Lecture 4 | Localization and Sensor Fusion

Secure Autonomous Systems

CSCI 6907/3907 88

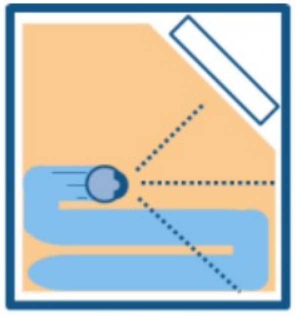
Fall 2022

Prof. Sibin Mohan

https://bit.ly/secureauto-fall22

Localization

Without



With

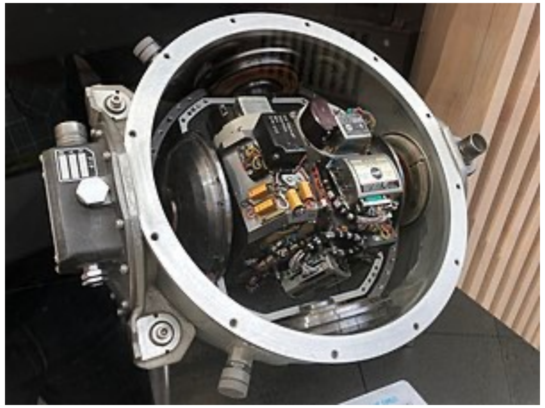
- We can use GPS to determine where we are
    - Not very precise ➔ errors from 1 to 10 meters

- Methods to localize?

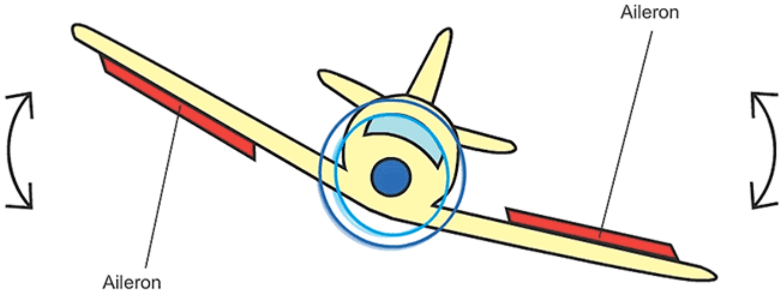
Localization Methods

- Odometry
- Movement of vehicles
- Kalman Filters
- Particle Filters
- SLAM

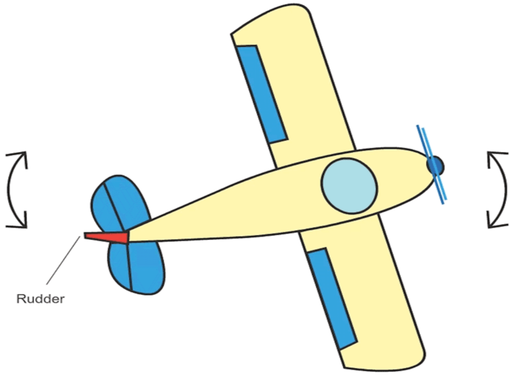
Inertial Measurement Units [IMUs]



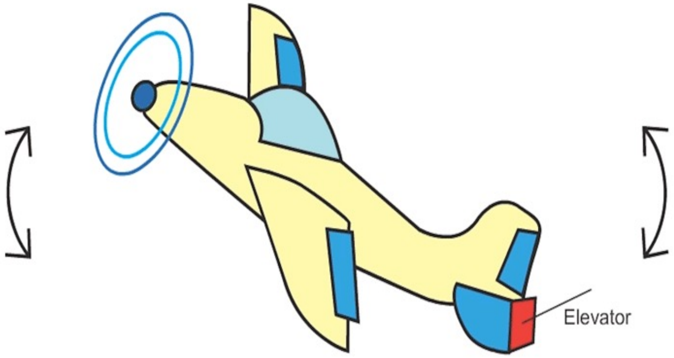
- Sensor to define **movement of vehicle**



