Smartphone-sensor based Localization of vehicle position on Lane level estimation

Taehwan Seo*

ABSTRACT

Localization of vehicles is one of the necessities for multiple vehicle operation purposes. In this work, a vehicle lane precision method was proposed with the sensor fusion of Global Navigation Satellite System(GNSS) and Inertial Measurement Unit(IMU) sensors that are built-in common smartphones inside. The process includes the contribution of vehicle localization using smart devices that are widespread, which can minimize the hardware cost of localization of the vehicle. The algorithm was tested over Atlanta metropolitan area I-85 and non-highway roads with 43 km drive and 31 lane changes, and accuracy showed about 94.5 % and 46.2 % of performance over the highway road and non-highway roads. The algorithm and dataset are fully available.

INTRODUCTION

The importance of mobility vehicles is rising worldwide with energy and technology development. For application of state-of-the-art technologies of ground vehicles, such as autonomous cars, requires the localization and precision of vehicle dynamics for operation[1]. In the paper, the protection boundary was calculated about 0.5m to 1.3m, which has smaller bound for a single lane of public road. Also for data-driven techniques, the robustness of the control is fairly limited, and the precise localization takes important role in term of providing correct data information for them.

Sensor fusion is widely used technique for localization of dynamical system. Global Navigation Satellite System(GNSS), Inertial Measurement Unit(IMU), and Light Detection and Ranging(Lidar) can contribute to localization [2], radar and camera also plays a role for the work [3]. In [4], The GNSS and IMU fusion is common usage for localization of vehicle, while the image processing used for motion of the vehicle.

Smartphone built-in sensors are efficient tools for providing either raw or filtered data of positioning. From the built-in IMU,GNSS smartphone sensors, it is widely used in detection including the report of hard-breaking vehicles for the road safety from acceleration sensors [5], Indoor localization with telemetry and IMU sensors [6], driving behavior of the driver with IMU sensors[7][8], accurate navigation method with smartphone sensor fusions [9].

*Graduate Student, Guggenheim School of Aerospace Engineering, Georgia Institute of Technology, 30318.

In this proposed work, the main contribution will be conventional sensor fusion techniques with Kalman Filter from Smartphone sensors that lots of people owns everyday, and verify it for lane level localization of the ground vehicle.

BACKGROUND EQUATIONS

Data Processing

The sensor measurements were processed before utilize the data for the dynamical equation. based on the measured bias from Table 1, sensor measurements were calibrated

$$\begin{bmatrix} calibrated_x \\ calibrated_y \\ calibrated_z \end{bmatrix} = \begin{bmatrix} raw_x \\ raw_y \\ raw_z \end{bmatrix} - \text{Bias}$$

Also for the Accelerometer, the gravity acceleration was removed from body y-axis, approximating the direction of the gravitational force direction because of the error of the orientation.

State Space Integration

From the sensor information, the dynamical system of the vehicle can be approximated with 10 states:

$$X = \begin{bmatrix} \text{Inertial Position}^T \\ \text{Quaternion Orientation Angle}^T \\ \text{Body frame Velocity}^T \end{bmatrix} = \begin{bmatrix} x_i \\ y_i \\ z_i \\ q_0 \\ q_1 \\ q_2 \\ q_3 \\ v_x^b \\ v_y^b \\ v_z^b \end{bmatrix} \quad (1)$$

The body frame is the frame that is defined over the Smartphone given in the figure 1, and it is fixed on the center of the Smartphone. The inertial frame defined as North-East-Down(NED) direction assigned as x-y-z respectively. Inertial frame is fixed in the initial position of the vehicle, providing relative position of the target. In the state parameters, the position and orientation angle are from inertial frame, while the velocity is from the body (smartphone) frame.

