Development of Novel Sensor Fusion Architecture for Autonomous Vehicles

Pranav Manpuria , CEO

Rajkumar Palanisamy , Team Lead, Sensor Fusion

About Flux Auto

- Founded in January 2017 with the goal of democratizing autonomous vehicle technology and bring its benefits to commercial vehicles around the world.
- Team of over 50, Flux Auto is still the only company to have successfully validated its technology on Indian roadways.
- Flux Auto can integrate into existing vehicle and technology platforms to quickly enable autonomous driving capabilities to clients

Economic Times

"Flux Auto believes the chaotic, poorly-kept roads of India are ideal grounds for them to master their self-driving technology."

Udacity

"Building self-driving trucks in India is no small feat. It is even more challenging than other parts of the world due to the complexity and infrastructure"

Forbes

"Flux Auto is gearing up to change the way trucks are driven on messy and chaotic roads. The technology can be retrofitted onto in-use trucks."

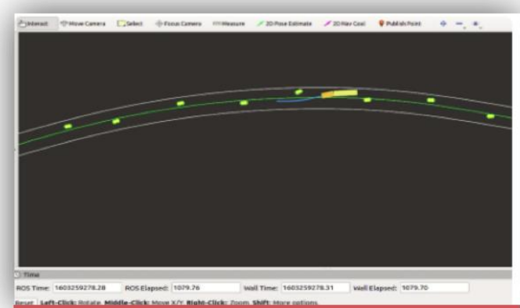
Technology at Flux Auto

Embedded Systems

Sensors

Perception

Path Planning



Localization

Simulation

Testing

<u>Autonomous Mobility</u>

The engineering challenge of a lifetime across hardware and software platforms, our engineers are working to solve new and unique challenges every day

Demo

Requirements

To develop a complete 360-degree perception stack for L3 and L4 autonomy so that it can be used for applications like Lane Change Assist, Lane Keep Assist and Behavior Prediction

- Support for 16 Cameras, 8 Radar and 6 Lidar
- Generic architecture independent of sensors
- Develop Sensor Fusion Algorithms
- Develop a validation framework

Requirements

To develop a complete 360-degree perception stack for L3 and L4 autonomy so that it can be used for applications like Lane Change Assist, Lane Keep Assist and Behavior Prediction