Roll

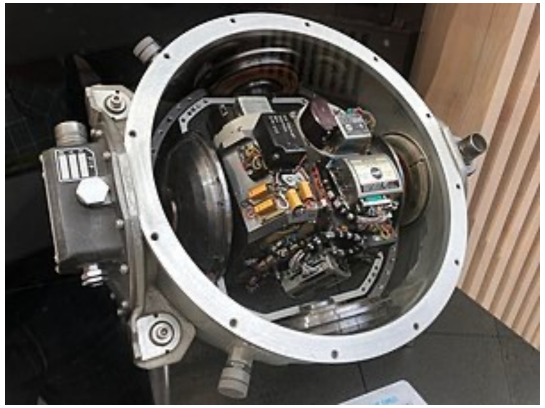


Yaw

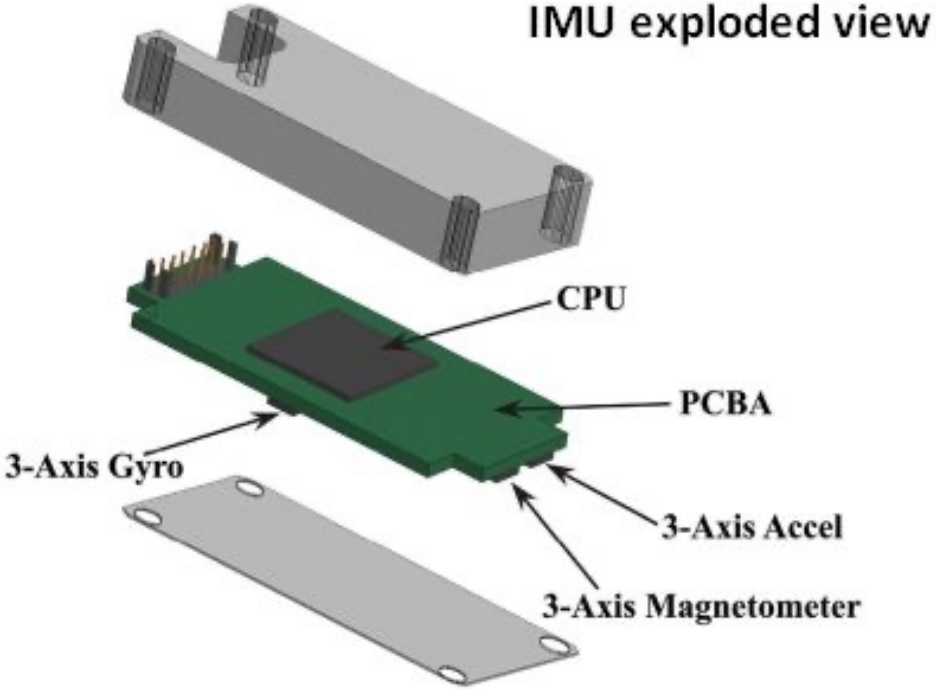


Pitch

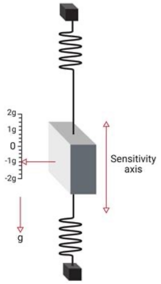
Inertial Measurement Units [IMUs]



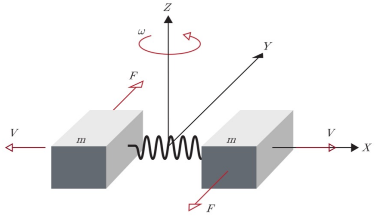
- Sensor to define **movement of vehicle**



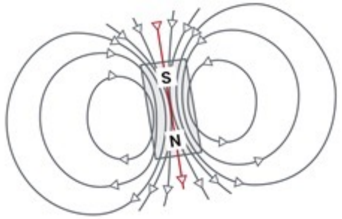
- IMU includes



accelerometer



gyroscope



magnetometer

Localization | Errors

- Each sensor builds up error over time
  - Drift in measurement from average values
  - Constant bias
  - Noise
  - Calibration errors
  - Scale factor
  - Vibration rectification errors

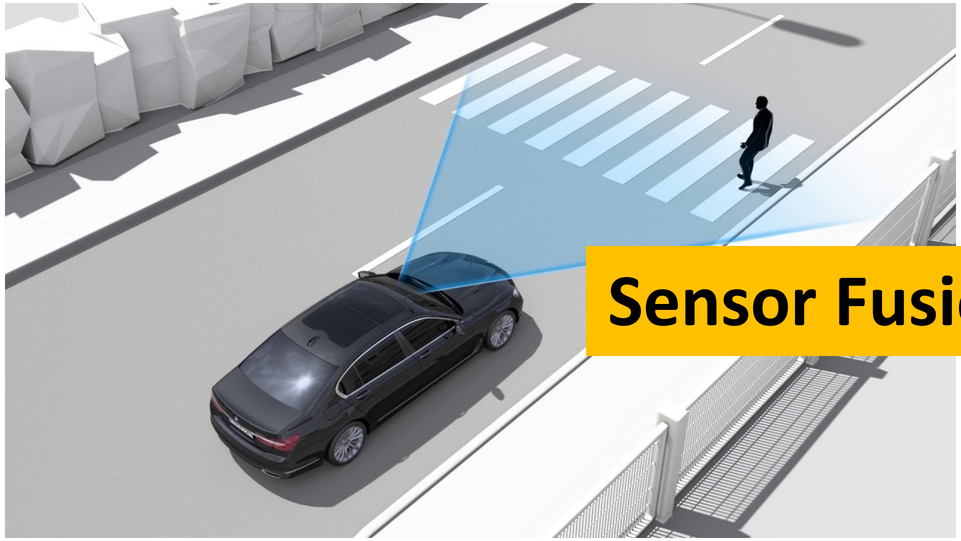
“fuse” data from multiple sensors | Sensor Fusion

Sensor Fusion

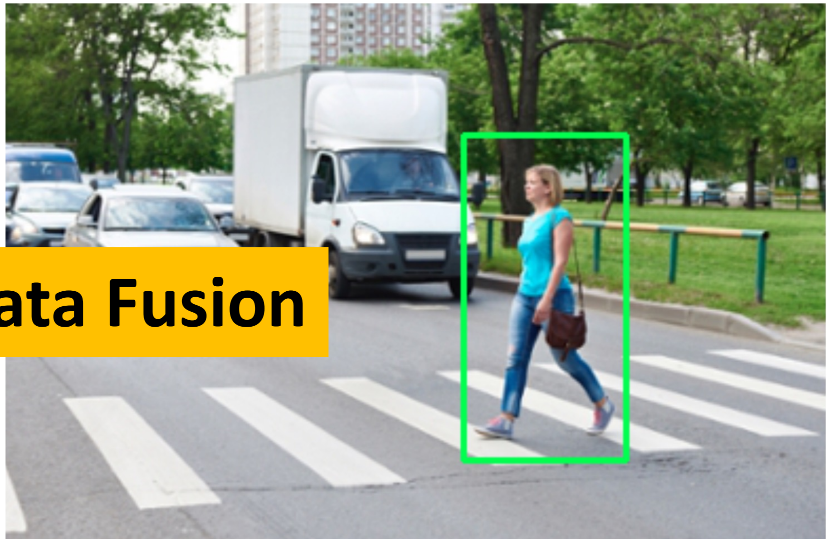
- Fusing data from multiple sensors
- Better **reliability**, **redundancy** and **safety**

Sensor Fusion

- Consider a LiDAR and a camera → looking at a pedestrian



Sensor Fusion → Data Fusion



Situation	Result
Only one detects the pedestrian	Use the other to increase chances
Both detect the pedestrian	Better **accuracy+confidence**

