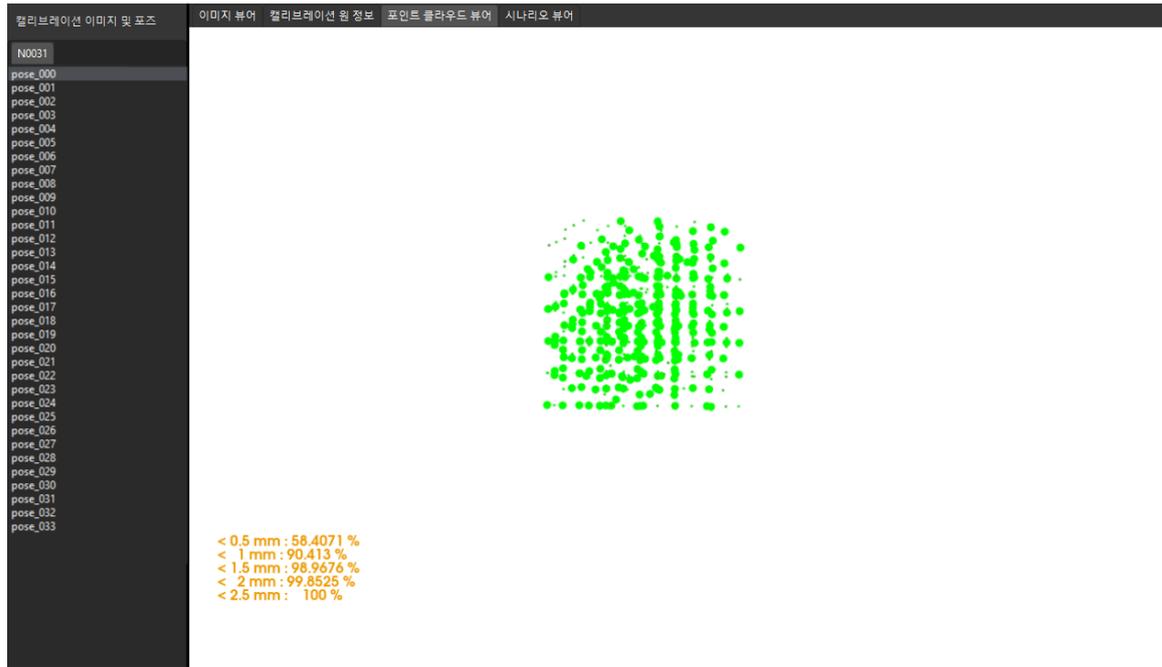
3. 오른쪽 **포인트 클라우드 뷰어** 패널에서 캘리브레이션 오차 포인트 클라우드를 확인합니다.



오차 포인트 클라우드는 각 캘리브레이션 포즈의 캘리브레이션 보드 원의 진릿값과 계산된 값 사이의 편차를 표시하는 데 사용됩니다.

4. 캘리브레이션 정확도가 예상 요구 사항을 충족하는지 확인합니다.

캘리브레이션의 정확도를 대략적으로 확인하기 위해 100%를 차지하는 오차 값을 찾으십시오. 예를 들어, 아래 그림의 정확도는 2.5mm 미만입니다.



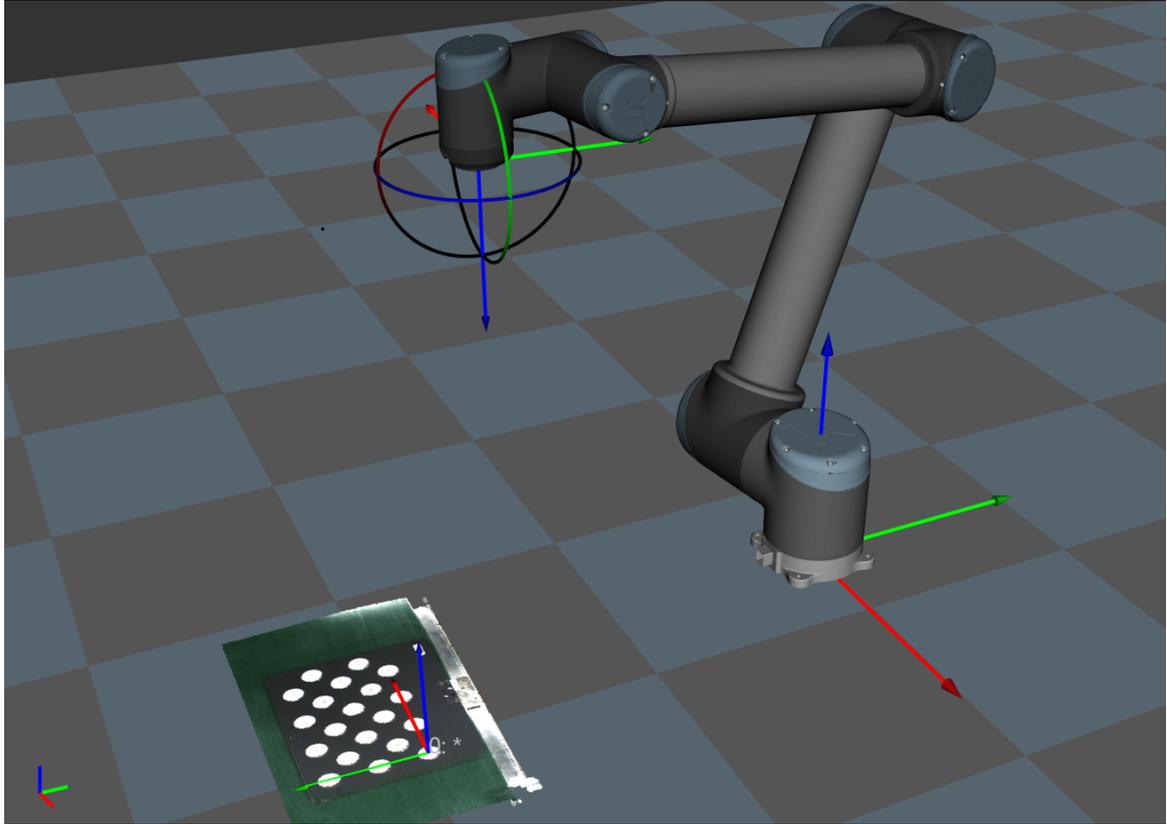
캘리브레이션 정확도를 개선해야 하는 경우 [캘리브레이션 결과 분석](#) 내용을 참조하십시오.

### 캘리브레이션 결과 검증 및 저장

카메라 외부 파라미터를 계산한 후 시나리오 뷰어에서 고정점을 기준으로 하는 캘리브레이션 보드 포인트 클라우드(캘리브레이션 원 십자의 중심점)의 움셋을 보고 캘리브레이션 결과를 대략적으로 판단할 수 있습니다. 구체적인 작업은 다음과 같습니다.

1. 캘리브레이션 보드를 고정된 위치에 놓습니다.
2. Mech-Viz 소프트웨어를 열고 고정점을 추가하고 캘리브레이션 보드에 있는 캘리브레이션 원 중 하나의 교차 중심점과 겹치도록 합니다.
  - a. 스텝 라이브러리에서 "이동" 스텝을 찾아 프로젝트 편집 영역으로 드래그합니다.
  - b. 이 스텝을 선택한 다음 파라미터 편집 영역에서 **이동 목표 유형** 파라미터를 "작업물 포즈"로 설정하고 포즈의 X/Y/Z 좌표를 조정하여 포즈가 캘리브레이션 보드의 캘리브레이션 원 중 하나의 교차 중심점과 겹치도록 합니다.
3. 로봇을 제어하여 카메라 포즈를 여러 번 변경하고 캘리브레이션 프로세스의 **카메라 파라미터를 계산하기** 스텝에서 [ **외부 파라미터 재계산** ] 버튼을 클릭합니다. 이 작업은 카메라가 사진을 캡처하도록 트리거합니다.
4. 시나리오 뷰어에서 캘리브레이션 보드의 포인트 클라우드가 고정점을 기준으로 분명히 움셋되는지 여부를 관찰합니다.

캘리브레이션 보드의 포인트 클라우드에 고정점에 대한 명확한 움셋이 없으면 캘리브레이션 결과를 사용할 수 있음을 의미합니다.



1. EIH 시나리오에서 로봇이 연결되지 않은 상태에서 실시간 로봇 포즈를 획득할 수 없습니다. 로봇 현재의 촬영 포인트 포즈를 입력해야 Mech-Viz에서 표시된 포인트 클라우드를 확인할 수 있습니다.
2. 시나리오 뷰어 외에도 Mech-Viz 소프트웨어에서 캘리브레이션 보드의 포인트 클라우드가 고정점에 대해 명확한 움셋을 가지고 있는지 여부를 확인할 수 있습니다.

1. 하단 표시줄에서 [ 저장 ] 버튼을 클릭하고 팝업한 **캘리브레이션 파일 저장** 다이얼로그 박스에서 [ 확인 ] 버튼을 클릭합니다. 카메라 캘리브레이션 결과는 프로젝트의 "calibration" 디렉터리에 자동으로 저장됩니다.

이로써 캘리브레이션 프로세스가 완료됩니다.

#### 4.4.8. 트러스 로봇의 핸드-아이 캘리브레이션 완료

이 부분에서는 트러스 로봇의 핸드-아이 캘리브레이션을 완료하는 방법에 대해 설명합니다.



트러스 로봇의 핸드-아이 캘리브레이션 관련 개념과 외부 파라미터 파일의 사용과 관련된 내용은 [트러스 로봇의 핸드-아이 캘리브레이션 설명](#)을 참조하십시오.

##### 캘리브레이션 사전 준비

핸드-아이 캘리브레이션을 수행하기 전에 다음 준비를 완료해야 합니다.

- Mech-Mind 비전 시스템 구축을 완료합니다.
- 캘리브레이션에 필요한 재료를 준비합니다.
- 캘리브레이션 보드 포인트 클라우드 이미징을 조정합니다.

## 비전 시스템 전체 구축 완료

Mech-Mind 비전 시스템 구축을 완료하려면 비전 시스템 구축 부분 내용을 참조하십시오.

핸드-아이 캘리브레이션에는 Mech-Eye Viewer, Mech-Center, Mech-Vision 및 Mech-Viz 소프트웨어가 필요합니다. 위의 소프트웨어가 설치되고 최신 버전으로 업데이트되었는지 확인하십시오.

### 캘리브레이션에 필요한 재료를 준비

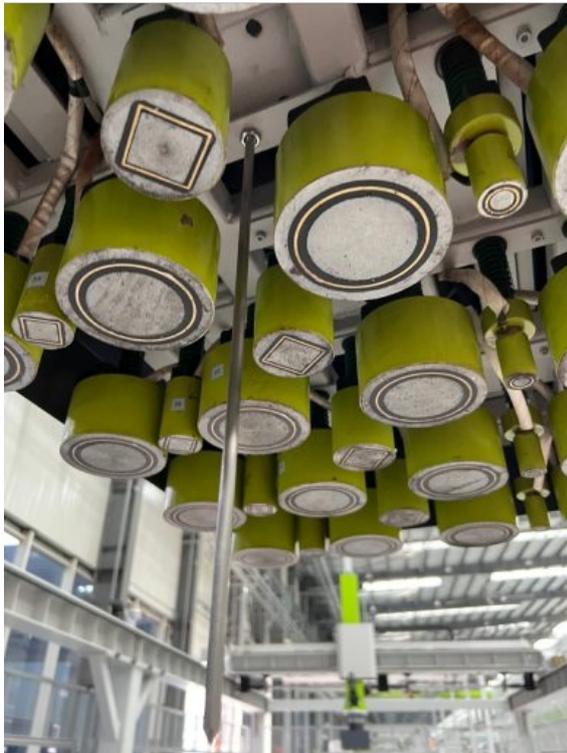
트러스 로봇의 핸드-아이 캘리브레이션은 다음과 같은 재료들이 필요합니다.

- 캘리브레이션 보드
- 끝단점
- 줄자

다음 요구 사항에 따라 캘리브레이션 보드를 준비하십시오.

- 캘리브레이션 보드에서 도트가 잘 보이고 눈에 띄는 굵은 흔적이 없으며 캘리브레이션 보드가 뒤틀리지 않아야 합니다.
- 트러스 로봇의 핸드-아이 캘리브레이션 시나리오에서 캘리브레이션 보드를 작업대의 중앙에 평평하게 놓으십시오.

끝단점은 트러스 로봇의 그리퍼에 설치하십시오. 줄자는 끝단점에서 그리퍼 플랜지 중심까지의 X, Y, Z 방향의 편차를 측정하는 데 사용됩니다.



### 캘리브레이션 보드 포인트 클라우드 이미지를 조정하기

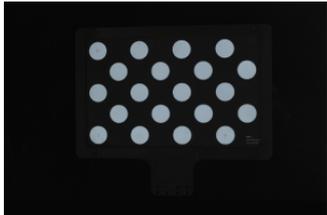
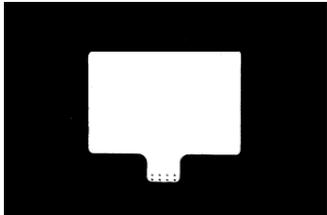
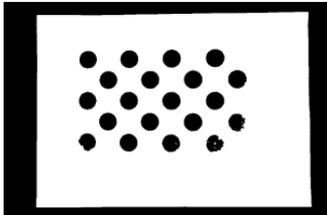
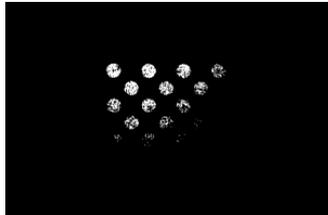
1. Mech-Eye Viewer 소프트웨어를 열고 카메라 파라미터를 조정합니다.
2. 2D 파라미터를 조정함으로써 2D 맵의 캘리브레이션 보드가 뚜렷하고 노출 과다나 노출 부족 등 현상이 없도록 합니다.
3. 3D 파라미터를 조정함으로써 캘리브레이션 보드의 원의 포인트 클라우드를 완전하게 하고 **포인트**

클라우드 후처리에서 **포인트 클라우드 평활화** 및 **노이즈 제거**를 **일반**으로 변경하는 것이 좋으며 포인트 클라우드 변동 범위를 줄입니다.



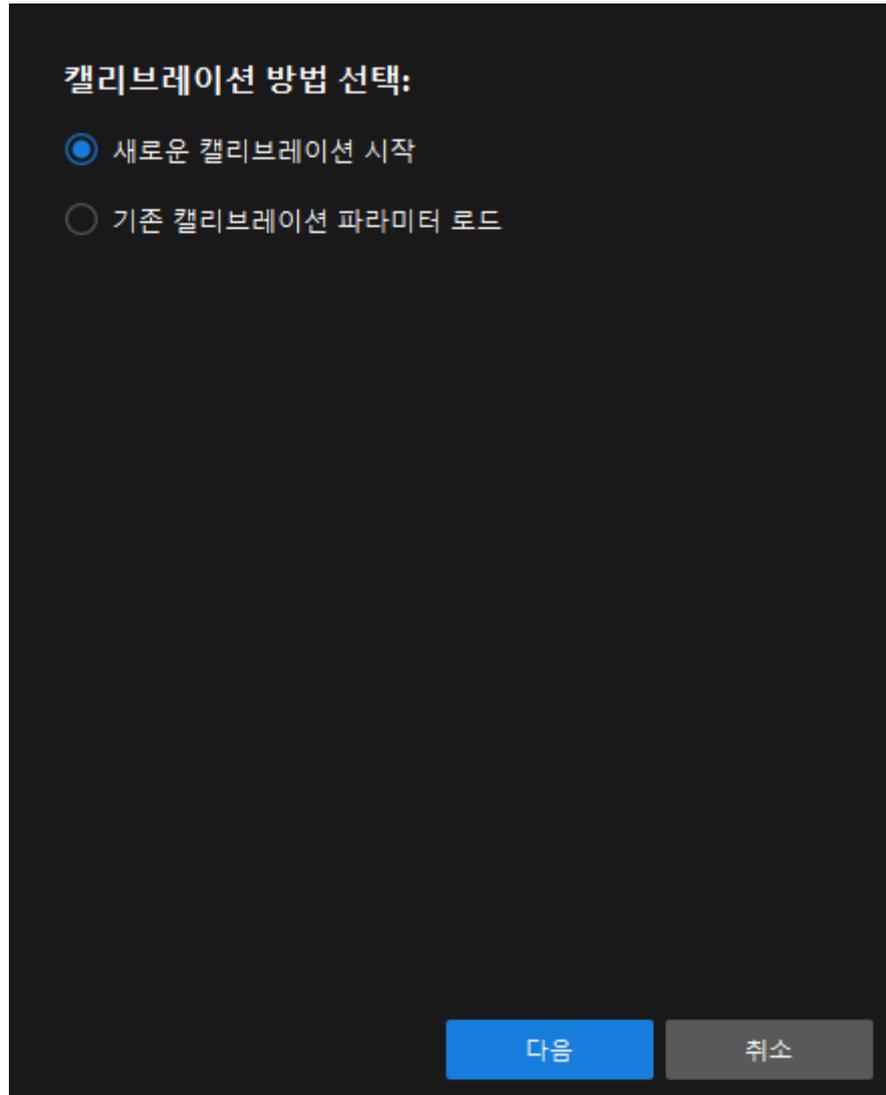
현장에 환경광이 복잡한 경우, 환경광이 2D 맵과 포인트 클라우드에 주는 영향을 줄이기 위해 광원을 차단하거나 보충하기 바랍니다.

4. 위의 단계를 완료하여 캘리브레이션 보드의 포인트 클라우드 이미징 품질이 요구 사항을 충족하는지 확인하십시오.

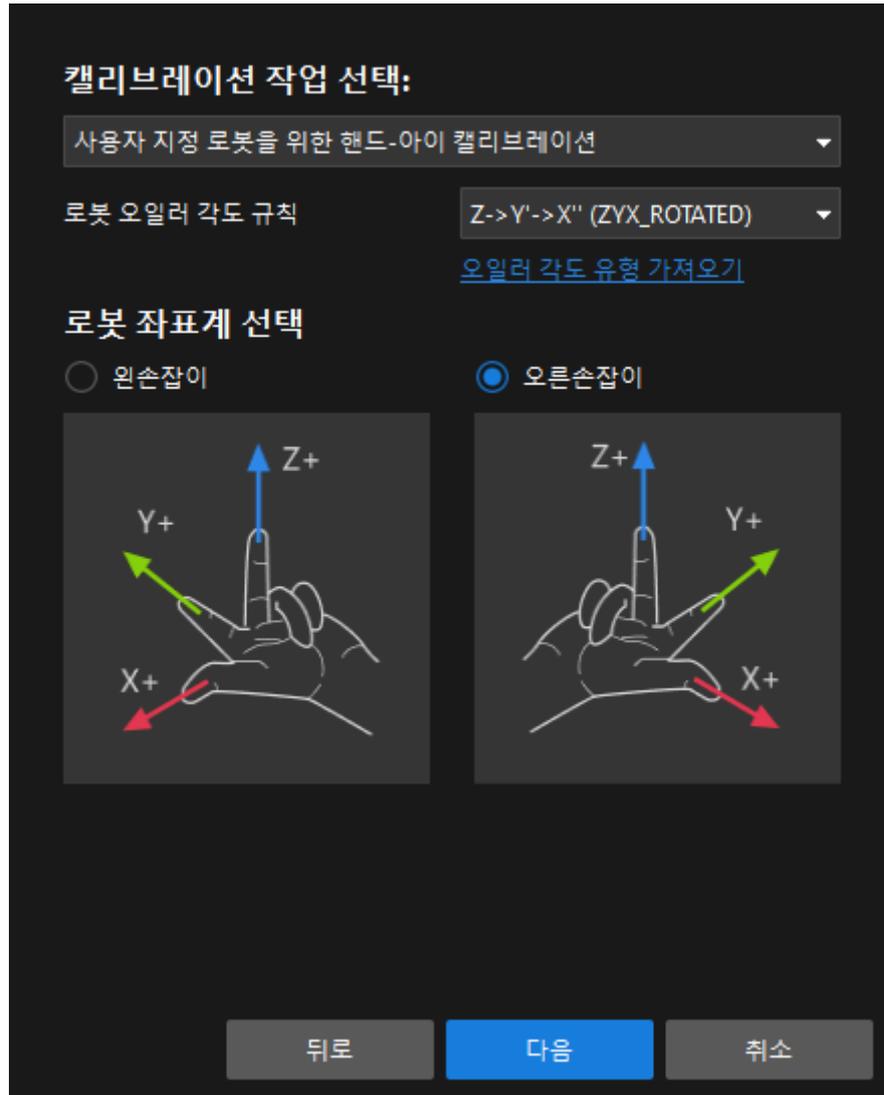
	정상	노출 과다	노출 부족
2D 맵			
포인트 클라우드			

#### 캘리브레이션 사전 구성

1. Mech-Vision 소프트웨어에서 톨 바의 [ **카메라 캘리브레이션(표준)** ] 버튼을 클릭합니다. **캘리브레이션 사전 구성** 창이 팝업됩니다.
2. **캘리브레이션 방법 선택** 창에서 **새로운 캘리브레이션 시작** 라디오 버튼을 선택한 후 [ **다음** ] 버튼을 클릭합니다.

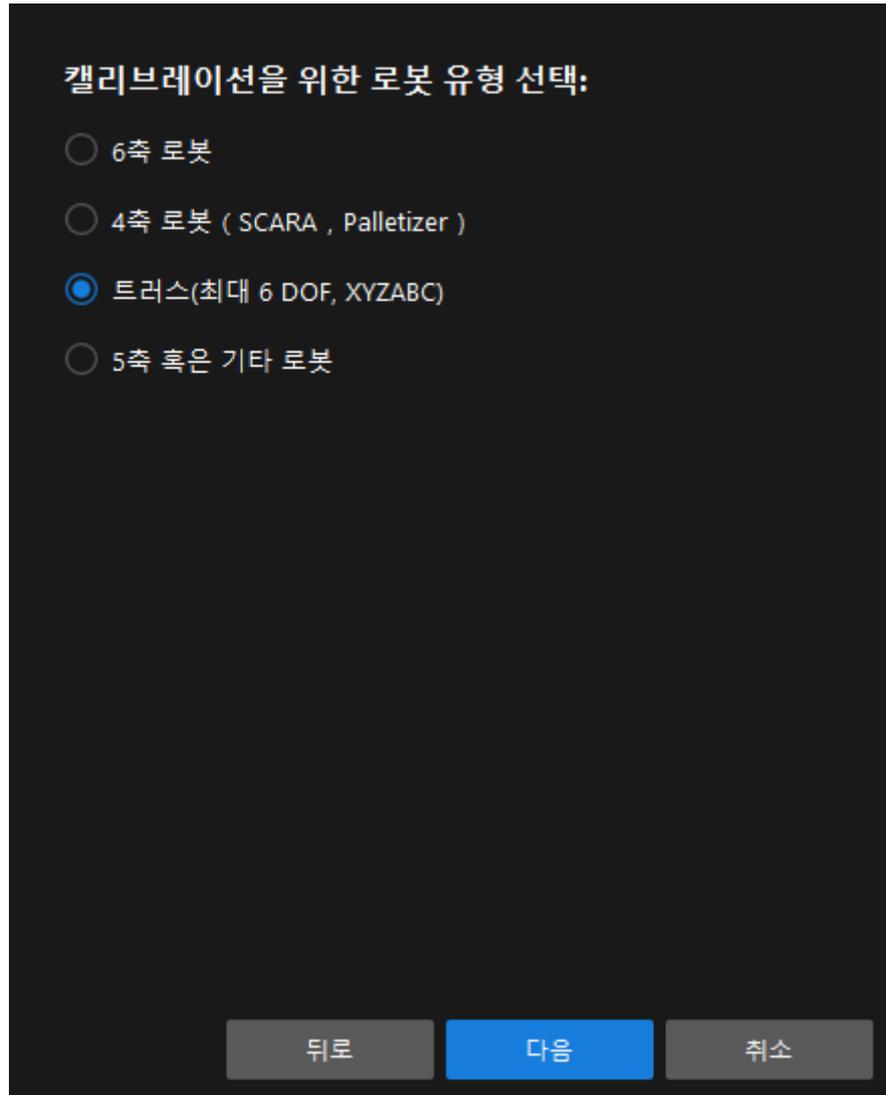


3. **캘리브레이션 작업 선택** 창의 드롭다운 목록 상자에서 **사용자 지정 로봇을 위한 핸드-아이 캘리브레이션**을 선택하고 **로봇 오일러 각도 유형** 파라미터를 지정하여 로봇 좌표계 유형을 선택한 후 [ **다음** ] 버튼을 클릭합니다.

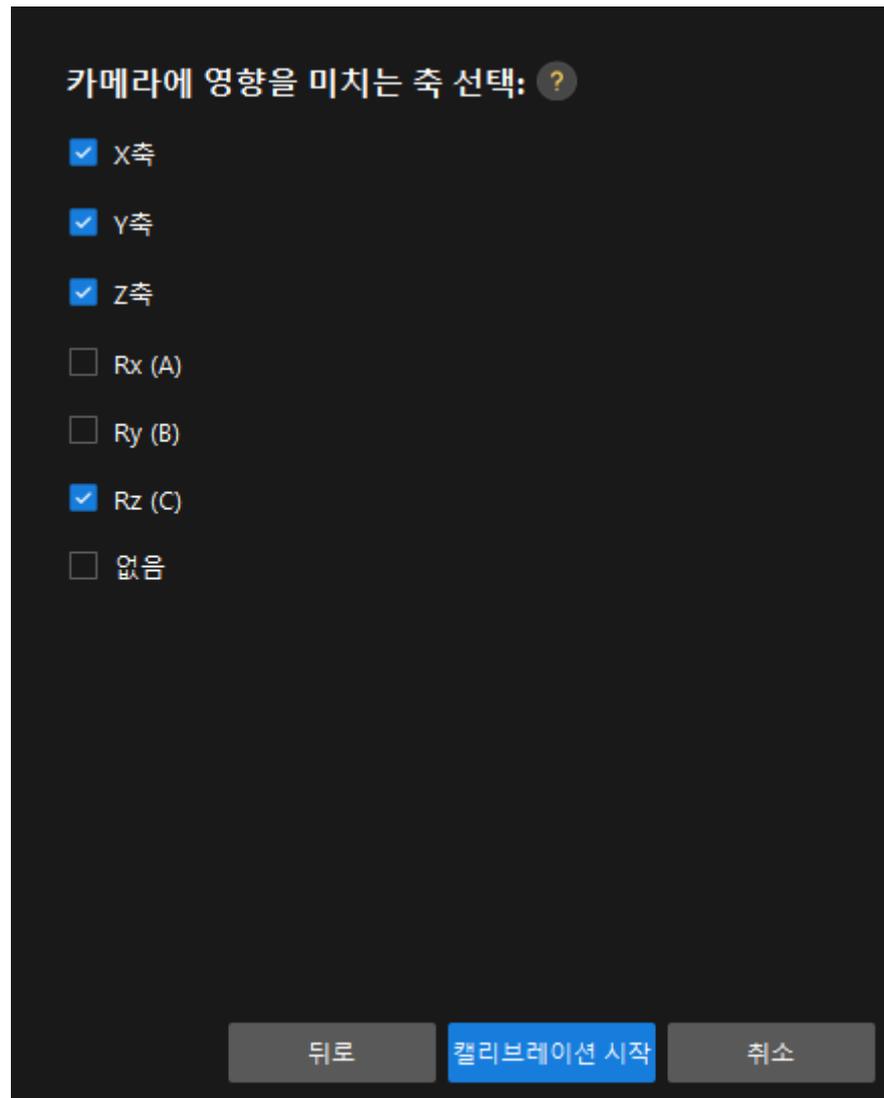


- 로봇 좌표계 유형을 확인하려면 **왼손/오른손 좌표계** 내용을 참조하십시오. 트러스 로봇이 왼손 좌표계를 사용하는 경우 여기에서 왼손 좌표계를 선택하십시오.
- 로봇과 통신 할 때 통신 모듈은 포즈를 송수신 할 때 좌표계를 자동으로 변환합니다. 로봇 측에서 보낸 로봇 포즈를 수신 할 때 왼손 좌표계의 포즈를 오른손 좌표계의 포즈로 자동 변환합니다. 소프트웨어에서 출력한 포즈를 로봇 측으로 전송하기 전에 오른손 좌표계의 포즈를 왼손 좌표계의 포즈로 변환합니다.
- 캘리브레이션 프로세스에서 로봇 플랜지 포즈를 수동으로 입력해야 하는 경우 y 축 값의 반대를 취해야 합니다.

4. 캘리브레이션을 위한 로봇 유형 선택 창에서 트러스 로봇(최대 6 DOF, XYZABC) 라디오 상자를 선택한 후 [ 다음 ] 버튼을 클릭합니다.



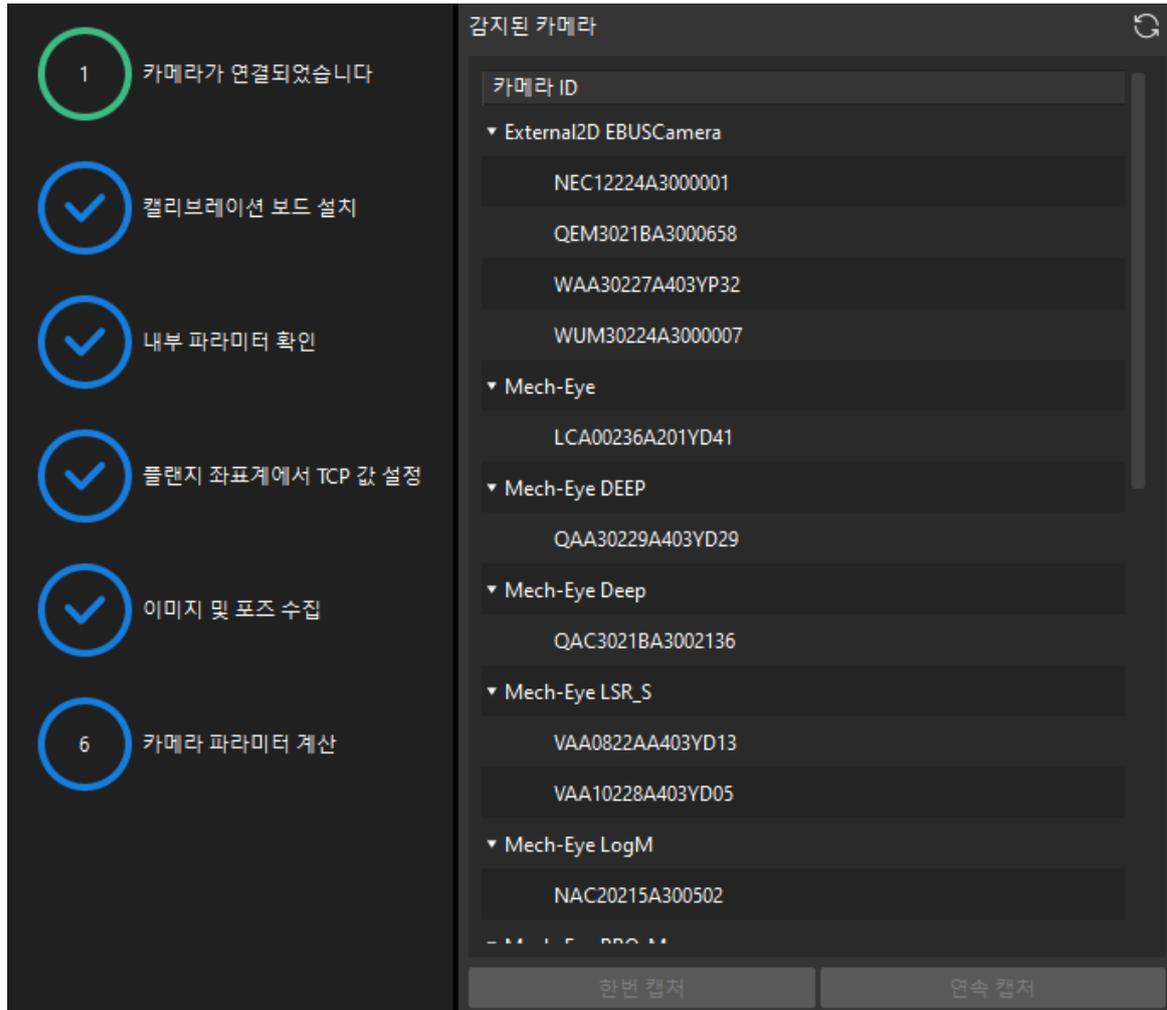
5. 카메라에 영향을 미치는 축 선택 창에서 필요에 따라 해당 확인란 (예 : X축, Y축, Z축, Rz)을 선택한 다음 [ 캘리브레이션 시작 ] 버튼을 클릭하십시오.



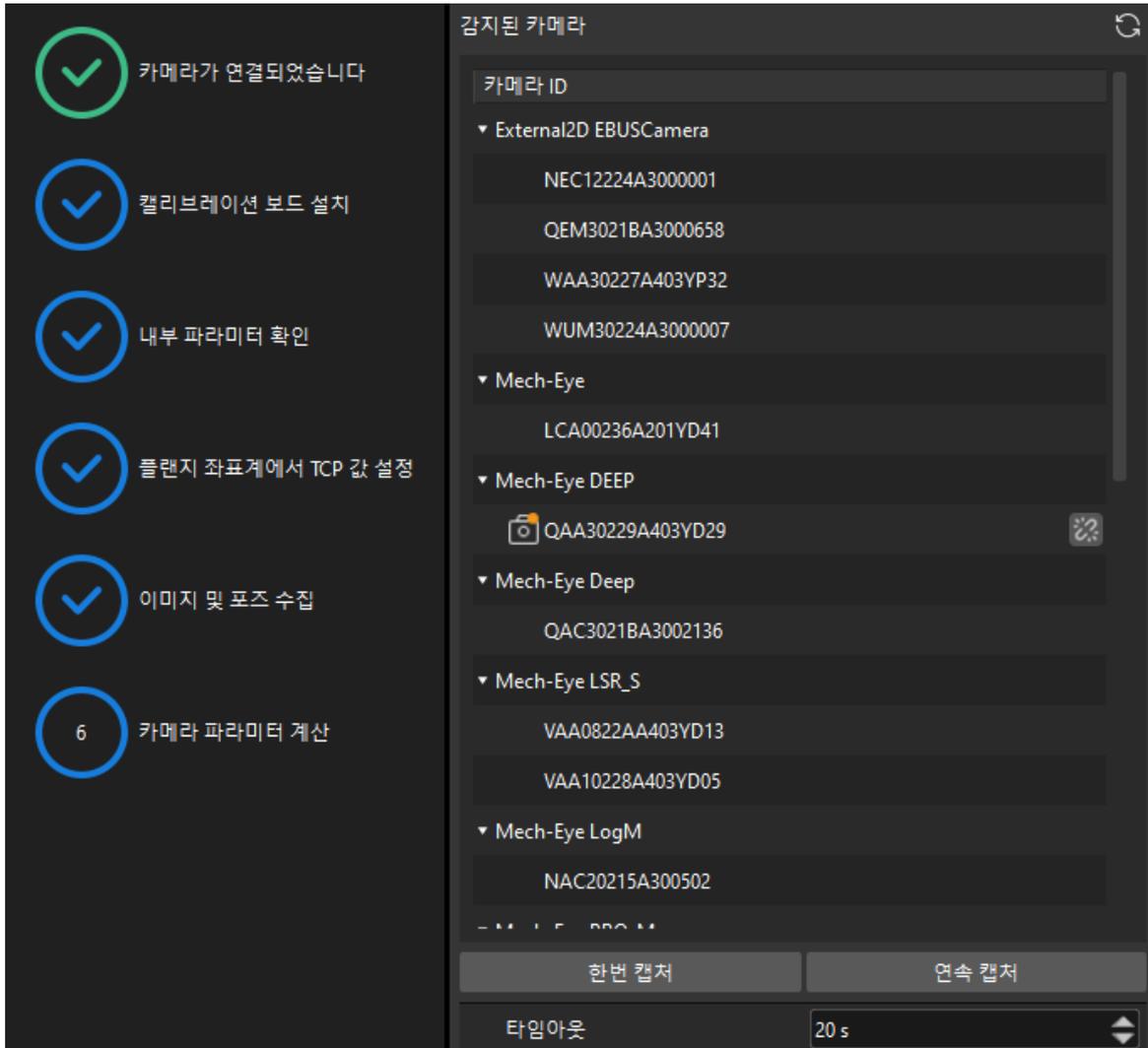
## 캘리브레이션 프로세스

### 카메라 연결

1. **카메라 연결** 단계에서 **감지된 카메라** 리스트에서 연결할 카메라를 찾아  버튼을 클릭합니다.



2. 카메라를 연결한 후 [한번 캡처] 또는 [연속 캡처] 버튼을 선택할 수 있습니다.



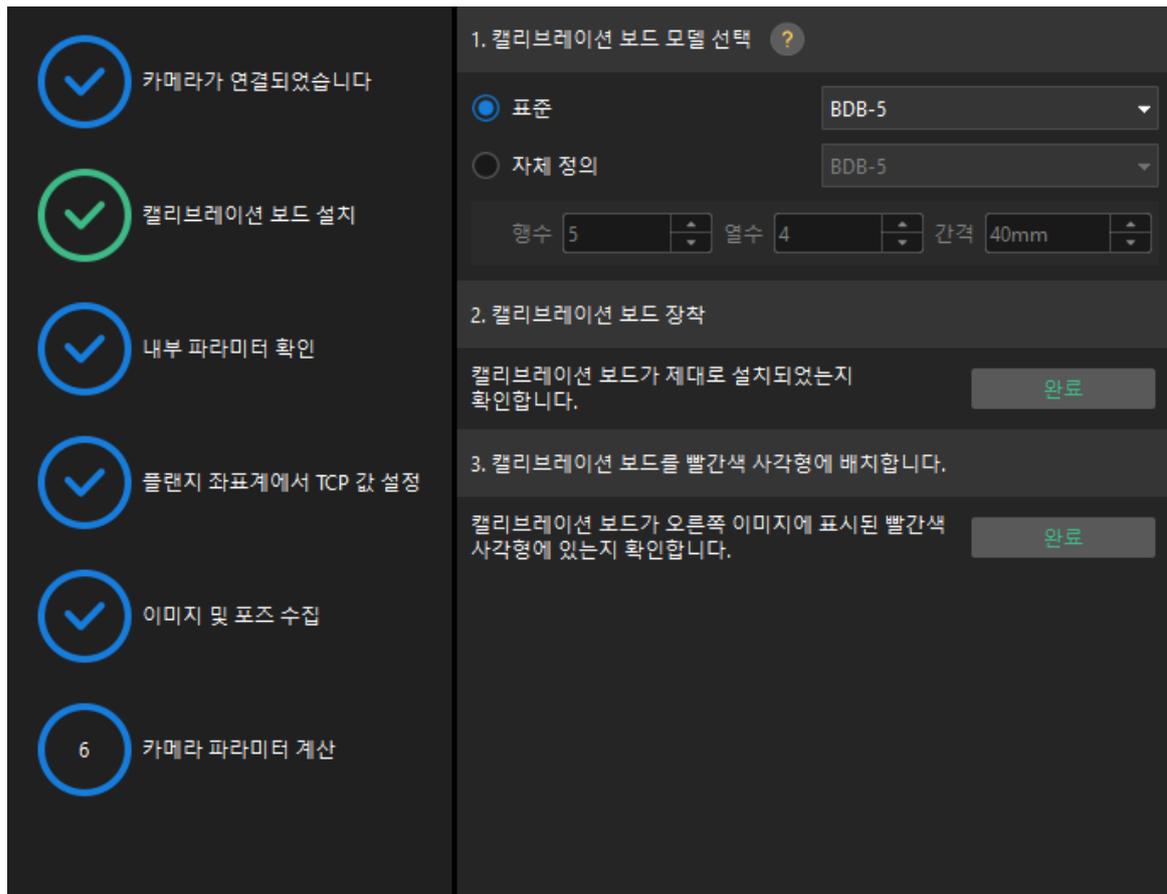
3. 이미지 뷰어에서 카메라가 캡처한 2D 맵과 덤스 맵이 캘리브레이션 요구 사항을 충족하는지 확인한 다음 하단 표시줄에서 [ 다음 ] 버튼을 클릭합니다.



캡처한 이미지가 캘리브레이션 요구 사항을 충족하지 않으면 Mech-Eye Viewer 소프트웨어를 열어 **카메라의 2D 및 3D 노출 파라미터를 조정**하고 다시 캡처해야 합니다.

#### 캘리브레이션 보드 설치

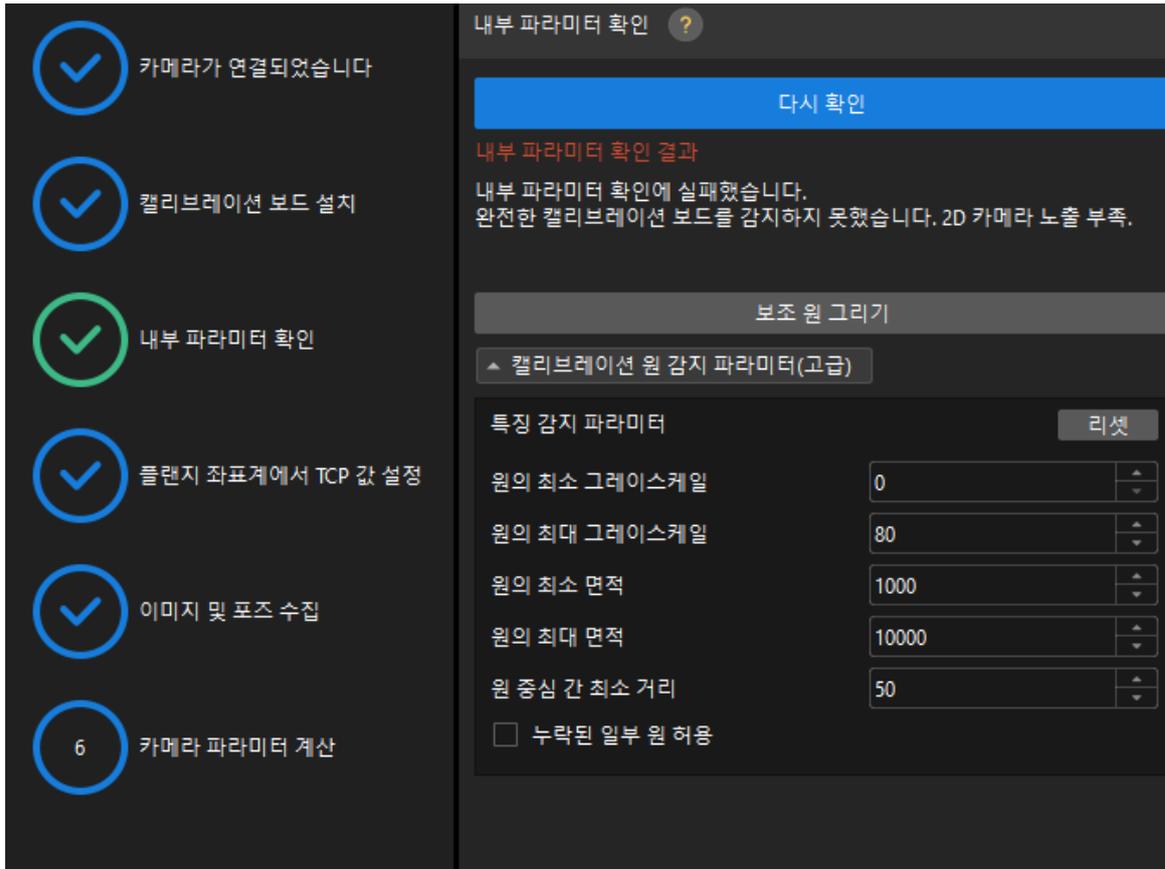
1. 캘리브레이션 보드 설치 단계의 1. 캘리브레이션 보드 유형 선택 영역에서 **표준** 라디오 버튼을 선택한 후 캘리브레이션 보드 모델 레이블에 따라 해당 캘리브레이션 유형을 선택합니다.



2. 캘리브레이션 보드가 작업 평면 중앙에 평평하게 놓여 있는지 확인한 후 **2. 캘리브레이션 보드 설치** 영역에서 [ **확인** ] 버튼을 클릭합니다.
3. 캘리브레이션 보드가 카메라 시야의 중앙(빨간색 사각형 내)에 있는지 확인한 다음 **3.캘리브레이션 보드를 빨간색 사각형에 배치** 영역에서 [ **확인** ] 버튼을 클릭합니다.
4. 캘리브레이션 보드 관련 작업을 모두 완료한 후 하단 표시줄에서 [ **다음** ] 버튼을 클릭합니다.

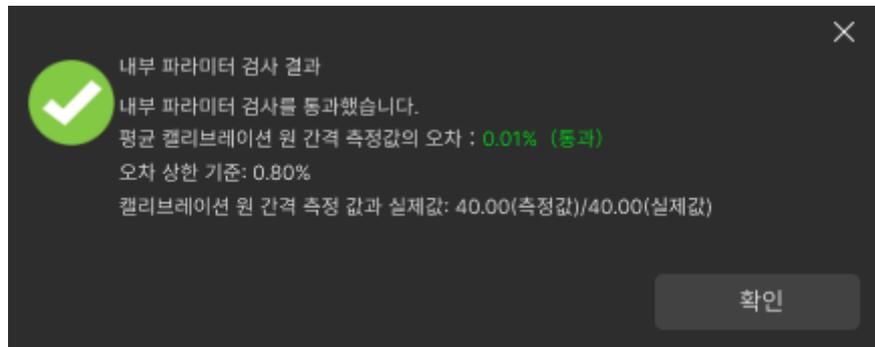
#### 카메라 내부 파라미터를 검사하기

1. 내부 파라미터 검사 단계에서 [ **검사 시작** ] 버튼을 클릭합니다.



## 2. 카메라의 내부 파라미터 검사 결과를 확인합니다.

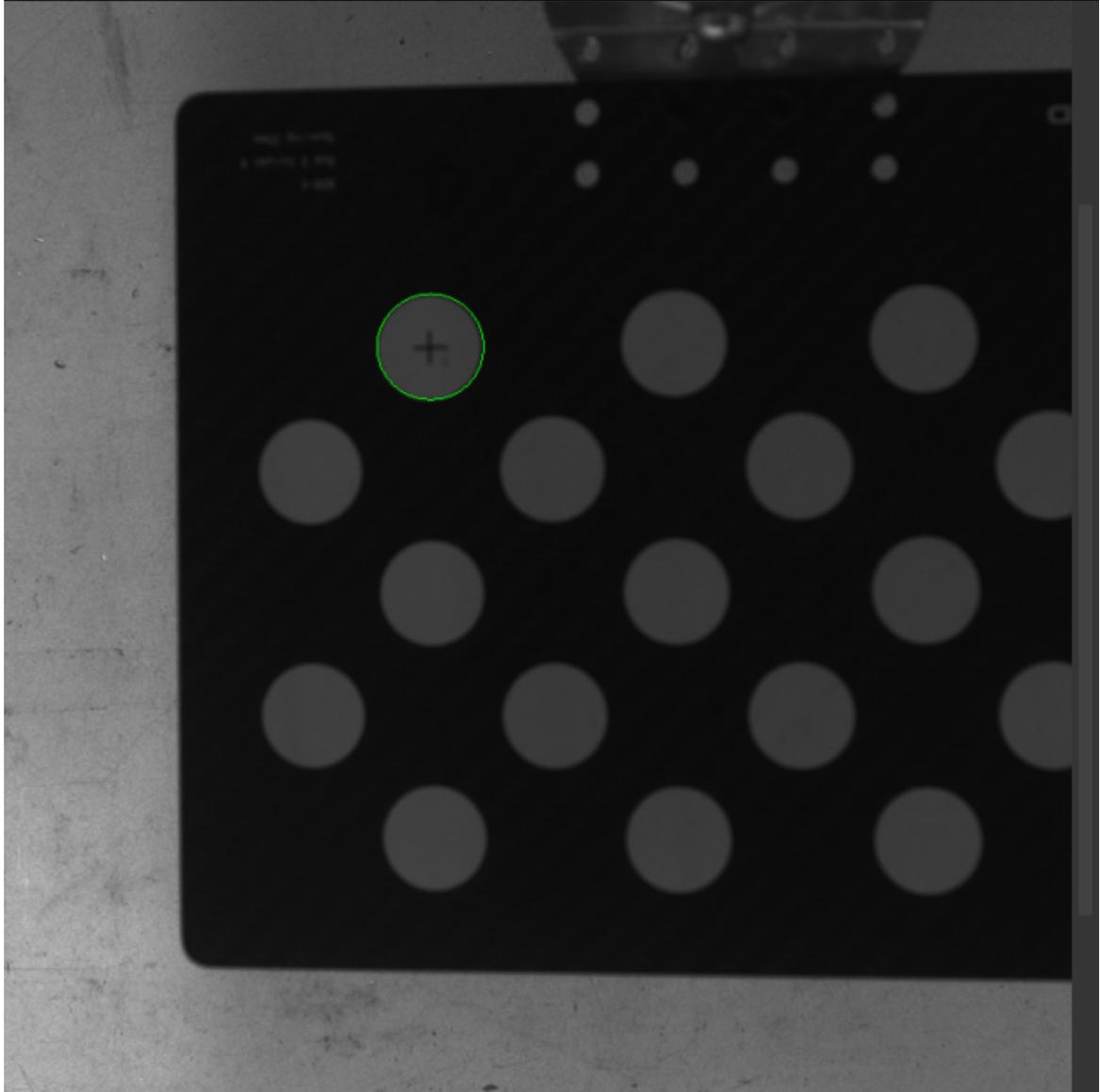
- 카메라의 내부 파라미터 검사에 성공하면 [ 확인 ]을 클릭한 후 [ 다음 ] 버튼을 클릭하십시오.



- 카메라의 내부 파라미터 검사에 실패하면 보조원을 그리거나 캘리브레이션 원을 수동으로 편집하여 파라미터를 감지한 다음 [ 다시 검사 ] 버튼을 클릭하십시오.

### 보조원을 그리기

1. 보조원을 그리기로 선택한 경우 [ 보조원 그리기 ] 버튼을 클릭하십시오.
2. 오른쪽 이미지 뷰어 패널에서 마우스 오른쪽으로 캘리브레이션 보드 이미지를 클릭하고 해당 창 확인란을 언체크 한 다음 **Ctrl** 키를 누른 상태에서 롤러를 드래그하여 이미지를 적절한 크기로 조정하십시오.
3. 마우스 포인터를 캘리브레이션 원의 중간 줄 지점으로 이동하고 왼쪽 마우스 버튼을 누른 다음 보조원에 캘리브레이션 원을 완전히 포함시킨 다음 해제하십시오.



4. [ 다시 검사 ] 버튼을 클릭하고 카메라 내부 파라미터 검사가 통과되었는지 확인하십시오.

#### 감지 파라미터 수동으로 조정

감지 파라미터를 수동으로 조정할 경우 [ 캘리브레이션 원 감지 파라미터(고급) ]을 클릭하여 감지 파라미터의 값을 변경하면 됩니다.

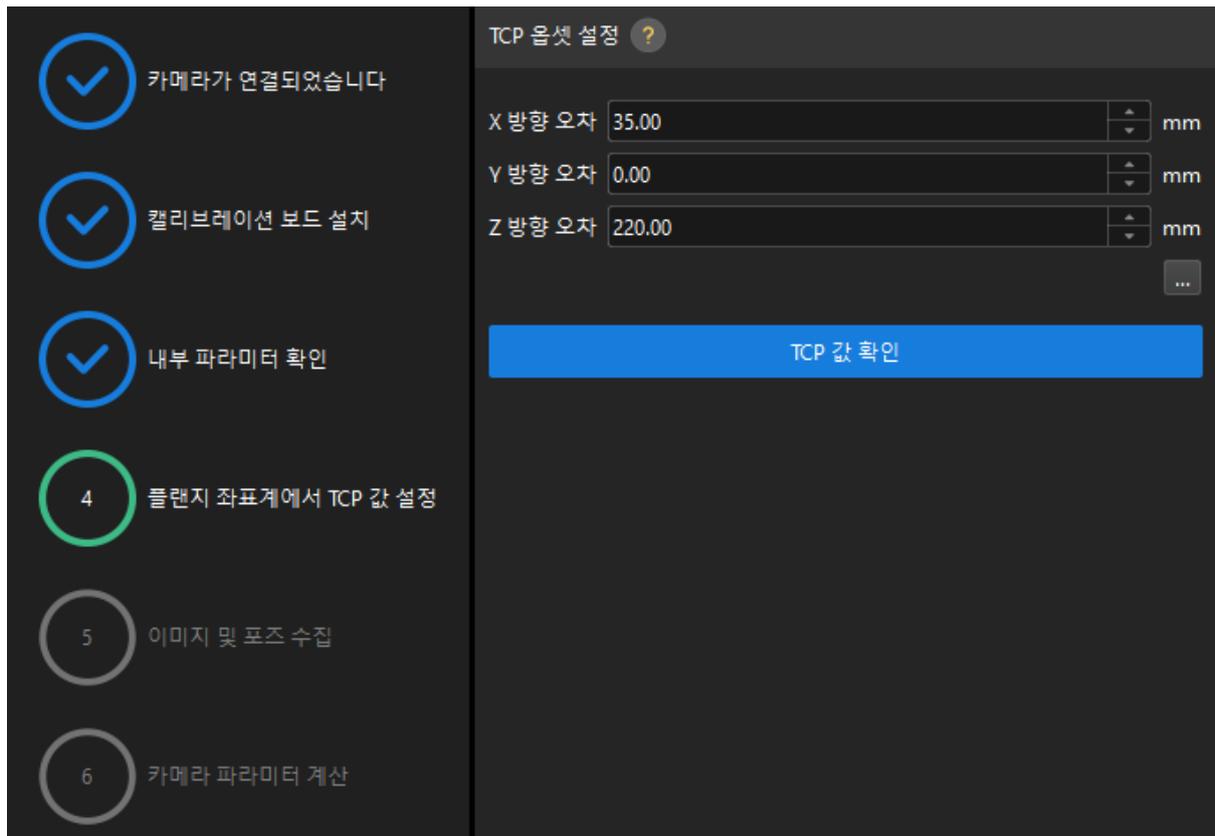
캘리브레이션 원이 여전히 감지되지 않은 경우 실제 작업 조건에 따라 카메라 관련 파라미터를 조정해야 하며 카메라 파라미터는 [파라미터 조정](#) 내용을 참조하십시오.

#### 플랜지 좌표계에서 TCP 값 설정



이 단계를 수행하기 전에 줄자를 사용하여 끝단점에서 그리퍼의 플랜지 중심까지의 X, Y, Z 방향의 편차를 측정하며 단위는 mm입니다.

플랜지 좌표계에서 TCP 값 설정 단계에서 X, Y, Z 텍스트 상자에 측정하여 얻은 X, Y, Z 방향의 옵셋을 입력하고 [ TCP 값 확인 ]을 클릭한 다음 하단 열에서 [ 다음 ] 버튼을 클릭하십시오.



**i** | 트러스 로봇이 왼손 좌표계인 경우 y 축 오프셋의 반대를 취한 후 입력합니다.

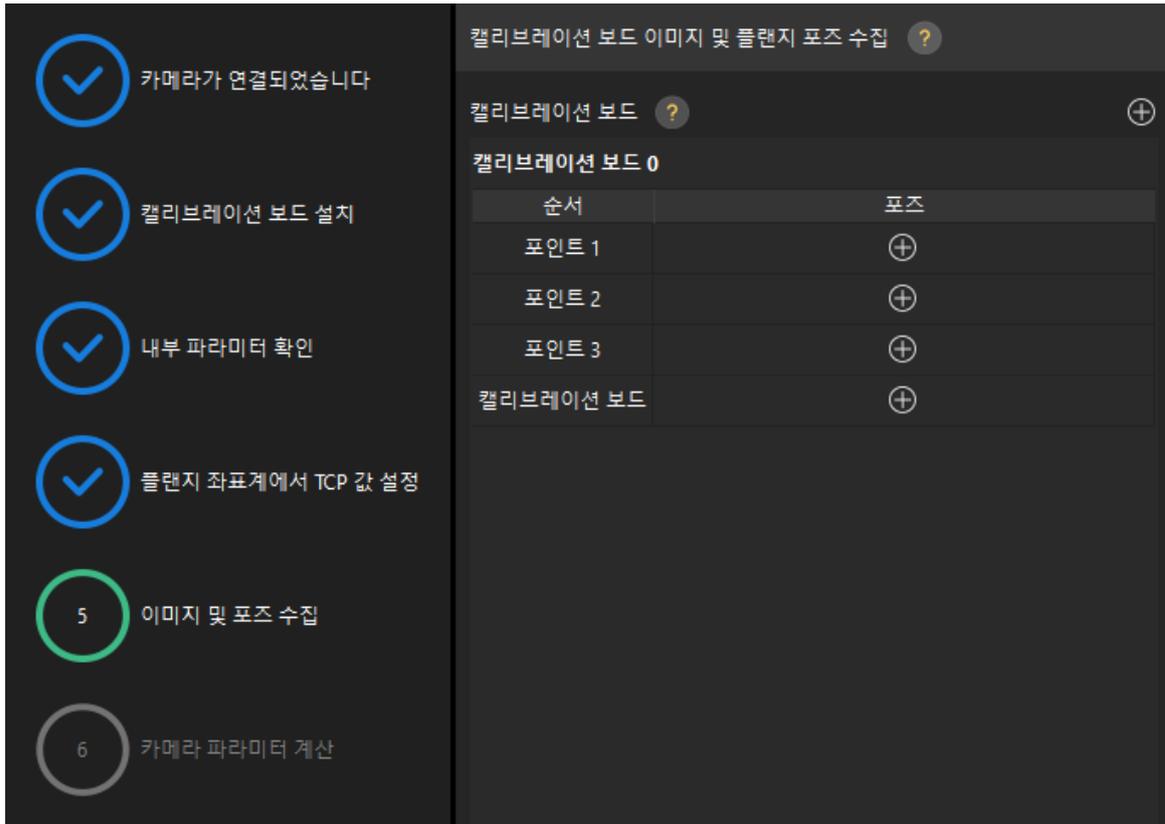
#### 이미지 및 포즈 가져오기

1. 그리퍼의 끝점이 캘리브레이션 보드의 포인트 1의 교차 중심점에 당도록 트러스 로봇의 움직임을 제어하고 티치 펜던트에서 로봇 플랜지의 포즈를 기록합니다.
2. **이미지 및 포즈 가져오기** 단계에서 포인트 1의 **+** 버튼을 클릭하고 팝업된 **로봇 플랜지 포즈를 입력하기** 다이얼로그 박스에서 로봇의 플랜지 포즈를 입력한 후 **[확인]** 버튼을 클릭합니다.

**i** | 트러스 로봇이 왼손 좌표계인 경우 로봇 플랜지 포즈를 입력할 때 먼저 y 축 값의 반대를 취해야 합니다.



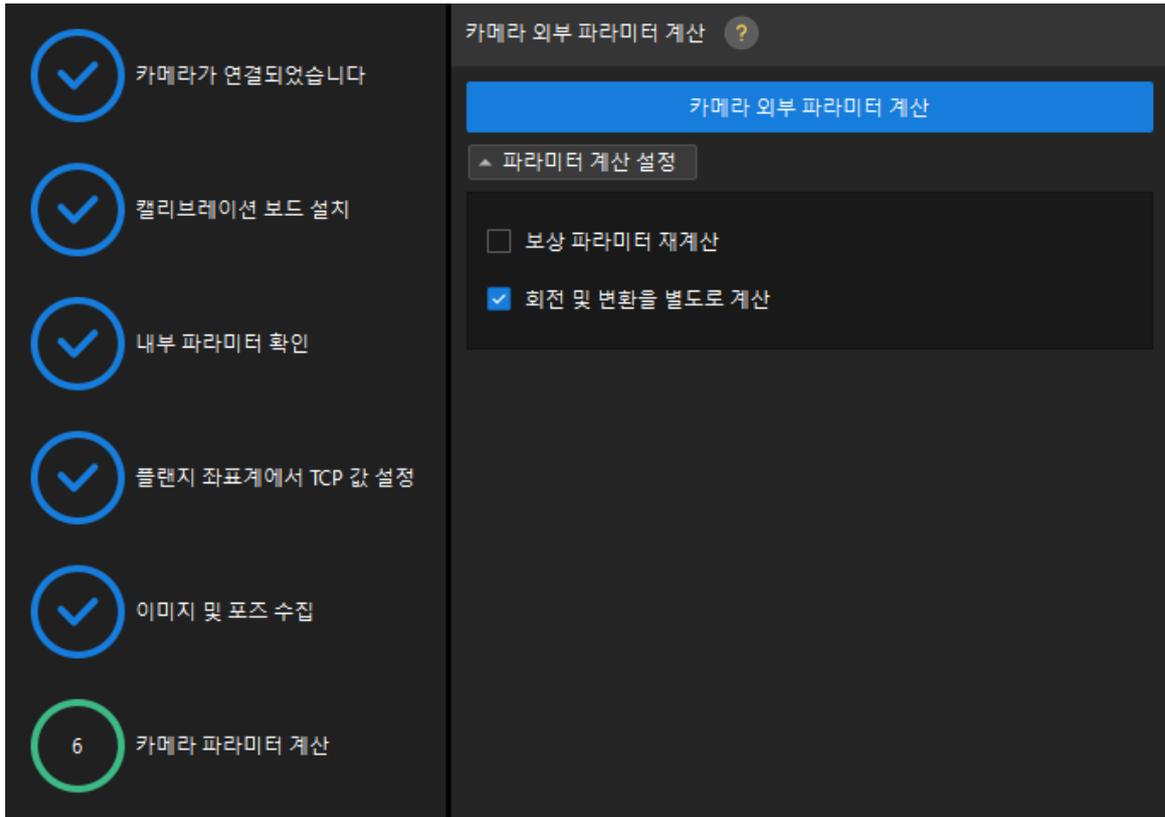
3. 위의 단계를 반복하여 TCP 끝단점이 포인트 2와 포인트 3의 교차 중심점에 닿도록 하고 로봇 플랜지의 포즈를 입력합니다.
4. 현재 캘리브레이션 보드의  버튼을 클릭하여 캘리브레이션 보드의 이미지를 캡처합니다.
5. [ 데이터 업데이트 ] 버튼을 클릭한 후 하단 표시줄에서 [ 다음 ] 버튼을 클릭합니다.



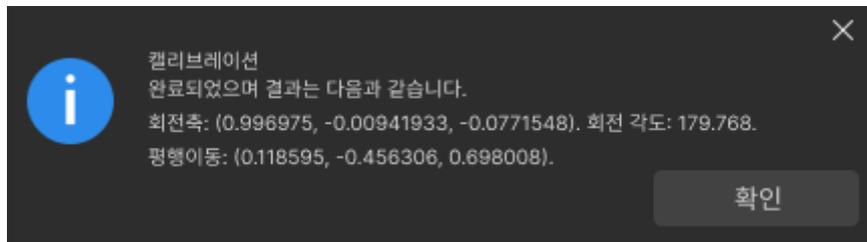
- 캘리브레이션 절차에서는 외부 파라미터를 계산하기 위해 최소 3개의 비동일선 지점을 터치해야 합니다.
- 트러스 로봇에 특정 방향 (예 : Y 방향의 자유도가 부족함)의 자유도가 부족한 경우 여러 캘리브레이션 보드의 포즈를 추가 할 수 있습니다. 우선, 캘리브레이션 보드의 방향을 조정하여 끝단점이 캘리브레이션 보드의 두 지점에 닿을 수 있도록 하고 동시에 로봇 플랜지 포즈를 입력하고 캘리브레이션 보드 이미지를 캡처합니다. 그 다음 Z 방향으로 캘리브레이션 보드를 올린(혹은 낮춤) 다음 X 방향을 따라 캘리브레이션 보드의 포인트를 터치하고 동시에 로봇 플랜지 포즈를 입력하고 캘리브레이션 보드 이미지를 캡처합니다. 캘리브레이션 보드의 새로운 위치가 이전 위치와 크게 다른지 확인하십시오.

## 카메라 파라미터를 계산하기

1. 카메라 파라미터 계산 단계에서 [ 카메라 외부 파라미터 계산 ] 버튼을 클릭합니다.



- 팝업한 캘리브레이션 성공 다이얼로그 박스에서 [ 확인 ] 버튼을 클릭합니다.



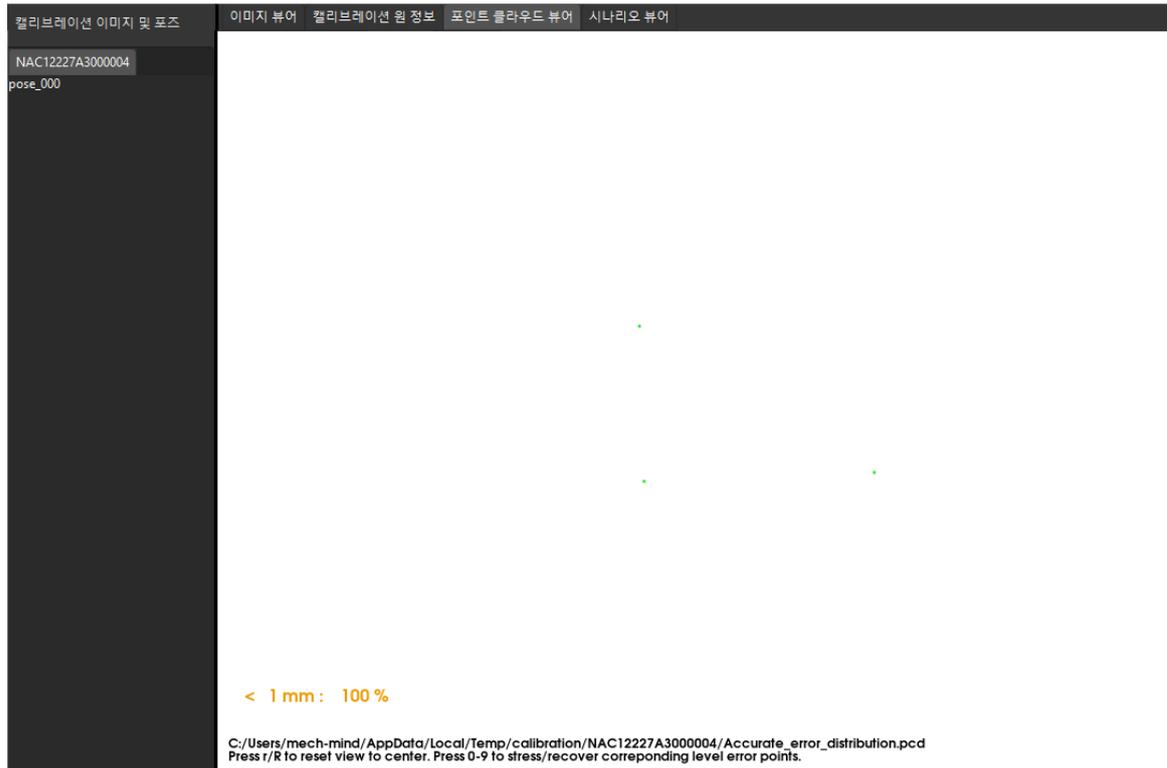
- 오른쪽 **포인트 클라우드 뷰어** 패널에서 캘리브레이션 오차 포인트 클라우드를 확인합니다.



오차 포인트 클라우드는 각 캘리브레이션 포즈의 캘리브레이션 보드 원의 참값과 계산된 값 사이의 편차를 표시하는 데 사용됩니다.

- 캘리브레이션 정확도가 예상 요구 사항을 충족하는지 확인합니다.

100%를 차지하는 오차 값을 찾으면 캘리브레이션 정확도를 대략 확인할 수 있습니다. 예를 들어, 아래 그림의 정확도는 1 mm 미만입니다.



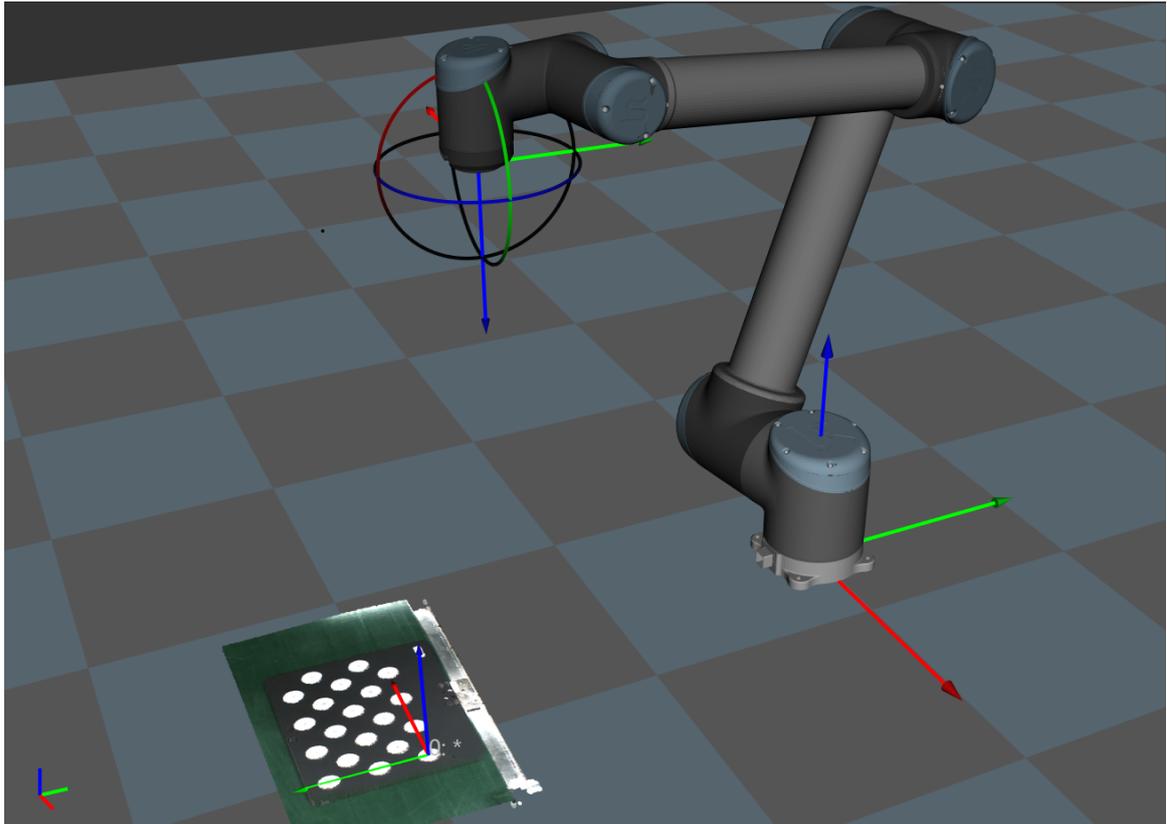
캘리브레이션 정확도를 개선해야 하는 경우 [캘리브레이션 결과 분석](#) 내용을 참조하십시오.

### 캘리브레이션 결과 검증 및 저장

카메라 외부 파라미터를 계산한 후 시나리오 뷰어에서 고정점을 기준으로 하는 캘리브레이션 보드 포인트 클라우드(캘리브레이션 원 십자의 중심점)의 움직임을 보고 캘리브레이션 결과를 대략적으로 판단할 수 있습니다. 구체적인 작업은 다음과 같습니다.

1. 캘리브레이션 보드를 고정된 위치에 놓습니다.
2. Mech-Viz 소프트웨어를 열고 고정점을 추가하고 캘리브레이션 보드에 있는 캘리브레이션 원 중 하나의 교차 중심점과 겹치도록 합니다.
  - a. 스텝 라이브러리에서 "이동" 스텝을 찾아 프로젝트 편집 영역으로 드래그합니다.
  - b. 이 스텝을 선택한 다음 파라미터 편집 영역에서 **이동 목표 유형** 파라미터를 "작업물 포즈"로 설정하고 포즈의 X/Y/Z 좌표를 조정하여 포즈가 캘리브레이션 보드의 캘리브레이션 원 중 하나의 교차 중심점과 겹치도록 합니다.
3. 로봇을 제어하여 카메라 포즈를 여러 번 변경하고 캘리브레이션 프로세스의 **카메라 파라미터를 계산하기** 스텝에서 [ **외부 파라미터 재계산** ] 버튼을 클릭합니다. 이 작업은 카메라가 사진을 캡처하도록 트리거합니다.
4. 시나리오 뷰어에서 캘리브레이션 보드의 포인트 클라우드가 고정점을 기준으로 분명히 움직이는지 여부를 관찰합니다.

캘리브레이션 보드의 포인트 클라우드에 고정점에 대한 명확한 움직임이 없으면 캘리브레이션 결과를 사용할 수 있음을 의미합니다.



1. EIH 시나리오에서 로봇이 연결되지 않은 상태에서 실시간 로봇 포즈를 획득할 수 없습니다. 로봇 현재의 촬영 포인트 포즈를 입력해야 Mech-Viz에서 표시된 포인트 클라우드를 확인할 수 있습니다.
2. 시나리오 뷰어 외에도 Mech-Viz 소프트웨어에서 캘리브레이션 보드의 포인트 클라우드가 고정점에 대해 명확한 움직임을 가지고 있는지 여부를 확인할 수 있습니다.

1. 하단 표시줄에서 [ 저장 ] 버튼을 클릭하고 팝업한 **캘리브레이션 파일 저장** 다이얼로그 박스에서 [ 확인 ] 버튼을 클릭합니다. 카메라 캘리브레이션 결과는 프로젝트의 "calibration" 디렉터리에 자동으로 저장됩니다.

이로써 캘리브레이션 프로세스가 완료됩니다.

여기까지 트러스 로봇의 핸드-아이 캘리브레이션 작업을 완료하였습니다.

#### 4.4.9. Eye to Eye 시나리오에서의 핸드-아이 캘리브레이션

이 부분에서는 Eye to Eye 시나리오에서 핸드-아이 캘리브레이션을 완료하는 방법에 대해 설명합니다.



Eye to Eye 시나리오에서의 핸드-아이 캘리브레이션 관련 설명은 [두 대의 카메라\(Eye to Eye\) 캘리브레이션](#) 내용을 참조하십시오.

##### 캘리브레이션 사전 준비

핸드-아이 캘리브레이션을 수행하기 전에 다음 준비를 완료해야 합니다.

- Mech-Mind 비전 시스템 구축을 완료합니다.
- 캘리브레이션에 필요한 재료를 준비합니다.
- 캘리브레이션 보드 포인트 클라우드 이미징을 조정합니다.  
로봇 및 인터페이스 구성을 완료합니다.

## ● 비전 시스템 전체 구축 완료

Mech-Mind 비전 시스템 구축을 완료하려면 비전 시스템 구축 부분 내용을 참조하십시오.

핸드-아이 캘리브레이션에는 Mech-Eye Viewer, Mech-Center, Mech-Vision 및 Mech-Viz 소프트웨어가 필요합니다. 위의 소프트웨어가 설치되고 최신 버전으로 업데이트되었는지 확인하십시오.

## 캘리브레이션에 필요한 재료를 준비

Eye to Eye 시나리오에서의 캘리브레이션을 수행할 때 캘리브레이션 보드가 필요합니다. 다음 요구 사항에 따라 캘리브레이션 보드를 준비하십시오.

- 캘리브레이션 보드에서 도트가 잘 보이고 눈에 띄는 굵은 흔적이 없으며 캘리브레이션 보드가 뒤틀리지 않아야 합니다.
- Eye to Eye 시나리오에서 먼저 로봇의 말단 플랜지에 캘리브레이션 보드 커넥터를 설치한 다음 캘리브레이션 보드를 커넥터에 설치합니다. 캘리브레이션 보드가 단단히 설치되어 있고 캘리브레이션 보드가 로봇 말단의 XY 평면과 평행한지 확인하십시오.



로봇 플랜지에 분리 불가능한 그리퍼를 장착한 경우에 캘리브레이션 보드를 그리퍼에 직접 고정하면 됩니다.

또한 캘리브레이션 시작 전 로봇을 캘리브레이션 시작 포즈로 이동합니다. 즉, 두 카메라의 시야가 겹치는 영역에서 최하위 레이어의 중심 위치입니다.

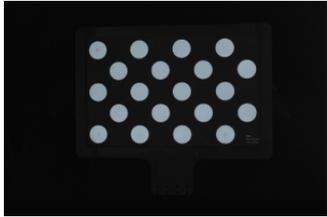
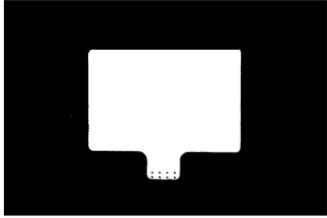
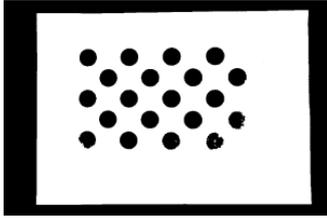
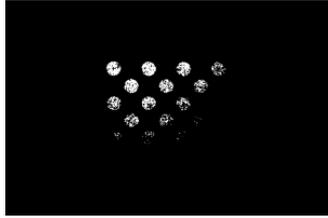
## 캘리브레이션 보드 포인트 클라우드 이미지를 조정하기

1. Mech-Eye Viewer 소프트웨어를 열어서 각 카메라를 차례로 연결하고 카메라 파라미터를 조정합니다.
2. 2D 파라미터를 조정함으로써 2D 맵의 캘리브레이션 보드가 뚜렷하고 노출 과다나 노출 부족 등 현상이 없도록 합니다.
3. 3D 파라미터를 조정함으로써 캘리브레이션 보드의 원의 포인트 클라우드를 완전하게 하고 **포인트 클라우드 후처리에서 포인트 클라우드 평활화 및 노이즈 제거를 일반으로 변경하는 것이 좋으며** 포인트 클라우드 변동 범위를 줄입니다.



현장에 환경광이 복잡한 경우, 환경광이 2D 맵과 포인트 클라우드에 주는 영향을 줄이기 위해 광원을 차단하거나 보충하기 바랍니다.

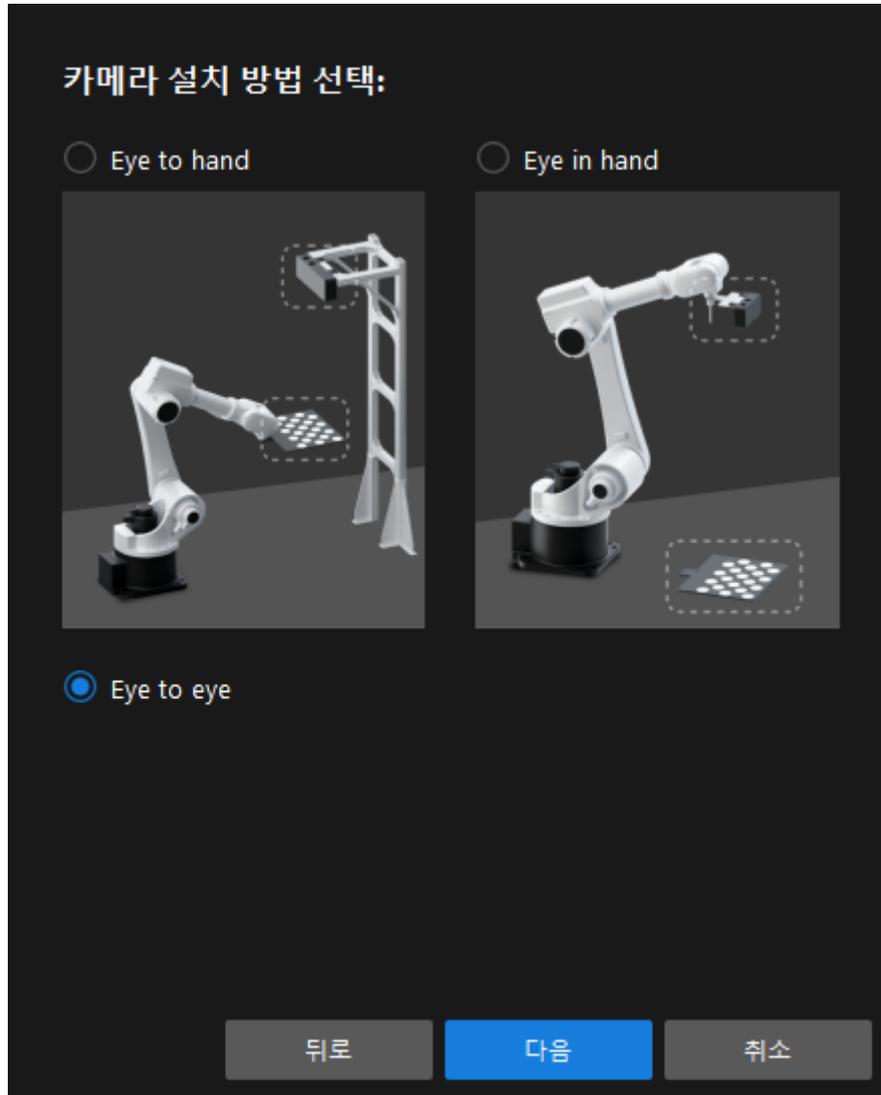
4. 위의 단계를 완료하여 캘리브레이션 보드의 포인트 클라우드 이미지 품질이 요구 사항을 충족하는지 확인하십시오.

	정상	노출 과다	노출 부족
2D 맵			
포인트 클라우드			

## 캘리브레이션 사전 구성

로봇이 표준 인터페이스 또는 마스터 컨트롤 통신 방식을 지원하는 경우 Eye to Hand 시나리오에서의 자동 캘리브레이션 프로세스의 **캘리브레이션 사전 구성** 부분을 참조하여 캘리브레이션 사전 구성을 완료하십시오.

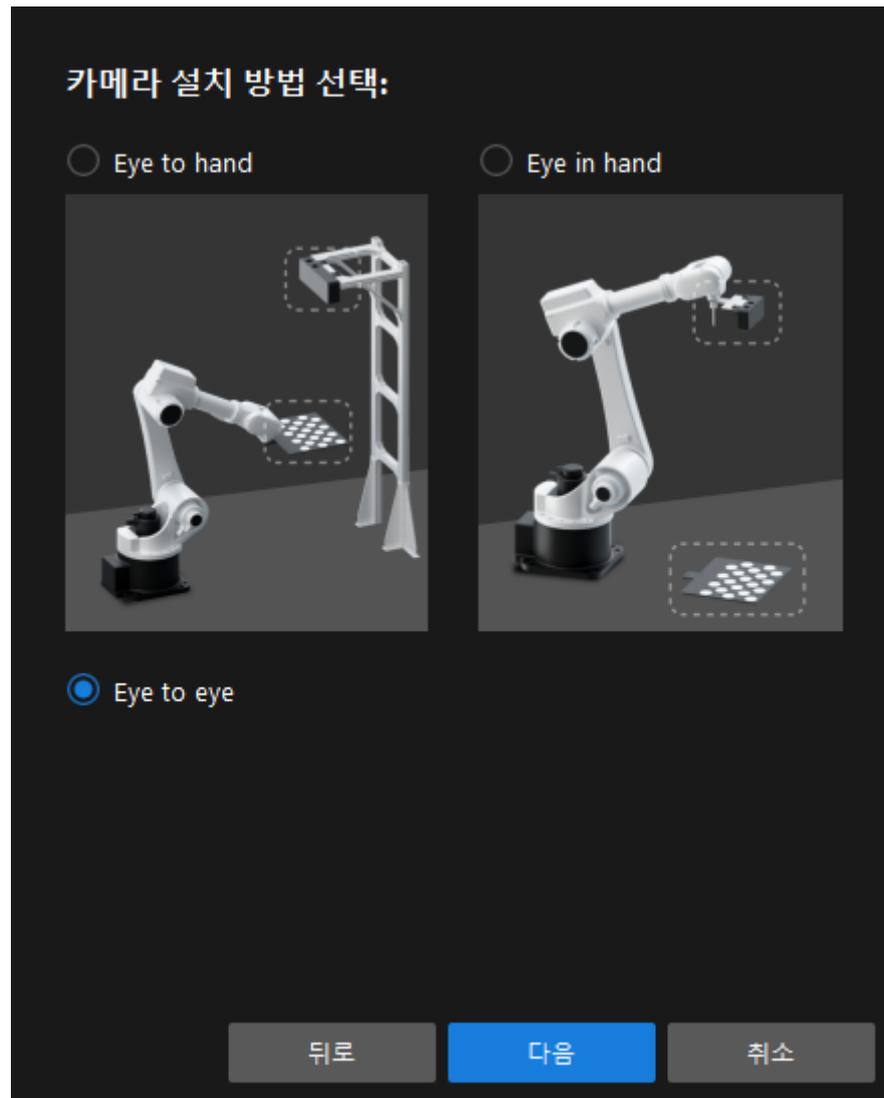
위의 캘리브레이션 프로세스와 비교했을 때 Eye to Eye 시나리오의 캘리브레이션 사전 구성은 **카메라 설치 방법 선택** 창에서 **Eye to eye** 라디오 버튼을 선택해야 합니다.



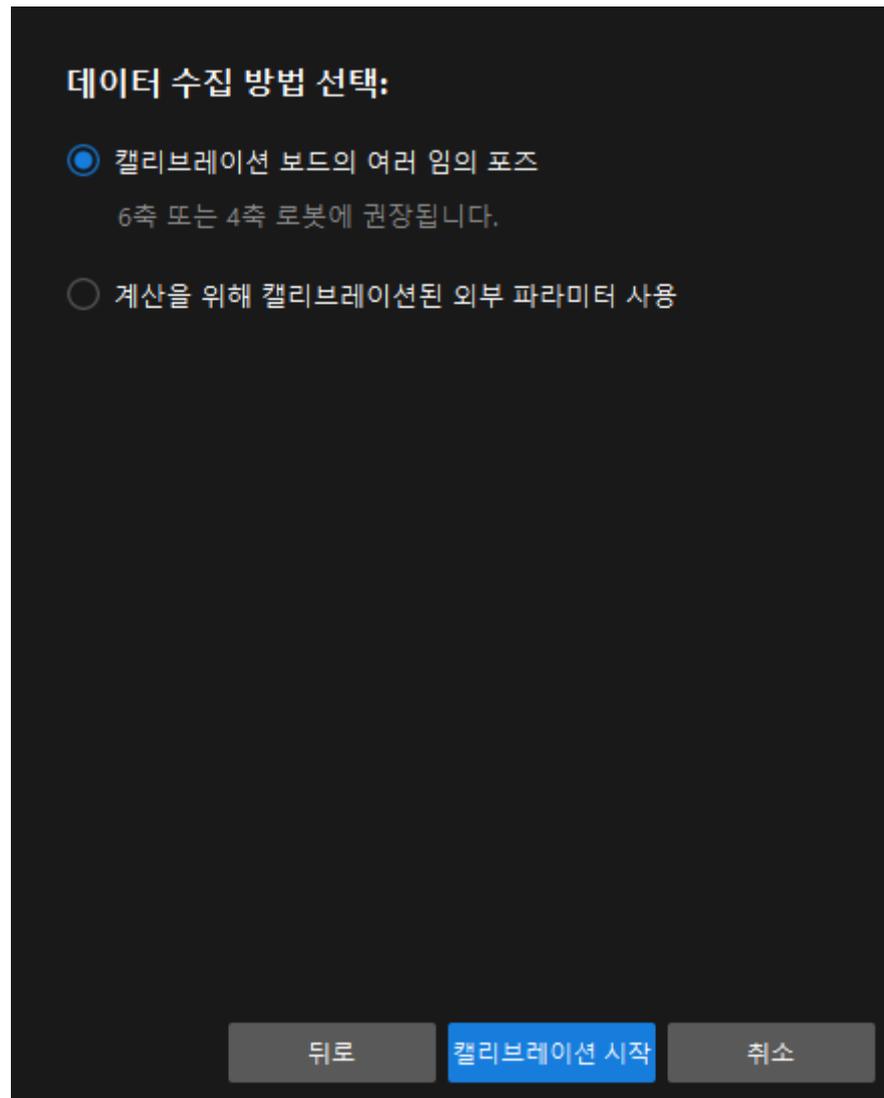
로봇이 표준 인터페이스 또는 마스터 컨트롤 통신 방식을 지원하는 경우 Eye to Hand 시나리오에서의 수동 캘리브레이션(캘리브레이션 보드의 여러 임의의 포즈) 프로세스의 **캘리브레이션 사전 구성** 부분을 참조하여 캘리브레이션 사전 구성을 완료하십시오.

위의 캘리브레이션 프로세스와 비교했을 때 Eye to Eye 시나리오의 캘리브레이션 사전 구성은 다음 단계에서 차이가 있습니다.

1. **카메라 설치 방법 선택** 창에서 **Eye to eye** 라디오 버튼을 선택합니다.



2. 데이터 수집 방법 선택 창에서 캘리브레이션 보드의 여러 임의 포즈 라디오 버튼을 선택합니다.



## 캘리브레이션 프로세스

### 카메라 연결

- 1. 카메라 연결** 단계에서 **감지된 카메라** 리스트에서 연결할 카메라를 찾은 다음 을 클릭하면 카메라가 연결되며 또는 직접 연결할 카메라를 더블클릭합니다.



2. 위의 단계를 반복하여 메인 카메라를 연결합니다. 메인 카메라가 연결되면 카메라 번호 앞에  아이콘이 표시됩니다.



메인 카메라를 전환하려면 카메라를 선택하고 [메인 카메라로 설정] 버튼을 클릭합니다.

3. 카메라를 연결한 후 [연속 캡처] 또는 [한번 캡처] 버튼을 선택할 수 있습니다.



4. 이미지 뷰어에서 카메라가 캡처한 2D 맵과 뎀스 맵이 캘리브레이션 요구 사항을 충족하는지 확인한 다음 하단 표시줄에서 [ 다음 ] 버튼을 클릭합니다.



- 캡처한 이미지가 캘리브레이션 요구 사항을 충족하지 않으면 Mech-Eye Viewer 소프트웨어를 열어 **카메라의 2D 및 3D 노출 파라미터를 조정**하고 다시 캡처해야 합니다.
- 이 단계는 메인 카메라의 2D 맵과 뎀스 맵만 캡처합니다. 서브 카메라의 이미지 캡처 효과를 확인하려면 메인 카메라로 전환한 후 이미지 캡처 효과를 확인한 후 기존 메인 카메라를 다시 메인 카메라로 설정할 수 있습니다.

#### 기타 단계

로봇이 표준 인터페이스 또는 마스터 컨트롤 통신 방식을 지원하는 경우 Eye to Hand 시나리오에서의 자동 캘리브레이션 프로세스의 **캘리브레이션 프로세스** 부분을 참조하여 남은 단계의 구성을 완료하십시오.

로봇이 표준 인터페이스 또는 마스터 컨트롤 통신 방식을 지원하는 경우 Eye to Hand 시나리오에서의 수동 캘리브레이션(캘리브레이션 보드의 여러 임의 포즈) 프로세스의 **캘리브레이션 프로세스** 부분을 참조하여 남은 단계의 구성을 완료하십시오.



- **이미지 및 포즈 가져오기** 단계에서 소프트웨어는 두 카메라의 이미지 및 포즈의 캘리브레이션 데이터 수집을 동시에 완료합니다. **캘리브레이션 이미지 및 포즈 리스트** 패널에서 카메라를 전환하여 다양한 카메라에서 캡처한 캘리브레이션 보드 이미지 및 캘리브레이션 원 정보를 볼 수 있습니다.
- **카메라 파라미터 계산** 단계에서 소프트웨어는 두 카메라의 외부 파라미터 계산 및 서브 카메라가 메인 카메라에서의 포즈 관계 변환을 동시에 완료합니다.

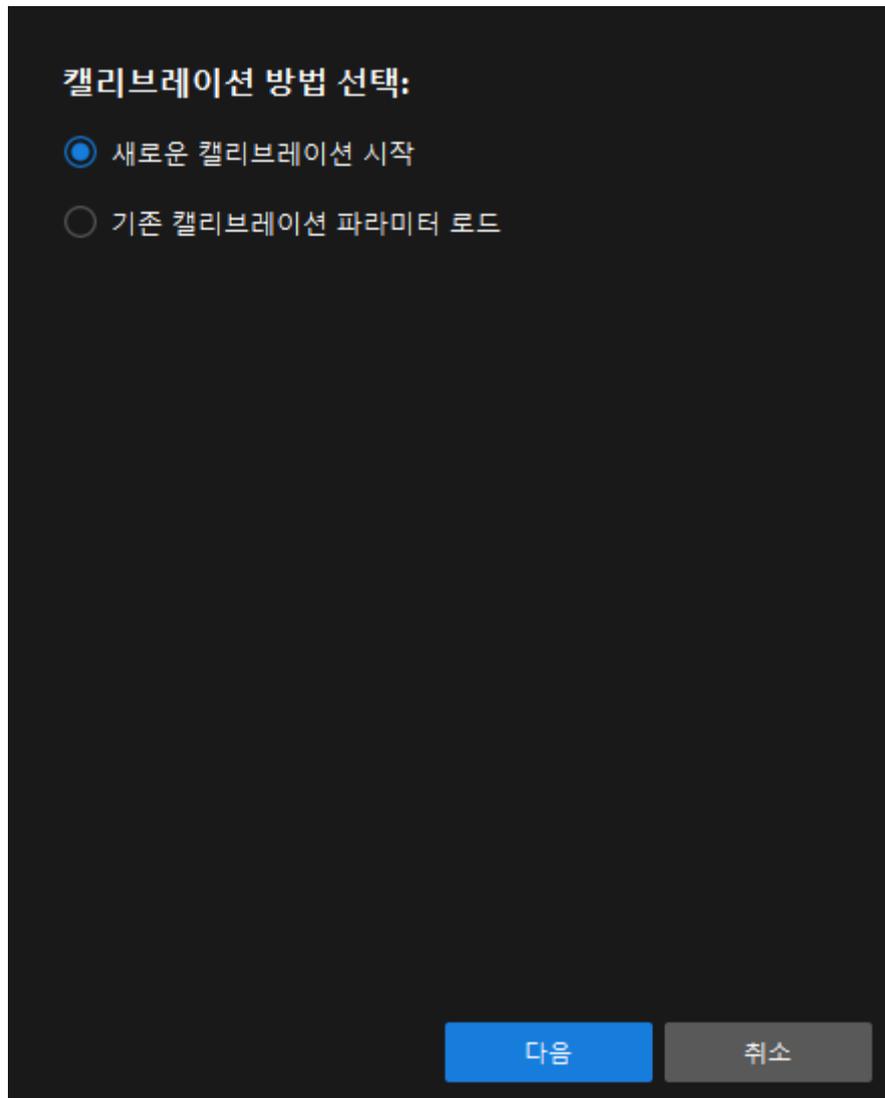
이로써 캘리브레이션 프로세스가 완료됩니다.

### 기존 캘리브레이션 외부 파라미터를 사용하여 두 카메라의 포즈 관계를 직접 계산

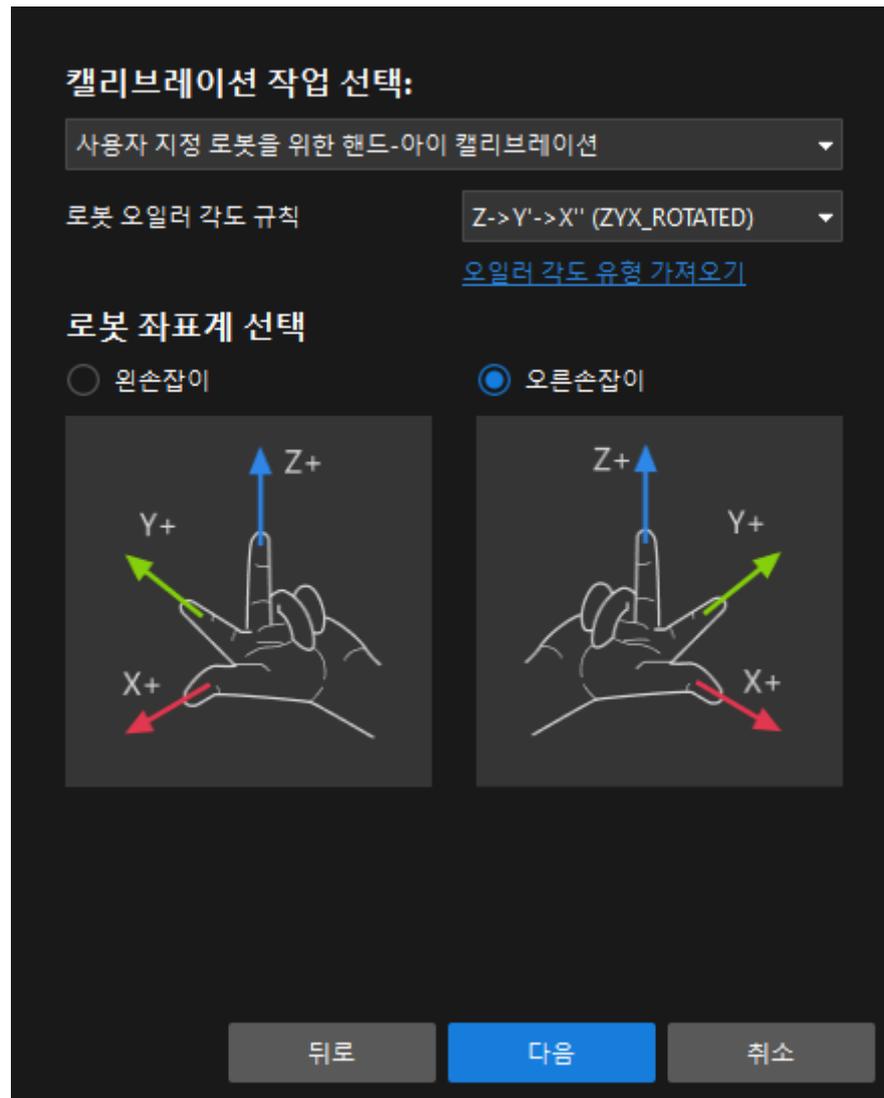
두 카메라의 외부 파라미터 캘리브레이션이 완료되면 Eye to Eye 캘리브레이션 프로세스를 사용하여 두 카메라의 포즈 관계를 직접 계산할 수 있습니다.

기존 캘리브레이션 외부 파라미터를 사용하여 두 카메라의 포즈 관계를 직접 계산하려면 다음 단계를 수행하십시오.

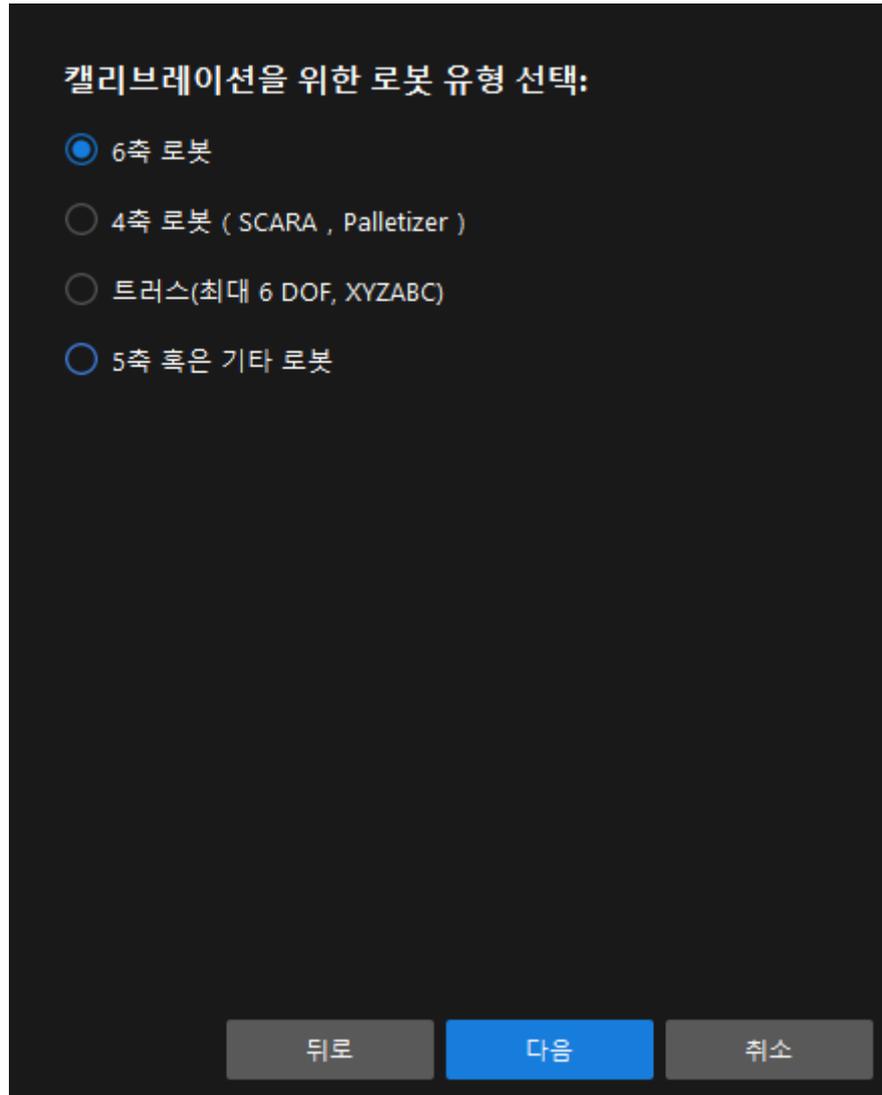
1. Mech-Vision 소프트웨어에서 톨 바의 [ 카메라 캘리브레이션(표준) ] 버튼을 클릭합니다. **캘리브레이션 사전 구성** 창이 팝업됩니다.
2. **캘리브레이션 방법 선택** 창에서 **새로운 캘리브레이션 시작** 라디오 버튼을 선택한 후 [ 다음 ] 버튼을 클릭합니다.



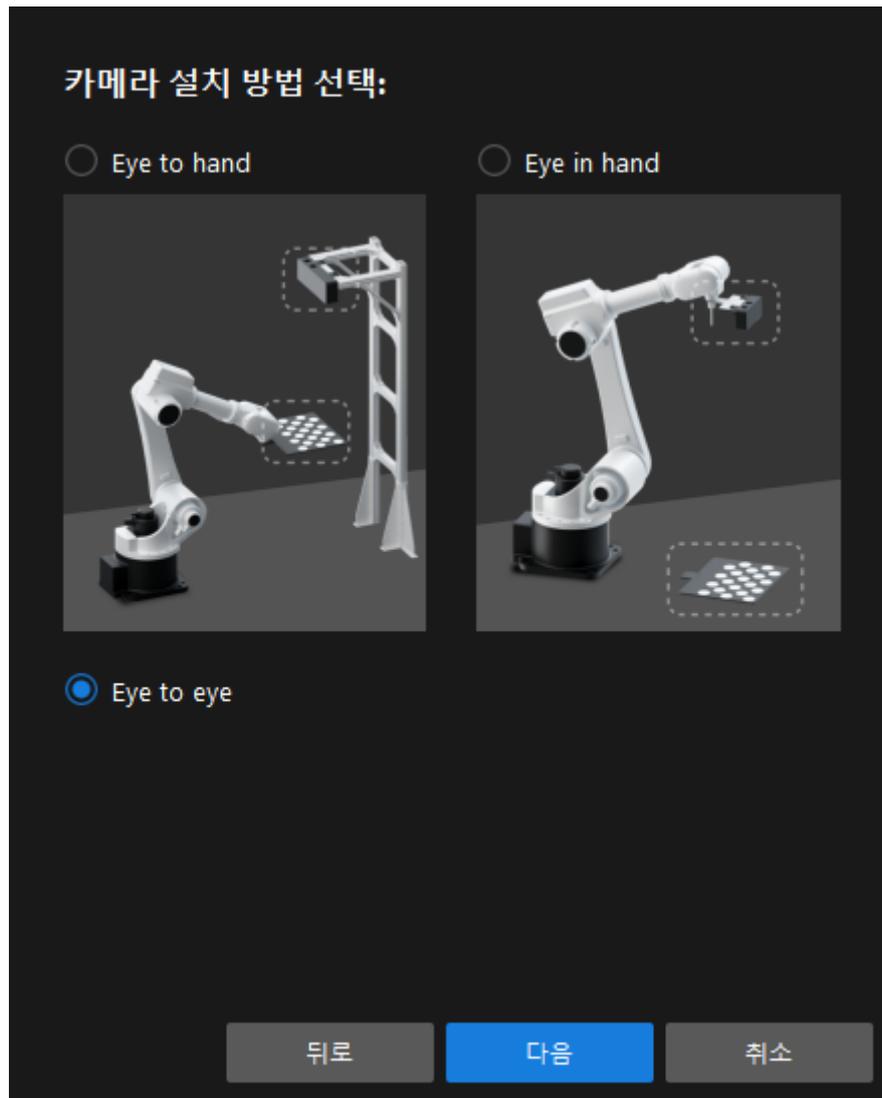
3. **캘리브레이션 작업 선택** 창의 드롭다운 목록 상자에서 **사용자 지정 로봇을 위한 핸드-아이** **캘리브레이션**을 선택하고 **로봇 오일러 각도 유형** 파라미터를 지정하여 로봇 좌표계 유형을 선택한 후 [ 다음 ] 버튼을 클릭합니다.



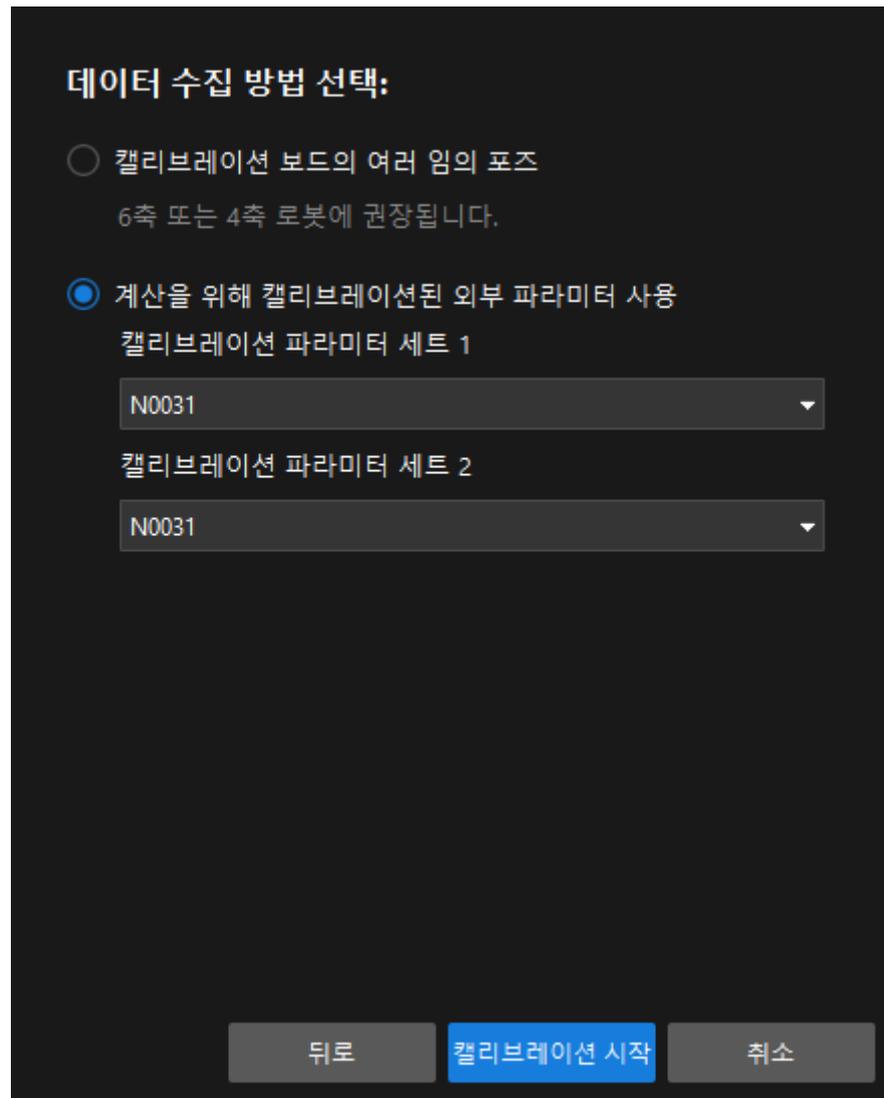
4. 캘리브레이션을 위한 로봇 유형 선택 창에서 로봇 유형에 따라 6축 로봇, 4축 로봇(SCARA, Palletizer) 또는 5축 혹은 기타 로봇 라디오 상자를 선택한 후 [ 다음 ] 버튼을 클릭합니다.



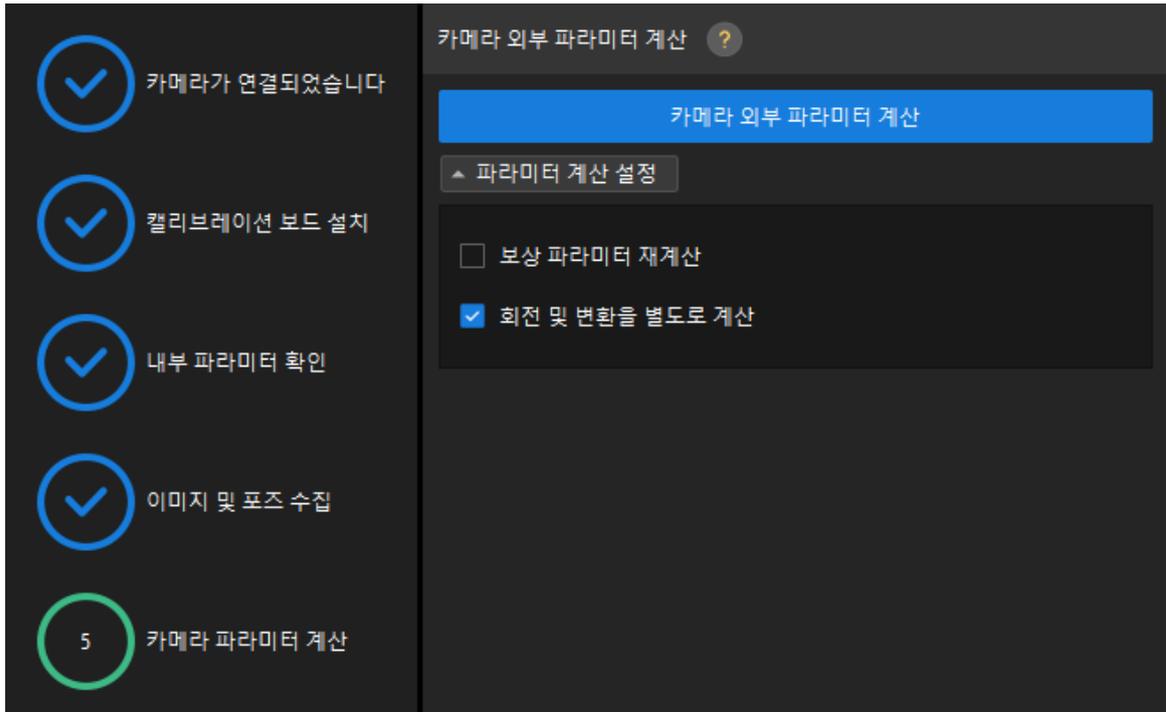
5. 카메라 설치 방법 선택 창에서 **Eye to eye** 라디오 버튼을 선택한 후 [ 다음 ] 버튼을 클릭합니다.



6. 데이터 수집 방법 선택 창에서 기존 캘리브레이션 외부 파라미터를 사용하여 직접 계산 라디오 버튼을 선택하고 두 카메라의 외부 파라미터 파일을 선택한 다음 [캘리브레이션 시작] 버튼을 클릭합니다. 캘리브레이션(Eye to Eye) 창이 나타납니다.



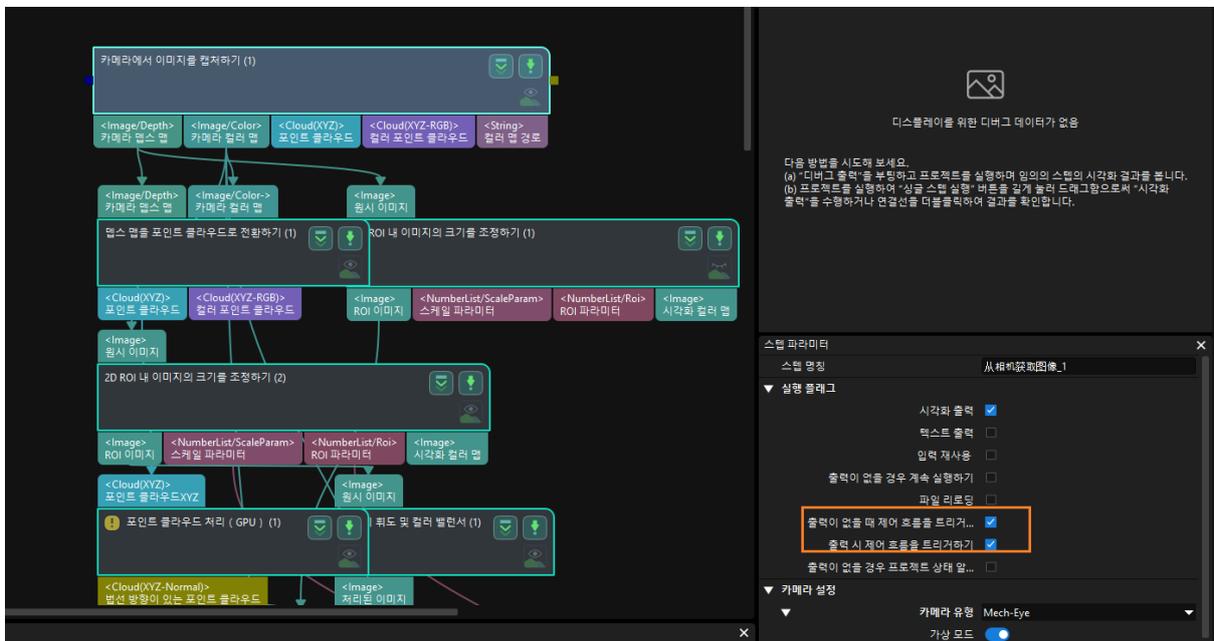
7. 카메라 연결 단계에서 감지된 카메라 리스트에서 연결할 카메라를 찾아  버튼을 클릭하여 카메라를 연결합니다.
8. 위의 단계를 반복하여 두 번째 카메라를 연결한 후 하단 표시줄에서 [ 다음 ] 버튼을 클릭합니다.
9. 카메라 파라미터를 계산하기 단계에서 회전 및 변환을 별도로 계산 파라미터를 선택한 다음 [ 카메라 외부 파라미터 계산 ] 버튼을 클릭하십시오.



카메라 외부 파라미터를 계산한 후 **포인트 클라우드 뷰어** 패널에서 생성된 융합 포인트 클라우드를 볼 수 있습니다.

#### Mech-Vision 프로젝트를 통해 두 카메라 캘리브레이션의 융합 효과 보기

프로젝트 구축은 아래 그림과 같습니다. ”출력이 없을 때 제어 흐름을 촉발하기” 나 ”출력이 있을 때 제어 흐름을 촉발하기”를 체크하세요.



“데이터를 합치기” 스텝을 한 단계씩 실행하여 융합된 포인트 클라우드를 표시합니다. 데이터를 합친 후 출력한 포인트 클라우드는 아래 그림과 같이 융합된 전체 포인트 클라우드입니다. 왼쪽 상단 모서리에서 **View as Whole | 1 | 2**를 클릭하여 포인트 클라우드를 전환할 수 있습니다.



#### 4.4.10. 참조 정보

이 부분에서는 로봇 핸드-아이 캘리브레이션과 관련된 참조 정보를 제공합니다.

[캘리브레이션 관련 개념](#)

[캘리브레이션 원리](#)

[트러스 로봇 캘리브레이션 설명](#)

[캘리브레이션 결과 체크 및 분석](#)

[자주 나타나는 문제와 해결 방법](#)

##### 4.4.10.1. 캘리브레이션 관련 개념

###### 로봇 통신 방식

로봇이 비전 측과 통신하는 방법을 나타냅니다. Mech-Mind 비전 시스템은 표준 인터페이스, Adapter 및 마스터 컨트롤의 세 가지 통신 방법을 지원합니다. 자세한 내용은 [통신 개요](#)를 참조하십시오.

###### 카메라 설치 방식

작업 셀에서의 카메라 설치 방식을 가리킵니다. 일반적인 설치 방식은 ETH(Eye to Hand) 및 EIH(Eye in Hand)입니다. 또한 카메라의 시야를 확장하고 포인트 클라우드의 품질을 향상시키기 위해 프로젝트는 두 대의 카메라를 설치할 수도 있으며 Eye to Eye라고 합니다.

###### 캘리브레이션 방식

캘리브레이션 이미지와 포즈를 캡처하는 과정의 자동화 여부에 따라 캘리브레이션 방식은 자동 캘리브레이션과 수동 캘리브레이션으로 구분됩니다. 수동 캘리브레이션 작업은 비교적 복잡하므로 자동 캘리브레이션 방식을 사용하는 것이 좋습니다.

### 자동 캘리브레이션(권장)

캘리브레이션 시 로봇을 연결하면 Mech-Vision이 자동으로 캘리브레이션 경로를 계획하고 계획된 경로에 따라 이동하도록 로봇을 제어하고 캘리브레이션 이미지 및 포즈 캡처를 완료합니다.

### 수동 캘리브레이션

캘리브레이션 시 로봇을 연결하지 않고 사용자는 수동으로 로봇을 제어하고 본인이 계획한 경로에 따라 이동하거나 캘리브레이션 원을 터치하며 수동으로 로봇의 포즈를 입력하고 캘리브레이션 이미지 캡처를 트리거해야 합니다.

### 캘리브레이션 데이터 수집 방법

캘리브레이션 데이터를 수집하는 방법을 말합니다. Mech-Vision은 두 가지 캘리브레이션 데이터 수집 방법(캘리브레이션 보드의 여러 임의 포즈 및 TCP 끝단점 터치)을 지원합니다.

### 다수의 랜덤 캘리브레이션 보드 포즈(권장)

로봇이 소프트웨어에 의해 자동으로 생성된 웨이포인트 또는 수동으로 계획된 웨이포인트를 순차적으로 지나가도록 하고 각 웨이포인트에서 캘리브레이션 보드의 이미지를 캡처하고 캘리브레이션 원을 인식하고 로봇 플랜지의 포즈를 캡처하고 최종적으로 캘리브레이션 포인트의 열을 형성하여 캘리브레이션 보드, 카메라 및 로봇 간의 관계를 구성합니다. 이 방법은 과정이 간단하고 캘리브레이션 정확도가 높습니다. 이 방법은 6축 또는 4축 로봇에 사용하는 것을 권장합니다.

### TCP 끝단점 터치

3점법으로 캘리브레이션 보드의 포즈를 확인한 후 캘리브레이션 보드와 카메라 및 로봇 간의 관계를 구성합니다. 이 방법은 로봇의 활동 공간이 제한되어 있고 캘리브레이션 보드를 설치할 수 없는 상황에 적합합니다. 이 방법은 5축 또는 기타 로봇에 사용하는 것을 권장합니다.

### 카메라 내부 파라미터

카메라 내부 파라미터는 렌즈 초점 거리, 왜곡 등 카메라 내부의 기본 파라미터입니다. 일반적으로 카메라가 출고될 때 내부 파라미터가 캘리브레이션 완료된 상태로 카메라 내부에 저장되어 있습니다.

### 카메라 외부 파라미터

카메라 외부 파라미터는 로봇과 카메라 간의 포즈 변환 관계(즉, 핸드-아이 관계이므로 로봇 핸드-아이 캘리브레이션을 카메라 외부 파라미터 캘리브레이션이라고도 함)를 나타냅니다. 로봇과 카메라의 상대적인 포즈는 다양한 사용 시나리오에서 고정적이지 않으며 카메라와 로봇 간의 핸드-아이 관계는 작업 현장에서 캘리브레이션해야 합니다.

### 캘리브레이션 포인트

캘리브레이션 과정에서 카메라가 캘리브레이션 보드의 이미지를 캡처할 때 로봇의 포즈를 말합니다. 캘리브레이션 보드의 여러 임의 포즈를 사용하여 캘리브레이션 데이터를 수집하는 경우 캘리브레이션 포인트는 캘리브레이션 경로의 웨이포인트입니다. TCP 끝단점 터치 방법을 사용하여 캘리브레이션 데이터를 수집할 때 캘리브레이션 보드에서 세 개의 비동일선 캘리브레이션 원의 중심을 터치한 후 캘리브레이션 보드의 이미지를 한 번 캡처하면 됩니다.

### 캘리브레이션 원

캘리브레이션 보드의 원형 특징점을 나타냅니다. 캘리브레이션 시 소프트웨어는 캡처한 캘리브레이션 보드 이미지(2D 맵 및 덤스 맵)에서 캘리브레이션 원 중심의 픽셀 좌표와 원 중심이 카메라 좌표계에서의 좌표를 계산합니다. 그다음 수집된 캘리브레이션 원 데이터를 기반으로 카메라 외부 파라미터를 계산합니다.

