NIO

Vision-Guided Self-Localization for Autonomous Cars

2017

11/7/2017
MATLAB EXPO 2017
San Jose, California

Veera Ganesh Yalla
Sathya Narayanan K*
Davide Bacchet

** Sathya is an intern from WPI and continuing his MS research at NIO*

Agenda

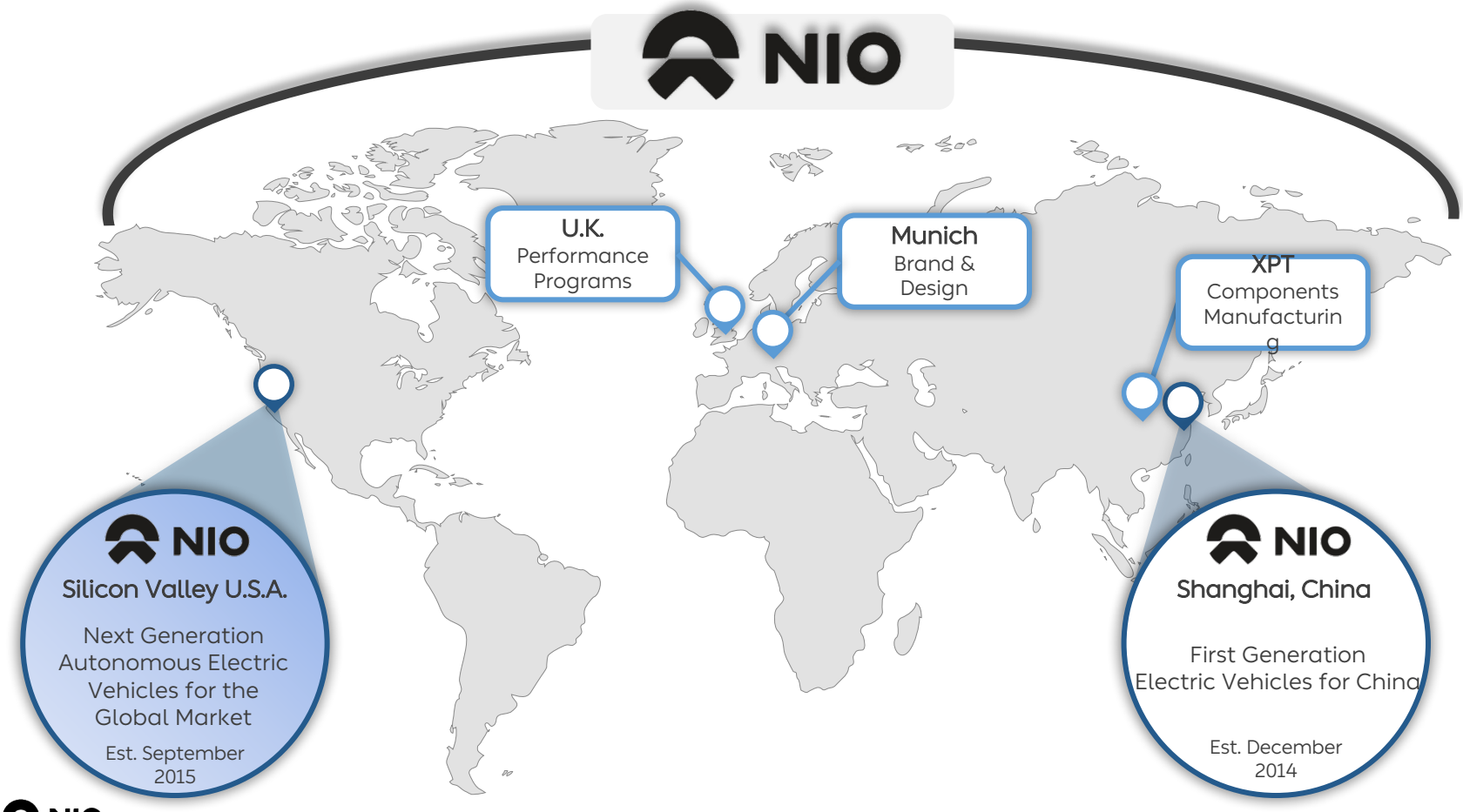
- About NIO
- Use of Various MATLAB Toolboxes at NIO
  - Project #1: EP9 Driving Autonomously at COTA
    - Control Systems Toolbox
    - MATLAB Coder
    - SIMULINK Coder
  - Project #2: Camera Calibration
    - Computer Vision System Toolbox
  - Project #3: Vision-Guided Self-Localization for Autonomous Cars
    - Image Processing Toolbox
    - Computer Vision System Toolbox
- Conclusions

