2023 IEEE International Conference on Mobility, Operations, Services and Technologies (MOST) | 979-8-3503-1995-8/23/$31.00 ©2023 IEEE | DOI: 10.1109/MOST57249.2023.00015

Real-Time Vehicle Localization Using Steering Wheel Angle in Urban Cities

Raef Abdallah
Department of Computer Science
Wayne State University
Detroit, Michigan, USA
raef@wayne.edu

Baofu Wu
School of Computer Science
Hangzhou Dianzi University
Hangzhou, Zhejiang, China
baofu.wu@hdu.edu.cn

Jian Wan
School of Computer Science
Hangzhou Dianzi University
Hangzhou, Zhejiang, China
wanjian@hdu.edu.cn

***Abstract*—Whether it is a small autonomous shuttle picking up and dropping off passengers, a robot navigating a large warehouse, a pizza delivery autonomous vehicle, or a truck fleet delivering goods and services, the precise localization of a moving object plays an important role in its safety and reliability. An integrated system composed of an onboard Inertial Measurement Unit (IMU) and a Global Positioning System (GPS) is utilized in vehicles nowadays and can precisely determine the location of a vehicle in real time; however, the vehicle localization accuracy degrades significantly even during the short duration of unavailable satellite signal. In this work, we propose using steering wheel angle and odometer data to determine the vehicle location during GPS satellite outages. Several test-driving experiments were conducted using an OBD-II Vehicle Interface (VI) and a tablet. We augmented our approach using reference GPS coordinates to enhance the vehicle location and rectify bias caused by odometer readings. We compared our approach with the vehicle's GPS navigation system in an urban environment and verified that our proposed approach performs better. Comparing our approach to IMU has shown that the former predicted locations more accurately and with fewer error drifts.**

***Index Terms*—GPS outage, Location, Urban City, Steering wheel angle, Odometer**

I. Introduction

The Global Positioning System (GPS) is a satellite-based navigation system that enables the determination of the precise location, speed, and time of a moving object [1]. It is widely used in various applications such as vehicle navigation, transportation management, and package delivery. The accuracy of GPS-based location information is generally high, but it can be affected by various factors such as interference from tall buildings [2], atmospheric conditions [3], and solar flares [4]. Also, the accuracy degrades in areas where signals are weak or blocked, such as in congested cities and inside tunnels. When GPS signals are lost or degraded, the location accuracy of the system is significantly reduced, leading to problems such as navigation errors and safety risks.

To address the issue of GPS outages, researchers and engineers have developed various methods to mitigate the impact on location accuracy. They use the inertial measurement units (IMUs) [8], which can determine the location of a moving object based on its acceleration and angular rate. With IMU data, they designed many dead reckoning algorithms [9]–[11] to estimate the location of a moving object based on its previous location and the distance and direction traveled. While dead reckoning algorithms can be effective for short-term location estimation, they are prone to errors due to the accumulation of errors over time, which can limit their accuracy in long-term applications.

Besides, researchers have also proposed the use of additional sensors such as cameras and lidar [12]–[15] to improve the accuracy of location estimation during GPS outages. These sensors can provide additional information about the environment and can be used to correct errors in the location estimation. For example, Bosch utilizes the fusion of camera and radar sensor systems to collect information about the vehicle's surroundings in real-time and compare it with an independent map layer stored in the Cloud that contains unique road features [18].

Global Navigation Satellite Systems (GNSS) and Inertial Measurement Units (IMUs) are widely utilized in modern vehicles to provide accurate position, velocity, and altitude information. GNSS relies on constellation of satellites transmitting signals that carry timing and positioning data to GNSS receivers. In ideal conditions, GNSS is accurate and reliable; however, this accuracy degrades in areas where signals are weak or blocked such as in congested cities and inside tunnels. Companies such as Bosch provide localization solutions for autonomous vehicles so that the vehicle is aware of its current location at all times. To do so, Bosch uses different technologies to pinpoint the exact location of an autonomous vehicle. Bosch utilizes the fusion of camera and radar sensor systems to collect information about the vehicle's surroundings in real time and compare it with an independent map layer stored in the Cloud that contains unique road features.

Moreover, technologies such as satellite navigation and inertial sensors assist vehicle sensors in providing reliable and precise positioning information. When the satellite signal is interrupted due to an outage or obstruction, the sensors in the IMU activate to measure the vehicle's acceleration and its angular motion. Combining this information with steering wheel angle sensors and wheel speed sensors, the relative change in the vehicle's position is calculated making localization possible even when satellite connection malfunctions.

To address this problem, we propose a new method for determining the location of a moving object during GPS

979-8-3503-1995-8/23/$31.00 ©2023 IEEE
DOI 10.1109/MOST57249.2023.00015

outages without the reliance on sensors in the vehicle. Instead, our approach relies on steering wheel angle and odometer CAN signals. We demonstrate its effectiveness through a series of experimental evaluations. Our results show that our method can provide reliable and accurate location information during GPS outages, making it suitable for use in various applications.

In this paper, we present the background and related work in Section II. In Section III, we focus on the design of methodology. In Section IV, we introduce the data flow of the proposed method, and in Section V, we describe the implementation of our approach. In Section VI, we provide an evaluation of the system. In Section VII, we discuss features of the proposed method and future work, and finally, we summarize the whole paper.