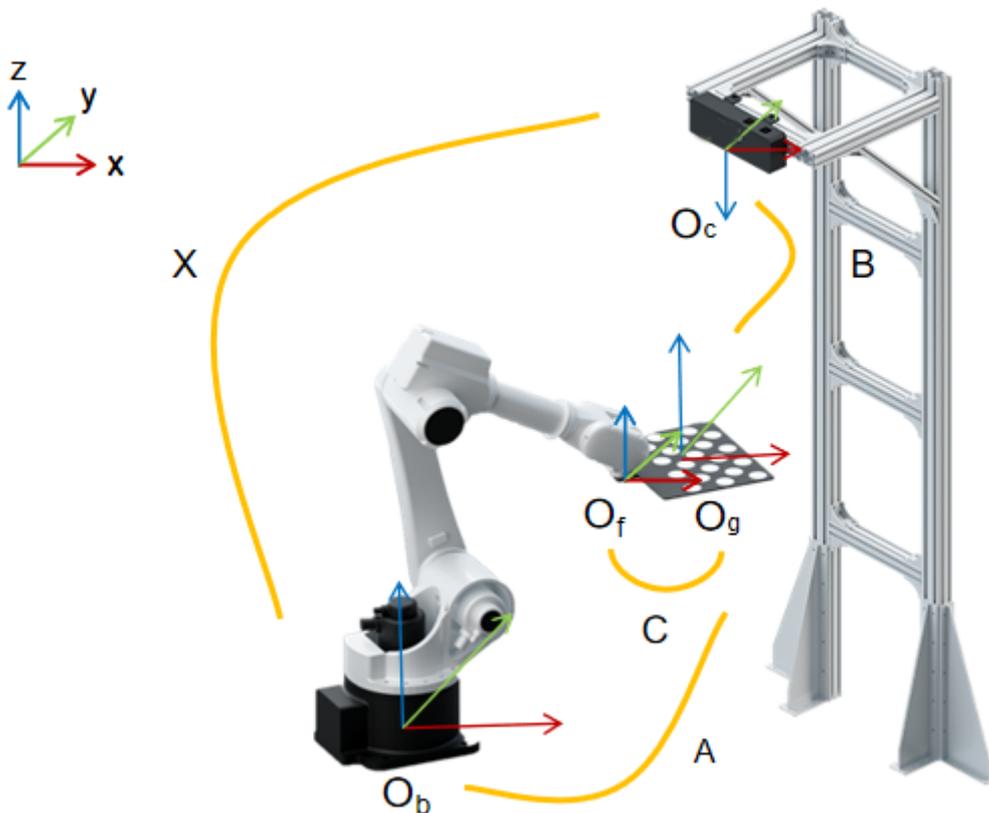
### 4.4.10.2. 캘리브레이션 원리

이 부분에서는 각 시나리오에서의 캘리브레이션의 작업 원리를 소개합니다.

#### ETH 시나리오에서의 캘리브레이션(캘리브레이션 보드의 여러 임의 포즈)



알려진 치수의 캘리브레이션 보드가 로봇 플랜지에 설치되고 로봇 베이스 좌표계(Base)에서 A로 표시된 캘리브레이션 보드의 각 원의 좌표를 얻을 수 있습니다. 캘리브레이션 보드의 이미지를 캡처하면 B로 표시된 각 캘리브레이션 원에 대한 카메라 광학 중심의 좌표를 얻을 수 있습니다. 카메라 광학 중심의 베이스 좌표계(Base)와 X로 표시된 로봇 베이스 간의 변환 관계는 미지수입니다. 따라서 A, B, X는 폐쇄 루프를 형성하고, 루프에서 생성된 방정식을 통해 X의 값을 계산할 수 있습니다. 캘리브레이션 보드와 플랜지 끝의 위치 관계 C를 모르는 경우 캘리브레이션 프로세스에서 캘리브레이션 보드 일련의 상대적인 이동을 통해 수치법에 따라 캘리브레이션 보드와 플랜지 끝의 위치 관계를 계산하여 획득할 수 있습니다. 이에 따라 A를 계산할 수 있습니다. 로봇을 움직이고 카메라를 기준으로 캘리브레이션 보드의 포즈를 변경함으로써 여러 방정식 세트를 얻을 수 있으며 이러한 방정식의 값을 피팅하고 최적화하여 최종적으로 최적의 X 값을 얻을 수 있습니다. 포즈 관계는 아래 그림과 같습니다.



카메라 좌표계와 로봇 좌표계의 대응 관계를 계산하는 공식은 아래 그림과 같습니다.

$$B_1 \bullet X \bullet C_1 = A$$

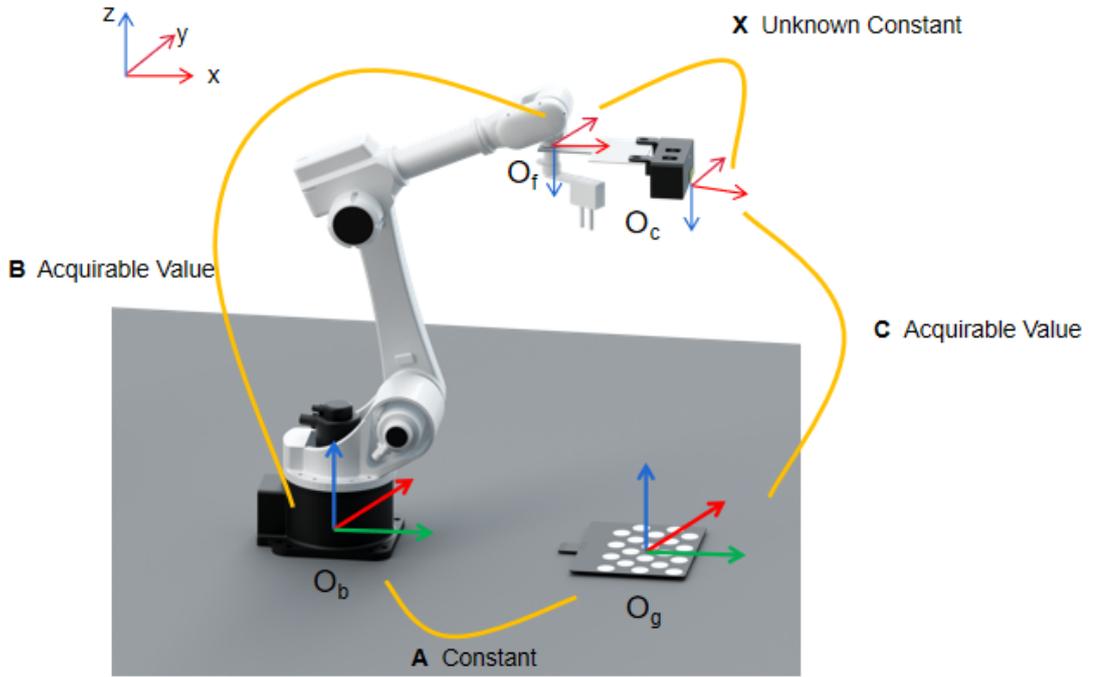
$$B_2 \bullet X \bullet C_2 = A$$

$$B_1 \bullet X \bullet C_1 = B_2 \bullet X \bullet C_2$$

EIH 시나리오에서의 캘리브레이션(캘리브레이션 보드의 여러 임의 포즈)



카메라는 고정 프레임을 통해 로봇의 말단에 고정되는데 이때 로봇 말단의 플랜지 중심과 카메라의 광학 중심 사이의 포즈는 상대적으로 고정되어 있습니다. 즉 아래의 그림 중의 알 수 없는 변수 X입니다. 로봇 베이스 좌표계(Base)에 대한 로봇 말단 플랜지 중심의 자세는 알려진 양 B입니다. 캘리브레이션 보드의 사진을 찍음으로써 카메라는 카메라의 광학 중심과 캘리브레이션 보드의 각 점 사이의 포즈 관계를 얻고 알려진 양 C를 얻을 수 있습니다. 캘리브레이션 보드는 카메라 시야의 도달 가능한 영역에 평평하게 배치되며 로봇의 기본 좌표에 대한 포즈 관계는 고정 값 A입니다. 이러한 방식으로 변수 A, B, C 및 X는 폐쇄 루프 관계를 형성합니다. 다음 방정식에서 A는 고정 값이므로 처음 두 방정식을 결합하고 X만 새로 얻은 방정식에서 풀어야 할 미지의 양입니다. 로봇 말단 포즈를 변환하고 다양한 각도에서 사진을 캡처하여 A, B, C 값의 여러 세트를 얻고 이 값을 사용하여 최적의 X 값을 얻기 위한 피팅 계산을 수행합니다.



카메라 좌표계와 로봇 플랜지 좌표계의 대응 관계를 계산하는 공식은 아래 그림과 같습니다.

$$B_1 \cdot X \cdot C_1 = A$$

$$B_2 \cdot X \cdot C_2 = A$$

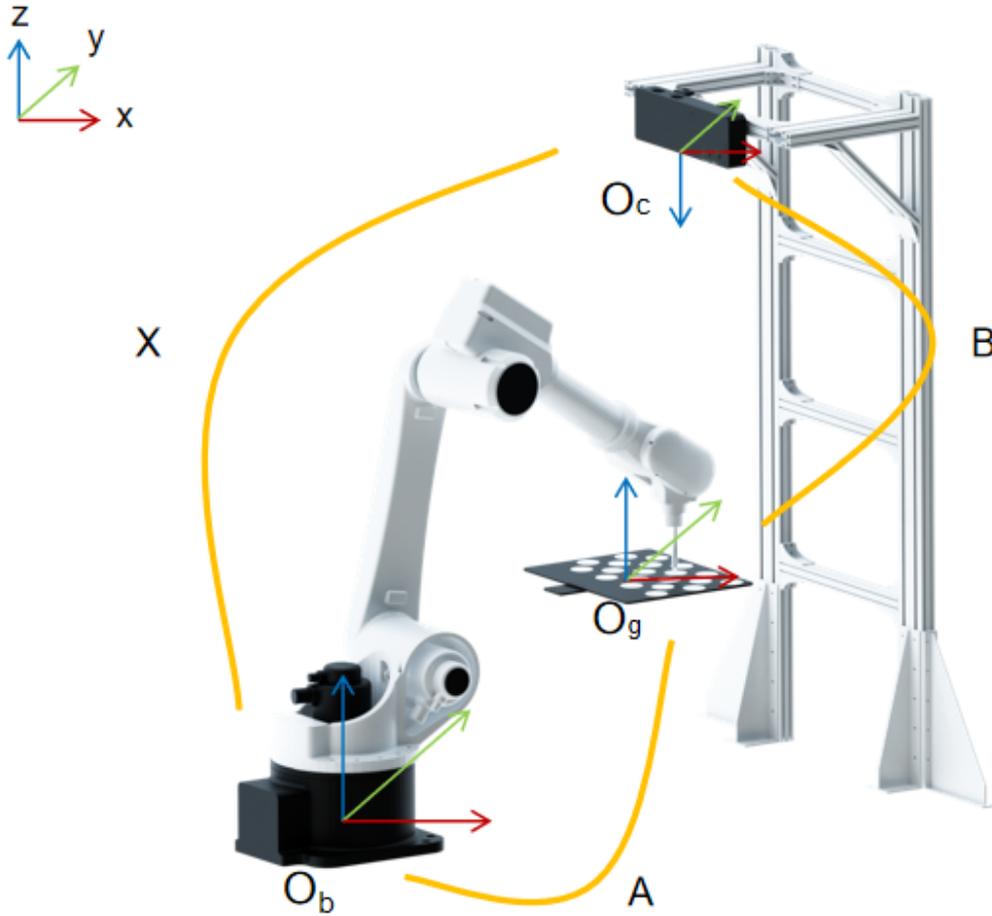
$$B_1 \cdot X \cdot C_1 = B_2 \cdot X \cdot C_2$$

ETH 시나리오에서의 캘리브레이션(TCP 끝단점 터치)



TCP 끝단점 터치 방법을 사용해 캘리브레이션을 하는 경우 캘리브레이션 보드를 물체체 평면에 놓고 알려진 크기의 끝단점을 로봇 말단에 추가하여 캘리브레이션 보드 원의 교차 중심을 터치합니다. 다음

그림과 같이 A와 B의 값을 알고 X의 값을 구할 수 있습니다. 캘리브레이션 보드가 플랜지에 고정되어 있지 않은 경우 TCP 좌표가 알려진 끝단점을 사용하여 캘리브레이션 보드에서 캘리브레이션 원의 교차 중심점을 터치하여 A 값을 계산할 수 있습니다.



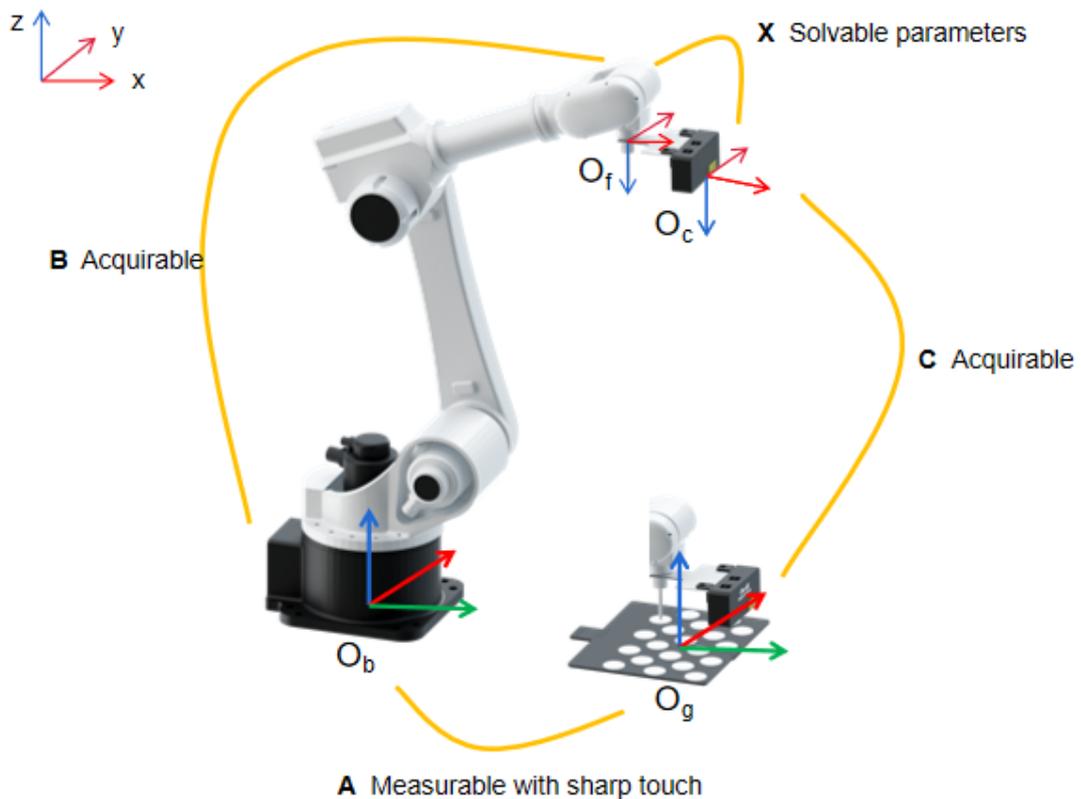
카메라 좌표계와 로봇 좌표계의 대응 관계를 계산하는 공식은 아래 그림과 같습니다.

$$\begin{aligned}
 B_1 \bullet X \bullet C_1 &= A \\
 B_2 \bullet X \bullet C_2 &= A \\
 B_1 \bullet X \bullet C_1 &= B_2 \bullet X \bullet C_2
 \end{aligned}$$

EIH 시나리오에서의 캘리브레이션(TCP 끝단점 터치)



TCP 끝단점 터치 방법을 사용하여 캘리브레이션할 때 캘리브레이션 보드를 작업면에 놓고 로봇 말단에 알려진 크기의 TCP 끝단점을 설치하고 캘리브레이션 보드의 원점을 터치합니다. 원리는 아래의 그림과 같습니다. 여기서 A, B 및 C를 알고 있으면 X 값도 얻을 수 있습니다.



카메라 좌표계와 로봇 플랜지 좌표계의 대응 관계를 계산하는 공식은 아래 그림과 같습니다.

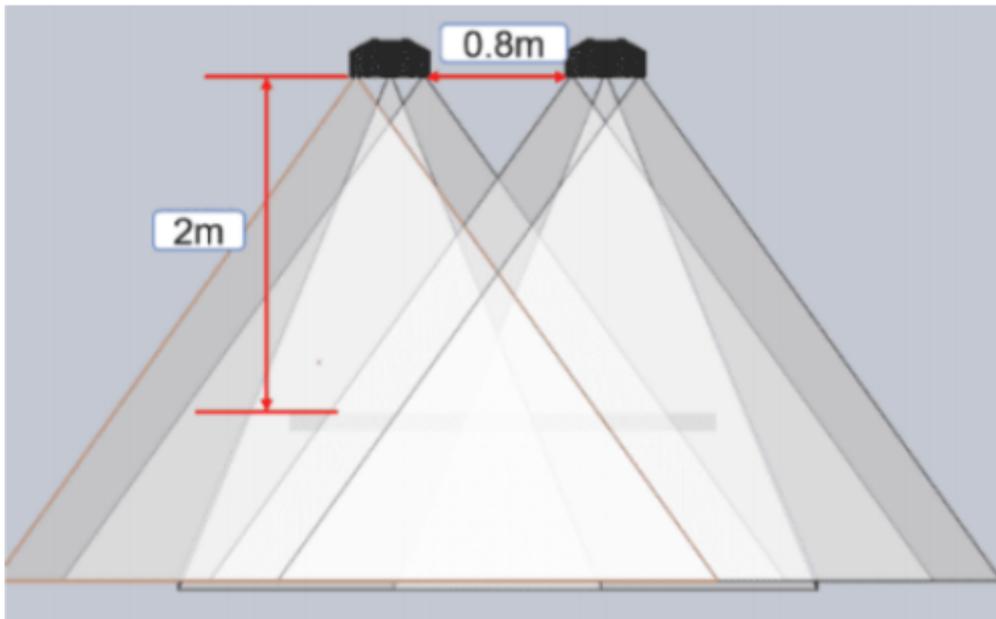
$$B_1 \bullet X \bullet C_1 = A$$

$$B_2 \bullet X \bullet C_2 = A$$

$$B_1 \bullet X \bullet C_1 = B_2 \bullet X \bullet C_2$$

#### 두 대의 카메라(Eye to Eye) 캘리브레이션

두 대의 카메라를 사용하면 다음 그림과 같이 카메라 시야를 확장하고 두 카메라의 겹치는 부분의 포인트 클라우드 품질을 향상시킬 수 있습니다.



Eye to Eye 캘리브레이션 시나리오에서 두 대의 카메라(하나는 메인 카메라이고 다른 하나는 서브 카메라임)는 스탠드에 고정적으로 장착됩니다. Eye to Eye 캘리브레이션은 두 카메라의 외부 파라미터를 캘리브레이션할 뿐만 아니라 두 카메라 간의 포즈 관계도 캘리브레이션해야 합니다. Eye to Eye 캘리브레이션 시나리오의 경우 Mech-Vision은 표준 Eye to Eye 캘리브레이션 절차를 제공합니다.



사용되는 두 대의 카메라는 동일한 해상도를 가져야 하며 카메라 시야(2D 및 3D)의 겹친 구역이 전체 작업 구역을 커버해야 합니다.

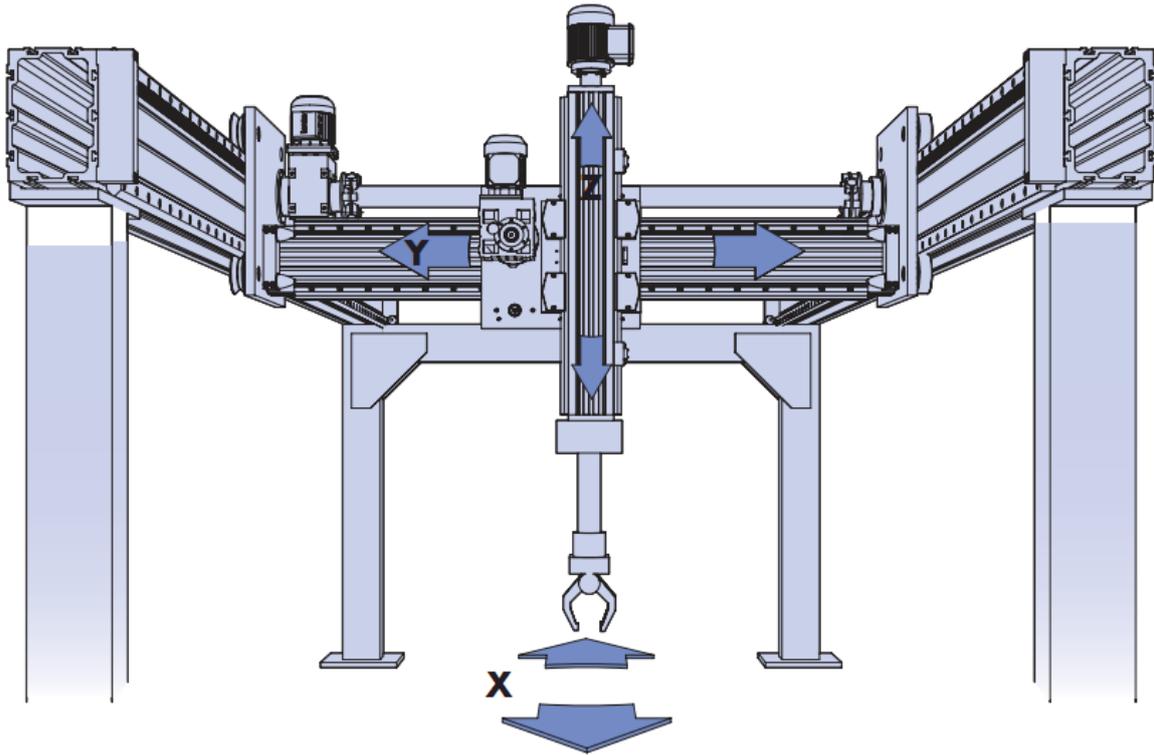
Eye to Eye 캘리브레이션은 캘리브레이션 데이터 수집을 위한 캘리브레이션 보드의 여러 임의 포즈 방법만 지원하며 자동 및 수동 캘리브레이션 모드를 모두 지원합니다.

#### 4.4.10.3. 트러스 로봇 캘리브레이션 설명

##### 관련 개념

##### 트러스 로봇

트러스 로봇은 자동 제어를 실현할 수 있는 자동화 장치이며 공간의 XYZ 데카르트 좌표계를 기반으로 재프로그래밍이 가능하고 여러 자유도를 가지며 다양한 작업에 적합합니다.



### 트러스 로봇의 자유도

트러스 로봇은 다자유도 운동을 지원하며 각 자유도 사이의 공간 각도는 직각입니다.

트러스 로봇은 일반적으로 2~6개 자유도를 지원합니다. 예를 들면 다음과 같습니다.

- XYZ 축은 선형으로 이동할 수 있습니다(3자유도).
- XYZ 축은 선형으로 이동할 수 있고 Z 축(Rz 또는 C 축이라고 함)은 회전(4개 자유도)할 수 있습니다.
- XZ 축은 선형으로 이동할 수 있습니다(2개 자유도).
- XZ 축은 선형으로 이동할 수 있고 Z 축은 회전할 수 있습니다(3개 자유도).

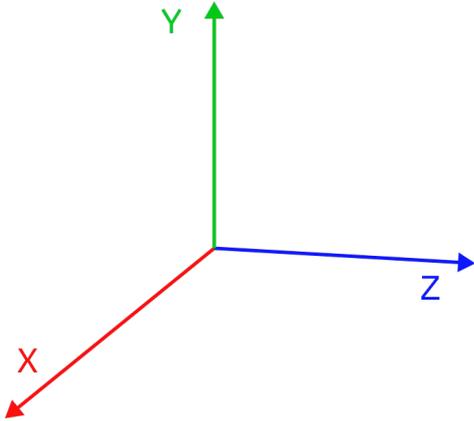
카메라는 일반적으로 트러스의 특정 축(예: Z 또는 C 축)에 장착되며 카메라의 사진 캡처 위치는 축의 움직임에 영향을 받습니다. 따라서 캘리브레이션 사전 구성에서 어느 축의 이동이 카메라의 위치에 영향을 주는지를 지정해야 합니다.

### 왼손/오른손 좌표계

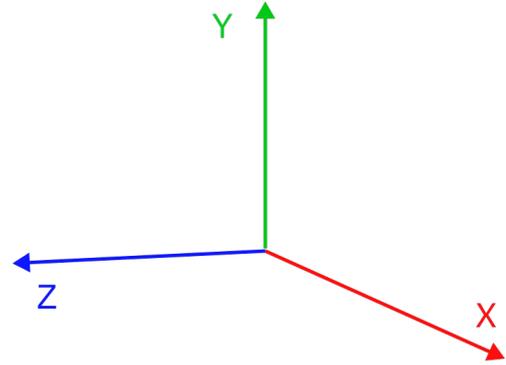
Mech-Mind 비전 시스템은 오른손 좌표계에서 로봇의 플랜지 포즈를 사용하여 외부 파라미터를 계산합니다. 따라서 트러스 로봇을 사용할 때는 트러스 좌표계가 오른손 좌표계인지 여부를 판단해야 합니다.

왼손 좌표계와 오른손 좌표계의 판단 방법은 다음과 같습니다.

- 왼쪽 엄지손가락이 X축의 정방향을 가리키고 검지가 Y축의 정방향을 가리키도록 합니다. 중지가 Z축의 정방향을 가리킬 수 있는 경우 이 좌표계를 왼손 좌표계라고 합니다(아래 왼쪽 그림 참조).
- 오른쪽 엄지손가락이 X축의 정방향을 가리키고 검지가 Y축의 정방향을 가리키도록 합니다. 중지가 Z축의 정방향을 가리킬 수 있는 경우 이 좌표계를 오른손 좌표계라고 합니다(아래 오른쪽 그림 참조).



왼손 좌표계



오른손 좌표계

트러스 로봇이 왼손 좌표계인 경우 후속 캘리브레이션 및 비전 처리를 용이하게 하기 위해 오른손 좌표계로 변경하는 것이 좋습니다. 일반적으로 트러스 로봇의 XYZ 축에 대한 엔코더 동작의 정방향의 반대를 취해야 합니다. 보통 트러스 베이스에 설치된 축의 반대를 취한 다음 Mech-Vision에서 오른손 좌표계를 선택하여 캘리브레이션하면 됩니다.

트러스 로봇에서 이 변경을 수행할 수 있는 경우 Mech-Vision에서 왼손 좌표계만 선택하면 됩니다. 그러면 소프트웨어가 입력 및 출력한 포즈 데이터를 자동으로 변환합니다.

#### 캘리브레이션 원리

트러스 로봇의 경우 카메라는 일반적으로 로봇의 특정 축(예를 들면 Z축)에 장착됩니다. 즉, 카메라의 설치 방식은 Eye in Hand입니다. 따라서 트러스 로봇의 캘리브레이션은 카메라 좌표계와 로봇 TCP 좌표계 사이의 상대적 관계를 결정하는 것을 목표로 합니다.

트러스 로봇의 제한된 자유도와 활동 공간을 고려하여 Mech-Vision 소프트웨어는 TCP 끝단점 터치 방식을 사용하여 캘리브레이션 데이터를 수집하고 캘리브레이션 보드, 카메라 및 로봇 간의 관계를 구성합니다. 또한 트러스 로봇의 캘리브레이션 프로세스는 여러 캘리브레이션 보드의 포즈 추가를 지원합니다. 자유도가 제한되어 트러스 로봇이 캘리브레이션 보드의 세 특징점을 터치할 수 없는 경우 여러 캘리브레이션 보드를 사용하고 트러스 로봇이 여러 캘리브레이션 보드의 세 특징점(세 개의 비동일선)을 터치하도록 제어할 수 있습니다.

트러스 로봇의 캘리브레이션 원리는 [EIH 시나리오에서의 캘리브레이션\(TCP 끝단점 터치\)](#) 부분을 참조하십시오.

#### 외부 파라미터 파일의 사용

트러스 로봇의 캘리브레이션 프로세스는 캘리브레이션 지점에 대한 외부 파라미터 파일을 생성하고 Mech-Vision은 트러스 로봇의 포즈에 따라 동적 외부 파라미터를 실시간으로 계산합니다. Mech-Vision 프로젝트에서 동적 외부 파라미터를 계산하려면 [포인트 클라우드 변환\(트러스\)](#) 및 [포즈 변환\(트러스\)](#) 스텝을 사용하여 카메라 좌표계의 포인트 클라우드/포즈를 트러스 좌표계로 변환해야 합니다.

프로젝트가 실행 중일 때 트러스 로봇이 Mech-Mind 비전 시스템과 정상적으로 통신할 수 있는지 확인하십시오. 또한 로봇 프로그램은 Mech-Vision 프로젝트 실행을 트리거할 때 로봇 플랜지 포즈를 제공해야 "카메라에서 이미지를 캡처하기" 스텝에서 이미지를 캡처할 때 로봇 플랜지의 포즈를 성공적으로 얻을 수 있습니다.



트러스 로봇이 왼손 좌표계를 사용하는 경우 Mech-Center 소프트웨어는 로봇 프로그램에서 보낸 로봇 플랜지 포즈를 오른손 좌표계의 포즈로 자동 변환한 다음 비전 처리를 위해 Mech-Vision으로 보냅니다. 비전 결과가 반환되면 Mech-Center 소프트웨어는 먼저 Mech-Vision이 출력한 트러스 좌표계의 포즈를 왼손 좌표계의 포즈로 자동 변환한 다음 로봇 프로그램으로 반환합니다.

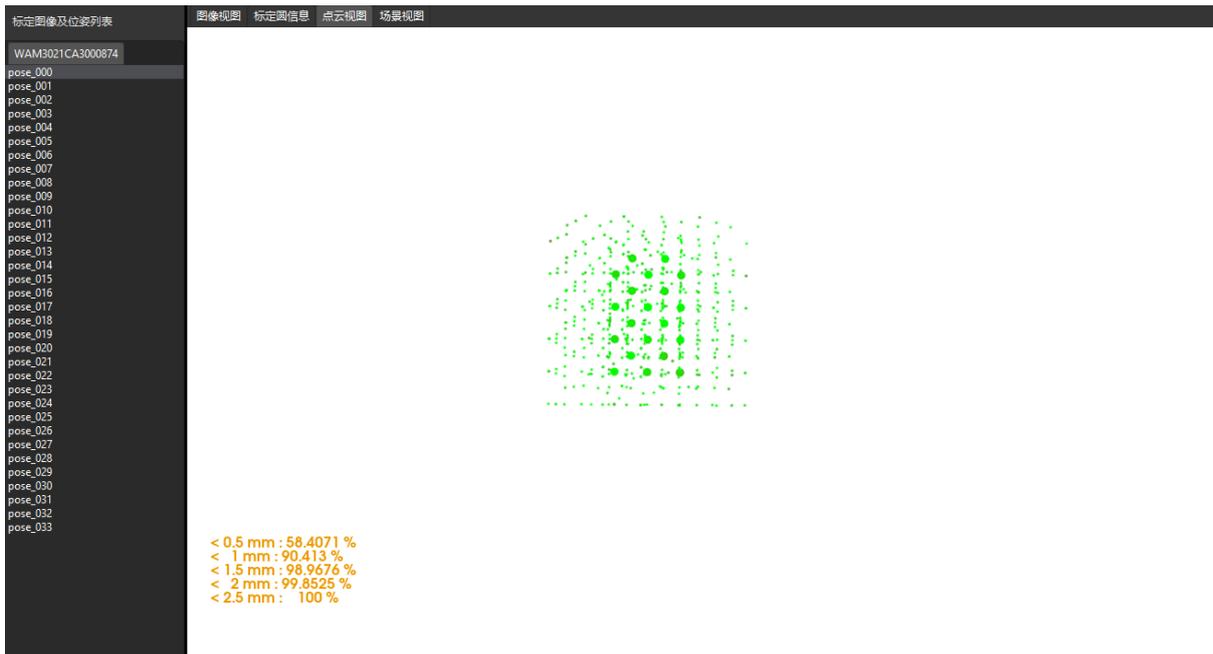
#### 4.4.10.4. 캘리브레이션 결과 체크 및 분석

캘리브레이션 완료된 외부 파라미터의 경우 캘리브레이션 정확도가 요구 사항을 충족하는지 확인해야 합니다. 오차가 정상 범위를 벗어나는 경우 다시 캘리브레이션하기 전에 오차의 원인이 되는 문제를 진단하고 해결하여 적합한 외부 파라미터를 얻어야 합니다.

##### 캘리브레이션 결과 검증

##### 포인트 클라우드 뷰어에서 오류 포인트 클라우드 확인

카메라 파라미터 계산 단계 화면에서 캘리브레이션 결과 계산 완료 후 **포인트 클라우드 뷰어** 패널은 다음 그림과 같이 오차 포인트 클라우드를 표시합니다. 오차 포인트 클라우드는 각 캘리브레이션 포즈의 캘리브레이션 보드 원의 참값과 계산된 값 사이의 편차를 표시하는 데 사용됩니다.

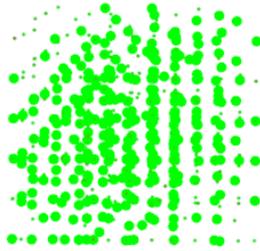


오차 포인트 클라우드의 컬러는 해당 포인트의 오차 등급 (0.5mm당 오차 레벨 하나로 보기)을 표시하며, 녹색에서 빨간색으로 색깔이 어두울수록 해당 포인트의 오차가 크다는 것을 의미합니다.

각 오차 등급의 포인트를 보려면 **포인트 클라우드 뷰어** 패널의 영문 입력 모드에서 숫자 키 0~9를 누르십시오. 소프트웨어는 포인트 클라우드에서 해당 오차 등급의 포인트를 강조 표시합니다(0은 0.5mm 미만의 오차가 있는 지점에 해당하고 1은 0.5mm에서 1mm 사이의 오차가 있는 지점에 해당하는 등).

다음 그림은 오차가 0.5mm 미만인 데이터 포인트를 표시합니다(숫자 키 0 누름).

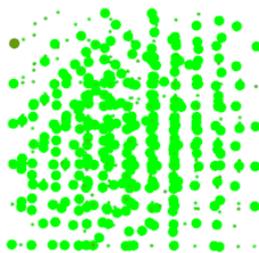
이미지 뷰어   캘리브레이션 원 정보   포인트 클라우드 뷰어   시나리오 뷰어



< 0.5 mm : 58.4071 %  
 < 1 mm : 90.413 %  
 < 1.5 mm : 98.9676 %  
 < 2 mm : 99.8525 %  
 < 2.5 mm : 100 %

다음 그림은 오차가 2.0mm에서 2.5mm 사이인 데이터 포인트를 표시합니다(숫자 키 4 누름).

이미지 뷰어   캘리브레이션 원 정보   포인트 클라우드 뷰어   시나리오 뷰어



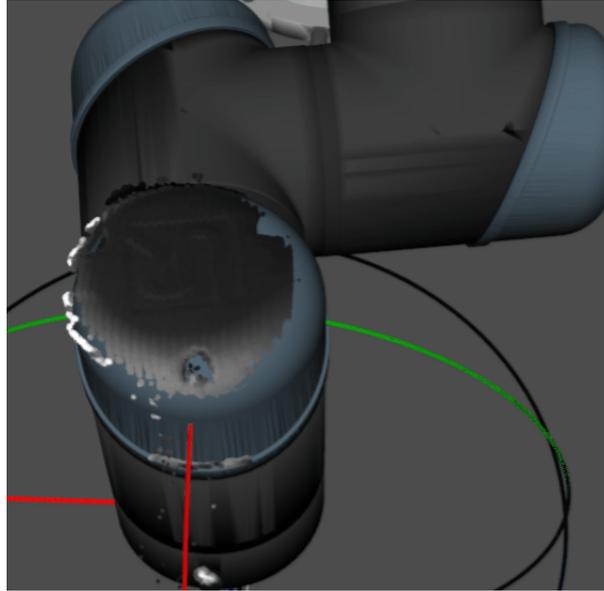
< 0.5 mm : 58.4071 %  
 < 1 mm : 90.413 %  
 < 1.5 mm : 98.9676 %  
 < 2 mm : 99.8525 %  
 < 2.5 mm : 100 %

### 시나리오 뷰어에서 로봇의 포인트 클라우드와 로봇 모델의 매칭 정도 확인(ETH 시나리오에 적용)

tag::calib\_result\_verification\_eth[] 카메라 외부 파라미터를 계산한 후 시나리오 뷰어에서 로봇 포인트 클라우드와 로봇 모델의 매칭 정도를 보고 캘리브레이션 결과를 대략적으로 판단할 수 있습니다. 구체적인 작업은 다음과 같습니다.

1. 로봇 팔을 카메라의 시야 내로 이동합니다.
2. **카메라 파라미터 계산** 단계에서 [ **카메라 외부 파라미터 계산** ] 버튼을 클릭합니다. 이 작업은 카메라가 사진을 캡처하도록 트리거합니다.
3. **시나리오 뷰어**를 클릭하여 로봇 포인트 클라우드와 로봇 모델 간의 매칭 정도를 확인합니다.

로봇의 포인트 클라우드가 로봇 모델과 대략적으로 매칭하면 캘리브레이션 결과를 사용할 수 있음을 의미합니다.



1. 로봇 모델은 실제 로봇의 모양과 완전히 매칭하지 않을 수 있으며 외부 파라미터를 미세 조정하기 위한 기준으로 사용할 수 없습니다.
2. **시나리오 뷰어** 외에도 Mech-Viz 소프트웨어에서 로봇의 포인트 클라우드와 로봇 모델 간의 매칭 정도를 확인할 수 있습니다.

end::calib\_result\_verification\_eth[]

### 시나리오 뷰어에서 고정점을 기준으로 캘리브레이션 보드 포인트 클라우드의 움직임 보기(EIH 시나리오에 적용)

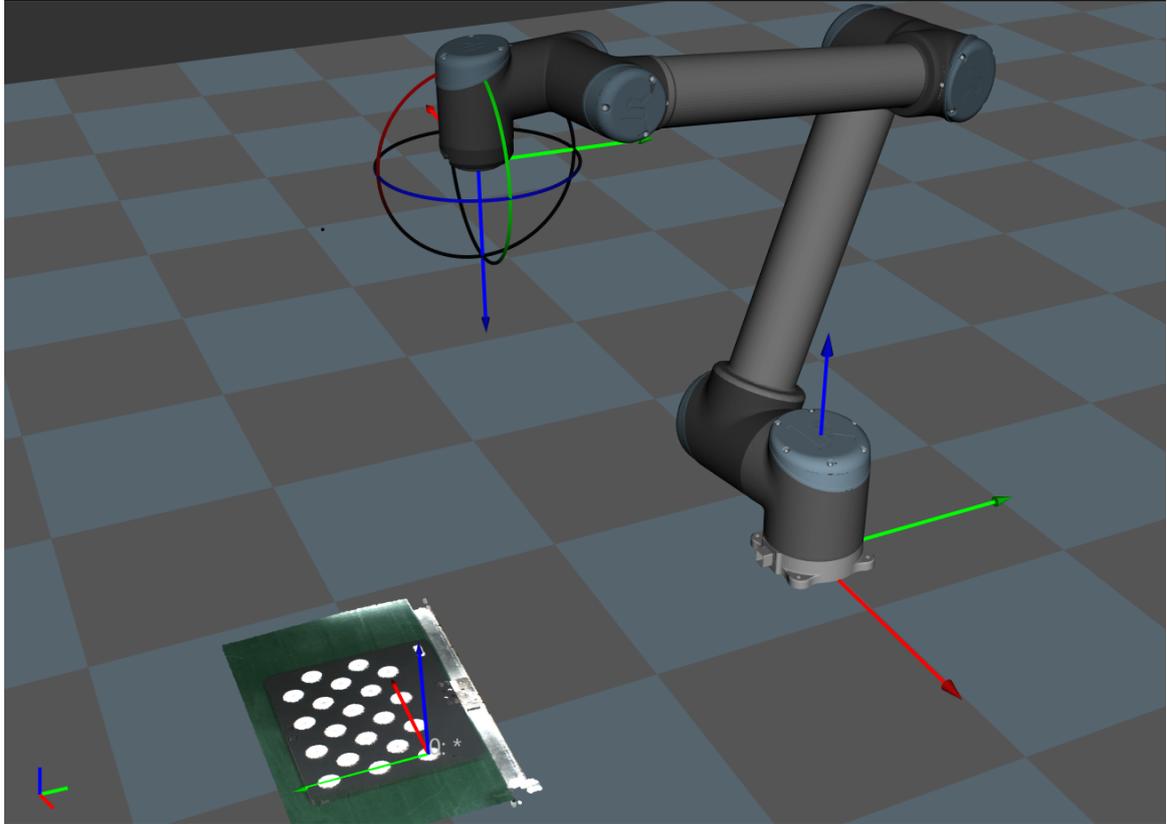
tag::calib\_result\_verification\_eih[] 카메라 외부 파라미터를 계산한 후 시나리오 뷰어에서 고정점을 기준으로 하는 캘리브레이션 보드 포인트 클라우드(캘리브레이션 원 십자의 중심점)의 움직임을 보고 캘리브레이션 결과를 대략적으로 판단할 수 있습니다. 구체적인 작업은 다음과 같습니다.

1. 캘리브레이션 보드를 고정된 위치에 놓습니다.
2. Mech-Viz 소프트웨어를 열고 고정점을 추가하고 캘리브레이션 보드에 있는 캘리브레이션 원 중 하나의 교차 중심점과 겹치도록 합니다.
  - a. 스텝 라이브러리에서 "이동" 스텝을 찾아 프로젝트 편집 영역으로 드래그합니다.
  - b. 이 스텝을 선택한 다음 파라미터 편집 영역에서 **이동 목표 유형** 파라미터를 "작업물 포즈"로 설정하고 포즈의 X/Y/Z 좌표를 조정하여 포즈가 캘리브레이션 보드의 캘리브레이션 원 중 하나의 교차 중심점과 겹치도록 합니다.
3. 로봇을 제어하여 카메라 포즈를 여러 번 변경하고 캘리브레이션 프로세스의 **카메라 파라미터**를

**계산하기** 스텝에서 [ **외부 파라미터 재계산** ] 버튼을 클릭합니다. 이 작업은 카메라가 사진을 캡처하도록 트리거합니다.

4. 시나리오 뷰어에서 캘리브레이션 보드의 포인트 클라우드가 고정점을 기준으로 분명히 움직이는지 여부를 관찰합니다.

캘리브레이션 보드의 포인트 클라우드에 고정점에 대한 명확한 움직임이 없으면 캘리브레이션 결과를 사용할 수 있음을 의미합니다.



1. EIH 시나리오에서 로봇이 연결되지 않은 상태에서 실시간 로봇 포즈를 획득할 수 없습니다. 로봇 현재의 촬영 포인트 포즈를 입력해야 Mech-Viz에서 표시된 포인트 클라우드를 확인할 수 있습니다.
2. 시나리오 뷰어 외에도 Mech-Viz 소프트웨어에서 캘리브레이션 보드의 포인트 클라우드가 고정점에 대해 명확한 움직임을 가지고 있는지 여부를 확인할 수 있습니다.

end::calib\_result\_verification\_eih[]

### 캘리브레이션 결과를 분석하기

#### 캘리브레이션 결과 평가 지표

캘리브레이션이 완료된 외부 파라미터의 경우 정밀도 요구에 만족하는지를 확인해야 합니다. 포인트 클라우드 오차의 참고 기준은 다음과 같습니다.

- 일반 프로젝트 시나리오에서는 모든 데이터 포인트의 오차가 3mm 미만으로 요구됩니다, 즉 <math><3\text{mm}</math>의 데이터 비율이 100%에 도달해야 합니다.
- 정밀도 요구가 높은 시나리오에서는 모든 데이터 포인트의 오차가 2mm 미만으로 요구됩니다, 즉 <math><2\text{mm}</math>의 데이터 비율이 100%에 도달해야 합니다.
- 팔레타이징의 시나리오에서는 모든 데이터 포인트의 오차가 5mm 미만으로 요구됩니다, 즉 <math><5\text{mm}</math>의 데이터 비율이 100%에 도달해야 합니다.

위의 표준은 참고용이며 특정 요구 사항은 실제 생산과정 중의 프로젝트 정밀도 요구 사항에 따릅니다.

#### 캘리브레이션의 정확도에 영향을 미치는 요인

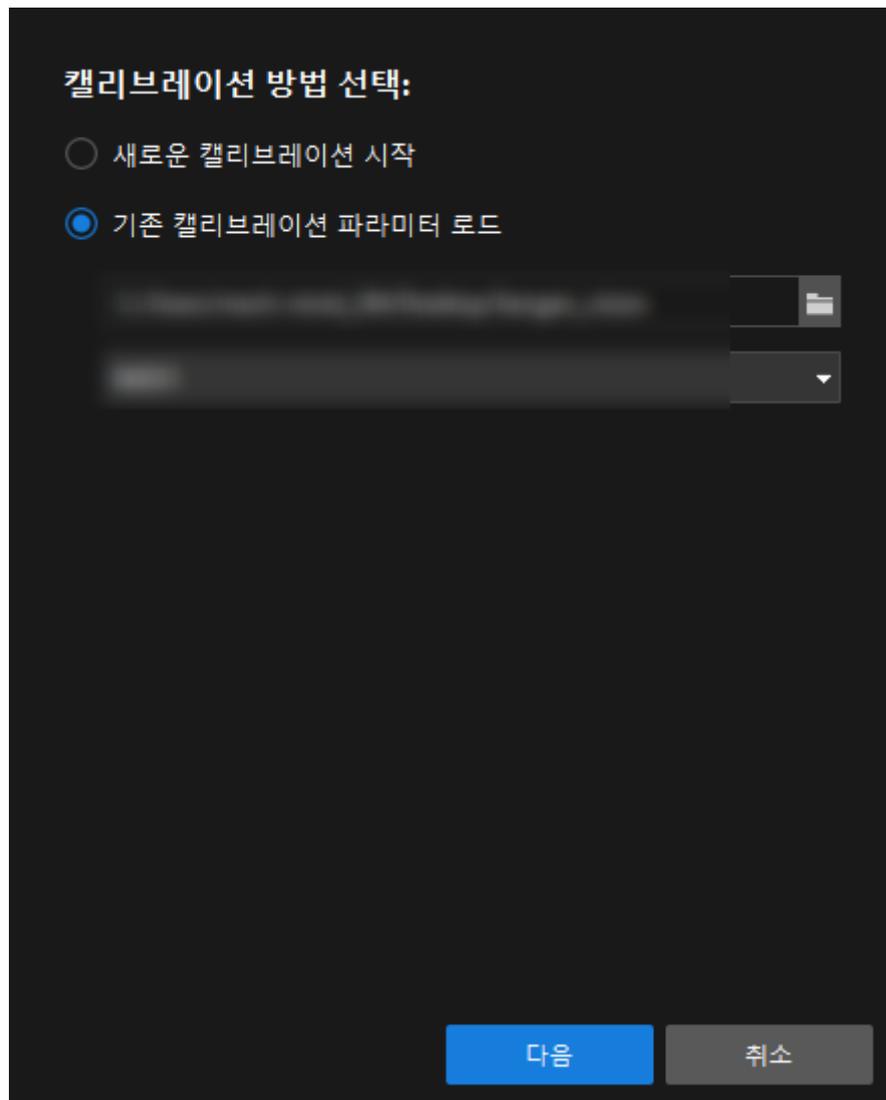
캘리브레이션이 완료된 외부 파라미터의 경우, 정확도가 요구 사항을 만족하지 못하는 경우 오차 발생의 원인을 찾아야 합니다. 다음과 같은 순서대로 캘리브레이션 데이터를 하나씩 확인하고 문제의 원인을 찾아야 합니다.

#### 캘리브레이션 데이터가 문제가 있는지 체크하기

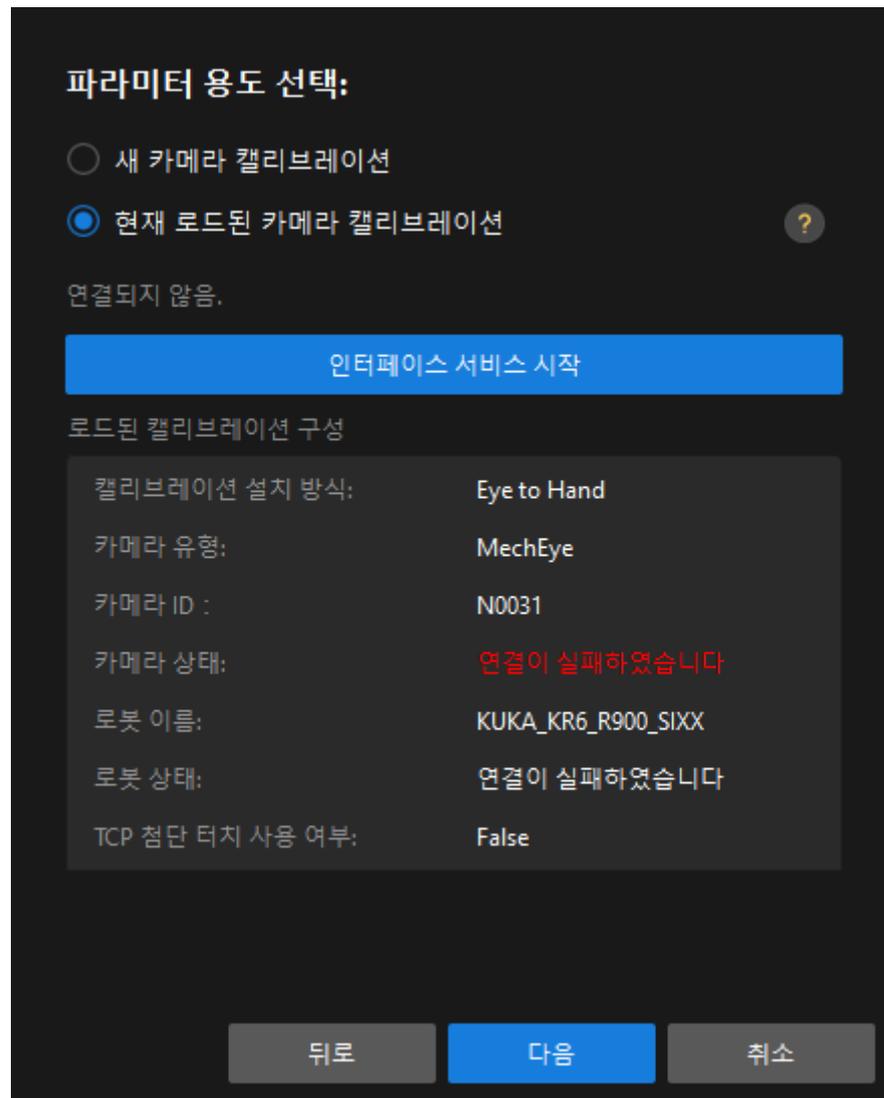
캘리브레이션 데이터는 캘리브레이션 작업 수행 시 생성한 **캘리브레이션 포인트**의 데이터입니다. 카메라 번호 폴더 아래의 `calib_data.json` 파일에 저장되며, 이 파일은 캘리브레이션 포인트의 플랜지 포즈, **캘리브레이션 원** 데이터 등을 기록합니다.

캘리브레이션 포인트 데이터를 보려면 다음 단계를 수행하십시오.

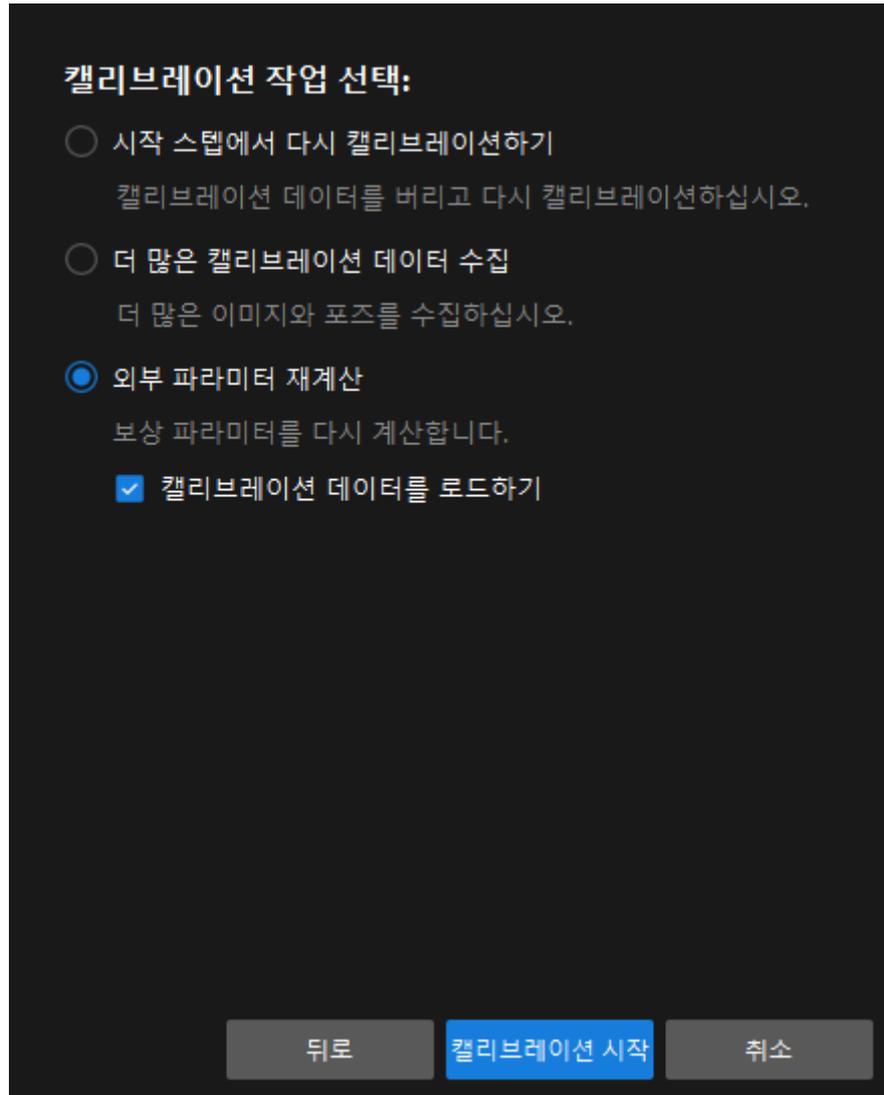
1. Mech-Vision 소프트웨어에서 톨 바의 [ **카메라 캘리브레이션(표준)** ] 버튼을 클릭합니다. **캘리브레이션 사전 구성** 창이 팝업됩니다.
2. **캘리브레이션 방법 선택** 창에서 **기존 캘리브레이션 파라미터 로드** 라디오 버튼을 선택하고 캘리브레이션 파라미터 파일을 선택한 후 [ **다음** ] 버튼을 클릭합니다.



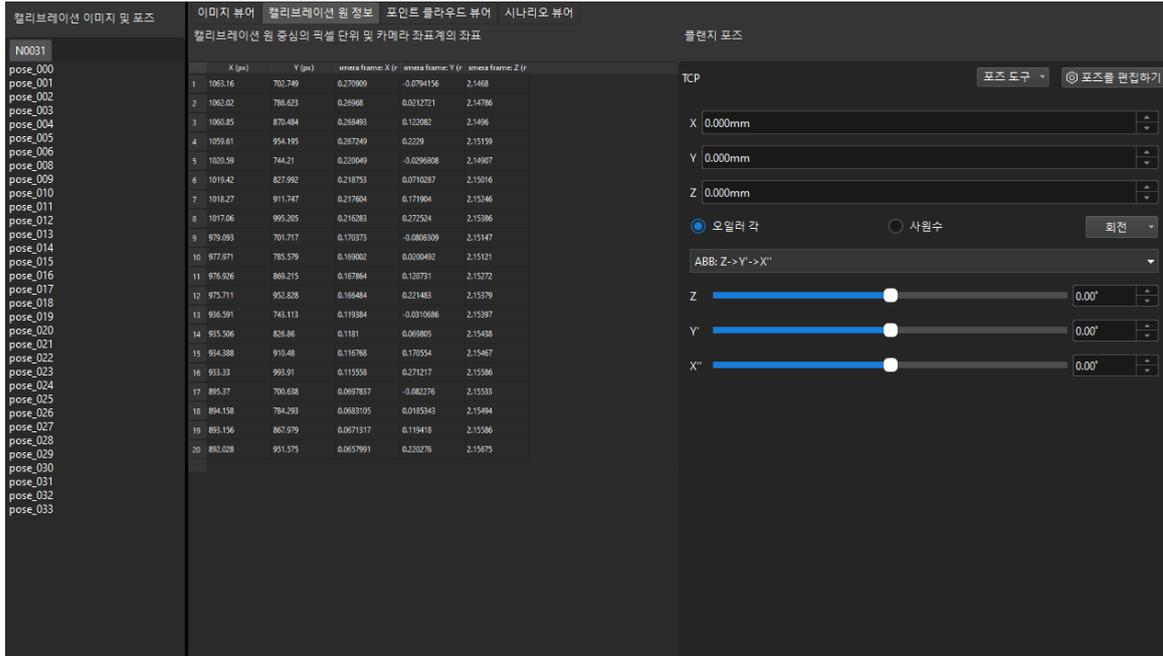
3. 캘리브레이션 구성 적용 방법 선택 창에서 **현재 로드된 카메라 캘리브레이션** 라디오 버튼을 선택한 후 [다음] 버튼을 클릭합니다.



4. 캘리브레이션 작업 선택 창에서 **외부 파라미터 재계산** 라디오 상자를 선택하고 **캘리브레이션 데이터를 로드하기** 라디오 상자를 선택한 다음 [캘리브레이션 시작] 버튼을 클릭합니다.



5. 캘리브레이션의 **카메라 파라미터 계산** 단계에서 **캘리브레이션 이미지 및 포즈 리스트** 패널에서 캘리브레이션 포인트(예: pose\_000)를 선택하고 **캘리브레이션 원 정보** 패널을 클릭하면 캘리브레이션 포인트의 플랜지 포즈와 캘리브레이션 보드의 캘리브레이션 원의 데이터를 볼 수 있습니다.



그림에서 X(pixel) 및 Y(pixel) 열은 현재 캘리브레이션 포인트의 2D 맵에서 캘리브레이션 보드의 모든 캘리브레이션 원 중심의 픽셀 좌표를 나타냅니다. Camera frame: X(m)/Y(m)/Z(m) 열은 카메라 좌표계에서 현재 캘리브레이션 포인트의 3D 맵에서 캘리브레이션 보드의 모든 캘리브레이션 원 중심의 좌표를 나타냅니다.

다음 세 가지 측면에서 캘리브레이션 포인트 데이터를 체크하십시오.

### 1. 피라미드 경로로 이동하는 캘리브레이션 포인트의 플랜지 포즈에서 오일러 각이 전후에 변화가 있는지 확인합니다.



주의해야 할 점은 피라미드의 캘리브레이션 포인트만 취하고 회전 캘리브레이션 포인트는 포함되지 않습니다. 이 체크는 캘리브레이션 보드에서 여러 임의의 포즈 방법을 사용하여 캘리브레이션 데이터를 수집하는 캘리브레이션 시나리오에만 적용할 수 있습니다. **캘리브레이션 이미지 및 포즈 리스트** 패널에서 캘리브레이션 포인트(예: pose\_000)를 선택한 다음 **캘리브레이션 원 정보** 탭을 클릭하여 캘리브레이션 포인트의 플랜지 포즈를 확인합니다. 피라미드 경로로 이동할 때 로봇은 회전 없이 자체 기본 좌표계 또는 플랜지 좌표계를 따라 이동하므로 오일러 각은 기본적으로 일정합니다.

다른 로봇의 경우 정확도에 따라 포즈가 위아래로 변동합니다. 플로트가 1도를 초과하면 로봇 자체의 제로 포인트가 손실되거나 로봇의 정밀도가 떨어질 수 있습니다.

**해결책:** 이 때 핸드-아이 캘리브레이션에 적용하지 않는 경우, 로봇의 제로 포인트 (재배치 방법을 사용하여 로봇 제로 포인트가 정확한지를 확인할 수 있음)를 확인하고 로봇의 문제를 해결해야 캘리브레이션이 계속 진행할 수 있습니다.

### 2. 캘리브레이션 포인트의 "평균 캘리브레이션 원 간격 측정 값 오차"가 기준 값을 초과하는지 확인하십시오.

**캘리브레이션 이미지 및 포즈 리스트** 패널에서 캘리브레이션 포인트를 선택하고 아래 메시지 박스에서 평균 캘리브레이션 원 간격 측정값 오차를 확인합니다.



오차 상한선 기준을 초과하면 캘리브레이션 포인트가 노란색 경고로 바뀝니다. 내부 파라미터 검사 결과의 일부인 평균 캘리브레이션 원 간격 측정 값의 오차는 현재 카메라 내부 파라미터의 오차 값을 부분적으로 반영할 수 있습니다.

## 메시지 박스

[정보] 캡처된 캘리브레이션 원은 캘리브레이션에 사용할 수 있습니다.

평균 캘리브레이션 원 간격 측정값의 오차: **0.77% (통과)**

오차 상한 기준: 2.20%

캘리브레이션 원 간격 측정 값과 실제 값: 100.77 (측정 값) / 100.00 (실제 값)

캘리브레이션 원 포인트 클라우드 변동 (mm) : 0.574698 (평균값) / 1.15823 (최대 값) / 20 (캘리브레이션 원 수량)

감지된 이동 데이터 비교(로봇 vs 카메라) : 로봇 경로에서 상대적인 평행이동된 부분. 이동: [평행이동]0.00/0.00mm. [회전 각도]0.00/0.00°

**해결책:** 정확도 요구 사항이 높지 않은 시나리오에는 표준 값을 약간 초과하는 내부 파라미터를 계속 사용할 수 있습니다. 그러나 고정밀 또는 오차가 큰 상황에서는 카메라의 내부 파라미터를 다시 캘리브레이션하거나 카메라를 공장으로 반환하여 교체하는 것이 좋습니다.



내부 파라미터의 오차는 캘리브레이션 결과에 영향을 미치며 일반적으로 내부 파라미터의 오차가 표준값 안에 있는 경우 문제가 없을 것입니다.

### 3. 캘리브레이션 포인트의 “캘리브레이션 원 포인트 클라우드의 변동“의 최대값이 3mm를 넘었는지를 체크하기

**캘리브레이션 이미지 및 포즈 리스트** 패널에서 캘리브레이션 포인트를 선택하고 아래 메시지 박스에서 **캘리브레이션 원 포인트 클라우드의 변동**을 확인합니다. **캘리브레이션 원 포인트 클라우드의 변동**은 현재 캘리브레이션 포인트에서 평면에 피팅된 캘리브레이션 보드의 모든 캘리브레이션 원 중심 좌표의 위아래 변동을 통계합니다. 이 값은 외부 파라미터의 정확도에 직접적인 영향을 미칩니다.

## 메시지 박스

[정보] 캡처된 캘리브레이션 원은 캘리브레이션에 사용할 수 있습니다.

평균 캘리브레이션 원 간격 측정값의 오차: **0.77% (통과)**

오차 상한 기준: 2.20%

캘리브레이션 원 간격 측정 값과 실제 값: 100.77 (측정 값) / 100.00 (실제 값)

캘리브레이션 원 포인트 클라우드 변동 (mm) : 0.574698 (평균값) / 1.15823 (최대 값) / 20 (캘리브레이션 원 수량)

감지된 이동 데이터 비교(로봇 vs 카메라) : 로봇 경로에서 상대적인 평행이동된 부분. 이동: [평행이동]0.00/0.00mm. [회전 각도]0.00/0.00°

정확도 요구 사항이 높을수록 포인트 클라우드의 변동 값이 작아야 합니다. 일반적으로 최대값은 3mm를 초과하지 않아야 합니다. 세 개의 캘리브레이션 포인트를 초과하는 포인트 클라우드 변동의 최대값이 3mm보다 크면 전체 정확도를 보장할 수 없음을 나타냅니다. 문제를 배제하고 다시 캘리브레이션해야 합니다.

캘리브레이션 원 포인트 클라우드의 변동에 영향을 미치는 가능한 이유는 다음과 같습니다.

**이유 1:** 카메라의 3D 노출 파라미터가 제대로 조정되지 않았거나 포인트 클라우드가 크게 변동하거나 **포인트 클라우드 평활화** 및 **포인트 클라우드 노이즈 제거** 기능이 활성화되지 않았거나 **카메라 게인**이 커져 있습니다. **해결책:** **포인트 클라우드 평활화** 및 **포인트 클라우드 노이즈 제거** 기능을 활성화하고 **카메라 게인**을 비활성화하며 3D 노출을 조정합니다. 캘리브레이션 보드의 도트 포인트 클라우드가 여전히 불완전한 경우 현장 음영 처리가 필요합니다. **이유 2:** ETH 시나리오에서 자동 캘리브레이션을 수행할 때 캘리브레이션 보드가 로봇 플랜지 끝에 단단히 고정되지 않았습니다. 캘리브레이션 중에 로봇이 너무 빨리 움직여 카메라가 사진을 캡처할 때 캘리브레이션 보드가 진동하여 포인트 클라우드가 크게 변동합니다. **해결책:** 로봇의 이동 속도를 낮추고 캘리브레이션 보드를 더 굳게 고정시키며 카메라 캡처의 대기시간을 늘립니다.

#### 보상 파라미터를 재계산하기

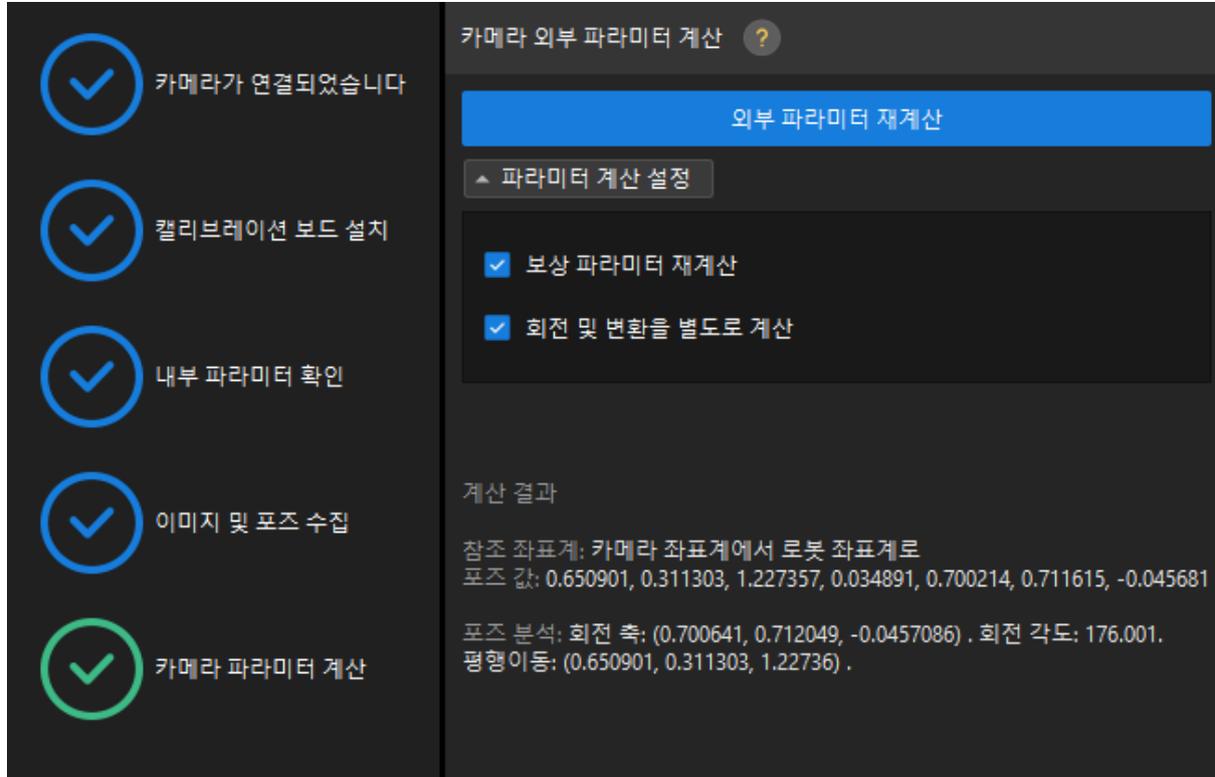
보상 파라미터는 캘리브레이션 구역에 대한 블록 분할 계산을 위한 파라미터의 값입니다. 일반적으로 카메라가 공장에서 출하될 때 카메라의 실제 최대 작동 범위에 따라 이 파라미터가 표시됩니다.



실제 현장의 캘리브레이션 조건과 카메라 출하시의 보상 파라미터는 일치하지 않기 때문에 제품 출하시 보상 파라미터가 적용되지 않을 가능성이 있습니다.

위의 순서대로 캘리브레이션 데이터를 하나씩 확인하여 문제가 없는지 확인하되 그래도 결과가 만족스럽지 않을 경우 보상 파라미터를 다시 계산할 수 있습니다.

보상 파라미터를 다시 계산하려면 캘리브레이션의 **카메라 파라미터를 계산하기** 단계에서 **카메라 파라미터를 계산하기**를 선택한 다음 [ **카메라 외부 파라미터 계산하기** ] 버튼을 클릭합니다.



카메라 외부 파라미터 계산 ?

외부 파라미터 재계산

▲ 파라미터 계산 설정

- 보상 파라미터 재계산
- 회전 및 변환을 별도로 계산

계산 결과

참조 좌표계: 카메라 좌표계에서 로봇 좌표계로  
 포즈 값: 0.650901, 0.311303, 1.227357, 0.034891, 0.700214, 0.711615, -0.045681

포즈 분석: 회전 축: (0.700641, 0.712049, -0.0457086) . 회전 각도: 176.001.  
 평행이동: (0.650901, 0.311303, 1.22736) .



카메라 파라미터를 다시 계산하기 전에 원본 extri\_param.json 외부 파라미터 파일을 백업하고 공장 보상 파라미터로 기록해 두십시오.

#### 4.4.10.5. 자주 나타나는 문제와 해결 방법

출하 시의 보상 파라미터가 무효할 때 추천하는 처리법

출하 시 보상 파라미터에 대한 설명

보상 파라미터는 카메라 외부 파라미터 파일인 extri\_param.json에 저장합니다. 아래 그림과 같습니다.

```

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             ],

```

카메라에 보상 파라미터가 없는 경우, 보상 파라미터의 값은 단위 행렬 (identity matrix) 입니다. 다음 그림과 같습니다.

```

"depthToCloud_offset": {
  "Region0": {
    "Layer0": [
      1,
      0,
      0,
      0,
      0,
      1,
      0,
      0,
      0,
      0,
      1,
      0,
      0,
      0,
      1
    ]
  }
}
    
```

보상 파라미터 재계산하기를 선택하고 계산하기를 클릭하면 현재 모든 캘리브레이션 포인트의 데이터를 다시 최적 보상 파라미터로 피팅할 것입니다. 이때 외부 파라미터 파일 중의 보상 파라미터가 새로고침이 될 것입니다.

출하 시의 보상 파라미터: 출하 시의 보상 파라미터는 이 카메라의 초점 거리에 따라 카메라의 가장 큰 뷰 안에서 캘리브레이션을 한 보상 파라미터의 값입니다. 출하 시의 보상 파라미터의 장단점은 다음과 같습니다.

- **장점:** 출하 시의 보상 파라미터를 사용하면 캘리브레이션 포인트가 비교적 적고 제한된 시야 조건에서 전체 작업 공간의 정확도를 나타내는 외부 파라미터를 계산할 수 있습니다. TCP 터치법을 사용하며, 로봇이 캘리브레이션 보드를 설치한 후 캘리브레이션 공간이 제한되어 전체 작업 영역을 캘리브레이션 할 수 없으면 출하 시의 보상 파라미터가 우선 선택됩니다.
- **단점:** 실제 현장 로봇과 캘리브레이션 경로의 공간 등 조건이 다르기 때문에 출하 시의 보상 파라미터를 사용하면 매년 최적의 결과를 얻을 수 있다고 보장할 수 없습니다. 작업 공간이 캘리브레이션으로 완전히 커버되는 경우 보상 파라미터를 다시 계산하여 더 나은 결과를 얻을 수 있습니다.



출하 시의 보상 파라미터를 계산하고 정밀도가 요구에 맞는 외부 파라미터를 획득한 다음, 작업 구역 전체를 캘리브레이션하지 않은 경우, 더 높은 정밀도를 추구하기 위해 보상 파라미터 재계산하기를 체크하고 재계산하는 것을 권장하지 않습니다.

출하 시의 보상 파라미터가 무효한 것은 현장 보상 파라미터가 새로 고침이 되거나, 출하 시의 보상 파라미터의 백업이 없거나, 카메라 출하 시의 보상 파라미터가 캘리브레이션을 하지 않는 등을 포함합니다.

### TCP 끝단점 터치법을 사용하는 솔루션

이런 경우에 부정확한 보상 파라미터의 문제를 없애기 위해, 다수 세트의 터치 포인트를 추가하고 같이 계산하는 것으로 해결할 수 있습니다.

다수 세트의 터치 포인트를 추가하기: 캘리브레이션 보드를 작업 구역의 최하층 중앙에 배치하고, 끝단점을 사용해서 포인트 3개를 터치하고 터치 포인트 한 세트를 생성합니다. 다음에 캘리브레이션 보드를

테두리에 배치하고 포인트를 한 번 더 터치해서 터치 포인트 한 세트를 더 생성합니다. 이런 식으로 다른 층에서 다수 세트의 터치 포인트를 추가합니다 (일반적으로 3층으로 생성하고 층마다 터치 포인트 2개를 생성하면 됨).

### 다수의 랜덤 캘리브레이션 보드 포즈를 사용하는 솔루션

다수의 랜덤 캘리브레이션 보드의 포즈를 사용하면, 부정확한 보상 파라미터의 문제를 해결하기 위해 작업 구역 전체에 대해 캘리브레이션을 하고 [ **보상 파라미터를 재계산하기** ]를 선택하면 보상 파라미터가 정확하지 않는 문제를 해결할 수 있습니다.

## 비6축 로봇의 캘리브레이션

### 4축 로봇

4축 로봇은 일반적으로 트러스 로봇, SCARA 로봇, 팔레타이징 로봇으로 나뉩니다.

사용 빈도를 고려하여 Mech-Viz는 소수의 SCARA 및 팔레타이징 로봇에만 적용됩니다(적용 범위는 향후 지속적으로 추가될 예정입니다).

#### ● TCP 끝단점 터치법을 사용할 때 끝단점의 고정 문제

4축에 끝단점을 고정하고 4축을 회전시켜 끝단점이 한 지점에서 안정적인지 확인하여 끝단점이 플랜지 중앙에 설치되어 있는지 확인합니다. 4축 로봇의 TCP는 캘리브레이션을 통해 얻을 수 없으며 TCP의 XYZ 값은 수동으로 측정해야 합니다.

#### ● 4축 로봇이 캘리브레이션 할 때 Z 방향을 조정하는 방법

다수의 랜덤 캘리브레이션 보드의 포즈를 사용할 경우, 4축 로봇에 회전 자유도가 부족하기 때문에 캘리브레이션 중 회전값이 부족합니다. 캘리브레이션이 완료된 후, 외부 파라미터의 Z 방향을 수동으로 조정해야 합니다.

1. 로봇의 베이스 좌표 위치를 획득합니다.

2. 캘리브레이션 보드를 로봇의 베이스 좌표의 XY 평면과 평행한 작업면 (지면, 작업대 등을 포함)에 배치합니다.

3. 로봇의 베이스 좌표에서 작업면까지의 거리를 측정합니다. Mech-Viz에서 실제 로봇의 베이스 좌표에서 작업면까지의 거리와 일치하도록 지면 높이를 물체 포즈 Z 방향으로 설정합니다. 아래 그림과 같습니다.

+ image::calibration-reference-adjust\_z.png[align="center"]

4. 외부 파라미터의 Z 방향을 조정하여 캘리브레이션 보드의 포인트 클라우드가 Mech-Viz의 작업면에 위치하도록 하여 Z 방향의 조정이 완료됩니다.

#### ● 트러스 로봇 사용 설명

로봇의 종류에 따라 트러스 로봇의 베이스 좌표의 정의한 위치가 다르기 때문에 베이스 좌표의 위치를 기반으로 Z 방향을 조정하기 어려워서 트러스 로봇의 경우 TCP 터치법만을 사용하기 바랍니다.

트러스 로봇에 대해 가장 많이 사용하는 설치 방식은 카메라를 세 번째 축에 고정시키는 것입니다.

EIH 방식: 이 경우 트러스 로봇은 3축 로봇으로 되어, 네 번째 축은 사용할 수 없으므로 어댑터를 통해서만 로봇에 고정 각도를 보낼 수 있습니다.

ETH 방식: 제한된 수의 고정 촬영 포인트가 존재하는 것을 조건으로 봅니다. 촬영 포인트마다의 옵셋을

기지로 삼아, 매번 다른 촬영 포인트를 바꿔서 어댑터에 로봇 베이스 좌표의 옵셋을 추가합니다.

### 7축 로봇/슬라이드 레일 6축 로봇/5축 로봇



여기에서 언급된 슬라이드 레일 6축 로봇은 슬라이드 레일을 로봇 터치 펜던트에 통합하였기 때문에 7축 로봇에 해당합니다.

위의 제시한 로봇 3 가지 종류를 사용하는 경우, TCP 끝단점 터치법을 추천합니다.

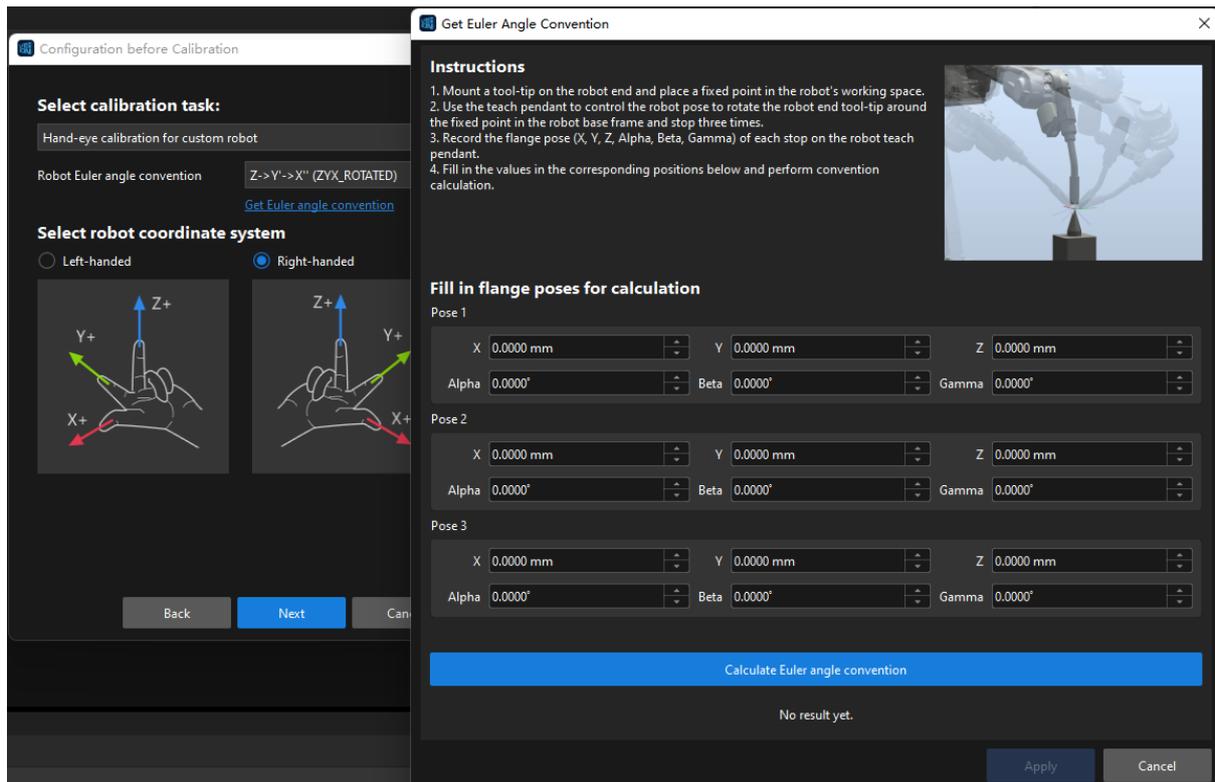
#### ● 다수의 랜덤 캘리브레이션 보드 포즈를 사용하는 7축 로봇의 작업 표준

현장에 적합한 끝단점이 없거나 끝단점을 고정할 수 없는 경우 여러 가지 임의의 캘리브레이션 보드 포즈 방법을 사용하여 7축 로봇을 캘리브레이션할 수 있습니다.

캘리브레이션 프로세스 중에 축 중 하나의 움직임을 제한하고 6축 로봇으로 이상화해야 합니다. 나머지 작업은 6축 로봇의 캘리브레이션과 거의 동일합니다.

#### 로봇 오일러 각의 유형이 불확실할 때 사용하는 캘리브레이션 방법

로봇의 오일러 각이 확실하지 않는 경우, 카메라 캘리브레이션의 인터페이스에 메뉴 바의 “오일러 각 유형 획득하기”를 통해 현재 로봇의 오일러 각 유형을 획득합니다. 아래 그림과 같습니다.



로봇의 포즈를 3번 변환하여 로봇 끝과 테이블에 고정한 끝단점 두 개를 터치합니다. 터치할 때마다 로봇 터치 펜던트에 표시된 포즈를 입력합니다. 터치한 다음 “오일러 각 유형을 획득하기”를 클릭하면 추천하는 오일러 각을 획득할 수 있습니다.

### 현장 작업대에 적당한 고정 끝단점이 없을 때 캘리브레이션하는 방법

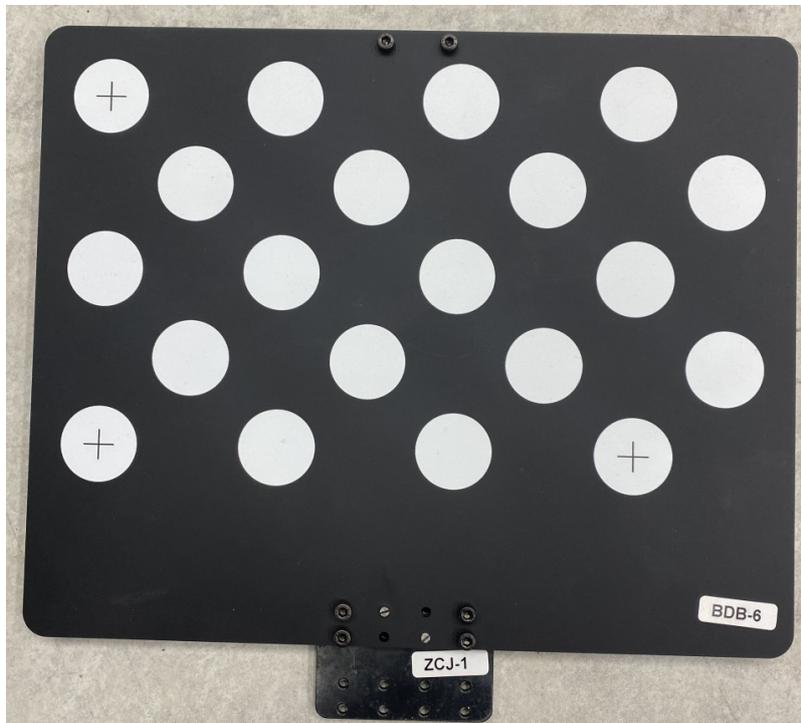
로봇의 정확한 오일러 각 유형을 획득할 수 없기 때문에 TCP 끝단점 터치법을 사용하여 캘리브레이션을 할 때, 오일러 각이 정확한 유형으로 입력할 수 없는 가능성이 있습니다.

이 때 로봇 터치 펜던트에 있는 포즈를 TCP로 전환할 수 있습니다. 전환하면 끝단점의 포즈를 읽게 될 것입니다.

3개의 포인트를 차례로 터치하여 입력에 해당하는 터치 펜던트의 XYZ 3개 값을 읽습니다. 오일러 각 유형 중 임의로 하나를 선택하고 세 입력 포즈의 오일러 각이 일관되도록 상수 값을 입력합니다.

### 장거리 캘리브레이션을 할 때, 대량 캘리브레이션 포인트의 포인트 클라우드 변동이 허용범위를 초과하고 카메라 파라미터 조정이 미비한 경우에 대한 솔루션

캘리브레이션 보드 상태를 체크해야 합니다. 캘리브레이션 보드의 포인트에 크로스가 존재하면 포인트 클라우드에 큰 변동이 생길 것입니다. 아래 그림과 같습니다.



A4 용지에서 몇 개의 원을 잘라내어 각 캘리브레이션 원의 중앙을 덮습니다(원의 가장자리를 덮지 마십시오). 포인트 클라우드 변동에 대한 크로스의 영향은 완료된 후 감소되어야 합니다.

위의 방법으로 여전히 문제가 해결되지 않으면 수동으로 더 많은 포즈를 추가하고 포인트 클라우드 변동이 심한 포즈를 삭제하십시오.

### 경로 캘리브레이션에 대해 흔히 발생하는 문제

#### ● 캘리브레이션 포인트 개수가 많을수록 결과가 좋아지는가?

캘리브레이션 포인트가 너무 많으면 비정상적인 포인트가 생겨 전체 오차 비율이 증가할 수 있습니다.

출하 시의 보상 파라미터를 사용하여 외부 파라미터를 계산할 때 카메라 초점 거리, 캘리브레이션 보드의 크기 등에 따라 각 레이어의 포인트 수를 결정합니다.

- 초점 거리가 300~2000mm인 경우 레이어당 4개의 캘리브레이션 포인트가 있는 2\*2 구성을 사용하는 것이 좋습니다. 레이어 수는 일반적으로 3개입니다. 높은 스택에는 4개의 레이어를 사용합니다.
- 초점 거리가 2000~3500mm인 경우 레이어당 9개의 캘리브레이션 포인트가 있는 3\*3 구성을 사용하는 것이 좋습니다. 레이어 수는 일반적으로 3개입니다. 높은 스택에는 4~5개의 레이어를 사용합니다.
- **캘리브레이션 범위는 모든 작업 구역을 포함해야 하는가?**

출하 시의 보상 파라미터를 사용하는 경우 초점 중심 주변과 카메라의 초점 거리 주변에서 레이어로 출하 시의 캘리브레이션을 수행해야 합니다.

출하 시의 보상 파라미터를 사용하지 않거나 카메라의 초점 거리 주변에서 캘리브레이션이 불가능한 경우 전체 작업 영역에서 캘리브레이션하는 것이 좋습니다.

- **캘리브레이션이 완료된 구역의 외부 파라미터는 정확하지만, 캘리브레이션을 하지 않는 구역의 외부 파라미터는 부정확한가?**

캘리브레이션을 한 후에 생성된 오차 포인트 클라우드는 캘리브레이션이 완료된 구역의 외부 파라미터의 오차를 표시하지만 캘리브레이션을 하지 않는 구역의 외부 파라미터가 부정확하다고는 할 수 없습니다. 반대로 출하 시의 교정 파라미터를 사용하는 경우, 캘리브레이션 구역의 외부 파라미터도 기본적으로 정확하다고 할 수 있습니다.

## 4.5. 측정 모드 사용 가이드

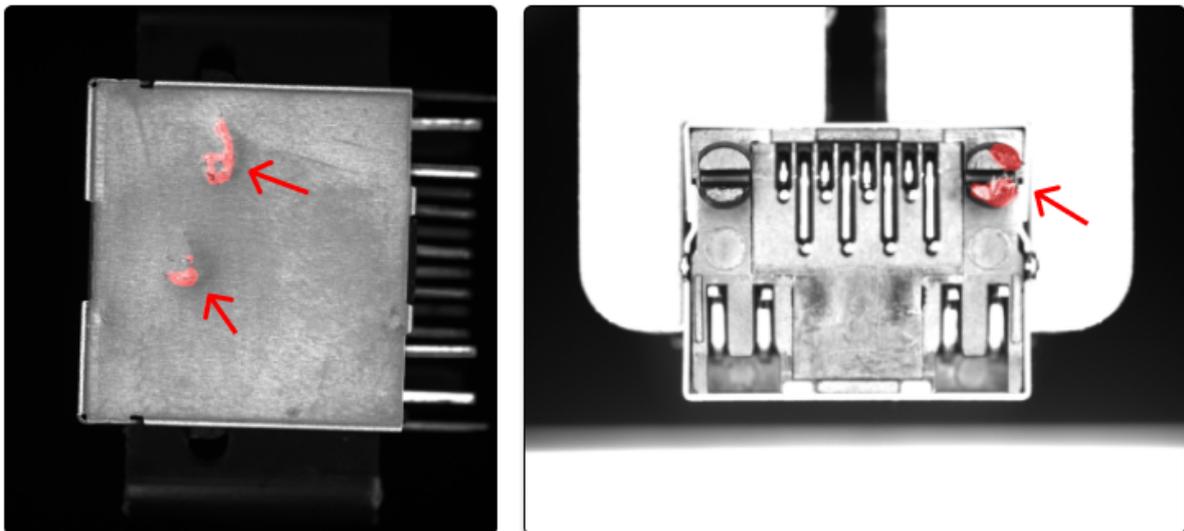
### 측정 모드 퀵 스타트

측정, 검사, 정렬, 인식은 머신 비전이 공업 분야에서 가장 흔히 볼 수 있는 응용 방향입니다.

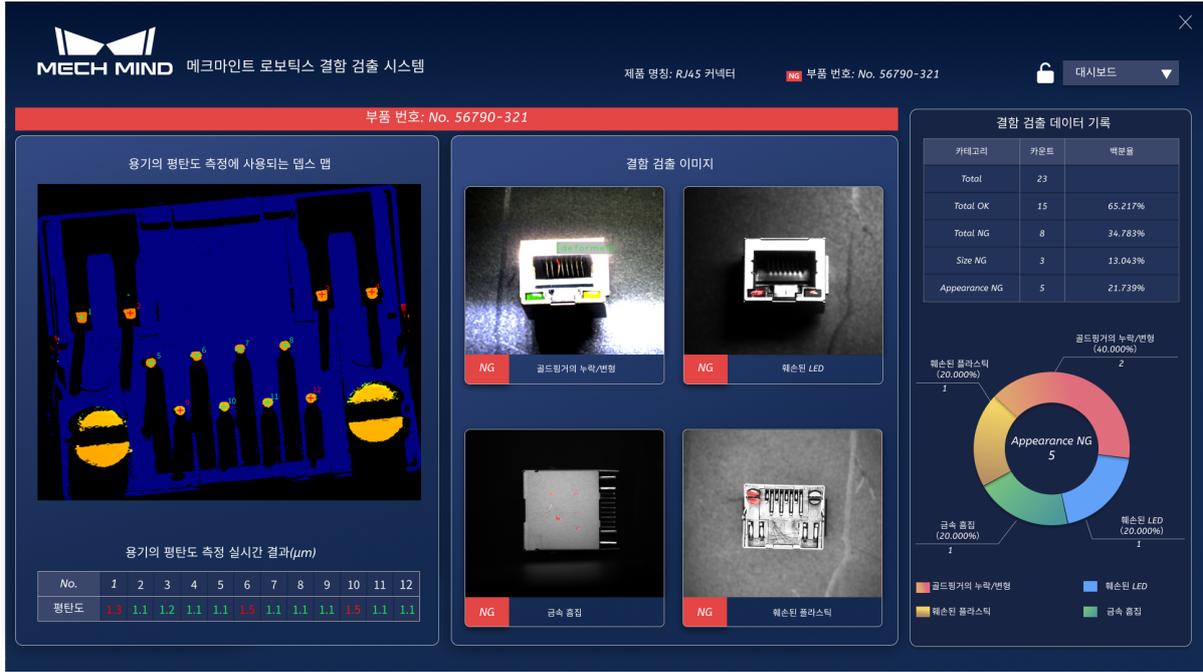
Mech-Mind에서 제공한 측정 모드는 전자 제조, 리튬전지, 자동차, 가전 등 업종에 적용할 수 있습니다.

측정 모드에는 다양한 측정 알고리즘 도구가 포함되어 있으며 패널에서 퀵 디버깅이 가능하며 사용자 정의 오퍼레이터 인터페이스에서 실제 수요에 의해 **정밀 측정** 및 **결합 검출**을 빠르게 배치할 수 있습니다.

아래의 그림은 커넥터 품질 검측(측정+검사) 예시도입니다.



아래의 그림은 사용자 정의 오퍼레이터 인터페이스입니다.



다음 내용을 참고하여 **측정 모드 시작하기**를 통해 첫 응용을 완성해보세요.

### 측정 모드 시작하기

다음 내용을 참고하여 측정 모드 중의 **주요 알고리즘**을 알아보세요.

### 알고리즘 소개

#### 4.5.1. 측정 모드 시작하기

첫번째 응용 -- 물체의 원형 부분 사이즈(원의 반경) 합격 여부를 확인하기 위해 사용자는 원형 부분을 갖춘 물체를 준비해야 합니다.

#### 프로젝트를 새로 만들어 측정 모드로 들어가기

1. **설정 > 옵션 > 고급** 차례로 클릭하여 측정 프로젝트를 선택한 후 **소프트웨어 재부팅** 합니다.
2. **파일 > 새로운 프로젝트** 차례로 클릭합니다.
3. 톨 바에 있는 **측정 모드** 버튼을 클릭하여 측정 모드로 들어갑니다.

#### 입력

측정 모드에서 원시 데이터를 획득하는 방법은 여러 가지 있습니다.

- [카메라에서 이미지를 캡처하기](#)
- [라인 스캐닝 레이저 카메라](#)
- [포인트 클라우드를 읽기\(V2\)](#)
- [이미지 읽기\(V2\)](#)

(라인 스캐닝 레이저 카메라와 **포인트 클라우드를 읽기(V2)**는 **직교 투영**과 같이 2.5D 알고리즘 처리 시 사용함)

본 예시는 **카메라에서 이미지를 캡처하기** 스텝을 통해 원시 이미지를 캡처한 것입니다.

1. 스텝 라이브러리에서 **카메라에서 이미지를 캡처하기**를 찾아 프로젝트 편집 구역으로 드래그합니다.



스텝 라이브러리 오른쪽 하단에 있는 [ > ]를 클릭하면 표시 모드를 바꿀 수 있습니다.

2. **카메라에서 이미지를 캡처하기**에 있는 해당 소개를 참고하여 스텝 파라미터 칸에서 본 스텝의 파라미터를 설정하십시오.

## 처리

본 예시의 목적은 물체 원형 부분 사이즈의 합격 여부를 검사하는 것이므로 **원을 측정하기** 스텝을 사용해야 합니다.

1. 스텝 검색:

스텝 라이브러리에서 **원을 측정하기** 스텝을 찾아 프로젝트 편집 구역으로 드래그합니다.

2. 스텝 연결:

해당 스텝을 클릭하여 소프트웨어 인터페이스 오른쪽 **스텝 입/출력 포트 선택** 창에서 “입력1(컬러 맵)”을 “컬러 맵”으로 설정한 후 스텝이 자동으로 연결될 것입니다.



입력 옵션 선택 시 데이터 유형은 일치해야 합니다. 매칭하지 않는 데이터 유형을 선택하면 팝업창 알림이 나올 것입니다.

3. 프로젝트 실행:

인터페이스 왼쪽 상단에 있는 [ **실행** ]이나 첫 스텝의 을 클릭하면 패널에 이미지가 나올 것입니다.

4. 감지할 원형 부분을 선택하기:

**원을 측정하기** 스텝을 클릭하고 선택 박스는 이미지의 왼쪽 상단에 있습니다. 마우스를 선택 박스로 이동하여(먼저 이미지 크기 조정) 마우스의 화살표 표기가 손 모양으로 바뀔 때 마우스를 길게 눌러 “선택 박스 드래그” 및 “선택 박스 크기 설정”을 진행하십시오.

패널은 **시각화 표시**, **ROI 선택** 및 **포즈 표기** 등 기능을 실현할 수 있습니다.

- a. 패널 왼쪽 상단에서 컬러 맵, 그레이스케일 이미지, 바이너리 이미지를 스위치할 수 있습니다.
- b. 패널 오른쪽 상단에 있는 톨 바에서 표시된 이미지의 크기, 펜의 크기, 명암비를 조절할 수 있습니다.



표시 크기는 마우스 휠로 조절할 수 있습니다.

## 출력

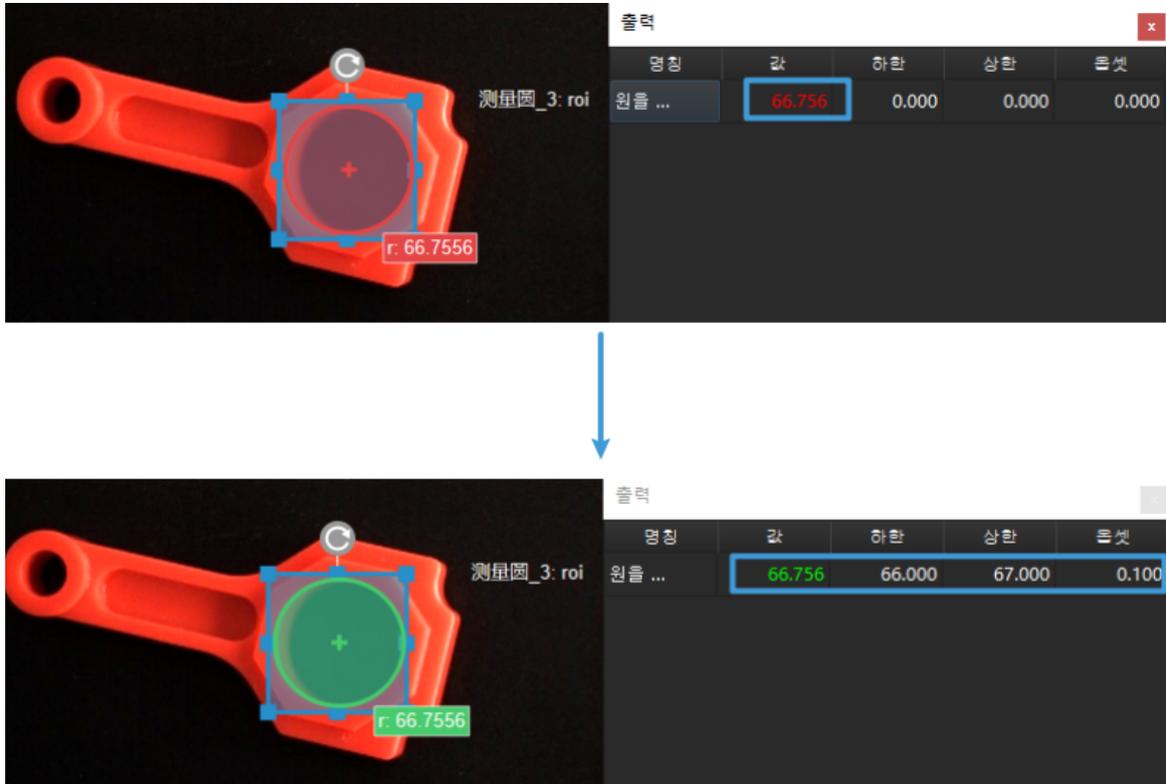
1. 측정된 수치 읽기:

- 패널 구역: ROI 내의 원형 부분의 측정된 수치가 실시간으로 표시됩니다.

- 소프트웨어 오른쪽에 있는 **측정 결과 출력** 창에서 측정된 수치가 표시될 것입니다.

## 2. 측정 결과 합격 여부 판단하기:

상/하한을 설정하지 않으므로 측정된 수치가 빨간색으로 표시됩니다. 이때 상/하한을 따로 설정하면 (측정치는 상/하한 수치 중간이어야 함) 측정치가 초록색으로 변할 것입니다. 아래와 같습니다.



사용자는 상/하한 수치와 오프셋(즉 **합격 기준**)을 설정합니다. 측정치의 컬러를 통해 물체 원형 부분의 사이즈가 합격인지를 판단할 수 있습니다.

## 3. 결과 보기:

측정 모드의 **결과 보기**는 사용자에게 더 나은 결과 판단 표시 방식을 제공합니다. 툴 바에 있는 **+ ▾**를 클릭하고 **[결과 보기]**를 클릭하십시오. 결과 보기의 왼쪽 상단에서 설정된 상/하한 수치에 근거하여 물체가 합격한지를 판단할 것입니다.합격은 OK이고 불합격은 NG입니다.

OK

검색 설정      결과 기록

	명칭	값	하한	상한	오프셋
1	가장 긴 ...	76.977	76.000	77.000	0.000
2	원을 측정하기 ...	66.749	66.000	67.000	0.000



NG

검색 설정      결과 기록

	명칭	값	하한	상한	오프셋
1	가장 긴 ...	76.977	76.000	77.000	0.000
2	원을 측정하기 ...	66.749	66.000	66.700	0.000

#### 4. 오퍼레이터 인터페이스:

툴 바에 있는 [ 오퍼레이터 인터페이스(사용자 정의) ]를 클릭하여 사용자 정의 오퍼레이터 인터페이스로 들어갈 수 있습니다. 이 인터페이스는 맞춤형 디자인이 가능합니다. 필요하시면 Mech-Mind 기술 지원과 연락하십시오.

이로써 첫 응용이 완성됩니다!

### 측정 모드의 일반 사용 프로세스

첫 번째 응용 프로그램을 완료하면 측정 모드의 일반적인 사용 프로세스를 요약 할 수 있습니다.

1. 프로젝트를 만들어 측정 모드로 들어갑니다.
2. 알고리즘 처리 프로세스를 만들어 입력 스텝, 알고리즘 처리 스텝 등을 배치합니다. 측정 모드의 주요 알고리즘은 다음 부분에서 소개해 드리겠습니다.
3. 패널에서 ROI를 선택하고 기준 포인트, 감지 포인트 등을 추가합니다.
4. 결과 미리보기를 열어 출력할 스텝을 선택하여 합격 기준을 설정합니다.
5. 맞춤형 오퍼레이터 인터페이스에서 실시간으로 검사 결과, 제품 이미지, 생산 데이터 통계 결과 등 정보를 볼 수 있습니다.

## 4.5.2. 주요 알고리즘 소개

다양한 스텝을 통해 다양한 알고리즘을 활용합니다.

주요 알고리즘:

1. 2D 알고리즘: 2D 이미지를 기반으로 하여 측정 및 위치 지정 등 기능을 실현합니다.

주요 관련 스텝: 원을 측정하기, 원과 원 간의 거리를 측정하기, 원과 라인 세그먼트 간의 거리를 측정하기, 점과 원 간의 거리를 측정하기, 점과 점 간의 거리를 측정하기, 점과 라인 세그먼트 간의 거리를 측정하기, 라인 세그먼트 간의 거리를 측정하기, 가장 긴 라인 세그먼트를 측정하기, 라인 세그먼트 간의 협각을 측정하기 등.

2. 2.5D 알고리즘: 뎁스 맵을 기반으로 하여 뎁스 차이를 계산합니다.

주요 관련 스텝: 점 간의 높이 차이를 측정하기, 점과 기준선 간의 높이 차이를 측정하기, 점과 평면 간의 높이 차이를 측정하기 등.

3. 3D 알고리즘: 3D 포인트 클라운드를 기반으로 하여 거리를 계산합니다.

주요 관련 스텝: 대각선 길이를 계산하기 등.

4. 기하공차: 제품의 평탄도를 계산하는 데 사용됩니다.

주요 관련 스텝: 평탄도 오차를 계산하기, 평행도 오차를 계산하기 등.

5. 딥 러닝: 딥 러닝 알고리즘을 통해 제품 결함 검출에 사용됩니다.

## 4.6. 보조 도구 사용 가이드

이 부분에서는 Mech-Vision의 보조 도구를 소개할 것입니다. 이러한 도구를 합리적으로 사용하면 프로젝트 실행 속도와 비전 방안의 정확도를 높일 수 있습니다.

- [카메라 뷰어](#)
- [고정 배경을 설정하여 노이즈를 제거하기](#)
- [ROI 설정](#)
- [시나리오 포인트 클라운드 설정](#)
- [딥 러닝 서버](#)
- [매칭 모델 및 픽 포인트 편집기](#)
- [딥 러닝 모델 패키지 관리 도구](#)
- [파라미터 레시피](#)
- [데이터 저장](#)
- [로그](#)

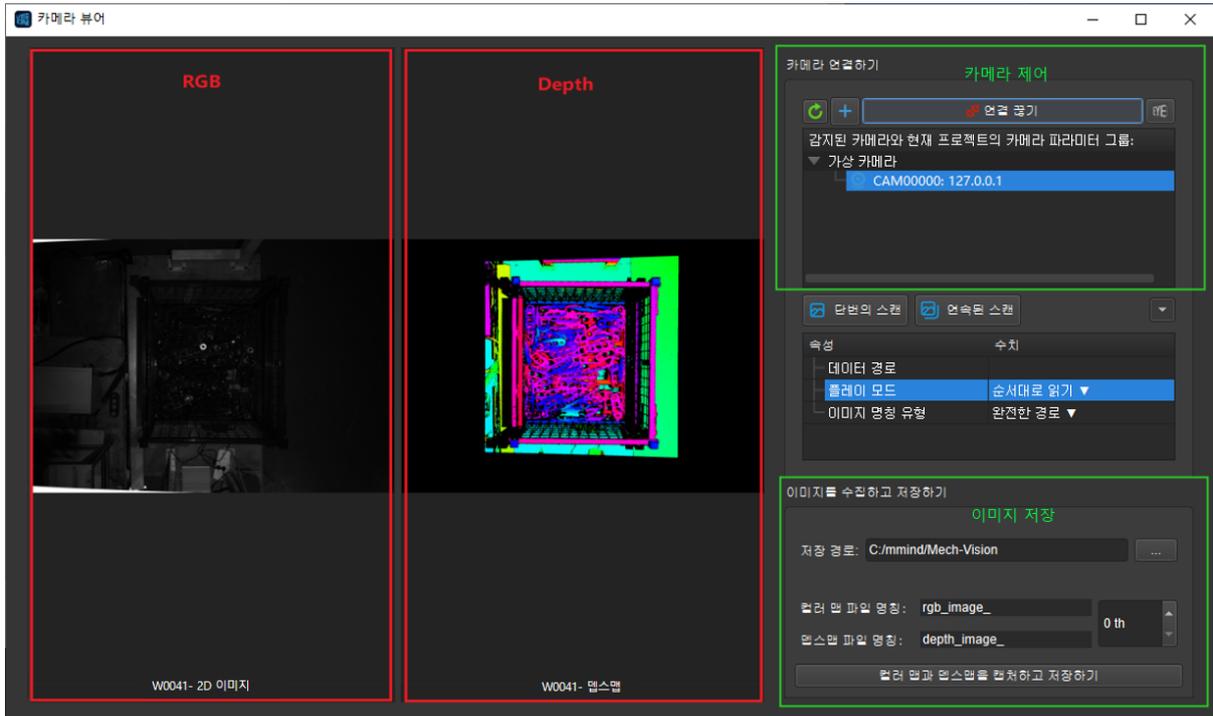
### 4.6.1. 카메라 뷰어

#### 도구 소개

카메라 뷰어는 카메라를 연결하고 이미지 캡처 기능을 제어할 수 있으며 카메라의 속성을 설정할 수 있고 컬러 맵과 뎁스 맵을 표시할 수 있습니다. 또한 저장 경로를 지정해서 이미지를 저장할 수 있습니다.

메뉴 바에서 [카메라 도우미](#) > [카메라 뷰어](#)를 차례로 클릭하여 아래 그림과 같이 카메라 뷰어 인터페이스로

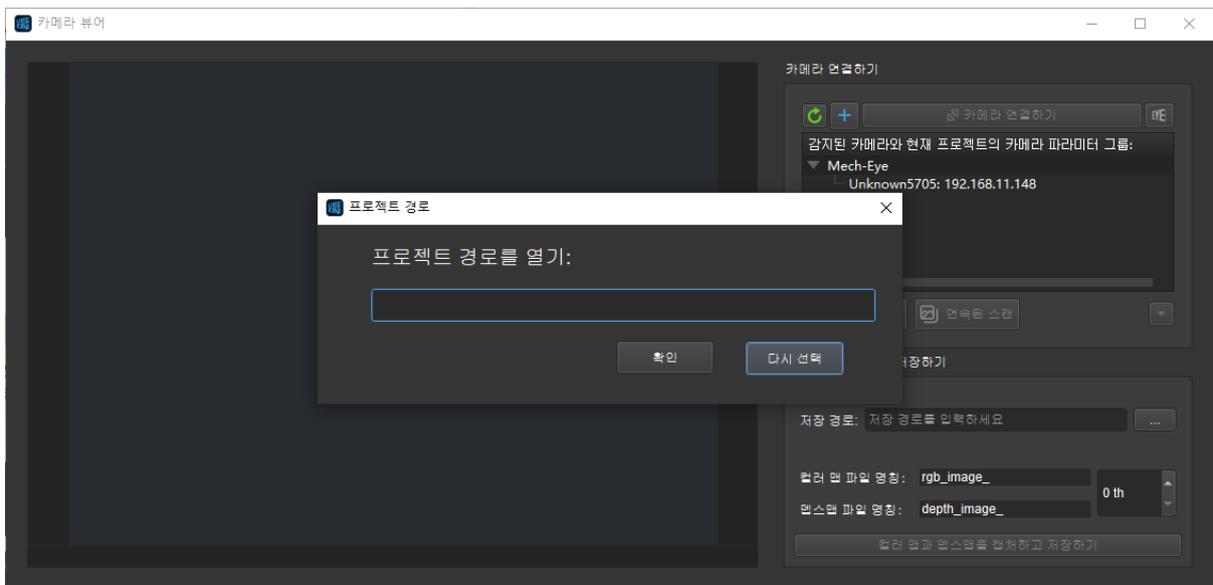
들어갈 수 있습니다.



## 사용 가이드

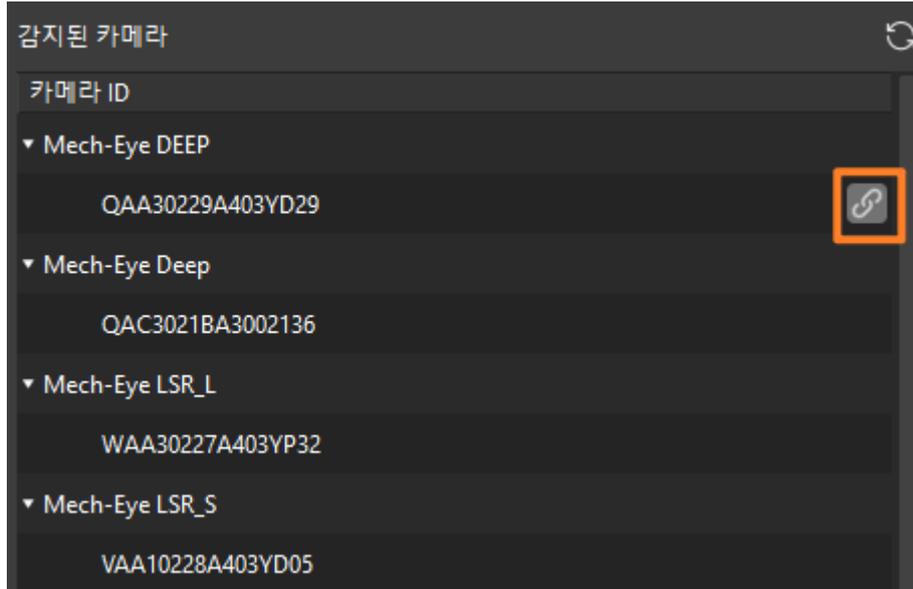
### 카메라 뷰어로 들어가기

메뉴 바를 통해 카메라 뷰어를 연 후 카메라로 촬영할 때 해당 내부/외부 파라미터를 로드해야 하고 내부/외부 파라미터는 해당 프로젝트 리스트에 저장하기 때문에 해당 프로젝트 리스트를 로드하면 됩니다. 아무런 리스트를 로드하지 않고 카메라 뷰어에 들어갈 수도 있습니다.



### 카메라 연결

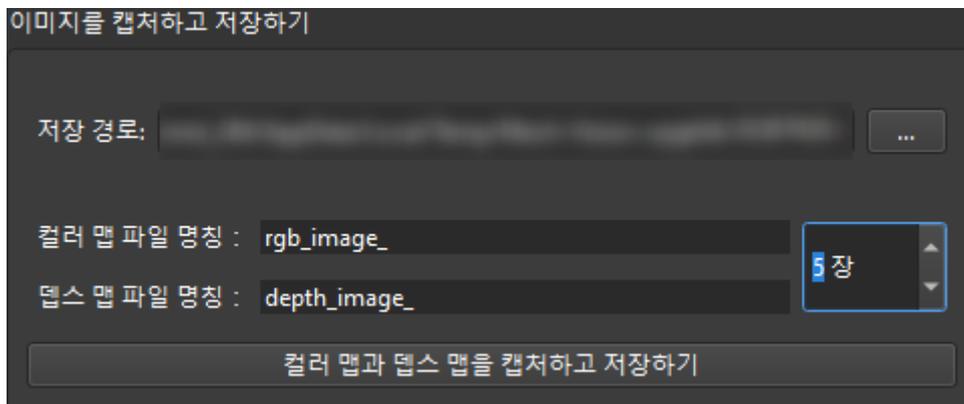
1. 카메라 리스트에서 연결하고자 하는 카메라를 찾은 후, 카메라 번호 우측의  버튼을 클릭하면 카메라와 연결됩니다. 카메라가 성공적으로 연결되면  버튼이 로 변경됩니다.



2. 카메라 리스트 아래의 [ 단일 캡처 ] 또는 [ 연속 캡처 ]를 클릭하면 이미지 캡처 결과를 확인할 수 있습니다.

#### 이미지 획득 및 저장하기

이미지 획득 및 저장이 필요한 경우 이미지 획득 및 저장 영역에서 **경로 저장**을 먼저 설정한 후 **컬러 맵/덱스 맵 파일 이름** 및 수집할 이미지 수를 설정하고 마지막으로 [ **컬러 맵과 덱스 맵을 캡처하고 저장하기** ] 버튼을 클릭하면 해당 경로에 저장됩니다.

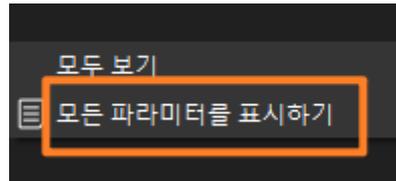


#### 4.6.2. 고정 배경을 설정하여 노이즈를 제거하기

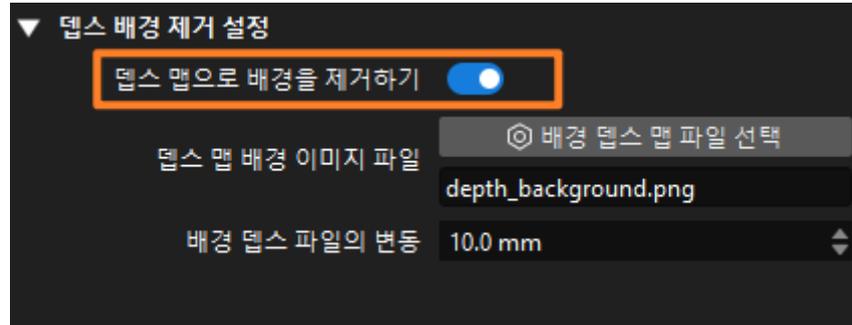
고정 배경을 설정함으로써 배경 덱스 정보를 캡처하고 후속 스텝에서 배경의 간섭을 제거하거나 물체 높이 정보를 계산할 수 있습니다.

고정 배경을 설정하는 작업 프로세스는 아래와 같습니다.

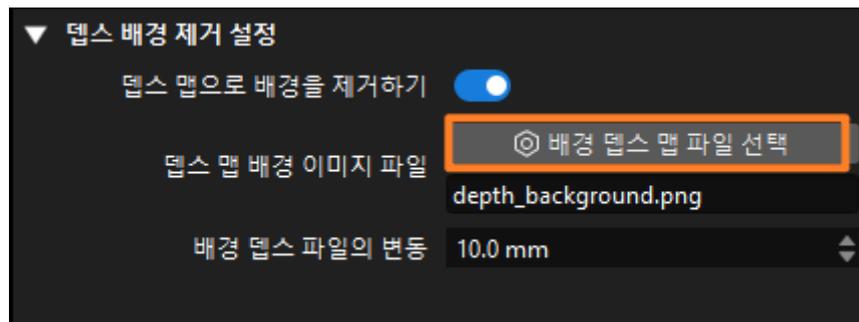
1. Mech-Vision에서 프로젝트를 열어 **카메라에서 이미지를 캡처하기** 스텝을 클릭하고 오른쪽 파라미터 구역에서 마우스 오른쪽 키로 클릭하여 [ **모든 파라미터를 표시하기** ] 옵션을 선택하십시오. 아래 그림과 같습니다.



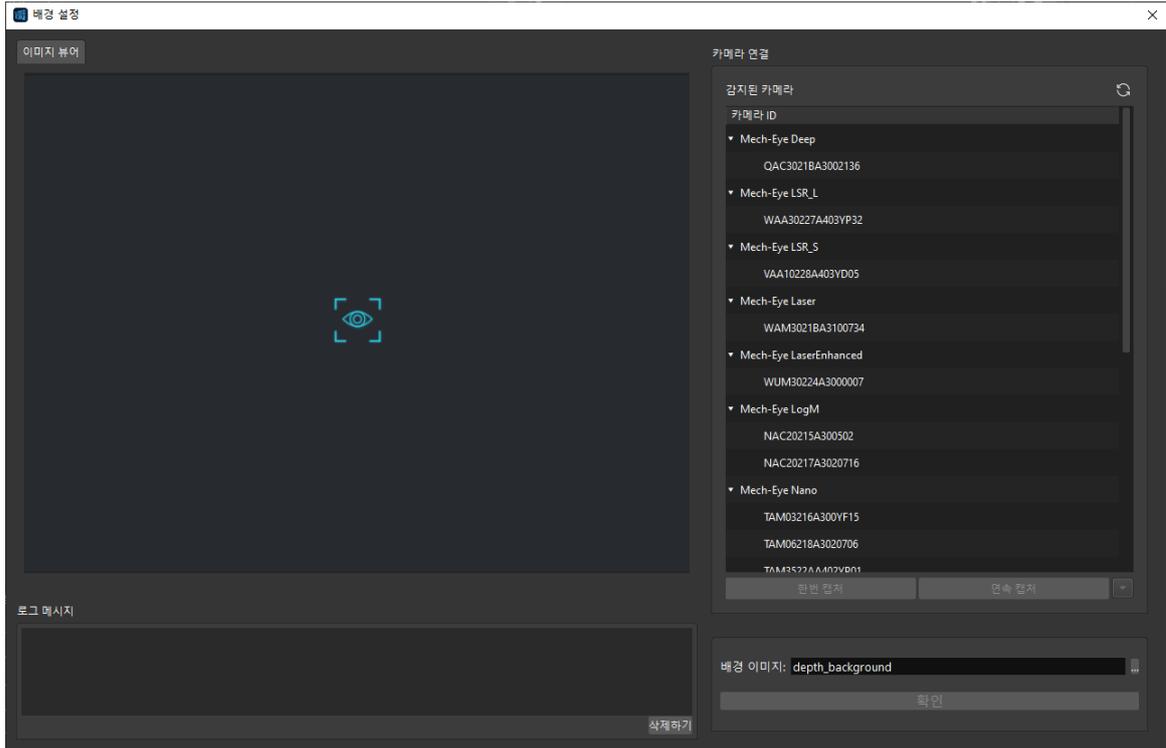
2. 카메라에서 이미지를 캡처하기 스텝의 파라미터에서 [덱스 맵으로 배경을 제거하기]를 선택하십시오. 아래 그림과 같습니다.



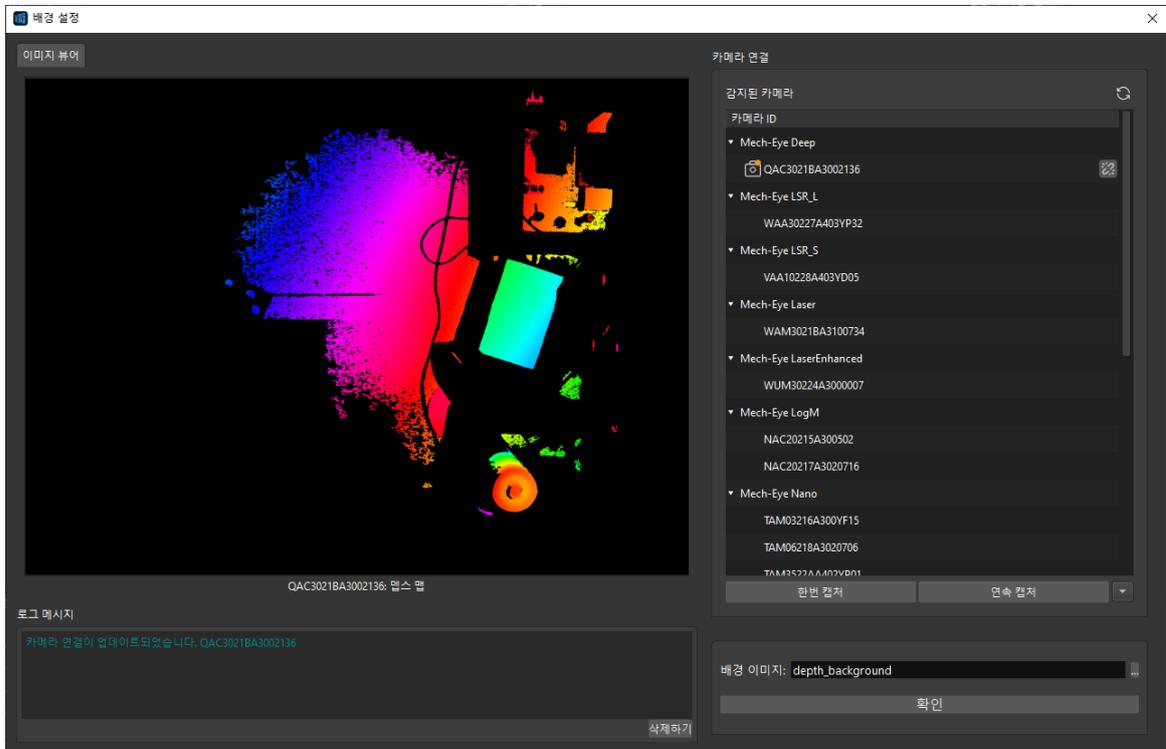
3. [배경 덱스 맵 파일 선택] 버튼을 클릭하여 배경 설정 화면으로 들어갑니다.



4. 배경 설정 화면은 카메라 연결 구역, 이미지 뷰어 구역, 로그 메시지 구역 및 배경 이미지 저장 구역 4가지 구역이 있습니다. 아래 그림과 같습니다.