NIO

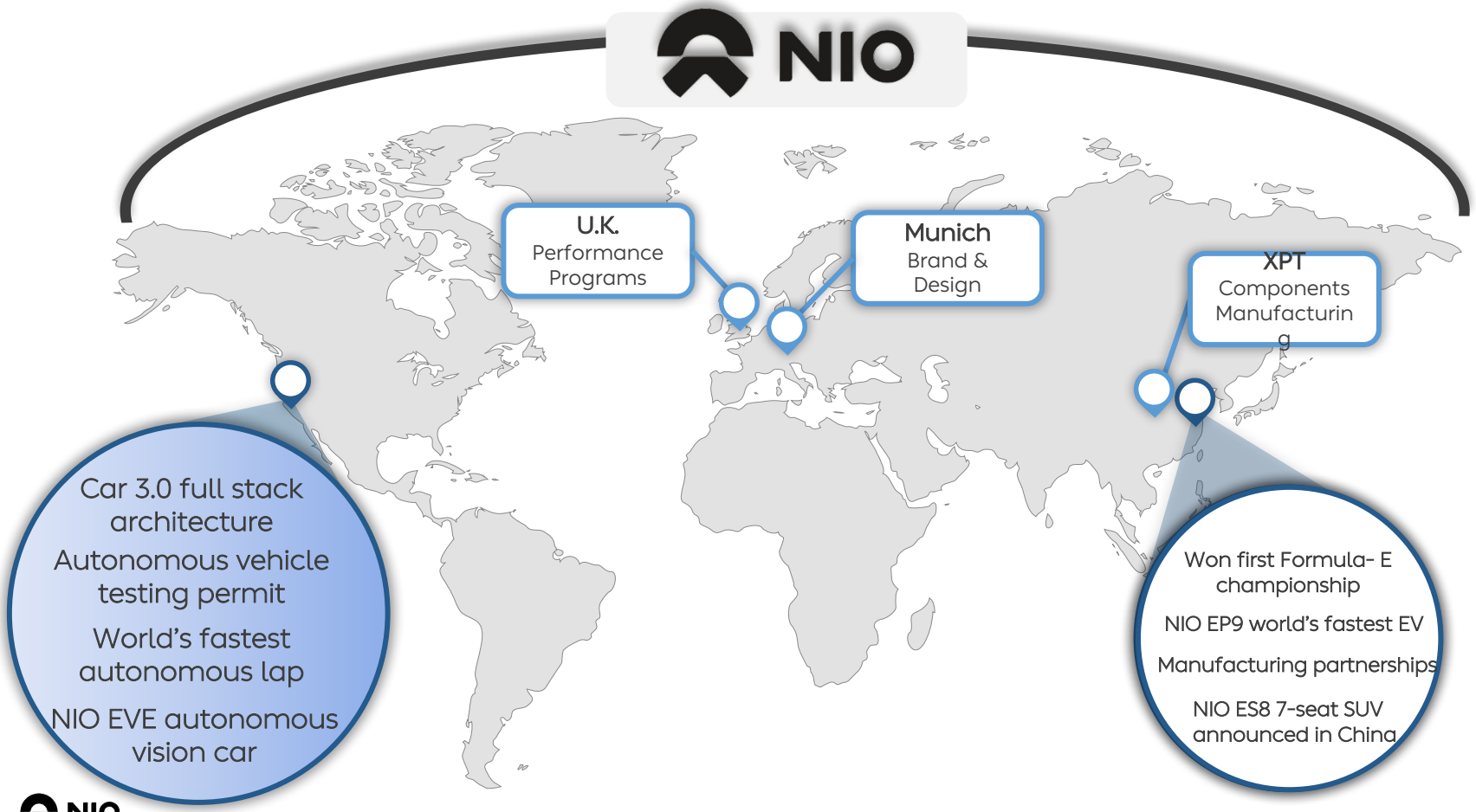
About NIO

NIO

We are a Global Startup



Significant Progress To Date



EP9 : Autonomous Mode at COTA

Key Personnel:

Aaron Bailey (Head of Driving Dynamics)
Dennis Polischuk (Head of Platform Engineering, Firmware)
Kamran Turkoglu (Head of Controls/Algorithms)

NIO

NIO EP9

A limited edition 1 megawatt, 3G capable, electric supercar, born to push limits.


