Vision and Image Sciences Laboratory

Final Presentation

SLAM for Formula Driverless

Students:
Lital Guy, Maxim Aslyansky

Supervisors:
Eli Appelboim, Israel Berger

Context: Project B

Semester: Winter, 2019

Date: 02/04/2019

Presentation Outline

- Introduction and Prior Work
- Our Setup and System Considerations
- LIDAR Perception and LIDAR SLAM
- Visual Cone Detection
- Visual SLAM
- Building Cone Map
- Conclusions and Future Work

Project overview

Formula Student is a student engineering competition where student teams from around the world design, build, test, and race a small-scale formula style racing car. In 2017 Driverless competition was added.

Our contribution to the team:

Reconstruct the race track and estimate the vehicle position on the map while driving (SLAM)

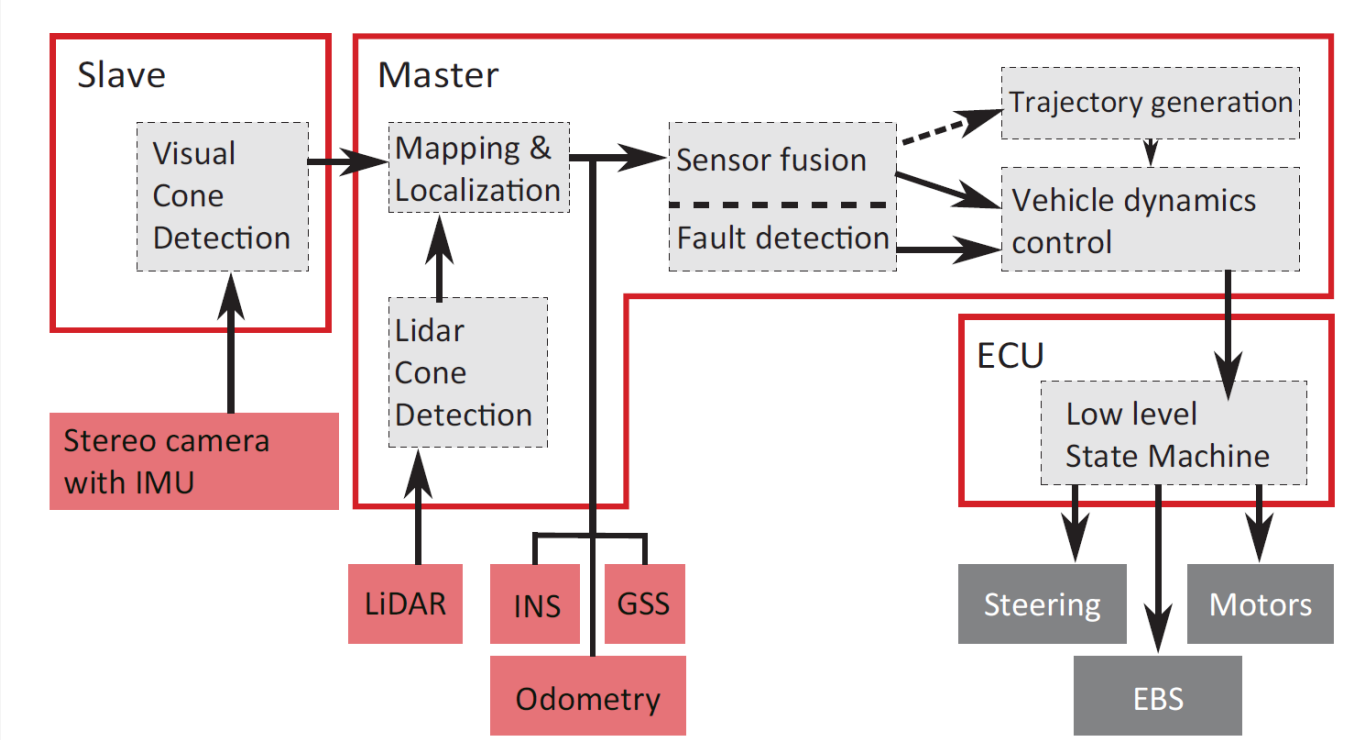
Vision and Image Sciences Laboratory

Project Goal

- Mapping is crucial to achieve high speed
  - During the first loop environment map is built
  - Allows to significantly improve speed by planning beforehand
- Main challenges
  - The algorithm must run in real time
  - SLAM should be robust to fast motions and tracking failures
  - Sunlight, uneven road conditions, missing cones