5. 배경을 설정할 때 먼저 카메라를 연결해야 합니다. 카메라 번호를 더블 클릭하거나 카메라를 선택하고  을 클릭하고 연결합니다. 연결한 후 [ 단번 캡처 ] 혹은 [ 연속 캡처 ] 버튼을 클릭하고 이미지를 촬영합니다. 마지막으로 필요한 배경 이미지를 획득한 후 [ 확인 ]을 클릭하고 이미지를 저장합니다. 아래 그림과 같습니다.

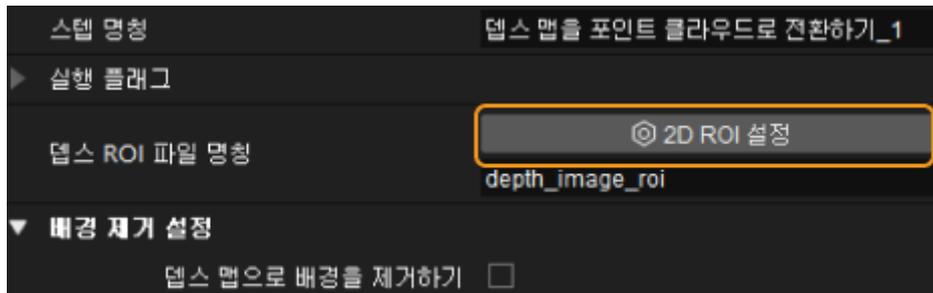


### 4.6.3. ROI 설정

Mech-Vision에서 ROI 설정에는 2D ROI 설정과 3D ROI 설정이 있으며 구체적인 작업 과정은 다음과 같습니다.

#### 2D ROI 설정 프로세스

1. 2D ROI를 설정해야 할 스텝 파라미터에서 [ **2D ROI 설정** ] 버튼을 클릭하고 ROI 설정 화면으로 들어갑니다. 아래 그림과 같습니다.



2. 이미지 표시 구역에서 마우스 왼쪽 버튼을 클릭하고 마우스를 드래그하여 컬러 맵 또는 딥스 맵의 ROI를 선택한 후, 다시 마우스 왼쪽 버튼을 클릭하면 선택이 완료됩니다. ROI를 다시 선택하려면 마우스 왼쪽 버튼을 다시 클릭하여 ROI를 선택합니다.
3. 선택한 ROI의 좌표 범위는 ROI 파라미터에 나타날 것이고 [ **확인** ]을 클릭하면 저장되며 창을 닫습니다.



2D ROI 설정과 관련된 스텝: [2D ROI 내 이미지의 크기를 조정하기](#), [딥스 맵을 분할하기](#), [딥스 맵을 포인트 클라우드로 전환하기](#).

#### 3D ROI 설정 프로세스



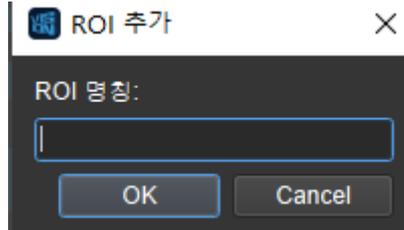
3D ROI를 설치하기 전에 **프로젝트 도우미**에서 시나리오 포인트 클라우드 하나를 기준으로 선택해야 합니다. 그렇지 않으면 오른쪽 하단에 알람 창이 뜰 것이고 설치를 진행하지 못하게 될 것입니다.

1. 3D ROI를 설정해야 할 스텝 파라미터에서 [ **3D ROI 설정** ] 버튼을 클릭하고 ROI 설정 화면으로 들어갑니다. 아래 그림과 같습니다.



2. 3D ROI 설치 창도 포인트 클라우드 구역, 3D ROI 파라미터 구역, 표시 설정 구역 그리고 작업 제시 구역 총 네 구역으로 나뉩니다.
  - 포인트 클라우드 표시 구역은 주로 로봇 베이스 좌표계, 카메라 좌표계, 3D ROI, 목표 포인트 클라우드가 포함됩니다.
  - 3D ROI 파라미터 구역에서 현재 ROI의 명칭, 사이즈, 구역의 중심의 위치 그리고 자세 등 파라미터를 볼 수 있습니다.
  - 디스플레이 구역에서 지면과 로봇 베이스 좌표계의 높이를 조절할 수 있습니다.

- 작업 제시 구역에 3D ROI 설치에 대한 제시가 있습니다.
3. **Ctrl** 키를 누르고 선택 상자의 표면을 드래그하여 3D ROI를 적합한 위치로 조정하고 다음 **Ctrl** 키를 누르고 선택 상자의 측면을 드래그하여 3D ROI를 적합한 사이즈로 조정할 수 있습니다. 그리고 3D ROI 파라미터 구역의 사이즈, 구역의 중심의 위치 그리고 자세 등 파라미터를 조절함으로써 3D ROI를 조절할 수 있습니다.
  4. 자동으로 생성된 3D ROI를 직접 편집하는 방법 외에 새로운 3D ROI를 만들 수도 있습니다. **+** 버튼을 클릭하면 아래 그림과 같이 3D ROI 추가 팝업창이 나올 것입니다. 거기서 추가할 3D ROI의 명칭을 입력하고 **[확인]**을 클릭하면 추가가 완성됩니다. 다음으로 표시 구역에서 3D ROI를 조절하면 됩니다. 대상 영역에서 폴다운 메뉴를 클릭하여 다른 3D ROI를 전환 할 수 있습니다.



3D ROI 설치에 관련된 스텝: 덤스 맵 3D ROI 외의 영역을 무효로 설정하기, 덤스 맵 3D ROI에 있는 0인 영역을 추출하기, 3D ROI 내의 포인트 클라우드를 추출하기, 포즈가 3D ROI에 있는지 확인하기, 3D ROI 내의 포즈를 획득하기, 3D ROI 중심을 읽기.

2D ROI를 설정하고 3D ROI를 설정할 때 다음 문제에 주의하십시오.



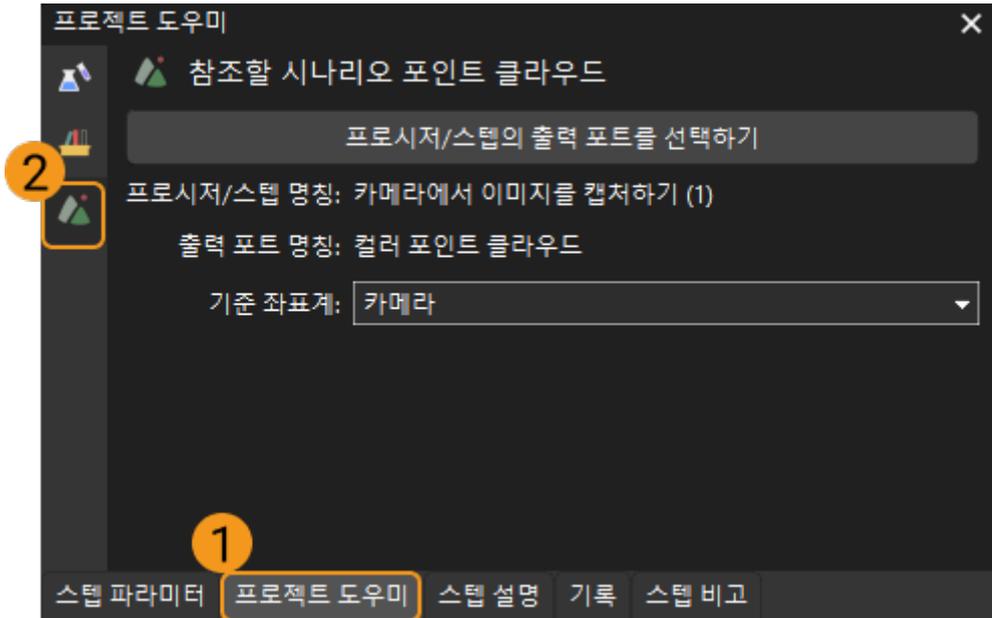
- 다른 3D ROI를 만들었다면 적합한 3D ROI를 사용하고 있는지를 확인하세요.
- 2D ROI 및 3D ROI를 선택할 때 상자 또는 빈 등 용기도 고려해야 합니다. 용기의 훼손 및 흔들림으로 인한 피킹 편차를 방지하기 위해 작업물과 용기를 함께 선택해야 합니다. 동시에 선택 범위는 작업 현장의 실제 크기보다 크고 여백이 있어야 합니다.

#### 4.6.4. 시나리오 포인트 클라우드 설정

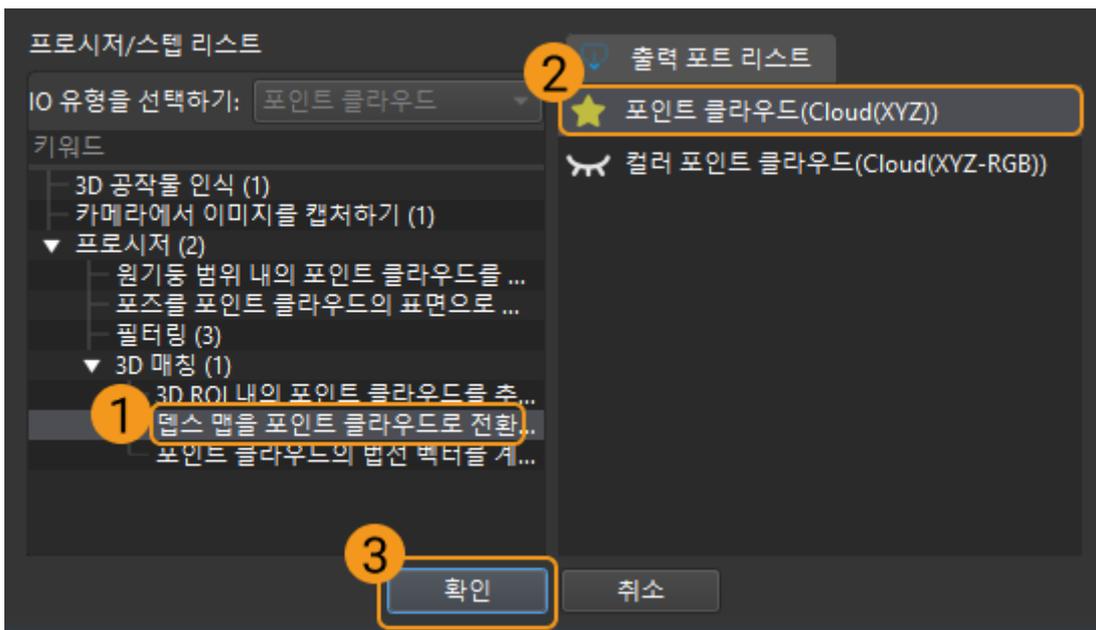
카메라 시야 내의 모든 포인트 클라우드를 시나리오 포인트 클라우드라고 하며 시나리오 포인트 클라우드에 따라 ROI를 설정할 수 있습니다. 따라서 ROI를 설정하기 전에 시나리오 포인트 클라우드를 설정해야 합니다.

이 부분에서는 시나리오 포인트 클라우드를 설정하는 방법에 대해 설명할 것입니다.

1. 먼저 **[프로젝트 도우미]**를 클릭한 다음 **+** 버튼을 클릭하여 시나리오 포인트 클라우드 설정 화면으로 들어갑니다.



2. [프로시저/스텝의 출력 포트를 선택하기]를 클릭하고 팝업 창에서 시나리오 포인트 클라우드 스텝과 대응하는 포트를 출력하기를 차례로 선택한 다음 [확인]을 클릭합니다.



시나리오 포인트 클라우드를 출력하는 스텝을 선택할 때 후속 스텝의 처리 속도를 보장하기 위해 일반적으로 **덱스 맵을 포인트 클라우드로 전환하기**를 선택합니다. 스텝에서 출력한 포인트 클라우드가 시나리오 포인트 클라우드인 경우 **카메라에서 이미지를 캡처하기** 스텝에서 출력한 포인트 클라우드를 선택하지 않는 것이 좋습니다.



- 아이콘 는 이 출력 포트가 다음 스텝의 입력 포트에 연결되었음을 나타냅니다.
- 아이콘 은 이 출력 포트가 다음 스텝의 입력 포트에 연결되지 않았음을 나타냅니다.

3. 시나리오 포인트 클라우드 선택이 완료된 후 실제 상황에 따라 참조 좌표계도 선택할 수 있습니다.



시나리오 포인트 클라우드를 설정한 후 **ROI 설정**을 참조하여 ROI 설정 관련 내용에 대해 알아볼 수 있습니다.

#### 4.6.5. 매칭 모델 및 픽 포인트 편집기

Mech-Vision은 사용자가 대상 물체에 대한 포인트 클라우드 모델을 만들고 포인트 클라우드 모델을 편집하고 픽 포인트를 추가할 수 있도록 매칭 모델 및 픽 포인트 편집기 도구를 제공하며 일반적으로 부품 로드 및 언로드와 같은 작업물을 식별해야 하는 프로젝트에서 사용됩니다.

매칭 모델 및 픽 포인트 편집기는 프로젝트의 데이터와 구성을 호출해야 하므로 이 도구를 사용하기 전에 해당 프로젝트를 먼저 열거나 새 프로젝트를 만들어야 합니다. 다음으로 툴 바에 있는 매칭 모델 및 픽 포인트 편집기 버튼을 클릭하면 이 도구를 사용할 수 있습니다.

매칭 모델 및 픽 포인트 편집기에서 픽 포인트를 추가하는 방법은 주로 두 가지가 있습니다. 매칭 모델 및 픽 포인트 편집을 시작하기 전에 [인터페이스 소개](#)에서 도구의 인터페이스 및 기능에 대해 자세히 알아보는 것이 좋습니다.

##### 티치 펜던트를 통해 로봇을 이동하여 픽 포인트를 획득하기

- 적용 가능한 시나리오: 피킹 정확도에 대한 요구 사항이 높고 픽 포인트가 공작물 표면에 없습니다.
- 사용 프로세스: 먼저 픽 포인트를 얻은 다음 포인트 클라우드 모델을 생성하고 마지막으로 포인트 클라우드 모델을 편집합니다.
  - [티치 펜던트 픽 포인트를 추가하기](#)
  - [포인트 클라우드 모델을 생성하기](#)
  - [포인트 클라우드 모델을 편집하기](#)

##### 매칭 모델 및 픽 포인트 편집기에서 직접 픽 포인트 추가

- 적용 가능한 시나리오: 피킹 정확도에 대한 요구 사항이 낮고 픽 포인트가 공작물 표면에 있습니다.
- 사용 프로세스: 먼저 포인트 클라우드 모델을 생성한 다음 포인트 클라우드 모델을 편집하고 마지막으로 픽 포인트를 추가합니다.
  - [포인트 클라우드 모델을 생성하기](#)
  - [포인트 클라우드 모델을 편집하기](#)
  - [포즈를 추가함으로써 픽 포인트를 추가하기](#)

위의 두 가지 방법 외에도 기존 픽 포인트 파일을 가져올 수도 있습니다. 툴 바의  버튼을 클릭하고 팝업한 파일 선택 창에서 기존의 픽 포인트 파일을 선택하면 빠르게 픽 포인트를 추가할 수 있습니다.



일부 스텝은 실행할 때 매칭 모델 및 픽 포인트 편집기를 통해 생성된 파일을 사용해야 합니다. 예를 들면 아래와 같습니다.

- 포인트 클라우드 모델 파일을 모델 파일로 사용해야 하는 스텝: [3D 근사 매칭](#), [3D 근사 매칭\(멀티 모델\)](#), [3D 상세 매칭](#) 및 [3D 상세 매칭\(멀티 모델\)](#).
- 픽 포인트 파일을 사용하는 데 필요한 스텝: [여러 개의 픽 포인트에 매핑하기](#).

#### 4.6.6. 딥 러닝 서버

이 부분에서는 딥 러닝 서버를 사용하는 방법을 소개합니다.

##### 도구 소개

딥 러닝 서버는 Mech-Vision 소프트웨어에서 딥 러닝 서비스를 관리하기 위해 제공하는 도구입니다. 소프트웨어 기본 창의 메뉴 바에서 [딥 러닝 > 딥 러닝 서버](#)를 차례로 선택하여 딥 러닝 서버를 열 수 있습니다.

일반적으로 사용자는 딥 러닝 서버 페이지에서 어떤 작업도 수행할 필요가 없습니다. Mech-Vision 프로젝트를 열 때 딥 러닝 서버는 딥 러닝 스텝 파라미터 **서버 IP** 및 **서버 포트(1~65535)** 설정이 자동으로 활성화됩니다.

### 호환성 설명

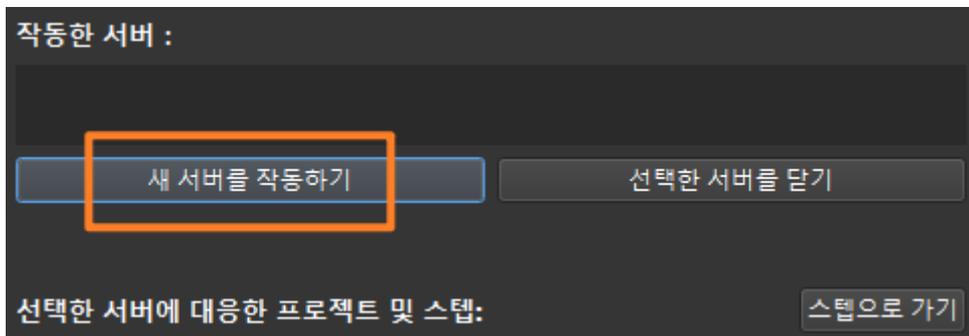
딥 러닝 유형 스텝에서 딥 러닝 서버를 열어야 하는지 여부는 [딥 러닝 유형 스텝에 대한 호환성 설명](#)을 참조하십시오.

### 작업

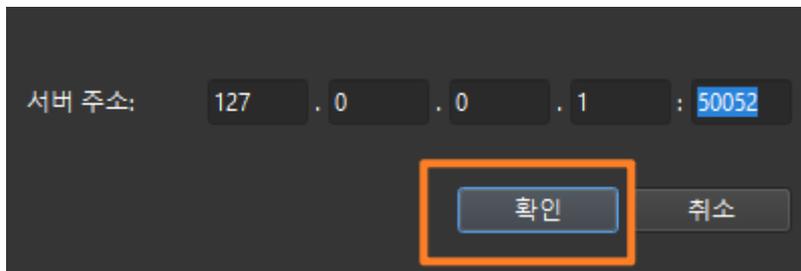
이 부분에서는 딥 러닝 서버에 대한 일반적인 작업을 제공합니다.

#### 새로운 딥 러닝 서버를 열기

1. 딥 러닝 서버 인터페이스에서 [ **새 서버를 작동하기** ] 버튼을 클릭합니다.

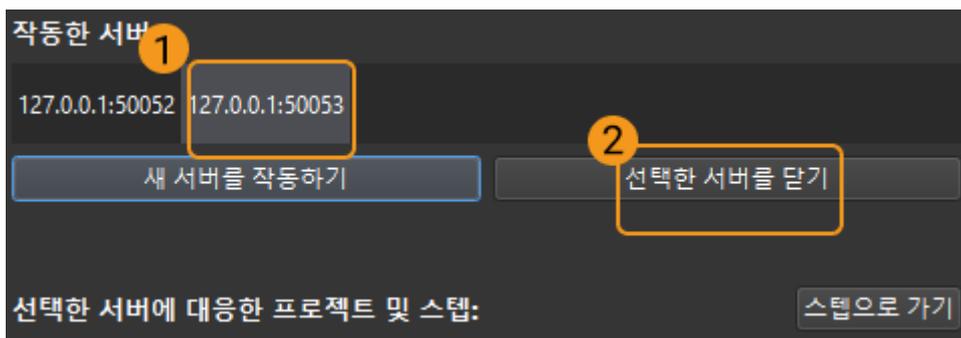


2. 팝업 창에 서버 주소와 포트 번호를 입력한 후 [ **확인** ]을 클릭하면 새로운 딥 러닝 서버가 시작됩니다.

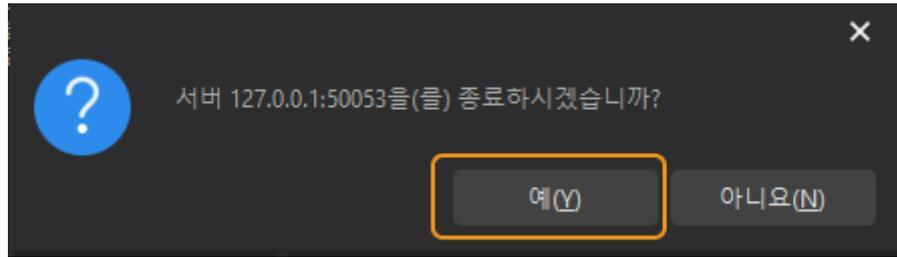


#### 선택한 딥 러닝 서버를 닫기

1. 여러 개의 딥 러닝 서버가 활성화된 경우 딥 러닝 서버 인터페이스에서 종료할 딥 러닝 서버를 선택한 후 [ **선택한 서버를 닫기** ] 버튼을 클릭합니다.



2. 팝업 창에서 [예(Y)]를 클릭하여 선택한 딥 러닝 서버를 닫습니다.



#### 딥 러닝 서버 재시작

딥 러닝 서버를 다시 시작해야 하는 경우 먼저 딥 러닝 서버를 닫은 다음 동일한 서버 주소와 포트 번호로 딥 러닝 서버를 열어야 합니다.

### 4.6.7. 딥 러닝 모델 패키지 관리 도구

이 부분은 딥 러닝 모델 패키지 관리 도구의 사용 방법과 주의사항에 대한 소개입니다.

#### 도구 소개

딥 러닝 모델 패키지 관리 도구는 딥 러닝 모델 패키지를 관리하기 위해 Mech-Vision 소프트웨어에서 제공하는 도구로, Mech-DLK 2.2.0 이후에 도출한 딥 러닝 모델 패키지(단일 수준 모델 패키지 또는 캐스케이딩 모델 패키지)를 최적화하고 운영 모드, 하드웨어 유형, 모델 효율성 및 모델 패키지 상태를 관리할 수 있습니다. 또한 이 도구는 IPC의 GPU 사용량을 모니터링할 수 있습니다.

프로젝트에서 딥 러닝 관련 스텝을 사용할 때 먼저 모델 패키지를 딥 러닝 모델 패키지 관리 도구로 도입한 후 딥 러닝 관련 스텝에서 사용할 수 있습니다. 이 도구로 모델 패키지를 미리 도입하면 모델 패키지의 최적화를 미리 완료할 수 있습니다.



Mech-DLK 2.4.1 버전 이후 모델 패키지는 단일 및 캐스케이딩 모델 패키지 두 가지로 나뉩니다.

- 단일 모델 패키지: 모델 패키지에 단 하나의 딥 러닝 알고리즘 모듈이 있는 모델. 예를 들면 "인스턴스 세그멘테이션" 모델과 같습니다.
- 캐스케이딩 모델 패키지: 모델 패키지에는 딥 러닝 알고리즘 모듈의 여러 모델이 직렬로 존재하며 이전 모델의 출력은 다음 모델의 입력입니다. 예를 들면 모델 패키지에 "물체 검출"과 "인스턴스 세그멘테이션" 두 모델이 있으면 추론 순서는 **물체 검출** > **인스턴스 세그멘테이션**이며 "물체 검출" 모델의 출력은 "인스턴스 세그멘테이션" 모델의 입력이 됩니다.

#### 인터페이스 소개

이 도구는 다음 두 가지 방법으로 열 수 있습니다.

- 프로젝트를 새로 만들거나 기존 프로젝트를 연 후 메뉴 바에서 menu:딥 러닝 [ 딥 러닝 모델 패키지 관리 도구]를 선택합니다.
- 프로젝트에서 "딥 러닝 모델 패키지 추론" 스텝을 선택하고 스텝 파라미터에서 [ 모델 패키지 관리 도구 ] 버튼을 클릭합니다.

딥 러닝 모델 패키지 관리 도구 인터페이스의 각 옵션 필드에 대한 소개는 아래 표와 같습니다.

필드 명칭	소개
캐시(cache) 모델 패키지 명칭	도입한 딥 러닝 모델 패키지의 명칭을 확인할 수 있습니다.

필드 명칭	소개
프로젝트 사용	모델 패키지를 사용하는 Mech-Vision 프로젝트를 볼 수 있습니다.
모델 패키지 유형	모델 패키지 유형을 볼 수 있습니다. 예: "물체 검출"(단일 모델 패키지), "물체 검출 + 결합 세그먼테이션"(캐스케이딩 모델 패키지)
실행 모드	딥 러닝 모델 패키지 추론의 실행 모드를 선택할 수 있습니다("공유 모드" 및 "성능 모드"를 포함).
하드웨어 유형	GPU(기본값), GPU(최적화), CPU 등 모델 패키지 추론에 사용되는 하드웨어 유형을 볼 수 있습니다.
모델 효율성	모델 패키지 추론의 효율성을 구성할 수 있습니다.
모델 패키지 상태	"최적화 중", "준비 완료", "최적화 실패" 등 모델 패키지의 상태를 확인할 수 있습니다.

### 실행 모드

- 공유 모드: 동일한 모델을 사용하는 다른 스텝은 추론을 위해 대기합니다. 이 옵션은 런타임 리소스를 절약합니다.
- 성능 모드: 동일한 모델을 사용하는 다른 스텝은 병렬로 추론합니다. 이 옵션은 실행 속도를 향상시키지만 더 많은 런타임 리소스를 소비합니다.



### 하드웨어 유형

- CPU: CPU를 사용하여 딥 러닝 모델 추론을 수행하며 GPU에 비해 추론 시간이 늘어나고 인식 정확도가 떨어집니다.
- GPU(기본값): 하드웨어 장치에 따라 최적화하지 않고 모델 패키지 추론을 수행합니다. 모델 패키지 추론은 가속화되지 않습니다.
- GPU(최적화): 하드웨어 장치에 따라 최적화 후 모델 패키지 추론을 수행합니다. 최적화는 한 번만 수행하면 되며 5~15분 소요되고 추론 시간은 최적화 후 줄어든 것입니다.

## 사용 가이드

이 부분에서는 딥 러닝 모델 패키지 관리 도구에 대한 일반적인 작업에 대해 소개합니다.

### 딥 러닝 모델 패키지를 도입하기

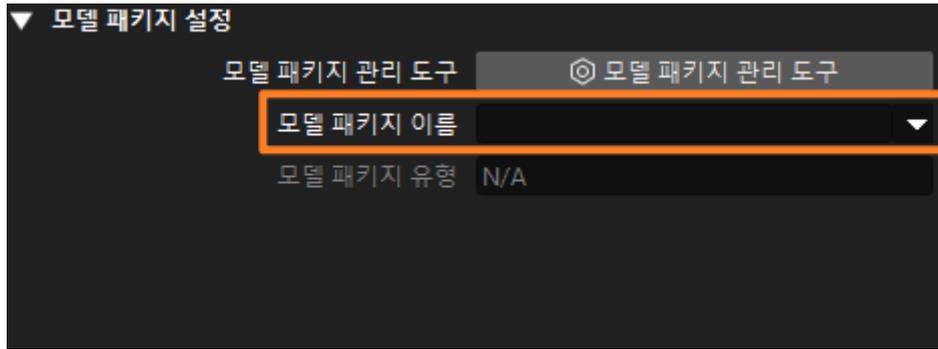
1. 딥 러닝 모델 패키지 관리 도구를 열고 인터페이스의 왼쪽 상단 모서리에 있는 [ **모델 패키지 도입하기** ]를 클릭하십시오.
2. 팝업한 파일 선택창에서 도입할 딥 러닝 모델 패키지를 선택한 후 [ **열기** ]를 클릭하면 이 딥 러닝 모델 패키지가 딥 러닝 모델 관리 도구 리스트에 나타납니다. 이로써 딥 러닝 모델 패키지의 도입이 완료됩니다.



딥 러닝 모델 패키지를 도입할 때 GPU 드라이버의 최소 버전은 472.50(딥 러닝 스텝의 실행 시간에 변동이 발생할 수 있으므로 버전 500 이상의 GPU 드라이버 사용을 권장하지 않음)이며, CPU의 최소 요구 사양은 인텔 6세대 코어입니다. 하드웨어 조건이 충족되지 않으면 딥 러닝 모델 패키지 도입하기가 실패할 수 있습니다.

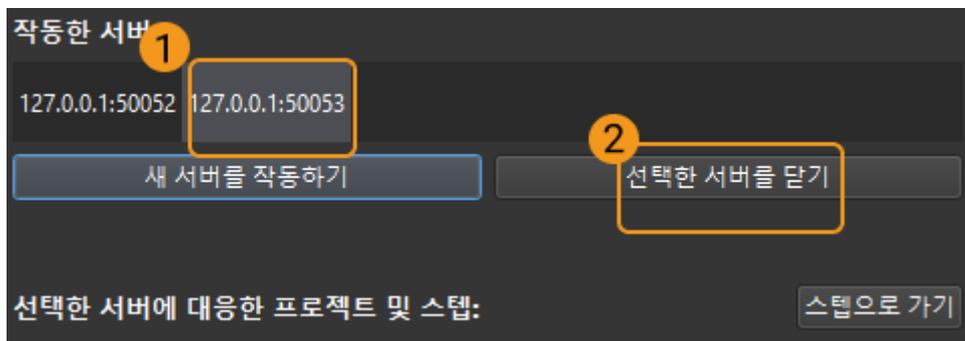
### 딥 러닝 스텝에서 딥 러닝 모델 패키지 선택

딥 러닝 모델 패키지를 도입한 후 "딥 러닝 모델 패키지 추론" 스텝에서 딥 러닝 모델 패키지를 선택하려면 그 파라미터에서 **모델 패키지 이름**의 드롭다운 목록에서 도입한 딥 러닝 모델 패키지를 선택할 수 있습니다.



### 딥 러닝 모델 패키지 제거하기

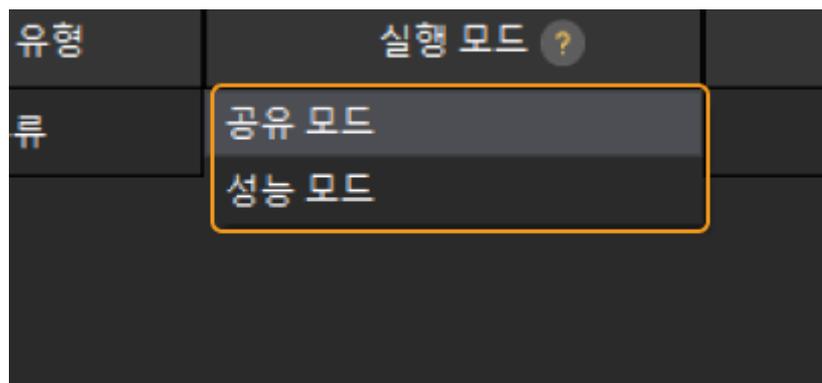
도입한 딥 러닝 모델 패키지를 제거하려면 먼저 딥 러닝 모델 패키지를 선택한 다음 딥 러닝 모델 패키지 관리 도구의 오른쪽 상단에 있는 [ 모델 패키지 제거하기 ] 버튼을 클릭해야 합니다.



**최적화 중 또는 사용 중**(해당 딥 러닝 모델 패키지를 사용하는 프로젝트가 실행될 때)인 모델 패키지는 제거될 수 없습니다.

### 딥 러닝 모델 패키지 추론의 실행 모드 전환

딥 러닝 모델 패키지 추론의 실행 모드를 전환하려면 딥 러닝 모델 패키지 관리 도구의 실행 모드 필드 아래에 있는 ▼ 버튼을 클릭한 후 **공유 모드** 또는 **성능 모드**를 선택합니다.



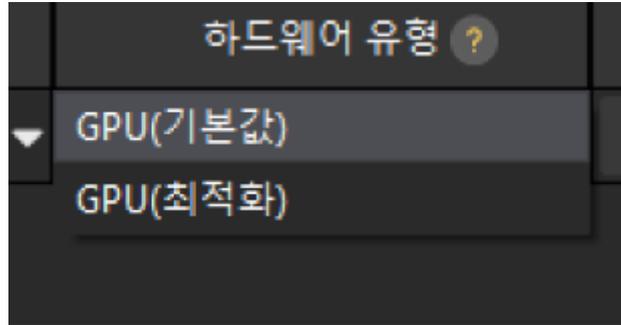
- 딥 러닝 모델 패키지의 최적화 상태가 **최적화 중** 또는 **사용 중**(해당 딥 러닝 모델 패키지를 사용하는 프로젝트가 실행될 때)인 경우 **실행 모드** 전환이 금지됩니다.
- 딥 러닝 모델 패키지의 실행 모드가 **공유 모드**인 경우 **딥 러닝 모델 패키지 추론** 스텝의 파라미터에서 **GPU ID**를 전환할 수 없습니다.

### 딥 러닝 모델 패키지 추론을 위한 하드웨어 유형 전환

딥 러닝 모델 패키지 추론을 위한 하드웨어 유형은 GPU(기본값), GPU(최적화), CPU로 전환할 수

있습니다.

딥 러닝 모델 패키지 관리 도구에서 **하드웨어 유형** 필드 아래의  버튼을 클릭하고 **GPU(기본값)**, **GPU(최적화)** 또는 **CPU**를 선택합니다.



- Mech-DLK 2.4.1 및 이상 버전에서 도출한 모델 패키지만 CPU와 GPU 간 전환을 지원합니다.
- 다음 도출한 모델 패키지는 **GPU(기본)** 하드웨어 유형의 사용을 지원하지 않습니다.
  - Mech-DLK 2.2.0에서 도출한 인스턴스 세그먼테이션 및 인스턴스 세그먼테이션 슈퍼 모델 패키지.
  - Mech-DLK에서 도출한 .dlkmt 파일만 포함하는 모델 패키지.
- 딥 러닝 모델 패키지의 최적화 상태가 **최적화 중** 또는 **사용 중**(해당 딥 러닝 모델 패키지를 사용하는 프로젝트가 실행될 때)인 경우 **하드웨어 유형** 전환이 금지됩니다.

#### 모델 효율성 구성

Mech-DLK 2.4.1 및 이상 버전에서 도출한 모델 패키지를 사용하는 경우 다음과 같이 모델 효율성을 구성할 수 있습니다.

1. 딥 러닝 모델 패키지를 도입합니다.
2. "모델 효율성" 아래의 [구성] 버튼을 클릭하면 모델 효율성 구성 창이 팝업하여 "배치 크기" 및 "정확도"를 구성할 수 있습니다.

모델 실행 효율성은 "배치 크기" 및 "정확도" 파라미터의 영향을 받습니다.

- 배치 크기: 모델 추론 중에 한 번에 신경망으로 전송되는 이미지의 수이며 범위는 1~128입니다. 이 값을 늘리면 모델 추론 속도가 빨라지지만 더 많은 그래픽 카드 메모리를 차지하게 됩니다. 이 값을 무리하게 설정하면 추론 속도가 느려집니다. 인스턴스 세그먼테이션 모델은 "배치 크기" 설정을 지원하지 않으며 "배치 크기"는 1이어야 합니다.
- 정확도("하드웨어 유형"이 "GPU 최적화"인 경우에만 활성화할 수 있음): FP32: 높은 모델 정확도, 느린 추론 속도; FP16: 낮은 모델 정확도, 빠른 추론 속도.



"배치 크기" 값은 실제로 신경망에 공급되는 이미지 수와 일치하는 것이 좋습니다.

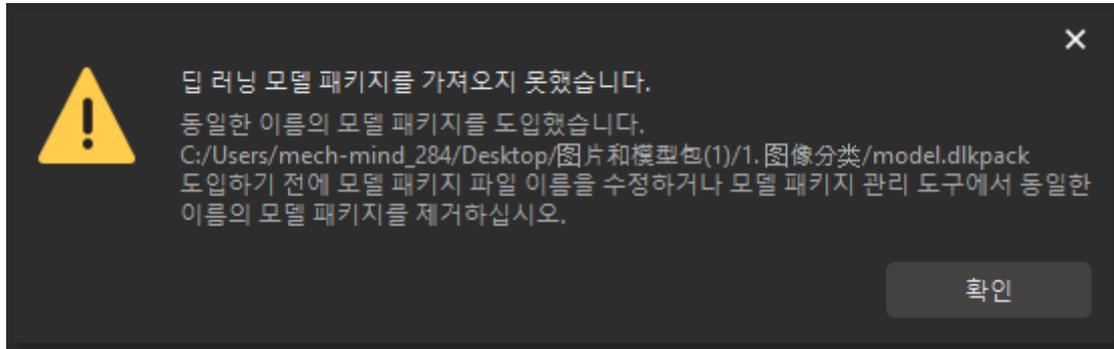
"배치 크기"가 실제로 신경망에 공급되는 이미지 수보다 훨씬 크게 설정되면 일부 리소스가 낭비되어 추론 속도가 느려집니다.

예: 이미지 수가 26개이고 "배치 크기"가 20으로 설정된 경우 추론이 두 번 수행되며 처음에는 20개의 이미지가 신경망에 전송되고 두 번째에는 6개의 이미지가 신경망에 전송됩니다. 두 번째 추론에서 "배치 크기" 설정이 신경망에 실제로 전송되는 이미지 수보다 훨씬 커서 일부 리소스가 낭비되고 추론 속도가 느려집니다. 따라서 리소스를 합리적으로 사용할 수 있도록 "배치 크기"를 합리적으로 설정하십시오.

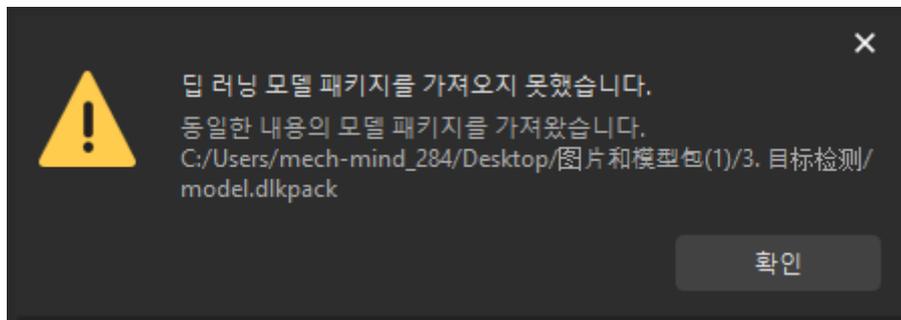
#### 주의사항

### 딥 러닝 모델 패키지 도입 실패

- 딥 러닝 모델 패키지를 이미 도입한 후 동일한 명칭의 딥 러닝 모델을 도입할 때 **딥 러닝 모델 패키지의 이름을 수정하거나 동일한 이름의 모델 패키지를 제거하여 도입해야 합니다.**



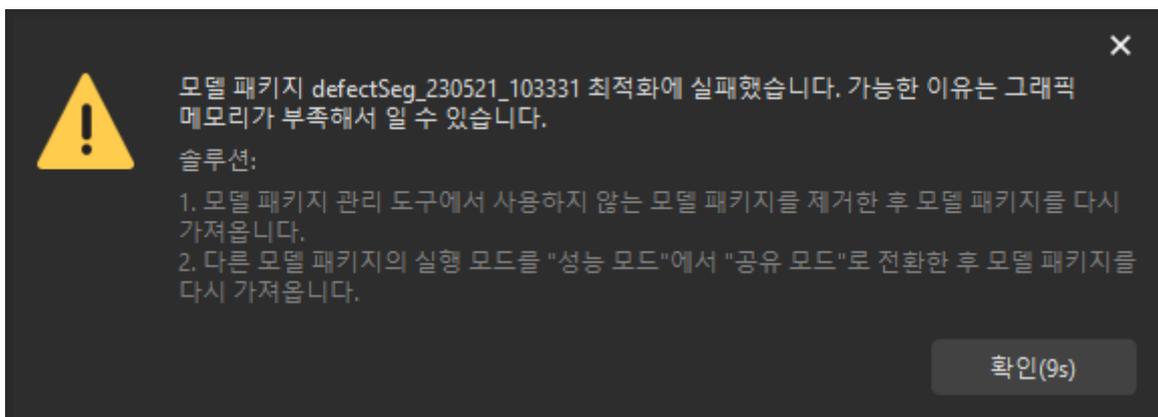
- 동일한 내용의 딥 러닝 모델 패키지를 반복적으로 도입하면 **딥 러닝 모델 패키지를 가져오지 못했습니다**라는 메시지가 표시됩니다.



- 소프트웨어 및 하드웨어 조건이 충족되지 않으면 딥 러닝 모델 패키지 도입하기가 실패할 수 있습니다. 최소 GPU 드라이버 버전 요구 사항은 472.50이고 CPU 최소 요구 사항은 Intel 6세대 코어입니다.

### 딥 러닝 모델 패키지 최적화 실패

그래픽 카드 메모리가 부족하면 아래 그림과 같이 딥 러닝 모델 패키지의 최적화가 실패할 수 있습니다. 프롬프트 창의 해결 방법에 따라 이 문제를 해결할 수 있습니다.



### 호환성 설명

- Mech-Vision 1.7.1 버전에서도 Mech-DLK 2.4.1 및 이상 버전에서 도출한 딥 러닝 모델 패키지를 사용할 수 있지만 다음의 호환성 문제에 주의해야 합니다. Mech-Vision 1.7.2 및 이상 버전과 Mech-DLK 2.4.1 및 이상 버전에서 도출한 딥 러닝 모델 패키지와 결합해서 사용하는 것을 권장합니다.

- Mech-Vision은 캐스케이딩 모델 패키지를 사용할 수 없습니다.
- 모델 효율성을 조정할 수 없습니다.
- 이미지 분류 효과가 나빠질 수 있습니다.
- 모델 패키지를 CPU 장치에서 사용할 수 없습니다.
- Mech-Vision 1.7.2에서 Mech-Vision 1.7.1 최적화된 모델 패키지를 사용하는 경우 "딥 러닝 모델 패키지 추론" 스텝에서 모델 패키지의 첫 번째 실행 시간이 더 오래 걸릴 수 있습니다.
- Mech-DLK 2.4.1 버전을 사용하여 물체 검출 모델 패키지를 도출할 때 "추론 시 최대 인스턴스 수"를 1로 설정하고 딥 러닝 모델 패키지 관리 도구의 하드웨어 유형이 CPU인 경우, 모델 패키지의 추론 속도가 매우 느려질 것입니다. "추론 시 최대 인스턴스 수"의 값은 1보다 큰 것이 좋습니다.

#### 4.6.8. 파라미터 레시피

이 부분에서는 **파라미터 레시피**와 해당 응용 프로그램 시나리오 및 구성을 소개합니다.

##### 소개

파라미터 레시피는 프로젝트가 다양한 상황에서 실행될 때 조정해야 하는 파라미터 설정 모음입니다. 다양한 상황에 대해 서로 다른 파라미터 레시피를 구성함으로써 반복적인 프로젝트 구축을 피하고 프로젝트 적응성이 향상됨으로써 생산 효율성이 향상됩니다.

Mech-Vision은 파라미터 레시피 기능을 지원하기 위해 **파라미터 레시피 편집기** 도구를 제공합니다.

##### 응용 시나리오

파라미터 레시피는 프로젝트가 다양한 상황에서 차별화된 파라미터 설정이 필요한 시나리오에서 사용됩니다. 대표적인 적용 시나리오는 다음과 같습니다.

- 동일한 Mech-Vision 프로젝트를 사용하여 다양한 공작물 또는 물체를 처리합니다. 이런 시나리오에서 사용자는 다양한 파라미터 레시피를 구성하여 공작물 또는 물체에 대한 파라미터 설정을 동적으로 전환할 수 있습니다.
- 동일한 Mech-Vision 프로젝트를 사용하여 다양한 빈에 있는 공작물을 처리합니다. 이런 시나리오에서 사용자는 다양한 파라미터 레시피를 구성하여 다양한 상자에 따라 파라미터 설정을 동적으로 전환할 수 있습니다.
- 다양한 조명 조건(예: 낮과 밤)에서 Mech-Vision 프로젝트에는 다양한 카메라 구성 파라미터 그룹 및 기타 파라미터 설정이 필요합니다. 이런 시나리오에서 사용자는 다양한 파라미터 레시피를 구성하여 다양한 조명 조건에 대해 카메라 구성 파라미터 그룹 및 기타 파라미터 설정을 동적으로 전환할 수 있습니다.
- Eye In Hand 카메라 설치 모드에서 Mech-Vision 프로젝트는 다른 영역에서 사진을 캡처해야 합니다. 이런 시나리오에서 사용자는 다른 파라미터 레시피를 구성하여 다른 촬영 영역에 대한 카메라 구성 파라미터 그룹 및 기타 파라미터 설정을 동적으로 전환할 수 있습니다.

마스터 컨트롤 통신을 사용하는 경우 사용자는 Mech-Vision 프로젝트를 해당 파라미터 레시피로 수동으로 전환해야 합니다. 또한 표준 인터페이스 또는 Adapter 통신을 사용할 때 사용자는 로봇 측 프로그램에서 전송한 레시피 전환 명령이 필요하고 Mech-Vision이 수신된 명령에 따라 파라미터 레시피를 자동으로 전환합니다.

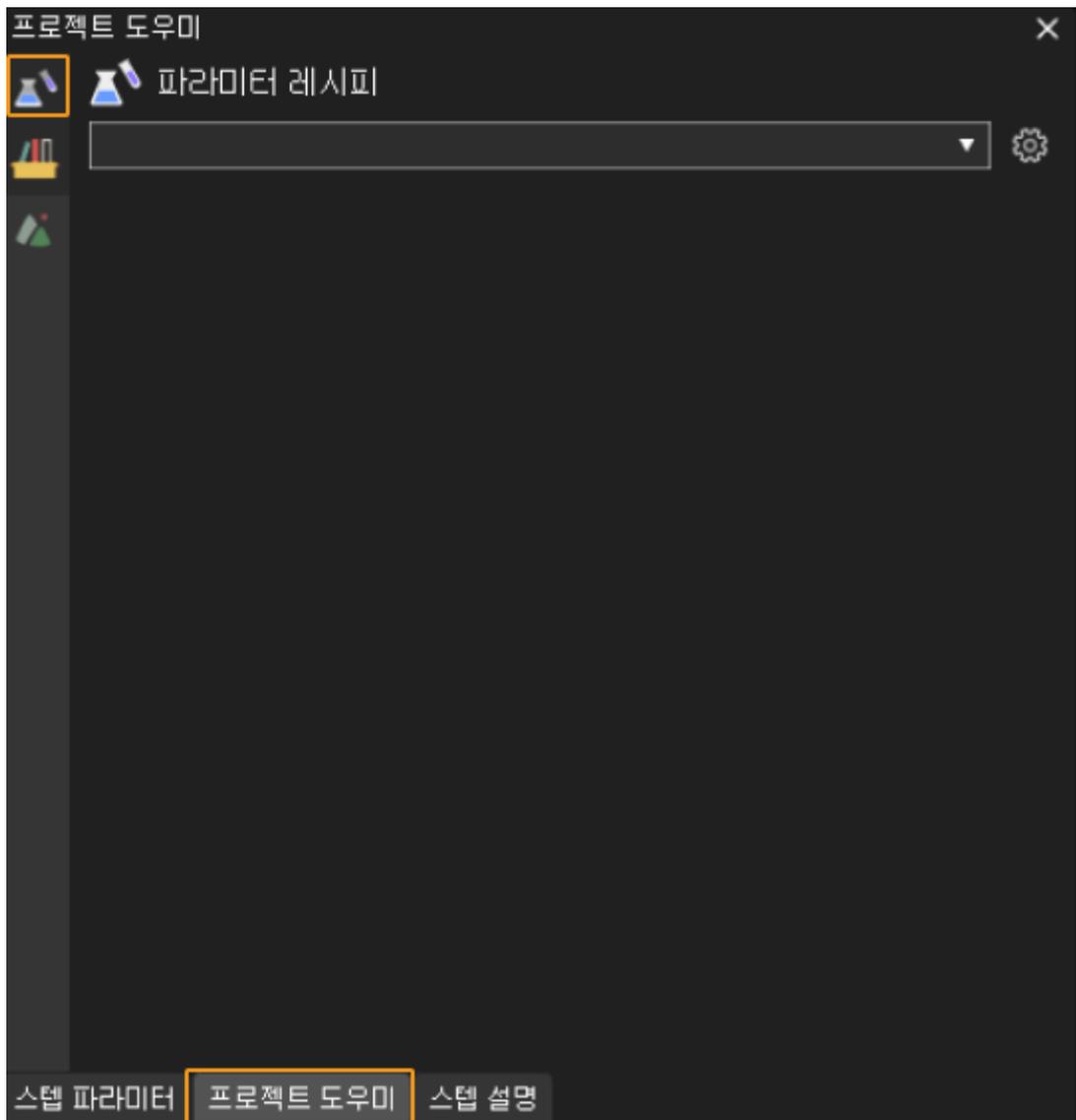
#### 파라미터 레시피 구성

##### 파라미터 레시피 추가

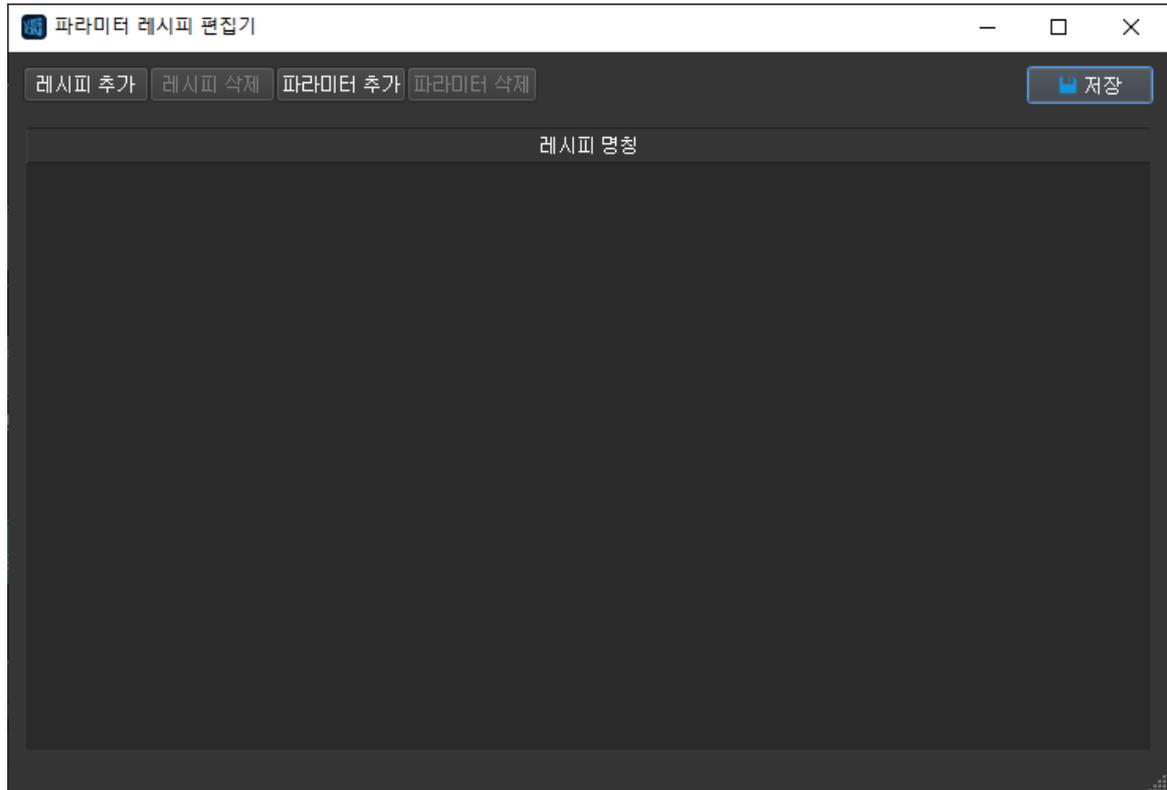
파라미터 레시피를 추가하려면 다음 단계를 수행하십시오.

1. 프로젝트를 연 후 프로젝트 구성 영역에서 **프로젝트 도우미** 탭을 클릭한 다음 을 클릭하여 파라미터

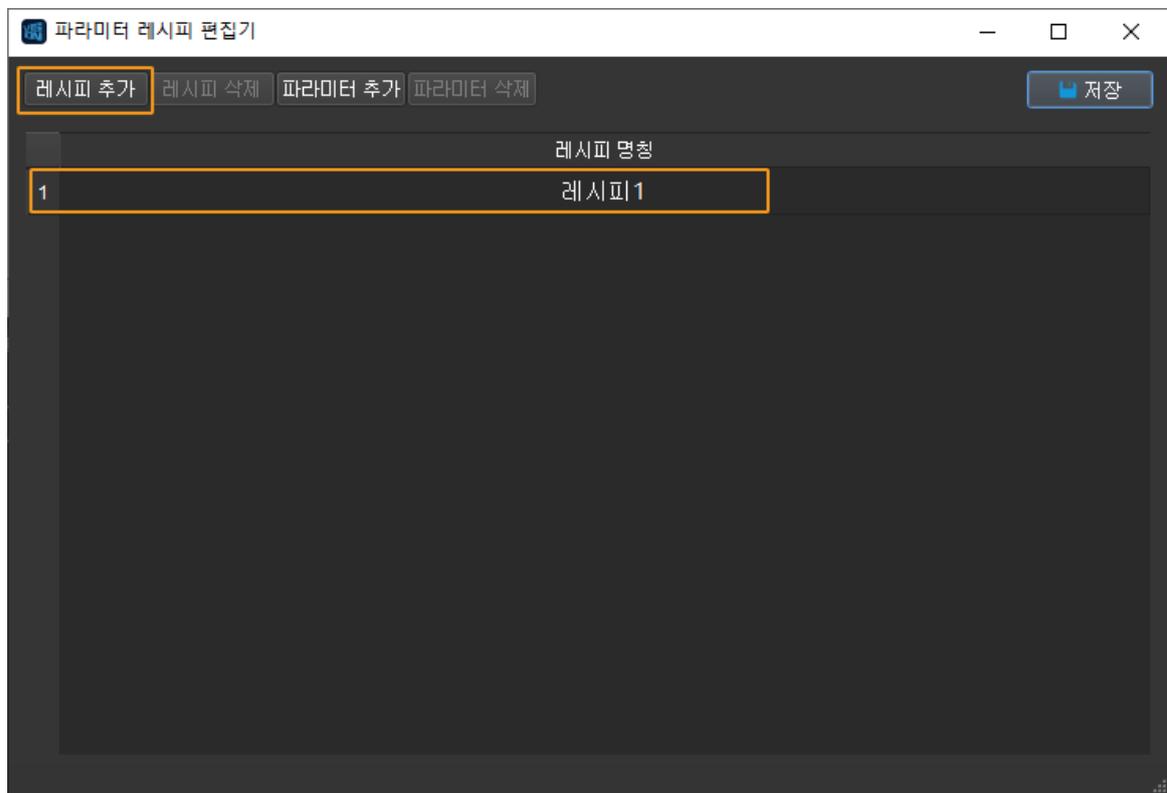
레시피 인터페이스로 들어갑니다.



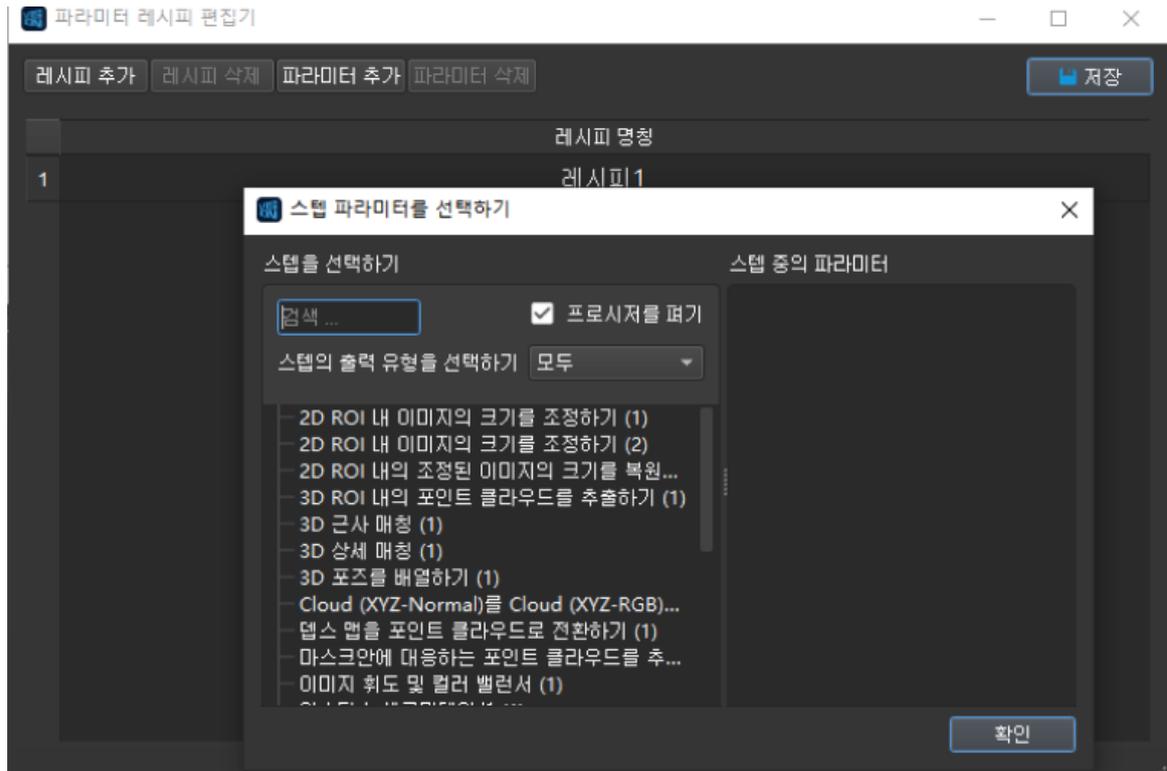
2. 파라미터 레시피 인터페이스에서  을 클릭하십시오.
3. 팝업한 **파라미터 레시피 편집기** 다이얼로그 박스에서 [레시피 추가] 버튼을 클릭하십시오.



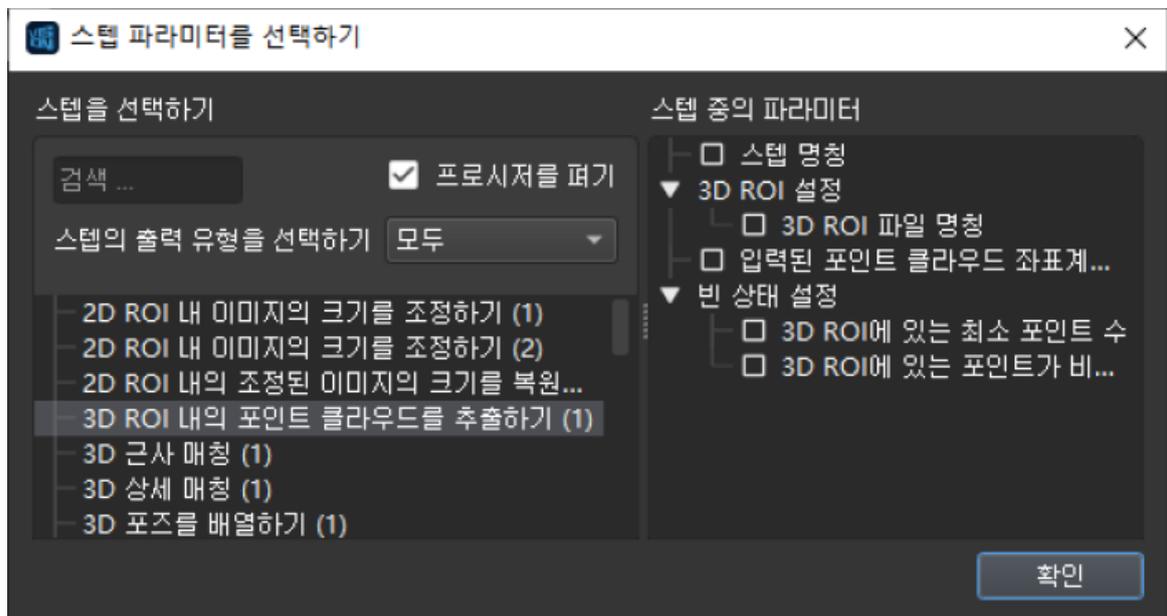
4. 레시피 리스트에서 새로 추가된 레시피 항목을 선택한 후 레시피 이름을 입력하십시오.



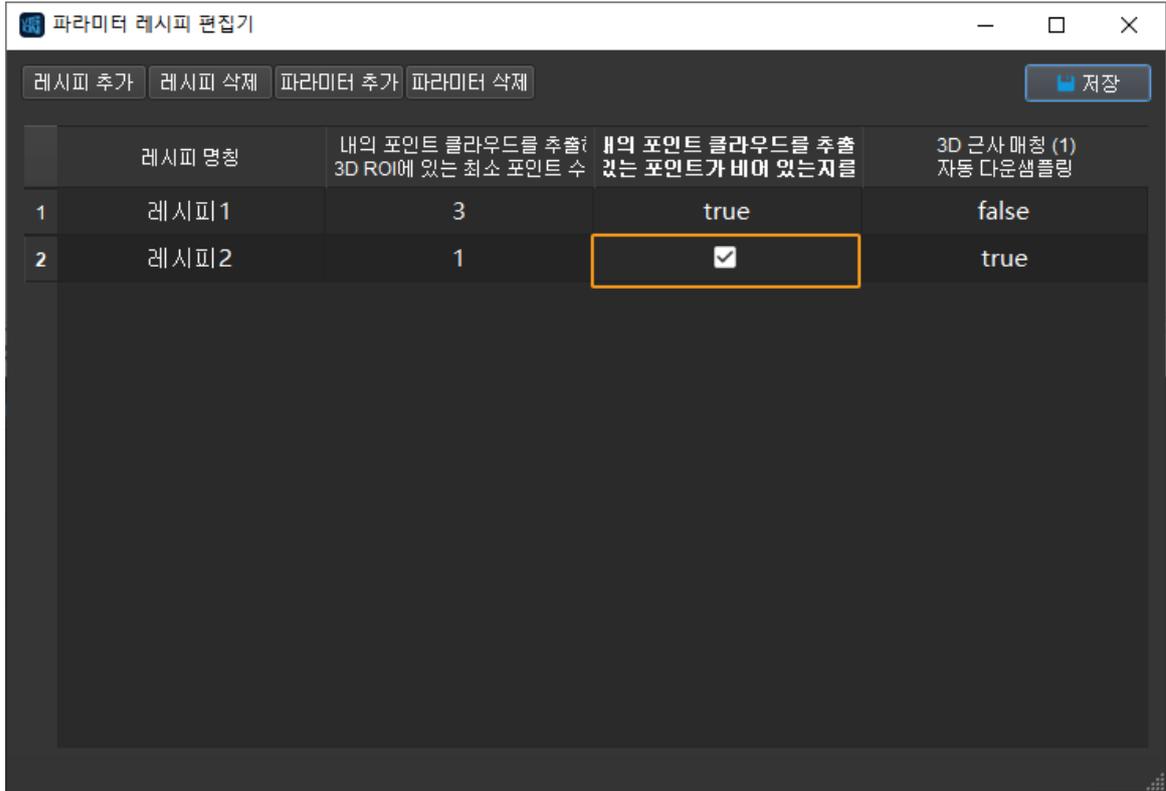
5. 다이얼로그 박스의 빈 영역을 클릭 한 다음 새로 추가 된 레시피를 선택하고 [파라미터 추가] 버튼을 클릭하십시오. 스텝 파라미터를 선택하기 다이얼로그 박스가 나타납니다.



6. **스텝 파라미터를 선택하기** 다이얼로그 박스의 왼쪽 패널에서 추가할 스텝을 선택하고 오른쪽 패널에서 레시피에 추가할 스텝 파라미터를 선택합니다.

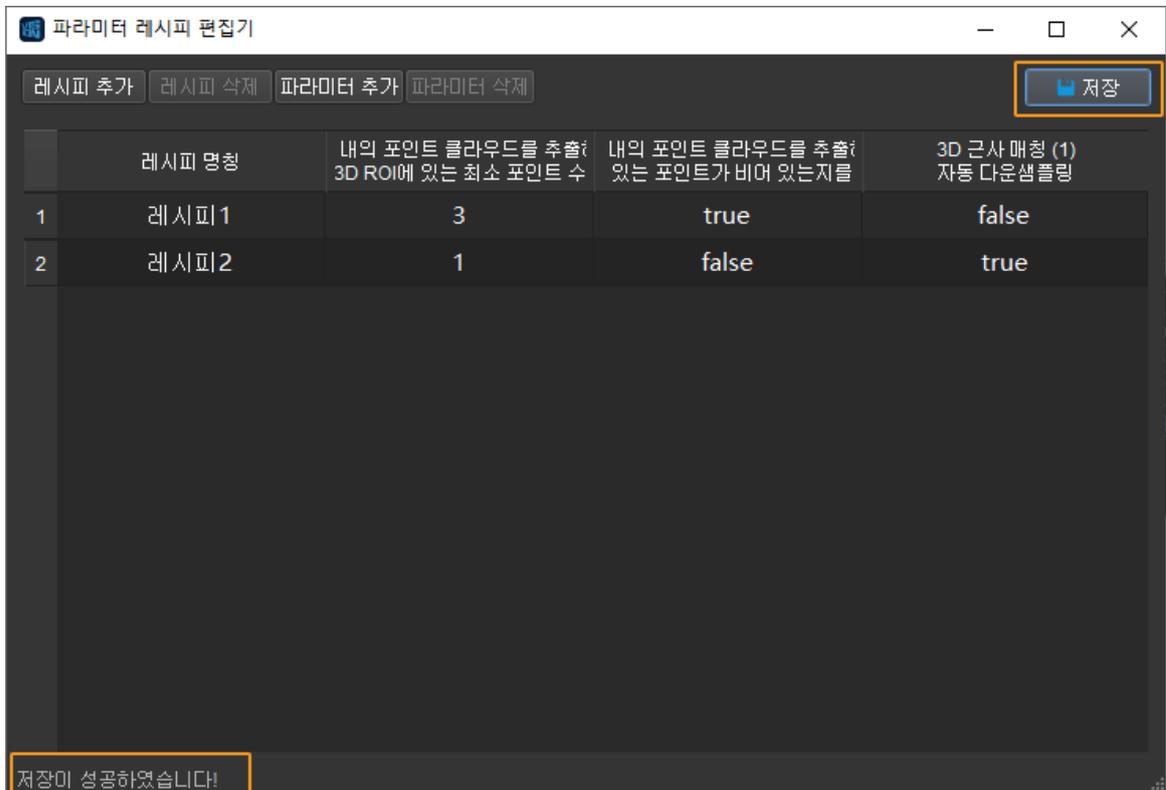


7. 모든 필수 스텝 파라미터가 선택될 때까지 위의 스텝을 반복한 다음 [ **확인** ] 버튼을 클릭하십시오.
8. 레시피 리스트에서 레시피의 각 파라미터 설정을 더블 클릭하여 수정합니다.



파라미터 순서 문제로 인해 특정 파라미터의 설정이 실패하지 않도록 레시피에 설정된 파라미터의 순서에 주의하십시오.

9. 팝업한 **파라미터 레시피 편집기** 다이얼로그 박스에서 [ **저장** ] 버튼을 클릭하십시오. 저장에 성공하면 다이얼로그 박스의 왼쪽 하단에 "저장 성공"이라는 메시지가 표시됩니다.



프로젝트에서 여러 파라미터 레시피를 사용하는 경우 위의 스텝을 반복하여 여러 파라미터 레시피를 추가합니다. 새로 추가된 파라미터 레시피는 현재 프로젝트의 스텝 파라미터 설정을 자동으로 상속합니다.

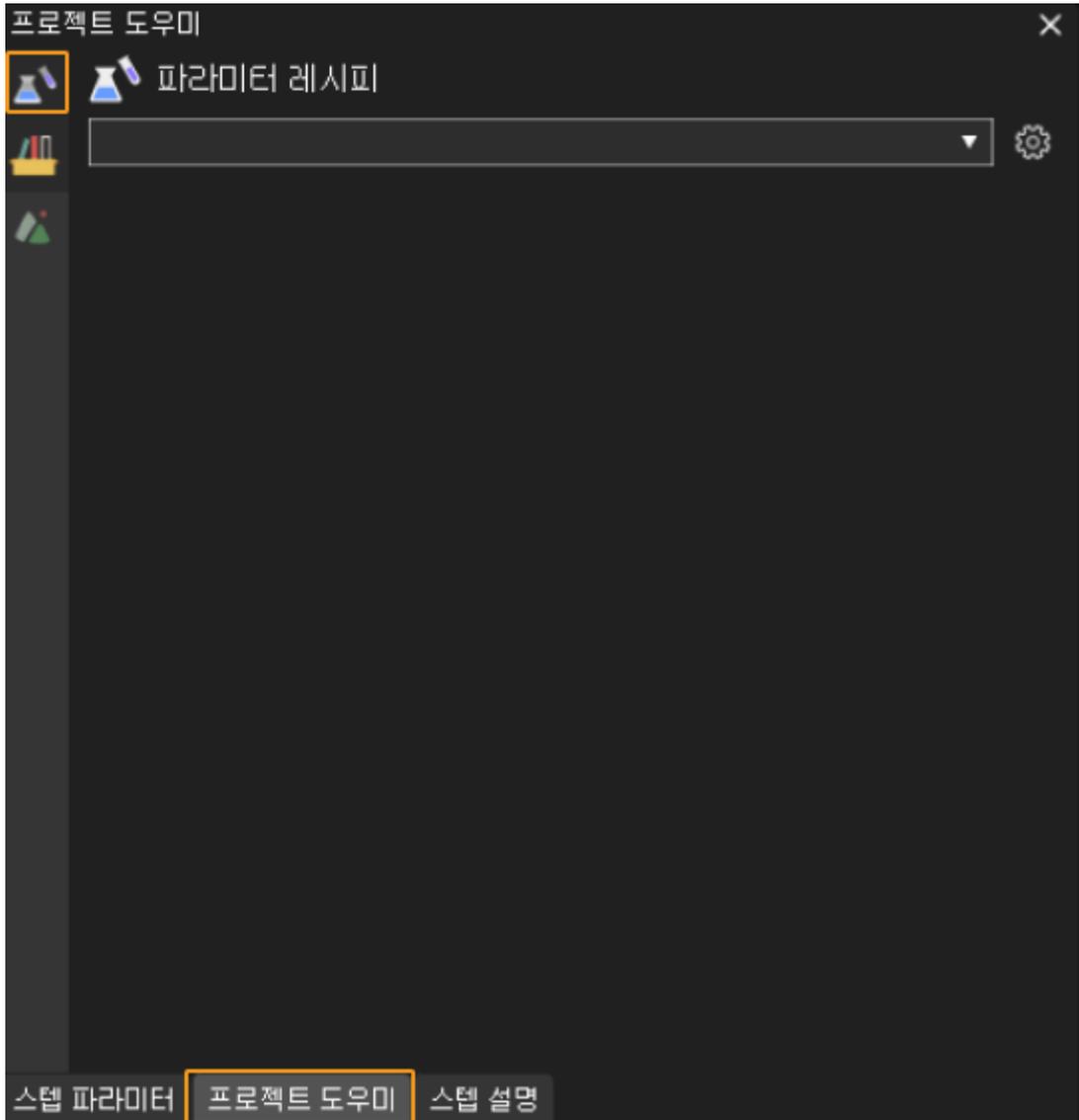
### 파라미터 레시피 적용 또는 전환



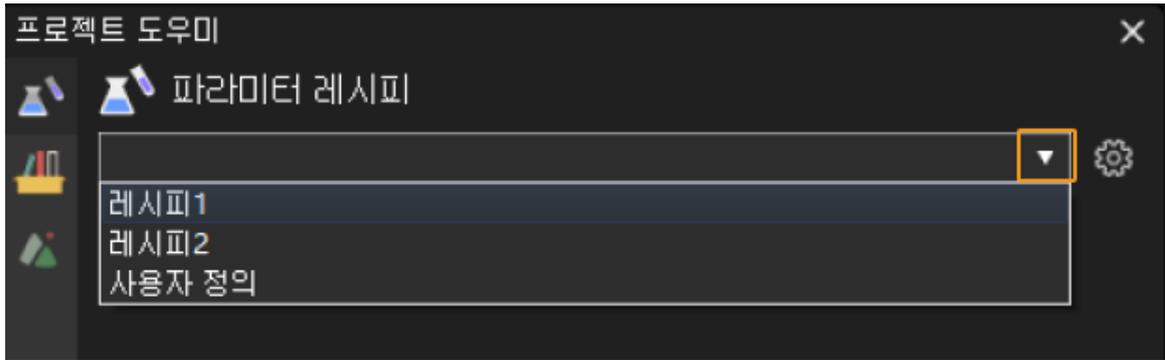
표준 인터페이스 또는 Adapter 통신을 사용할 때 사용자는 로봇 측 프로그램에서 파라미터 레시피를 전환하는 명령을 보내야 합니다. 파라미터 레시피의 자동 전환을 실현합니다.

파라미터 레시피를 수동으로 응용하거나 전환하려면 다음 단계를 수행하십시오.

1. 프로젝트 구성 영역에서 **프로젝트 도우미** 탭을 클릭한 다음  을 클릭하여 파라미터 레시피 인터페이스로 들어갑니다.



2. 파라미터 레시피 인터페이스의 드롭다운 목록에서 공작물에 해당하는 파라미터 레시피를 선택합니다. 프로젝트를 실행하면 설정된 파라미터 레시피가 적용됩니다.



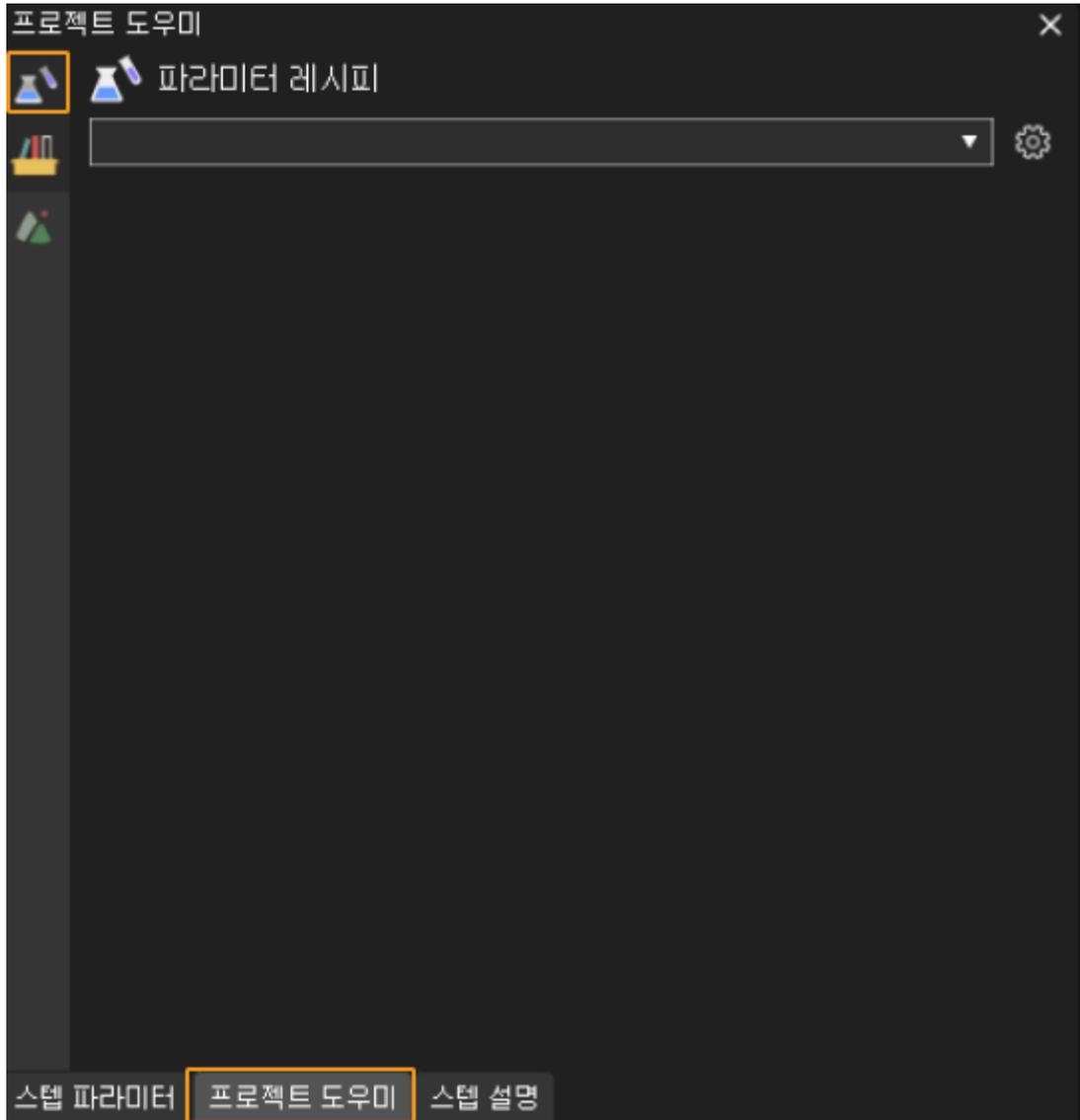
파라미터 레시피를 적용하고 프로젝트를 실행한 후 레시피에서 파라미터 설정을 수정해야 하는 경우 [파라미터 레시피를 위해 특정 스텝의 파라미터 설정 사항을 수정하기](#) 내용을 참조하여 수정하십시오. 파라미터 설정이 프로젝트에서 직접 수정되면 프로젝트가 다시 실행될 때 효력이 발생한 레시피가 자동으로 "사용자 지정" 레시피로 전환됩니다.

### 파라미터 레시피 번호 보기

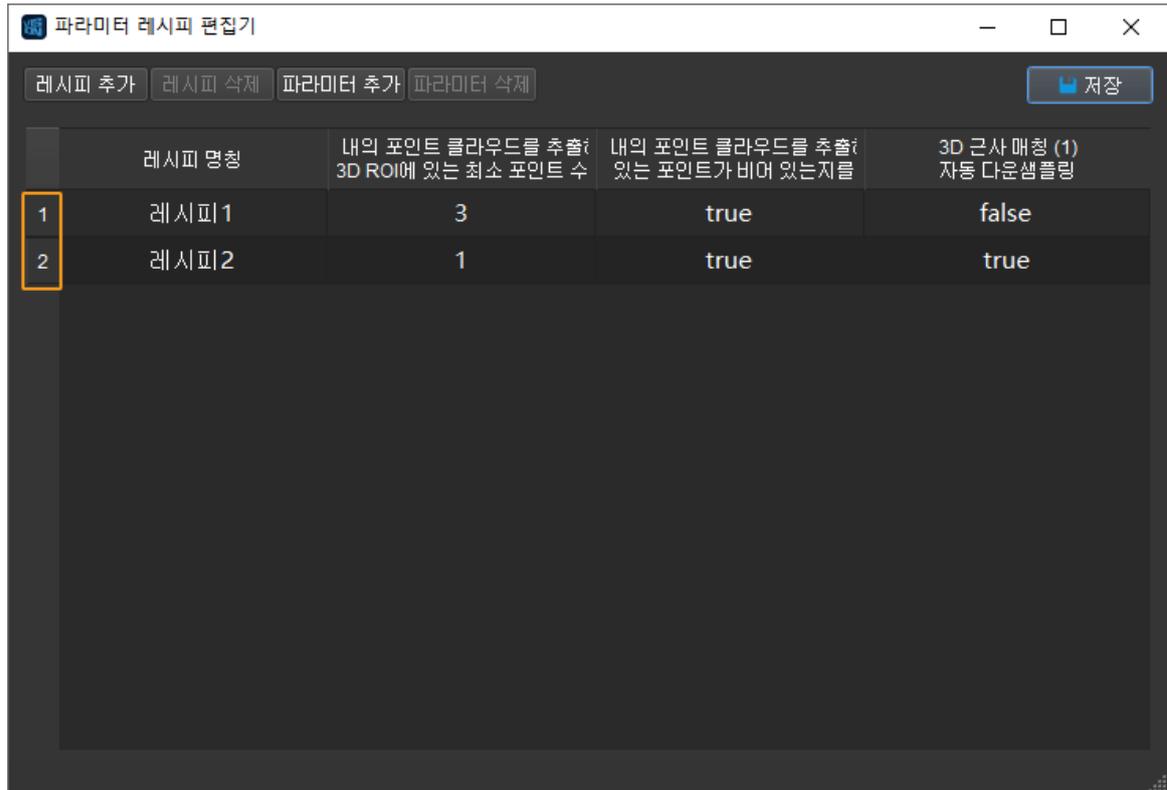
표준 인터페이스 또는 Adapter 통신을 사용할 때 사용자는 로봇 측 프로그램에서 전송한 레시피 전환 명령에서 이를 대신 파라미터 레시피 번호를 지정해야 합니다.

파라미터 레시피의 번호를 확인하려면 다음 단계를 수행하십시오.

1. 프로젝트를 연 후 프로젝트 구성 영역에서 **프로젝트 도우미** 탭을 클릭한 다음  을 클릭하여 파라미터 레시피 인터페이스로 들어갑니다.



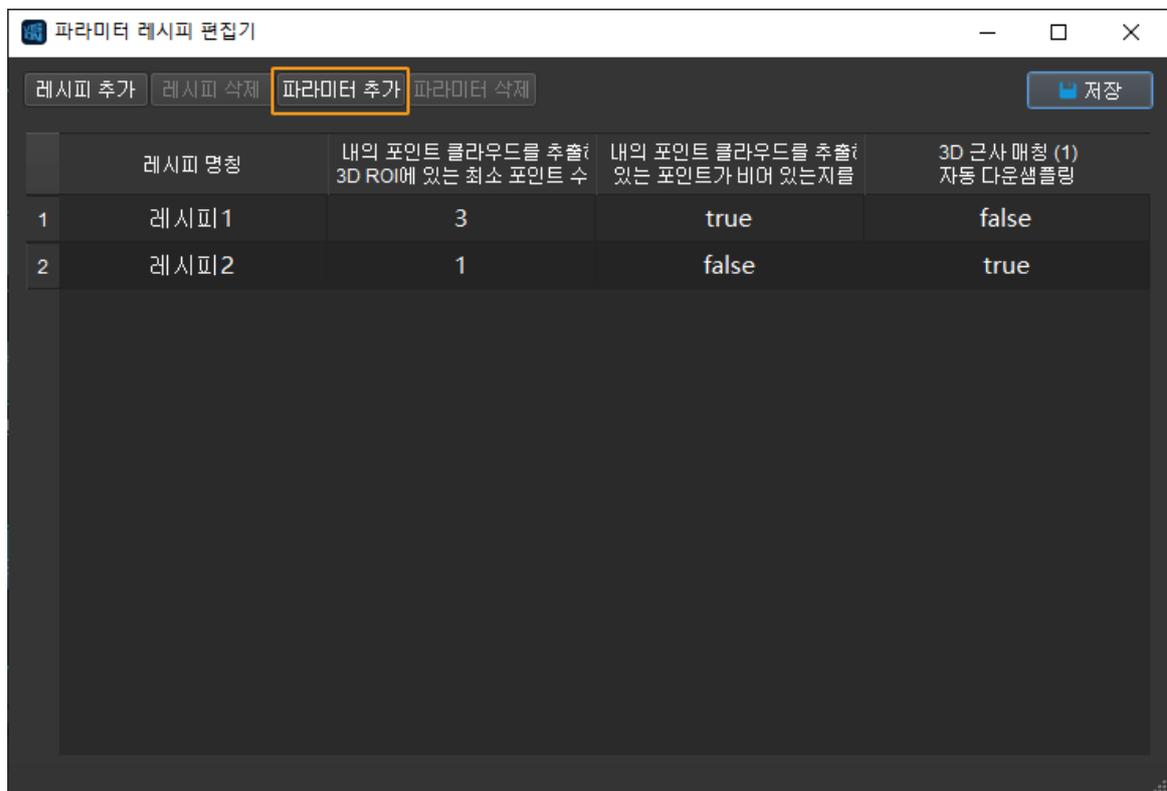
2. 파라미터 레시피 인터페이스에서  을 클릭하십시오.
3. 레시피 리스트에서 필요한 파라미터 레시피를 찾고 테이블 항목의 맨 왼쪽에 있는 레시피 번호를 확인합니다.



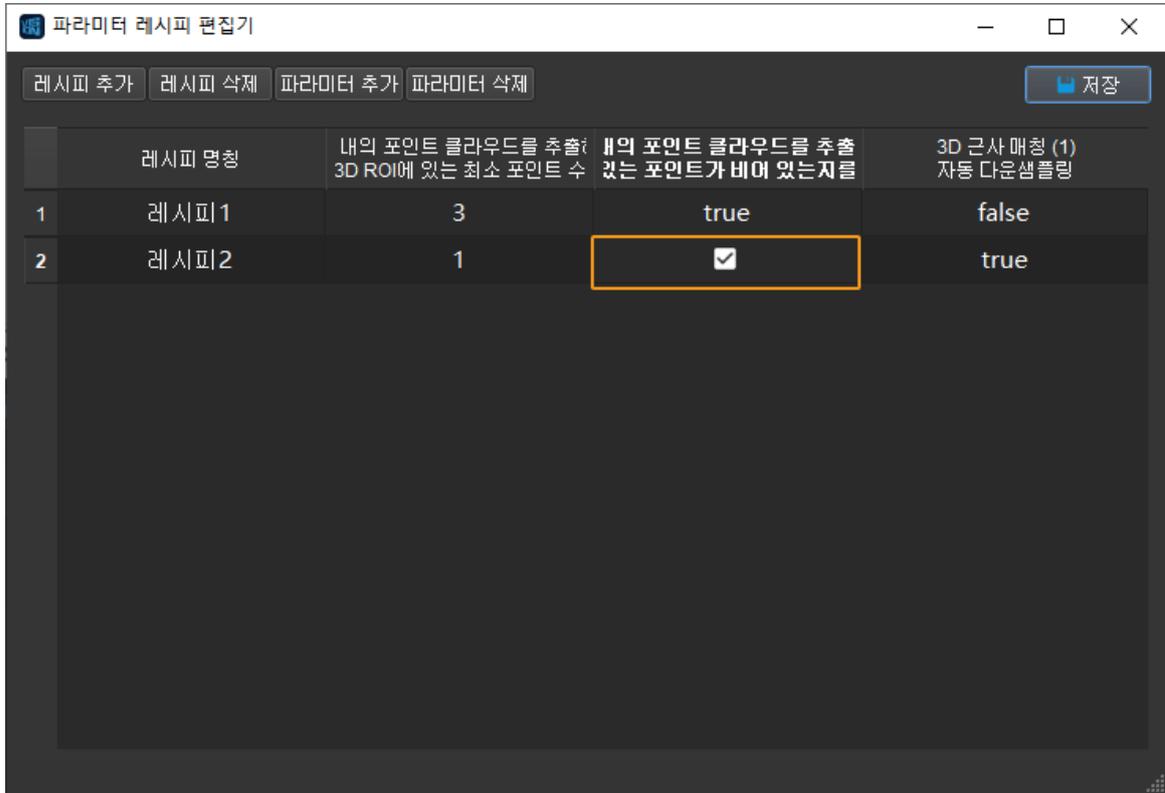
#### 파라미터 레시피에 스텝 파라미터 추가

파라미터 레시피에서 스텝 파라미터를 수정하려면 다음 단계를 수행하십시오.

1. **파라미터 레시피 편집기** 다이얼로그 박스에서 파라미터 레시피 항목을 선택하고 [ **파라미터 추가** ] 버튼을 클릭합니다.



2. **스텝 파라미터 선택** 다이얼로그 박스에서 새 스텝 파라미터를 파라미터 레시피에 계속 추가한 다음 [ **확인** ] 버튼을 클릭합니다.
3. 레시피 리스트에서 레시피의 새 파라미터 설정을 두 번 클릭하고 수정합니다.



4. 팝업한 **파라미터 레시피 편집기** 다이얼로그 박스에서 [ **저장** ] 버튼을 클릭하십시오.

#### 파라미터 레시피에서 스텝 파라미터 삭제

파라미터 레시피에서 스텝 파라미터를 삭제하려면 다음 단계를 수행하십시오.

1. **파라미터 레시피 편집기** 다이얼로그 박스의 파라미터 레시피 테이블 항목에서 삭제할 스텝 파라미터를 선택하고 [ **파라미터 삭제** ] 버튼을 클릭합니다.

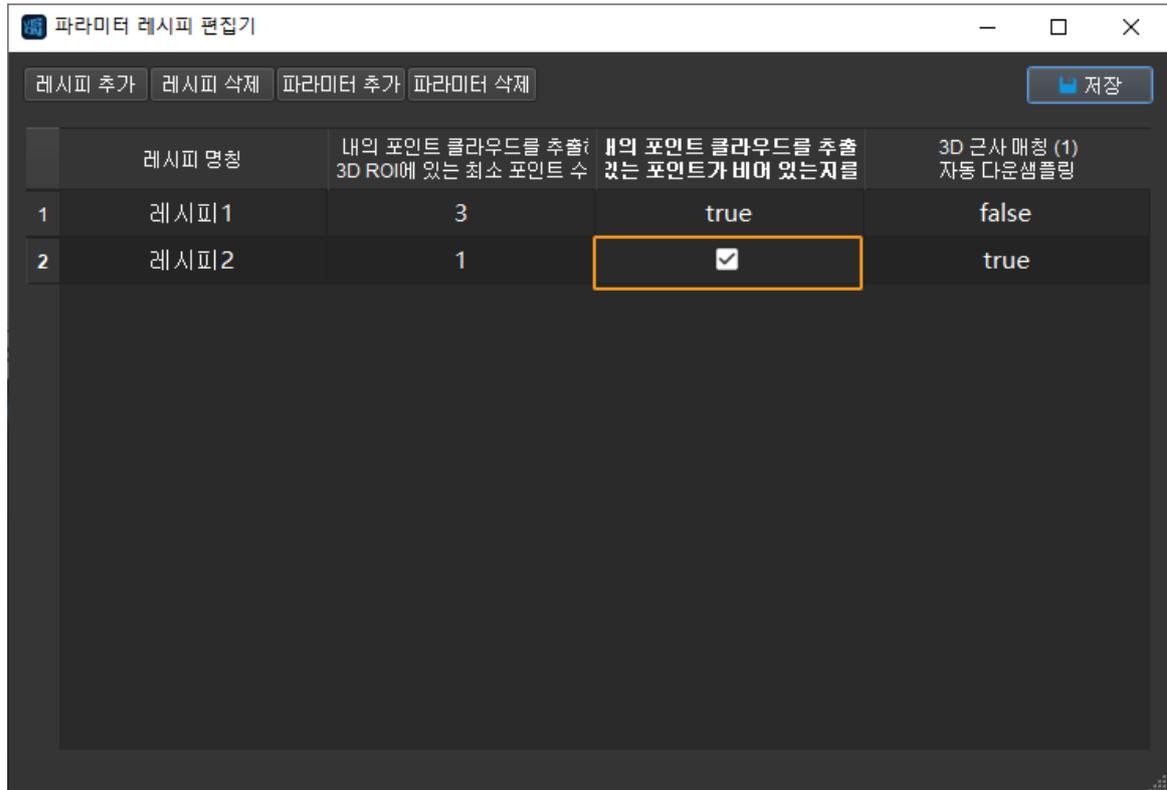


2. [ 저장 ] 버튼을 클릭하십시오.

**파라미터 레시피를 위해 특정 스텝의 파라미터 설정 사항을 수정하기**

파라미터 레시피를 위해 지정된 스텝의 파라미터 설정을 수정하려면 다음 단계를 수행하십시오.

1. 레시피 리스트에서 수정하고자 하는 파라미터 설정을 더블 클릭하여 수정합니다.



2. [ 저장 ] 버튼을 클릭하십시오.

#### 현재 프로젝트 구성에 따라 레시피의 스텝 파라미터 설정을 업데이트하기

파라미터 레시피 편집기는 현재 프로젝트 구성에 따라 레시피의 특정 또는 모든 스텝의 파라미터 설정 업데이트를 지원합니다.

현재 프로젝트 구성에 따라 레시피 중 특정 스텝의 파라미터 설정을 업데이트하려면 다음 단계를 수행하십시오.

1. **파라미터 레시피 편집기** 다이얼로그 박스에서 지정된 파라미터 레시피 테이블 항목에서 업데이트할 스텝 파라미터를 선택합니다.

 **Ctrl** 키를 누른 상태에서 여러 스텝 파라미터를 선택합니다.

2. 툴 바에서 **파라미터 업데이트** 버튼을 클릭합니다.

	从相机获取图像 (1) 相机标定参数组	从相机获取图像 (1) 使用深度图去背景	3D工件识别 (1) 选择工件	路径规划 (1) 工作流程配置	路径规划 (1) 输入点云类型
1	WAM3021CA30008...	false	model_edge	0. 默认流程	CloudXYZ
2	WAM3021CA30008...	false	model_edge	0. 默认流程	CloudXYZRGB
3	WAM3021CA30008...	false	model_edge	0. 默认流程	CloudXYZRGB

현재 프로젝트 구성에 따라 레시피 중 모든 스텝의 파라미터 설정을 업데이트하려면 다음 단계를 수행하십시오.

1. **파라미터 레시피 편집기** 다이얼로그 박스에서 지정된 파라미터 레시피 테이블 항목의 파라미터 레시피 번호를 클릭하여 현재 항목을 선택합니다.
2. 톨 바에서 **파라미터 업데이트** 버튼을 클릭합니다.

#### 두 파라미터 레시피에 대한 스텝 파라미터 설정 동기화

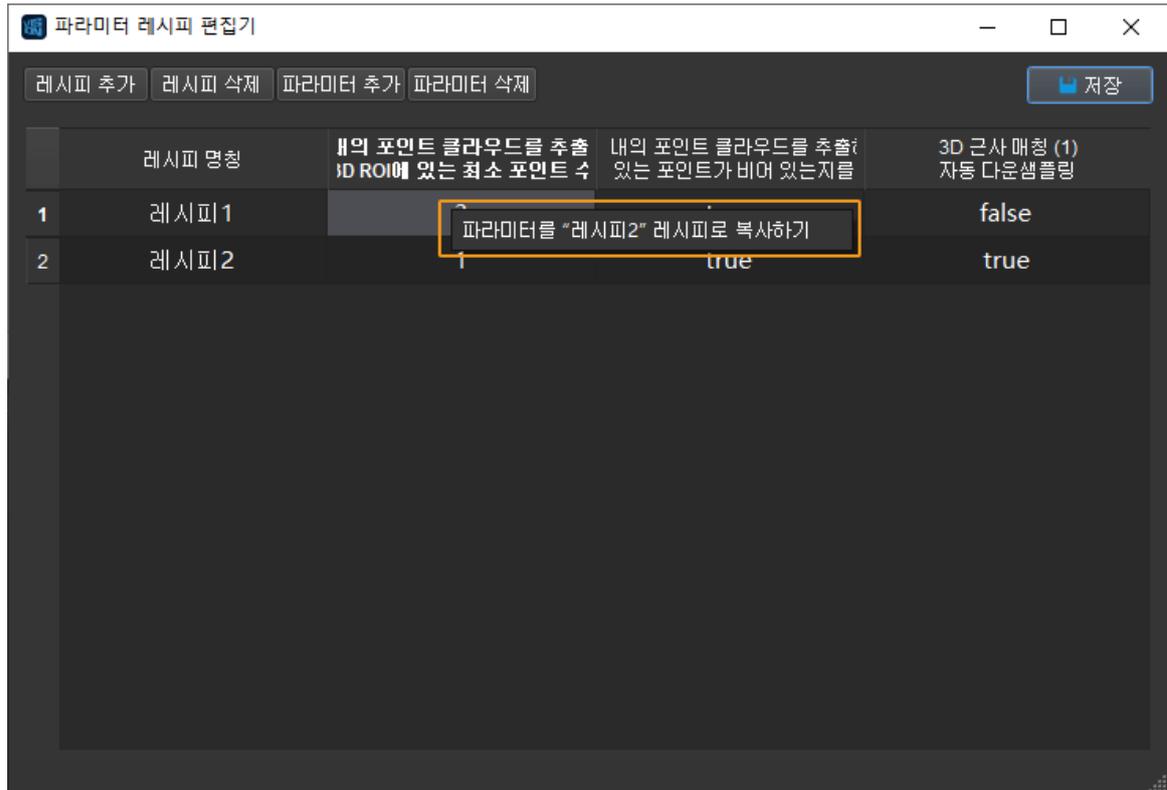
파라미터 레시피 편집기는 지정된 파라미터 또는 한 파라미터 레시피의 모든 파라미터 설정을 다른 파라미터 레시피에 동기화하는 것을 지원합니다.

한 파라미터 레시피의 지정된 파라미터 설정을 다른 파라미터 레시피와 동기화하려면 다음 단계를 수행하십시오.

1. **파라미터 레시피 편집기** 다이얼로그 박스에서 지정된 파라미터 레시피 테이블 항목에서 동기화할 스텝 파라미터를 선택합니다.

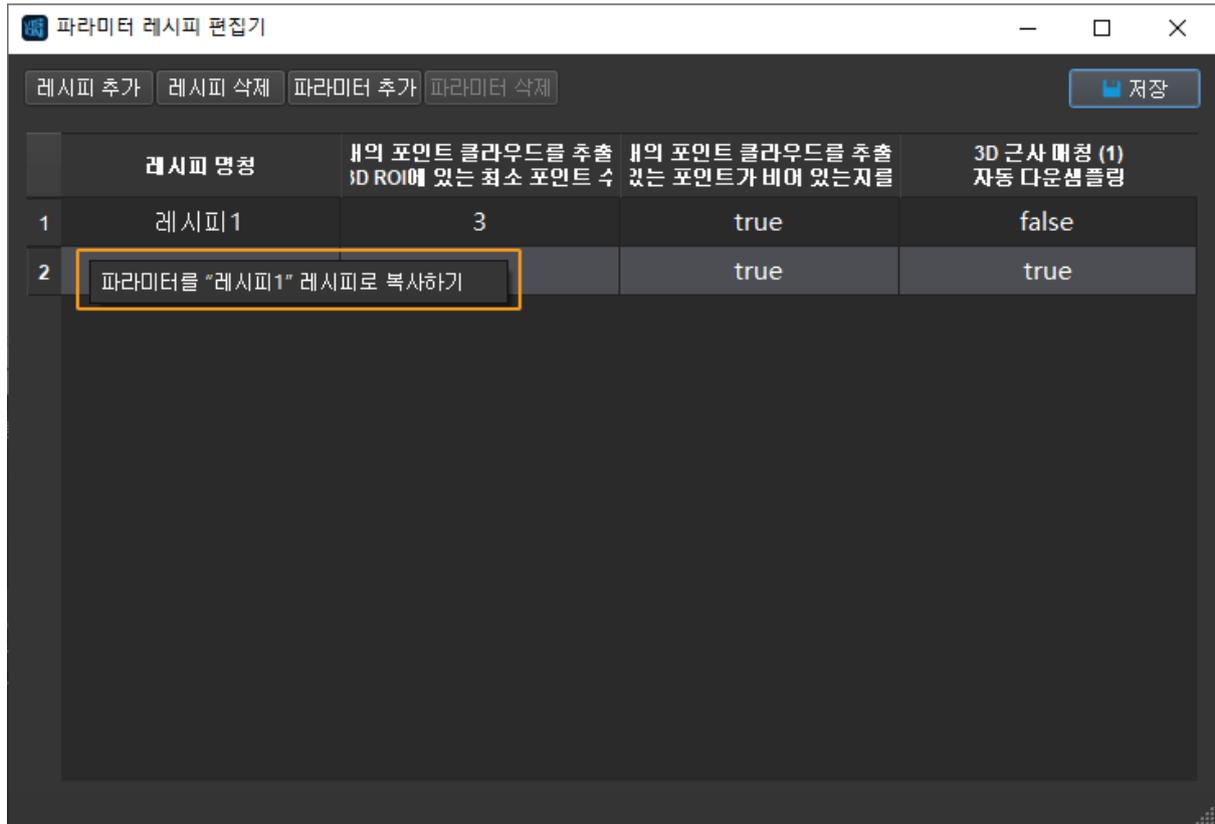
 **Ctrl** 키를 누른 상태에서 여러 스텝 파라미터를 선택합니다.

2. 마우스 오른쪽 버튼을 클릭하고 **파라미터를 "xx" 레시피로 복사하기**를 선택합니다.



한 파라미터 레시피의 모든 파라미터 설정을 다른 파라미터 레시피와 동기화하려면 다음 단계를 수행하십시오.

1. **파라미터 레시피 편집기** 다이얼로그 박스에서 지정된 파라미터 레시피 테이블 항목의 파라미터 레시피 번호를 클릭하여 현재 항목을 선택합니다.
2. 레시피 이름을 마우스 오른쪽 버튼으로 클릭하고 **파라미터를 "xx" 레시피로 복사하기**를 선택합니다.

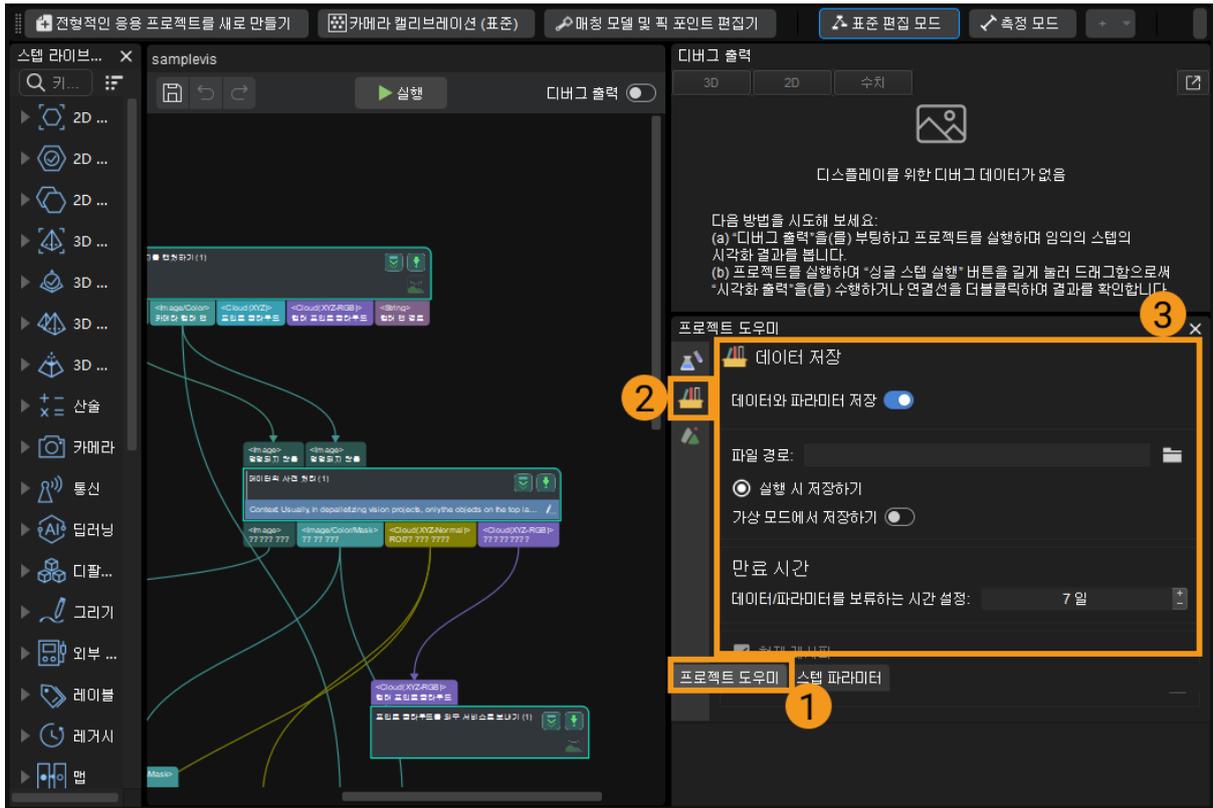


#### 4.6.9. 데이터 저장

##### 기능 설명

프로젝트 작업 현장에서 문제 발생 시 당시의 상황을 재현될 수 있도록 하고, 현장을 떠나기 전에 충분한 회귀 분석을 수행할 수 있도록 하기 위해 Mech-Vision은 데이터 저장 기능을 제공합니다.

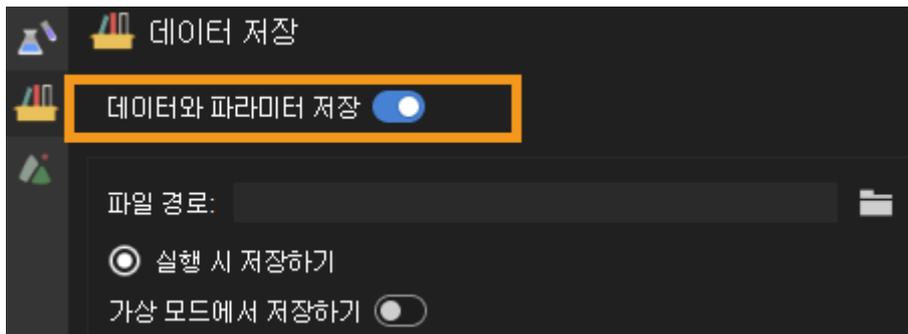
프로젝트를 열어 Mech-Vision 프로젝트 편집 구역 밑에서 [프로젝트 도우미]를 클릭하고 를 클릭하여 바로 데이터 저장 화면을 열 수 있습니다. 아래 그림과 같습니다.



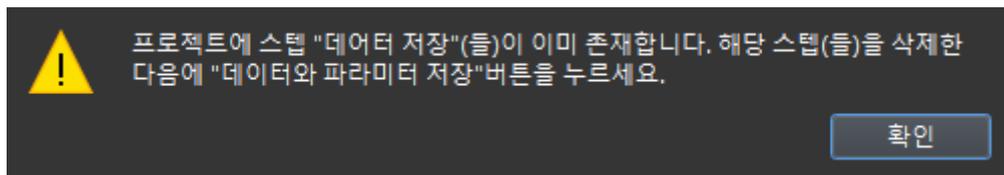
Mech-Vision 프로젝트 편집 구역에서 **프로젝트 도우미** 옵션을 찾을 수 없는 경우 Mech-Vision 메뉴 바의 **뷰**에서 **프로젝트 도우미** 옵션이 선택되어 있는지를 확인하십시오.

## 사용 가이드

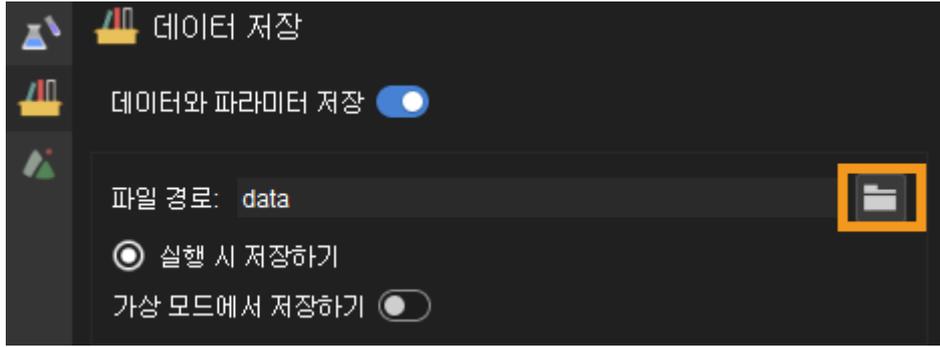
1. [데이터와 파라미터 저장] 버튼을 킨 후 프로젝트를 실행할 때 데이터를 지정한 경로에 저장할 것이며 기본적인 저장 경로는 프로젝트 폴더에 있는 **data** 폴더입니다.



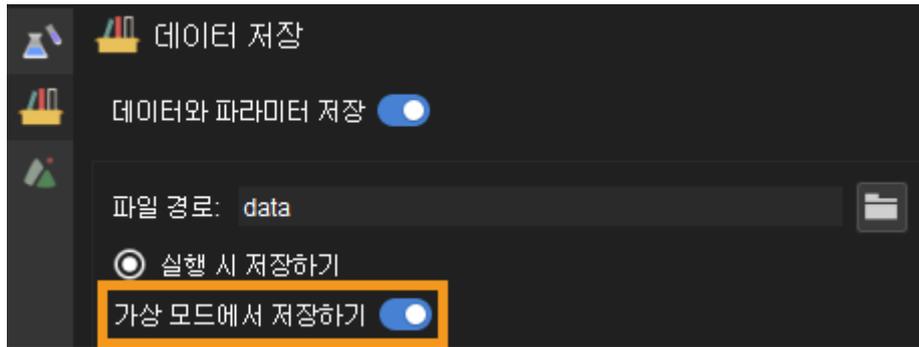
[데이터와 파라미터 저장] 기능을 켜면 프로젝트에 “이미지와 파라미터를 저장하기” 프로시저가 있는지를 감지할 것이며 해당 프로시저가 있는 경우 “이미지와 파라미터를 저장하기” 프로시저를 삭제하라는 알림이 나타날 것입니다.



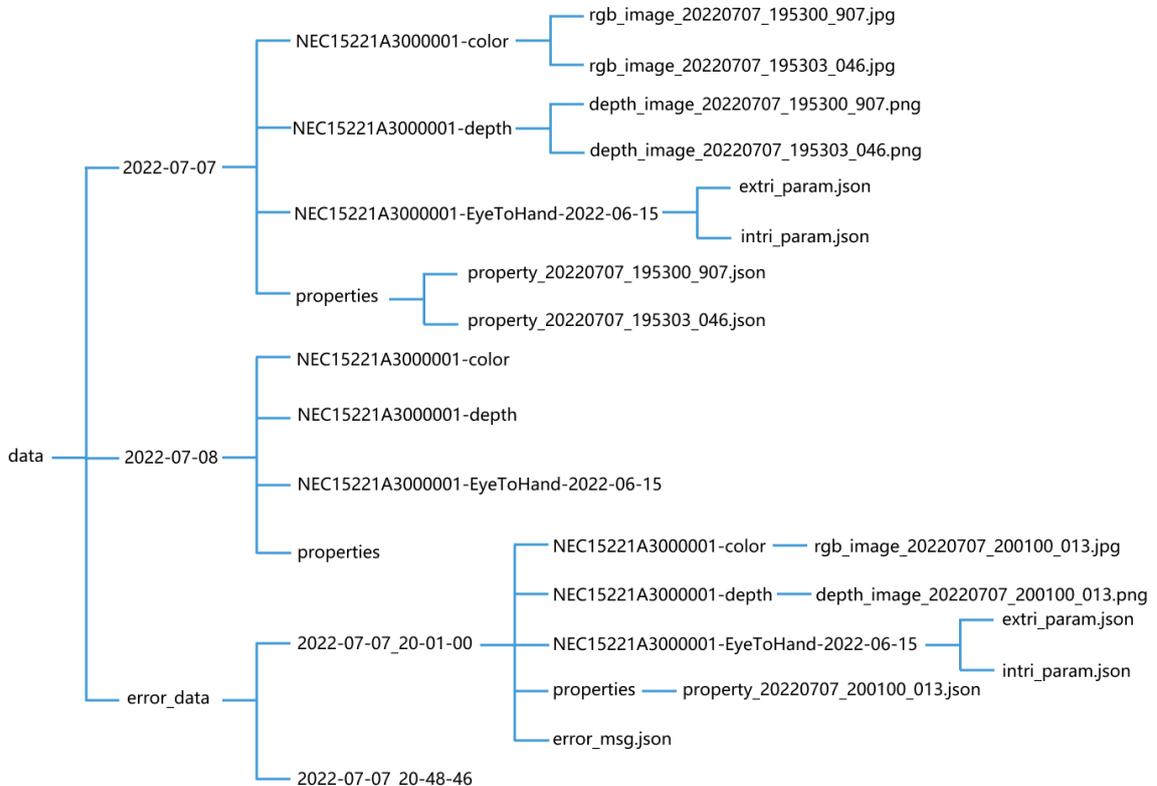
저장 경로를 조절하려면 를 클릭하여 경로를 변경할 수 있습니다.



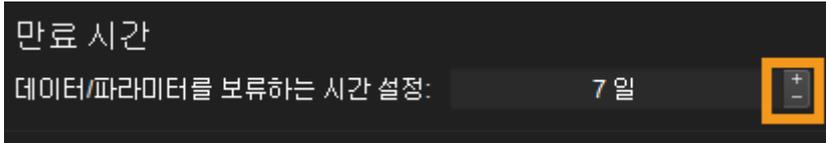
프로젝트 실행 과정에서 E 오류가 발생하면 오류 데이터가 data 폴더에 있는 **error\_data** 폴더에 저장될 것입니다.



한 프로젝트(이 프로젝트에서 “카메라에서 이미지를 캡처하기” 스텝을 통해 덤스 맵과 컬러 맵을 획득함) 를 예시로 설명하겠습니다. 데이터 파일 명칭과 저장 경로의 구조는 아래 그림과 같습니다. 그 중에 이미지 파일의 접미사는 타임스탬프 즉 프로젝트 실행 시간입니다.



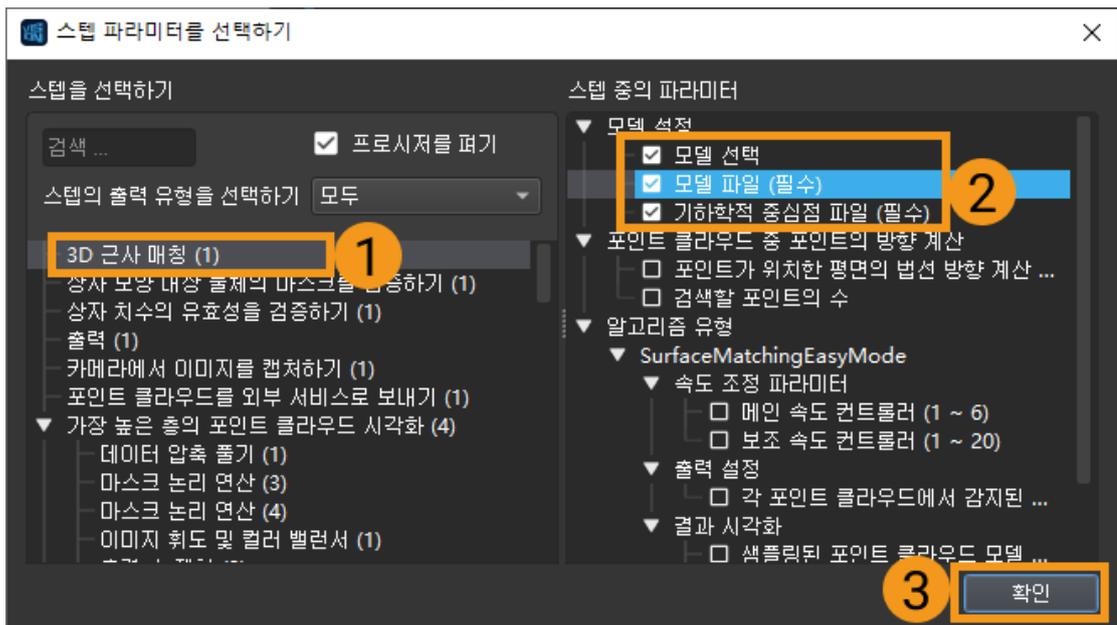
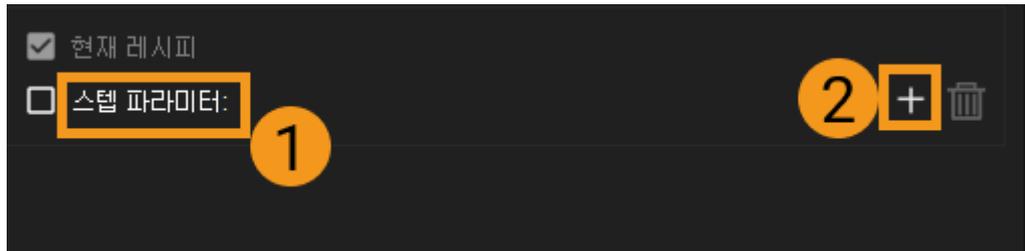
2. **만료 시간**에서 를 클릭하여 데이터를 보류할 수 있는 시간을 설정할 수 있습니다. 설정된 시간을 초과한 후 새로운 데이터가 원래의 데이터를 덮어씁니다.



### 3. 다른 설정

- 현재 레시피: 프로젝트를 실행하는 동안 사용한 레시피들이 저장될 것입니다.
- 스텝 파라미터: 데이터를 재현할 때 해당 스텝을 성공적으로 실행할 수 있도록 저장하려는 파라미터를 선택할 수 있습니다.

스텝 파라미터를 클릭하여 **+**를 클릭하고 팝업된 **스텝 파라미터를 선택하기** 창에서 저장하려는 파라미터를 선택하고 **[ 확인 ]**을 클릭합니다.



선택한 스텝 파라미터를 삭제하려면 스텝 파라미터 리스트에서 해당 파라미터를 선택하고 **🗑**을 클릭하면 됩니다.



- 가상 모드에서 재현할 때 저장된 파라미터를 사용하여 기존 파라미터를 덮어쓰려면 데이터 저장 화면 하단에 있는 **가상 모드의 파라미터를 덮어쓰기** 버튼을 클릭하십시오.



**데이터 저장** 기능은 다수 카메라의 데이터를 저장할 수 있습니다. 즉 Mech-Vision 프로젝트에서 “카메라에서 이미지를 캡처하기” 스텝이 여러 개가 있으면 여러 대 카메라의 데이터가 자동으로 저장될 것입니다.

#### 4.6.10. 로그

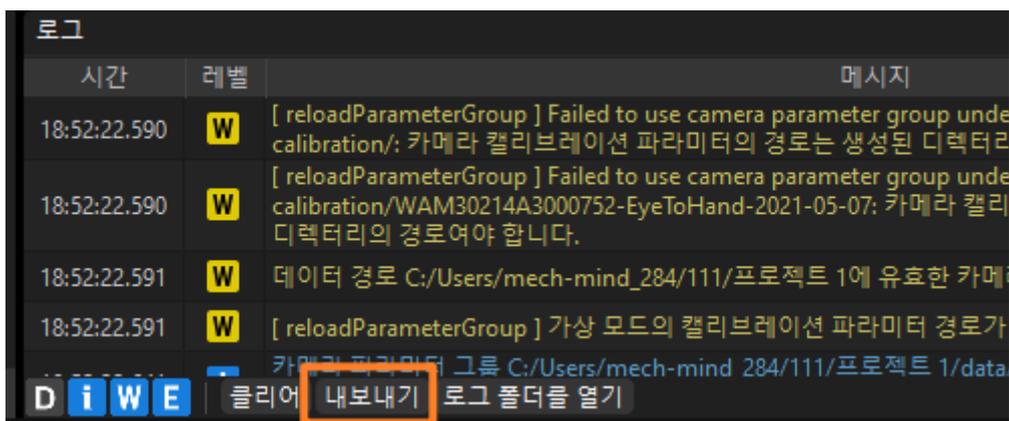
로그바는 사용자들이 구체적인 시각의 프로젝트 실행 기록을 쉽게 찾을 수 있도록 프로젝트 실행의 로그 정보를 실시간으로 표시합니다.

로그바에서 **Vision** 및 **Console** 로그라는 두 모듈의 로그를 보도록 전환할 수 있습니다. 그 중 **Vision** 로그는 비전 솔루션/프로젝트 실행 프로세스 정보, **Console** 로그는 콘솔 정보를 보는 데 사용됩니다.

로그의 관련 작업은 아래에서 소개합니다.

##### 로그를 도출하기

로그바 아래의 [출력] 버튼을 클릭하면 팝업창에서 Html 형식의 로그 파일을 찾을 수 있습니다.



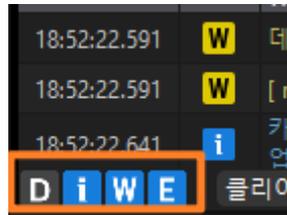
名称	修改日期	类型
2022-11-30.log	2022/11/30 10:46	文本文档
2022-11-30_11-08-54_489.html	2022/11/30 11:08	Chrome

## 로그 정보를 필터링하기

로그 레벨에 대한 설명은 아래와 같습니다.

로그 레벨	설명
D	결함 수정(Debug)
i	정상 정보(Info)
W	경고(Warning)
E	오류(Error)

특정 레벨의 로그 정보를 보려면 로그바의 왼쪽 하단에서 로그 레벨을 클릭합니다. 로그 레벨 버튼이 파란색이면 표시된 상태이고 회색이면 표시되지 않은 상태입니다.



## 기타 작업

### 클리어

현재 로그바의 정보를 지우려면 로그바에서 [클리어] 버튼을 클릭하십시오.

### 로그 정보를 복사하기

- **임의의 행의 일부 로그를 복사:** 로그바의 로그를 두 번 클릭하고 복사할 로그 정보를 선택한 후 마우스 오른쪽 버튼을 클릭하여 복사를 선택해 현재 선택된 로그를 복사합니다.
- **한 줄의 로그 내용을 전체적으로 복사:** 로그바에서 임의의 행의 로그 정보를 두 번 클릭하고 마우스 오른쪽 버튼을 클릭한 다음 모두 선택을 선택하면 현재 줄의 모든 로그 정보가 선택됩니다. 그런 다음 마우스 오른쪽 버튼을 클릭하고 복사를 선택하여 한 줄의 로그 내용을 전체적으로 복사합니다.

## 5. Mech-Viz

Mech-Viz 사용자 매뉴얼에 오신 것을 환영합니다!

Mech-Viz는 비전 시스템에서 제공하는 비전 결과에 따라 충돌 없는 로봇 이동 경로를 동적으로 생성할 수 있는 로봇 동적 경로 계획 소프트웨어입니다. 완전한 그래픽 인터페이스로 사용하기 쉬우며 사용자는 프로그래밍 전문 지식을 가질 필요가 없습니다. 이 소프트웨어는 로봇 동작의 원키 시뮬레이션을 지원하며 다양한 산업 현장의 실제 요구 사항을 충족하기 위해 충돌 감지, 상자 조합, 빨판의 지능형 오프셋 및 혼합 팔레타이징과 같은 다양한 도구가 내장되어 있습니다.