Support for 16 Cameras, 8 Radar and 6 Lidar

Generic architecture independent of sensors

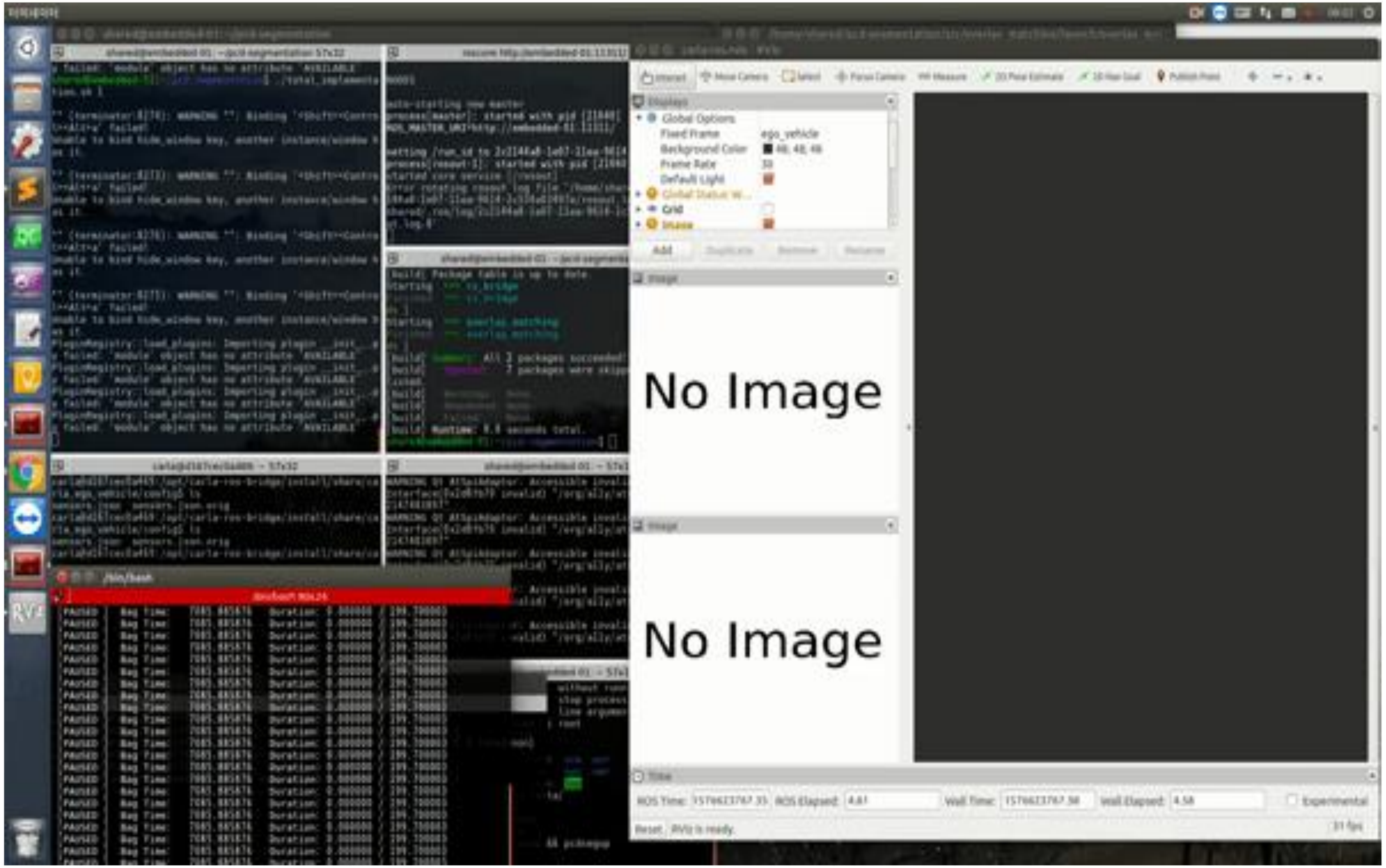
Develop Sensor Fusion Algorithms

Started Discussions With MathWorks

Develop a validation framework

Challenges in Sensor Fusion

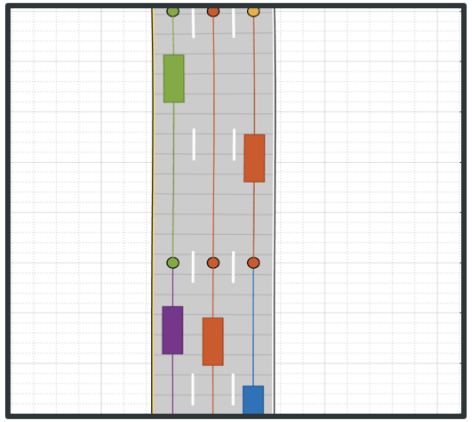
- Sensor pipeline
- Time synchronization
- Coordinate transformation
- Stack implementation

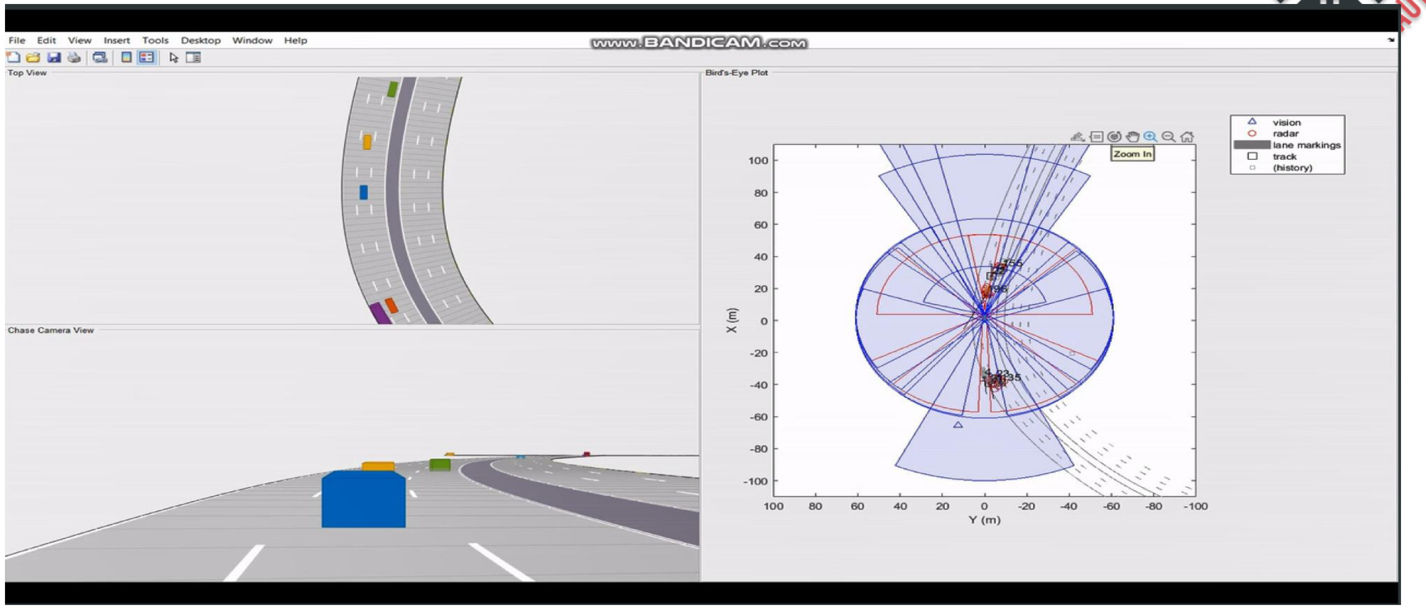


Discussion with MathWorks for Fusion Architecture

Simulation	Sensor configuration	Modularity	Architecture	Metrics for MOT	Code Generation
Software testing for developed algorithms using simulators in 2D and 3D	Evaluation of Types of sensors and placement techniques	Modular approach for development and testing	Feasibility for multi sensor fusion stack development	Metric to correlate and determinate multiple Multi Object Trackers	Code Generation to C++ for Embedded Platform

