Robotics
응용로봇공학및설계

Student Presentations

발표 주제 소개
(Research Topics)

SLAM · 자율주행 · 비전 기반 픽앤플레이스

Department of Control & Instrumentation Engineering
Pukyong National University, Busan, Korea
pnj@pknu.ac.kr

발표 일정 — 11월 12일부터 매주 목요일

발표 자료는 발표 전날 자정까지 메일로 보낸다: pnj@pknu.ac.kr

날짜	형식
11월 5일	발표 주제 소개 (오늘)
11월 12일	팀 발표 30분 + 질의응답 10분
11월 19일	개별 발표 15분 + 질의응답 10분 (2명)
11월 26일	개별 발표 15분 + 질의응답 10분 (2명)
12월 3일	개별 발표 15분 + 질의응답 10분 (2명)
12월 10일	개별 발표 15분 + 질의응답 10분 (1명)

[안내] 개별: 발표 15분 + 질의응답 10분 · 팀(11월 12일): 발표 30분 + 질의응답 10분
평가: 내용 20 · 전달 10 · 질의응답 10.

발표 주제 소개 세 가지 연구 주제

1 SLAM: 지도 없이 지도를 만들며 위치를 안다

2 자율주행: 측위 · 인지 · 예측 · 계획 · 제어

3 픽앤플레이스: 깊이 센서 없이 3차원 위치를 구한다

[안내] 발표 주제는 로봇 관련 자율 주제: 아래 셋 가운데 골라도, 스스로 정해도 된다.

발표 주제 소개 세 가지 연구 주제

1 SLAM: 지도 없이 지도를 만들며 위치를 안다

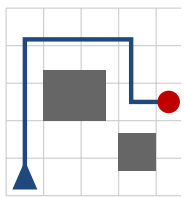
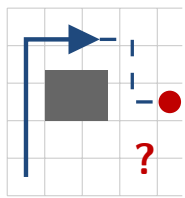
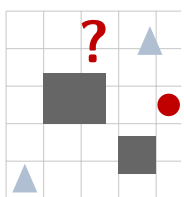
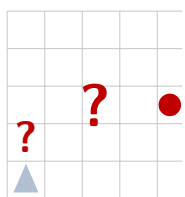
2 자율주행: 측위 · 인지 · 예측 · 계획 · 제어

3 픽앤플레이스: 깊이 센서 없이 3차원 위치를 구한다

[왜 필요한가] 지도도 GPS 도 없는 로봇청소기 · 지하주차장의 차량은 어떻게 다닐까? 보이는 주변 환경만으로 지도와 위치를 함께 구하는 방법이 SLAM 이다.

Motivation — 지도와 위치, 둘 다 있어야 길을 간다 (1/2)

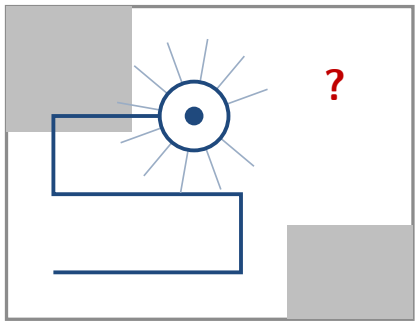
내비게이션: 미리 만든 지도 + GPS 가 주는 현재 위치 ⇒ 목적지까지 경로를 계산한다.

	지도 있음	지도 없음
위치 있음	 **내비게이션** 내 위치가 지도에 찍힌다 → 경로를 계산한다 가장 쉬운 경우	 **지도 작성** 목적지 방향만 안다 → 건물을 피해 가며 지나온 곳을 지도로
위치 없음	 **위치 추정** 지도는 있다, 나는 어디? → 보이는 것과 지도를 맞춰야 위치를 안다	 **SLAM** 지도도 위치도 없다 → 지도를 그리면서 그 위의 내 위치를 갱신

[핵심] 지도 작성과 위치 추정은 서로를 필요로 한다 ⇒ 둘을 **동시에** 진행 = SLAM

Motivation — 일상에서 보는 SLAM (2/2)

미리 만든 지도도, GPS 도 없는 곳에서 움직이는 기계는 전부 같은 문제를 푼다.

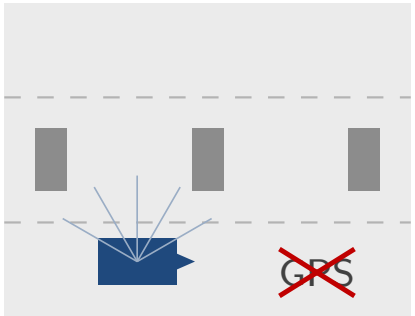


로봇청소기

처음 켜면 집 지도가 없다

→ 돌면서 지도를 그리고

그 위에서 내 위치를 갱신

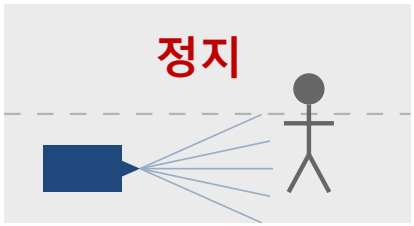


지하주차장 · 실내

GPS 신호가 끊긴다

→ 라이다(LiDAR)로 보고

기둥 · 벽으로 위치를 구한다



자율주행 차량

보행자를 보고 멈춘다

→ 본 것으로 지도를 그리며

차의 위치도 함께 구한다

[공통] 주변을 본 것만으로 **지도 작성** + **위치 추정**을 동시에 ⇒ 없으면 움직일 수 없다.

SLAM — Simultaneous Localization and Mapping

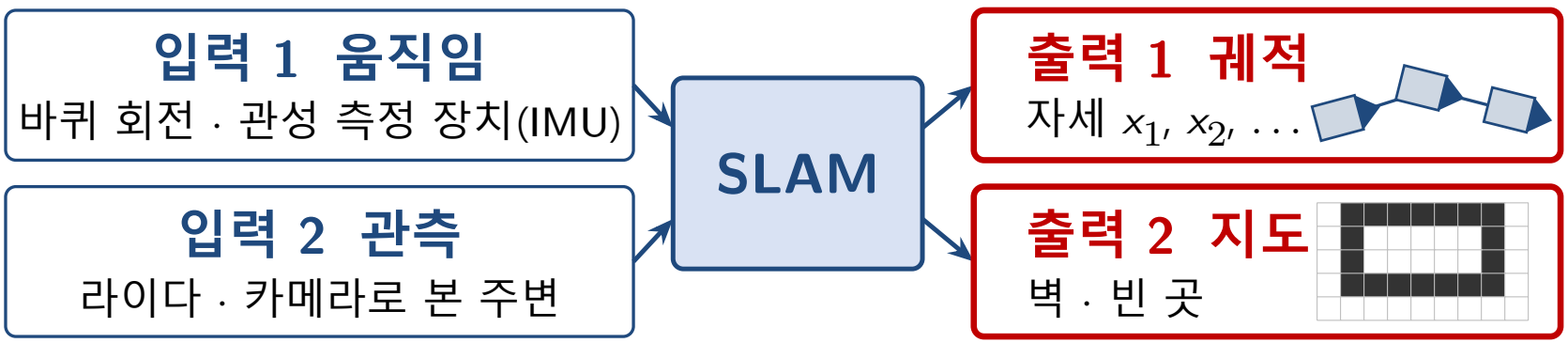
CONCEPT 정의

위치 추정(localization): 지도를 알 때 자신의 위치 · 방향을 구한다

지도 작성(mapping): 위치 · 방향을 알 때 주변 벽을 지도에 그린다

SLAM(동시적 위치 추정과 지도 작성): 둘 다 모를 때 **둘을 동시에** 추정

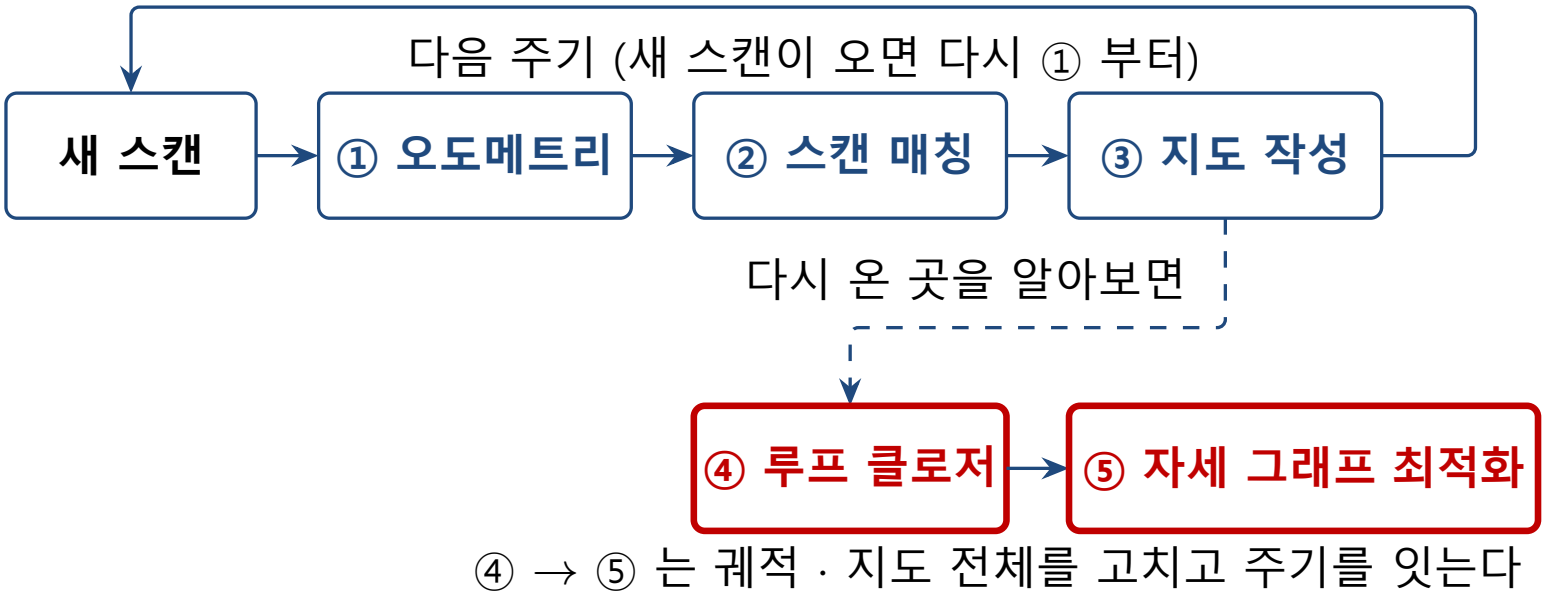
자세(pose) = 위치 + 방향, **궤적**(trajectory) = 시간 순서의 자세 $x_1, x_2, \ldots$



[직관] 궤적의 자세와 지도의 벽을 모두 출발 자세 x_1 의 좌표계 하나에서 표현한다.

SLAM 의 흐름 — 새 스캔마다 ① ② ③, 다시 온 곳에서 ④ ⑤

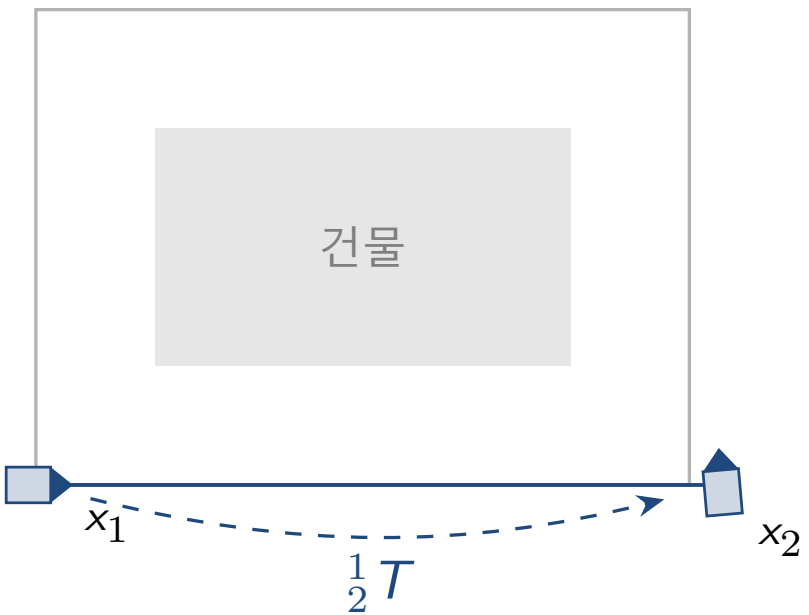
새 스캔마다 ① → ② → ③ 을 반복, 다시 온 곳을 알아보면 ④ → ⑤ 로 전체를 고친다.



[이 덱] 주기 1($x_1 \to x_2$)은 ① ② ③ 을 한 쪽씩, 주기 2∼4 는 한 쪽에 하나씩.
한 바퀴 뒤 $x_5 \approx x_1$ 을 알아보면 ④ → ⑤ 로 쌓인 오차를 없앤다.

주기 1 ① 오도메트리 — 상대 변환으로 x_2 를 추정한다

오도메트리(odometry): 바퀴 회전 · IMU 로 구한 상대 변환 ${}^1_2T$ 로 x_2 를 추정



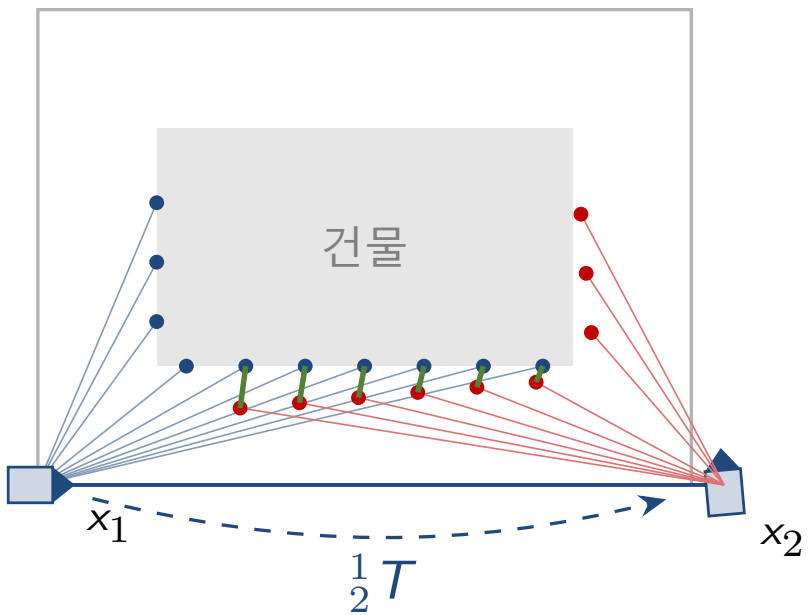
새 스캔마다 반복
① **오도메트리**
② 스캔 매칭
③ 지도 작성
다시 온 곳에서
④ 루프 클로저
⑤ 자세 그래프 최적화

실제 경로
추정 궤적 · 자세
상대 변환
스캔
최근접 짝 (ICP)
루프 클로저

[한계] 바퀴 미끄러짐 · 회전 오차가 그대로 ⇒ 추정한 $x_2 \neq$ 실제 모서리(회색)

주기 1 ② 스캔 매칭 — 같은 벽을 두 자세에서 본다 (1/2)

스캔 매칭(scan matching): 두 스캔이 겹치는 $\frac{1}{2}T$ 를 찾는다. 먼저 최근접 짝(초록)



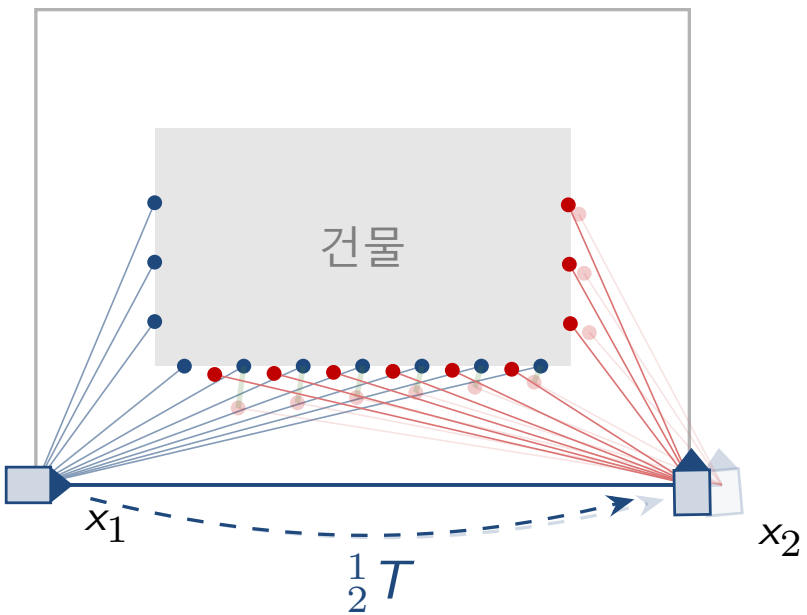
새 스캔마다 반복
① 오도메트리
② **스캔 매칭**
③ 지도 작성
다시 온 곳에서
④ 루프 클로저
⑤ 자세 그래프 최적화

