A Detection and Filtering Framework for Collaborative Localization

Thirumalaesh Ashokkumar
University of Michigan
Ann Arbor, Michigan

Katherine A. Skinner
University of Michigan
Ann Arbor, Michigan

Siddharth Agarwal
IEEE Senior Member

Ankit Vora
IEEE Member

Ashutosh Bhown
University of Michigan
Ann Arbor, Michigan

***Abstract*—Increasingly, autonomous vehicles (AVs) are becoming a reality, such as the Advanced Driver Assistance Systems (ADAS) in vehicles that assist drivers in driving and parking functions with vehicles today. The localization problem for AVs relies primarily on multiple sensors, including cameras, LiDARs, and radars. Manufacturing, installing, calibrating, and maintaining these sensors can be very expensive, thereby increasing the overall cost of AVs. This research explores the means to improve localization on vehicles belonging to the ADAS category in a platooning context, where an ADAS vehicle follows a lead 'Smart' AV equipped with a highly accurate sensor suite. We propose and produce results by using a filtering framework to combine pose information derived from vision and odometry to improve the localization of the ADAS vehicle that follows the smart vehicle.**

***Index Terms*—Autonomous Vehicles, Multi-Agent Systems, Localization, Mapping, Visual Odometry, Sensors, Extended Kalman Filter, Perception**

I. Introduction

Autonomous Vehicles (AVs) are a reality waiting to happen, with broad applications in logistics, travel, and service industries. Moreover, the ability of AVs to collaborate, forming multi-agent networks would boost the productivity and efficiency of tasks performed by these vehicles. While the development of standalone autonomous vehicles has seen great strides, the development of multi-agent systems requires highly accurate mapping and localization of all agents.

Due to cost and complexity, it is not always possible for all agents of a multi-agent autonomous vehicles network to possess equivalent sensing suites in terms of accuracy and precision. It would be ideal to exploit the sensing suites available on one vehicle with high-end sensors to improve the localization of other vehicles with a low-cost sensor suite. This premise is the focus of this work, where we will describe a mechanism of detecting and improving localization on a vehicle with a lower-grade sensor suite, by using the state estimations derived from a vehicle with a higher-grade sensor suite and relating these to the follower vehicle.

The specific focus of this work is a two-vehicle setup. The first is a lead vehicle having a robust and highly accurate sensor suite. We will refer to this vehicle as the smart vehicle. Improving the localization of the ADAS vehicle with diminished sensing capabilities is our primary goal. We propose a fusion framework to fuse the pose information acquired from the smart vehicle along with the odometry of the ADAS vehicle to improve its localization. We test our setup on the Ford Multi-AV Seasonal dataset [1]. The dataset contains odometry and other sensor data from multiple vehicles driving through the Michigan-Detroit area. The illustration of our problem statement is shown in Fig 1.

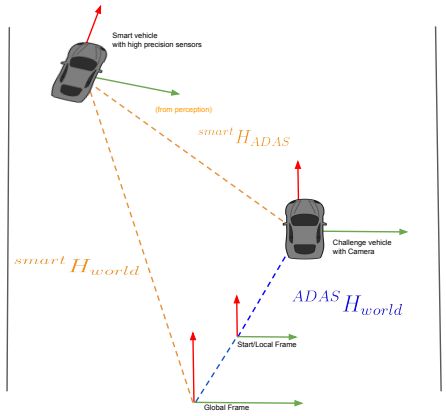

Fig. 1. Improving localization of an 'ADAS' vehicle by fusing pose from inertial odometry($^{ADAS}H_{world}$) and pose information derived by perceiving the smart vehicle($^{ADAS}H_{world}$). Fusion is done through an EKF that gives the improved pose estimate $^{ADAS}\hat{P}_{world}$. $^{A}H_{B}$ is the pose of A in frame B.

