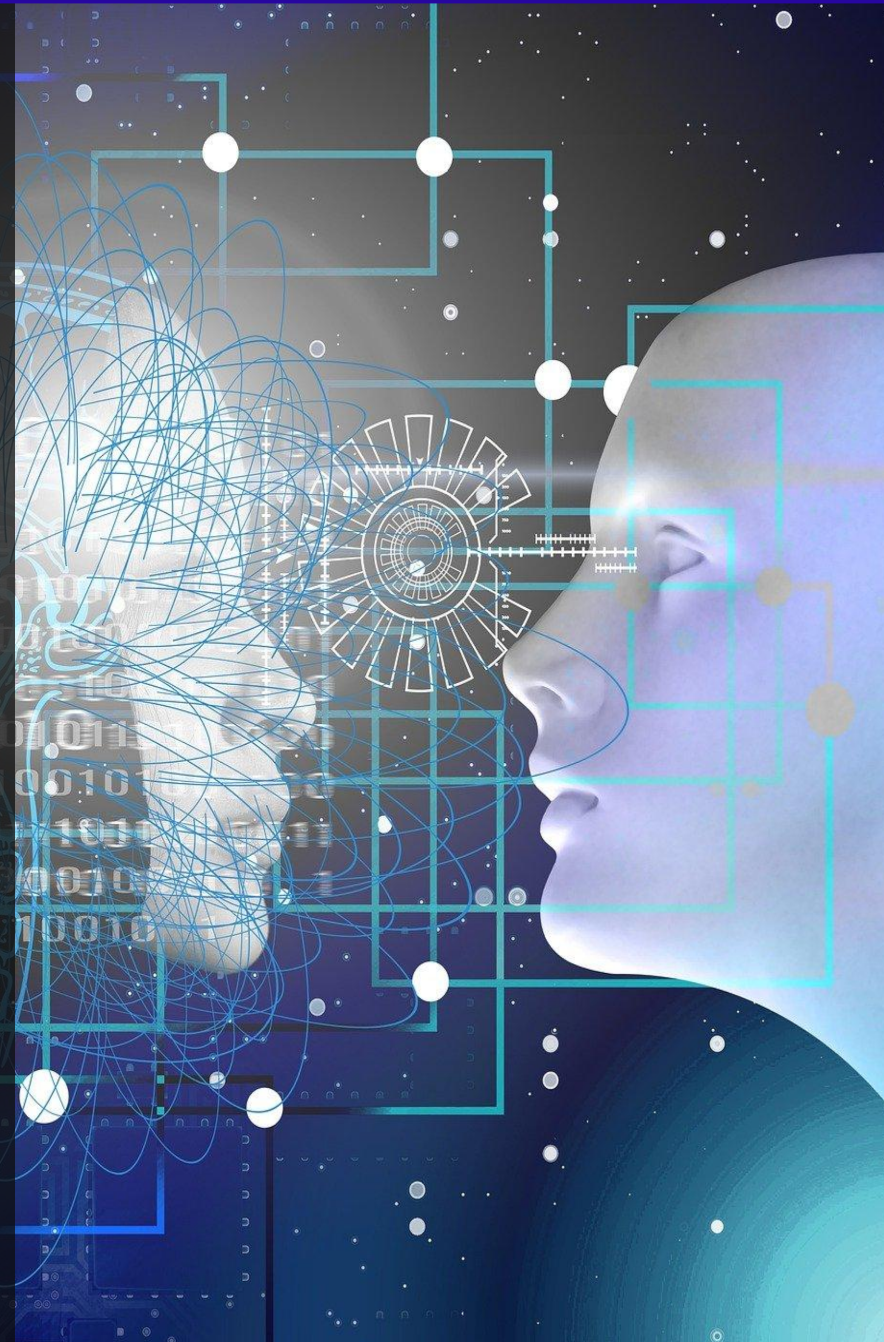
K-Digital Training

ADAS센서원리 및 퓨전이해



한양대학교 전기공학과
Prof. Chang Mook Kang

자율주행 주요 센서

센서 종류	설명	장점	단점	실제 영상 예시
카메라	2D 이미지 획득	고해상도, 저렴함	거리 측정 약함, 빛에 민감	Lane detection 영상
레이더	전파 기반 거리/속도 탐지	거리/속도 정확, 날씨 강인	해상도 낮음	자동차 전방 추적
라이다	레이저 기반 3D 거리 정보	정밀한 3D 인식	고가, 날씨 민감	LiDAR Point Cloud 애니메이션
IMU/GPS	움직임 + 위치 정보	자율 위치 추정에 필수	단독으론 누적 오차	GPS-IMU 통합 지도

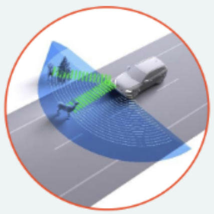
운전자가 수행 운전자가 조건부 수행 시스템이 수행

자율주행 주요 센서

Sensor Fusion

Multiple sensing modalities required; sensor fusion brings them together

RADAR



- Long-range sensing
- Object movement
- All-weather performance

LIDAR

- Precise 3D object detection
- Range accuracy
- Free-space detection

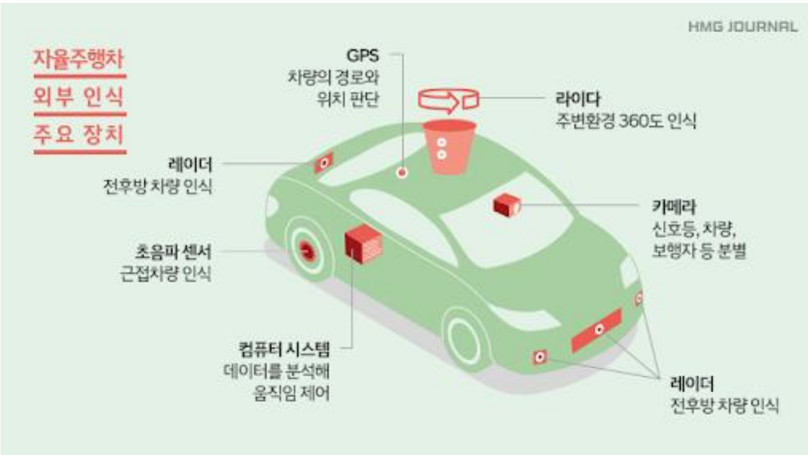
CAMERA

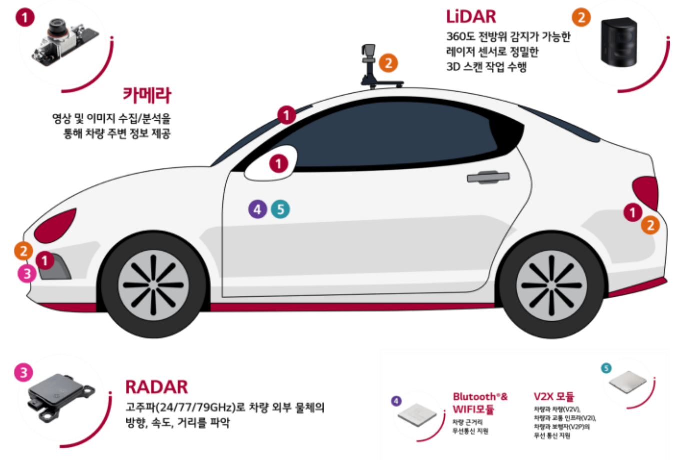
- Object classification
- Object angular position
- Scene context