실제 경로
추정 궤적 · 자세
상대 변환
스캔 (x_1 · x_2)
최근접 짝 (ICP)
루프 클로저

[직관] 바퀴는 미끄러져도 벽은 움직이지 않는다 ⇒ 같은 벽이 겹치는 변환 = 실제 이동

주기 1 ② 스캔 매칭 — ICP 로 보정한 x_2 (2/2)

ICP(Iterative Closest Point): 짝의 거리 제곱합이 최소인 변환 → 반복 ⇒ 보정한 $\frac{1}{2}T$, x_2



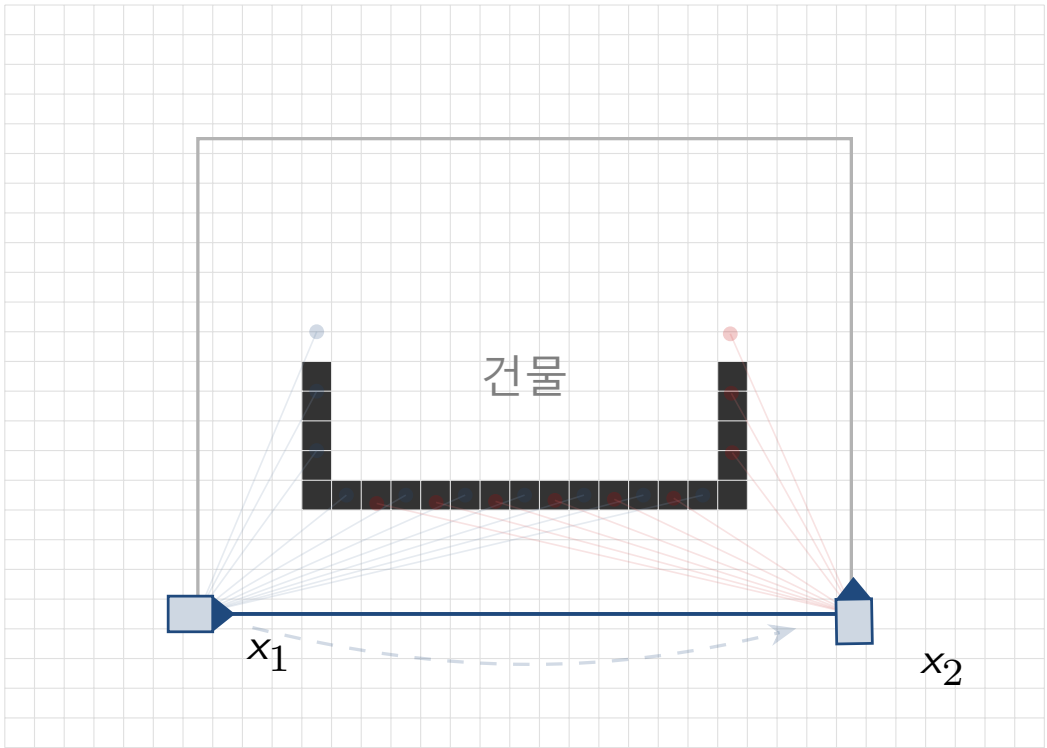
새 스캔마다 반복
① 오도메트리
② **스캔 매칭**
③ 지도 작성
다시 온 곳에서
④ 루프 클로저
⑤ 자세 그래프 최적화

실제 경로
추정 궤적 · 자세
상대 변환
스캔 (x_1 · x_2)
최근접 짝 (ICP)
루프 클로저

[핵심] 오도메트리의 $\frac{1}{2}T$ = ICP 의 초깃값, ICP 의 결과 = 보정한 $\frac{1}{2}T$

주기 1 ③ 지도 작성 — $x_1 \to x_2$ 의 스캔을 지도에 찍는다

점유 격자 지도(occupancy grid map): 빔의 끝 칸 = 벽(검정), 지나간 칸 = 빈 곳(흰)



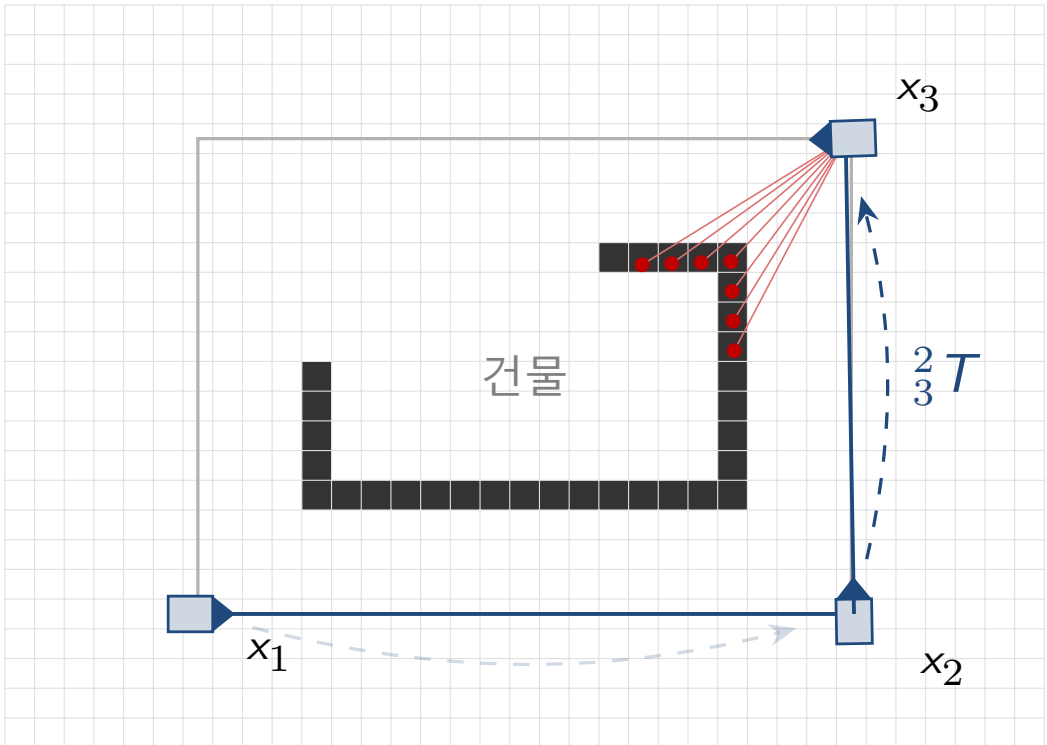
새 스캔마다 반복
① 오도메트리
② 스캔 매칭
③ **지도 작성**
다시 온 곳에서
④ 루프 클로저
⑤ 자세 그래프 최적화

실제 경로
추정 궤적 · 자세
상대 변환
스캔 (x_1 · x_2)
최근접 짝 (ICP)
루프 클로저

[주기 1 끝] 아래 벽이 생겼다. 새 스캔이 오면 $x_2 \to x_3$ 에서 ① ② ③ 을 다시

주기 2 — $x_2 \to x_3$: ① ② ③ 을 반복한다

오도메트리로 x_3 추정 → 스캔 매칭으로 보정 → x_3 의 스캔을 지도에 ⇒ 오른쪽 벽



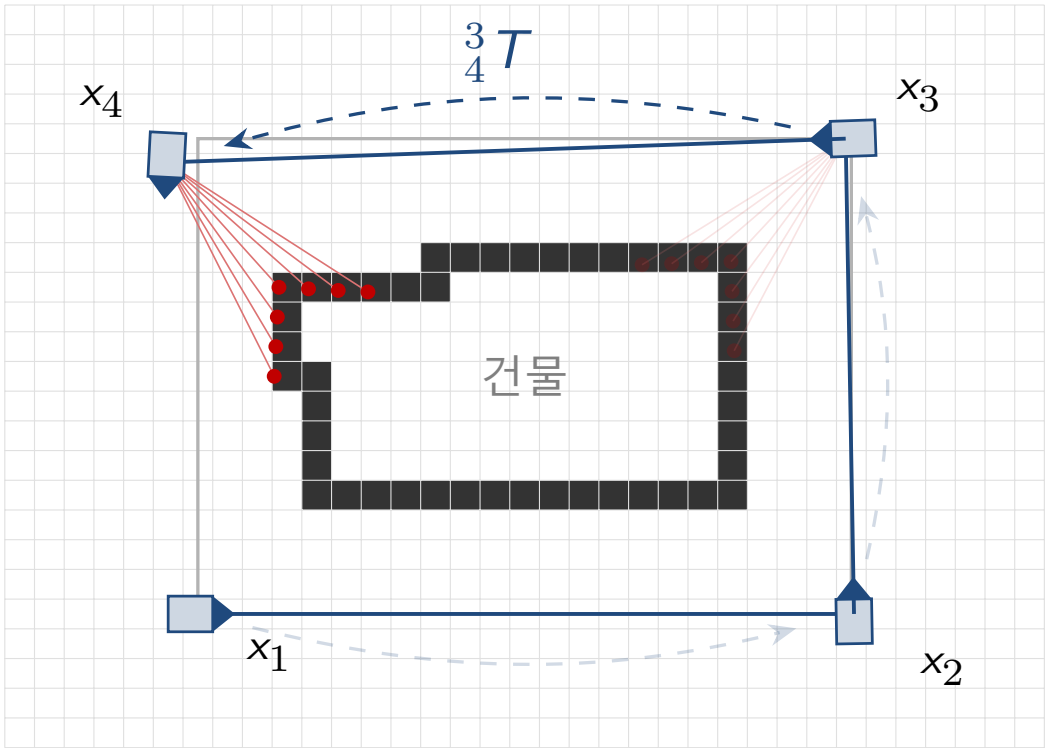
새 스캔마다 반복
① **오도메트리**
② **스캔 매칭**
③ **지도 작성**
다시 온 곳에서
④ 루프 클로저
⑤ 자세 그래프 최적화

실제 경로
추정 궤적 · 자세
상대 변환
스캔 (x_3)
최근접 짝 (ICP)
루프 클로저

[직관] ${}^1_3T = {}^1_2T \cdot {}^2_3T$ 는 연속 변환 ${}^A_CT = {}^A_BT \cdot {}^B_CT$ 와 같은 구조

주기 3 — $x_3 \to x_4$: 위 벽

같은 ① ② ③. 스캔 매칭 뒤에도 작은 오차가 남고, 주기마다 쌓인다



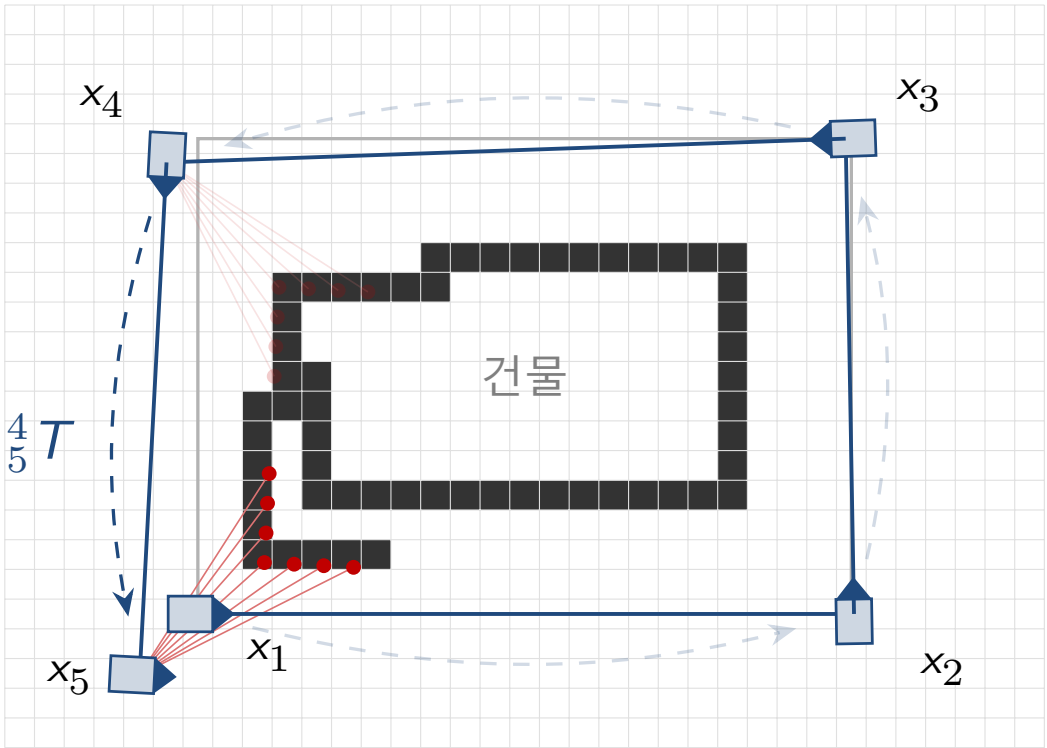
새 스캔마다 반복
① 오도메트리
② 스캔 매칭
③ 지도 작성
다시 온 곳에서
④ 루프 클로저
⑤ 자세 그래프 최적화

실제 경로
추정 궤적 · 자세
상대 변환
스캔 (x_4)
최근접 짝 (ICP)
루프 클로저

[남는 오차] x_4 가 실제보다 왼쪽 · 아래로 밀려 위 벽의 왼쪽이 아래로 찍힌다

주기 4 — $x_4 \to x_5$: 왼쪽 벽, 한 바퀴

같은 ① ② ③. 한 바퀴를 돌아 x_5 는 x_1 과 같아야 하는데 쌓인 오차만큼 차이가 난다



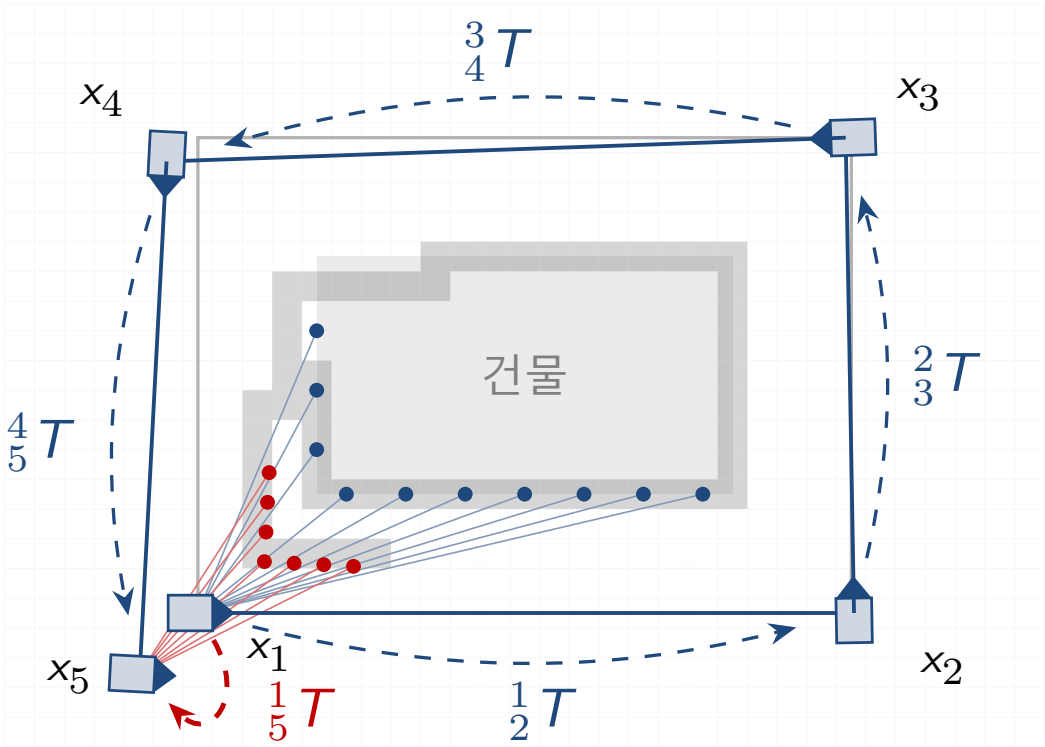
새 스캔마다 반복
① 오도메트리
② 스캔 매칭
③ 지도 작성
다시 온 곳에서
④ 루프 클로저
⑤ 자세 그래프 최적화