II. BACKGROUND AND RELATED WORK

In the last decade, consumer portable devices such as smartphones and tablets have been instrumental in providing solutions related to navigation, storage, safety and much more. Smartphones and other smart devices contain several sensors such as gyroscopes, accelerometers, magnetometers, GNSS receivers, and all of which can be utilized to assist in navigation. The relative low cost of smart devices has attracted researchers to use these devices as edge devices to provide additional computing power at the edge of the network, or to be utilized in autonomous vehicles and robots for navigation purposes [20]. [21] summarizes the first ten years of researching the use of smartphones in the field of vehicle telematics and surveys the latest smartphone applications used in cloud computing, vehicle navigation, road condition monitoring and other purposes. [22] describes a smartphone application that relies on smartphone GNSS receivers and leverages edge computing and machine learning to provide essential low-clearance overpass warning mechanism to tall vehicle drivers. Therefore, it comes to no surprise that research work has used smartphone sensors such as gyroscopes to estimate steering wheel angle. In order to rectify gyroscope biases and errors, accelerometer and magnetometer sensor measurements were introduced combined with the use of Extended Kalman Filter [23]. In other scenarios, GPS/INS (Inertial Navigation System) was integrated with a wheel speed sensor for land-vehicle positioning [25]. In [24], a steering angle sensor was integrated with Micro-Electro-Mechanical Systems Inertial Measurement Unit (MEMS IMU) to enhance positioning accuracy during GPS outages. To meet the stringent positioning accuracy requirements of autonomous vehicles, research work proposed the integration of GPS with odometer, compass, and gyroscope rather than relying on much more costly MEMS IMU. Nevertheless, odometers are prone to slippage; therefore, such readings can be biased. Other work incorporated consumer portable devices to measure heading changes by estimating the steering angle as an update in the navigation estimation filter. This approach was adopted since steering angle information is not typically provided by commercial OBD-II units and requires hardware and software customization [26].

[16] investigates a modular approach for odometry localization of vehicles with increased maneuverability, such as drones or autonomous cars. The approach consists of several modules, including data fusion, motion model prediction, and error correction. The data fusion module combines measurements from multiple sensors, including accelerometers, gyroscopes, and encoders, to estimate the vehicle's pose. The motion model prediction module uses the vehicle's kinematic and dynamic models to predict its future state based on previous estimates.The error correction module uses a Kalman filter to correct any errors in the position and orientation estimates. The approach also includes a method for detecting and mitigating sensor failures, which can be a common problem in autonomous systems.

In [17], the authors propose a solution to maintain an acceptable precision of the vehicle positioning even during the satellite signal outages. They tackle challenges of relying on a real-time kinematic (RTK) GPS system. RTK GPS is a type of GPS technology that can be used for precise positioning in real-time applications. RTK GPS uses a network of reference stations to provide differential corrections to the GPS signals received by the receiver on the vehicle. These differential corrections can improve the accuracy of GPS positioning to within a few centimeters. The challenge is that RTK GPS deliver position data with a delay. The proposed solution identifies this latency by using a combination of GPS measurements and data from other sensors, such as accelerometers and wheel encoders, in a sensor fusion approach. The sensor fusion approach uses a Kalman filter to estimate the vehicle's position and orientation based on the different sensor measurements.

Previous studies have demonstrated that malfunctions in onboard sensors can lead to catastrophic consequences for autonomous vehicles, such as vehicle accidents and unmanned aerial vehicle malfunctions. Our proposed model offers an alternative approach for autonomous vehicles to operate safely during sensor failures, especially for ground vehicles travelling on straight roadways. However, further research is required to investigate the appropriate integration of this method, and related mechanistic methods aimed at other types of autonomous vehicles, to ensure the resilience of the system against any combination of sensor failures. Jha et al [6] and Taylor et al [5] have provided comprehensive model checking techniques for such integration, while Pritam et al. [7] have extended these approaches to encompass a broader range of sensor failures.

A. Location without GPS

Lack of access to GPS can present a significant challenge for autonomous vehicles seeking to accurately estimate their position. To address this problem, researchers have proposed a range of approaches that can be used to predict the vehicle's location in the absence of GPS. These methods can be broadly grouped into two categories: those that rely on onboard sensors and those that rely on external information.

Sensors: One approach to predicting the vehicle's location in the absence of GPS is to use onboard sensors to estimate the vehicle's movement over time. Inertial measurement units

(IMUs) are one type of sensor that can be used for this purpose. IMUs measure the vehicle's acceleration and angular velocity, which can be used to calculate the vehicle's position and orientation over time. Odometers, which measure the distance traveled by the vehicle, can also be used to estimate the vehicle's position based on its starting location. Laser scanners, which can create a map of the surrounding environment, can be used to estimate the vehicle's position relative to the map.

External information: In addition to using onboard sensors, vehicles can also use external information to estimate their location. For example, they can use map data or location data from other vehicles in the area to determine their position. They can also use information about the surrounding environment, such as the location of landmarks or the layout of the road network, to estimate their location.

Overall, the accuracy and reliability of these approaches will depend on the quality and availability of the sensors and external information used. Methods that rely on sensors are generally more accurate in the short term, while those that rely on external information are more accurate in the long term. Further research is needed to determine the most effective combination of sensor and external information sources for predicting the vehicle's location in the absence of GPS.

B. Location with IMU

Inertial measurement units (IMUs) are commonly used in autonomous vehicles as a way to estimate the vehicle's position and orientation in the absence of GPS. IMUs consist of sensors that measure the vehicle's acceleration and angular velocity, which can be used to calculate the vehicle's movement over time.

There are several approaches to using IMUs to predict the vehicle's location without GPS. One approach is to use a Kalman filter, which is a statistical algorithm that can estimate the vehicle's position and orientation by fusing data from the IMU with other sources of information, such as odometry or map data. Kalman filters can effectively reduce the error in the IMU measurements over time, allowing for more accurate estimates of the vehicle's location.

