MCC-EKF for Autonomous Car Security

Authors: Ashutosh Singandhupe, Hung Manh La

Presented by: Jason Pratama (ECE M.Eng student)

MCC-EKF for Autonomous Car Security

- 2020 Fourth IEEE International Conference on Robotic Computing (IRC)
- Authors: Ashutosh Singandhupe (PhD Fall 2021), Hung Manh La (Associate Professor)
- Advanced Robotics and Automation (ARA) Lab, Department of Computer Science and Engineering, University of Nevada, Reno

General Overview

- Motivation: Prevent the autonomous system to change its estimated trajectory when an attack happens
- Introduce MCC-EKF as a defense against attacks or shot noise
- Sensors: LiDAR and stereo camera
- Simulated in Gazebo
- Benchmarked using the KITTI dataset

Problems

"Can we use the odometry information from two different Simultaneous Localization And Mapping (SLAM) algorithms to get a better estimate of the odometry?"

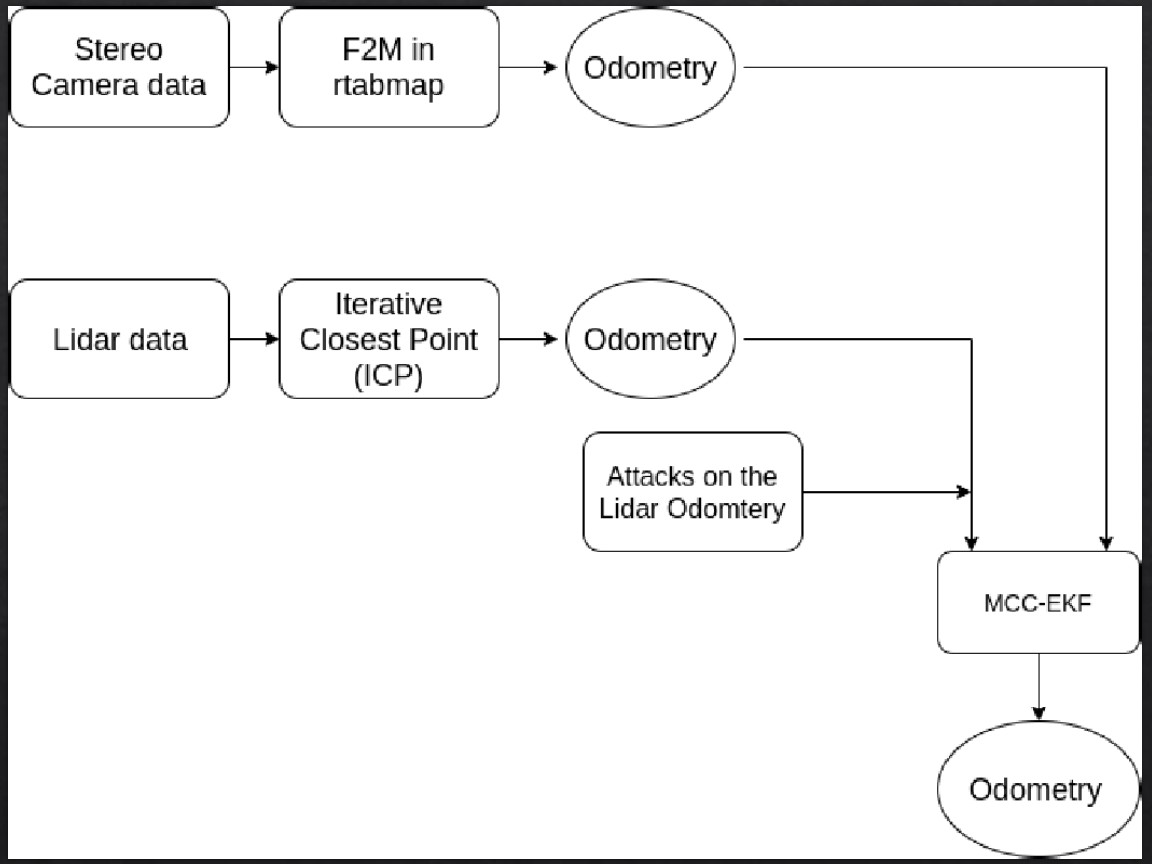
- "We focus on fusing odometries from Lidar-based SLAM and Visual based SLAM using the Extended Kalman Filter (EKF) algorithm"

"What if one of the SLAM algorithms gets affected by shot noise or by attack vectors, and can we resolve this situation?"

- "Introducing the Maximum Correntropy Criterion - Extended Kalman Filter (MCC-EKF), which assists in removing/minimizing shot noise or attack vectors injected into the system."

Paper Link: https://ara.cse.unr.edu/wp-content/uploads/2014/12/IEEE_IRC2020_Ashu.pdf

Concepts



System Architecture

*F2M =Frame-To-Map

Concepts

- MCC-EKF = Maximum Correntropy Criterion - Extended Kalman Filter
- Correntropy
  - X and y are 2 scalar variables
  - Kernel function: *κσ*

$$V_\sigma(X,Y) = E[\kappa_\sigma(X-Y)].$$

- Correntropy with Kalman Filtering
  - State vector: *xk*
  - Measurement vector: *yk*
  - System Noise: *wk*
  - Measurement Noise: *vk*
  - F and H are the system matrix and the observation matrix

$$x_k = F * x_{k-1} + w_k$$

$$y_k = H * x_k + v_k$$

Concepts

For initialization:

$$\hat{x_0} = E[x_0] \tag{13}$$

$$P_0 = E[e_0 e_0^T] \tag{14}$$

where, $e_0 = x_0 - \hat{x}_0$ is the error term, and $E(.)$ is the expected value operation. Prior estimation can be calculated as

$$\hat{x}_k^- = F\hat{x}_{k-1} \tag{15}$$

$$P_{k|k-1} = F P_{k-1|k-1} F^T + Q_k. \tag{16}$$

$$K_k = (P_{k|k-1}^{-1} + L_k * H^T * R_k^{-1} * H)^{-1} * L_K * H^T * R_K^{-1} \tag{18}$$

$$\hat{x}_k = \hat{x}_k^- + K_k(y_k - H * \hat{x}_k^-) \tag{19}$$

$$P_{k|k} = (I - K_k * H) * P_{k|k-1} * (I - K_k * H)^T + K_k * R_k * K_k^T. \tag{20}$$

$$L_k = \frac{G_\sigma(||y_k' - H * \hat{x}_k||_{R_k^{-1}})}{G_\sigma(||\hat{x}_k - F * \hat{x}_{k-1}||_{P_{k|k-1}^{-1}})}. \tag{24}$$