실제 경로
추정 궤적 · 자세
상대 변환
스캔 (x_5)
최근접 짝 (ICP)
루프 클로저

[드리프트] x_4 · x_5 에 쌓인 오차(drift) ⇒ 왼쪽 · 아래 벽이 두 겹으로 남는다

④ 루프 클로저 — 출발점을 다시 인식한다

루프 클로저(loop closure): x_5 스캔이 x_1 스캔과 같은 벽 $\Rightarrow {}_5^1T$ 를 스캔 매칭으로 직접



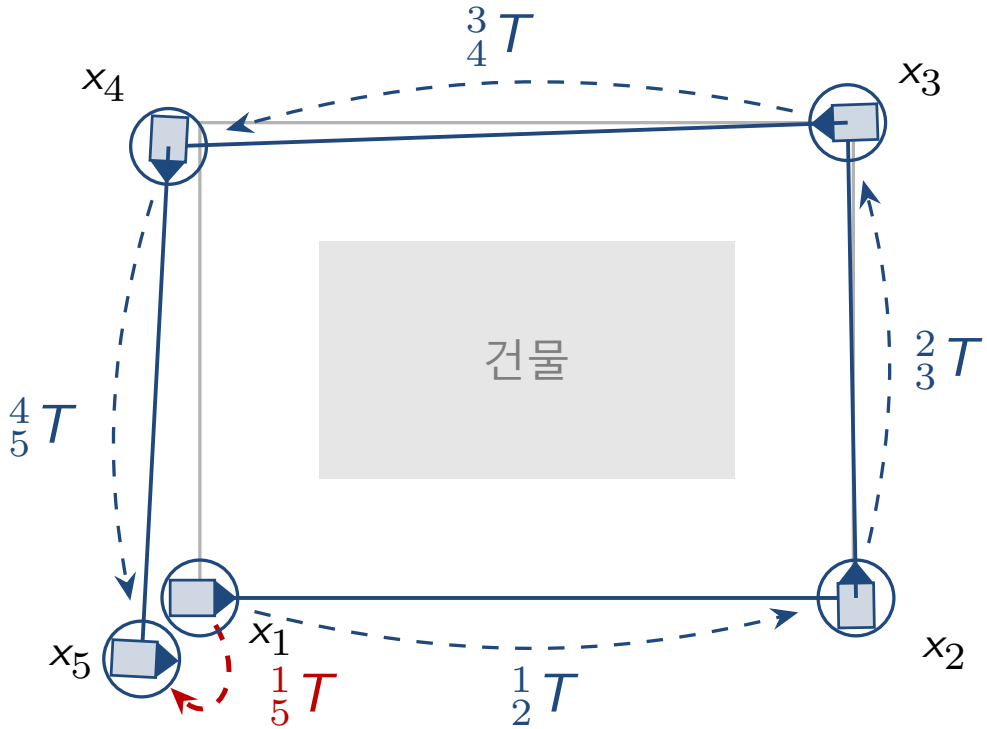
새 스캔마다 반복
① 오도메트리
② 스캔 매칭
③ 지도 작성
다시 온 곳에서
④ 루프 클로저
⑤ 자세 그래프 최적화

실제 경로
추정 궤적 · 자세
상대 변환
스캔 (x_1 · x_5)
최근접 짝 (ICP)
루프 클로저

[직관] 다시 온 곳이니 ${}_5^1T \approx I$(항등). 쌓은 곱 ${}_2^1T \cdots {}_5^4T \neq I \Rightarrow$ 그 차이 = 쌓인 오차

⑤ 자세 그래프 최적화 — 노드는 자세, 간선은 상대 변환 (1/2)

자세 그래프(pose graph): 노드(node) = 자세, 간선(edge) = 상대 변환 측정값



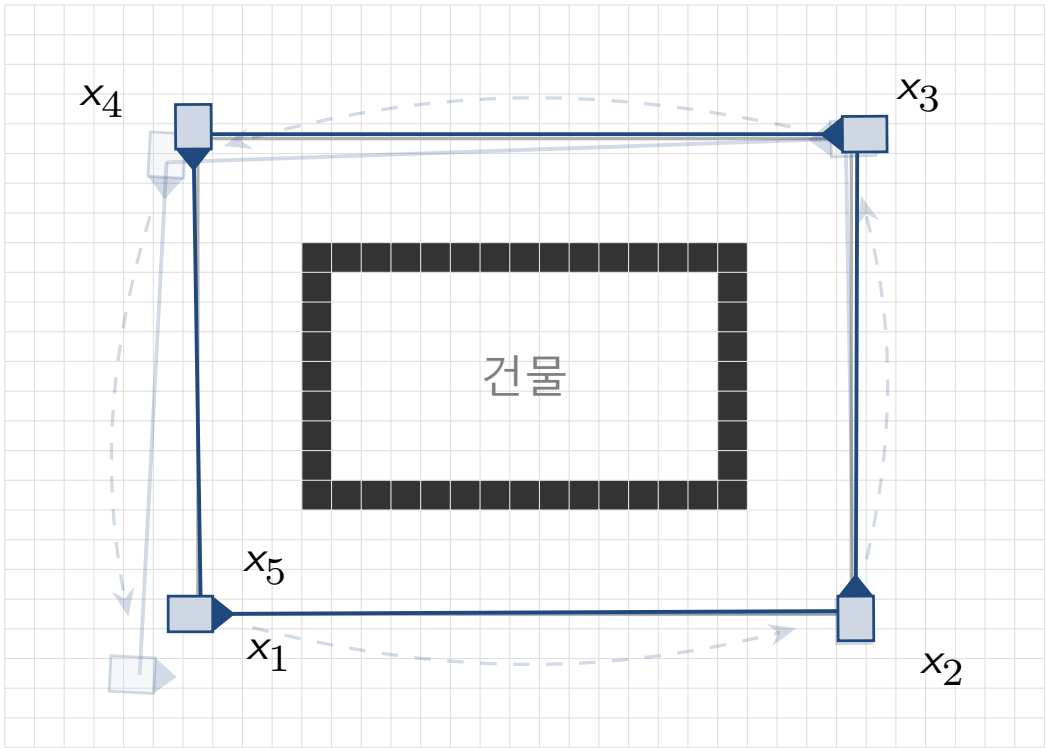
새 스캔마다 반복
① 오도메트리
② 스캔 매칭
③ 지도 작성
다시 온 곳에서
④ 루프 클로저
⑤ 자세 그래프 최적화

실제 경로
추정 궤적 · 자세
상대 변환
스캔
최근접 짝 (ICP)
루프 클로저

[직관] 측정값끼리 $\frac{1}{5}T \neq \frac{1}{2}T \cdot \frac{2}{3}T \cdot \frac{3}{4}T \cdot \frac{4}{5}T \Rightarrow$ 다섯 간선을 동시에 만족하는 자세는 없다

⑤ 자세 그래프 최적화 — 루프가 닫히고 벽이 하나로 (2/2)

자세 그래프 최적화(pose graph optimization): 간선 오차의 제곱합을 최소화



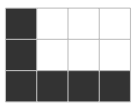
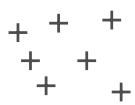
새 스캔마다 반복
① 오도메트리
② 스캔 매칭
③ 지도 작성
다시 온 곳에서
④ 루프 클로저
⑤ 자세 그래프 최적화

실제 경로
추정 궤적 · 자세
상대 변환
스캔
최근접 짝 (ICP)
루프 클로저

[결론] 쌓인 오차를 궤적 전체에 분산 $\Rightarrow x_5 \approx x_1$, 겹치던 지도의 벽도 하나로

대표 기법 — 구조는 같고 입력 센서가 다르다

세 기법의 공통 구조: 상대 변환 추정 → 루프 클로저 → 자세 그래프 최적화

이름	입력 센서	지도	특징
Cartographer	라이다 2D · 3D		점유 격자 지도	서브맵 단위 스캔 매칭 + 루프 클로저
ORB-SLAM	카메라		특징점 지도	특징점 매칭 + 루프 클로저
LIO-SAM	라이다 + IMU		3차원 점군 지도	IMU 와 라이다를 한 그래프에서 최적화

[용어] **서브맵**(submap): 스캔 몇 개로 만든 작은 지도
특징점(feature point): 영상에서 다시 찾기 쉬운 점(모서리 등)
점군(point cloud): 3차원 점의 집합

대표 기법 비교 — 영상에 나오는 이름들

차이: 추정 방식(입자 필터 / 그래프) · 루프 클로저 유무 · IMU 결합 정도

영상 1 (TurtleBot3): 2차원 라이다, 둘 다 ROS(Robot Operating System) 패키지

GMapping
입자 필터: 입자마다 궤적 + 격자 지도
명시적 루프 클로저 · 자세 그래프 없음

Cartographer
서브맵 스캔 매칭 + 자세 그래프
루프 클로저로 큰 지도도 닫힌다

영상 2 (LIO-SAM 데모의 비교): 3차원 라이다 + IMU

LOAM
라이다 특징 매칭
IMU 는 보조
루프 클로저 없음

LIO-mapping
라이다 + IMU 강결합
한 최적화에서 같이
계산이 무겁다

LIO-SAM
팩터 그래프로 강결합
루프 클로저 · GPS 선택
실시간

[용어] **입자 필터**(particle filter): 궤적 후보(입자) 여럿을 관측으로 걸러 내는 추정
팩터 그래프(factor graph): IMU · GPS 측정도 간선으로 넣은 자세 그래프
강결합(tightly coupled): 라이다 · IMU 를 한 식에서 함께 최적화 (보조로만 쓰면 약결합)

직접 보기 — SLAM 실행 영상

GMapping · Cartographer(TurtleBot3)와 LIO-SAM 의 실행 영상.

TurtleBot3 SLAM

ROBOTIS 공식 영상

GMapping · Cartographer 로 SLAM

youtu.be/lkW4-dG2BCY

LIO-SAM 데모

GitHub README 의 영상 (TixiaoShan)

라이다 + IMU 를 함께 쓰는 SLAM

youtu.be/A0H8CoORZJU

발표 주제 소개 세 가지 연구 주제

1 SLAM: 지도 없이 지도를 만들며 위치를 안다

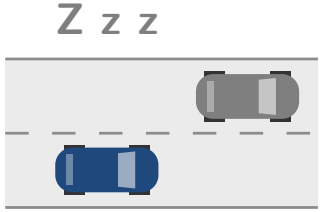
2 자율주행: 측위 · 인지 · 예측 · 계획 · 제어

3 픽앤플레이스: 깊이 센서 없이 3차원 위치를 구한다

[왜 필요한가] 사람이 하던 인지 · 판단 · 조작을 차량의 센서와 컴퓨터가 수행한다.
그 구조: 측위 · 인지 · 예측 · 계획 · 제어 다섯 단계와 단계별 대표 기법.

Motivation — 왜 자율주행이 필요한가

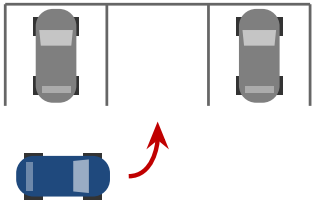
누구나 한 번은 겪는 장면들: 이때 차가 스스로 운전한다면?



장거리 주행

졸음운전
눈이 자꾸 감긴다

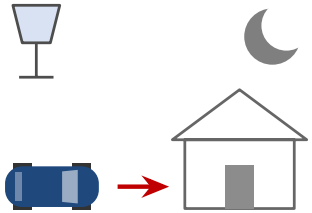
고속도로는 차가 운전, 나는 휴식



빈자리

빈 칸 찾아 몇 바퀴
좁은 칸에 넣다 뺐다

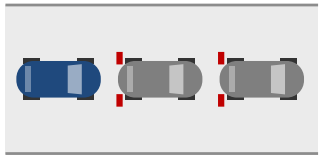
내리기만 하면 차가 스스로 주차



대리운전

운전할 수 없다
늦은 밤 기사 대기

기다림 없이 차가 집까지 운전



출퇴근 정체

가다 서다 한 시간
브레이크만 밟는다

정체 구간은 차가 알아서 주행

[공통] 네 장면 모두 **사람이 운전대를 잡아야** 해서 생기는 불편이다.

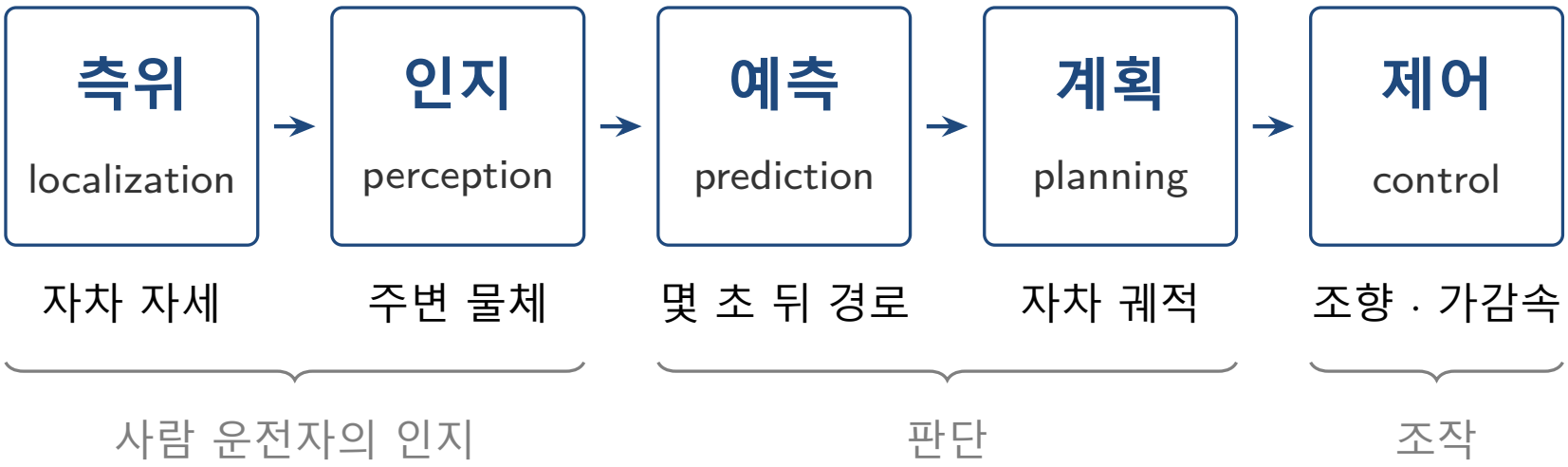
SAE 자동화 레벨 — 누가 운전하는가

미국자동차공학회(SAE)의 표준 J3016: 운전 자동화를 레벨 0∼5 로 나눈다.

레벨	이름	시스템이 하는 일	운전 주체	사람의 개입
0	자동화 없음	경고 · 순간 개입	사람	항상: 직접 운전
1	운전자 지원	조향 또는 가감속	사람	항상: 직접 운전
2	부분 자동화	조향과 가감속	사람	항상: 계속 감시
3	조건부 자동화	정해진 조건에서 운전 전부	시스템	요청이 오면
4	고도 자동화	정해진 조건에서 운전 전부	시스템	필요 없다
5	완전 자동화	모든 조건에서 운전 전부	시스템	필요 없다

[읽는 법] 레벨 3 부터(옅은 파랑 행) 운전하는 주체가 **사람에서 시스템으로** 바뀐다.
예: 1 = 적응형 순항 제어(ACC) · 2 = 테슬라 FSD(Supervised) · 고속도로 주행 보조
3 = 메르세데스 Drive Pilot · 4 = Waymo 로보택시 · 2 와 3 의 차이 = 감시 의무

전체 구조 — 다섯 단계



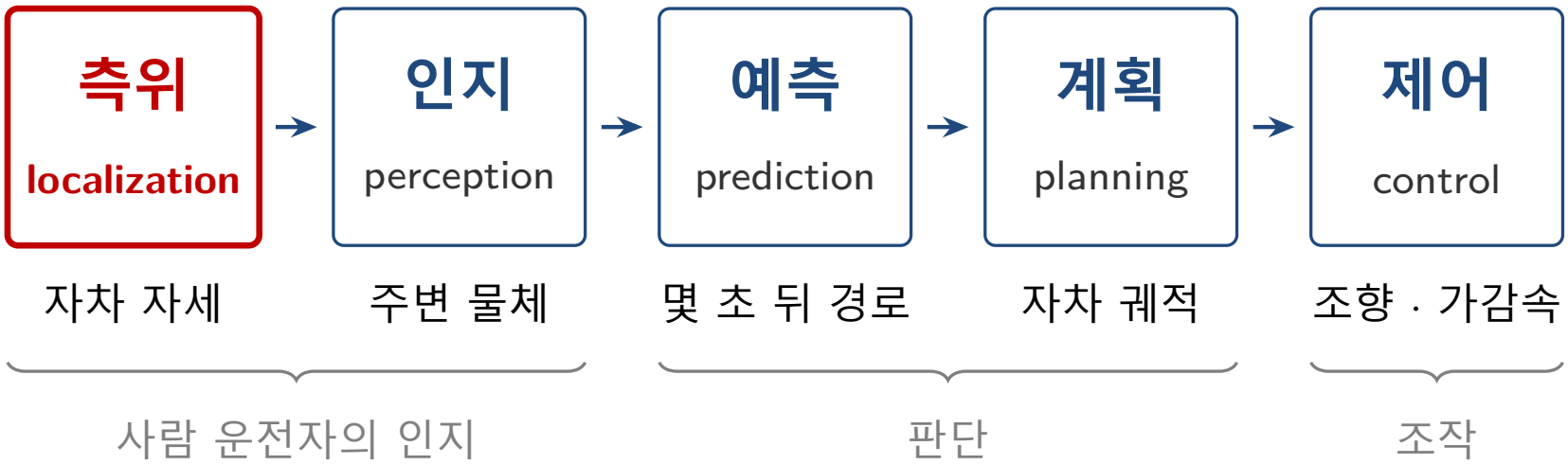
CONCEPT 정의

자율주행(autonomous driving): 차량이 센서로 주변을 보고 스스로 운전

예: 오른쪽 차로의 차가 **자차**(ego vehicle) 앞으로 끼어든다.

