Sensor Fusion for Autonomous Parking

Sensor Fusion Team

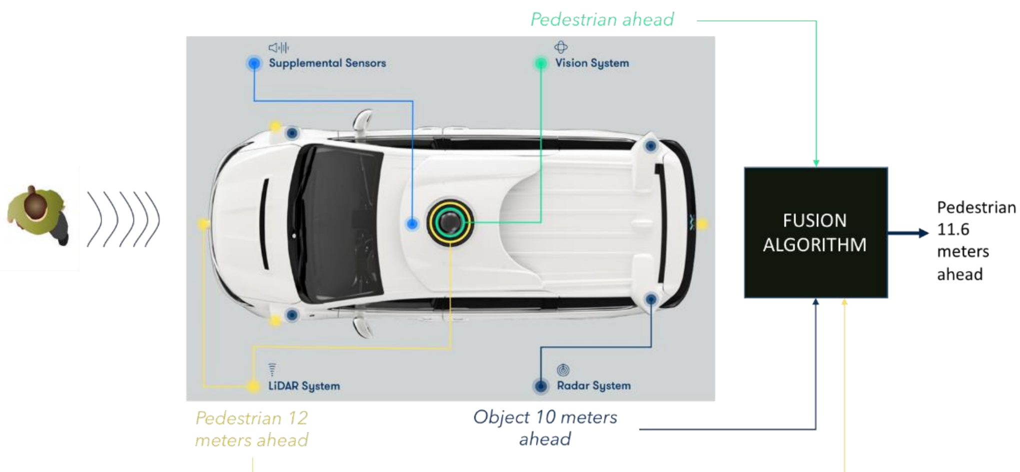
Hot Topics in Computer Vision | 25.03.2019

Motivation

Why Sensor Fusion?

- Combines sensory data
- Allows for more accurate estimations (inference)
- Alleviates weaknesses found in stand-alone sensors

Essentially, sensor fusion allows us to maximize the parameters and specifications of multiples sensors simultaneously.



Udacity [1]

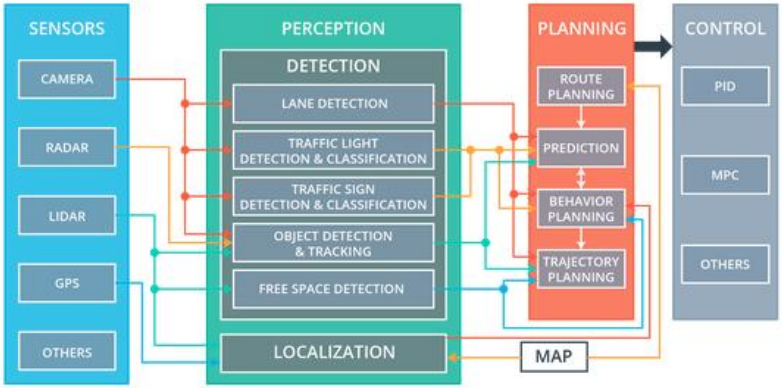
Design Specifications

Case for Hella Aglaia:

- Seek out parking spaces
- Confirm availability
- Not concerned with detecting moving objects
- Real-time functional and quality requirements

Design Considerations:

- Access to RADAR and LiDAR data
- The data is non-linear
- Cassandra synchronizes the sampling rate
- No External Map



Udacity [1]

Research Objectives

Implement Sensor Fusion:

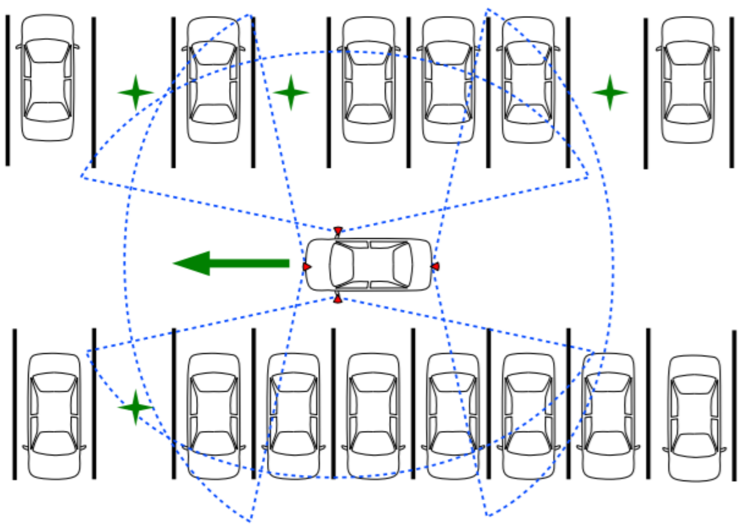
- Select an appropriate algorithm
- Apply it based on data and model

Object Tracking:

- Classify objects
- Assign objects IDs

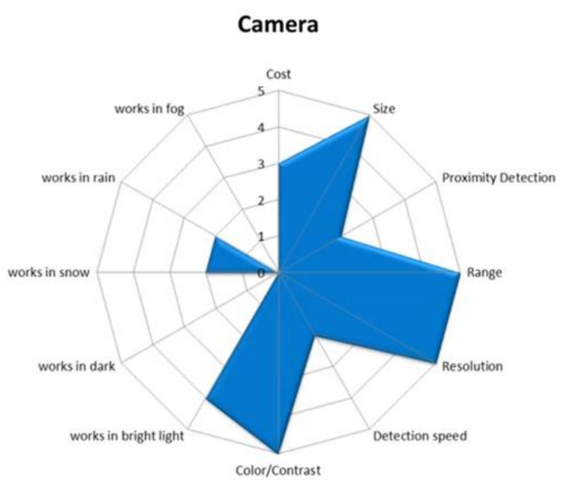
Find Unoccupied Parking Spaces:

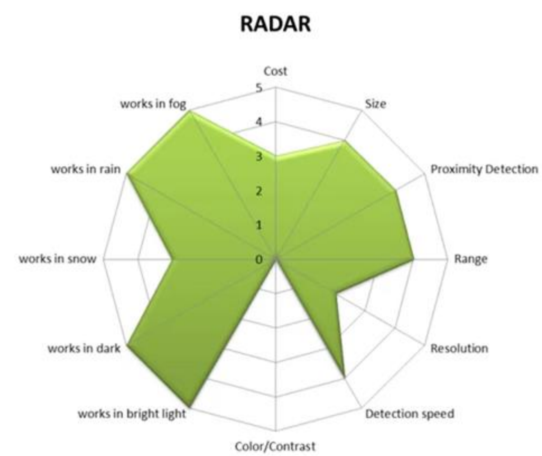
- At least, find where we cannot park (safety)

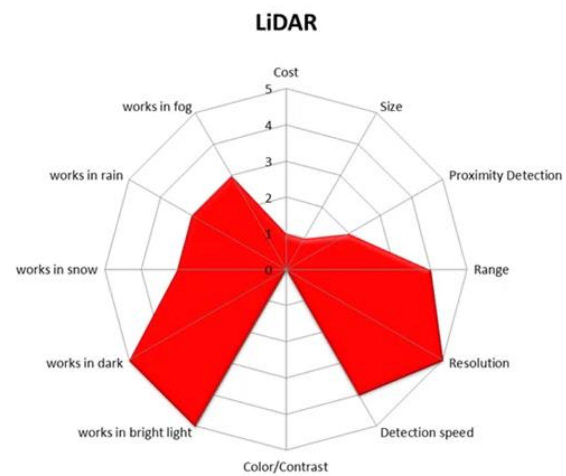


Houben et al. [5]