Sensor Fusion | Classification

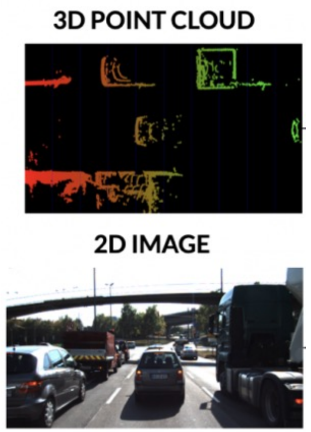
Abstraction Level	“when”
Centralization Level	“where”
Competition Level	“what”

Sensor Fusion | Abstraction Level

- *"when should we do the fusion?"*

- **Low-level fusion**
  - Fusing the raw data multiple sensors
  - E.g. point clouds from LiDARs and pixels from cameras

- Object detection
- Projecting 3D point clouds onto image
- Associating with the pixels



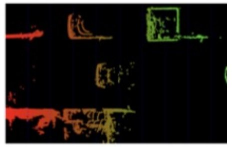
Pros	Cons
Future proof	Huge processing requirements

Abstraction Level | Mid Level Fusion

- **Fusing objects detected independently**
  - Each sensor does its own detection
  - E.g. camera and radar detect objects and these are fused
  - **Kalman Filter**

- 3D bounding box (LiDAR)+2D bounding box (camera)
- Projecting 3D result into 2D
- **Data fusion in 2D**

3D POINT CLOUD



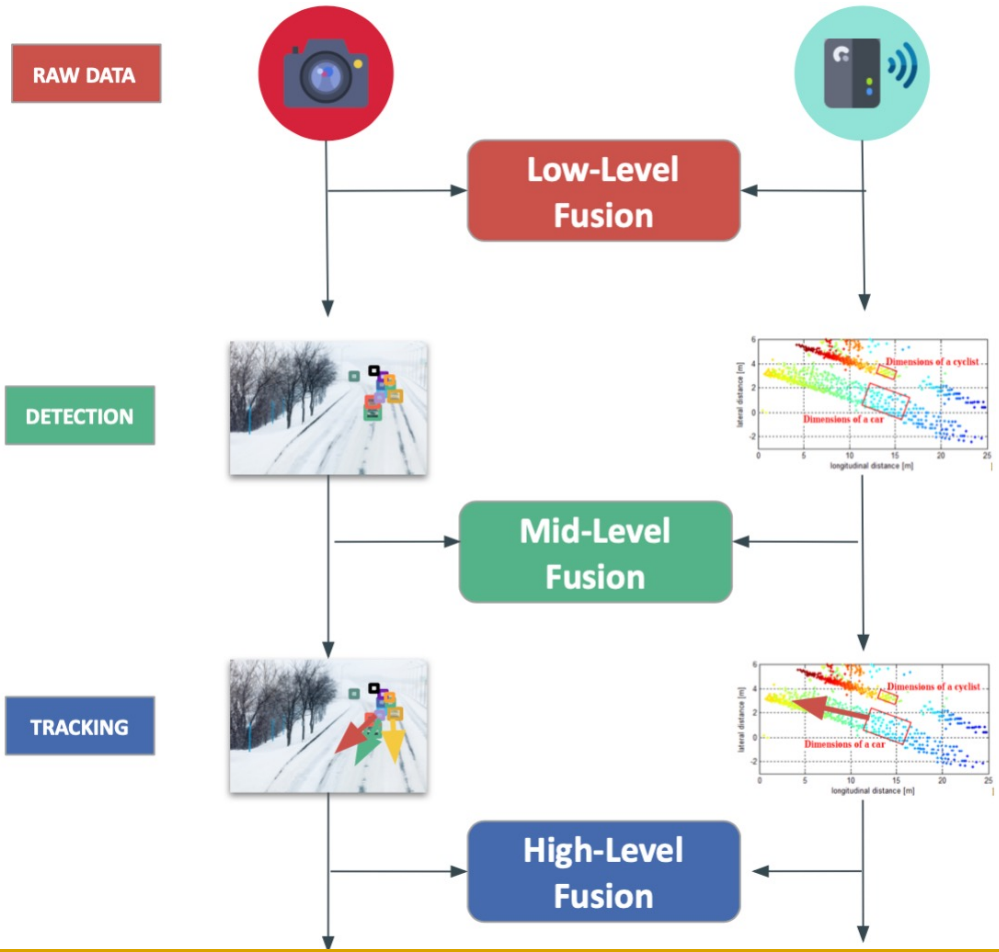
2D IMAGE

Pros	Cons
Simplicity	Potential to lose information

Abstraction Level | High Level Fusion

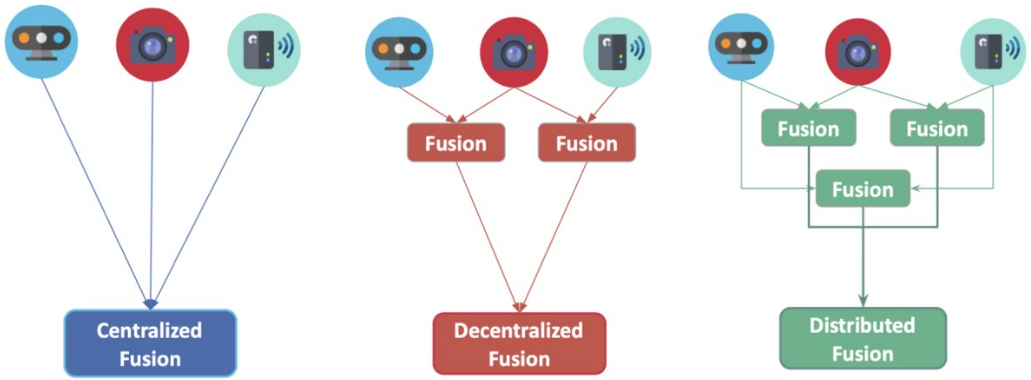
- **Fusing the tracks**
  - Fuse objects and their trajectories
  - Relying not only on detections
  - **Also on predictions+tracking**