Another approach is to use a particle filter, which is a probabilistic algorithm that can estimate the vehicle's position and orientation by generating a large number of possible locations and orientations and weighing them based on the likelihood that they correspond to the true position and orientation of the vehicle. Particle filters can also be used in conjunction with other sources of information, such as odometry or map data, to further improve the accuracy of the estimate.

The disadvantage of using inertial measurement units (IMUs) to predict the vehicle's location without GPS is that IMUs are subject to drift over time. This drift occurs because the measurements from the IMU are subject to error, and these errors can accumulate over time, resulting in a growing discrepancy between the estimated position and the true position of the vehicle.

Overall, the accuracy of IMU-based approaches to predict the vehicle's location without GPS will depend on the quality of the IMU sensors and the effectiveness of the estimation algorithm being used. Further research is needed to determine the optimal combination of IMU sensors and estimation algorithms for predicting the vehicle's location in the absence of GPS.

III. Methodology

This paper proposes a new approach to estimate the positioning of a vehicle when GNSS is unavailable by relying on real-time vehicle information received via the On Board Diagnostic (OBD-II). An OBD2 message structure is comprised of identifier and data. The first field in an OBD2 messasge is the 11-bit CAN identifier. This identifier ID is used to distinguish between request and response messages. The second field is the length and contains the remaining number of bytes. For example, a steering wheel angle request will contain 2 bytes in the length field whereas it will have 3 bytes in the same field in a response message. The extra byte in the response message reflects the steering wheel angle value. The third field is the mode and its values range from 01 to 0A for a request message. The range of the mode field is between 41 and 4A for a response message. SAE J1979 OBD2 standard identifies 10 modes as listed in Table I.

Mode	Description
01	Show current data
02	Show freeze frame data
03	Show stored Diagnostic Troubled Codes (DTCs)
04	Clear DTCs and stored values
05	Test results for oxygen sensors (non CAN only)
06	Test results for system monitoring (and oxygen sensors for CAN)
07	show pending DTCs
08	Control operation of on-board system
09	Request vehicle information
0A	Permanent DTCs

TABLE I: OBD2 Diagnostic Message Mode Field.

A. System Architecture

The proposed navigation system architecture is shown in Figure 1.

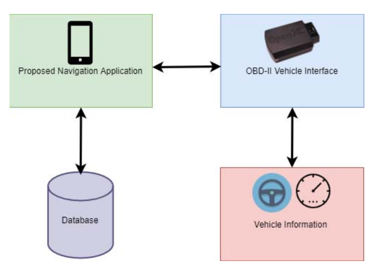

Fig. 1: Navigation System Architecture.

When a GNSS outage occurs at time t, the proposed navigation system activates and is composed of the following:

1) A navigation application running on a mobile device that makes a request to the OBD-II Vehicle Interface for specific vehicle information. This information includes GPS coordinates and the bearing at time t where t is the time when the GPS outage occurred.
2) In addition to GPS and bearing data, the OBD-II provides steering wheel angle and odometer data which can be used to estimate the vehicle heading and the distance traveled respectively. The navigation application requests vehicle information from the OBD-II Vehicle Interface. The steering wheel angle and the odometer signals received via the OpenXC Vehicle Interface have frequencies of 10Hz. OpenXC is basically a small hardware module that plugs into the OBD-II port to read and translate signals received from the vehicle's internal network.
3) A database that stores GPS coordinates is used to adjust the calculated coordinates of the vehicle's new position. This adjustment is necessary due to errors introduced by the odometer reading biases. The proposed navigation algorithm makes these coordinates' adjustments when necessary. More details on that will be discussed in IV.

IV. DATA FLOW

This section introduces the overview data flow of our proposed method. It includes three main parts: data retrieval, data processing, and data correction.

The data retrieval process is shown in blue in Figure 2. The OBD-II captures the steering wheel angle, odometer, bearing, and GPS signals. When a GPS outage occurs, the distance traveled is calculated by taking the difference between two odometer readings at time t and $t+1$ respectively. The distance traveled along with the steering wheel angle, last bearing reading and the last GPS reading are passed as input to the function that calculates the vehicle's new position. If the steering wheel angle $\alpha \leq p$, then the vehicle's new position is calculated by repeating the same process. If $\alpha > p$, reference points are retrieved from the database to adjust the bias. Additionally, the new bearing and the traveled distance are calculated to pinpoint the new vehicle's position. More detail will be presented in subsection IV-C.

A. Data Retrieval

1) Bearing data: The data retrieval process starts when GNSS outage occurs at time t. At time t, the mobile application communicates with the OBD-II Vehicle Interface (VI) to receive steering wheel angle and odometer data. The VI then requests steering wheel angle and odometer data by setting the mode in the OBD-II message to 01 (Table I). The mobile application requires two other pieces of information at time t before the next data processing stage starts. One is the GPS coordinates, and the second is the bearing. Bearing is the angle measured clockwise from the north. The frequency of the bearing angle retrieved from the vehicle interface device is 1Hz. Figure 3 shows the bearing of a vehicle on a trajectory from A to B. The bearing θ between two GPS coordinates can be calculated as follows:

$$\begin{gathered}\theta = atan2(X, Y) \\ Where \\ X = cos(LatitudeB) * sin(LongitudeB - LongitudeA) \\ Y = cos(LatitudeA) * sin(LatitudeB) - sin(LatitudeA) \\ * \, cos(LatitudeB) * cos(LongitudeB - LongitudeA)\end{gathered} \tag{1}$$

2) Odometer Data: Using the odometer signal frequency of 10Hz, the distance traveled is calculated every 100 milliseconds as the difference between odometer readings between time $t+1$ and t as in the following:

$$d = Odometer(t+1) - Odometer(t) \tag{2}$$

where d is the distance traveled between time t and $t+1$

3) Steering Wheel Data: The obtained steering wheel data at a frequency of 10 Hz will determine whether a vehicle is making a turn or not. In other words, the steering wheel angle is used to determine if the vehicle's orientation has changed. The steering wheel angle data indicates whether the vehicle is making a left turn, right turn or continuing to move in the same direction. In subsection IV-C, we will explain in more detail how the algorithm determines if the vehicle changed orientation based on the steering wheel angle value.