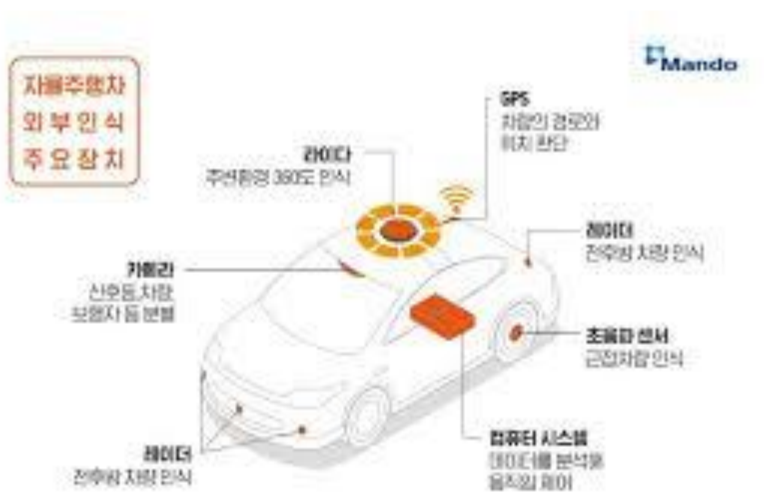
	RADAR	LIDAR	CAMERA	FUSION
Object detection	+	+	O	+
Pedestrian detection	—	O	+	+
Weather conditions	+	O	—	+
Lighting conditions	+	+	—	+
Dirt	+	O	—	+
Velocity	+	O	O	+
Distance - accuracy	+	+	O	+
Distance - range	+	O	O	+
Data density	—	O	+	+
Classification	—	O	+	+
Packaging	+	—	O	+

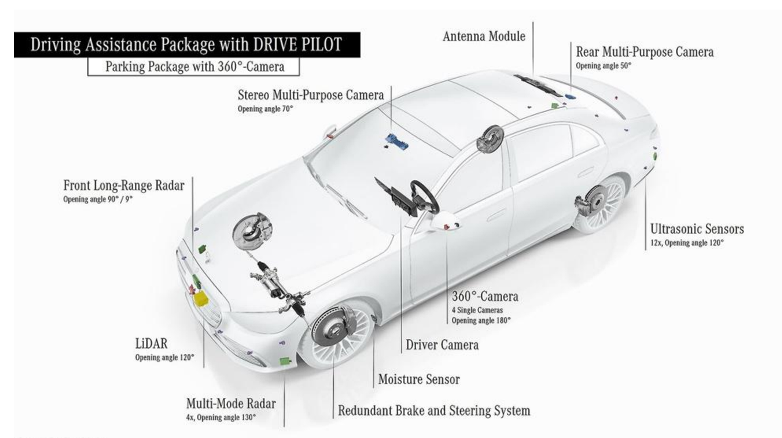
+ = Strength O = Capability — = Weakness

자율주행 주요 센서


HMG JOURNAL
자율주행차
외부 인식
주요 장치
GPS
차량의 경로와
위치 판단
라이다
주변환경 360도 인식
레이더
전후방 차량 인식
카메라
신호등, 차량,
보행자 등 분별
초음파 센서
근접차량 인식
컴퓨터 시스템
데이터를 분석해
움직임 제어
레이더
전후방 차량 인식


카메라
영상 및 이미지 수집/분석을
통해 차량 주변 정보 제공
LiDAR
360도 전방위 감지가 가능한
레이저 센서로 정밀한
3D 스캔 작업 수행
RADAR
고주파(24/77/79GHz)로 차량 외부 물체의
방향, 속도, 거리를 파악
Blutooth®&
WIFI모듈
V2X 모듈


Mando
자율주행차
외부인식
주요장치
GPS
라이다
레이더
카메라
초음파 센서
컴퓨터 시스템
레이더


Driving Assistance Package with DRIVE PILOT
Parking Package with 360°-Camera
Antenna Module
Rear Multi-Purpose Camera
Opening angle 50°
Stereo Multi-Purpose Camera
Opening angle 70°
Front Long-Range Radar
Opening angle 90° / 9°
Ultrasonic Sensors
12x, Opening angle 120°
360°-Camera
4 Single Cameras
Opening angle 180°
LiDAR
Opening angle 120°
Driver Camera
Moisture Sensor
Multi-Mode Radar
4x, Opening angle 130°
Redundant Brake and Steering System

자율주행 주요 센서

자율주행차 브랜드별 센서 탑재 현황

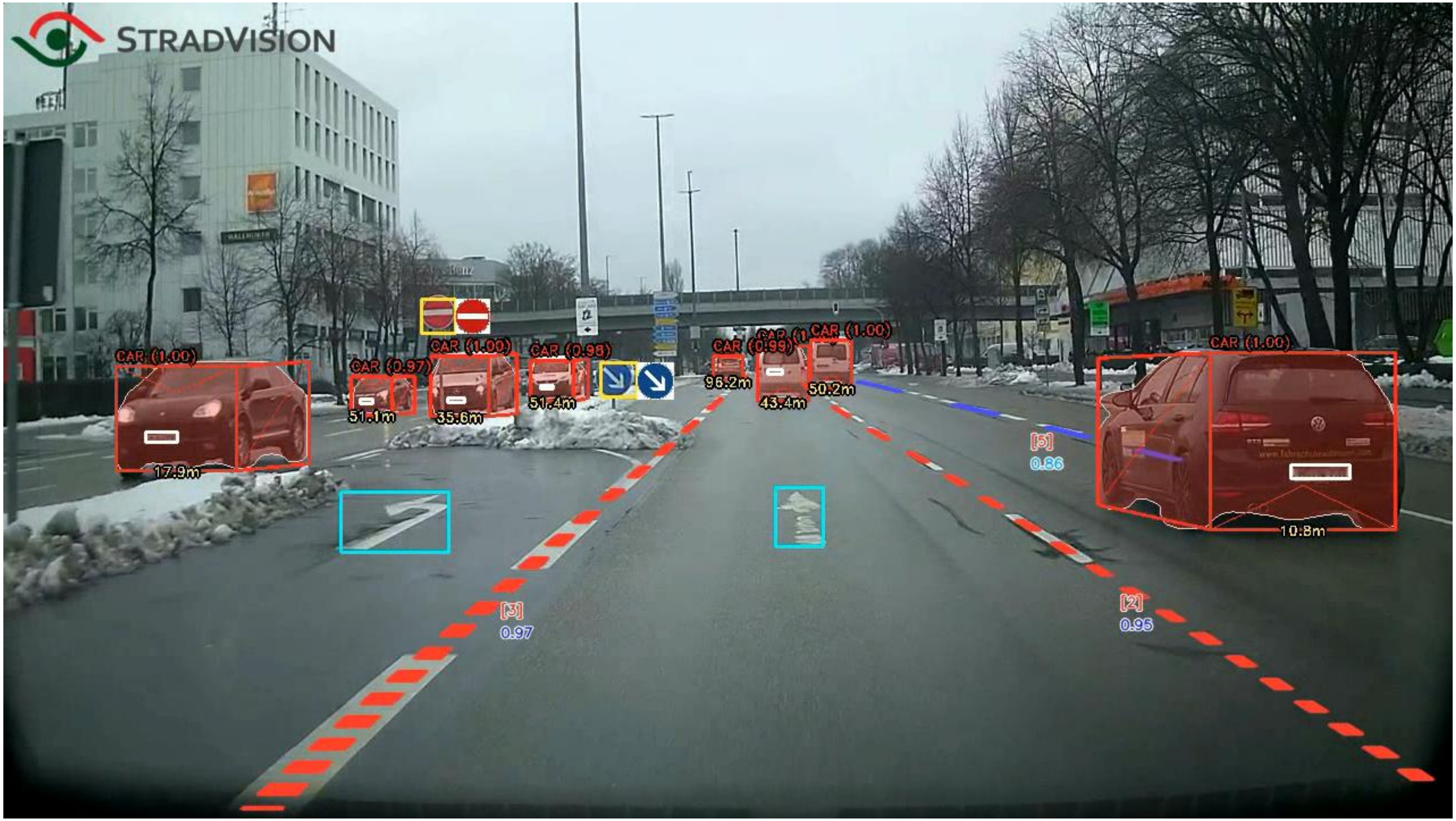
업체(브랜드명)	자율주행 단계	총 개수	카메라	레이더	라이다
테슬라(FSD)	2.5	8	8(100%)		
바이두(아폴로)	3	15	6(40%)	4(27%)	5(33%)
구글(웨이모)	3	18	8(44%)	6(33%)	4(22%)
현대차(아이오닉5)	4	30	13(43%)	10(33%)	7(23%)
바이두(아폴로RT6)	4	38	12(32%)	18(47%)	8(21%)
GM(크루즈)	4	40	14(35%)	21(53%)	21(53%)
합계		149	61(41%)	59(40%)	29(18%)