Background and Prior Work

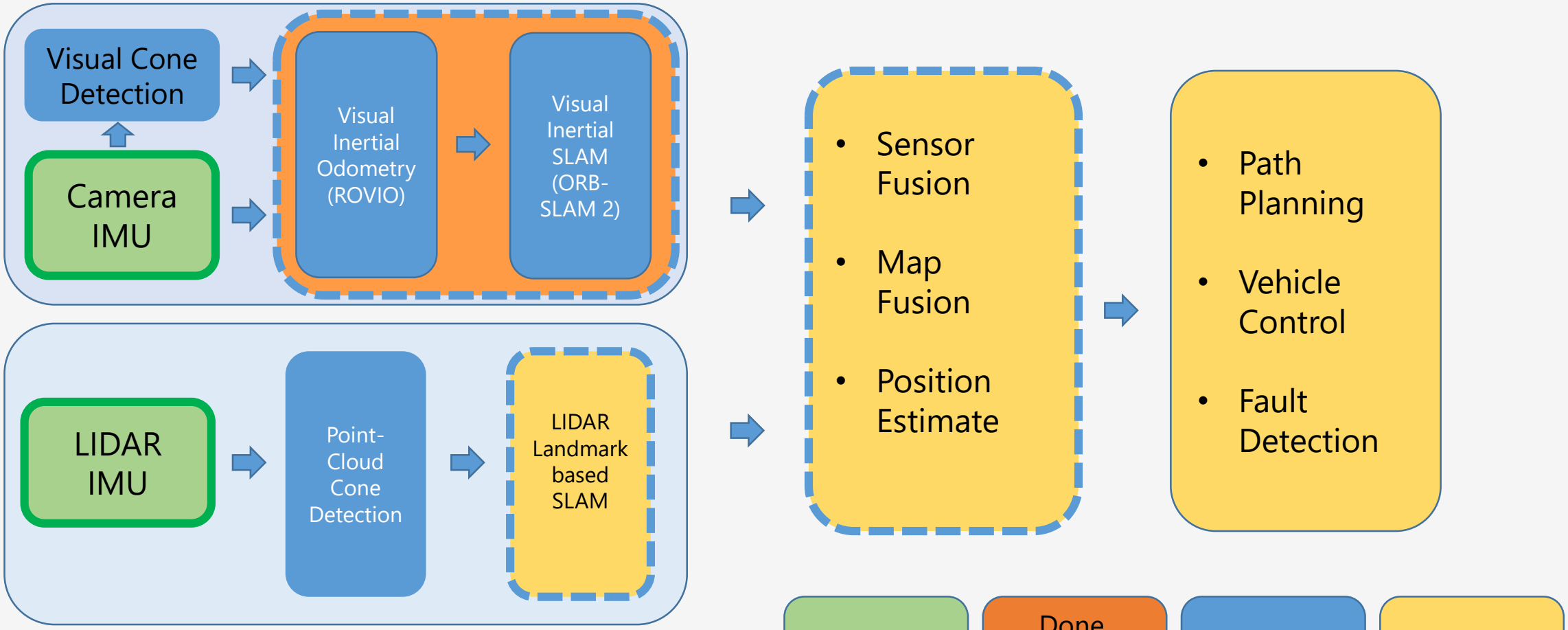
- AMZ Driverless
- Two SLAM pipelines:
  - visual SLAM (+cones)
  - LIDAR SLAM (cones only)
- Fused by Kalman Filter
- Effective path planning