[직관] 각 단계의 출력이 다음 단계의 입력 → **앞 단계의 오차는 뒤 단계로 전달**된다.

전체 구조 — 측위 (1/5)



CONCEPT 정의

측위(localization): 지도 위 자차의 **자세**(pose) = 위치 + 방향을 구하는 일

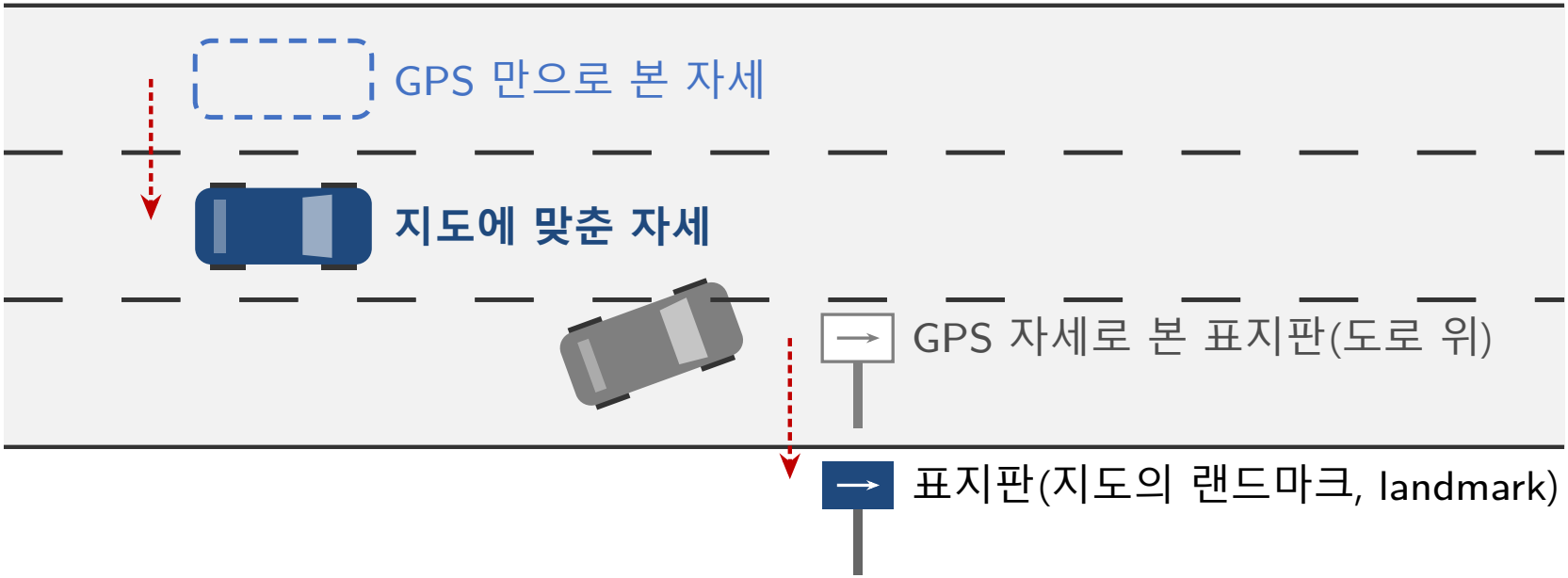
정밀 지도(HD map): 차선 · 표지판의 위치를 담아 미리 만든 지도

[직관] 주제 1 의 위치 추정: 지도는 **SLAM 으로 미리 작성**, 주행 중에는 자세만

측위 — 내 차는 지금 어느 차로의 어디에?

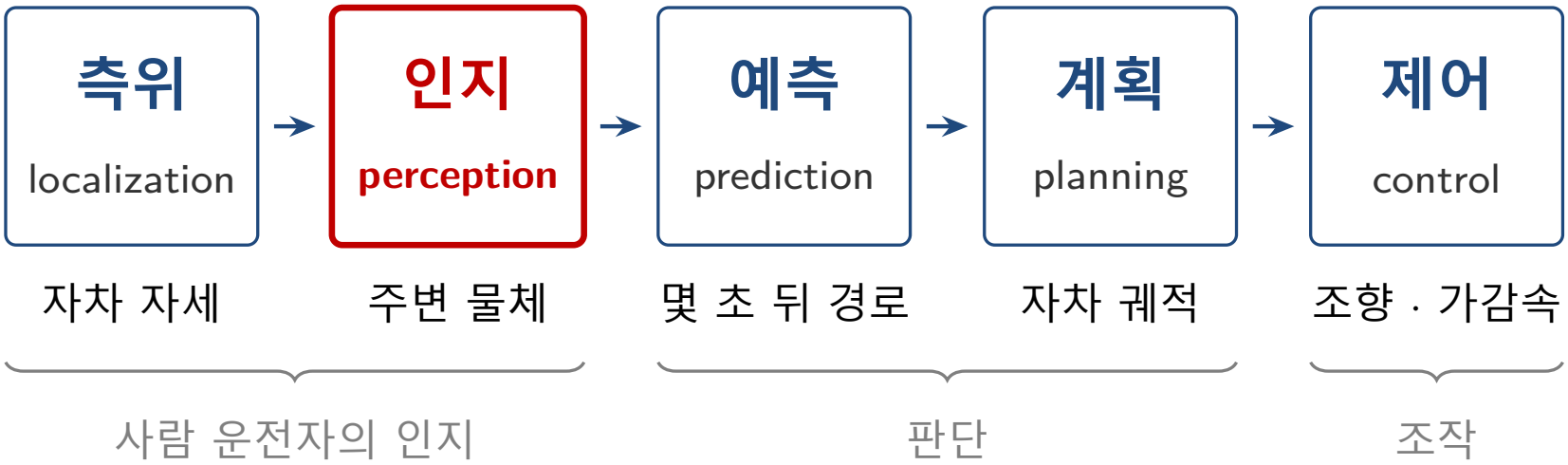
예: 편도 3차로 도로, 오른쪽 차로의 차(회색)가 자차(파랑) 앞으로 끼어든다

– – 정밀 지도의 차선 ⇢ 보정: 지도에 맞추는 변환



[읽는 법] GPS 만으로는 차로 하나가 틀릴 수 있다: 표지판이 도로 위에 놓인다.
라이다로 본 표지판 · 차선을 정밀 지도에 맞추면(스캔 매칭) 자세가 **cm 급**

전체 구조 — 인지 (2/5)



CONCEPT 정의

인지(perception): 카메라 · 라이다로 물체의 위치 · 크기 · 속도를 구하는 일

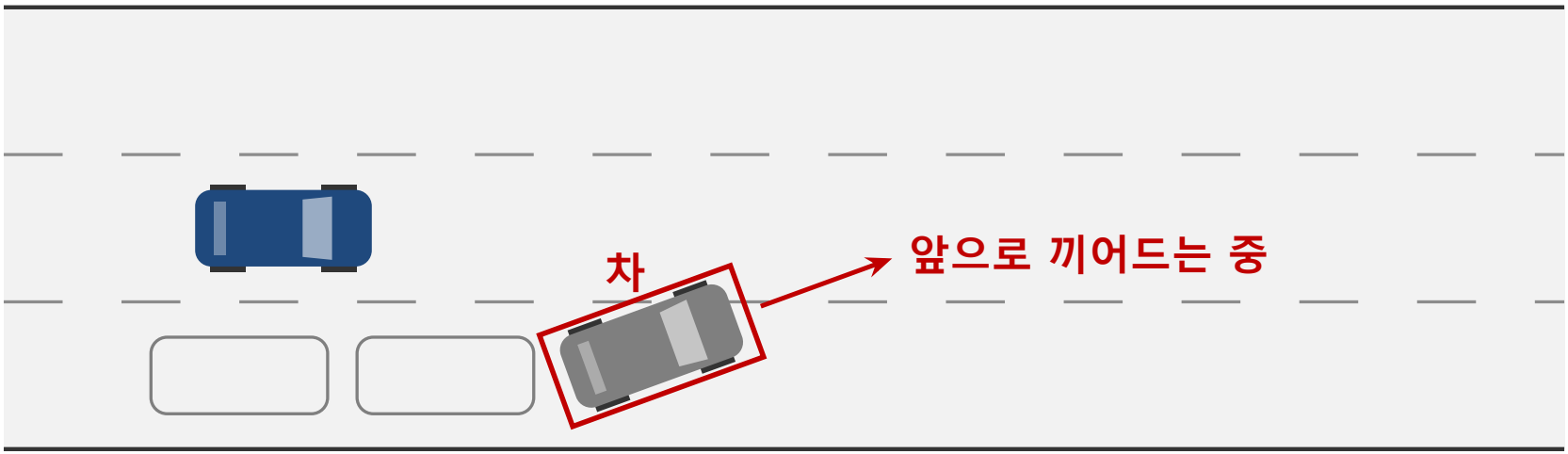
검출(detection): 물체 찾기, **추적**(tracking): 시점 사이 같은 물체 연결

[직관] 결과는 조감도(bird's-eye view, BEV): 위에서 본 평면 위의 박스

인지 — 주변에 무엇이 있고 어디로 움직이나?

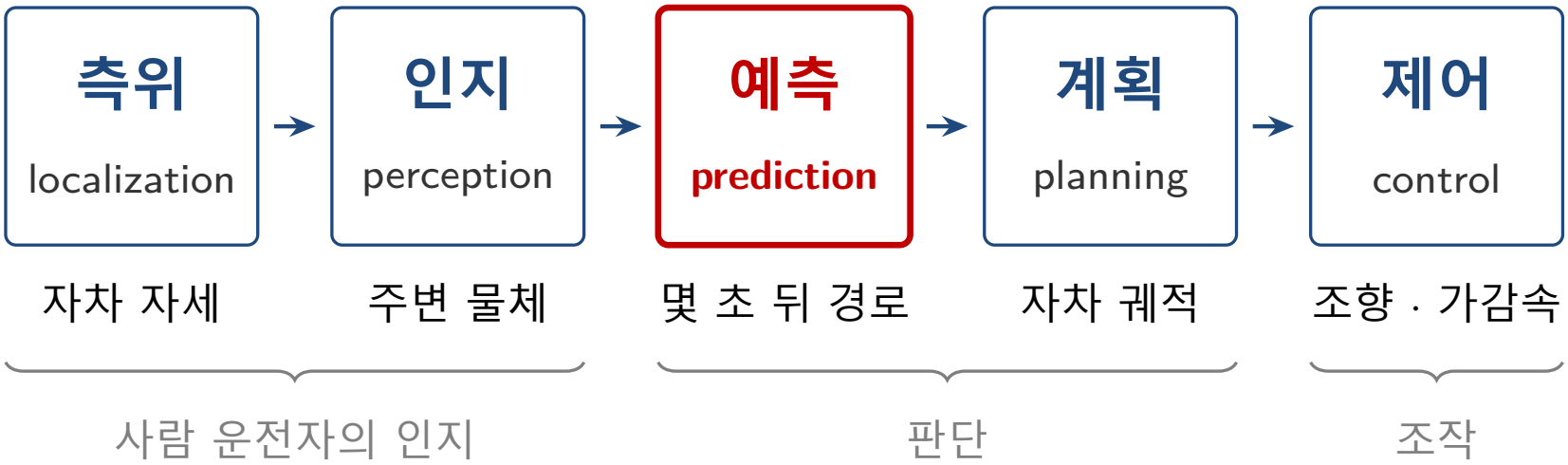
예: 편도 3차로 도로, 오른쪽 차로의 차(회색)가 자차(파랑) 앞으로 끼어든다

검출 박스 추적: 이전 자리 → 속도



[읽는 법] 검출: 차를 찾아 박스, 추적: 시점마다 같은 차를 이어 **속도**
조감도: 카메라 기준 위치를 좌표 변환으로 차량 기준 평면에 나타낸 것

전체 구조 — 예측 (3/5)



CONCEPT 정의

예측(prediction): 주변 차가 몇 초 뒤 어디에 있을지, 그 경로를 구하는 일

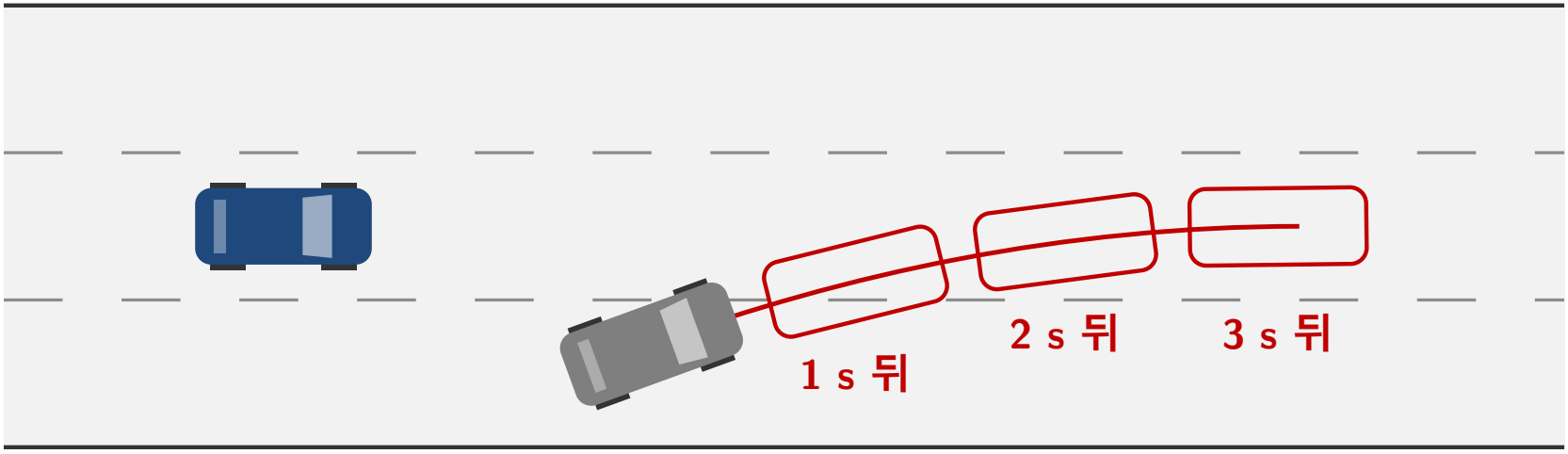
등속 가정(constant velocity): 지금 속도 그대로 직선 연장, $\mathbf{p}(t) = \mathbf{p}_0 + \mathbf{v}\,t$

[직관] 계획은 지금 자리가 아니라 **몇 초 뒤의 자리**를 피해야 한다.

예측 — 저 차는 몇 초 뒤 어디에 있을까?