현대차 아이오닉5는 2023년 출시예정으로 공개된 로보택시 제원 기준. [자료=우리금융경영연구소]

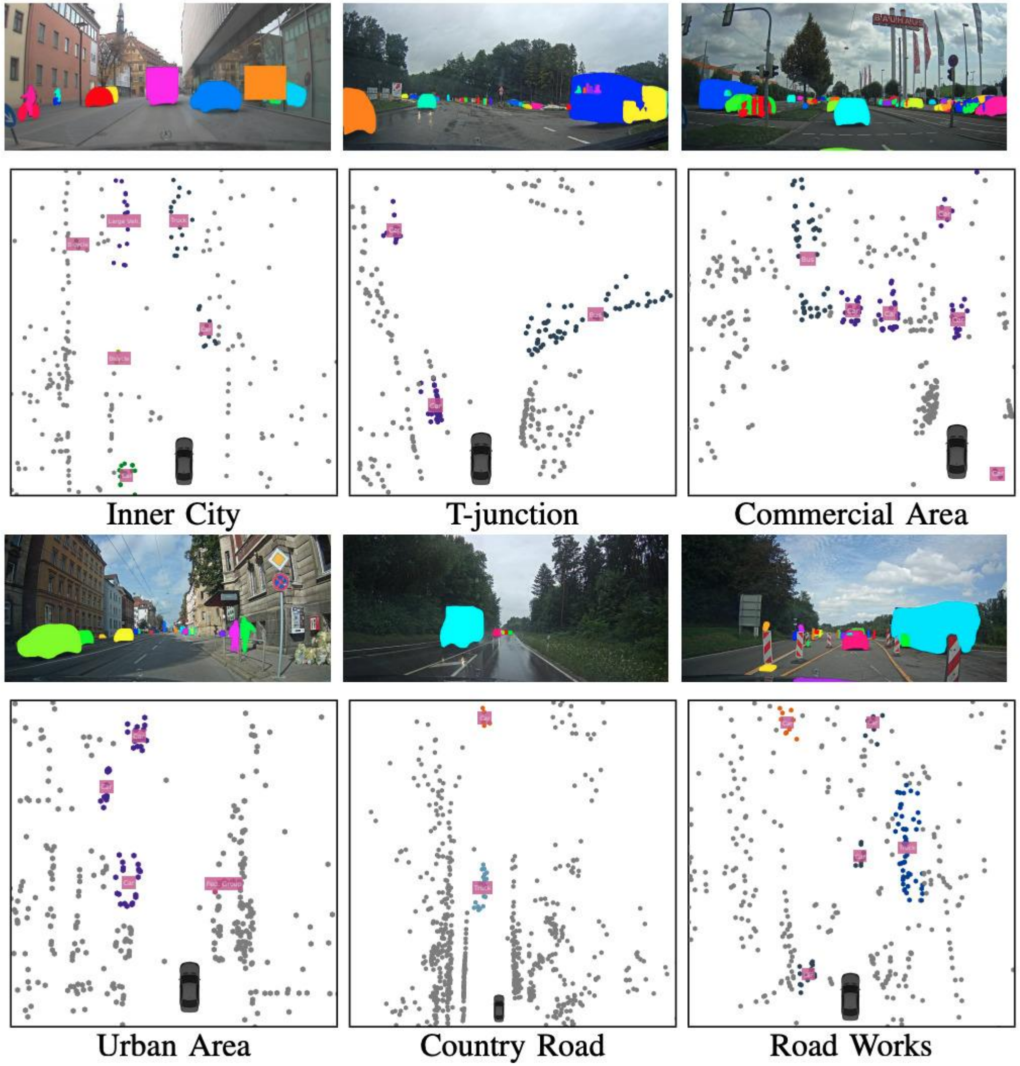
현대아이오닉5 '로보택시'.

Sensor fusion?