AMZ Driverless pipeline

Vision and Image Sciences Laboratory

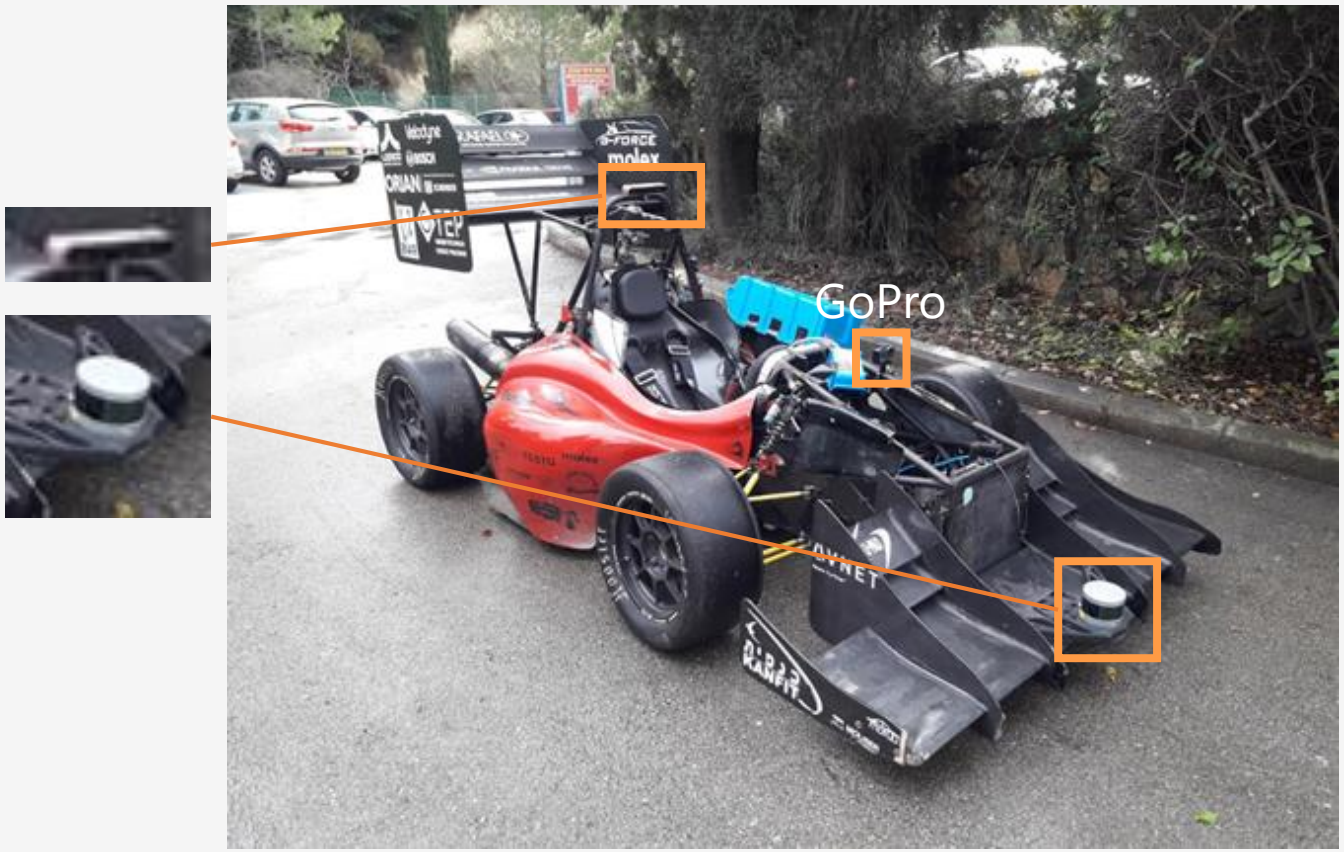
System Overview



VISL Vision and Image Sciences Laboratory

Our setup

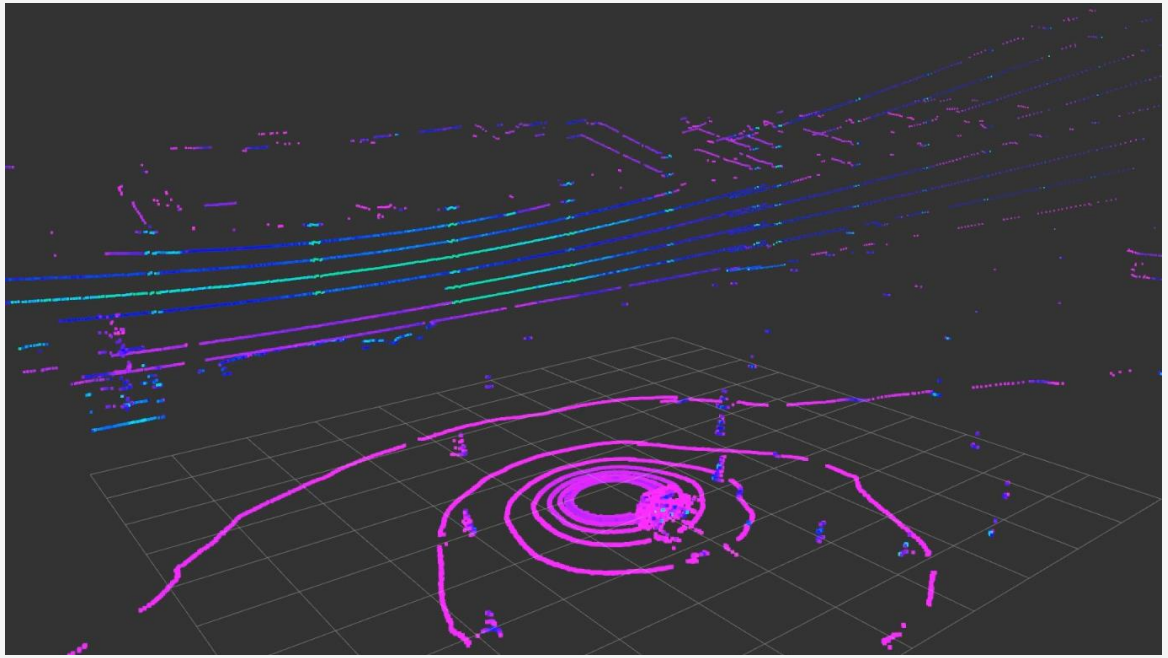
- ROS Kinetic and Ubuntu 16.06
- Our hardware:
  - ZED stereo camera
  - Velodyne VLP-16 LIDAR
  - Jetson TX-1 embedded PC
  - WIFI for remote control



racing car with the sensors

LIDAR perception and SLAM

- LIDAR provides great accuracy, robustness and range
- When placed correctly, it is especially useful for cone detection