The vector conversion between the inertial and body frame can be done with Direction Cosine Matrix(DCM) from

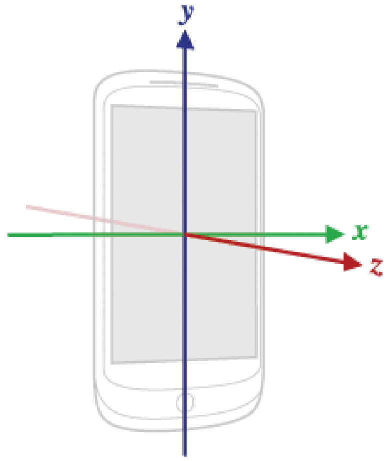


Figure 1: Smartphone body axis

quaternion orientation angle, which expressed in [10] as

$$C_{b/i} = \begin{bmatrix} (q_0^2 + q_1^2 - q_2^2 - q_3^2) & 2(q_1q_2 + q_0q_3) & 2(q_1q_3 - q_0q_2) \\ 2(q_1q_2 - q_0q_3) & (q_0^2 - q_1^2 + q_2^2 - q_3^2) & 2(q_2q_3 - q_0q_1) \\ 2(q_1q_3 - q_0q_2) & 2(q_2q_3 - q_0q_1) & (q_0^2 - q_1^2 - q_2^2 + q_3^2) \end{bmatrix} \tag{2}$$

$$\begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} = {C_{b/i}}^T \cdot \begin{bmatrix} x_b \\ y_b \\ z_b \end{bmatrix} \tag{3}$$

Now, It is required to calculate for the rate of change of each state parameters. For the integration of the inertial position, the integration is defined as

$$\begin{bmatrix} \dot{x}_i \\ \dot{y}_i \\ \dot{z}_i \end{bmatrix} = {C_{b/i}}^T \cdot \begin{bmatrix} v_x^b \\ v_y^b \\ v_z^b \end{bmatrix} \tag{4}$$

To integrate the rate of change of the quaternion orientation angle, the quaternion kinematic equation is used from

[10]

$$\begin{bmatrix} \dot{q_0} \\ \dot{q_1} \\ \dot{q_2} \\ \dot{q_3} \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 0 & -P & -Q & -R \\ P & 0 & R & -Q \\ Q & -R & 0 & P \\ R & Q & -P & 0 \end{bmatrix} \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} \tag{5}$$

where

$$\begin{bmatrix} P \\ Q \\ R \end{bmatrix} = \begin{bmatrix} \omega_x^b \\ \omega_y^b \\ \omega_z^b \end{bmatrix}$$

ω^b is angular velocity of with respect to the body frame, which collected from the built-in gyroscope sensor.

Lastly, the body velocity state v^b is updated based on the accelerometer sensor measurement and correction for torque from center of the mass, expressed as

$$\begin{bmatrix} \dot{v}_x^b \\ \dot{v}_y^b \\ \dot{v}_z^b \end{bmatrix} = \begin{bmatrix} a_x^b \\ a_y^b \\ a_z^b \end{bmatrix} - \begin{bmatrix} \omega_x^b \\ \omega_y^b \\ \omega_z^b \end{bmatrix} \times \vec{r} \tag{6}$$

where a^b is accelerometer measurement, $\vec{r}$ is direction vector from rear wheel of the vehicle, set as $[0, 0, 3]^T$ for this test.

From the equation (4), (5) and (6), state can be integrated for the estimation.

Fusion algorithm

Since the system has nonlinear movement from the defined state space, the Unscented Kalman Filter(UKF) was used for the fusion algorithm. Initial covariance matrix was arbitrarily set as

$$P_0 = diag([100, 100, 100, 10^{-4}, 10^{-5}, 10^{-5}, 10^{-5}, 0.1, 0.1, 0.1]) \tag{7}$$

while the state and measurement covariance matrix were referenced by the pre-test variance measurements, which is written in Table 1.