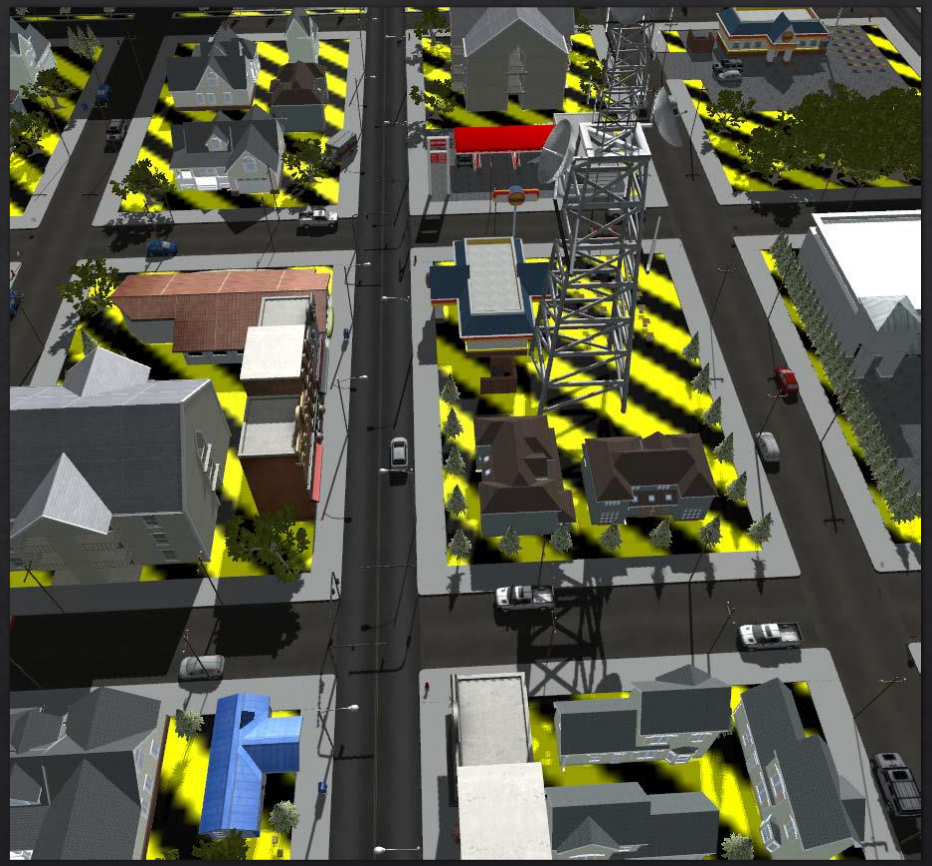
- “So when yk is large as defined by the kernel bandwidth, Lk is 0, which forces the Kalman Gain K to be 0. So the next updated state is updated by the system state as given in Equ. (15). This process, as one can see, can reject the attacks/outliers thus securing the system”

Concepts

Result: Computed odometry from MCC-EKF SLAM
Lidar Odom (ICP) $\rightarrow \mathrm{L}_o(x, y, z, r, p, y)$
Stereo Odom (F2M) $\rightarrow \mathrm{S}_o(x, y, z, r, p, y)$
Initialization;
Lidar odom $\rightarrow (\mathrm{L}_o)$
Stereo odom $\rightarrow (\mathrm{S}_o)$
Compute x_0 from Equ. (13)
Compute P_0 from Equ. (14)
Prior Estimation;
Compute $\hat{x}_k^-$ from Equ. (15)
Compute $P_{k|k-1}$ from Equ. (16)
while *get* L_o *and* S_o*;*
do
  Compute L_k from Equ. (24)
  Compute Gain K_k from Equ. (18)
  Update state x_k from Equ. (19)
  Update $P_{k|k}$ from Equ. (20)
end