Sensor Characteristics

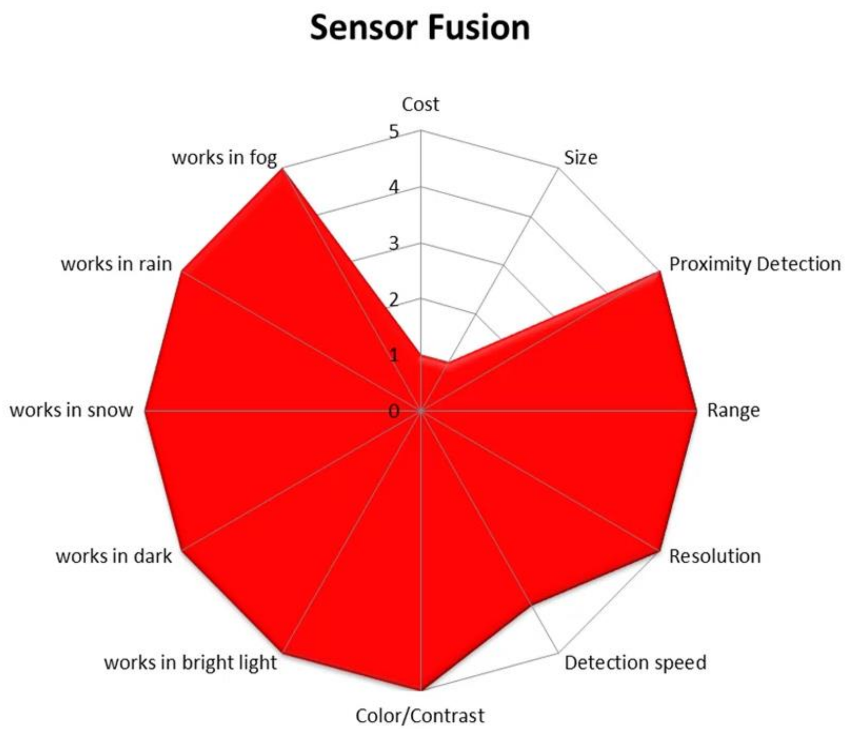






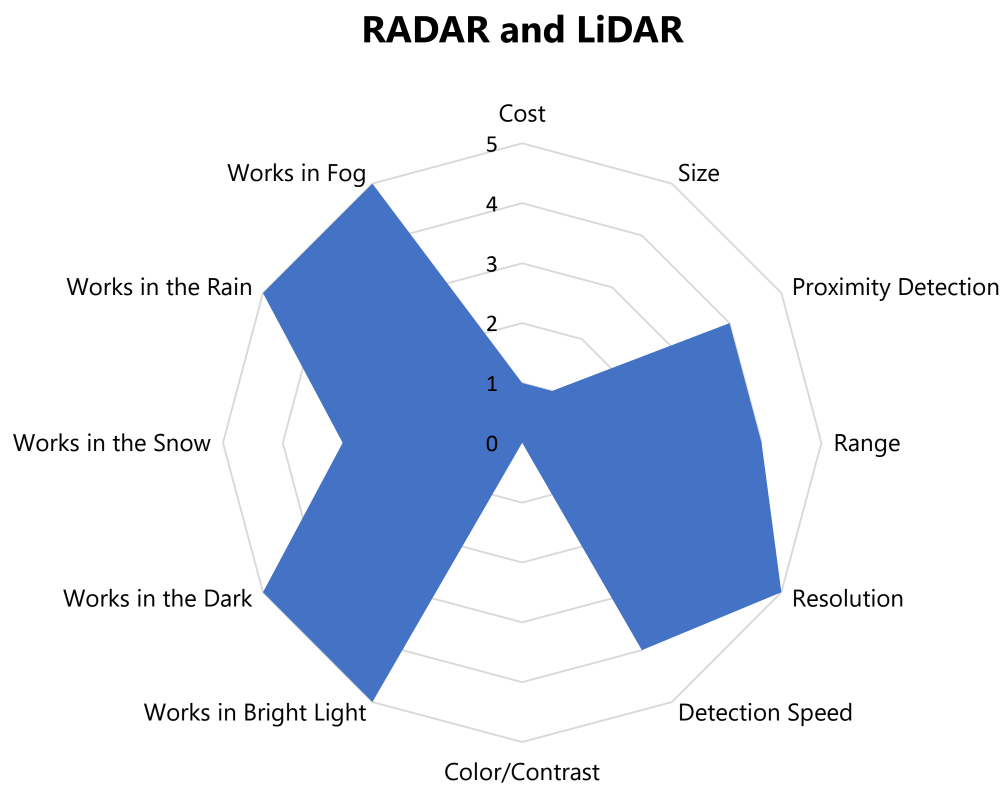
TI [3]

Fused Sensor Characteristics



TI [3]

Fused Sensor Characteristics



What do we get from the other teams?

Covariance

- Measurement Noise
  - 'Adjustable'

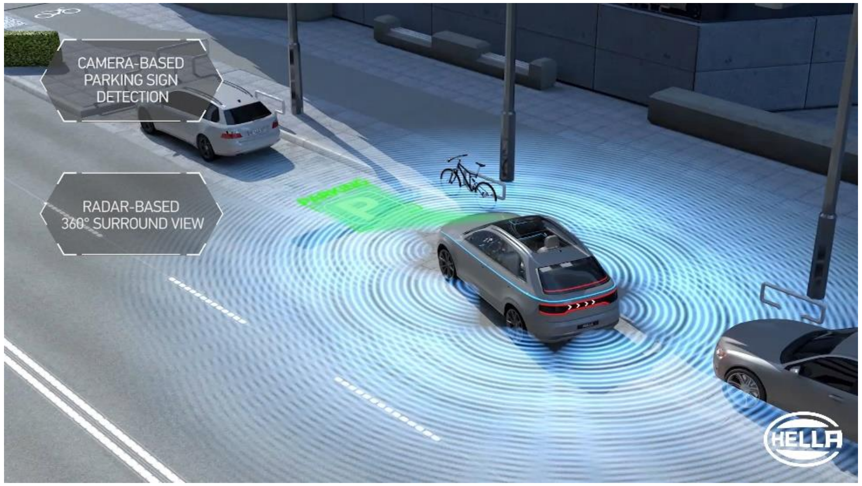
  → Weighted Reliability

Bounding Boxes

- Coordinates (x_min, x_max, y_min, y_max)

Time Stamps

- Synchronization
- Object Identification (assignment)

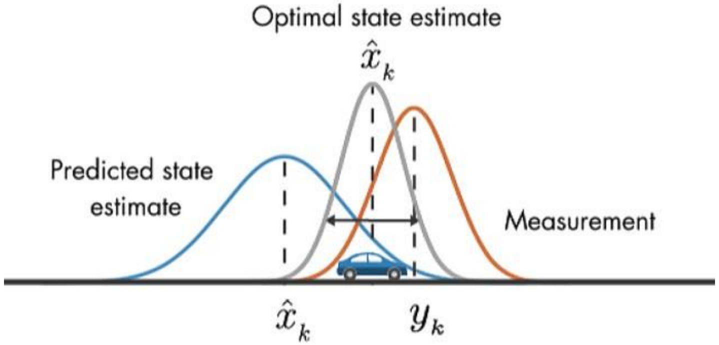


Hella [10]

Extended Kalman Filter

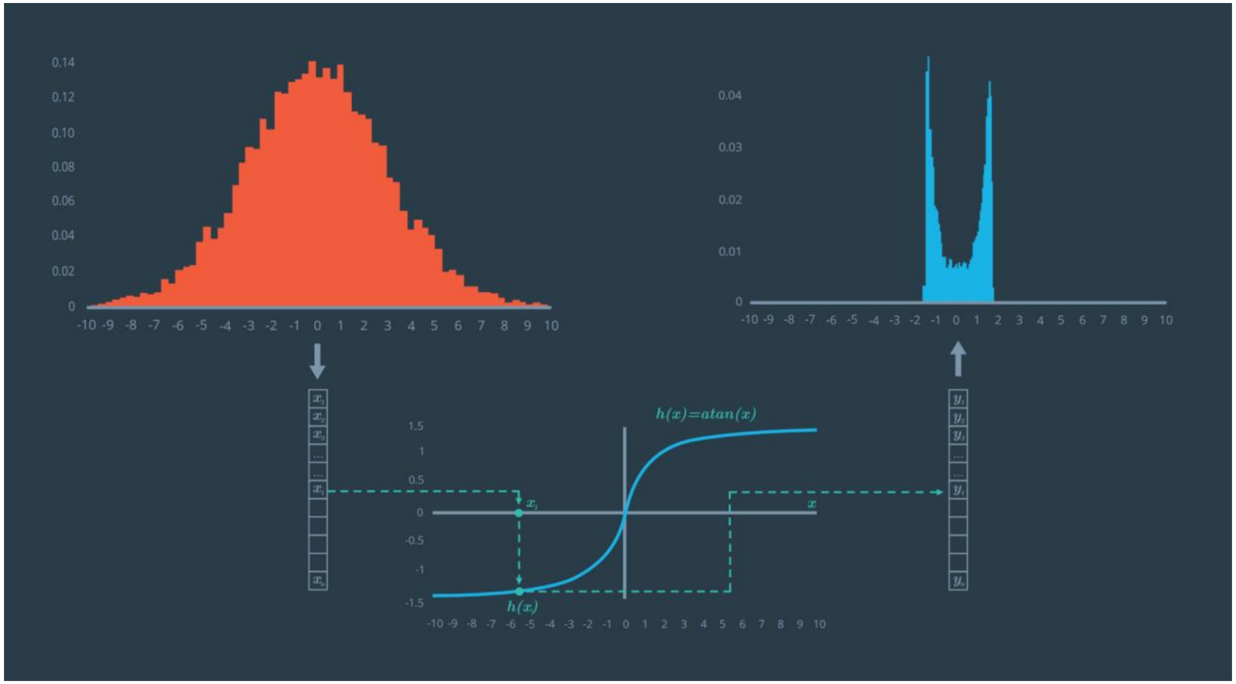
Why EKF?