Simulation and Sensor Configuration





Sensor Stack Design

Scenario Generation and Testing

Stack Coverage and Blind Spots

Fusion Stack and Visualization in 2D/3D

Simulation

Sensor configuration

Modularity

Architecture

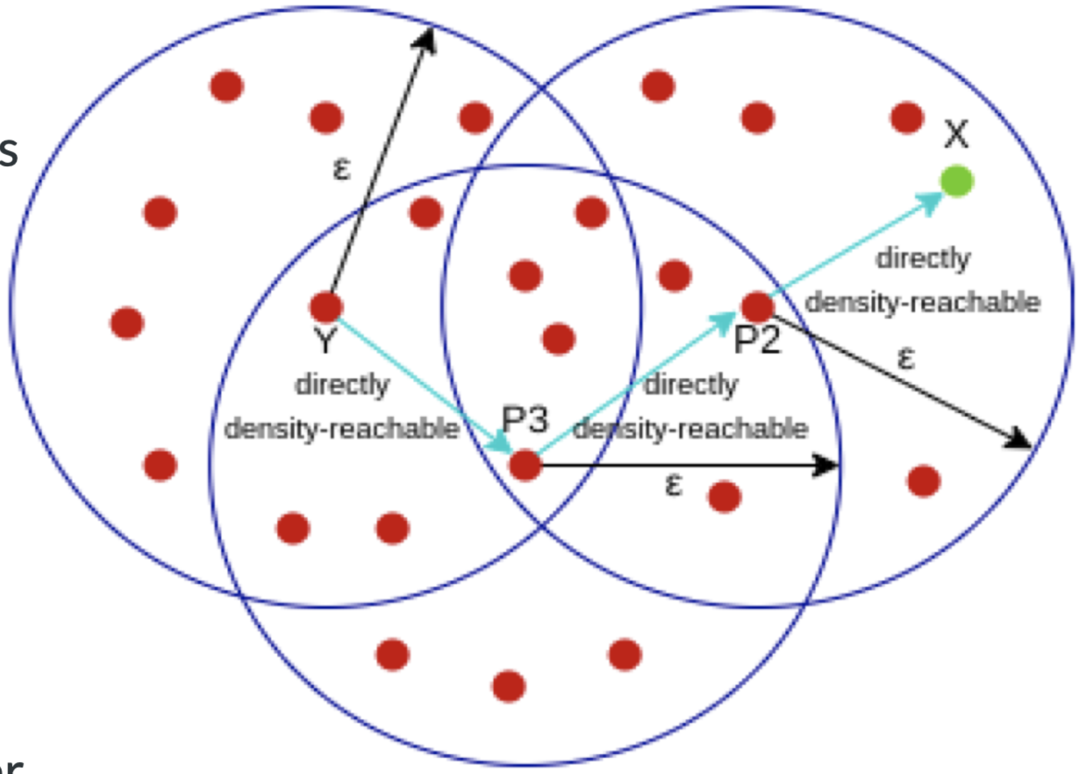
Metrics for MOT

Code Generation

Pre-Processing

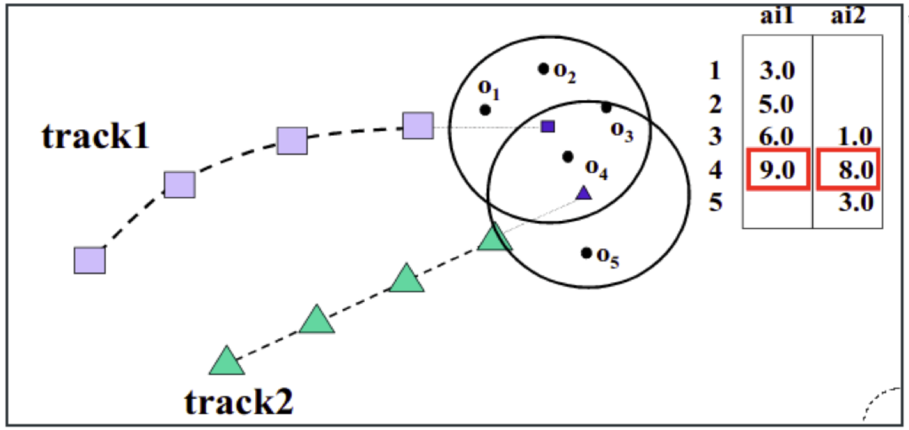
- **Filters (Interacting Multiple Model)**
  - Uses Multiple model like EKF and KF in a single filter
  - Filter uses transition probability to switch motion models
  - Optimal Filter to solve nonlinearity

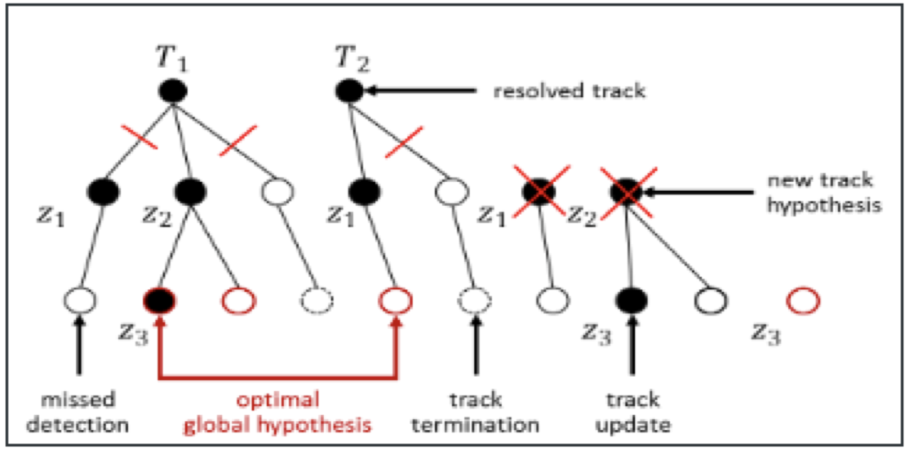
- **DBSCAN Clustering**
  - Density-Based Spatial Clustering of Applications with Noise
  - DBSCAN works based on EPS and Min points
  - **EPS :** Minimum distance for points to cluster
  - **Min points :** Minimum number off points to form a cluster



Multi Object Trackers

- **Global Nearest Neighbor**
  - Euclidean Distance based method
  - Linear Assignment solver
- **Joint Probabilistic Data Association**
  - Probabilistic data association method
  - Overlap problem in PDA
  - Each possible assignment becomes a hypothesis with an associated probability
- **Multi Hypothesis Tracker**
  - Merge Multiple hypothesis
  - Common observation history
  - Similar current state estimates
  - Discard low probability hypotheses
- **Gaussian Mean Probability Hypothesis Density**
  - Filter assumes the target states are in Gaussian
  - Pruning Gaussian terms of smaller weights