Pros	Cons
Further Simplicity	Too much information loss



Sensor Fusion | Centralization Level

- “***where is the fusion happening?***”
  - Main computer
  - Each sensor does it independently



- Three types:
  - **Centralized**: one central unit deals with it [**low-level**]
  - **Decentralized**: each sensor fuses data and forwards to next one
  - **Distributed**: each sensor processes data locally and sends to next unit [**late**]

Centralization Level | Satellite Architecture

- Plug many sensors [satellites]
- Fuse together on a single central unit [**active safety domain controller**]
- 360 degree fusion+detection on controller
- Sensors do not have to be extremely good

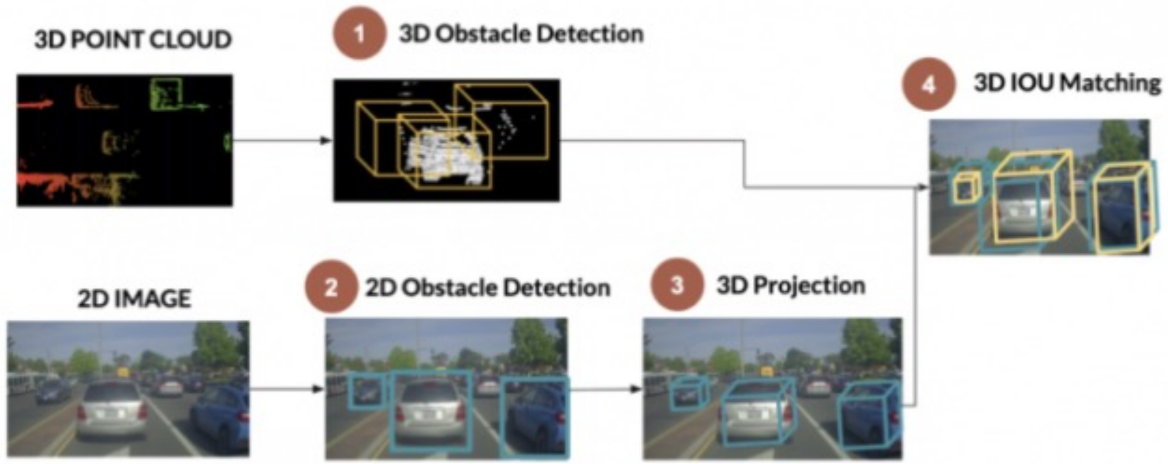
Sensor Fusion | Competition Level

- "***what should the fusion do?***"
- Three types
  - Competitive: sensors meant for same purpose [RADAR+LiDAR]
  - Complementary: different sensors looking at different scenes [multiple cameras]
  - Coordinated: sensors produce a new scene from same object [3d reconstruction]

Sensor Fusion | Competition Level

- *"what should the fusion do?"*

- Three types
  - **Competitive**: sensors meant for same purpose
  - E.g. Camera+LiDAR

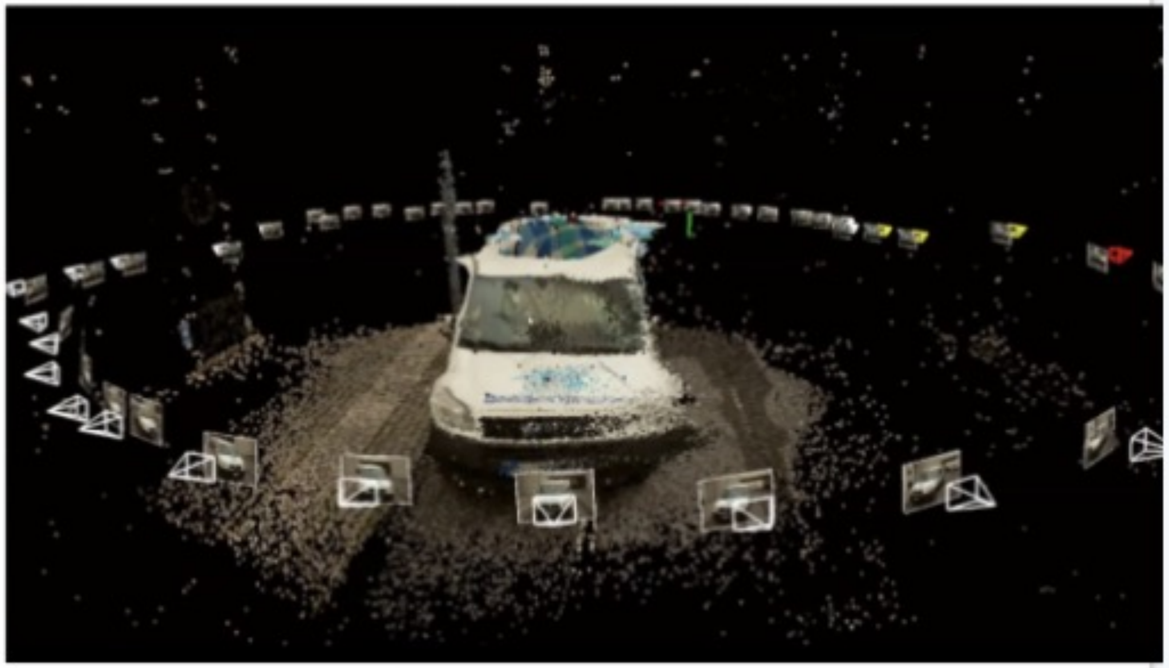


Sensor Fusion | Competition Level [contd.]