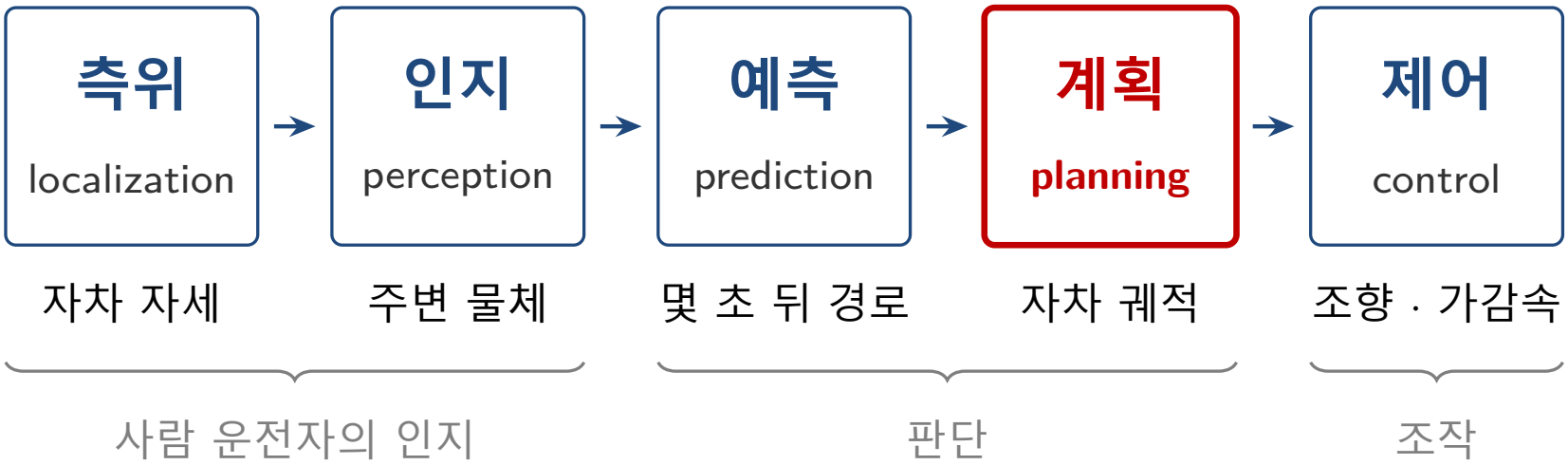
예: 편도 3차로 도로, 오른쪽 차로의 차(회색)가 자차(파랑) 앞으로 끼어든다

— 예측 경로 ▭ 몇 초 뒤 끼어드는 차의 자리



[읽는 법] 학습 기반(learning-based): 차선 · 방향지시등을 보고 **자차 차로 진입**을 예측
2 s 뒤부터 자차 차로 안 → 계획이 피해야 할 자리

전체 구조 — 계획 (4/5)



CONCEPT 정의

계획(planning): 예측된 경로를 피해 자차가 따라갈 궤적을 정하는 일

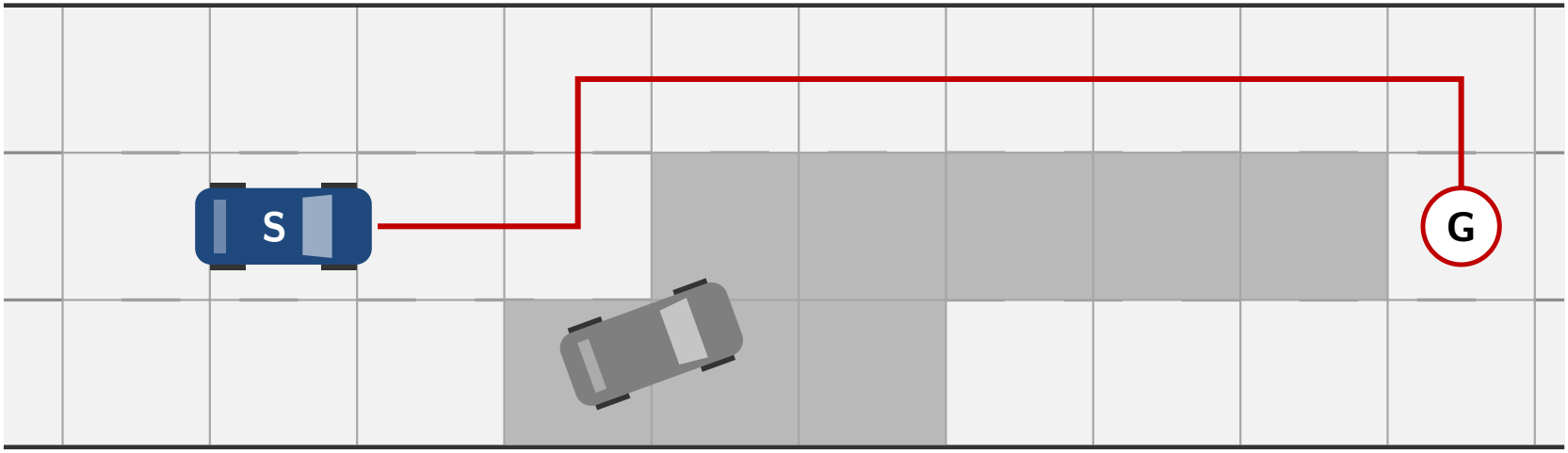
궤적(trajectory) = 경로 + 시간: 언제 어디에 있을지까지 정한 경로

[직관] 경로 탐색 → 후보 궤적 → **비용 함수**(cost function)로 하나를 고른다.

계획 — 어느 길로 갈까? (1/2)

예: 편도 3차로 도로, 오른쪽 차로의 차(회색)가 자차(파랑) 앞으로 끼어든다

막힌 칸: 끼어드는 차의 0∼3 s 예측 자리 — A* 가 찾은 경로

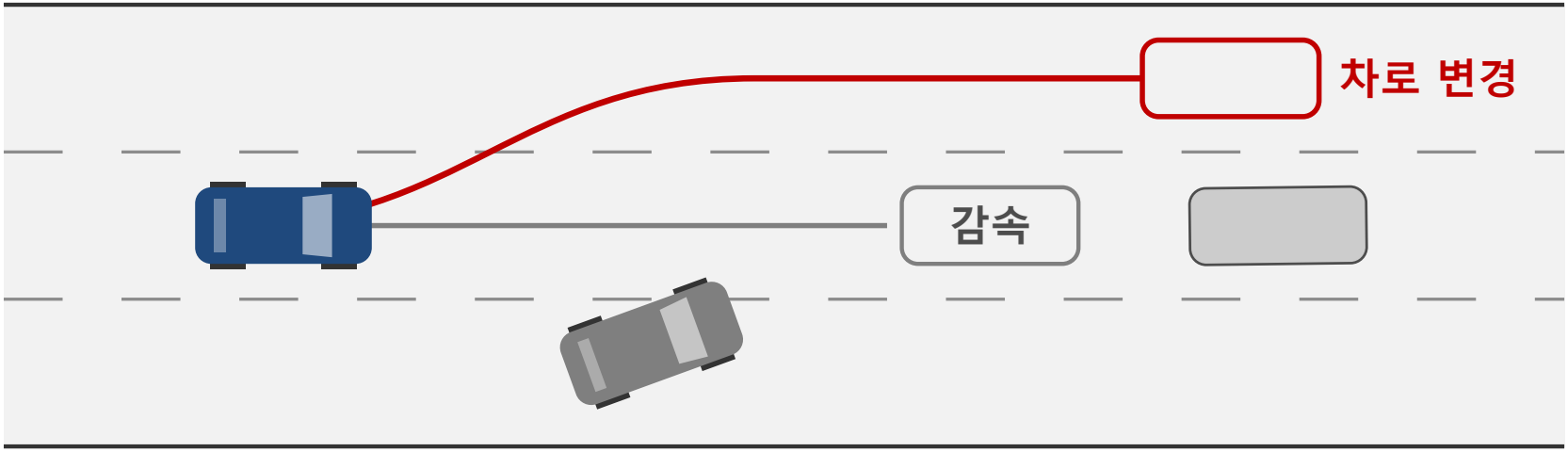


[읽는 법] A*: 막힌 칸을 피해 출발 S → 목표 G 의 **가장 짧은 칸 경로**를 찾는다.
격자 탐색: 목표 쪽 칸부터 먼저 살핀다(휴리스틱, heuristic). 막힌 칸은 예측에서 온다.

계획 — 감속할까, 차로를 바꿀까? (2/2)

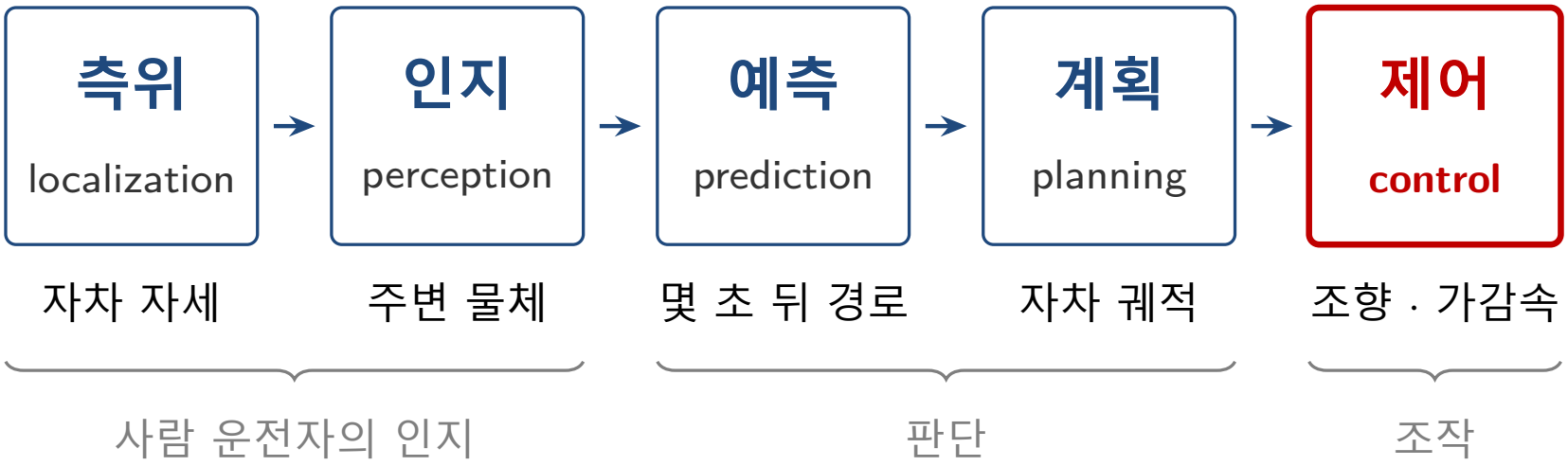
예: 편도 3차로 도로, 오른쪽 차로의 차(회색)가 자차(파랑) 앞으로 끼어든다

후보 궤적과 3 s 뒤 자차(빨강 = 고른 것)　　3 s 뒤 끼어드는 차



[읽는 법] 비용 함수: 안전 · 승차감 · 진행 비용의 합이 가장 작은 후보를 고른다.
감속은 늦고 차로 변경은 속도 유지 → 이 예는 **차로 변경**(가중치에 따라 감속도 가능)

전체 구조 — 제어 (5/5)



CONCEPT 정의

제어(control): 계획한 궤적을 따라가도록 조향각과 가감속을 정하는 일

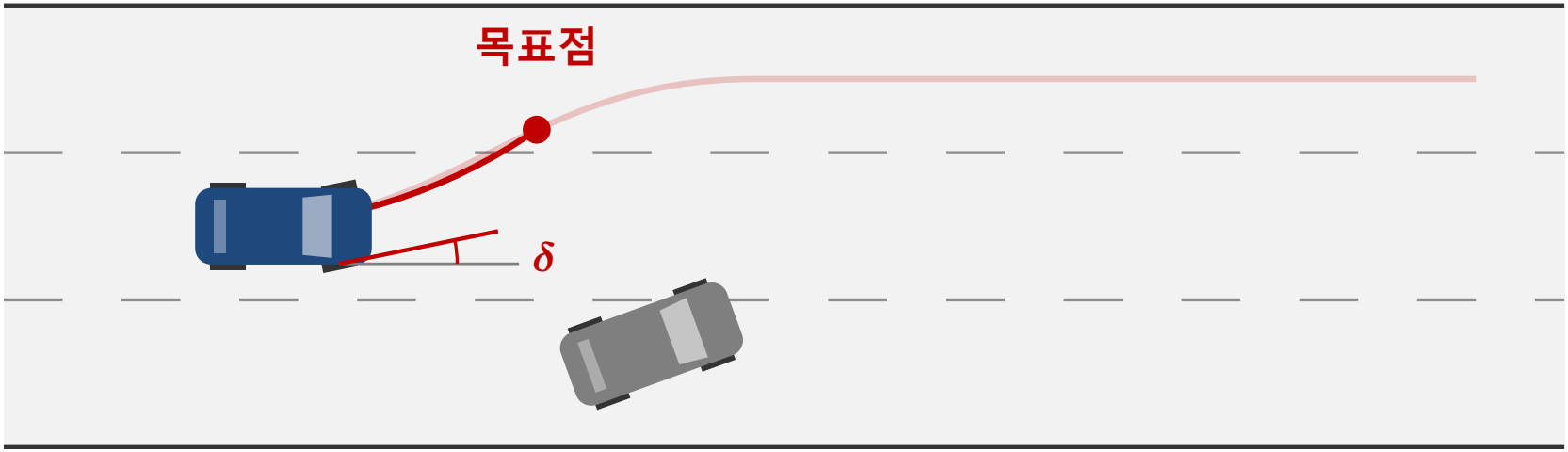
조향은 **Pure Pursuit**(목표점을 지나는 원호), 속도는 **PID**(비례 · 적분 · 미분)

[직관] 운전자가 **앞쪽의 한 점**을 보며 조향하는 것과 같다.

제어 — 핸들 · 페달을 얼마나?

예: 편도 3차로 도로, 오른쪽 차로의 차(회색)가 자차(파랑) 앞으로 끼어든다

따라갈 원호 ● 궤적 위 전방의 목표점 계획한 궤적 조향각 δ



[읽는 법] Pure Pursuit: 전방의 목표점을 지나는 원호를 따라가도록 **조향각 δ**를 정한다.
PID: 목표 속도와 지금 속도의 차이로 가감속(비례 · 적분 · 미분)

두 구조 — 모듈형과 end-to-end

모듈형(modular): 지금까지 본 다섯 단계



해석 가능(단계마다 출력을 본다) · 모듈별로 개선

end-to-end: 입력에서 출력까지 신경망(neural network) 하나

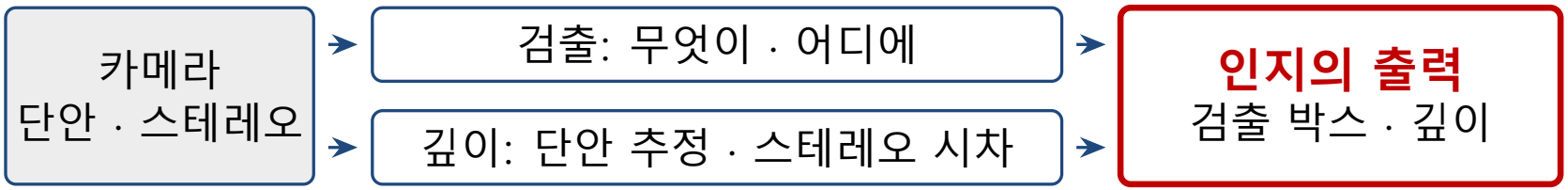


구조가 단순 · 데이터로 개선, 실패 원인 추적이 어렵다

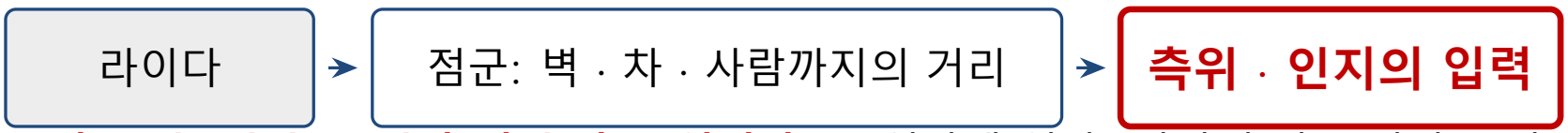
[직관] 프로그램 기반 대 학습 기반 로봇과 같은 구도다.
단계를 사람이 나눠 설계하는가, 입력 → 출력을 **데이터로 한 번에 학습**하는가.