The main contributions of this work are:

- Development of a filtering framework for collaborative multi-agent localization
- Testing of the framework on a real-world dataset - Ford Multi-AV Seasonal dataset

II. Related Work

While the solution to the localization problem on standalone vehicles can be viewed as part of the SLAM (Simultaneous Localization and Mapping) problem, the research into collaborative multi-agent localization takes varied approaches. A feature detection and two-stage filtering mechanism has been described in [2]. Another approach to solving this problem is to treat the multiple agents as parts of the same entity, which has been detailed by authors in [3]. An approach using particle filters, where the particles are shared between the agents has been proposed by [4]. [5] is a detailed survey that points

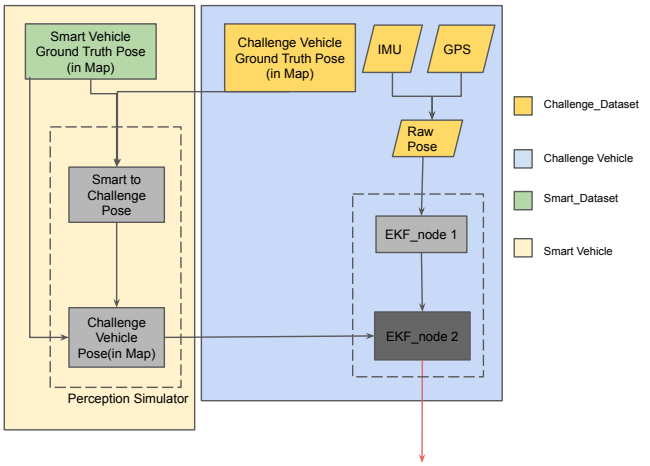

Fig. 2. The proposed Framework: Perception module inputs a pose information to the filter, odometry information is also input to the network to obtain an improved pose estimate

out the various methods and problems related to Multi-Agent Localization.

The literature on localization approaches can be classified into three broad categories: algorithms based on the Extended Kalman Filter, the Particle Filter, and Graph-based approaches. Solutions using a camera as the primary source of data are also being developed, such as solutions to the Visual SLAM problem. In this work, our goal is to establish the advantages of using a perception system on an ADAS vehicle with a lower-grade sensor suite to detect the smart vehicle with a higher-grade sensor suite, thereby improving the localization of the ADAS vehicle. Thus, for localization, we focus on using the Extended Kalman Filter and we make the assumption that there is a communications link between the two vehicles.

III. Technical Approach

Consider the three important frames in Fig. 1. The Global frame(H_{world}) or Map frame is the frame from which all other frames are defined. This is usually dependent on the map being used. The map is considered to be known for the purposes of this work, and thus raw poses of both vehicles are defined in this frame. The Local or Start frame is the frame from which we begin the experiments. The measurements of the IMU are defined with respect to this frame. The Body frame is the frame of the vehicle in each instance.

For our experiments, since the map is as available from the dataset, we know the start position of the robot, implying we know the transformation between the Global frame and the Local frame, which is simply the start position of the vehicle. To perform filtering and get pose estimates in the Global frame, we need transforms from the Local frame to the Body frame at every instance of fusion, along with the transformation from the Global frame to the Local frame. Only then can the Global $\rightarrow$ Local $\rightarrow$ Body transform be achieved, through which the measurements can be fused.

To achieve the Local to Body transform at each instance, we use EKF_node 1 as shown in Fig. 2, which fuses only the raw poses. Using the generated poses and a static transform to describe the Global to Local transform, we are able to use EKF_node 2, to perform sensor fusion on the raw poses and the measurements from the perception module. Particularly, node 2 uses the poses of the ADAS vehicle(T_O^C), and output of the perception module(T_W^C) as input, fusing them based on the Extended Kalman Filter algorithm, to generate state estimates. We note that T_O^C is fused relatively, while T_W^C are fused as absolute measurements of the pose.