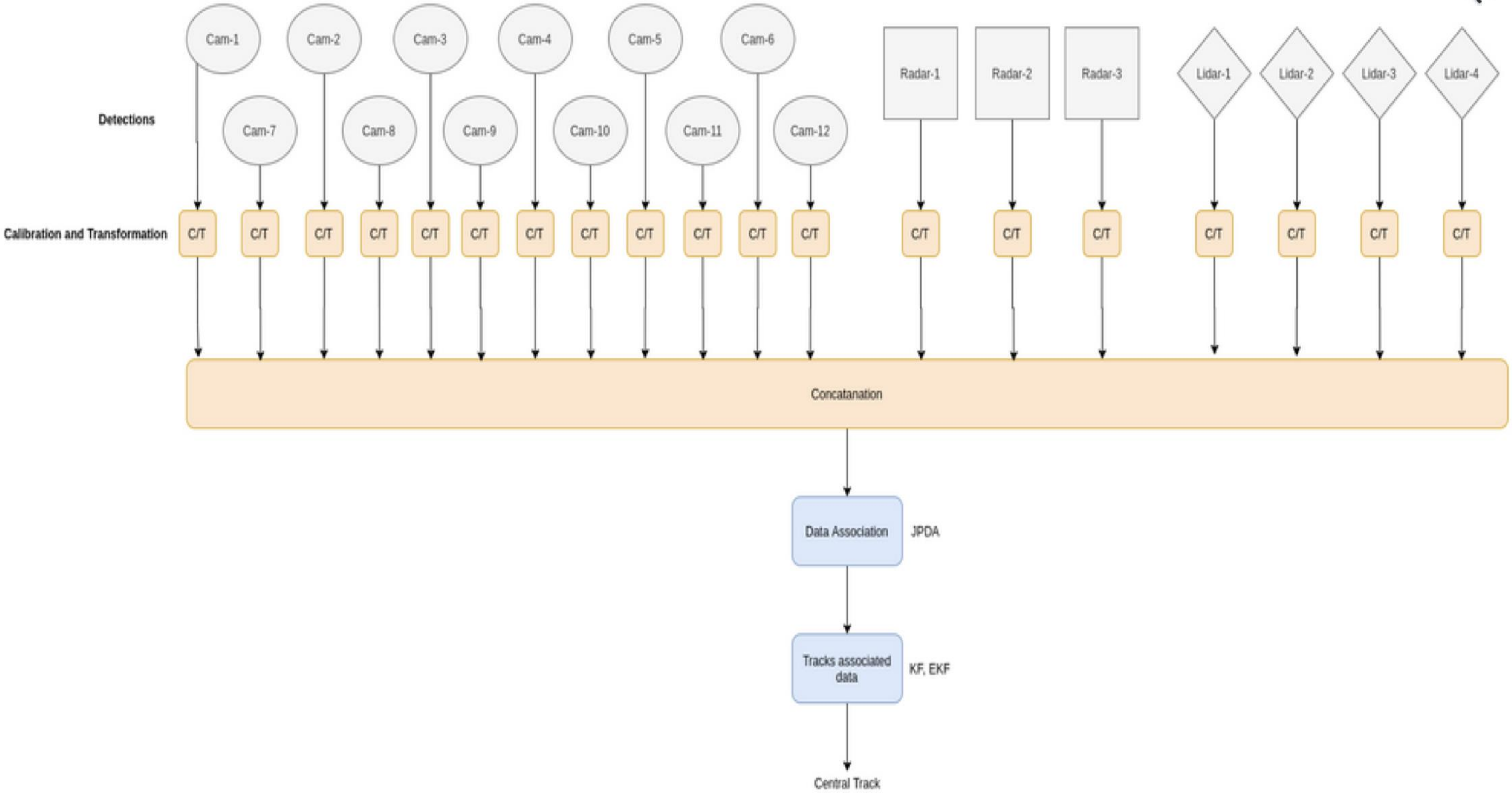




Pipeline & Data Processing

- Sensor Abstract layer
- ROS Sensor Drivers
- Coordinate transformation
- Concatenation
- Filters
- MOT / Data association
- Central tracks

Detection Level Fusion

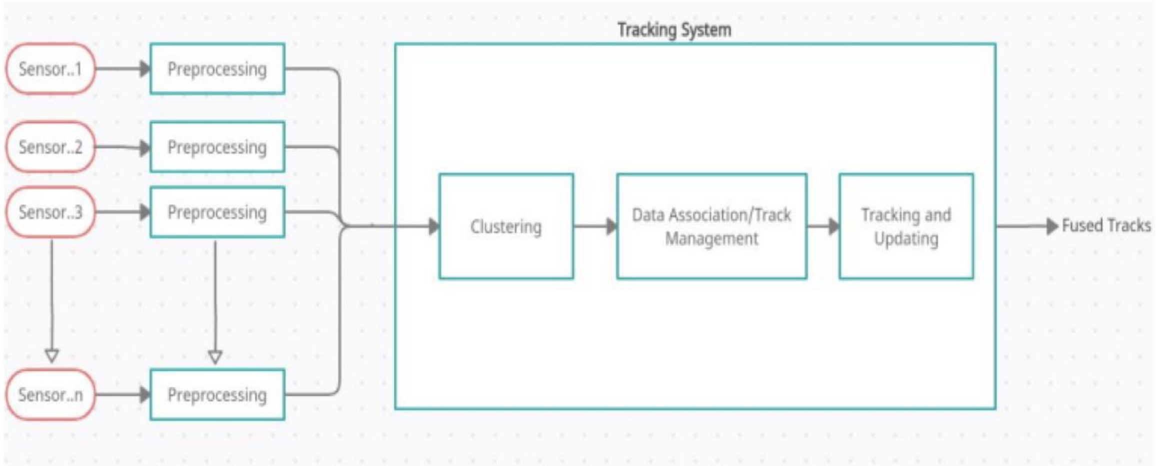


Simulation | Sensor configuration | Modularity | Architecture | Metrics for MOT | Code Generation