MCC-EKF Algorithm

Testing Environment



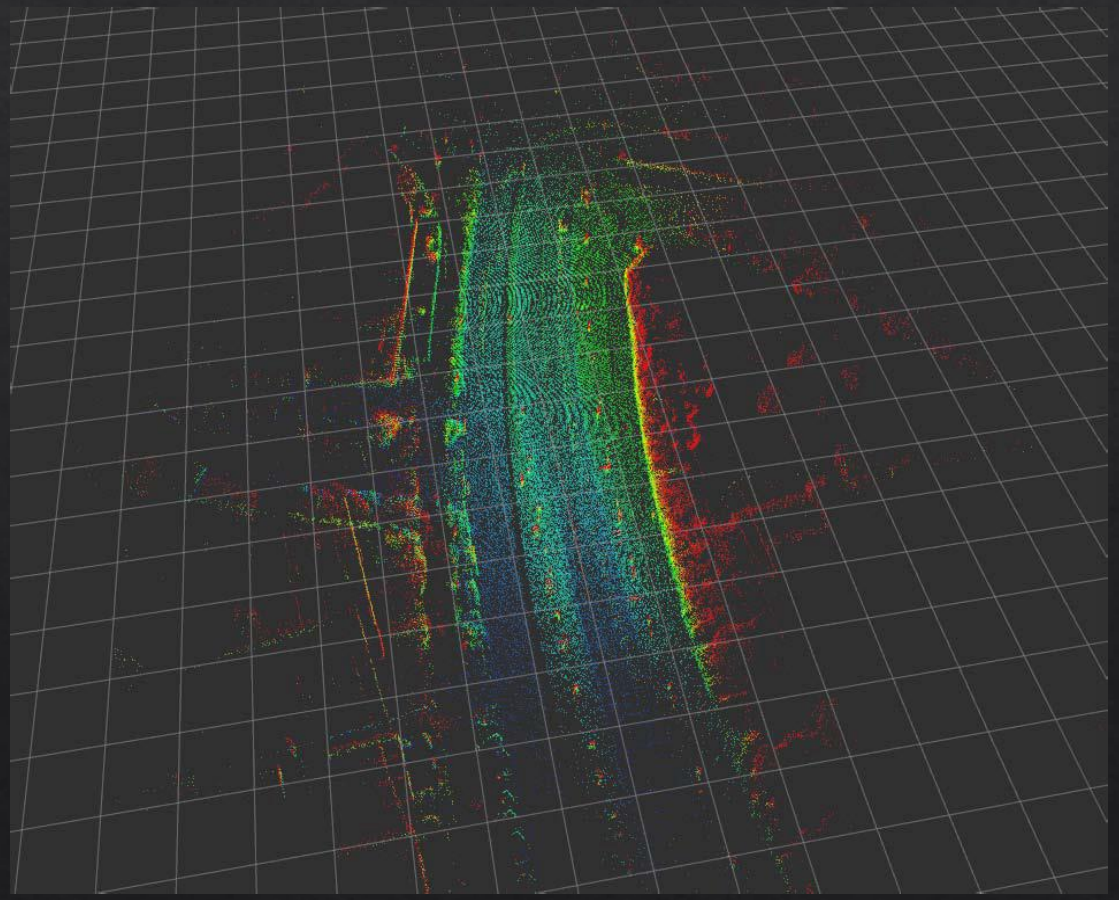
Gazebo Simulation Environment

Testing Environment

KITTI dataset data (SEQ 11) Environment

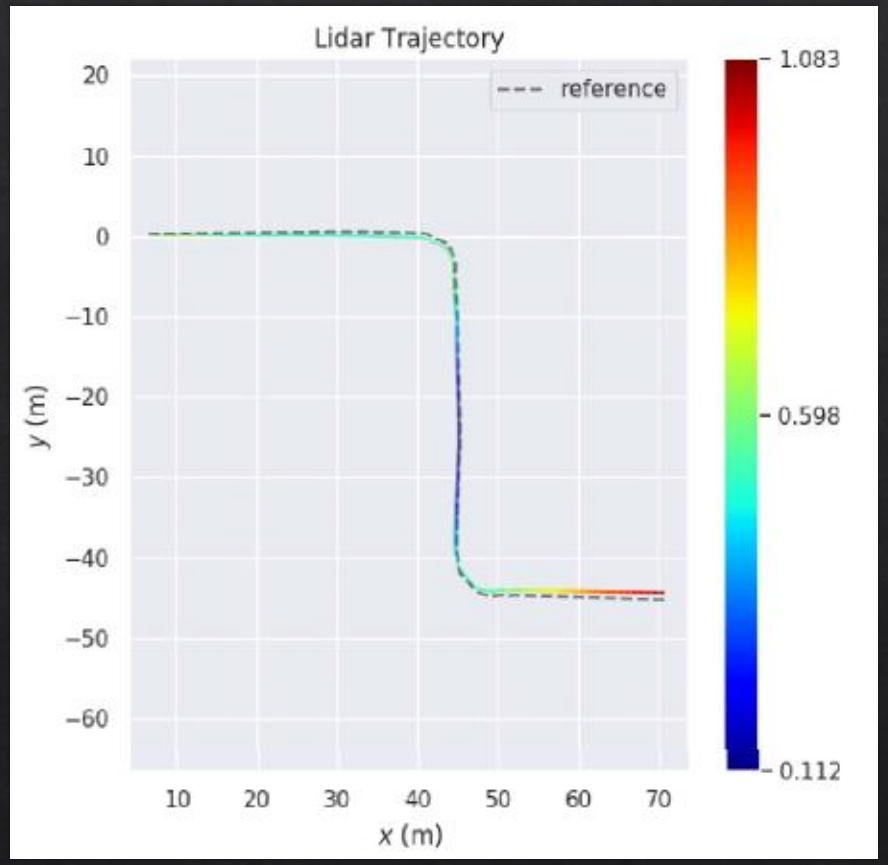
KITTI Dataset Project:

- Karlsruhe Institute of Technology (KIT) and Toyota Technological Institute at Chicago (TTI-C)
- Datasets are captured by driving around the mid-size city of Karlsruhe
- http://www.cvlibs.net/datasets/kitti/