A. Localization

For localization and filtering, we use the filtering setup provided by the robot_localization [6] ROS package. We set up two nodes of the Extended Kalman Filter for the reasons stated earlier. The first node, filters the pose odometry of the ADAS vehicle, to produce the transform between the start frame and the body frame. Using this, measurements of raw poses and perception are fused in the second node. The setup is described in detail below.

1) Extended Kalman Filter: The Extended Kalman filter (EKF) is a nonlinear estimation algorithm used to estimate the states of a system with nonlinear dynamics. It is an extension of the Kalman filter, which handles the nonlinearities by using a first-order linear approximation of the nonlinear system. The EKF works by linearizing the system around the current estimate and then updating the estimate using a combination of the linearized model and the measurements. We model the system as the linear approximation of a nonlinear system, given by

$$X_{k+1} = A_k X_k + \xi, \xi \sim \mathcal{N}(0, \Sigma) \tag{1}$$

where A_k is the linear approximation of the posterior at step k. Here, A_k used in the robot_localization package is the standard motion model for a rigid body. We also note that by implication of using the EKF, we assume a zero mean Gaussian noise in the system dynamics, with covariance Σ as shown above.

To fuse the odometry of the ADAS vehicle with the pose estimates from the smart vehicle, we consider a linear model for the measurements given by

$$Y_k = H_k X_k + \eta, \eta \sim \mathcal{N}(0, \Gamma) \tag{2}$$

Similar to the motion model, the noise here is considered to be zero mean Gaussian, with covariance given by Γ.

For the state estimation of the ADAS vehicle, we consider a 15 dimensional state vector, containing the positions, orientations, linear velocities, angular velocities and linear accelerations. The dataset contains IMU, GPS data and Pose data derived from the IMU and GPS data. Therefore, we would use the Pose data directly for fusion as the odometry data. Consequently, the measurement model is simplified to being an identity transformation. Based on the frequency of the odometry measurements, pose estimates from the smart vehicle, we carry out the prediction and update steps sequentially.

2) Perception Simulator: We simulate the perception module by using the noisy ground truth data. Using the ground truth pose of the ADAS vehicle (P_W^C) and ground truth pose of the smart vehicle (P_W^S), we arrive at the pose of the ADAS vehicle in the frame of the smart vehicle (P_S^C).

$$P_S^C = P_W^C P_S^W \tag{3}$$

During the simulations, independent noises are added to the translation along x and y (T) and rotation (R) of these poses, such that:

$$T' = T + \Sigma, \Sigma \sim (0, \sigma) \tag{4}$$

We work with rotations as quaternions, and rotation noise is added such that,

$$R' = \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} * \begin{bmatrix} 0 \\ 0 \\ cos(\theta/2) \\ sin(\theta/2) \end{bmatrix}, \theta \sim (0, \gamma) \tag{5}$$

where, θ is the rotation about z. This transformation is then combined back with the pose of the smart vehicle in the world frame (P_W^S) to give us The new pose of the ADAS vehicle in the world frame (T_W^C). The translation noises are added to only x,y coordinates.

$$P_S^{''C} = \left[\; R' \; \middle| \; T' \; \right] \tag{6}$$

$$T_W^C = P_S^{''C} P_W^S \tag{7}$$

This measurement T_C^W is used as the input to the second EKF node.

3) Raw Poses: Raw poses are taken from the rosbag, which contains position and orientation information along with timestamps. These are fused pose estimates from the Applanix POS module, which uses its internal IMU, GPS and estimator. Given that the accuracy of the sensors are high, we will perturb these measurements with noises, supplanting for a noisy sensor. For translations T_{raw} and rotation quaternions R_{raw} from the bag,

$$T_{raw}^{'} = T_{raw} + \Sigma_{raw}, \Sigma \sim (0, \sigma_{raw}) \tag{8}$$

$$R'_{raw} = \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} * \begin{bmatrix} 0 \\ 0 \\ cos(\theta_{raw}/2) \\ sin(\theta_{raw}/2) \end{bmatrix}, \theta_{raw} \sim (0, \gamma_{raw}) \tag{9}$$