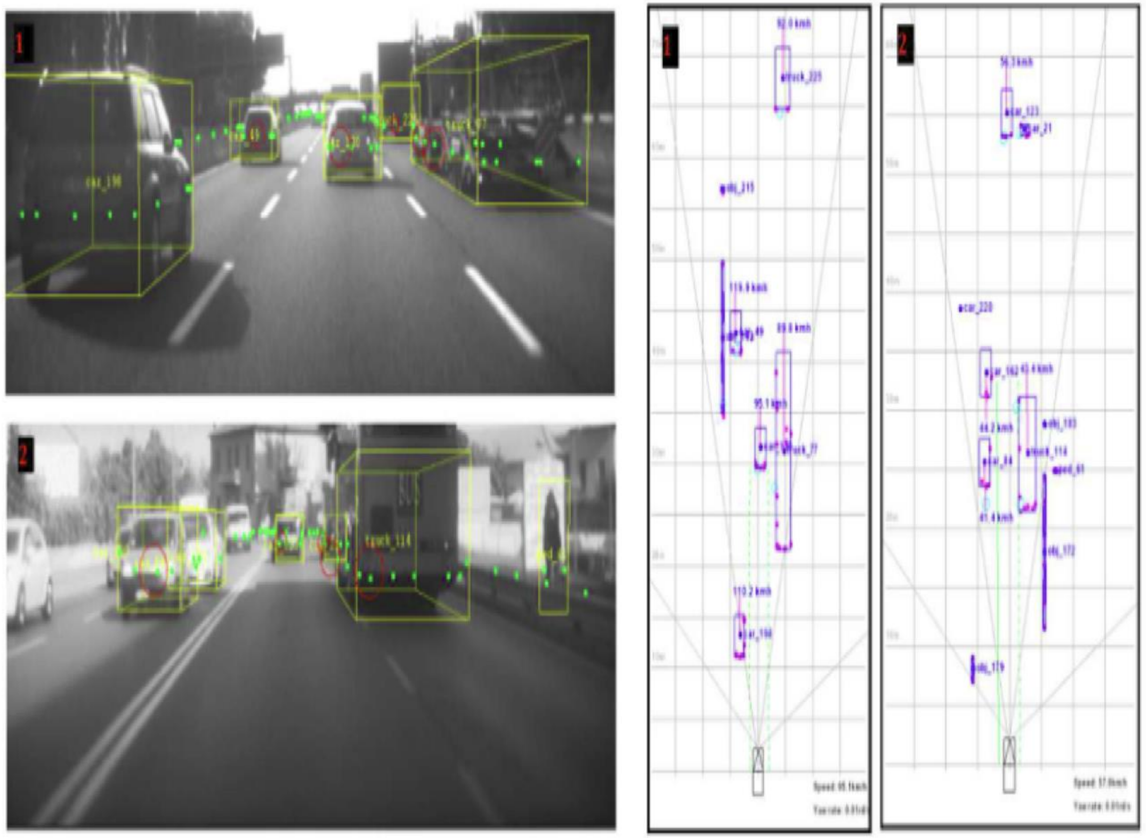
High level fusion algorithm architecture

Detection Level Fusion

- Receiving detection list from multiple sensors
- Preprocess the sensor data to remove clutters
- Data is sent to a clustering algorithm
- Data association and track management system for initialization and deletion of tracks



Detection Level Fusion Architecture



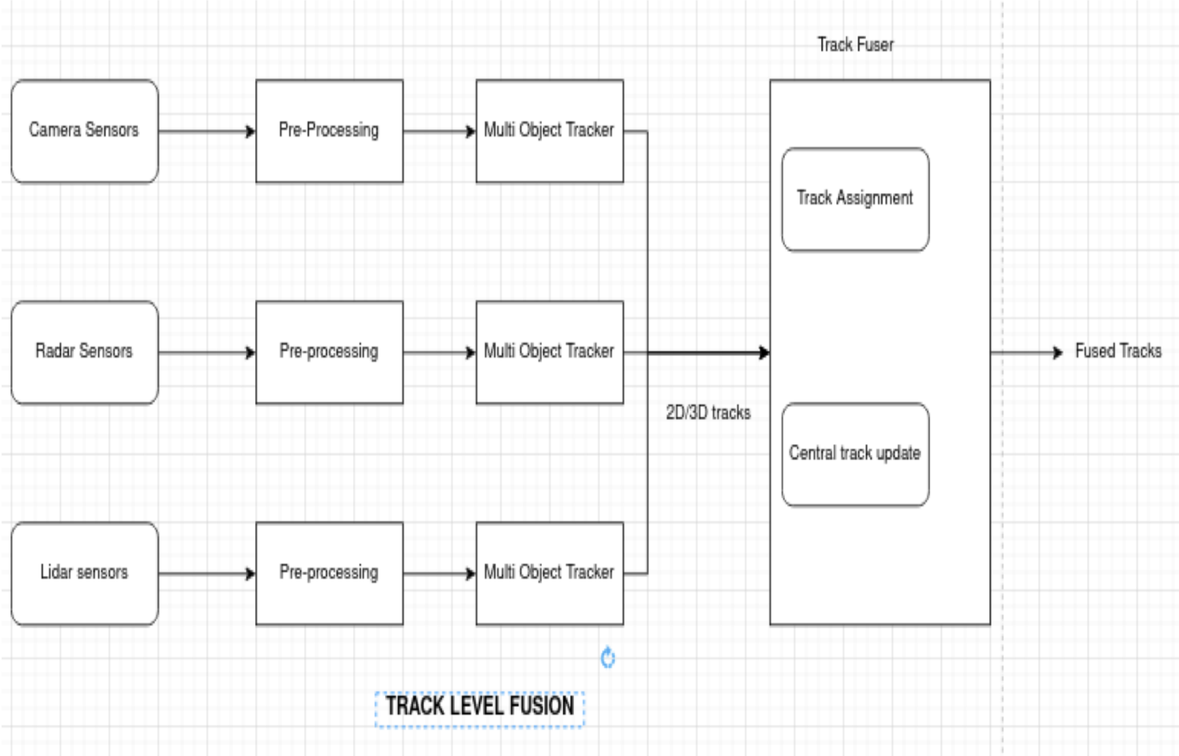
High level fusion algorithm architecture