- **Complementary**
  - different sensors looking at different scenes
  - E.g. multiple cameras for creating panorama

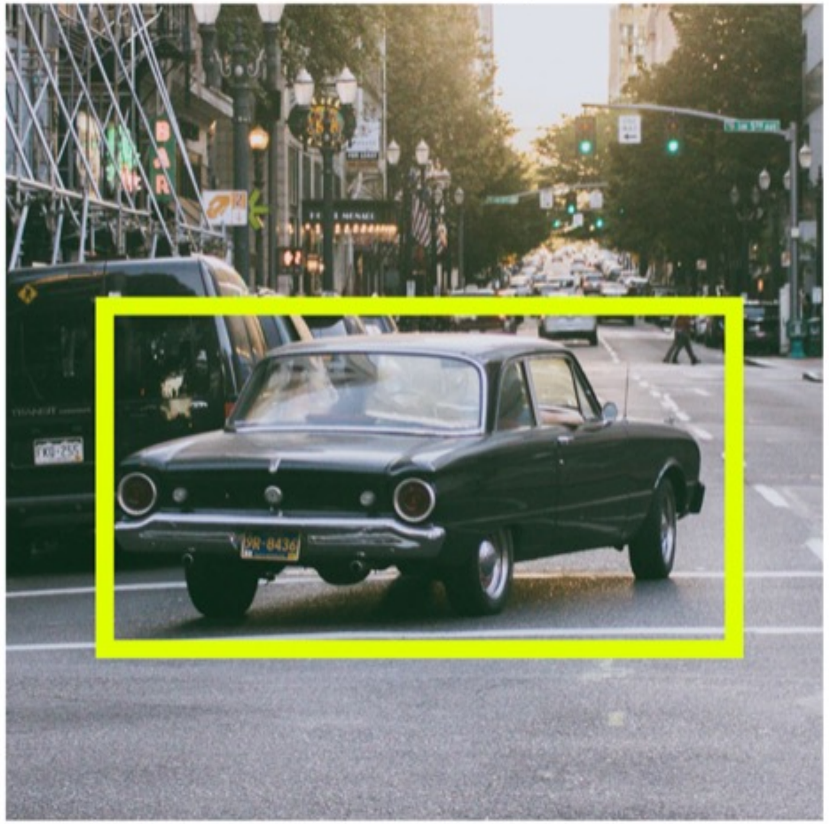
Sensor Fusion | Competition Level [contd.]

- **Coordinated**
  - sensors produce a new scene from same object
  - E.g. 3d reconstruction



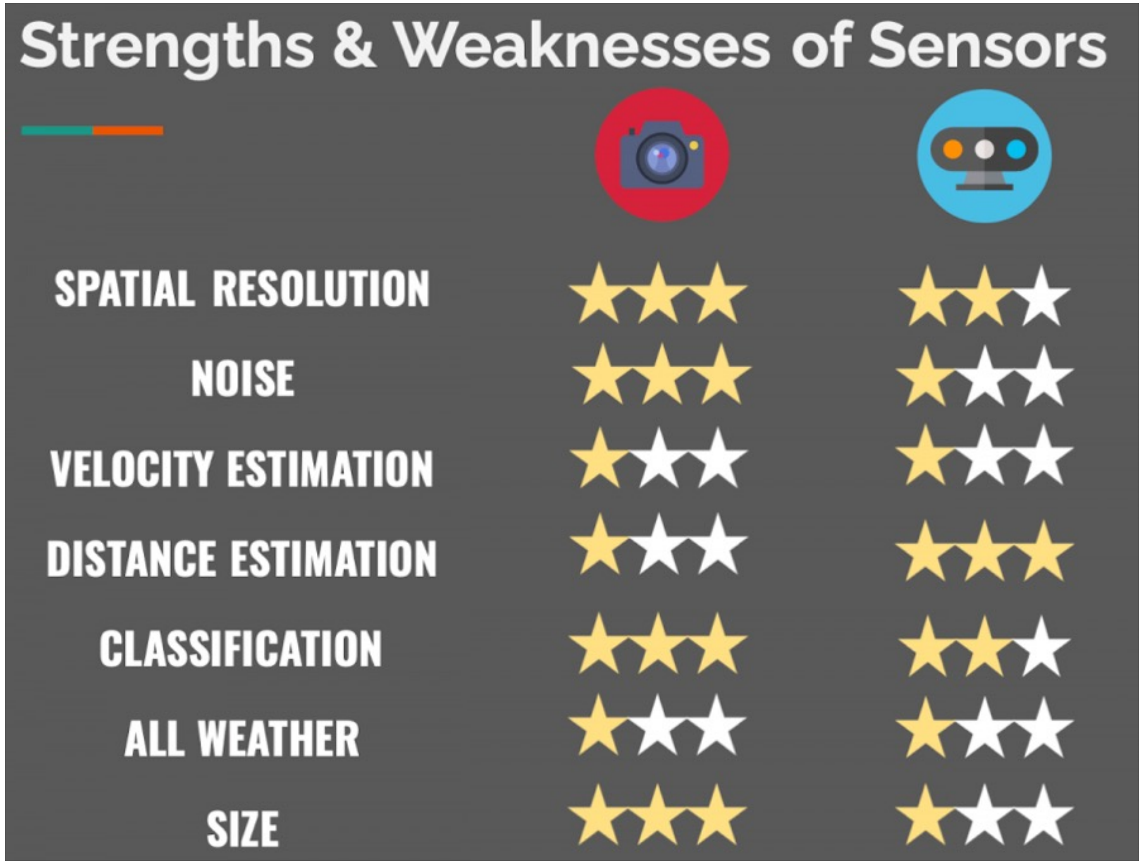
Sensor Fusion Example

Camera and LiDAR



Sensor Fusion | Camera+LiDAR

- Camera → excellent for **object classification** and **understand scenes**
- LiDAR → good for **estimating distances**