CIRCUIT OF THE AMERICAS
FEBRUARY 23RD 2017

Camera Calibration

NIO

Camera Calibration

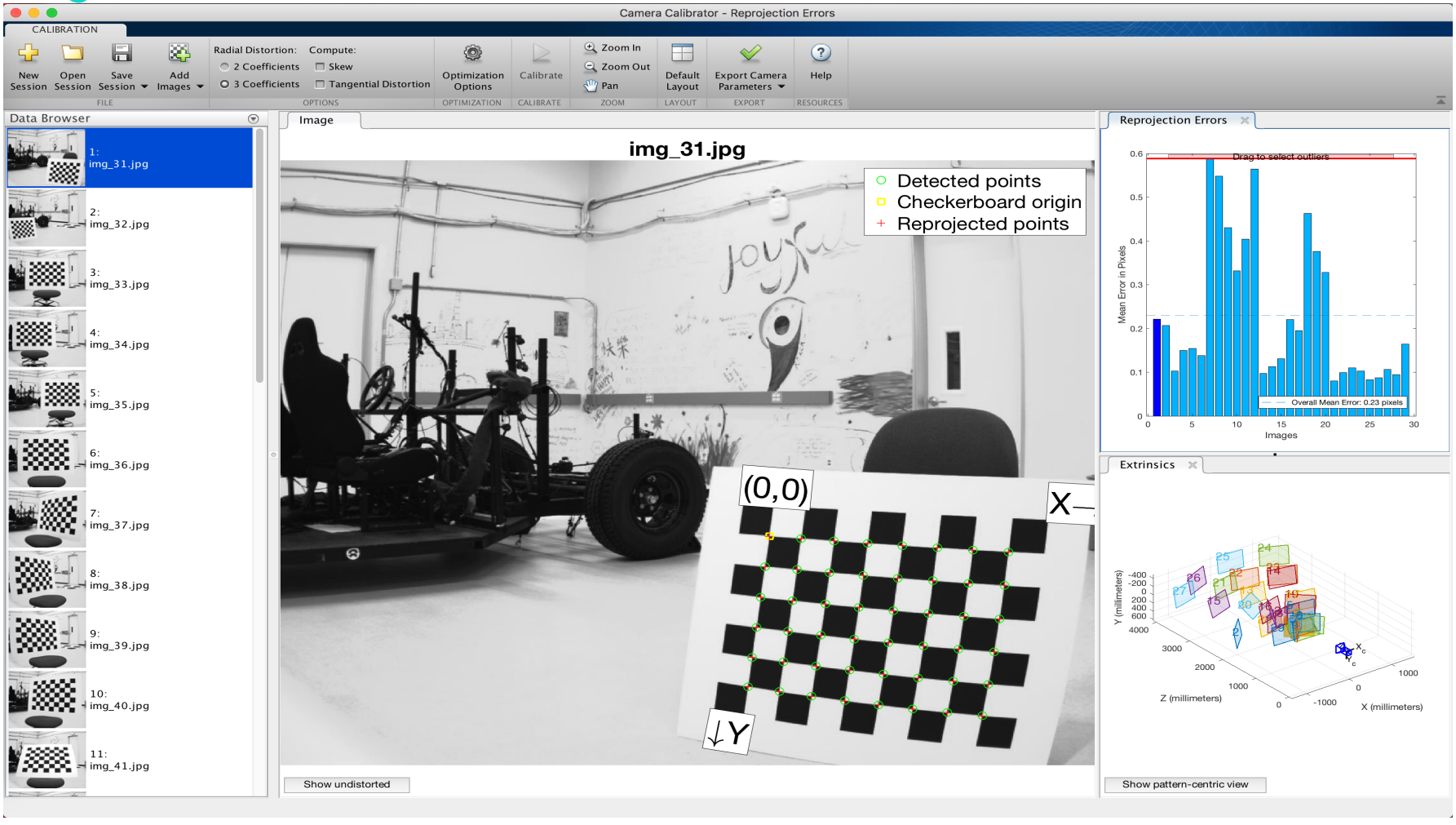
Motivation

- Rapid prototyping of calibration methods for an ADAS tri-focal camera system with three different FoVs.
  - Narrow Field of View (28 degree)
  - Wide Field of View (150 degree) *
  - Main Field of View (52 degree)

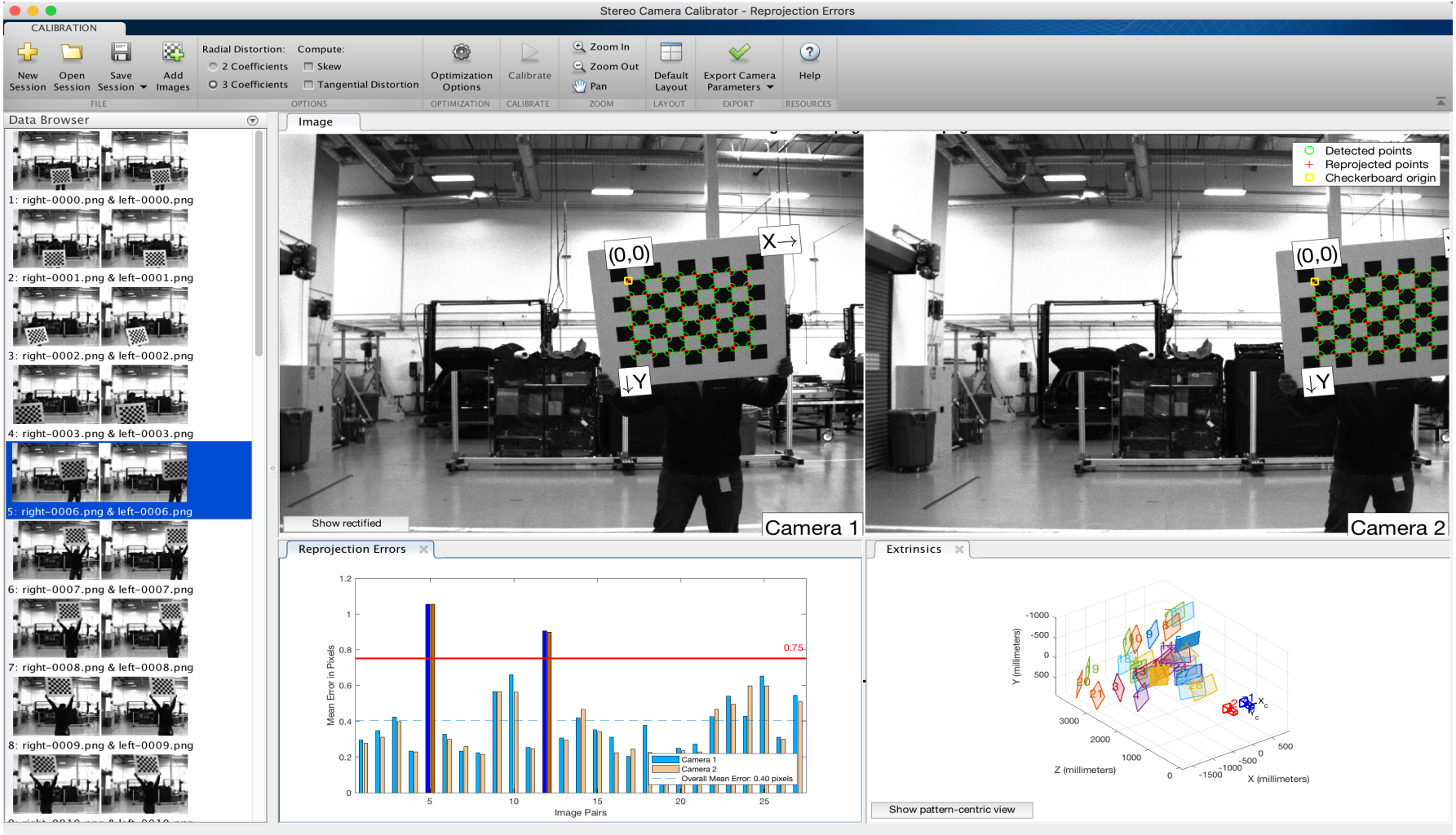
- Calibrate a stereo camera system for a mapping pilot project.