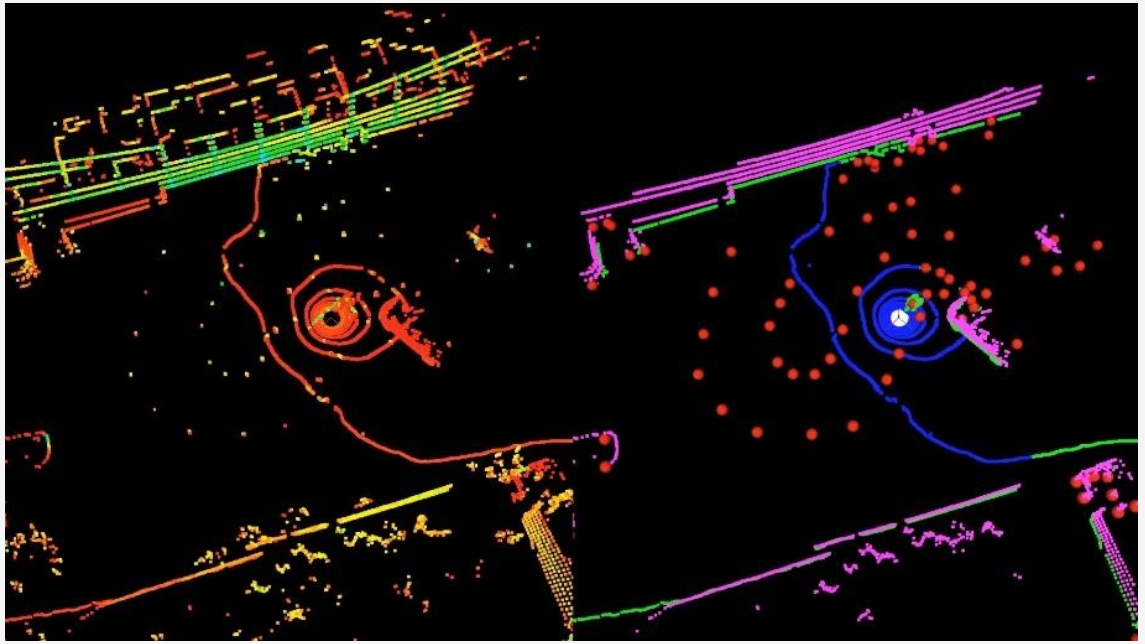
a LIDAR scan

LIDAR Cone Detection

- Steps of the algorithm
  1. Remove distant points
  2. Use RANSAC to detect the ground plane
  3. Filter all points based on distance to ground
  4. Apply Euclidean clustering on the points left
  5. Filter clusters by size and extract centroids
- Further filtering needed (spatial or temporal) to remove outliers

Implemented using PCL library and ROS

Input point cloud **Segmentation**



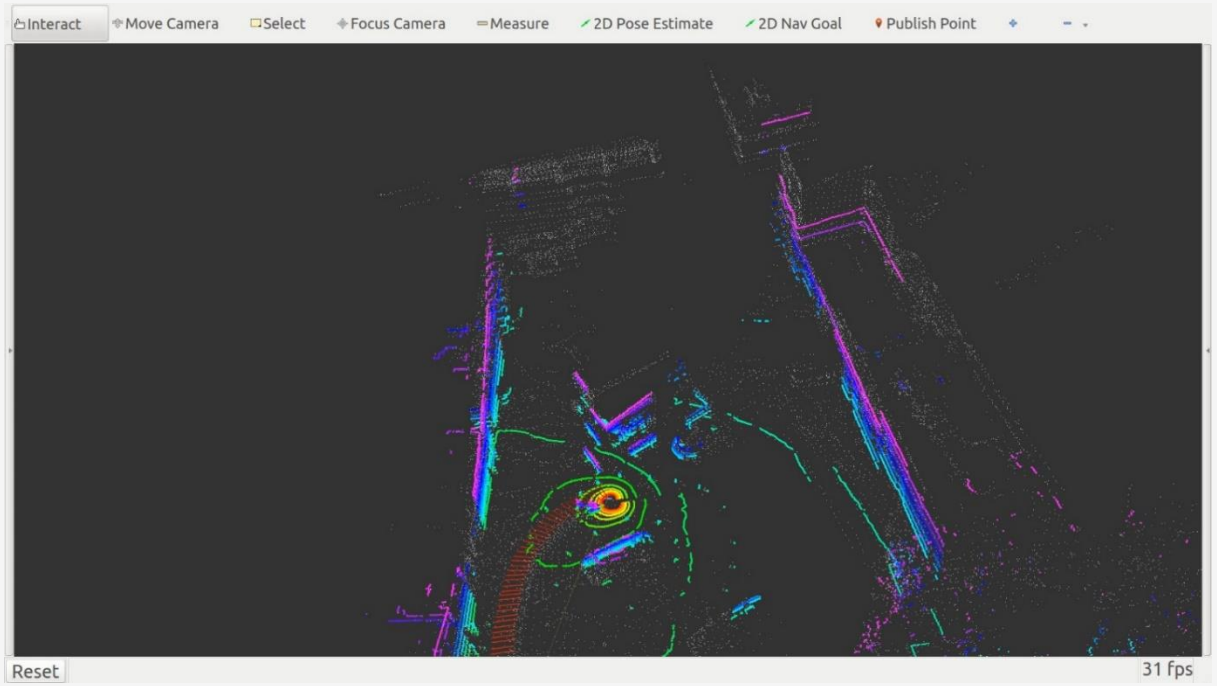
color represents LIDAR intensity

cone clusters, ground points, too high, everything else

VISL Vision and Image Sciences Laboratory

LIDAR SLAM

- We tested several algorithms:
  - LOAM, HDL Graph SLAM
    LeGO-LOAM, Cartographer
- None of them was good enough
- Probably due to:
  - High speed, uncompensated ego-motion, not enough features for alignment (ICP)



Frame from LOAM map: current frame, map, vehicle pose

Visual-inertial perception and SLAM

- Recently new CNN-based object detection algorithms were developed
- Camera based SLAM has also became widely used in Robotics