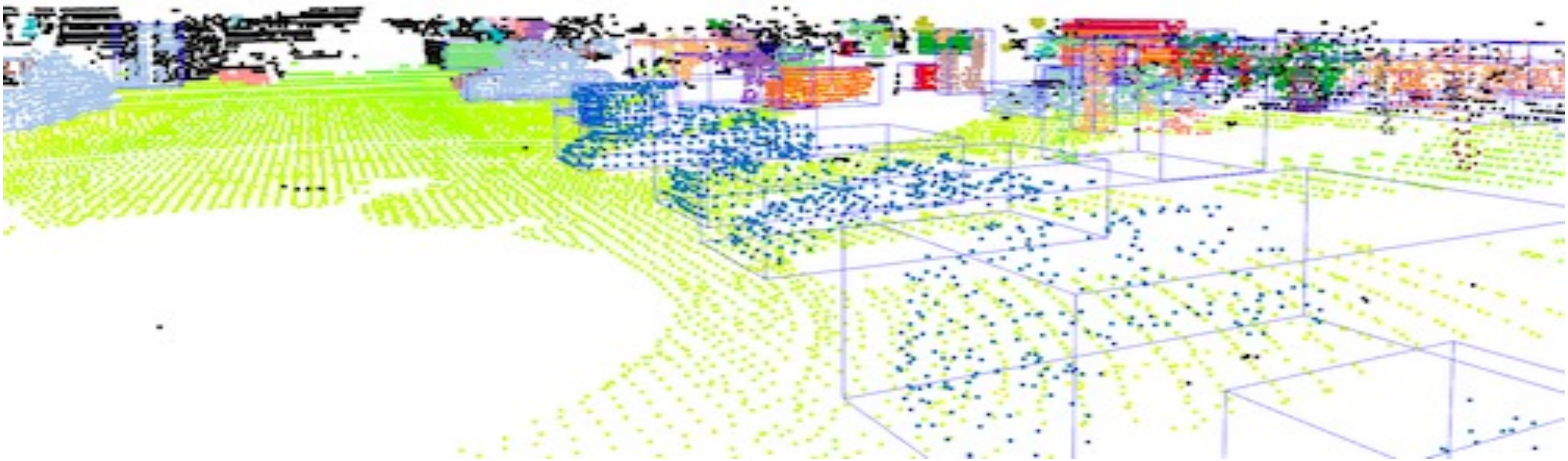


Sensor Fusion Example | Camera

- Outputs bounding boxes
- 2D

Sensor Fusion Example | LiDAR

- Outputs point clouds
- 3D



Sensor Fusion Example | Classes

“what”

competition and redundancy

“where”

doesn’t matter

(for now; lots of options)

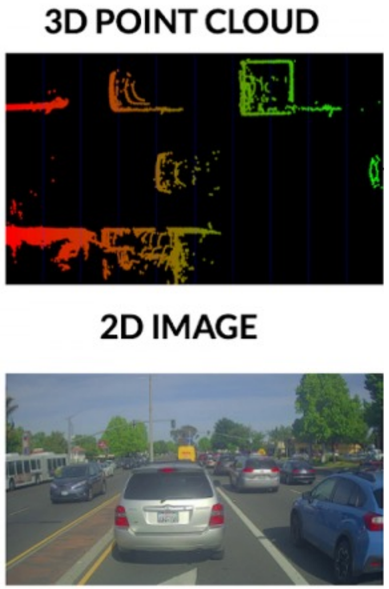
“when” | multiple options

early: fuse the raw data ➔ pixels and point clouds

late: fuse the results ➔ bounding boxes

Sensor Fusion Example | Early Fusion

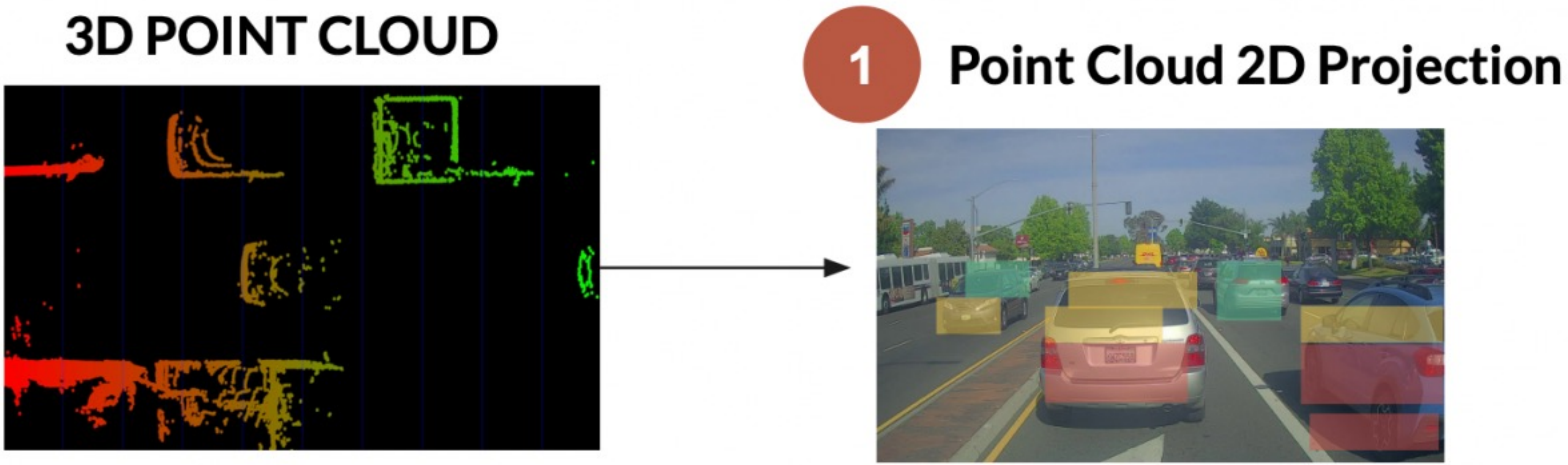
- Fuse raw data **as soon as sensors are plugged**
- Project 3D LiDAR point clouds onto 2D image
- Check whether point clouds belong to 2D bounding boxes from camera



Sensor Fusion Example | Point Cloud Projection in 2D

Translate 3D point cloud [LiDAR frame] ➔ 2D projection [camera frame]

1. **Convert** each 3D LiDAR point into **homogeneous coordinates**
2. Apply **projection equations** [translation/rotation] to convert from LiDAR to camera
3. **Transform back** into Euclidean coordinates



Sensor Fusion Example | Object Detection

- Detect the object using the camera
- **YOLO** again!