IV. Experiments and Results

A. Dataset

The Ford Multi-AV Seasonal dataset contains the sensor data in rosbags. These rosbags contain data in the NED (North-East-Down) frame, and this is converted to the ENU (East-North-Up) frame, to align with the standards set by ROS REP-103 for outdoor navigation. Vehicle 1 is considered the smart vehicle, while vehicle 2 is the ADAS vehicle. Given the internal clocks of the vehicles have not been synchronized, we perform this synchronization using GPS time. We compute and publish T_C^W only if the time difference between when P_W^S and P_W^C is received is less than a threshold of 0.1s. For purposes of this work, we perform filtration on a specific portion of the dataset, that consists of two vehicles in proximity to each other. The results shown below are derived using the data from Vehicle 1 and Vehicle 2 from the logs of 24^{th} July 2017.

B. Results

The experiments are structured to test the filter performance through various noise levels and frequencies of data availability. During the experiments, it was noted that the pose estimates along the z-axis for ADAS vehicle (Vehicle 2), were very noisy and uncharacteristic of the sensor specifications.

The frequency of the ADAS vehicle and smart vehicle's poses is around 200 Hz. In the first experiment, we simulate the perception module to also produce estimates using all of the measurements. However, we do add noises to the perception module measurements as described in previous sections. We set $\sigma = 5, \gamma = 5$, to simulate the noises that would arise from using visual feedback. Using this and the raw pose (odometry) measurements of the ADAS vehicle, we produce 6DoF state estimates for the ADAS vehicle, whose trajectory is shown in Fig 3. Note that the raw poses are only corrupted by the noise inherent in the sensor, Applanix POS and its internal estimator. We use the rpg_trajectory evaluation tool [7] on ROS to perform trajectory alignment and analysis.

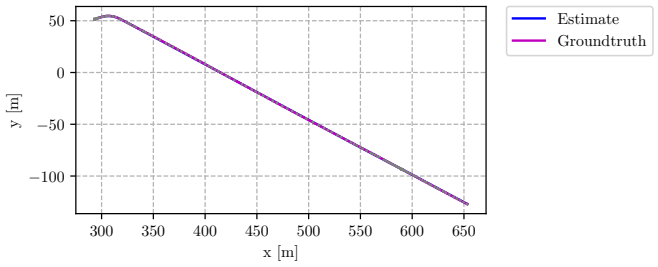

Fig. 3. Trajectory of ADAS Vehicle, σ=5, γ=5, no added noise on raw pose

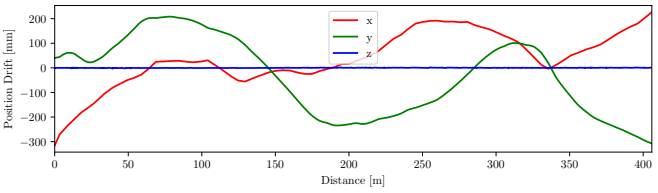

Fig. 4. Position error ADAS Vehicle, σ=5, γ=5, no added noise on raw pose

In Fig. 3, the estimated pose vs. ground truth pose is presented. The corresponding error is shown in Fig. 4 and Fig. 5. As expected, it is evident that when the odometry is accurate, the noises from the perception measurements are filtered out, and we get an absolute mean translation accuracy of 0.127m and an absolute mean orientation accuracy of 1.080 degrees.