영상의 구조 — 두 영상은 다섯 단계의 어디인가

카메라 영상: 단안(monocular) 1대, 또는 스테레오(stereo) 2대의 시차(disparity)



라이다 영상: 레이저 점군(point cloud)



보이는 것: 센서 → 인지(점선 칸 = 입력만) 영상에 없다: 인지 출력을 받아 동작



[직관] 두 영상은 **인지까지**: 카메라 = 인지 출력, 라이다 = 측위 · 인지의 입력
그 출력이 예측 → 계획 → 제어의 입력. 주제 3 도 같은 조합(검출 + 깊이 추정)

직접 보기 (1/2) — 차량이 인식한 주변

Waymo 360° experience

Waymo, 2018

차량이 인식한 주변을 겹쳐 보여 준다

youtu.be/B8R148hFxPw

직접 보기 (2/2) — 카메라와 라이다로 달린다

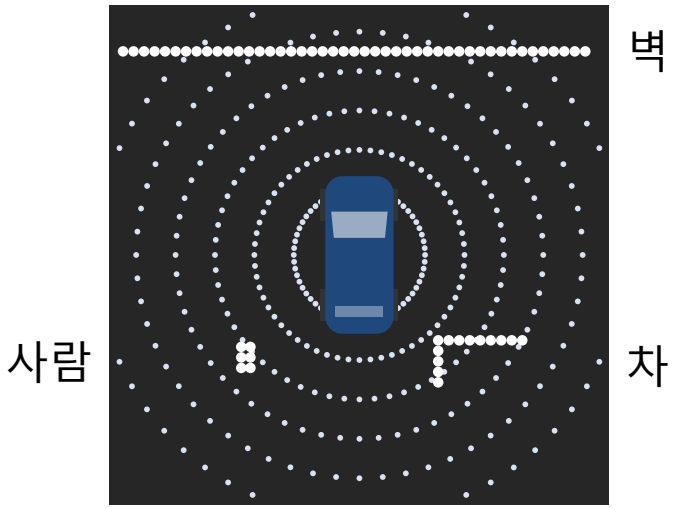
카메라 하나로, 라이다 하나로 주행하는 두 영상.

Depth Estimation and Object Detection for Mono Camera

Depth Estimation + Object Detection

(Hailo-8, Mono Depth2 + MobileNet SSD)

youtu.be/e8HrwjEoxto



라이다 점군 주행 영상 (로컬 재생)

LiDAR (능동 센서, 점군 자율주행)

위 그림은 설명용(영상 장면 아님)

발표 주제 소개 세 가지 연구 주제

1 SLAM: 지도 없이 지도를 만들며 위치를 안다

2 자율주행: 측위 · 인지 · 예측 · 계획 · 제어

3 픽앤플레이스: 깊이 센서 없이 3차원 위치를 구한다

Deployable Onboard Vision-based Pick-and-Place Automation System for Unstructured Environments

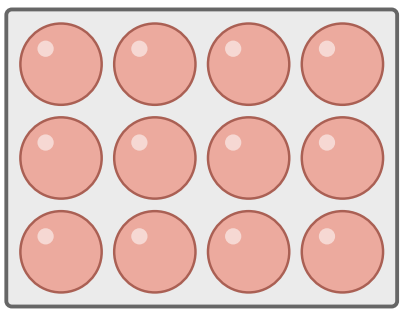
T.-W. Um, S.-W. Kang, K.-H. Kim, G.-H. Kim, H.-S. Ahn, W. Y. Choi, N.-J. Park

IEEE Robotics and Automation Letters, accepted 2026-10-11

[왜 필요한가] 상자 안에 무작위로 쌓인 사과를 집으려면 높이 z 까지 알아야 한다.
깊이 센서 없이 카메라 영상 한 장으로 3차원 위치를 구하는 시스템.

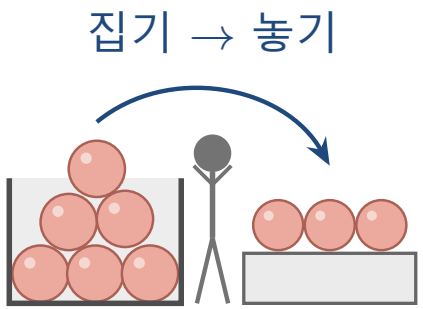
Motivation — 사과 한 상자는 누가 담았을까 (1/5)

마트 · 온라인 주문으로 받는 사과 한 상자 → 사과를 하나씩 집어 담는 작업이 있다.



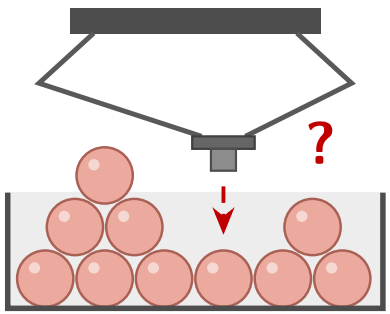
마트의 사과 한 상자

가지런히 담긴 사과
→ 산지에서 누군가
하나씩 집어 담았다



하루 종일 같은 동작

하나 집고, 하나 놓고
→ 단순 반복이라
일할 사람이 부족



로봇이 집는다면?

상자 속 사과는
높이가 제각각
→ 얼마나 내려야 하나?

[배경] 이 작업의 현장: 농산물 유통센터(Agricultural Produce Center, APC).

Motivation — 픽앤플레이스와 델타 로봇 (2/5)

CONCEPT 정의

픽앤플레이스(Pick-and-Place, PNP): 물체를 집어 정해진 자리에 놓는 작업

델타 로봇(delta robot): PNP 에 쓰이는 고속 병렬 로봇(parallel robot)



컨베이어 위 농산물의 델타 로봇 PNP

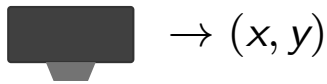


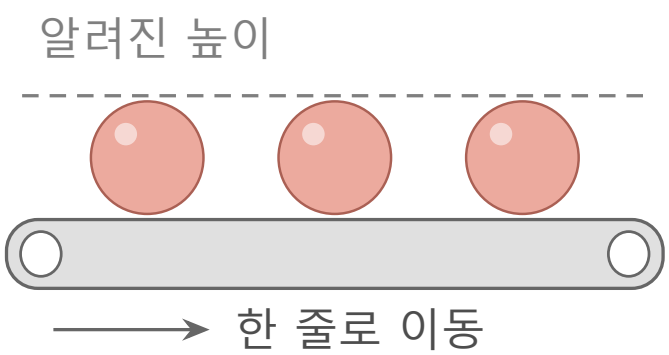
위에서 본 적재 상자(loading box) 속 사과

[직관] 사람이 하던「하나 집고, 하나 놓고」= PNP → 델타 로봇이 빠르게 반복한다.

Motivation — 컨베이어 대 적재 상자 (3/5)

기존: 컨베이어





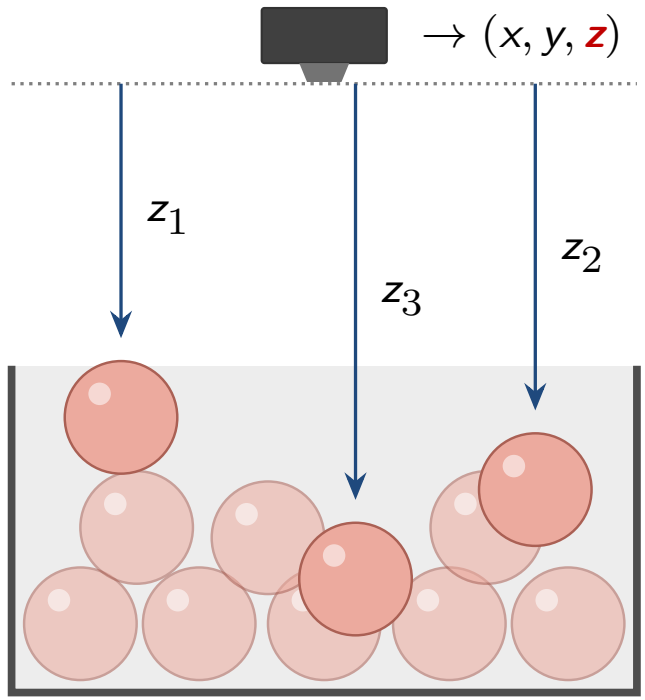
적재 상자

[직관] 기존 PNP 대부분: 평평한 컨베이어 $\rightarrow$ 높이를 알아 비전은 (x, y) 만 구한다.

Motivation — 컨베이어 대 적재 상자 (4/5)

기존: 컨베이어

적재 상자



[직관] 상자: 높이가 제각각 → 충돌 없이 확실하게 잡으려면 ***z* 까지** 구해야 한다.

Motivation — 깊이 측정 대 깊이 추정 (5/5)

CONCEPT 정의

단안 깊이 추정(monocular depth estimation): 영상 한 장 → 픽셀마다 깊이

임베디드 보드(embedded board): 장치에 넣는 소형 컴퓨터, 연산 자원이 제한

RGB-D 카메라 · 라이다(LiDAR)

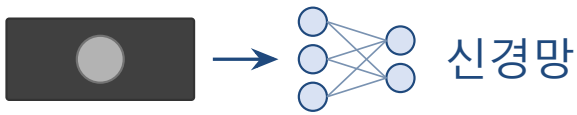
깊이를 직접 측정



– 높은 비용 · 제한된 수급
– 먼지 · 습기: 잦은 재보정 · 하우징
– 대규모 설치의 제약

RGB 카메라 + 단안 깊이 추정

영상 한 장에서 깊이를 추정



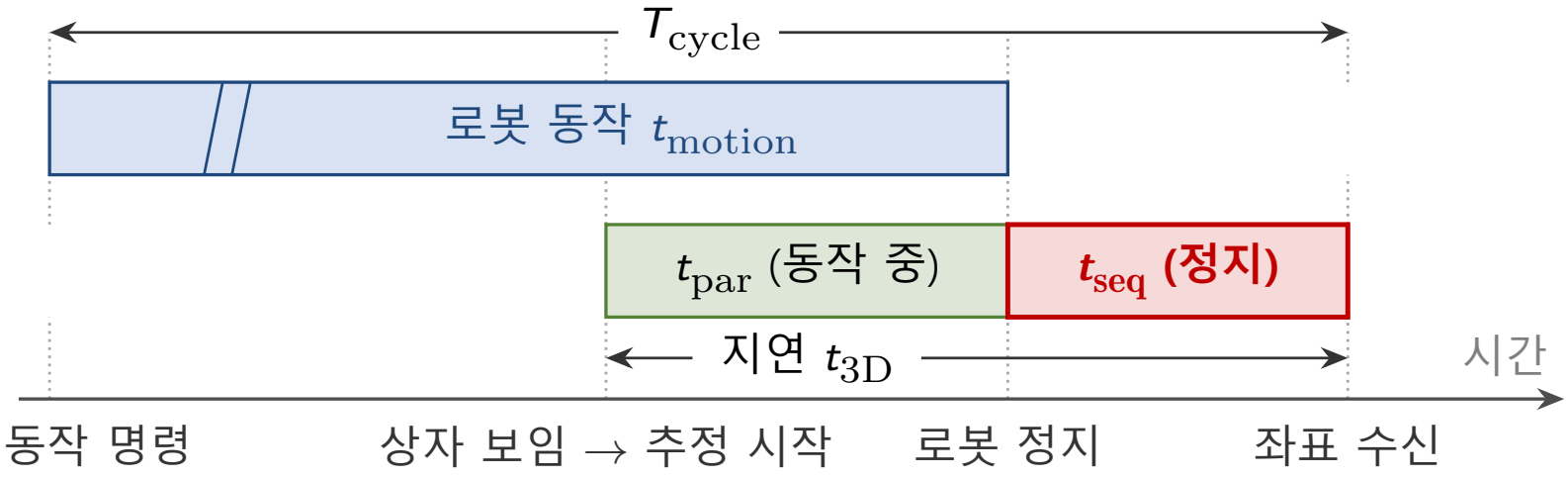
+ 깊이 센서 없음: 왼쪽의 제약 없음
– 파지에 충분한 정확도가 필요
– 신경망 연산 → 지연(latency)

[연구 질문] 단안 깊이 추정으로 **파지에 충분한가**, 임베디드 보드에서 **지연은 얼마인가**.

문제 정식화 — 한 사이클의 시간 (1/2)

CONCEPT 정의

3D 위치 추정(3D localization): 영상에서 목표의 (x, y, z) 를 구하는 과정



$$t_{\text{3D}} = t_{\text{par}} + t_{\text{seq}}, \qquad T_{\text{cycle}} = t_{\text{motion}} + t_{\text{seq}}$$

[직관] 로봇이 움직이는 동안 위치 추정을 시작 → 사이클을 늘리는 것은 t_{seq} 뿐이다.

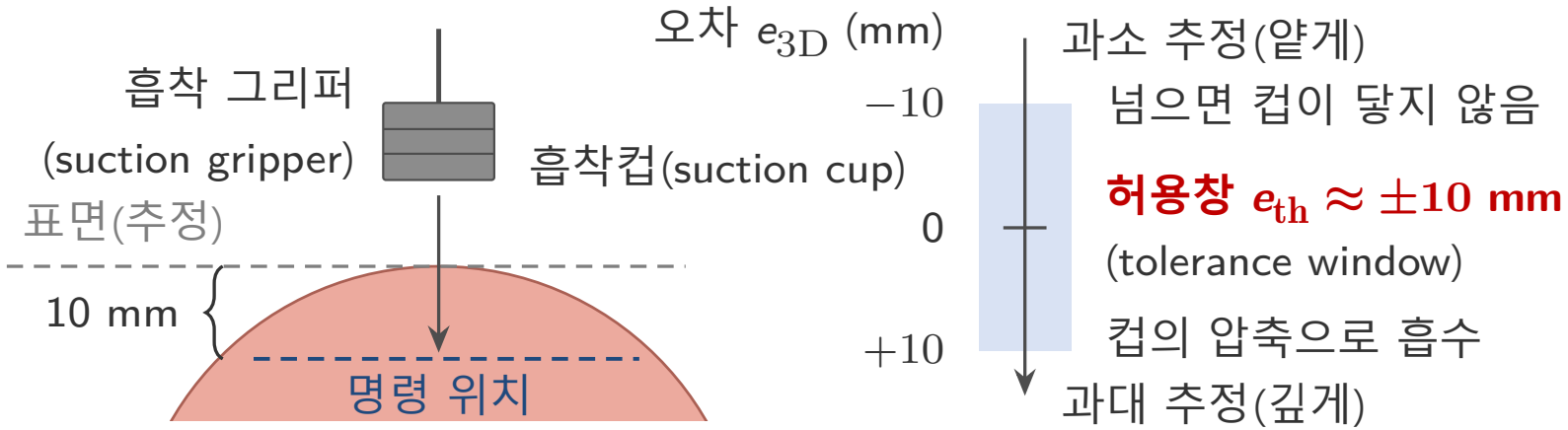
문제 정식화 — 지연-정확도 trade-off (2/2)

CONCEPT 정의

지연-정확도 trade-off(latency-accuracy trade-off): 허용 오차 안의 최소 지연

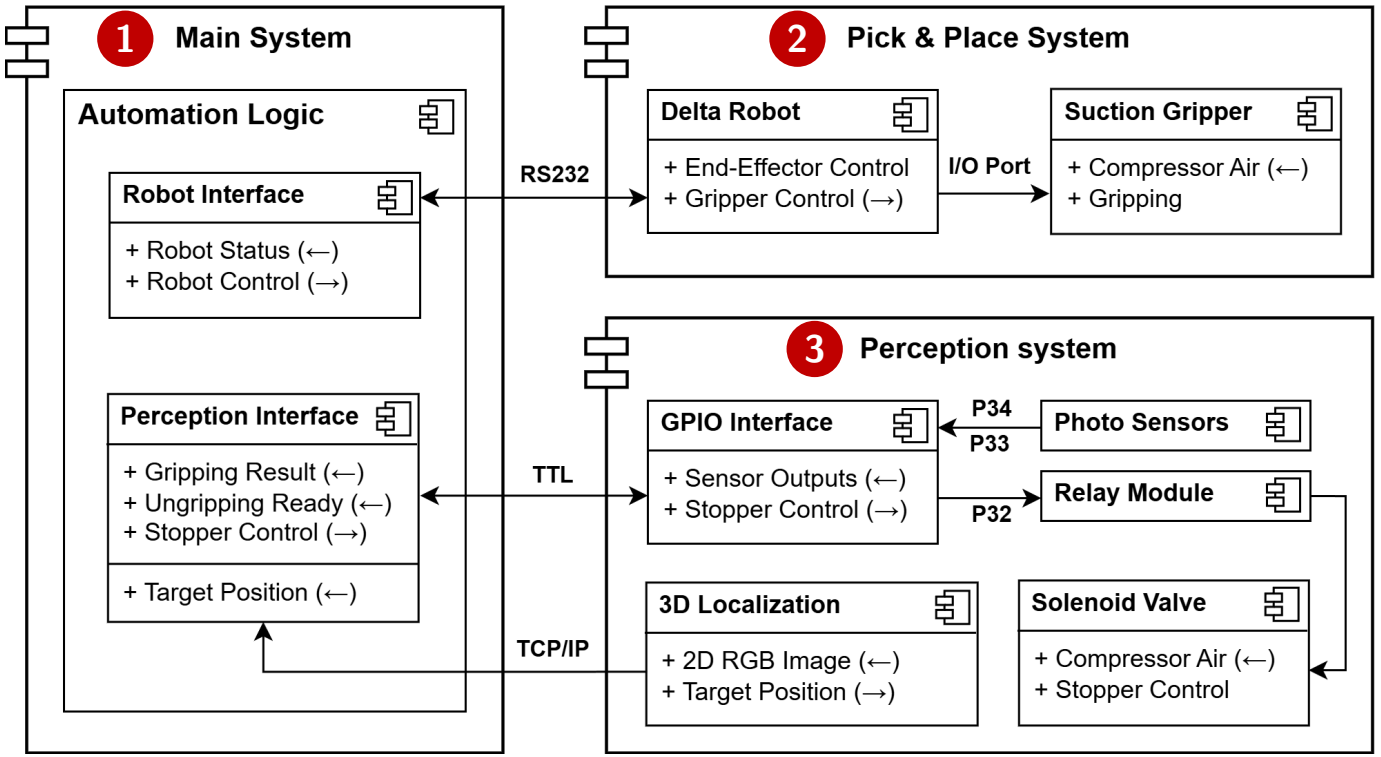
$$\min_{\theta} \; t_{3\mathrm{D}}(\theta) \quad \text{s.t.} \quad e_{3\mathrm{D}}(\theta) \le e_{\mathrm{th}}$$