Sensor Fusion Example | ROI Matching

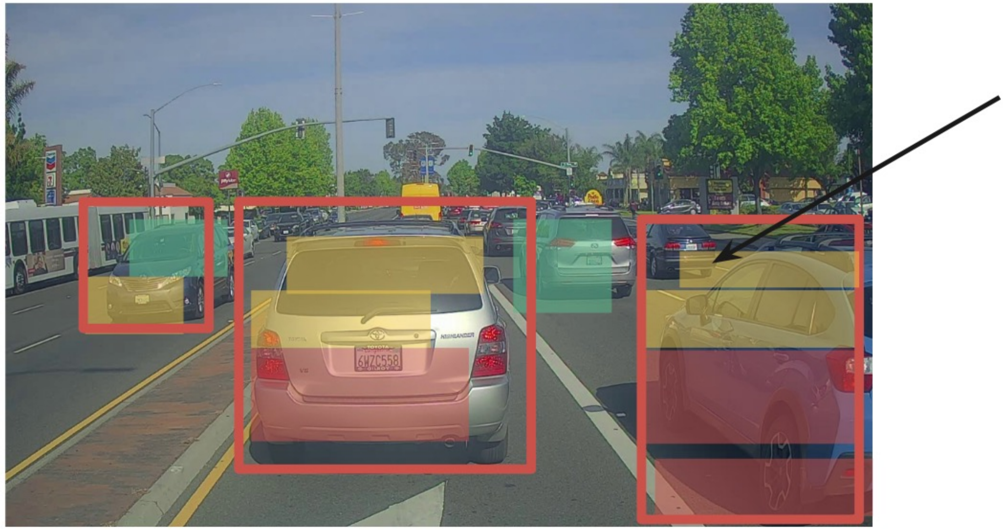
- “**region of interest**” mapping
- Fuse the data inside each bounding box
- Outputs?
  - For each bounding box → camera gives **classification**
  - For each LiDAR projected point → **accurate distance**

- Objects **are measured accurately and classified**

Sensor Fusion Example | Problems in ROI matching

- Which point to pick for distance?
  - Average/median/center point/closest?

- Point belong to another bounding box?

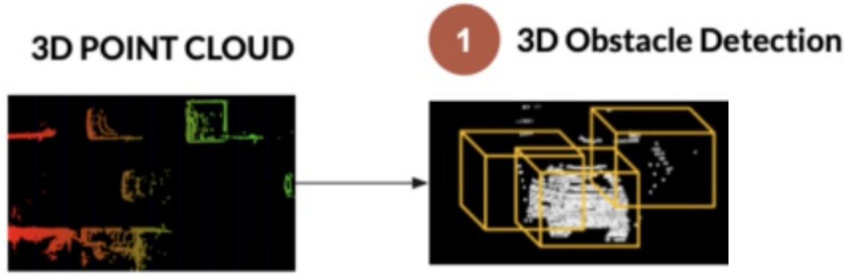


Sensor Fusion Example | Late Fusion

- Fusing result **after** independent detection
  - Get 3D bounding boxes on both ends, fuse results
  - Get 2D bounding boxes on both sides, fuse results

Sensor Fusion Example | Late Fusion in 3D

- Multiple Steps:
  1. 3D Obstacle Detection [LiDAR]