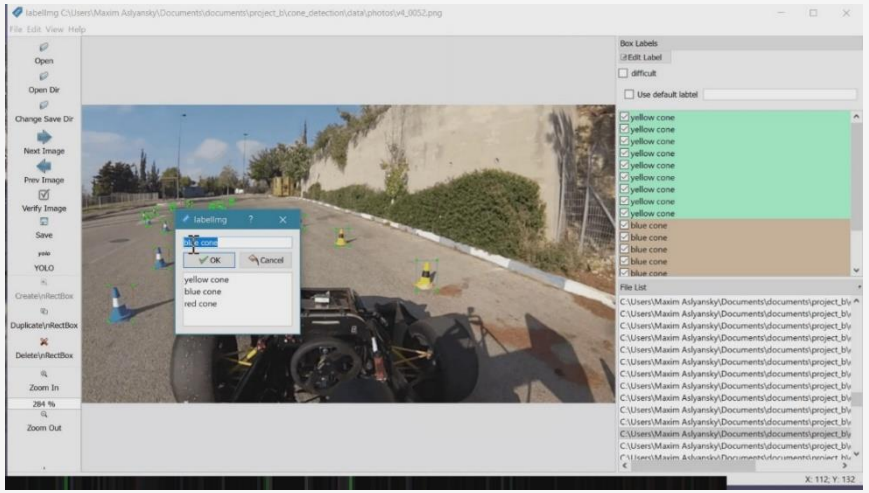
a frame from onboard camera

VISL

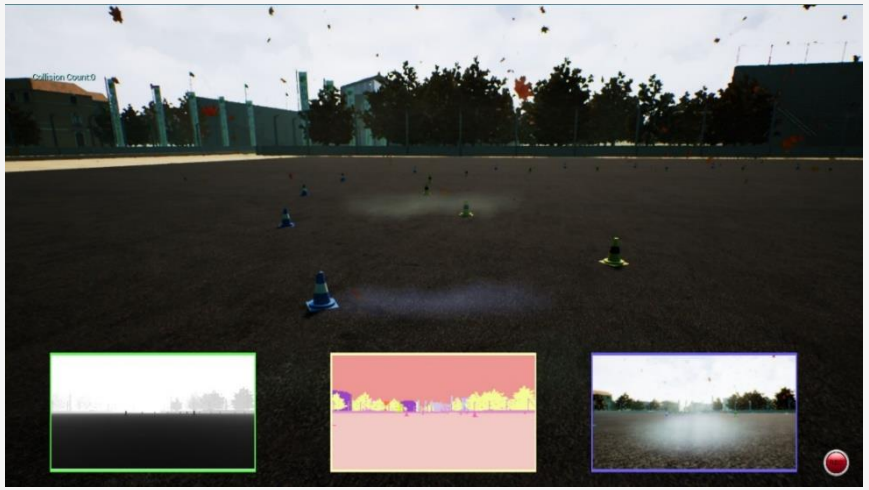
Vision and Image Sciences Laboratory

Dataset Visual Cone Detection

- Prepared and annotated dataset with ~400 images and ~8k cones
  - Focused on dense annotation in real-world scenario
  - Then also used pre-trained network
- Later started modifying AirSim simulator
  - Higher accuracy, weather, automatic
  - Ground-truth for depth prediction



LabelImg

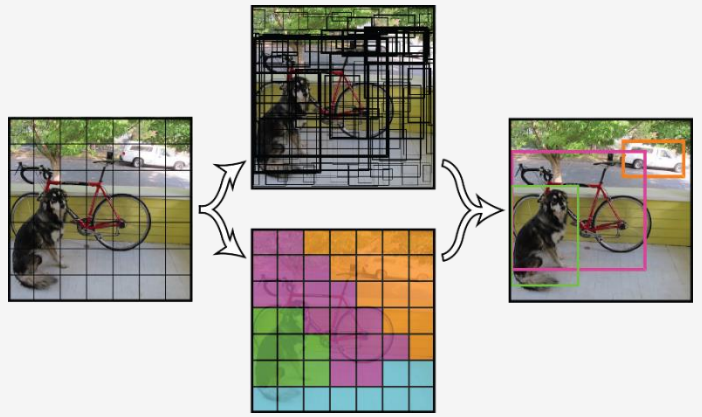


AirSim

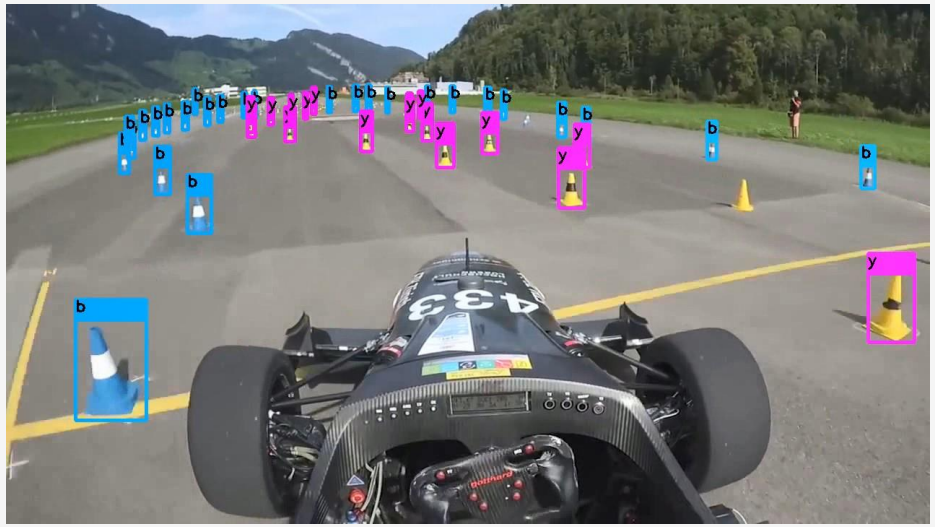
Vision and Image Sciences Laboratory

Visual Cone Detection

- YOLOv3 detector implemented in Darknet
  - Fastest while being accurate enough
- Optimized network architecture for accuracy
  - Added skip connections
  - Calibrated and added YOLO layers
  - Increased grid resolution
  - Trained on grayscale dataset