Algorithm goes as follows from time update:

1-1. From the Simon [11], from the system

$$x_{k+1} = f(x_k, u_k) + w_k \tag{8}$$

$$y_k = h(x_k) + v_k \tag{9}$$

$$w_k \sim (0, Q_k) \tag{10}$$

$$v_k \sim (0, R_k) \tag{11}$$

1-2. Choose sigma points from x_{k-1}^{+} and P_{k-1}^{+} where n is state parameter numbers,

$$\tilde{x}^{(i)} = \left(\sqrt{nP_{k-1}^{+}}\right)_i^T \qquad \text{(i = 1,...,n)} \tag{12}$$

$$\tilde{x}^{(n+i)} = -\left(\sqrt{nP_{k-1}^{+}}\right)_i^T \qquad \text{(i = 1,...,n)} \tag{13}$$

$$\hat{x}_{k-1}^{(i)} = \hat{x}_{k-1}^{+} + \tilde{x}^{(i)} \qquad \text{(i = 1,...,2n)} \tag{14}$$

where subscript i stands for ith row of the matrix.

1-3. Propagate the sigma points with state equation.

$$\hat{x}_k^{(i)} = f\left(\hat{x}_{k-1}^{(i)}, u_k, t_k\right) \tag{15}$$

1-4. Obtain a priori state estimate from the mean of propagated sigma points.

$$\hat{x}_k^{-} = \frac{1}{2n}\sum_{i=1}^{2n} \hat{x}_k^{(i)} \tag{16}$$

1-5. Obtain a priori covariance estimate.

$$P_k^{-} = \frac{1}{2n}\sum_{i=1}^{2n}\left(\hat{x}_k^{(i)} - \hat{x}_k^{-}\right)\left(\hat{x}_k^{(i)} - \hat{x}_k^{-}\right)^T + Q_{k-1} \tag{17}$$

For the timeframe that the observation is not available, which is the time between a second on GPS for this test, a priori estimate of state mean and covariance substituted to the posterior state.

$$\hat{x}_k^{+} = \hat{x}_k^{-} \tag{18}$$

$$P_k^{+} = P_k^{-} \tag{19}$$

And return to 1-2 for another update. Also, For the cases that the measurement is not available or statistical outlier, it does not proceed to measurement update but just perform time update as above. The detection of such statistical outliers was based on the covariance matrix P, ignoring measurement if it is out of 3σ bound from state a priori.

For the points where the observations are available, now goes for measurement update,

2-1. Choose sigma points from a priori information.

$$\tilde{x}^{(i)} = \left(\sqrt{nP_{k-1}^{-}}\right)_i^T \qquad \text{(i = 1,...,n)} \tag{20}$$

$$\tilde{x}^{(n+i)} = -\left(\sqrt{nP_{k-1}^{-}}\right)_i^T \qquad \text{(i = 1,...,n)} \tag{21}$$

$$\hat{x}_{k-1}^{(i)} = \hat{x}_{k-1}^{-} + \tilde{x}^{(i)} \qquad \text{(i = 1,...,2n)} \tag{22}$$

2-2. Obtain measurements from sigma points.

$$\hat{y}_k^{(i)} = h\left(\hat{x}_k^{(i)}\right) \tag{23}$$

2-3. Combine to generate predicted measurement at time t_k

$$\hat{y}_k = \frac{1}{2n}\sum_{i=1}^{2n}\hat{y}_k^{(i)} \tag{24}$$

2-4. Generate measurement variance

$$P_y = \frac{1}{2n}\sum_{i=1}^{2n}\left(\hat{y}_k^{(i)} - \hat{y}_k\right)\left(\hat{y}_k^{(i)} - \hat{y}_k\right)^T + R_k \tag{25}$$

2-5. Estimate cross variance between $\hat{x}_k$ and $\hat{y}_k$.

$$P_{xy} = \frac{1}{2n}\sum_{i=1}^{2n}\left(\hat{x}_k^{(i)} - \hat{x}_k\right)\left(\hat{y}_k^{(i)} - \hat{y}_k\right)^T \tag{26}$$

2-6. Calculate kalman gain

$$K_k = P_{xy}P_y^{-1} \tag{27}$$

2-7. Compute measurement update of state

$$\hat{x}_k^{+} = \hat{x}_k^{-} + K_k(y_k - \hat{y}_k) \tag{28}$$

$$P_k^{+} = P_k^{-} - K_kP_yK_k^T \tag{29}$$

And return to 1-2 for another update.