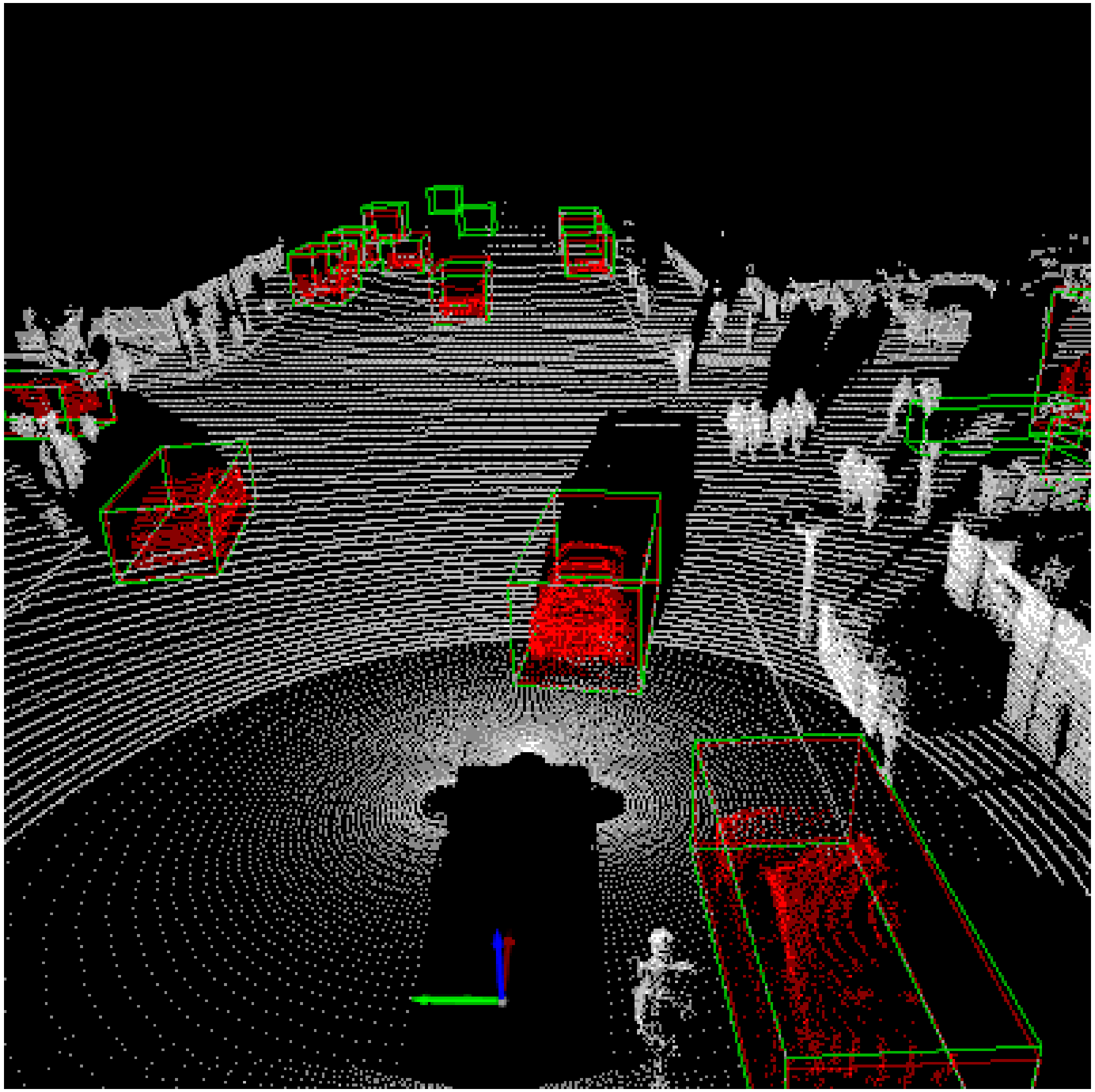
Camera view


STRADVISION
CAR (1.00)
CAR (0.97)
CAR (1.00)
CAR (0.98)
CAR (0.99)
CAR (1.00)
CAR (1.00)
17.9m
51.1m
35.6m
51.4m
96.2m
43.4m
50.2m
10.8m
[3]
0.97
[2]
0.95
[5]
0.86

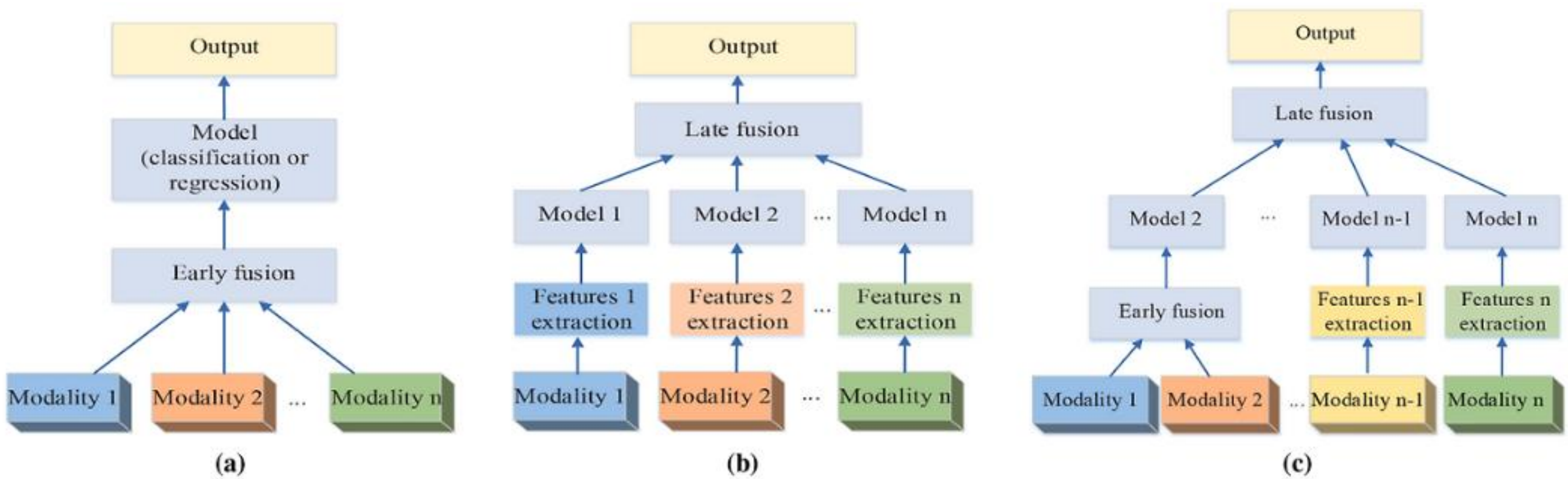
RADAR view


Inner City
T-junction
Commercial Area
Urban Area
Country Road
Road Works

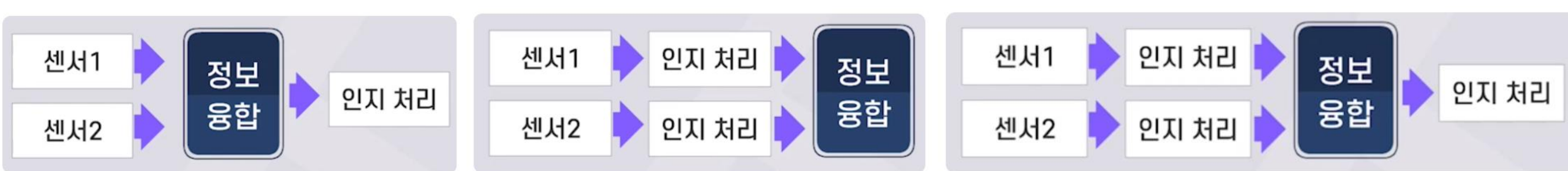
LiDAR view



Multi-sensor data fusion frameworks



(a) Early fusion (data or feature-level); (b) Late fusion (decision-level); (c) Hybrid fusion.



Multi-sensor data fusion frameworks

Raw level fusion / Raw data fusion

- 센서에서 수집한 **원시(raw) 데이터**를 가공하지 않고 직접 합침.
- 센서들의 해상도, 위치 보정, 시간 동기화가 선행되어야 함.

▶ **예시**

- 카메라 이미지의 픽셀 위치와 LiDAR 포인트의 위치를 정렬해서 같은 위치에 표시.
- 레이더 거리 데이터 + 카메라 RGB 영상을 하나의 화면에 오버레이.
- 여러 IMU의 가속도 데이터를 평균해서 노이즈를 줄임.

▶ 장점

- 정보 손실 없이 다양한 특성 결합 가능
- 복합적인 특성을 가진 입력 생성

▶ 단점

- 시간/공간 정렬이 어렵고 계산량 큼
- 센서 특성 차이(해상도, 범위 등) 보정 필요

Multi-sensor data fusion frameworks

Feature level fusion

- 각 센서의 데이터를 먼저 **가공해서 '특징(feature)'만 추출**한 뒤, 이를 결합.
- 특징이란? → 엣지(edge), 코너, 객체 후보 박스 등

▶ **예시**

- 카메라에서 "사람 얼굴"이라는 객체 특징을 추출, 레이더에서 "속도" 정보를 추출 → 합쳐서 "달리는 사람" 판단
- 영상에서 차선 검출, LiDAR에서 평면 특징 검출 → 도로 형태 예측

▶ **장점**

- 센서 특성 간 균형 유지
- 처리 효율이 좋음

▶ **단점**

- 중요한 정보가 feature 추출 단계에서 손실될 수 있음
- 특징 추출 알고리즘의 성능에 의존

Multi-sensor data fusion frameworks

Decision level fusion

- 각 센서가 **독립적으로 판단**을 내린 후, 그 판단을 **결합**해서 최종 결정.