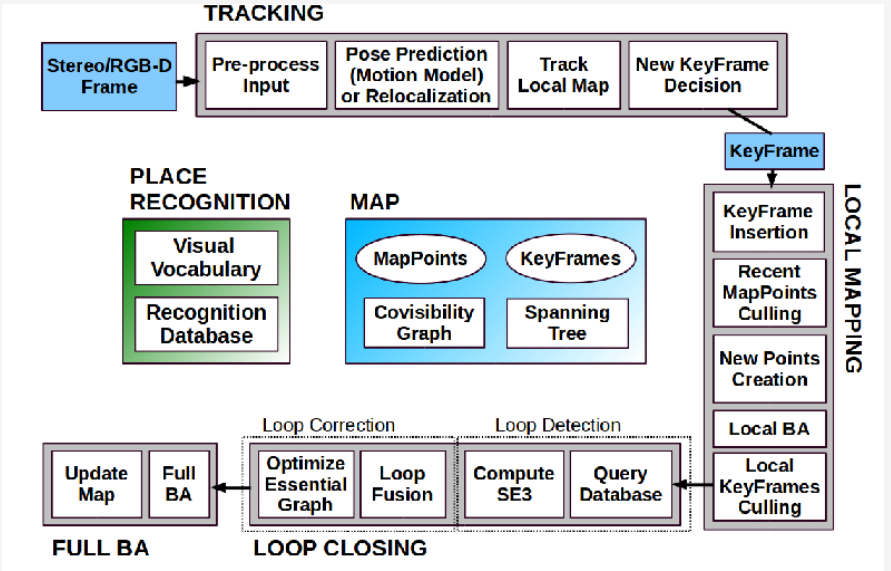
YOLO detector

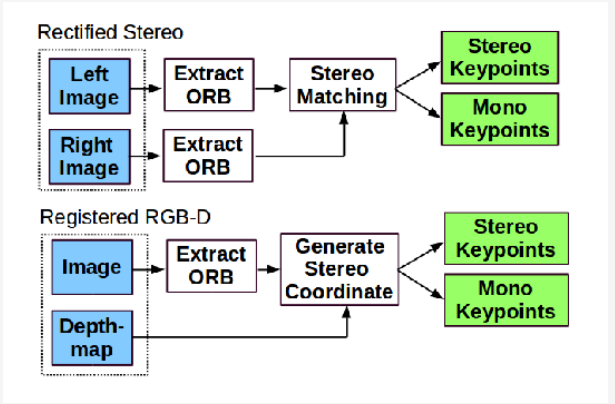


our results

Visual SLAM

- ORB-SLAM 2 is a very popular algorithm
  - Highly efficient and robust
  - Supports loop closure, localization mode
- We also tested several extensions
- SERVO is robust extension to ORB-SLAM 2 used by AMZ Driverless team
- Benefits from ROVIO (robust visual-inertial odometry) initialization