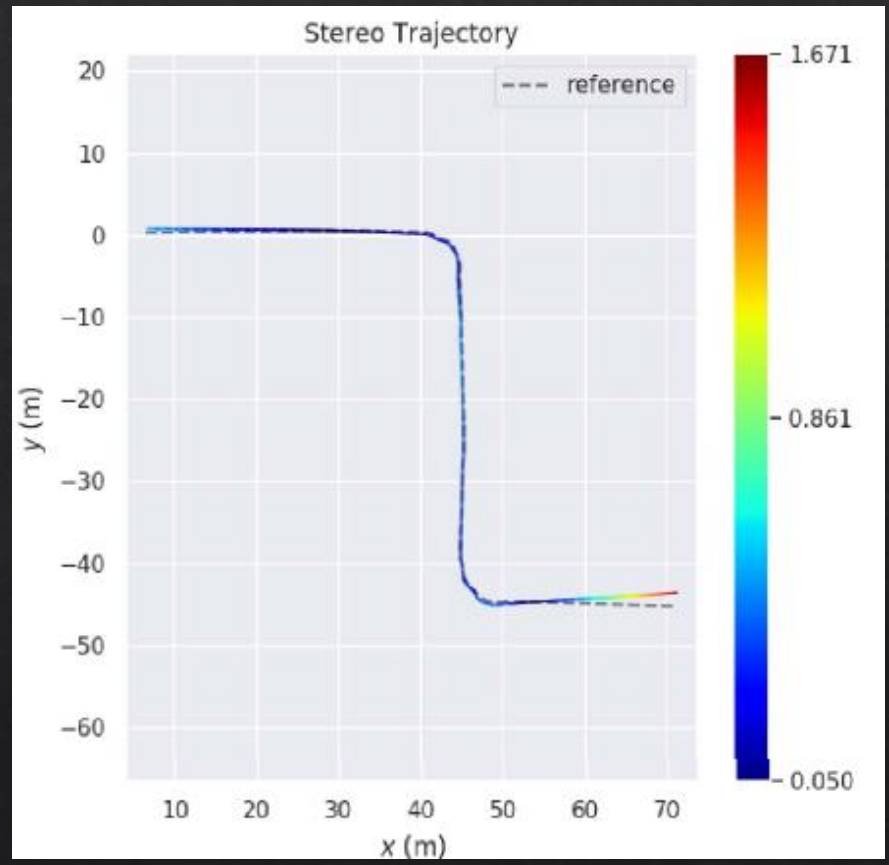


KITTI dataset data (SEQ 11) 3D Lidar Map

Results

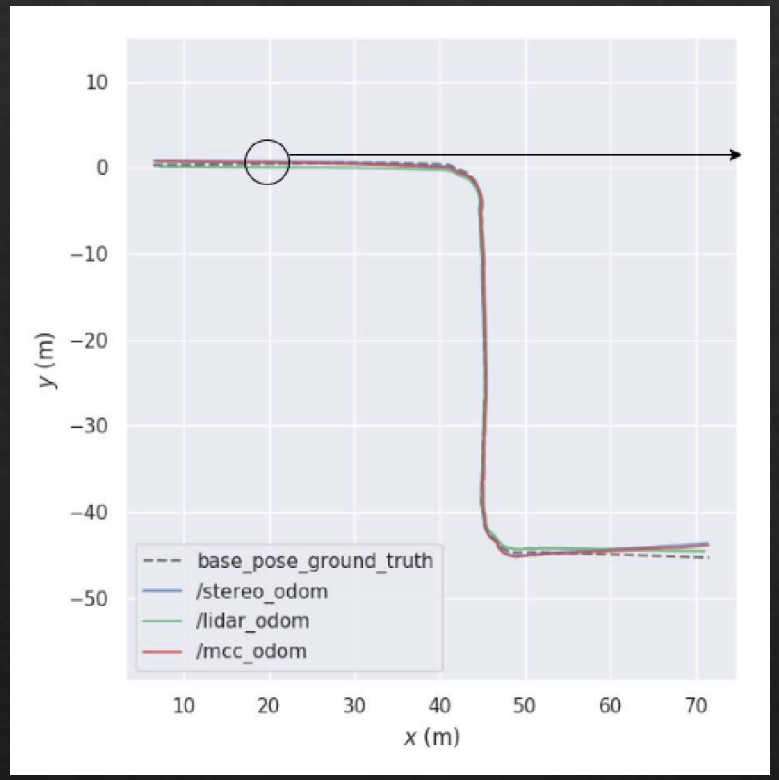


Lidar odometry

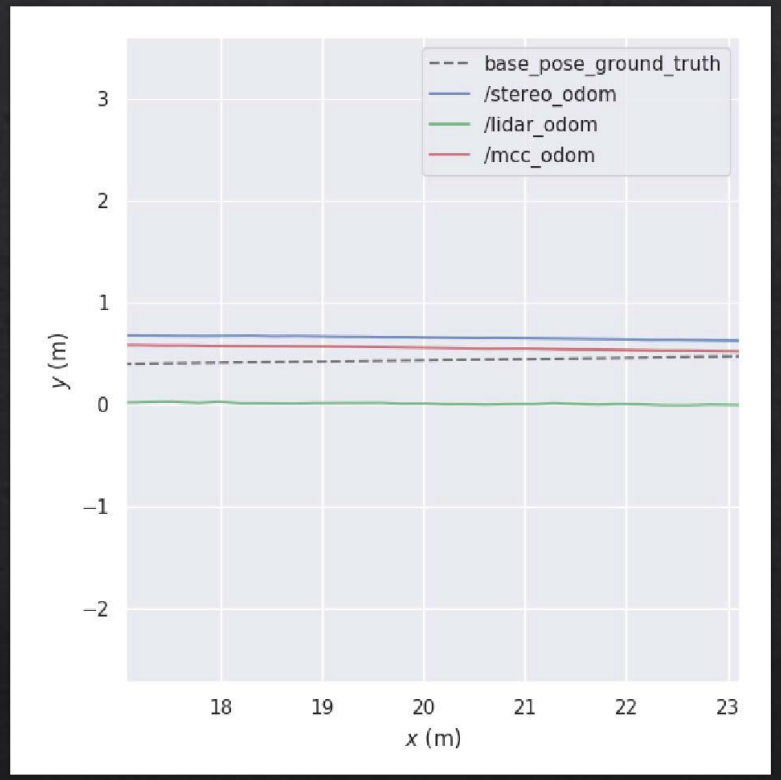


Stereo odometry

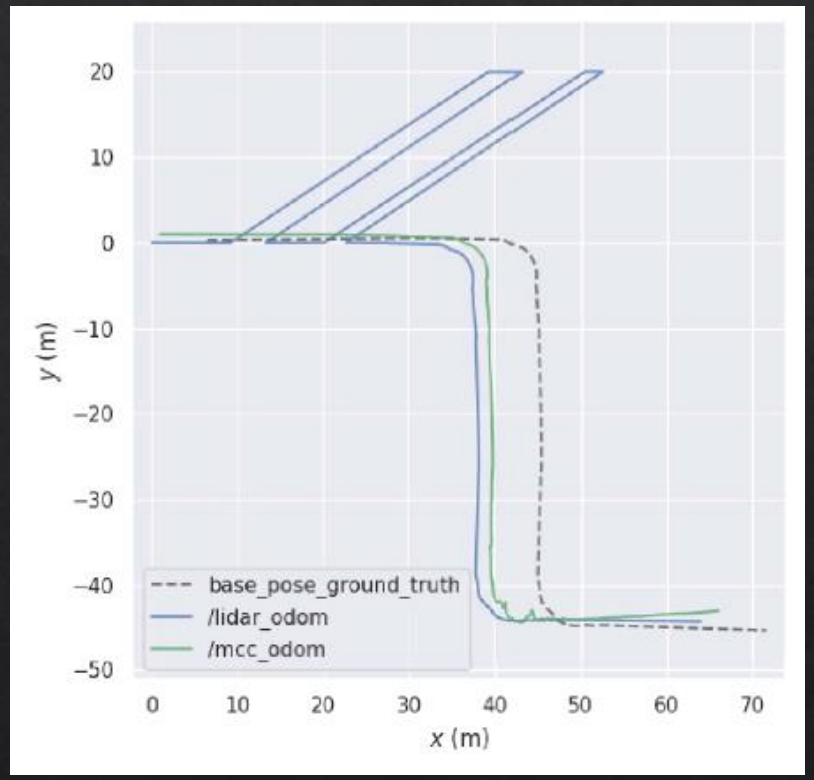
Results



Odometry



Odometry (Zoom-in)



MCC-EKF response to attacks on Lidar data

Results

	Traj 1	Traj 2