**Special Thanks to Avi Nehemiah & Mathworks Team for providing NIO access to pre-release version of MATLAB for the Wide FoV Calibration*

Single Camera Calibration – cameraCalibrator APP



Stereo Camera Calibration – stereoCameraCalibrator APP



Vision-Guided Self-Localization for Autonomous Cars

NIO

Overview

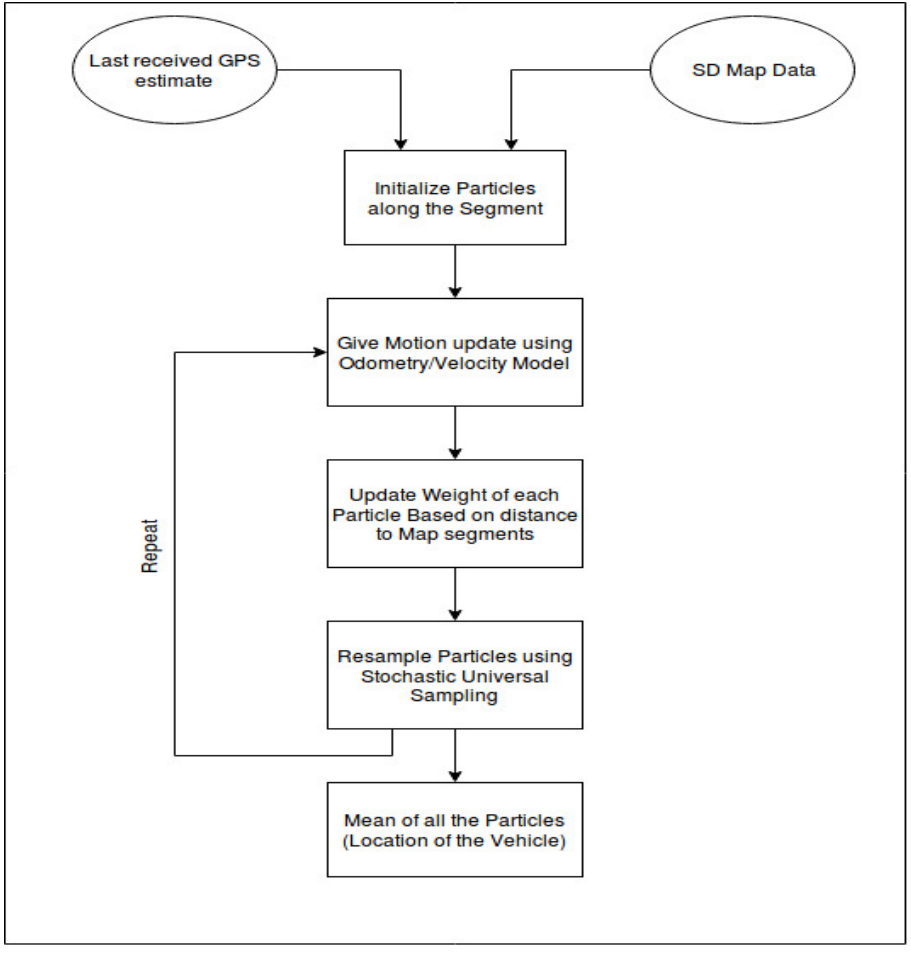
- **Motivation**
  - Limited Accuracy of GPS in Production Car (> 5 meters).
  - Poor GPS Signal Reception problems
    - Urban Areas / Urban Canyons.
    - Inside a Tunnel etc.