Algorithm 1: Unscented Kalman Filter Algorithm with Ignorance Criterion

begin
  for t = *0* : t_f **do**
    Apply through (12) - (17)
    if *GPS measurement available* ***and*** *GPS measurement is in 3σ error bound from state priori* **then**
      Apply through (20)-(29)
    else
      apply (18), (19)

METHODOLOGY

Test setup

For this test setup, the Samsung Galaxy S10 was used for the sensor source. The smartphone sensor data can be obtained from the Android API *, and Accelerometer, Gyroscope, and GPS data were collected for this test. To obtain the initial information of the sensors, the bias and variance of sensor measurement were analyzed from known and constant movement of the device. For the GPS information, according to google policy information[12], the providing GPS information includes fusion of Router location, Cell tower, and GNSS systems. It is already a filtered signal from multiple sources, but not including IMU information. Due to the filtered signal property of the given data, the GPS almost showed close zero variance and bias from the measurement. However, the update frequency of GPS limited to a second, requires additional information for trajectory in between.

Table 1: Measured bias and variance data

Sensor	Bias	Variance
Accelerometer - x	0.0251 [m/s^2]	$1.755 \cdot 10^{-4}$
Accelerometer - y	-0.1228	$1.352 \cdot 10^{-4}$
Accelerometer - z	0.0409	$0.983 \cdot 10^{-4}$
Gyroscope - x	-0.0023 [rad/s]	0.0075
Gyroscope - y	0.0009	0.0004
Gyroscope - z	0.0047	0.0049
GPS - Latitude	$7.3181 \cdot 10^{-8}$ [deg]	$1.1964 \cdot 10^{-12}$
GPS - Longitude	0	$1.1964 \cdot 10^{-12}$
GPS - Altitude	0	0.0947 [m]

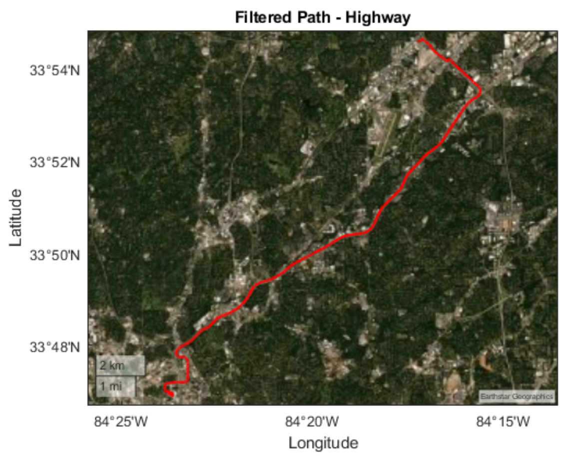
The test apparatus was the smartphone and the mount, and the camera recorder for validation of estimation.

Test drive

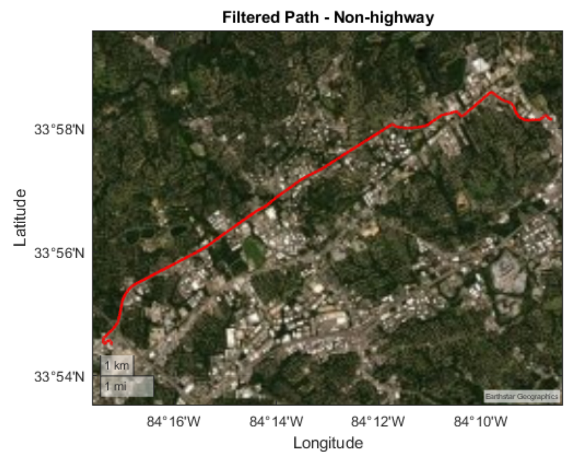
For the collection of the sensor data, test drive was conducted in the Atlanta Metropolitan Area, total 42.6 km with 31 lane changes, 23.3 km of phase 1 drive including highway, 19.3 km of phase 2, Non-highway drive.