Stereo (RMSE)1	0.478404	1.840472
Lidar (RMSE)	0.557076	0.505731
MCC-EKF (RMSE)	0.419823	0.49761

RSME Comparison

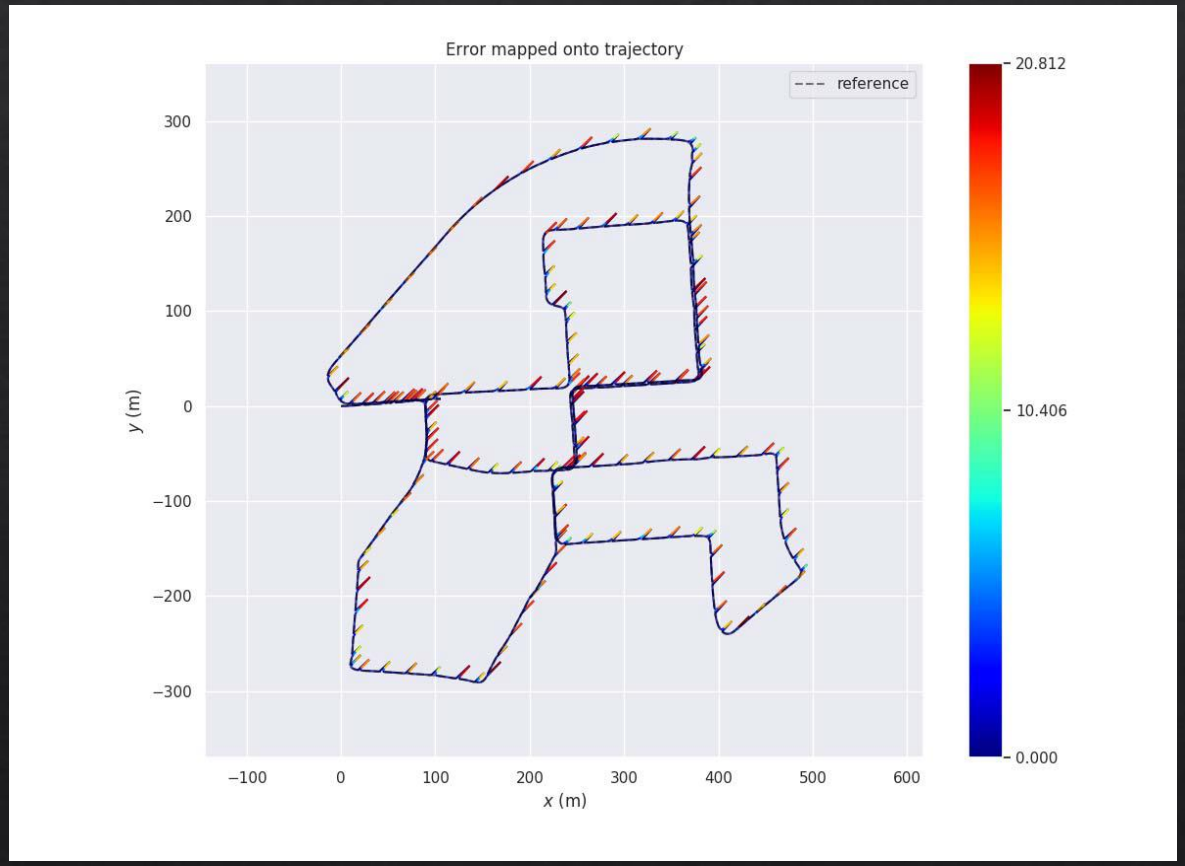
	Traj 1	Traj 2
Random attacks 1		
Normal EKF (RMSE)	3.624596	5.01472
MCC-EKF (RMSE)	0.420437	0.81037
Random attacks 2		
Normal EKF (RMSE)	4.624596	7.01472
MCC-EKF (RMSE)	0.43317	0.85132

RSME Comparison (after attack)