- **Available Input Signals**
  - Last received GPS estimate.
  - SD Maps
    - *HD maps are currently being investigated.*
  - Odometry
    - Change in position over time.
  - Camera + LiDAR
    - *Work in Progress*

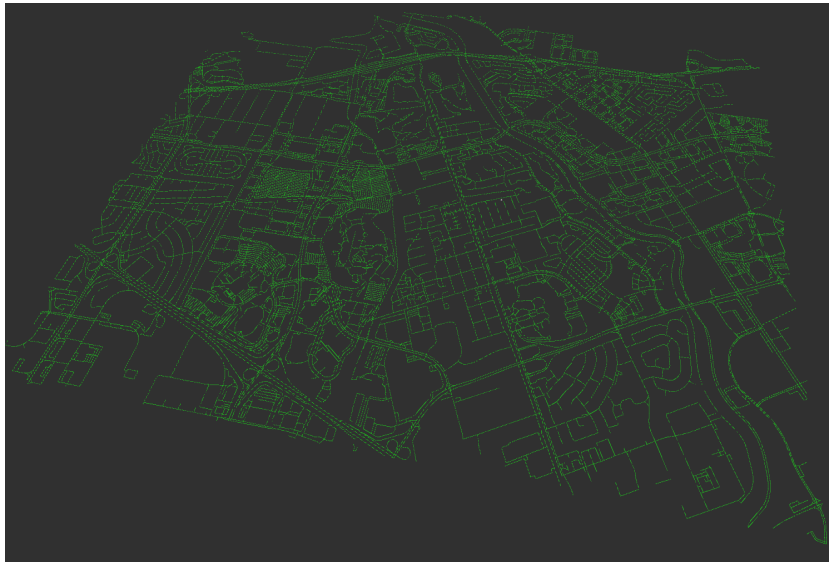
- **Method Implemented**
  - Particle Filtering

NIO

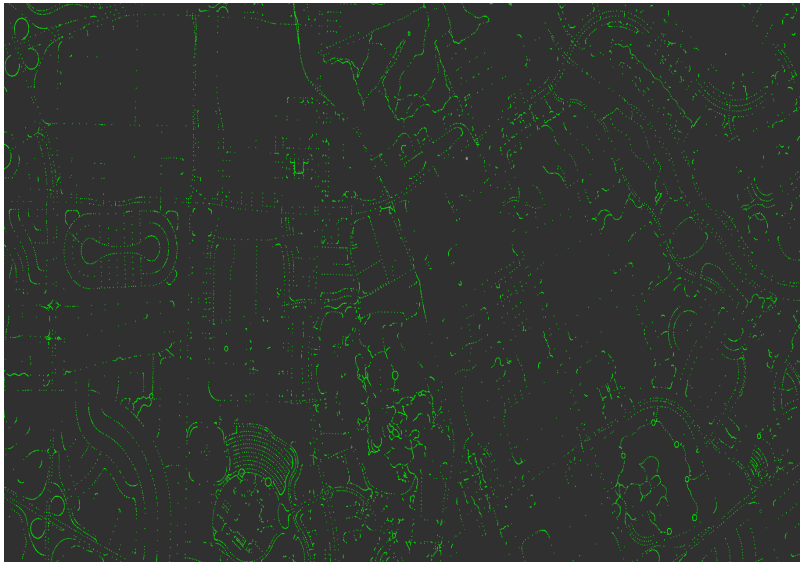
Algorithm: Using Particle Filters for Localization