*https://source.android.google.cn/devices/sensors?hl=en

Figure 2: Test setup for sensor recordings



(a) Phase 1 path



(b) Phase 2 path

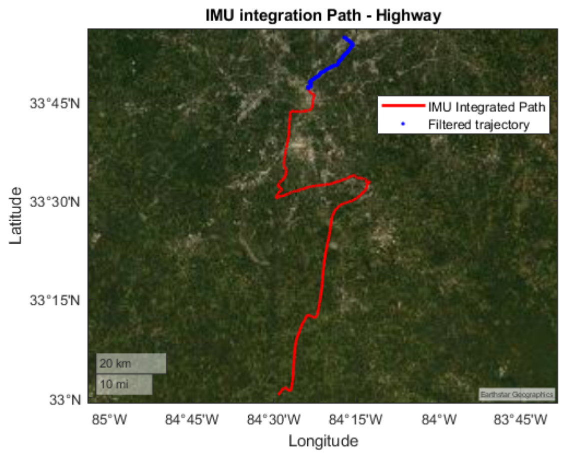
Figure 3: Two test drive paths

RESULT

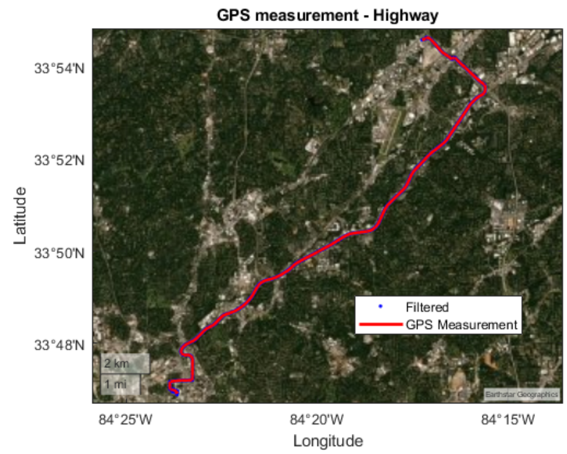
Filtering Efficacy

From the Filtering algorithm shown in (8) to (29), the vehicle position was estimated for every 0.1 seconds. Entire loop took approximately 10 seconds for each of the paths, which is far less time to real driving time. From the comparison with mere state integration with IMU, GPS measurements, filtered response showed better accuracy in position and movement especially involving sharp turns.

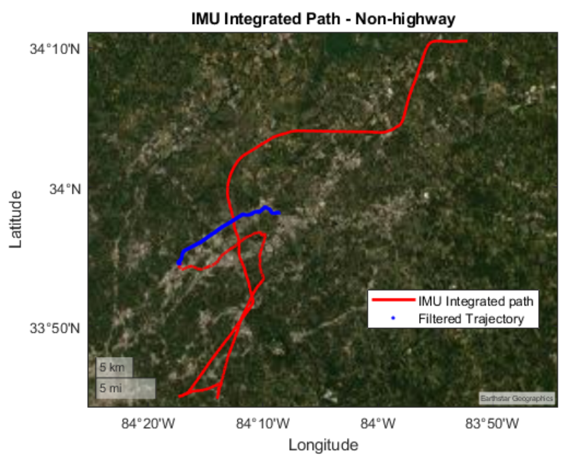
In Figure 4, it is clearly shown mere IMU integration is far different in real trajectory. GPS measurements were precise enough to see the path, but since the GPS updates location about 1 Hz, the gap between measured points were simply interpolated and it stands out in sharp turn, shown in Figure 5 (b). This was improved in Figure 5 (a) with IMU