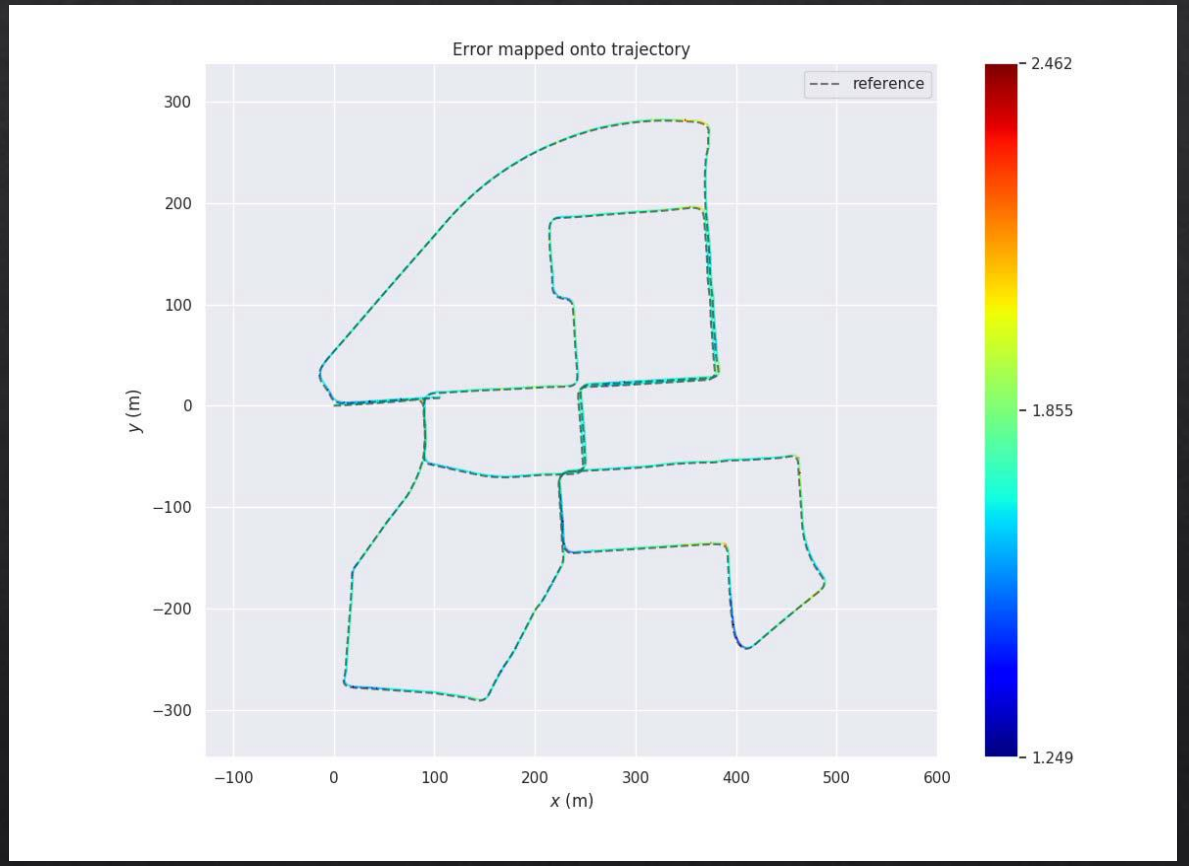
	SEQ 01	SEQ 05	SEQ 11	SEQ 27
Random attacks 1 EKF	2.04	2.42	1.475	3.193625
MCC-EKF	0.0198	0.0073	0.110	0.142083
Random attacks 2 EKF	3.79	7.0685	2.1875	4.1989
MCC-EKF	0.0198	0.092263	0.1842	0.1693

RSME Comparison on KITTI Dataset (after attack)