▶ 예시

- 카메라가 "차 있음" 판단 + 레이더도 "차 있음" 판단 → 최종적으로 "앞에 차 있음" 확정
- 차량 주행 중: GPS는 "위치 정확", IMU는 "직진 중" → 판단 일치 시 신뢰도 상승

▶ **장점**

- 구현이 간단하고 유연성 있음
- 센서 간 독립적 유지 가능

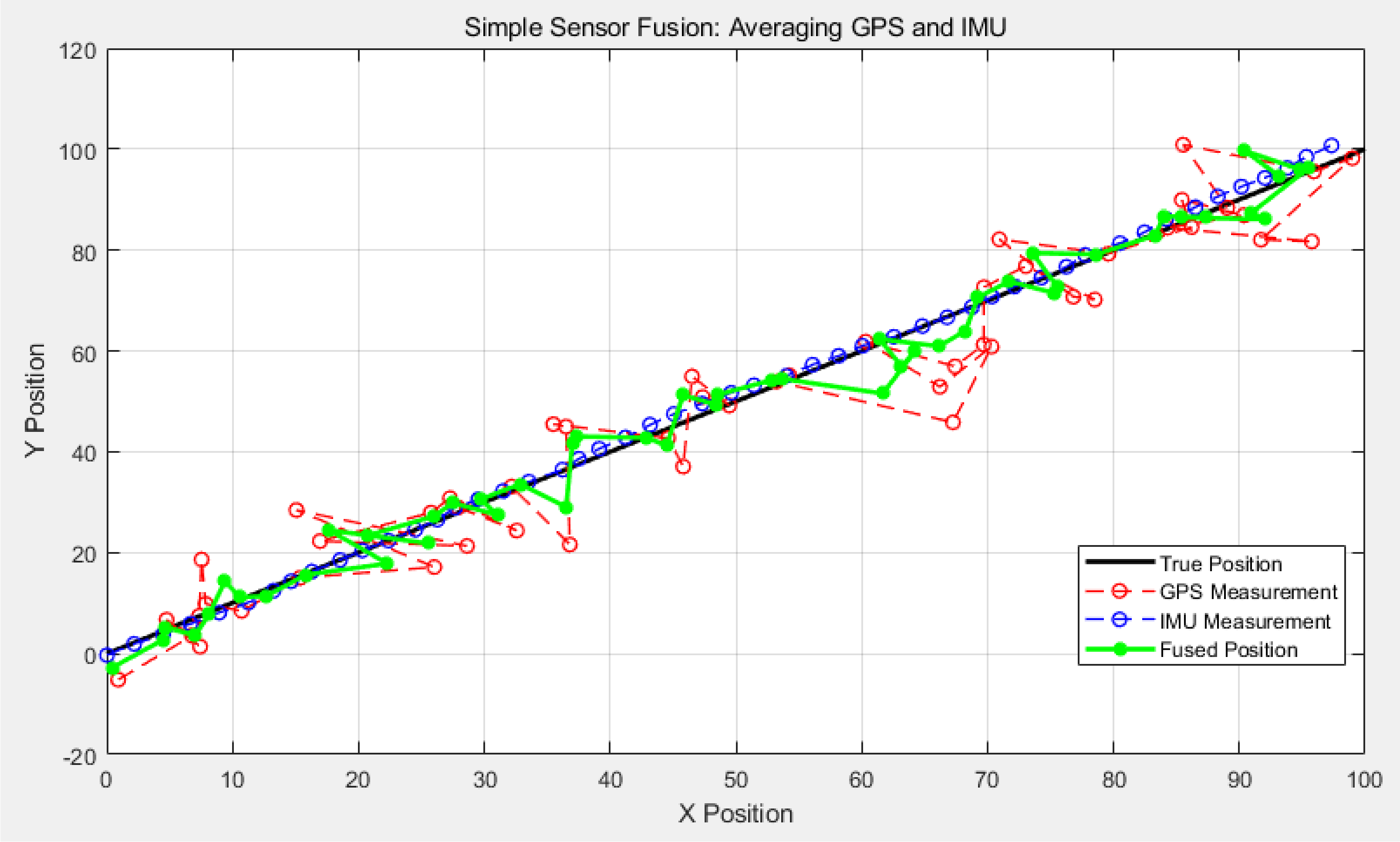
▶ **단점**

- 정보 손실 큼 (센서가 판단한 '결과'만 활용)
- 결정 오류 시 수정 어려움

https://www.youtube.com/watch?v=wKNvzLgTYhQ

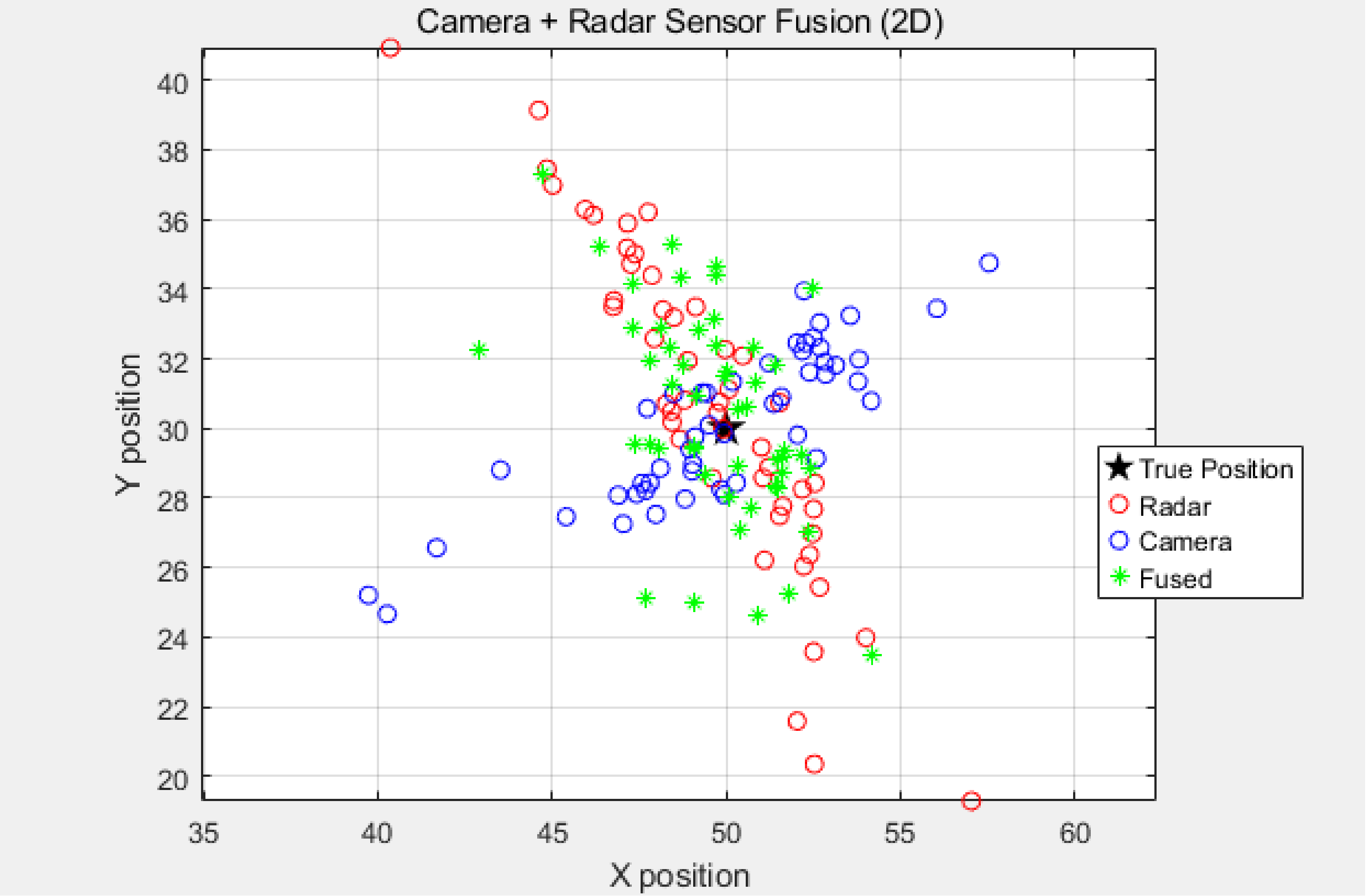
https://www.youtube.com/watch?v=ZauJbv4AdWI