Track Level Fusion

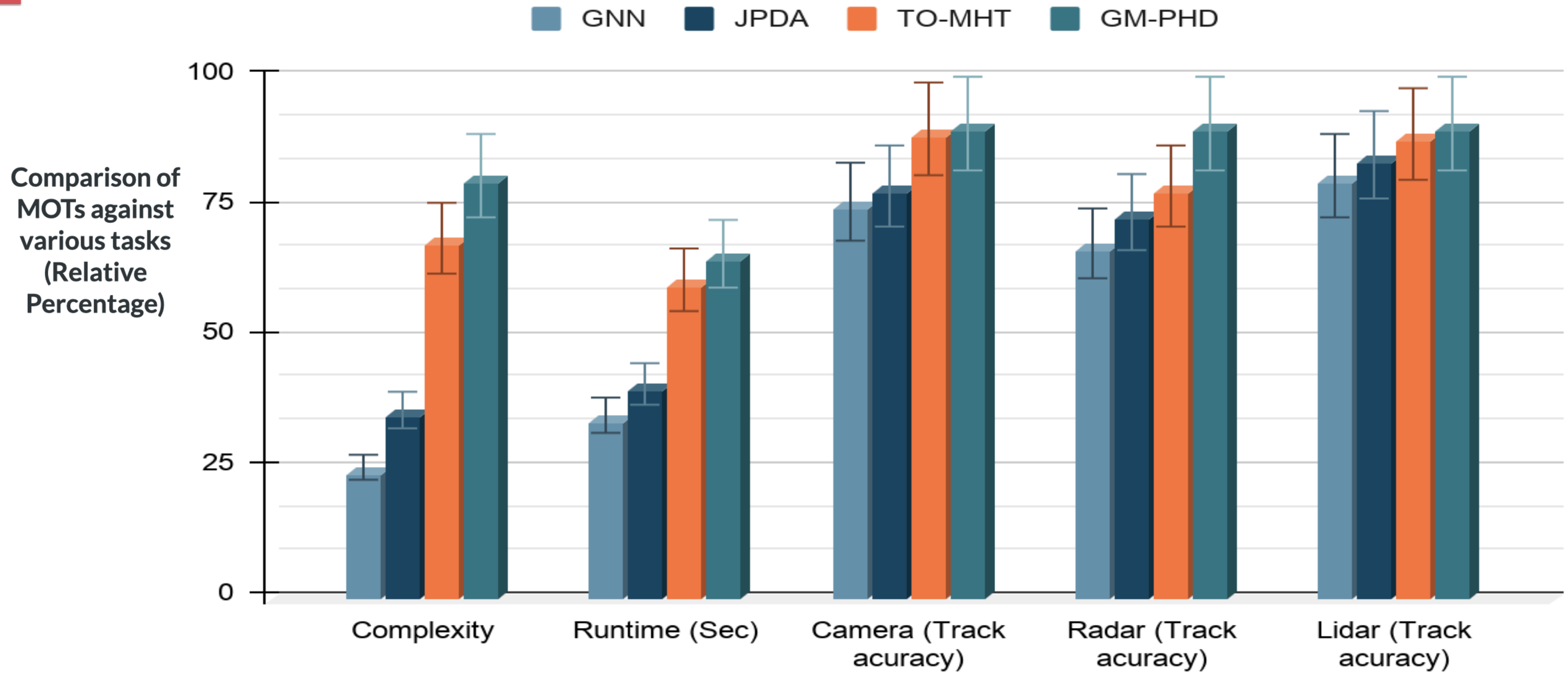
- High Level fusion is characterized by distributed tracking in each of the sensor and a central fusion combining the already tracked targets from multiple sensors, for which it is also called track-to-track, sensor-level, or distributed fusion.

- Distributed fusion architecture can handle nonlinearity of the sensor at the sensor level which would increase our accuracy and minimize noise and false alarms picked up by the sensors

- The main module for track level fusion is the central track fuser where all the tracks from individual sensors are assigned and updated.