SD Map



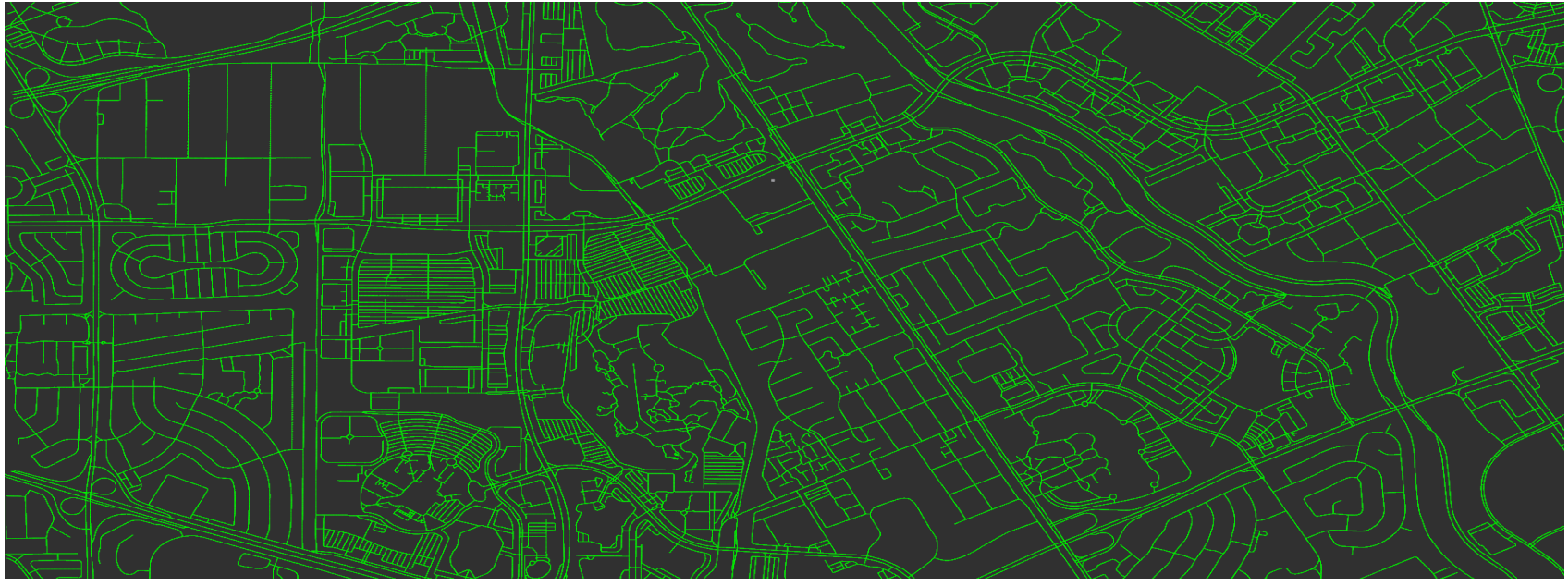
SD Map



Close Up of SD Map

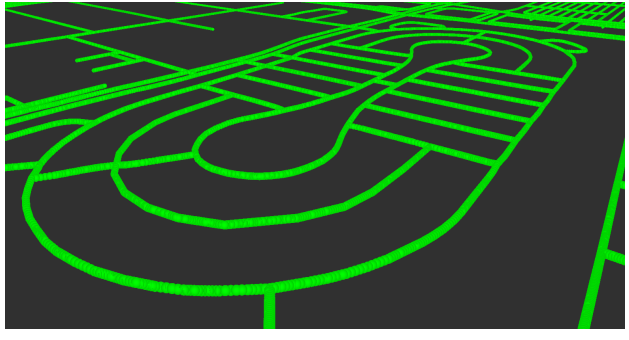
- The SD map information that is currently available contains only the Segment Geometry.
- Each segment has N points represented in the local frame.
- Number of points in the segment are sparse.
- For our application
    - We need closely spaced points to generate particles all along the segment.

SD Map - Augmentation



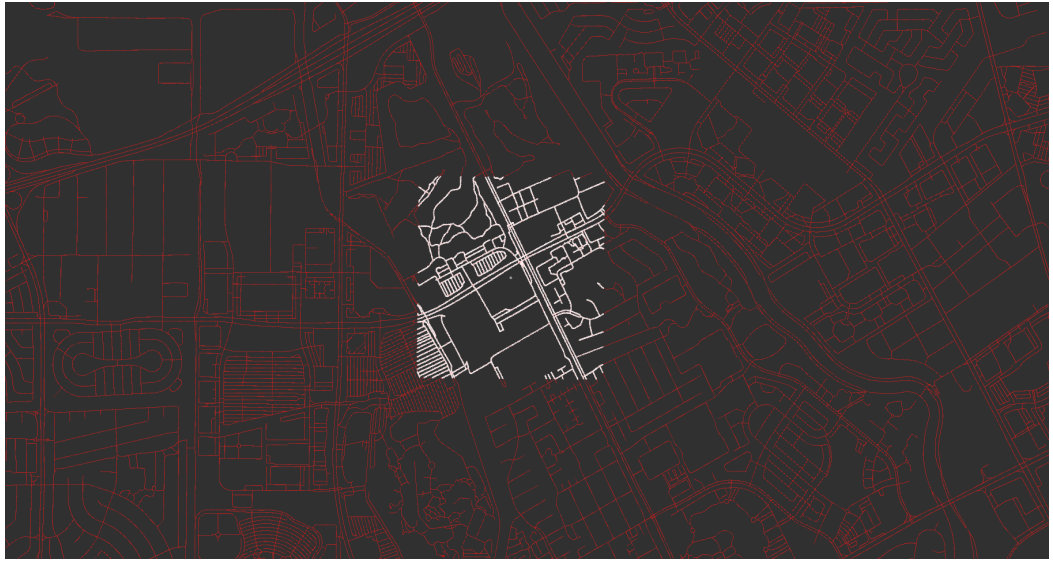
- Number points along each segment along the map is augmented.

- Due to this, we have a more dense map information for localization.