본 사용자 매뉴얼은 다음과 같은 부분으로 구성됩니다:

번호	제목	내용
1	<a href="#">Mech-Viz 소개</a>	기본 개념 소개, 사용자 인터페이스 소개 및 프로젝트 구축 방안 소개
2	<a href="#">프로젝트 리소스 구성</a>	로봇, 말단장치, 작업물, 시나리오 물체 등 프로젝트 리소스를 구성하는 방법에 관한 소개
3	<a href="#">작업 흐름 구축</a>	작업 흐름 구축에 관한 소개 및 예시
4	<a href="#">스텝 라이브러리</a>	스텝 파라미터에 관한 상세한 설명
5	<a href="#">프로젝트 시뮬레이션 최적화</a>	충돌 감지 설정, 프로젝트 시뮬레이션, 계획 기록 및 로그의 사용 방법
6	<a href="#">도구 사용 가이드</a>	모델 편집기, 빨판 구성기, 파렛트 편집기 등 도구의 사용 방법
7	<a href="#">부록</a>	로봇 모델 제작 및 도입, 옵션 설정, 디스플레이 설정 등

### 5.1. Mech-Viz 소개

#### 5.1.1. 기본 개념

##### 프로젝트

“프로젝트”는 Mech-Viz 소프트웨어에서 생성된 로봇 경로 계획 항목을 말합니다. Mech-Viz에서 프로젝트에 필요한 모든 구성을 완료하면 이 프로젝트를 통해 로봇의 경로를 계획하고 로봇의 이동을 컨트롤할 수 있습니다. 프로젝트의 모든 구성 사항은 프로젝트와 이름이 동일한 폴더에 저장됩니다.

##### 프로젝트 리소스

“프로젝트 리소스”는 로봇, 말단장치, 작업물, 시나리오 물체 등 프로젝트에 필요한 다양한 리소스를 가리킵니다.

##### 로봇

Mech-Viz에서 말하는 “로봇”은 일반적으로 사용되는 다관절 산업용 로봇 또는 트러스 로봇입니다.

##### 말단장치

“말단장치”란 그리퍼, 빨판과 같이 로봇이 작업을 완료하기 위해 로봇의 플랜지에 특별히 설계되고

장착되는 장치입니다.

### 작업물

“작업물”은 상자, 금속 부품, 접착 또는 용접할 부품 등과 같은 로봇 말단장치의 작업 대상을 가리킵니다.

### 시나리오 물체

“시나리오 물체”는 로봇이 실제로 작업할 때 시나리오에 있는 각종 물체를 가리킵니다. 일반적으로 로봇 안전 펜스, 빈, 팔레트, 카메라 및 카메라 스탠드 등 물체들이 포함됩니다.

### 작업 흐름

“작업 흐름”이란 Mech-Viz 에서 플로우 차트의 식으로 구축되는 로봇 이동 컨트롤 프로그램입니다.

### 좌표계

Mech-Viz 프로젝트에 다음과 같은 몇 가지 좌표계가 사용될 수 있습니다.

#### 세계 좌표계

3D 시뮬레이션 공간의 중심점을 원점으로 하는 좌표계이며 위치는 고정됩니다.

#### 로봇 베이스 좌표계

원점은 로봇 베이스의 밑면에 위치하며 좌표계는 로봇 위치와 함께 움직입니다. 비전 결과가 카메라 외부 파라미터를 통해 로봇 좌표계로 변환될 때 이 좌표계로 변환됩니다. 기본적으로 세계 좌표계와 겹칩니다.

#### 로봇 TCP 좌표계

원점이 말단장치 끝점에 위치하는 좌표계입니다.

#### 시나리오 물체 좌표계

원점이 시나리오 물체 모델 중심점에 위치하는 좌표계입니다. 각 시나리오 물체 모델에는 시나리오 물체 좌표계가 있습니다.

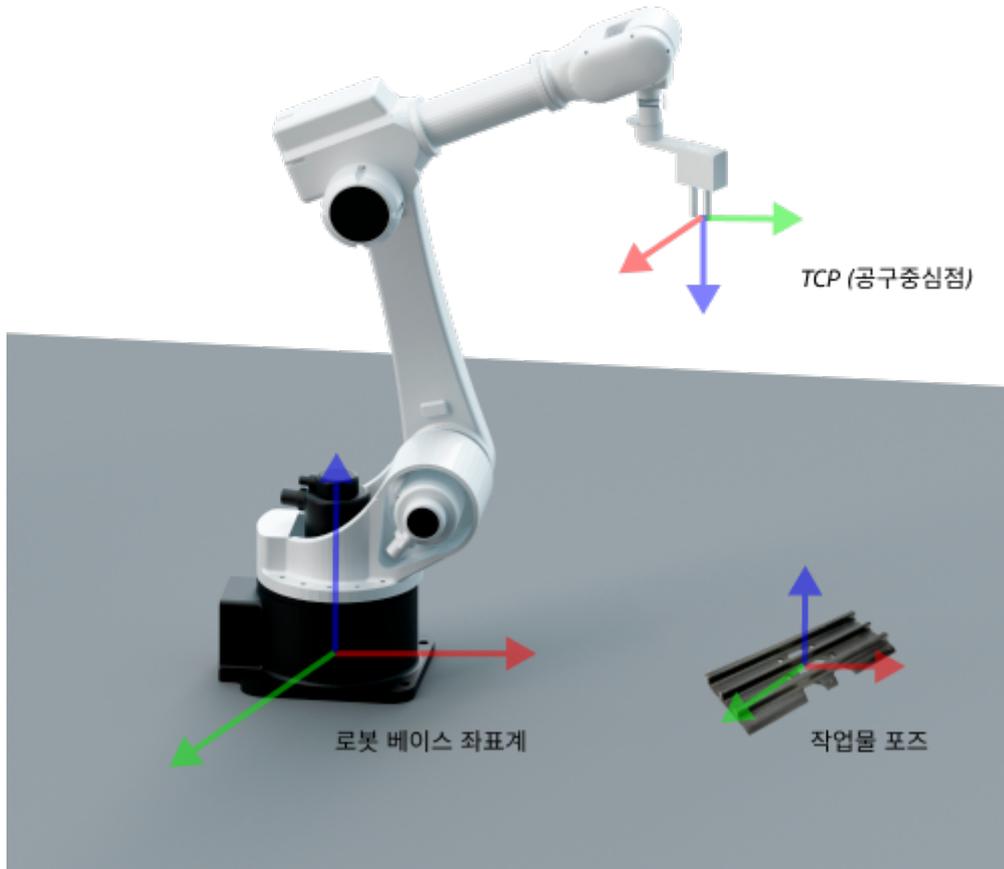
### TCP(Tool Center Point)(공구중심점)

TCP(공구중심점)는 말단장치 끝쪽에 위치하는 점입니다.

작업물 피킹 등 작업을 수행하기 위해 로봇을 공간의 특정 지점으로 이동시키는 것은 본질적으로 TCP를 해당 지점으로 이동시키는 것입니다.

### 작업물 포즈

작업물 포즈는 작업물의 어떤 포인트가 로봇 베이스를 기준으로 한 포즈이며 이 포인트는 말단장치의 대상 포인트입니다.



### 말단장치 대칭성

말단장치의 대칭성은 말단장치의 대칭축을 중심으로 일정 각도 회전 후 형상이 회전 전과 일치할 수 있는 특성을 나타내는 것입니다.

### 작업물 대칭성

작업물의 대칭성은 작업물의 대칭축을 중심으로 일정 각도 회전 후 형상이 회전 전과 일치할 수 있는 특성을 나타내는 것입니다.

상세한 내용은 [작업물의 회전 대칭성](#)을 참조하십시오.

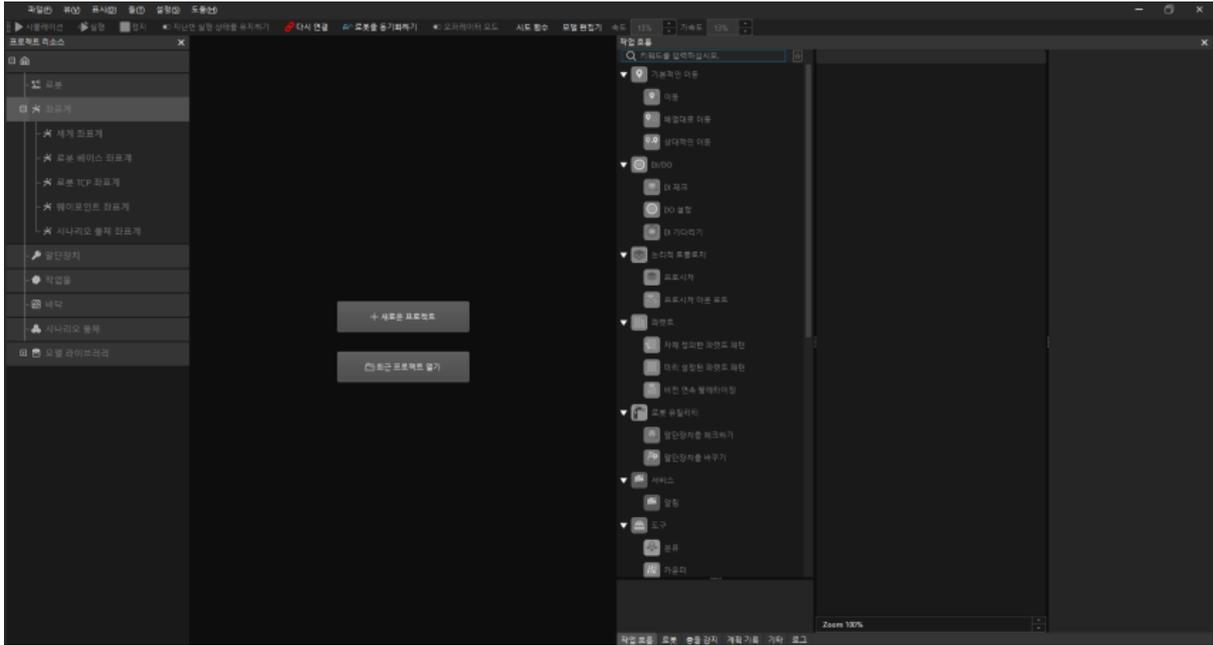
### 작업물 피킹 허용 편차

일부 작업물을 피킹할 때 로봇의 피킹 포즈가 일정한 각도 범위 내에서 유연하게 조절될 수 있으며 이 각도 범위는 피킹 허용 편차입니다.

상세한 내용은 [작업물 피킹 허용 편차](#)를 참조하십시오.

## 5.1.2. 사용자 인터페이스

Mech-Viz의 시작 화면은 아래 그림과 같습니다. 여기서 **새로운 프로젝트**를 만들거나 **최근에 실행했던 프로젝트**를 열 수 있습니다.



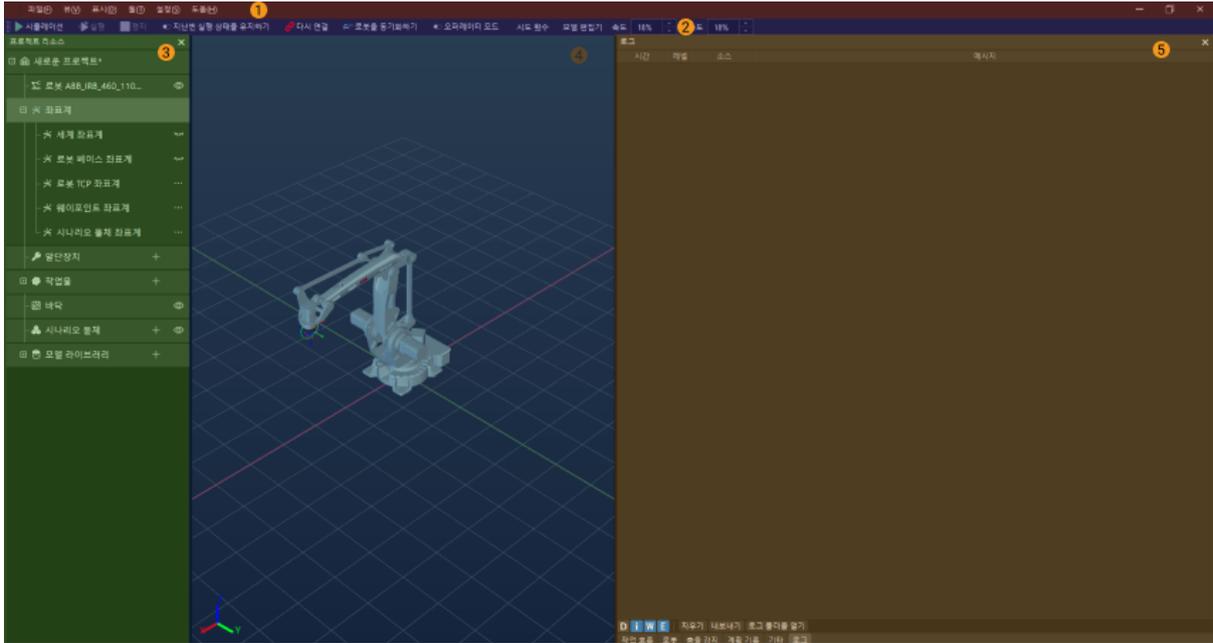
## ● 새로운 프로젝트

1. [ **새로운 프로젝트** ] 버튼을 클릭하십시오.
2. 검색창에서 원하는 로봇의 모델을 입력하거나 선택 조건을 수정하여 로봇 모델을 선택합니다. 원하는 로봇 모델을 찾지 못하는 경우 다음과 같이 해결해 보십시오:
  - a. **온라인 로봇 라이브러리** 버튼을 클릭하여 필요한 로봇 모델을 다운로드합니다. 온라인 로봇 라이브러리에 원하는 로봇 모델을 찾을 수 없는 경우, **로봇 모델 제작 및 도입** 내용을 참조하십시오.
  - b. **새로운 프로젝트** 화면을 닫은 후 로봇 모델 파일을 소프트웨어로 드래그하거나 **툴 > 로봇 라이브러리 도구 > 로봇을 도입하기** 버튼을 클릭하여 팝업창에서 로봇 모델 파일을 선택하십시오.
  - c. [ **새로운 프로젝트** ] 버튼을 클릭한 후 다시 검색하십시오.
3. 해당 로봇 카드를 더블클릭하거나 마우스를 로봇 카드 위에 넣어 [ **선택** ] 버튼을 클릭하여 로봇 모델을 선택하십시오.

## ● 최근에 실행했던 프로젝트

[ **최근에 실행했던 프로젝트** ] 버튼을 클릭하여 프로젝트 리스트에서 원하는 프로젝트를 더블클릭하거나 프로젝트를 선택한 후 [ **열기** ] 버튼을 클릭하십시오.

시작 화면을 닫은 후의 메인 인터페이스는 아래 그림과 같습니다.



주로 다섯 가지 부분으로 구성됩니다:

1	메뉴 바	파일, 뷰, 표시, 툴, 설정, 도움말 등이 포함됩니다. 상세한 내용은 해당 링크를 클릭하여 참조하십시오.
2	툴 바	로봇 이동 컨트롤, 실행 속도 조절 등 프로젝트 시뮬레이션에 사용됩니다. 상세한 내용은 툴 바를 참조하십시오.
3	프로젝트 리소스	말단장치, 작업물, 시나리오 물체 등 프로젝트에 사용되는 리소스를 관리합니다. 상세한 내용은 프로젝트 리소스를 참조하십시오.
4	3D 시뮬레이션 공간	프로젝트 실행 또는 시뮬레이션 과정에서 로봇의 이동 경로, 충돌 감지 결과, 비전 시스템을 통해 획득한 포인트 클라우드 및 인식된 대상 물체의 포즈 등이 여기에서 표시됩니다. 상세한 내용은 3D 시뮬레이션 공간을 참조하십시오.
5	기능 패널	작업 흐름, 로봇, 충돌 감지, 계획 기록, 기타, 로그 등이 포함됩니다. 상세한 내용은 해당 링크를 클릭하여 참조하십시오.

### 5.1.2.1. 메뉴 바

이 부분에서는 메뉴 바에 있는 각 옵션에 대해 소개하겠습니다.

파일(F) 뷰(V) 표시(O) 툴(T) 설정(S) 도움말(H)

#### 파일

옵션	설명	단축키
새로운 프로젝트	새로운 프로젝트를 만들기	Ctrl + N
프로젝트 열기	기존 프로젝트 폴더를 선택하여 열기	Ctrl + O

옵션	설명	단축키
최근에 실행했던 프로젝트	최근에 사용했던 프로젝트를 표시하고 프로젝트 이름을 클릭하면 바로 열 수 있음	없음
탐색기에서 실행 파일 열기	Mech-Viz의 설치 경로를 열기	없음
프로젝트를 저장하기	현재 프로젝트에 대한 변경 사항을 저장하기	Ctrl + S
프로젝트를 JSON 파일로 저장하기	프로젝트를 저장하고 프로젝트 폴더에 있는 .viz 파일을 .json 포맷으로 저장하기	없음
저장 경로를 선택하기	프로젝트를 지정된 위치에 저장하기	Ctrl + Shift + S
프로젝트를 닫기	현재 프로젝트를 닫기	Ctrl + W
종료	Mech-Viz 소프트웨어를 종료하기	Ctrl + Q



Mech-Viz 1.4.0 및 더 높은 버전에서는 프로젝트의 실행 파일을 viz 포맷으로 저장하며 이전 버전의 json 파일과 대응합니다.

## 뷰

소프트웨어 UI에 대해 조정할 수 있습니다.

옵션	설명
프로젝트 리소스	선택하면 "프로젝트 리소스" 패널이 표시되며 기본적으로 선택되어 있음
작업 흐름	선택하면 "작업 흐름" 패널이 표시되며 기본적으로 선택되어 있음
로봇	선택하면 "로봇" 패널이 표시되며 기본적으로 선택되어 있음
충돌 모델	선택하면 "충돌 감지" 패널이 표시되며 기본적으로 선택되어 있음
계획 기록	선택하면 "계획 기록" 패널이 표시되며 기본적으로 선택되어 있음
기타	선택하면 "기타" 패널이 표시되며 기본적으로 선택되어 있음
로그	선택하면 "로그" 패널이 표시되며 기본적으로 선택되어 있음
기능 설명	선택하면 스텝 파라미터 표시줄에서 선택된 스텝의 기능 설명이 표시됨
기본적인 레이아웃	선택하면 소프트웨어 UI 레이아웃이 초기 상태로 리셋됨

## 표시

3D 시뮬레이션 공간의 표시 사항을 조정할 수 있습니다.

옵션	설명
수신된 비전 포즈를 표시하기	선택하면 수신된 비전 포즈, 레이블, 번호 등 정보가 표시되며 기본적으로 선택되어 있음
포인트 클라우드를 표시하기	선택하면 수신된 포인트 클라우드를 표시할 것이며 기본적으로 선택되어 있음
피킹된 물체를 표시하기	선택하면 피킹된 물체의 모델을 표시할 것이며 기본적으로 선택되어 있음
계획 시 충돌을 표시하기	선택하면 3D 시뮬레이션 공간에서 계획 시 감지된 충돌의 위치를 표시할 것이며 기본적으로 선택되어 있습니다.
팔진트리를 표시하기	선택하면 물체 포인트 클라우드가 팔진트리로 표시될 것이며 기본적으로 선택되지 않음
수신된 상자 모델을 표시하기	선택하면 3D 시뮬레이션 공간에서 수신된 상자 모델을 표시할 것이며 기본적으로 선택되어 있음
물체 포즈를 표시하기	선택하면 3D 시뮬레이션 공간에서 피킹된 물체의 포즈를 표시할 것이며 기본적으로 선택되지 않음
<b>디스플레이 설정</b>	3D 시뮬레이션 공간의 더 많은 디테일을 설정하기
포즈의 상태 컬러	포즈의 번호와 레이블의 컬러가 대표하는 포즈의 상태를 확인할 수 있음

## 툴

옵션	설명
저장한 비전 기록을 사용하기	선택 가능한 옵션이며 선택하면 저장한 비전 기록을 사용하여 프로젝트를 실행함
<b>비전 기록 설정</b>	비전 기록을 저장하고 호출하여 디버깅할 때 문제를 재현하는 데 사용됨
비전 결과를 찾기	3D 시뮬레이션 공간에서 찾는 번호와 대응하는 비전 결과를 하이라이트로 표시함
로봇 라이브러리 도구	로봇 모델 파일을 도입하거나 현재 로봇 모델을 지정된 경로로 도출하기

## 설정

옵션	설명
Mech-Center 주소 설정	Mech-Center 소프트웨어의 IP 주소를 설정하기
프로젝트 잠금/잠금 해제	이 옵션을 선택하면 현재 프로젝트의 잠금이 해제됨
<b>옵션</b>	소프트웨어의 일반적인 설정 사항을 수정하는 데 사용됨

옵션	설명
로그 레벨	로그 레벨을 설정하면 로그 패널에서 설정된 레벨 및 이상의 로그 내용을 표시함

#### 도움

버전 정보	소프트웨어의 버전 정보를 확인하기
업데이트 설명	브라우저에서 소프트웨어 업데이트 설명을 열기
사용자 안내서	브라우저에서 소프트웨어 사용자 안내서를 열기

#### 5.1.2.2. 툴 바

이 부분에서는 툴 바의 각 옵션에 대해서 소개하겠습니다.

<b>시뮬레이션</b>	가상 로봇을 실행하기
<b>실행</b>	실제 로봇을 실행하기
<b>정지</b>	현재 프로젝트의 실행 과정을 정지하기
<b>지난 실행 상태를 유지하기</b>	팔레타이징 시나리오의 디버깅에 사용되며 이미 쌓인 위치에서 계속 배치함
<b>다시 연결</b>	Mech-Center와 연결하여 로봇과의 데이터 통신을 실현하기
<b>로봇을 동기화하기</b>	가상 로봇을 실제 로봇의 포즈로 동기화하기
<b>오퍼레이터 모드</b>	실행 효율성을 향상시키기 위해 오퍼레이터 모드에서 포인트 클라우드에 대해 다운 샘플링을 수행하기
<b>시도 횟수</b>	각 작업물을 피킹할 때의 시도 횟수
<b>모델 편집기</b>	모델 편집기를 열기
<b>속도</b>	로봇 실행 시의 속도를 조절하기
<b>가속도</b>	로봇 실행 시의 가속도를 조절하기

#### 5.1.2.3. 프로젝트 리소스

프로젝트 리소스 패널은 아래 그림과 같습니다.



이 부분에서는 프로젝트 리소스 패널에서 마우스 오른쪽 버튼을 클릭하면 나오는 옵션 및 버튼의 기능에 대해 소개하겠습니다.

[프로젝트 리소스 구성](#) 내용을 참조하여 프로젝트에 다양한 리소스를 구성하는 방법을 알아보십시오.

#### 프로젝트

프로젝트 이름, 프로젝트 변경 사항 및 프로젝트가 자동으로 로드되는지 여부를 표시합니다.

마우스 오른쪽 버튼으로 프로젝트 이름을 클릭하면 아래와 같은 옵션들이 나타납니다:

- |                           |   |
|---------------------------|---|
| <b>프로젝트 폴더를 열기</b>        | 프로젝트가 위치하는 폴더를 엽니다.                     |
| <b>해당 프로젝트를 자동으로 로드하기</b> | 선택하면 다음에 소프트웨어를 열 때 해당 프로젝트가 자동으로 열립니다. |
| <b>해당 프로젝트를 백업하기</b>      | 현재 프로젝트를 백업합니다.                         |

작업 흐름에 대한 변경 사항을 저장하지 않거나 프로젝트를 자동으로 로드하는 것으로 설정하면 아래 그림 속의 표지가 표시될 것입니다.



#### 로봇

현재 프로젝트에 사용되는 로봇을 표시합니다.

마우스 오른쪽 버튼으로 로봇 이름을 클릭하면 아래와 같은 옵션들이 나타납니다:

- |                       |                      |
|-----------------------|----------------------|
| <b>로봇 파일 디렉터리를 열기</b> | 로봇 모델이 위치하는 폴더를 엽니다. |
|-----------------------|----------------------|

또한,  아이콘을 클릭하면 로봇을 반투명 상태로 전환할 수 있으며  아이콘을 클릭하면 원래 상태로 복원합니다.

## 좌표계

3D 시뮬레이션 공간의 모든 좌표계를 표시합니다.

 아이콘을 클릭하면 대응한 좌표계를 표시하고  아이콘을 클릭하면 대응한 좌표계를 표시하지 않습니다.  아이콘을 클릭하면 드래그 볼의 "평행이동"과 "회전" 기능을 활성화하거나 비활성화할 수 있습니다.

## 말단장치

현재 프로젝트에 구성된 로봇 말단장치를 표시합니다. [말단장치](#) 내용을 참조하여 말단장치를 구성하는 방법을 알아보십시오.

[+] 버튼을 클릭하면 말단장치 구성 창을 열어 말단장치를 추가합니다.

마우스 오른쪽 버튼으로 이미 구성된 말단장치의 이름을 클릭하면 아래와 같은 옵션들이 나타납니다:

옵션	설명	단축키
복사하기	현재 사용되는 말단장치를 복사합니다.	Ctrl + C
붙여넣기	말단장치를 붙여넣습니다.	Ctrl + V
삭제하기	선택된 말단장치를 삭제합니다.	Delete
이름을 바꾸기	현재 말단장치의 이름을 수정합니다.	
말단장치 구성	현재 말단장치의 구성 창을 엽니다.	
현재 작업 도구로 설정하기	해당 말단장치를 현재 작업 도구로 설정합니다.	

 아이콘을 클릭하면 말단장치를 반투명 상태로 전환할 수 있으며  아이콘을 클릭하면 원래 상태로 복원합니다.

## 작업물

현재 프로젝트에 구성된 작업물을 표시합니다. [작업물](#) 내용을 참조하여 작업물을 구성하는 방법을 알아보십시오.

[+] 버튼을 클릭하면 작업물 구성 창을 열어 작업물을 추가할 수 있습니다.

마우스 오른쪽 버튼으로 이미 구성된 작업물의 이름을 클릭하면 아래와 같은 옵션들이 나타납니다:

옵션	설명	단축키
삭제하기	현재 작업물을 삭제합니다.	Delete
이름을 바꾸기	현재 작업물의 이름을 바꿉니다.	
작업물 구성	현재 작업물의 구성 창을 엽니다.	



레이블이 없는 비전 결과인 경우, "기본값"을 사용합니다.

## 바닥

바닥 높이 및 표시 상태를 설정합니다.

마우스 오른쪽 버튼으로 바닥을 클릭하면 바닥의 높이를 조정할 수 있습니다.

 아이콘을 클릭하면 바닥을 표시하지 않고  아이콘을 클릭하면 바닥을 표시합니다.

### 시나리오 물체

현재 프로젝트에 구성한 시나리오 물체를 표시합니다. **시나리오 물체** 내용을 참조하여 시나리오 물체를 구성하는 방법을 알아보십시오.

[+] 버튼을 클릭하면 시나리오 물체의 구성 창을 열어 시나리오 물체를 추가할 수 있습니다.

마우스 오른쪽 버튼으로 이미 구성한 시나리오 물체의 이름을 클릭하면 아래와 같은 옵션들이 나타납니다:

옵션	설명	단축키
복사하기	현재 시나리오 물체를 복사합니다.	Ctrl + C
붙여넣기	현재 시나리오 물체를 붙여넣습니다.	Ctrl + V
잘라내기	현재 시나리오 물체를 잘라냅니다.	Ctrl + X
삭제하기	현재 시나리오 물체를 삭제합니다.	Delete
이름을 바꾸기	현재 시나리오 물체의 이름을 바꿉니다.	
시나리오 물체 구성	현재 시나리오 물체의 구성 창을 엽니다.	

 아이콘을 클릭하면 시나리오 물체를 반투명 상태로 전환하고  아이콘을 클릭하면 완전한 투명 상태로 전환하며  아이콘을 클릭하면 원래 상태로 복원합니다.

또한 시나리오 물체의 이름을 드래그하면 시나리오 물체 모델을 하위 모델 또는 독립적인 모델로 변환할 수 있습니다.

### 모델 라이브러리

소프트웨어에 추가된 모든 사용자 자체 정의 모델을 표시하며 모델 포맷에 따라 분류합니다.

[+] 버튼을 클릭하면 사용자가 자체 정의한 모델을 도입하기 위해 파일 선택 창을 엽니다.

마우스 오른쪽 버튼으로 이미 도입한 모델의 이름을 클릭하면 아래와 같은 옵션들이 나타납니다:

옵션	설명	단축키
복사하기	현재 모델을 복사합니다.	Ctrl + C
붙여넣기	복사한 모델을 붙여넣습니다.	Ctrl + V
삭제하기	현재 모델을 삭제합니다.	Delete
모델 구성	현재 모델의 편집 창을 엽니다.	
모델 편집기	모델 편집기를 열어 현재 모델을 기준 모델로 사용합니다.	
모델 파일 폴더를 열기	현재 모델 파일이 위치하는 디렉토리를 엽니다.	

### 5.1.2.4. 3D 시뮬레이션 공간

3D 시뮬레이션 공간에는 로봇, 시나리오 물체, 말단장치, 작업물 모델, 로봇 이동 경로, 픽 포인트 및 충돌 등을 표시하며 시나리오 물체의 모델을 드래그할 수도 있습니다.

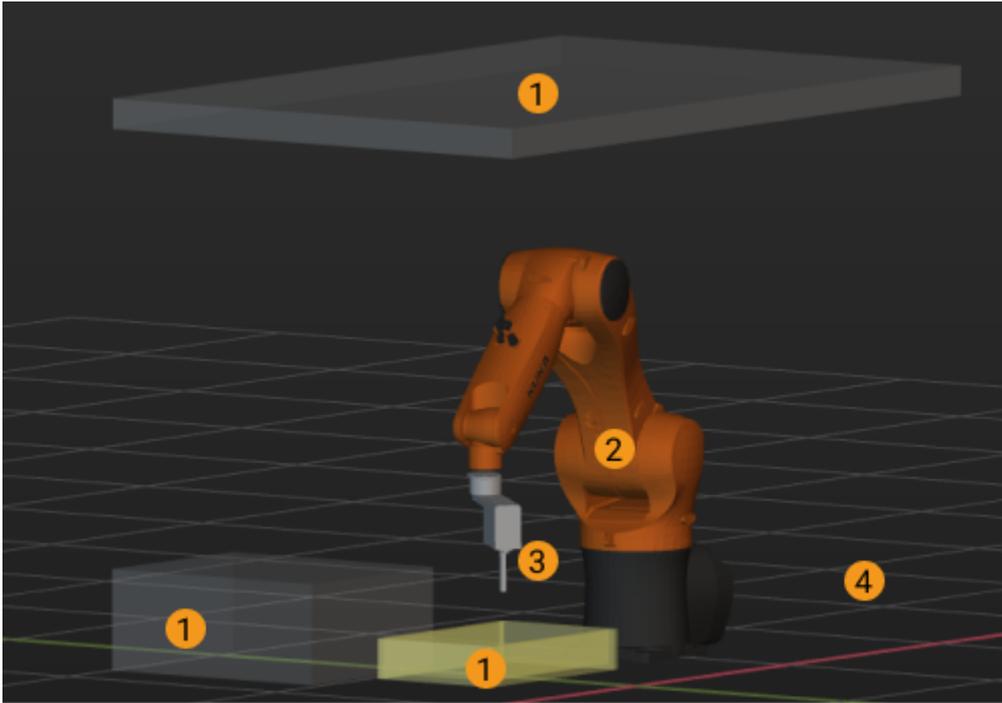
#### 기본 작업

3D 시뮬레이션 공간의 표시는 다음 마우스 조작으로 조정할 수 있습니다.

시각을 회전하기	마우스 왼쪽 버튼을 길게 눌러 임의의 방향으로 드래그합니다.	
뷰 전환 메뉴	마우스 오른쪽 버튼을 클릭합니다.	
시각을 평행이동하기	마우스 가운데 버튼을 길게 눌러 임의의 방향으로 드래그합니다.	
크기 조정	마우스 휠을 조절합니다.	

#### 시나리오에 있는 물체를 표시하기

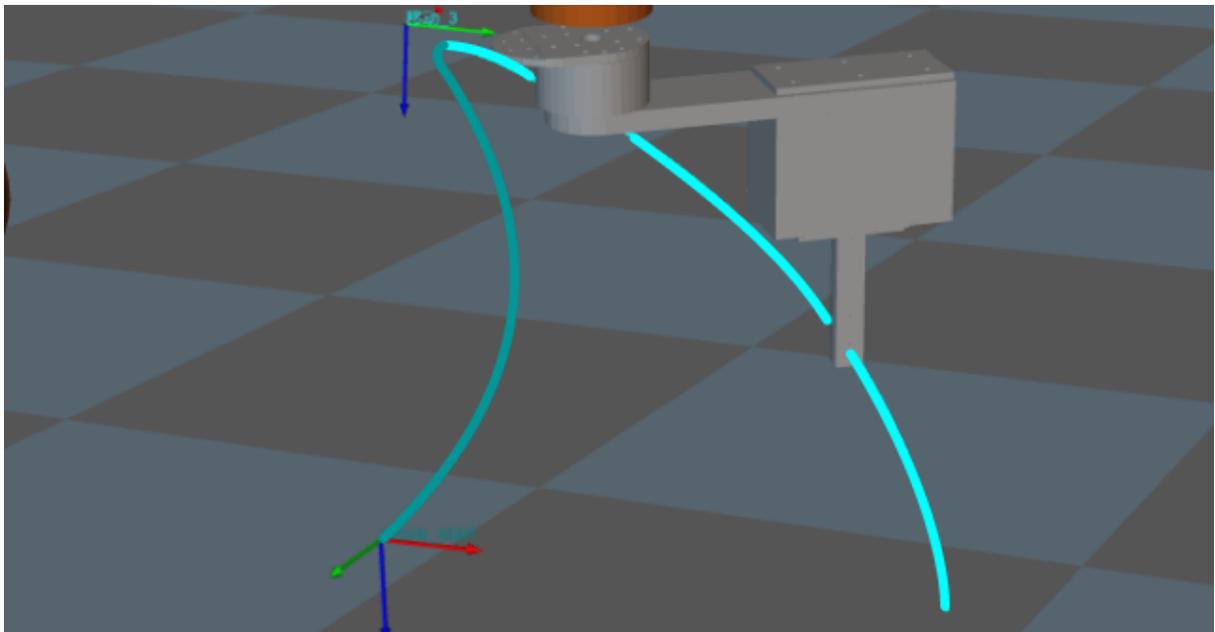
프로젝트 구축 시의 3D 시뮬레이션 공간은 주로 아래 그림에 표시된 부분으로 구성됩니다.



① 시나리오 물체 ② 로봇 ③ 말단장치 ④ 바닥

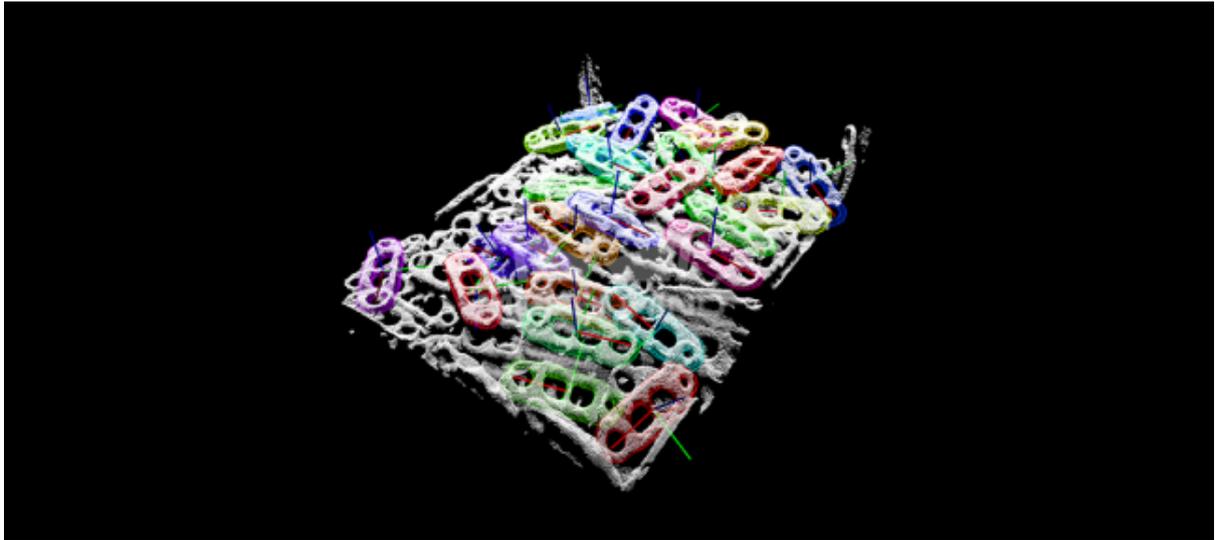
#### 이동 경로를 표시하기

실제 로봇을 컨트롤하거나 가상 로봇의 운동 과정을 시뮬레이션하는 과정에서 로봇이 이동할 경로를 표시할 수 있습니다. 시뮬레이션 경로를 통해 로봇의 실제 이동 경로를 더 합리적으로 계획할 수 있습니다.



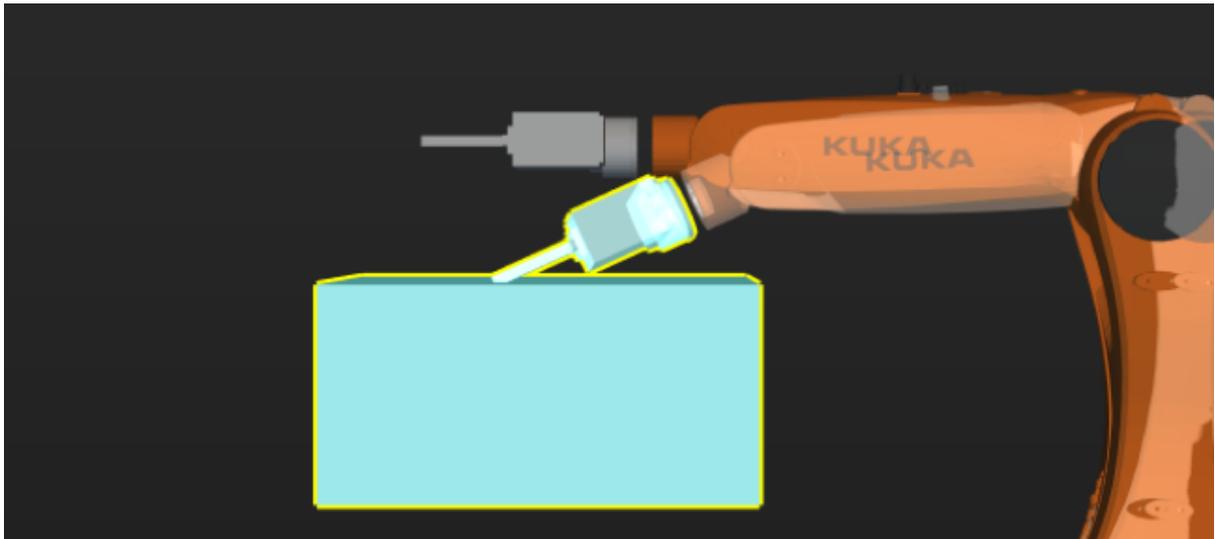
#### 포인트 클라우드 및 픽 포인트를 표시하기

소프트웨어는 비전 서비스가 보내온 비전 결과를 3D 시뮬레이션 공간에서 표시할 것입니다. 일반적으로 완전한 비전 결과라면 다음과 같은 것을 포함해야 합니다. 작업물 포인트 클라우드, 시나리오 포인트 클라우드, 작업물 포즈, 작업물 인덱스 및 작업물 특성을 나타내는 레이블.



### 충돌 예측을 표시하기

프로젝트 실행 시 Mech-Viz는 로봇의 이동 경로에서 로봇과 말단장치가 시나리오에 있는 다른 물체, 작업물 또는 상자과 충돌할지를 감지합니다. 충돌을 감지하면 3D 시뮬레이션 공간에서 충돌이 발생한 대상을 하이라이트로 표시할 것입니다. 아래 그림과 같습니다.



### 5.1.2.5. 작업 흐름

작업 흐름의 패널은 아래 그림과 같습니다.



작업 흐름 패널은 ① 스텝 라이브러리, ②작업 흐름 편집 영역, ③ 파라미터 편집 영역, ④ 작업 기록 영역으로 구성됩니다.

로봇 프로그래밍을 완료하기 위한 작업 흐름을 구축하려면 상세한 내용은 [작업 흐름 구축](#)을 참조하십시오.

### 스텝 라이브러리

스텝들은 기능에 따라 다양한 클래스로 나뉩니다. 예를 들어 "기본 운동" 클래스에 이동, 상대적인 이동, 배열대로 이동 등 로봇의 기본적인 운동과 관련된 스텝을 포함합니다. 스텝 클래스를 클릭하여 해당 클래스 아래의 모든 스텝을 표시하거나 숨길 수 있습니다.

스텝은 주로 "이동" 및 "비이동" 스텝으로 나뉩니다.

- 이동 스텝: 기본 운동, 팔레타이징, 비전 이동 등
- 비이동 스텝: DI DO, 논리적 토폴로지, 로봇 유틸리티, 서비스, 도구, 비전(비전 이동을 제외함), 기타

스텝 라이브러리에서 다음과 같은 작업을 통해 스텝의 표시를 조정할 수 있습니다.

- 스텝 라이브러리 오른쪽 상단에 있는 아이콘을 클릭하면 모든 스텝이 표시될 것입니다.

아이콘을 클릭하면 자주 사용되는 스텝만 표시됩니다.

- 마우스 오른쪽 버튼을 클릭하고 팝업창에서 **모두 표시하기** 버튼을 클릭하면 모든 스텝을 볼 수 있습니다.
- 마우스 오른쪽 버튼을 클릭하고 팝업창에서 **자주 사용되는 스텝만 표시하기** 버튼을 클릭하면 자주 사용하는 스텝만 볼 수 있습니다.
- 마우스 오른쪽 버튼을 클릭하고 팝업창에서 **모두 보기** 버튼을 클릭하면 모든 스텝 클래스를 펼칠 것입니다.
- 마우스 오른쪽 버튼을 클릭하고 팝업창에서 **모두 숨기기** 버튼을 클릭하면 모든 스텝 클래스를 숨길 것입니다(즉 스텝 클래스의 이름만 볼 수 있음).

또한, 스텝 라이브러리 위에 있는 검색창에서 키워드를 입력하면 스텝을 검색할 수 있습니다.

### 프로젝트 편집 영역

프로젝트 편집 영역은 다양한 스텝을 조합하여 그래픽 프로그래밍을 진행하는 영역입니다.

프로젝트 편집 영역에서 마우스 오른쪽 버튼으로 클릭하면 다음과 같은 옵션이 나타납니다:

이전으로	프로시저의 이전 화면으로 돌아갑니다.
다음으로	프로시저 화면으로 들어갑니다(프로시저를 마우스 오른쪽 버튼으로 클릭해야 함).
복사하기	선택된 스텝을 복사합니다.
잘라내기	선택된 스텝을 잘라냅니다.
붙여넣기	복사한 스텝을 붙여넣습니다.
삭제하기	선택된 스텝을 삭제합니다.
모두 선택	프로젝트 편집 구역에 있는 모든 스텝을 선택합니다.
프로젝트 스크린샷을 저장하기	프로젝트 스크린샷을 임의의 경로로 저장합니다.
자동으로 배치하기	현재 모든 스텝에 대해 자동으로 배치합니다.
프로시저로 합치기	선택된 여러 개 스텝을 하나의 프로시저로 합칩니다.
프로시저를 분할하기	선택된 프로시저를 분할합니다.
프로시저 도입/도출	선택된 프로시저를 도출하거나 프로시저를 새로 도입합니다.
사용자 정의 설명	선택된 스텝에 대해 자체 정의한 설명을 추가합니다.
DO를 리셋하기	DO 신호를 0으로 리셋하며 일반적으로 빨판을 닫는 데 사용됩니다.

프로젝트 편집 영역에서 마우스 왼쪽 버튼으로 클릭하고 **Ctrl + F** 버튼을 누른 후 검색창이 나올 것이며 검색창에서 스텝을 검색하거나 선택할 수 있습니다.

또한 **ctrl** 버튼을 길게 누르고 마우스 휠을 슬라이드하여 확대/축소할 수 있습니다.

### 파라미터 편집 영역

프로젝트 편집 영역에서 어떤 스텝을 선택하면 해당 스텝에 관한 기능 설명 및 파라미터들이 표시됩니다.

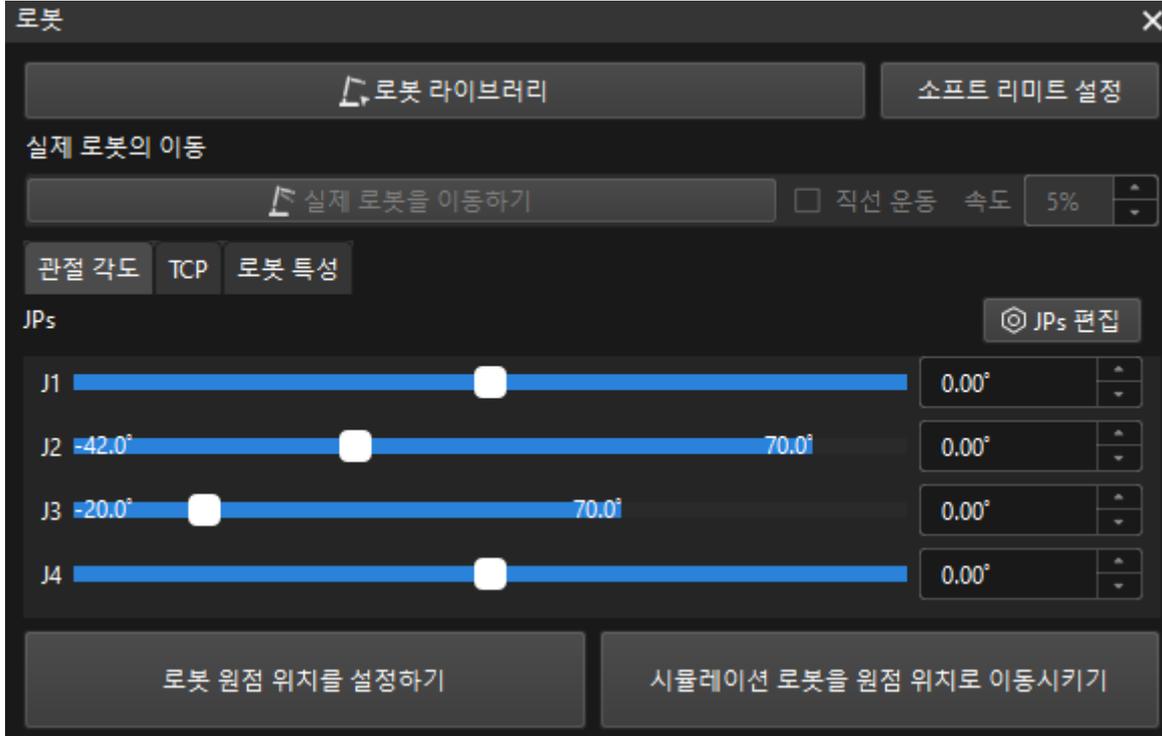
## 작업 기록 영역

스텝 추가/삭제 및 연결선 추가/삭제 등 프로젝트 편집 영역에서 진행된 작업들을 표시합니다.

구체적인 기록 항목을 클릭하면 해당 기록이 위치하는 상태로 돌아갈 것입니다.

### 5.1.2.6. 로봇

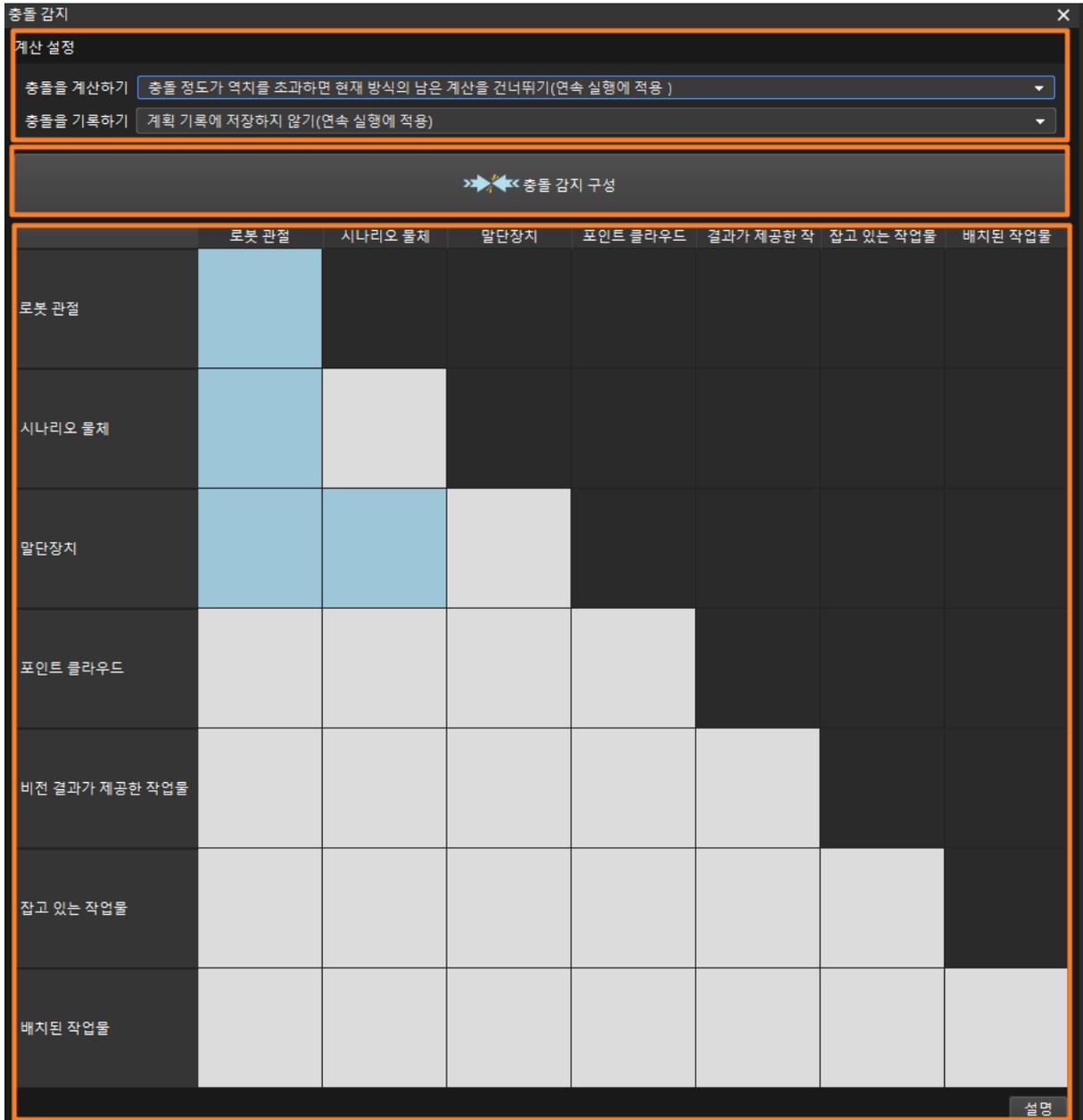
로봇 패널은 아래 그림과 같습니다.



로봇 패널에서 로봇 모델 바꾸기, 로봇 소프트 리미트 설정하기, 실제 로봇을 이동하기, 로봇 포즈를 조정하기, 로봇 원점 위치를 설정하기 등 기능을 실현할 수 있습니다. 상세한 설명은 [로봇 구성](#) 내용을 참조하십시오.

### 5.1.2.7. 충돌 감지

충돌 감지 패널은 아래 그림과 같습니다.



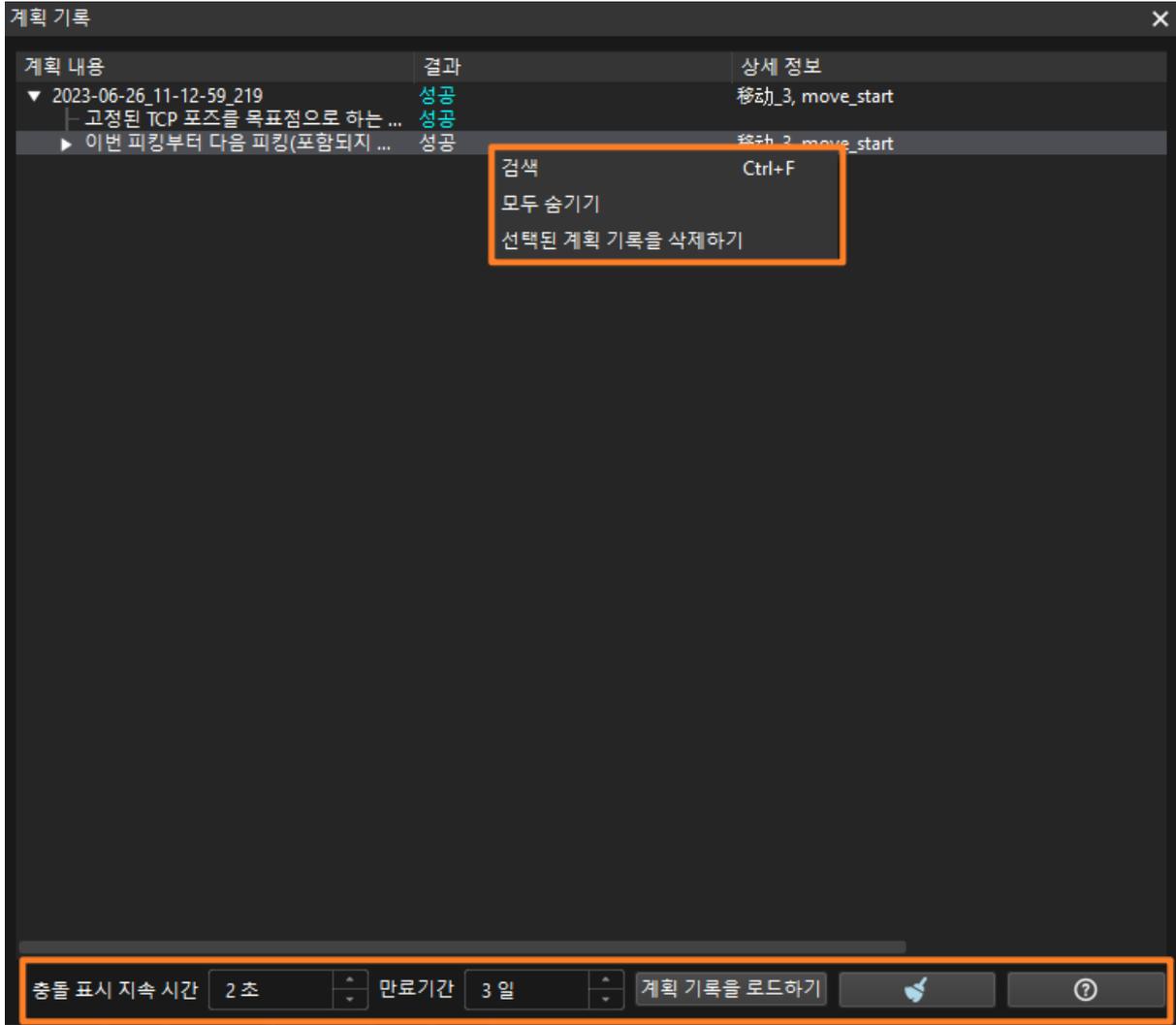
계산 설정: 계산 설정을 수정하면 소프트웨어의 실행 속도를 높일 수 있습니다. 상세한 설명은 [계산 설정](#)을 참조하십시오.

충돌 감지 구성: [충돌 감지 구성] 버튼을 클릭하면 충돌 감지 구성 창을 열 수 있으며 여기서 포인트 클라우드, 말단장치, 로봇 관절, 피킹된 물체와의 충돌 감지를 구성할 수 있습니다. 상세한 작업 설명은 [충돌 감지](#) 내용을 참조하십시오.

충돌 감지 도표: 충돌 감지 도표는 모든 충돌 감지의 구성을 보여줍니다. [설명] 아이콘을 클릭하여 도표에서 다른 색상의 의미를 봅니다.

### 5.1.2.8. 계획 기록

계획 기록 패널의 화면은 아래 그림과 같습니다.



계획 기록 패널에서 Mech-Viz 상세한 계획 기록을 볼 수 있고 문제 발생 원인을 분석하며 프로젝트를 최적화할 수 있습니다. 상세한 정보는 [계획 기록](#) 내용을 참조하십시오.

이 패널 하단에서 계획 기록과 관련된 설정 사항을 제공합니다:

<b>충돌 표시 지속 시간</b>	충돌로 인한 실패 항목을 선택한 후 3D 시뮬레이션 공간에서 지속적으로 충돌을 표시하는 시간입니다.
<b>만료기간</b>	계획 기록의 데이터를 저장하는 시간 제한입니다. 이 시간을 초과하면 현재 데이터가 기존 데이터를 덮어씁니다.
<b>계획 기록을 로드하기</b>	저장한 계획 기록을 로드합니다.
<b>지우기</b>	현재 패널에 표시된 모든 계획 기록 내용을 지웁니다.
<b>도움</b>	계획 기록에 관한 소개입니다.

또한 마우스 오른쪽 버튼으로 계획 기록 항목을 클릭하면 아래와 같은 옵션들이 나타날 것입니다:

<b>검색</b>	“검색” 버튼을 클릭하면 패널 하단에서 관련 검색 설정 사항이 나타날 것이며 유형에 따라 계획 기록 결과를 선택할 수 있습니다.
<b>모두 숨기기</b>	계획 기록에 있는 상위 모드만 표시하고 하위 모드를 모두 숨깁니다.

**선택된 계획 기록을 삭제하기**      선택된 계획 기록을 삭제합니다.

#### 5.1.2.9. 기타

이 패널에서 실행, 스텝 수집, 싱글래리티, 전역 JPs 제한 등 설정을 할 수 있습니다.

##### 실행 설정

##### 작업 흐름의 스텝별 진행 상황을 추정하기(로봇이 move ID를 반환할 수 없는 경우)

로봇은 move ID를 반환할 수 없고 프로젝트의 사이클 타임에 대한 요구 사항이 높은 경우 이 기능을 사용할 수 있습니다. 이런 경우, Mech-Viz는 현재 로봇의 포즈에 따라 작업 흐름의 진행 상황을 판단합니다. 하지만 이 기능을 사용하면 작업 흐름에서 인접한 이동과 비이동 스텝의 실행 시작 시간이 부정확해질 수 있습니다.

##### 비슷한 웨이포인트를 건너뛰기

두 웨이포인트가 매우 비슷하면(위치가 매우 가까우면) 이전 웨이포인트만 로봇에 전송합니다.

##### 기본적인 작업물 구성만 사용하기

이 옵션을 선택하면 물체의 레이블을 수신해도 작업물 구성의 기본값을 사용합니다.

##### 로봇 서비스 제한시간

이동 스텝 및 "제어권 이전" 스텝을 제외하고 Mech-Viz와 로봇 사이의 통신할 때의 제한 시간입니다. 기본값은 1000ms.

##### 스텝 수집

##### 수집된 이동 스텝의 최대 수

계획에 사용되는 이동 스텝의 최대 수량, 프로젝트에 사용되는 이동 스텝의 수가 설정된 수치를 초과하면 계획이 중단됩니다.

##### 수집된 비이동 스텝의 최대 수

계획에 사용되는 비이동 스텝의 최대 수량, 프로젝트에 사용되는 비이동 스텝의 수가 설정된 수치를 초과하면 계획이 중단됩니다.

##### 싱글래리티

로봇이 싱글래리티 근처에 있을 때 일부 직선 운동(MoveL) 명령어를 실행하지 못할 수도 있습니다. 소프트웨어가 전송한 운동 명령으로 인해 로봇의 로봇 싱글래리티 알람이 발생하지 않도록 하려면 소프트웨어 경로 계획 프로세스 중에 조기 경고를 제공하고 경로의 로봇 싱글래리티를 피하도록 노력해야 합니다.

\*싱글래리티 감지 방식\*에 아래와 두 가지 방식을 포함합니다:

- **로봇의 각 관절 속도를 감지하기**

관절의 최대 속도와 감속비를 설정함으로써 로봇 싱글래리티를 초래할 수 있는 계획 결과를 필터링합니다.

##### 관절 속도 상한

관절 속도가 이 수치보다 크면 싱글래리티로 간주될 수 있습니다.

##### 싱글래리티 속도 감속비

싱글래리티를 판단하기 위한 로봇 감속비의 역치, 로봇의 실제 감속비가 이 수치보다 크면 싱글래리티로 판단할 수 있습니다. 감속비 = 낮춰진 속도(또는 가속도) / 설정한 초기 속도(또는 가속도).

- 로봇의 관절 각도를 감지하기

로봇 관절과 각도 범위를 설정함으로써 로봇 싱글래리티를 초래할 수 있는 계획 결과를 필터링합니다.

#### 관절을 선택하기

로봇의 관절을 선택합니다.

#### 각도 하한

관절 각도가 이 수치보다 작으면 싱글래리티로 간주될 수 있습니다.

#### 각도 상한

관절 각도가 이 수치보다 크면 싱글래리티로 간주될 수 있습니다.

#### 전역 JPs 제한

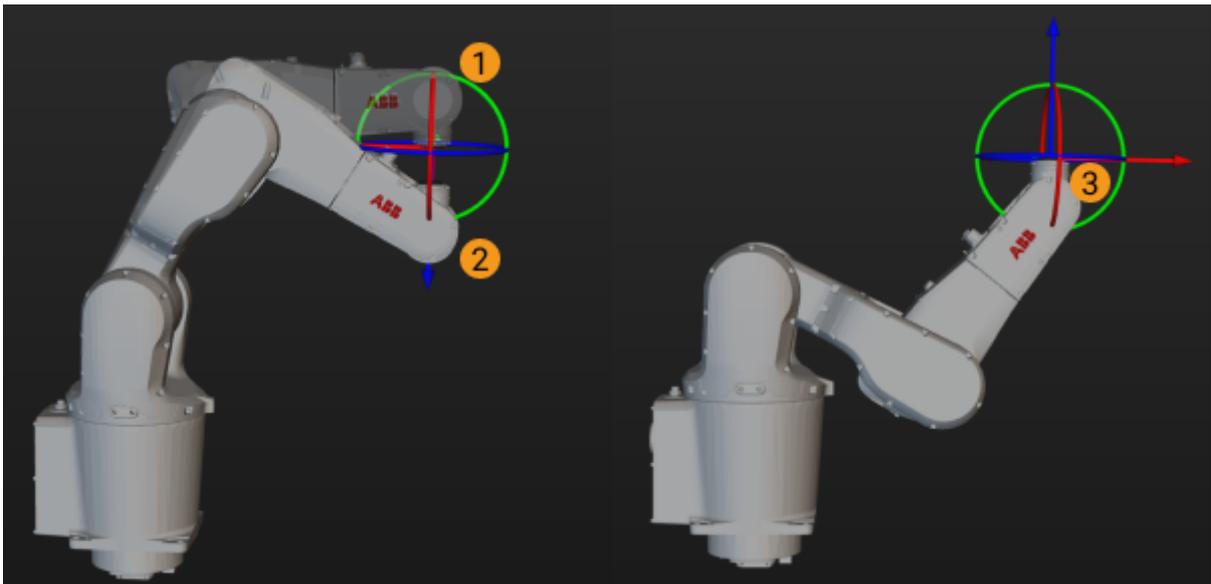


이동 스텝의 파라미터에 있는 "관절 각도 제한 조건"을 "Auto"로 설정하면 Mech-Viz는 이 설정 사항대로 실행하고 그렇지 않으면 이동 스텝의 "관절 각도 제한 조건"을 사용하여 실행합니다.

#### 어깨/팔꿈치/손목을 뒤척이지 않기

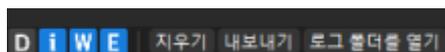
작업물을 잡고 있을 때 또는 실행 과정 전반 상태에서 불필요한 로봇의 뒤척임을 줄입니다.

어깨/팔꿈치/손목을 모두 "뒤척이지 않기"로 설정하는 것은 반드시 좋은 것이 아닙니다. 예를 들어:



로봇이 1에서 2로 이동하는 경우, "손목을 뒤척이지 않기"로 설정하면 로봇의 뒤척임이 오히려 더 커질 수 있고 로봇은 결국은 3의 상태가 될 수 있으므로 이런 경우에 "어깨를 뒤척이지 않기" 및 "팔꿈치를 뒤척이지 않기"만 설정하면 됩니다.

#### 5.1.2.10. 로그

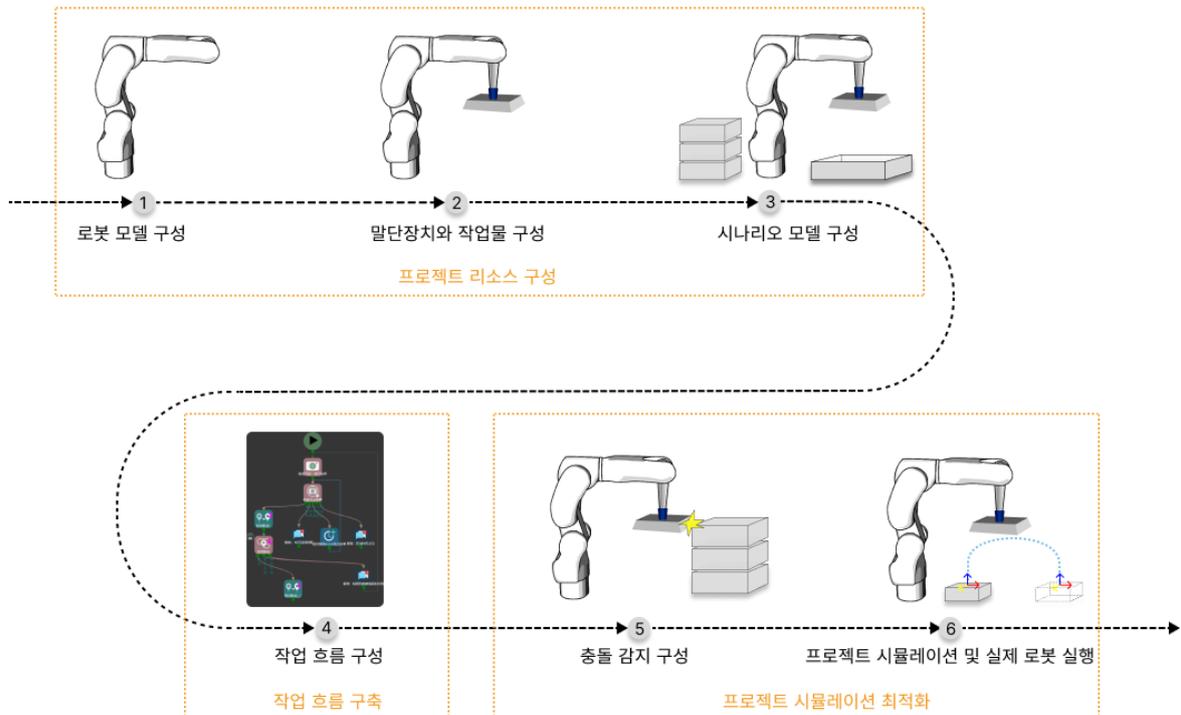


로그 패널에서 자세한 소프트웨어 실행 로그를 볼 수 있으며 로그의 관련 설정은 기능 패널 아래에 제공됩니다. 또한 하나의 로그 항목을 선택하면 **Ctrl + C**를 눌러 항목 내용을 복사할 수 있습니다.

D, i, W, E	설정한 로그 레벨에 따라 표시하려는 로그를 필터링합니다. 선택된 레벨의 로그 내용만 표시하며 여러 개를 동시에 선택할 수 있습니다.  D: 결함 수정(Debug) I: 정상 정보(Info) W: 경고(Warning) E: 오류(Error)
지우기	현재 패널에 표시된 로그를 지웁니다.
내보내기	현재 화면의 로그 정보를 HTML 형식으로 내보냅니다. 내보낸 파일은 Mech-Viz 설치 디렉터리 아래의 로그 폴더에 있습니다.
로그 폴더를 열기	Mech-Viz 설치 경로에 있는 ``logs`` 폴더를 엽니다. 로그 폴더에는 날짜별로 기록된 로그 파일이 있습니다.

### 5.1.3. 프로젝트 구축 방안

실제 프로젝트에서 Mech-Viz를 사용하여 프로젝트를 구축하는 일반적인 프로세스는 아래와 같습니다:



## 5.2. 프로젝트 리소스 구성

프로젝트 리소스에서 프로젝트를 구축할 때 필요한 기본적인 리소스(예: 로봇, 말단장치, 작업물, 시나리오 물체 등)를 구성할 수 있습니다.

### 프로젝트 리소스를 구성하는 방법

다음으로 각 프로젝트 리소스를 구성하는 방법에 대해 설명하겠습니다.

- **로봇**
- **말단장치**

- [공작물](#)
- [바닥](#)
- [시나리오 물체](#)

프로젝트 이름을 마우스 오른쪽 버튼으로 클릭한 후 팝업창에서 프로젝트 파일의 목록을 열 수 있고 자동 로드를 설정할 수 있으며 해당 프로젝트를 백업할 수 있습니다.

## 프로젝트 리소스 모델에 관한 설명

3D 시뮬레이션 공간에 있는 모든 기본적인 프로젝트 리소스(예: 로봇, 말단장치, 작업물, 시나리오 물체 등)에 대해 외부에서 모델을 도입해야 하면 \*시각화 모델\*과 \*충돌 모델\*을 모두 도입해야 합니다.

**시각화 모델**     3D 시뮬레이션 공간에만 표시되고 충돌 감지 과정에 참여하지 않습니다.

**충돌 모델**        충돌 감지만 참여하고 3D 시뮬레이션 공간에 표시되지 않습니다.

다른 프로젝트 리소스가 지원하는 모델 모맷은 다음 표를 참조하십시오:

		STL	OBJ	DAE	Binvx
로봇	시각화 모델	√	√	√	×
	충돌 모델	√	√	×	×
시나리오 물체	시각화 모델	√	√	√	×
	충돌 모델	√	√	×	×
말단장치	시각화 모델	√	√	√	×
	충돌 모델	×	√	×	×
공작물	시각화 모델	√	×	×	×
	충돌 모델	×	×	×	√

### 5.2.1. 로봇

이 부분에서는 로봇과 관련 구성에 대해 소개하겠습니다.

#### 소개

프로젝트를 구축하기 전에 우선 로봇 모델을 선택해야 하며 실제 작업 현장에서 사용되는 로봇과 일치하는 로봇 모델을 선택해야 합니다.

#### 로봇 모델

Mech-Viz의 로봇 라이브러리에 ABB, FANUC, KAWASAKAI, KUKA, UR, YASKAWA 등 다양한 브랜드의 로봇 모델이 내장되어 있고 [온라인 로봇 라이브러리](#)를 제공하며 온라인 로봇 라이브러리에서 원하는 로봇 모델을 다운로드 하고 [로봇 모델을 소프트웨어로 도입할 수 있습니다](#). 온라인 로봇 라이브러리에 필요한 로봇 모델이 없으면 [로봇 모델 제작 및 도입](#) 내용을 참조하십시오.

## 로봇 특성

Mech-Viz는 다른 로봇에 대한 특성 적응은 다르기 때문에 Mech-Viz가 로봇의 일부 기능 제어에 영향을 미칩니다. **로봇 특성**에서 세부 정보를 찾아볼 수 있습니다.

## 로봇 소프트 리미트

소프트 리미트를 설정하면 로봇 축의 운동 범위를 제한할 수 있습니다. 로봇 축의 운동 범위 내에 다른 장비가 있으면 로봇과 충돌할 수 있습니다. 비표준 로봇의 클램프와 로봇 파이프라인 패키지도 로봇 운동에 영향을 줄 수 있습니다. 위와 같은 상황이 발생하면 계획된 이동 경로에서 로봇이 도달할 수 없는 포즈를 피하기 위해 소프트 리미트를 미리 설정할 수 있습니다.

## 로봇 원점 위치

로봇 원점 위치는 사용자가 자체적으로 설정한 로봇의 초기 포즈입니다.

## 로봇 구성

로봇 패널에서 가상 로봇을 바꾸기, 로봇 소프트 리미트를 설정하기, 가상 로봇의 포즈를 편집하기, 로봇 원점 위치를 설정하기, 실제 로봇을 이동시키기, 로봇 모델을 소프트웨어로 도입하기 등 기능을 실현할 수 있습니다.

## 가상 로봇을 바꾸기

**시작 화면**에서 이미 선택한 로봇 모델을 바꾸려면 다음과 같이 작업하십시오:

1. [ **로봇 라이브러리** ]를 클릭하십시오.
2. 검색창에서 원하는 로봇의 모델을 입력하거나 선택 조건을 수정하여 로봇 모델을 선택합니다. 원하는 로봇 모델을 찾지 못하는 경우 다음과 같이 해결해 보십시오:
  - a. **온라인 로봇 라이브러리** 버튼을 클릭하여 필요한 로봇 모델을 다운로드합니다. 온라인 로봇 라이브러리에 원하는 로봇 모델을 찾을 수 없는 경우, **로봇 모델 생성 및 도입** 내용을 참조하십시오.
  - b. 로봇 모델 파일을 소프트웨어로 드래그하거나 **툴 > 로봇 라이브러리 도구 > 로봇을 도입하기**를 클릭하고 팝업 창에서 로봇 모델 파일을 선택합니다.
  - c. 다시 검색하십시오.
3. 로봇 카드를 더블클릭하거나 마우스를 로봇 카드로 이동하고 [ **선택** ] 버튼을 클릭합니다.

## 소프트 리미트를 설정하기

1. [ **소프트 리미트를 설정하기** ]를 클릭합니다.
2. 실제 수요에 따라 각 축의 최댓값과 최솟값을 설정하십시오.
3. [ **확인** ] 버튼을 클릭하십시오.



여기서 설정한 소프트 리미트는 Mech-Viz 소프트웨어에서 정의된 것이며 Mech-Viz 소프트웨어에만 적용됩니다. 여기서 한 설정은 로봇 티치 펜던트의 소프트 리미트에 영향을 미치지 않습니다.

## 가상 로봇의 포즈를 편집하기

관절 각도(JPs)를 통해 가상 로봇의 포즈를 편집하려면 다음 방법을 시도하십시오:

- 관절 각도 파라미터 패널에서 각 축의 슬라이더를 드래그하거나 오른쪽의 입력창에 값을 입력하면 가상 로봇이 조정에 따라 실시간으로 이동합니다.

- [ **JP**s 편집 ] 버튼을 클릭하여 각 관절 각도의 값을 입력한 후 [ **확인** ] 버튼을 클릭하십시오.

TCP 포즈를 통해 가상 로봇의 포즈를 편집하려면 다음 방법을 시도하십시오:

- TCP 포즈 파라미터 패널에서 오일러 각 또는 사원수를 조정하면 가상 로봇이 조정에 따라 실시간으로 이동합니다.
- [ **포즈 편집** ] 버튼을 클릭하여 각 TCP 포즈의 값을 입력한 후 [ **확인** ] 버튼을 클릭하십시오.
- 실제 로봇이 이미 연결되어 있는 경우 [ **TCP 포즈를 업데이트하기** ] 버튼을 클릭하면 가상 로봇은 실제 로봇의 TCP 포즈로 업데이트될 것입니다.

#### 로봇 원점 위치를 설정하기

1. **가상 로봇의 포즈를 수정하기** 내용을 참조하여 로봇 포즈를 수정합니다.
2. [ **로봇 원점 위치를 설정하기** ] 버튼을 클릭하십시오.



가상 로봇이 다른 포즈에 위치할 때 [ **가상 로봇을 원점 위치로 이동시키기** ] 버튼을 클릭하여 가상 로봇은 원점 위치로 즉시 이동할 것입니다.

#### 실제 로봇을 이동시키기

실제 로봇이 이미 연결되어 있는 경우 실제 로봇을 가상 로봇의 현재 포즈로 이동시킬 수 있습니다.

1. **가상 로봇의 포즈를 수정하기** 내용을 참조하여 로봇 모델의 포즈를 수정합니다.
2. 속도 파라미터를 5% 및 이하로 설정하십시오.
3. [ **실제 로봇을 이동하기** ] 버튼을 클릭하십시오.

#### 로봇 모델을 소프트웨어로 도입하기

온라인 로봇 라이브러리에서 로봇 모델 파일을 다운받거나 로봇 모델 파일을 자체적으로 만든 후 다음과 같이 작업하십시오:

- 직접 로봇 모델 파일을 소프트웨어로 그래프하여 도입하십시오.
- **툴 > 로봇 라이브러리 도구 > 로봇을 도입하기**를 클릭하고 팝업 창에서 로봇 모델 파일을 선택합니다.

## 5.2.2. 말단장치

이 부분에서는 로봇 말단장치 및 구성 방법을 소개하겠습니다.

### 소개

말단장치는 로봇이 작업을 완료할 수 있도록 특별히 설계되고 기계 끝쪽에 장착되는 장치(예: 클램프, 빨판)입니다.

### 말단장치 모델

3D 시뮬레이션 공간에서 말단장치를 표시하고 충돌 감지에 참여시키기 위해 말단장치의 시각화 모델과 충돌 모델을 만들어서 소프트웨어의 모델 라이브러리에 도입해야 합니다.

지원하는 말단장치 시각화 모델 및 충돌 모델의 포맷은 다음과 같습니다:

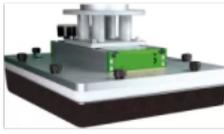
포맷	STL	OBJ	DAE	Binvox
시각화 모델	√	√	√	×
충돌 모델	×	√	×	×



- 소프트웨어에 사용되는 말단장치 충돌 모델의 각 부분이 반드시 볼록다면체(볼록 겹질)의 obj 모델이어야 합니다. [모델 편집기](#)를 사용하면 말단장치 모델을 위해 볼록다면체를 생성하고 obj 모델로 전환할 수 있습니다.
- 시각화 모델은 3D 시뮬레이션 공간에만 표시되고 충돌 감지에 참여하지 않습니다. 충돌 모델은 충돌 감제에 사용되고 3D 시뮬레이션 공간에 표시되지 않습니다.

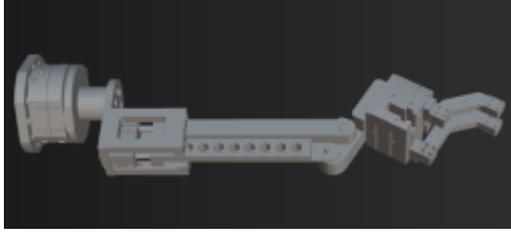
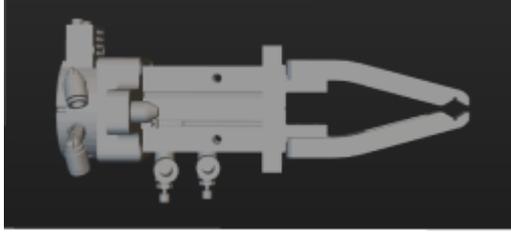
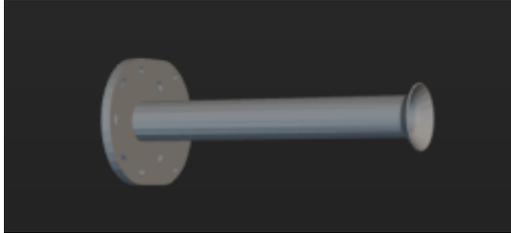
### 말단장치 유형

Mech-Viz에서 지원하는 말단장치의 유형은 일반적인 공구, 디팔레타이징 빨판, 측면 빨판, 어레이 그리퍼 등 있습니다.

말단장치 유형	설명	예시 그림
일반적인 공구	디팔레타이징 빨판 또는 어레이 그리퍼 이외의 모든 공구. 대표적인 공구 유형은 실린더 그리퍼, 싱글로드 빨판 등 있습니다.	
디팔레타이징 빨판	직사각형 디팔레타이징 빨판 공구로 다중 블록을 지원합니다.	
어레이 그리퍼	여러 개의 말단을 가지고 여러 개 말단이 동시에 작동할 수 있는 말단장치.	

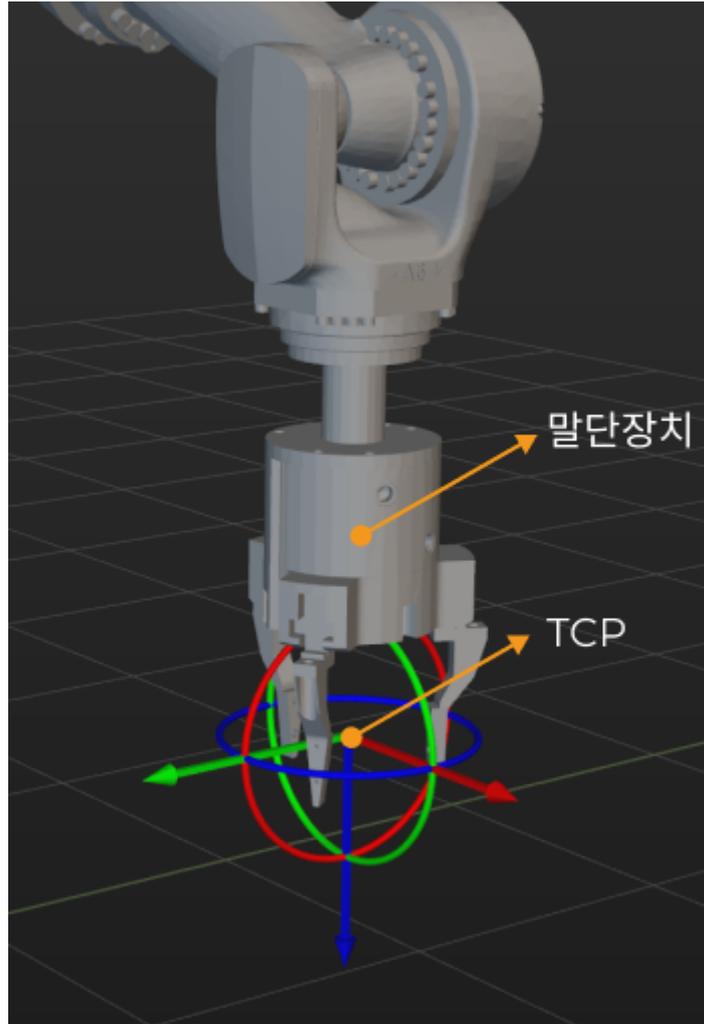
### 말단장치 대칭성

말단장치의 대칭성은 말단장치의 대칭축을 중심으로 일정 각도 회전 후 형상이 회전 전과 일치할 수 있는 특성을 나타내는 것입니다. 말단장치의 대칭성을 구성하면 피킹 및 배치 프로세스 중에 로봇의 말단 회전을 줄이고 계획 성공률을 높이고 계획 시간을 단축하고 로봇을 보다 원활하고 빠르게 움직일 수 있습니다.

회전 대칭이 없음	
180° 회전 대칭	
원형 대칭(임의각도 회전대칭)	

#### 공구중심점(TCP)

말단장치는 물체를 피킹하고 배치하는 데 사용되며 로봇을 공간의 어떤 지점으로 이동시킬 때 사실은 로봇의 TCP를 그 지점으로 이동시키는 것입니다. 기본적으로 TCP는 로봇의 말단에 있으므로 공구의 중심점(TCP)을 말단장치의 말단까지 조정하여 실제 피킹-배치 프로세스를 정확하게 시뮬레이션해야 합니다.



## 말단장치를 구성하기

말단장치의 충돌 모델과 시각화 모델을 도입하기

**프로젝트 리소스 > 모델 라이브러리** 패널의 [ + ] 버튼을 클릭하여 팝업 창에서 충돌 모델 파일과 시각화 모델 파일을 선택한 후 [ 열기 ] 버튼을 클릭하십시오.

로봇 말단장치 추가

**프로젝트 리소스 > 말단장치** 패널의 [ + ] 버튼을 클릭하여 말단장치 구성 창으로 들어갑니다.

1. **말단장치 이름** 텍스트 상자에서 자체 정의한 말단장치의 이름을 입력하십시오.
2. 실제 상황에 따라 **말단장치 유형** 파라미터를 설정하십시오.
3. **충돌 모델** 파라미터를 충돌 감지에 사용되는 말단장치 모델로 설정하십시오.
4. **시각화 모델** 파라미터를 3D 시뮬레이션 공간에 표시되는 시나리오 물체 모델로 설정하십시오. 시각화 모델의 위치와 크기는 실제 상황과 일치하지 않으면 **말단장치 모델의 위치나 크기를 조정하기** 내용을 참조하십시오.
5. 실제 상황에 따라 **회전 대칭** 파라미터를 **없음**, **N회 대칭** 또는 **원형 대칭\***으로 설정하십시오. \***N회 대칭\***으로 설정하면 **\*대칭 횟수** 파라미터도 설정해야 합니다.
6. 다음 방법을 통해 TCP를 설정하십시오.

<b>로봇을 통해 TCP를 업데이트하기</b>	이 버튼을 클릭하면 실제 로봇의 TCP 상태를 소프트웨어로 동기화할 수 있습니다.
<b>TCP 캘리브레이션</b>	로봇을 제어하여 특정 지점을 중심으로 회전하여 여러 세트의 플랜지 포즈를 기록한 다음 TCP를 계산합니다.
<b>TCP 파라미터를 수정하기</b>	오일러 각 또는 사원수의 각 파라미터를 수정합니다.
<b>정확한 TCP 포즈 수치를 사용하기</b>	다른 방법을 통해 정확한 포즈 데이터를 이미 획득하면 직접 포즈 편집 도구에 붙여넣을 수 있습니다.

7. 일반적인 장치를 사용하는 경우 컨트롤 로직을 구성할 필요가 없습니다. 빨판 또는 어레이 그리퍼를 사용하는 경우 **말단장치의 컨트롤 로직 구성** 내용을 참조하여 컨트롤 로직을 구성해야 합니다.

8. **[확인]** 버튼을 클릭하십시오.

필요하시면 위 작업을 반복하여 더 많은 말단장치를 추가할 수 있습니다.

#### 말단장치 삭제

다음 방법을 통해 말단장치를 삭제하십시오.

- **프로젝트 리소스 > 말단장치** 패널에서 말단장치 이름을 클릭한 후 **Delete** 버튼을 클릭하십시오.
- **프로젝트 리소스 > 말단장치** 패널에서 마우스 오른쪽 버튼으로 말단장치 이름을 클릭한 후 팝업창에서  버튼을 클릭하십시오.

#### 말단장치 수정

1. 다음 방법을 통해 말단장치의 구성 창구를 열어 주십시오.
  - **프로젝트 리소스 > 말단장치** 패널에서 말단장치 이름을 더블클릭하십시오.
  - menu:프로젝트 리소스 [말단장치] 패널에서 마우스 오른쪽 버튼으로 말단장치 이름을 클릭한 후 팝업창에서 **[말단장치 구성]** 버튼을 클릭하십시오.
2. 실제 수요에 따라 각 파라미터를 수정하십시오.
3. **[확인]** 버튼을 클릭하십시오.

#### 현재 작업 공구 설정

말단장치가 하나만 추가되면 해당 장치는 현재 작업 공구입니다. 말단장치가 여러 개 추가되면 추가된 첫 번째 것은 기본적으로 현재 작업 공구입니다.

현재 작업 공구를 바꾸려면 **프로젝트 리소스 > 말단장치** 에서 말단 공구를 클릭하고 팝업창에서 **[현재 작업 공구로 설정하기]** 를 클릭하십시오.

#### 말단장치 모델의 위치나 크기 조정

추가한 말단장치 모델의 위치가 올바르지 않을 때 다음과 같이 조정하십시오.

1. menu:프로젝트 리소스 [모델 라이브러리] 중의 말단장치 모델파일을 더블클릭하십시오.
2. 모델 변환의 편집창에서 포즈 파라미터를 수정하여 말단장치가 3D 시뮬레이션 공간에 있는 위치를 수정합니다.

추가한 말단장치 모델의 크기가 실제 상황에 부합하지 않을 때 다음과 같이 조정하십시오:

1. **프로젝트 리소스 > 모델 라이브러리** 중의 말단장치 모델 파일을 더블클릭하십시오.
2. 모델 변환의 편집창에서 다음 작업을 수행하십시오:
  - 모델 전체 스케일을 수정하려면 **스케일** 파라미터를 수정하십시오.
  - X, Y, Z 방향에서 스케일 비율을 별도로 수정하려면 **X,Y,Z에 같은 스케일을 응용하기** 옵션을 언체크하고 각 방향에서의 스케일 비율을 따로 설정하십시오.

#### 말단장치의 컨트롤 로직 구성

일반적인 장치를 사용하는 경우 컨트롤 로직을 구성할 필요가 없습니다. 빨판 또는 어레이 그리퍼를 사용하는 경우 실제 상황에 따라 컨트롤 로직을 구성해야 합니다. 말단장치 구성 창에서 [ **일반적인 빨판 구성** ] 또는 [ **어레이 그리퍼 구성** ]을 클릭하여 해당 구성기를 열어 주십시오.

**디팔레타이징 빨판** 상세한 작업 설명은 **일반적인 빨판 구성기** 내용을 참조하십시오.

**어레이 그리퍼** 상세한 작업 설명은 **어레이 그리퍼 구성기** 내용을 참조하십시오.

### 5.2.3. 작업물

이 부분에서는 작업물과 관련 구성에 대해 소개하겠습니다.

#### 소개

작업물은 상자, 금속 부품, 접착 또는 용접할 부품 등과 같은 로봇 말단장치의 작업 대상을 가리킵니다.

#### 작업물 대칭성

작업물의 대칭성은 작업물의 대칭축을 중심으로 일정 각도 회전 후 형상이 회전 전과 일치할 수 있는 특성을 나타내는 것입니다. 말단장치의 대칭성을 구성하면 피킹 및 배치 프로세스 중에 로봇의 말단 회전을 줄이고 계획 성공률을 높이고 계획 시간을 단축하고 로봇을 보다 원활하고 빠르게 움직일 수 있습니다.

상세한 정보는 **작업물 회전 대칭성** 내용을 참조하십시오.

#### 작업물 피킹 허용 편차

일부 작업물을 피킹할 때 로봇의 피킹 포즈가 일정한 각도 범위 내에서 유연하게 조절될 수 있으며 이 각도 범위는 피킹 허용 편차입니다.

상세한 정보는 **작업물 피킹 허용 편차** 내용을 참조하십시오.

#### 솔루션 선택 전략

##### 말단장치의 회전폭이 가장 작음

이 전략을 선택하면 “피킹-배치” 과정에서 말단장치 Z축의 회전폭이 가장 작은 점을 픽 포인트로 우선으로 선택할 것입니다. 이 전략은 물체를 잡은 후 로봇이 무의미하게 회전하여 공작물이 떨어지는 것을 방지할 수 있습니다.

##### 말단장치와 비전 포즈 사이의 차이가 가장 작음

이 전략을 선택하면 말단장치와 비전 포즈 사이의 각도 차이가 가장 작은 점을 픽 포인트로 우선으로 선택할 것입니다.

##### 말단장치와 포인트 클라우드 사이의 충돌이 가장 작음

이 전략을 선택하면 말단장치와 작업물의 포인트 클라우드 사이의 충돌 부피가 가장 작은 점을 픽 포인트로 우선으로 선택할 것입니다.

## 작업물 구성

### 작업물 추가

1. **프로젝트 리소스 > 작업물** 패널의 [ + ] 버튼을 클릭하여 작업물 구성 창을 엽니다.
2. **작업물 이름** 텍스트 상자에서 작업물 이름을 입력하십시오.



입력한 작업물 이름은 Mech-Vision에서 출력한 "labels" 필드 이름과 일치해야 합니다. Mech-Vision에서 다음 두 가지 방식으로 이름을 확인할 수 있습니다.

- 로그에서 “출력” 스텝의 상세한 내용을 통해 확인합니다. "labels":["XXX"]에 있는 “XXX”는 바로 작업물 이름입니다.
- “출력” 스텝의 **labels** 입력 포트의 데이터 스트림을 더블 클릭하면 오른쪽 디버그 출력 창에서 "다음표 안의 내용"이 바로 작업물 이름입니다.

3. 작업물의 실제 상황에 따라 **회전대칭** 파라미터를 **Z축 대칭**, **Y축 대칭** 또는 **X축 대칭** (여러 개 동선에 선택 가능) 으로 설정하고 **대칭 횟수** 및 \***시도 범위**\* 를 설정해야 합니다. 작업물이 회전 대칭성을 갖추지 않으면 이 파라미터를 설정할 필요가 없습니다. 작업물이 회전 대칭성을 갖추지 않으면 이 파라미터를 설정할 필요가 없습니다.
4. 작업물의 실제 상황에 따라 **피킹 허용 편차** 파라미터를 **Y축 대칭** 또는 **X축 대칭** (여러 개 동선에 선택 가능) 으로 설정하고 **대칭 횟수** 및 \***시도 범위**\* 를 설정해야 합니다. 작업물이 피킹 허용 편차를 응용하지 않으면 이 파라미터를 설정할 필요가 없습니다.
5. 피킹 전략에 대한 요구에 따라 **솔루션 선택 전략** 파라미터를 설정하십시오.
6. [ **확인** ] 버튼을 클릭하십시오.

필요가 있으면 위 단계를 반복으로 수행하여 더 많은 작업물을 추가하십시오.

### 작업물 삭제

다음 방법을 통해 작업물을 삭제하십시오.

- **프로젝트 리소스 > 작업물** 패널에서 작업물 이름을 클릭한 후 **Delete** 버튼을 클릭하십시오.
- **프로젝트 리소스 > 작업물** 패널에서 마우스 오른쪽 버튼으로 작업물 이름을 클릭한 후 팝업창에서 [ **삭제하기** ] 버튼을 클릭하십시오.

### 작업물 수정

1. 다음 방법을 통해 작업물 구성 창을 열어 주십시오.
  - **프로젝트 리소스 > 작업물** 패널에서 작업물 이름을 더블클릭하십시오.
  - **프로젝트 리소스 > 작업물** 패널에서 마우스 오른쪽 버튼으로 작업물 이름을 클릭한 후 팝업창에서 [ **작업물 구성** ] 버튼을 클릭하십시오.
2. 실제 수요에 따라 각 파라미터를 수정하십시오.
3. [ **확인** ] 버튼을 클릭하십시오.

#### 5.2.3.1. 작업물의 회전 대칭성

이 부분에서는 작업물의 회전 대칭성에 대해 소개하겠습니다.

## 소개

**i** 이 부분에서 말하는 "대칭성"은 모두 "회전 대칭성"을 가리킵니다.

작업물의 대칭성은 작업물의 대칭축을 중심으로 일정 각도 회전 후 형상이 회전 전과 일치할 수 있는 특성을 나타내는 것입니다. 작업물이 대칭성을 가진 경우 사용자는 특정 요구에 따라 작업물 구성에서 작업물의 대칭성과 관련된 파라미터를 설정할 수 있습니다. 작업물의 대칭성을 구성하면 피킹 및 배치 프로세스 중에 로봇의 말단 회전을 줄이고 계획 성공률을 높이고 계획 시간을 단축하고 로봇을 보다 원활하고 빠르게 움직일 수 있습니다.

## 작업물 대칭 유형

### 회전 대칭축을 결정하기

회전 대칭축은 **매칭 모델 및 픽 포인트 편집기**에서 설정한 기하학적 중심점이 위치하는 축입니다. 회전 대칭축은 유일하지 않습니다. 작업물이 배치되는 방식, 말단장치 유형 등 요소로 인해 Mech-Vision에서 동일한 작업물 기하학적 중심점의 설정 방식이 다양할 수 있으며 회전 대칭축의 결정에 영향을 미칠 수 있습니다.

일반적인 배치 방식에서 일부 일반적인 작업물의 대칭 유형은 다음과 같습니다. 사용하는 작업물의 회전 대칭축을 모르면 **매칭 모델 및 픽 포인트 편집기**에서 기하학적 중심점의 설정을 참조하십시오.

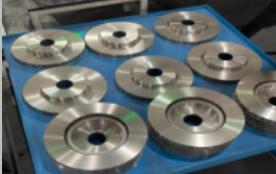
### 대칭성이 없는 작업물

다음 도표를 참조하십시오.

			
트랙슈	커넥터	차문 판금 부품	케이스

### Z축 대칭 작업물

다음 도표를 참조하십시오(일반적인 배치 방식).

			
브레이크 디스크	플랜지	기어	종이 상자

### X/Y축 대칭 작업물

다음 도표를 참조하십시오(일반적인 배치 방식).

			
질서정연하게 배치된 철강	무질서하게 쌓여 있는 철강	호스 커넥터	볼트

### 대칭 횡수

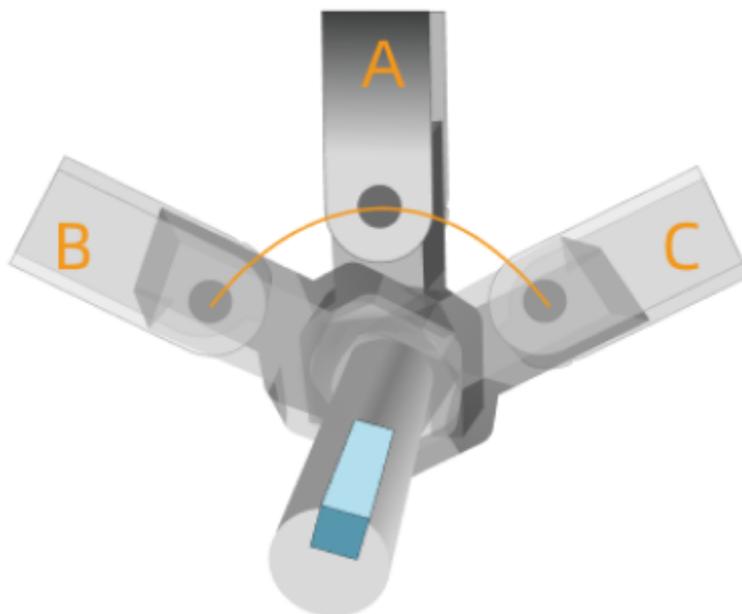
하나의 작업물이 대칭축을 중심으로 하여 일정한 각도( $a^\circ$ )로 회전했을 때 회전 전의 모양과 완벽하게 겹칠 수 있습니다. 대칭 횡수  $N = 360^\circ/a^\circ$ .

일반적으로 직육면체 종이 상자의 대칭 횡수  $N=2$ , 삼각기둥의 대칭 횡수  $N=3$ , 원기둥의 대칭 횡수  $N$ =양의 무한대  $\infty$  (소프트웨어에 있는 "원형 대칭"에 해당함)입니다.

			
$N = 2$	$N = 4$	$N = 9$	원형 대칭

### 시도 범위

아래 그림과 같이 B,C 사이의 각도 범위는 바로 시도 범위입니다.



A는 말단장치가 픽 포인트에 있을 때의 피킹 포즈이며, B와 C는 시도 범위 경계에 있을 때의 피킹 포즈입니다

시도 범위는 작업물의 배치 방식, bin의 모양 및 프로젝트 사이클 타임 등 요소에 따라 설정해야 합니다. 시도 범위를 매우 크게 설정하면 계획 속도를 낮출 수 있으며 매우 작게 설정하면 도달 가능한 픽 포인트를 놓칠 수 있습니다.

#### 시도 횟수

시도 횟수는 대칭 횟수와 시도 범위에 따라 자동으로 계산됩니다.

대칭 횟  $N = 10$ 이고 시도 범위가  $\pm 80^\circ$ 라고 가정하면 대칭 각도 간격은  $360^\circ/10 = 36^\circ$ 입니다. 편측 범위 내의 시도 횟수는  $80/36 = 2 \text{ mod } 8$ , 즉 2회(각각  $36^\circ$  및  $72^\circ$ )입니다. 총 시도 횟수는  $1+2*2 = 5$ 회(각각  $-72^\circ, -36^\circ, 0^\circ, 36^\circ, 72^\circ$ )입니다.

#### 5.2.3.2. 작업물 피킹 허용 편차

이 부분에서는 작업물의 피킹 허용 편차에 대해 설명하겠습니다.

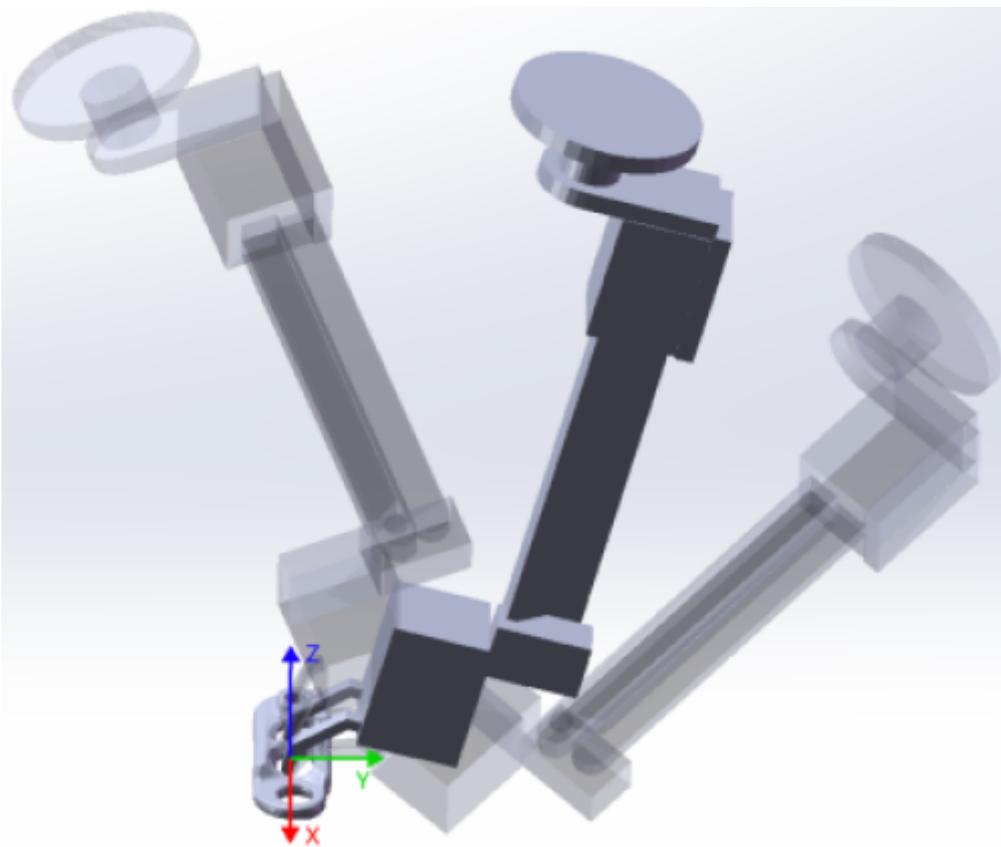
##### 소개

일부 작업물을 피킹할 때 로봇의 피킹 포즈가 일정한 각도 범위 내에서 유연하게 조절될 수 있으며 이 각도 범위는 피킹 허용 편차입니다. 피킹 허용 편차를 설정함으로써 로봇은 이러한 "허용 가능한 편차"를 적극적으로 활용하여 충돌 및 싱귤래리티와 같은 문제를 피할 수 있습니다.

배치 포즈가 "작업물 포즈"로 설정된 경우, 소프트웨어는 피킹 허용 편차를 적용할 때 피킹 시의 실제 포즈와 배치 포즈 사이의 각도를 자동으로 기록합니다. 물체를 배치할 때 소프트웨어는 이 각도에 따라 TCP 포즈를 조정하여 말단장치가 올바른 배치 포즈에 따라 물체를 내려놓도록 합니다.

##### 기준축을 확인하기

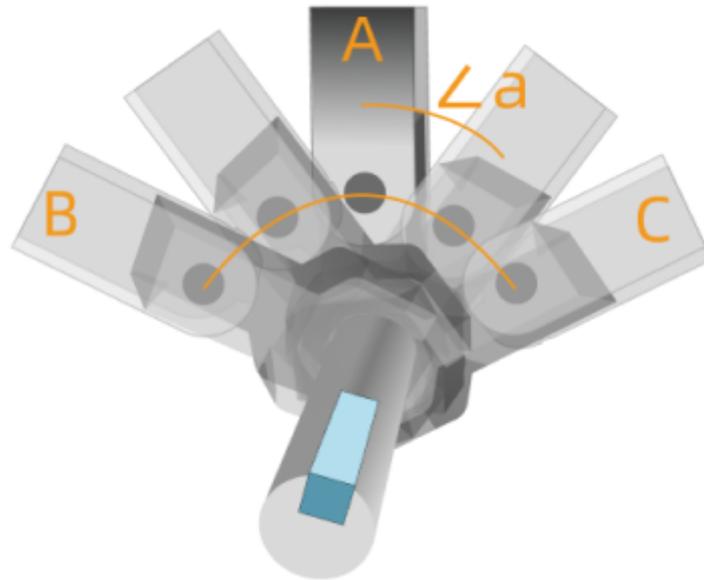
예를 들어 아래 그림에서 말단장치는 작업물 포즈 X축을 중심으로 하여 일정한 각도 범위 내에서 피킹 포즈를 유연하게 선택할 수 있으며 이때 기준축은 X축입니다.



### 허용 편차 범위 및 시도 간격

허용 편차 범위는 피킹 포즈와 비전 포즈 사이에서 각도 편차의 최대한 허용 범위입니다(아래 그림에서 B,C 사이의 각도는 허용 편차 범위입니다).

시도 간격은 한번 피킹을 시도할 때마다의 각도 간격을 가리킵니다(아래 그림 속의  $\angle a$ 는 시도 간격입니다).



A는 말단장치가 픽 포인트에 있을 때의 피킹 포즈이며, B와 C는 시도 범위 경계에 있을 때의 피킹 포즈입니다.

### 시도 횟수

시도 횟수는 허용 편차 범위와 시도 간격에 의해 자동으로 계산됩니다.

예를 들어 시도 간격이  $5^\circ$ , 허용 편차 범위  $\pm 10^\circ$ 인 경우, 한 쪽 범위 내의 시도 횟수는  $10/5 = 2$ 회(각각  $5^\circ, 10^\circ$ )입니다. 총 시도 횟수는  $1+2*2 = 5$ 회(각각  $-10^\circ, -5^\circ, 0^\circ, 5^\circ, 10^\circ$ )입니다.

## 5.2.4. 바닥

실제 작업 현장의 바닥을 시뮬레이션합니다. 기본적으로 바닥 평면은 로봇 베이스의 아랫면과 동일하며 실제 작업 현장의 바닥 높이에 따라 바닥 평면의 높이를 수정해야 합니다.

**바닥 높이를 조정하기**      마우스 오른쪽 버튼으로 **프로젝트 리소스 > 바닥**을 클릭하고 팝업창에서 슬라이드를 조정하거나 직접 수치를 수정하면 바닥의 높이를 조정할 수 있습니다.

**바닥 유형을 설정하기**      **설정 > 옵션 > 기본 설정 > 시뮬레이션 바닥의 패턴**에서 선호에 따라 바닥 패턴을 설정할 수 있습니다.

## 5.2.5. 시나리오 물체

이 부분에서는 시나리오에 있는 모든 물체와 관련된 구성에 대해서 소개하겠습니다.

## 소개

시나리오 물체란 로봇이 실제로 작업할 때 시나리오에 있는 각종 물체를 가리킵니다. 일반적으로 로봇 안전 펜스, 빈, 팔레트, 카메라 및 카메라 스탠드 등 물체들이 포함됩니다.

### 시나리오 물체 모델

시나리오 물체 모델은 주로 소프트웨어에서 실제 시나리오를 복원하고 충돌 감지 및 경로 계획 완료를 지원하는 데 사용됩니다. 소프트웨어에서 직육면체, 원기둥, 빈 모델을 생성할 수 있고 사용자가 자체 정의한 모델을 사용할 수도 있습니다. 지원하는 사용자 정의 모델의 시각화 모델과 충돌 모델의 포맷은 아래와 같습니다.

포맷	STL	OBJ	DAE	Binvov
시각화 모델	√	√	√	×
충돌 모델	√	√	×	×

### 시나리오 물체 사이의 "상하위 관계"

시나리오 물체 사이에서 상하위 관계를 형성할 수 있습니다. 상위 모델의 위치가 변하면 하위 모델도 함께 변하는 반면, 하위 모델의 위치가 변하면 상위 모델은 변하지 않습니다. 또한 상위 모델을 삭제하면 하위 모델도 함께 삭제될 것입니다.

### 시나리오 물체 구성

프로젝트 리소스 패널에서 시나리오 물체 모델을 추가, 수정, 삭제할 수 있습니다.

#### 시나리오 물체 추가

시나리오 모델은 **직육면체**, **원기둥**, **빈**인 경우 다음과 같이 작업하십시오:

1. menu:프로젝트 리소스 [시나리오 물체] 패널의 [ + ] 버튼을 클릭하여 시나리오 물체 구성 창을 엽니다.
2. **물체 설정** 화면에서 실제 상황에 따라 **시나리오 물체** 파라미터를 설정하십시오.
  - 직육면체인 경우, **X, Y, Z**를 설정해야 합니다.
  - 원기둥인 경우, **바닥면 반경, 높이**를 설정해야 합니다.
  - 빈인 경우, **X, Y, Z 두께와 빈 비전 포즈의 유효 범위**를 설정해야 합니다.
3. **물체 이름** 입력창에서 자체 정의한 시나리오 물체 모델의 이름을 입력하십시오.



- 현재 추가한 시나리오 물체 모델이 충돌 감지에 참여하지 않으려면 **충돌 감지에 참여하기** 옵션을 언체크하십시오.
- 현재 추가된 시나리오 물체 모델이 3D 시뮬레이션 공간에서 마우스로 선택되지 않으려면 **모델 선택 가능** 옵션을 언체크하십시오.

4. 새로 추가된 시나리오 물체 모델의 기본 위치는 로봇 기준 좌표계 원점입니다. [ **물체 포즈** ] 버튼을 클릭하고 해당 패널에 있는 각 파라미터를 수정하여 시나리오 물체 모델의 포즈를 조정하십시오.
5. [ **확인** ] 버튼을 클릭하십시오.

사용자 정의 모델인 경우 다음과 같이 작업하십시오:

1. menu:프로젝트 리소스 [모델 라이브러리] 패널의 [ + ] 버튼을 클릭하여 팝업 창에서 충돌 모델 파일과 시각화 모델 파일을 선택한 후 [ **열기** ] 버튼을 클릭하십시오.

2. menu:프로젝트 리소스 [시나리오 물체] 패널의 [ + ] 버튼을 클릭하여 시나리오 물체 구성 창을 엽니다.
3. 물체 설정 패널에서 시나리오 모델을 사용자 정의 모델로 설정하십시오.
4. 충돌 모델 파라미터를 충돌 감지에 사용되는 시나리오 물체 모델로 설정하십시오.
5. 시각화 모델 파라미터를 3D 시뮬레이션 공간에 표시되는 시나리오 물체 모델로 설정하십시오.
6. 물체 이름 입력창에서 자체 정의한 시나리오 물체의 이름을 입력하십시오.



- 현재 추가한 시나리오 물체 모델이 충돌 감지에 참여하지 않으려면 **충돌 감지에 참여하기** 옵션을 언체크하십시오.
- 현재 추가된 시나리오 물체 모델이 3D 시뮬레이션 공간에서 마우스로 선택되지 않으려면 **모델 선택 가능** 옵션을 언체크하십시오.

7. 새로 추가된 시나리오 물체 모델의 기본 위치는 로봇 기준 좌표계 원점입니다. [ 물체 포즈 ] 버튼을 클릭하고 해당 패널에 있는 각 파라미터를 수정하여 시나리오 물체 모델의 포즈를 조정하십시오.
8. [ 확인 ] 버튼을 클릭하십시오.



"사용자 정의 모델"의 "충돌 모델" 형식이 STL인 경우, 충돌 감지의 정확도를 향상시키기 위해 "사용자 정의 모델"을 덮을 수 있는 새로운 "직육면체" 또는 "원기둥"을 생성하는 것이 좋습니다.

더 많은 시나리오 물체 모델을 추가할 필요가 있으면 위 작업을 반복하십시오.



- 프로젝트 리소스에서 "모델 이름"을 선택한 후 [ + ] 버튼을 클릭하여 현재 모델의 하위 모델을 추가하면 하위 모델의 기본 초기 포즈가 상위 모델의 포즈와 일치합니다.
- 또한 시나리오 물체의 이름을 드래그하면 시나리오 물체 모델을 하위 모델 또는 독립적인 모델로 변환할 수 있습니다.

### 시나리오 물체 삭제

시나리오 물체를 삭제하려면 다음과 같이 작업하십시오.

- 3D 시뮬레이션 공간에서 시나리오 물체 모델을 선택하거나 **프로젝트 리소스 > 시나리오 물체** 패널에서 시나리오 물체의 이름을 클릭한 후 [ Delete ] 버튼을 클릭하십시오.
- 3D 시뮬레이션 공간에서 시나리오 물체 모델을 마우스 오른쪽 버튼으로 클릭하거나 **프로젝트 리소스 > 시나리오 물체** 패널에서 시나리오 물체 모델의 이름을 마우스 오른쪽 버튼으로 클릭한 후 팝업 창에서 [ 삭제하기 ] 버튼을 누르십시오.

### 시나리오 물체 수정

1. 다음 방법을 통해 해당 시나리오 물체의 구성 창을 열어 주십시오.
  - 3D 시뮬레이션 공간에 있는 물체 모델을 더블클릭하십시오.
  - **프로젝트 리소스 > 시나리오 물체** 패널에서 시나리오 물체 이름을 더블클릭하십시오.
  - **프로젝트 리소스 > 시나리오 물체** 패널에서 시나리오 물체 모델의 이름을 마우스 오른쪽 버튼으로 클릭한 후 팝업 창에서 [ 시나리오 물체 구성 ] 버튼을 누르십시오.
2. 실제 수요에 따라 각 파라미터를 수정하십시오.
3. [ 확인 ] 버튼을 클릭하십시오.



3D 시뮬레이션 공간에서 시나리오 물체 모델을 클릭하여 선택한 후 드래그 볼을 조정함으로써 드래그하여 시나리오 물체의 포즈를 조정할 수 있습니다.

### 빈의 비전 포즈의 유효한 범위를 설정하기

시나리오 물체 모델은 빈인 경우, **빈 설정**에서 비전 포즈의 유효한 범위를 설정해야 합니다.

- 빈이 시나리오 물체로만 사용되고 비전 포즈의 유효한 범위를 제한할 필요가 없는 경우 **포즈의 유효 범위를 설정하기** 파라미터를 언체크하십시오.
- 비전 포즈가 반드시 빈 안에 있어야 하는 경우 **빈 안에 있어야 함** 을 선택하십시오.
- 비전 포즈가 빈 상단을 초과해도 되는 경우 **빈 안에 있거나 빈 상단을 초과할 수 있음**을 선택하고 **빈 상단을 초과한 높이 범위** 파라미터를 설정하십시오.
- 포즈의 유효한 범위를 자체 정의하려면 **유효 범위를 자체 정의하기** 를 선택하고 **포즈의 유효 범위** 에 있는 각 파라미터를 설정하십시오.

## 5.3. 작업 흐름 구축

이 부분에서는 주로 스텝들을 사용하여 작업 흐름을 구축하는 방법에 대해 소개하겠습니다. 구체적으로 다음과 같습니다.

### 소개

프로그램의 실행 논리에 따라 다양한 스텝(step, 로봇 프로그래밍 기능을 실현하는 모듈)을 순서대로 연결하고 관련 파라미터를 조절함으로써 로봇을 컨트롤하는 작업 흐름을 구축합니다.

Mech-Viz 소프트웨어에서 다양하고 기능이 강한 스텝(step, 로봇 프로그래밍 기능을 실현하는 모듈)들이 내장되어 있으며 이런 스텝들을 사용하면 로봇 운동 컨트롤, 비전 시스템과 송수신, 비전 결과 처리, DI&DO 컨트롤, 논리적 토폴로지, 팔레타이징, 논리 판단, 피킹, 배치, 통신 등 다양한 작업 현장의 대표적인 수요를 충족할 수 있습니다.

### 기본 작업

작업	설명
스텝 라이브러리에서 스텝을 찾기	스텝 라이브러리의 검색창에서 키워드를 직접 입력하거나 스텝 클래스 분류를 통해 클래스별로 찾으십시오.
스텝을 작업 흐름 편집 영역으로 드래그하기	마우스 왼쪽 버튼을 누른 상태에서 스텝을 작업 흐름 편집 영역으로 드래그하십시오.
작업 흐름 편집 영역에 있는 스텝을 삭제하기	삭제하려는 스텝을 마우스 왼쪽 버튼으로 클릭하고 선택한 후 키보드에 있는 <b>Delete</b> 버튼을 누르거나 마우스 오른쪽 버튼으로 클릭하고 팝업창에서 <b>[ 삭제 ]</b> 버튼을 클릭하십시오.
스텝을 연결하기	마우스 포인터를 스텝의 아웃 포트 위치로 이동한 후 왼쪽 버튼을 길게 누른 상태에서 다른 스텝으로 드래그한 다음 놓습니다.
스텝 사이의 연결선을 삭제하기	삭제하려는 스텝 사이의 연결선을 마우스 왼쪽 버튼으로 클릭하고 선택한 후 키보드에 있는 <b>Delete</b> 버튼을 누르거나 마우스 오른쪽 버튼으로 클릭하고 팝업창에서 <b>[ 삭제 ]</b> 버튼을 클릭하십시오.
스텝에 대해 자체 정의한 설명을 추가하기	스텝을 마우스 오른쪽 버튼으로 클릭하고 팝업창에서 <b>[ 사용자 정의 설명 ]</b> 버튼을 클릭하십시오.

## 작업 흐름 구축 예시

1. 시뮬레이션된 로봇을 움직이게 하는 방법
2. 실제 로봇을 움직이게 하는 방법

### 5.3.1. 작업 흐름 구축

이 부분에서는 주로 스텝들을 사용하여 작업 흐름을 구축하는 방법에 대해 소개하겠습니다. 구체적으로 다음과 같습니다.

#### 소개

프로그램의 실행 논리에 따라 다양한 스텝(step, 로봇 프로그래밍 기능을 실현하는 모듈)을 순서대로 연결하고 관련 파라미터를 조절함으로써 로봇을 컨트롤하는 작업 흐름을 구축합니다.

Mech-Viz 소프트웨어에서 다양하고 기능이 강한 스텝(step, 로봇 프로그래밍 기능을 실현하는 모듈)들이 내장되어 있으며 이런 스텝들을 사용하면 로봇 운동 컨트롤, 비전 시스템과 송수신, 비전 결과 처리, DI&DO 컨트롤, 논리적 토폴로지, 팔레타이징, 논리 판단, 피킹, 배치, 통신 등 다양한 작업 현장의 대표적인 수요를 충족할 수 있습니다.

#### 기본 작업

작업	설명
스텝 라이브러리에서 스텝을 찾기	스텝 라이브러리의 검색창에서 키워드를 직접 입력하거나 스텝 클래스 분류를 통해 클래스별로 찾으십시오.
스텝을 작업 흐름 편집 영역으로 드래그하기	마우스 왼쪽 버튼을 누른 상태에서 스텝을 작업 흐름 편집 영역으로 드래그하십시오.
작업 흐름 편집 영역에 있는 스텝을 삭제하기	삭제하려는 스텝을 마우스 왼쪽 버튼으로 클릭하고 선택한 후 키보드에 있는 <b>Delete</b> 버튼을 누르거나 마우스 오른쪽 버튼으로 클릭하고 팝업창에서 [ <b>삭제</b> ] 버튼을 클릭하십시오.
스텝을 연결하기	마우스 포인터를 스텝의 아웃 포트 위치로 이동한 후 왼쪽 버튼을 길게 누른 상태에서 다른 스텝으로 드래그한 다음 놓습니다.
스텝 사이의 연결선을 삭제하기	삭제하려는 스텝 사이의 연결선을 마우스 왼쪽 버튼으로 클릭하고 선택한 후 키보드에 있는 <b>Delete</b> 버튼을 누르거나 마우스 오른쪽 버튼으로 클릭하고 팝업창에서 [ <b>삭제</b> ] 버튼을 클릭하십시오.
스텝에 대해 자체 정의한 설명을 추가하기	스텝을 마우스 오른쪽 버튼으로 클릭하고 팝업창에서 [ <b>사용자 정의 설명</b> ] 버튼을 클릭하십시오.