- High measurement frequencies
  - Jacobian useful for "local" linearity
- Real-time desirability
- Multiple instances of EKF (a problem for UKF complexity)
  - We use individual (weighted) EKFs for each sensor
- Sparseness in matrices
  - Less computation for EKF



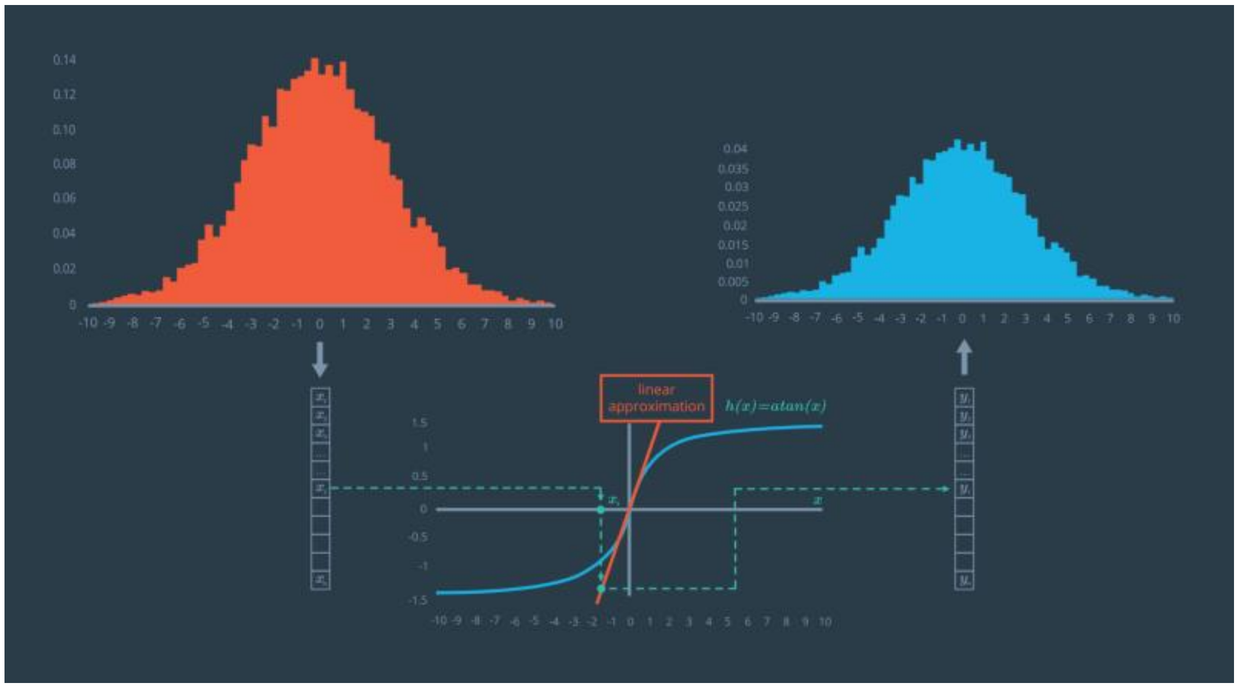
Udacity [1]

Why EKF?



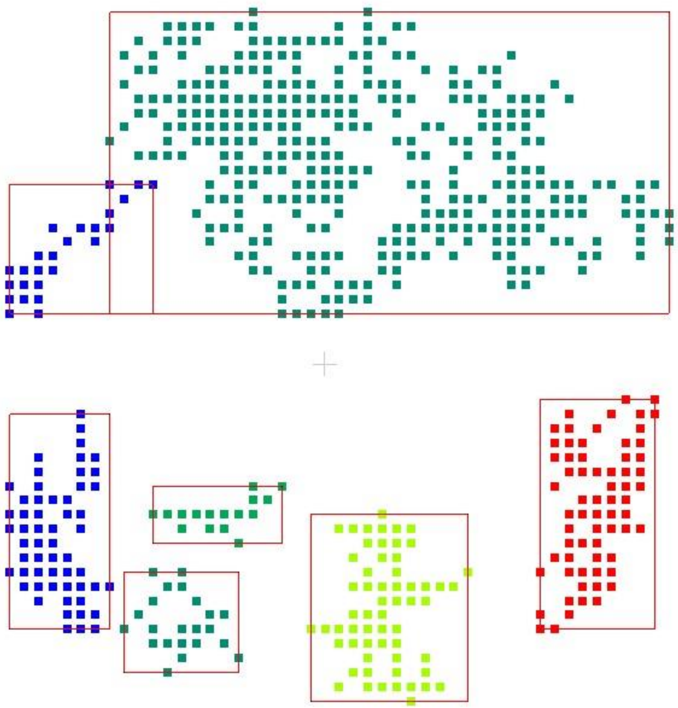
Udacity [1]

Why EKF?

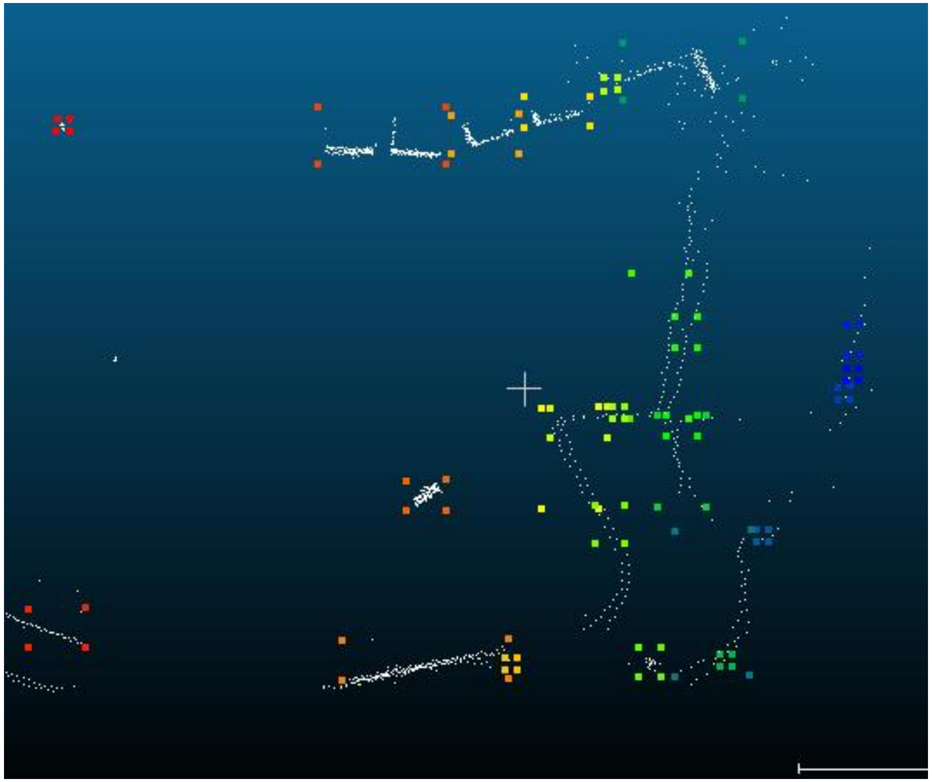


Udacity [1]

What is the problem?