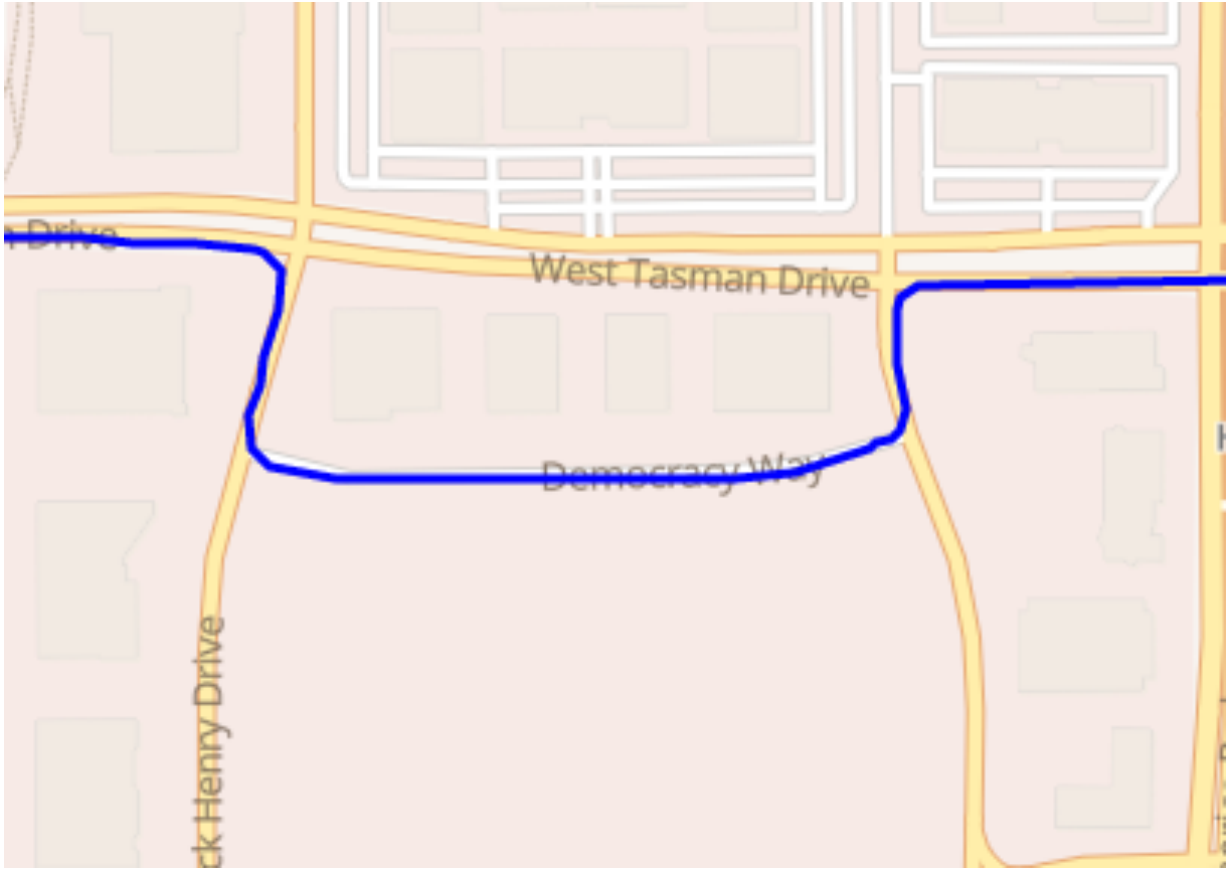
NIO

Particle Filtering for Localization



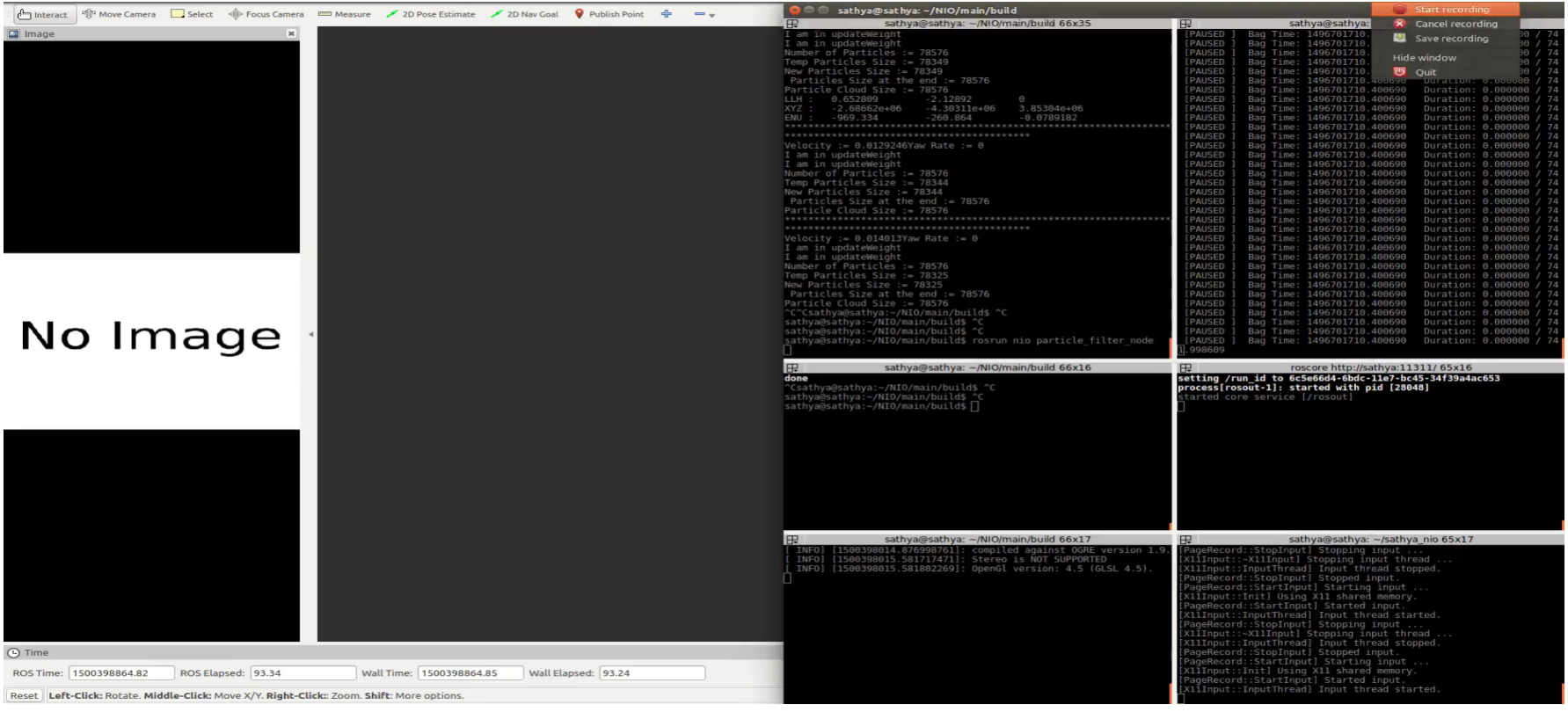
- Once we have the augmented map, we need to sample particles along the segments of the map.
    - States of the particles are x, y, θ.
- First we are extracting a 500m x 500m area using the last received GPS point.
- For augmenting, each point along the segment is considered as a possible location of the vehicle and the particle's orientation θ is the orientation of the segment.
- Using this information the particles are generated with orientations equal to θ, 180 + θ .