#### 작업 흐름 구축 예시

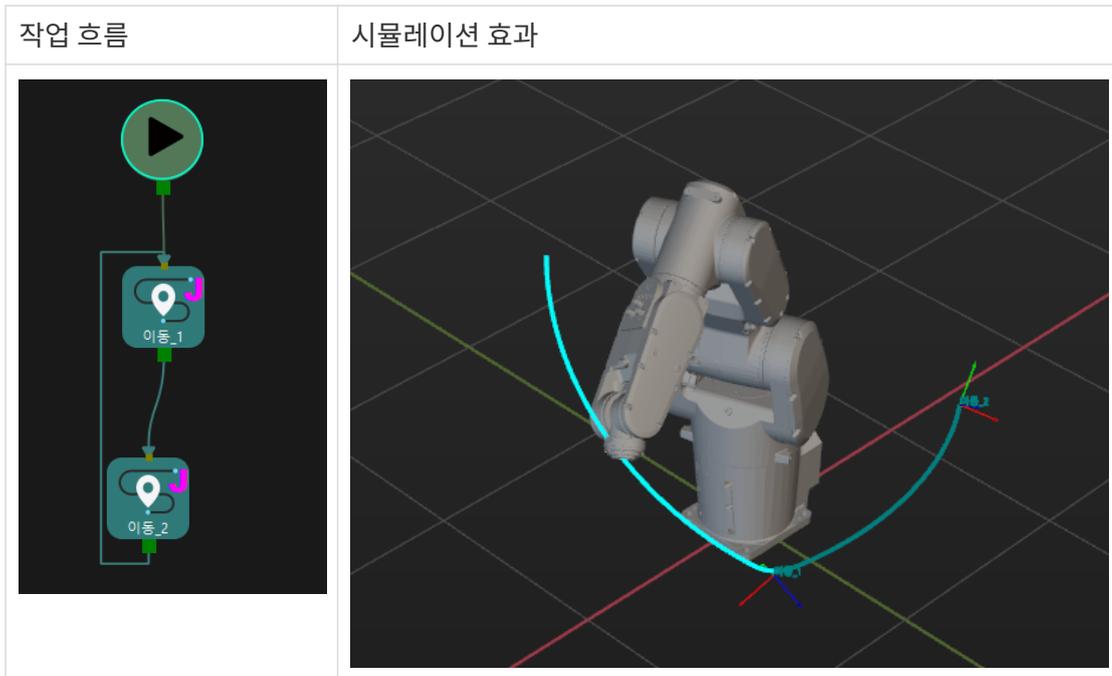
1. 시뮬레이션된 로봇을 움직이게 하는 방법
2. 실제 로봇을 움직이게 하는 방법

### 5.3.2. 시뮬레이션된 로봇을 움직이게 하는 방법

시뮬레이션된 로봇을 움직이게 하려면 로봇을 위해 이동 경로를 계획해야 합니다. “경로”는 수많은 “웨이포인트”로 구성되기 때문에 웨이포인트를 추가하면 경로를 계획할 수 있습니다. “이동” 스텝을 사용하여 웨이포인트를 추가하고 설정할 수 있습니다.

● 작업 프로세스 설명

1. Mech-Viz를 시작하고 원하는 로봇 모델을 선택하여 새로운 프로젝트를 만듭니다.
2. 스텝 라이브러리에서 "이동" (이동1)스텝을 찾아 편집 영역으로 드래그합니다.
3. 오른쪽의 "JPs" 파라미터 패널에서 현재 초기 상태에서 시뮬레이션된 로봇의 각 관절 각도 값을 볼 수 있습니다. 원하는 대로 JPs(관절 각도)의 값을 수정하면(수정 범위가 조금 커도 됨) 시뮬레이션된 로봇의 포즈에서 해당 변화를 볼 수 있습니다. 이때 수정한 후의 포즈는 웨이포인트로 간주될 수 있습니다.
4. 시작 버튼(작업 흐름 패널에 있는 삼각형 아이콘)을 “이동” 스텝과 연결하여 [ 시뮬레이션 ] 버튼을 클릭하면 시뮬레이션된 로봇은 이전 단계에서 설정한 웨이포인트 위치로 이동하는 것을 볼 수 있습니다.
5. 웨이포인트를 계속해서 추가할 수 있습니다. “이동 “(이동2) 스텝을 선택하고 작업 흐름 편집 영역으로 드래그하여 원하는 대로 JPs 값을 수정합니다. “이동1” 스텝의 아웃 포트를 “이동2” 스텝의 인포트와 연결하고 [ 시뮬레이션 ] 버튼을 클릭하면 시뮬레이션된 로봇은 이전 웨이포인트 위치에서 다음 웨이포인트 위치로 이동하는 것을 볼 수 있습니다.
6. “이동2” 스텝의 아웃 포트를 “이동1” 스텝의 인포트와 연결하고 [ 시뮬레이션 ] 버튼을 클릭하면 시뮬레이션된 로봇은 두 웨이포인트 사이에서 순환적으로 이동하는 것을 볼 수 있습니다. [ 정지 ] 버튼을 클릭하면 로봇의 운동을 정지할 수 있습니다.



이로써 가장 간단한 로봇 경로 계획 방법을 소개했습니다.

### 5.3.3. 실제 로봇을 움직이게 하는 방법

실제 로봇을 움직이게 하려면 우선 소프트웨어와 로봇 사이의 통신을 실현해야 합니다. 즉 로봇이 소프트웨어에서 전송하는 명령어를 수신하도록 합니다. 이 예시에서 사용하는 통신 방식은 마스터 컨트롤이기 때문에 **마스터 컨트롤 통신** 내용을 참조하여 해당 로봇의 통신 구성을 완료하십시오.

통신 구성이 완료되었는지 확인한 후 다음 단계는 로봇의 웨이포인트를 설정하는 것입니다. "이동" 스텝을 편집 영역으로 드래그하고 "JPs" 파라미터(**안전한 작업 범위 내에서 수정하는 것을 권장함**)를 수정합니다. 이때 시뮬레이션된 로봇의 포즈는 실제 로봇의 웨이포인트에 해당합니다. 마지막으로 시작 버튼(작업 흐름 패널에 있는 삼각형 아이콘)을 “이동1” 스텝과 연결합니다.



- 실제 로봇을 작동하기 전에 로봇의 속도와 가속도를 우선 낮춰야 합니다. **툴 바에 있는 “속도”와 “가속도”를 모두 “5%”로 낮추십시오.**
- 로봇이 움직이고 있을 때 비상시 터치 펜던트의 비상정지 버튼을 제때 눌러주십시오.

[ 실행 ] 버튼을 클릭하면 로봇은 “이동1” 스텝과 해당하는 포즈에 이동할 것입니다.

이로써 Mech-Viz를 사용하여 가장 간단한 운동을 완료하기 위해 로봇을 컨트롤하는 방법을 소개했습니다.

## 5.4. 스텝 라이브러리

스텝의 파라미터에 관한 상세한 설명은 해당 링크를 클릭하여 확인하십시오.

이동	<a href="#">동적 이동</a>
	<a href="#">이동</a>
	<a href="#">배열대로 이동</a>
	<a href="#">순서대로 이동</a>
	<a href="#">외부 이동</a>
	<a href="#">상대적인 이동</a>
DI DO	<a href="#">DI 체크</a>
	<a href="#">DI 리스트 체크</a>
	<a href="#">DO 설정</a>
	<a href="#">DO 리스트 설정</a>
	<a href="#">DI 기다리기</a>
논리적 토폴로지	<a href="#">길표지 분기</a>
	<a href="#">메시지 분기</a>
	<a href="#">프로시저</a>
	<a href="#">프로시저 아웃 포트</a>
	<a href="#">길표지를 설정하기</a>
팔레타이징	<a href="#">자체 정의한 파렛트 패턴</a>
	<a href="#">혼합 팔레타이징</a>
	<a href="#">미리 설정된 파렛트 패턴</a>
	<a href="#">비전 연속 팔레타이징</a>
	<a href="#">다수 피킹 팔레타이징</a>

로봇 유틸리티	말단장치를 체크하기
	페이로드 설정
	말단장치를 바꾸기
	피킹 상태 설정
	관절 각도를 획득하기
	제어권 이전
서비스	알림
도구	분류
	카운터
	완료 여부 확인
	인덱스 변경
	리셋
	기다리기
경로	경로 프로시저
비전	비전 결과 체크
	파렛트 포즈 업데이트
	피킹된 물체를 업데이트하기
	시나리오 물체 업데이트
	비전 결과를 모두 사용했음
	비전 인식
	비전 이동
기타	작업 흐름 종료

## 5.4.1. 이동

### 5.4.1.1. 동적 이동

#### 기능 설명

지난 “비전 인식” 결과에서 가장 높은 작업물 포즈에 따라 이동하여 카메라와 대상 물체와의 거리가 적당하도록 합니다.

#### 응용 시나리오

일반적으로 TIH 모드에 사용됩니다.

## 파라미터 설명

### 이동 스텝 기본 파라미터

#### 웨이포인트를 전송하기

기본적으로 선택되어 있으며 로봇 등 수신자에 웨이포인트를 전송합니다. 선택하지 않으며 웨이포인트를 전송하지 않지만 해당 웨이포인트는 여전히 경로 계획 중에 있습니다.

#### 후속 비이동 명령을 원활성있게 수행하기

기본적으로 선택되어 있지 않습니다. **이동 스텝** 사이에 **비이동 스텝**(예: "비전 인식", "DI 체크" 등)이 연결되면 로봇의 경로 계획을 방해하고 실제 로봇이 실행 중일 때 짧은 일시 중지가 발생하여 로봇이 원활하지 않게 실행됩니다. 이 옵션을 선택하면 현재 이동 스텝의 실행 종료를 기다릴 필요가 없고 계속해서 작업 흐름에 따라 실행할 수 있습니다. 이렇게 하면 로봇 실행 과정에서 자주 멈추는 문제를 피할 수 있고 로봇 동작의 원활성을 높일 수 있지만 스텝이 너무 일찍 종료될 수 있습니다.



스텝은 일찍 종료되는 이유가 무엇입니까?

Mech-Viz 소프트웨어가 실행 중일 때 동시에 로봇에 여러 포즈를 보냅니다. 소프트웨어는 로봇에 보낸 마지막 포즈가 로봇이 반환한 관절 각도와 동일인지 여부만 판단하고 동일하면 로봇이 마지막 위치에 이미 도달했다는 것으로 간주합니다. 예를 들어 경로에는 10개의 이동 스텝으로 구성되고 이동 스텝 5의 포즈는 마지막 이동 스텝의 포즈와 동일합니다. 로봇이 느린 속도로 이동할 때 웨이포인트 5로 이동하고 현재 관절 각도를 Mech-Viz로 보냅니다. 이동 스텝 5의 포즈는 마지막 이동 스텝의 포즈와 동일하기 때문에 Mech-Viz 소프트웨어는 경로 중의 모든 스텝 실행이 이미 완료된 것으로 잘못 판단하여 일찍 명령을 종료합니다.

#### 배치된 물체와의 충돌을 감지하지 않기

기본적으로 선택되어 있지 않습니다. 즉 이미 배치된 물체와 사이의 충돌을 감지합니다. 이 옵션을 선택하면 로봇, 말단장치 및 배치된 물체 사이의 충돌을 감지하지 않습니다.

팔레타이징 응용 시나리오에 다음과 같은 두 가지 경우가 있습니다.

1. 팔레타이징 시나리오에서 상자를 쌓을 때 로봇 자체가 이미 배치된 상자와 접촉할 경우가 있습니다(상자가 압착되거나 변형되지 않음). 이 충돌을 감지하면 Mech-Viz는 충돌을 피하기 위해 다른 배치 포인트를 계획하기 때문에 오히려 파レット를 완전히 채우지 못하게 됩니다.
2. 일반적으로 진공 그리퍼의 TCP는 그리퍼 표면이 아닌 모델 내부에 설정되어 있으며, 이로 인해 물체를 피킹할 때 말단장치와 피킹된 상자 모델이 중첩되는 상황이 발생합니다(소프트웨어는 말단장치와 작업물 사이의 충돌을 감지하지 않음). 로봇은 상자를 놓고 배치한 후 배치된 상자 모델은 시나리오 모델이 되며 이때 소프트웨어는 말단장치와 시나리오 속의 상자 모델과의 충돌을 감지하여 소프트웨어에서 충돌 경보가 발생하여 팔레타이징 스텝을 완료할 수 없습니다.

이 파라미터를 선택하면 소프트웨어는 로봇, 말단장치와 배치된 작업물 사이의 충돌을 감지하지 않고 위 문제를 피할 수 있습니다.

#### 포인트 클라우드 충돌 감지 모드

작업 현장의 실제 상황에 따라 파라미터를 설정하며 일반적으로 **Auto\***를 사용하면 됩니다. **로봇이 물체를 피킹하기 전의 이동 스텝은 \*NOTCHECK** 모드, 물체를 피킹한 후의 이동 스텝은 **CHECK** 모드를 선택할 수 있습니다.

**Auto** 기본값. "비전 이동" 스텝 및 "비전 이동"에 의존하는 "상대적인 이동" 스텝에 대해서만 포인트 클라우드와의 충돌을 감지하고 다른 이동 스텝에 대해 감지하지 않습니다.

**NOTCHECK** 모든 이동 스텝에 대해 포인트 클라우드와의 충돌을 감지하지 않습니다.

## CHECK 모든 이동 스텝에 대해 포인트 클라우드와의 충돌을 감지합니다.



**충돌 감지 > 충돌 감지 구성 > 포인트 클라우드와 다른 물체의 충돌을 감지하기** 기능을 활성화하면 Mech-Viz 소프트웨어가 경로 계획을 수행할 때 로봇 모델, 말단장치와 포인트 클라우드 사이의 충돌을 감지합니다. 일반적으로 포인트 클라우드 충돌 감지 설정은 로봇이 피킹 과정에서 작업물과의 충돌 여부를 확인하는 것입니다. 공간에 노이즈가 있으면 소프트웨어가 물체 피킹 전의 경로를 계획할 때 로봇 모델, 말단장치 모델이 노이즈와 접촉하므로 포인트 클라우드 사이의 충돌로 잘못 감지하여 소프트웨어의 계획 오류를 초래할 수 있습니다.

### 물체의 대칭성을 사용하지 않기

이 파라미터는 \*웨이포인트 유형\*을 \*작업물 포즈\*로 설정한 경우에만 적용됩니다. 예를 들어 웨이포인트 유형을 작업물 포즈로 설정한 이동/팔레타이징 스텝에 적용되며 웨이포인트 유형이 JPs 또는 TCP 포즈인 이동 스텝에는 적용되지 않습니다.

**None** 기본값. 모든 축의 대칭성을 사용합니다.

**AxisZ** Z축의 대칭성만 사용하지 않습니다.

**AxisXy** X, Y축의 대칭성을 사용하지 않습니다.

**All** 모든 축의 대칭성을 사용하지 않습니다.

대칭성을 사용하지 않으면 로봇은 물체 포즈에 따라 물체를 정확하게 배치합니다.



하지만 일부 특수한 상황에서 물체를 피킹하지 못할 때 **작업물 > 작업물 구성** 중의 \*회전 대칭\*을 설정할 수 있습니다. 인식된 물체에 대해 물체 대칭성을 설정하고 대칭 각도에 따라 작업물이 여러 개의 후보 포즈를 가질 수 있습니다. Mech-Viz 소프트웨어는 물체 피킹을 계획할 때 기본 포즈를 피킹할 수 없는 경우 후보 포즈를 피킹할 수 있는지 여부를 시도합니다. 회전 대칭 설정을 기반으로 계산된 후보 포즈는 Mech-Vision에서 출력된 원시 포즈와 다르며 물체 배치 포즈의 일관성을 보장할 수 없습니다.

### 피킹된 물체의 충돌 감지 모드

#### 시나리오&로봇과의 충돌 감지를 하지 않기

기본적으로 선택되어 있지 않습니다. 이 옵션을 선택한 후에는 **피킹된 물체\*와 \*시나리오 모델, 로봇** 사이의 충돌을 감지하지 않으므로 소프트웨어 충돌 감지의 계산량을 줄이고 Mech-Viz의 계획 속도를 최적화하며 전반적인 사이클 타임을 향상시킵니다. 일반적으로 로봇이 물체를 피킹한 후의 1~2개 이동 스텝에서 사용합니다.

충돌 감지를 수행하지 않으면 충돌 발생의 위험이 있으므로 주의하여 사용하시기 바랍니다.



menu:충돌 감지 구성 [잡고 있는 물체 구성>잡고 있는 작업물과 다른 물체 사이의 충돌을 감지하기] 기능을 활성화하면 잡고 있는 작업물과 시나리오 모델&로봇 사이의 충돌을 감지합니다. 팔레타이징 시나리오에서 비전을 통해 계산된 상자 치수와 실제 치수 사이에는 밀리미터 수준의 편차가 있으면 피킹 과정에서 상자 사이에 마찰이 발생할 수 있으며 충돌이 발생하지 않습니다. 분명히 충돌하지 않는 일부 움직임의 경우, 이러한 충돌을 감지하면 소프트웨어 계산량이 증가하고 Mech-Viz 계획 시간이 늘어나며 사이클 타임이 낭비됩니다. 시나리오 물체와의 충돌을 감지하지 않는 기능을 활성화하면 팔레타이징 시나리오에서 피킹된 작업물과 이미 배치된 상자 간의 충돌 감지에 영향을 미치지 않습니다. 파렛트 밑에 시나리오 물체가 있을 때 이 기능을 사용하면 팔레타이징 솔루션 선택 실패 문제를 피할 수 있습니다.

#### 포인트 클라우드와의 충돌 감지를 하지 않기

기본적으로 선택되어 있지 않습니다. 선택하면 **피킹된 물체\*와 \*시나리오 속의 포인트 클라우드** 사이의 충돌을 감지하지 않고 소프트웨어의 충돌 감지 계산량을 줄여 Mech-Viz의 계획 속도를 향상시키며 전체적인 사이클 타임을 최적화할 수 있습니다. 또한 피킹된 물체와 포인트 클라우드 노이즈 사이의 충돌을 잘못 감지하는 문제를 피할 수 있습니다.



- menu:충돌 감지 구성 [잡고 있는 물체 구성>잡고 있는 작업물과 다른 물체 사이의 충돌을 감지하기] 기능과 **포인트 클라우드 구성 > 포인트 클라우드와 다른 물체의 충돌을 감지하기** 기능을 동시에 활성화하면 잡고 있는 작업물 모델과 시나리오 속의 포인트 클라우드 사이의 충돌을 감지합니다.
- Mech-Vision이 포인트 클라우드와 물체 모델의 정보를 Mech-Viz로 보내면 포인트 클라우드와 물체 모델이 맞춰집니다. 로봇이 물체를 피킹한 후 모델은 계획된 경로를 따라 이동하고 피킹된 작업물 모델과 포인트 클라우드 간의 충돌이 발생합니다.
- 피킹된 작업물 모델과 포인트 클라우드 사이에서 이미 존재한 잘못된 충돌을 피할 수 없는 경우에, 이러한 충돌을 감지하면 불필요한 계산 작업량이 증가하고 Mech-Viz의 계획 시간이 늘어날 수 밖에 없습니다.

### Z방향 오프셋(웨이포인트에 대한 가장 높은 작업물)

**기본값** 0

**조절 설명** 카메라가 EIH 모드로 장착되면 로봇이 작업물을 피킹할 때 카메라와 작업물 사이의 거리가 변하지 않아야 합니다. 이 파라미터가 설정되면 "동적 이동" 스텝을 실행할 때마다 **비전 인식** 스텝에서 획득한 가장 높은 포즈의 Z 값에 따라 "동적 이동" 스텝에 해당하는 웨이포인트의 Z 좌표(웨이포인트 Z 좌표 = Z 오프셋 + 가장 높은 작업물 포즈의 Z 좌표)를 조정합니다.

### 로봇 포즈 Z의 하한

**기본값** -10 m

**조절 설명** 실제 로봇 TCP가 도달할 수 있는 로봇 포즈 Z의 최소값입니다.

### 기본 이동 설정

“기본 이동 설정”은 이동 스텝을 사용할 때 조절해야 하는 중요한 파라미터이며 로봇이 웨이포인트 위치로 이동하는 과정의 **속도** 및 \*운동 방식\*을 컨트롤하는 데 사용됩니다.

### 피킹&배치

**지정하지 않음** 기본값

**피킹** “비전 이동” 전의 이동 스텝에 적용됩니다.

**배치** “비전 이동” 후의 이동 스텝에 적용됩니다.

**조절 설명** Mech-Viz 프로젝트 실행 논리를 검사하는 데 사용될 수 있습니다. 작업 현장의 실제 작업 프로세스에 따라 피킹한 다음에 배치한다는 순서를 기본 원칙으로 이동 스텝에 대해 피킹&배치를 설정합니다.

### 운동 방식

**관절 운동** 로봇의 실행 경로가 원호이기 때문에 더 원활하게 이동할 수 있어 이동 중에 싱귤래리티가 나타나기 쉽지 않음을 의미합니다. 관절 운동은 경로 계획의 정밀도에 대한 요구가 그리 높지 않고 로봇이 큰 범위에서 이동하는 시나리오에 적용됩니다.

**직선 운동** 로봇의 실행 경로가 직선이며 로봇 경로의 정밀도에 대한 요구가 높습니다. 용접, 접착제 도포, 피킹 등 경로 정밀도에 대한 요구가 상대적으로 높은 시나리오에 적용됩니다.

### 속도&가속도

로봇의 이동 속도를 결정합니다. 일반적으로 가속도 파라미터의 수치가 속도보다 작아야 합니다. 가속도 파라미터의 수치가 속도보다 클 때 로봇의 동작이 원활하지 않습니다.

- ❗ 피킹의 안정성을 위해 "비전 이동"과 전/후의 이동 속도를 낮게 설정해야 합니다.

### 회전 반경

**기본값** 50.00mm

**조절 설명** 일반적으로 조절할 필요가 없고 기본값을 사용합니다.

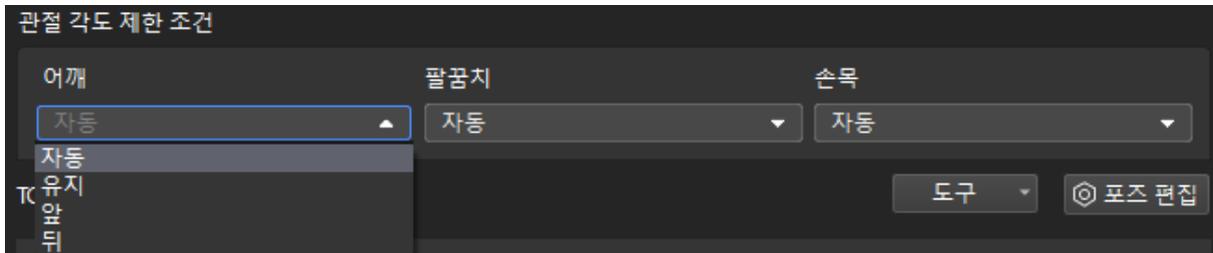
회전 반경은 웨이포인트와 로봇이 회전하기 시작하는 지점 사이의 거리를 나타냅니다. 회전 반경이 클수록 로봇의 이동이 더 원활해집니다. 로봇이 비교적 작은 공간에서 이동하는 경우 회전 반경을 더 작은 값으로 설정하십시오.

로봇이 넓은 공간에서 작업하고 다른 장애물이 없으며 로봇의 두 경로 사이의 거리가 멀었을 때, 회전 반경을 적절히 크게 조절하여 로봇의 움직임을 보다 원활하게 할 수 있습니다.

### 관절 각도 제한 조건

#### 개념

- 어깨** 손목 관절 중심과 Axis1의 상대적 위치 관계입니다. Axis1은 로봇 축1의 회전 중심 축을 나타냅니다.
- 팔꿈치** 손목 관절과 LowerArm 의 상대적 위치 관계입니다. LowerArm은 로봇의 축2, 축3의 회전 중심의 연결선을 나타냅니다.
- 손목** 손목 관절은 사실 로봇의 축5입니다. 축5 각도의 양/음의 관계는 손목이 뒤집히는 상태를 나타내고, Wrist center는 손목 관절의 중심을 나타냅니다.



#### 값 리스트

- 자동** 로봇 관절의 움직임에는 제한 조건이 없습니다. 최적의 솔루션은 각 축의 회전폭이 가장 작은 솔루션입니다.
- 유지** 현재 JPs 상태에 따라 다음 JPs 솔루션을 제한합니다. 예를 들어, 로봇 축3의 현재 JPs가 양수이면 로봇 축3의 JPs가 양수인 솔루션만 다음 웨이포인트로 고려됩니다.
- 앞** 손목 관절의 중심은 Axis1 앞에 있습니다.
- 뒤** 손목 관절의 중심은 Axis1뒤에 있습니다.



버튼을 클릭하면 현재 포즈에 대한 모든 JPs 솔루션을 표시하는 창이 나타납니다. 포즈 중 하나를 클릭하면 솔루션의 해당 로봇 포즈가 3D 시뮬레이션 영역에 표시됩니다. 따라서 아래와 같이 다양한 제한 조건에서 가능한 솔루션을 알 수 있습니다.

- ❗ 1. 관절 각도 제한 조건은 6축 로봇에만 적용되며 4축 로봇은 어깨, 팔꿈치, 손목을 뒤척이지 않습니다.

2. “상대적인 이동”, “미리 설정한 파レット 패턴”, “자체 정의한 파レット 패턴” 스텝에서 이 파라미터를 설정할 수 없으며 기본적으로 로봇의 어깨, 팔꿈치 및 손목이 변경되지 않습니다. 즉, 이러한 스텝에서 로봇이 싱글래리티를 거치지 않도록 제한합니다.

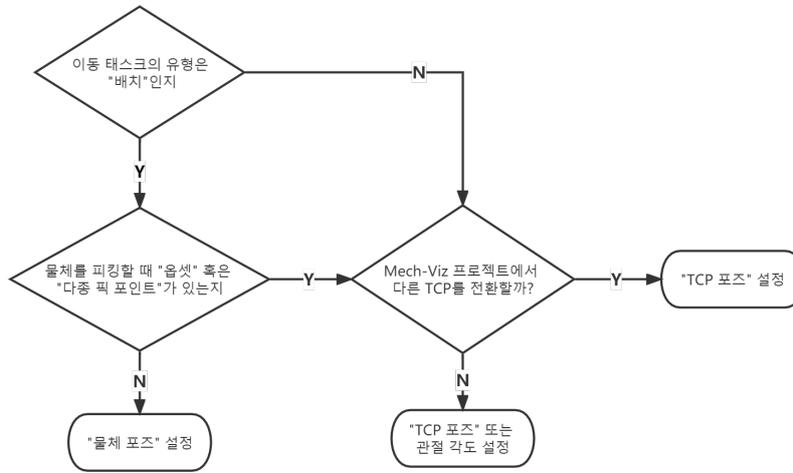
### 웨이포인트 유형

**TCP** 웨이포인트는 TCP 좌표계의 X, Y, Z 값과 오일러 각/사원수로 표시됩니다.

**JPs** 웨이포인트는 로봇 각 관절의 수치로 표시됩니다.

**작업물 포즈** 웨이포인트는 물체 좌표계의 X, Y, Z 값과 오일러 각/사원수로 표시됩니다.

웨이포인트 유형을 선택할 때 판단 방법은 다음 그림과 같습니다.



**포즈 편집** 포즈를 직접 편집하고 복사/붙여넣을 수 있으며 사원수와 오일러 각 두 가지 형식을 지원합니다.

**포즈 변환** 자체 정의를 통해 현재 포즈를 새 포즈로 변환하는 데 사용되며 포즈 파인튜닝에 적용됩니다.

**포즈 보정** 작업물의 좌표계를 계산하는 ABB 로봇 삼점법과 유사하며 대상 물체가 회전할 때 포즈를 확인하기 쉽지 않은 시나리오에 적합합니다. 예를 들어 기울어진 직육면체의 회전 포즈를 계산하기 어려운 경우, 포즈 보정을 통해 직육면체의 회전 포즈를 먼저 계산하고 로봇이 계산된 포즈에 따라 실행하도록 합니다.

**JPs 편집** 포즈를 편집하는 기능과 유사하여 관절 각도를 복사/붙여넣을 수 있고 라디안과 각도 두 가지 단위를 지원합니다.

 가상 로봇을 설정된 웨이포인트 위치로 이동합니다.

 배치된 로봇의 포즈를 읽어 웨이포인트로 설정합니다.

 현재 웨이포인트에 대한 모든 JPs 솔루션을 표시합니다.

### 5.4.1.2. 이동

#### 기능 설명

로봇 이동 경로에서의 목표 포즈 및 목표 포즈로 이동하는 방식을 설정합니다.

## 파라미터 설명

### 이동 스텝 기본 파라미터

#### 웨이포인트를 전송하기

기본적으로 선택되어 있으며 로봇 등 수신자에 웨이포인트를 전송합니다. 선택하지 않으며 웨이포인트를 전송하지 않지만 해당 웨이포인트는 여전히 경로 계획 중에 있습니다.

#### 후속 비이동 명령을 원활성있게 수행하기

기본적으로 선택되어 있지 않습니다. **이동 스텝** 사이에 **비이동 스텝**(예: "비전 인식", "DI 체크" 등)이 연결되면 로봇의 경로 계획을 방해하고 실제 로봇이 실행 중일 때 짧은 일시 중지가 발생하여 로봇이 원활하지 않게 실행됩니다. 이 옵션을 선택하면 현재 이동 스텝의 실행 종료를 기다릴 필요가 없고 계속해서 작업 흐름에 따라 실행할 수 있습니다. 이렇게 하면 로봇 실행 과정에서 자주 멈추는 문제를 피할 수 있고 로봇 동작의 원활성을 높일 수 있지만 스텝이 너무 일찍 종료될 수 있습니다.



스텝은 일찍 종료되는 이유가 무엇입니까?

Mech-Viz 소프트웨어가 실행 중일 때 동시에 로봇에 여러 포즈를 보냅니다. 소프트웨어는 로봇에 보낸 마지막 포즈가 로봇이 반환한 관절 각도와 동일한지 여부만 판단하고 동일하면 로봇이 마지막 위치에 이미 도달했다는 것으로 간주합니다. 예를 들어 경로에는 10개의 이동 스텝으로 구성되고 이동 스텝 5의 포즈는 마지막 이동 스텝의 포즈와 동일합니다. 로봇이 느린 속도로 이동할 때 웨이포인트 5로 이동하고 현재 관절 각도를 Mech-Viz로 보냅니다. 이동 스텝 5의 포즈는 마지막 이동 스텝의 포즈와 동일하기 때문에 Mech-Viz 소프트웨어는 경로 중의 모든 스텝 실행이 이미 완료된 것으로 잘못 판단하여 일찍 명령을 종료합니다.

#### 배치된 물체와의 충돌을 감지하지 않기

기본적으로 선택되어 있지 않습니다. 즉 이미 배치된 물체와 사이의 충돌을 감지합니다. 이 옵션을 선택하면 로봇, 말단장치 및 배치된 물체 사이의 충돌을 감지하지 않습니다.

팔레타이징 응용 시나리오에 다음과 같은 두 가지 경우가 있습니다.

1. 팔레타이징 시나리오에서 상자를 쌓을 때 로봇 자체가 이미 배치된 상자와 접촉할 경우가 있습니다(상자가 압착되거나 변형되지 않음). 이 충돌을 감지하면 Mech-Viz는 충돌을 피하기 위해 다른 배치 포인트를 계획하기 때문에 오히려 파レット를 완전히 채우지 못하게 됩니다.
2. 일반적으로 진공 그리퍼의 TCP는 그리퍼 표면이 아닌 모델 내부에 설정되어 있으며, 이로 인해 물체를 피킹할 때 말단장치와 피킹된 상자 모델이 중첩되는 상황이 발생합니다(소프트웨어는 말단장치와 작업물 사이의 충돌을 감지하지 않음). 로봇은 상자를 놓고 배치한 후 배치된 상자 모델은 시나리오 모델이 되며 이때 소프트웨어는 말단장치와 시나리오 속의 상자 모델과의 충돌을 감지하여 소프트웨어에서 충돌 경보가 발생하여 팔레타이징 스텝을 완료할 수 없습니다.

이 파라미터를 선택하면 소프트웨어는 로봇, 말단장치와 배치된 작업물 사이의 충돌을 감지하지 않고 위 문제를 피할 수 있습니다.

#### 포인트 클라우드 충돌 감지 모드

작업 현장의 실제 상황에 따라 파라미터를 설정하며 일반적으로 **Auto\***를 사용하면 됩니다. **로봇이 물체를 피킹하기 전의 이동 스텝은 \*NOTCHECK** 모드, 물체를 피킹한 후의 이동 스텝은 **CHECK** 모드를 선택할 수 있습니다.

**Auto** 기본값. "비전 이동" 스텝 및 "비전 이동"에 의존하는 "상대적인 이동" 스텝에 대해서만 포인트 클라우드와의 충돌을 감지하고 다른 이동 스텝에 대해 감지하지 않습니다.

**NOTCHECK** 모든 이동 스텝에 대해 포인트 클라우드와의 충돌을 감지하지 않습니다.

## CHECK 모든 이동 스텝에 대해 포인트 클라우드와의 충돌을 감지합니다.



**충돌 감지 > 충돌 감지 구성 > 포인트 클라우드와 다른 물체의 충돌을 감지하기** 기능을 활성화하면 Mech-Viz 소프트웨어가 경로 계획을 수행할 때 로봇 모델, 말단장치와 포인트 클라우드 사이의 충돌을 감지합니다. 일반적으로 포인트 클라우드 충돌 감지 설정은 로봇이 피킹 과정에서 작업물과의 충돌 여부를 확인하는 것입니다. 공간에 노이즈가 있으면 소프트웨어가 물체 피킹 전의 경로를 계획할 때 로봇 모델, 말단장치 모델이 노이즈와 접촉하므로 포인트 클라우드 사이의 충돌로 잘못 감지하여 소프트웨어의 계획 오류를 초래할 수 있습니다.

### 물체의 대칭성을 사용하지 않기

이 파라미터는 \*웨이포인트 유형\*을 \*작업물 포즈\*로 설정한 경우에만 적용됩니다. 예를 들어 웨이포인트 유형을 작업물 포즈로 설정한 이동/팔레타이징 스텝에 적용되며 웨이포인트 유형이 JPs 또는 TCP 포즈인 이동 스텝에는 적용되지 않습니다.

**None** 기본값. 모든 축의 대칭성을 사용합니다.

**AxisZ** Z축의 대칭성만 사용하지 않습니다.

**AxisXy** X, Y축의 대칭성을 사용하지 않습니다.

**All** 모든 축의 대칭성을 사용하지 않습니다.

대칭성을 사용하지 않으면 로봇은 물체 포즈에 따라 물체를 정확하게 배치합니다.



하지만 일부 특수한 상황에서 물체를 피킹하지 못할 때 **작업물 > 작업물 구성** 중의 \*회전 대칭\*을 설정할 수 있습니다. 인식된 물체에 대해 물체 대칭성을 설정하고 대칭 각도에 따라 작업물이 여러 개의 후보 포즈를 가질 수 있습니다. Mech-Viz 소프트웨어는 물체 피킹을 계획할 때 기본 포즈를 피킹할 수 없는 경우 후보 포즈를 피킹할 수 있는지 여부를 시도합니다. 회전 대칭 설정을 기반으로 계산된 후보 포즈는 Mech-Vision에서 출력된 원시 포즈와 다르며 물체 배치 포즈의 일관성을 보장할 수 없습니다.

### 피킹된 물체의 충돌 감지 모드

#### 시나리오&로봇과의 충돌 감지를 하지 않기

기본적으로 선택되어 있지 않습니다. 이 옵션을 선택한 후에는 **피킹된 물체\*와 \*시나리오 모델, 로봇** 사이의 충돌을 감지하지 않으므로 소프트웨어 충돌 감지의 계산량을 줄이고 Mech-Viz의 계획 속도를 최적화하며 전반적인 사이클 타임을 향상시킵니다. 일반적으로 로봇이 물체를 피킹한 후의 1~2개 이동 스텝에서 사용합니다.

충돌 감지를 수행하지 않으면 충돌 발생의 위험이 있으므로 주의하여 사용하시기 바랍니다.



menu:충돌 감지 구성 [잡고 있는 물체 구성>잡고 있는 작업물과 다른 물체 사이의 충돌을 감지하기] 기능을 활성화하면 잡고 있는 작업물과 시나리오 모델&로봇 사이의 충돌을 감지합니다. 팔레타이징 시나리오에서 비전을 통해 계산된 상자 치수와 실제 치수 사이에는 밀리미터 수준의 편차가 있으면 피킹 과정에서 상자 사이에 마찰이 발생할 수 있으며 충돌이 발생하지 않습니다. 분명히 충돌하지 않는 일부 움직임의 경우, 이러한 충돌을 감지하면 소프트웨어 계산량이 증가하고 Mech-Viz 계획 시간이 늘어나며 사이클 타임이 낭비됩니다. 시나리오 물체와의 충돌을 감지하지 않는 기능을 활성화하면 팔레타이징 시나리오에서 피킹된 작업물과 이미 배치된 상자 간의 충돌 감지에 영향을 미치지 않습니다. 파렛트 밑에 시나리오 물체가 있을 때 이 기능을 사용하면 팔레타이징 솔루션 선택 실패 문제를 피할 수 있습니다.

#### 포인트 클라우드와의 충돌 감지를 하지 않기

기본적으로 선택되어 있지 않습니다. 선택하면 **피킹된 물체\*와 \*시나리오 속의 포인트 클라우드** 사이의 충돌을 감지하지 않고 소프트웨어의 충돌 감지 계산량을 줄여 Mech-Viz의 계획 속도를 향상시키며 전체적인 사이클 타임을 최적화할 수 있습니다. 또한 피킹된 물체와 포인트 클라우드 노이즈 사이의 충돌을 잘못 감지하는 문제를 피할 수 있습니다.



- menu:충돌 감지 구성 [잡고 있는 물체 구성>잡고 있는 작업물과 다른 물체 사이의 충돌을 감지하기] 기능과 **포인트 클라우드 구성 > 포인트 클라우드와 다른 물체의 충돌을 감지하기** 기능을 동시에 활성화하면 잡고 있는 작업물 모델과 시나리오 속의 포인트 클라우드 사이의 충돌을 감지합니다.
- Mech-Vision이 포인트 클라우드와 물체 모델의 정보를 Mech-Viz로 보내면 포인트 클라우드와 물체 모델이 맞춰집니다. 로봇이 물체를 피킹한 후 모델은 계획된 경로를 따라 이동하고 피킹된 작업물 모델과 포인트 클라우드 간의 충돌이 발생합니다.
- 피킹된 작업물 모델과 포인트 클라우드 사이에서 이미 존재한 잘못된 충돌을 피할 수 없는 경우에, 이러한 충돌을 감지하면 불필요한 계산 작업량이 증가하고 Mech-Viz의 계획 시간이 늘어날 수 밖에 없습니다.

### 기본 이동 설정

“기본 이동 설정”은 이동 스텝을 사용할 때 조절해야 하는 중요한 파라미터이며 로봇이 웨이포인트 위치로 이동하는 과정의 **속도** 및 \***운동 방식**\*을 컨트롤하는 데 사용됩니다.

### 피킹&배치

<b>지정하지 않음</b>	기본값
<b>피킹</b>	“비전 이동” 전의 이동 스텝에 적용됩니다.
<b>배치</b>	“비전 이동” 후의 이동 스텝에 적용됩니다.
<b>조절 설명</b>	Mech-Viz 프로젝트 실행 논리를 검사하는 데 사용될 수 있습니다. 작업 현장의 실제 작업 프로세스에 따라 피킹한 다음에 배치한다는 순서를 기본 원칙으로 이동 스텝에 대해 피킹&배치를 설정합니다.

### 운동 방식

<b>관절 운동</b>	로봇의 실행 경로가 원호이기 때문에 더 원활하게 이동할 수 있어 이동 중에 싱귤래리티가 나타나기 쉽지 않음을 의미합니다. 관절 운동은 경로 계획의 정밀도에 대한 요구가 그리 높지 않고 로봇이 큰 범위에서 이동하는 시나리오에 적용됩니다.
<b>직선 운동</b>	로봇의 실행 경로가 직선이며 로봇 경로의 정밀도에 대한 요구가 높습니다. 용접, 접착제 도포, 피킹 등 경로 정밀도에 대한 요구가 상대적으로 높은 시나리오에 적용됩니다.

### 속도&가속도

로봇의 이동 속도를 결정합니다. 일반적으로 가속도 파라미터의 수치가 속도보다 작아야 합니다. 가속도 파라미터의 수치가 속도보다 클 때 로봇의 동작이 원활하지 않습니다.

- ❗ 피킹의 안정성을 위해 "비전 이동"과 전/후의 이동 속도를 낮게 설정해야 합니다.

### 회전 반경

<b>기본값</b>	50.00mm
<b>조절 설명</b>	일반적으로 조절할 필요가 없고 기본값을 사용합니다.  회전 반경은 웨이포인트와 로봇이 회전하기 시작하는 지점 사이의 거리를 나타냅니다. 회전 반경이 클수록 로봇의 이동이 더 원활해집니다. 로봇이 비교적 작은 공간에서 이동하는 경우 회전 반경을 더 작은 값으로 설정하십시오.  로봇이 넓은 공간에서 작업하고 다른 장애물이 없으며 로봇의 두 경로 사이의 거리가 멀었을 때, 회전 반경을 적절히 크게 조절하여 로봇의 움직임을 보다 원활하게 할 수 있습니다.

## 스마트 장애물 회피

이 기능은 로봇이 더 많은 경로를 계획하고 디버깅의 어려움을 줄이며 경로 계획의 성공률을 향상시킬 수 있도록 “고정점 이동” 스텝과 대응하는 웨이포인트에 대해 여러 후보 지점을 추가하는 것입니다.

### 장애물 회피 모드

경로 계획의 속도와 성공률에 대한 요구 사항에 따라 \*장애물 회피 모드\*를 설정하십시오.

<b>사용하지 않기</b>	장애물 회피 모드를 사용하지 않습니다.
<b>속도 우선</b>	경로는 속도 우선으로 계획되며 장애물 회피 포인트의 수가 가장 적습니다.
<b>성공률 우선</b>	경로는 성공률 우선으로 계획되며 장애물 회피 포인트의 수가 가장 많습니다.
<b>밸런스</b>	속도와 성공률의 균형을 맞춰 경로를 계획하며 장애물 회피 포인트의 수가 중간 정도입니다.

### 위치 조정 범위(반경)

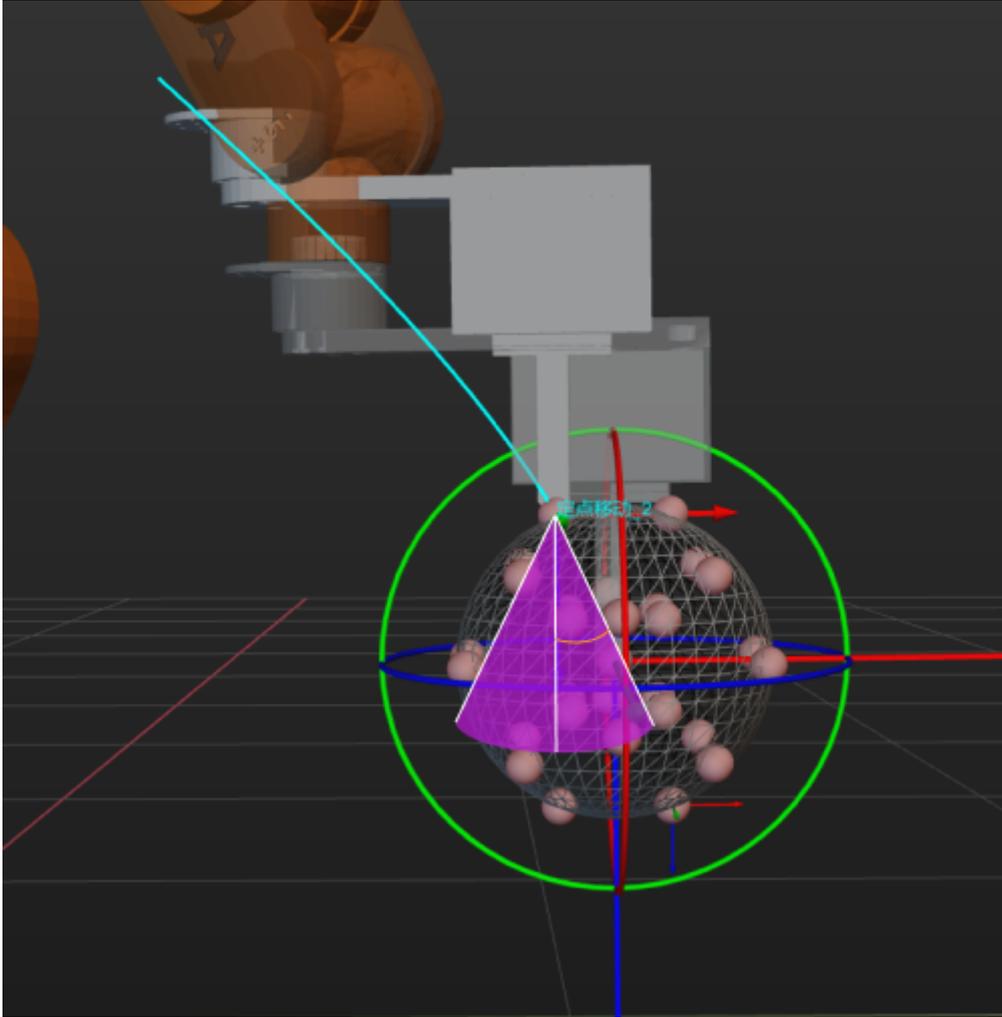
기본값: 0.005

이 반경을 가진 구체 내에 장애물 회피 포인트가 생성됩니다.

### 각도 조정 범위

기본값: 0. 조정 범위: 0 - 79

장애물 회피 포인트 범위 내에서 TCP가 조정됩니다.

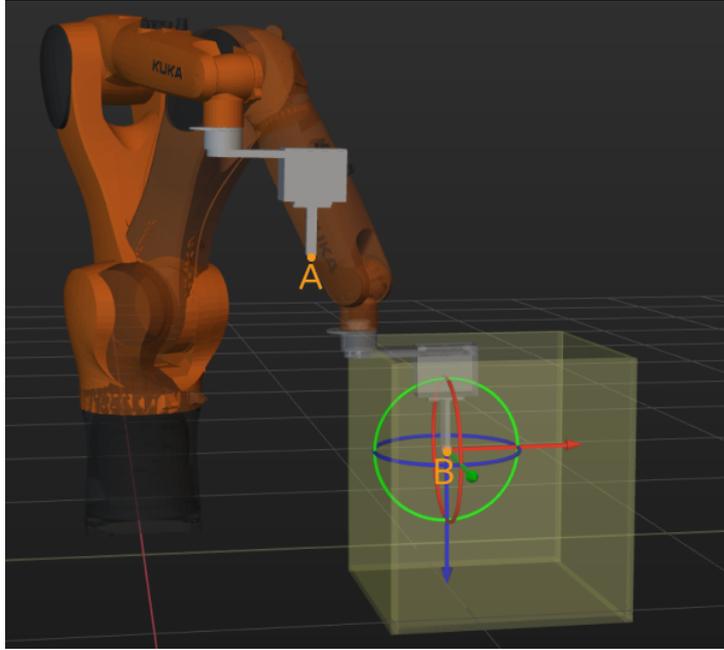


#### 모든 후보 포즈를 표시하기

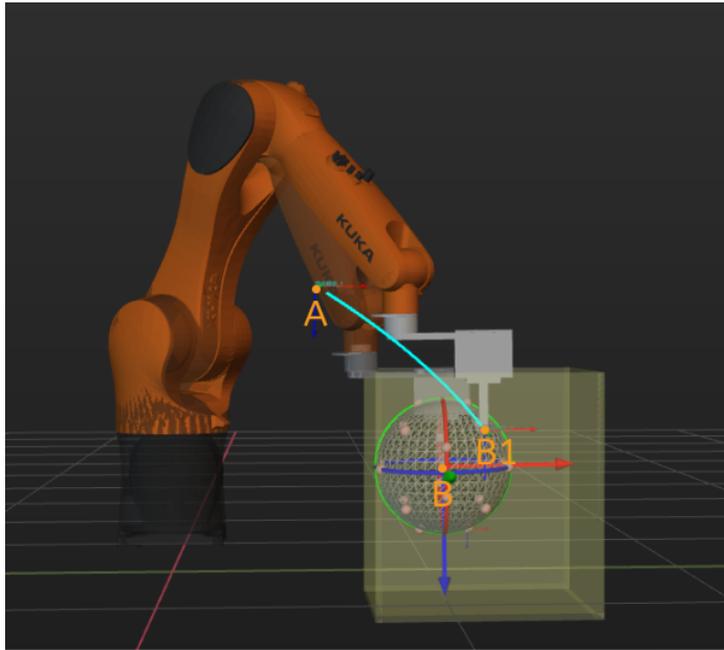
기본적으로 선택되어 있으며 모든 후보 포인트가 표시됩니다. 언체크하면 후보 포인트가 표시되지 않습니다.

#### 응용 예시

아래 그림과 같이 로봇이 웨이포인트 A에서 웨이포인트 B로 이동할 때 로봇 말단장치는 빈과 충돌합니다.



웨이포인트 B에서 스마트 장애물 회피 기능을 활성화한 후 소프트웨어는 아래와 같이 충돌을 피하기 위해 장애물 회피 포인트 B1을 자동으로 선택합니다.



#### 5.4.1.3. 배열대로 이동

##### 기능 설명

로봇이 이동 경로에서 배열에 따라 여러 개의 \*목표 포즈\*와 해당 포즈로 이동하는 \*이동 방식\*을 설정하고 설정한 순서에 따라 각 포즈로 이동합니다.

##### 파라미터 설명

## 이동 스텝 기본 파라미터

### 웨이포인트를 전송하기

기본적으로 선택되어 있으며 로봇 등 수신자에 웨이포인트를 전송합니다. 선택하지 않으며 웨이포인트를 전송하지 않지만 해당 웨이포인트는 여전히 경로 계획 중에 있습니다.

### 후속 비이동 명령을 원활성있게 수행하기

기본적으로 선택되어 있지 않습니다. **이동 스텝** 사이에 **비이동 스텝**(예: "비전 인식", "DI 체크" 등)이 연결되면 로봇의 경로 계획을 방해하고 실제 로봇이 실행 중일 때 짧은 일시 중지가 발생하여 로봇이 원활하지 않게 실행됩니다. 이 옵션을 선택하면 현재 이동 스텝의 실행 종료를 기다릴 필요가 없고 계속해서 작업 흐름에 따라 실행할 수 있습니다. 이렇게 하면 로봇 실행 과정에서 자주 멈추는 문제를 피할 수 있고 로봇 동작의 원활성을 높일 수 있지만 스텝이 너무 일찍 종료될 수 있습니다.



스텝은 일찍 종료되는 이유가 무엇입니까?

Mech-Viz 소프트웨어가 실행 중일 때 동시에 로봇에 여러 포즈를 보냅니다. 소프트웨어는 로봇에 보낸 마지막 포즈가 로봇이 반환한 관절 각도와 동일한지 여부만 판단하고 동일하면 로봇이 마지막 위치에 이미 도달했다는 것으로 간주합니다. 예를 들어 경로에는 10개의 이동 스텝으로 구성되고 이동 스텝 5의 포즈는 마지막 이동 스텝의 포즈와 동일합니다. 로봇이 느린 속도로 이동할 때 웨이포인트 5로 이동하고 현재 관절 각도를 Mech-Viz로 보냅니다. 이동 스텝 5의 포즈는 마지막 이동 스텝의 포즈와 동일하기 때문에 Mech-Viz 소프트웨어는 경로 중의 모든 스텝 실행이 이미 완료된 것으로 잘못 판단하여 일찍 명령을 종료합니다.

### 배치된 물체와의 충돌을 감지하지 않기

기본적으로 선택되어 있지 않습니다. 즉 이미 배치된 물체와 사이의 충돌을 감지합니다. 이 옵션을 선택하면 로봇, 말단장치 및 배치된 물체 사이의 충돌을 감지하지 않습니다.

팔레타이징 응용 시나리오에 다음과 같은 두 가지 경우가 있습니다.

1. 팔레타이징 시나리오에서 상자를 쌓을 때 로봇 자체가 이미 배치된 상자와 접촉할 경우가 있습니다(상자가 압착되거나 변형되지 않음). 이 충돌을 감지하면 Mech-Viz는 충돌을 피하기 위해 다른 배치 포인트를 계획하기 때문에 오히려 파렛트를 완전히 채우지 못하게 됩니다.
2. 일반적으로 진공 그리퍼의 TCP는 그리퍼 표면이 아닌 모델 내부에 설정되어 있으며, 이로 인해 물체를 피킹할 때 말단장치와 피킹된 상자 모델이 중첩되는 상황이 발생합니다(소프트웨어는 말단장치와 작업물 사이의 충돌을 감지하지 않음). 로봇은 상자를 놓고 배치한 후 배치된 상자 모델은 시나리오 모델이 되며 이때 소프트웨어는 말단장치와 시나리오 속의 상자 모델과의 충돌을 감지하여 소프트웨어에서 충돌 경보가 발생하여 팔레타이징 스텝을 완료할 수 없습니다.

이 파라미터를 선택하면 소프트웨어는 로봇, 말단장치와 배치된 작업물 사이의 충돌을 감지하지 않고 위 문제를 피할 수 있습니다.

### 포인트 클라우드 충돌 감지 모드

작업 현장의 실제 상황에 따라 파라미터를 설정하며 일반적으로 **Auto\***를 사용하면 됩니다. **로봇이 물체를 피킹하기 전의 이동 스텝은 \*NOTCHECK** 모드, 물체를 피킹한 후의 이동 스텝은 **CHECK** 모드를 선택할 수 있습니다.

<b>Auto</b>	기본값. "비전 이동" 스텝 및 "비전 이동"에 의존하는 "상대적인 이동" 스텝에 대해서만 포인트 클라우드와의 충돌을 감지하고 다른 이동 스텝에 대해 감지하지 않습니다.
<b>NOTCHECK</b>	모든 이동 스텝에 대해 포인트 클라우드와의 충돌을 감지하지 않습니다.
<b>CHECK</b>	모든 이동 스텝에 대해 포인트 클라우드와의 충돌을 감지합니다.



**충돌 감지 > 충돌 감지 구성 > 포인트 클라우드와 다른 물체의 충돌을 감지하기** 기능을 활성화하면

Mech-Viz 소프트웨어가 경로 계획을 수행할 때 로봇 모델, 말단장치와 포인트 클라우드 사이의 충돌을 감지합니다. 일반적으로 포인트 클라우드 충돌 감지 설정은 로봇이 피킹 과정에서 작업물과의 충돌 여부를 확인하는 것입니다. 공간에 노이즈가 있으면 소프트웨어가 물체 피킹 전의 경로를 계획할 때 로봇 모델, 말단장치 모델이 노이즈와 접촉하므로 포인트 클라우드 사이의 충돌로 잘못 감지하여 소프트웨어의 계획 오류를 초래할 수 있습니다.

### 물체의 대칭성을 사용하지 않기

이 파라미터는 \*웨이포인트 유형\*을 \*작업물 포즈\*로 설정한 경우에만 적용됩니다. 예를 들어 웨이포인트 유형을 작업물 포즈로 설정한 이동/팔레타이징 스텝에 적용되며 웨이포인트 유형이 JPs 또는 TCP 포즈인 이동 스텝에는 적용되지 않습니다.

- None** 기본값. 모든 축의 대칭성을 사용합니다.
- AxisZ** Z축의 대칭성만 사용하지 않습니다.
- AxisXy** X, Y축의 대칭성을 사용하지 않습니다.
- All** 모든 축의 대칭성을 사용하지 않습니다.

대칭성을 사용하지 않으면 로봇은 물체 포즈에 따라 물체를 정확하게 배치합니다.



하지만 일부 특수한 상황에서 물체를 피킹하지 못할 때 **작업물 > 작업물 구성** 중의 \*회전 대칭\*을 설정할 수 있습니다. 인식된 물체에 대해 물체 대칭성을 설정하고 대칭 각도에 따라 작업물이 여러 개의 후보 포즈를 가질 수 있습니다. Mech-Viz 소프트웨어는 물체 피킹을 계획할 때 기본 포즈를 피킹할 수 없는 경우 후보 포즈를 피킹할 수 있는지 여부를 시도합니다. 회전 대칭 설정을 기반으로 계산된 후보 포즈는 Mech-Vision에서 출력된 원시 포즈와 다르며 물체 배치 포즈의 일관성을 보장할 수 없습니다.

### 피킹된 물체의 충돌 감지 모드

#### 시나리오&로봇과의 충돌 감지를 하지 않기

기본적으로 선택되어 있지 않습니다. 이 옵션을 선택한 후에는 **피킹된 물체\*와 \*시나리오 모델, 로봇** 사이의 충돌을 감지하지 않으므로 소프트웨어 충돌 감지의 계산량을 줄이고 Mech-Viz의 계획 속도를 최적화하며 전반적인 사이클 타임을 향상시킵니다. 일반적으로 로봇이 물체를 피킹한 후의 1~2개 이동 스텝에서 사용합니다.

충돌 감지를 수행하지 않으면 충돌 발생의 위험이 있으므로 주의하여 사용하시기 바랍니다.



menu:충돌 감지 구성 [잡고 있는 물체 구성>잡고 있는 작업물과 다른 물체 사이의 충돌을 감지하기] 기능을 활성화하면 잡고 있는 작업물과 시나리오 모델&로봇 사이의 충돌을 감지합니다. 팔레타이징 시나리오에서 비전을 통해 계산된 상자 치수와 실제 치수 사이에는 밀리미터 수준의 편차가 있으면 피킹 과정에서 상자 사이에 마찰이 발생할 수 있으며 충돌이 발생하지 않습니다. 분명히 충돌하지 않는 일부 움직임의 경우, 이러한 충돌을 감지하면 소프트웨어 계산량이 증가하고 Mech-Viz 계획 시간이 늘어나며 사이클 타임이 낭비됩니다. 시나리오 물체와의 충돌을 감지하지 않는 기능을 활성화하면 팔레타이징 시나리오에서 피킹된 작업물과 이미 배치된 상자 간의 충돌 감지에 영향을 미치지 않습니다. 파렛트 밑에 시나리오 물체가 있을 때 이 기능을 사용하면 팔레타이징 솔루션 선택 실패 문제를 피할 수 있습니다.

#### 포인트 클라우드와의 충돌 감지를 하지 않기

기본적으로 선택되어 있지 않습니다. 선택하면 **피킹된 물체\*와 \*시나리오 속의 포인트 클라우드** 사이의 충돌을 감지하지 않고 소프트웨어의 충돌 감지 계산량을 줄여 Mech-Viz의 계획 속도를 향상시키며 전체적인 사이클 타임을 최적화할 수 있습니다. 또한 피킹된 물체와 포인트 클라우드 노이즈 사이의 충돌을 잘못 감지하는 문제를 피할 수 있습니다.



- menu:충돌 감지 구성 [잡고 있는 물체 구성>잡고 있는 작업물과 다른 물체 사이의 충돌을 감지하기] 기능과 **포인트 클라우드 구성 > 포인트 클라우드와 다른 물체의 충돌을 감지하기** 기능을 동시에 활성화하면 잡고 있는 작업물 모델과 시나리오 속의 포인트 클라우드 사이의



충돌을 감지합니다.

- Mech-Vision이 포인트 클라우드와 물체 모델의 정보를 Mech-Viz로 보내면 포인트 클라우드와 물체 모델이 맞춰집니다. 로봇이 물체를 피킹한 후 모델은 계획된 경로를 따라 이동하고 피킹된 작업물 모델과 포인트 클라우드 간의 충돌이 발생합니다.
- 피킹된 작업물 모델과 포인트 클라우드 사이에서 이미 존재한 잘못된 충돌을 피할 수 없는 경우에, 이러한 충돌을 감지하면 불필요한 계산 작업량이 증가하고 Mech-Viz의 계획 시간이 늘어날 수 밖에 없습니다.
- 일반적으로 **충돌 감지를 하지 않기** 기능은 로봇이 물체를 피킹한 후의 1~2 개 이동 스텝에 사용되며 로봇이 충돌할 위험이 있으므로 주의하여 사용하십시오.
- \*시나리오와의 충돌 감지를 하지 않기\*를 활성화하면 팔레타이징 시나리오에서 피킹된 작업물과 이미 배치된 상자 간의 충돌 감지에 영향을 미치지 않습니다. 팔레트 밑에 시나리오 물체가 있는 경우, 이 기능을 사용하여 팔레타이징 솔루션 선택 실패 문제를 피할 수 있습니다.

## 인덱스

### 시작 인덱스

수행할 웨이포인트의 인덱스입니다. 예시: 첫 번째 웨이포인트와 대응한 인덱스는 0이므로 배열의 셋 번째 포인트부터 이동하려면 이 파라미터를 2로 설정해야 합니다.

### 현재 인덱스

현재 실행 중인 웨이포인트의 인덱스입니다.

### 배열 설정

이 파라미터는 주로 현재 배열의 각 방향(X, Y,Z)에서 웨이포인트의 수량과 간격을 설정하는 데 사용됩니다.

#### X 방향 웨이포인트 수량

배열의 X 방향에서 웨이포인트의 수량.

**X 방향 웨이포인트 간의 간격**      배열의 X 방향에서 두 웨이포인트 사이의 간격.

**Y 방향 웨이포인트 수량**              배열의 Y 방향에서 웨이포인트의 수량.

**Y 방향 웨이포인트 간의 간격**      배열의 Y 방향에서 두 웨이포인트 사이의 간격.

### 기본 이동 설정

“기본 이동 설정”은 이동 스텝을 사용할 때 조절해야 하는 중요한 파라미터이며 로봇이 웨이포인트 위치로 이동하는 과정의 **속도** 및 **\*운동 방식\***을 컨트롤하는 데 사용됩니다.

### 피킹&배치

**지정하지 않음**      기본값

**피킹**                  “비전 이동” 전의 이동 스텝에 적용됩니다.

**배치**                  “비전 이동” 후의 이동 스텝에 적용됩니다.

**조절 설명**              Mech-Viz 프로젝트 실행 논리를 검사하는 데 사용될 수 있습니다. 작업 현장의 실제 작업 프로세스에 따라 피킹한 다음에 배치한다는 순서를 기본 원칙으로 이동 스텝에 대해 피킹&배치를 설정합니다.

## 운동 방식

**관절 운동** 로봇의 실행 경로가 원호이기 때문에 더 원활하게 이동할 수 있어 이동 중에 싱귤래리티가 나타나기 쉽지 않음을 의미합니다. 관절 운동은 경로 계획의 정밀도에 대한 요구가 그리 높지 않고 로봇이 큰 범위에서 이동하는 시나리오에 적용됩니다.

**직선 운동** 로봇의 실행 경로가 직선이며 로봇 경로의 정밀도에 대한 요구가 높습니다. 용접, 접착제 도포, 피킹 등 경로 정밀도에 대한 요구가 상대적으로 높은 시나리오에 적용됩니다.

## 속도&가속도

로봇의 이동 속도를 결정합니다. 일반적으로 가속도 파라미터의 수치가 속도보다 작아야 합니다. 가속도 파라미터의 수치가 속도보다 클 때 로봇의 동작이 원활하지 않습니다.

**!** 피킹의 안정성을 위해 "비전 이동"과 전/후의 이동 속도를 낮게 설정해야 합니다.

## 회전 반경

**기본값** 50.00mm

**조절 설명** 일반적으로 조절할 필요가 없고 기본값을 사용합니다.

회전 반경은 웨이포인트와 로봇이 회전하기 시작하는 지점 사이의 거리를 나타냅니다. 회전 반경이 클수록 로봇의 이동이 더 원활해집니다. 로봇이 비교적 작은 공간에서 이동하는 경우 회전 반경을 더 작은 값으로 설정하십시오.

로봇이 넓은 공간에서 작업하고 다른 장애물이 없으며 로봇의 두 경로 사이의 거리가 멀었을 때, 회전 반경을 적절히 크게 조절하여 로봇의 움직임을 보다 원활하게 할 수 있습니다.

## 웨이포인트 그리드 표시

이 옵션을 선택하면 Mech-Viz 화면에서 배열의 웨이포인트 및 각 웨이포인트의 번호를 확인할 수 있습니다.

## 로봇 표시

이 옵션을 선택하면 Mech-Viz 화면에서 **배열대로 이동** 스텝을 실행하는 과정에서 첫 번째 포인트 위치에 로봇 또는 물체 포즈를 확인할 수 있습니다.

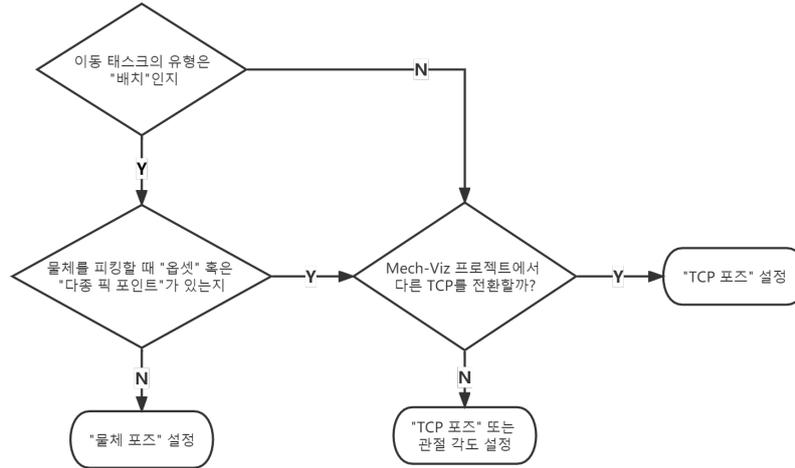
## 웨이포인트 유형

**TCP** 웨이포인트는 TCP 좌표계의 X, Y, Z 값과 오일러 각/사원수로 표시됩니다.

**JPs** 웨이포인트는 로봇 각 관절의 수치로 표시됩니다.

**작업물 포즈** 웨이포인트는 물체 좌표계의 X, Y, Z 값과 오일러 각/사원수로 표시됩니다.

웨이포인트 유형을 선택할 때 판단 방법은 다음 그림과 같습니다.



### 포즈 편집

포즈를 직접 편집하고 복사/붙여넣을 수 있으며 사원수와 오일러 각 두 가지 형식을 지원합니다.

### 포즈 변환

자체 정의를 통해 현재 포즈를 새 포즈로 변환하는 데 사용되며 포즈 파인튜닝에 적용됩니다.

### 포즈 보정

작업물의 좌표계를 계산하는 ABB 로봇 삼점법과 유사하며 대상 물체가 회전할 때 포즈를 확인하기 쉽지 않은 시나리오에 적합합니다. 예를 들어 기울어진 직육면체의 회전 포즈를 계산하기 어려운 경우, 포즈 보정을 통해 직육면체의 회전 포즈를 먼저 계산하고 로봇이 계산된 포즈에 따라 실행하도록 합니다.

### JPs 편집

포즈를 편집하는 기능과 유사하여 관절 각도를 복사/붙여넣을 수 있고 라디안과 각도 두 가지 단위를 지원합니다.



가상 로봇을 설정된 웨이포인트 위치로 이동합니다.



배치된 로봇의 포즈를 읽어 웨이포인트로 설정합니다.



현재 웨이포인트에 대한 모든 JPs 솔루션을 표시합니다.

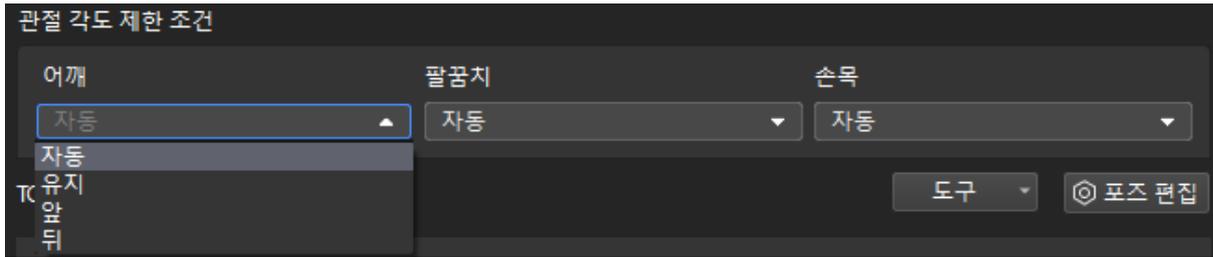
### 관절 각도 제한 조건

#### 개념

**어깨** 손목 관절 중심과 Axis1의 상대적 위치 관계입니다. Axis1은 로봇 축1의 회전 중심 축을 나타냅니다.

**팔꿈치** 손목 관절과 LowerArm의 상대적 위치 관계입니다. LowerArm은 로봇의 축2, 축3의 회전 중심의 연결선을 나타냅니다.

**손목** 손목 관절은 사실 로봇의 축5입니다. 축5 각도의 양/음의 관계는 손목이 뒤집히는 상태를 나타내고, Wrist center는 손목 관절의 중심을 나타냅니다.



## 값 리스트

- 자동** 로봇 관절의 움직임에는 제한 조건이 없습니다. 최적의 솔루션은 각 축의 회전폭이 가장 작은 솔루션입니다.
- 유지** 현재 JPs 상태에 따라 다음 JPs 솔루션을 제한합니다. 예를 들어, 로봇 축3의 현재 JPs가 양수이면 로봇 축3의 JPs가 양수인 솔루션만 다음 웨이포인트로 고려됩니다.
- 앞** 손목 관절의 중심은 Axis1 앞에 있습니다.
- 뒤** 손목 관절의 중심은 Axis1뒤에 있습니다.



버튼을 클릭하면 현재 포즈에 대한 모든 JPs 솔루션을 표시하는 창이 나타납니다. 포즈 중 하나를 클릭하면 솔루션의 해당 로봇 포즈가 3D 시뮬레이션 영역에 표시됩니다. 따라서 아래와 같이 다양한 제한 조건에서 가능한 솔루션을 알 수 있습니다.



1. 관절 각도 제한 조건은 6축 로봇에만 적용되며 4축 로봇은 어깨, 팔꿈치, 손목을 뒤척이지 않습니다.
2. “상대적인 이동”, “미리 설정한 파렛트 패턴”, “자체 정의한 파렛트 패턴” 스텝에서 이 파라미터를 설정할 수 없으며 기본적으로 로봇의 어깨, 팔꿈치 및 손목이 변경되지 않습니다. 즉, 이러한 스텝에서 로봇이 싱글래티를 거치지 않도록 제한합니다.

### 5.4.1.4. 순서대로 이동

#### 기능 설명

로봇이 이동 경로에서 순서에 따라 여러 개의 **목표 포즈** 와 해당 포즈로 이동하는 **\*이동 방식\***을 설정하고 설정한 순서에 따라 각 포즈로 이동합니다.

#### 파라미터 설명

##### 이동 스텝 기본 파라미터

##### 웨이포인트를 전송하기

기본적으로 선택되어 있으며 로봇 등 수신자에 웨이포인트를 전송합니다. 선택하지 않으며 웨이포인트를 전송하지 않지만 해당 웨이포인트는 여전히 경로 계획 중에 있습니다.

##### 후속 비이동 명령을 원활성있게 수행하기

기본적으로 선택되어 있지 않습니다. **이동 스텝** 사이에 **비이동 스텝**(예: “비전 인식”, “DI 체크” 등)이 연결되면 로봇의 경로 계획을 방해하고 실제 로봇이 실행 중일 때 짧은 일시 중지가 발생하여 로봇이 원활하지 않게 실행됩니다. 이 옵션을 선택하면 현재 이동 스텝의 실행 종료를 기다릴 필요가 없고 계속해서 작업 흐름에 따라 실행할 수 있습니다. 이렇게 하면 로봇 실행 과정에서 자주 멈추는 문제를 피할 수 있고 로봇 동작의 원활성을 높일 수 있지만 스텝이 너무 일찍 종료될 수 있습니다.



스텝은 일찍 종료되는 이유가 무엇입니까?

Mech-Viz 소프트웨어가 실행 중일 때 동시에 로봇에 여러 포즈를 보냅니다. 소프트웨어는 로봇에 보낸 마지막 포즈가 로봇이 반환한 관절 각도와 동일한지 여부만 판단하고 동일하면 로봇이 마지막 위치에 이미 도달했다는 것으로 간주합니다. 예를 들어 경로에는 10개의 이동 스텝으로 구성되고 이동 스텝 5의 포즈는 마지막 이동 스텝의 포즈와 동일합니다. 로봇이 느린 속도로 이동할 때 웨이포인트 5로 이동하고 현재 관절 각도를 Mech-Viz로 보냅니다. 이동 스텝 5의 포즈는 마지막 이동 스텝의 포즈와 동일하기 때문에 Mech-Viz 소프트웨어는 경로 중의 모든 스텝 실행이 이미 완료된 것으로 잘못 판단하여 일찍 명령을 종료합니다.

### 배치된 물체와의 충돌을 감지하지 않기

기본적으로 선택되어 있지 않습니다. 즉 이미 배치된 물체와 사이의 충돌을 감지합니다. 이 옵션을 선택하면 로봇, 말단장치 및 배치된 물체 사이의 충돌을 감지하지 않습니다.

팔레타이징 응용 시나리오에 다음과 같은 두 가지 경우가 있습니다.

1. 팔레타이징 시나리오에서 상자를 쌓을 때 로봇 자체가 이미 배치된 상자와 접촉할 경우가 있습니다(상자가 압착되거나 변형되지 않음). 이 충돌을 감지하면 Mech-Viz는 충돌을 피하기 위해 다른 배치 포인트를 계획하기 때문에 오히려 파レット를 완전히 채우지 못하게 됩니다.
2. 일반적으로 진공 그리퍼의 TCP는 그리퍼 표면이 아닌 모델 내부에 설정되어 있으며, 이로 인해 물체를 피킹할 때 말단장치와 피킹된 상자 모델이 중첩되는 상황이 발생합니다(소프트웨어는 말단장치와 작업물 사이의 충돌을 감지하지 않음). 로봇은 상자를 놓고 배치한 후 배치된 상자 모델은 시나리오 모델이 되며 이때 소프트웨어는 말단장치와 시나리오 속의 상자 모델과의 충돌을 감지하여 소프트웨어에서 충돌 경보가 발생하여 팔레타이징 스텝을 완료할 수 없습니다.

이 파라미터를 선택하면 소프트웨어는 로봇, 말단장치와 배치된 작업물 사이의 충돌을 감지하지 않고 위 문제를 피할 수 있습니다.

### 포인트 클라우드 충돌 감지 모드

작업 현장의 실제 상황에 따라 파라미터를 설정하며 일반적으로 **Auto\***를 사용하면 됩니다. **로봇이 물체를 피킹하기 전의 이동 스텝은 \*NOTCHECK** 모드, 물체를 피킹한 후의 이동 스텝은 **CHECK** 모드를 선택할 수 있습니다.

<b>Auto</b>	기본값. "비전 이동" 스텝 및 "비전 이동"에 의존하는 "상대적인 이동" 스텝에 대해서만 포인트 클라우드와의 충돌을 감지하고 다른 이동 스텝에 대해 감지하지 않습니다.
<b>NOTCHECK</b>	모든 이동 스텝에 대해 포인트 클라우드와의 충돌을 감지하지 않습니다.
<b>CHECK</b>	모든 이동 스텝에 대해 포인트 클라우드와의 충돌을 감지합니다.



**충돌 감지 > 충돌 감지 구성 > 포인트 클라우드와 다른 물체의 충돌을 감지하기** 기능을 활성화하면 Mech-Viz 소프트웨어가 경로 계획을 수행할 때 로봇 모델, 말단장치와 포인트 클라우드 사이의 충돌을 감지합니다. 일반적으로 포인트 클라우드 충돌 감지 설정은 로봇이 피킹 과정에서 작업물과의 충돌 여부를 확인하는 것입니다. 공간에 노이즈가 있으면 소프트웨어가 물체 피킹 전의 경로를 계획할 때 로봇 모델, 말단장치 모델이 노이즈와 접촉하므로 포인트 클라우드 사이의 충돌로 잘못 감지하여 소프트웨어의 계획 오류를 초래할 수 있습니다.

### 물체의 대칭성을 사용하지 않기

이 파라미터는 \*웨이포인트 유형\*을 \*작업물 포즈\*로 설정한 경우에만 적용됩니다. 예를 들어 웨이포인트 유형을 작업물 포즈로 설정한 이동/팔레타이징 스텝에 적용되며 웨이포인트 유형이 JPs 또는 TCP 포즈인 이동 스텝에는 적용되지 않습니다.

<b>None</b>	기본값. 모든 축의 대칭성을 사용합니다.
<b>AxisZ</b>	Z축의 대칭성만 사용하지 않습니다.

**AxisXy** X, Y축의 대칭성을 사용하지 않습니다.

**All** 모든 축의 대칭성을 사용하지 않습니다.

대칭성을 사용하지 않으면 로봇은 물체 포즈에 따라 물체를 정확하게 배치합니다.



하지만 일부 특수한 상황에서 물체를 피킹하지 못할 때 **작업물 > 작업물 구성** 중의 \*회전 대칭\*을 설정할 수 있습니다. 인식된 물체에 대해 물체 대칭성을 설정하고 대칭 각도에 따라 작업물이 여러 개의 후보 포즈를 가질 수 있습니다. Mech-Viz 소프트웨어는 물체 피킹을 계획할 때 기본 포즈를 피킹할 수 없는 경우 후보 포즈를 피킹할 수 있는지 여부를 시도합니다. 회전 대칭 설정을 기반으로 계산된 후보 포즈는 Mech-Vision에서 출력된 원시 포즈와 다르며 물체 배치 포즈의 일관성을 보장할 수 없습니다.

### 피킹된 물체의 충돌 감지 모드

#### 시나리오&로봇과의 충돌 감지를 하지 않기

기본적으로 선택되어 있지 않습니다. 이 옵션을 선택한 후에는 **피킹된 물체\*와 \*시나리오 모델, 로봇** 사이의 충돌을 감지하지 않으므로 소프트웨어 충돌 감지의 계산량을 줄이고 Mech-Viz의 계획 속도를 최적화하며 전반적인 사이클 타임을 향상시킵니다. 일반적으로 로봇이 물체를 피킹한 후의 1~2개 이동 스텝에서 사용합니다.

충돌 감지를 수행하지 않으면 충돌 발생의 위험이 있으므로 주의하여 사용하시기 바랍니다.



menu:충돌 감지 구성 [잡고 있는 물체 구성>잡고 있는 작업물과 다른 물체 사이의 충돌을 감지하기] 기능을 활성화하면 잡고 있는 작업물과 시나리오 모델&로봇 사이의 충돌을 감지합니다. 팔레타이징 시나리오에서 비전을 통해 계산된 상자 치수와 실제 치수 사이에는 밀리미터 수준의 편차가 있으면 피킹 과정에서 상자 사이에 마찰이 발생할 수 있으며 충돌이 발생하지 않습니다. 분명히 충돌하지 않는 일부 움직임의 경우, 이러한 충돌을 감지하면 소프트웨어 계산량이 증가하고 Mech-Viz 계획 시간이 늘어나며 사이클 타임이 낭비됩니다. 시나리오 물체와의 충돌을 감지하지 않는 기능을 활성화하면 팔레타이징 시나리오에서 피킹된 작업물과 이미 배치된 상자 간의 충돌 감지에 영향을 미치지 않습니다. 파렛트 밑에 시나리오 물체가 있을 때 이 기능을 사용하면 팔레타이징 솔루션 선택 실패 문제를 피할 수 있습니다.

#### 포인트 클라우드와의 충돌 감지를 하지 않기

기본적으로 선택되어 있지 않습니다. 선택하면 **피킹된 물체\*와 \*시나리오 속의 포인트 클라우드** 사이의 충돌을 감지하지 않고 소프트웨어의 충돌 감지 계산량을 줄여 Mech-Viz의 계획 속도를 향상시키며 전체적인 사이클 타임을 최적화할 수 있습니다. 또한 피킹된 물체와 포인트 클라우드 노이즈 사이의 충돌을 잘못 감지하는 문제를 피할 수 있습니다.



- menu:충돌 감지 구성 [잡고 있는 물체 구성>잡고 있는 작업물과 다른 물체 사이의 충돌을 감지하기] 기능과 **포인트 클라우드 구성 > 포인트 클라우드와 다른 물체의 충돌을 감지하기** 기능을 동시에 활성화하면 잡고 있는 작업물 모델과 시나리오 속의 포인트 클라우드 사이의 충돌을 감지합니다.

- Mech-Vision이 포인트 클라우드와 물체 모델의 정보를 Mech-Viz로 보내면 포인트 클라우드와 물체 모델이 맞춰집니다. 로봇이 물체를 피킹한 후 모델은 계획된 경로를 따라 이동하고 피킹된 작업물 모델과 포인트 클라우드 간의 충돌이 발생합니다.

- 피킹된 작업물 모델과 포인트 클라우드 사이에서 이미 존재한 잘못된 충돌을 피할 수 없는 경우에, 이러한 충돌을 감지하면 불필요한 계산 작업량이 증가하고 Mech-Viz의 계획 시간이 늘어날 수 밖에 없습니다.



일반적으로 **충돌 감지를 하지 않기** 기능은 로봇이 물체를 피킹한 후의 1~2 개 이동 스텝에 사용되며 로봇이 충돌할 위험이 있으므로 주의하여 사용하십시오. \*시나리오와의 충돌 감지를 하지 않기\*를 활성화하면 팔레타이징 시나리오에서 피킹된 작업물과 이미 배치된 상자 간의 충돌 감지에 영향을 미치지 않습니다. 파렛트 밑에 시나리오 물체가 있는 경우, 이 기능을 사용하여 팔레타이징 솔루션 선택 실패 문제를 피할 수 있습니다.

## 인덱스

### 시작 인덱스

수행할 웨이포인트의 인덱스입니다.

예시: 첫 번째 웨이포인트와 대응한 인덱스는 0이므로 이동 리스트의 셋 번째 포인트부터 이동하려면 이 파라미터를 2로 설정해야 합니다.

### 현재 인덱스

현재 실행 중인 웨이포인트의 인덱스입니다.

### 한번의 이동으로 모든 웨이포인트에 도달하기

이 옵션을 선택하면 **인덱스** 설정이 무효로 되고 Mech-Viz 프로젝트가 **순서대로 이동** 스텝을 실행할 때 로봇은 한번에 이동 리스트의 모든 웨이포인트를 순서대로 실행합니다.

### 전역 이동 파라미터를 사용하기

이 옵션을 선택하면 리스트에 있는 모든 이동 스텝은 동일한 **\*기본 운동 설정\***을 사용합니다.

## 기본 이동 설정

“기본 이동 설정”은 이동 스텝을 사용할 때 조절해야 하는 중요한 파라미터이며 로봇이 웨이포인트 위치로 이동하는 과정의 **속도** 및 **\*운동 방식\***을 컨트롤하는 데 사용됩니다.

### 피킹&배치

**지정하지 않음** 기본값

**피킹** “비전 이동” 전의 이동 스텝에 적용됩니다.

**배치** “비전 이동” 후의 이동 스텝에 적용됩니다.

**조절 설명** Mech-Viz 프로젝트 실행 논리를 검사하는 데 사용될 수 있습니다. 작업 현장의 실제 작업 프로세스에 따라 피킹한 다음에 배치한다는 순서를 기본 원칙으로 이동 스텝에 대해 피킹&배치를 설정합니다.

### 운동 방식

**관절 운동** 로봇의 실행 경로가 원호이기 때문에 더 원활하게 이동할 수 있어 이동 중에 싱귤래리티가 나타나기 쉽지 않음을 의미합니다. 관절 운동은 경로 계획의 정밀도에 대한 요구가 그리 높지 않고 로봇이 큰 범위에서 이동하는 시나리오에 적용됩니다.

**직선 운동** 로봇의 실행 경로가 직선이며 로봇 경로의 정밀도에 대한 요구가 높습니다. 용접, 접착제 도포, 피킹 등 경로 정밀도에 대한 요구가 상대적으로 높은 시나리오에 적용됩니다.

### 속도&가속도

로봇의 이동 속도를 결정합니다. 일반적으로 가속도 파라미터의 수치가 속도보다 작아야 합니다. 가속도 파라미터의 수치가 속도보다 클 때 로봇의 동작이 원활하지 않습니다.



피킹의 안정성을 위해 "비전 이동"과 전/후의 이동 속도를 낮게 설정해야 합니다.

**회전 반경**

**기본값** 50.00mm

**조절 설명** 일반적으로 조절할 필요가 없고 기본값을 사용합니다.

회전 반경은 웨이포인트와 로봇이 회전하기 시작하는 지점 사이의 거리를 나타냅니다. 회전 반경이 클수록 로봇의 이동이 더 원활해집니다. 로봇이 비교적 작은 공간에서 이동하는 경우 회전 반경을 더 작은 값으로 설정하십시오.

로봇이 넓은 공간에서 작업하고 다른 장애물이 없으며 로봇의 두 경로 사이의 거리가 멀었을 때, 회전 반경을 적절히 크게 조절하여 로봇의 움직임을 보다 원활하게 할 수 있습니다.

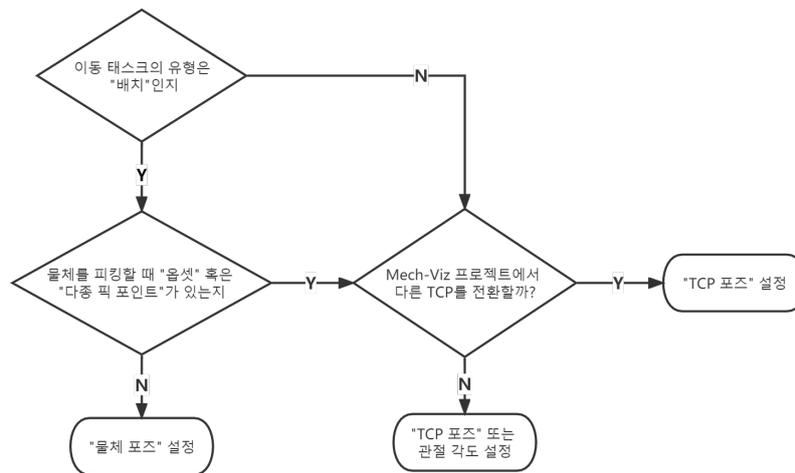
**웨이포인트 유형**

**TCP** 웨이포인트는 TCP 좌표계의 X, Y, Z 값과 오일러 각/사원수로 표시됩니다.

**JPs** 웨이포인트는 로봇 각 관절의 수치로 표시됩니다.

**작업물 포즈** 웨이포인트는 물체 좌표계의 X, Y, Z 값과 오일러 각/사원수로 표시됩니다.

웨이포인트 유형을 선택할 때 판단 방법은 다음 그림과 같습니다.



**포즈 편집** 포즈를 직접 편집하고 복사/붙여넣을 수 있으며 사원수와 오일러 각 두 가지 형식을 지원합니다.

**포즈 변환** 자체 정의를 통해 현재 포즈를 새 포즈로 변환하는 데 사용되며 포즈 파인튜닝에 적용됩니다.

**포즈 보정** 작업물의 좌표계를 계산하는 ABB 로봇 삼점법과 유사하며 대상 물체가 회전할 때 포즈를 확인하기 쉽지 않은 시나리오에 적합합니다. 예를 들어 기울어진 직육면체의 회전 포즈를 계산하기 어려운 경우, 포즈 보정을 통해 직육면체의 회전 포즈를 먼저 계산하고 로봇이 계산된 포즈에 따라 실행하도록 합니다.

**JPs 편집** 포즈를 편집하는 기능과 유사하여 관절 각도를 복사/붙여넣을 수 있고 라디안과 각도 두 가지 단위를 지원합니다.



가상 로봇을 설정된 웨이포인트 위치로 이동합니다.



배치된 로봇의 포즈를 읽어 웨이포인트로 설정합니다.



현재 웨이포인트에 대한 모든 JPs 솔루션을 표시합니다.

## 이동 리스트

\*이동 리스트\*에 추가하려는 포즈를 순서에 따라 수동으로 추가하고 관련 이동 파라미터를 설정하십시오.

이동 리스트 **끝**에 웨이포인트를 추가하려면 +를 클릭하고 선택한 웨이포인트를 삭제하려면 -를 클릭합니다.

## TCP 포즈 리스트 추가

\*TCP 포즈\*에 따라 한번에 여러 개 이동 포즈를 이동 리스트로 추가할 수 있습니다.

추가 화면에서 개별 라인으로 추가하려는 \*TCP 포즈\*를 입력하면 됩니다.

### 5.4.1.5. 외부 이동

#### 기능 설명

이 스텝은 Mech-Viz가 외부 서비스에서 웨이포인트를 획득하여 로봇을 해당 웨이포인트로 이동하는 데 사용됩니다. Adapter와 함께 사용해야 합니다.

#### 파라미터 설명

##### 이동 스텝 기본 파라미터

##### 웨이포인트를 전송하기

기본적으로 선택되어 있으며 로봇 등 수신자에 웨이포인트를 전송합니다. 선택하지 않으며 웨이포인트를 전송하지 않지만 해당 웨이포인트는 여전히 경로 계획 중에 있습니다.

##### 후속 비이동 명령을 원활성있게 수행하기

기본적으로 선택되어 있지 않습니다. **이동 스텝** 사이에 **비이동 스텝**(예: “비전 인식”, “DI 체크” 등)이 연결되면 로봇의 경로 계획을 방해하고 실제 로봇이 실행 중일 때 짧은 일시 중지가 발생하여 로봇이 원활하지 않게 실행됩니다. 이 옵션을 선택하면 현재 이동 스텝의 실행 종료를 기다릴 필요가 없고 계속해서 작업 흐름에 따라 실행할 수 있습니다. 이렇게 하면 로봇 실행 과정에서 자주 멈추는 문제를 피할 수 있고 로봇 동작의 원활성을 높일 수 있지만 스텝이 너무 일찍 종료될 수 있습니다.

스텝은 일찍 종료되는 이유가 무엇입니까?



Mech-Viz 소프트웨어가 실행 중일 때 동시에 로봇에 여러 포즈를 보냅니다. 소프트웨어는 로봇에 보낸 마지막 포즈가 로봇이 반환한 관절 각도와 동일한지 여부만 판단하고 동일하면 로봇이 마지막 위치에 이미 도달했다는 것으로 간주합니다. 예를 들어 경로에는 10개의 이동 스텝으로 구성되고 이동 스텝 5의 포즈는 마지막 이동 스텝의 포즈와 동일합니다. 로봇이 느린 속도로 이동할 때 웨이포인트 5로 이동하고 현재 관절 각도를 Mech-Viz로 보냅니다. 이동 스텝 5의 포즈는 마지막 이동 스텝의 포즈와 동일하기 때문에 Mech-Viz 소프트웨어는 경로 중의 모든 스텝 실행이 이미 완료된 것으로 잘못 판단하여 일찍 명령을 종료합니다.

##### 배치된 물체와의 충돌을 감지하지 않기

기본적으로 선택되어 있지 않습니다. 즉 이미 배치된 물체와 사이의 충돌을 감지합니다. 이 옵션을 선택하면 로봇, 말단장치 및 배치된 물체 사이의 충돌을 감지하지 않습니다.

팔레타이징 응용 시나리오에 다음과 같은 두 가지 경우가 있습니다.

1. 팔레타이징 시나리오에서 상자를 쌓을 때 로봇 자체가 이미 배치된 상자와 접촉할 경우가 있습니다(상자가 압착되거나 변형되지 않음). 이 충돌을 감지하면 Mech-Viz는 충돌을 피하기 위해 다른 배치 포인트를 계획하기 때문에 오히려 파レット를 완전히 채우지 못하게 됩니다.
2. 일반적으로 진공 그리퍼의 TCP는 그리퍼 표면이 아닌 모델 내부에 설정되어 있으며, 이로 인해 물체를 피킹할 때 말단장치와 피킹된 상자 모델이 중첩되는 상황이 발생합니다(소프트웨어는 말단장치와 작업물 사이의 충돌을 감지하지 않음). 로봇은 상자를 놓고 배치한 후 배치된 상자 모델은 시나리오 모델이 되며 이때 소프트웨어는 말단장치와 시나리오 속의 상자 모델과의 충돌을 감지하여 소프트웨어에서 충돌 경보가 발생하여 팔레타이징 스텝을 완료할 수 없습니다.

이 파라미터를 선택하면 소프트웨어는 로봇, 말단장치와 배치된 작업물 사이의 충돌을 감지하지 않고 위 문제를 피할 수 있습니다.

#### 포인트 클라우드 충돌 감지 모드

작업 현장의 실제 상황에 따라 파라미터를 설정하며 일반적으로 **Auto\***를 사용하면 됩니다. **로봇이 물체를 피킹하기 전의 이동 스텝은 \*NOTCHECK** 모드, 물체를 피킹한 후의 이동 스텝은 **CHECK** 모드를 선택할 수 있습니다.

<b>Auto</b>	기본값. "비전 이동" 스텝 및 "비전 이동"에 의존하는 "상대적인 이동" 스텝에 대해서만 포인트 클라우드와의 충돌을 감지하고 다른 이동 스텝에 대해 감지하지 않습니다.
<b>NOTCHECK</b>	모든 이동 스텝에 대해 포인트 클라우드와의 충돌을 감지하지 않습니다.
<b>CHECK</b>	모든 이동 스텝에 대해 포인트 클라우드와의 충돌을 감지합니다.



**충돌 감지 > 충돌 감지 구성 > 포인트 클라우드와 다른 물체의 충돌을 감지하기** 기능을 활성화하면 Mech-Viz 소프트웨어가 경로 계획을 수행할 때 로봇 모델, 말단장치와 포인트 클라우드 사이의 충돌을 감지합니다. 일반적으로 포인트 클라우드 충돌 감지 설정은 로봇이 피킹 과정에서 작업물과의 충돌 여부를 확인하는 것입니다. 공간에 노이즈가 있으면 소프트웨어가 물체 피킹 전의 경로를 계획할 때 로봇 모델, 말단장치 모델이 노이즈와 접촉하므로 포인트 클라우드 사이의 충돌로 잘못 감지하여 소프트웨어의 계획 오류를 초래할 수 있습니다.

#### 물체의 대칭성을 사용하지 않기

이 파라미터는 **\*웨이포인트 유형\***을 **\*작업물 포즈\***로 설정한 경우에만 적용됩니다. 예를 들어 웨이포인트 유형을 작업물 포즈로 설정한 이동/팔레타이징 스텝에 적용되며 웨이포인트 유형이 JPs 또는 TCP 포즈인 이동 스텝에는 적용되지 않습니다.

<b>None</b>	기본값. 모든 축의 대칭성을 사용합니다.
<b>AxisZ</b>	Z축의 대칭성만 사용하지 않습니다.
<b>AxisXy</b>	X, Y축의 대칭성을 사용하지 않습니다.
<b>All</b>	모든 축의 대칭성을 사용하지 않습니다.

대칭성을 사용하지 않으면 로봇은 물체 포즈에 따라 물체를 정확하게 배치합니다.



하지만 일부 특수한 상황에서 물체를 피킹하지 못할 때 **작업물 > 작업물 구성** 중의 **\*회전 대칭\***을 설정할 수 있습니다. 인식된 물체에 대해 물체 대칭성을 설정하고 대칭 각도에 따라 작업물이 여러 개의 후보 포즈를 가질 수 있습니다. Mech-Viz 소프트웨어는 물체 피킹을 계획할 때 기본 포즈를 피킹할 수 없는 경우 후보 포즈를 피킹할 수 있는지 여부를 시도합니다. 회전 대칭 설정을 기반으로 계산된 후보 포즈는 Mech-Vision에서 출력된 원시 포즈와 다르면 물체 배치 포즈의 일관성을 보장할 수 없습니다.

## 피킹된 물체의 충돌 감지 모드

### 시나리오&로봇과의 충돌 감지를 하지 않기

기본적으로 선택되어 있지 않습니다. 이 옵션을 선택한 후에는 **피킹된 물체\*와 \*시나리오 모델, 로봇** 사이의 충돌을 감지하지 않으므로 소프트웨어 충돌 감지의 계산량을 줄이고 Mech-Viz의 계획 속도를 최적화하며 전반적인 사이클 타임을 향상시킵니다. 일반적으로 로봇이 물체를 피킹한 후의 1~2개 이동 스텝에서 사용합니다.

충돌 감지를 수행하지 않으면 충돌 발생의 위험이 있으므로 주의하여 사용하시기 바랍니다.



menu:충돌 감지 구성 [잡고 있는 물체 구성>잡고 있는 작업물과 다른 물체 사이의 충돌을 감지하기] 기능을 활성화하면 잡고 있는 작업물과 시나리오 모델&로봇 사이의 충돌을 감지합니다. 팔레타이징 시나리오에서 비전을 통해 계산된 상자 치수와 실제 치수 사이에는 밀리미터 수준의 편차가 있으면 피킹 과정에서 상자 사이에 마찰이 발생할 수 있으며 충돌이 발생하지 않습니다. 분명히 충돌하지 않는 일부 움직임의 경우, 이러한 충돌을 감지하면 소프트웨어 계산량이 증가하고 Mech-Viz 계획 시간이 늘어나며 사이클 타임이 낭비됩니다. 시나리오 물체와의 충돌을 감지하지 않는 기능을 활성화하면 팔레타이징 시나리오에서 피킹된 작업물과 이미 배치된 상자 간의 충돌 감지에 영향을 미치지 않습니다. 파렛트 밑에 시나리오 물체가 있을 때 이 기능을 사용하면 팔레타이징 솔루션 선택 실패 문제를 피할 수 있습니다.

### 포인트 클라우드와의 충돌 감지를 하지 않기

기본적으로 선택되어 있지 않습니다. 선택하면 **피킹된 물체\*와 \*시나리오 속의 포인트 클라우드** 사이의 충돌을 감지하지 않고 소프트웨어의 충돌 감지 계산량을 줄여 Mech-Viz의 계획 속도를 향상시키며 전체적인 사이클 타임을 최적화할 수 있습니다. 또한 피킹된 물체와 포인트 클라우드 노이즈 사이의 충돌을 잘못 감지하는 문제를 피할 수 있습니다.



- menu:충돌 감지 구성 [잡고 있는 물체 구성>잡고 있는 작업물과 다른 물체 사이의 충돌을 감지하기] 기능과 **포인트 클라우드 구성 > 포인트 클라우드와 다른 물체의 충돌을 감지하기** 기능을 동시에 활성화하면 잡고 있는 작업물 모델과 시나리오 속의 포인트 클라우드 사이의 충돌을 감지합니다.
- Mech-Vision이 포인트 클라우드와 물체 모델의 정보를 Mech-Viz로 보내면 포인트 클라우드와 물체 모델이 맞춰집니다. 로봇이 물체를 피킹한 후 모델은 계획된 경로를 따라 이동하고 피킹된 작업물 모델과 포인트 클라우드 간의 충돌이 발생합니다.
- 피킹된 작업물 모델과 포인트 클라우드 사이에서 이미 존재한 잘못된 충돌을 피할 수 없는 경우에, 이러한 충돌을 감지하면 불필요한 계산 작업량이 증가하고 Mech-Viz의 계획 시간이 늘어날 수 밖에 없습니다.

## 서비스 명칭

Adapter가 Mech-Center에서 등록된 외부 서비스의 명칭이며 서비스 명칭을 통해 이 스텝의 인터페이스를 획득하고 웨이포인트를 전송하기 위해 이 파라미터는 Adapter와 반드시 일치해야 합니다.

### 외부 서비스에서 획득한 JPs를 사용하기

기본적으로 선택되어 있지 않습니다. 소프트웨어는 경로를 계획할 때 이전 웨이포인트를 초기 위치로 사용합니다. 이 옵션을 선택하면 경로 계획의 초기 위치는 외부 서비스에서 획득한 로봇 JPs 위치가 됩니다. 이 기능은 일반적으로 Mech-Viz가 로봇의 마스터 컨트롤에 사용되지 않는 시나리오에서 사용됩니다.

## 5.4.1.6. 상대적인 이동

### 기능 설명

이 스텝은 로봇이 이동 유형 스텝의 알려진 웨이포인트를 기준으로 이동하도록 가이드합니다. 다른 좌표계에서 상대적인 이동을 설정하고 다른 방향으로 회전을 설정할 수 있습니다.

## 응용 시나리오

일반적으로 “비전 이동” 스텝 전/후에 있으며 로봇이 비전 포인트의 Z 방향을 따라 상대적인 이동하도록 가이드합니다.

## 파라미터 설명

### 이동 스텝 기본 파라미터

#### 웨이포인트를 전송하기

기본적으로 선택되어 있으며 로봇 등 수신자에 웨이포인트를 전송합니다. 선택하지 않으며 웨이포인트를 전송하지 않지만 해당 웨이포인트는 여전히 경로 계획 중에 있습니다.

#### 후속 비이동 명령을 원활성있게 수행하기

기본적으로 선택되어 있지 않습니다. **이동 스텝** 사이에 **비이동 스텝**(예: “비전 인식”, “DI 체크” 등)이 연결되면 로봇의 경로 계획을 방해하고 실제 로봇이 실행 중일 때 짧은 일시 중지가 발생하여 로봇이 원활하지 않게 실행됩니다. 이 옵션을 선택하면 현재 이동 스텝의 실행 종료를 기다릴 필요가 없고 계속해서 작업 흐름에 따라 실행할 수 있습니다. 이렇게 하면 로봇 실행 과정에서 자주 멈추는 문제를 피할 수 있고 로봇 동작의 원활성을 높일 수 있지만 스텝이 너무 일찍 종료될 수 있습니다.



스텝은 일찍 종료되는 이유가 무엇입니까?

Mech-Viz 소프트웨어가 실행 중일 때 동시에 로봇에 여러 포즈를 보냅니다. 소프트웨어는 로봇에 보낸 마지막 포즈가 로봇이 반환한 관절 각도와 동일한지 여부만 판단하고 동일하면 로봇이 마지막 위치에 이미 도달했다는 것으로 간주합니다. 예를 들어 경로에는 10개의 이동 스텝으로 구성되고 이동 스텝 5의 포즈는 마지막 이동 스텝의 포즈와 동일합니다. 로봇이 느린 속도로 이동할 때 웨이포인트 5로 이동하고 현재 관절 각도를 Mech-Viz로 보냅니다. 이동 스텝 5의 포즈는 마지막 이동 스텝의 포즈와 동일하기 때문에 Mech-Viz 소프트웨어는 경로 중의 모든 스텝 실행이 이미 완료된 것으로 잘못 판단하여 일찍 명령을 종료합니다.

#### 배치된 물체와의 충돌을 감지하지 않기

기본적으로 선택되어 있지 않습니다. 즉 이미 배치된 물체와 사이의 충돌을 감지합니다. 이 옵션을 선택하면 로봇, 말단장치 및 배치된 물체 사이의 충돌을 감지하지 않습니다.

팔레타이징 응용 시나리오에 다음과 같은 두 가지 경우가 있습니다.

1. 팔레타이징 시나리오에서 상자를 쌓을 때 로봇 자체가 이미 배치된 상자와 접촉할 경우가 있습니다(상자가 압착되거나 변형되지 않음). 이 충돌을 감지하면 Mech-Viz는 충돌을 피하기 위해 다른 배치 포인트를 계획하기 때문에 오히려 파レット를 완전히 채우지 못하게 됩니다.
2. 일반적으로 진공 그리퍼의 TCP는 그리퍼 표면이 아닌 모델 내부에 설정되어 있으며, 이로 인해 물체를 피킹할 때 말단장치와 피킹된 상자 모델이 중첩되는 상황이 발생합니다(소프트웨어는 말단장치와 작업물 사이의 충돌을 감지하지 않음). 로봇은 상자를 놓고 배치한 후 배치된 상자 모델은 시나리오 모델이 되며 이때 소프트웨어는 말단장치와 시나리오 속의 상자 모델과의 충돌을 감지하여 소프트웨어에서 충돌 경보가 발생하여 팔레타이징 스텝을 완료할 수 없습니다.

이 파라미터를 선택하면 소프트웨어는 로봇, 말단장치와 배치된 작업물 사이의 충돌을 감지하지 않고 위 문제를 피할 수 있습니다.

#### 포인트 클라우드 충돌 감지 모드

작업 현장의 실제 상황에 따라 파라미터를 설정하며 일반적으로 **Auto\***를 사용하면 됩니다. **로봇이 물체를 피킹하기 전의 이동 스텝은 \*NOTCHECK** 모드, 물체를 피킹한 후의 이동 스텝은 **CHECK** 모드를 선택할 수 있습니다.

- Auto** 기본값. "비전 이동" 스텝 및 "비전 이동"에 의존하는 "상대적인 이동" 스텝에 대해서만 포인트 클라우드와의 충돌을 감지하고 다른 이동 스텝에 대해 감지하지 않습니다.
- NOTCHECK** 모든 이동 스텝에 대해 포인트 클라우드와의 충돌을 감지하지 않습니다.
- CHECK** 모든 이동 스텝에 대해 포인트 클라우드와의 충돌을 감지합니다.



**충돌 감지 > 충돌 감지 구성 > 포인트 클라우드와 다른 물체의 충돌을 감지하기** 기능을 활성화하면 Mech-Viz 소프트웨어가 경로 계획을 수행할 때 로봇 모델, 말단장치와 포인트 클라우드 사이의 충돌을 감지합니다. 일반적으로 포인트 클라우드 충돌 감지 설정은 로봇이 피킹 과정에서 작업물과의 충돌 여부를 확인하는 것입니다. 공간에 노이즈가 있으면 소프트웨어가 물체 피킹 전의 경로를 계획할 때 로봇 모델, 말단장치 모델이 노이즈와 접촉하므로 포인트 클라우드 사이의 충돌로 잘못 감지하여 소프트웨어의 계획 오류를 초래할 수 있습니다.

#### 물체의 대칭성을 사용하지 않기

이 파라미터는 \*웨이포인트 유형\*을 \*작업물 포즈\*로 설정한 경우에만 적용됩니다. 예를 들어 웨이포인트 유형을 작업물 포즈로 설정한 이동/팔레타이징 스텝에 적용되며 웨이포인트 유형이 JPs 또는 TCP 포즈인 이동 스텝에는 적용되지 않습니다.

- None** 기본값. 모든 축의 대칭성을 사용합니다.
- AxisZ** Z축의 대칭성만 사용하지 않습니다.
- AxisXy** X, Y축의 대칭성을 사용하지 않습니다.
- All** 모든 축의 대칭성을 사용하지 않습니다.

대칭성을 사용하지 않으면 로봇은 물체 포즈에 따라 물체를 정확하게 배치합니다.



하지만 일부 특수한 상황에서 물체를 피킹하지 못할 때 **작업물 > 작업물 구성** 중의 \*회전 대칭\*을 설정할 수 있습니다. 인식된 물체에 대해 물체 대칭성을 설정하고 대칭 각도에 따라 작업물이 여러 개의 후보 포즈를 가질 수 있습니다. Mech-Viz 소프트웨어는 물체 피킹을 계획할 때 기본 포즈를 피킹할 수 없는 경우 후보 포즈를 피킹할 수 있는지 여부를 시도합니다. 회전 대칭 설정을 기반으로 계산된 후보 포즈는 Mech-Vision에서 출력된 원시 포즈와 다르면 물체 배치 포즈의 일관성을 보장할 수 없습니다.

#### 피킹된 물체의 충돌 감지 모드

##### 시나리오&로봇과의 충돌 감지를 하지 않기

기본적으로 선택되어 있지 않습니다. 이 옵션을 선택한 후에는 **피킹된 물체\*와 \*시나리오 모델, 로봇** 사이의 충돌을 감지하지 않으므로 소프트웨어 충돌 감지의 계산량을 줄이고 Mech-Viz의 계획 속도를 최적화하며 전반적인 사이클 타임을 향상시킵니다. 일반적으로 로봇이 물체를 피킹한 후의 1~2개 이동 스텝에서 사용합니다.

충돌 감지를 수행하지 않으면 충돌 발생의 위험이 있으므로 주의하여 사용하시기 바랍니다.



menu:충돌 감지 구성 [잡고 있는 물체 구성>잡고 있는 작업물과 다른 물체 사이의 충돌을 감지하기] 기능을 활성화하면 잡고 있는 작업물과 시나리오 모델&로봇 사이의 충돌을 감지합니다. 팔레타이징 시나리오에서 비전을 통해 계산된 상자 치수와 실제 치수 사이에는 밀리미터 수준의 편차가 있으면 피킹 과정에서 상자 사이에 마찰이 발생할 수 있으며 충돌이 발생하지 않습니다. 분명히 충돌하지 않는 일부 움직임의 경우, 이러한 충돌을 감지하면 소프트웨어 계산량이 증가하고 Mech-Viz 계획 시간이 늘어나며 사이클 타임을 낭비됩니다. 시나리오 물체와의 충돌을 감지하지 않는 기능을 활성화하면 팔레타이징 시나리오에서 피킹된 작업물과 이미 배치된 상자 간의 충돌 감지에 영향을 미치지 않습니다. 파レット 밑에 시나리오 물체가 있을 때 이 기능을 사용하면 팔레타이징 솔루션 선택 실패 문제를 피할 수 있습니다.

### 포인트 클라우드와의 충돌 감지를 하지 않기

기본적으로 선택되어 있지 않습니다. 선택하면 **피킹된 물체\*와 \*시나리오 속의 포인트 클라우드** 사이의 충돌을 감지하지 않고 소프트웨어의 충돌 감지 계산량을 줄여 Mech-Viz의 계획 속도를 향상시키며 전체적인 사이클 타임을 최적화할 수 있습니다. 또한 피킹된 물체와 포인트 클라우드 노이즈 사이의 충돌을 잘못 감지하는 문제를 피할 수 있습니다.



- menu:충돌 감지 구성 [잡고 있는 물체 구성>잡고 있는 작업물과 다른 물체 사이의 충돌을 감지하기] 기능과 **포인트 클라우드 구성 > 포인트 클라우드와 다른 물체의 충돌을 감지하기** 기능을 동시에 활성화하면 잡고 있는 작업물 모델과 시나리오 속의 포인트 클라우드 사이의 충돌을 감지합니다.
- Mech-Vision이 포인트 클라우드와 물체 모델의 정보를 Mech-Viz로 보내면 포인트 클라우드와 물체 모델이 맞춰집니다. 로봇이 물체를 피킹한 후 모델은 계획된 경로를 따라 이동하고 피킹된 작업물 모델과 포인트 클라우드 간의 충돌이 발생합니다.
- 피킹된 작업물 모델과 포인트 클라우드 사이에서 이미 존재한 잘못된 충돌을 피할 수 없는 경우에, 이러한 충돌을 감지하면 불필요한 계산 작업량이 증가하고 Mech-Viz의 계획 시간이 늘어날 수 밖에 없습니다.

### 기본 이동 설정

“기본 이동 설정”은 이동 스텝을 사용할 때 조절해야 하는 중요한 파라미터이며 로봇이 웨이포인트 위치로 이동하는 과정의 **속도** 및 **\*운동 방식\***을 컨트롤하는 데 사용됩니다.

#### 피킹&배치

<b>지정하지 않음</b>	기본값
<b>피킹</b>	“비전 이동” 전의 이동 스텝에 적용됩니다.
<b>배치</b>	“비전 이동” 후의 이동 스텝에 적용됩니다.
<b>조절 설명</b>	Mech-Viz 프로젝트 실행 논리를 검사하는 데 사용될 수 있습니다. 작업 현장의 실제 작업 프로세스에 따라 피킹한 다음에 배치한다는 순서를 기본 원칙으로 이동 스텝에 대해 피킹&배치를 설정합니다.

#### 운동 방식

<b>관절 운동</b>	로봇의 실행 경로가 원호이기 때문에 더 원활하게 이동할 수 있어 이동 중에 싱글래티가 나타나기 쉽지 않음을 의미합니다. 관절 운동은 경로 계획의 정밀도에 대한 요구가 그리 높지 않고 로봇이 큰 범위에서 이동하는 시나리오에 적용됩니다.
<b>직선 운동</b>	로봇의 실행 경로가 직선이며 로봇 경로의 정밀도에 대한 요구가 높습니다. 용접, 접촉제 도포, 피킹 등 경로 정밀도에 대한 요구가 상대적으로 높은 시나리오에 적용됩니다.

#### 속도&가속도

로봇의 이동 속도를 결정합니다. 일반적으로 가속도 파라미터의 수치가 속도보다 작아야 합니다. 가속도 파라미터의 수치가 속도보다 클 때 로봇의 동작이 원활하지 않습니다.



피킹의 안정성을 위해 "비전 이동"과 전/후의 이동 속도를 낮게 설정해야 합니다.

#### 회전 반경

<b>기본값</b>	50.00mm
------------	---------

**조절 설명**      일반적으로 조절할 필요가 없고 기본값을 사용합니다.

회전 반경은 웨이포인트와 로봇이 회전하기 시작하는 지점 사이의 거리를 나타냅니다. 회전 반경이 클수록 로봇의 이동이 더 원활해집니다. 로봇이 비교적 작은 공간에서 이동하는 경우 회전 반경을 더 작은 값으로 설정하십시오.

로봇이 넓은 공간에서 작업하고 다른 장애물이 없으며 로봇의 두 경로 사이의 거리가 멀었을 때, 회전 반경을 적절히 크게 조절하여 로봇의 움직임을 보다 원활하게 할 수 있습니다.

### 상대적인 이동의 기준

값 리스트: 이전 웨이포인트, 다음 웨이포인트

설명: \*이전 웨이포인트\*를 선택하면 로봇은 이전 웨이포인트를 기준으로 이동합니다. \*다음 웨이포인트\*를 선택하면 로봇은 다음 웨이포인트를 기준으로 이동합니다.

### 웨이포인트 유형

말단장치, 로봇과 기준점을 선택할 수 있습니다.

### 말단장치

로봇은 “기준 웨이포인트”의 “TCP 원점”을 기준으로 움직입니다.

 “기준 웨이포인트”는 “이전 웨이포인트” 또는 “다음 웨이포인트”를 가리킵니다.

**좌표**      움직 거리를 설정하는 데 사용됩니다.

아래 그림과 같이 A는 기준 웨이포인트이고 B는 X, Y 파라미터를 300으로 설정한 후의 움직 결과입니다. 이 스텝을 수행하면 로봇은 A부터 B까지 이동합니다.

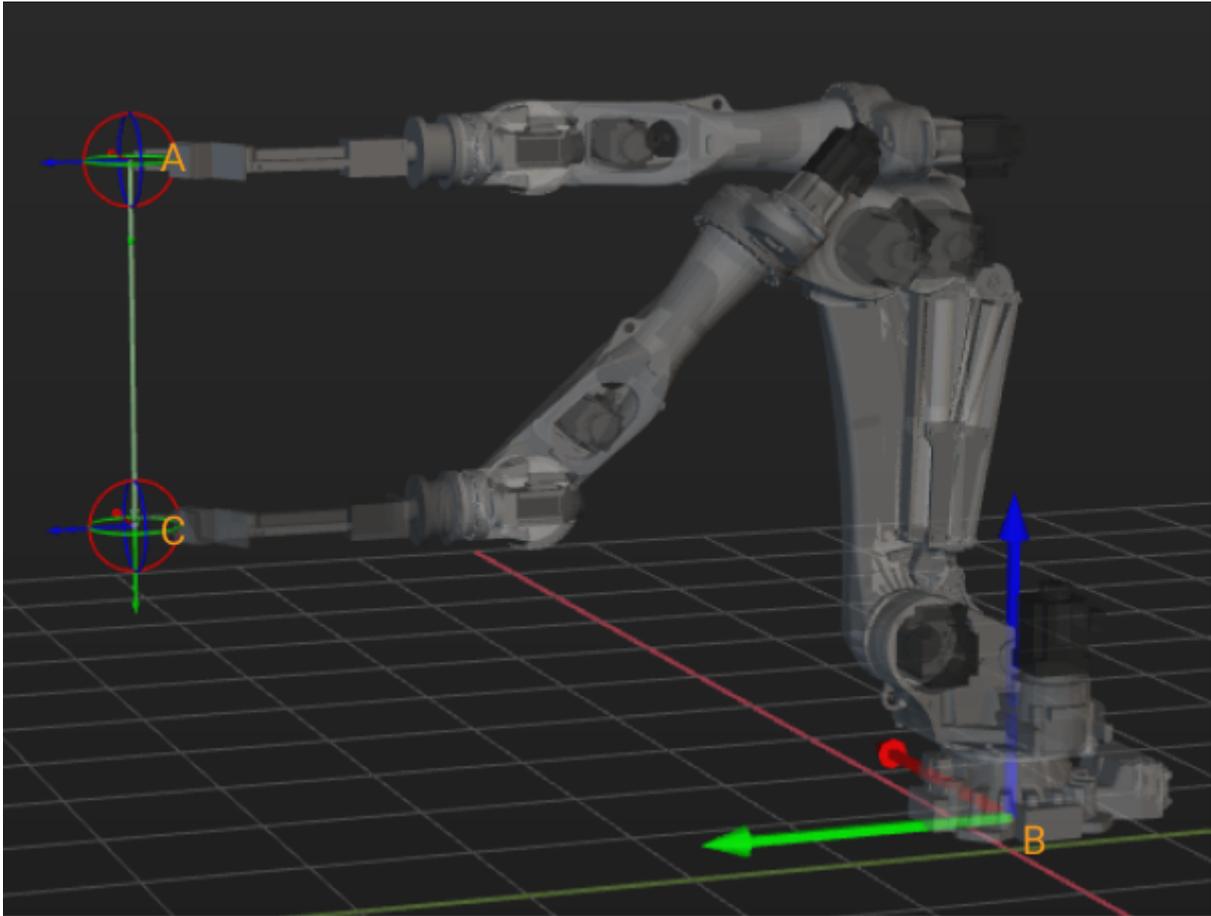
[relative move tool] | *relative\_move\_tool.png*

### 로봇

로봇은 “로봇 기준 좌표계의 방향”을 따라 기준 웨이포인트를 기준으로 움직입니다.

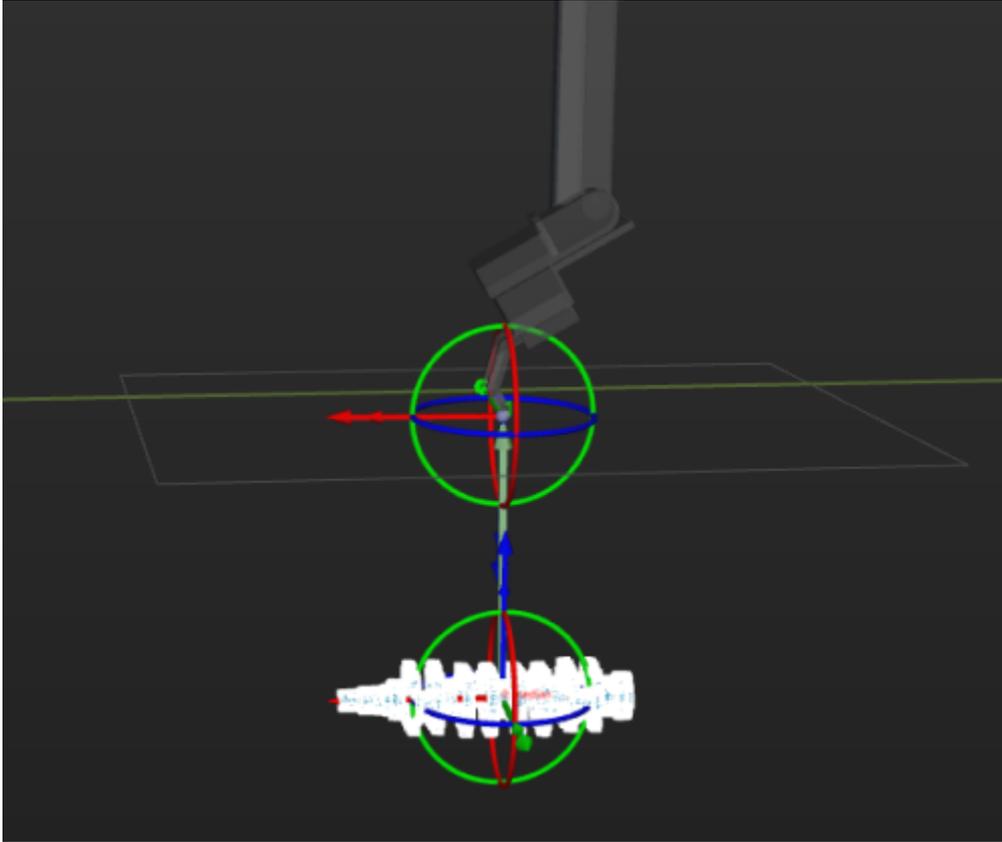
**좌표**      움직 거리를 설정하는 데 사용됩니다.

아래 그림과 같이 A는 기준 웨이포인트이고 B는 로봇 기준 좌표계의 원점이며 C는 Z 파라미터 값을 -1000으로 설정한 후의 움직 결과입니다. 이 스텝을 수행하면 로봇은 A부터 C까지 이동합니다.



### X/Y/Z 고정시키기

일반적으로 **Z축을 고정시키기** 옵션을 많이 사용하며 \*Z축을 고정시키기\*를 선택한 후 아래와 같이 평면이 나타납니다. 이 스텝은 일반적으로 **비전 이동**과 함께 사용되어 현재 “상대적인 이동” 스텝에서 웨이포인트의 Z 값을 지정된 값으로 설정할 수 있습니다. 로봇은 아래와 같이 매번 피킹 후 지정된 높이로 이동할 수 있습니다.



### 기준점

기준점을 사용하면 기준 방향을 자체 정의할 수 있습니다. 예를 들어 물체가 깊은 빈에 놓여 있고 수직 방향으로 물체를 피킹하면 빈의 표면과 충돌이 발생할 수 있는 경우, \*기준점\*을 설정하면 “상대적인 이동” 스텝은 설정된 기준 방향에서 움직입니다. 이 스텝은 일반적으로 **비전 이동**과 함께 사용됩니다.

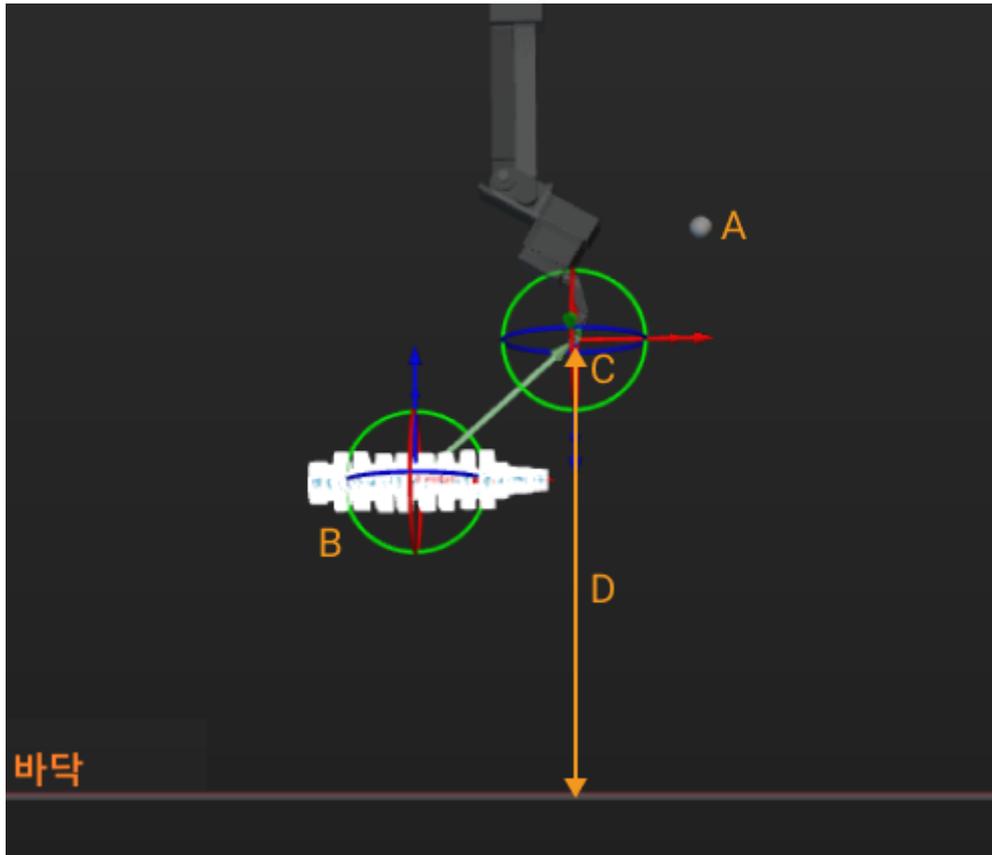
### 기준점 X/Y/Z

기준점의 위치를 설정합니다.