4) GPS Data: GPS readings are retrieved at a frequency of 1Hz from OpenXC (vehicle interface device). When a GPS outage occurs, the application captures the last GPS reading. The proposed application then uses this last reading to continuously generate latitude and longitude data to approximate the new vehicle's position. This process continues until GPS service is restored.

B. Data Processing

All the input collected as explained in the previous subsections are processed to calculate the new vehicle's position. The distance traveled d, the bearing θ, and the last GPS coordinate retrieved or calculated act as input parameters to calculate the new vehicle's position at time $t+1$ (Figure 2). The first time the new position is calculated will use the last GPS and bearing readings and the distance traveled between time t and $t+1$ where time t is when the outage occurred. As long as the GPS outage persists, the proposed application uses the new generated GPS coordinates, the bearing angle θ, and the distance traveled to continuously provide the vehicle's real-time position. If the vehicle's orientation does not change, we assume that θ remains constant. If a change in orientation takes place, the bearing angle θ has to be updated. Additionally, the GPS coordinates have to be corrected using reference points that will be explained in the next subsection.

The approach we used to calculate the new GPS coordinates uses a spherical coordinate system. Equation 3 calculates the target longitude and latitude given the source longitude and latitude, the bearing, and the distance the vehicle traveled from the source to the target.

$$\begin{gathered}lat_{t+1} = lat_t + d/R * (180/\pi) * \cos\theta \\ lon_{t+1} = lon_t + d/R * (180/\pi) * \cos\theta / cos(lat_t)\end{gathered} \tag{3}$$

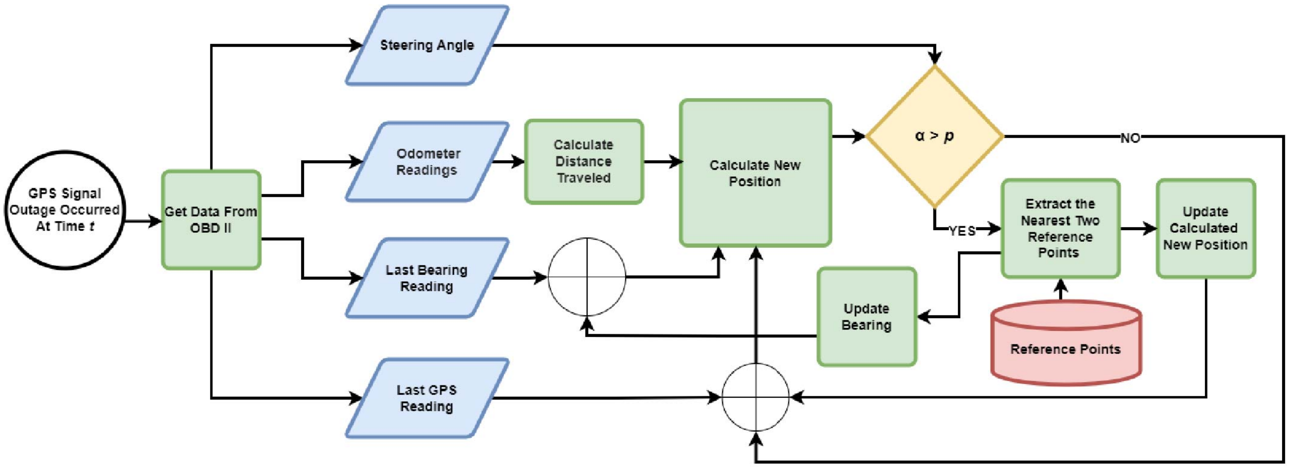

Fig. 2: Proposed Navigation System Data Flow.

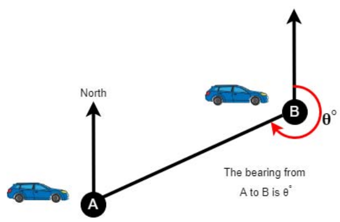

Fig. 3: Bearing Angle of a Vehicle from on a course from A to B.

In Equation 3, R is the radius of the earth, d is the distance traveled. Both latitude and longitude are expressed in radians.

C. Data Correction

After calculating the new vehicle's position, the steering wheel angle α is compared to a predetermined threshold value p. The threshold value is dependent on the vehicle's year and model and varies from one vehicle model to another. If $\alpha > p$, it indicates that the moving vehicle is changing heading and that consequently results in a change of bearing. At the time instant when this change in heading occurs, GPS coordinates are retrieved from the database. The database contains reference points consisting of street intersection GPS coordinates. The GPS coordinates retrieved from the database represents the nearest coordinates to those coordinates at the time instant when $\alpha > p$ occurs. Once the reference point is retrieved from the database, the previous process (Data Processing) continues and the new vehicle's position is calculated. On the other hand, if the vehicle does not change heading, the navigation application does not need to connect to the database to retrieve reference points, but instead it uses the previously calculated position at time t to calculate the new position of the vehicle at time $t + 1$. To calculate the new vehicle's position at time $t + 1$, the vehicle's position at time t is used and can be one of the following three sources: 1) Last GPS reading when outage occurred, 2) Position at time t, or 3) Reference GPS coordinates retrieved from database. This is represented by the plus sign inside the circle in Figure 2 above. The pseudocode below summarizes the data correction process.