GPS+IMU



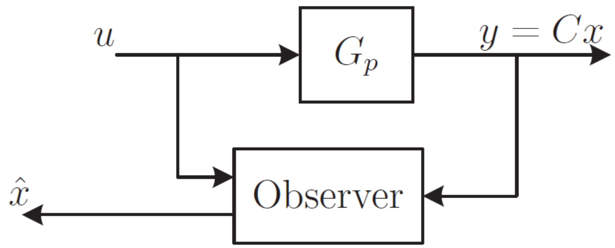
Simple Sensor Fusion: Averaging GPS and IMU
Y Position
X Position
120
100
80
60
40
20
0
-20
0
10
20
30
40
50
60
70
80
90
100
True Position
GPS Measurement
IMU Measurement
Fused Position

Camera + RADAR



State observer

State Observer:



$$G_p : \begin{cases} x(k+1) = \Phi x(k) + \Gamma u(k) \\ y(k) = Cx(k) \end{cases}$$

State observer

$$\hat{x}(k+1) = \Phi \hat{x}(k) + \Gamma u(k) + L(y(k) - C\hat{x}(k))$$

Estimation error $e(k) = x(k) - \hat{x}(k)$

$$e(k+1) = \Phi e(k) - LCe(k) = (\Phi - LC)e(k)$$ state observation error dynamics

The observer gain L can be chosen such that $e \to 0$, irrespective of u (provided it's known and used)

State observer

Prediction-Correction State Observer:

An LTI system

$$G_p : \begin{cases} x(k+1) = \Phi x(k) + \Gamma u(k) \\ y(k) = Cx(k) \end{cases}$$

Prediction:

$$\bar{x}(k+1) = \Phi \hat{x}(k) + \Gamma u(k)$$

Correction:

$$\hat{x}(k) = \bar{x}(k) + L(y(k) - C\bar{x}(k))$$

Estimation error $\bar{e}(k) = x(k) - \bar{x}(k)$

$$\bar{e}(k+1) = (\Phi - \Phi LC)\bar{e}(k)$$ state observation error dynamics

Note: $\bar{e}(\cdot)$ convergent $\Rightarrow$ $e(\cdot) = x(\cdot) - \hat{x}(\cdot)$ convergent.

Kalman filter: white Gaussian noise

Discrete Kalman Filter

Let us consider a discrete-time system

$$G_p : \begin{cases} x_{k+1} = \Phi_k x_k + w_k \\ y_k = C_k x_k + v_k \end{cases}$$

where

w_k : white sequence with known covariance, $\sim N(0, Q_k)$

v_k : white sequence measurement error with known covariance, $\sim N(0, R_k)$

The covariance matrices for the w_k and v_k

$$\mathbb{E}[w_k w_k^T] = Q_k, \;\; \mathbb{E}[w_k w_j^T] = 0 \;\; (j \neq k)$$

$$\mathbb{E}[v_k v_k^T] = R_k, \;\; \mathbb{E}[v_k v_j^T] = 0 \;\; (j \neq k)$$

$$\mathbb{E}[w_k v_j^T] = 0 \;\; (\forall k, \; j)$$

Kalman filter

Prediction (*a priori*) estimate $\bar{x}_k$

Prediction (*a priori*) estimation error

$$\bar{e}_k = x_k - \bar{x}_k$$

Prediction (*a priori*) error covariance matrix

$$\bar{\Sigma}_k = \mathbb{E}[\bar{e}_k \bar{e}_k^T] = \mathbb{E}[(x_k - \bar{x}_k)(x_k - \bar{x}_k)^T]$$

How to use the measurement y_k to improve the prior estimate $\bar{x}_k$

Kalman filter

We choose

$$\hat{x}_k = \bar{x}_k + L_k(y_k - C_k\bar{x}_k)$$

where

$\hat{x}_k$: the updated (*a posteriori*) estimate

L_k : a gain to be determined

How to find the gain L_k that yields an updated estimate that is optimal in some sense

- minimum mean-square error as a performance criterion

Kalman filter

Updated (*a posteriori*) estimation error:

$$e_k = x_k - \hat{x}_k$$

The covariance associated with the updated estimate error:

$$\Sigma_k = \mathbb{E}[e_k e_k^T] = \mathbb{E}[(x_k - \hat{x}_k)(x_k - \hat{x}_k)^T]$$

Using

$$\hat{x}_k = \bar{x}_k + L_k(y_k - C\bar{x}_k) = (I - L_k C)\bar{x}_k + L_k C x_k + L_k v_k$$

$$e_k = x_k - \hat{x}_k = (I - L_k C)\bar{e}_k - L_k v_k$$

we have

$$\Sigma_k = \mathbb{E}[e_k e_k^T] = (I - L_k C)\bar{\Sigma}_k (I - L_k C)^T + L_k R_k L_k^T$$

(The *a priori* estimation error $\bar{e}_k$ uncorrelated with v_k, $\mathbb{E}[\bar{e}_k v_k^T] = 0$)

Kalman filter: optimization

Optimization:

Need to solve an optimization problem

$$\min_{L_k} \operatorname{tr}[\Sigma_k]$$

subject to Convergence

where

$$\begin{aligned}\Sigma_k = \mathbb{E}[e_k e_k^T] &= (I - L_k C_k)\bar{\Sigma}_k (I - L_k C_k)^T + L_k R_k L_k^T \\ &= \bar{\Sigma}_k - L_k C_k \bar{\Sigma}_k - \bar{\Sigma}_k C_k^T L_k^T + L_k (C_k \bar{\Sigma}_k C_k^T + R_k) L_k^T\end{aligned}$$

Good to note:

For $A = [a_{kl}] \in \mathbb{C}^{n\times m}$, $B = [b_{kl}] \in \mathbb{C}^{n\times m}$,

$$\operatorname{tr}\left[A^T B\right] = \sum_{k=1}^{n}\sum_{l=1}^{m} a_{kl} b_{kl} = a_{11}b_{11} + \cdots + a_{nm}b_{nm}$$

If $A = B$,

$$\operatorname{tr}\left[A^T A\right] = \sum_{k=1}^{n}\sum_{l=1}^{m} a_{kl}^2 = a_{11}^2 + \cdots + a_{nm}^2$$

Kalman filter: optimization

$$\frac{d\mathrm{tr}[\Sigma_k]}{dL_k} = -2(C_k\bar{\Sigma}_k)^T + 2L_k(C_k\bar{\Sigma}_kC_k^T + R_k) = 0$$

Optimal gain

$$\boxed{L_k = \bar{\Sigma}_kC_k^T(C_k\bar{\Sigma}_kC_k^T + R_k)^{-1} \qquad \text{(Kalman gain)}}$$

The covariance matrix associated with the optimal estimate

$$\Sigma_k = (I - L_kC_k)\bar{\Sigma}_k(I - L_kC_k)^T + L_kR_kL_k^T$$

Substituting the optimal gain leads to

$$\boxed{\Sigma_k = (I - L_kC_k)\bar{\Sigma}_k}$$

Kalman filter

Prediction model

$$\bar{x}_{k+1} = \Phi_k \hat{x}_k$$

The error covariance matrix associated with $\bar{x}_{k+1}$

$$\bar{e}_{k+1} = x_{k+1} - \bar{x}_{k+1} = \Phi_k x_k + w_k - \Phi_k \hat{x}_k = \Phi_k e_k + w_k$$

The prediction error covariance matrix $\bar{\Sigma}_{k+1} = \mathbb{E}[\bar{e}_{k+1}\bar{e}_{k+1}^T]$

$$\boxed{\bar{\Sigma}_{k+1} = \Phi_k \Sigma_k \Phi_k^T + Q_k}$$

($\mathbb{E}[e_k w_k^T] = 0,\ e_k$ uncorrelated with w_k)

Kalman filter

Convergence?