RADAR

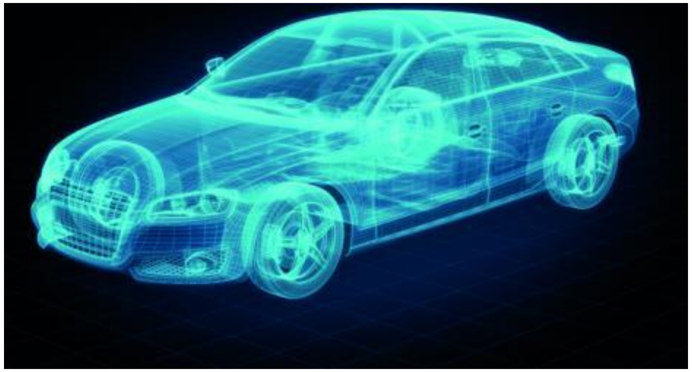


LiDAR

Tracking (Data Association)

Tracking:

- Objects change over time
  - How do you know when there is a new object?
  - Classification!
    - Multi-Object Tracking (ID)
    - ‘Freshness’ Rate
- Bounding boxes are not equally sized
  - How do we differentiate discrete objects?
- Noise is still an issue



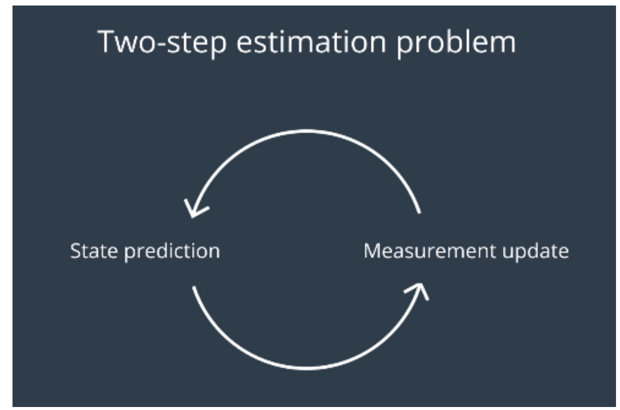
Hella Aglaia [2]

Nearest Neighbors (NN)

Motivation:

- EKF does not differentiate objects!
- Track objects based on locality
  - Track only meaningful objects
  - Global: multiple objects
- Deterministic or Probabilistic?
  - We used deterministic (difference of bounding boxes)
  - Threshold
    - Size of object
    - Evolution over time (Δt_{obj})

Result: NN acts like a filter



Udacity [1]

Technische Universität Berlin

Vehicle Model

$$x_k = [x, y]\ ;\ z_k = [x, y]$$

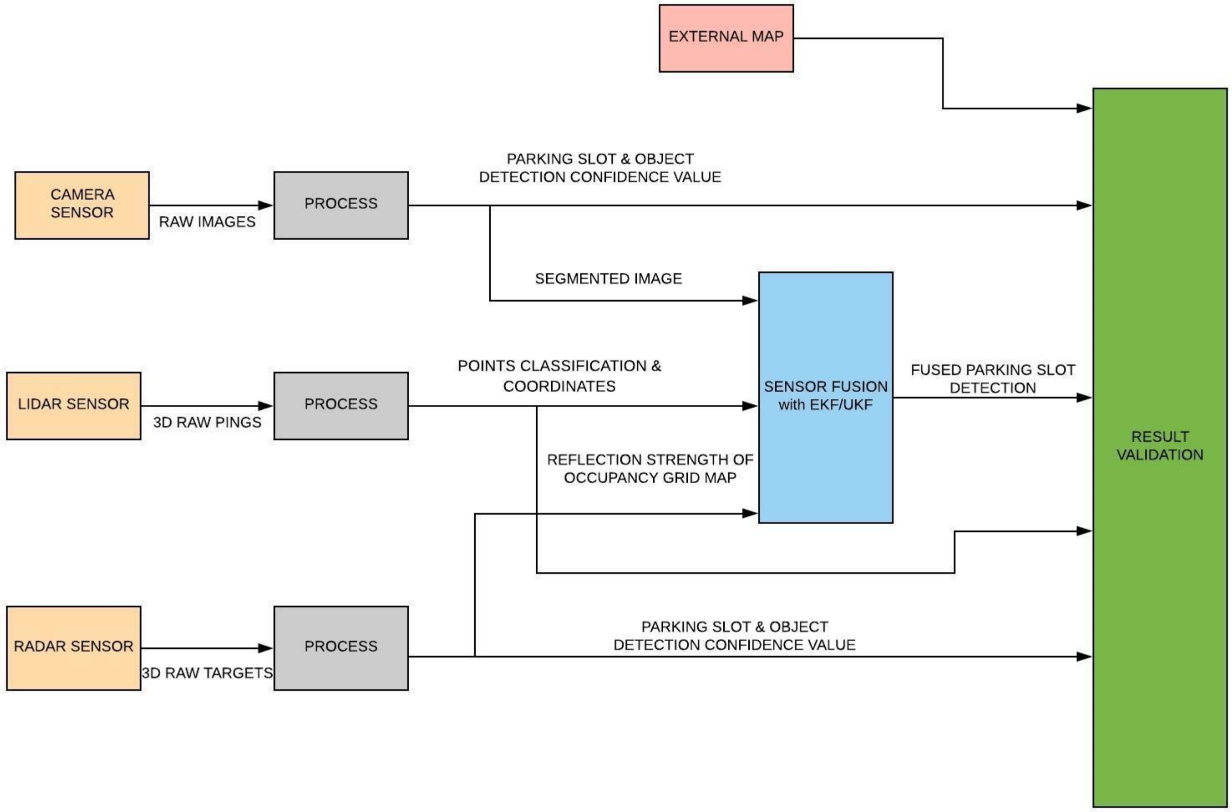
$$f(x) = x' = x \cos v_{yaw}t - y \sin v_{yaw}t - v_x t - \frac{1}{2} a_x t^2$$

$$u_k = [a_x\ a_y\ v_x\ v_y\ v_{yaw}]$$

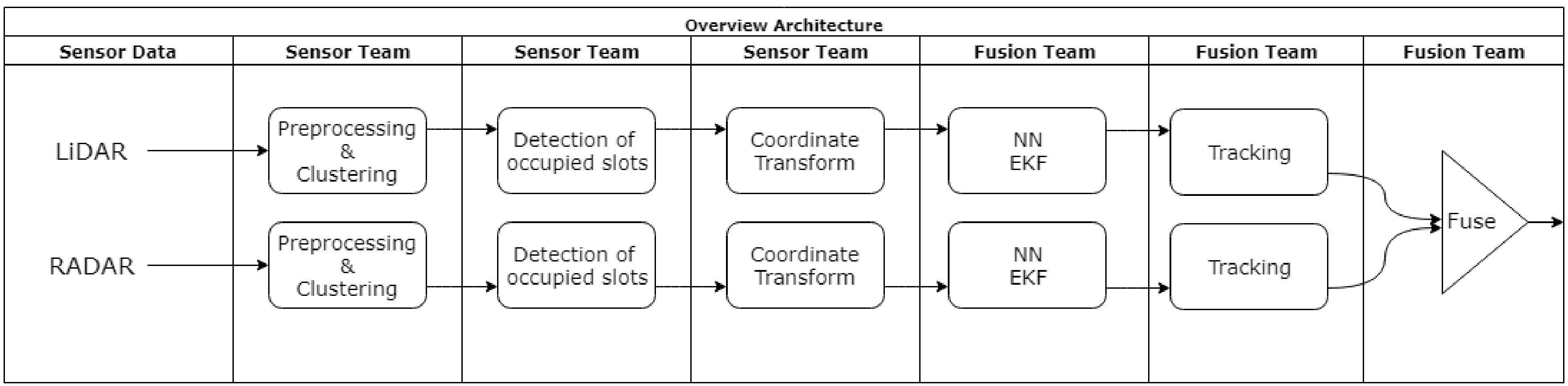
$$f(y) = y' = x \sin v_{yaw}t - y \cos v_{yaw}t - v_y t - \frac{1}{2} a_y t^2$$