Results

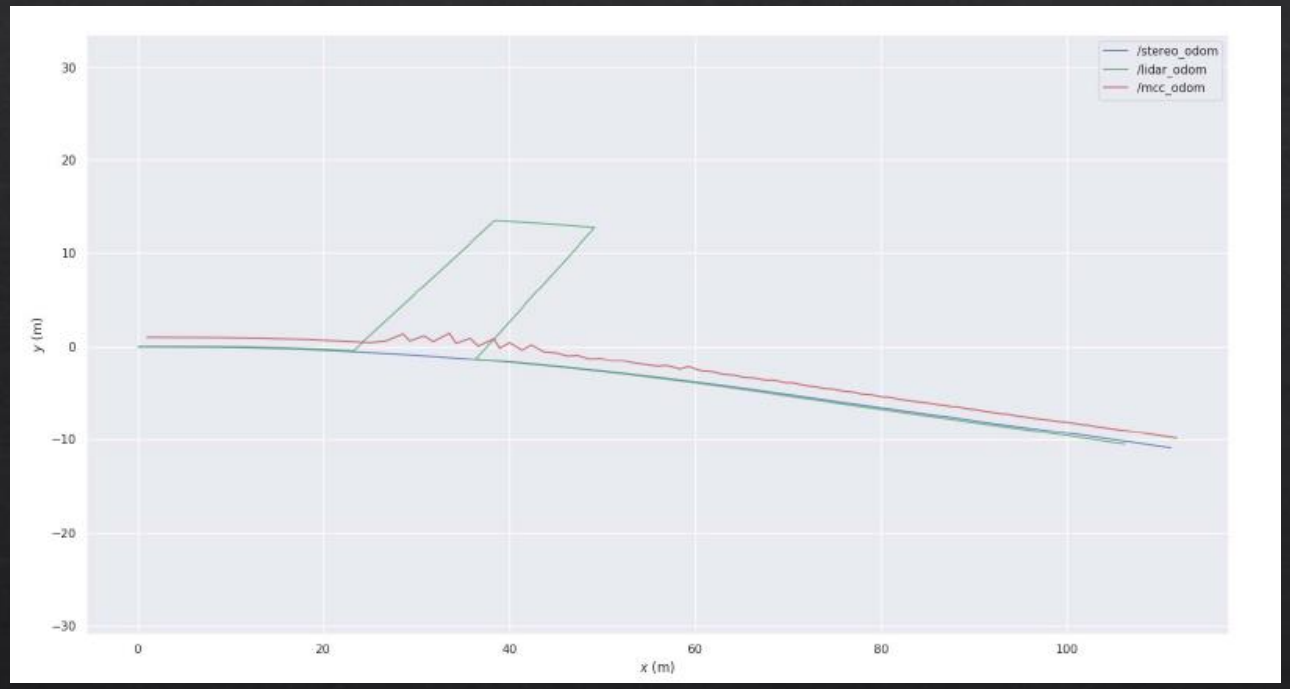


Normal EKF on KITTI dataset SEQ 27

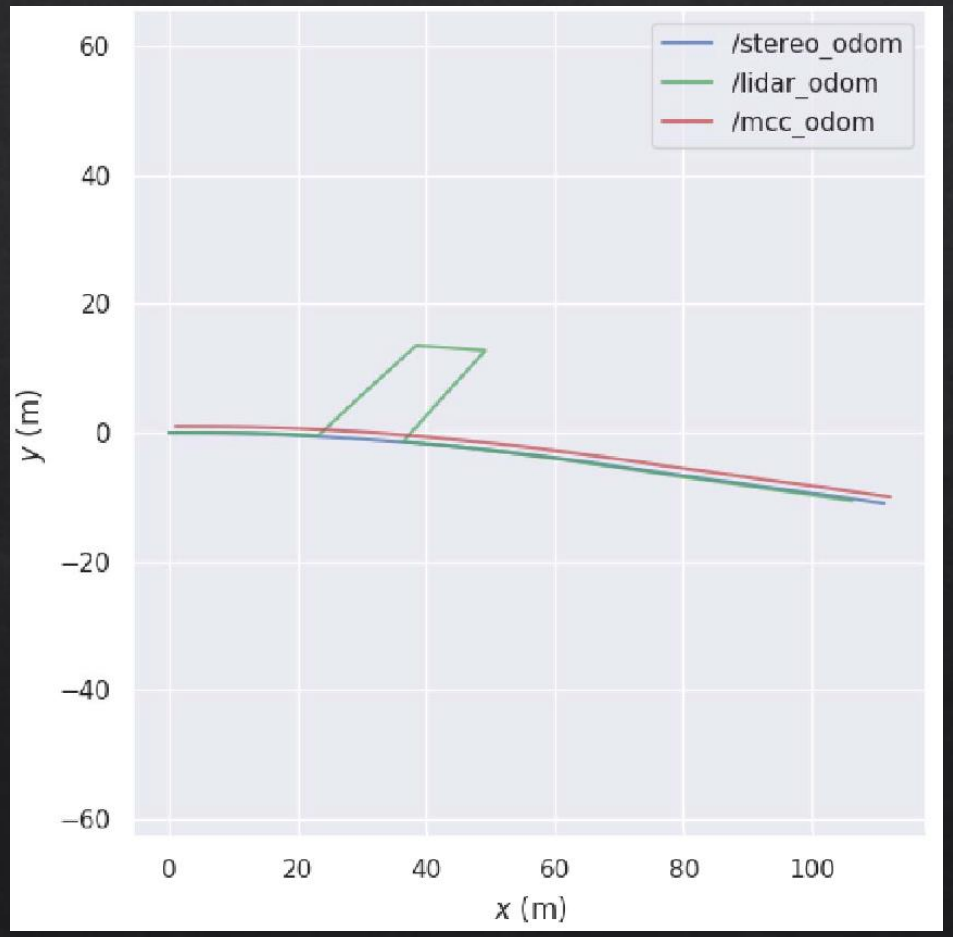


MCC-EKF on KITTI dataset SEQ 27

Results

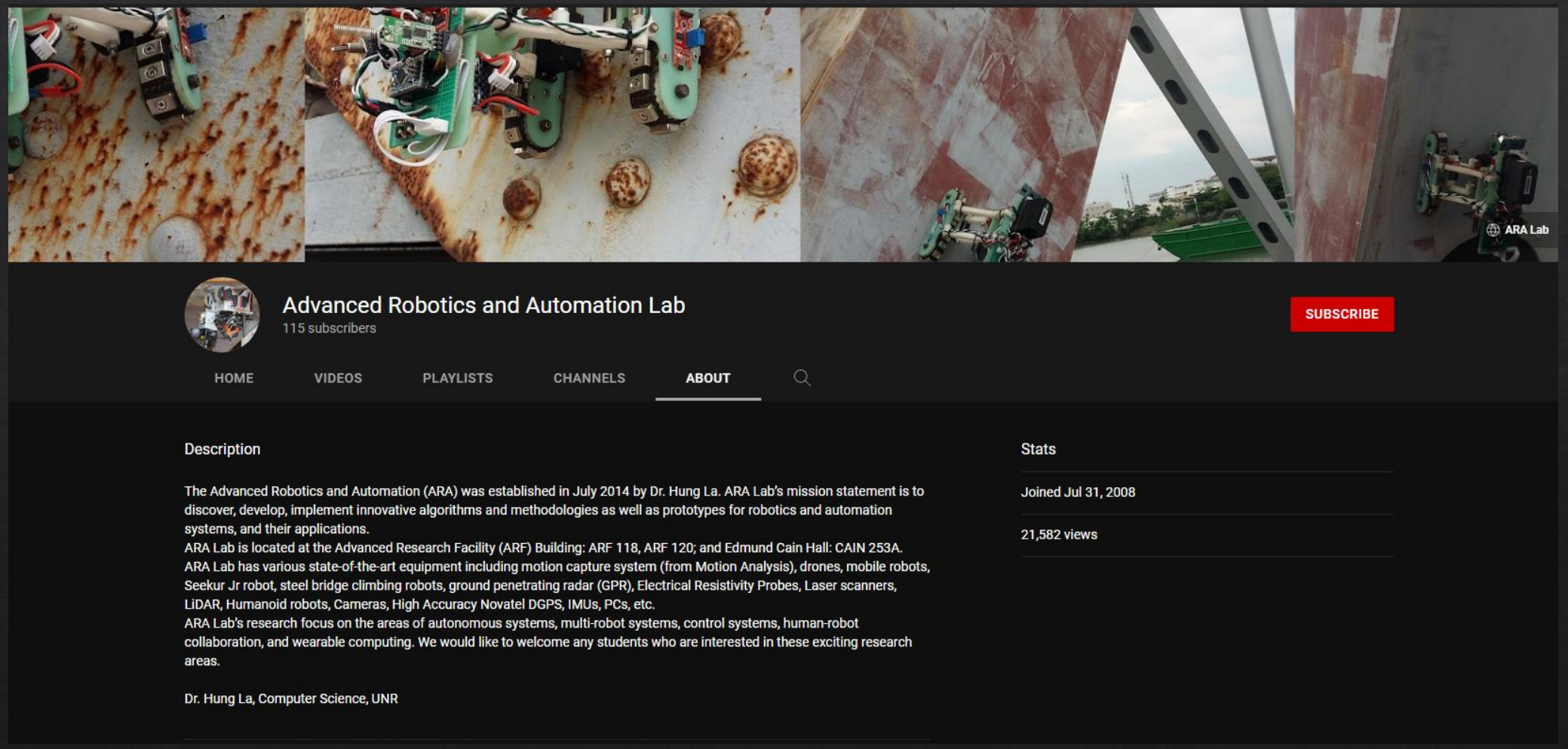


EKF response on KITTI dataset SEQ 01 (with attack)