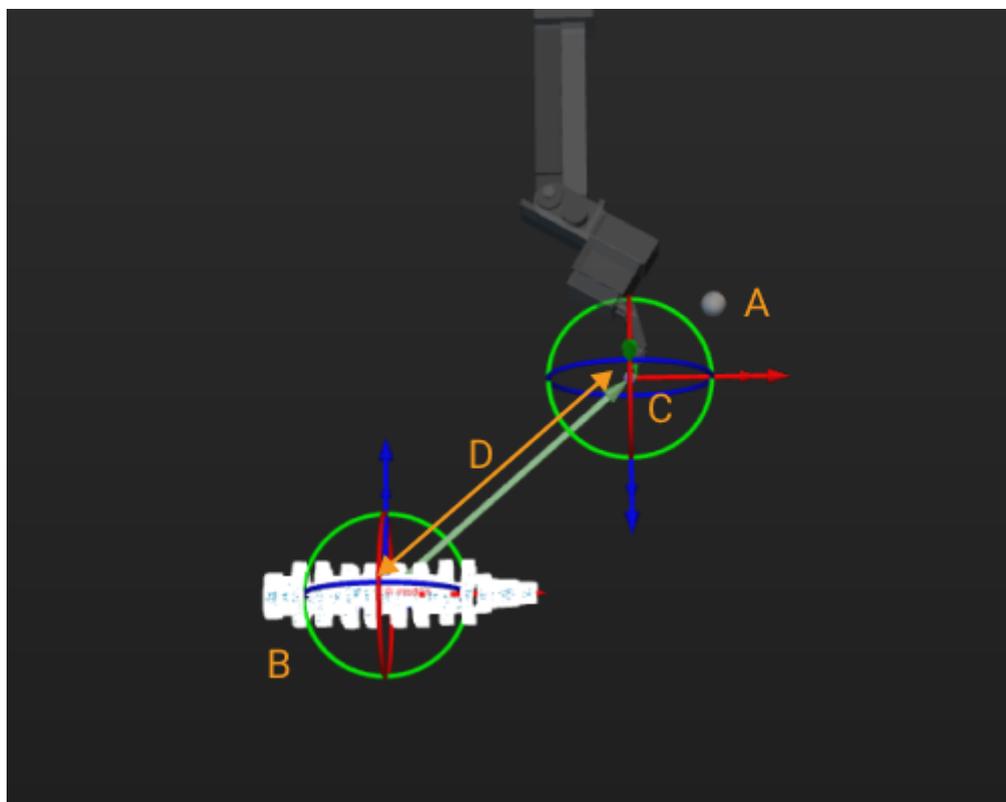
### 웨이포인트의 위치

이 파라미터는 움직이거리를 설정하는 데 사용됩니다.

- 세계 좌표계의 Z값을 기준으로: 세계 좌표계의 XOY 평면으로부터 움직이 포인트의 수직 거리를 나타냅니다. 아래 그림과 같이 A는 기준점이고 B는 “기준 웨이포인트”이며 C는 움직이 포인트이고 D는 세계 좌표계의 XOY 평면으로부터 움직이 포인트의 수직 거리입니다.



- 원래 포즈와의 거리를 기준으로: 기준 방향에서의 이동 거리입니다. 아래 그림과 같이 A는 기준점이고 B는 “기준 웨이포인트”이며 C는 옵셋 포인트이고 D는 기준 방향에서의 이동 거리입니다.



\*XOY 평면에서 이동하기\*를 선택하면 “기준 웨이포인트”의 XOY 평면에서만 옵셋 포인트를 설정할 수

있습니다.

## 5.4.2. 논리적 토폴로지

### 5.4.2.1. 길표지 분기

#### 기능 설명

이 스텝은 보통 “길표지를 설정하기” 스텝과 함께 사용합니다. 여러 분기에 공통된 부분이 있는 경우, 공통된 부분을 여러 번 복제할 필요 없이 **길표지를 설정하기** 스텝을 사용하여 현재 분기를 표기하고 공통된 부분으로 설정하며 **길표지 분기** 스텝을 사용하여 프로그램 실행 논리를 원래 분기로 복원할 수 있습니다.

#### 파라미터 설명

##### 비이동 스텝 기본 파라미터

##### 실행을 건너뛰기

<b>없음</b>	기본값. 해당 스텝의 실행을 건너뛰지 않습니다.
<b>시뮬레이션할 때만</b>	시뮬레이션을 할 때만 해당 스텝의 실행을 건너뛸니다. 아웃 포트는 <b>실행 건너뛸 때의 아웃 포트</b> 파라미터를 통해 지정됩니다.
<b>항상</b>	실제 실행 또는 시뮬레이션을 할 때 모두 해당 스텝의 실행을 건너뛸니다. 아웃 포트는 <b>실행 건너뛸 때의 아웃 포트</b> 파라미터를 통해 지정됩니다.
<b>조절 설명</b>	<b>시뮬레이션할 때만</b> 또는 *항상*으로 설정할 때 프로젝트 실행 중에는 현재 스텝을 건너뛰고 계속 다음 스텝을 실행합니다. 예를 들어, "DI 체크" 스텝이 프로젝트에 연결되어 있고 시뮬레이션 중 외부 입력 신호가 없는 경우, “실행을 건너뛰기” 파라미터를 설정하지 않으면 해당 스텝을 실행할 때 프로젝트 실행이 중지됩니다. 이 파라미터를 설정한 후에는 이 스텝을 실행하지 않고 시뮬레이션이 완료될 때까지 계속 다음 스텝을 실행합니다.

##### 실행 건너뛸 때의 아웃 포트

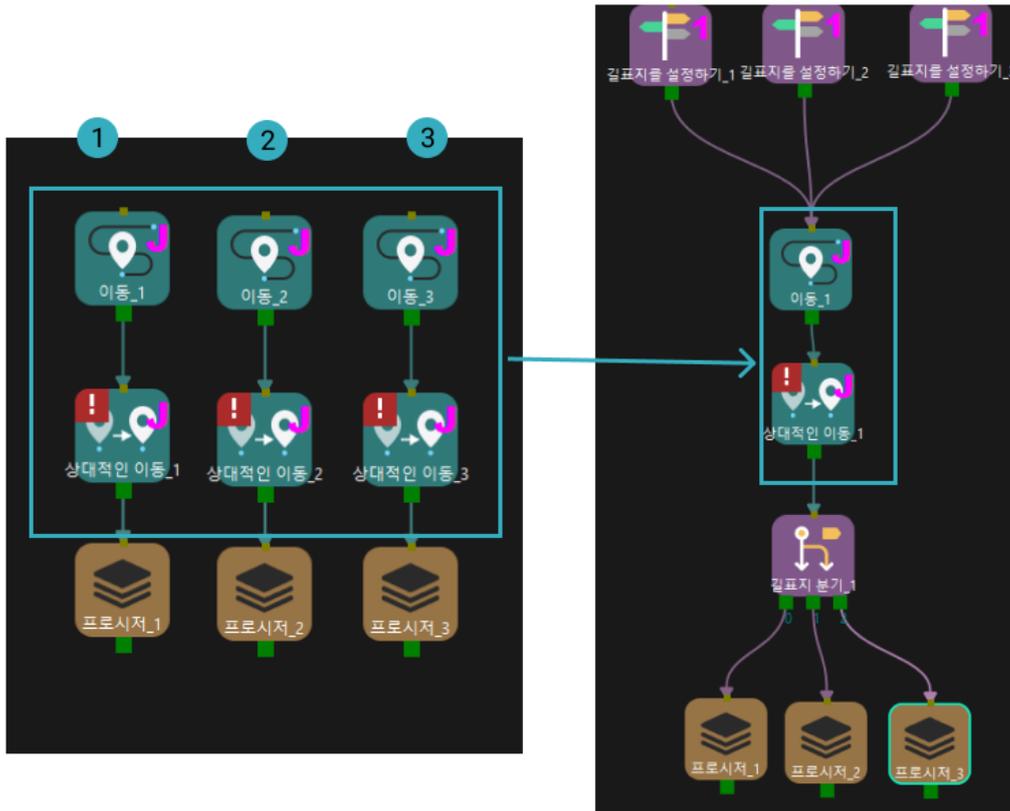
**실행을 건너뛰기** 파라미터의 값은 **시뮬레이션할 때만** 또는 \*항상\*으로 설정해야 이 기능을 사용할 수 있습니다. 해당 스텝을 건너뛰고 계속 실행될 때의 아웃 포트를 지정합니다.

<b>아웃 포트의 수량</b>	길표지 분기의 아웃 포트 수량을 설정하며 일반적으로 <b>길표지를 설정하기</b> 스텝의 *표기값*과 결합하여 사용합니다. 설정한 길표지의 수량은 아웃 포트의 수량과 일치해야 합니다. 예를 들어 길표지 2개를 설정하면 아웃 포트의 수량도 2로 설정해야 합니다. 또한 표기값 1은 <b>길표지 분기</b> 의 아웃 포트 0에 해당하고, 표기값 2는 <b>길표지 분기</b> 의 아웃 포트 1에 해당합니다.
------------------	---

#### 응용 예시

다음 그림과 같이 3개의 논리적 분기에는 동일한 스텝("이동" 및 "상대적인 이동")을 사용하며 이러한 스텝들의 파라미터는 동일한 값으로 설정됩니다.

1. "이동" 및 "상대적인 이동" 스텝의 한 세트만 유지하십시오.
2. 각 분기에 **길표지를 설정하기** 스텝을 추가하고 각 분기(1, 2, 3)에 대한 태그를 설정합니다.
3. "이동" 및 "상대적인 이동" 스텝이 실행된 후 **길표지 분기** 스텝은 현재 분기의 태그에 따라 실행 논리를 공통 스텝 이전 분기를 따라 진행됩니다.



#### 5.4.2.2. 메시지 분기

##### 응용 시나리오

- 외부 장치의 데이터에 따라 해당 분기 프로세스를 실행합니다.
- 프로그램의 실행을 중단하고 외부 신호를 수신한 후 계속 실행합니다.

##### 파라미터 설명

##### 비이동 스텝 기본 파라미터

##### 실행을 건너뛰기

없음	기본값. 해당 스텝의 실행을 건너뛰지 않습니다.
시뮬레이션할 때만	시뮬레이션을 할 때만 해당 스텝의 실행을 건너뛵니다. 아웃 포트는 <b>실행 건너뛴 때의 아웃 포트</b> 파라미터를 통해 지정됩니다.
항상	실제 실행 또는 시뮬레이션을 할 때 모두 해당 스텝의 실행을 건너뛵니다. 아웃 포트는 <b>실행 건너뛴 때의 아웃 포트</b> 파라미터를 통해 지정됩니다.
조절 설명	<b>시뮬레이션할 때만</b> 또는 *항상*으로 설정할 때 프로젝트 실행 중에는 현재 스텝을 건너뛰고 계속 다음 스텝을 실행합니다. 예를 들어, "DI 체크" 스텝이 프로젝트에 연결되어 있고 시뮬레이션 중 외부 입력 신호가 없는 경우, "실행을 건너뛰기" 파라미터를 설정하지 않으면 해당 스텝을 실행할 때 프로젝트 실행이 중지됩니다. 이 파라미터를 설정한 후에는 이 스텝을 실행하지 않고 시뮬레이션이 완료될 때까지 계속 다음 스텝을 실행합니다.

### 실행 건너뛰기 때의 아웃 포트

실행을 건너뛰기 파라미터의 값은 **시뮬레이션할 때만** 또는 **\*항상\***으로 설정해야 이 기능을 사용할 수 있습니다. 해당 스텝을 건너뛰고 계속 실행될 때의 아웃 포트를 지정합니다.

**아웃 포트의 수량** 이 스텝의 아웃 포트 수입입니다. 예를 들어 분기 수는 3개이고 이 스텝에는 3개의 아웃 포트가 있으며 각각 3개의 프로세스에 연결할 수 있습니다. 이 스텝을 사용하여 프로그램 실행을 중단하려면 분기 수를 1로 설정하면 됩니다.

\*아웃 포트의 수량\*은 1로 설정되면 아래와 같은 파라미터들을 설정할 수 있습니다.

#### 후속 이동에 영향을 미치기(계획을 중단할 수 있음)

이 파라미터를 선택하면 소프트웨어 후속 논리 계획을 중단할 수도 있습니다. 이 스텝은 다른 분기를 수행하는 데 사용되는 경우, 후속 단계에서 어느 분기를 통해 수행될지 모르기 때문에 프로그램 계획에 영향을 끼칠 수 밖에 없습니다. 하지만 외부 신호를 중단하고 기다리는 데만 사용되면 후속 프로그램의 논리는 변하지 않으므로 이 파라미터를 선택하지 않습니다.

#### 계획 완료 후 미리 실행하기

소프트웨어는 외부 신호를 임시로 저장했다가 이 스텝까지 실행되면 해당 분기를 직접 실행하여사이클 타임을 개선할 수 있습니다. 이 옵션을 언체크하면 이 스텝은 외부 장치에서 보내는 신호를 미리 수신하지 않습니다.

#### 타임아웃

이 파라미터를 선택하면 **\*제한 시간\***을 설정할 수 있습니다. 외부 신호를 기다리는 시간이 **\*제한 시간\***을 초과한 경우 타임아웃 분기 이후에 프로세스를 실행합니다. 또한 사전 계획의 아웃 포트를 지정할 수도 있습니다. 일반적으로 타임아웃 시의 아웃 포트가 아닌 정상적인 아웃 포트를 선택합니다.

\*아웃 포트의 수량\*이 1이 아니면 아래와 같은 파라미터들을 설정할 수 있습니다.

#### 계획 완료 후 미리 실행하기

소프트웨어는 외부 신호를 임시로 저장했다가 이 스텝까지 실행되면 해당 분기를 직접 실행하여사이클 타임을 개선할 수 있습니다. 이 옵션을 언체크하면 이 스텝은 외부 장치에서 보내는 신호를 미리 수신하지 않습니다.

#### 사전 계획 아웃 포트

계획의 연속성을 보장하기 위해서는 사전 계획의 아웃 포트를 지정해야 합니다. 일반적으로 다시 계획의 확률을 줄이고 로봇의 대기 시간을 줄이기 위해 실행 가능성이 가장 큰 분기를 지정하는 것이 좋습니다.

#### 타임아웃

이 파라미터를 선택하면 **\*제한 시간\***을 설정할 수 있습니다. 외부 신호를 기다리는 시간이 **\*제한 시간\***을 초과한 경우 타임아웃 분기 이후에 프로세스를 실행합니다.

### 5.4.2.3. 프로시저

#### 기능 설명

여러 스텝을 하나의 프로시저로 그룹화할 수 있습니다. 프로시저 내에 또 프로시저를 중첩하는 것도 가능합니다. 프로시저 내부의 워크플로우에 가능한 결과가 여러 개인 경우 **프로시저 아웃 포트** 스텝과 함께 사용하십시오.

## 파라미터 설명

없습니다.

## 명칭

사용자가 자체 정의한 프로시저의 명칭입니다.

### 5.4.2.4. 프로시저 아웃 포트

## 기능 설명

“프로시저” 스텝을 종료할 때 사용하는 아웃 포트를 지정하는 데 사용됩니다.

## 응용 시나리오

프로시저 내부의 워크플로우에 가능한 결과가 여러 개인 경우 \*프로시저 아웃 포트\*를 사용하여 사용할 종료 포트를 선택하십시오.

## 파라미터 설명

### 비이동 스텝 기본 파라미터

#### 실행을 건너뛰기

<b>없음</b>	기본값. 해당 스텝의 실행을 건너뛰지 않습니다.
<b>시뮬레이션할 때만</b>	시뮬레이션을 할 때만 해당 스텝의 실행을 건너뛸니다. 아웃 포트는 <b>실행 건너뛸 때의 아웃 포트</b> 파라미터를 통해 지정됩니다.
<b>항상</b>	실제 실행 또는 시뮬레이션을 할 때 모두 해당 스텝의 실행을 건너뛸니다. 아웃 포트는 <b>실행 건너뛸 때의 아웃 포트</b> 파라미터를 통해 지정됩니다.
<b>조절 설명</b>	<b>시뮬레이션할 때만</b> 또는 *항상*으로 설정할 때 프로젝트 실행 중에는 현재 스텝을 건너뛰고 계속 다음 스텝을 실행합니다. 예를 들어, "DI 체크" 스텝이 프로젝트에 연결되어 있고 시뮬레이션 중 외부 입력 신호가 없는 경우, “실행을 건너뛰기” 파라미터를 설정하지 않으면 해당 스텝을 실행할 때 프로젝트 실행이 중지됩니다. 이 파라미터를 설정한 후에는 이 스텝을 실행하지 않고 시뮬레이션이 완료될 때까지 계속 다음 스텝을 실행합니다.

#### 실행 건너뛸 때의 아웃 포트

**실행을 건너뛰기** 파라미터의 값은 **시뮬레이션할 때만** 또는 \*항상\*으로 설정해야 이 기능을 사용할 수 있습니다. 해당 스텝을 건너뛰고 계속 실행될 때의 아웃 포트를 지정합니다.

#### 종료 시의 포트 번호

<b>기본값</b>	0
<b>조절 설명</b>	수행할 “프로시저”의 아웃 포트를 선택하십시오. 각 아웃 포트는 다른 후속 실행 논리에 연결됩니다.

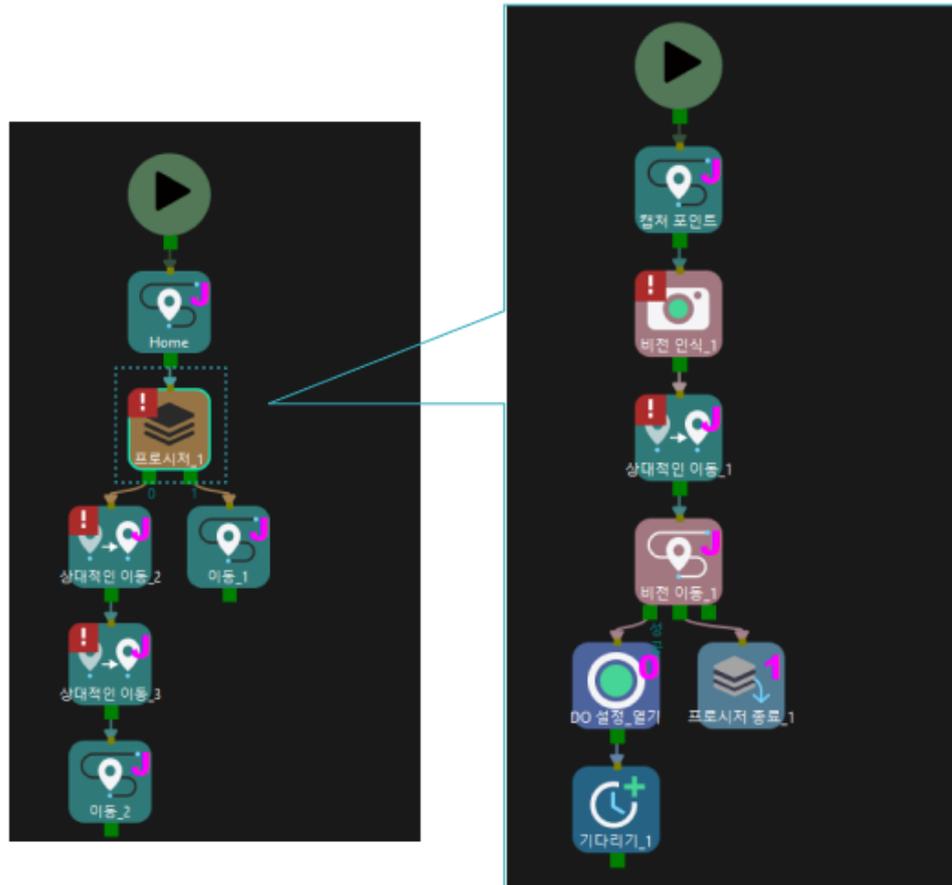
종료 시의 포트 번호는 1부터 순서대로 설정해야 합니다. 아웃 포트 0은 프로시저에서 **프로시저 아웃 포트** 스텝과 연결되지 않은 분기에 해당합니다. 자세한 내용은 “응용 예시”를 참조하십시오.

## 응용 예시

프로시저의 "비전 이동" 스텝에는 성공, 계획 실패 및 기타 실패의 세 가지 가능한 결과가 있습니다. 성공 및 실패에 대한 후속 논리에 연결하려면 프로시저에 두 개의 아웃 포트가 필요합니다.

“비전 이동” 스텝의 “계획 실패” 아웃 포트에 **프로시저 아웃 포트** 스텝을 연결하고 \*프로시저 아웃 포트\*의 \*종료 시의 포트 번호\*를 1로 설정합니다. 이때 프로시저에 아웃 포트 1, 즉 ‘비전 이동’ 스텝이 계획 실패 시 실행하는 아웃 포트가 추가됩니다.

“비전 이동” 스텝의 “성공” 아웃 포트는 \*프로시저 아웃 포트\*와 연결되어 있지 않으며, 이 분기가 실행되면 아웃 포트 0을 통해 프로시저를 종료하게 됩니다.



### 5.4.2.5. 길표지를 설정하기

#### 기능 설명

이 스텝은 보통 “길표지 분기” 스텝과 함께 사용됩니다. 여러 분기에 공통된 부분이 있는 경우, 공통된 부분을 여러 번 복제할 필요 없이 **길표지를 설정하기** 스텝을 사용하여 현재 분기를 표기하고 공통된 부분으로 설정하며 **길표지 분기** 스텝을 사용하여 프로그램 실행 논리를 원래 분기로 복원할 수 있습니다.

#### 파라미터 설명

##### 비이동 스텝 기본 파라미터

### 실행을 건너뛰기

<b>없음</b>	기본값. 해당 스텝의 실행을 건너뛰지 않습니다.
<b>시뮬레이션할 때만</b>	시뮬레이션을 할 때만 해당 스텝의 실행을 건너뛩니다. 아웃 포트는 <b>실행 건너뛩 때의 아웃 포트</b> 파라미터를 통해 저장됩니다.
<b>항상</b>	실제 실행 또는 시뮬레이션을 할 때 모두 해당 스텝의 실행을 건너뛩니다. 아웃 포트는 <b>실행 건너뛩 때의 아웃 포트</b> 파라미터를 통해 저장됩니다.
<b>조절 설명</b>	<b>시뮬레이션할 때만</b> 또는 *항상*으로 설정할 때 프로젝트 실행 중에는 현재 스텝을 건너뛩고 계속 다음 스텝을 실행합니다. 예를 들어, "DI 체크" 스텝이 프로젝트에 연결되어 있고 시뮬레이션 중 외부 입력 신호가 없는 경우, "실행을 건너뛩기" 파라미터를 설정하지 않으면 해당 스텝을 실행할 때 프로젝트 실행이 중지됩니다. 이 파라미터를 설정한 후에는 이 스텝을 실행하지 않고 시뮬레이션이 완료될 때까지 계속 다음 스텝을 실행합니다.

### 실행 건너뛩 때의 아웃 포트

**실행을 건너뛩기** 파라미터의 값은 **시뮬레이션할 때만** 또는 \*항상\*으로 설정해야 이 기능을 사용할 수 있습니다. 해당 스텝을 건너뛩고 계속 실행될 때의 아웃 포트를 지정합니다.

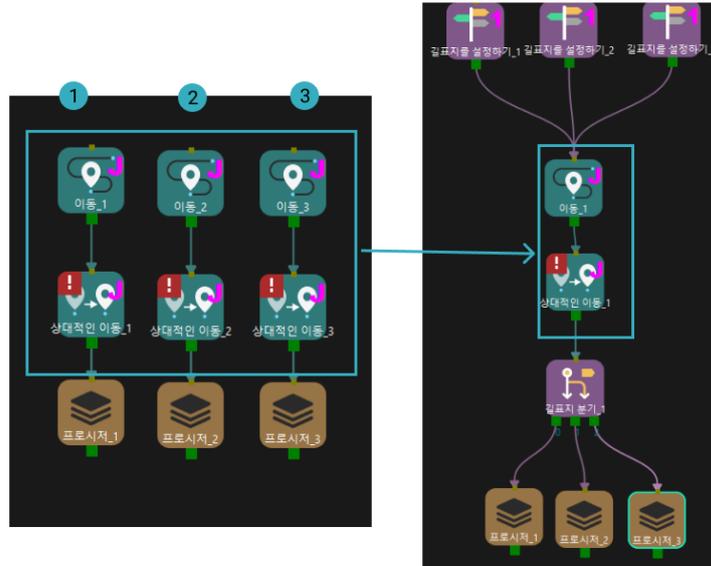
### 표기값

현재 논리 분기의 번호를 설정하는 데 사용되며 일반적으로 "길표지 분기" 스텝의 **아웃 포트의 수량** 파라미터를 결합하여 설정해야 합니다. 아웃 포트의 수량은 설정한 분기 길표지의 수와 같아야 합니다. 예를 들어 2개의 길표지가 설정된 경우 아웃 포트의 수량도 2로 설정되어야 합니다. 또한 표기값 1은 "길표지 분기"의 아웃 포트 0에 해당하고, 표기값 2는 "길표지 분기"의 아웃 포트 1에 해당합니다.

### 응용 예시

다음 그림과 같이 3개의 논리적 분기에는 동일한 스텝("이동" 및 "상대적인 이동")을 사용하며 이러한 스텝들의 파라미터는 동일한 값으로 설정됩니다.

1. "이동" 및 "상대적인 이동" 스텝의 한 세트만 유지하십시오.
2. 각 분기에 **길표지를 설정하기** 스텝을 추가하고 각 분기(1, 2, 3)에 대한 태그를 설정합니다.
3. "이동" 및 "상대적인 이동" 스텝이 실행된 후 **길표지 분기** 스텝은 현재 분기의 태그에 따라 실행 논리를 공통 스텝 이전 분기를 따라 진행됩니다.



### 5.4.3. 로봇 유틸리티

#### 5.4.3.1. 말단장치를 체크하기

##### 기능 설명

말단장치를 바꿔야 하는 시나리오에 적용됩니다. 이 스텝은 선택된 말단장치가 현재 말단장치와 일치하는지를 체크하는 데 사용되며 일치하면 왼쪽 아웃 포트(same as current tcp)를 통해 실행되고 말단장치를 바꾸는 이동 또는 비이동 스텝을 실행할 필요가 없습니다. 일치하지 않으면 오른쪽 아웃 포트(different from current tcp)를 통해 실행되고 "말단장치를 바꾸기" 등 스텝을 실행해야 합니다.

##### 파라미터 설명

##### 비이동 스텝 기본 파라미터

##### 실행을 건너뛰기

없음	기본값. 해당 스텝의 실행을 건너뛰지 않습니다.
시뮬레이션할 때만	시뮬레이션을 할 때만 해당 스텝의 실행을 건너뛵니다. 아웃 포트는 <b>실행 건너뛴 때의 아웃 포트</b> 파라미터를 통해 지정됩니다.
항상	실제 실행 또는 시뮬레이션을 할 때 모두 해당 스텝의 실행을 건너뛵니다. 아웃 포트는 <b>실행 건너뛴 때의 아웃 포트</b> 파라미터를 통해 지정됩니다.
조절 설명	<b>시뮬레이션할 때만</b> 또는 *항상*으로 설정할 때 프로젝트 실행 중에는 현재 스텝을 건너뛰고 계속 다음 스텝을 실행합니다. 예를 들어, "DI 체크" 스텝이 프로젝트에 연결되어 있고 시뮬레이션 중 외부 입력 신호가 없는 경우, "실행을 건너뛰기" 파라미터를 설정하지 않으면 해당 스텝을 실행할 때 프로젝트 실행이 중지됩니다. 이 파라미터를 설정한 후에는 이 스텝을 실행하지 않고 시뮬레이션이 완료될 때까지 계속 다음 스텝을 실행합니다.

##### 실행 건너뛴 때의 아웃 포트

**실행을 건너뛰기** 파라미터의 값은 **시뮬레이션할 때만** 또는 \*항상\*으로 설정해야 이 기능을 사용할 수 있습니다. 해당 스텝을 건너뛰고 계속 실행될 때의 아웃 포트를 지정합니다.

## 말단장치를 구성하기

원하는 말단장치 모델의 이름을 선택하여 현재 말단장치와 원하는 것과 일치하는지를 체크합니다.

### 5.4.3.2. 관절 각도를 획득하기

#### 기능 설명

현재 연결된 로봇의 관절 각도 정보를 획득하고 관절 각도 정보를 Mech-Viz에 동기화합니다.

#### 응용 시나리오

Mech-Viz가 로봇의 모든 움직임을 완전히 컨트롤하지 못하는 경우, 이 스텝을 사용하여 로봇이 자율적으로 이동한 후의 관절 각도를 획득하고 로봇에 대한 컨트롤을 유지합니다.

#### 파라미터 설명

##### 비이동 스텝 기본 파라미터

##### 실행을 건너뛰기

없음	기본값. 해당 스텝의 실행을 건너뛰지 않습니다.
시뮬레이션할 때만	시뮬레이션을 할 때만 해당 스텝의 실행을 건너뛵니다. 아웃 포트는 <b>실행 건너뛴 때의 아웃 포트</b> 파라미터를 통해 지정됩니다.
항상	실제 실행 또는 시뮬레이션을 할 때 모두 해당 스텝의 실행을 건너뛵니다. 아웃 포트는 <b>실행 건너뛴 때의 아웃 포트</b> 파라미터를 통해 지정됩니다.
조절 설명	<b>시뮬레이션할 때만</b> 또는 *항상*으로 설정할 때 프로젝트 실행 중에는 현재 스텝을 건너뛰고 계속 다음 스텝을 실행합니다. 예를 들어, "DI 체크" 스텝이 프로젝트에 연결되어 있고 시뮬레이션 중 외부 입력 신호가 없는 경우, "실행을 건너뛰기" 파라미터를 설정하지 않으면 해당 스텝을 실행할 때 프로젝트 실행이 중지됩니다. 이 파라미터를 설정한 후에는 이 스텝을 실행하지 않고 시뮬레이션이 완료될 때까지 계속 다음 스텝을 실행합니다.

##### 실행 건너뛴 때의 아웃 포트

**실행을 건너뛰기** 파라미터의 값은 **시뮬레이션할 때만** 또는 \*항상\*으로 설정해야 이 기능을 사용할 수 있습니다. 해당 스텝을 건너뛰고 계속 실행될 때의 아웃 포트를 지정합니다.

#### 응용 예시

**비전 인식** 스텝은 해당 Mech-Vision 프로젝트를 트리거하기 전에 **관절 각도를 획득하기** 스텝을 통해 실제 로봇의 관절 각도를 획득하고 이를 경로 계획의 시작 위치로 사용하기 위해 Mech-Viz에 동기화합니다.



### 5.4.3.3. 페이로드 설정

#### 기능 설명

이 스텝은 로봇의 탑재하중을 설정하는 데 사용됩니다.

#### 파라미터 설명

#### 비이동 스텝 기본 파라미터

#### 실행을 건너뛰기

없음	기본값. 해당 스텝의 실행을 건너뛰지 않습니다.
시뮬레이션할 때만	시뮬레이션을 할 때만 해당 스텝의 실행을 건너뛵니다. 아웃 포트는 <b>실행 건너뛸 때의 아웃 포트</b> 파라미터를 통해 지정됩니다.
항상	실제 실행 또는 시뮬레이션을 할 때 모두 해당 스텝의 실행을 건너뛵니다. 아웃 포트는 <b>실행 건너뛸 때의 아웃 포트</b> 파라미터를 통해 지정됩니다.
조절 설명	<b>시뮬레이션할 때만</b> 또는 *항상*으로 설정할 때 프로젝트 실행 중에는 현재 스텝을 건너뛰고 계속 다음 스텝을 실행합니다. 예를 들어, "DI 체크" 스텝이 프로젝트에 연결되어 있고 시뮬레이션 중 외부 입력 신호가 없는 경우, "실행을 건너뛰기" 파라미터를 설정하지 않으면 해당 스텝을 실행할 때 프로젝트 실행이 중지됩니다. 이 파라미터를 설정한 후에는 이 스텝을 실행하지 않고 시뮬레이션이 완료될 때까지 계속 다음 스텝을 실행합니다.

#### 실행 건너뛸 때의 아웃 포트

**실행을 건너뛰기** 파라미터의 값은 **시뮬레이션할 때만** 또는 \*항상\*으로 설정해야 이 기능을 사용할 수 있습니다. 해당 스텝을 건너뛰고 계속 실행될 때의 아웃 포트를 지정합니다.

#### 페이로드 ID

페이로드가 여러 개인 경우에 사용하며 페이로드가 하나만 있는 경우 기본값을 사용합니다.

## 페이로드

말단장치와 작업물을 포함한 최대 탑재하중(kg)을 입력하십시오.

## COG 위치

X/Y/Z: 플랜지 좌표계에서 페이로드의 무게 중심을 설정합니다(일부 로봇 브랜드에서만 필요함).

### 5.4.3.4. 피킹 상태 설정

#### 기능 설명

이 스텝을 사용하여 피킹 상태를 설정하며 일반적으로 피킹에 실패한 후 피킹 상태를 업데이트하는 데 사용됩니다.

#### 응용 시나리오

이 스텝은 피킹 상태를 업데이트하기 위해 이동 유형 스텝의 “피킹&배치” 파라미터와 함께 사용됩니다.

#### 파라미터 설명

##### 비이동 스텝 기본 파라미터

##### 실행을 건너뛰기

없음	기본값. 해당 스텝의 실행을 건너뛰지 않습니다.
시뮬레이션할 때만	시뮬레이션을 할 때만 해당 스텝의 실행을 건너뛵니다. 아웃 포트는 <b>실행 건너뛴 때의 아웃 포트</b> 파라미터를 통해 지정됩니다.
항상	실제 실행 또는 시뮬레이션을 할 때 모두 해당 스텝의 실행을 건너뛵니다. 아웃 포트는 <b>실행 건너뛴 때의 아웃 포트</b> 파라미터를 통해 지정됩니다.
조절 설명	<b>시뮬레이션할 때만</b> 또는 *항상*으로 설정할 때 프로젝트 실행 중에는 현재 스텝을 건너뛰고 계속 다음 스텝을 실행합니다. 예를 들어, "DI 체크" 스텝이 프로젝트에 연결되어 있고 시뮬레이션 중 외부 입력 신호가 없는 경우, “실행을 건너뛰기” 파라미터를 설정하지 않으면 해당 스텝을 실행할 때 프로젝트 실행이 중지됩니다. 이 파라미터를 설정한 후에는 이 스텝을 실행하지 않고 시뮬레이션이 완료될 때까지 계속 다음 스텝을 실행합니다.

##### 실행 건너뛴 때의 아웃 포트

**실행을 건너뛰기** 파라미터의 값은 **시뮬레이션할 때만** 또는 \*항상\*으로 설정해야 이 기능을 사용할 수 있습니다. 해당 스텝을 건너뛰고 계속 실행될 때의 아웃 포트를 지정합니다.

#### 피킹 상태

프로젝트의 실제 상황에 따라 “물체를 잡고 있음” 또는 “물체를 잡고 있지 않음”을 선택하면 됩니다.

### 5.4.3.5. 말단장치를 바꾸기

#### 기능 설명

이 스텝은 다른 TCP로 전환해야 하는 시나리오에 사용됩니다. 피킹 작업을 수행하기 전에 이 스텝을 수행하면 올바른 말단장치를 선택하고 다른 TCP로 전환하여 피킹할 수 있습니다.

## 파라미터 설명

### 비이동 스텝 기본 파라미터

#### 실행을 건너뛰기

<b>없음</b>	기본값. 해당 스텝의 실행을 건너뛰지 않습니다.
<b>시뮬레이션할 때만</b>	시뮬레이션을 할 때만 해당 스텝의 실행을 건너뛵니다. 아웃 포트는 <b>실행 건너뛴 때의 아웃 포트</b> 파라미터를 통해 지정됩니다.
<b>항상</b>	실제 실행 또는 시뮬레이션을 할 때 모두 해당 스텝의 실행을 건너뛵니다. 아웃 포트는 <b>실행 건너뛴 때의 아웃 포트</b> 파라미터를 통해 지정됩니다.
<b>조절 설명</b>	<b>시뮬레이션할 때만</b> 또는 *항상*으로 설정할 때 프로젝트 실행 중에는 현재 스텝을 건너뛰고 계속 다음 스텝을 실행합니다. 예를 들어, "DI 체크" 스텝이 프로젝트에 연결되어 있고 시뮬레이션 중 외부 입력 신호가 없는 경우, "실행을 건너뛰기" 파라미터를 설정하지 않으면 해당 스텝을 실행할 때 프로젝트 실행이 중지됩니다. 이 파라미터를 설정한 후에는 이 스텝을 실행하지 않고 시뮬레이션이 완료될 때까지 계속 다음 스텝을 실행합니다.

#### 실행 건너뛴 때의 아웃 포트

실행을 건너뛰기 파라미터의 값은 **시뮬레이션할 때만** 또는 \*항상\*으로 설정해야 이 기능을 사용할 수 있습니다. 해당 스텝을 건너뛰고 계속 실행될 때의 아웃 포트를 지정합니다.

#### 말단장치 구성

플다운 메뉴에 추가된 말단장치가 표시되며 현재 사용할 말단장치를 선택합니다.

### 5.4.3.6. 제어권 이전

#### 기능 설명

이 스텝은 일시적으로 제어권을 로봇으로 이전하고 제어권이 다시 전송될 때까지 대기하는 데 사용됩니다.

## 파라미터 설명

### 비이동 스텝 기본 파라미터

#### 실행을 건너뛰기

<b>없음</b>	기본값. 해당 스텝의 실행을 건너뛰지 않습니다.
<b>시뮬레이션할 때만</b>	시뮬레이션을 할 때만 해당 스텝의 실행을 건너뛵니다. 아웃 포트는 <b>실행 건너뛴 때의 아웃 포트</b> 파라미터를 통해 지정됩니다.
<b>항상</b>	실제 실행 또는 시뮬레이션을 할 때 모두 해당 스텝의 실행을 건너뛵니다. 아웃 포트는 <b>실행 건너뛴 때의 아웃 포트</b> 파라미터를 통해 지정됩니다.
<b>조절 설명</b>	<b>시뮬레이션할 때만</b> 또는 *항상*으로 설정할 때 프로젝트 실행 중에는 현재 스텝을 건너뛰고 계속 다음 스텝을 실행합니다. 예를 들어, "DI 체크" 스텝이 프로젝트에 연결되어 있고 시뮬레이션 중 외부 입력 신호가 없는 경우, "실행을 건너뛰기" 파라미터를 설정하지 않으면 해당 스텝을 실행할 때 프로젝트 실행이 중지됩니다. 이 파라미터를 설정한 후에는 이 스텝을 실행하지 않고 시뮬레이션이 완료될 때까지 계속 다음 스텝을 실행합니다.

### 실행 건너뛰기 때의 아웃 포트

실행을 건너뛰기 파라미터의 값은 **시뮬레이션할 때만** 또는 **\*항상\***으로 설정해야 이 기능을 사용할 수 있습니다. 해당 스텝을 건너뛰고 계속 실행될 때의 아웃 포트를 지정합니다.

#### 아웃 포트의 수량

스텝을 종료할 때 아웃 포트의 수량입니다.

#### 관절 각도를 체크하기

제어권을 이전하기 전에 로봇의 관절 각도를 체크할지 여부를 설정합니다.

#### 제어권 이전하기 전의 관절 각도

**관절 각도를 체크하기** 옵션을 선택한 후 이 파라미터가 표시됩니다. 체크할 관절 각도 값을 입력하십시오.

## 5.4.4. 서비스

### 5.4.4.1. 알림

#### 기능 설명

외부 서비스에 자체 정의한 메시지를 전송하는 데 사용됩니다.

#### 응용 시나리오

이 스텝은 일반적으로 통신을 위한 외부 장치가 필요한 시나리오에서 사용됩니다.

#### 파라미터 설명

##### 비이동 스텝 기본 파라미터

##### 실행을 건너뛰기

<b>없음</b>	기본값. 해당 스텝의 실행을 건너뛰지 않습니다.
<b>시뮬레이션할 때만</b>	시뮬레이션을 할 때만 해당 스텝의 실행을 건너뛵니다. 아웃 포트는 <b>실행 건너뛰기 때의 아웃 포트</b> 파라미터를 통해 지정됩니다.
<b>항상</b>	실제 실행 또는 시뮬레이션을 할 때 모두 해당 스텝의 실행을 건너뛵니다. 아웃 포트는 <b>실행 건너뛰기 때의 아웃 포트</b> 파라미터를 통해 지정됩니다.
<b>조절 설명</b>	<b>시뮬레이션할 때만</b> 또는 <b>*항상*</b> 으로 설정할 때 프로젝트 실행 중에는 현재 스텝을 건너뛰고 계속 다음 스텝을 실행합니다. 예를 들어, "DI 체크" 스텝이 프로젝트에 연결되어 있고 시뮬레이션 중 외부 입력 신호가 없는 경우, "실행을 건너뛰기" 파라미터를 설정하지 않으면 해당 스텝을 실행할 때 프로젝트 실행이 중지됩니다. 이 파라미터를 설정한 후에는 이 스텝을 실행하지 않고 시뮬레이션이 완료될 때까지 계속 다음 스텝을 실행합니다.

### 실행 건너뛰기 때의 아웃 포트

실행을 건너뛰기 파라미터의 값은 **시뮬레이션할 때만** 또는 **\*항상\***으로 설정해야 이 기능을 사용할 수 있습니다. 해당 스텝을 건너뛰고 계속 실행될 때의 아웃 포트를 지정합니다.

### Adapter 서비스 명칭

Adapter 서비스 명칭\*을 통해 이 스텝의 정보를 획득합니다. 프로젝트에 여러 \*알림 스텝이 있는 경우, 모든 알림 스텝은 동일한 \*서비스 명칭\*을 사용해야 합니다.

### 메시지

자체 정의한 메시지입니다. Adapter는 메시지 내용에 따라 다른 논리를 실행합니다. 예를 들어 “분기 실행 종료”의 뜻을 전송하려면 메시지는 \*finish\*로 설정됩니다.

### 실패 시 처리

설명: 메시지를 전송하는 데 실패하면 소프트웨어의 처리입니다.

값:

- Warning: 경고 알림(기본값).
- Retry: 재시도.
- Throw: 해당 스텝을 버리기.

### 로봇 정지 필요

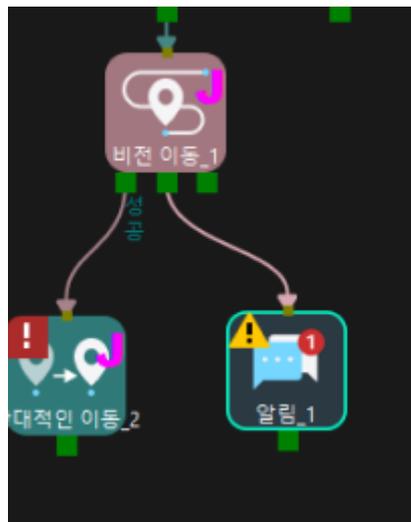
이 파라미터를 선택하면 프로젝트가 이 스텝까지 실행될 때 로봇이 일시 정지됩니다. 이 파라미터를 선택하지 않으면 메시지를 보내는 동안 로봇은 계속 움직입니다.

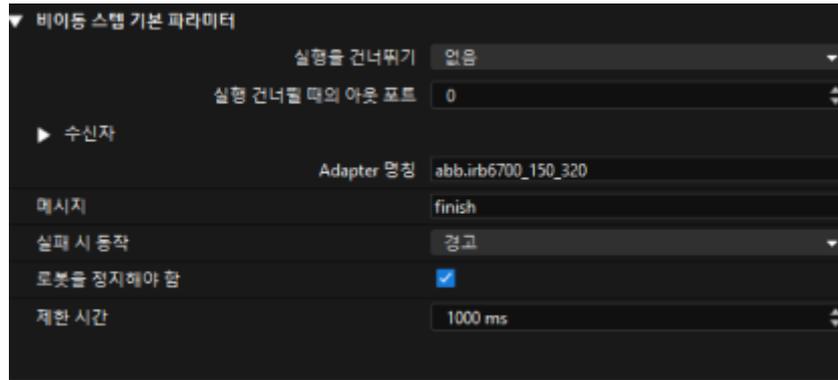
### 제한 시간

메시지가 이 제한 시간(ms) 내에 전송되지 않으면 로봇이 \*실패 시 처리\*를 수행합니다.

### 응용 예시

\*서비스 명칭\*을 설정하여 계획 실패의 메시지를 Adapter로 전송하며 메시지 내용은 \*finish\*입니다.





## 5.4.5. 도구

### 5.4.5.1. 분류

#### 기능 설명

이 스텝은 Mech-Vision에서 인식된 작업물 레이블에 따라 서로 다른 분기를 선택하기 위해 “비전 이동” 스텝 전/후에 사용됩니다.

#### 응용 시나리오

Mech-Vision에서 인식된 작업물 레이블에 따라 분류합니다.

- “분류” 스텝은 “비전 이동” 스텝 전에 사용되는 경우, 기본적인 피킹 과정은 다음과 같습니다: 물체 인식 > 다른 물체 레이블에 따라 서로 다른 분기를 선택해서 작업을 수행하기 > 해당 레이블과 대응하는 물물체를 피킹하기. 예시: 다른 작업물을 피킹할 때 말단장치를 변경합니다.
- “분류” 스텝은 “비전 이동” 스텝 후에 사용되는 경우, 기본적인 피킹 과정은 다음과 같습니다: 물체 인식 > 물체 피킹 > 물체 종류에 따라 서로 다른 분류 작업을 수행하기. 예시: 동일한 말단장치를 사용하여 작업물을 피킹한 후 다른 빈에 배치합니다.

#### 파라미터 설명

##### 비이동 스텝 기본 파라미터

##### 실행을 건너뛰기

없음	기본값. 해당 스텝의 실행을 건너뛰지 않습니다.
시뮬레이션할 때만	시뮬레이션을 할 때만 해당 스텝의 실행을 건너뛵니다. 아웃 포트는 <b>실행 건너뛴 때의 아웃 포트</b> 파라미터를 통해 지정됩니다.
항상	실제 실행 또는 시뮬레이션을 할 때 모두 해당 스텝의 실행을 건너뛵니다. 아웃 포트는 <b>실행 건너뛴 때의 아웃 포트</b> 파라미터를 통해 지정됩니다.

##### 조절 설명

**시뮬레이션할 때만** 또는 \*항상\*으로 설정할 때 프로젝트 실행 중에는 현재 스텝을 건너뛰고 계속 다음 스텝을 실행합니다. 예를 들어, "DI 체크" 스텝이 프로젝트에 연결되어 있고 시뮬레이션 중 외부 입력 신호가 없는 경우, “실행을 건너뛰기” 파라미터를 설정하지 않으면 해당 스텝을 실행할 때 프로젝트 실행이 중지됩니다. 이 파라미터를 설정한 후에는 이 스텝을 실행하지 않고 시뮬레이션이 완료될 때까지 계속 다음 스텝을 실행합니다.

### 실행 건너뛰기 때의 아웃 포트

실행을 건너뛰기 파라미터의 값은 **시뮬레이션할 때만** 또는 **\*항상\***으로 설정해야 이 기능을 사용할 수 있습니다. 해당 스텝을 건너뛰고 계속 실행될 때의 아웃 포트를 지정합니다.

#### “피킹된 물체를 업데이트하기”에서 레이블을 획득하기

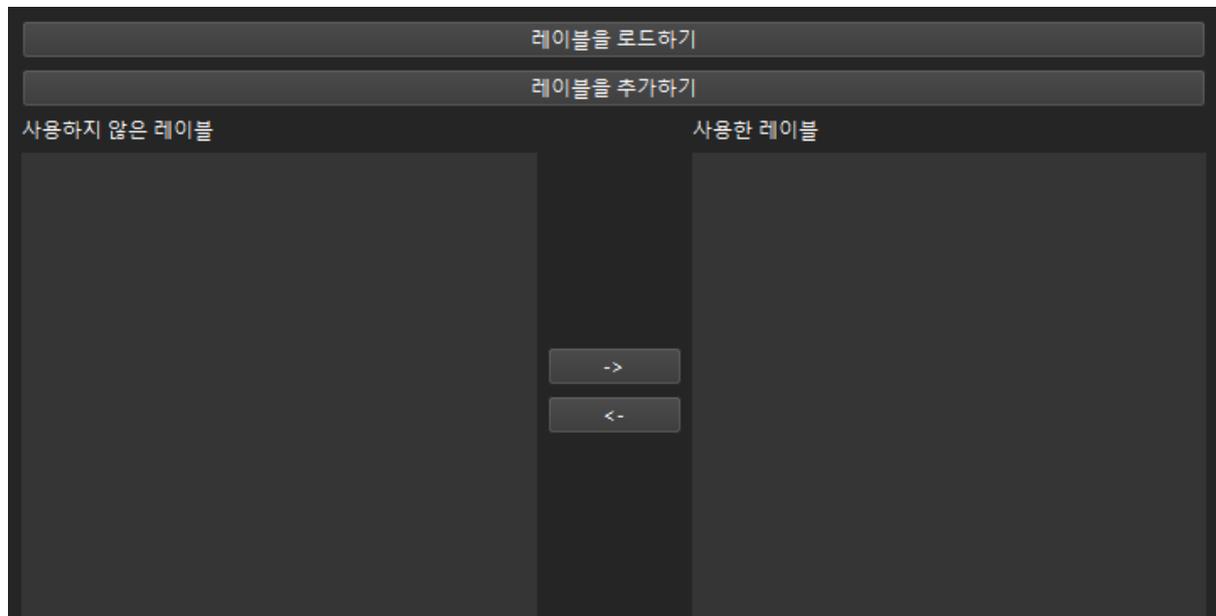
##### 이 옵션을 선택함

설명: 이 스텝은 “피킹된 물체를 업데이트하기” 스텝 후에 사용되며 현재 피킹된 작업물의 레이블 정보를 획득한 다음에 분류에 따라 서로 다른 작업을 수행합니다. 배경: 작업물이 적층되는 등 이유로 인식된 대상 물체의 레이블 정보를 직접적으로 획득할 수 없기 때문에 작업물을 피킹한 후 작업물에 대한 2차 인식 및 분류를 한 후 레이블을 제공해야 합니다.

##### 이 옵션을 선택하지 않음

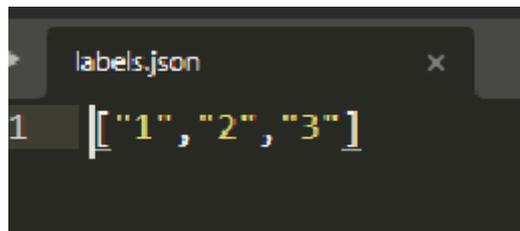
기본값. Mech-Vision에서 인식된 물체 레이블을 사용합니다. “분류”가 “비전 이동” 전에 있음 파라미터의 값을 대응하는 “비전 이동” 스텝으로 설정해야 합니다.

### 레이블을 로드하기 & 레이블을 추가하기



#### 레이블을 로드하기

**사용하지 않은 레이블** 목록에 한 번에 여러 레이블을 추가합니다. 아래 그림과 같이 **레이블을 로드하기** 버튼을 클릭하고 여러 레이블이 포함된 **.json** 파일을 선택합니다. 모든 레이블 파일은 **사용하지 않은 레이블** 목록에 로드되며 원래 레이블은 자동으로 삭제됩니다.



#### 레이블을 추가하기

**사용하지 않은 레이블** 목록에 **\*unnamed\_label\***이라는 레이블을 추가하며 이 레이블을 더블클릭하면 명칭을 수정할 수 있습니다.

## 레이블 목록

\*사용한 레이블\*과 \*사용하지 않은 레이블\*로 나뉘지며  또는  를 클릭하면 서로 전환할 수 있습니다.

**사용된 레이블\*은 이스텝의 아웃 포트에 나타납니다. \*사용한 레이블 목록에 없는 레이블을 필터링하기 위해 \*기타\*라는 레이블이 있습니다.**

### 5.4.5.2. 카운터

#### 기능 설명

현재 프로그램 또는 분기의 실행 횟수를 계산합니다. 카운트 역치를 설정하고 카운트가 역치를 충족하면 다른 분기를 실행할 수 있습니다. 이 스텝은 일반적으로 리셋 스텝과 함께 사용되어 누적된 수를 재설정합니다.

#### 응용 시나리오

실행 횟수를 계산해야 하는 시나리오에 적합합니다.

#### 파라미터 설명

##### 비이동 스텝 기본 파라미터

##### 실행을 건너뛰기

<b>없음</b>	기본값. 해당 스텝의 실행을 건너뛰지 않습니다.
<b>시뮬레이션할 때만</b>	시뮬레이션을 할 때만 해당 스텝의 실행을 건너뛵니다. 아웃 포트는 <b>실행 건너뛴 때의 아웃 포트</b> 파라미터를 통해 지정됩니다.
<b>항상</b>	실제 실행 또는 시뮬레이션을 할 때 모두 해당 스텝의 실행을 건너뛵니다. 아웃 포트는 <b>실행 건너뛴 때의 아웃 포트</b> 파라미터를 통해 지정됩니다.
<b>조절 설명</b>	<b>시뮬레이션할 때만</b> 또는 *항상*으로 설정할 때 프로젝트 실행 중에는 현재 스텝을 건너뛰고 계속 다음 스텝을 실행합니다. 예를 들어, "DI 체크" 스텝이 프로젝트에 연결되어 있고 시뮬레이션 중 외부 입력 신호가 없는 경우, "실행을 건너뛰기" 파라미터를 설정하지 않으면 해당 스텝을 실행할 때 프로젝트 실행이 중지됩니다. 이 파라미터를 설정한 후에는 이 스텝을 실행하지 않고 시뮬레이션이 완료될 때까지 계속 다음 스텝을 실행합니다.

##### 실행 건너뛴 때의 아웃 포트

**실행을 건너뛰기** 파라미터의 값은 **시뮬레이션할 때만** 또는 \*항상\*으로 설정해야 이 기능을 사용할 수 있습니다. 해당 스텝을 건너뛰고 계속 실행될 때의 아웃 포트를 지정합니다.

#### 카운트

계산할 횟수입니다. 해당 스텝의 실행 횟수가 설정된 값보다 크면 **완성** 포트가 사용됩니다.

#### 카운트 유형

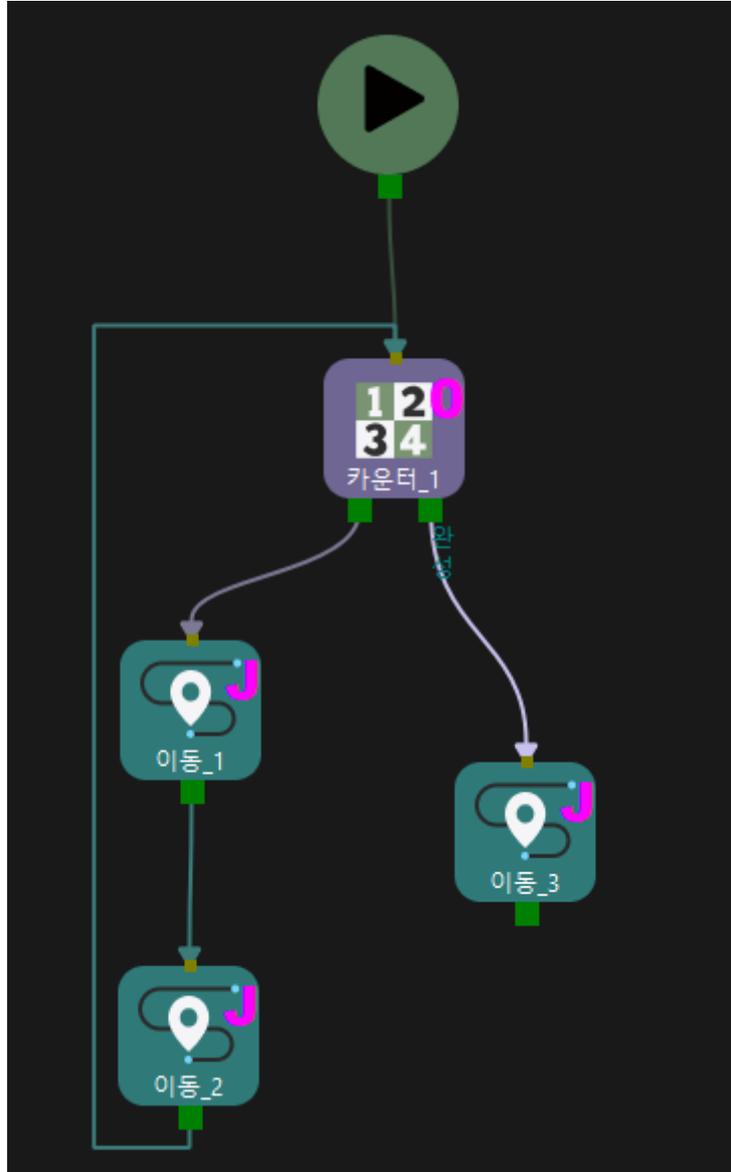
Execution: "카운터" 스텝이 실행되는 횟수를 통해 계산합니다. Picked Count: "비전 이동" 스텝이 실행되는 횟수를 통해 계산합니다.

### 현재 카운트

“카운터” 또는 “비전 이동” 스텝이 실행되는 횟수이며 \*카운트 유형\*에서 설정됩니다.

### 응용 예시

경로를 두 번 실행해야 하는 시나리오..



카운트 파라미터의 값을 2로 설정하는 경우, 해당 스텝은 3 번째로 실행될 때 **완성** 포트가 사용됩니다.

### 5.4.5.3. 완료 여부 확인

#### 기능 설명

이 스텝은 인덱스가 있는 모든 이동 유형 스텝(배열대로 이동, 순서대로 이동등)이 루프를 피하기 위해 단일 주기에서 실행되는지 여부를 확인하는 데 사용됩니다.

## 응용 시나리오

일반적으로 “배열대로 이동”, “순서대로 이동” 스텝과 함께 사용됩니다.

단일 주기 실행이 완료되면 **완성** 아웃 포트가 사용되고 그렇지 않으면 **미완성** 아웃 포트가 사용됩니다.

## 파라미터 설명

### 비이동 스텝 기본 파라미터

#### 실행을 건너뛰기

<b>없음</b>	기본값. 해당 스텝의 실행을 건너뛰지 않습니다.
<b>시뮬레이션할 때만</b>	시뮬레이션을 할 때만 해당 스텝의 실행을 건너뛵니다. 아웃 포트는 <b>실행 건너뛴 때의 아웃 포트</b> 파라미터를 통해 지정됩니다.
<b>항상</b>	실제 실행 또는 시뮬레이션을 할 때 모두 해당 스텝의 실행을 건너뛵니다. 아웃 포트는 <b>실행 건너뛴 때의 아웃 포트</b> 파라미터를 통해 지정됩니다.
<b>조절 설명</b>	<b>시뮬레이션할 때만</b> 또는 *항상*으로 설정할 때 프로젝트 실행 중에는 현재 스텝을 건너뛰고 계속 다음 스텝을 실행합니다. 예를 들어, "DI 체크" 스텝이 프로젝트에 연결되어 있고 시뮬레이션 중 외부 입력 신호가 없는 경우, “실행을 건너뛰기” 파라미터를 설정하지 않으면 해당 스텝을 실행할 때 프로젝트 실행이 중지됩니다. 이 파라미터를 설정한 후에는 이 스텝을 실행하지 않고 시뮬레이션이 완료될 때까지 계속 다음 스텝을 실행합니다.

#### 실행 건너뛴 때의 아웃 포트

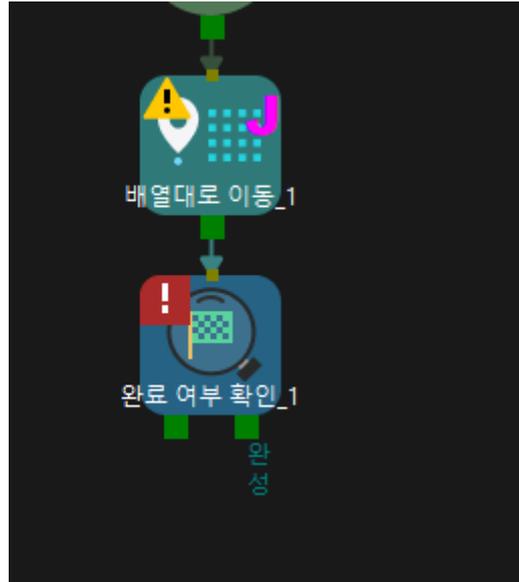
**실행을 건너뛰기** 파라미터의 값은 **시뮬레이션할 때만** 또는 \*항상\*으로 설정해야 이 기능을 사용할 수 있습니다. 해당 스텝을 건너뛰고 계속 실행될 때의 아웃 포트를 지정합니다.

#### 이동 스텝을 선택하기

실제 프로젝트에 있는 “이동” 유형 스텝을 선택합니다.

#### 응용 예시

“배열대로 이동” 스텝과 함께 사용되며 이 스텝은 루프가 끝날 때까지 실행이 성공적으로 완료되었는지 여부를 판단합니다.



#### 5.4.5.4. 인덱스 변경

##### 기능 설명

현재 인덱스 파라미터를 갖춘 “이동 유형” 스텝 또는 “카운터” 스텝과 결합하여 사용됩니다.

이 스텝은 실행될 때마다 **보폭** 파라미터 값을 통해 지정된 “이동” 유형 스텝의 **현재 인덱스** 또는 “카운터” 스텝의 **현재 카운트** 파라미터 값을 수정합니다.

##### 응용 시나리오

현재 인덱스 파라미터를 갖춘 “이동 유형” 스텝 또는 “카운터” 스텝과 결합하여 사용됩니다.

##### 파라미터 설명

##### 비이동 스텝 기본 파라미터

##### 실행을 건너뛰기

<b>없음</b>	기본값. 해당 스텝의 실행을 건너뛰지 않습니다.
<b>시뮬레이션할 때만</b>	시뮬레이션을 할 때만 해당 스텝의 실행을 건너뛩니다. 아웃 포트는 <b>실행 건너뛴 때의 아웃 포트</b> 파라미터를 통해 지정됩니다.
<b>항상</b>	실제 실행 또는 시뮬레이션을 할 때 모두 해당 스텝의 실행을 건너뛩니다. 아웃 포트는 <b>실행 건너뛴 때의 아웃 포트</b> 파라미터를 통해 지정됩니다.
<b>조절 설명</b>	<b>시뮬레이션할 때만</b> 또는 *항상*으로 설정할 때 프로젝트 실행 중에는 현재 스텝을 건너뛰고 계속 다음 스텝을 실행합니다. 예를 들어, "DI 체크" 스텝이 프로젝트에 연결되어 있고 시뮬레이션 중 외부 입력 신호가 없는 경우, “실행을 건너뛰기” 파라미터를 설정하지 않으면 해당 스텝을 실행할 때 프로젝트 실행이 중지됩니다. 이 파라미터를 설정한 후에는 이 스텝을 실행하지 않고 시뮬레이션이 완료될 때까지 계속 다음 스텝을 실행합니다.

##### 실행 건너뛴 때의 아웃 포트

**실행을 건너뛰기** 파라미터의 값은 **시뮬레이션할 때만** 또는 \*항상\*으로 설정해야 이 기능을 사용할 수 있습니다. 해당 스텝을 건너뛰고 계속 실행될 때의 아웃 포트를 지정합니다.

## 보폭

기본값: 0.



“이동” 유형 스텝의 **현재 인덱스\***와 “카운터” 수텝의 **\*현재 카운트** 파라미터는 스텝 자체의 규칙에 따라 실행될 때마다 값에 1을 더합니다.

“이동” 유형 스텝의 **현재 인덱스\***와 “카운터” 스텝의 **\*현재 카운트** 값은 **보폭** 파라미터 값에 따라 업데이트됩니다.

업데이트 후의 **현재 인덱스 / 현재 카운트** = 업데이트 전의 **현재 인덱스 / 현재 카운트** + 1

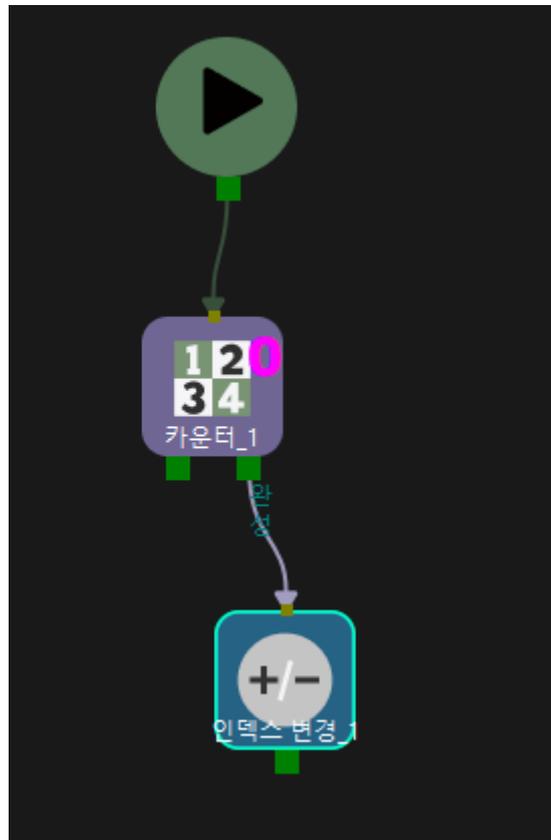
- 보폭 = 0: “이동” 유형 스텝의 **현재 인덱스\***와 “카운터” 스텝의 **\*현재 카운트** 값에 영향을 미치지 않습니다.
- 보폭 = -1: 스텝이 실행될 때마다 “이동” 유형 스텝의 **\*현재 인덱스\***와 “카운터” 스텝의 **\*현재 카운트\***는 1씩 감소합니다.
- 보폭 = 1: 스텝이 실행될 때마다 “이동” 유형 스텝의 **\*현재 인덱스\***와 “카운터” 스텝의 **\*현재 카운트\***는 1씩 증가합니다.

### 인덱스를 갖춘 이동 유형 스텝 또는 카운터 스텝을 선택하기

실제 수요에 따라 인덱스를 갖춘 “이동” 유형 스텝 또는 “카운터” 스텝을 선택합니다.

### 응용 예시

“카운터” 스텝 뒤에 “인덱스 변경” 스텝을 연결하면 수요에 따라 **현재 카운트** 파라미터의 값을 수정할 수 있습니다.



보폭은 -1이면, 한번 실행된 후 "카운터" 스텝의 **\*현재 카운트\***는 1씩 감소합니다.

#### 5.4.5.5. 리셋

##### 기능 설명

설정에 따라 선택한 스텝을 리셋합니다.

##### 응용 시나리오

이 스텝은 계산, 피킹 및 배치 판단, 비전 결과 재사용 등이 필요한 시나리오에 사용됩니다.

##### 파라미터 설명

##### 비이동 스텝 기본 파라미터

##### 실행을 건너뛰기

<b>없음</b>	기본값. 해당 스텝의 실행을 건너뛰지 않습니다.
<b>시뮬레이션할 때만</b>	시뮬레이션을 할 때만 해당 스텝의 실행을 건너뛵니다. 아웃 포트는 <b>실행 건너뛴 때의 아웃 포트</b> 파라미터를 통해 지정됩니다.
<b>항상</b>	실제 실행 또는 시뮬레이션을 할 때 모두 해당 스텝의 실행을 건너뛵니다. 아웃 포트는 <b>실행 건너뛴 때의 아웃 포트</b> 파라미터를 통해 지정됩니다.
<b>조절 설명</b>	<b>시뮬레이션할 때만</b> 또는 *항상*으로 설정할 때 프로젝트 실행 중에는 현재 스텝을 건너뛰고 계속 다음 스텝을 실행합니다. 예를 들어, "DI 체크" 스텝이 프로젝트에 연결되어 있고 시뮬레이션 중 외부 입력 신호가 없는 경우, "실행을 건너뛰기" 파라미터를 설정하지 않으면 해당 스텝을 실행할 때 프로젝트 실행이 중지됩니다. 이 파라미터를 설정한 후에는 이 스텝을 실행하지 않고 시뮬레이션이 완료될 때까지 계속 다음 스텝을 실행합니다.

##### 실행 건너뛴 때의 아웃 포트

**실행을 건너뛰기** 파라미터의 값은 **시뮬레이션할 때만** 또는 \*항상\*으로 설정해야 이 기능을 사용할 수 있습니다. 해당 스텝을 건너뛰고 계속 실행될 때의 아웃 포트를 지정합니다.

##### 일부 리셋

기본적으로 선택되어 있지 않으며 하나의 스텝만 리셋할 수 있습니다. 선택된 스텝 전에 있는 계산, 피킹 및 배치 판단, 비전 결과 재사용 등 스텝을 리셋하는 데 사용되며 \*스텝을 선택하기\*에서 설정합니다.

이 옵션을 선택하면 카운트, 피킹 및 배치 판단, 비전 결과 재사용과 관련된 여러 스텝이 있는 경우, \*스텝을 선택하기\*에서 스텝을 선택하면 "리셋 리스트"에 선택 가능한 모든 스텝들이 표시됩니다. 필요에 따라 리셋할 스텝을 선택하십시오.

##### 스텝을 선택하기

리셋할 스텝을 선택합니다.

#### 5.4.5.6. 기다리기

##### 기능 설명

이 스텝을 실행할 때 기다리는 제한 시간(단위: ms)을 설정합니다.

## 응용 시나리오

기다리는 시간\*을 설정해야 하는 시나리오에 적용되며 일반적으로 “DO설정” 또는 **비전 결과 체크** 스텝의 \*결과 없음 아웃 포트 뒤에 연결합니다.

## 파라미터 설명

### 비이동 스텝 기본 파라미터

#### 실행을 건너뛰기

<b>없음</b>	기본값. 해당 스텝의 실행을 건너뛰지 않습니다.
<b>시뮬레이션할 때만</b>	시뮬레이션을 할 때만 해당 스텝의 실행을 건너뛵니다. 아웃 포트는 <b>실행 건너뛴 때의 아웃 포트</b> 파라미터를 통해 지정됩니다.
<b>항상</b>	실제 실행 또는 시뮬레이션을 할 때 모두 해당 스텝의 실행을 건너뛵니다. 아웃 포트는 <b>실행 건너뛴 때의 아웃 포트</b> 파라미터를 통해 지정됩니다.
<b>조절 설명</b>	<b>시뮬레이션할 때만</b> 또는 *항상*으로 설정할 때 프로젝트 실행 중에는 현재 스텝을 건너뛰고 계속 다음 스텝을 실행합니다. 예를 들어, "DI 체크" 스텝이 프로젝트에 연결되어 있고 시뮬레이션 중 외부 입력 신호가 없는 경우, “실행을 건너뛰기” 파라미터를 설정하지 않으면 해당 스텝을 실행할 때 프로젝트 실행이 중지됩니다. 이 파라미터를 설정한 후에는 이 스텝을 실행하지 않고 시뮬레이션이 완료될 때까지 계속 다음 스텝을 실행합니다.

#### 실행 건너뛴 때의 아웃 포트

실행을 건너뛰기 파라미터의 값은 **시뮬레이션할 때만** 또는 \*항상\*으로 설정해야 이 기능을 사용할 수 있습니다. 해당 스텝을 건너뛰고 계속 실행될 때의 아웃 포트를 지정합니다.

#### 기다리는 시간

실제 수요에 따라 기다리는 시간을 입력합니다. 기본값: 1000ms(1s).

## 5.4.6. 경로

### 5.4.6.1. 경로 프로시저

#### 기능 설명

이 스텝은 웨이포인트의 템플릿 파일에 따라 여러 개의 연속 세그먼트로 배열된 조밀한 포인트를 따라 이동하도록 로봇을 가이드합니다. 일반적으로 **비전 이동** 스텝과 함께 사용됩니다.

#### 파라미터 설명

없음.

#### 웨이포인트의 Json 파일 경로

미리 작성한 웨이포인트 Json 파일. 프로그램이 실행될 때 비전 서비스에서 제공한 포인트의 수는 템플릿 파일 속의 포인트 수와 일치해야 합니다.

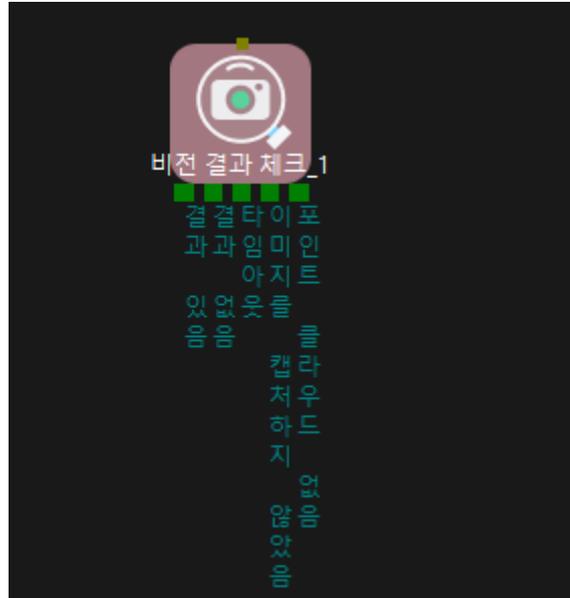
## 5.4.7. 비전

### 5.4.7.1. 비전 결과 체크

#### 기능 설명

비전 결과를 체크하고 다른 결과에 대해 서로 다른 아웃 포트를 지정합니다.

아웃 포트가 총 5개 있습니다.



- 결과 있음: “비전 인식” 스텝에 출력된 결과가 있으면 프로젝트는 이 포트를 통해 실행됩니다.
- 결과 없음: 오류로 인해 “비전 인식” 스텝에 출력된 결과가 없으면 프로젝트는 이 포트를 통해 실행됩니다.
- 이미지를 캡처하지 않았음: 여러 개 “비전 인식” 스텝을 사용한 프로젝트에 적용됩니다. 만약 이전 작업 흐름에서 현재 “비전 결과 체크” 스텝에 시야 서비스를 제공하는 “비전 인식” 스텝이 없으면, 프로젝트는 이 아웃 포트를 사용합니다. **비전 결과 체크** 스텝은 바로 “비전 인식” 스텝 뒤에 사용되는 경우, 이 아웃 포트를 연결하지 않아도 됩니다.
- 타이아웃: 설정된 **기다리는 데 제한 시간** 이후에도 비전 처리 프로세스가 종료되지 않으면 프로젝트는 이 아웃 포트를 사용합니다. 해당 아웃 포트는 **비전 결과 체크** 스텝의 입구와 연결하여 스텝을 다시 실행할 수 있고 **기다리기** 스텝과 연결되어 다시 인포트로 돌아갈 수도 있습니다.
- 포인트 클라우드 없음: ROI 내 포인트 클라우드가 없는 경우, 즉 인식된 물체가 없는 경우, 프로젝트는 이 아웃 포트를 사용합니다.
  - 포인트 클라우드가 없는 경우 로봇은 원점 위치로 돌아가고 비전 결과를 다시 체크하며 물체 유무를 다시 판단합니다. 물체가 없으면 우선 로봇을 정지하고 모든 대상 물체가 이미 피킹되었는지 확인해야 합니다.

#### 응용 시나리오

이는 "비전 인식"에서 얻은 비전 결과에 따라 프로젝트를 지정된 분기로 진행시킵니다.

#### 파라미터 설명

##### 비이동 스텝 기본 파라미터

##### 실행을 건너뛰기

<b>없음</b>	기본값. 해당 스텝의 실행을 건너뛰지 않습니다.
<b>시뮬레이션할 때만</b>	시뮬레이션을 할 때만 해당 스텝의 실행을 건너뛵니다. 아웃 포트는 <b>실행 건너뛵 때의 아웃 포트</b> 파라미터를 통해 지정됩니다.
<b>항상</b>	실제 실행 또는 시뮬레이션을 할 때 모두 해당 스텝의 실행을 건너뛵니다. 아웃 포트는 <b>실행 건너뛵 때의 아웃 포트</b> 파라미터를 통해 지정됩니다.
<b>조절 설명</b>	<b>시뮬레이션할 때만</b> 또는 *항상*으로 설정할 때 프로젝트 실행 중에는 현재 스텝을 건너뛰고 계속 다음 스텝을 실행합니다. 예를 들어, "DI 체크" 스텝이 프로젝트에 연결되어 있고 시뮬레이션 중 외부 입력 신호가 없는 경우, "실행을 건너뛰기" 파라미터를 설정하지 않으면 해당 스텝을 실행할 때 프로젝트 실행이 중지됩니다. 이 파라미터를 설정한 후에는 이 스텝을 실행하지 않고 시뮬레이션이 완료될 때까지 계속 다음 스텝을 실행합니다.

#### 실행 건너뛵 때의 아웃 포트

**실행을 건너뛰기** 파라미터의 값은 **시뮬레이션할 때만** 또는 \*항상\*으로 설정해야 이 기능을 사용할 수 있습니다. 해당 스텝을 건너뛰고 계속 실행될 때의 아웃 포트를 지정합니다.

#### 자체 정의 키를 체크하기

기본적으로 선택되지 않으며 자체 정의 키를 체크하지 않습니다. 이 파라미터를 선택하면 자체 정의 키를 체크합니다. \*체크할 키\*를 설정할 수 있으며 Adapter와 통신합니다.

#### 사전 계획 아웃 포트

계획의 연속성을 보장하기 위해서는 사전 계획의 아웃 포트를 지정해야 합니다. 일반적으로 다시 계획의 확률을 줄이고 로봇의 대기 시간을 줄이기 위해 실행 가능성이 가장 큰 분기를 지정합니다.

- 1 사전 계획 아웃 포트를 지정하지 않습니다.
- 0 프로젝트는 **결과 있음** 포트를 통해 실행되며 경로를 계속 계획합니다.

#### 기다리는 데 제한 시간

비전 서비스의 처리를 기다리는 데 제한 시간(단위: s)을 지정합니다. 기본값: 5s. 이 파라미터는 **타임아웃** 포트와 결합하여 사용됩니다. 구체적인 수요에 따라 설정해 주십시오.

#### 비전 서비스 명칭

비전 서비스를 제공하는 Mech-Vision 프로젝트 명칭을 지정합니다. 명칭은 이전 "비전 인식" 스텝에 비전 서비스를 제공하는 Mech-Vision 프로젝트와 동일해야 합니다.

### 5.4.7.2. 파렛트 포즈 업데이트

#### 기능 설명

비전 서비스를 통해 파렛트 포즈를 업데이트합니다.

#### 응용 시나리오

팔레타이징 작업이 완료되면 새 파렛트가 배치되고 파렛트 포즈를 업데이트해야 합니다. 비전 서비스를 통해 지정된 팔레타이징 스텝에서 파렛트 포즈를 업데이트합니다.

## 파라미터 설명

### 비이동 스텝 기본 파라미터

#### 실행을 건너뛰기

<b>없음</b>	기본값. 해당 스텝의 실행을 건너뛰지 않습니다.
<b>시뮬레이션할 때만</b>	시뮬레이션을 할 때만 해당 스텝의 실행을 건너뛵니다. 아웃 포트는 <b>실행 건너뛴 때의 아웃 포트</b> 파라미터를 통해 지정됩니다.
<b>항상</b>	실제 실행 또는 시뮬레이션을 할 때 모두 해당 스텝의 실행을 건너뛵니다. 아웃 포트는 <b>실행 건너뛴 때의 아웃 포트</b> 파라미터를 통해 지정됩니다.
<b>조절 설명</b>	<b>시뮬레이션할 때만</b> 또는 *항상*으로 설정할 때 프로젝트 실행 중에는 현재 스텝을 건너뛰고 계속 다음 스텝을 실행합니다. 예를 들어, "DI 체크" 스텝이 프로젝트에 연결되어 있고 시뮬레이션 중 외부 입력 신호가 없는 경우, "실행을 건너뛰기" 파라미터를 설정하지 않으면 해당 스텝을 실행할 때 프로젝트 실행이 중지됩니다. 이 파라미터를 설정한 후에는 이 스텝을 실행하지 않고 시뮬레이션이 완료될 때까지 계속 다음 스텝을 실행합니다.

#### 실행 건너뛴 때의 아웃 포트

**실행을 건너뛰기** 파라미터의 값은 **시뮬레이션할 때만** 또는 \*항상\*으로 설정해야 이 기능을 사용할 수 있습니다. 해당 스텝을 건너뛰고 계속 실행될 때의 아웃 포트를 지정합니다.

#### 파렛트 포즈를 저장하기

기본적으로 선택되어 있지 않습니다. 즉 업데이트된 파렛트 포즈를 저장하지 않습니다. 이 파라미터를 선택하면 업데이트된 파렛트 포즈를 저장합니다. **파렛트 포즈 파일의 저장 경로** 뒤에 있는  아이콘을 클릭하면 파렛트 포즈의 저장 경로를 선택할 수 있습니다.

#### 파렛트 포즈를 로드하기

기본적으로 선택되어 있지 않습니다. 즉 업데이트된 파렛트 포즈를 로드하지 않습니다. 이 파라미터를 선택하면 업데이트된 파렛트 포즈를 로드합니다. **파렛트 포즈를 로드하는 경로** 뒤에 있는  아이콘을 클릭하면 파렛트 포즈를 로드할 수 있습니다.

#### 팔레타이징 스텝을 선택하기

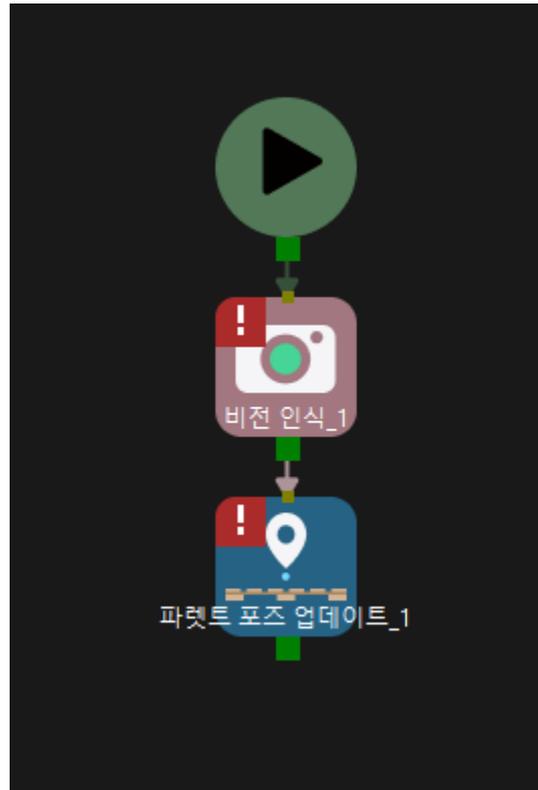
파렛트 포즈를 업데이트하려는 팔레타이징 스텝을 선택합니다.

#### 비전 서비스 명칭

등록된 비전 서비스의 명칭. 해당 비전 프로젝트가 신규 파렛트의 포즈를 제공합니다.

#### 응용 예시

로봇이 팔레타이징 작업을 완료한 후 신규 파렛트는 다른 위치에 배치되므로 이때 파렛트 포즈를 업데이트해야 합니다. **비전 인식** 스텝과 결합하여 사용해야 합니다.



### 5.4.7.3. 피킹된 물체를 업데이트하기

#### 기능 설명

비전 서비스를 통해 이미 피킹된 작업물의 포즈나 치수를 업데이트하여 충돌을 방지합니다.

 | 작업물 포즈는 로봇이 물체를 잡고 내려놓았을 때의 포즈입니다.

#### 응용 시나리오

획득한 물체의 포즈나 치수가 부정확할 때 사용됩니다.

충돌을 피하기 위해 정확한 작업물의 포즈와 치수를 다시 획득하고 충돌 모델을 다시 선택하며 충돌 감지를 수행해야 합니다.

- 작업물은 직육면체, 수직 원기둥 또는 수평 원기둥일 수 있습니다.
- 이 스텝을 사용하면 이동 유형 스텝의 계획을 중단할 수 있습니다.

#### 파라미터 설명

##### 비이동 스텝 기본 파라미터

##### 실행을 건너뛰기

<b>없음</b>	기본값. 해당 스텝의 실행을 건너뛰지 않습니다.
<b>시뮬레이션할 때만</b>	시뮬레이션을 할 때만 해당 스텝의 실행을 건너뛵니다. 아웃 포트는 <b>실행 건너뛰기</b> 때의 아웃 포트 파라미터를 통해 지정됩니다.

**항상** 실제 실행 또는 시뮬레이션을 할 때 모두 해당 스텝의 실행을 건너뛴니다. 아웃 포트는 **실행 건너뛴 때의 아웃 포트** 파라미터를 통해 저장됩니다.

**조절 설명** **시뮬레이션할 때만** 또는 \*항상\*으로 설정할 때 프로젝트 실행 중에는 현재 스텝을 건너뛰고 계속 다음 스텝을 실행합니다. 예를 들어, "DI 체크" 스텝이 프로젝트에 연결되어 있고 시뮬레이션 중 외부 입력 신호가 없는 경우, "실행을 건너뛰기" 파라미터를 설정하지 않으면 해당 스텝을 실행할 때 프로젝트 실행이 중지됩니다. 이 파라미터를 설정한 후에는 이 스텝을 실행하지 않고 시뮬레이션이 완료될 때까지 계속 다음 스텝을 실행합니다.

#### 실행 건너뛴 때의 아웃 포트

**실행을 건너뛰기** 파라미터의 값은 **시뮬레이션할 때만** 또는 \*항상\*으로 설정해야 이 기능을 사용할 수 있습니다. 해당 스텝을 건너뛰고 계속 실행될 때의 아웃 포트를 지정합니다.

#### 업데이트 방식

**파라미터를 통해** 및 **비전 서비스를 통해** 두 가지 선택할 수 있습니다.

#### 파라미터를 통해

작업물의 치수 정보가 없거나 로봇이 비전 서비스에 의해 가이드되지 않는 경우 이 방식을 사용하십시오. **피킹된 물체를 업데이트하기** 스텝을 통해 로봇 말단에 생성되며 충돌 감지에 사용될 수 있는 작업물 모델을 설정합니다.

**물체 유형** 작업물은 직육면체, 수직 원기둥, 수평 원기둥 또는 브레이크 디스크.

**물체 포즈** TCP 좌표계의 포즈. 기본적으로 선택하지 않습니다. 즉 기본적으로 피킹된 물체의 포즈를 업데이트하지 않습니다. 이 파라미터를 선택하면 피킹된 물체의 포즈를 업데이트합니다. 이 때 **포즈 업데이트** 영역에서 포즈 업데이트와 관련된 내용을 설정하십시오.

**물체 치수** 피킹된 작업물의 치수입니다. 기본적으로 선택하지 않습니다. 즉 기본적으로 피킹된 물체의 치수를 업데이트하지 않습니다. 이 파라미터를 선택하면 피킹된 물체의 치수를 업데이트합니다. 이 때 **치수 업데이트** 영역에서 치수 업데이트와 관련된 내용을 설정하십시오.

**브레이크 디스크 구성 파일 경로** 를 클릭하여 **json** 포맷의 브레이크 디스크 구성 파일을 선택합니다. 브레이크 디스크인 경우, 포즈만 업데이트할 수 있습니다.

#### 비전 서비스를 통해

로봇은 물체를 적절한 위치로 피킹한 후 재차 이미지를 캡처하여 피킹된 물체의 치수와 TCP 좌표계의 포즈를 다시 추정합니다.

#### 업데이트 내용

**물체 포즈** 기본적으로 선택되어 있습니다. 즉 TCP 좌표를 프로젝트로 자동으로 업데이트합니다. 선택하지 않으면 피킹된 작업물의 포즈를 업데이트하지 않습니다.

**물체 치수** 기본적으로 선택되어 있습니다. 즉 피킹된 물체의 치수 정보를 프로젝트로 자동으로 업데이트합니다. 선택하지 않으면 피킹된 작업물의 치수를 업데이트하지 않습니다.

## 이동의 사전 계획

**사전 계획 아웃 포트** 계획의 연속성을 보장하기 위해서는 사전 계획의 아웃 포트를 지정해야 합니다. 일반적으로 다시 계획의 확률을 줄이고 로봇의 대기 시간을 줄이기 위해 실행 가능성이 가장 큰 분기를 지정합니다.

- 0: **업데이트할 필요 없음** 아웃 포트를 사용하며 재차 촬영 결과의 편차가 설정된 \*업데이트 역치\*보다 작은 경우에 사용됩니다.
- 1: **업데이트** 아웃 포트를 사용하며 재차 촬영 결과의 편차가 설정된 \*업데이트 역치\*보다 큰 경우에 사용됩니다.
- 2: **비정상** 아웃 포트를 사용하며 비전 처리가 비정상적인 경우에 사용됩니다.

## 업데이트 역치

**정확성 역치로 필터링해야 함** 기본적으로 선택되어 있습니다. 즉 **치수 역치**, **평행이동 역치** 및 **회전 역치** 파라미터를 통해 정확성 역치를 설정합니다. 이 파라미터를 선택하지 않으면 정확성 역치를 설정하지 않습니다.

## 응용 예시

\*분류\*와 “혼합 팔레타이징” 스텝의 특정 기능과 결합하여 사용될 수 있습니다.

분류 스텝과 결합하여 사용될 경우, 분류 스텝 중의 “**피킹된 물체를 업데이트하기**”에서 레이블을 획득하기 파라미터를  상태로 선택해야 합니다.

혼합 팔레타이징 스텝과 결합하여 사용될 경우, “혼합 팔레타이징” 스텝의 파라미터 패널에서 **파렛트 유형** > **낙하 방식으로 배치하기**를 선택하여 \*상자 치수를 업데이트하기 위해 이미지를 재차 캡처하기\*를  상태로 선택합니다. 고정적인 사용 방식은 다음 그림과 같습니다.



#### 5.4.7.4. 시나리오 물체 업데이트

##### 기능 설명

비전 서비스 또는 파라미터 설정에 따라 시나리오 물체의 포즈와 치수를 설정합니다.

##### 응용 시나리오

Mech-Viz 실행 과정에서 시나리오 모델의 위치, 치수를 조정하여 다양한 충돌 문제를 해결하는 데 사용됩니다.

## 파라미터 설명

### 비이동 스텝 기본 파라미터

#### 실행을 건너뛰기

<b>없음</b>	기본값. 해당 스텝의 실행을 건너뛰지 않습니다.
<b>시뮬레이션할 때만</b>	시뮬레이션을 할 때만 해당 스텝의 실행을 건너뛵니다. 아웃 포트는 <b>실행 건너뛴 때의 아웃 포트</b> 파라미터를 통해 지정됩니다.
<b>항상</b>	실제 실행 또는 시뮬레이션을 할 때 모두 해당 스텝의 실행을 건너뛵니다. 아웃 포트는 <b>실행 건너뛴 때의 아웃 포트</b> 파라미터를 통해 지정됩니다.
<b>조절 설명</b>	<b>시뮬레이션할 때만</b> 또는 *항상*으로 설정할 때 프로젝트 실행 중에는 현재 스텝을 건너뛰고 계속 다음 스텝을 실행합니다. 예를 들어, "DI 체크" 스텝이 프로젝트에 연결되어 있고 시뮬레이션 중 외부 입력 신호가 없는 경우, "실행을 건너뛰기" 파라미터를 설정하지 않으면 해당 스텝을 실행할 때 프로젝트 실행이 중지됩니다. 이 파라미터를 설정한 후에는 이 스텝을 실행하지 않고 시뮬레이션이 완료될 때까지 계속 다음 스텝을 실행합니다.

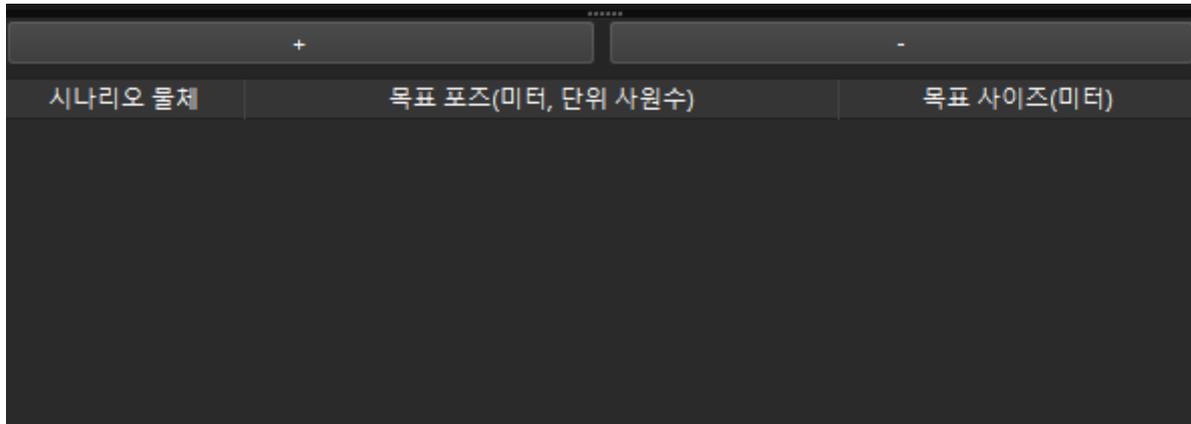
#### 실행 건너뛴 때의 아웃 포트

**실행을 건너뛰기** 파라미터의 값은 **시뮬레이션할 때만** 또는 \*항상\*으로 설정해야 이 기능을 사용할 수 있습니다. 해당 스텝을 건너뛰고 계속 실행될 때의 아웃 포트를 지정합니다.

#### 정보 소스 설정

시나리오 물체의 포즈와 치수 정보를 업데이트하는 데 사용되는 정보 소스를 지정합니다. **파라미터를 통해** 및 **비전 서비스를 통해** 두 가지 선택할 수 있습니다.

#### 파라미터를 통해



[+]: 시나리오 물체를 추가합니다.

[-]: 선택한 시나리오 물체를 삭제합니다.

실제 요구 사항에 따라 시나리오 물체를 추가하거나 삭제할 수 있습니다. 시나리오 물체 목록의 순서에 따라 시나리오 물체의 포즈와 치수가 업데이트됩니다. 시나리오 물체 목록의 각 항목에 대해 해당 드롭다운 목록에서 시나리오 물체를 선택한 다음 **목표 포즈** 및 \*목표 치수\*를 설정할 수 있습니다. 목표 포즈의 포맷은 XYZ(단위: m) 및 사원수 총 7개 숫자이며 목표 치수는 X, Y, Z 방향의 치수(단위: m) 데이터입니다.

예를 들어, 목표 포즈: **1.00, 0.00, 0.00, 1.00, 0.00, 0.00, 0.00**. 직육면체의 목표 치수: **0.40,**

0.40, 1.00(길이, 너비, 높이). 원기둥의 목표 치수: 0.50, 1.00, 0.00(반경, 높이, 0).

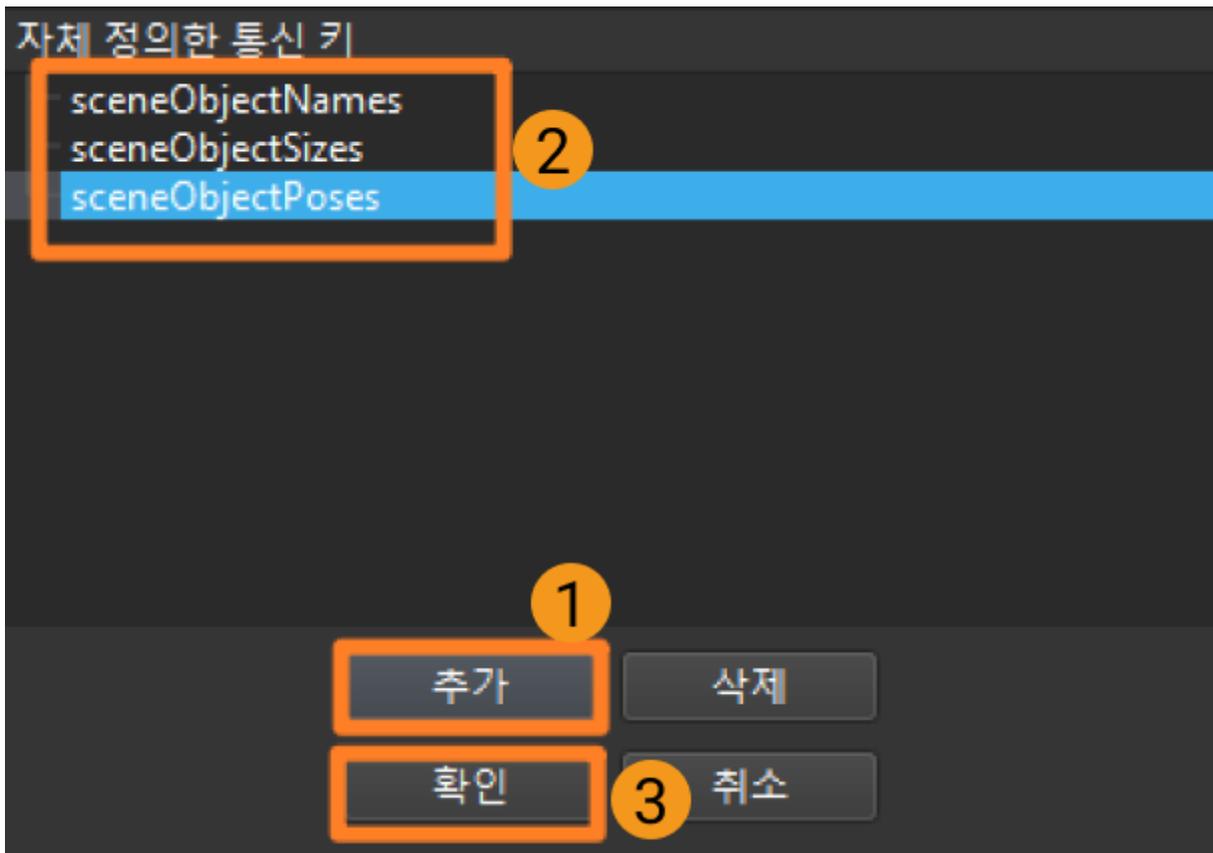
비전 서비스를 통해

비전 서비스 명칭

비전 서비스를 제공하는 Mech-Vision 프로젝트 명칭을 지정합니다. 이미 등록된 Mech-Vision 프로젝트를 선택해야 합니다.

Mech-Vision 프로젝트를 선택한 후 다음과 같이 Mech-Vision 프로젝트에서 “출력” 스텝을 설정하십시오.

1. 스텝 파라미터 패널에서 \*포트 유형\*을 \*Dynamic\*로 설정합니다.
2. [ 외부 통신 출력 데이터 유형 설정 ]을 클릭하여 “출력 설정 도우미”를 엽니다.
3. [ 추가 ] 버튼을 클릭하여 3개의 사용자 정의 키를 생성하십시오.
4. 사용자 정의 키를 순서대로 더블클릭하여 각각 `sceneObjectNames`, `sceneObjectSizes` 및 `sceneObjectPoses`로 이름을 수정하십시오.
5. [ 확인 ] 버튼을 클릭하여 설정 사항을 저장하십시오.
6. 이 스텝의 입력 포트와 이전 스텝의 출력 포트를 연결하십시오.

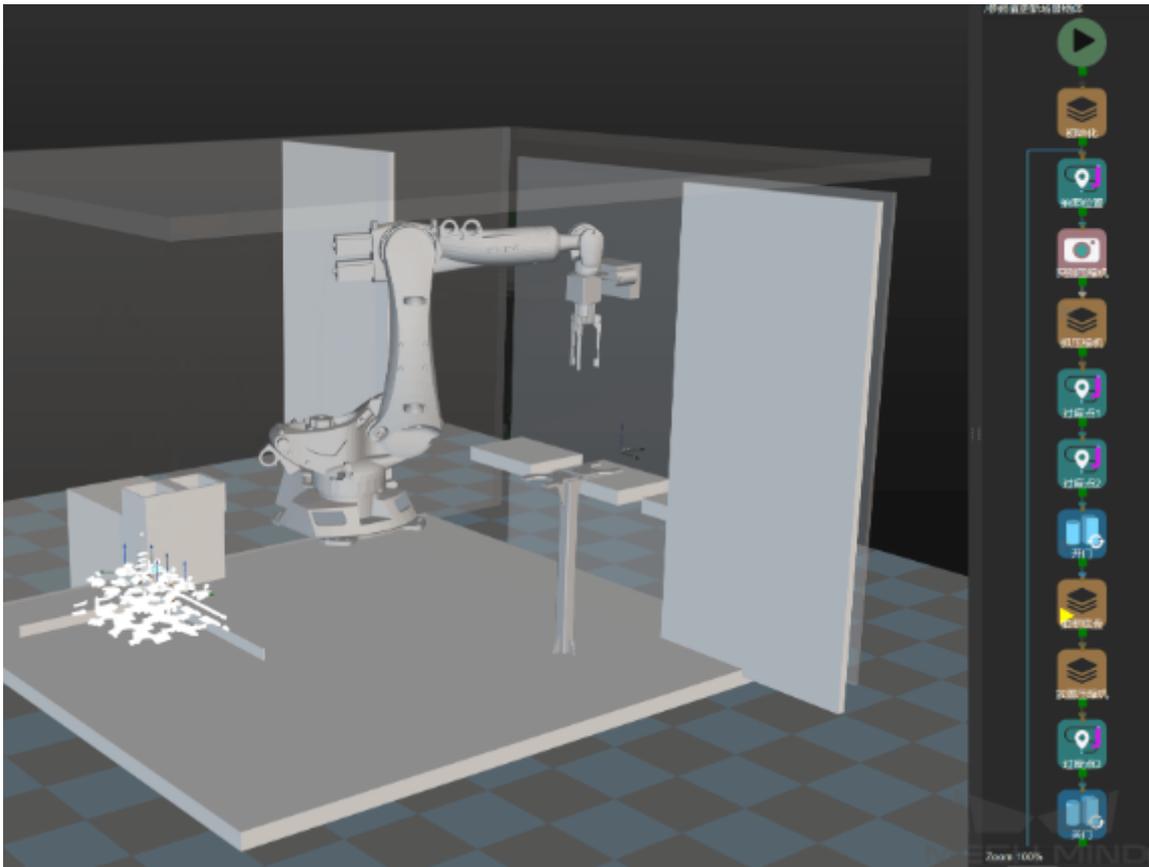


비전 서비스에서 반환된 결과	설명
sceneObjectNames	Mech-Viz 중의 업데이트될 시나리오 물체의 이름과 대응해야 합니다.

비전 서비스에서 반환된 결과	설명
sceneObjectSizes	직육면체의 경우 (길이, 너비, 높이), 원기둥의 경우 (반경, 높이, 0) 단위의 물체 치수를 포함합니다.
sceneObjectPoses	XYZ(단위: m)와 사원수를 포함한 7개의 숫자를 포함합니다.

### 응용 예시

이 예시에서는 다양한 작업 프로세스에서 충돌을 방지하기 위해 센서 도어 열기 및 닫기 프로세스를 시뮬레이션합니다. 파라미터를 설정함으로써 시나리오 물체를 업데이트합니다.



#### 5.4.7.5. 비전 결과를 모두 사용했음

##### 기능 설명

선택한 “비전 이동” 스텝 중의 비전 결과가 모두 사용됐는지를 체크하는 데만 사용됩니다.



이 시스템을 사용하려면 우선 “비전 이동” 스텝의 파라미터 \*비전 결과를 재사용하기\*를 선택해야 합니다.

##### 응용 시나리오

“비전 이동”의 비전 결과에 여러 사용 가능한 포즈가 있는 경우, 소프트웨어는 로봇을 한 번에 하나씩 피킹해서 현재 비전 결과에 있는 사용 가능한 포즈를 모두 사용할 때까지 진행합니다.

비전 결과 중의 비전 포즈가 모두 사용되면 이미지를 한번 캡처합니다. 사이클 타임에 대한 요구가 높은

프로젝트에 비전 결과를 재사용하면 비전 처리 시간을 단축할 수 있습니다.

## 파라미터 설명

### 비이동 스텝 기본 파라미터

#### 실행을 건너뛰기

<b>없음</b>	기본값. 해당 스텝의 실행을 건너뛰지 않습니다.
<b>시뮬레이션할 때만</b>	시뮬레이션을 할 때만 해당 스텝의 실행을 건너뛵니다. 아웃 포트는 <b>실행 건너뛴 때의 아웃 포트</b> 파라미터를 통해 저장됩니다.
<b>항상</b>	실제 실행 또는 시뮬레이션을 할 때 모두 해당 스텝의 실행을 건너뛵니다. 아웃 포트는 <b>실행 건너뛴 때의 아웃 포트</b> 파라미터를 통해 저장됩니다.
<b>조절 설명</b>	<b>시뮬레이션할 때만</b> 또는 *항상*으로 설정할 때 프로젝트 실행 중에는 현재 스텝을 건너뛰고 계속 다음 스텝을 실행합니다. 예를 들어, "DI 체크" 스텝이 프로젝트에 연결되어 있고 시뮬레이션 중 외부 입력 신호가 없는 경우, "실행을 건너뛰기" 파라미터를 설정하지 않으면 해당 스텝을 실행할 때 프로젝트 실행이 중지됩니다. 이 파라미터를 설정한 후에는 이 스텝을 실행하지 않고 시뮬레이션이 완료될 때까지 계속 다음 스텝을 실행합니다.

#### 실행 건너뛴 때의 아웃 포트

**실행을 건너뛰기** 파라미터의 값은 **시뮬레이션할 때만** 또는 \*항상\*으로 설정해야 이 기능을 사용할 수 있습니다. 해당 스텝을 건너뛰고 계속 실행될 때의 아웃 포트를 지정합니다.

#### 사전 계획 아웃 포트

계획의 연속성을 보장하기 위해서는 사전 계획의 아웃 포트를 지정해야 합니다. 일반적으로 다시 계획의 확률을 줄이고 로봇의 대기 시간을 줄이기 위해 실행 가능성이 가장 큰 분기를 지정합니다.

- 1 사전 계획 아웃 포트를 지정하지 않습니다.
- 0 프로젝트는 **다 사용하지 않음** 포트를 통해 실행됩니다.
- 1 프로젝트는 **모두 사용했음** 포트를 통해 실행됩니다.

#### 비전 이동을 선택하기

체크할 "비전 이동" 스텝을 선택합니다.

#### 응용 예시

디팔레타이징 시나리오에서 보통 이미지를 한번 캡처하고 상자 한 층을 피킹합니다. **비전 결과를 모두 사용했음** 스텝을 통해 \*비전 이동\*의 결과를 모두 써 버린지 체크합니다. 효과는 다음과 같습니다.



- \*다 사용하지 않음\*인 경우: 피킹을 반복하여 “이동\_2 → 상대적인 이동\_1 → 비전 이동\_1 → 상대적인 이동\_3 → 배열대로 이동\_1 → 비전 결과를 모두 사용했음”.
- 모두 사용했음: 로봇은 모두 사용했음 아웃 포트를 사용합니다.

#### 5.4.7.6. 비전 인식

##### 기능 설명

Mech-Vision 프로젝트를 시작하고 비전 결과를 Mech-Viz로 전송합니다.

##### 응용 시나리오

모든 프로젝트에 적용되며 일반적으로 “비전 결과 체크”, “상대적인 이동” 및 “비전 이동” 스텝과 함께 사용됩니다.

##### 파라미터 설명

##### 비이동 스텝 기본 파라미터

##### 실행을 건너뛰기

<b>없음</b>	기본값. 해당 스텝의 실행을 건너뛰지 않습니다.
<b>시뮬레이션할 때만</b>	시뮬레이션을 할 때만 해당 스텝의 실행을 건너뛵니다. 아웃 포트는 <b>실행 건너뛴 때의 아웃 포트</b> 파라미터를 통해 지정됩니다.
<b>항상</b>	실제 실행 또는 시뮬레이션을 할 때 모두 해당 스텝의 실행을 건너뛵니다. 아웃 포트는 <b>실행 건너뛴 때의 아웃 포트</b> 파라미터를 통해 지정됩니다.
<b>조절 설명</b>	<b>시뮬레이션할 때만</b> 또는 *항상*으로 설정할 때 프로젝트 실행 중에는 현재 스텝을 건너뛰고 계속 다음 스텝을 실행합니다. 예를 들어, "DI 체크" 스텝이 프로젝트에 연결되어 있고 시뮬레이션 중 외부 입력 신호가 없는 경우, “실행을 건너뛰기” 파라미터를 설정하지 않으면 해당 스텝을 실행할 때 프로젝트 실행이 중지됩니다. 이 파라미터를 설정한 후에는 이 스텝을 실행하지 않고 시뮬레이션이 완료될 때까지 계속 다음 스텝을 실행합니다.

##### 실행 건너뛴 때의 아웃 포트

**실행을 건너뛰기** 파라미터의 값은 **시뮬레이션할 때만** 또는 \*항상\*으로 설정해야 이 기능을 사용할 수 있습니다. 해당 스텝을 건너뛰고 계속 실행될 때의 아웃 포트를 지정합니다.

##### 이동의 정확한 도달을 기다리기

기본적으로 선택되어 있지 않습니다. 이 스텝 전에 로봇이 지정한 위치에 정확히 도달하지 않아도 해당 스텝을 실행할 수 있습니다.

##### 비전 결과 미사용 허용

기본적으로 선택되어 있지 않습니다. 즉 비전 결과를 사용해야 합니다. 이 파라미터를 선택하면 인식된 비전 결과를 사용하지 않습니다.

##### 비전 서비스 명칭

Mech-Vision 프로젝트를 선택합니다.

## 5.4.8. 기타

### 5.4.8.1. 작업 흐름 종료

#### 기능 설명

하나의 프로시저가 다른 \*프로시저\*를 중첩할 수 있는 복잡한 사용 시나리오에서 프로시저의 특정 스텝에서 결과를 출력하지 않을 때 이 스텝을 사용하여 전체 작업 흐름을 종료할 수 있습니다.

#### 파라미터 설명

없음.

#### 종료 원인

여기에서 작업 흐름을 종료해야 하는 이유를 설명하면 프로젝트 논리를 명확하게 기록할 수 있습니다.

## 5.4.9. 일반적인 파라미터

### 5.4.9.1. 이동 스텝 기본 파라미터

#### 웨이포인트를 전송하기

기본적으로 선택되어 있으며 로봇 등 수신자에 웨이포인트를 전송합니다. 선택하지 않으며 웨이포인트를 전송하지 않지만 해당 웨이포인트는 여전히 경로 계획 중에 있습니다.

#### 후속 비이동 명령을 원활성있게 수행하기

기본적으로 선택되어 있지 않습니다. **이동 스텝** 사이에 **비이동 스텝**(예: “비전 인식”, “DI 체크” 등)이 연결되면 로봇의 경로 계획을 방해하고 실제 로봇이 실행 중일 때 짧은 일시 중지가 발생하여 로봇이 원활하지 않게 실행됩니다. 이 옵션을 선택하면 현재 이동 스텝의 실행 종료를 기다릴 필요가 없고 계속해서 작업 흐름에 따라 실행할 수 있습니다. 이렇게 하면 로봇 실행 과정에서 자주 멈추는 문제를 피할 수 있고 로봇 동작의 원활성을 높일 수 있지만 스텝이 너무 일찍 종료될 수 있습니다.

스텝은 일찍 종료되는 이유가 무엇입니까?



Mech-Viz 소프트웨어가 실행 중일 때 동시에 로봇에 여러 포즈를 보냅니다. 소프트웨어는 로봇에 보낸 마지막 포즈가 로봇이 반환한 관절 각도와 동일한지 여부만 판단하고 동일하면 로봇이 마지막 위치에 이미 도달했다는 것으로 간주합니다. 예를 들어 경로에는 10개의 이동 스텝으로 구성되고 이동 스텝 5의 포즈는 마지막 이동 스텝의 포즈와 동일합니다. 로봇이 느린 속도로 이동할 때 웨이포인트 5로 이동하고 현재 관절 각도를 Mech-Viz로 보냅니다. 이동 스텝 5의 포즈는 마지막 이동 스텝의 포즈와 동일하기 때문에 Mech-Viz 소프트웨어는 경로 중의 모든 스텝 실행이 이미 완료된 것으로 잘못 판단하여 일찍 명령을 종료합니다.

#### 배치된 물체와의 충돌을 감지하지 않기

기본적으로 선택되어 있지 않습니다. 즉 이미 배치된 물체와 사이의 충돌을 감지합니다. 이 옵션을 선택하면 로봇, 말단장치 및 배치된 물체 사이의 충돌을 감지하지 않습니다.

팔레타이징 응용 시나리오에 다음과 같은 두 가지 경우가 있습니다.

1. 팔레타이징 시나리오에서 상자를 쌓을 때 로봇 자체가 이미 배치된 상자와 접촉할 경우가 있습니다(상자가 압착되거나 변형되지 않음). 이 충돌을 감지하면 Mech-Viz는 충돌을 피하기 위해 다른 배치 포인트를 계획하기 때문에 오히려 파レット를 완전히 채우지 못하게 됩니다.
2. 일반적으로 진공 그리퍼의 TCP는 그리퍼 표면이 아닌 모델 내부에 설정되어 있으며, 이로 인해 물체를 피킹할 때 말단장치와 피킹된 상자 모델이 중첩되는 상황이 발생합니다(소프트웨어는 말단장치와 작업물 사이의 충돌을 감지하지 않음). 로봇은 상자를 놓고 배치한 후 배치된 상자 모델은 시나리오

모델이 되며 이때 소프트웨어는 말단장치와 시나리오 속의 상자 모델과의 충돌을 감지하여 소프트웨어에서 충돌 경보가 발생하여 팔레타이징 스텝을 완료할 수 없습니다.

이 파라미터를 선택하면 소프트웨어는 로봇, 말단장치와 배치된 작업물 사이의 충돌을 감지하지 않고 위 문제를 피할 수 있습니다.

#### 포인트 클라우드 충돌 감지 모드

작업 현장의 실제 상황에 따라 파라미터를 설정하며 일반적으로 **Auto\***를 사용하면 됩니다. **로봇이 물체를 피킹하기 전의 이동 스텝은 \*NOTCHECK** 모드, 물체를 피킹한 후의 이동 스텝은 **CHECK** 모드를 선택할 수 있습니다.

<b>Auto</b>	기본값. "비전 이동" 스텝 및 "비전 이동"에 의존하는 "상대적인 이동" 스텝에 대해서만 포인트 클라우드와의 충돌을 감지하고 다른 이동 스텝에 대해 감지하지 않습니다.
<b>NOTCHECK</b>	모든 이동 스텝에 대해 포인트 클라우드와의 충돌을 감지하지 않습니다.
<b>CHECK</b>	모든 이동 스텝에 대해 포인트 클라우드와의 충돌을 감지합니다.



**충돌 감지 > 충돌 감지 구성 > 포인트 클라우드와 다른 물체의 충돌을 감지하기** 기능을 활성화하면 Mech-Viz 소프트웨어가 경로 계획을 수행할 때 로봇 모델, 말단장치와 포인트 클라우드 사이의 충돌을 감지합니다. 일반적으로 포인트 클라우드 충돌 감지 설정은 로봇이 피킹 과정에서 작업물과의 충돌 여부를 확인하는 것입니다. 공간에 노이즈가 있으면 소프트웨어가 물체 피킹 전의 경로를 계획할 때 로봇 모델, 말단장치 모델이 노이즈와 접촉하므로 포인트 클라우드 사이의 충돌로 잘못 감지하여 소프트웨어의 계획 오류를 초래할 수 있습니다.

#### 물체의 대칭성을 사용하지 않기

이 파라미터는 \*웨이포인트 유형\*을 \*작업물 포즈\*로 설정한 경우에만 적용됩니다. 예를 들어 웨이포인트 유형을 작업물 포즈로 설정한 이동/팔레타이징 스텝에 적용되며 웨이포인트 유형이 JPs 또는 TCP 포즈인 이동 스텝에는 적용되지 않습니다.

<b>None</b>	기본값. 모든 축의 대칭성을 사용합니다.
<b>AxisZ</b>	Z축의 대칭성만 사용하지 않습니다.
<b>AxisXy</b>	X, Y축의 대칭성을 사용하지 않습니다.
<b>All</b>	모든 축의 대칭성을 사용하지 않습니다.

대칭성을 사용하지 않으면 로봇은 물체 포즈에 따라 물체를 정확하게 배치합니다.



하지만 일부 특수한 상황에서 물체를 피킹하지 못할 때 **작업물 > 작업물 구성** 중의 \*회전 대칭\*을 설정할 수 있습니다. 인식된 물체에 대해 물체 대칭성을 설정하고 대칭 각도에 따라 작업물이 여러 개의 후보 포즈를 가질 수 있습니다. Mech-Viz 소프트웨어는 물체 피킹을 계획할 때 기본 포즈를 피킹할 수 없는 경우 후보 포즈를 피킹할 수 있는지 여부를 시도합니다. 회전 대칭 설정을 기반으로 계산된 후보 포즈는 Mech-Vision에서 출력된 원시 포즈와 다르면 물체 배치 포즈의 일관성을 보장할 수 없습니다.

#### 5.4.9.2. 피킹된 물체의 충돌 감지 모드

##### 시나리오&로봇과의 충돌 감지를 하지 않기

기본적으로 선택되어 있지 않습니다. 이 옵션을 선택한 후에는 **피킹된 물체\*와 \*시나리오 모델, 로봇** 사이의 충돌을 감지하지 않으므로 소프트웨어 충돌 감지의 계산량을 줄이고 Mech-Viz의 계획 속도를 최적화하며 전반적인 사이클 타임을 향상시킵니다. 일반적으로 로봇이 물체를 피킹한 후의 1~2개 이동 스텝에서 사용합니다.

충돌 감지를 수행하지 않으면 충돌 발생의 위험이 있으므로 주의하여 사용하시기 바랍니다.



menu:충돌 감지 구성 [잡고 있는 물체 구성>잡고 있는 작업물과 다른 물체 사이의 충돌을 감지하기] 기능을 활성화하면 잡고 있는 작업물과 시나리오 모델&로봇 사이의 충돌을 감지합니다. 팔레타이징 시나리오에서 비전을 통해 계산된 상자 치수와 실제 치수 사이에는 밀리미터 수준의 편차가 있으면 피킹 과정에서 상자 사이에 마찰이 발생할 수 있으며 충돌이 발생하지 않습니다. 분명히 충돌하지 않는 일부 움직임의 경우, 이러한 충돌을 감지하면 소프트웨어 계산량이 증가하고 Mech-Viz 계획 시간이 늘어나며 사이클 타임이 낭비됩니다. 시나리오 물체와의 충돌을 감지하지 않는 기능을 활성화하면 팔레타이징 시나리오에서 피킹된 작업물과 이미 배치된 상자 간의 충돌 감지에 영향을 미치지 않습니다. 파렛트 밑에 시나리오 물체가 있을 때 이 기능을 사용하면 팔레타이징 솔루션 선택 실패 문제를 피할 수 있습니다.

#### 포인트 클라우드와의 충돌 감지를 하지 않기

기본적으로 선택되어 있지 않습니다. 선택하면 **피킹된 물체\*와 \*시나리오 속의 포인트 클라우드** 사이의 충돌을 감지하지 않고 소프트웨어의 충돌 감지 계산량을 줄여 Mech-Viz의 계획 속도를 향상시키며 전체적인 사이클 타임을 최적화할 수 있습니다. 또한 피킹된 물체와 포인트 클라우드 노이즈 사이의 충돌을 잘못 감지하는 문제를 피할 수 있습니다.



- menu:충돌 감지 구성 [잡고 있는 물체 구성>잡고 있는 작업물과 다른 물체 사이의 충돌을 감지하기] 기능과 **포인트 클라우드 구성 > 포인트 클라우드와 다른 물체의 충돌을 감지하기** 기능을 동시에 활성화하면 잡고 있는 작업물 모델과 시나리오 속의 포인트 클라우드 사이의 충돌을 감지합니다.
- Mech-Vision이 포인트 클라우드와 물체 모델의 정보를 Mech-Viz로 보내면 포인트 클라우드와 물체 모델이 맞춰집니다. 로봇이 물체를 피킹한 후 모델은 계획된 경로를 따라 이동하고 피킹된 작업물 모델과 포인트 클라우드 간의 충돌이 발생합니다.
- 피킹된 작업물 모델과 포인트 클라우드 사이에서 이미 존재한 잘못된 충돌을 피할 수 없는 경우에, 이러한 충돌을 감지하면 불필요한 계산 작업량이 증가하고 Mech-Viz의 계획 시간이 늘어날 수 밖에 없습니다.