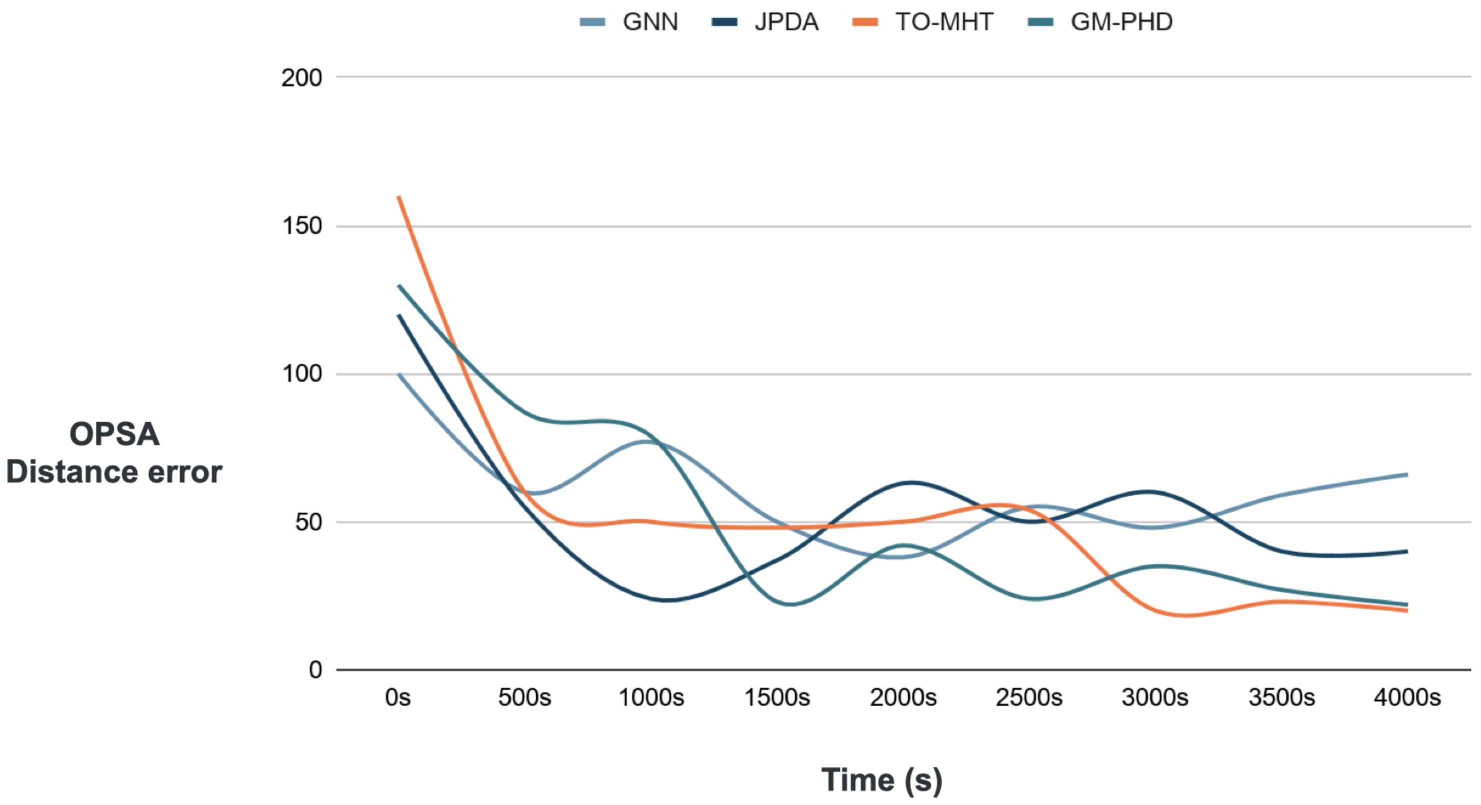
Performance comparison of Multi Object trackers



Simulation → Sensor configuration → Modularity → Architecture → **Metrics for MOT** → Code Generation

Metrics for Multi Object Tracker

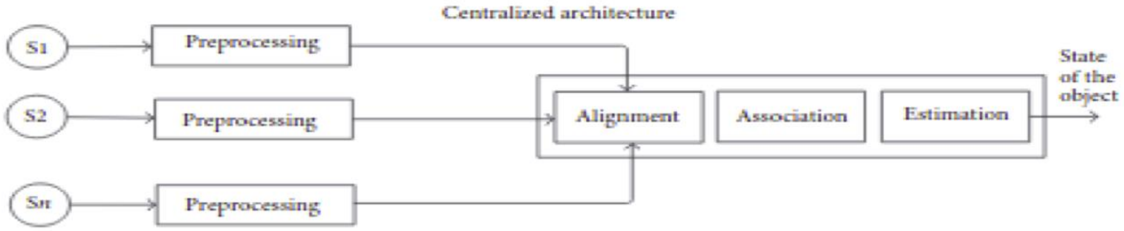
- The metrics we used to evaluate out Multi object trackers are based on optimal sub-pattern assignment
- GOSPA : GOSPA is not normalized by the cardinality of the largest set and it penalizes cardinality errors differently, which enables us to express it as an optimization over assignments instead of permutations
- Traditionally one needs to compute the similarity between the ground truth and the estimated set.

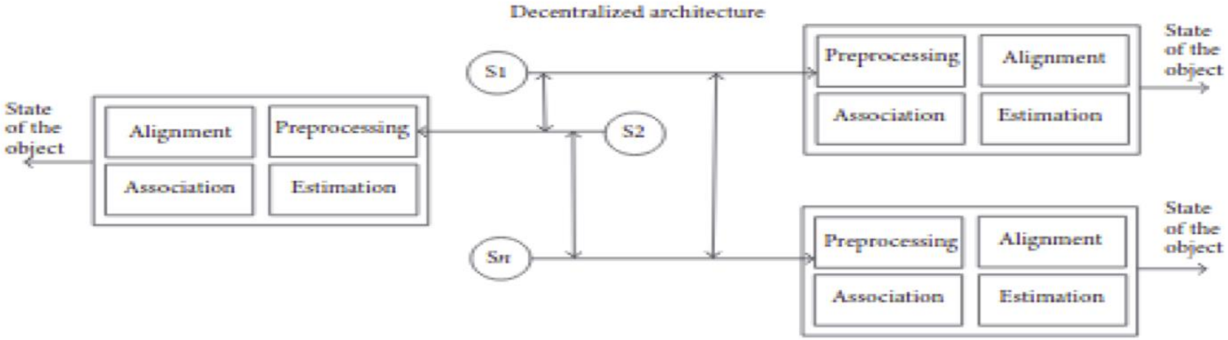


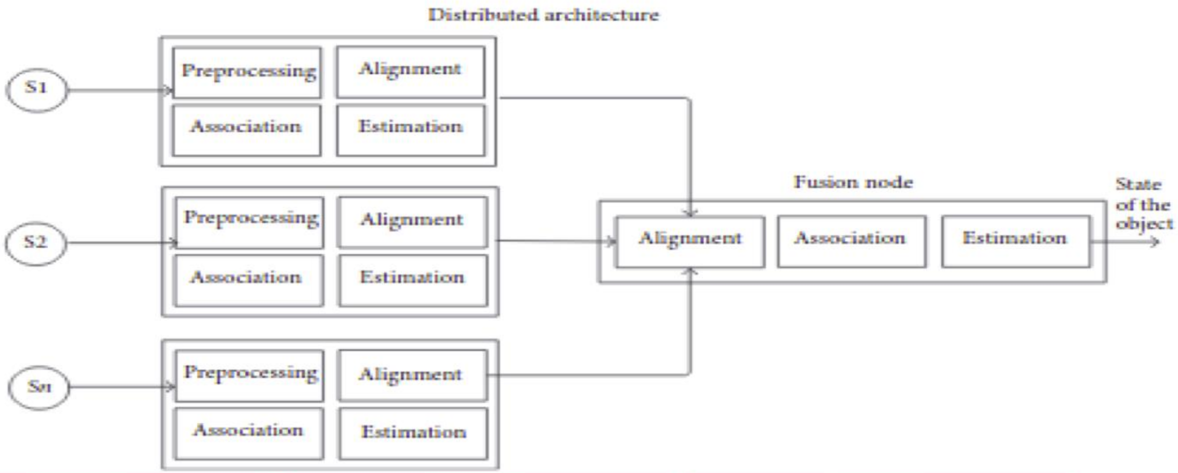
MOT advantages based on Fusion Architecture