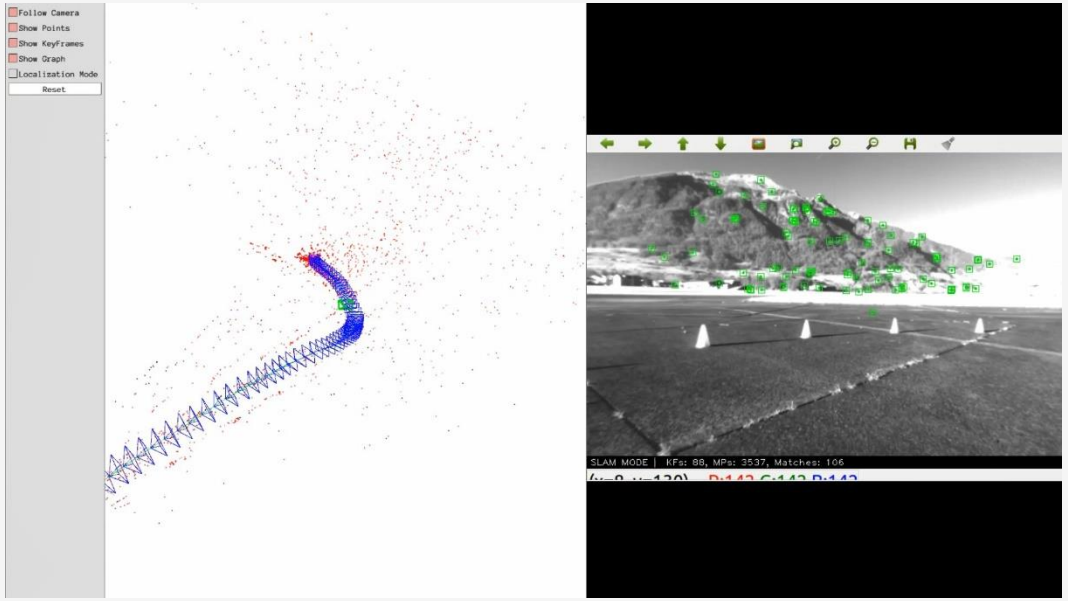
ORB-SLAM 2 algorithm

VISL

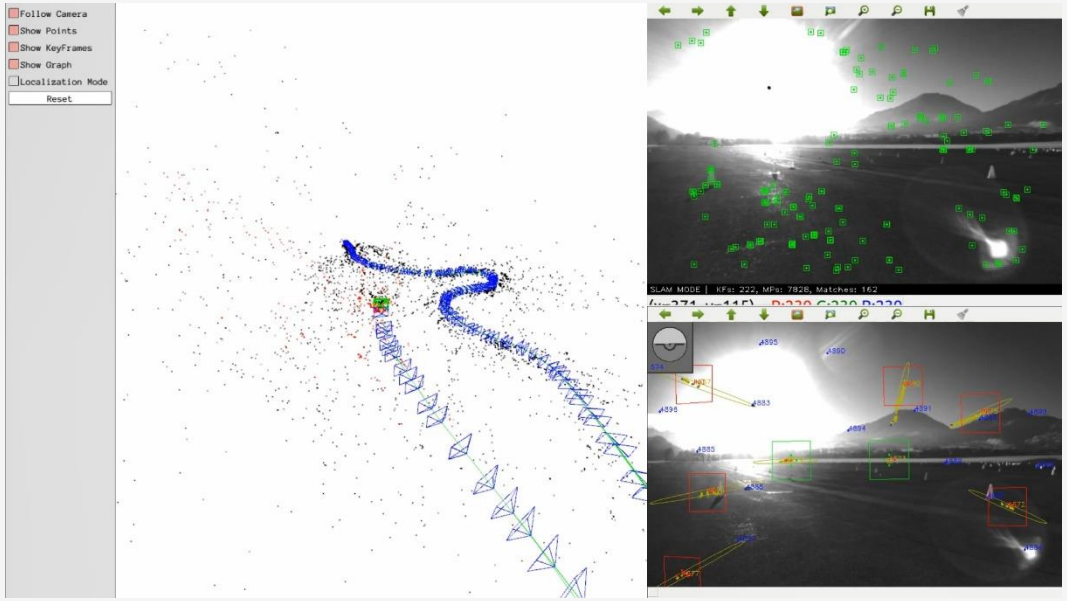
Vision and Image Sciences Laboratory

Visual SLAM - Results

- Comparison:



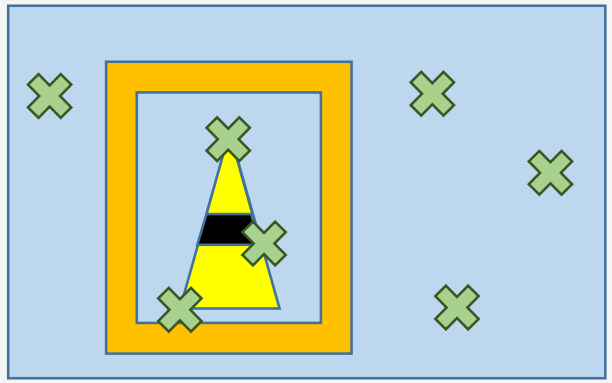
ORBSLAM 2



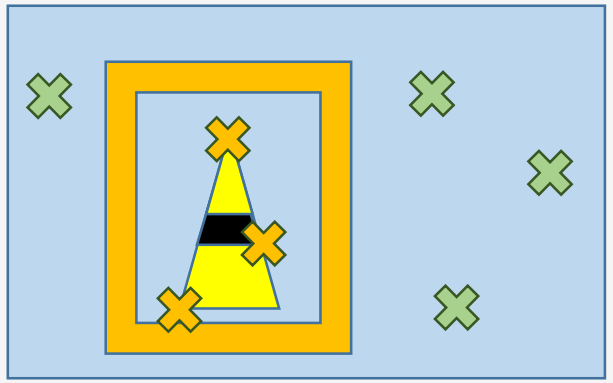
SERVO

Adding Cones to the Map

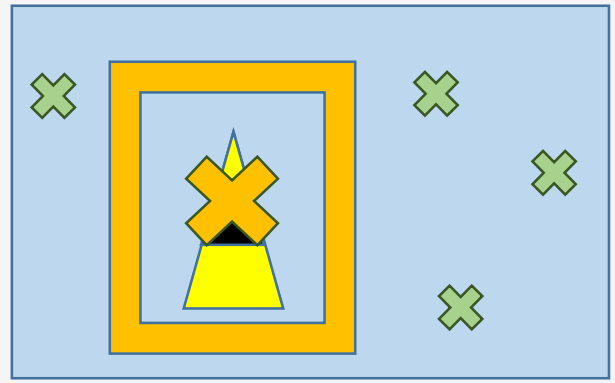
- Unlike AMZ we decided to integrate cones directly into ORBSLAM
- Option A – Semantic SLAM
  - Mark all features that overlap with cone bounding boxes
- Option B – cone feature
  - We've implemented this one