#### 5.4.9.3. 비이동 스텝 기본 파라미터

##### 실행을 건너뛰기

없음	기본값. 해당 스텝의 실행을 건너뛰지 않습니다.
시뮬레이션할 때만	시뮬레이션을 할 때만 해당 스텝의 실행을 건너뛵니다. 아웃 포트는 <b>실행 건너뛴 때의 아웃 포트</b> 파라미터를 통해 지정됩니다.
항상	실제 실행 또는 시뮬레이션을 할 때 모두 해당 스텝의 실행을 건너뛵니다. 아웃 포트는 <b>실행 건너뛴 때의 아웃 포트</b> 파라미터를 통해 지정됩니다.
조절 설명	<b>시뮬레이션할 때만</b> 또는 *항상*으로 설정할 때 프로젝트 실행 중에는 현재 스텝을 건너뛰고 계속 다음 스텝을 실행합니다. 예를 들어, "DI 체크" 스텝이 프로젝트에 연결되어 있고 시뮬레이션 중 외부 입력 신호가 없는 경우, "실행을 건너뛰기" 파라미터를 설정하지 않으면 해당 스텝을 실행할 때 프로젝트 실행이 중지됩니다. 이 파라미터를 설정한 후에는 이 스텝을 실행하지 않고 시뮬레이션이 완료될 때까지 계속 다음 스텝을 실행합니다.

##### 실행 건너뛴 때의 아웃 포트

**실행을 건너뛰기** 파라미터의 값은 **시뮬레이션할 때만** 또는 \*항상\*으로 설정해야 이 기능을 사용할 수 있습니다. 해당 스텝을 건너뛰고 계속 실행될 때의 아웃 포트를 지정합니다.

#### 5.4.9.4. 기본 이동 설정

"기본 이동 설정"은 이동 스텝을 사용할 때 조절해야 하는 중요한 파라미터이며 로봇이 웨이포인트 위치로 이동하는 과정의 **속도** 및 \*운동 방식\*을 컨트롤하는 데 사용됩니다.

**피킹&배치**

<b>지정하지 않음</b>	기본값
<b>피킹</b>	“비전 이동” 전의 이동 스텝에 적용됩니다.
<b>배치</b>	“비전 이동” 후의 이동 스텝에 적용됩니다.
<b>조절 설명</b>	Mech-Viz 프로젝트 실행 논리를 검사하는 데 사용될 수 있습니다. 작업 현장의 실제 작업 프로세스에 따라 피킹한 다음에 배치한다는 순서를 기본 원칙으로 이동 스텝에 대해 피킹&배치를 설정합니다.

**운동 방식**

<b>관절 운동</b>	로봇의 실행 경로가 원호이기 때문에 더 원활하게 이동할 수 있어 이동 중에 싱귤래리티가 나타나기 쉽지 않음을 의미합니다. 관절 운동은 경로 계획의 정밀도에 대한 요구가 그리 높지 않고 로봇이 큰 범위에서 이동하는 시나리오에 적용됩니다.
<b>직선 운동</b>	로봇의 실행 경로가 직선이며 로봇 경로의 정밀도에 대한 요구가 높습니다. 용접, 접착제 도포, 피킹 등 경로 정밀도에 대한 요구가 상대적으로 높은 시나리오에 적용됩니다.

**속도&가속도**

로봇의 이동 속도를 결정합니다. 일반적으로 가속도 파라미터의 수치가 속도보다 작아야 합니다. 가속도 파라미터의 수치가 속도보다 클 때 로봇의 동작이 원활하지 않습니다.

- ! 피킹의 안정성을 위해 "비전 이동"과 전/후의 이동 속도를 낮게 설정해야 합니다.

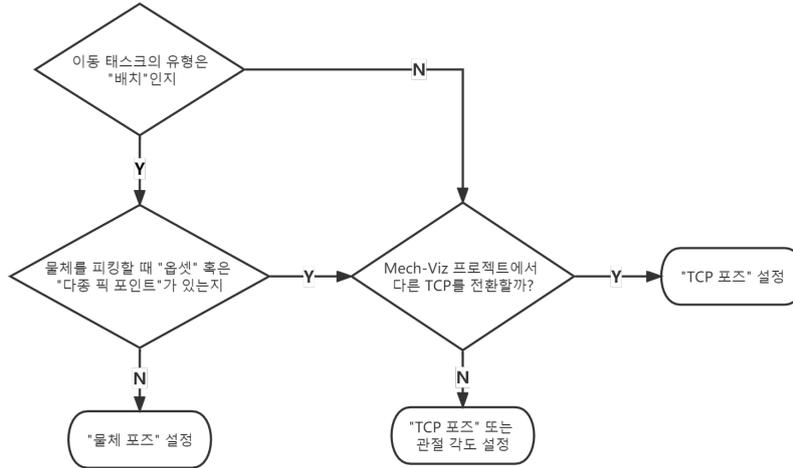
**회전 반경**

<b>기본값</b>	50.00mm
<b>조절 설명</b>	일반적으로 조절할 필요가 없고 기본값을 사용합니다.  회전 반경은 웨이포인트와 로봇이 회전하기 시작하는 지점 사이의 거리를 나타냅니다. 회전 반경이 클수록 로봇의 이동이 더 원활해집니다. 로봇이 비교적 작은 공간에서 이동하는 경우 회전 반경을 더 작은 값으로 설정하십시오.  로봇이 넓은 공간에서 작업하고 다른 장애물이 없으며 로봇의 두 경로 사이의 거리가 멀었을 때, 회전 반경을 적절히 크게 조절하여 로봇의 움직임을 보다 원활하게 할 수 있습니다.

**5.4.9.5. 웨이포인트 유형**

<b>TCP</b>	웨이포인트는 TCP 좌표계의 X, Y, Z 값과 오일러 각/사원수로 표시됩니다.
<b>JPs</b>	웨이포인트는 로봇 각 관절의 수치로 표시됩니다.
<b>작업물 포즈</b>	웨이포인트는 물체 좌표계의 X, Y, Z 값과 오일러 각/사원수로 표시됩니다.

웨이포인트 유형을 선택할 때 판단 방법은 다음 그림과 같습니다.



#### 포즈 편집

포즈를 직접 편집하고 복사/붙여넣을 수 있으며 사원수와 오일러 각 두 가지 형식을 지원합니다.

#### 포즈 변환

자체 정의를 통해 현재 포즈를 새 포즈로 변환하는 데 사용되며 포즈 파인튜닝에 적용됩니다.

#### 포즈 보정

작업물의 좌표계를 계산하는 ABB 로봇 삼점법과 유사하며 대상 물체가 회전할 때 포즈를 확인하기 쉽지 않은 시나리오에 적합합니다. 예를 들어 기울어진 직육면체의 회전 포즈를 계산하기 어려운 경우, 포즈 보정을 통해 직육면체의 회전 포즈를 먼저 계산하고 로봇이 계산된 포즈에 따라 실행하도록 합니다.

#### JPs 편집

포즈를 편집하는 기능과 유사하여 관절 각도를 복사/붙여넣을 수 있고 라디안과 각도 두 가지 단위를 지원합니다.



가상 로봇을 설정된 웨이포인트 위치로 이동합니다.



배치된 로봇의 포즈를 읽어 웨이포인트로 설정합니다.



현재 웨이포인트에 대한 모든 JPs 솔루션을 표시합니다.

### 5.4.9.6. 관절 각도 제한 조건

#### 개념

##### 어깨

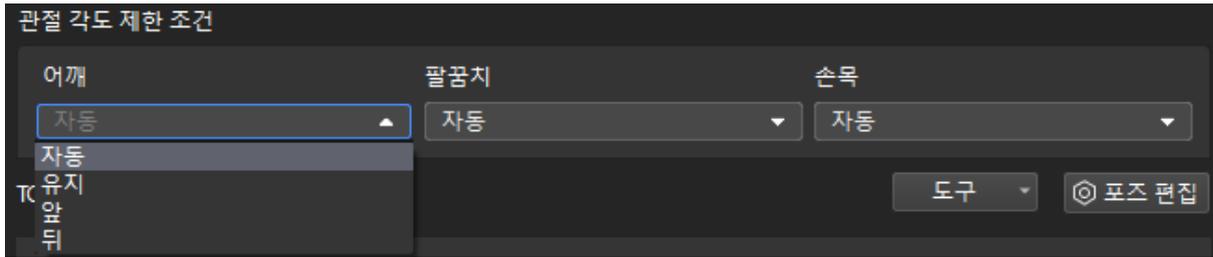
손목 관절 중심과 Axis1의 상대적 위치 관계입니다. Axis1은 로봇 축1의 회전 중심 축을 나타냅니다.

##### 팔꿈치

손목 관절과 LowerArm의 상대적 위치 관계입니다. LowerArm은 로봇의 축2, 축3의 회전 중심의 연결선을 나타냅니다.

##### 손목

손목 관절은 사실 로봇의 축5입니다. 축5 각도의 양/음의 관계는 손목이 뒤집히는 상태를 나타내고, Wrist center는 손목 관절의 중심을 나타냅니다.



## 값 리스트

- 자동** 로봇 관절의 움직임에는 제한 조건이 없습니다. 최적의 솔루션은 각 축의 회전폭이 가장 작은 솔루션입니다.
- 유지** 현재 JPs 상태에 따라 다음 JPs 솔루션을 제한합니다. 예를 들어, 로봇 축3의 현재 JPs가 양수이면 로봇 축3의 JPs가 양수인 솔루션만 다음 웨이포인트로 고려됩니다.
- 앞** 손목 관절의 중심은 Axis1 앞에 있습니다.
- 뒤** 손목 관절의 중심은 Axis1뒤에 있습니다.

**모든 JP 표시** 버튼을 클릭하면 현재 포즈에 대한 모든 JPs 솔루션을 표시하는 창이 나타납니다. 포즈 중 하나를 클릭하면 솔루션의 해당 로봇 포즈가 3D 시뮬레이션 영역에 표시됩니다. 따라서 아래와 같이 다양한 제한 조건에서 가능한 솔루션을 알 수 있습니다.



1. 관절 각도 제한 조건은 6축 로봇에만 적용되며 4축 로봇은 어깨, 팔꿈치, 손목을 뒤척이지 않습니다.
2. “상대적인 이동”, “미리 설정한 파렛트 패턴”, “자체 정의한 파렛트 패턴” 스텝에서 이 파라미터를 설정할 수 없으며 기본적으로 로봇의 어깨, 팔꿈치 및 손목이 변경되지 않습니다. 즉, 이러한 스텝에서 로봇이 싱글래리터를 거치지 않도록 제한합니다.

## 5.5. 프로젝트 시뮬레이션 최적화

프로젝트 리소스를 구성하고 작업 흐름을 구축한 후 시뮬레이션을 통해 프로젝트의 실행 효과를 검증할 수 있습니다.

새로운 프로젝트를 구축한 후 다양한 문제들이 발생할 수 있습니다. 작업 현장의 구체적인 수요를 충족할 수 있도록 이 부분에서 소개한 작업 설명에 따라 프로젝트를 구성하십시오.

깊은 바구니에서 물체를 피킹하는 등 제한이 있는 응용 시나리오에서 말단장치와 바구니 또는 다른 장애물 사이의 충돌을 방지하는 것이 매우 중요합니다. 시뮬레이션을 시작하기 전에 “충돌 감지” 기능을 활성화하여 발생할 수 있는 충돌을 감지해야 합니다. 충돌 감지에 대한 구성을 완성하면 툴바 왼쪽에 있는 “시뮬레이션” 버튼을 클릭하여 프로젝트 시뮬레이션을 시작할 수 있습니다.

### 충돌 감지

시뮬레이션을 진행하는 동안에는 충돌이 발생하거나 웨이포인트에 도달하지 못하는 등 문제들이 나타나면 계획 기록 또는 로그에서 문제가 발생한 구체적인 원인을 분석하고 프로젝트를 최적화할 수 있습니다.

### 계획 기록

### 로그

프로젝트를 최적화한 후 실제 로봇을 실행함으로써 작업이 올바른지 검증할 수 있습니다.

## 실제 로봇을 실행하기

시뮬레이션과 실제 실행 과정에서 문제가 없으면 계산 설정을 수정하여 소프트웨어의 실행 속도를 높일 수 있습니다.

## 계산 설정

### 5.5.1. 충돌 감지

이 부분에서는 충돌 감지 구성과 관련된 내용을 소개하겠습니다. 주로 다음과 몇 부분으로 나뉘집니다:

#### 소개

머신 텐징, 디/팔레타이징 등 응용 시나리오에서 프로젝트의 중단 없는 실행을 보장하려면 로봇이 빈 또는 기타 장애물과 충돌하지 않도록 하는 것이 매우 중요합니다. Mech-Viz는 충돌 감지 기능을 통해 이동 중 로봇의 불필요한 충돌을 방지하고 로봇의 경로를 계획할 때 Mech-Viz는 3D 시뮬레이션 공간에서 충돌이 발생한 부분을 하이라이트로 표시하여 사용자에게 상기시키고 충돌이 실제로 발생하지 않도록 프로젝트 실행을 종료합니다.

#### 계산 설정

계산 설정은 소프트웨어에 의한 충돌 계산 및 기록을 수정하는 데 사용되어 프로젝트 실행 속도에 영향을 미칩니다.

계획 기록을 통해 신속하게 문제의 원인을 찾아 문제를 해결하기 위해 프로젝트 디버깅 단계에서 **계산 설정**의 파라미터를 **각 호부 방식에 해당하는 완전한 충돌을 계산하기 + 계획 기록에 저장하기로** 설정하는 것을 권장합니다.

#### 충돌 감지 구성

충돌 감지는 충돌 모델의 쌍별 조합으로 구현되며 여러 유형의 충돌 감지 결합을 포함합니다. 충돌 감지를 활성화하기 전에 프로젝트 리소스에서 해당 충돌 모델을 추가하고 구성해야 합니다.

- **말단장치** 내용을 참조하여 말단장치 모델을 추가하고 구성합니다.
- **시나리오 물체** 내용을 참조하여 시나리오 물체 모델을 추가하고 구성합니다.
- 작업물 모델은 Mech-Vision에서 출력한 데이터로 자동으로 생성되며 구성 방법은 다음과 같습니다.

Mech-Viz는 기본적으로 다음 물체 사이의 충돌을 감지합니다:

1	로봇 관절	로봇 관절
2	로봇 관절	시나리오 물체
3	로봇 관절	말단장치
4	시나리오 물체	말단장치

뿐만 아니라 다음과 같은 물체 사이의 충돌 감지를 구성할 수도 있습니다.

**충돌 감지** 패널에서 “충돌 감지 구성” 버튼을 클릭하여 충돌 감지 구성 화면으로 들어가서 구성할 수 있습니다.

#### 포인트 클라우드와 다른 물체의 충돌을 감지하기

왼쪽 “포인트 클라우드 구성” 패널에서 **포인트 클라우드와 다른 물체의 충돌을 감지하기** 버튼을 선택하면 포인트 클라우드와 말단장치 사이의 충돌을 감지합니다. 뿐만 아니라 포인트 클라우드와 로봇 관절, 피킹된 물체의 충돌 감지를 구성할 수 있습니다.



“비전 이동” 스텝 및 “비전 이동”에 의존하는 “상대적인 이동” 스텝에 대해서만 포인트 클라우드와의 충돌을 감지하고 다른 이동 스텝에 대해 포인트 클라우드와의 충돌을 감지하지 않습니다.

#### 포인트 클라우드 구성 - 포인트 클라우드 정육면체 변의 길이

포인트 클라우드 정육면체는 포인트 클라우드 중의 포인트를 중심으로 생성된 정육면체입니다. 포인트 클라우드 정육면체가 다른 물체와 충돌하면 포인트 클라우드가 물체와 충돌하는 것으로 간주합니다.

충돌 감지 정밀도와 속도에 대한 구체적인 요구 사항에 따라 이 파라미터를 조절하십시오.

#### 말단장치 구성 - 충돌 부피 역치

포인트 클라우드 정육면체가 말단장치의 충돌 모델과 충돌할 수 있는 최대 부피입니다. 충돌 부피가 이 역치를 초과하면 충돌로 판단됩니다.

실제 프로젝트 요구 사항에 따라 이 파라미터를 설정하십시오.

#### 로봇 관절 구성 - 로봇 관절

로봇 관절은 손목, 아랫팔, 윗팔, 베이스가 포함되며 기본적으로 모든 관절이 포인트 클라우드와의 충돌 감지에 참여하지 않습니다.

실제 요구 사항에 따라 해당 관절을 선택하여 포인트 클라우드와의 충돌 감지를 수행하십시오.

#### 로봇 관절 구성 - 충돌 부피 역치

포인트 클라우드 정육면체가 로봇 관절의 충돌 모델과 충돌할 수 있는 최대 부피입니다. 충돌 부피가 이 역치를 초과하면 충돌로 판단됩니다.

실제 프로젝트 요구 사항에 따라 이 파라미터를 설정하십시오.

#### 피킹된 작업물과 다른 물체 사이의 충돌 구성

피킹된 작업물과 다른 물체 간의 충돌을 감지하며 주로 작업물을 옮기는 과정에서의 충돌을 감지하는 데 사용됩니다.

이미 피킹된 작업물 유형에 따라 **직육면체** 또는 **사용자 정의 모델** 을 선택하십시오.

#### 직육면체

##### 직육면체 모델을 추가하는 방식

Mech-Viz는 Mech-Vision에서 제공한 작업물 치수에 따라 자동으로 직육면체 작업물의 충돌 모델을 생성할 수 있습니다. 따라서 Mech-Vision에서 관련 구성을 수행하여 작업물의 치수를 출력할 수 있습니다.

1. Mech-Vision에서 “평면 포인트 클라우드의 포즈와 크기를 계산하기”(다중 물체 시나리오에 사용하는 것을 권장함) 또는 “물체 사이즈 읽기”(단일 물체 시나리오에 사용하는 것을 권장함) 스텝을 사용하여 작업물의 치수 정보를 획득하십시오.
2. “출력” 스텝을 통해 치수 정보를 출력하면 Mech-Viz는 작업물의 치수 정보를 획득하고 대응한 충돌 모델을 생성할 수 있습니다.

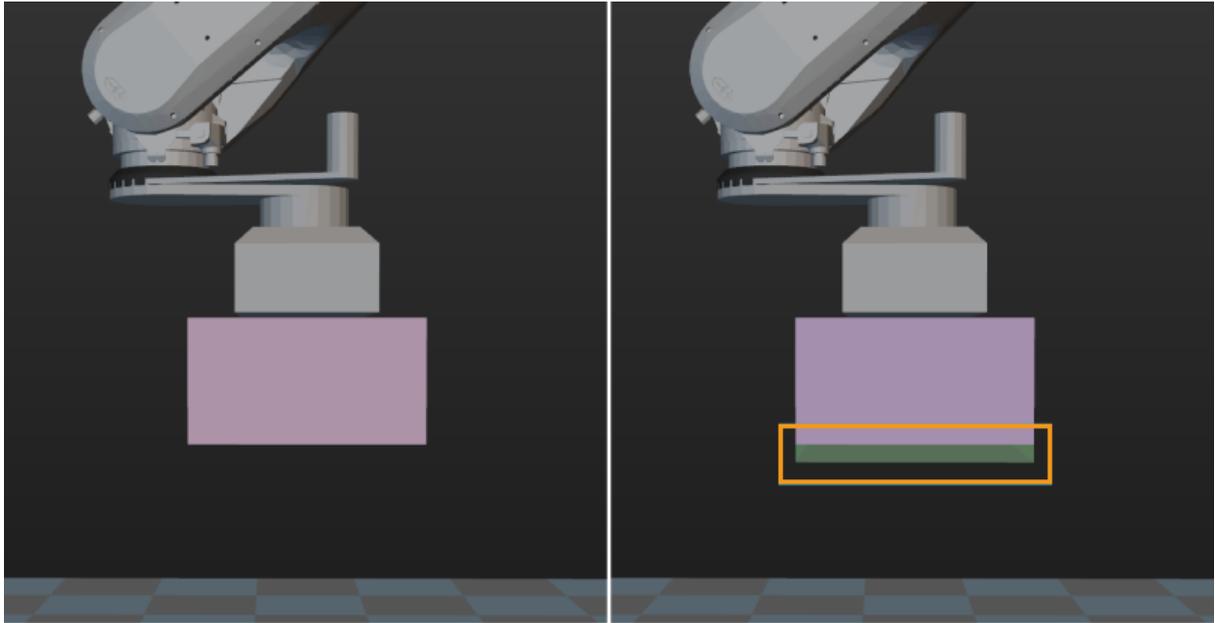


출력한 "픽 포인트의 포즈" 와 "물체 치수" 의 수가 일치해야 합니다.

### 직육면체 하단 안전 거리 설정

혼합(치수) 팔레타이징 프로젝트의 경우 직육면체 작업물 모델에 대한 하단 안전 거리를 설정해야 합니다. 하단 안전 거리를 설정한 후 작업물 모델 하단에 해당 충돌 감지 범위가 추가됩니다. 다른 물체가 이 범위에 들어간 것이 감지되면 충돌로 판단됩니다. 따라서 하단 안전 거리를 설정하면 작업물이 이동할 때 배치된 작업물과 충돌하는 것을 보다 효과적으로 방지할 수 있습니다.

직육면체 작업물의 경우, 혼합 팔레타이징 시나리오에서 물체를 잡고 있거나 배치하는 동안 하단 안전 거리 범위내에서 충돌도 감지해야 합니다. 옵션을 선택한 후 직육면체 하단 안전 거리를 설정하십시오.



### 충돌 부피 역치

포인트 클라우드 정육면체와 말단장치 모델 사이에 허용되는 최대 충돌 부피입니다. 충돌 부피가 이 역치보다 크면 충돌로 간주됩니다. **포인트 클라우드와 다른 물체의 충돌을 감지하기** 기능을 활성화한 후 이 파라미터를 설정할 수 있습니다.

실제 프로젝트 요구 사항에 따라 이 파라미터를 설정하십시오.

### 사용자 정의 모델

#### 사용자 정의 모델을 추가하는 방식

1. 작업물의 stl 및 binvox 모델을 Mech-Viz 프로젝트 폴더에 있는 **collision\_models** 파일(이 파일이 없으면 새로 생성해야 함)에 옮깁니다.
2. Mech-Vision에서 해당 스텝의 "포즈 분류 레이블" 포트를 통해 레이블 이름을 확인하고 모델 파일의 이름을 대응한 레이블 이름으로 수정합니다. Mech-Viz이 프로젝트 리소스에 작업물 구성을 추가한 경우, 작업물 이름도 레이블 이름과 일치해야 합니다.
3. Mech-Vision에서 **포인트 클라우드를 외부 서비스로 보내기** 스텝의 **물체 정보를 전송하기** 파라미터를 선택하여 **입력은 카메라 좌표계에 있는지**를 연체크하며 해당 스텝의 모든 입력 포트에 데이터 스트림을 연결시켜야 합니다.



출력한 포인트 클라우드와 포즈는 모두 로봇 좌표계에 있어야 합니다.

### 충돌 부피 역치

포인트 클라우드 정육면체와 말단장치 모델 사이에 허용되는 최대 충돌 부피입니다. 충돌 부피가 이 역치보다 크면 충돌로 간주됩니다. **포인트 클라우드와 다른 물체의 충돌을 감지하기** 기능을 활성화한

후 이 파라미터를 설정할 수 있습니다.

실제 프로젝트 요구 사항에 따라 이 파라미터를 설정하십시오.

## 5.5.2. 계획 기록

이 부분은 주로 계획 기록에 관한 내용입니다. 다음과 같이 몇 부분으로 나뉩니다:

### 소개

계획 기록 결과의 전체 구조는 Mech-Viz 계획의 전 과정을 상세하고 완전하게 기록하는 트리 구조입니다.

로그와 비교하면 계획 기록은 계획의 각 노드(특히 실패한 노드)를 자세히 볼 수 있습니다.

- 일부 오류는 일련의 오류로, 하나의 자식 노드 계획이 실패하면 전체 계획이 실패합니다. 예를 들어 어떤 관절 각도를 잘못 설정하면 로봇은 계획한 위치에 도달할 수 없습니다.
- 일부 오류는 일련의 오류로, 하나의 자식 노드 계획이 실패하면 전체 계획이 실패합니다. 예를 들어 Mech-Vision에서 출력한 픽 포인트에서 노드 중 하나라도 성공적으로 실행되면 해당 픽 포인트를 사용하여 물체를 피킹할 수 있음을 의미합니다.



프로젝트를 실행하는 과정에서 실패한 노드에 중점을 둘 수 있습니다.

### 충돌 결과 시각화

#### 충돌 계산 & 기록

일반적으로 충돌 계산 & 기록은 충돌 결과의 시각화에 사용됩니다.

**계산 설정**의 파라미터를 각 호부 방식에 해당하는 완전한 충돌을 계산하기 + 계획 기록에 저장하기로 설정하면 소프트웨어가 완전한 충돌을 기록하고 계획 기록으로 저장할 것입니다.

#### 계획 기록 중 충돌 감지 결과의 그래픽 프롬프트

소프트웨어의 왼쪽 상단 메뉴 바에서 “표시” 버튼을 클릭한 다음 **계획 시 충돌을 표시하기**(기본적으로 소프트웨어를 부팅할 때 이미 선택되어 있음)를 선택합니다.

시나리오 충돌	충돌이 발생하는 스텝의 이름을 클릭하면 충돌하는 모든 물체가 하이라이트로 표시됩니다.
	충돌이 발생하는 물체를 클릭하면 해당 충돌 물체가 하이라이트로 표시됩니다.
포인트 클라우드 충돌	충돌이 발생하는 스텝의 이름을 클릭하면 충돌하는 모든 물체가 하이라이트로 표시됩니다.
	<b>충돌 부피</b> 를 클릭하면 충돌한 포인트 클라우드가 하이라이트로 표시됩니다.
	포인트 클라우드가 충돌할 때 <b>충돌을 기록하지 않으면</b> 나머지 충돌 물체만 하이라이트로 표시됩니다.
대상 물체 충돌	대상 물체가 충돌하면 하이라이트로 표시됩니다.

### 일반적인 실패한 경우 및 해당 솔루션

#### 로봇 자체 충돌

계획 내용	결과	상세 정보	자식 노트 실패한 모든 원인
2022-07-07_19-24-38_267 현재 태스크 관절1 관절2	로봇 자체 충돌 移动_2 손목 위팔	손목과 위팔 충돌 관절 각도: (3,40339,-1.5708,3.21141,-1.5708,0,0)	로봇 자체 충돌.

**실패 원인**     로봇의 아랫팔이 베이스와 충돌합니다.

**솔루션**         자체 충돌을 피하기 위해 현재 스텝에 해당하는 포즈를 조정합니다.

#### 시나리오 물체와의 충돌

계획 내용	결과	상세 정보
2022-07-19_15-27-53_590	성공	Home, 이미지 캡처 포인트
2022-07-19_15-27-58_429	성공	상대적인 이동_1, 비전 이동_1
2022-07-19_15-28-23_205 현재 태스크 관절1 시나리오 물체 명칭 시나리오 물체 포즈 시나리오 물체 크기	시나리오 물체와의 충돌 Home, 엔드 이펙터 명칭: sample 베이스 boxbox[-x] {(0.0360265, 0, 9.65953e-17), (1... (0.002, 0.4, 0.2)	베이스와 시나리오 물체 [box] 관절 각도: {1.5245, -2.23394,

**실패 원인**     말단장치가 시나리오 [1]과 충돌합니다.

**솔루션**         충돌이 발생한 시나리오 물체의 위치를 확인하고 시나리오 물체의 위치를 변경할 수 없는 경우 현재 스텝의 해당 포즈를 조정합니다.

#### 목표점 위치에 도달하지 못함

계획 내용	결과	상세 정보	자식 노트 실패한 모든 원인
2022-07-07_19-24-16_373 현재 태스크 관절 각도를 선택할 때 참조하는 태...	운동 대상 도달하지 못하기 移动_1, 엔드 이펙터 명칭: 0 deg...	0/0/0-초시/소프트 리미트/최종 포즈 개수, {어깨: ... TCP 포즈: (1.7,2.54569,2.14608,0.5,-0.5,0.5,0.5) 관절 각도: {}	운동 대상 도달하지 못하기.

**실패 원인**     로봇의 운동 범위를 초과하여 로봇은 이 포즈에 도달할 수 없습니다.

**솔루션**         로봇이 도달할 수 있는 범위 내에 있도록 현재 스텝의 해당 포즈를 조절하십시오.

#### 충돌 부피가 역치를 초과함

"visual_move_1"에 대칭성/... 현재 태스크 피킹 해결 (필요할 때 선택... 피킹에 관절 각도를 선택... "visual_move_1"의 트랙... 현재 태스크 트랙 시작 트랙 엔드 관절1 충돌 면적	대칭성 시도 실패 visual_move_1, 엔드 이펙터 명칭... 성공 성공 포인트 클라우드 충돌 면적이 한... visual_move_1, 엔드 이펙터 명칭... relative_move_4, 엔드 이펙터 명... visual_move_1, 엔드 이펙터 명칭... 엔드 이펙터 91.60000000000015	0°, -10° (물체 Z축, X축회전... TCP 포즈: (-1.76712,0.3785... 엔드 이펙터 스캔한 포인트 ... TCP 포즈: (-1.76347,0.4046... TCP 포즈: (-1.50659,0.2226...	포인트 클라우드 충돌 면적이 한계를 초과하기. 포인트 클라우드 충돌 면적이 한계를 초과하기.
--	---	---	--

**실패 원인**     말단장치와 포인트 클라우드 간의 충돌 부피가 설정된 역치를 초과합니다.

**솔루션** 충돌 부피의 역치는 감지된 충돌 역치와 충돌에 대한 실제 프로젝트의 허용 편차에 따라 수정될 수 있습니다. 또한 실제 요구에 따라 TCP 포즈, 픽 포인트와 말단장치 충돌 모델의 포즈를 조정할 수 있습니다.

### 5.5.3. 실제 로봇을 실행하기

실제 로봇을 실행하려면 아래 작업을 수행해야 합니다.

#### 1. 로봇 연결 상태를 확인하기

Mech-Viz를 사용하여 실제 로봇의 운동을 컨트롤하기 전에 **로봇 통신 구성**을 우선 완성해야 합니다.

#### 2. 시뮬레이션 실행을 비활성화하기

**Mech-Center > 구성 설정 > Mech-Viz**화면에서 **시뮬레이션 실행** 버튼을 언체크하십시오.

#### 3. 로봇 속도를 낮추기

 안전 사고를 방지하기 위해 실제 로봇을 실행하기 전에 반드시 로봇의 속도를 낮춰야 합니다!

Mech-Viz 툴 바에 있는 **속도** 및 **가속도**를 5% 로 설정하십시오.

#### 4. 로봇을 실행하기

Mech-Viz 왼쪽 상단에 있는 [ **실행** ] 버튼을 클릭하거나 Mech-Center 메인 인터페이스에 있는 [ **실행** ] 버튼을 클릭하면 로봇은 계획한 경로에 따라 이동할 것입니다.

 로봇이 움직이고 있을 때 비상시 티치 펜던트의 비상정지 버튼을 제때 눌러주십시오.

#### 5. 다음 방법을 통해 프로젝트의 실행 과정을 멈출 수 있습니다.

Mech-Viz 왼쪽 상단에 있는 [ **정지** ] 버튼을 클릭합니다.

실제 로봇을 실행할 때 오류가 발생하면 **계획 기록** 및 **로그** 내용을 통해 오류를 해결해 보십시오. 오류가 없으면 **계산 설정**을 수정하여 실행 속도를 향상시킬 수 있습니다.

### 5.5.4. 계산 설정

계산 설정은 소프트웨어에서 감지된 충돌에 대한 계산 방식 및 충돌을 계획 기록에 저장할지 여부를 설정하는 데 사용되며 프로젝트의 실행 속도에 영향을 미칩니다.

<p>충돌을 계산하기: 각 후보 방식에 해당하는 완전한 충돌을 계산하기(파라미터 조정 및 충돌 시각화를 위해) -</p> <p>충돌을 기록하기: 계획 기록에 저장하기(속도가 느리긴, 파라미터 조정 및 시각화를 위해) -</p>	<p><b>프로젝트 디버깅 단계에 사용되며 계산 속도가 느립니다.</b></p>
<p>충돌을 계산하기: 충돌 정도가 역치를 초과하면 현재 방식의 남은 계산을 건너뛰기(연속 실행에 적합) -</p> <p>충돌을 기록하기: 계획 기록에 저장하기(속도가 느리긴, 파라미터 조정 및 시각화를 위해) -</p>	<p>계획 기록 중의 충돌 감지와 관련된 항목을 클릭하면 충돌을 시각화할 수 있지만 일부만 표시됩니다.</p>
<p>충돌을 계산하기: 각 후보 방식에 해당하는 완전한 충돌을 계산하기(파라미터 조정 및 충돌 시각화를 위해) -</p> <p>충돌을 기록하기: 계획 기록에 저장하지 않기(연속 실행에 적합) -</p>	<p>계획 기록에서 완전한 충돌 접촉을 표시할 수 있지만 구체적인 충돌이 발생한 위치를 시각화하지 못합니다.</p>
<p>충돌을 계산하기: 충돌 정도가 역치를 초과하면 현재 방식의 남은 계산을 건너뛰기(연속 실행에 적합) -</p> <p>충돌을 기록하기: 계획 기록에 저장하지 않기(연속 실행에 적합) -</p>	<p><b>안정적인 생산 단계에 사용되며 충돌을 계획 기록으로 저장하지 않아 계산 속도가 빠릅니다.</b></p>

## 5.6. 도구 사용 가이드

### 5.6.1. 모델 편집기

이 부분에서는 모델 편집기의 인터페이스와 사용 방법에 대해 소개하겠습니다.

#### 소개

Mech-Viz 소프트웨어에 사용되는 말단장치 충돌 모델의 각 부분이 반드시 볼록다면체(볼록 껍질)의 obj 모델이어야 합니다. 모델 편집기를 사용하면 말단장치 모델을 위해 볼록다면체를 생성하고 obj 모델로 전환할 수 있습니다. 또한, 모델 편집기를 사용하면 말단장치 모델의 좌표계를 재구성하고 모델을 stl 포맷으로 도출할 수 있습니다.

#### 지원하는 포맷

- 도입할 수 있는 기준 모델 포맷 : stp, step, stl, obj
- 도출할 수 있는 기준 모델 포맷 : stl
- 도출할 수 있는 볼록다면체 모델 포맷: obj

#### 인터페이스 소개

주로 다섯 가지 부분으로 구성됩니다:



상세한 내용은 해당 링크를 클릭하여 참조하십시오.

<b>메뉴 바</b>	모델 파일을 도입&도출하기. 작업을 편집하기.사용자 매뉴얼을 열기
<b>툴 바</b>	모델 편집기의 기본 도구
<b>기준 모델</b>	도입한 기준 모델 파일을 표시하기
<b>리소스</b>	기본 기하학체,볼록 껍질과 중심점을 표시하기

메뉴 바	모델 파일을 도입&도출하기. 작업을 편집하기.사용자 매뉴얼을 열기
3D 편집 영역	모델의 디스플레이와 편집

### 블록다면체 모델 생성

stl, stp, step, 무효한 obj 모델을 전체적으로 블록다면체로 구성된 obj 모델로 전환하려면 다음과 같이 작업하십시오.

"좌표계 설정" 및 "블록다면체 생성"의 데모 비디오:

▶ <https://docs.mech-mind.net/download/ModelEditor/ModelEditorDemoen.mp4> (video)

### 기준 모델을 도입하기

도입할 수 있는 기준 모델의 포맷: stl, stp, step, obj.

기준 모델을 도입하려면 다음 단계 중 하나를 수행하십시오.

- **파일 > 기준 모델을 도입하기** 버튼을 클릭한 후 파일 선택 창에서 기준 모델 파일을 선택하십시오.
- 기준 모델 파일을 모델 편집기로 드래그하십시오.

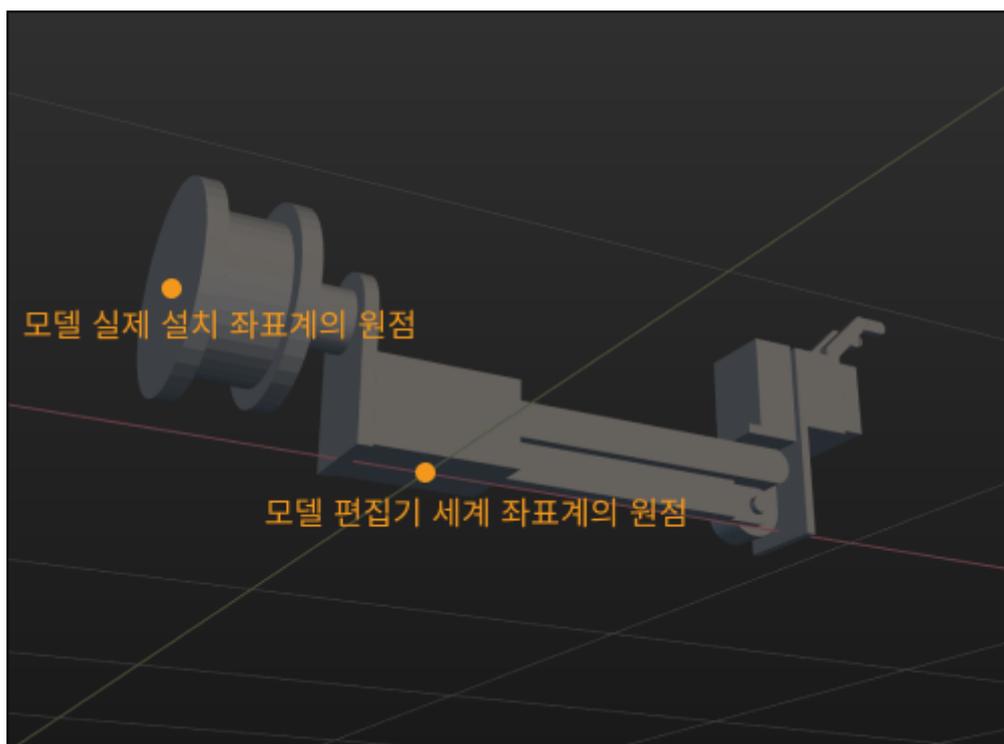
모델의 실제 치수에 따라 “단위”를 선택한 후 **[확인]**을 클릭하십시오.



“모델을 로드하지 못했음”이라는 알람이 나타나면 **STEP/STP 파일의 유효성 판정** 내용을 참조하여 모델 파일을 확인하십시오.

### 좌표계 설정

"모델 편집기의 세계 좌표계"가 "모델의 실제 설치 좌표계"와 일치하지 않는 경우 좌표계를 재설정해야 합니다. 일치한다면 이 부분 내용은 건너뛰고 **블록다면체 생성**부터 계속 읽으시면 됩니다.

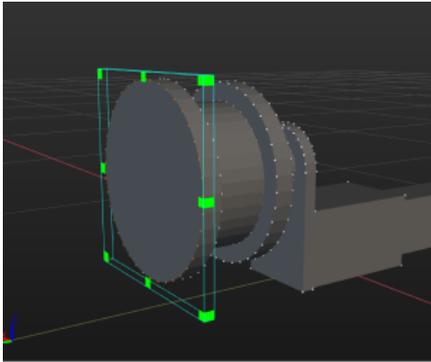
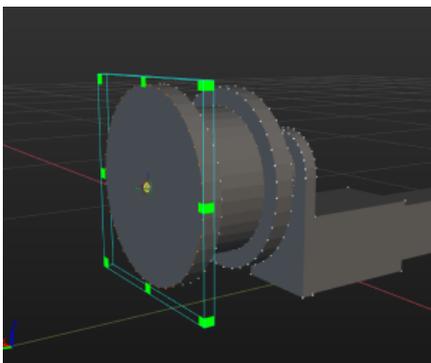
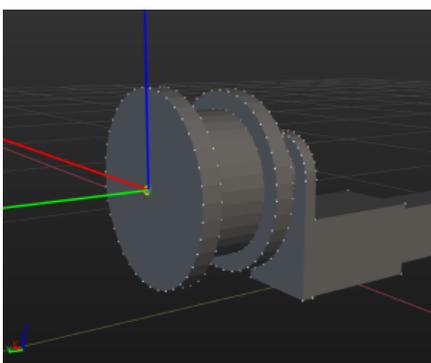


좌표계를 설정할 때 새로운 원점을 설정해야 하거나 축 방향을 조정해야 합니다.

### 새로운 좌표계 원점을 설정하기

모델의 꼭짓점을 선택하고 이 꼭짓점들을 참조로 사용하여 중심점을 만든 다음 이 "중심점"을 "좌표계 원점"으로 설정하거나 "모델 꼭짓점"을 "좌표계 원점"으로 직접 선택합니다.

다음 내용은 중심점을 만드는 방법을 예로 들어 설명합니다.

<p>“기준 모델” 표시줄에 모델 이름을 선택합니다.</p> <p><b>[3D 선택 도구]</b>를 클릭하고 플랜지의 가장 바깥쪽 레이어의 모든 꼭짓점을 선택하도록 3D 선택 프레임을 조정합니다. 구체적인 사용법은 <b>3D 선택 프레임 사용 가이드</b> 내용을 참조하십시오.</p> <p>중심점은 모델의 꼭짓점을 기준으로 생성되기 때문에 선택할 때 처리할 부분의 “꼭짓점”이 모두 선택되어야 합니다.</p>	
<p><b>[중심점을 생성하기]</b>를 클릭하여 이 점을 좌표계의 원점으로 사용하십시오.</p>	
<p><b>[좌표계 설정]</b> 버튼을 클릭하십시오. 오른쪽 좌표계 파라미터 표시줄에서 <b>중심점 선택</b> 버튼을 선택하십시오.</p> <p><b>좌표 원점</b> 오른쪽의 <b>[선택]</b> 버튼을 단번 클릭하십시오. 3D 편집 영역에서 중심점을 클릭하면 이 중심점을 좌표 원점으로 좌표계를 생성합니다.</p>	

사용 중인 모델에 좌표 원점으로 사용할 수 있는 적합한 모델 꼭짓점이 있는 경우 다음 단계를 수행하십시오:

1. **[좌표계 설정]** 버튼을 클릭하십시오.
2. 좌표계 파라미터 표시줄에서 “좌표 원점” 오른쪽의 **[선택]** 버튼을 클릭하십시오.
3. 3D 편집 영역에서 모델의 꼭짓점을 선택하십시오.



- **중심점 선택** 버튼을 언체크하면 좌표 원점, X/Y/Z축은 “모델 꼭짓점”만 선택할 수 있습니다. **중심점 선택** 버튼을 선택하면 좌표 원점, X/Y/Z축은 “중심점”만 선택할 수 있습니다.

- 좌표축의 방향을 더 조정하려면 [ 확인 ] 버튼을 클릭하지 마십시오.

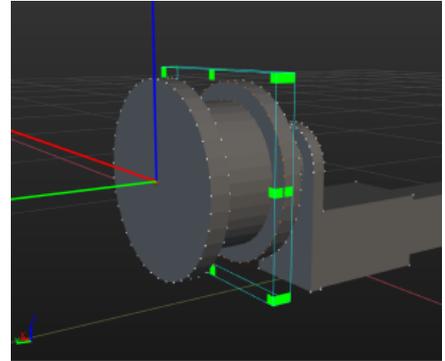
### 축 방향 조정

이전 단계를 설정한 후 좌표계의 각 축의 방향이 맞으면 방향을 조정할 필요가 없습니다. 그렇지 않으면 "중심점" 또는 "모델 꼭짓점"을 통해 각 축의 방향을 조정할 수 있습니다. 구체적으로 다음과 같습니다.

**중심점 선택** 버튼을 언체크하십시오.

[ 3D 선택 도구 ]를 클릭하고 선택 프레임을 조정하며 가장 바깥쪽 꼭짓점에 평행한 꼭짓점을 선택하십시오.

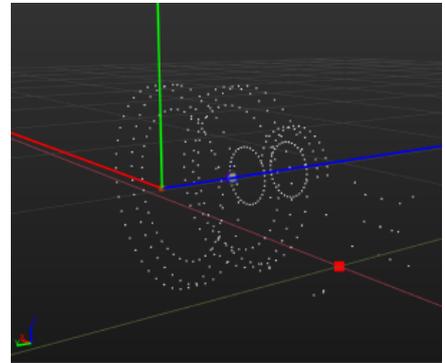
[ 중심점을 생성하기 ]를 클릭하여 Z축의 기준 중심점을 생성합니다.



오른쪽 파라미터 표시줄에서 **중심점 선택** 버튼을 선택하십시오.

Z 오른쪽의 [ 선택 ] 버튼을 클릭하십시오.

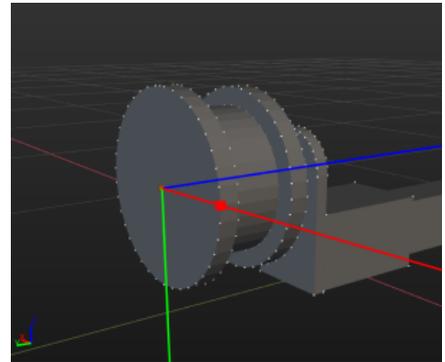
3D 편집 영역에서 Z축 기준 중심점을 클릭하면 Z축의 방향은 변경됩니다.



**중심점 선택** 버튼을 언체크하십시오.

X 오른쪽의 [ 선택 ] 버튼을 클릭하십시오.

3D 편집 영역에서 좌표 원점 평면에 있는 꼭짓점을 클릭하면 X축의 방향은 변경됩니다.



[ 확인 ] 버튼을 클릭하십시오.

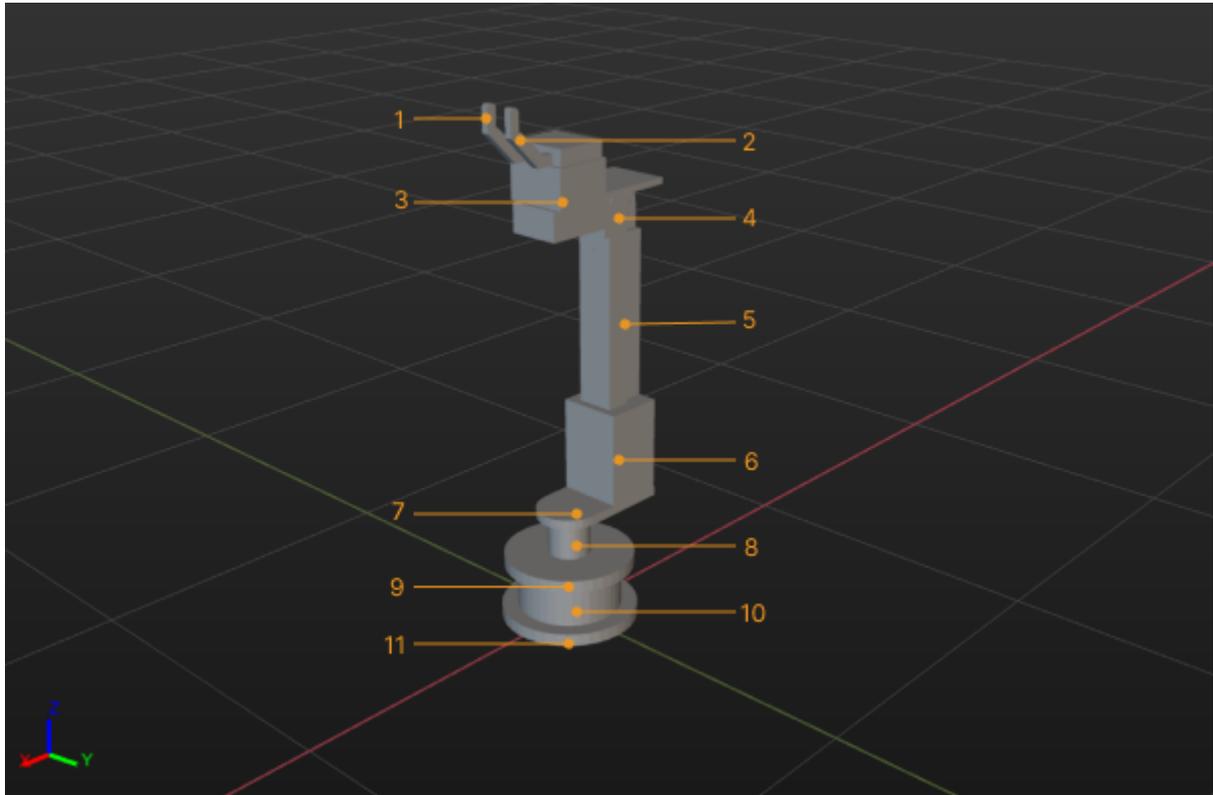
이때 모델 포즈가 변경되며 "모델 편집기의 세계 좌표계"가 "모델의 실제 설치 좌표계"와 겹치게 됩니다.



### 블록다면체 생성

Mech-Viz에는 말단장치 충돌 모델의 각 부분은 반드시 블록다면체이어야 합니다. 블록다면체는 기준 모델에 따라 구성되어야 하며 최종 모델은 블록다면체만 포함하고 기준 모델에 최대한 비슷한 것이 좋습니다.

블록다면체를 생성할 때 3D 선택 프레임으로 전체 모델을 선택하여 블록 껍질을 직접 생성하지 말고 말단장치의 구조적 특성에 따라 여러 부분으로 나눈 후 3D 선택 프레임을 사용하여 블록 껍질을 하나씩 선택하여 생성해야 합니다. 아래 그림에 표시된 모델을 예로 들어 대략 11개의 부분으로 나눌 수 있습니다.



라이선스 활성화의 프로세스는 다음과 같습니다:

1. “기준 모델” 표시줄에 모델 이름을 선택한 후 [3D 선택 도구]를 클릭합니다.
2. 일부 영역의 모든 꼭짓점을 선택할 수 있도록 3D 선택 프레임을 조정하고 [블록 껍질을 생성하기] 버튼을 클릭합니다.
3. 모델의 11개 부분에 대해 블록 껍질이 생성될 때까지 단계2를 반복하십시오.
4. 기준 모델 왼쪽의  아이콘을 클릭하면 기준 모델을 표시하지 않고 블록 모델을 볼 수 있습니다.



- 블록 껍질 또는 중심점은 모델의 꼭짓점을 기준으로 생성되기 때문에 선택할 때 처리할 부분의 “꼭짓점”이 모두 선택되어야 합니다.
- 3D 선택 프레임을 사용하여 해당 블록 껍질을 정확하게 생성할 수 없는 부분에 대해 [직육면체 또는 원기둥을 생성할 수 있습니다.](#)

### 프로젝트 파일을 저장하기

모델 재편집을 용이하게 하려면 다음 단계를 수행하여 모델을 m3d 포맷으로 저장하십시오.

1. **파일 > 저장** 버튼을 클릭합니다.
2. 저장 경로를 선택하고 파일 이름을 입력하십시오.

### 3. [ 저장 ] 버튼을 클릭하십시오.

#### 편집한 후의 모델을 도출하기

다음 단계를 수행하여 편집한 후의 모델을 obj 포맷으로 저장하십시오.

1. **파일 > 블록 모델을 도출하기** 버튼을 클릭하십시오.
2. 저장 경로를 선택하고 파일 이름을 입력하십시오.
3. [ 저장 ] 버튼을 클릭하십시오.

모델을 성공적으로 도출하면 **프로젝트 리소스 > 모델 라이브러리**에 해당 모델을 추가하고 **프로젝트 리소스 > 말단장치**에서 해당 단장치의 충돌 모델을 설정할 수 있습니다.

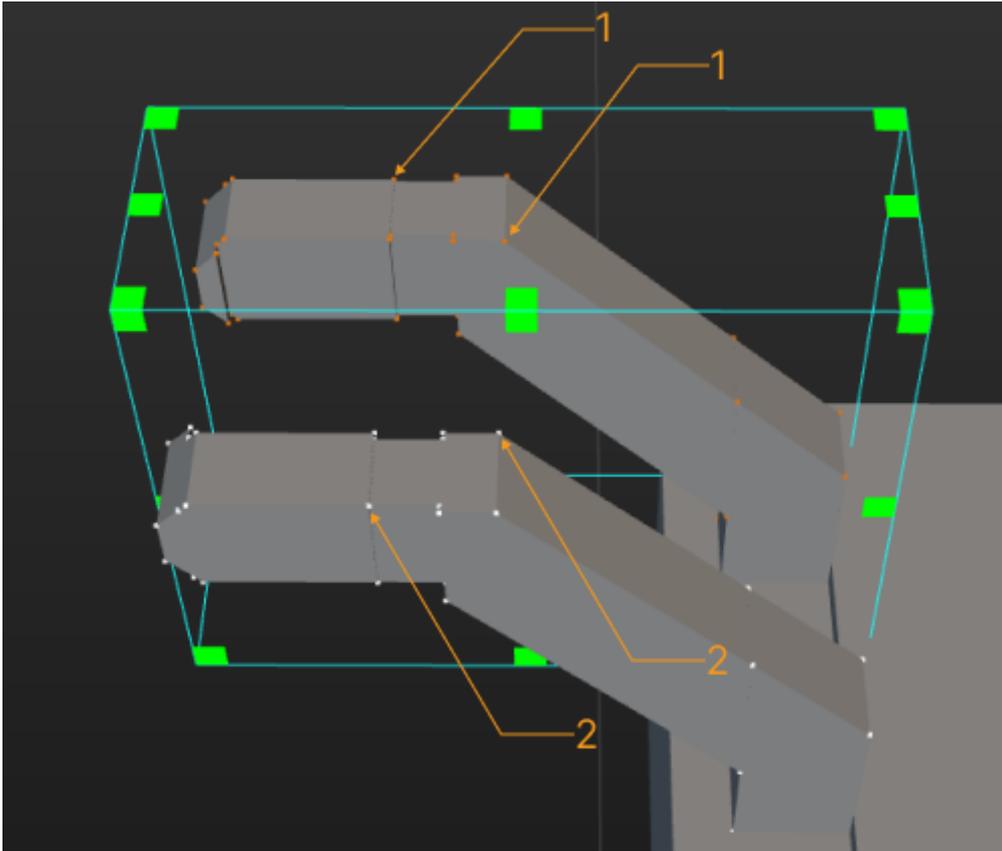
#### 3D 선택 프레임 사용 가이드



모델 이름이 선택된 상태에서  아이콘을 클릭하면 모델의 꼭짓점만 표시하고 모델을 표시하지 않습니다.

[ 3D 선택 도구 ]를 클릭하면 모델 꼭짓점을 선택하는 데 사용되는 3D 선택 프레임이 나타납니다. 3D 선택 프레임의 크기와 위치는 두 가지 방법으로 조정할 수 있습니다.

- 3D 편집 영역에서 다음과 같이 작업하십시오:
  - **Ctrl** 키를 누른 상태에서 마우스 왼쪽 버튼을 누르면서 3D 선택 도구의 꼭짓점을 드래그하여 프레임의 크기를 변경합니다.
  - **Ctrl** 키를 누른 상태에서 마우스 왼쪽 버튼을 누르면서 3D 선택 도구 표면을 드래그하여 프레임의 위치를 이동합니다.
- 오른쪽 파라미터 표시줄에서 다음과 같이 조정하십시오:
  - **치수**를 설정하여 선택 프레임의 크기를 수정합니다.
  - **중심, 회전**을 설정하여 선택 프레임의 포즈를 조정합니다.



1: 선택된 모델 꼭짓점. 2: 선택되지 않은 모델 꼭짓점

#### 직육면체나 원기둥을 만들기

3D 선택 도구를 사용하여 볼록 껍질을 만드는 것 외에도 직육면체 또는 원기둥을 만들어 모델의 일부를 덮을 수 있습니다.

직육면체 생성을 예로 들어 보겠습니다(원기둥의 생성 과정은 비슷합니다).

1. [ 직육면체를 만들기 ] 버튼을 클릭하십시오.
2. 팝업 창에 사용자가 정의한 물체의 이름을 입력하고 물체 치수 파라미터를 대략적으로 수정한 다음 [ 확인 ] 버튼을 클릭합니다.
3. 새로 추가된 직육면체는 3D 편집 영역의 좌표 원점에 위치합니다. **Ctrl** 키를 누른 상태에서 마우스 왼쪽 버튼으로 매니플레이터를 드래그하고 직육면체를 덮을 위치로 드래그합니다.
4. 직육면체 모델을 더블 클릭하여 볼록 물체 구성 창을 띄우고 처리할 구조를 덮도록 치수와 포즈를 정확하게 조정합니다.



### 시각화 모델 파일을 stl 포맷으로 전환하기

stp, step포맷의 모델 파일을 소프트웨어에 지원되는 stl 포맷으로 전환합니다.

1. 처리할 모델을 도입하려면 다음 단계 중 하나를 수행하십시오.
  - **파일 > 기준 모델을 도입하기** 버튼을 클릭한 후 파일 선택 창에서 기준 모델 파일을 선택하십시오.
  - 기준 모델 파일을 모델 편집기로 드래그하십시오.
2. **파일 > 기준 모델을 도출하기** 버튼을 클릭하십시오.
3. **모델 도출** 창에서 파일의 저장 경로를 선택하고 파일 이름을 입력한 후 [저장] 버튼을 클릭하십시오.

모델을 성공적으로 도출하면 **프로젝트 리소스 > 모델 라이브러리**에 해당 모델을 추가하고 **프로젝트 리소스 > 팔단장치**에서 해당 단장치의 시각화 모델을 설정하거나 **프로젝트 리소스 > 시나리오 물체** 패널에서 해당 시나리오 물체의 시각화 모델을 설정할 수 있습니다.

#### 5.6.1.1. 인터페이스 소개

이 부분에서는 모델 편집기의 인터페이스에 대해 소개하겠습니다.

##### 메뉴 바

##### 파일

옵션	설명	단축키
새로 만들기	새로운 편집 작업을 만들기	Ctrl + N
열기	m3d 포맷의 파일을 열기	Ctrl + O

옵션	설명	단축키
저장	편집 작업을 m3d 파일로 저장하기	Ctrl + S
저장 경로를 선택하기	m3d 파일로 편집 작업을 지정한 경로에 저장하기	Ctrl + Shift + S
기준 모델을 도입하기	기준으로 사용될 수 있는 모델을 모델 라이브러리에 도입하기	
기준 모델을 도출하기	기준 모델을 stl 파일로 도출하기	
블록 모델을 도출하기	블록 모델을 obj 파일로 도출하기	

### 편집

옵션	설명	단축키
취소	작업을 취소하기	Ctrl + Z
다시 하기	작업을 다시 하기	Ctrl + Y

### 도움

옵션	설명
사용자 안내서	모델 편집기의 사용자 안내서를 온라인으로 열기

### 툴 바

옵션	설명	단축키
3D 선택 도구	3D 선택 박스를 생성하기	
직육면체를 만들기	직육면체 하나를 생성하기	
원기둥을 만들기	원기둥 하나를 생성하기	
블록 껍질을 만들기	3D 선택 프레임의 꼭짓점을 기준으로 블록 껍질을 만들기	Shift + C
중심점을 생성하기	3D 선택 프레임의 꼭짓점을 기준으로 중심점을 만들기	
좌표계 설정	중심점이나 꼭짓점을 기준으로 좌표계를 만들기	

### 기준 모델

도입된 기준 모델 파일을 표시하는 데 사용됩니다.

- 모델 파일의 이름을 클릭하면 3D 편집 영역에서 모델의 꼭짓점을 표시할 것입니다.
- 모델 이름이 선택된 상태에서  아이콘을 클릭하면 모델의 꼭짓점만 표시하고 모델을 표시하지 않습니다.

- 모델 이름이 선택되지 않은 상태에서  아이콘을 클릭하면 모델과 모델 꼭짓점은 모두 표시하지 않습니다.

### 리소스

기본 기하학체, 볼록 껍질과 중심점을 표시하는 데 사용됩니다.

 아이콘을 클릭하면 해당 볼록 껍질과 중심점이 표시되지 않습니다.

마우스 오른쪽 버튼으로 기본적인 기하학체의 이름을 클릭하면 다음과 같은 옵션들이 나옵니다:

옵션	설명	단축키
복사하기	해당 기하학체를 복사하기	Ctrl + C
붙여넣기	해당 기하학체를 붙여넣기	Ctrl + V
삭제하기	해당 기하학체를 삭제하기	Delete
이름을 바꾸기	해당 기하학체의 이름을 재설정하기	

마우스 오른쪽 버튼으로 볼록 껍질의 이름을 클릭하면 다음과 같은 옵션들이 나옵니다:

옵션	설명	단축키
복사하기	해당 볼록 껍질을 복사하기	Ctrl + C
붙여넣기	해당 볼록 껍질을 붙여넣기	Ctrl + V
삭제하기	해당 볼록 껍질을 삭제하기	Delete
이름을 바꾸기	해당 볼록 껍질의 이름을 재설정하기	
생성 시의 3D 선택 도구를 표시하기	해당 볼록 껍질을 생성 시의 3D 선택 도구를 표시하기	

마우스 오른쪽 버튼으로 중심점의 이름을 클릭하면 다음과 같은 옵션들이 나옵니다:

옵션	설명	단축키
복사하기	해당 중심점을 복사하기	Ctrl + C
붙여넣기	해당 중심점을 붙여넣기	Ctrl + V
삭제하기	해당 중심점을 삭제하기	Delete
이름을 바꾸기	해당 중심점의 이름을 재설정하기	
생성 시의 3D 선택 도구를 표시하기	해당 중심점을 생성 시의 3D 선택 도구를 표시하기	
맨 위로 표시하기	이 옵션을 선택하면 모델 내부의 중심점을 볼 수 있음	

### 3D 편집 영역

3D 편집 영역에서는 모델을 표시하고 편집할 수 있으며 다음과 같이 시각을 전환할 수도 있습니다.

- 마우스 휠을 통해 3D 편집 영역의 크기를 조정할 수 있습니다.
- 마우스 왼쪽 버튼을 길게 누르고 원하는 방향으로 드래그하여 시각을 회전할 수 있습니다.
- 마우스 가운데 버튼을 길게 누르고 원하는 방향으로 드래그하여 시각을 평행이동할 수 있습니다.
- 마우스 오른쪽 버튼을 클릭하고 시각 전환 메뉴를 통해 시각을 전환할 수 있습니다.

#### 5.6.1.2. STEP/STP 파일의 유효성 반정

STEP/STP 파일이 다음 세 가지 사항을 충족하지 않으면 모델을 도입할 때 "모델을 로드하지 못했음"이라는 팝업 창이 표시됩니다.

- STEP/STP 파일은 ASCII 인코딩 파일입니다.
- STEP/STP 파일은 "ISO 10303-21" 표준을 기반으로 하며 파일의 첫 번째 줄에는 ISO 10303-21 로고가 있습니다.
- 표준 로고 외에 HEADER와 DATA 두 영역이 있으며 각각 파일 정보와 파일 데이터를 설명하는 데 사용됩니다.

또한 STEP/STP 파일을 열었을 때 글자가 깨져 보이면 파일이 암호화된 것일 수 있으므로 파일을 제공한 담당자에게 연락하여 해독해야 합니다.

#### 5.6.2. 일반적인 빨판 구성기

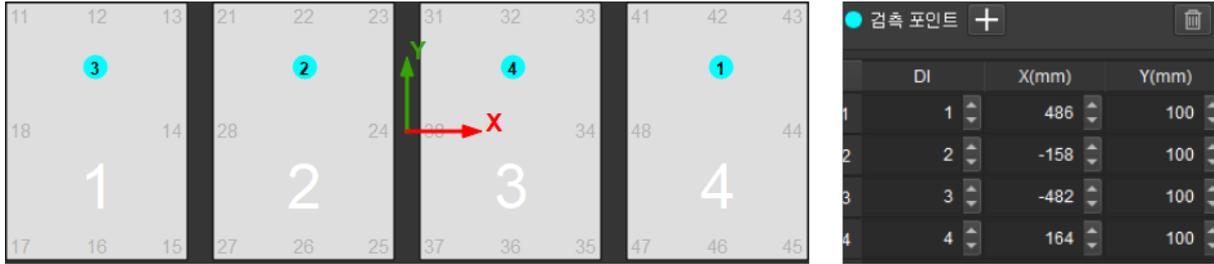
이 부분에서는 일반적인 빨판 구성기 및 관련 설정에 대해 소개하겠습니다.

##### 소개

빨판 구성기는 현재 직사각형, 단일 파티션, 다중 파티션, 단일 행, 파티션 크기가 서로 같은 빨판 구성만 지원합니다.

##### 빨판 구성 프로세스

1. 빨판 구조 구성
  - a. 실제 사용하는 빨판의 블록 수에 따라 **블록 수** 파라미터를 설정하십시오. 여러 개 블록을 하나의 DO 신호로 컨트롤하면 한 완전한 블록으로 간주되어 하나의 큰 블록만 만들면 됩니다.
  - b. 빨판 블록 사이의 실제 거리에 따라 **간격** 파라미터를 설정하십시오.
  - c. 빨판 블록의 실제 치수에 따라 **블록 길이/너비** 파라미터를 설정하십시오.
  - d. **완충구역 두께** 파라미터를 충돌 감지에 사용되는 스펀지층 두께, 즉 TCP 평면 아래의 스펀지층의 두께로 설정하십시오.
2. 말단장치 구성 창에서 설정한 TCP에 따라 **각도** 파라미터를 설정하십시오.
3. 실제 상황에 따라 **DO/DI 활성화값** 파라미터를 설정하십시오.
4. DI 감지 포인트 구성
  - a. **[+]** 버튼을 클릭하면 DI 감지 포인트가 새로 추가됩니다. 감지 포인트의 DI값은 기본적으로 0이고 빨판 가운데에 위치합니다.
  - b. **X/Y**파라미터를 수정하거나 감지 포인트를 직접 드래그하여 위치를 변경합니다.
  - c. **DI** 파라미터를 수정하여 감지 포인트의 번호를 변경합니다.



감지 포인트를 삭제하려면 감지 포인트를 먼저 선택한 다음에 를 클릭하십시오.

1. 실제로 빨판의 각 파티션을 컨트롤하는 포트 번호에 따라 해당한 **DO 값**을 입력하십시오.
2. **파일 > 저장** 버튼을 클릭하여 빨판 구성을 해당 말단장치의 구성으로 저장하십시오.

### 빨판 구성 파일을 도출하기

**파일 > 도출** 버튼을 클릭하면 구성 파일을 지정된 경로에 저장할 수 있습니다.

### 빨판 구성 파일을 도입하기

**파일 > 도입** 버튼을 클릭하면 빨판 구성 파일을 도입할 수 있습니다.

## 5.6.3. 어레이 그리퍼 구성기

이 부분에서는 어레이 그리퍼 구성기 및 관련 설정에 대해 소개하겠습니다.

### 어레이 그리퍼 구성 프로세스

1. 실제로 사용하는 어레이 그리퍼의 말단 수와 간격에 따라 **총수** 및 **간격** 파라미터를 수정하십시오.
2. 실제 상황에 따라 **DO 활성화값** 파라미터를 설정하십시오.
3. 말단장치 구성 창에서 설정한 TCP에 따라 **어레이 각도** 파라미터를 설정하십시오.
4. **파일 > 저장** 버튼을 클릭하여 어레이 그리퍼의 구성을 해당 말단장치의 구성으로 저장하십시오.

### 어레이 그리퍼 구성 파일을 도출하기

**파일 > 도출** 버튼을 클릭하면 구성 파일을 지정된 경로에 저장할 수 있습니다.

### 어레이 그리퍼 구성 파일을 도입하기

**파일 > 도입** 버튼을 클릭하면 어레이 그리퍼 구성 파일을 도입할 수 있습니다.

## 5.6.4. 파렛트 패턴 편집기

파렛트 패턴 편집기는 스텝"자체 정의한 파렛트 패턴"에서 사용자가 정의한 파렛트 패턴을 편집하는 데 사용됩니다. 스텝"자체 정의한 파렛트 패턴"의 파라미터 표시줄에서 **[ 파렛트 패턴 편집기 ]** 버튼을 클릭하면 이 도구를 사용할 수 있습니다.

### 사전 준비

파렛트 패턴을 편집하기 전에 실제 상황에 따라 다음 사항을 확인하십시오:

- 파렛트 길이와 너비

- 파렛트의 층수
- 파렛트의 층 높이
- 파렛트 패턴의 층 레이아웃
- 상자 길이와 너비

## 파렛트 패턴을 편집하기

파렛트 패턴을 편집하려면 **층 레이아웃**과 **파렛트 층**의 관련 설정을 완료해야 합니다.



**파일 > 파일에서 로드하기**를 클릭하면 이전에 편집한 파렛트 패턴 파일을 로드할 수 있습니다.

### 층 레이아웃 설정

파렛트 길이, 너비, 레이아웃 수와 물체의 위치를 설정해야 합니다.

1. **파렛트 길이**와 **파렛트 너비** 파라미터를 설정합니다. 실제 파렛트의 길이와 너비에 따라 입력하십시오.
2. **[위치 추가]** 버튼을 클릭하고 마우스를 상자 위로 이동하며 마우스 왼쪽 버튼을 길게 누르면 위치를 변경할 수 있습니다.
3. **위치 설정** 패널에서 상자의 치수, 회전 각도, 입사각을 설정할 수 있습니다.
4. 단계 2 및 3을 반복하여 현재 층에 대한 레이아웃 설정을 완료합니다.
5. 여러 층에 대해 레이아웃을 설정하려면 **[층 레이아웃 추가]** 버튼을 클릭하면 됩니다.

### 파렛트 층 설정

층 수, 층 높이, 각 층의 레이아웃을 설정해야 합니다.

1. 실제 수요에 따라 **[층 추가]** 버튼을 클릭하여 층을 추가하십시오.
2. 한 층을 선택하고 **층 높이** 파라미터를 설정하여 해당 층의 높이를 수정합니다.
3. **레이아웃 바꾸기** 버튼을 클릭하면 현재 선택된 층의 레이아웃을 바꿀 수 있습니다.
4. 단계 2 및 3을 반복하여 모든 층에 대한 파라미터 설정을 완료합니다.

### 저장 및 사용

**파일 > 저장**(Ctrl + S) 버튼을 클릭하여 파렛트 패턴의 변경 사항을 저장하며 소프트웨어의 3D 시뮬레이션 공간에서 편집한 후의 파렛트 패턴을 표시합니다.

**파일 > 파일로 저장하기** 버튼을 클릭하여 현재 설정한 내용을 **.json** 파일로 저장합니다. 스텝 “자체 정의한 파렛트 패턴”의 파라미터 표시줄에서 **[동적 로딩]** 버튼을 클릭한 후 “파렛트 파일 경로”에서 저장한 파일을 선택하면 파렛트 패턴을 로드할 수 있습니다.

## 5.7. 부록

### 5.7.1. 로봇 모델의 제작 및 도입

소프트웨어의 빌트인 로봇 라이브러리와 온라인 로봇 라이브러리에서 필요한 로봇을 찾을 수 없는 경우, 이 부분 내용을 참조하여 로봇 모델을 제작하고 소프트웨어로 도입할 수 있습니다. 구체적으로 다음과 같습니다:

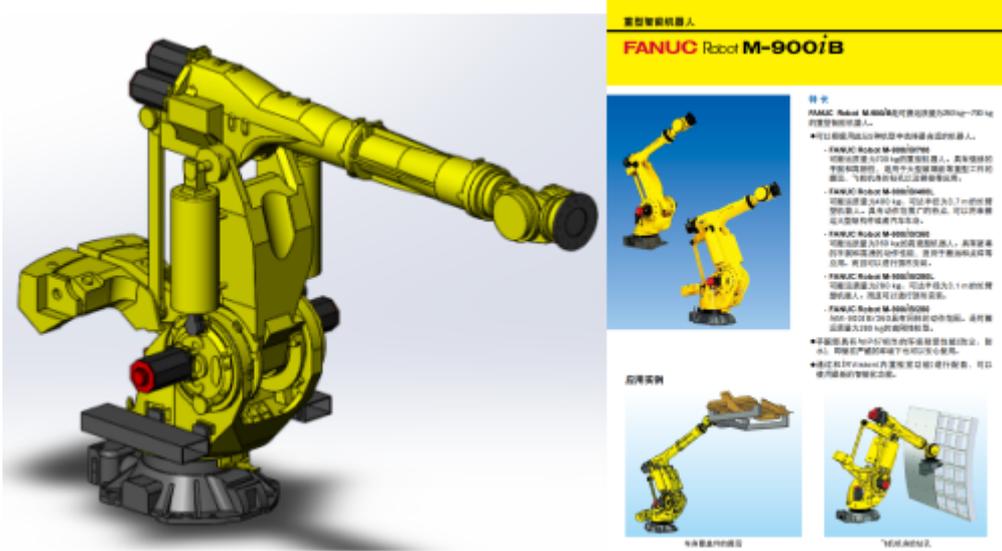
FANUC M-900iB/400L 로봇을 예시로 설명하도록 하겠습니다.

## 로봇 CAD 파일과 설명서를 준비하기

로봇 모델을 제작하기 전에 로봇의 CAD 모델과 DH 파라미터, 각 관절 제한 등이 표시된 설명서 파일을 준비해야 합니다.

사용자는 특정 로봇의 공식 웹사이트에서 해당 모델의 관련 파일을 검색하고 다운로드 할 수 있습니다.

아래 그림은 FANUC M-900iB/400L 로봇의 CAD 모델 및 설명서입니다.



일부 로봇 웹사이트에서는 x\_t 형식의 모델을 제공하고 있으며, STEP 형식에 비해 x\_t 형식의 모델이 재구성될 때 성공률이 더 높고 속도가 더 빠르므로 x\_t 형식의 모델을 먼저 사용하는 것이 좋습니다.

## [robot]\_algo.json 파라미터 파일을 작성하기

### [robot]\_algo.json 템플릿 파일

[robot]\_algo.json의 템플릿 파일Mech-Viz은 설치 경로 `C:/Users/Administrator/AppData/Roaming/Mmind/robot/profile\_example.json`에 위치하며 이 파일을 기반으로 새로운 로봇의 \_algo.json 파일을 작성할 수 있습니다.



[robot]\_algo.json 파일 속의 각 파라미터의 의미는 [robot\\_profile.json 파일 파라미터 설명](#) 내용을 참조하십시오.

### 로봇 구성(algo\_type)을 확인하기

로봇은 다양한 구성으로 나뉘지며 로봇의 구성이 다르면 좌표계와 DH 파라미터에 관한 정의도 다릅니다.

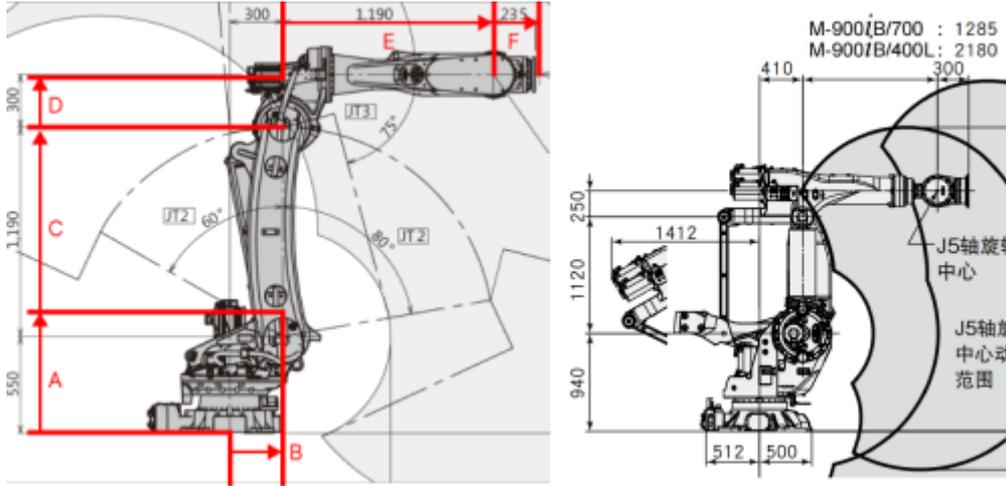
상세한 정보는 [로봇 구성](#) 내용을 참조하십시오.

FANUC M-900iB/400L 로봇은 일반적인 6축 볼형 손목 산업용 로봇으로 SphericalWrist\_SixAxis 구성에 속합니다.

### DH 파라미터(dh, dhPassive)를 확인하기

[로봇 구성](#) 부분에서 로봇의 구성 유형을 참조하여 해당 로봇의 구성 다이어그램을 찾아 로봇 설명서에 따라 [robot]\_algo.json 속의 dh 값을 확인하십시오.

FANUC M-900iB/400L 로봇의 dh 파라미터는 다음과 같습니다: a=0.940, b=0.410, c=1.120, d=0.250, e=2.180, f=0.300.



왼쪽은 SphericalWrist\_SixAxis 로봇 구성 다이어그램이고 오른쪽은 FANUC M-900iB/400L 로봇의 치수가 표시된 다이어그램입니다.

### 로봇 각 축의 리미트(minlimits, maxlimits)를 확인하기

보통 로봇의 각 축의 이동 범위는 로봇 매뉴얼에서 알 수 있지만 FANUC 로봇은 특수하므로 FANUC 로봇 시뮬레이션 소프트웨어 RoboGuide에서 해당 내용을 찾아야 합니다.



### 기타 파라미터를 확인하기

일반적인 로봇 브랜드의 경우 추가된 다른 로봇의 기존 데이터를 참조할 수 있습니다. 참조할 수 있는 기존 데이터가 없는 경우, 작업 현장에서 일하는 엔지니어가 실제 로봇에 따라 다음과 같은 파라미터를 확인해야 합니다: mastering\_joints, axis\_flip, base\_z\_offset, axis\_flip, mastering\_joints.

- json 파라미터 파일 속의 서로 다른 데이터 항목을 구분하기 위해 영어 문장 부호(예: 쉼표)를 사용해야 합니다.

정확성을 보장하기 위해 로봇 모델을 제작한 후

- mastering\_joints, axis\_flip, base\_z\_offset 속성을 **로봇 시뮬레이션 소프트웨어**와 대조해야 합니다.
- axis\_flip 속성은 minlimits, maxlimits와 결합되며 필요한 경우 한 관절 각의 상한 및 하한의 부호를 반전하고 위치를 바꿔야 합니다.
- mastering\_joints 속성의 마지막 J6은 종종 무시되며 이 속성은 **오일러 각**에 영향을 미치므로 정확성을 점검하고 보장해야 합니다.

### [robot]\_profile.json 파라미터 파일을 작성하기

[robot]\_profile.json 로봇의 템플릿 파일은 Mech-Viz의 설치 경로 **resource/robot/profile/example.json**에 위치하 사용자는 이 파일을 기반으로 새로운 로봇의 \_profile.json 파일을 작성할 수 있습니다.

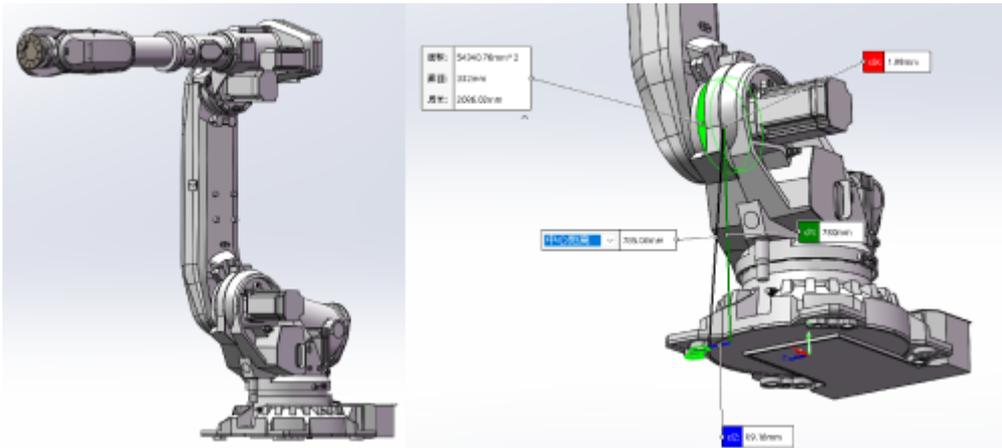


- [robot]\_profile.json 파일 속의 각 파라미터의 의미는 [robot\\_profile.json 파일 파라미터 설명](#) 내용을 참조하십시오.
- 이 파일은 필수적이지 않습니다.

### Solidworks를 사용하여 완전한 로봇 STL 모델을 만들기

#### 모델을 SolidWorks로 도입하기

아래 그림과 같이 SolidWorks를 사용하여 미리 준비된 로봇 CAD 모델을 엽니다.



위 그림에서 왼쪽은 3D 소프트웨어에 표시된 로봇 모델이고 오른쪽은 조립할 때의 참조 그림입니다.

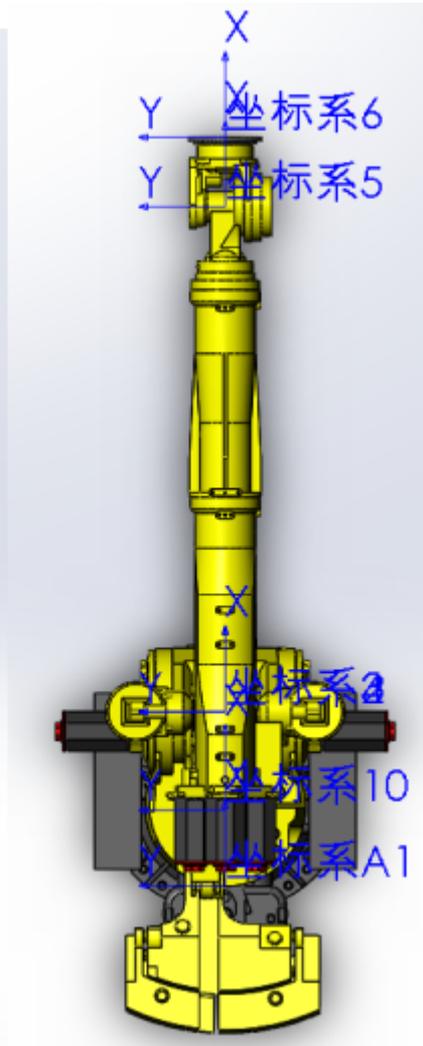
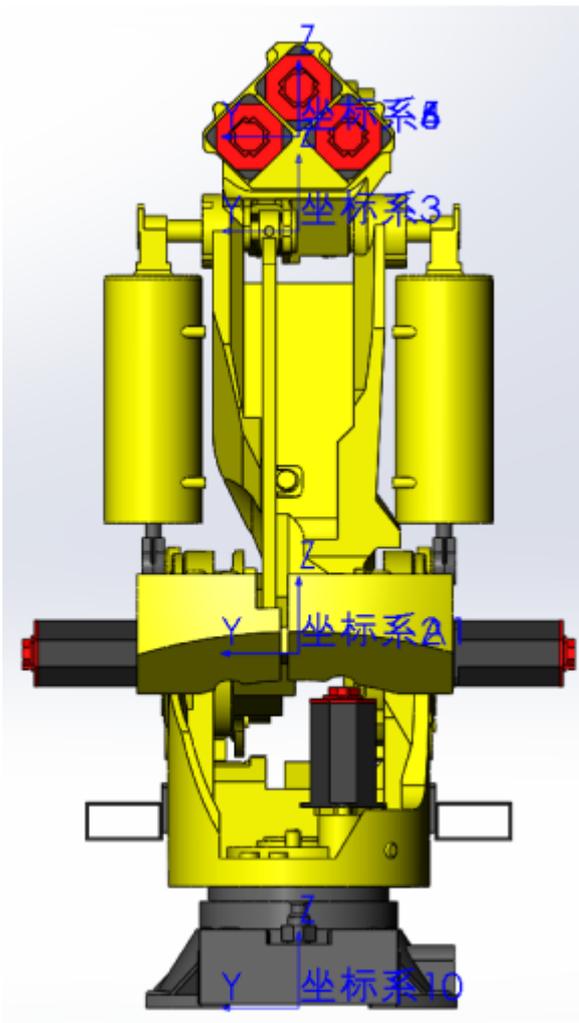
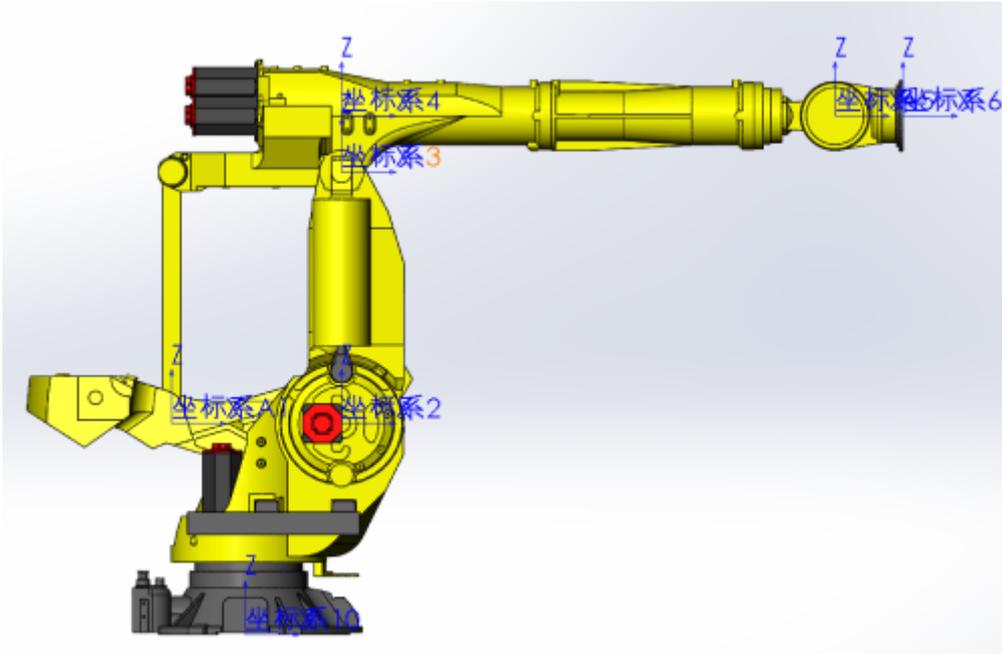


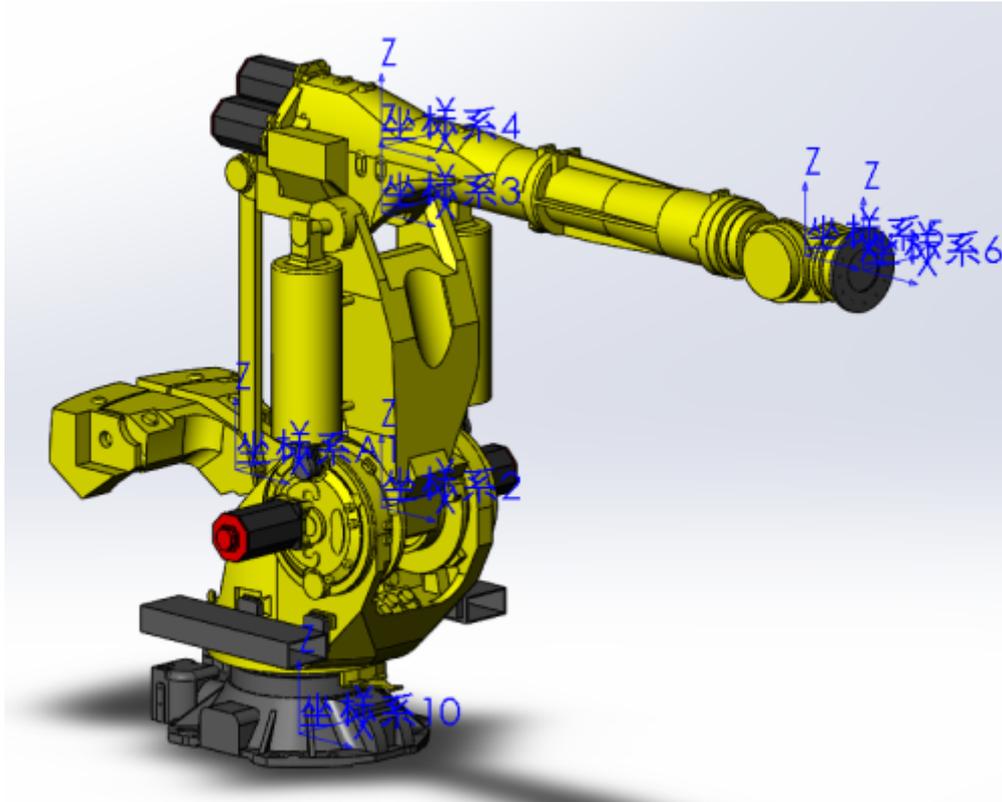
- 로봇 모델을 다운로드할 때, 로봇 공식 홈페이지에서 조립된 완전한 모델을 다운로드해 보십시오.
- 부품만 다운받을 수 있는 경우 부품들을 자체적으로 조립해야 하며 dh 파라미터에 따라 각 관절을 확인해야 합니다.
- 모델의 도입 속도를 높이기 위해 일부 모델 디테일을 적당히 삭제할 수 있으며 충돌 감지에 영향을 미치는 모델 수조만 보류합니다.

#### 좌표계를 구축하기

**로봇 구성** 내용을 참조하여 각 축의 좌표계를 구축합니다.

FANUC M-900iB/400L의 구성은 SphericalWrist\_SixAxis이며, 로봇의 자세는 다음과 같아야 합니다: 1축은 초기 위치로 복원되고 2축은 수직으로 위쪽을 가리키며 3,4축은 수평으로 앞쪽을 가리키고 5축은 아래가 아닌 앞쪽 방향을 가리킵니다. 로봇 자세를 확인한 후 모든 부품을 고정시키는 것을 권장합니다.

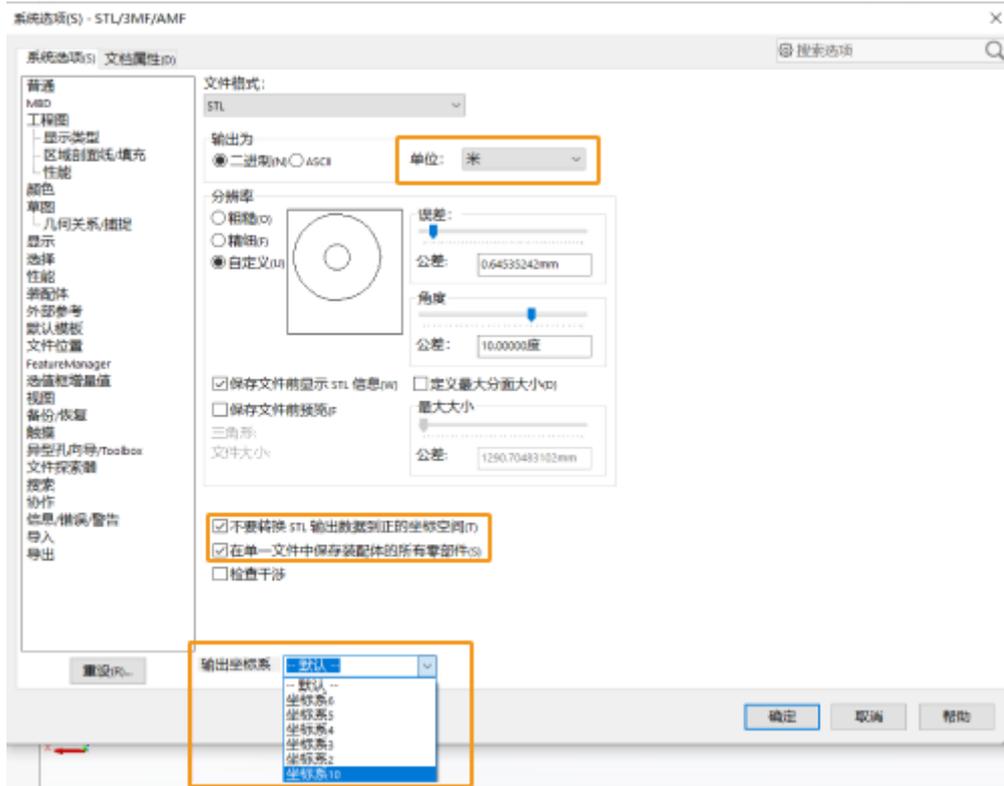




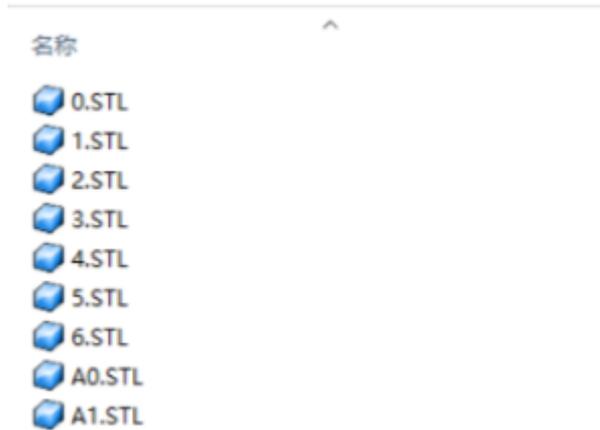
#### 로봇 각 관절의 STL 모델을 도출하기

로봇 모델을 도출할 때 장착체에서 다른 구성 요소의 표시를 숨겨 개별 도출하기를 용이하게 합니다.

모델을 도출할 때 파일을 stl 파일로 저장하고 출력은 바이너리, 단위는 미터, 파일의 네이밍 규칙과 부위별로 선택해야 하는 좌표계는 [로봇 구성](#) 내용을 참조하십시오. 도출할 때의 다른 설정 사항은 아래 그림과 같습니다.



FANUC M-900iB/400L 로봇이 모두 저장된 후의 완전한 STL 모델 파일은 아래 그림과 같습니다.

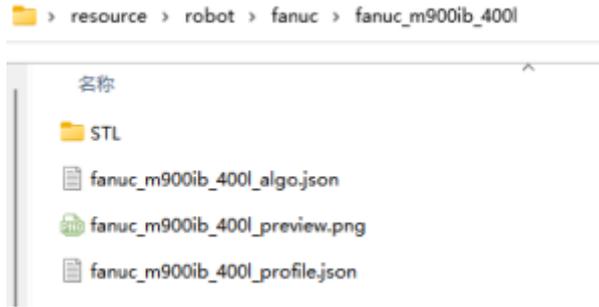


이로써 로봇 모델의 제작 과정은 끝났습니다.

### 모델 파일을 Mech-Viz 소프트웨어로 도입하기

Mech-Viz 소프트웨어가 로봇 라이브러리를 저장하는 경로는 설치 경로 아래의 **resource/robot** 폴더입니다. 동일한 브랜드의 로봇은 로봇 브랜드의 이름으로 명명된 동일한 폴더에 저장됩니다. 각 로봇의 모든 파일은 브랜드 폴더 아래에 있는 로봇 모델에 의해 명명된 폴더에 있습니다.

예를 들어, FANUC M900iB/400L 로봇의 모델 파일은 **resource/robot/fanuc\_m900ib\_400l** 폴더에 있으며 폴더의 구조는 아래 그림과 같습니다.



- STL: 모델 파일
- fanuc\_m900ib\_400l\_algo.json: 로봇 운동학 파라미터 파일
- fanuc\_m900ib\_400l\_preview.png: 로봇 이미지
- fanuc\_m900ib\_400l\_profile.json: 로봇 관련 설명 파라미터 파일



- 로봇 폴더 이름은 모두 소문자여야 합니다.
- 로봇 모델 파일은 해당 폴더에 위치하며 포맷은 STL, DAE, OBJ일 수 있습니다.
- STL 모델은 시각화 또는 충돌 모델로 사용될 수 있습니다.
- DAE 모델은 시각화 모델로만 사용될 수 있습니다.
- OBJ 모델은 충돌 모델로만 사용될 수 있습니다.
- 완전한 로봇 모델에는 적어도 하나의 시각화 모델로 사용될 수 있는 모델과 하나의 충돌 모델로 사용될 수 있는 모델이 필요합니다.

모든 파일을 해당 경로에 넣으면 로봇 모델 도입 과정이 완료됩니다. 도입된 로봇 모델은 Mech-Viz 소프트웨어의 로봇 라이브러리에서 선택될 수 있습니다.

#### 5.7.1.1. 로봇 구성

축의 수		6축	6축
구성	구성 명칭	UR_UR5_Like	SphericalWrist_SixAxis
	구성에 관한 설명	6축 협업 로봇	일반 산업용 6축 구형 손목 로봇

축의 수		6축	6축
로봇 구성 다이아그 램	DH 파라미터 그림		
	좌표계 각 축 이름		
커넥팅 로드와 좌표계의 대응 관계		0: 좌표계0. 1: 좌표계10. 2: 좌표계2. 3: 좌표계3. 4: 좌표계4. 5: 좌표계5. 6: 좌표계6.	

축의 수	6축	6축
[robot]_algo 예시	<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 파일의 파라미터 설명

[robot]\_algo.json은 로봇 구성 파일입니다. 로봇의 다양한 파라미터의 정보를 기록하고 로봇의 분류, 각 연결봉의 DH 파라미터, 각 관절 운동의 상한 및 하한, 각 관절의 시작 위치 및 회전 방향 등 내용을 결정합니다.

다음은 [robot]\_algo.json 파일의 일부 코드입니다.

```
{
  "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
}
```

<b>algo_type</b>	로봇 구성. Mech-Viz에서 로봇은 13 가지 유형으로 나뉘지며 상세한 정보는 <a href="#">로봇 구성</a> 내용을 참조하십시오.
<b>robot_type</b>	로봇 이름. 로봇 모델 폴더 이름과 일치해야 합니다. 그렇지 않으면 로봇 모델을 찾을 수 없습니다.
<b>dh</b>	로봇의 능동관절의 DH 파라미터.

<b>minlimits, maxlimits</b>	<p>minlimits 및 maxlimits는 각각 로봇의 각 축의 최대 및 최소 이동 범위입니다.</p> <p>현재 FANUC과 Nachi를 제외하고 다른 로봇의 각 축의 정확한 이동 범위는 로봇 매뉴얼에서 찾을 수 있습니다.</p> <p>모든 관절에는 상한 및 하한 두 개의 파라미터가 포함됩니다. 이 파라미터는 Axis_flip와 결합되어 있으며 특정 상황에서 상한 및 하한 파라미터를 서로 바꿔야 하며 부호를 반전해야 합니다.</p>
<b>mastering_joints</b>	<p>mastering_joints 파라미터는 로봇 각 축이 0일 때의 위치를 결정합니다. 모든 파라미터가 0일 경우, 로봇 각 축의 0 위치(초기 위치)는 모델링 시의 자세와 일치합니다. KUKA와 같은 일부 로봇의 경우, J2 및 J3의 값이 0일 때 Mech-Viz 소프트웨어의 기본적인 자세가 아니므로, J2 및 J3 mastering_joints 파라미터를 조정해야 합니다.</p>
<b>axis_flip</b>	<p>axis_flip 파라미터는 각 축의 회전 방향을 결정하며, 로봇 모델이 완성된 후에는 Mech-Viz 및 로봇 시뮬레이션 소프트웨어 또는 실제 로봇과 대조하여 각 축의 회전 방향이 일치하는지 확인해야 합니다. 일치하지 않는 경우 이 파라미터를 조정해야 합니다.</p> <p>이 파라미터는 로봇 관절 각도 상한 및 하한 minlimits, maxlimits 및 동적 리미트 link3_dynamic_limits, link4_dynamic_limits 파라미터에 영향을 줍니다.</p>

새로 만든 로봇을 사용하는 경우 실제 로봇에서 다음 파라미터를 확인해야 합니다:

<b>axis_flip</b>	<p>Mech-Viz에서 로봇 축의 회전 방향이 실제 로봇과 일치하는지를 확인합니다.</p>
<b>dh</b>	<p>다음 두 가지 방법을 사용하여 Mech-Viz에서 시뮬레이션된 로봇과 실제 로봇의 포즈 데이터가 일치하는지 비교합니다.</p> <ul style="list-style-type: none"> <li>● 관절 각도를 동기화하여 TCP 포즈와 비교하기</li> <li>● TCP 포즈를 동기화하여 관절 각도와 비교하기</li> </ul> <p>실제 로봇의 DH 파라미터가 파라미터 파일 속의 이론값에 가까울수록 로봇의 정밀도가 높아집니다. 일반적으로 1mm 미만의 오류가 허용됩니다.</p>
<b>mastering_joints</b>	<p>Mech-Viz에서 시뮬레이션된 로봇의 포즈 데이터가 실제 로봇의 포즈 데이터와 일치하는지 비교하고 확인합니다.</p> <p>검사하는 동안 축 1, 4, 6에 특별한 주의를 기울여야 합니다.</p>
<b>min_limits/max_limit</b>	<p>Mech-Viz에서 시뮬레이션된 로봇의 포즈 데이터가 실제 로봇의 포즈 데이터와 일치하는지 비교하고 확인합니다.</p> <p>소프트 리미트는 실제 로봇의 하드 리미트계보다 크면 안됩니다.</p>

### 5.7.1.3. [robot]\_profile.json 파일의 파라미터 설명

[robot]\_profile.json 파일에는 로봇의 소프트웨어 내의 초기 자세, 하중, 도달범위, 축 수 등 일부 로봇의 기본 정보를 기록합니다.

다음은 [robot]\_profile.json 파일의 일부 코드입니다.

```
{
  "home_jps": [0,90,0,0,90,0],
  "max_tcp_vel": 7,
  "max_joint_vel": 500,
  "reach": 2.55,
  "payload": 40,
  "axes": 6
}
```

<b>home_jps</b>	로봇이 Mech-Viz에서의 기본적인 자세를 결정합니다. 단위는 *도*입니다.
<b>max_tcp_vel</b>	최대 TCP 속도, 단위는 *m/s*이며 싱글래리티 필터링에 사용됩니다.
<b>max_joint_vel</b>	최대 관절 속도, 단위는 *도/s*이며 싱글래리티 필터링에 사용됩니다.
<b>reach</b>	로봇의 도달 가능한 범위, 단위는 *m*이며 로봇 라이브러리에 표시됩니다.
<b>payload</b>	로봇 부하능력, 단위는 *kg*이며 로봇 라이브러리에 표시됩니다.
<b>axes</b>	로봇 축수, 로봇 라이브러리에 표시됩니다.

## 5.7.2. 기타

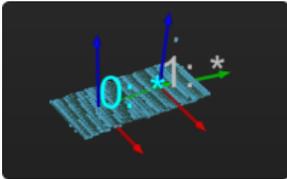
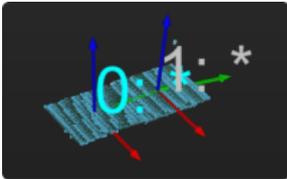
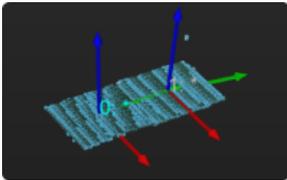
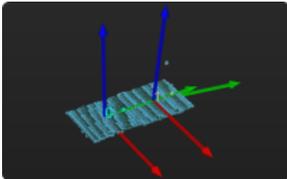
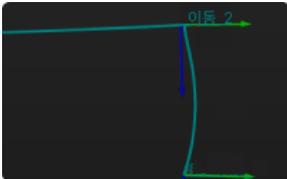
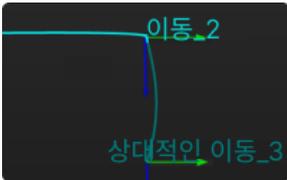
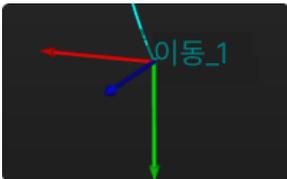
### 5.7.2.1. 설정 - 옵션

옵션 창에서 다음과 같은 내용을 조정할 수 있습니다.

- 일반 설정
    - 리소스 경로를 자체적으로 정의할 수 있습니다.
    - 로그와 관련된 파라미터를 설정할 수 있습니다.
    - 3D 시뮬레이션 화면의 파라미터를 설정할 수 있습니다.
    - 메인 인터페이스를 닫을 수 있습니다.
    - 소프트웨어의 시스템 언더, 길이 단위, 각도 단위를 설정할 수 있습니다.
  - 기본 설정
    - 실제 로봇의 중지 기능을 활성화할 수 있습니다.
    - 3D 시뮬레이션 공간의 바닥 유형을 설정할 수 있습니다.
- 💡 \* 표기를 갖춘 설정은 소프트웨어 재부팅한 뒤 적용될 것입니다.

### 5.7.2.2. 표시 - 디스플레이 설정

디스플레이와 관련된 사항을 설정할 수 있으며 프로젝트 디버깅에 도움이 됩니다.

파라미터	값	예시 그림
비전 포즈 레이블의 크기	0.05	
	0.07	
비전 포즈 드래거의 크기	0.05	
	0.07	
경로 텍스트 크기	0.02	
	0.04	
경로 드래거 크기	0.05	
	0.1	

### 5.7.2.3. 툴 - 비전 기록 설정

저장한 비전 기록을 사용하여 시뮬레이션 과정 또는 실제 생산 과정에서 나타나는 문제를 재현할 수 있습니다.

Mech-Viz가 실행되는 과정에서 문제가 발생하면 저장한 비전 기록을 통해 대응한 Mech-Vision 프로젝트를 실행하지 않고 Mech-Viz 프로젝트를 직접 실행해도 문제가 발생하는 원인을 찾아 문제를 해결할 수 있습니다.

#### 비전 기록 저장

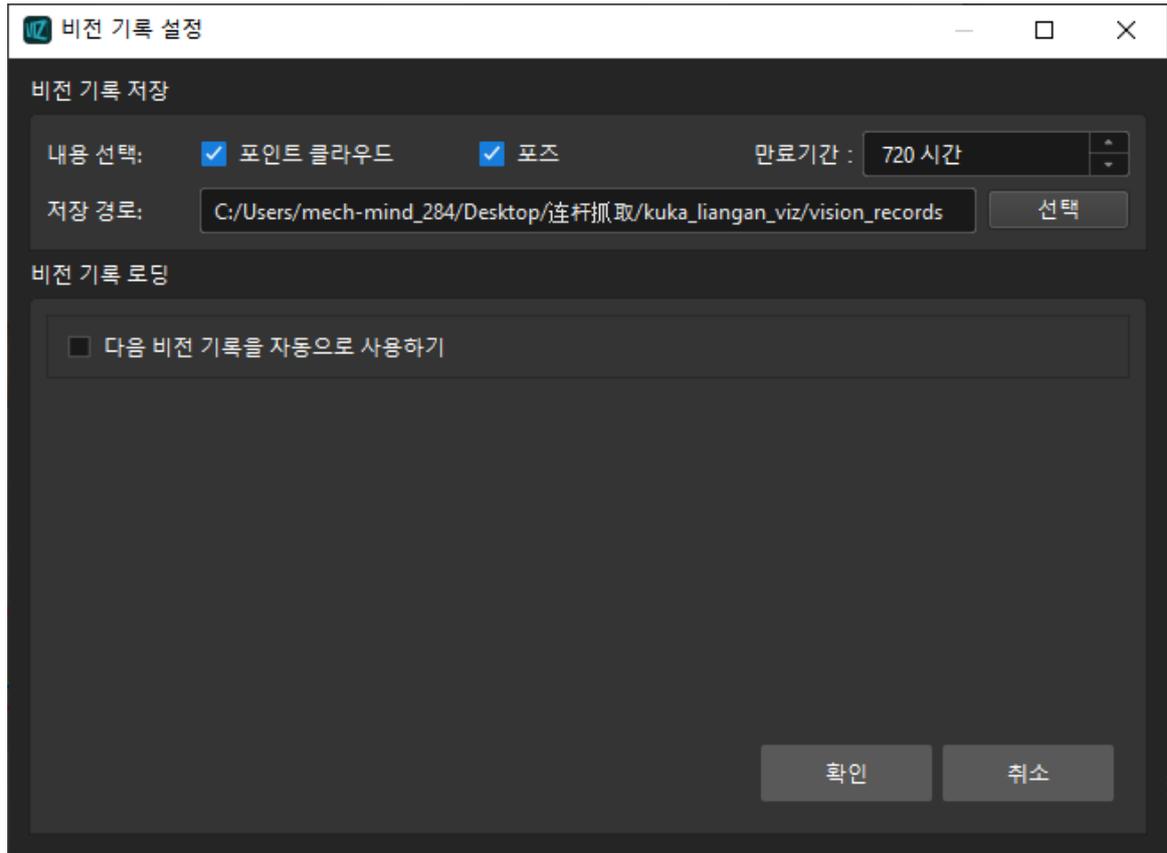
1. Mech-Viz메뉴 바의 **툴 > 비전 기록 설정**을 클릭하여 비전 기록을 설정하는 창을 엽니다.
2. 비전 기록에는 **포즈**만 저장할 수도 있고 동시에 **포즈**와 **포인트 클라우드**정보를 함께 저장할 수도 있습니다.
  - 비전 포즈만 저장하려면 **포즈**만 선택하십시오.
  - 비전 포즈와 포인트 클라우드를 모두 저장하려면 **포즈**를 선택한 후 **포인트 클라우드**를 선택하십시오.
3. 실제 수요에 따라 **만료기간**을 수정할 수 있으며 설정된 시간 제한을 초과하면 저장된 비전 기록은 삭제될 것입니다.
4. 실제 수요에 따라 **저장 경로**를 선택할 수 있으며 [ **선택** ] 버튼을 클릭하면 구체적인 저장 경로를 설정할 수 있습니다. 주의해야 할 것은 vision\_records 폴더만 선택 가능합니다.
5. [ **확인** ] 버튼을 클릭하십시오.
6. 프로젝트를 시뮬레이션하거나 실제로 실행하십시오.



- “포즈”와 “포인트 클라우드” 정보는 \_vision\_records/와 대응하는 Mech-Vision 프로젝트와 이름이 동일한 폴더/날짜에 따라 생성된 폴더\_에 저장합니다.
- 포인트 클라우드가 있는 시나리오에서 이미지를 한번 캡처할 때마다 포인트 클라우드 파일이 저장됩니다. 하지만 포인트 클라우드의 파일 용량이 비교적 커서 하드 디스크 공간을 많이 차지하므로 테스트 단계에만 **포인트 클라우드**를 선택하는 것을 권장하며 안정적인 실행 단계에서 선택하지 말아야 합니다. 비전 기록의 재현을 위해 vision\_records 폴더 뿐만 아니라 해당 Mech-Viz 프로젝트 파일도 함께 보류해야 합니다.

#### 비전 기록 재현

1. 해당 Mech-Viz 프로젝트, 즉 실행할 때 문제가 발생하고 비전 기록을 저장한 Mech-Viz 프로젝트를 열어 주십시오.
2. Mech-Vision이 시작되지 않았거나 Mech-Vision 프로젝트가 Mech-Center에 등록되었는지를 확인해야 합니다. 그렇지 않으면 소프트웨어는 저장된 비전 기록을 사용하는 대신 우선 Mech-Vision 프로젝트를 호출할 것입니다.
3. Mech-Viz 프로젝트의 비전 기록 파일을 소프트웨어 루트 디렉터리의 vision\_records 폴더에 저장합니다(**파일 > 탐색기에서 실행 파일을 열기** 버튼을 순서대로 클릭하여 소프트웨어의 루트 디렉터리를 빠르게 열 수 있음).
4. 툴 에 있는 **저장한 비전 기록을 사용하기(비전 서비스가 등록되지 않을 때만)**기능을 선택하십시오.
5. [ **시뮬레이션** ] 버튼을 클릭하면 소프트웨어는 vision\_records 폴더에 저장된 비전 기록이 있는지를 확인하고 **비전 기록 설정** 창이 팝업됩니다.



6. 재현할 비전 기록을 선택하여 [ 확인 ] 버튼을 클릭하면 현재 기록을 자동으로 재현할 것입니다.

**다음 비전 기록을 자동으로 사용하기** 버튼을 선택하지 않으면 선택한 비전 기록을 반복적으로 사용할 것입니다. 이 옵션을 선택하면 선택한 비전 기록을 재현한 후 자동으로 다음 기록으로 넘어갈 것이고 마지막 기록까지 재현한 후 재현 과정이 종료됩니다.



비전 기록의 재현이 끝난 후 오류를 방지하기 위해 **저장한 비전 기록을 사용하기(비전 서비스가 등록되지 않을 때만)** 기능을 언체크해야 합니다.

## 6. Mech-Center

Mech-Center 사용자 매뉴얼에 오신 것을 환영합니다!

Mech-Center는 Mech-Mind 소프트웨어 시스템의 중추이고 제어 센터이며, 관련 소프트웨어의 전역 설정 및 프로젝트의 전체 백업과 복원을 완료할 수 있습니다. Mech-Vision, Mech-Viz, Mech-Eye Viewer, 로봇, 표준 인터페이스, Adapter의 상태도 직접적으로 검사하고 외부 인터페이스 Mech-Interface의 구성 및 관리도 할 수 있습니다.

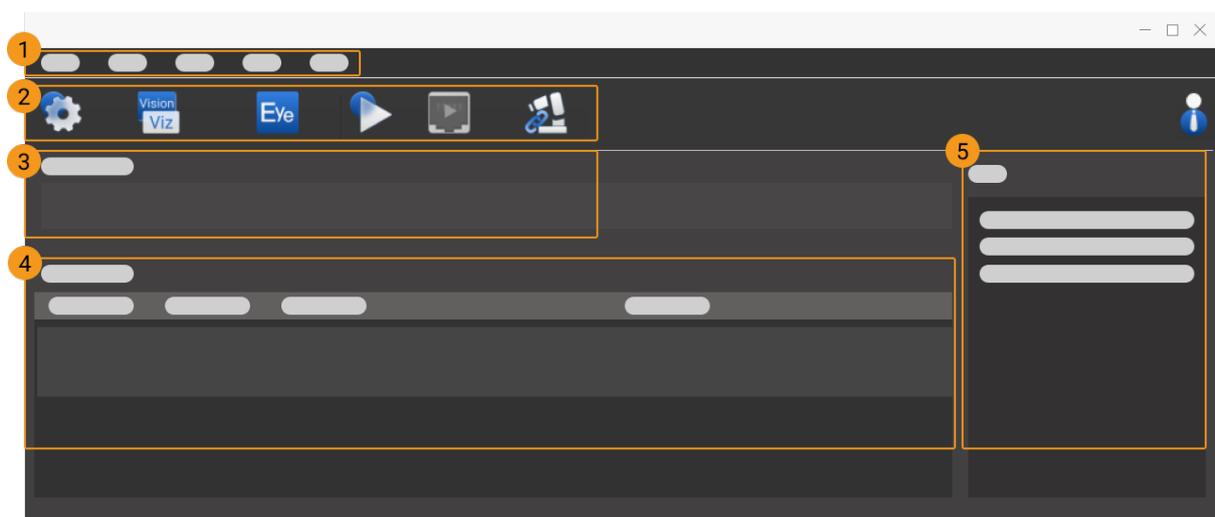
이 매뉴얼은 다음 부분으로 구성됩니다.

번호	제목	내용
1	<a href="#">Mech-Center 퀵 가이드</a>	사용자 인터페이스 및 소프트웨어 기능 소개
2	<a href="#">Mech-Center 시작하기</a>	기본 사용 프로세스
3	<a href="#">Mech-Interface</a>	외부 인터페이스 서비스
4	<a href="#">표준 인터페이스</a>	표준 인터페이스 소개
5	<a href="#">Adapter</a>	Adapter 사용 설명

### 6.1. Mech-Center 퀵 가이드

Mech-Center는 Mech-Mind 소프트웨어 시스템의 중추이고 제어 센터이며, 관련 소프트웨어의 전반 설정 및 프로젝트의 전체 백업과 복원을 완료할 수 있습니다. Mech-Vision, Mech-Viz, Mech-Eye Viewer, 로봇, 표준 인터페이스, Adapter의 상태도 직접적으로 확인하고 외부 인터페이스 Mech-Interface의 구성 및 관리도 할 수 있습니다.

Mech-Center 메인 인터페이스는 주로 다음과 같이 다섯 부분으로 구성됩니다.



- 1. 메뉴 바:** 프로젝트 실행, 사용자 및 뷰 관리, Adapter 생성 및 소프트웨어 정보 확인 등 기능이 있습니다.
- 2. 툴 바:** 구성 설정, 소프트웨어 및 인터페이스 서비스 실행, 로봇 연결 및 실행 등 버튼이 있습니다.

3. **서비스 상태 표시줄:** 등록된 소프트웨어, 카메라 및 로봇 등을 표시합니다.
4. **프로젝트 상태 표시줄:** Mech-Vision/Mech-Viz의 프로젝트 상태, 실행 시간 및 세부 정보를 표시합니다.
5. **로그바:** 현재 실행 중인 프로젝트 및 서비스의 로그 정보를 실시간으로 표시합니다.

### 6.1.1. 메뉴 바

메뉴 바는 파일, 툴, 사용자, 뷰, 도움말 등 작업과 관련된 기본 기능을 제공합니다.

#### 파일

옵션	설명	단축키
프로젝트 백업	이 옵션은 Mech-Mind 소프트웨어 시스템의 프로젝트 백업을 생성하는 데 사용됩니다. 자동 로드한 Mech-Viz/Mech-Vision 프로젝트, Mech-Center의 어댑터 프로젝트 및 Mech-Center의 구성 설정을 저장합니다.	Ctrl+B
복원하기	이 옵션은 선택한 백업에서 프로젝트 및 설정을 복원하는 데 사용됩니다. 자동으로 로드된 프로젝트는 복원 후 대체되므로 프로젝트가 백업되었는지 확인하십시오.	Ctrl+R
프로젝트를 도입하기	이 옵션은 다른 경로에서 프로젝트를 도입할 때 사용됩니다.	Ctrl+I
종료	Mech-Mind 소프트웨어 시스템을 종료합니다.	Ctrl+Q

 오퍼레이터 모드에서는 프로젝트 백업 및 종료 옵션만 사용할 수 있습니다.

#### 툴

옵션	설명
디버그 데이터를 패키징하기	Mech-Mind 소프트웨어 시스템 프로젝트의 디버그 데이터를 패키징하는 옵션으로 시간, 패키징 경로, 저장 옵션을 선택할 수 있습니다.
Adapter 생성기	이 옵션은 사용자 정의 Adapter 프로그램을 생성하는 데 사용됩니다.
로그 뷰어	이 옵션은 Mech-Viz/Mech-Vision의 로그 정보를 확인할 수 있습니다. 수동으로 로그 파일을 선택하고 로딩을 해야 확인할 수 있습니다. 또한 이 옵션은 로그 레벨에 따라 해당 로그를 확인 및 키워드(정규 표현식, 대소문장 구분 및 와일드카드를 선택할 수 있음) 입력을 통해 로그를 필터링하거나 검색하는 것을 지원합니다. 시간 축을 슬라이드하여 해당 시간 이후의 로그를 선택할 수도 있습니다.
서비스 상태 표시	이 옵션은 소프트웨어, 로봇 및 다양한 인터페이스와 같은 서비스의 연결 상태를 표시하는 데 사용됩니다.

 오퍼레이터 모드에서는 Adapter 생성기 옵션이 없습니다.

#### 사용자

**비밀번호를 수정하기:** 관리자 모드에서만 관리자 비밀번호를 수정할 수 있습니다. 오퍼레이터는 이 기능을 사용할 수 없습니다.

 IPC에 처음으로 Mech-Center 소프트웨어를 설치하는 경우에 관리자의 기본 비밀번호는 123456입니다. 또한 비밀번호를 수정한 후에 비밀번호를 타당하게 보관하시고 잊지 마십시오.

## 부

로그: 메인 인터페이스에서 로그 바 표시 여부를 선택할 수 있습니다.

## 도움

옵션	설명	단축키
사용자 매뉴얼	이 옵션은 인터페이스 사용자 매뉴얼을 보는 데 쓰입니다.	F1
업데이트 설명	이 옵션은 각 버전의 새로운 기능과 수정 사항을 설명하는 데 사용됩니다.	없음
버전에 대하여	이 옵션은 소프트웨어 버전 및 저작권 정보를 표시합니다.	없음

## 6.1.2. 툴 바

Mech-Center 툴 바에 구성 설정, Mech-Vision/Mech-Viz 실행, Mech-Eye Viewer 실행, 실행, 인터페이스 서비스 실행, 로봇 컨트롤 및 관리자 총 7 개 버튼이 있습니다.

### 구성 설정

구성 설정은 Mech-Center의 기본 설정을 구현하기, 각 소프트웨어의 열기 경로를 구성하기, 자동 로드하는 프로젝트 경로를 확인하기 및 외부 서비스를 선택하는 데 사용됩니다.

### 선호 설정

선호 설정 페이지는 전반의 개인화 설정에 사용됩니다. 사용자는 소프트웨어의 테마 색상 전환, Mech-Center 소프트웨어 화면 언어 전환(현재 한국어, 중국어, 영어 및 일본어의 4개 언어를 지원합니다), 로그 저장 설정과 같은 것을 사용 습관에 따라 관련 설정을 할 수 있습니다.

- 실행 시 콘솔을 숨기기: Mech-Center 소프트웨어를 시작하면 콘솔이 숨겨집니다.
- PC 부팅 시 Mech-Center 자동으로 실행하기: 컴퓨터 부팅 후에 Mech-Center 소프트웨어가 자동으로 실행됩니다.
- Mech-Vision/Mech-Viz/Mech-Interface 자동으로 실행하기 (있으면): Mech-Center 소프트웨어 실행한 후에 Mech-Vision/Mech-Viz/Mech-Interface도 자동으로 실행됩니다.
- Mech-Vision/Mech-Viz 실행한 후 시스템 트레이로 최소화하기: 실행한 Mech-Vision/Mech-Viz 소프트웨어는 컴퓨터 메시지 표시줄의 트레이로 최소화됩니다.

 변경 사항을 저장하고 Mech-Center를 재부팅해야 설정이 적용됩니다.

### Mech-Viz

Mech-Viz 옵션은 Mech-Viz 소프트웨어의 프로그램 경로를 설정하고 자동으로 로드된 프로젝트의 경로를 표시하는 데 사용됩니다.

- Mech-Viz 실행하기: 기본으로 선택되어 있습니다.
- 프로그램 경로: 설치된 Mech-Viz 소프트웨어의 프로그램 경로를 설정합니다. [...]를 단번 클릭해서 Mech-Viz 설치 목차 아래의 .exe 실행 가능 파일을 선택한 후 경로를 설정할 수 있습니다.
- 프로젝트 경로: Mech-Viz 소프트웨어에서 해당 프로젝트를 자동으로 로드하기를 설정하고 이 부분에서 나오는 것은 그 프로젝트의 경로입니다. 수동으로 경로를 설정할 필요가 없습니다.
- 시뮬레이션 실행: Mech-Viz 소프트웨어는 시뮬레이션만 실행하고 실제 로봇을 이동시키지 않습니다.