θ: 모델 구성(예: 신경망 복잡도), $e_{3\mathrm{D}}$: 위치 추정 오차, e_{th}: 그리퍼 형상이 정하는 허용 오차



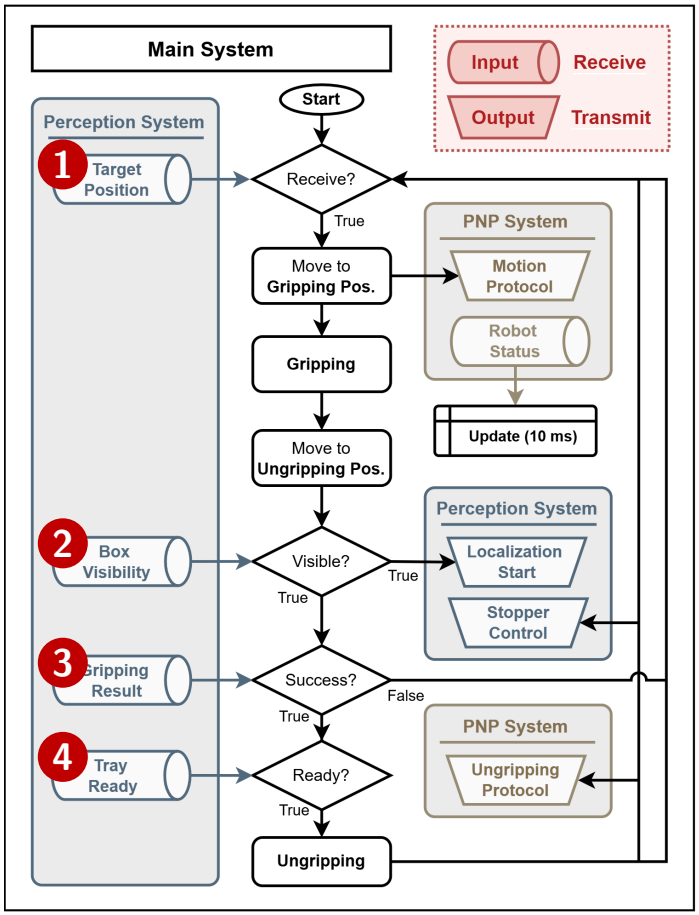
[직관] 가벼운 깊이 모델: 빠르지만 파지점(gripping point) 오차가 큼 → 기준은 e_{th}.

시스템 — 세 시스템 (1/2)



1. Main: 자동화 로직 실행, 델타 로봇에 동작 명령(RS-232)
2. Pick-and-Place: 델타 로봇 + 흡착 그리퍼, 로봇 상태를 10 ms 마다 보고
3. Perception: 머리 위 웹캠 + Jetson Orin NX 16GB(임베디드 보드), 3D 위치 추정

시스템 — Main 의 자동화 로직 (2/2)



1. 목표 위치 수신 → 동작 명령: 잡는 위치로, 파지, 놓는 위치로 복귀
2. 복귀 중 상자가 다 보이면 다음 목표의 위치 추정 요청
3. 파지 실패면 놓기를 건너뛰고 1 로
4. 트레이가 준비되면 놓고 1 로

실린더 = Main 이 받는 신호, 사다리꼴 = 보내는 신호

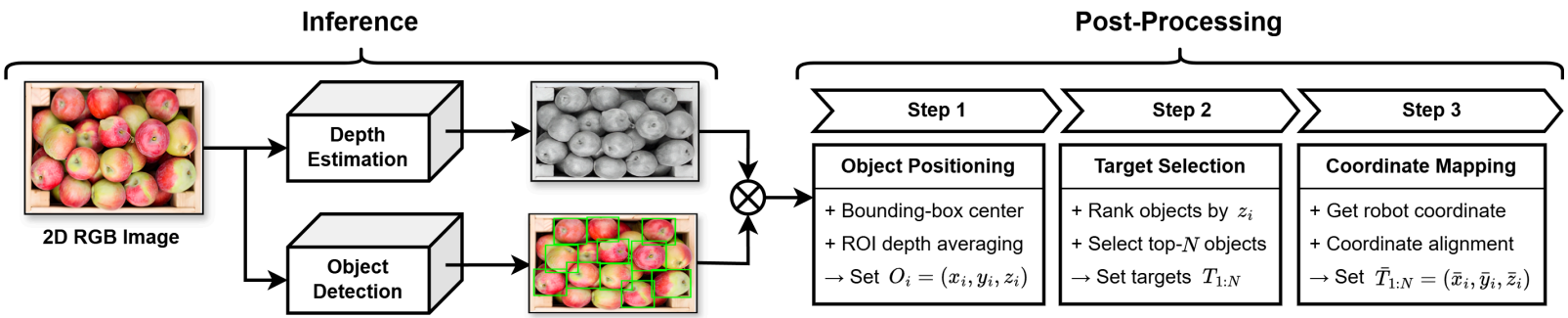
[직관] 2 에서 위치 추정이 로봇 동작과 겹친다 → t_{par} 최대, t_{seq} 최소.

3D 위치 추정 — 추론과 후처리

CONCEPT 정의

객체 검출(object detection): 경계 상자(bounding box) → (x, y), YOLOv8s

단안 깊이 추정: 같은 영상에서 픽셀마다 깊이 → z, AdaBins



후처리: Step 1 물체 위치 → Step 2 목표 선정 → Step 3 로봇 좌표로 매핑

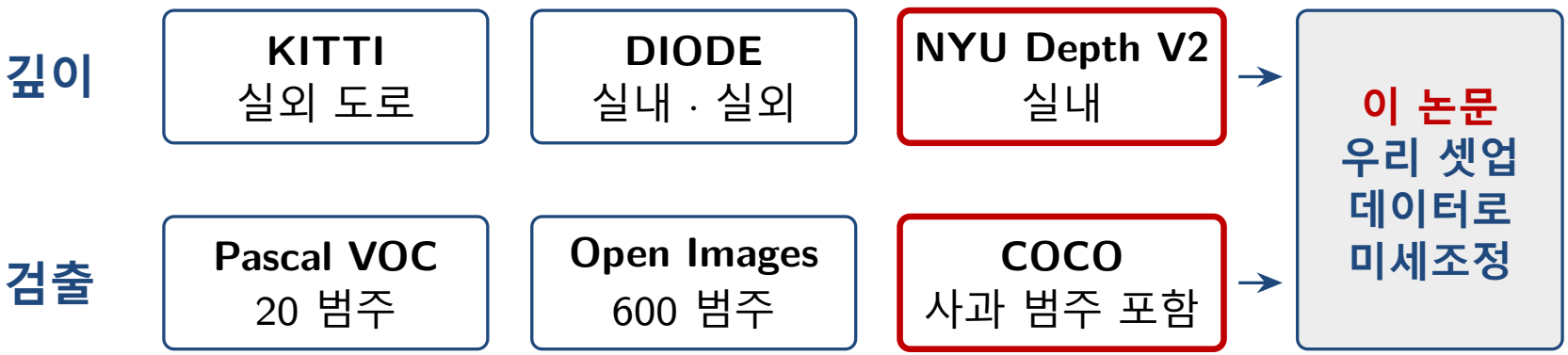
[직관] 깊이 센서(L515 · D435i)는 개발 단계에서만 쓰고, 설치된 시스템에는 없다.

모델 선정 — 사전학습과 미세조정 (1/3)

CONCEPT 정의

사전학습(pre-training): 공개 데이터셋으로 먼저 학습한 모델

미세조정(fine-tuning): 그 모델을 우리 셋업에서 모은 적은 데이터로 다시 학습



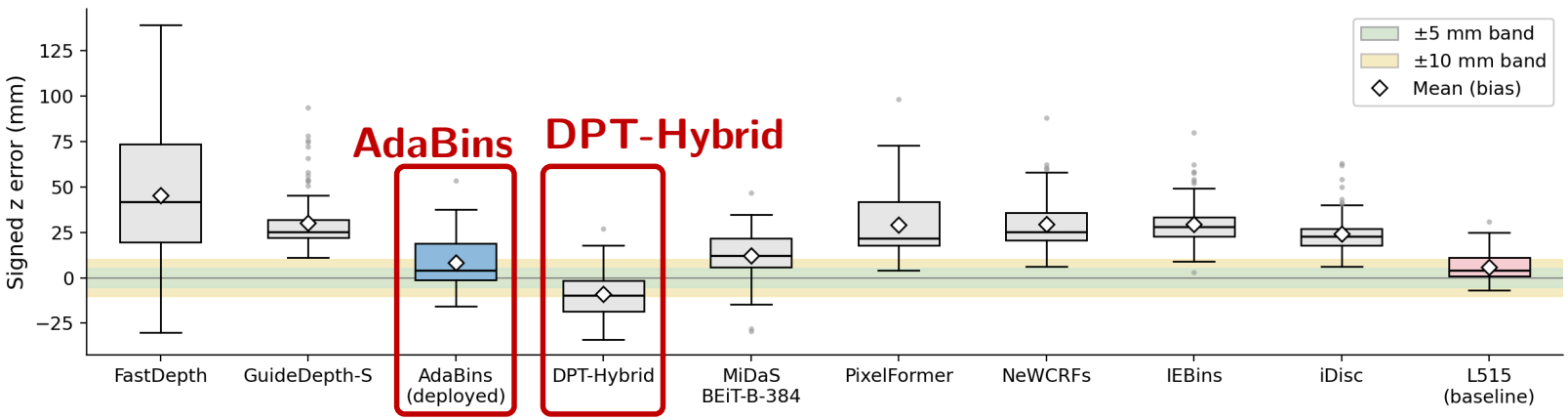
[직관] 공개 셋(빨강)으로 먼저 배우고, 우리 장면은 적은 데이터로 미세조정.

모델 선정 — 실제 접촉 평가 (2/3)

CONCEPT 정의

실제 접촉(physical touch) **평가**: 닿을 때까지 내려 추정 z 의 오차를 측정

깊이 9종 · 검출 5종을 같은 조건으로 **미세조정**(fine-tuning)



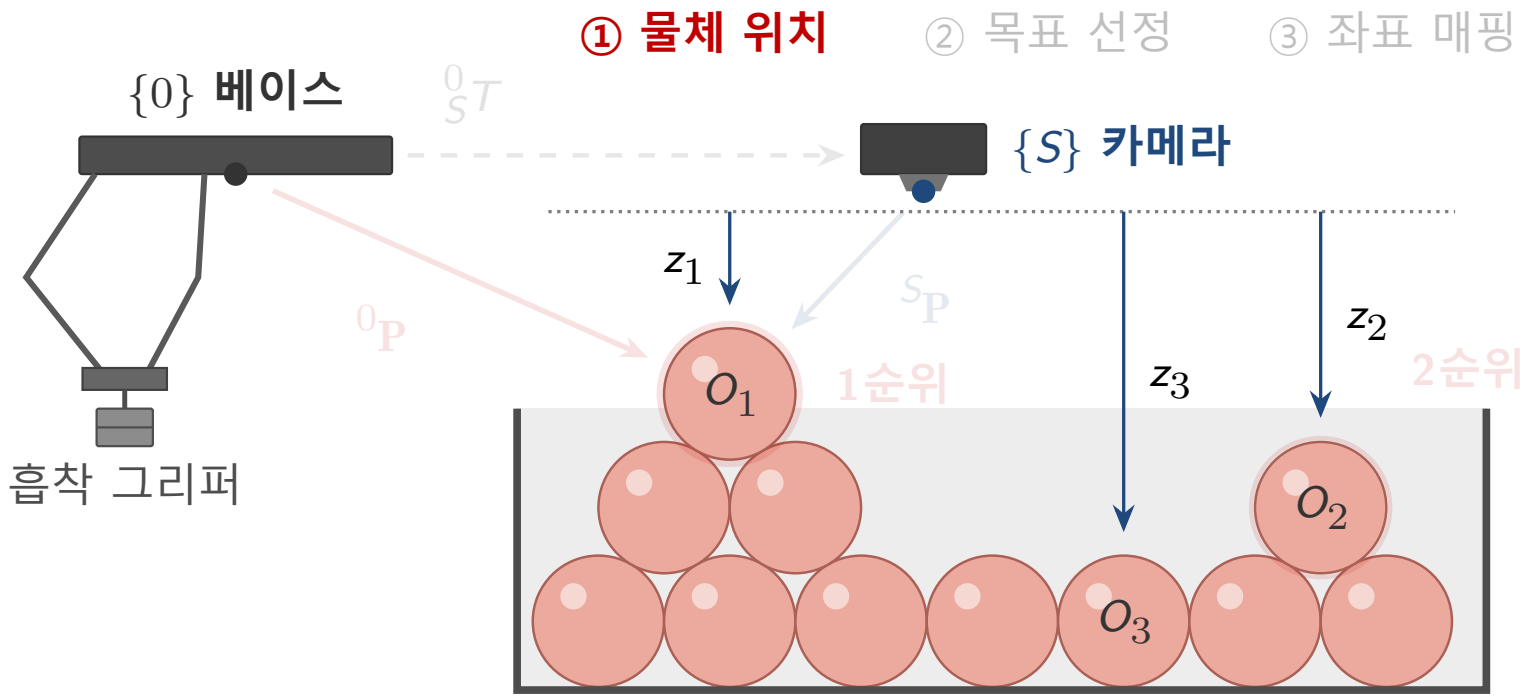
[읽는 법] ◇ = 평균 오차(bias). 깊이 9종 중 ± 10 mm 띠 안은 둘뿐(빨간 테두리).
데이터셋 지표(픽셀 전체 평균)로는 상위 모델 구별 불가 → 파지점에서 직접 측정.