feature and object detections



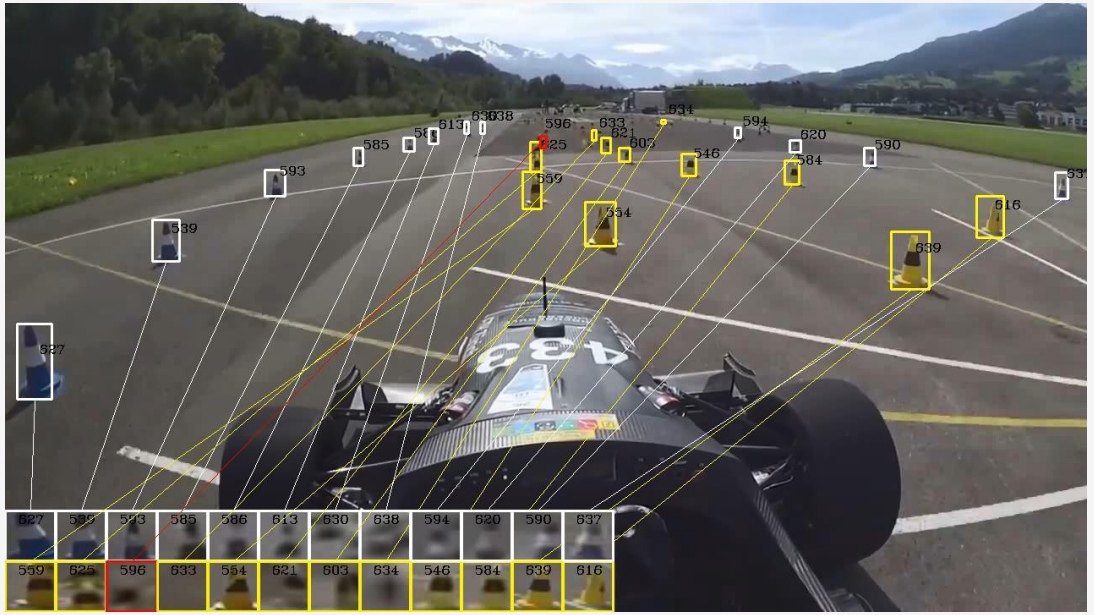
Semantic SLAM



Our approach

Cone Tracking

- Inspired by the success of ROVIO we decided to implement cone tracking
- Worked good in most cases but sometimes tracking was lost due to fast motion, especially near the edges

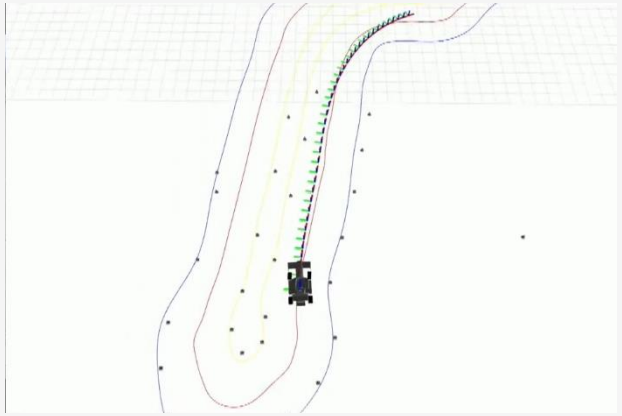


KLT object tracking

Conclusions and Future work

- We developed several crucial algorithms with emphasis on high accuracy and performance
- A lot of work to be done to achieve final working solution

AMZ demo



reconstructed track and path

VISL

Vision and Image Sciences Laboratory

Conclusions

- LIDAR SLAM
  - cone detection algorithm is fast and robust but needs some refinement
  - existing SLAM algorithms performed poorly, custom solution is needed
- Visual Cone Detection
  - our fine-tuned network achieved better accuracy compared to the competition (subjectively)
  - we believe the network can be further improved with synthetic data and extended with depth prediction
- Visual SLAM
  - SERVO was more robust on provided data but needs to be tested on our data
  - developed algorithm for cone tracking (which would benefit from depth from the network and IMU)

Future Work

- Data and Equipment Availability
  - LIDAR should be placed better (higher) and calibrated with camera
  - Need for a calibrated stereo rig with IMU
- Better Cone Detection Dataset and Network
  - Use AirSim: diverse and accurate data, depth prediction and modify the network accordingly
  - Add depth and IMU information to improve tracking (similarly to ROVIO)
- Integrating the Pipelines
  - Option A: Finish our work, Implement Kalman Filter and FastSLAM/g2o-based similarly to AMZ
  - Option B: Develop robust fusion of LIDAR and camera data and use single custom SLAM algorithm

References

1. Miguel I. Valls, Hubertus F.C. Hendrikx , Victor J.F. Reijgwart, Fabio V. Meier, Inkyu Sa, Renaud Dube, Abel Gawel, Mathias B ´ urki, and Roland Siegwart. “Design of an Autonomous Racecar: Perception, State Estimation and System Integration” International Conference on Robotics and Automation (2018)

2. Nikhil Bharadwaj Gosala, Andreas Bühler, Manish Prajapat, Claas Ehmke, Mehak Gupta, Ramya Sivanesan, Abel Gawel, Mark Pfeiffer, Mathias Bürki, Inkyu Sa, Renaud Dubé, Roland Siegwart. “Redundant Perception and State Estimation for Reliable Autonomous Racing”. International Conference on Robotics and Automation (2019)

3. Raul Mur-Artal and Juan D. Tardos. “ORB-SLAM2: an Open-Source SLAM System for Monocular, Stereo and RGB-D Cameras” IEEE Transactions on Robotics (2017)

4. Joseph Redmon, Ali Farhadi . “YOLOv3: An Incremental Improvement”. Computer Vision and Pattern Recognition (2018)

Thanks!

- Questions?