if $\alpha > p$ **then**
  extract the nearest reference coordinates from database to adjust odometer bias
  update calculated GPS coordinates with the new reference coordinates retrieved from database
else
  use previously calculated GPS coordinates
end if

V. Implementation

To validate our proposed navigation approach, we utilized our autonomous vehicle, Hydra to drive through the streets of Detroit midtown. Hydra collects vehicle data in real time via several onboard computers. For our purposes, we collected Hydra's GPS location, IMU data, steering wheel angle, and odometer readings. We also utilized a Vehicle Interface (VI), OpenXC, which is an open source composed of software and hardware that supports reading CAN messages via OBD-II. OpenXC VI reads data from CAN bus and displays it in JSON format. In our vehicle test, we connected a tablet to the OpenXC VI device via a USB port interface (4). The system was implemented using Python and C# programming languages data retrieval and processing respectively. Our navigation system can be described as 3-tier architecture.

A. Presentation Layer

The presentation layer displays vehicle's GPS position computed by the middle tier (application layer). it simply receives input for the middle tier and makes API call to pinpoint the current vehicle location on a map (See Figure 9).

B. Middle Tier

The middle tier is where the algorithm resides. It is the core of the system, and it starts by making requests via the OBD-

Fig. 4: Proposed Navigation System.

II VI to receive GPS coordinates, steering wheel angle, and odometer readings in JSON format. If $|\alpha(t+1) - \alpha(t)| \leq p$ (small steering wheel angle change), we assume that the vehicle did not change heading; therefore, bearing holds the same value θ. In the case where $|\alpha(t+1) - \alpha(t)| > p$, bearing is then computed from two GPS coordinates on two different streets. The first coordinate is the intersection reference point (intersection point is the intersection between the current street vehicle is on and the next street vehicle is adopting) retrieved from the database to adjust bias. The second point is the other intersection point on the street the vehicle just adopted. To illustrate this further (refer to Figure 5), assume the vehicle is at point A at time t and reaches point B at time $t1$. When the vehicle makes a right turn, the other intersection point (point C) is retrieved from the reference point database. To extract point C, the vehicle relies on a previous calculated point prior to reaching point B and the steering wheel angle value (positive or negative) to determine point C. The bearing θ is calculated using Eq.(1). This layer also receives the odometer readings at time t and $t+1$ to calculate the distance as shown in Eq.(2).

C. Backend

The reference points stored in a SQL database played an essential role in correcting odometer bias and calculating the bearing when the vehicle changes heading. The database only contains key reference points (latitude and longitude) of city streets. These reference GPS coordinates are simply collected from Google Maps and stores in our database.

VI. SYSTEM EVALUATION

To evaluate our proposed method, we will design three comparison experiments: First, we will verify the benefits of our reference point-based scheme design. Second, we will verify that our proposed scheme performs better than GPS in complex urban scenarios. Finally, we will verify that our proposed scheme performs better than using IMU alone without GPS.

Before comparison experiments, we introduce the dataset and experiment setup to make the experiment clearer.

To evaluate the performance of our proposed method, we calculated the average distance between predicted locations and the ground truth.

Fig. 5: Ground Truth Route Adopted During Test Drive.

A. Dataset

We will use a dataset collected from the autonomous vehicle, Hydra [27]. We drove Hydra around Wayne State University campus and collected a 103 second dataset. The ground truth trajectory is shown in Figure 5. Hydra mounts many sensors and can provide GPS, IMU, odometer, and steering wheel data to support our task, which can be collected from CAN bus. Three sets of data are central to our experiment, and we will describe each in more detail next.

1) Ground Truth: Ground truth is the true position of the vehicle on the road. As shown in Figure 5, we drove Hydra clockwise along the following route: `York St 2 -> Second Ave -> Burroughs St -> Cass Ave -> York St 1`, with a tall building located along the roadside of Cass Ave. During the data collection process, the Hydra vehicle strictly follows the center of the lane. It is worth noting that the reference points used in this paper are the intersection GPS coordinates in the ground truth.

2) GPS Data: We took a test drive to collect GPS data from Hydra's navigation system and from our proposed navigation system. The GPS dataset contains longitude and latitude coordinates retrieved by the vehicle's navigation system. In our case, the collected GPS data is not the ground truth. The blue route in Figure 9 shows the GPS coordinates collected by Hydra's GPS navigation system while the orange route shows the same for the data collected by the proposed navigation system. The GPS position received by a vehicle deviates from the true position due to the lack of RTK correction for the vehicle's position and the interference of complex scenarios such as tall buildings on the GPS signal.

3) Steering Wheel Data: The test drive consisted of a clockwise movements in the city of Detroit. Just as Figure 6 displays, there are noticeable spikes in the steering wheel angle data every time the vehicle makes a sharp turn, which was an important basis for our algorithm to determine whether the vehicle changed direction. Figure 6 describes the steering wheel angle changes. The x-axis represents the time in seconds, and the y-axis shows the steering wheel angle in degrees. At the start of the trip, the steering wheel angle is 0°. The first downward dip occurred when the vehicle was making the first right turn. The minimum value in the first dip is -238.299927° at approximately time t=25 seconds. After making a right turn, the vehicle then moves in the same heading for almost 17 seconds before changing heading again. This is depicted in the second downward slope in Figure 6 where the minimum steering wheel angle is -264.900024°. Similarly, the vehicle moves again in one direction for approximately 25 seconds before changing heading at approximately time t=72 seconds where the steering wheel angle value is -283.699951. The last turn the vehicle made at approximately time t=90 seconds with a minimum steering wheel angle value of -236. In summary, a negative steering wheel angle indicates that the steering wheel is turning clockwise while a positive value indicates the steering wheel is turning counterclockwise.