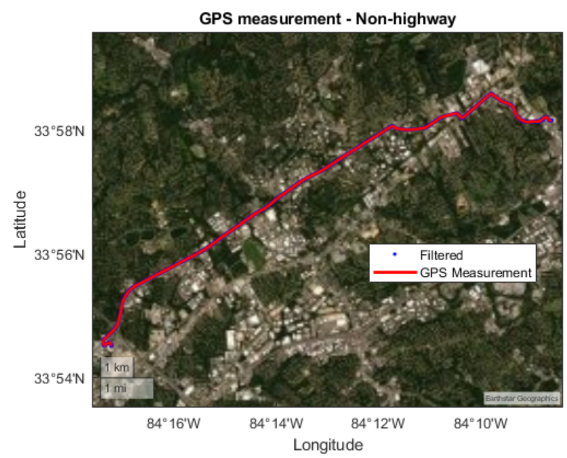
(a) Mere IMU propagation - Phase 1



(b) Mere GPS measurement - Phase 1

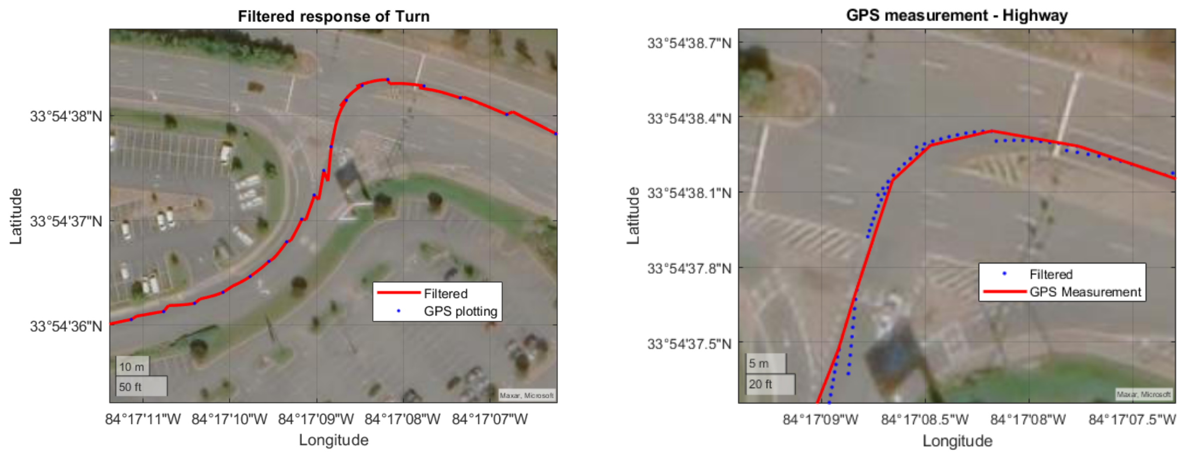


(c) Mere IMU propagation - Phase 2



(d) Mere GPS measurement - Phase 2

Figure 4: Comparison of propagation without filter algorithm



(a) Filtered path (b) Mere GPS measurement

Figure 5: Comparison over sharp turn

fusion.

Also, the from ignorance criterion in Algorithm 1, it showed the improvement over the estimate the path under/after the bridge, where the GPS signals is not available. It was clearly shown in Figure 6, where the statistical outlier peak of position estimate was smoothed to plausible estimate merely with IMU integration with ignorance of possible incorrect measurements in certain circumstances.

(a) Before ignorance criterion (b) After ignorance criterion

Figure 6: Comparison for Ignorance criterion application

Lane change accuracy

The lane level estimation was evaluated with lane change accuracy. From recorded data of test drive, 31 lane changes were identified, and compared it with the timestamp of estimation from the filtered result. Lane keeping was also evaluated, counting incorrect lane keeping over the trajectory.