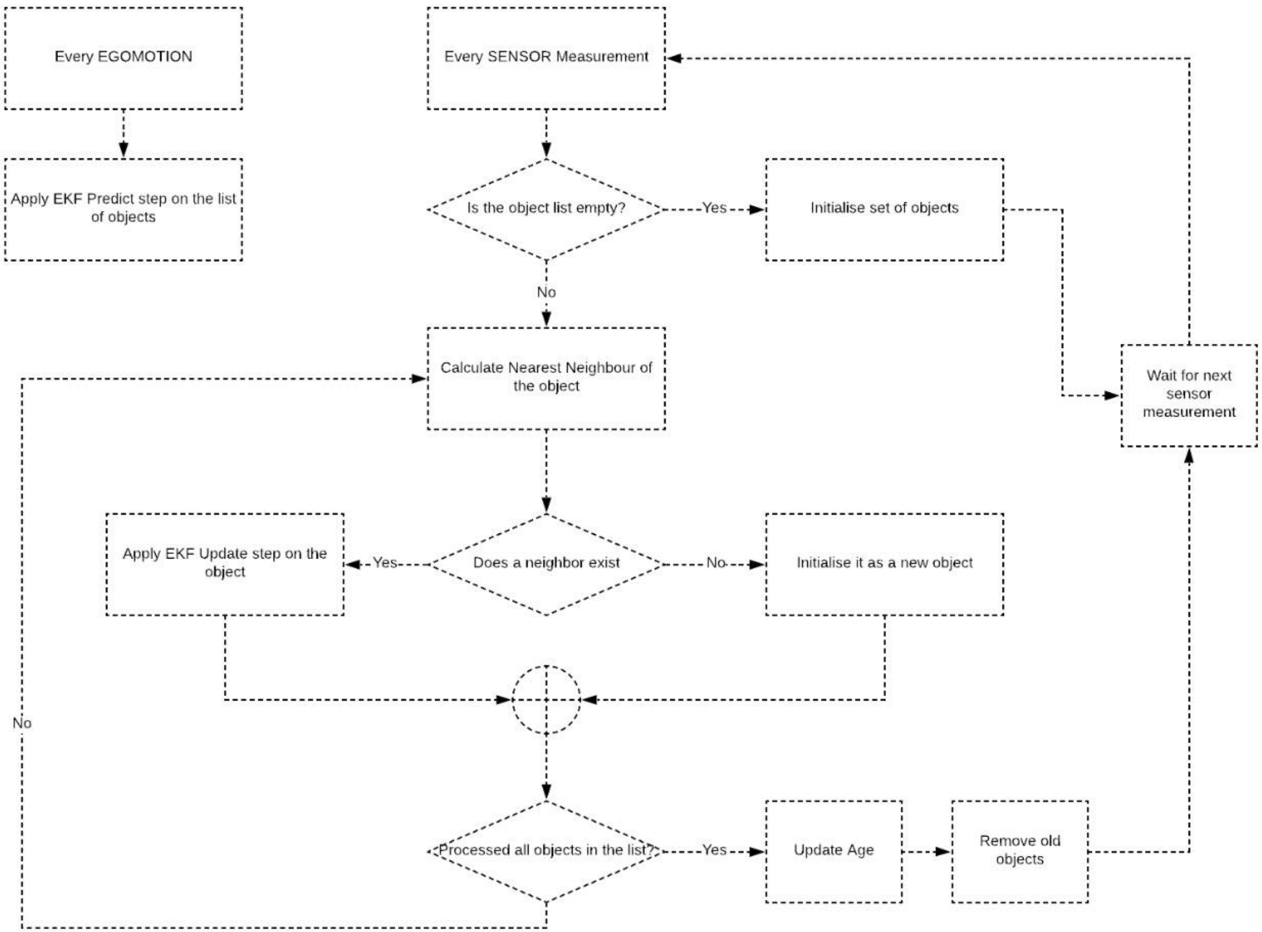
$$h(x, y) : x' = x,\ y' = y$$

Original Architecture:



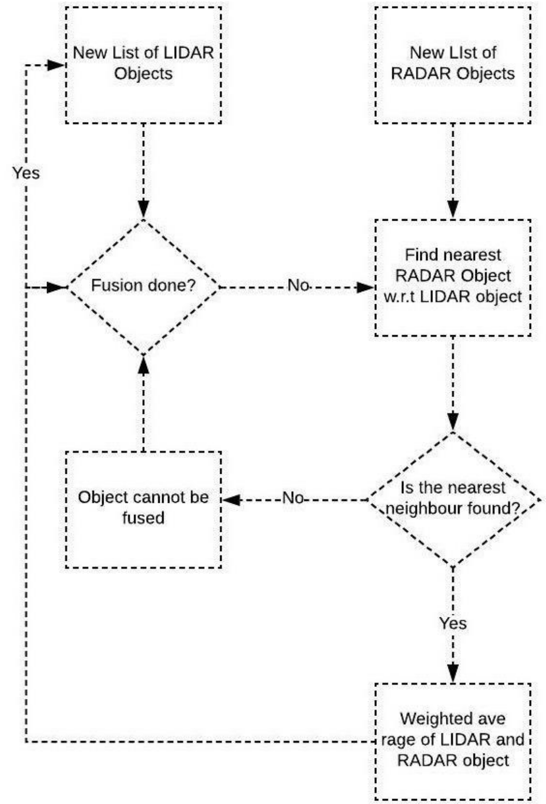
New Architecture:



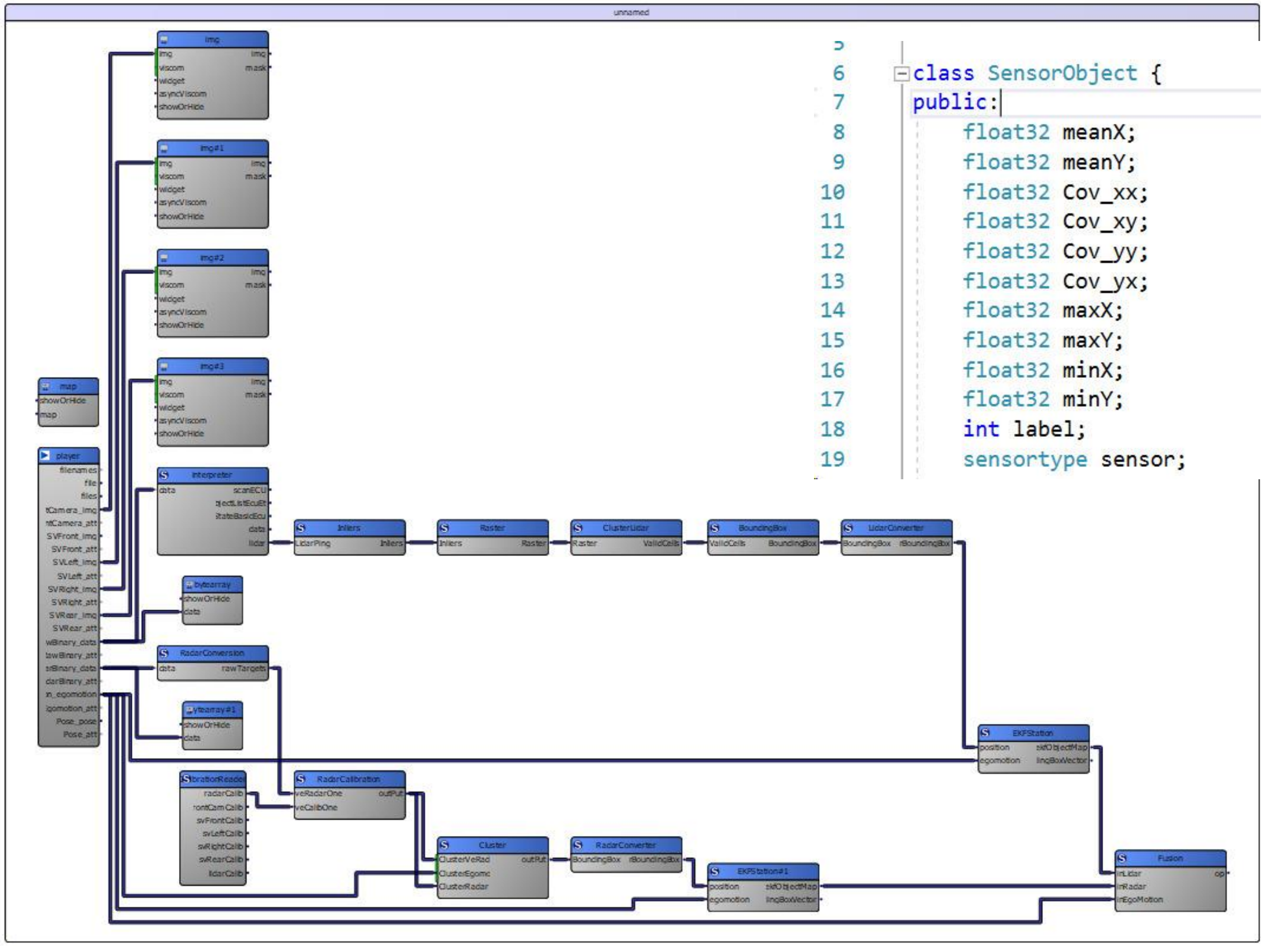


Every EGOMOTION
Apply EKF Predict step on the list of objects
Every SENSOR Measurement
Is the object list empty?
Yes
Initialise set of objects
No
Calculate Nearest Neighbour of the object
Wait for next sensor measurement
Apply EKF Update step on the object
Yes
Does a neighbor exist
No
Initialise it as a new object
No
Processed all objects in the list?
Yes
Update Age
Remove old objects