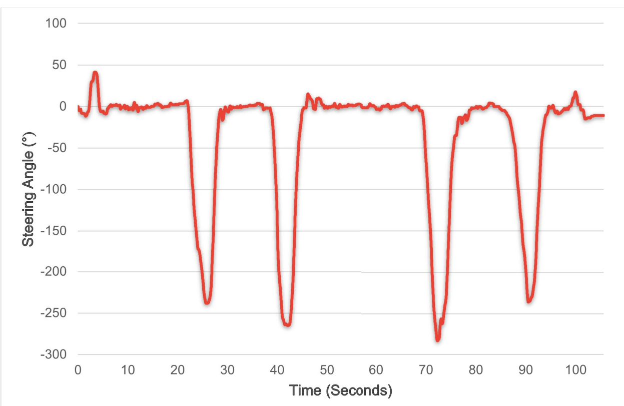

Fig. 6: Steering wheel angle.

B. Experimental Setup

We conducted experiments in the city of Detroit using Hydra, an autonomous vehicle, collecting GPS and IMU data and saving it locally. We also collected steering wheel angle and odometer in real time using OpenXC. We used the proposed navigation system shown in Figure 4 to estimate the vehicle's location. We then compared the results with other methods, which we will introduce in detail in the following subsections.

C. Performance of Reference Point-Based Method

Our proposed approach is characterized by two key design features: 1) Utilizing the steering wheel angle and odometry for localization, and 2) Employing reference points to improve vehicle's location. To illustrate the benefits of using reference points, we compared our initial approach (without reference points) to the same using reference points. This comparison is shown in Figure 7. The yellow line represents the predicted route without using reference points, while the orange dashed line represents the route using reference points. Figure 7 shows the vehicle's location prediction using reference points is more accurate than its counterpart because it is closer to the green route that represents the ground truth.

Overall, the use of reference points improves the accuracy of the vehicle location prediction.

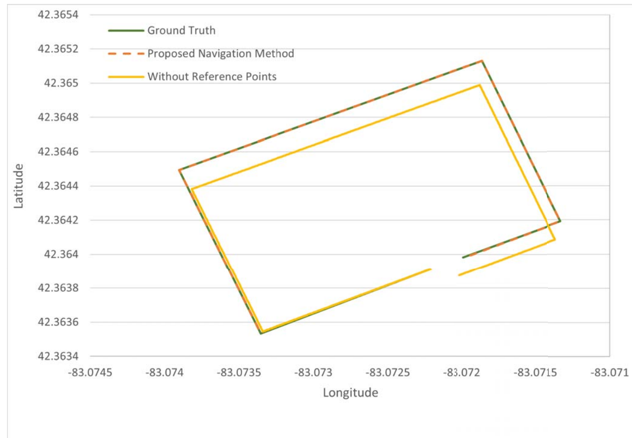

Fig. 7: Reference Point Use Versus No Reference Point Use.

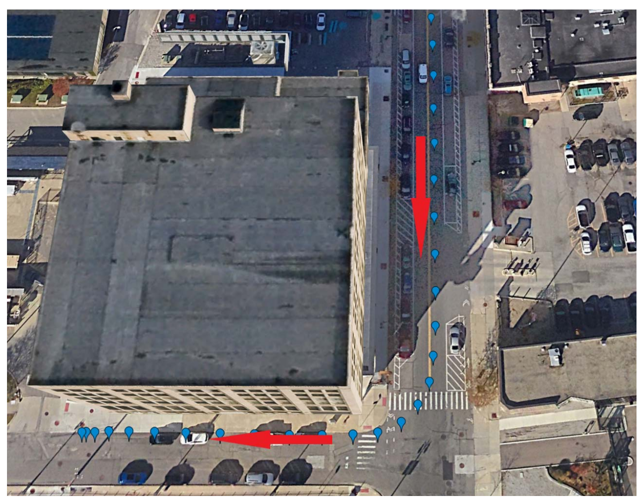

Fig. 8: Lane level Error in GPS Location in Urban Scenario.

D. Methods Comparison in Complex Urban Scenarios

In urban scenarios, the interference caused by the refraction of high-rise buildings on the GPS signal leads to errors in vehicle localization. Figure 8 shows Hydra traveling in the lane indicated by the red arrow. The GPS data collected by the vehicle's navigation system shows that the vehicle is travelling in the other lane indicated by the blue markers. This GPS data discrepancy is very common in urban cities due to tall buildings blocking the GPS signal. In Figure 8, we can also see a building in the left side of the image.

To showcase the performance of our proposed approach in an urban scenario, we compared the vehicle localization results from the GPS navigation system with the location predictions of our method.

Fig. 9: Vehicle Location (Proposed Navigation System, Ground Truth and Vehicle GPS Comparison.

Figure 9 plots the ground truth, vehicle GPS coordinates collected by Hydra's navigation system, and the coordinates generated by our proposed navigation system. The x-axis and the y-axis represent longitude and latitude, respectively. In Figure 10, we compare the performance of our method to the vehicle's GPS navigation system. We perform that by calculating the average distance between the coordinates predicted by our method and the ground truth on one hand and the average distance between coordinates collected by Hydra and the ground truth on the other hand. We repeated this process for the four different streets. By using this metric, we demonstrated that our method consistently outperforms GPS on each street. Specifically, our approach achieved a lower average distance from the ground truth than that of the vehicle's navigation system. indicating higher accuracy and better reliability than the vehicle's GPS navigation system. The results suggest that our method significantly performs better and has higher accuracy than its counterpart in an urban environment.