Table 2: Lane level estimation result

Category	Phase 1 (Highway)	Phase 2 (Non-Highway)
Lane change accuracy	94.5 %	46.2 %
Lane keeping accuracy	75.6 %	53.3 %

From the analysis data in Table 2, the highway condition showed better performance over both lane changing and lane keeping. It is small data and requires more of the test to see the general trend, but the GPS measurement showed better performance over the highway than the non-highway roads, and this affected to the performances over the lane level estimation of the vehicle. The sensor data and filtering code is attached.*

(a) Lane change path

(b) Lane change dash camera

Figure 7: Lane change situation

CONCLUSION

From this research, Smartphone, which is globally and widely used device, was utilized for Lane-level vehicle position estimation. Built-in IMU sensors and GPS measurements were used for dynamical estimate and sensor measurement of the vehicle position. Unscented Kalman Filter and 3σ outlier ignorance algorithm was used for sensor fusion, and the result shown 94.5 % lane change accuracy and 75.6 % lane keeping accuracy on the highway, and 46.2 % lane change accuracy and 53.3 % lane keeping accuracy over the non-highway.

Several future work can be proposed with the topic. First, the data may be computed through algorithm real-time with the smartphone, using Smartphone processors to compute. Second, develop image processing for detecting lanes in satellite images to automatically detect the current lane and lane changes. Lastly, research for special circumstances, such as GNSS denied spots in the urban areas.

*https://github.com/jjffkkgg/AE6505-KalmanFiltering-Vehicle-Lane-Estimation

REFERENCES

[1] T. G. R. Reid, S. E. Houts, R. Cammarata, G. Mills, S. Agarwal, A. Vora, and G. Pandey, "Localization Requirements for Autonomous Vehicles," *CoRR*, Vol. abs/1906.01061, 2019.

[2] X. Meng, H. Wang, and B. Liu, "A robust vehicle localization approach based on GNSS/IMU/DMI/LiDAR sensor fusion for autonomous vehicles," *Sensors (Switzerland)*, Vol. 17, No. 9, 2017, 10.3390/s17092140.

[3] D. Kang and D. Kum, "Camera and Radar Sensor Fusion for Robust Vehicle Localization via Vehicle Part Localization," *IEEE Access*, Vol. 8, 2020, 10.1109/ACCESS.2020.2985075.

[4] Z. Wang, Y. Wu, and Q. Niu, "Multi-Sensor Fusion in Automated Driving: A Survey," 2020, 10.1109/ACCESS.2019.2962554.

[5] L. Liu, D. Racz, K. Vaillancourt, J. Michelman, M. Barnes, S. Mellem, P. Eastham, B. Green, C. Armstrong, R. Bal, S. O'Banion, and F. Guo, "Smartphone-based Hard-braking Event Detection at Scale for Road Safety Services," 2022, 10.48550/ARXIV.2202.01934.

[6] A. Poulose, O. S. Eyobu, and D. S. Han, "An Indoor Position-Estimation Algorithm Using Smartphone IMU Sensor Data," *IEEE Access*, Vol. 7, 2019, pp. 11165–11177, 10.1109/ACCESS.2019.2891942.

[7] P. Handel, I. Skog, J. Wahlstrom, F. Bonawiede, R. Welch, J. Ohlsson, and M. Ohlsson, "Insurance Telematics: Opportunities and Challenges with the Smartphone Solution," *IEEE Intelligent Transportation Systems Magazine*, Vol. 6, No. 4, 2014, pp. 57–70, 10.1109/MITS.2014.2343262.

[8] J. Wahlström, I. Skog, and P. Händel, "Detection of Dangerous Cornering in GNSS-Data-Driven Insurance Telematics," *IEEE Transactions on Intelligent Transportation Systems*, Vol. 16, No. 6, 2015, pp. 3073–3083, 10.1109/TITS.2015.2431293.

[9] O. Walter, J. Schmalenstroeer, A. Engler, and R. Haeb-Umbach, "Smartphone-based sensor fusion for improved vehicular navigation," *2013 10th Workshop on Positioning, Navigation and Communication (WPNC)*, 2013, pp. 1–6, 10.1109/WPNC.2013.6533261.

[10] B. L. Stevens and F. L. Lewis, "Aircraft Control and Simulation," *Aircraft Engineering and Aerospace Technology*, Vol. 76, No. 5, 2004, 10.1108/aeat.2004.12776eae.001.

[11] D. Simon, *Optimal State Estimation*. 2006, 10.1002/0470045345.

[12] Google Inc., "How Google uses location information,"