- **Track Level Fusion :**
  - Distributed Architecture for Different types of sensors
  - Sensor dependent MOT
  - Handles nonlinearity in sensor level, reduced cardinality
  - Handles sensor Alignment
  - Handles Association in Individual sensor groups
  - Multiple levels of state estimators reduce mismatch and false alarms

- **Detection level Fusion :**
  - Lesser Compute required
  - Alignment / Association is handled Independent of the sensors
  - JPDA / GNN based architecture
  - Takes advantage of JPDA clusters and PDA based fusion
  - Sensor Independent

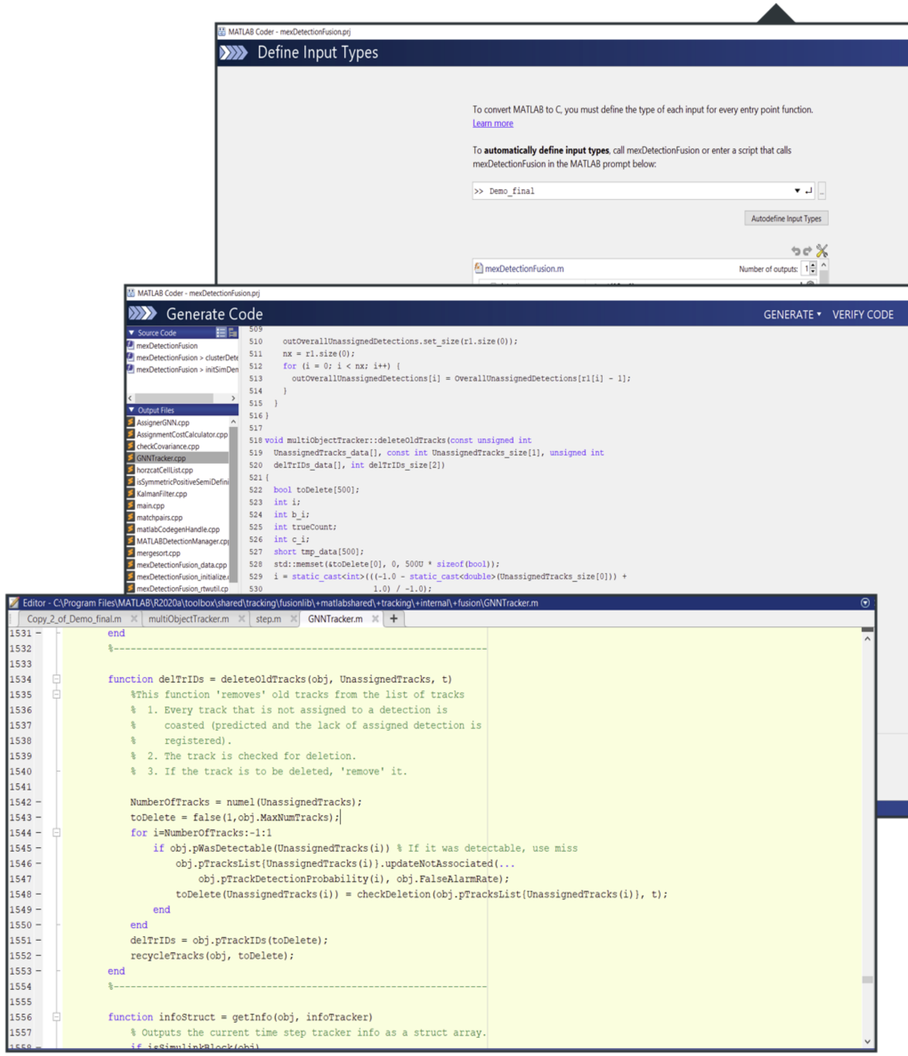






Code Generation

- Code generation was done using MATLAB Coder
- Code generation was done for each individual module
- Compatible with multiple hardware - Architectures like x86-64, ARM 64-bit, ARM-Cortex etc.
- Catkin/MinGW64/Visual studio can be used as compiler and test the generated C++ code.
- ROS integration was done manually
- SIL based testing was done by converting sensor data to JSON format and was passed as publisher to the Fusion node
- SIL was done Frame by Frame and time synchronization was handled at the sensor level
- Each as multiple detections from multiple sensor



Results Achieved

- Multi-Level Sensor Fusion
- 3D Perception Stack
- Application oriented Architecture
- Metrics

Benefits of MathWorks Tools

- Readily available packages for tracker modules (Sensor Fusion and Tracking Toolbox)
- Rich available framework for SIL (Automated Driving Toolbox)
- State of the art clustering algorithm for multiple sensors
- Stable Central track management systems based on history and score-based logic
- Reduction of Development time by 4 months
- MathWorks support at every step of the development cycle

Next Steps with MathWorks

- Roadrunner : Evaluation completed and good results of integration with CARLA
- LIDAR and Camera calibration
- ISO certification for controls
- Introduction of ROS toolbox in the deployment framework

Thank You!

Any Questions?

contact@fluxauto.xyz

www.fluxauto.xyz