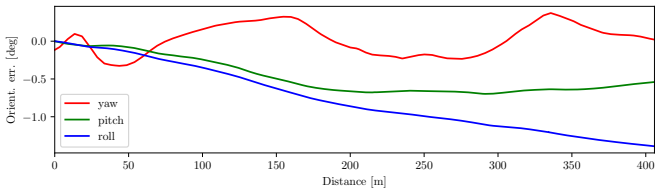

Fig. 5. Orientation error of ADAS Vehicle,, σ=5, γ=5, no added noise on raw pose

In the next experiment, we add noises to the raw pose measurements. We perform ablation on various noise levels across each state. For purposes of this analysis, we study the effects of various noise levels in the perception module, for a particular noise in the raw pose measurements. Tables I and II show the errors in translation at different perception noise levels and the error with only the odometry. Note that fixing $\sigma_{raw} = 2.5m$, $\gamma_{raw} = 0°$ corresponds to having an accurate heading with a high translation noise, where 68% of the additive noises are within (-2.5, 2.5).

TABLE I
TRANSLATION RMSE FOR $\sigma_{raw} = 2.5m$, $\gamma_{raw} = 0°$

Noises Std Dev	$\sigma = 0.3m$	$\sigma = 0.6m$	$\sigma = 0.9m$
w/o perception		0.846	
$\gamma = 10°$	0.095	0.155	0.183
$\gamma = 15°$	0.097	0.144	0.185

TABLE II
TRANSLATION RMSE FOR $\sigma_{raw} = 2.5m$, $\gamma_{raw} = 1°$

Noises Std Dev	$\sigma = 0.3m$	$\sigma = 0.6m$	$\sigma = 0.9m$
w/o perception		1.311	
$\gamma = 10°$	0.111	0.230	0.217
$\gamma = 15°$	0.123	0.235	0.250

However, in practice, the perception module is not expected to produce data at such a high frequency. While the above experiments showcase the potential of using the perception module, to be a practical solution, we need to test its performance at lower frequencies. To sub-sample the data in the rosbag, we make use of the drop tool in ROS. For the experiment, the frequency of the raw messages is dropped down to approximately 100 Hz, while the frequency of the smart vehicle's pose information is dropped down to approximately to 5 Hz. And the results of this are shown in Table III.

TABLE III
TRANSLATION RMSE FOR $\sigma_{raw} = 2.5m$, $\gamma_{raw} = 0.5°$

Noises Std Dev	$\sigma = 0.3m$	$\sigma = 0.6m$	$\sigma = 0.9m$
w/o perception		1.061	
$\gamma = 10°$	0.318	0.825	0.560
$\gamma = 15°$	0.446	0.641	0.499

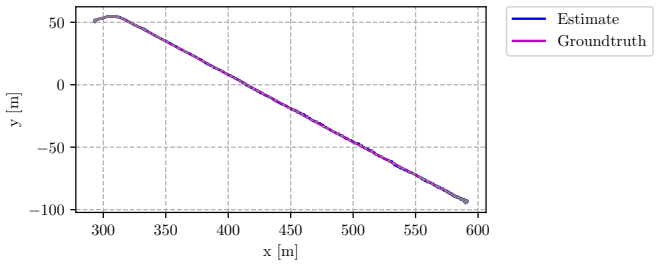

Fig. 6. Trajectory for $\gamma = 10°, \sigma = 0.3m$, reduced frequency

As an example, Fig. 6 shows the trajectory of the vehicle along with the ground truth at one of the noise levels, where we can see a near smooth trajectory.

V. DISCUSSION

From the experiments, it is evident that, even in the presence of noisy odometry, the use of the perception module can improve the overall localization of the ADAS vehicle. Furthermore, the improvement in translation is more prevalent for a wider range of raw pose noises. Using perception does improve heading estimates in some cases. However, it fails to do so when the odometry heading noise levels are beyond a bound. This is likely because given poor heading estimates from both sensor inputs, the filter is unable to filter out the heading noises, which worsens when integrated over time. We find that results are promising even when the frequencies of the measurements are low, as shown in Table III. Another behavior that was noticed was high heading errors with high oscillations in the beginning, which settled to a lower value after a period of time.

VI. CONCLUSION AND FUTURE WORK

In this work, we have developed and presented a localization mechanism that can be used for multi-agent localization, using odometry and visual feedback. The experiments show a clear and consistent improvement in translational accuracy, and bounded errors. The framework when tested on the Ford Multi-AV Seasonal Dataset is robust over various noise levels.