Fusion Flowchart

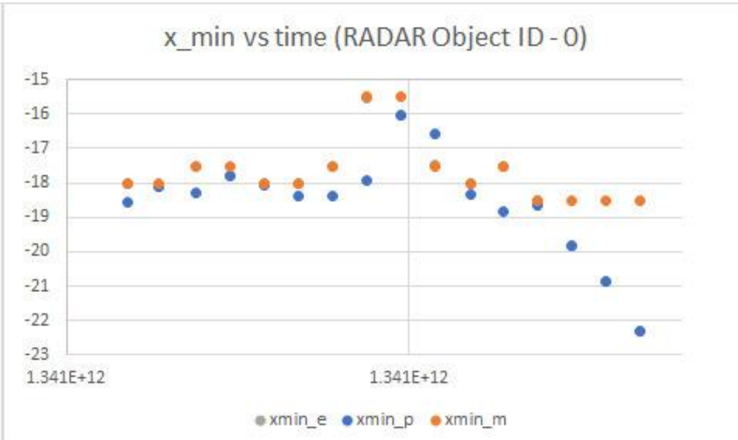


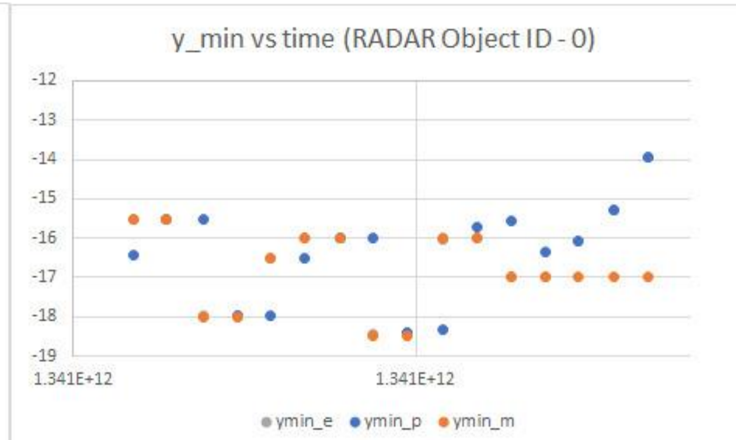
Stations:

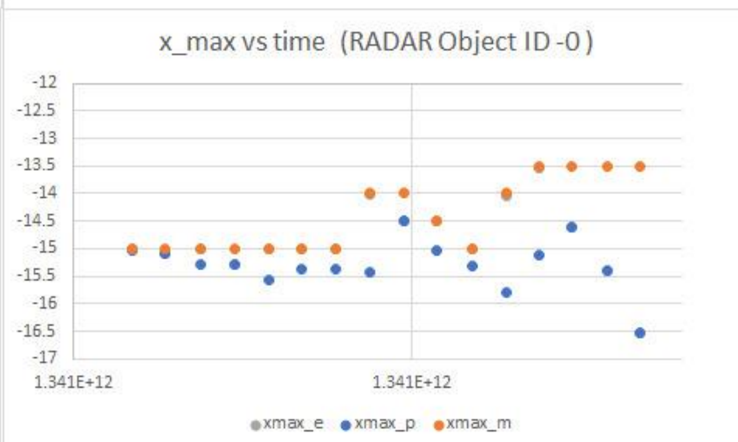


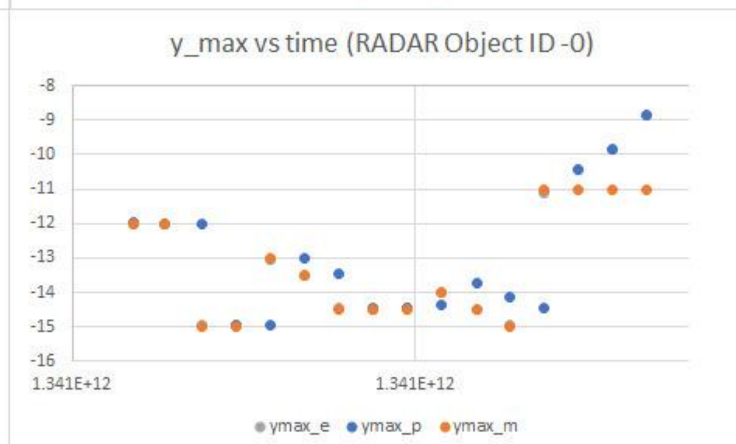
```cpp
class SensorObject {
public:
    float32 meanX;
    float32 meanY;
    float32 Cov_xx;
    float32 Cov_xy;
    float32 Cov_yy;
    float32 Cov_yx;
    float32 maxX;
    float32 maxY;
    float32 minX;
    float32 minY;
    int label;
    sensortype sensor;
```