E. Methods Comparison in GPS Outage Scenarios

In the absence of GPS, the common approach for trajectory prediction is to rely solely on IMU data. However, this method is prone to large errors due to accumulated drift. Figure 11 depicts how IMU-only trajectory exhibits significant errors in the vehicle's position.

In contrast, our proposed method effectively addresses this issue by incorporating reference points to adjust bias and wheel slippage. In our experiments, we compared the performance of our method to the IMU-only approach. The results demonstrated the effectiveness of our method in overcoming drift errors and improving the accuracy of trajectory prediction without the use of GPS.

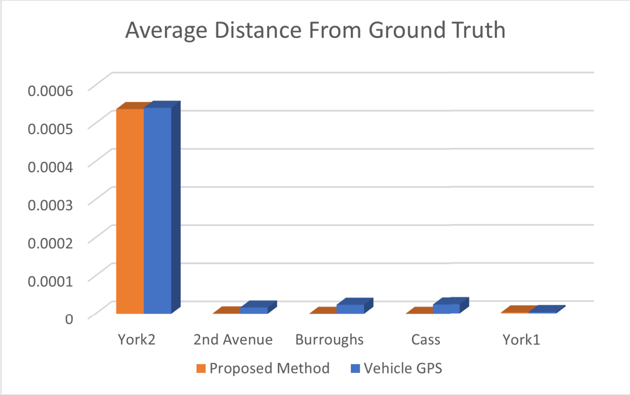

Fig. 10: Average Distance from Ground Truth.

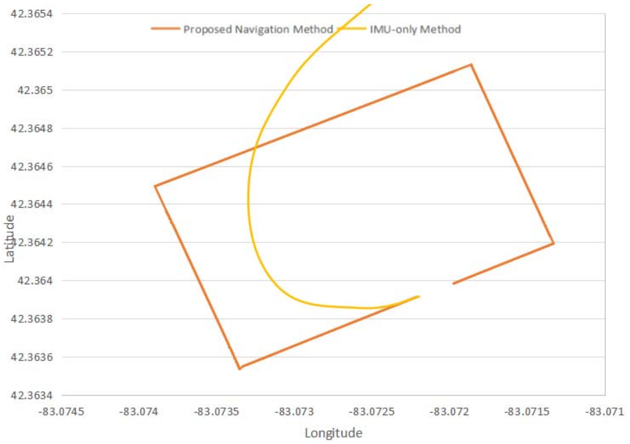

Fig. 11: Vehicle Location (Proposed Navigation System and IMU Comparison).

VII. Discussion and Future Work

We presented a position prediction method to localize a vehicle in an urban environment during intermittent GPS service or a short GPS outage. We showed that our proposed method; which incorporates steering wheel angle, odometer, and reference points; was able to achieve better location prediction than the vehicle's GPS navigation system and the IMU. However, there are still many areas that deserve further research and can be summarized as follows:

- Our proposed method is currently only suitable for vehicles adopting straight roads. In future work, we will enhance the proposed navigation method to tackle more complex road scenarios.

- Our approach requires a database to store GPS coordinates of intersections and intersection nearby points to adjust bias and calculate bearing respectively. Currently, we have tested four intersection points in a simple scenario. In the future, it will be necessary to test longer routes. Our approach relies on saved data as a basis for correcting position, and therefore, is better suited for self-driving vehicles or shuttles strictly following a predetermined route.

VIII. Conclusion

The localization of autonomous vehicles, autonomous shuttles, robots, and delivery vehicles is essential for their safety and reliability. Urban environments increase the probability of GPS signal loss. To address this problem, we proposed a navigation method that combines steering wheel angle, odometer, and reference points for accurate real-time vehicle localization without the use of sensors. We conducted test driving experiments using an OBD-II Vehicle Interface, and showed that our proposed method outperforms IMU-based vehicle localization and even surpasses GPS signal-based localization in non-complex routes. More complex routes will be the focus of future research.

References

[1] Ananda, M. P., Bernstein, H., Cunningham, K. E., Feess, W. A., & Stroud, E. G. (1990, March). Global positioning system (GPS) autonomous navigation. In IEEE symposium on position location and navigation. A decade of excellence in the navigation sciences (pp. 497-508). IEEE.

[2] Garvanov, I., & Garvanova, M. (2020, June). Improving GPS positioning of vehicles in urban environment. In 2020 21st International Symposium on Electrical Apparatus Technologies (SIELA) (pp. 1-5). IEEE.

[3] Spivak, A. A., Riabova, S. A., Rybnov, Y. S., & Kharlamov, V. A. (2021, October). GPS Positioning Errors during Periods of Magnetic Storms. In Doklady Earth Sciences (Vol. 500, pp. 852-856). Pleiades Publishing.

[4] Nie, W., Rovira-Garcia, A., Wang, Y., Zheng, D., Yan, L., & Xu, T. (2022). On the global kinematic positioning variations during the September 2017 solar flare events. Journal of Geophysical Research: Space Physics, 127(8), e2021JA030245.

[5] Taylor, M., Chen, H., Qin, F., Stewart, C. (2021, June). Avis: In-situ model checking for unmanned aerial vehicles. In 2021 51st Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN) (pp. 471-483). IEEE.

[6] Jha, S., Banerjee, S., Tsai, T., Hari, S. K., Sullivan, M. B., Kalbarczyk, Z. T., ... Iyer, R. K. (2019, June). Ml-based fault injection for autonomous vehicles: A case for bayesian fault injection. In 2019 49th annual IEEE/IFIP international conference on dependable systems and networks (DSN) (pp. 112-124). IEEE.