With a working localization pipeline, the future scope of this work is to use a detection and association system instead of the simulator. This would give us a complete detection and filtering framework that can be deployed and tested in real-world systems. Timestamp matching for the data was done using a constant offset which was calculated with GPS time as the reference. However, a better approach can be used to further synchronize the data from both vehicles accurately. Also, in terms of the filter, we have not deployed any outlier rejection techniques or applied any constraints on the evolution of the system dynamics, so this is left to future work.

REFERENCES

[1] Agarwal S, Vora A, Pandey G, Williams W, Kourous H, McBride J. Ford Multi-AV Seasonal Dataset. The International Journal of Robotics Research. 2020;39(12):1367-1376. doi:10.1177/0278364920961451

[2] Z. Zhou, W. Tang, Z. Wang, L. Wang and R. Zhang, "Multi-robot Real-time Cooperative Localization Based on High-speed Feature Detection and Two-stage Filtering," 2021 IEEE International Conference on Real-time Computing and Robotics (RCAR), Xining, China, 2021, pp. 690-696, doi: 10.1109/RCAR52367.2021.9517423.

[3] Roumeliotis, S.I., Bekey, G.A. (2000). Distributed Multi-Robot Localization. In: Parker, L.E., Bekey, G., Barhen, J. (eds) Distributed Autonomous Robotic Systems 4. Springer, Tokyo. https://doi.org/10.1007/978-4-431-67919-6_17

[4] Xian-lun TANG, La-mei LI, Bo-jie JIANG, Mobile robot SLAM method based on multi-agent particle swarm optimized particle filter, The Journal of China Universities of Posts and Telecommunications, Volume 21, Issue 6, 2014, Pages 78-86.

[5] Saeedi S, Trentini M, Seto M, Li H. Multiple-robot simultaneous localization and mapping: A review. Journal of Field Robotics, 2016, 33(1): 3–46 DOI:10.1002/rob.21620

[6] Moore T, Stouch D, A Generalized Extended Kalman Filter Implementation for the Robot Operating System, Proceedings of the 13th International Conference on Intelligent Autonomous Systems (IAS-13), July 2014

[7] Zhang, Zichao and Scaramuzza, Davide, A Tutorial on Quantitative Trajectory Evaluation for Visual(-Inertial) Odometry, IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS), 2018

[8] P. Biber and W. Strasser, "The normal distributions transform: a new approach to laser scan matching," Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453), Las Vegas, NV, USA, 2003, pp. 2743-2748 vol.3, doi: 10.1109/IROS.2003.1249285.

[9] Danping Zou, Ping Tan, Wenxian Yu, Collaborative visual SLAM for multiple agents:A brief survey, Virtual Reality & Intelligent Hardware, Volume 1, Issue 5, 2019, Pages 461-482

[10] Fox, D., Burgard, W., Kruppa, H., Thrun, S. (1999). Collaborative Multi-Robot Localization. In: Förstner, W., Buhmann, J.M., Faber, A., Faber, P. (eds) Mustererkennung 1999. Informatik aktuell. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-60243-6_2

[11] Nadia Nedjah, Luiza Macedo Mourelle, Pedro Jorge Albuquerque de Oliveira, Simultaneous localization and mapping using swarm intelligence based methods, Expert Systems with Applications, Volume 159, 2020, 113547.

[12] Takashi Matsubara, Masao Kubo & Yusuke Murachi (2010) Particle Filter for Collaborative Multi-Robot Localization Tolerant to Recognition Error, Advanced Robotics, 24:15,2043-2058, DOI: 10.1163/016918610X534259

[13] Reid, Tyler & Houts, Sarah & Cammarata, Robert & Mills, Graham & Agarwal, Siddharth & Vora, Ankit & Pandey, Gaurav. (2019). Localization Requirements for Autonomous Vehicles.