Technische Universität Berlin



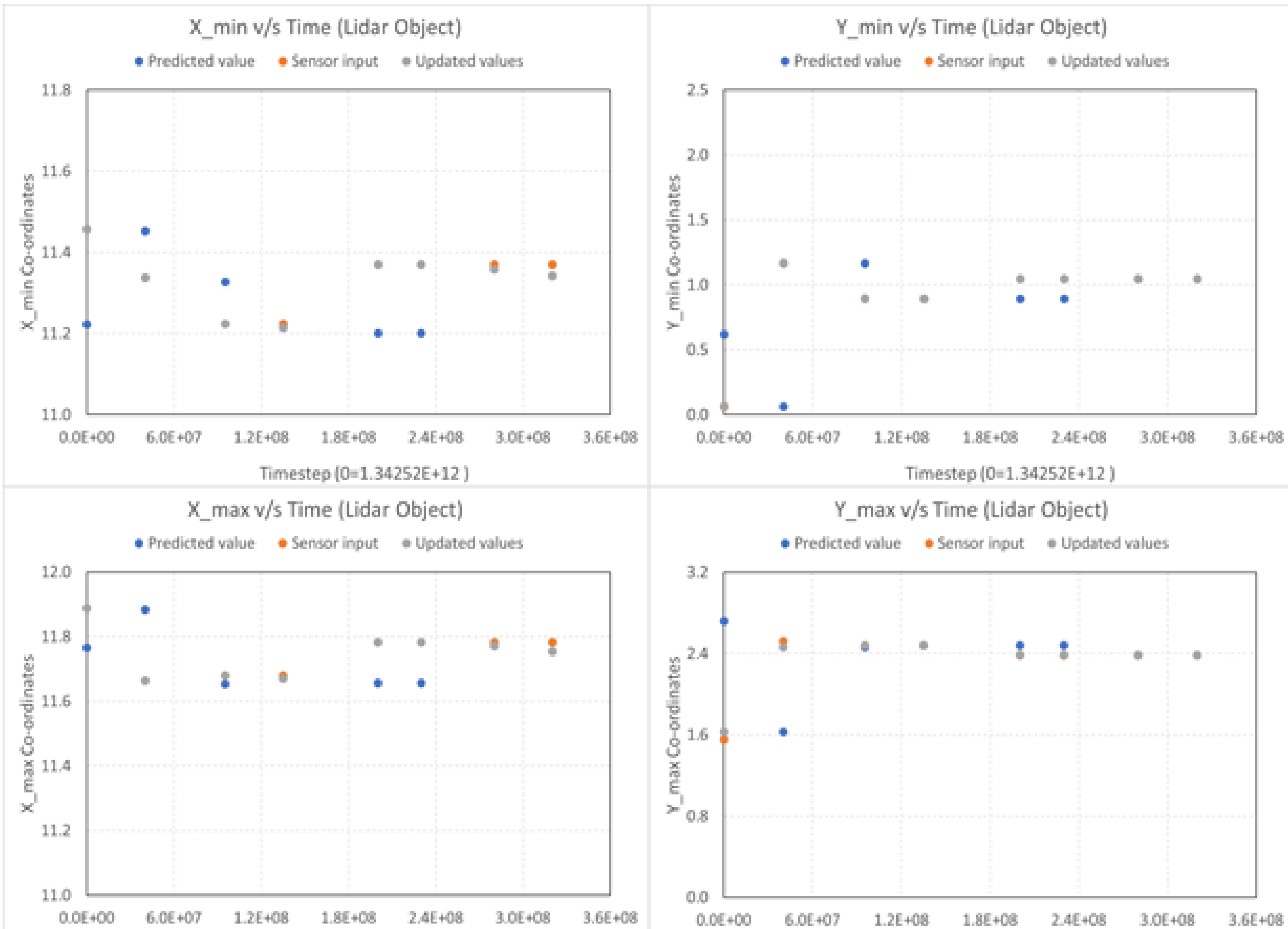






Tracking Results of RADAR

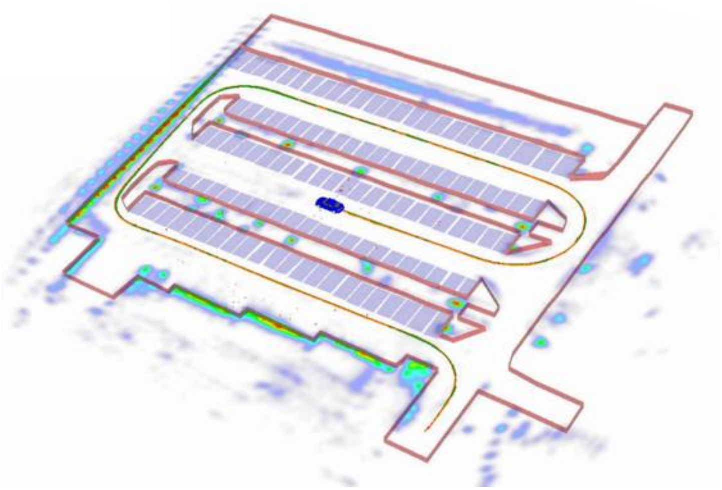
Hella Aglaia



Tracking Results of LiDAR

Results:

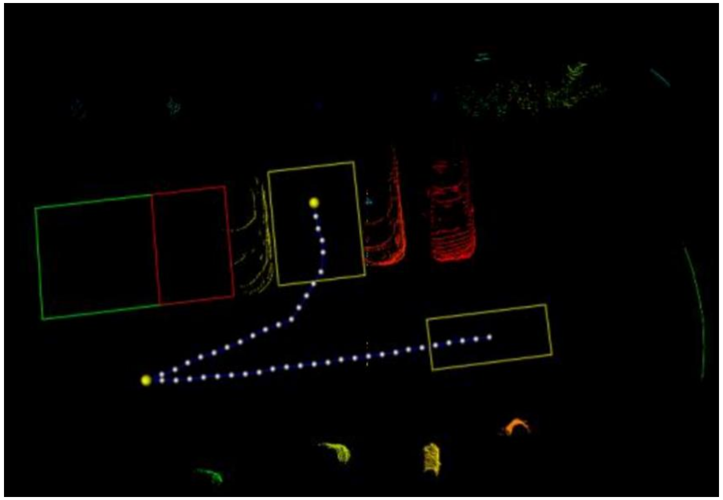
→ LiDAR and RADAR are properly tracked
→ Nearest Neighbors correctly implemented
→ Objects have been given IDs
→ Objects have been tracked
→ EFK station configured to automatically switches between sensor input



Hella Aglaia [2]

Conclusions:

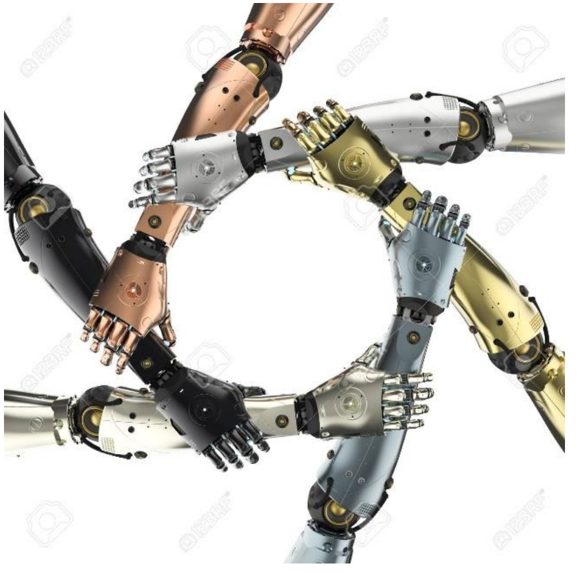
- → Sensor Fusion is a non-trivial topic
- → Documentation is often contradictory
- → (Free spots) != (Full spots)^{c}
- → The process is highly iterative
- → Communication is crucial
- → Flexibility is required for dead-ends



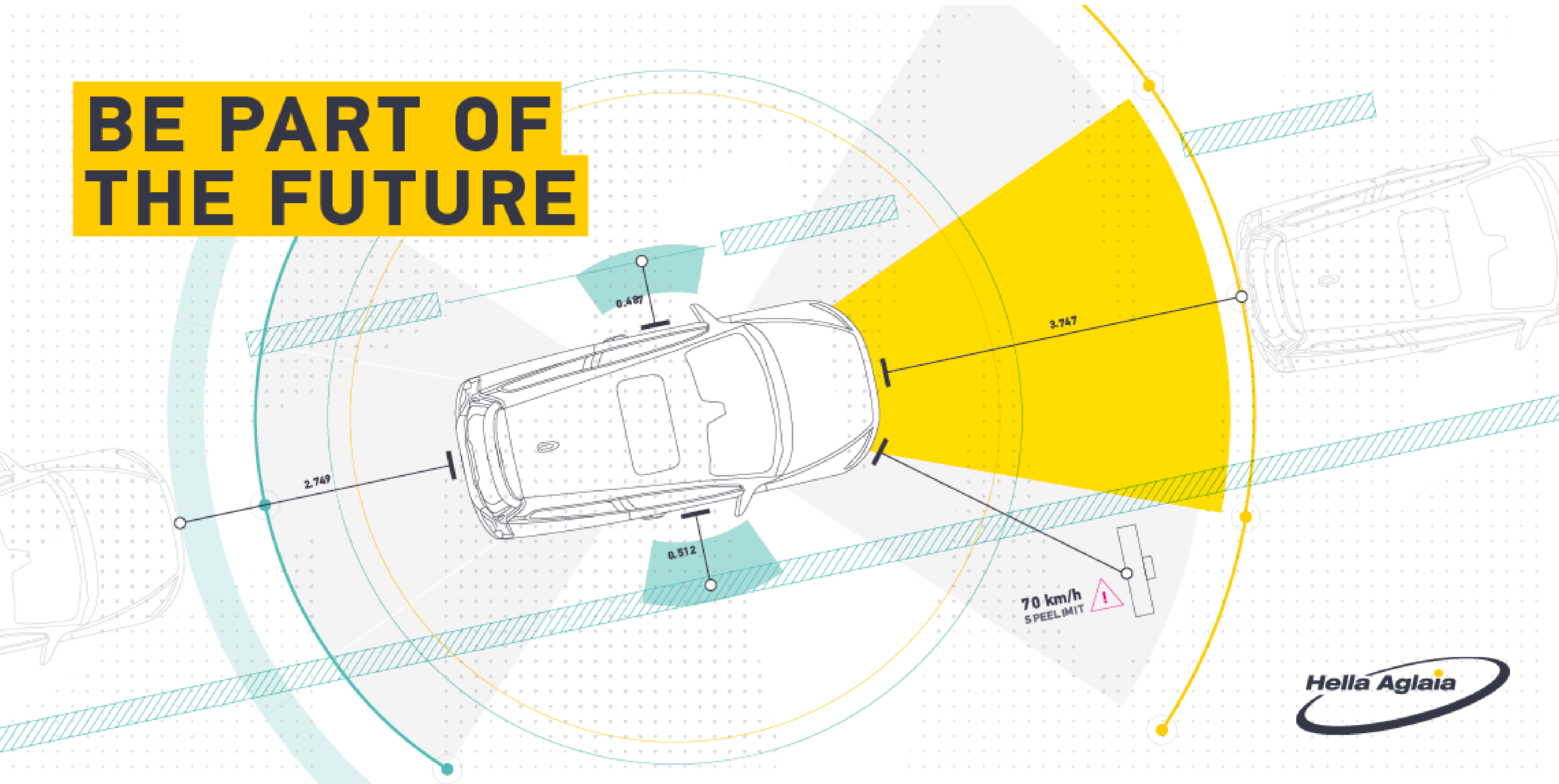
Lee et. Al [7]

Future Work:

- Incorporate Camera
  - Parking spots can be explicitly identified
- Documentation of issues
  - Likely more important than results…
  - Debugging/Approach more useful for Hella Aglaia
  - Code Refactoring
- Improve Object Tracking
- Improve VisCom Station
- Fuse with UKF (post-course)
- Implement on Audi R8 ("borrowed" from Hella Aglaia, of course)



123rf.com[4]


BE PART OF
THE FUTURE
0.487
3.747
2.749
0.512
70 km/h
SPEEL IMIT
Hella Aglaia

Thank you for your attention.