Assume stationary, fixed sample period process

$$\bar{\Sigma}_{k+1} = \Phi \Sigma_k \Phi^T + Q_k = (\Phi - \Phi L_k C)\bar{\Sigma}_k(\Phi - \Phi L_k C)^T + \Phi L_k R_k L_k^T \Phi^T + Q_k$$

assures

$$(\Phi - \Phi L_k C)\bar{\Sigma}_k(\Phi - \Phi L_k C)^T - \bar{\Sigma}_{k+1} + Q_k < 0$$

Note:

- Steady state prediction error dynamics
- $\bar{\Sigma} > 0$ for the Lyapunov inequality

Kalman filter: summary

Discrete Kalman Filter Algorithm:

- Correction update (using measurement y_k):

$$L_k = \bar{\Sigma}_k C_k^T (C_k \bar{\Sigma}_k C_k^T + R_k)^{-1}$$

$$\hat{x}_k = \bar{x}_k + L_k (y_k - C_k \bar{x}_k)$$

$$\Sigma_k = (I - L_k C_k) \bar{\Sigma}_k$$

- Prediction update:

$$\bar{x}_{k+1} = \Phi_k \hat{x}_k$$

$$\bar{\Sigma}_{k+1} = \Phi_k \Sigma_k \Phi_k^T + Q_k$$

Kalman filter: summary

Time invariant system: steady state soln.

Using

$$\Sigma_k = (I - LC)\bar{\Sigma}_k$$

and

$$\bar{\Sigma}_{k+1} = \Phi \Sigma_k \Phi^T + Q$$

we obtain

$$\bar{\Sigma}_{k+1} = \Phi (I - LC)\bar{\Sigma}_k \Phi^T + Q$$

Let $Y = \bar{\Sigma}_\infty$, then

$$\boxed{L = YC^T(CYC^T + R)^{-1}}$$

$$Y = \Phi(I - LC)Y\Phi^T + Q$$

Substituting $I - LC = I - YC^T(CYC^T + R)^{-1}C$ leads to

$$\boxed{Y = \Phi Y \Phi^T - \Phi Y C^T(CYC^T + R)^{-1}CY\Phi^T + Q}$$

Extended Kalman filter

Extended Kalman Filter

Most realistic robotic problems involve nonlinear functions

$$\begin{cases} x_{k+1} = f_k(x_k, u_{k+1}) + w_k \\ z_k = h_k(x_k) + v_k \end{cases}$$

The covariance matrices for w_k and v_k

$$\mathbb{E}[w_k w_k^T] = Q_k, \;\; \mathbb{E}[w_k w_j^T] = 0 \;\; (j \neq k)$$

$$\mathbb{E}[v_k v_k^T] = R_k, \;\; \mathbb{E}[v_k v_j^T] = 0 \;\; (j \neq k)$$

$$\mathbb{E}[w_k v_j^T] = 0 \;\; (\forall k, \; j)$$

Extended Kalman filter

Kalman Filter Revisited:

- Correction update (using measurement z_k):

$$L_k = \bar{\Sigma}_k C_k^T (C_k \bar{\Sigma}_k C_k^T + R_k)^{-1} \qquad \text{(Kalman gain)}$$

$$\hat{x}_k = \bar{x}_k + L_k (z_k - C_k \bar{x}_k)$$

$$\Sigma_k = (I - L_k C_k) \bar{\Sigma}_k$$

- Prediction update:

$$\bar{x}_{k+1} = \Phi_k \hat{x}_k$$

$$\bar{\Sigma}_{k+1} = \Phi_k \Sigma_k \Phi_k^T + Q_k$$

Extended Kalman filter

EKF Linearization: First Order Taylor Series Expansion

- Prediction:

$$f(x_{k-1}, u_k) \approx f(\hat{x}_{k-1}, u_k) + \frac{\partial f(x_{k-1}, u_k)}{\partial x_{k-1}}\Big|_{\hat{x}_{k-1}} (x_{k-1} - \hat{x}_{k-1})$$
$$= f(\hat{x}_{k-1}, u_k) + \Phi_k (x_{k-1} - \hat{x}_{k-1})$$

- Correction:

$$h(x_k) \approx h(\bar{x}_k) + \frac{\partial h(x_k)}{\partial x_k}\Big|_{\bar{x}_k} (x_k - \bar{x}_k)$$
$$= h(\bar{x}_k) + C_k (x_k - \bar{x}_k)$$

Extended Kalman filter

Extended Kalman Filter Algorithm:

- Correction update (using measurement z_k):

$$L_k = \bar{\Sigma}_k C_k^T (C_k \bar{\Sigma}_k C_k^T + R_k)^{-1}$$

$$\hat{x}_k = \bar{x}_k + L_k(z_k - h_k(\bar{x}_k))$$

$$\Sigma_k = (I - L_k C_k)\bar{\Sigma}_k$$

- Prediction update:

$$\bar{x}_{k+1} = f_k(\hat{x}_k, u_{k+1})$$

$$\bar{\Sigma}_{k+1} = \Phi_k \Sigma_k \Phi_k^T + Q_k$$

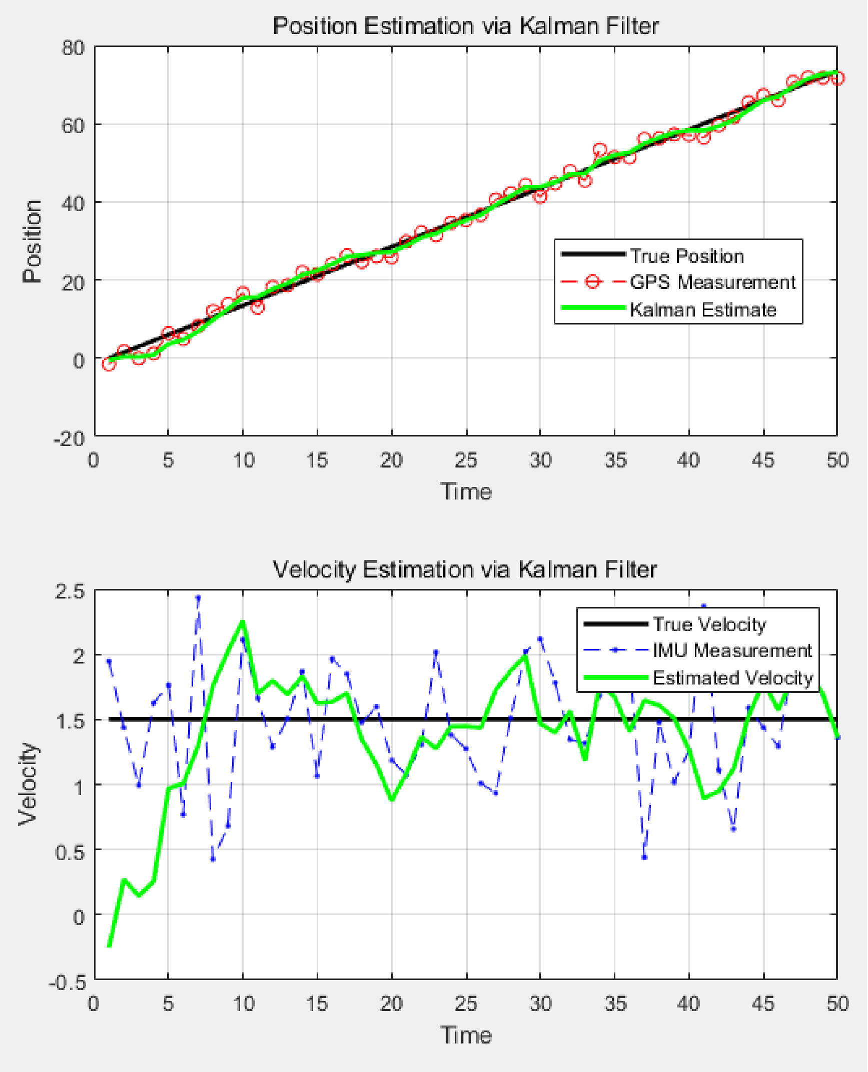
$$\Phi_k = \left.\frac{\partial f(x_{k-1}, u_k)}{\partial x_{k-1}}\right|_{\hat{x}_{k-1}}, \quad C_k = \left.\frac{\partial h(x_k)}{\partial x_k}\right|_{\bar{x}_k}$$