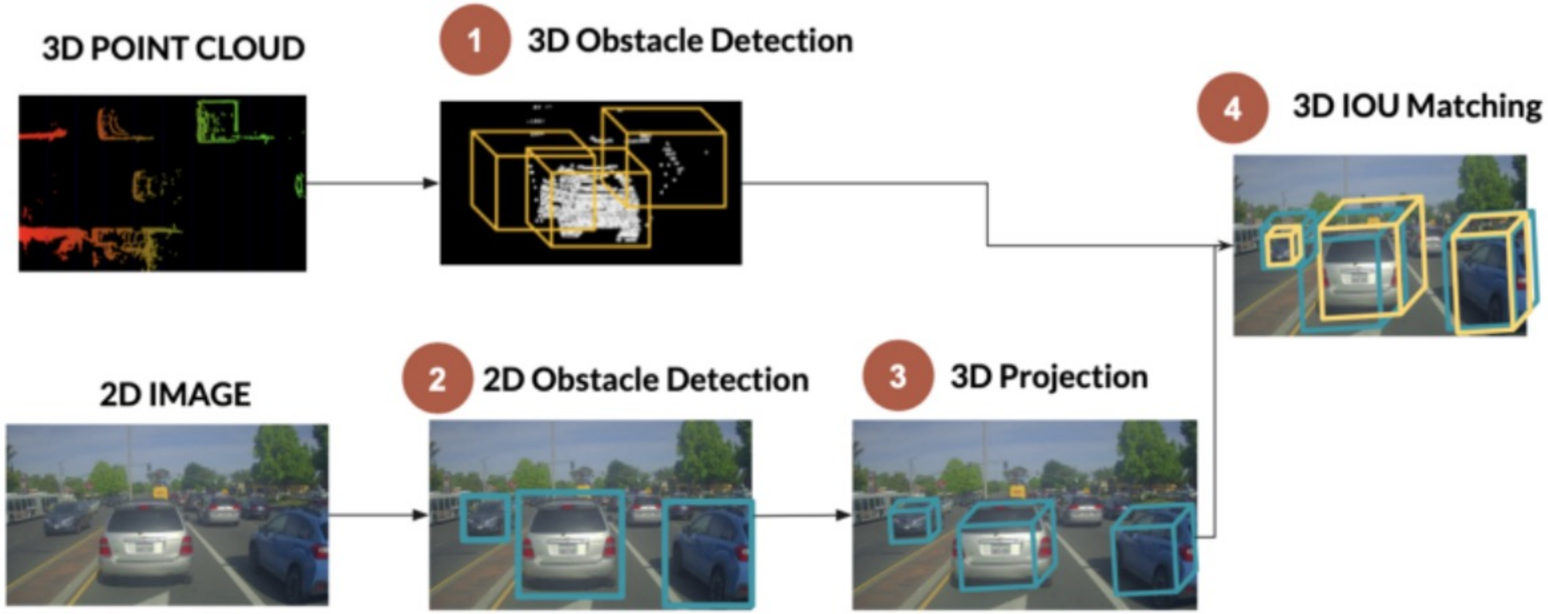


Sensor Fusion Example | Late Fusion in 3D

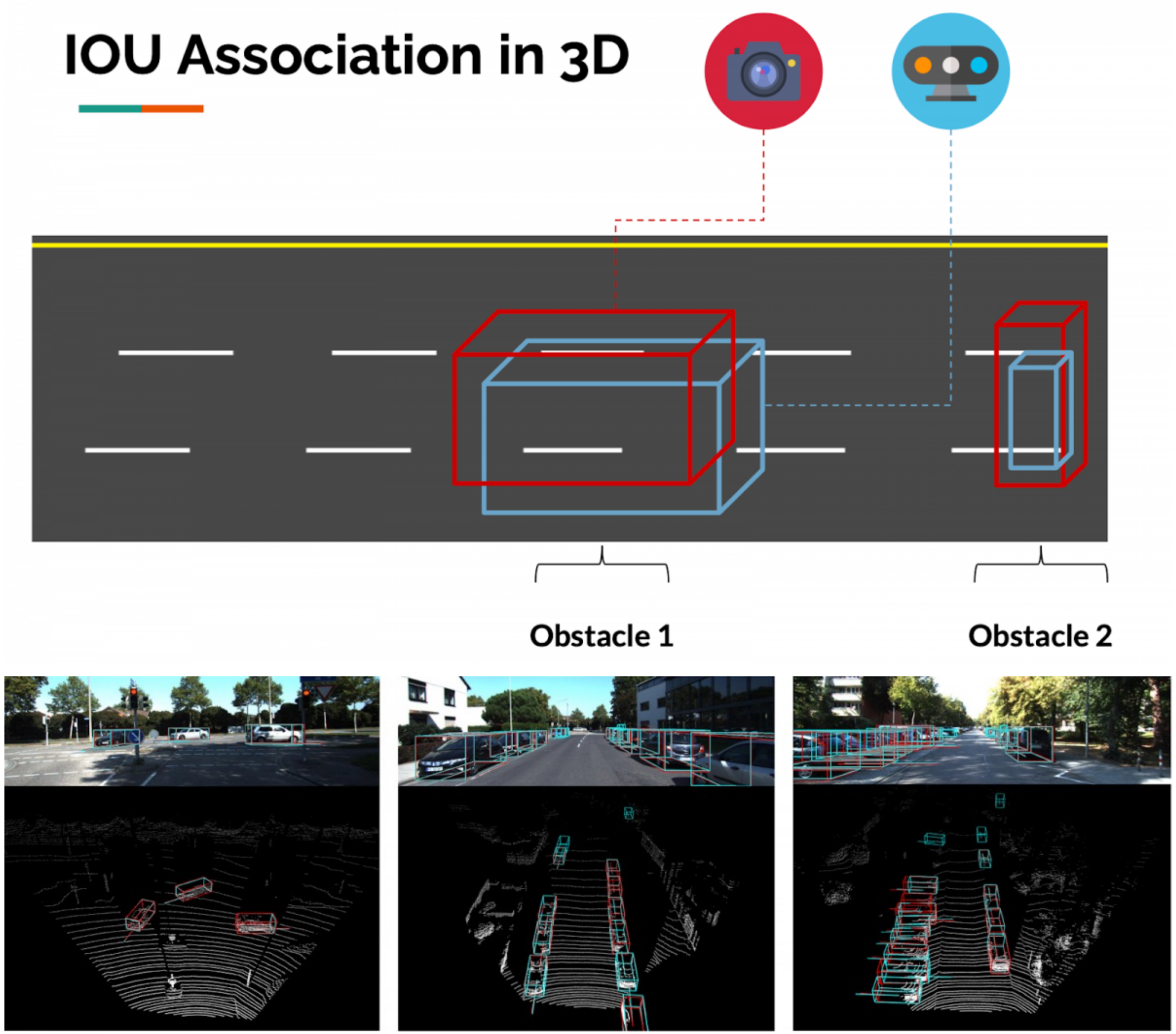
- Multiple Steps:
  1. 3D Obstacle Detection [LiDAR]
  2. 3D Obstacle Detection [Camera]
  3. IOU Matching in Space

Sensor Fusion Example | Late Fusion in 3D

- Multiple Steps:
    1. 3D Obstacle Detection [LiDAR]
    2. 3D Obstacle Detection [Camera]
    3. IOU Matching in Space



Sensor Fusion Example | IOU Matching



Sensor Fusion Example | IOU Matching in **Time**

Need to ensure the **frames also match in time**!

Associate objects in time, from frame to frame

Also predict next positions

Bounding boxes **overlap** between consecutive frames → same obstacle

Kalman Filter, Hungarian Algorithm, SORT

References

- IMUs

https://www.vectornav.com/resources/inertial-navigation-articles/what-is-an-inertial-measurement-unit-imu

- Sensor fusion classes

https://www.thinkautonomous.ai/blog/?p=9-types-of-sensor-fusion-algorithms

- Sensor fusion example (camera+LiDAR)

https://www.thinkautonomous.ai/blog/?p=lidar-and-camera-sensor-fusion-in-self-driving-cars

- 3D Bounding Box Estimation – one technique

https://arxiv.org/pdf/1612.00496.pdf