Questions?

Contact:

Our Team Is:

→ Anjaly Ealias
→ Shreyas Gokhale
→ Tuo Kang
→ Christopher O'Hara
→ Haripressana Prabakaran

Please contact:

ohara124c41@gmail.com

Research:

[1] www.udacity.com

[2] http://www.aglaia-gmbh.de/sites/default/files/styles/frontpageteaser/public/auto2_gross-neu_0.jpg?itok=ZSI3Yu95

[3] https://training.ti.com/adas-advanced-driver-assistance-systems-overview?cu=1136060&keyMatch=camera%20radar%20lidar%20sensor%20fusion&tisearch=Search-EN-Everything

[4] https://www.123rf.com/photo_95162952_3d-rendering-robot-hand-holding-together-or-robot-teamwork.html

[5] S. Houben, M. Komar, A. Hohm, S. Lüke, M. Neuhausen and M. Schlipsing, "On-vehicle video-based parking lot recognition with fisheye optics," *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*,

[6] https://news.voyage.auto/an-introduction-to-lidar-the-key-self-driving-car-sensor-a7e405590cff

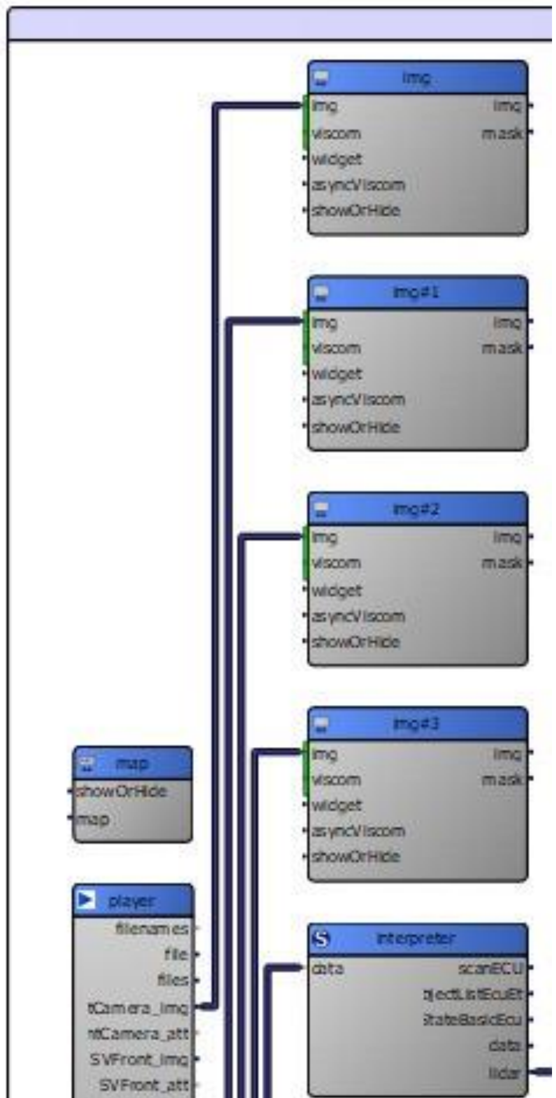
[7] Bijun Lee a , Yang Weia , I. Yuan Guo a "AUTOMATIC PARKING OF SELF-DRIVING CAR BASED ON LIDAR." The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume XLII-2/W7, 2017 ISPRS Geospatial Week 2017, 18–22 September 2017, Wuhan, China

[8] https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcRYaQ45MoMEgRsHwXy4TYL8w2l0UH2b4Ak6Ym-r1cy7gxx3I2WZ

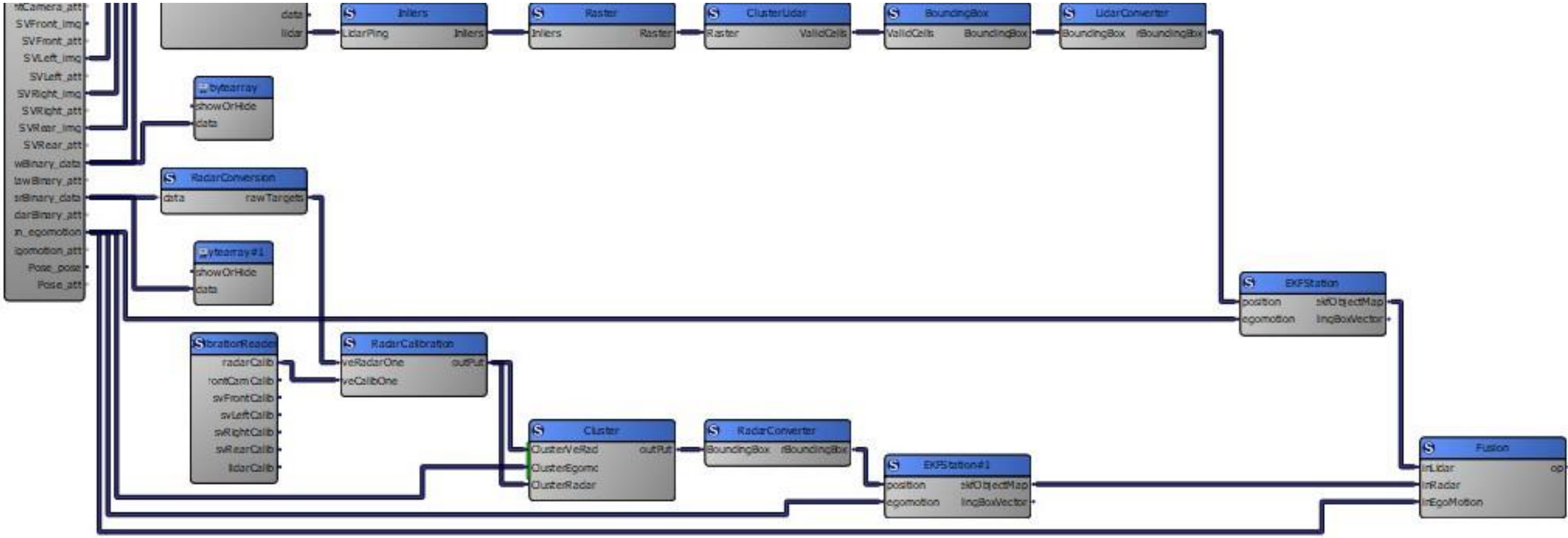
[9] https://ccv.wordpress.fos.auckland.ac.nz/

[10] https://www.hella.com/press/en/Technology-Products-Company-19-01-2018-16391.html

Stations:


img
img
viscom
widget
asyncViscom
showOrHide
img
mask
img#1
img
viscom
widget
asyncViscom
showOrHide
img
mask
img#2
img
viscom
widget
asyncViscom
showOrHide
img
mask
img#3
img
viscom
widget
asyncViscom
showOrHide
img
mask
map
showOrHide
map
player
filenames
file
files
interpreter
data
scanECU
data
lidar

Stations:



Inliers
LidarPing
Inliers
Raster
Inliers
Raster
ClusterLidar
Raster
ValidCells
BoundingBox
ValidCells
BoundingBox
LidarConverter
BoundingBox
data
lidar
bytearray
showOrHide
data
RadarConversion
data
rawTargets
bytearray#1
showOrHide
data
RadarCalibration
outPut
EKFStation
position
egomotion
Cluster
ClusterVeRad
outPut
ClusterEgomo
ClusterRadar
RadarConverter
BoundingBox
EKFStation#1
position
egomotion
Fusion
inLidar
inRadar
inEgoMotion
radarCalib
svFrontCalib
svLeftCalib
svRightCalib
svRearCalib
lidarCalib
SVFront_img
SVFront_att
SVLeft_img
SVLeft_att
SVRight_img
SVRight_att
SVRear_img
SVRear_att
Pose_pose
Pose_att