[7] Dash, P., Li, G., Chen, Z., Karimibiuki, M., Pattabiraman, K. (2021, June). Pid-piper: Recovering robotic vehicles from physical attacks. In 2021 51st Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN) (pp. 26-38). IEEE.

[8] Faisal, I. A., Purboyo, T. W., & Ansori, A. S. R. (2019). A Review of accelerometer sensor and gyroscope sensor in IMU sensors on motion capture. J. Eng. Appl. Sci, 15(3), 826-829.

[9] Bangera, S. S., Shiyana, T. D., Srinidhi, G. K., Vasani, Y. R., & Sukesh Rao, M. (2020). MEMS-based IMU for pose estimation. In Advances in Communication, Signal Processing, VLSI, and Embedded Systems: Select Proceedings of VSPICE 2019 (pp. 1-14). Springer Singapore.

[10] Kuang, J., Niu, X., & Chen, X. (2018). Robust pedestrian dead reckoning based on MEMS-IMU for smartphones. Sensors, 18(5), 1391.

[11] Brossard, M., Barrau, A., & Bonnabel, S. (2020). AI-IMU dead-reckoning. IEEE Transactions on Intelligent Vehicles, 5(4), 585-595.

[12] Poulose, A., & Han, D. S. (2019). Hybrid indoor localization using IMU sensors and smartphone camera. Sensors, 19(23), 5084.

[13] Nguyen, T. H., Nguyen, T. M., & Xie, L. (2021). Range-focused fusion of camera-IMU-UWB for accurate and drift-reduced localization. IEEE Robotics and Automation Letters, 6(2), 1678-1685.

[14] Fu, B., Han, F., Wang, Y., Jiao, Y., Ding, X., Tan, Q., ... & Xiong, R. (2021). High-precision multicamera-assisted camera-IMU calibration: Theory and method. IEEE Transactions on Instrumentation and Measurement, 70, 1-17.

[15] Ilci, V., & Toth, C. (2020). High definition 3D map creation using GNSS/IMU/LiDAR sensor integration to support autonomous vehicle navigation. Sensors, 20(3), 899.

[16] Han, C., Frey, M., & Gauterin, F. (2020). Modular approach for odometry localization method for vehicles with increased maneuverability. Sensors, 21(1), 79.

[17] Bouvet, D., & Garcia, G. (2000, April). Civil-engineering articulated vehicle localization: solutions to deal with GPS masking phases. In Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065) (Vol. 4, pp. 3499-3504). IEEE.

[18] Drews, F., Feng, D., Faion, F., Rosenbaum, L., Ulrich, M., & Gläser, C. (2022, October). DeepFusion: A Robust and Modular 3D Object Detector for Lidars, Cameras and Radars. In 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (pp. 560-567). IEEE.

[19] Xiao, Z., Li, P., Havyarimana, V., Hassana, G. M., Wang, D., & Li, K. (2018). GOI: A novel design for vehicle positioning and trajectory prediction under urban environments. IEEE Sensors Journal, 18(13), 5586-5594.

[20] Eren, H., Makinist, S., Akin, E., & Yilmaz, A. (2012, June). Estimating driving behavior by a smartphone. In 2012 IEEE Intelligent Vehicles Symposium (pp. 234-239). IEEE.

[21] Wahlström, J., Skog, I., & Händel, P. (2017). Smartphone-based vehicle telematics: A ten-year anniversary. IEEE Transactions on Intelligent Transportation Systems, 18(10), 2802-2825.

[22] Abdallah, R., & Shi, W. (2021, April). SafeOverPass: An Edge-Based Low-Clearance Overpass Warning System. In 2021 Fourth International Conference on Connected and Autonomous Driving (MetroCAD) (pp. 17-22). IEEE.

[23] Drosescu, R., & Zamfir, S. (2016). MEMS based device for steering wheel angle experimental measuring. In Applied engineering, materials and mechanics: Proceedings of the 2016 International Conference on Applied Engineering, Materials and Mechanics (ICAEMM 2016) (pp. 304-313).

[24] Gao, J., Petovello, M., & Cannon, M. E. (2008). Integration of steering angle sensor with global positioning system and micro-electro-mechanical systems inertial measurement unit for vehicular positioning. Journal of Intelligent Transportation Systems, 12(4), 159-167.

[25] Kubo, Y., Kindo, T., Ito, A., & Sugimoto, S. (1999, September). DGPS/INS/VVheelSensor Integration for High Accuracy Land-Vehicle Positioning. In Proceedings of the 12th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GPS 1999) (pp. 555-564).

[26] Moussa, M., Moussa, A., & El-Sheimy, N. (2019). Steering angle assisted vehicular navigation using portable devices in GNSS-denied environments. Sensors, 19(7), 1618.

[27] Liu, L., Lu, S., Zhong, R., Wu, B., Yao, Y., Zhang, Q., & Shi, W. (2020). Computing systems for autonomous driving: State of the art and challenges. IEEE Internet of Things Journal, 8(8), 6469-6486.

[28] Farroni, F., Pasquino, N., Rocca, E., & Timpone, F. (2015, July). A comparison among different methods to estimate vehicle sideslip angle. In Proceedings of the World Congress on Engineering (Vol. 2).

[29] Liu, L., Li, H., Liu, J., Karatas, C., Wang, Y., Gruteser, M., ... & Martin, R. P. (2017, June). Bigroad: Scaling road data acquisition for dependable self-driving. In Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services (pp. 371-384).