Experiment: Map Segment to Localize



Test Route: Trying to localize within this map segment

Results



When the Green (ground truth) and Blue dots (computed position value) overlap, the vehicle has self-localized in the map

Current Limitations

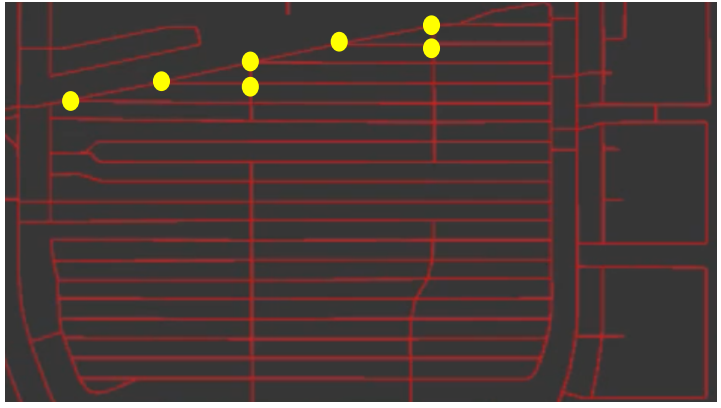
- Its an iterative approach.
- Solution takes time to converge to a location.
- Computed location can have ambiguities if the driving path is mostly along straight lines.

How can we improve ?

Proposed Solution: Add Visual Features to the Map Nodes

- Extract visual features from the camera sensors and send it to a map cloud.
- Visual features enhanced map data can then be used in the real-time localization.
- The localization itself can be done in real-time
  - Feature extraction and matching
  - Reduces the number of potential locations for the particle filters before we converge to a solution.
- This is a form of map crowdsourcing: every car contributes to build a better localization map

Augmenting the map with visual place features



Nodes

- The yellow dots represent the map nodes.
- For illustration, we show a few nodes. In reality each intersection in the map is a node point.
- The road segments (red lines) connect the different nodes in a map

Visual image captured at an intersection

Visual Features

$$\bar{x}_1 = [x_{1,1} \quad x_{1,2} \ldots\ldots\ldots\ldots\ldots\ldots x_{1,p}]$$

$$\bar{x}_2 = [x_{2,1} \quad x_{2,2} \ldots\ldots\ldots\ldots\ldots\ldots x_{2,p}]$$

$$\bar{x}_n = [x_{n,1} \quad x_{n,2} \ldots\ldots\ldots\ldots\ldots\ldots x_{n,p}]$$

The red boxes indicate some robust visual features we would like to extract.

- We would store this info at each yellow dot above during mapping and also as a dynamic update.
- During localization, we extract the features and do feature matching. This will reduce the number of particle filters we need and the places we search for a matching location.

References

- Levinson, Jesse, Michael Montemerlo, and Sebastian Thrun. "Map-Based Precision Vehicle Localization in Urban Environments." *Robotics: Science and Systems*. Vol. 4. 2007.

- Brubaker, Marcus A., Andreas Geiger, and Raquel Urtasun. "Lost! leveraging the crowd for probabilistic visual self-localization." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2013.

- Ruchti, Philipp, et al. "Localization on openstreetmap data using a 3d laser scanner." *Robotics and Automation (ICRA), 2015 IEEE International Conference on*. IEEE, 2015.

Conclusions

NIO

Conclusions

- Presented an overview of the various projects at NIO using Matlab Toolboxes
  - Project #1: EP9 Driving Autonomously at COTA
    - With the help of Control Systems Toolbox, MATLAB Coder and SIMULINK Coder, NIO was able to implement the autonomous driving feature (GPS waypoint following) and successfully demo EP9 at the COTA.
    - Our Controls and Algorithms team use Control Systems Toolbox, MATLAB Coder and SIMULINK Coder for the L2/L4 features currently in development.
  - Project #2: Camera Calibration
    - With the Computer Vision System Toolbox, NIO was able to do rapid prototyping of the various calibration algorithms for the ADAS trifocal camera system.
  - Project #3: Vision-Guided Self-Localization for Autonomous Cars
    - NIO currently relies on the Image Processing Toolbox and the Computer Vision System Toolbox for the visual feature recognition that we are incorporating into improving the self-localization algorithm.
- In addition,
  - NIO is currently using Automated Driving System Toolbox for some its R&D projects.

NIO

Thank you !!

NIO is hiring
https://www.nio.io/careers

NIO