모델 선정 — 깊이 9종의 통합 평가 (3/3)

모델	파라미터 [M]	데이터셋 지표 $\delta_1 \uparrow$	AbsRel ↓	RMSE ↓ [m]	지연 [ms]	실제 접촉 평가(파지점) MAE [mm]	bias [mm]	±5 mm 안	±10 mm 안
FastDepth	3.9	0.9666	0.0869	0.1750	**15.7**	48.1	+45.3	6 %	9 %
GuideDepth-S	5.7	**0.9748**	0.0434	0.1363	20.9	30.3	+30.3	0 %	0 %
AdaBins	78	0.9746	**0.0368**	**0.1330**	239.4	**10.8**	**+8.2**	**44 %**	**64 %**
DPT-Hybrid	123	0.9741	0.0409	0.1387	393.9	12.5	−9.1	16 %	46 %
MiDaS-BEiT	111	0.9744	0.0407	0.1371	407.4	15.1	+12.0	11 %	39 %
PixelFormer	271	0.9621	0.0470	0.1480	605.9	29.2	+29.2	1 %	5 %
NeWCRFs	270	0.9455	0.0630	0.2336	608.9	29.4	+29.4	0 %	2 %
IEBins	272	0.9593	0.0505	0.1627	861.2	29.6	+29.6	1 %	2 %
iDisc	207	0.9705	0.0420	0.1387	1,890.3	24.1	+24.1	0 %	2 %
L515	–	–	–	–	–	7.1	+5.9	51 %	74 %

[결론] 깊이: 띠 안의 둘 중 AdaBins 가 빠르고(239.4 ms) MAE 도 최소(10.8 mm). 검출: F1(정밀도 · 재현율의 조화 평균)이 5종 중 가장 높은 YOLOv8s(0.942, 23.3 ms). (굵게 = 열의 최선 · L515 = 깊이 센서 · bias = 오차 평균 · MAE = 절댓값 평균)

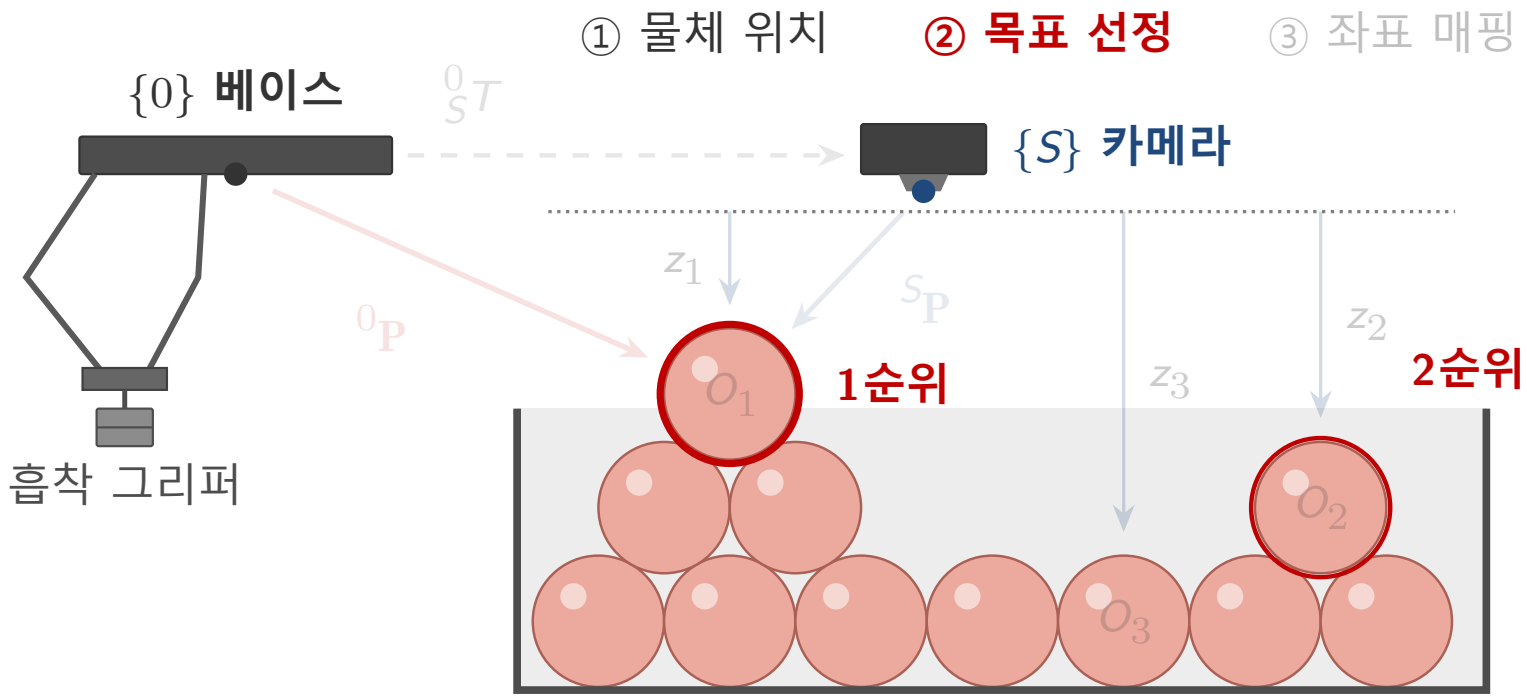
후처리 — 물체 위치에서 로봇 좌표까지 (1/3)



$$O_i = (x_i, y_i, z_i) \Rightarrow T_{1:N} \Rightarrow \bar{T}_{1:N} = (\bar{x}_i, \bar{y}_i, \bar{z}_i)$$

[직관] 경계 상자 중심 $\rightarrow (x_i, y_i)$, 관심 영역(region of interest, ROI) 평균 깊이 $\rightarrow z_i$.
ROI 는 중심 둘레, 카메라에 가까운 사과일수록 z_i 가 작다.

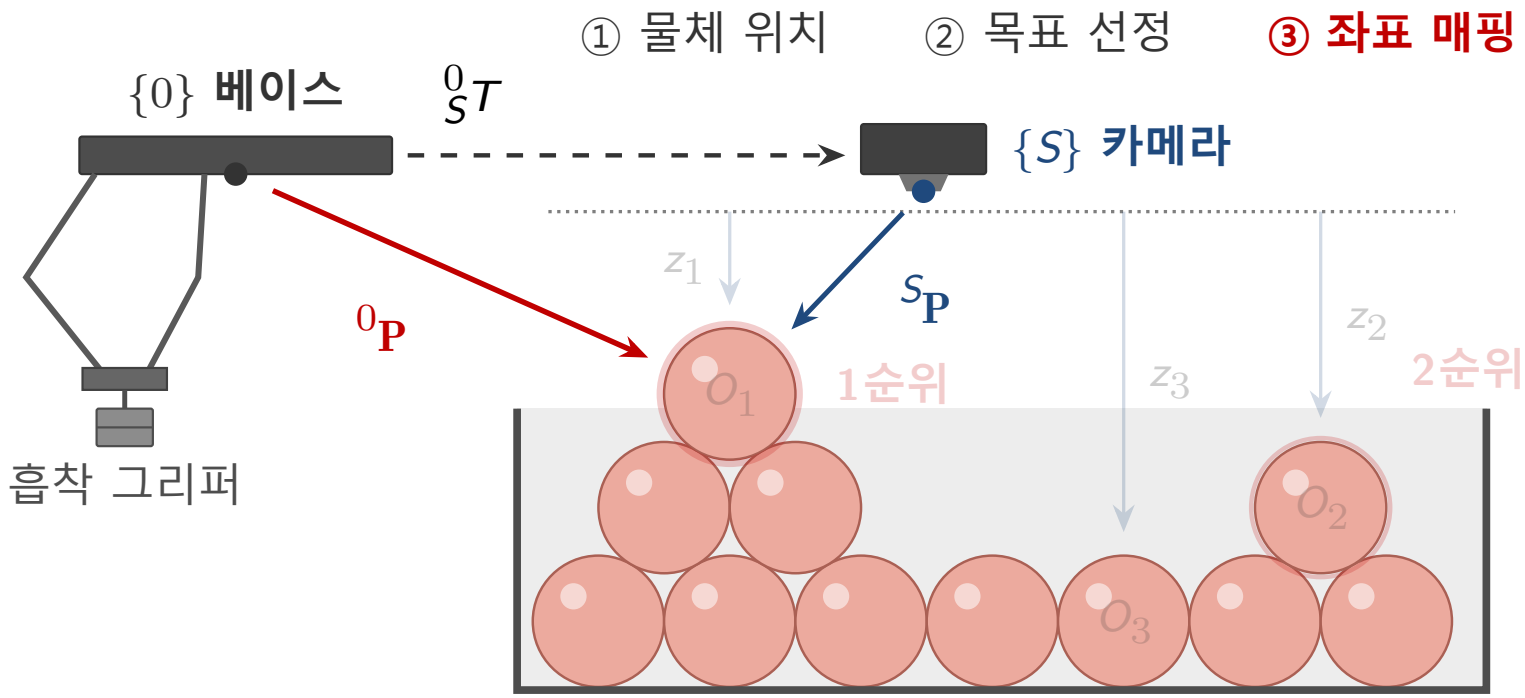
후처리 — 물체 위치에서 로봇 좌표까지 (2/3)



$$O_i = (x_i, y_i, z_i) \;\Rightarrow\; \boldsymbol{T_{1:N}} \;\Rightarrow\; \bar{T}_{1:N} = (\bar{x}_i, \bar{y}_i, \bar{z}_i)$$

[직관] z_i 로 정렬해 가장 높은 것부터 위 N 개를 목표 $T_{1:N}$ 으로(실험에서 $N = 2$). 2순위는 1순위가 이미 두 번 실패한 목표일 때 쓴다.

후처리 — 물체 위치에서 로봇 좌표까지 (3/3)



$$O_i = (x_i, y_i, z_i) \;\Rightarrow\; T_{1:N} \;\Rightarrow\; \bar{T}_{1:N} = (\bar{x}_i, \bar{y}_i, \bar{z}_i)$$

[직관] 좌표 변환 ${}^0\mathbf{P} = {}^0_S T \cdot {}^S\mathbf{P}$ 와 같은 구조.
${}^0_S T$: 로봇 밖에 고정한 카메라의 eye-to-hand 캘리브레이션으로 구한 고정 변환.

결과 — 파지 성공률 (1/2)

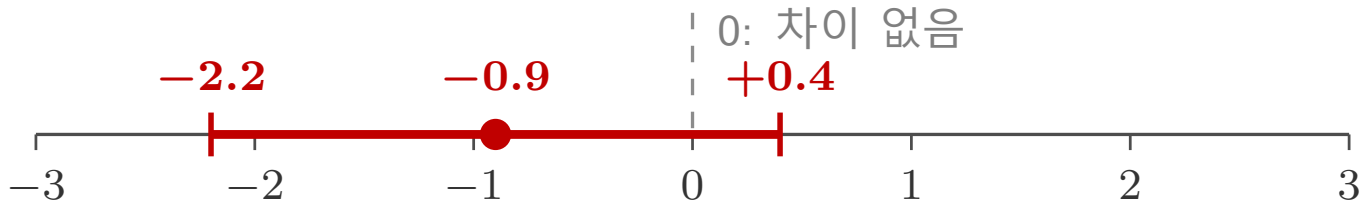
CONCEPT 정의

파지 성공률(gripping success rate): 목표당 성공 비율, 재시도 1회 허용

기준선(baseline): 같은 시스템에서 z 만 RGB-D 카메라(L515)로 직접 측정

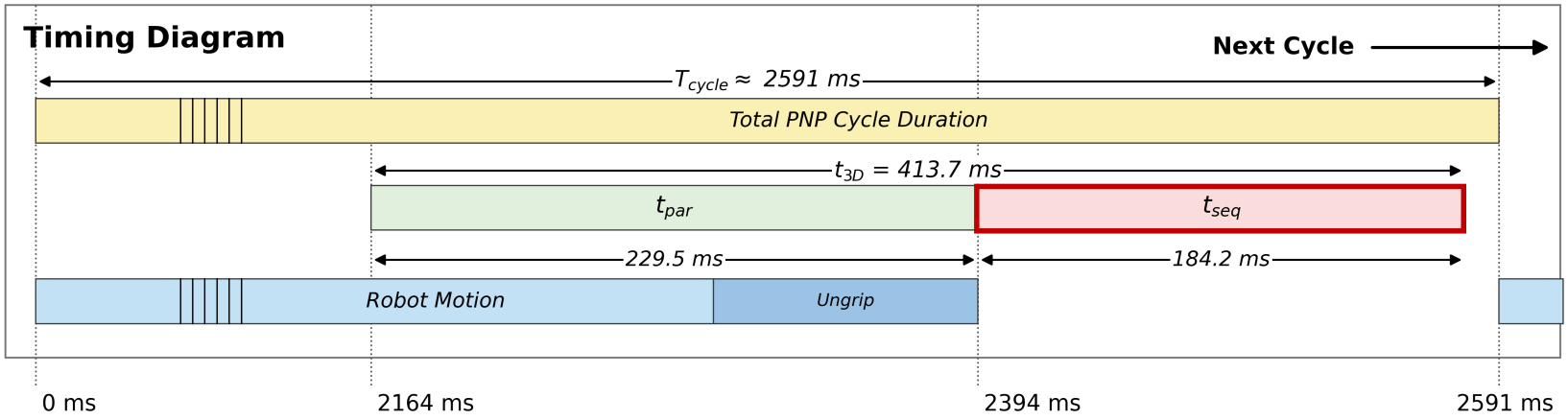
제안: 단안 깊이 추정	기준선: L515(RGB-D)
95.05 %	**95.96 %**
1,958 / 2,060 목표	2,020 / 2,105 목표

차이(제안 − 기준선)의 95 % 신뢰구간(confidence interval), pp = 퍼센트포인트



[결론] 신뢰구간이 0 을 포함 → 유의한 차이 없음(Fisher 정확 검정, $p = 0.156$).
RGB-D 의 우위는 많아야 2.2 pp.

결과 — 한 사이클의 시간 (2/2)



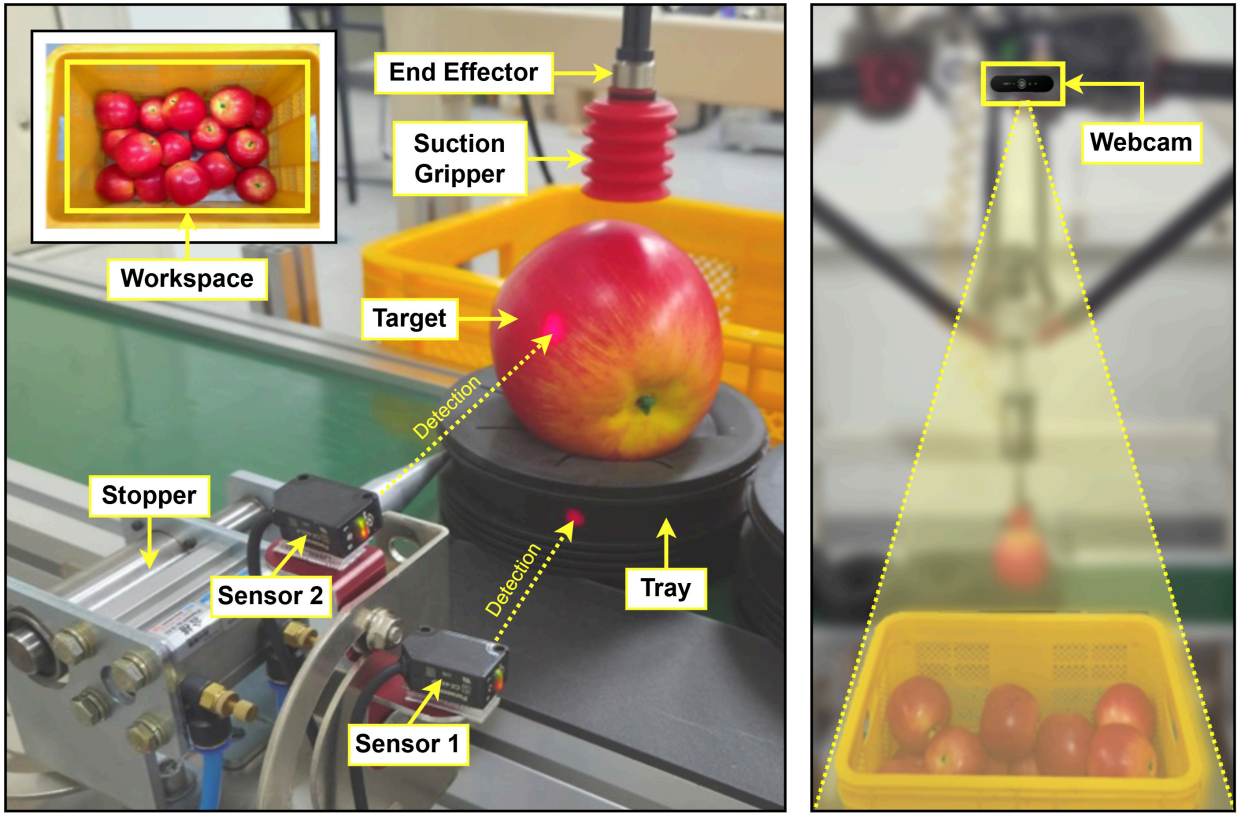
$$t_{3D} = t_{par} + \mathbf{t_{seq}} \approx 229.5 + 184.2 = 413.7 \text{ ms}$$

$$T_{cycle} = t_{motion} + \mathbf{t_{seq}} \approx 2{,}407 + 184.2 \approx 2{,}591 \text{ ms}$$

RGB-D 기준선의 t_{3D} 는 107.3 ms, 차이 306.4 ms 는 깊이 추론(240.1 ms)과 영상 획득 경로.

[결론] 오차가 허용창 안이면 성공률은 포화 → 그 뒤는 **지연이 처리량을 좌우**.
센서 선택이 설계 변수. 결론은 위에서 본 상자 속 사과라는 조건에 한정된다.

영상 — 논문 멀티미디어 자료



실험 장치: 적재 상자 · 흡착 그리퍼 · 머리 위 웹캠 · 트레이

영상: 논문 멀티미디어 자료(실패 · 재시도 장면 포함)