## Mech-Vision ,Mech-Eye Viewer

Mech-Vision, Mech-Eye Viewer의 설정 방식은 Mech-Viz와 비슷합니다.

### Robot Server

이 옵션은 Mech-Viz를 통해 로봇을 컨트롤 하거나 임시적 컨트롤 하는 데 사용됩니다.

- 로봇 서버를 실행하기: Mech-Viz를 통해 로봇을 컨트롤 하거나 임시적으로 컨트롤 할 때 이 옵션을 선택하십시오.
- 로봇 서버 경로: 로봇 서버는 현재 Mech-Center가 설치된 로봇 서버입니다. 특별한 필요가 있으면 현재 경로 아래의 main.py를 대체할 수 있습니다.
- Mech-Viz 프로젝트의 로봇 유형: 현재 자동으로 실행하는 Mech-Viz 프로젝트의 로봇 유형을 표시합니다.
- 로봇 IP: 처음으로 실행할 때 로봇을 연결하기 위해 실제 로봇의 IP 주소를 입력해야 합니다.

### Mech-Interface

Mech-Interface는 통합된 인터페이스로 타사 소프트웨어와 통신 하는 데 사용됩니다.

## Mech-Vision, Mech-Viz 실행하기

이 옵션은 Mech-Vision와 Mech-Viz를 빠르게 실행하는 데 사용됩니다. 소프트웨어가 성공적으로 시작되면 서비스 상태 표시줄에 소프트웨어 아이콘과 해당 버전이 표시됩니다.



- 프로젝트를 실행할 때 Mech-Center는 버전의 호환성 검사를 실행합니다. V1.4.0 및 이상 버전의 Mech-Viz, Mech-Vision, Mech-Center와 함께 사용하는 것이 좋습니다.
- Mech-Center에서 Mech-Vision와 Mech-Viz 소프트웨어 버전이 모두 1.4.0보다 낮은 것으로 감지되면 소프트웨어 버전이 1.4.0보다 높아야 한다는 메시지가 표시되고 [실행]을 단번 클릭하면 오류 프롬프트 박스가 나타납니다.

## Mech-Eye Viewer 실행하기

이 옵션은 Mech-Eye Viewer를 빠르게 실행하는 데 사용됩니다. 소프트웨어가 성공적으로 시작되면 서비스 상태 표시줄에 소프트웨어 아이콘과 해당 버전이 표시됩니다.

### 실행

이 옵션은 Mech-Viz와 Mech-Vision에서 설정한 자동 로드된 프로젝트를 실행하는 데 사용됩니다.

## 인터페이스 서비스를 실행하기

이 옵션은 Mech-Interface를 실행하는 데 사용됩니다.

### 로봇 컨트롤

이 옵션은 로봇을 컨트롤하는 데 사용됩니다. 마스터 컨트롤이 연결될 때 로봇 아이콘 및 번호는 서비스 상태 표시줄에 표시됩니다.

### 관리자

Mech-Center에 관리자와 오퍼레이터 두 가지 모드가 있고 기본으로 관리자 모드로 실행합니다. 오퍼레이터 모드로 실행할 때 프로젝트 편집과 설정을 수정하지 못합니다. 아이콘은 오퍼레이터 모드입니다.

- 모드를 바꾸려면 아이콘을 한번 클릭해서 로그인 다이얼로그 박스에서 사용자 유형을 선택하여 [로그인]을 한번 클릭하면 됩니다.
- 관리자 모드에 로그인 하려면 비밀번호가 필요합니다. 비밀번호를 변경하려면 메뉴 바에 있는 **사용자 > 비밀번호 변경**을 한번 클릭하여 나오는 비밀번호 변경 다이얼로그 박스에서 비밀번호와 새 비밀번호를 입력하면 됩니다.

### 6.1.3. 서비스 상태 표시줄

Mech-Center가 실행할 때 서비스 상태 표시줄에 실행한 소프트웨어, 카메라, 로봇 및 인터페이스 서비스가 표시됩니다.



서비스 상태 표시줄에서 소프트웨어 아이콘을 한번 클릭하면 해당 소프트웨어의 메인 화면을 열 수 있습니다.

### 6.1.4. 프로젝트 상태 표시줄

프로젝트 상태 표시줄은 Mech-Viz/Mech-Vision 프로젝트 명칭, 상태, 실행 시간 및 세부 정보를 표시할 수 있습니다. 일반적으로 오른쪽의 로그 바와 결합되어 프로젝트 실행 상태를 볼 수 있습니다.

옵션	설명
프로젝트 명칭	Mech-Viz/Mech-Vision 프로젝트의 명칭입니다.
상태	프로젝트의 현재 상태(유휴 또는 실행 중)입니다.
실행 시간	프로젝트 실행 시작부터 실행 종료까지 걸리는 시간입니다.
세부 정보	프로젝트 실행 시간과 해당 상태를 기록하고 프로젝트 실행 과정에서 오류가 나타난 경우 오류 정보도 여기에 기록됩니다.

### 6.1.5. 로그바

로그바는 현재 실행중인 프로젝트 및 서비스의 로그 정보를 실시간으로 표시합니다.

옵션	설명
D, i, W, E	로그 레벨을 표시합니다. D는 결함 수정(Debug), i는 정상 정보(Info), W는 경고(Warning), E는 오류(Error)입니다. 레벨 옵션을 한번 클릭하면 로그 정보를 필터링 할 수 있고, 여러 레벨 옵션을 동시에 선택할 수도 있습니다.
지우기	모든 로그 내용을 지웁니다.
폴더를 열기	Mech-Center 설치 경로에 있는 <b>logs</b> 폴더를 엽니다.



오류 메시지 앞의 오류 코드를 클릭하면 자세한 정보를 볼 수 있는 온라인 매뉴얼로 이동합니다.

## 6.2. Mech-Center 시작하기

### 소프트웨어 설정

Mech-Center를 시작하여 **툴** 바에 있는 [구성 설정]을 한번 클릭합니다. 구성 설정 화면에서 필요에 따라 해당 옵션을 설정하고 [저장]을 한번 클릭합니다. 해당 옵션은 **구성 설정**을 참조하십시오.



실제로 사용할 때의 상황에 따라 설정하십시오.

- Mech-Vision, Mech-Viz, Mech-Eye Viewer 소프트웨어가 설치된 후에 구성 설정에 있는 경로는 자동으로 추가되며 수동으로 추가할 필요가 없습니다.

## Mech-Vision과 Mech-Viz를 열기

툴 바에서 [ **Mech-Vision/Mech-Viz 실행** ]을 단번 클릭하여 Mech-Vision과 Mech-Viz 소프트웨어를 열 수 있습니다. 소프트웨어 아이콘은 서비스 상태 표시줄에 표시됩니다.



Mech-Vision 또는 Mech-Viz 프로젝트를 자동으로 로드하기를 설정하면 Mech-Center는 해당 프로젝트를 바로 엽니다.

## Mech-Viz 프로젝트 열기

1. Mech-Center 소프트웨어 인터페이스로 들어가기: Mech-Viz 아이콘을 단번 클릭하여 Mech-Viz 메인 인터페이스로 들어갑니다.
2. 프로젝트 구축: 프로젝트를 새로 만들거나 기존 프로젝트를 엽니다.
3. 해당 프로젝트를 자동으로 로드하기: Mech-Viz 프로젝트 리소스 화면에서 프로젝트 명칭을 마우스 오른쪽 버튼으로 클릭하여 \*해당 프로젝트를 자동으로 로드하기\*를 선택합니다.
4. 프로젝트 상태를 확인하기: 프로젝트 상태 표시줄에서 현재 프로젝트의 실행 상태가 표시됩니다.



\*해당 프로젝트를 자동으로 로드하기\*를 선택한 후에 Mech-Viz 프로젝트의 구체적인 경로는 자동으로 Mech-Center의 **구성 설정 > Mech-Viz > 프로젝트 경로**에 추가됩니다.

## Mech-Vision 프로젝트 열기

1. Mech-Center 소프트웨어 인터페이스로 들어가기: Mech-Vision 아이콘을 단번 클릭하여 Mech-Vision 메인 인터페이스로 들어갑니다.
2. 솔루션이나 프로젝트 만들기: 새로운 솔루션이나 프로젝트를 만듭니다.
3. 자동으로 로드하기: 현재 솔루션을 자동으로 로드하거나 현재 프로젝트를 자동으로 로드합니다.
4. 프로젝트 상태를 확인하기: 프로젝트 상태 표시줄에서 현재 프로젝트의 실행 상태가 표시됩니다.

## 카메라 설정을 변경하기

실행 과정에 카메라 설정을 확인하거나 조정할 필요가 있으면 툴 바에서 [ **Mech-Eye Viewer** ]를 단번 클릭하십시오. Mech-Eye Viewer 소프트웨어에서 카메라와 관련된 옵션을 설정합니다. Mech-Eye Viewer 소프트웨어를 시작한 후에 해당 아이콘이 서비스 상태 표시줄에 표시됩니다.

## 로봇을 연결하기

- 마스터 컨트롤 통신을 선택하면 선택한 로봇의 해당 마스터 컨트롤 구성 파일을 참조하여 툴 바에서 [ **로봇 컨트롤** ]을 단번 클릭하십시오. Mech-Center가 로봇과 연결되면 로봇 아이콘이 서비스 상태 표시줄에 표시됩니다.
- 표준 인터페이스나 Adapter 통신을 선택하면 표준 인터페이스나 Adapter 통신 파일을 참조하여 툴 바에서 [ **인터페이스 서비스를 실행하기** ]를 단번 클릭하십시오. 인터페이스 서비스가 시작되면 서비스 상태 표시줄에 해당 아이콘이 표시됩니다.

## 프로젝트를 실행하기



실제 로봇을 실행하기 전에 **구성 설정 > Mech-Viz**에서 **시뮬레이션 실행** 옵션을 언체크했는지 확인하십시오.

툴 바의 [ **실행** ]을 단번 클릭하면 “실행 스타트” 화면이 나옵니다. [ **Yes** ]를 단번 클릭하여 로드된

프로젝트를 실행합니다. 프로젝트 상태 표시줄에서 실행에 관한 정보를 확인할 수 있습니다.



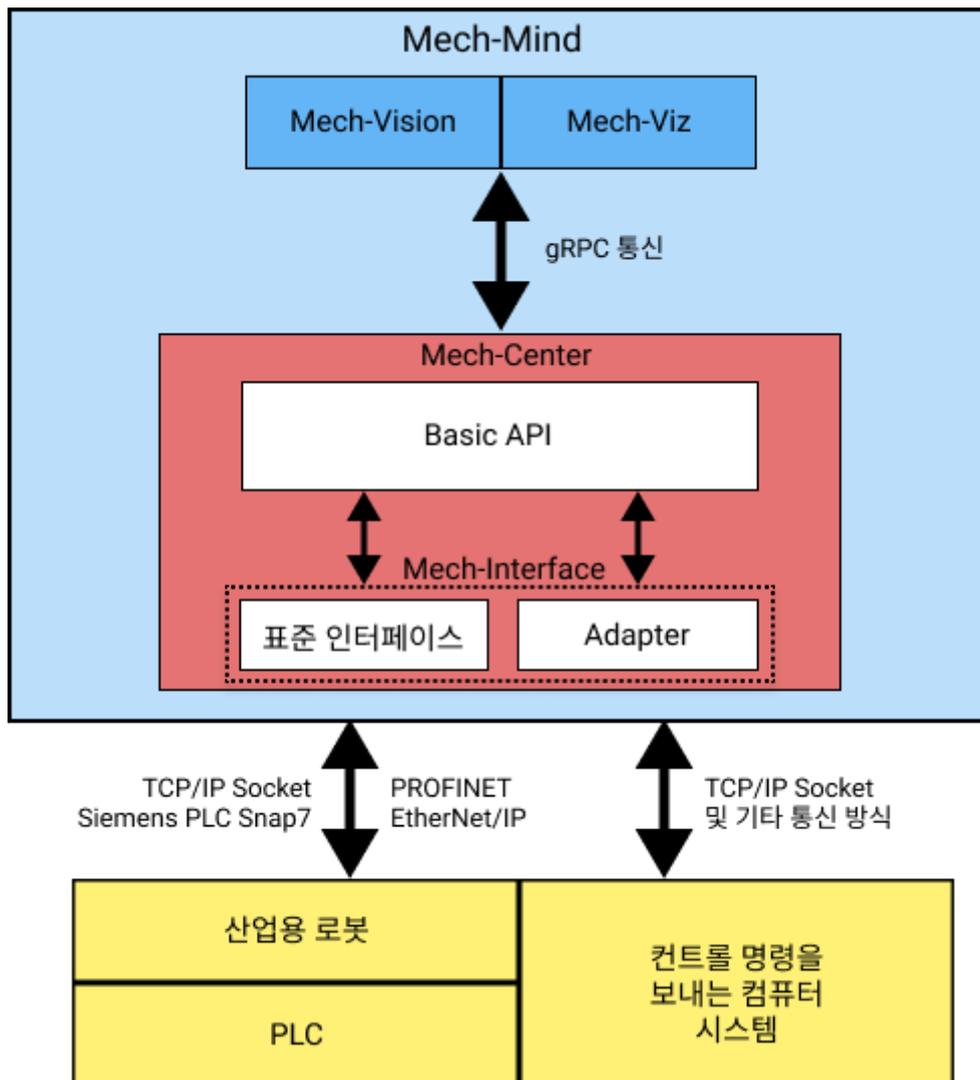
- 사용자가 Mech-Viz 프로젝트를 열 때 나온 \*소프트웨어 호환성 제시\*를 처리하지 않고 직접 Mech-Center에서 [실행]을 단번 클릭하면, Mech-Center의 로그에서 “Mech-Viz는 프로젝트를 로드하고 있기 때문에 Mech-Viz 프로젝트를 실행하지 못합니다: XXX” 정보를 프린트합니다. 이럴 때 사용자가 소프트웨어 호환성 제시를 처리해야 합니다. 즉 소프트웨어 호환성 제시에서 [Yes]를 단번 클릭하여 Mech-Center 실행 버튼을 활성화하면 다시 실행할 수 있습니다.
- 로봇을 실행할 때 안정을 부디 주의하십시오. 비상시에는 터치 펜던트의 비상정지 버튼을 눌러주십시오.

## 6.3. Mech-Interface

외부 통신과의 브릿지 역할을 하는 Mech-Interface는 Mech-Mind 소프트웨어 시스템의 외부 인터페이스 서비스로서 **표준 인터페이스** 및 \*Adapter\*를 포함하며 외부 정보를 수신하고 시스템 내부 정보를 보내는 기능을 실현할 수 있습니다. 이 부분에서 주로 Mech-Interface의 두 가지 통신 방식(표준 인터페이스와 Adapter)의 메커니즘, 차이점 및 응용 시나리오에 대해 설명합니다.

### 통신 메커니즘

두 가지 통신 방식의 메커니즘은 다음 그림과 같습니다.



- Adapter는 외부 통신 장치 (산업 로봇, 상위 컴퓨터, PLC)와 Mech-Vision 및 Mech-Viz 소프트웨어를 연결하는 Python 어댑터 프로그램입니다. 내부적으로는 Mech-Vision 및 Mech-Viz와 통신하고 외부적으로는 외부 장치와 통신합니다.
- 표준 인터페이스는 Mech-Mind에서 제공하는 완전한 Adapter 프로그램 세트로, 다양한 통신 프로토콜을 지원하며 강력한 제어 명령어 세트와 비정상적인 경보 시스템을 갖추고 있어 대부분의 사용자 요구를 충족할 수 있습니다.

## 표준 인터페이스와 Adapter의 비교

Adapter와 표준 인터페이스는 다 외부 장치, Mech-Vision 및 Mech-Viz 소프트웨어와 연결하는 어댑터 프로그램입니다. 표준 인터페이스는 Mech-Mind가 제공하는 고정적인 Adapter 프로그램 세트로 2차 개발 기능을 지원하지 않습니다.

내부적으로는 Basic API를 호출하여 Mech-Vision과 Mech-Viz와 통신하며, 외부적으로는 특정 통신 프로토콜을 통해 외부 장치와 통신합니다.

표준 인터페이스와 Adapter의 자세한 비교는 다음 표에서 볼 수 있습니다.

비교	표준 인터페이스	Adapter
내부 통신	모두 Basic API를 호출하여 Mech-Vision과 Mech-Viz와 통신합니다.	
외부 통신 프로토콜	표준 인터페이스는 다음과 같은 통신 프로토콜만 통해 외부 장치와 통신합니다. - TCP/IP Socket - UDP - Siemens PLC Snap7 - PROFINET - EtherNet/IP - Modbus TCP - 미쓰비시 MC	Adapter는 Python에서 지원하는 통신 프로토콜을 사용하여 외부 장치와의 통신을 지원합니다.
기능	표준 인터페이스는 비전 결과를 제공하는 것만 지원합니다.	Adapter는 비전 관련 기능뿐만 아니라 인터페이스, 데이터 베이스 및 주문 시스템과 같은 Python 지원 기능을 제공합니다.
구성 난이도	표준 인터페이스는 사용하기 쉽고 신속하게 구성을 완료할 수 있습니다.	Adapter는 사용자가 프로그램을 작성해야 하며 시간과 인건비가 비교적 높습니다.
확장 가능성	기능 확장을 지원하지 않습니다.	더 많은 통신 프로토콜과 더 많은 기능을 지원하도록 확장할 수 있습니다.

## 응용 설명

실제 응용 시나리오에서 일반적으로 외부 통신 대상, 사용하는 통신 프로토콜 및 프로젝트에서 필요한 통신 기능 등을 고려하여 어떤 유형의 Mech-Interface를 사용할지 결정해야 합니다.

표준 인터페이스와 Adapter가 지원하는 일반적인 통신 대상 및 통신 프로토콜은 다음 표와 같습니다.

통신 대상	통신 프로토콜	Mech-Interface 유형	설명
로봇	TCP/IP Socket	표준 인터페이스	Mech-Interface는 TCP/IP Socket의 서버로 사용됩니다.
	UDP		Mech-Interface는 UDP의 서버로 사용됩니다.
	PROFINET		Mech-Interface는 PROFINET의 슬레이브 장치로 사용됩니다.
	EtherNet/IP		Mech-Interface는 EtherNet/IP의 슬레이브 장치로 사용됩니다.
	Modbus TCP		Mech-Interface는 Modbus TCP의 슬레이브 장치로 사용됩니다.
마스터 컴퓨터(컨트롤 명령을 보내는 컴퓨터)	HTTP	Adapter	통합 프로젝트에 적합하며 로봇 마스터 제어 모드를 채택합니다.
	WebSocket		
	TCP/IP Socket	표준 인터페이스	Mech-Interface는 TCP/IP Socket의 서버로 사용됩니다.
PLC	TCP/IP Socket	표준 인터페이스	Mech-Interface는 TCP/IP Socket의 서버로 사용됩니다.
	Siemens PLC Snap7		Mech-Interface는 지멘스 PLC Snap7의 클라이언트로 사용됩니다.
	PROFINET		Mech-Interface는 PROFINET의 슬레이브 장치로 사용됩니다.
	EtherNet/IP		Mech-Interface는 EtherNet/IP의 슬레이브 장치로 사용됩니다.
	Modbus TCP		Mech-Interface는 Modbus TCP의 슬레이브 장치로 사용됩니다.
	미쓰비시 MC		Mech-Interface는 MC의 클라이언트로 사용됩니다.



- 프로젝트 요구 사항을 충족시키는 경우 표준 인터페이스를 사용하는 것이 좋습니다. 표준 인터페이스가 프로젝트 요구 사항을 충족시키지 못하는 경우 (예: 외부 장치가 표준 인터페이스에서 지원하지 않는 통신 프로토콜을 사용하는 경우, 프로젝트가 표준 인터페이스에서 지원하지 않는 기능을 필요로 하는 경우) Adapter를 사용하십시오.
- 표준 인터페이스에서 지원하는 기능 목록은 "[표준 인터페이스 개발자 참고 매뉴얼](#)" 부분에서 확인하실 수 있습니다.
- Adapter에서 지원하는 기능 목록은 "[Adapter 기능](#)" 부분에서 확인하실 수 있습니다.

표준 인터페이스와 Adapter에 대한 더 자세한 내용을 확인하시려면 다음 내용을 읽어보십시오.

- [표준 인터페이스](#)
- [Adapter](#)

## 6.4. 표준 인터페이스

Mech-Mind 소프트웨어 시스템을 사용하여 로봇 운동을 제어하지 않고 포즈만 필요한 경우 사용자는 표준 인터페이스를 사용하여 데이터를 전송할 수 있습니다. 표준 인터페이스는 포즈와 작업 데이터를 보내는 것과 같은 가장 기본적인 인터페이스 기능만 제공합니다. 더 많은 인터페이스 기능이 필요한 경우 사용자는 Adapter를 사용하여 정의할 수 있습니다. 표준 인터페이스는 Mech-Center 소프트웨어에 통합되어 별도로 생성할 필요가 없습니다. 사용하려면 소프트웨어에서 표준 인터페이스를 실행해야 합니다.

### 표준 인터페이스 실행하기

**구성 설정 > Mech-Interface** 옵션에서 \*Mech-Interface 실행하기\*를 선택하고 인터페이스 서비스 유형은 \*표준 인터페이스\*를 선택합니다.

### 인터페이스 옵션 및 호스트 주소

사용자는 필요에 따라 외부 서비스 유형을 선택할 수 있습니다.

#### Siemens PLC Client

Siemens PLC Client는 호스트 IP 주소, PLC의 **슬롯 번호** 및 \*DB 블록 번호\*를 설정해야 합니다.

#### TCP Server

TCP Server는 프로토콜 포맷(ASCII 혹은 HEX)을 선택해야 합니다. HEX 포맷에서 \*빅 엔디안 / 리틀 엔디안\*을 선택해야 합니다.

TCP Server는 실제 상황에 따라 포트 번호를 설정해야 하며 기본 포트는 50000입니다.

사용자는 로봇에 따라 사용 가능한 통신 형식을 선택하여 인터페이스 프로그램을 작성할 수 있습니다.

로봇 유형		표준 인터페이스 프로그램 샘플
산업	ABB	HEX 프로그램 패키지
	FANUC	HEX 프로그램 패키지
	KUKA	HEX 프로그램 패키지
	YASKAWA	ASCII 프로그램 패키지
	KAWASAKI	ASCII 프로그램 패키지
	ROKAE	ASCII 프로그램 패키지
	NACHI	ASCII 프로그램 패키지
협력	FANUC CRX	Plugin 플러그인 (HEX)
	UR	URCap 플러그인 (ASCII)
	TM	Plug and Play 플러그인 (ASCII)
	JAKA	ASCII 프로그램 패키지, Addon 플러그인과 함께 제공
	ELITE	ASCII 프로그램 패키지

로봇 유형	표준 인터페이스 프로그램 샘플
기타	사용자는 로봇 프로그램을 작성해야 함, Mech-Center는 샘플을 제공하지 않음



로봇 인터페이스 샘플 프로그램 폴더와 실행 파일은 소프트웨어의 설치 디렉터리에 있는 **Mech-Center/Robot\_Interface** 폴더에 있습니다.

## PROFINET

PROFINET는 로봇 모델 설정이 필요합니다.

## EtherNet/IP

EtherNet/IP는 로봇 모델 설정이 필요합니다.

## MODBUS TCP SLAVE

Modbus TCP SLAVE는 슬레이브 IP, 포트 번호, 장치 주소 및 바이트 순서를 설정해야 합니다(사용자는 일치하는 바이트 순서를 선택하기 위해 테스트할 수 있음).

## 로봇을 선택하기

현장 상황에 따라 해당 로봇 브랜드와 모델을 선택합니다.

설정을 완료한 후 Mech-Center를 저장하고 재부팅하며, \*인터페이스 실행\*을 단번 클릭하면 표준 인터페이스를 실행할 수 있습니다.

## 고급 설정

고급 설정에서 사용자는 단번에 보낼 수 있는 포즈의 최대 수량, Mech-Viz에서 데이터를 획득하는 데 제한 시간 및 Mech-Vision에서 데이터를 획득하는 데 제한 시간을 설정할 수 있습니다.

## 표준 인터페이스를 사용하는 Mech-Viz 예시 프로젝트

Mech-Mind 시스템 소프트웨어 설치 디렉터리의 `Mech-Center/tool/viz\_project`에는 표준 인터페이스를 사용하는 4 개의 Mech-Viz 예시 프로젝트가 포함되어 있습니다.

### check\_collision

이 예시 프로젝트는 비전 가이드 작업에서 간단한 경로 계획 및 충돌 감지를 하는 데 사용됩니다.

### outer\_move

이 예시 프로젝트는 Mech-Viz 시뮬레이션 로봇이 외부 클라이언트에서 전달된 위치로 이동해야 하는 경우에 사용됩니다.

### suction\_zone

이 예시 프로젝트는 Mech-Viz가 분할 제어가 있는 빨판을 사용하여 종이 상자를 다중으로 잡는 경우 빨판의 분할 제어 신호를 출력하는 데 사용됩니다.

### vision\_result\_reuse

이 예시 프로젝트는 Mech-Viz가 동일한 사진 캡처 결과를 사용하여 비전 가이드로 여러 번 경로 계획 및 충돌 감지를 하는 데 사용됩니다.

## 6.5. Adapter

표준 인터페이스의 기능으로 프로젝트의 수요를 충족하지 못했을 때 Adapter 프로그램을 작성함으로써 복잡한 컨트롤 과정을 실현할 수 있습니다.

이 부분에는 주로 아래 내용을 포함합니다:

- [Adapter 퀵 가이드](#)
- [Adapter 생성기 매뉴얼](#)
- [Adapter 프로그래밍 가이드](#)
- [Adapter 프로그래밍 샘플](#)



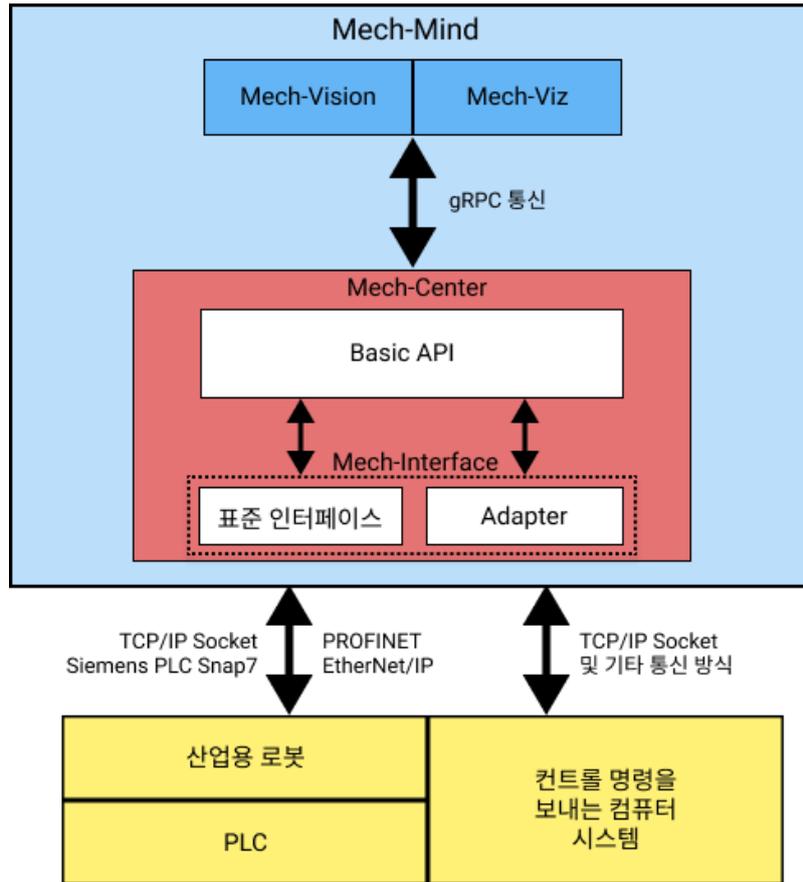
Adapter 프로젝트에서 다른 Python 라이브러리를 사용해야 하는 경우 Mech-Mind 소프트웨어 시스템의 **Mech-Center/python** 디렉터리에 설치하십시오. Python 라이브러리의 설치 방식은 다음과 같습니다:

1. CMD 혹은 PowerShell 프로그램을 엽니다.
2. Mech-Mind 소프트웨어 시스템의 설치 디렉터리에 있는 **Mech-Center/python** 폴더로 전환합니다.
3. `./python -m pip install library_name` 명령어를 입력하고 실행합니다.

### 6.5.1. Adapter 퀵 가이드

#### Adapter에 관한 소개

Adapter는 Mech-Center 소프트웨어의 통신 컴포넌트로 Basic API 인터페이스를 통해 Mech-Vision, Mech-Viz와 gRPC 통신을 진행하며 외부 장치와 TCP/IP Socket, HTTP 또는 PLC 데이터 블록 전송 프로토콜(예를 들어 미쓰비시 PLC MC) 등과 같이 다양한 산업적 통신 방식으로 통신을 진행합니다.



### Adapter 기능

Adapter를 통해 다음과 같은 기능을 실현할 수 있습니다:

- 비전 시스템 내부에서 Mech-Vision 및 Mech-Viz 소프트웨어를 컨트롤할 수 있습니다.

영역	기능
Mech-Vision에 관하여	Mech-Vision을 실행하여 비전 결과를 획득하기
	Mech-Vision 스텝 파라미터를 설정하기
	Mech-Vision 스텝 파라미터를 읽어내기
	Mech-Vision 파라미터 레시피를 전환하기

영역	기능
Mech-Viz에 관하여	Mech-Viz를 실행하기
	Mech-Viz를 종료하기
	Mech-Viz 스텝 파라미터를 설정하기
	Mech-Viz 스텝 파라미터를 읽어내기
	클램프 번호를 설정하기
	로봇의 실행 속도를 설정하기
	포인트 클라우드의 충돌 파라미터를 설정하기
	Mech-Viz의 실행 상태를 획득하고 반환하기
기타	상세한 기능은 “ <a href="#">Adapter 프로그래밍 샘플</a> ” 내용을 참조하십시오.

- 비전 시스템 외부에서 UI, 데이터 베이스, 파일 읽기 및 쓰기, Web 시스템과의 통신 등 비전 이외의 기능을 실현할 수 있습니다.

외부적인 기능은 Python 프로그래밍을 통해 실현되어야 합니다.

## Adapter 개발

TCP/IP Socket 통신 방식을 사용하는 경우, Mech-Center에서는 **Adapter 생성기** 기능을 제공하여 Adapter에 처음 사용하는 사용자가 빠르게 Adapter 프로그램을 생성하고 Adapter 프로젝트를 구축할 수 있도록 도와줍니다. 자세한 정보는 "[Adapter 생성기 매뉴얼](#)" 내용을 참조하십시오.

사용자는 생성된 Adapter를 기반으로 프로그램을 이차 개발할 수도 있고, 처음부터 완전한 Adapter 프로그램을 작성할 수도 있습니다. 자세한 정보는 "[Adapter 프로그래밍 가이드](#)" 및 "[Adapter 프로그래밍 샘플](#)" 내용을 참조하십시오.

## Adapter 프로젝트 배포

Adapter 프로그램을 작성한 다음에 다음 단계를 참조하여 Adapter 프로젝트를 배포할 수 있습니다:

1. Mech-Center 소프트웨어를 열고 툴 바에서 [구성 설정] 버튼을 클릭합니다.
2. 구성 설정 창에서 **Mech-Interface\***를 선택하고 **\*Mech-Interface를 실행하기** 버튼을 선택하고 \*인터페이스 서비스 유형\*을 \*Adapter\*로 설정합니다.
3. \*어댑터 프로젝트 폴더\*를 Adapter 프로그램이 저장되는 디렉터리로 설정하십시오.
4. 작업 현장 실제에 따라 \*호스트 주소\*를 설정합니다. 포트는 상대 포트와 일치해야 합니다.
  - 상대 쪽이 통신의 서버 역할을 하면 \*호스트 주소\*는 상대 쪽의 IP 주소로 설정해야 합니다.
  - 상대 쪽이 통신의 클라이언트 역할을 하면 \*호스트 주소\*는 “0.0.0.0”로 설정해야 합니다.
5. [저장] 버튼을 클릭한 다음 Mech-Center 소프트웨어를 다시 시작합니다.
6. 툴 바에 있는 [인터페이스 서비스를 실행하기] 버튼을 클릭하여 Adapter 서비스를 시작하십시오.

인터페이스 서비스를 실행하기 버튼이 인터페이스 서비스 정지 버튼으로 변하고 서비스 표시줄에 이미 실행된 Adapter 프로그램이 표시되면 Adapter 서비스가 이미 성공적으로 시작되었다는 뜻입니다.

Adapter에 대해 간단하게 알아본 후 "[Adapter 생성기 매뉴얼](#)" 내용을 참조하여 첫 번째 Adapter 프로그램을 생성하십시오.

## 6.5.2. Adapter 생성기 매뉴얼

이 부분에서는 Adapter 생성기와 Adapter 생성기를 사용하여 Adapter 프로그램을 빠르게 생성하는 방법에 대해 설명하겠습니다.

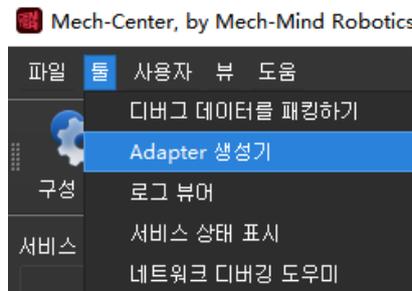
### Adapter 생성기 소개

Adapter 생성기는 Mech-Center에 통합된 작은 구성 요소입니다. TCP / IP Socket 통신 방식을 사용하며 Mech-Vision에서 제공하는 비전 포즈만 사용하는 시나리오에만 적용됩니다.

Adapter 생성기를 사용하여 다음과 같은 기능을 실현할 수 있습니다:

- Adapter 프로그램을 빠르게 생성하고 Adapter 프로젝트를 구축할 수 있습니다.
- Adapter 프로그래밍과 관련된 내용을 신속히 파악하여 복잡한 실제 요구 사항을 충족시킬 수 있습니다.

**틀** > **Adapter 생성기** 버튼을 클릭하여 Adapter 생성기 화면에서 관련 사항을 설정하십시오.



 Adapter 생성기는 관리자 모드에서만 사용할 수 있습니다.

### Adapter 생성기를 통해 Adapter 프로그램을 빠르게 생성하기



구성의 편의를 위해 해당 구성 요소에 대한 자세한 설명을 제공하며 구성 요소 위에 마우스 커서를 올려 놓으면 확인할 수 있습니다.

#### 네트워크 구성 - 서버 또는 클라이언트

이 단계에서 **Adapter 명칭**, **Adapter의 역할**, **스트림 포맷** 파라미터를 설정하고 아래와 같이 `:guilabel: `다음으로`` 를 클릭합니다.



### 파라미터에 관한 설명:

- **Adapter 명칭:** Adapter 프로그램의 명칭을 지정합니다.
- **Adapter의 역할:** Adapter는 TCP/IP Socket 통신의 서버(Server) 또는 클라이언트(Client)로 지정합니다. 클라이언트로서 작동하고 서버가 클라이언트에 대한 포트 제한이 있는 경우, **포트 바인딩** 버튼을 선택해야 합니다.



상대 쪽과의 정상적인 통신을 보장하기 위해 Adapter 프로젝트를 배포할 때 올바른 호스트 IP 주소와 포트를 지정해야 합니다. 자세한 내용은 [Adapter 프로젝트 배포](#)를 참조하십시오.

- **스트림 포맷:** 통신 전송 형식을 지정하며 ASCII 문자열 또는 HEX(16진법)를 지원합니다. \*스트림 포맷\*을 "HEX"로 설정한 경우, 바이트 순서(endianness)를 "빅 엔디안" 또는 "리틀 엔디안"으로 지정해야 합니다.

### Mech-Viz/Mech-Vision 구성 - 포즈와 관절 각도

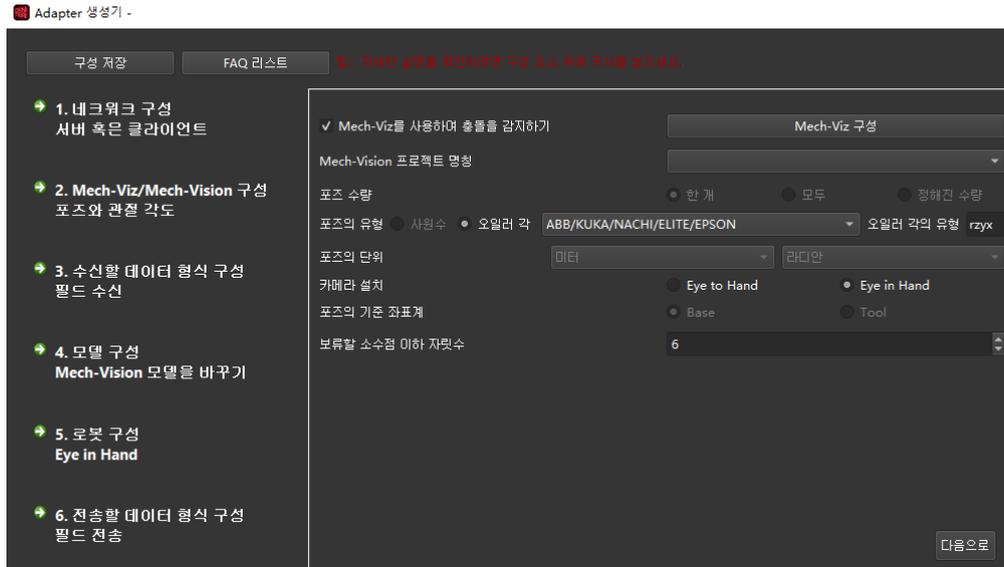


- 이 단계를 실행하기 전에 Mech-Vision 프로젝트와 충돌 감지에 사용할 Mech-Viz 프로젝트를 준비하고 자동로드 옵션을 활성화하여 Mech-Center에서 프로젝트가 로드되었는지 확인하십시오.
- Mech-Center는 충돌 감지를 위한 Mech-Viz 샘플 프로젝트 "check\_collision" 을 제공합니다. Mech-Center 설치 경로 아래의 "tool\viz\_project" 디렉터리에서 획득하십시오.



"check\_collision" 샘플 프로젝트에서 볼 수 있듯이 프로젝트는 "비전 인식" 스텝에 의해 트리거되며 프로젝트에 비이동 스텝을 포함해야 합니다. notify\_1, notify\_2 및 visual\_look\_1 등 스텝의 이름은 수정될 수 없습니다.

이 단계에서 Mech-Viz와 Mech-Vision 프로젝트의 파라미터를 수정해야 합니다.

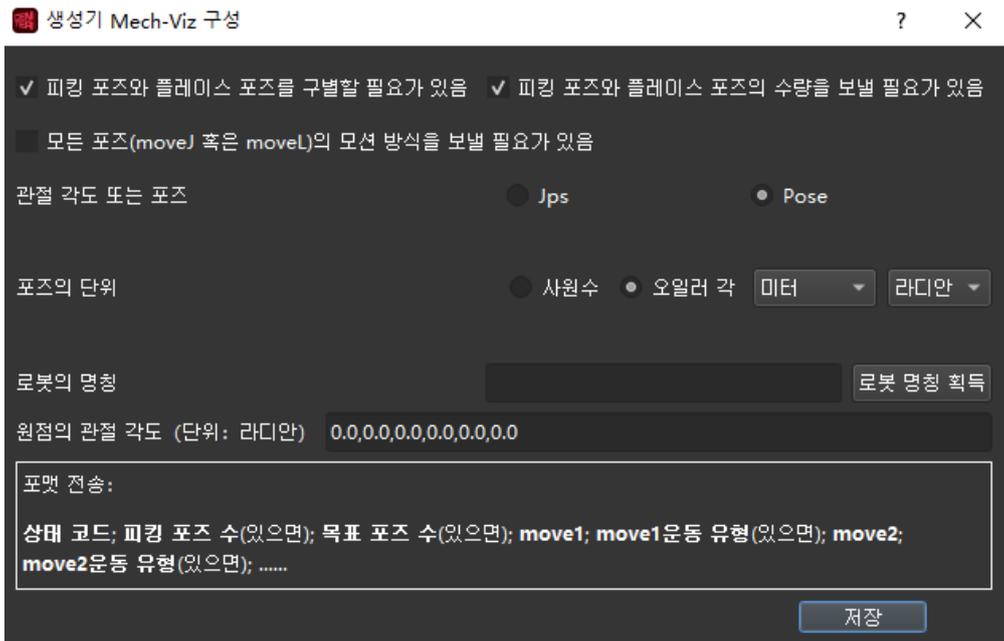


### Mech-Vision 파라미터에 관한 설명:

- **Mech-Vision 프로젝트 명칭:** Mech-Vision 프로젝트의 이름을 지정합니다. 드롭다운 목록에서 Adapter와 상호 작용할 Mech-Vision 프로젝트를 선택합니다.
- **포즈 수량:** 대상에게 보낼 포즈 수를 선택합니다.
- **포즈의 유형:** 사원수 또는 오일러 각을 선택할 수 있습니다.
- **포즈의 단위:** 일반적으로 밀리미터와 도를 사용하도록 선택합니다.
- **카메라 설치:** 카메라 설치 방식(ETH와 EIH 두 가지 경우가 있음)을 선택합니다.
- **포즈의 기준 좌표계:** 전송된 포즈의 기준 좌표계를 결정합니다. 일반적으로 포즈는 로봇 베이스 좌표계를 기준으로 합니다. EIH 시나리오에서 로봇이 로봇 말단의 포즈를 제공할 수 없는 경우 포즈는 TCP 좌표계를 기준으로 해야 합니다.
- **보류할 소수점 이하 자릿수:** 전송된 포즈의 소수점 이하 자릿수를 결정합니다. 최대 자릿수는 10입니다.

### Mech-Viz 파라미터에 관한 설명:

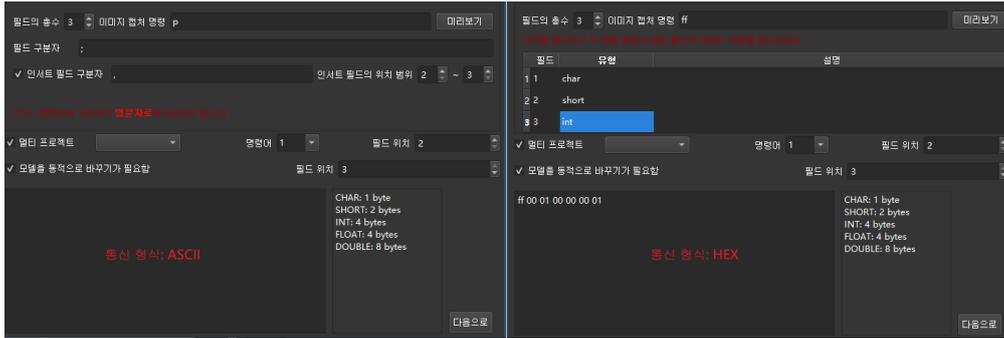
- **Mech-Viz를 사용하여 충돌을 감지하기:** 선택 후 비전 포즈는 Mech-Viz 프로젝트의 충돌 감지를 거쳐 계획에 실패한 포인트를 필터링하고 프로세스에서 충돌이 없는 피킹 포즈를 선택합니다.
- **Mech-Viz 구성.** Mech-Viz 충돌 감지 기능을 사용한 후 이 옵션이 활성화됩니다. 이 옵션을 클릭하면 **생성기 Mech-Viz 구성** 창이 나타나 관련 파라미터를 설정한 후 [저장] 버튼을 클릭합니다. 아래 그림과 같습니다.



- **피킹 포즈와 플레이스 포즈를 구별할 필요가 있음:** 피킹 포즈는 비전 이동 이전의 모든 포즈(비전 이동 포함)이며, 배치 포즈는 비전 이동 이후의 모든 포즈입니다. 일부 시나리오에서 로봇은 작업에 따라 피킹 동작과 배치 동작을 구분해야 합니다.
- **피킹 포즈와 플레이스 포즈의 수량을 보낼 필요가 있음:** 포즈의 수량이 많으면 수량 필드를 함께 보내도 됩니다. 선택하면 기본적으로 위치 수가 1 개보다 크면 필드를 보냅니다.
- **모든 포즈(moveJ 혹은 moveL)의 모션 방식을 보낼 필요가 있음:** Mech-Viz에서 이동 스텝의 운동 방식은 관절 운동(moveJ) 또는 직선 운동(moveL)으로 구분됩니다.
- **코드를 업데이트하기:** 기본적으로 관절 운동의 코드는 1이고 직선 운동의 코드는 2입니다. 사용자기 코드를 자체적으로 정의할 수 있습니다. 변경 후 [ **코드를 업데이트하기** ] 버튼을 클릭하면 적용됩니다.
- **관절 각도 또는 포즈:** 포즈를 전송할 때 포즈의 형식을 설정합니다. 기본적으로 관절 각도로 설정합니다.
- **관절 각도의 단위/ 포즈 단위:** 포즈를 보내는 형식이 \*Jps\*인 경우 Jps의 단위를 설정합니다. 일반적으로 도를 사용합니다. 포즈를 보내는 형식이 \*Pose\*인 경우 포즈의 단위를 설정합니다. 일반적으로 밀리미터를 사용합니다.
- **로봇의 명칭:** 로봇 서버의 이름을 지정합니다. Mech-Viz를 사용하여 로봇 운동을 시뮬레이션하려면 실제 로봇 서비스가 필요하며 생성된 Adapter는 이 서비스를 시뮬레이션합니다. 로봇의 명칭은 이 서비스의 명칭이며 Mech-Viz에서 사용되는 로봇의 이름과 일치해야 합니다. “로봇 명칭 획득” 버튼을 클릭하여 자동으로 로봇 이름을 추가할 수 있습니다.
- **원점의 관절 각도(단위:라디안):** Mech-Viz에서 운동의 시작점을 설정하며 단위는 라디안이며 심표로 구분됩니다. Mech-Viz에서 원점으로 사용할 수 있는 **이동** 스텝을 편집하고 해당 원점의 관절 위치를 복사할 수 있습니다.

#### 수신할 데이터 형식 구성 - 필드 수신

이 단계에서는 이미지 캡처 명령어, 멀티 프로젝트(명령어), 총 필드 수, 필드 유형, 필드의 구분자 및 하위 필드 구분자를 포함하여 수신된 필드의 형식을 설정한 다음 [ **다음으로** ] 버튼을 클릭합니다. 아래 그림과 같습니다.



- **필드의 총수:** 설정할 파라미터의 개수와 과련이 있고 값 범위는 1~10입니다. 필드 중에 이미지 캡처 명령어가 있어야 합니다.
- **이미지 캡처 명령:** 외부 통신 장비가 Mech-Mind 소프트웨어 시스템으로 이미지 캡처 명령어를 보내면 카메라를 사용하여 이미지를 캡처합니다. 통신 포맷이 ASCII 코드인 경우 문자를 사용하는 것이 좋으며, 예를 들어 문자 p를 사용하고 필드 위치는 기본값으로 1을 사용하는 것이 좋습니다. 통신 포맷이 16진법(HEX)인 경우, 16진법 형식의 정수를 사용해야 하며, 예를 들어 0xff 또는 ff를 사용해야 합니다.
- **필드 구분자 및 insert 필드 구분자:** ASCII 통신인 경우 설정해야 합니다. 필드가 2 개 이상인 경우, 필드 구분자를 입력해야 하며 추가 정보에 다른 구분자가 필요한 경우 insert 필드 구분자도 필요합니다. insert 필드의 시작 및 끝 범위를 지정할 수 있습니다.
- **필드 유형 및 필드 설명:** 16진법 (HEX) 통신인 경우 설정해야 합니다. CHAR, SHORT, INT, FLOAT 및 DOUBLE과 같은 선택 가능한 필드 유형이 있습니다. \*필드 설명\*은 필드의 기능 또는 의미를 설명하는데 사용됩니다.
- **멀티 프로젝트(명령어):** 선택 가능한 사항으로, 하나의 프로젝트에 여러 Mech-Vision 프로젝트가 있고 외부 명령에 따라 다른 Mech-Vision 프로젝트를 호출해야 하는 경우 명령 코드를 구성할 수 있습니다.



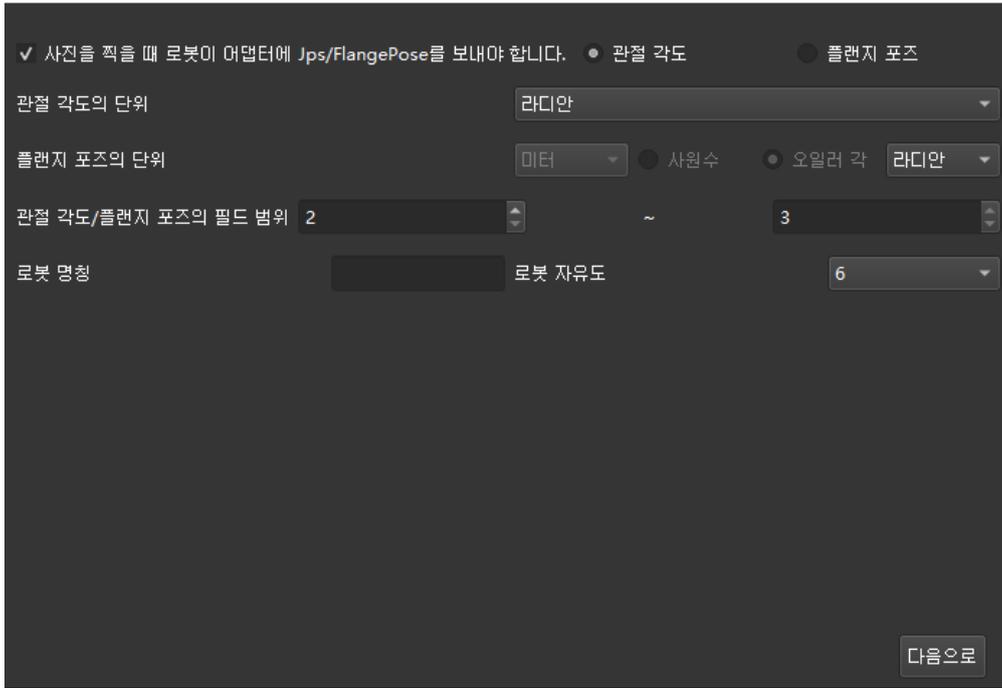
각 프로젝트에 대한 명령과 필드 위치는 모두 고유하며 다른 필드와 겹치지 않도록 해야 합니다.

## 로봇 구성 - Eye In Hand



**Mech-Viz/Mech-Vision 구성 - 포즈와 관절 각도** 스텝에서 **카메라 설치** 파라미터는 \*Eye In Hand\*로 설정하면 이 단계를 사용할 수 있습니다.

이 단계에서는 이미지 캡처 시 로봇의 포즈 형식을 설정하고 [ **다음으로** ] 버튼을 클릭합니다. 아래 그림과 같습니다.



### 파라미터에 관한 설명:

- **사진을 찍을 때 로봇이 Adapter에 관절 각도/플랜지 포즈를 보내야 합니다.:** 상대 쪽은 로봇 베이스 좌표계를 기준으로 한 물체 포즈가 필요한 경우, 로봇이 이미지를 캡처할 때의 관절 각도 또는 플랜지 포즈를 제공해야 합니다. 이때 이 옵션을 선택하십시오. 선택한 후 포즈의 유형을 지정하고 관절 각도 또는 플랜지 포즈를 선택할 수 있습니다.
- **관절 각도의 단위:** 관절 각도의 단위를 지정합니다. 도 또는 라디안을 선택할 수 있습니다.
- **플랜지 포즈의 단위:** 플랜지 포즈를 지정합니다. 사원수 또는 오일러 각을 선택할 수 있습니다. 사원수를 사용하는 경우, 미터 또는 밀리미터 단위를 선택할 수 있습니다. 오일러 각을 사용하는 경우, 도 또는 라디안 단위를 선택할 수 있습니다.
- **관절 각도/플랜지 포즈 필드 범위:** 전체 필드에서 관절 각 또는 포즈의 시작 및 종료 필드를 지정합니다.

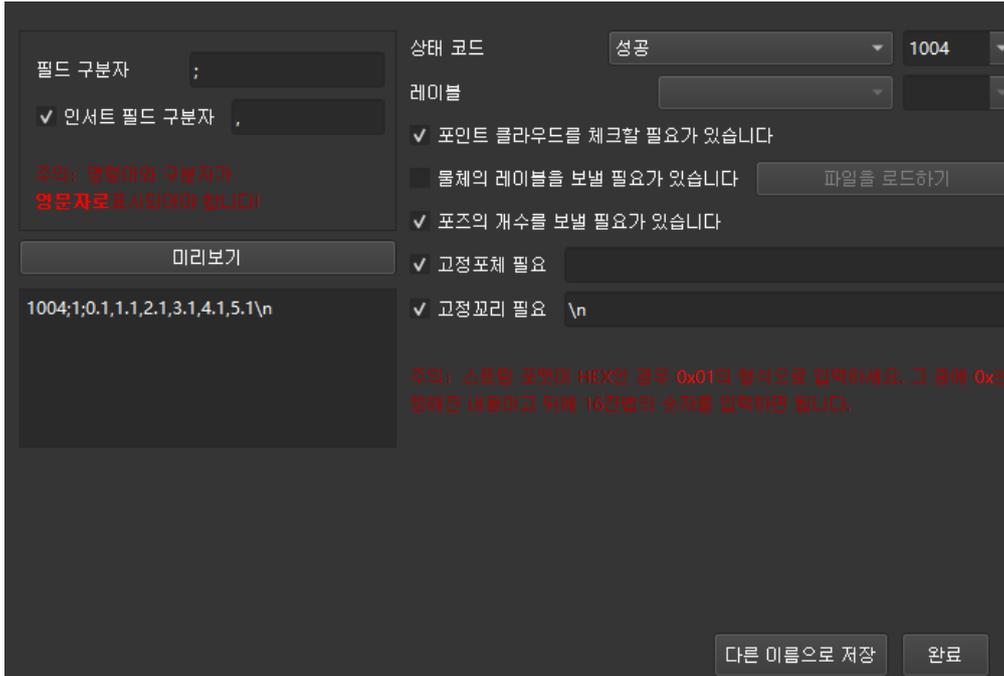


인덱스 위치는 1부터 시작하여 다른 필드와 중복되지 않아야 합니다. 예를 들어 인덱스 위치 1은 이미지 캡처 명령입니다.

- **로봇 명칭:** 로봇 서비스를 식별하는 데 사용되며, Mech-Viz에서 로봇 이름과 일치해야 합니다.
- **로봇의 자유도:** 4축 또는 6축의 로봇을 지원합니다. 프로젝트의 실제 상황에 따라 로봇의 자유도를 선택합니다.

### 전송할 데이터 형식 구성 - 필드 전송

이 스텝에서 전송할 포즈의 형식을 지정하고 [ 다음으로 ] 버튼을 클릭합니다. 아래 그림과 같습니다.



### 파라미터에 관한 설명:

- **필드 구분자 및 인서트 필드 구분자:** 구분자의 형식을 설정합니다. 통신 형식이 ASCII인 경우 설정해야 합니다. 필드가 2 개 이상인 경우, 필드 구분자를 입력해야 하며 추가 정보에 다른 구분자가 필요한 경우 인서트 필드 구분자도 필요합니다. 인서트 필드의 시작 및 끝 범위를 지정할 수 있습니다.
- **상태 코드:** 전송 상태를 설정합니다. 각 상태에 해당하는 상태 코드는 유일합니다.
- **포인트 클라우드를 체크하는 필요가 있습니다:** 이 옵션을 선택하면 Adapter는 포인트 클라우드를 체크합니다. 포인트 클라우드가 존재하지 않는 경우 해당 상태 코드를 출력합니다.
- **물체의 레이블을 보낼 필요가 있습니다.:** 물체의 레이블을 보내는 것은 Mech-Vision에서 인식된 레이블을 상대에게 보내는 것을 의미하며, 각 레이블은 포즈 뒤에 연결됩니다. 상대 쪽이 레이블 문자열을 해석하기 어려운 경우 해당 레이블의 코드를 지정할 수도 있으며, 모든 레이블 문자열을 포함하는 레이블 파일을 로드해야 합니다. 하지만 레이블 파일의 형식은 반드시 json 배열 형식이어야 합니다.
- **포즈의 개수용 보낼 필요가 있습니다:** 전송할 때 현재 전송된 포즈가 포함된 포즈의 수량입니다.
- **고정포체 필요:** 비전 인식 실패 시 상대 쪽에게 메시지(오류 코드 뒤의 메시지)를 보내야 합니다.
- **고정꼬리 필요:** 이 옵션을 선택하면 데이터 뒤에 고정된 테일 식별자를 추가해야 합니다.

**i** | 통신 형식이 16진법(HEX)인 경우 상태 코드, 포즈 수, 포즈 필드의 값 유형을 설정해야 합니다.

위의 모든 구성을 완료한 후 “완료” 또는 “다른 이름으로 저장” 버튼을 클릭하여 Adapter 프로젝트를 저장합니다. Adapter 프로젝트를 저장한 후 [Adapter 프로젝트 배포](#) 내용을 참조하여 Adapter 프로젝트를 배포하고 사용할 수 있습니다.

Adapter의 프로그래밍과 관련된 더 많은 정보는 아래 내용을 읽어 주십시오.

- [Adapter 프로그래밍 가이드](#)
- [Adapter 프로그래밍 샘플](#)

### 6.5.3. Adapter 프로그래밍 가이드

이 부분에서는 Adapter 프로그래밍 스타일 규범과 프로그래밍 구문 지식을 소개하겠습니다.

Adapter 개발자는 다음 지식을 구비해야 합니다:

- Python 기초 문법
- JSON 데이터 형식

먼저 **Adapter 프로그래밍 스타일 규범** 내용을 읽어주시기 바랍니다.

- [Adapter 프로그래밍 스타일 가이드라인](#)

다음으로 Adapter 프로그래밍에 관련된 **추상적인 상위 클래스 인터페이스\*와 \*Util** 패키지에 익숙해지고 Adapter의 가장 기본적인 상위 클래스와 자주 사용하는 함수를 파악해야 합니다.

- [추상 상위 클래스 인터페이스](#)
- [Adapter util 패키지](#)

마지막으로 요구 사항에 따라 추상적인 상위 클래스 인터페이스를 구현하여 내부(Mech-Vision 및 Mech-Viz)와 외부(외부 장치) 간의 통신을 실현합니다.

- [인터페이스를 획득하기](#)
- [Mech-Vision 인터페이스](#)
- [Mech-Viz 인터페이스](#)
- [RobotService 인터페이스](#)
- [기타 인터페이스](#)

### 6.5.3.1. Adapter 프로그래밍 스타일 가이드라인

Adapter는 Python 언어로 작성되었으므로, Adapter 개발자는 [Style Guide for Python Code](#) 내용을 엄격히 준수해야 합니다.

또한 프로그래밍된 Adapter의 품질과 가독성을 향상시키기 위해 Adapter 개발자는 이 부분의 내용도 따라야 합니다.

#### 명명

1. 클래스 이름은 카멜 표기법을 사용하십시오. 예를 들면 다음과 같습니다:

```
class AdapterWidget
```

2. 클래스의 멤버 변수와 멤버 함수는 밑줄로 연결된 소문자 단어를 사용합니다. 예를 들면 다음과 같습니다:

```
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. 클래스의 멤버 변수와 멤버 함수가 클래스 내부에서만 사용되는 경우 이름 앞에 밑줄을 추가하여 이름이 클래스 외부에서는 사용하지 않는 것이 좋지만 클래스 외부에서는 계속 사용할 수 있음을 나타냅니다. 예를 들면 다음과 같습니다.

```
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. 상수는 밑줄로 연결된 대문자를 사용합니다. 예를 들면 다음과 같습니다:

```
ADAPTER_DIR = "D:/adapter_for_communication"
```

## 주석

주석은 너무 많으면 안 되며 표현이 복잡하거나 표현된 의미가 비교적 중요할 때만 추가하면 됩니다.

- 한 줄 주석

#으로 한 줄 주석을 시작합니다.

- 여러 줄 주석

""" 및 """ 사이에 여러 줄 주석을 추가하십시오.

- 함수, 클래스 또는 패키지에 대한 소개 주석

주석은 함수 아래, 클래스 아래 또는 패키지 상단에 있습니다. 예를 들면 다음과 같습니다.

```
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
```

## 로그

로그는 문제가 발생했을 때 문제를 분석하는 데 도움이 될 수 있습니다. 로그 정보는 적절할 때(예를 들어 중요한 함수를 호출하거나 함수가 참조할 수 있는 데이터를 반환할 때) 출력하는 것이 좋습니다.

Adapter는 다음과 같은 두 가지 로그 출력 방법을 지원합니다:

- **print:** 이 방법은 로그만 콘솔에 출력합니다. 이 방법은 런타임에 실시간 관찰에 편리하지만 프로그램이 실패하면 메시지가 손실됩니다. 따라서 실제 생산 환경에서는 사용하지 않는 것이 좋습니다.
- **logging:** 이 방법은 형식화된 로그 표시 형식을 지원하고 로그 파일에 로그 저장도 지원합니다(옵션 기능). 따라서 이 방법을 사용하는 것이 좋습니다.

### 6.5.3.2. 추상 상위 클래스 인터페이스

추상 상위 클래스 인터페이스는 하위 클래스가 상위 클래스를 상속할 때 실제 필요에 따라 다시 작성할 수 있는 함수를 말합니다. 이 부분에서는 다음 추상 상위 클래스에 대해 소개합니다.

#### Communication

통신 관련 클래스의 소스 파일은 Mech-Mind 소프트웨어 시스템 설치 디렉터리의 [Mech-Center/src/interface/communication.py](#) 파일에 있습니다.

#### Communication 클래스

Communication 클래스는 통신을 담당하는 기본 클래스로 일련의 인터페이스를 제공하며, 서버나 클라이언트는 이 클래스의 인터페이스 기능을 다시 작성해야 합니다.

클래스 함수	설명
is_connected()	연결 상태 확인
set_recv_size()	수신 데이터의 길이를 설정, 기본값은 1024 바이트
send()	인터페이스 함수, 데이터 전송
recv()	인터페이스 함수, 데이터 수신
close()	인터페이스 함수, 연결 끊기
before_recv()	인터페이스 함수, 데이터를 수신하기 전에 실제 상황에 따라 논리를 추가할 수 있으며 이 함수를 다시 작성할 수 있음
after_recv()	인터페이스 함수, 데이터를 수신한 후 실제 상황에 따라 논리를 추가할 수 있으며 이 기능을 다시 작성할 수 있음
after_handle()	인터페이스 함수, 데이터를 수신한 후 실제 상황에 따라 논리를 추가할 수 있으며 이 함수를 다시 작성할 수 있음

#### TcpServer 클래스

TcpServer 클래스는 TCP/IP 소켓 서버를 캡슐화합니다.

클래스 함수	설명
bind_and_listen()	포트 바인딩

클래스 함수	설명
local_socket()	로컬 소켓 정보 제공
remote_socket()	원격 소켓 정보 제공
accept()	클라이언트 연결 수락
send()	데이터 전송
recv()	데이터 수신
close()	소켓 연결 끊기
close_client()	클라이언트 연결 끊기

### TcpClinet 클래스

TcpClient 클래스는 TCP/IP 소켓 서버를 캡슐화합니다.

클래스 속성	설명
is_bind_port	포트 바인딩 여부, 서버가 연결된 클라이언트의 포트를 제한하는 경우 이 변수는 True여야 함

클래스 함수	설명
send()	데이터 전송
recv()	데이터 수신
close()	연결 끊기
set_timeout()	제한 시간을 설정, 파라미터의 단위: 초
reconnect_server()	서버 다시 연결
after_connect_server()	인터페이스 함수, 최초 서버 접속 성공 후 작동
after_reconnect_server()	인터페이스 함수, 서버에 성공적으로 다시 연결한 후 작동
after_timeout()	인터페이스 함수, 타임아웃 후 작동

### Adapter

Adapter 관련 클래스의 소스 파일은 Mech-Mind 시스템 소프트웨어 설치 디렉터리의 [Mech-Center/src/interface/communication.py](#) 파일에 있습니다.

### Adapter 기본 클래스

Adapter는 Mech-Viz, Mech-Vision, Mech-Center 및 Robserver 관련 호출을 캡슐화하였으며 Mech-Viz 시작, Mech-Viz 중지, Mech-Vision 또는 Mech-Viz스텝 파라미터 설정 및 Mech-Vision 식별 시작과 같은 기능을 포함합니다. Adapter 프로그램이 Mech-Viz 또는 Mech-Vision을 호출하는 한 Adapter 클래스를 상속해야 합니다.

Adapter 클래스 속성은 아래 표에 나와 있습니다.

클래스 속성	설명
viz_project_dir	현재 Mech-Viz 프로젝트의 경로
vision_project_name	현재 Mech-Vision 프로젝트의 명칭
is_simulate	Mech-Viz 시뮬레이션 실행 여부
is_keep_viz_state	Mech-Viz가 마지막으로 중지했을 때의 실행 상태를 유지할지 여부
is_save_executor_data	Mech-Viz 이그제큐터의 데이터 저장 여부
is_force_simulate	Mech-Viz 시뮬레이션을 강제로 실행할지 여부
is_force_real_run	Mech-Viz를 실제로 강제 실행할지 여부
code_signal	Mech-Center 메인 인터페이스에 Adapter 정보의 신호를 표시(오류 코드가 있는 정보)
msg_signal	Mech-Center 메인 인터페이스에 Adapter 정보의 신호를 표시(오류 코드가 없는 정보)
i_code_signal	Mech-Center 메인 인터페이스에 Mech-Interface 정보의 신호를 표시(오류 코드가 있는 정보)
i_msg_signal	Mech-Center 메인 인터페이스에 Mech-Interface 정보의 신호를 표시(오류 코드가 없는 정보)
viz_finished_signal	Mech-Viz 실행 종료 신호(정상 종료 또는 비정상 종료)
connect_robot_signal	로봇 신호 연결/해제
start_adapter_signal	Adapter 신호 시작
service_name_changed	Mech-Center 메인 인터페이스에 Mech-Viz 및 Mech-Vision 상태의 신호를 표시
setting_infos	Mech-Center에 대한 구성 정보
service_name	등록된 서비스 이름

Adapter 클래스 함수는 아래 표에 나와 있습니다.

클래스 함수	설명
on_exec_status_changed()	Mech-Viz 및 Mech-Vision에서 상태 정보 수신
register_self_service()	Adapter 서비스 등록
vision_project_dirs(self):	Mech-Vision 프로젝트 폴더 경로 조회
vision_project_names()	모든 Mech-Vision 프로젝트 명칭 조회

클래스 함수	설명
vision_project_names_in_center()	Mech-Center에서 로드된 모든 Mech-Vision 프로젝트 명칭 조회
is_viz_registered()	Mech-Viz 등록 여부 판단
is_viz_in_running()	Mech-Viz가 실행 중인지 판단
is_vision_started()	Mech-Vision 프로젝트 등록 여부 판단
find_services()	서비스 찾기
before_start_viz()	Mech-Viz 작동 전에 호출되는 함수
after_start_viz()	Mech-Viz 작동 후 호출되는 함수
viz_not_registerd()	Mech-Viz 작동 후 Mech-Viz가 등록되지 않은 것으로 확인되면 이 함수가 호출됨
viz_is_running()	Mech-Viz 작동 후 Mech-Viz가 실행 중인 것으로 확인되면 이 함수가 호출됨
viz_run_error()	Mech-Viz 작동 후 Mech-Viz가 실행 중에 오류가 발생하면 이 함수가 호출됨
viz_run_finished()	Mech-Viz 실행이 완료되면 호출되는 함수
viz_plan_failed()	Mech-Viz 계획 실패 시 호출하는 함수
viz_no_targets()	Mech-Viz 계획 시 웨이포인트가 없을 때 호출하는 함수
viz_unreachable_targets()	Mech-Viz 계획 시 도달할 수 없는 웨이포인트가 있을 때 호출되는 함수
viz_collision_checked()	Mech-Viz 계획이 충돌을 감지하면 호출되는 함수
parse_viz_reply()	Mech-Viz의 답장 해석
wait_viz_result()	Mech-Viz의 답장 대기
start_viz()	Mech-Viz를 실행하기
stop_viz()	Mech-Viz를 정지하기
pause_viz()	Mech-Viz를 중지하기
find_vision_pose()	Mech-Vision 프로젝트를 트리거하여 사진 캡처하기
async_call_vision_run()	Mech-Vision 프로젝트를 비동기 트리거하여 사진 캡처하기
async_get_vision_callback()	Mech-Vision의 결과를 비동기적으로 수신하기
deal_vision_result()	Mech-Vision의 결과를 처리하기
set_step_property()	Mech-Vision의 스텝 파라미터를 설정하기
read_step_property()	Mech-Vision의 스텝 파라미터를 읽어내기

클래스 함수	설명
select_parameter_group()	Mech-Vision 프로젝트에서 레시피 템플릿을 선택하기
set_task_property()	Mech-Viz의 스텝 파라미터를 설정하기
read_task_property()	Mech-Viz의 스텝 파라미터를 읽어내기
get_digital_in()	DI를 획득하기
set_digital_out()	DO를 설정하기
before_start_adapter()	Adapter를 시작하기 전에 이 함수를 호출
start()	Adapter를 시작
close()	Adapter를 닫기
handle_command()	수신된 외부 명령 처리

### TcpServerAdapter 클래스

TcpServerAdapter 클래스는 Adapter에서 상속하고 아래와 같이 TcpServer의 기능을 캡슐화합니다.

```
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 클래스

TcpClientAdapter 클래스는 Adapter에서 상속하고 아래와 같이 TcpClient의 기능을 캡슐화합니다.

```

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_recv_size(self, size):
    self._client.set_recv_size(size)

def send(self, msg, is_logging=True):
    self._client.send(msg, is_logging)

def recv(self):
    return self._client.recv()

def start(self):
    self.reconnect_server(False)
    while not self.is_stop_adapter:
        try:
            self._client.before_recv()
            cmds = self._client.recv()
            if not cmds:
                self.reconnect_server()
                continue
            logging.info("Received command from server:{}".format(cmds))
            self._client.after_recv()
        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 클래스

TcpMultiplexingServerAdapter 클래스는 Adapter에서 상속되며 주로 아래와 같이 여러 클라이언트를 연결하는 데 사용됩니다.

```

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():
        available_sockets = list(self.sockets.values()) + list(self._servers.values())
        rs, _, _ = select(available_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 클래스

IOAdapter 클래스는 Adapter에서 상속하고 아래와 같이 루프(loop)에서 DI를 얻는 작업을 캡슐화합니다.

```

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 클래스

AdapterWidget 클래스는 Adapter 사용자 인터페이스를 사용자 지정하기 위한 상위 클래스이며 사용자 인터페이스를 사용자 지정하기 위한 모든 기능은 아래와 같이 이 클래스에서 상속해야 합니다.

```

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

Service 관련 클래스의 소스 파일은 Mech-Mind 시스템 소프트웨어 설치 디렉터리의 [Mech-Center/src/interface/communication.py](#) 파일에 있습니다.

### NotifyService 클래스

NotifyService의 클래스는 아래와 같습니다.

```
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)
```

기본 서비스 이름은 adapter이며, 프로젝트에 여러 알림 서비스가 필요한 경우 서브 클래스에서 service\_name을 다시 작성하여 서로 다른 서비스를 구분할 수 있습니다. 클래스 함수에 관한 설명은 아래와 같습니다.

클래스 함수	설명
handle_message()	인터페이스 함수, 하위 클래스는 이 함수를 다시 작성하고 이 함수에서 논리를 구현할 수 있음
notify()	메시지 구문 분석을 제공하며 하위 클래스는 일반적으로 다시 작성할 필요가 없음

### VisionResultSelectedAtService 클래스

VisionResultSelectedAtService 클래스는 다음과 같습니다.

```
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))
    
```

기본 서비스 유형은 vision\_watcher로 다른 유형으로 변경할 수 없으며 기본 명칭은 vision\_watcher\_adapter이며, 프로젝트에 여러 개의 vision\_watcher 서비스가 필요한 경우 하위 클래스에서 service\_name을 다시 작성하여 서비스를 구분할 수 있습니다. 클래스 함수에 관한 설명은 아래와 같습니다.

클래스 함수	설명
poses_found()	인터페이스 함수, 하위 클래스는 이 함수를 다시 작성하고 이 함수에서 논리를 구현할 수 있으며 파라미터는 Mech-Vision의 인식 결과임
posesFound()	Mech-Vision에서 인식한 문자를 분석하고 하위 클래스는 일반적으로 다시 작성할 필요가 없음
poses_planned()	인터페이스 함수, 파라미터는Mech-Viz 계획 시 선택한 웨이포인트
posesPlanned()	Mech-Viz 계획 시의 메시지 구문 분석을 지원

### RobotService 클래스

RobotService 클래스는 다음과 같습니다.

```

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
    
```

기본 서비스 유형은 robot으로 다른 유형으로 변경할 수 없으며, 기본 이름은 robot으로 하위 클래스에서 해당 로봇 이름으로 변경해야 합니다. jps 또는 pose 값은 하위 클래스에서 설정해야 하며 이는 Mech-Viz 실행 중 포즈를 고정하는 데 사용됩니다. 여기서 이 포즈는 전체 경로에서 시나리오와 충돌하지 않아야 합니다. 클래스 함수에 관한 설명은 아래와 같습니다.

클래스 함수	설명
getJ()	Mech-Viz / Mech-Vision에 관절 각도 반환
setJ()	관절 각도를 외부에서 설정, 단위는 라디안
getL()	Mech-Viz / Mech-Vision에 TCP 포즈 반환
getFL()	Mech-Viz / Mech-Vision에 플랜지 포즈 반환
setL()	외부에서 플랜지 포즈를 설정(사원수 형식), 단위는 미터
moveXs()	Mech-Viz 경로를 계획한 후 이 함수가 호출되며 파라미터에는 웨이포인트의 속성이 포함됨. 참고: Mech-Viz 프로젝트에 사전 계획한 스텝을 방해하는 DI, 분기 등이 있는 경우 Mech-Viz는 이 함수를 여러 번 호출함
stop()	로봇을 정지, 일반적으로 사용하지 않음
setTcp()	TCP 설정, 일반적으로 사용하지 않음
setDigitalOut()	DO 설정, 일반적으로 사용하지 않음
getDigitalIn()	DI 설정, 일반적으로 사용하지 않음

클래스 함수	설명
switchPauseContinue()	로봇 일시 중지/계속 실행, 일반적으로 사용하지 않음

### OuterMoveService 클래스

OuterMoveService 클래스는 다음과 같습니다.

```

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}
    
```

기본 서비스 이름은 adapter이며, 프로젝트에 여러 outer\_move 서비스가 필요한 경우 하위 클래스에서 service\_name을 다시 작성하여 서로 다른 서비스를 구분할 수 있습니다. 클래스 함수에 관한 설명은 아래와 같습니다.

클래스 함수	설명
move_target_type()	웨이포인트 유형, 0:jps, 1:tcp_pose, 2:obj_pose
velocity()	웨이포인트 속도, 기본값은 0.25
acceleration()	웨이포인트 가속도, 기본값은 0.25
blend_radius()	웨이포인트 회전 반경, 기본값은 0.05m
motion_type()	웨이포인트 이동 유형, 'J':moveJ, 'L':moveL
is_tcp_pose()	웨이포인트가 TCP인지 여부
gather_targets()	인터페이스 함수, 모든 웨이포인트를 수집하고 파라미터는 이때 로봇의 관절 각도, 플랜지 포즈 및 DI값이며 필요에 따라 서브클래스를 판단하고 수정할 수 있음
add_target()	단일 웨이포인트 추가, 이 함수는 하위 클래스에서 호출되어 웨이포인트를 추가할 수 있음
getMoveTargets()	Mech-Viz 외부로 이동 실행 시 이 함수가 호출되며 파라미터는 이때 로봇의 관절 각도, 플랜지 포즈 및 DI값을 포함함

#### 등록 서비스

위 4가지 항목에 해당하는 서비스는 회원가입을 해야만 이용이 가능하며, 회원가입 서비스 기능은 다음과 같습니다.

```

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 패키지

Adapter util 패키지는 Mech-Center 소프트웨어 설치 경로의 /src/util 폴더에 위치하며 많은 모듈을 포함하고 일반적인 함수를 제공합니다. 프로그래밍 과정에서는 util 패키지의 특정 기능이 이미 구현되어

있는지 확인하십시오. 기능이 구현되어 있으면 직접 사용하면 되고, 구현되어 있지 않고 일반적인 경우 작은 함수로 추상화하여 util 패키지에 추가할 수 있습니다.

다음은 각 모듈에 대한 간단한 소개입니다.

#### database 모듈

database 모듈은 database에 관한 조작을 제공합니다. Mech-Center는 실행될 때 기본적으로 mechmind.db 데이터 베이스 파일을 만들며 이 파일은 실행 중에 로그를 저장하는 데 사용됩니다. database 모듈은 SQL 문을 실행하고 하나 또는 모든 레코드를 조회하는 함수를 제공합니다.

#### json\_keys 모듈

json\_keys 모듈에는 Mech-Center에 사용하는 json 키/값 문자열이 저장되어 있으며 다른 모듈에서 직접 가져와 사용할 수 있습니다.

#### message\_box 모듈

message\_box 모듈은 팝업 창 알림 기능을 제공합니다. 팝업 창 알림 유형에는 information, warning 및 critical이 포함됩니다.

#### timestamp 모듈

timestamp 모듈은 현재 타임스탬프를 반환하는 기능을 제공합니다.

#### transforms 모듈

transforms 모듈은 오일러 각도를 사원수로 변환, 사원수를 오일러 각도로 변환, 포즈 곱셈, 물체 포즈를 TCP로 변환, TCP를 물체 포즈로 변환, 물체 회전을 계산하는 등의 함수를 제공합니다. 서드파티 라이브러리 transforms3d에서도 오일러 각도를 사원수로 변환하고 사원수를 오일러 각도로 변환하는 함수를 제공하지만 실제 사용 중 일부 경우 transforms3d 변환 값이 잘못될 수도 있습니다. 실제 계산 시 transforms3d 라이브러리를 우선 사용하고 결과가 올바르지 않으면 transforms 모듈에서 제공하는 사용자 정의 변환 함수를 사용할 수 있습니다.

#### util\_file 모듈

util\_file 모듈은 파일 읽기 및 쓰기 기능을 제공하며 일반적으로 사용되는 json 파일 읽기 및 쓰기가 포함됩니다.

#### timer 모듈

timer 모듈은 편리한 타이머 클래스를 제공합니다. 타이머 기능이 필요한 경우 Timer 객체를 생성하고 콜백 함수를 전달하여 start()를 호출하면 됩니다. 사용 후 Timer 객체를 없앨 필요가 없으며 프로그램이 종료되면 자동으로 없어집니다.

#### pose 모듈

pose 모듈은 Mech-Viz와 포즈 표현이 동일한 클래스를 제공하며 이동 (미터 단위)과 회전 (사원수 형식)을 포함하며 역 및 곱셈 작업을 수행할 수 있으며 list에서 변환하거나 list로 변환할 수 있습니다. 또한 pose 모듈은 포즈의 단위 변환 함수 몇 가지를 제공하며, 밀리미터를 미터로, 미터를 밀리미터로, 라디안으로, 도를 라디안으로, 사원수를 오일러 각도로, 오일러 각도를 사원수로 변환하는 것입니다.

### 6.5.3.4. 인터페이스를 획득하기

#### Mech-Viz 프로젝트에 사용되는 스텝을 획득하기

Mech-Viz 프로젝트에 사용되는 스텝을 획득하기 위한 함수는 아래와 같습니다.

```
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()를 호출한 후 json 형식의 문자열을 반환해 획득한 모든 스텝을 나타냅니다.

#### Mech-Viz 또는 Mech-Vision 프로젝트에 사용되는 파라미터를 획득하기

Mech-Viz 또는 Mech-Vision 프로젝트에 사용되는 파라미터를 획득하기 위한 함수는 아래와 같습니다.

```
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("{} property result: {}".format("Viz" if get_viz else "Vision", json.loads
    (result)))
    return result
```

함수를 호출할 때 msg 파라미터 중의 “type”을 지정하지 않으면 모든 파라미터를 획득합니다. 지정하면 지정된 파라미터만 획득합니다. 예를 들어 get\_property\_info(msg={"type": "move"}) 를 호출한 후 json 형식의 문자열을 반환해 획득한 이동 스텝의 파라미터를 나타냅니다.

#### 6.5.3.5. Mech-Vision 인터페이스

이 부분에서 아래와 같이 Mech-Vision과 관련된 인터페이스를 소개하겠습니다:

##### 비전 목표점을 획득하기

아래와 같이 adapter.py 중의 find\_vision\_pose()함수를 통해 Mech-Vision비전 결과를 획득합니다.

```
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 중의 비전 목표점은 보통 사원수 형식으로 출력된 물체 포즈(obj\_pose)입니다. TCP 포즈 형식으로 출력될 수도 있지만 Mech-Vision 프로젝트에서 변환되어야 합니다.

##### 예시

Adapter는 TCP 포즈(tcp\_pose)로 변환할 수 있는 함수를 생성해야 하며 때로는 사원수를 오일러 각도로 변환하거나 라디안을 각도로 변환하고 로봇 측에 전송해야 합니다 (변환 여부는 로봇 측과 일치해야 함). 또는 Adapter는 Mech-Vision에서 출력한 비전 포즈를 체크할 수도 있습니다. 예를 들어:

```
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", []) # Get pose
if len(poses) = 0: # 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
    
```

여기서 vision\_result는 find\_vision\_pose()에서 가져온 것이며, 다음과 같이 호출됩니다.

```
self.check_vision_result(json.loads(self.find_vision_pose().decode()))
```

vision\_result가 함수에 전송되면 ROI와 관련된 프로젝트에 대해 먼저 비어있는 빈인지 확인한 다음 비전 포즈를 얻습니다. 비전 포즈가 정상인 경우 비전 포즈를 변환 함수(pack\_pose())로 전달하고 전송합니다.

### 스텝 파라미터를 설정하기

Mech-Vision의 스텝에 대한 동적 파라미터를 설정할 때는 일반적으로 adapter.py의 set\_step\_property()를 호출하기만 하면 됩니다.

```

def set_step_property(self, msg, project_name=None, timeout=None):
    return self.call_vision("setStepProperties", msg, project_name, timeout)
    
```

여기서 msg는 구체적인 스텝의 이름과 구성해야 하는 파라미터를 결정합니다.

### 예시

공작물 종류에 따라 동적으로 Mech-Vision 매칭 모델을 설정해야 하는 경우, 다음과 같은 함수를 생성하여 msg를 설정할 수 있습니다.

```

def _step_matching_model_cell(step_name, model_type):
    msg = {"name": step_name,
          "values":
            {"modelFile": model_type["ply"],
             "pickPointFilePath": model_type["json"]}}
    return msg
    
```

여기서 step\_name은 설정해야 하는 Mech-Vision 스텝의 이름이며 model\_type은 해당 종류의 공작물과 대응하는 파일 경로입니다. "ply" 및 "json"을 통해 .ply 및 .json으로 끝나는 모델 파일의 경로를 획득하여 Mech-Vision에서 각각 해당 스텝의 파라미터에 입력할 수 있습니다.

step\_name은 "Local Matching"인 경우 다음과 같이 호출하면 됩니다.

```

msg = _step_matching_model_cell("Local Matching", model_type)
self.set_step_property(msg)
    
```

## 스텝 파라미터를 읽어내기

특정한 Mech-Vision 스텝의 파라미터를 획득하려면 adapter.py의 read\_step\_property()을 호출할 수 있습니다.

```
def read_step_property(self, msg):
    result = self.call_vision("readStepProperties", msg)
    logging.info("Property result: {}".format(result))
    return result
```

여기서 msg는 스텝의 이름과 획득할 파라미터의 값을 결정하며, 함수를 생성하여 msg를 다시 작성할 수 있습니다.

## 예시

카메라 IP를 획득하려면 다음과 같은 예제 코드를 사용할 수 있습니다.

```
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 프로젝트에 카메라가 하나만 있는 경우 유형(type)에 따라 스텝을 찾을 수 있습니다. 만약 Mech-Vision에 동일한 스텝이 여러 개 있지만 특정 스텝의 파라미터만 획득하거나 설정하려면 이름(name)으로 스텝을 찾을 수 있습니다. read\_step\_property() 함수를 호출하고 json 형식으로 변환하여 카메라의 모든 파라미터 값(json 형식)을 획득할 수 있습니다. 다음과 같습니다:

```
Property result:
{
  "MechEye": {
    "NetCamIp": "127.0.0.1",
    "TimeOut": "10",
    "configGroup": "",
  }
}
```

이 예에서 구체적인 파라미터의 필드("MechEye" 및 "NetCamIp")에 따라 최종적으로 카메라의 IP 주소(127.0.0.1)를 획득합니다.

## 파라미터 레시피를 전환하기

Mech-Vision을 사용할 때 일부 프로젝트의 프로세스가 동일하지만 일부 파라미터가 다른 경우, 파라미터 레시피를 구성하면 다른 프로젝트에 대해 해당하는 파라미터를 전환할 수 있습니다. 즉, adapter.py의 select\_parameter\_group()을 호출합니다.

```
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
```

여기서 `project_name` 파라미터는 Mech-Vision 프로젝트의 이름이며 `group_index` 파라미터는 레시피의 번호입니다.

## 예시

프로젝트에서 레시피를 전환해야 하는 경우 다음과 같은 예제 코드를 사용하여 `select_parameter_group()` 함수를 호출하고 예외 처리를 수행하십시오.

```
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
```

여기서 `self.vision_project_name`은 Mech-Vision 프로젝트의 이름이고 `model_code-1`은 레시피의 번호입니다.

### 6.5.3.6. Mech-Viz 인터페이스

이 부분에서 아래와 같이 Mech-Viz와 관련된 인터페이스를 소개하겠습니다:

#### Mech-Viz를 시작하기

`adapter.py` 파일의 `Adapter` 클래스에서 Mech-Viz를 시작하는 함수가 이미 정의되었으므로 `start_viz()`를 직접 호출할 수 있습니다. 또한 Mech-Viz를 시작하기 전/후의 작업을 자체적으로 설정하기 위해 프로젝트의 기능에 따라 `self.before_start_viz()` and `self.after_start_viz()` 내용을 다시 작성할 수 있습니다.

#### 함수 정의

```
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()는 기본적으로 새로운 스레드에서 Mech-Viz의 실행 과정이 완료될 때까지 기다립니다. 이 목적은 Mech-Viz 시작 이외의 다른 작업에 영향을 미치지 않도록 하는 것입니다.

다음으로 스텝의 파라미터 동적 설정을 예시로 self.before\_start\_viz()를 다시 작성하는 방법에 대해 소개하겠습니다.

```

def before_start_viz(self):
    self.set_move_offset(x, y, z)
    
```

Mech-Viz를 시작하기 전에 읽어낸 데이터에 따라 웨이포인트의 X,Y, Z 방향에서의 오프셋을 구성해야 합니다.

### Mech-Viz를 정지하기

adapter.py 파일의 Adapter 클래스에서 Mech-Viz를 정지하는 함수가 이미 정의되었으므로 stop\_viz()를 직접 호출할 수 있습니다.

### 함수 정의

```

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-Viz를 중지/계속 실행하기

adapter.py 파일의 Adapter 클래스에서 Mech-Viz를 중지/계속 실행하는 함수가 이미 정의되었으며 기능은 Mech-Viz 소프트웨어의 중지 버튼과 동일하여 시뮬레이션할 때만 사용될 수 있습니다.

#### 함수 정의

```
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)
```

### 스텝 파라미터를 설정하기

Mech-Viz의 스텝에 대한 동적 파라미터를 설정할 때는 일반적으로 Adapter 클래스의 set\_task\_property()를 호출하기만 하면됩니다.

#### 함수 정의

```
def set_task_property(self, msg, timeout=None):
    return self.call_viz("setTaskProperties", msg, timeout)
```

msg는 다양한 스텝에 대해 다른 파라미터를 구성하는 것을 결정합니다.

### 고정점 이동

Mech-Viz 실행 과정에서 “고정점 이동” 스텝 중 X,Y, Z의 오프셋 값을 조정해야 할 경우, Mech-Viz를 컨트롤하는 메인 프로그램에서 다음과 같은 함수를 작성할 수 있습니다.

#### 예시

```
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: “고정점 이동” 스텝의 이름. UNIT\_PER\_METER=1000, 일반적으로 x\_offset, y\_offset 및 z\_offset 데이터의 단위가 mm이며 Mech-Viz 중의 데이터의 단위가 m일 때 UNIT\_PER\_METER를 통해 단위를 전환합니다.

다음과 같은 방식으로 set\_move\_offset() 함수를 호출하면 Mech-Viz의 move\_1 스텝의 X, Y, Z 오프셋이

해당 변경 사항이 발생합니다.

```
self.set_move_offset("move_1", 100, 200, 300)
```

### 순서대로/배열대로 이동

순서대로/배열대로 이동은 일반적으로 Mech-Viz에서 미리 편집한 다음 Adapter가 로직에 따라 시작 인덱스를 수정하여 사용합니다. 사용 방법은 [고정점 이동](#), [팔레타이징](#)과 동일합니다.

### 외부 이동

여러 외부 목표 포즈를 Mech-Viz에 전송하여 계획 및 컨트롤할 필요가 있는 경우 “외부 이동” 스텝을 통해 구현할 수 있습니다. “외부 이동” 스텝은 JPs, TCP, 물체 포즈를 설정하는 데 사용됩니다. 사용법은 다음과 같습니다.

### 예시

```
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-Viz는 “외부 이동” 스텝까지 실행되면 getMoveTargets()를 호출합니다. 서비스 이름을 통해 서로 다른 외부 이동 스텝을 구분합니다.

```
def _register_service(self):
    self.outer_move_service = CustomOuterMoveService()
    self._outer_move_server, port = register_service(outer_move_service, port)
```



“외부 이동” 스텝 앞에는 피킹 스텝이 있어야 합니다. 그렇지 않으면 "물체를 잡고 있지 않을 때 물체 포즈가 무효합니다!" 오류 알람이 발생합니다.

### 팔레타이징

Mech-Viz 실행 과정에서 다른 팔레타이징 스텝에 따라 다른 파라미터를 설정해야 할 때, 해당 스텝의 이름을 통해 구체적인 스텝을 찾아야 합니다. 프로젝트 편집 영역에서 해당 스텝을 선택하면 Mech-Viz 파라미터 편집 영역에 나타나는 파라미터를 수정할 수 있습니다.

### 예시

예를 들어 “자체 정의한 파렛트 패턴” 스텝인 경우, 일반적으로 변경해야 하는 파라미터 값은 **시작 인덱스\*와 \*파일 이름**(\*동적 로딩\*을 선택해야 \*파렛트 파일 경로\*가 나타남)입니다. 따라서 다음과 같은 함수를 메인 프로그램에서 정의할 수 있습니다.

```
def set_stack_pallet(self, name, startIndex, fileName):
    msg = {
        "name": name,
        "values": {
            "startIndex": startIndex,
            "fileName": fileName,
```

```

    }
}
self.set_task_property(msg)
    
```

여기서 "startIndex"는 \*시작 인덱스\*를 나타내고 "fileName"은 \*파일 이름\*을 나타냅니다. "startIndex" 및 "fileName" 파라미터의 이름은 Mech-Viz에서 정의됩니다.

다음 방법을 사용하여 set\_stack\_pallet() 함수를 호출합니다.

```
self.set_stack_pallet("common_pallet_1", 2, "re.json")
```

“미리 설정된 팔레트 패턴” 스텝의 경우, 일반적으로 변경해야 하는 파라미터 값은 **시작 인덱스, 팔레트 패턴, 상자 길이, 상자 너비, 상자 높이, 행 수, 열 수 및 \*층 수\***입니다. 따라서 다음과 같은 함수를 메인 프로그램에서 정의할 수 있습니다.

```

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)
    
```

설정해야 할 파라미터가 많은 경우, 일반적으로 파라미터를 Excel 파일에 작성한 다음 Excel 파일에서 데이터를 읽고 self.box\_data\_info에 기록합니다. 그런 다음 stack\_type 값을 사용하여 나중에 참조할 수 있습니다. "startIndex", "palletType", "cartonLength" 등의 이름은 Mech-Viz에서 정해진 것입니다.

**자체 정의한 팔레트 패턴 및 미리 설정된 팔레트 패턴** 클래스의 msg 값을 비교하면 "values" 값이 다르다는 것을 쉽게 알 수 있습니다. 특정 스텝의 파라미터를 설정해야 하는 경우 해당 파라미터의 이름과 값을 "values"에 추가하기만 하면 됩니다. 기타 팔레타이징 스텝의 파라미터 설정에 대해서는 위에서 설명한 예제를 참조하십시오.

## 메시지 분기

Mech-Viz는 “메시지 분기” 스텝을 실행할 때, 외부 신호(Adapter)가 아웃 포트를 설정할 때까지 계속

기다립니다. “메시지 분기” 스텝에 대해서 다음과 같은 함수를 정의하면 분기를 컨트롤할 수 있습니다.

### 예시

```
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)
```

여기서 name은 스텝의 이름이고, area는 아웃 포트의 번호이며 왼쪽에서 오른쪽으로 각각 0, 1, 2...입니다. 예를 들어 이 스텝은 가장 왼쪽의 아웃 포트를 통해 실행되면 area 값은 0입니다. Mech-Viz를 시작하지 않고 직접 “메시지 분기” 스텝을 호출하면 Mech-Viz에서 “스텝 파라미터를 설정할 때 이그제큐터가 없습니다!”라는 오류 알람이 발생합니다.

### 카운터

“카운터” 스텝을 사용할 때 일반적으로 카운트 총수와 현재 카운트를 설정해야 합니다. 해당 함수를 정의하는 코드가 다음과 같습니다.

### 예시

```
def set_counter_property(self, name, count, curCount):
    msg = {"name": name,
          "values": {"count": count, "currentCount": curCount}}
    self.set_task_property(msg)
```

해당 함수를 호출하는 예시가 다음과 같습니다.

```
self.set_counter_property("counter_1", 5, self.success_stack_num)
```

self.success\_stack\_num은 성공적으로 적재된 팔레트의 수량을 나타냅니다. 팔레타이징을 수행하는 과정에서 상자가 떨어지고 작업자의 수동 개입으로 Mech-Viz가 정지되면, Mech-Viz 중 "counter\_1"로 명명된 카운트 스텝에 "currentCount"의 값을 저장하지 않습니다. Mech-Viz를 재시작하고 self.success\_stack\_num을 통해 이 스텝의 현재 카운트 수를 다시 설정할 수 있습니다.

### 스텝 파라미터를 읽어내기

Mech-Viz를 실행할 때 특정 스텝의 파라미터를 읽어내려면 메인 프로그램에서 다음과 같은 함수를 정의할 수 있습니다.

### 예시

```
def read_move_pallet(self, name):
```

```
msg = {"name": name,
       "properties": ["xOffset", "yOffset", "zOffset", ]}
return read_task_property(msg)
```

그 중 "name"을 통해 읽어야 할 스텝의 이름을 찾을 수 있습니다. "properties" 뒤의 목록에 있는 값은 필요에 따라 파라미터를 추가하거나 제거할 수 있습니다. 이 예시에서는 "xOffset", "yOffset" 및 "zOffset" 값을 읽으므로 이 세 가지 값을 "properties"에 추가하고 다음과 같이 호출합니다.

```
self.read_move_pallet("move_3")
```

호출한 후 다음과 같은 결과를 얻었습니다.

```
{'zOffset': -0.23, 'xOffset': -0.12, 'yOffset': -0.15}
```

하지만 주의해야 할 것은 Mech-Viz 스텝에서 획득한 파라미터의 일부 값은 계획된 파라미터 값이고 실제 실행 시와 다를 수도 있습니다. 계획 진행 상태는 실행 진행 상태보다 미리 진행됩니다(즉 Mech-Viz는 미래의 이동을 미리 계획할 수도 있음). 여기서는 “팔레타이징” 스텝의 현재 인덱스(curIndex) 파라미터 값을 예로 들어 설명하며, 사용자는 메인 프로그램에서 다음과 같은 함수를 정의할 수 있습니다.

```
def read_pallet_current_index(self, name):
    msg = {"name": name,
          "properties": ["curIndex"]}
    return read_task_property(msg)
```

다음과 같이 함수를 호출하여 curIndex의 값을 획득합니다.

```
self.read_pallet_current_index("common_pallet_1")
```

호출한 후 다음과 같은 결과를 얻을 수 있습니다.

```
{'curIndex': 5}
```

여기서의 5는 현재 계획이 5 라운드를 완료했음을 나타냅니다. 예를 들어 상자 배치 프로젝트의 경우, 5는 계획 진행 상황에서 5 개의 상자를 배치했음을 나타내며, 로봇이 실제로 배치한 상자는 5 개보다 적을 수 있습니다. 따라서 "curIndex"와 같은 파라미터의 값은 실행 시 값이 아니라 계획된 값을 나타냅니다.

### 말단장치를 바꾸기

Mech-Viz 말단장치 리스트에서 해당 인덱스를 지정하면 됩니다. Mech-Viz 말단장치 리스트의 인덱스는 위에서 아래로 0, 1, 2, 3 등으로 순서대로 지정됩니다. 범위를 벗어나지 않도록 주의해야 합니다. 구체적인 함수는 아래와 같습니다.

### 예시

```
def set_tcp(self, index):
    msg = {"function": "setTcp", "index": index}
```

```
self.call("executor", msg)
```

### 실행 시 전역 속도를 설정하기

Mech-Viz 실행 중에 로봇 실행 속도 백분율을 동적으로 조정해야 하는 경우 다음 함수를 메인 프로그램에 작성할 수 있습니다.

### 예시

```
def set_vel(self, vel_scale):
    msg = {"function": "setConfig",
          "velScale": vel_scale / 100, "accScale": vel_scale / 100}
    self.call("executor", msg)
```

속도를 80%로 설정하면 다음과 같이 함수를 호출할 수 있습니다.

```
self.set_vel(80)
```



이 함수는 Mech-Viz가 시작된 후에만 호출할 수 있습니다. 그렇지 않으면 오류가 발생합니다. 즉, 이 모듈의 호출 조건과 “메시지 분기” 스텝이 일치합니다. 따라서 일반적으로 “메시지 분기” 스텝에서 `set_vel()` 함수를 호출하여 새 스레드에서 `set_vel()` 함수를 호출하는 것을 피합니다. 다음과 같습니다.

```
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)
```

### 포인트 클라우드의 충돌 파라미터를 설정하기

포인트 클라우드 충돌 파라미터를 설정하려면 Mech-Viz에 해당하는 인터페이스인 `setConfig()`를 사용하면 됩니다. 실행 시 전역 속도 설정 인터페이스와 마찬가지로 설정 정보가 다를뿐입니다. 다음과 같습니다.

### 예시

```
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 반환값

Mech-Viz의 반환값 리스트는 다음과 같습니다.

반환값	설명
Finished	정상적으로 실행되었습니다. 설명: Mech-Viz 프로젝트에서 비전 이동의 오른쪽 분기에 다른 스텝과 연결될 때 Mech-Viz도 정상적으로 반환됩니다.
Command Stop	정지 또는 stop_viz () 함수 호출합니다.
No targets	비전 포즈가 없음을 나타냅니다. 즉, Mech-Vision이 빈 값을 반환합니다.
No proper vision poses	비전 포즈에 도달할 수 없으며 로봇이 현재 인식된 물체의 위치에 도달할 수 없거나 충돌합니다.
PlanFail	Mech-Viz 경로 계획 실패.
SceneCollision	충돌이 발생합니다.

Mech-Viz 반환 상황에 대해 Adapter 기본 클래스는 해당 인터페이스 함수를 제공합니다.

### 6.5.3.7. RobotService 인터페이스

이 부분에서 아래와 같이 RobotService와 관련된 인터페이스를 소개하겠습니다.

#### RobotServer 시뮬레이션 서비스

RobotServer 서비스 시뮬레이션은 일반적으로 로봇을 마스터 컨트롤하는 데 사용되지만 RobotServer를 통해 로봇을 제어할 수 없고 RobotService를 생성하여 RobotServer를 시뮬레이션합니다. RobotService가 Mech-Center에 등록되면 Mech-Viz는 실제 로봇과 연결하는 것과 같은 방식으로 RobotService와 상호 작용합니다.

RobotServer서비스 시뮬레이션의 주요 응용 시나리오는 아래와 같습니다.

- Mech-Vision 계산을 위해 로봇의 관절 각도를 획득합니다 (Eye In Hand).
- 실제 로봇에서 포즈를 획득하여 RobotService 서비스로 보냅니다.

#### 예시

Adapter 하위 클래스에서 호출하는 방식은 아래와 같습니다.

```
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])
    
```

RobotService의 클래스는 아래와 같습니다.

```

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()는 아래와 같이 주로 Mech-Viz와 Mech-Vision이 관절 각도를 획득하는 데 사용됩니다. 일반적으로 setJ()를 통해 현재 관절 각도를 먼저 설정한 다음 getJ()를 호출합니다.

## 예시

```

def getJ(self, *_):
    pose = {"joint_positions": self._jps}
    return pose
    
```

1. Eye In Hand : 로봇에서 보내 온 관절 각도를 setJ()에 입력합니다.

```

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))
    
```

JPs는 로봇이 보내는 관절 각도 데이터이며, 이를 self.\_jps에 할당하여 getJ()에서 호출됩니다. getJ()는 라디안 형식의 데이터를 획득해야 하므로 여기서 단위 변환에 유의해야 합니다.

2. Eye To Hand: 로봇의 현재 관절 각도에 따라 설정할 필요는 없지만 실제 로봇 상태를 시뮬레이션하려면 안전한 웨이포인트를 원점 위치(보통 Home 포인트)로 설정해야 합니다. 그렇지 않으면 임의의 숫자가 할당되어 오류가 발생하기 쉽습니다.

```

def getJ(self, *_):
    return {"joint_positions": [1.246689, -0.525868, -0.789761, -1.330814, 0.922581, 4.364021]}
    
```

## getFL

getFL() 함수는 이미지를 캡처할 때의 플랜지 포즈를 제공하는 데 사용됩니다. Eye In Hand 모드에서 진행되는 캘리브레이션은 플랜지와 카메라의 상대적인 위치 관계를 설명하지만 결국 베이스 좌표계 아래의 데이터를 사용하기 때문에 이미지를 캡처했을 때의 플랜지 포즈를 제공해야 합니다.

```

def getFL(self, *_):
    return {"flange_pose": self.pose}
    
```

Eye In Hand 방식을 통해 포즈를 제공할 때 다음과 같은 사항을 주의해야 합니다:

1. 로봇이 이미지를 캡처했을 때의 JPs를 반환하면 RobotService에서 직접 setJ()함수(단위: 라디안)를 호출하면 됩니다. 이 함수는 []를 반환할 것입니다.
2. 로봇이 플랜지 포즈를 반환하면 다음과 같이 처리하십시오:
  - Mech-Vision 프로젝트 중의 외부 파라미터 파일 extri\_param.json에 있는 파라미터 "is\_eye\_in\_hand"의 값은 true로 설정해야 합니다.
  - RobotService의 setFL()를 호출하십시오(단위: 미터, 포즈 유형: 사원수).

## moveXs

Adapter를 통해 생성한 RobotService 서비스는 moveXs()를 사용하여 Mech-Viz가 계획한 웨이포인트를 수신합니다. 구체적으로 아래와 같습니다.

## 함수 정의

```
def moveXs(self, params, _):
    with self.lock:
        for move in params["moves"]:
            self.targets.append(move)
    return {"finished": True}
```

그중에 params가 Mech-Viz 프로젝트의 모든 파라미터를 입력하고 params["moves"]를 통해 모든 웨이포인트(비전 이동, 상대적인 이동 등 포함)의 포즈를 획득할 수 있습니다. 포즈는 기본적으로 관절 각도의 형식으로 반환되고 포즈를 self.targets에 입력한 후 나중에 호출하는 데 사용될 것입니다.

## 예시

일반적으로 이 기능은 “알림” 스텝과 함께 사용되어야 합니다. Adapter가 “알림” 스텝에서 보내 온 메시지를 수신하면 self.targets를 함수에 입력하고 전환 및 압축한 후 로봇에 전송합니다. 예를 들면 아래와 같습니다.

```
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)
```

메시지 "started"를 수신했을 때 Mech-Viz 실행 오류로 인해 중단되어 이동 전/후 두 개의 웨이포인트가 서로 겹치는 것을 방지하기 위해 웨이포인트 리스트를 지워야 합니다. "finished"를 수신했을 때 웨이포인트 리스트를 pack\_move 함수에 입력해 나중에 데이터를 정리하고 보내는 데 사용됩니다.

```
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)
    
```

작업 현장의 실제 수요에 따라 Mech-Viz의 모든 웨이포인트를 전송해도 되고 index에 근거하여 몇 개를 선택해서 전송해도 됩니다. pack\_move() 함수는 각 로봇 브랜드가 필요한 데이터 포맷에 따라 데이터를 정리합니다(일반적으로 통신 프로토콜에서 미리 설정됨).

### 6.5.3.8. 기타 인터페이스

이 부분에서는 Adapter를 사용하는 기타 인터페이스에 대해 소개하겠습니다. 구체적으로 다음 인터페이스를 포함합니다:

#### 알림 서비스

Mech-Viz 프로젝트가 특정 분기 또는 특정 스텝으로 실행될 때 Adapter 프로그램의 해당 함수를 호출하려는 경우 Mech-Viz에 알림 스텝을 추가할 수 있습니다.

#### 예시

예를 들어, Adapter에서 이미 하역된 파렛트의 수량을 1씩 증가시키는 함수를 작성한다면, 디팔레타이징 과정의 마지막 스텝 뒤에 “알림” 스텝을 추가하여 여기에 도달하면 Adapter를 트리거하고 해당 함수를 호출할 수 있습니다. 이 기능을 구현하는 예시는 다음과 같습니다.

1. NotifyService를 상속하는 클래스를 작성하십시오.

```

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()
    
```

이 알림은 다음 기능을 수행할 수 있습니다. Mech-Viz가 "Success"을 보내는 알림 스텝까지 실행되면 Adapter는 update\_success\_num() 함수를 호출하며 "Fail" 을 보내는 알림 스텝까지 실행되면 Adapter는 update\_fail\_num() 함수를 호출합니다.

2. Mech-Viz 메인 프로그램을 컨트롤하는 클래스에서 NotifyService 클래스를 인스턴스화하고 이

서비스를 등록합니다.

```
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. Mech-Viz의 적당한 위치에 대응하는 “알림” 스텝을 추가하십시오.

“알림” 스텝에 대해 가장 중요한 것은 **Adapter 명칭** 및 \*메시지\*를 입력하는 것입니다. 이 두 파라미터의 값은 NotifyService 클래스 중의 service\_name 및 msg 값과 일치해야 합니다.

프로그램이 실행되면 Mech-Center 인터페이스에 service\_type 및 service\_name이 나타납니다. 이는 알림 서비스 등록이 성공했음을 나타냅니다.

#### VisionWatcher 서비스

Mech-Vision 실행 과정이 완료되면 결과가 출력됩니다. 예를 들어: vision result:{‘noCloudInRoi’: False, ‘function’: ‘posesFound’, ‘vision\_name’: ‘TJTvission-3’}. 일부 비정상적인 상황의 경우 Adapter는 VisionWatcher 서비스를 통해 오류 메시지를 보낼 수 있습니다.

#### 예시

##### 1. VisionResultSelectedAtService를 상속하는 클래스를 작성하십시오.

```
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()
```

하위 클래스 VisionWatcher는 상위 클래스의 poses\_found() 함수를 다시 작성해야 하므로 Adapter에서 send\_err\_no\_cloud()(오류 정보를 전송하는 함수)호출하는 논리가 poses\_found()에서 변할 것입니다. 실행 시 Mech-Vision이 반환한 포즈의 값은 poses\_found()의 result 파라미터로 전송됩니다.

## 2. Mech-Viz 메인 프로그램을 컨트롤하는 클래스에서 VisionWatcher를 인스턴스화하십시오.

```
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 클래스를 인스턴스화하는 과정에 send\_err\_no\_cloud() 함수를 파라미터로 VisionWatcher()에 전송합니다. 포인트 클라우드가 표시되지 않으면 poses\_found()의 논리에 따라 오류 메시지를 보내는 함수가 호출됩니다.

프로그램이 실행되면 등록된 서비스가 Mech-Center 인터페이스에 나타납니다. 그러면 Adapter가 VisionWatcher 서비스를 성공적으로 등록했음을 나타냅니다.

### 6.5.4. Adapter 프로그래밍 샘플

Adapter 프로그래밍 구문 및 프로그래밍 논리에 익숙해지면 이 부분에서 제공하는 샘플 프로그램을 참조하여 본인의 Adapter 프로그램을 작성할 수 있습니다.

- [Mech-Vision만 사용하여 비전 포즈를 전송하기](#)

#### 6.5.4.1. Mech-Vision만 사용하여 비전 포즈를 전송하기

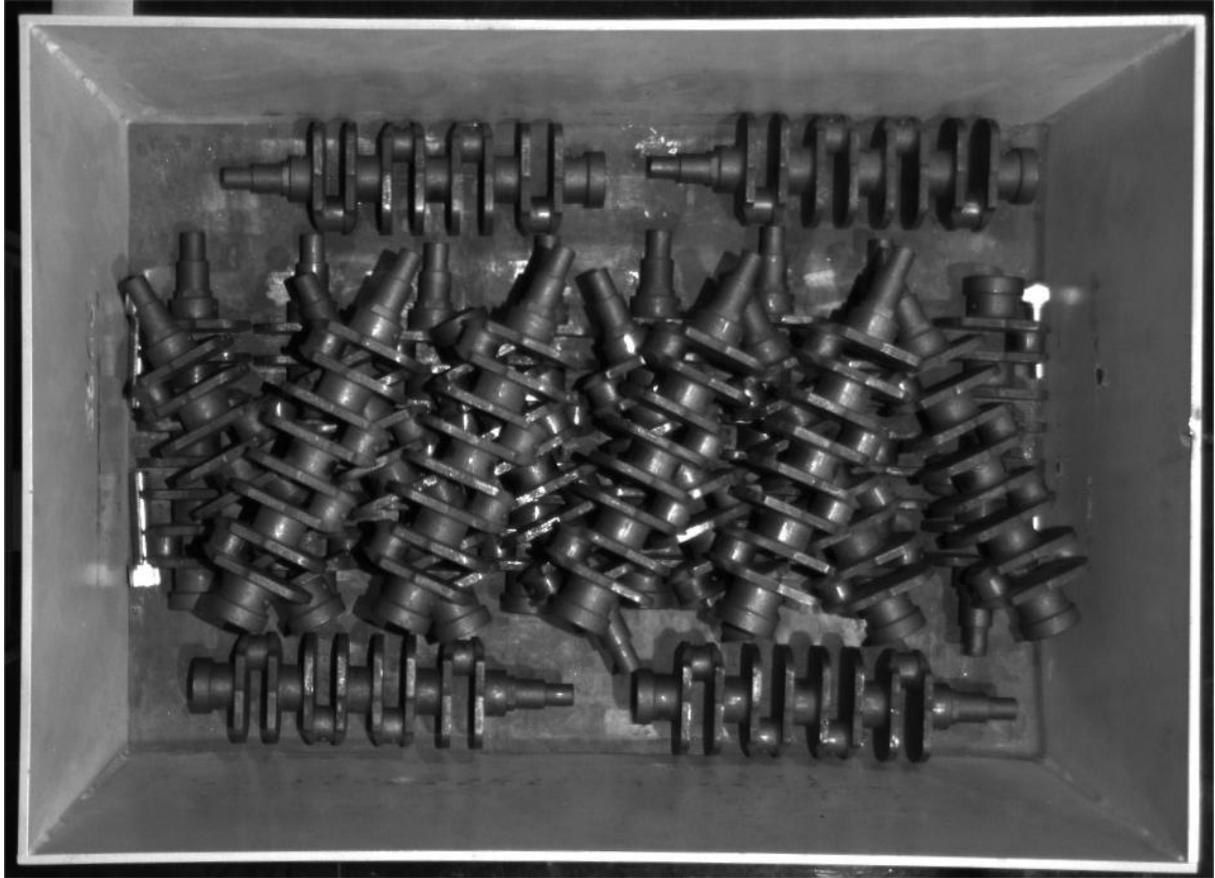
이 부분에서는 Mech-Vision만 사용하여 비전 포즈를 전송하는 Adapter 샘플 프로그램에 대해 자세히 소개하겠습니다.

##### 배경 소개

이 샘플은 크랭크 축 텐딩의 응용 시나리오에 관한 것입니다. 카메라는 bin 위의 브래킷에 고정되며 Mech-Vision은 이미지를 캡처하고 피킹할 수 있는 작업물의 좌표를 출력해 로봇 측에 보냅니다.

이 샘플은 Mech-Vision의 내장 샘플 프로젝트 “대형 비 평면 부품”(파일 > 샘플 프로젝트를 보기 > 부품 로드&언로드 > 대형 비 평면 부품)를 사용합니다.

이 프로젝트는 3D 모델 매칭 알고리즘을 사용하며 서로 다른 공작물에 대해 서로 다른 모델 파일과 픽 포인트를 설정해야 합니다. 따라서 Mech-Vision에서 파라미터 레시피를 설정해야 하고 레시피 번호(공작물 번호)는 로봇 측에서 이미지 캡처 명령을 보낼 때 설정합니다.

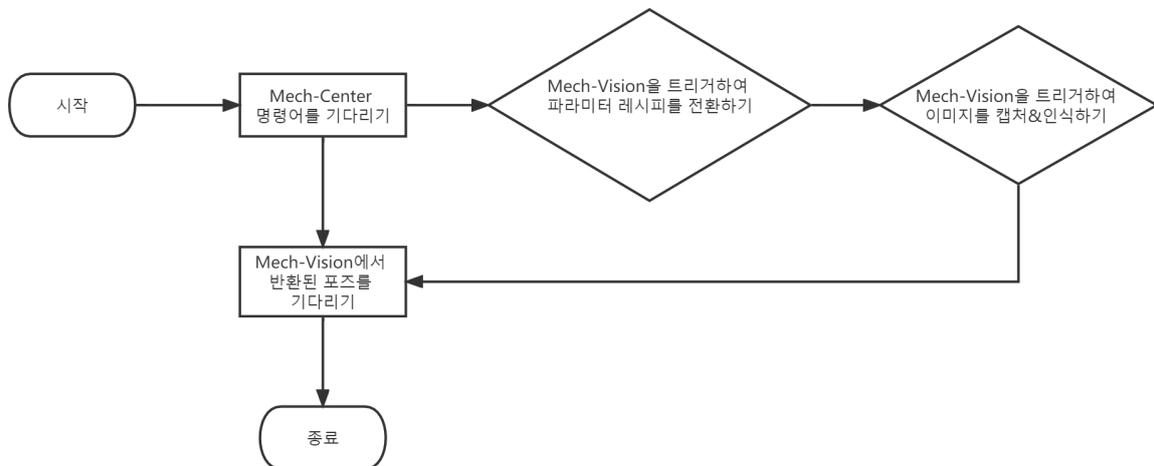


### 통신 방안

로봇과 Mech-Mind 비전 소프트웨어 시스템을 설치하는 IPC는 TCP/IP Socket 프로토콜을 사용하여 통신하며 통신 형식은 ASCII 문자열이며 데이터 구분 기호는 영어 쉼표(,)를 사용합니다.

그 중 비전 시스템은 통신 서버로, 로봇은 클라이언트로 사용됩니다.

통신 프로세스는 아래 그림과 같습니다.



통신 프로세스의 상세한 설명은 아래와 같습니다:

1. Mech-Center는 로봇이 이미지 캡처 명령 `P` 와 레시피 번호를 보낼 때까지 기다립니다.

2. Mech-Center는 Mech-Vision을 트리거하여 파라미터 레시피를 전환합니다.
3. Mech-Center는 Mech-Vision을 트리거하여 이미지를 캡처합니다.
4. 이미지 캡처와 물체 인식이 성공하면 Mech-Vision은 상태 코드와 비전 포즈를 Mech-Center로 전송합니다.
5. Mech-Center는 상태 코드와 비전 포즈를 로봇으로 전송합니다.



로봇 피킹을 용이하게 하기 위해 Mech-Center는 피킹할 작업물의 포즈를 로봇 TCP 포즈로 변환합니다.

### 통신 메시지 형식

자세한 통신 메시지 형식은 아래와 같습니다.

	요청 명령	공작물 번호
전송 (로봇 → IPC)	P	정수, 값 범위: 1~100
	상태 코드	비전 포즈(TCP 좌표)
수신 (IPC → 로봇)	정수, 값 범위: 0~4 (0-정상 인식, 1-잘못된 명령 코드, 2-Vision 프로젝트가 등록되지 않음, 3-포즈가 없음, 4-포인트 클라우드가 없음)	x,y,z,a,b,c 형식의 쉼표(,)로 구분된 6개의 부동 소수점 데이터입니다.



응답 메시지의 길이는 정해진 것입니다. 응답 메시지의 상태 코드가 비정상 코드(1~4)인 경우 비전 포인트 데이터는 0으로 채워집니다.

### 통신 메시지 예시

#### 요청 메시지

P,1

#### 정상적인 응답 메시지

0,1994.9217,-192.198,506.4646,-23.5336,-0.2311,173.6517



이 샘플에서 Mech-Vision은 정상적으로 인식하며 반환된 TCP 좌표는 1994.9217,-192.198,506.4646,-23.5336,-0.2311,173.6517입니다.

#### 비정상적인 응답 메시지: 잘못된 명령 코드

1,0,0,0,0,0,0

#### 비정상 응답 메시지: 비전 프로젝트가 등록되지 않음

2,0,0,0,0,0,0

#### 비정상적인 응답 메시지: 포즈가 없음

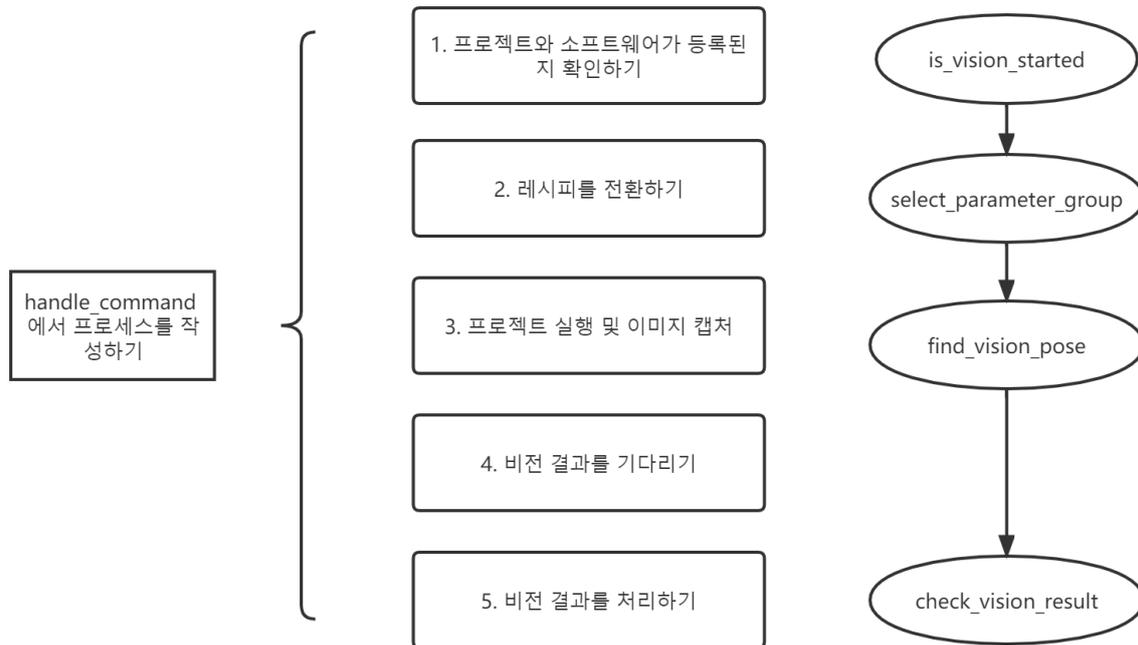
```
3,0,0,0,0,0,0
```

비정상적인 응답 메시지: 포인트 클라우드가 없음

```
4,0,0,0,0,0,0
```

### 프로그래밍 맥락

이 샘플은 프로젝트 목표를 달성하기 위해 다음 그림과 같은 맥락에 따라 Adapter 프로그램을 작성합니다.



위의 그림은 샘플 프로그램의 메시지 처리 논리만을 나열한 것입니다. 다음으로 샘플 프로그램에 대해 자세히 설명하겠습니다.

### 샘플 프로그램 상세 소개

 [Adpater 샘플 프로그램](#)을 다운로드하십시오.

### Python 패키지를 가져오기

Adapter 프로그램이 의존하는 모든 모듈을 도입합니다.

```
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
    
```

## 클래스 정의

“TcpServerAdapter” 상위 클래스를 상속하는 “TestAdapter” 하위 클래스를 정의하십시오.

```

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_rcv_size(1024)
    
```



이 샘플은 Adapter 프로그램을 TCP/IP 소켓 통신용 서버로 정의합니다.

## 명령 수신 및 논리 처리 설정

수신을 받기 위한 처리 논리를 설정합니다(이미지 캡처 및 파라미터 레시피 포함).

```

# 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” 함수는 TCP/IP 소켓 서버가 수신한 메시지에 대한 처리 인포트로 사용됩니다.

## Mech-Vision 비전 결과 체크를 정의하기

Mech-Vision에서 출력한 비전 결과를 체크하도록 Adapter를 설정합니다.

```

# 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')
    
```

## 포즈의 출력 형식을 설정하기

비전 포즈의 외부로 출력하는 형식을 설정합니다.

```
# 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 + ''
```

## Adapter를 닫는 작업을 정의하기

Adapter를 닫는 방법을 정의합니다.

```
def close(self):
    super().close()
```