GPS+IMU (CV)

등속(등속도, Constant Velocity, CV) 모델

$$\begin{bmatrix} x_{k+1} \\ v_{k+1} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_k \\ v_k \end{bmatrix} + w_k$$

- 위치 업데이트: $x_{k+1} = x_k + v_k \cdot \Delta t$
- 속도는 일정: $v_{k+1} = v_k$

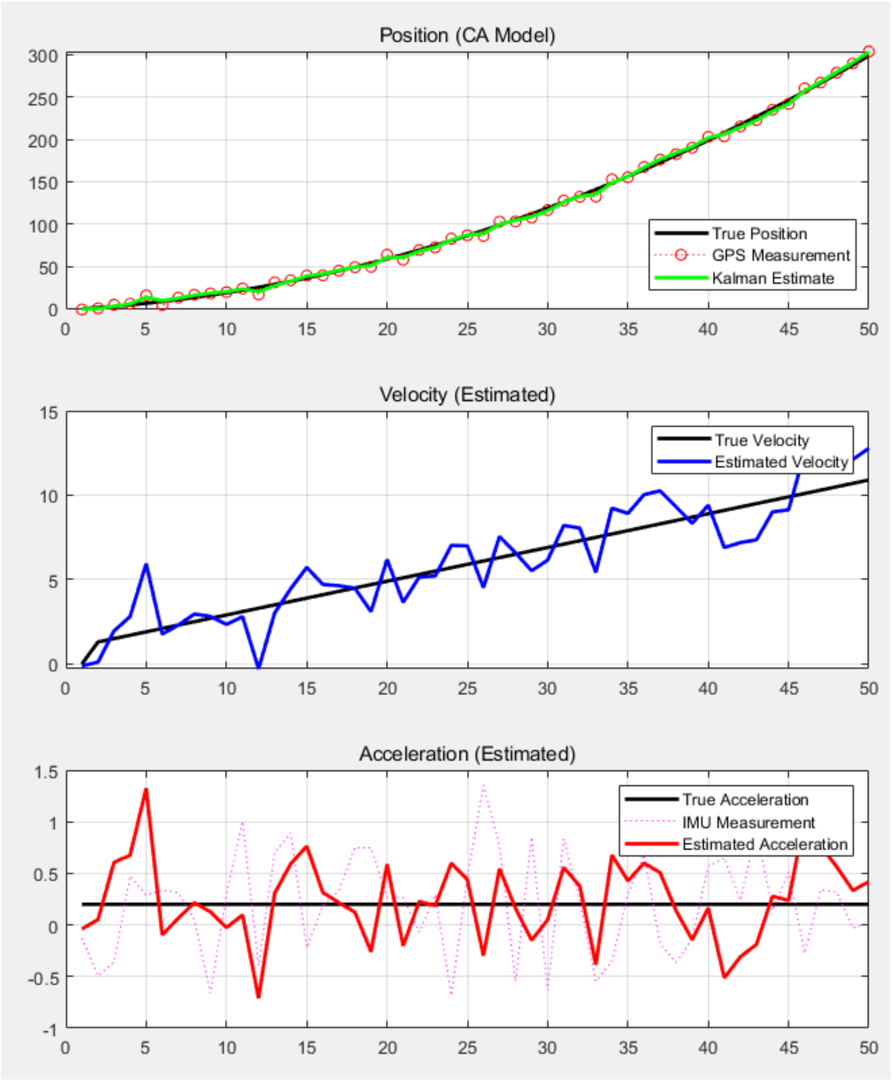


GPS+IMU (CA)

등가속도 모델(Constant Acceleration, CA)

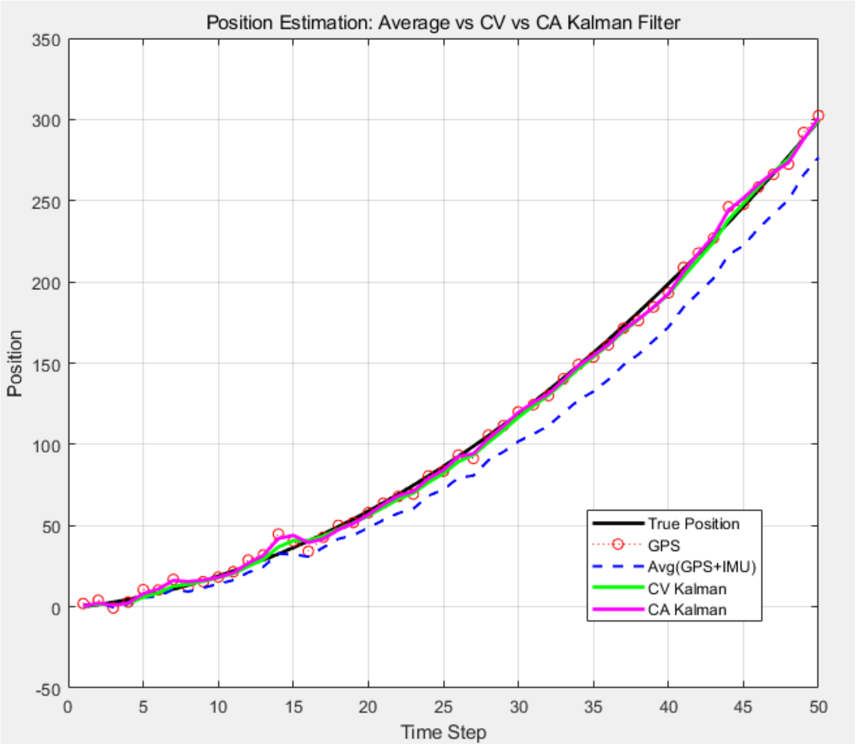
$$x = \begin{bmatrix} \text{position} \\ \text{velocity} \\ \text{acceleration} \end{bmatrix}$$

$$A = \begin{bmatrix} 1 & dt & 0.5dt^2 \\ 0 & 1 & dt \\ 0 & 0 & 1 \end{bmatrix}$$



GPS+IMU (Average, CV, CA)

1. **단순 평균** (GPS + IMU 가속도 기반 추정)
2. **등속 모델 Kalman Filter**
3. **등가속 모델 Kalman Filter**



- 실제 객체는 **등가속 운동** (position + velocity + acceleration)
- GPS: noisy한 위치 측정만 제공
- IMU: noisy한 가속도 측정만 제공
- 세 가지 추정 방식으로 **동일한 참값을 기반으로 추정**하고 비교