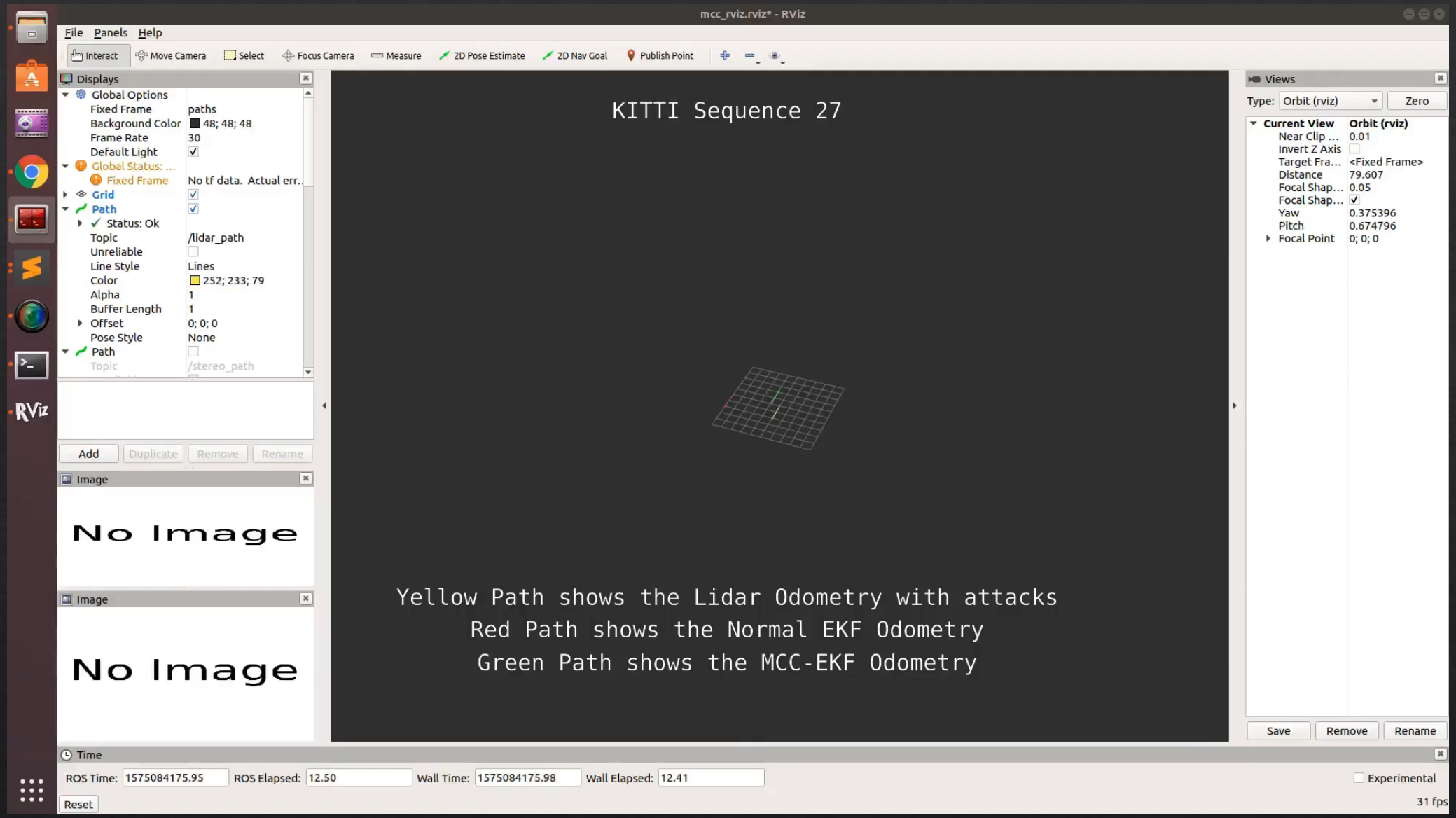


MCC-EKF response on KITTI dataset SEQ 01 (with attack)

Video



YouTube Page: https://www.youtube.com/user/hungmanhla



Original YouTube Video: https://www.youtube.com/watch?v=bKzN8NdQnaE

Related Publications

- Sehgal, A. *et al.* (2019) 'Lidar-Monocular Visual Odometry with Genetic Algorithm for Parameter Optimization', *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 11845 LNCS, pp. 358–370. doi: 10.1007/978-3-030-33723-0_29.
- Singandhupe, A. and La, H. (2019) 'A Review of SLAM Techniques and Security in Autonomous Driving', *Proceedings - 3rd IEEE International Conference on Robotic Computing, IRC 2019*, (19), pp. 602–607. doi: 10.1109/IRC.2019.00122.

Discussion

- Is the upgrade from normal EKF to MCC-EKF worth it?
- Would it still be an improvement (from EKF to MCC-EKF) if the GPS was used in the KITTI benchmark?
- Is there a better data fusion method for this scenario?

Links

- Paper Link: https://ara.cse.unr.edu/wp-content/uploads/2014/12/IEEE_IRC2020_Ashu.pdf
- YouTube Video: https://www.youtube.com/watch?v=bKzN8NdQnaE
- ARA Lab Website: https://ara.cse.unr.edu/
- GitHub: https://github.com/aralab-unr/MCC-EKF-SLAM
- Gazebo: http://gazebosim.org/
- KITTI database: http://www.cvlibs.net/datasets/kitti/