ENHANCING URBAN VEHICULAR NAVIGATION: IMPROVING CLASSICAL TOPOLOGICAL MAP MATCHING THROUGH RAY-CASTING

H. Ragab[1]*, S. Givigi[2], A. Noureldin[1,3]

[1] Dept. of Electrical and Computer Engineering, Queen's University,
Kingston, ON, Canada, K7L3N9 - (hany.ragab, nourelda)@queensu.ca
[2] School of Computing, Queen's University, Kingston, ON, Canada, K7L2N8 - sidney.givigi@queensu.ca
[3] Dept. of Electrical and Computer Engineering, Royal Military College of Canada,
Kingston, ON, Canada, K7K7B4 - aboelmagd.noureldin@rmc.ca

KEY WORDS: Urban Navigation, Sensor Fusion, Topological Map Matching, Ray-Casting.

ABSTRACT:

Navigation is of paramount importance for land vehicles as it enables efficient and accurate movement from one location to another. Whether it is for personal navigation, commercial transportation, or emergency services, reliable navigation systems play a crucial role in ensuring safety, optimizing routes, and enhancing overall operational efficiency. This paper presents the integration of classical Topological Map Matching (TMM) with the Global Navigation Satellite System (GNSS) and the Inertial Navigation System (INS), addressing the limitations of relying solely on road centerlines. A novel solution is proposed, leveraging the ray-casting algorithm to determine the predicted position's area and employing a two-stage kinematic update process for enhanced positioning accuracy. The solution's efficacy is evaluated through tests conducted on simulated GNSS outages within a road experiment conducted in the City of Toronto, demonstrating substantial improvements compared to the classical TMM approach. Notably, the proposed method achieved a considerable 82.33% reduction in RMS positioning error and a 33.71% improvement in maximum positioning error during the longest GNSS outage. By overcoming the limitations of classical TMM algorithms, this research contributes to the advancement of navigation and tracking systems, with future work focusing on practical implementations and optimization for diverse navigation scenarios.

1. INTRODUCTION

Navigating the intricacies of road traffic effectively is pivotal to the success of autonomous passenger vehicles, which span various scales of autonomy. Given that road traffic accidents are responsible for approximately $1.35 million deaths annually according to the World Health Organization, and that traffic congestion has been estimated to cost the US economy around $87 billion in 2018, it is evident that efficient and reliable navigation solutions are instrumental in optimizing our transport systems (Uhlemann, 2020). The benefits of such navigation solutions are manifold, ranging from increased passenger safety and improved route planning to environmental sustainability via reduced fuel consumption, and potentially even transformative urban design. Historically, Global Navigation Satellite Systems (GNSS) have been the favored navigation solution. However, their performance diminishes in urban environments due to obstructed sky views, necessitating the integration with Inertial Navigation Systems (INS) (Noureldin et al., 2013). Despite INS's capacity to compensate for short GNSS outages, it cannot sustain accuracy over extended periods due to the accumulation of errors over time. In response to this, researchers have sought to integrate other exteroceptive sensors such as cameras, LiDARs, and radars with INS to correct inertial measurement unit (IMU) sensor biases and position, velocity, and attitude (PVA) errors, thereby enhancing the overall navigation accuracy (Dawson et al., 2022). However, despite these advancements, odometry-based techniques remain susceptible to drift, particularly in prolonged GNSS outages. As a result, there has been a growing trend in building and utilizing maps at the topological as well as visual levels to enhance positioning accuracy and mitigate drift-related challenges. While visual mapping offers

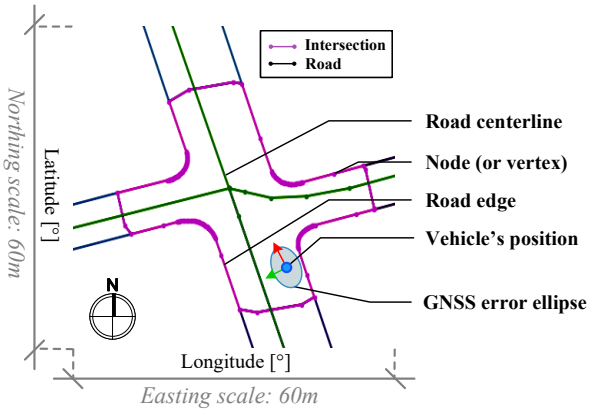


Figure 1. Topological map showcasing map elements and connectivity, integral to the art of land vehicle navigation.

* Corresponding author

a level of precision, it requires frequent updates and is memory-intensive compared to topological maps (Wei et al., 2019). Localization using HD maps has gained significant attention in the field of robot localization, offering centimeter-level accuracy in various environments while minimizing computational costs associated with full simultaneous localization and mapping (SLAM) systems (Ma et al., 2019). In fact, self-driving vehicles (SDVs) often rely on them for navigation, guidance, and control (NGC) (Liu et al., 2020). However, HD maps for cities around the world are predominantly owned by mapping companies or the automotive industry, creating challenges for collaborative map research efforts especially when dealing with different grades of navigation units, as part of our endeavor to reduce costs related to our integrated navigation solution that works in synergy with them. On the other hand, topological maps are widely available globally and offer lightweight solutions that require minimal hardware and computing resources while achieving lane-to-road level of positioning accuracy in most cases. Figure 1 illustrates a topological map, highlighting its crucial role in land vehicle navigation. Various map formats, such as shapefiles and raster maps, exist, but OpenStreetMap (OSM) has gained popularity among researchers (Kasmi et al., 2018) due to its openness and accessibility, enabling open access and unification of map research efforts. Topological map matching (TMM) encompasses a range of techniques (Chao et al., 2020), including HMM-based methods for matching GNSS measurements to road networks. The matching granularity can range from point-to-point to point-to-curve and curve-to-curve. Nevertheless, existing methods rely solely on road centerlines for matching (Cossaboom et al., 2012, Kasmi et al., 2018). Despite the abundance of road features, incorporating road edges to enhance the matching process has not been extensively explored. Moreover, only a few studies have addressed the problem of positioning using topological maps in conjunction with INS, which in itself provides uninterrupted positioning and navigation (POS/NAV) solutions. In this paper, we demonstrate the benefits of utilizing road edges to enhance classical TMM during GNSS outages. Our objective is to contribute to the development of vehicular navigation systems that achieve a harmonious blend of portability, cost efficiency, and reliable and robust performance. By doing so, we aim to make navigation technology more accessible and usable for a wide range of users. Our contributions can be summarized in the following aspects:

1. A thorough exploration of the merits and obstacles of classical topological map matching.
2. Introduction of a novel TMM paradigm, utilizing a two-stage kinematic update process.
3. Empirical validation of our proposed methodology through actual road test experiments.

The remainder of the paper is organized as follows: Section 2 dives into the methodology, first addressing the overall navigation system design, then detailing each subsystem, with an emphasis on the TMM paradigms under evaluation. Section 3 outlines the experimental setup, road test details, and displays the results along with discussions. Some concluding remarks are summarized in Section 4.

2. METHODOLOGY

2.1 System Design Choice

In this research, we utilize a simple single-antenna, low-cost, commercial-grade GNSS receiver to provide positioning, navigation, and timing (PNT) capabilities to a moving test vehicle. While dual-antenna designs have gained attention in recent navigation systems to enhance accuracy, our emphasis is on applications prioritizing mobility and simplicity, such as personal navigation. Dual-antenna GNSS systems offer improved heading estimation but require additional hardware and complexity. In contrast, single-antenna GNSS receivers provide a lightweight and straightforward solution, estimating the course over ground (COG) instead of direct heading information. For the fusion with GNSS, we choose a low-cost commercial-grade MEMS-based IMU and incorporate real-time Wheel Speed Sensors (WSS) for direct speed measurements.

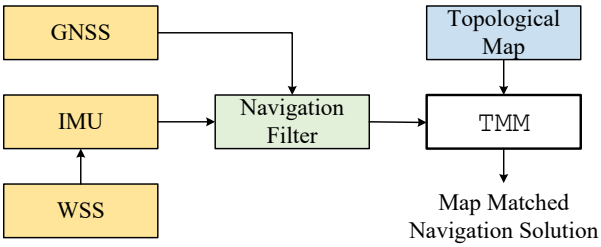


Figure 2. Navigation system architecture.

Lastly, we augment the conventional GNSS/IMU integrated navigation system with topological maps acquired through the City of Toronto Open Data Portal (City of Toronto Open Data, 2017). Figure 2 shows a block diagram of the overall system architecture. Each block will be described independently in the following sections beginning with the navigation filter design choice as it plays a central role in this architecture.

2.1.1 Navigation Filter The navigation filter utilized in this research adopts a loosely coupled (LC) integration approach, employing a combination of Global Navigation Satellite System (GNSS), Inertial Measurement Unit (IMU) data aided with an onboard WSS through an Extended Kalman Filter (EKF) integration scheme. The loosely coupled integration approach is favored over tightly coupled integration due to its ability to provide a more robust evaluation of map-matching algorithms (Cossaboom et al., 2012). Tightly coupled integration relies on a combination of GNSS and IMU measurements, where even during a partial GNSS outage, the positioning solution can still depend on two or three satellites (Georgy et al., 2010). This reliance on limited satellite data during an outage can potentially impact the accuracy of the map-matching results. In contrast, the loosely coupled integration approach does not have access to any satellite information during a GNSS outage. This lack of satellite data places greater emphasis on the performance of the map-matching algorithm itself, enabling it to play a more

significant role in enhancing the accuracy of the positioning solution. By employing a loosely coupled GNSS/IMU EKF-based integration approach, this research aims to explore the potential advantages of relying on map-matching algorithms during GNSS outages. This approach is expected to contribute to improved accuracy and reliability in navigation systems, particularly in challenging environments where GNSS signals may be compromised.

2.1.2 GNSS Measurables At time epoch k, the GNSS measurements yield latitude φ_k, longitude λ_k, and altitude h_k, eastward velocity v_{e_k}, northward velocity v_{n_k}, and upward velocity v_{u_k} components, respectively. Single-antenna GNSS receivers estimate the COG by leveraging the variations in the Doppler shift of received signals. This can be mathematically expressed using the following equation:

$$A^G_{\text{COG}_k} = \arctan\left(\frac{v_{e_k}}{v_{n_k}}\right), \tag{1}$$

where $A^G_{\text{COG}_k}$ represents the GNSS COG azimuth at epoch k. By employing this equation, the COG azimuth can be estimated based on the derived velocity components, enabling the GNSS to determine the direction of movement.

2.1.3 IMU Measurables For the IMU-based dead-reckoning solution, we opted for the use of the *reduced inertial sensor system* (RISS) as opposed to the full-IMU for its simplicity and relatively good performance, especially when considering the employment of low-cost commercial-grade MEMS-based IMUs for land vehicle navigation. To obtain a full 3D motion estimate, one gyroscope (gyro) was used to monitor the vertical axis of the vehicle while three accelerometers were used to monitor the pitch and roll rotational components. We denote the Earth's rotation rate as ω^e, the bias-compensated vertical gyro measurement as w_{z_c}, the forward and transversal accelerometer measurements as f_x and f_y, and the vehicle's forward speed measurements obtained through live readings of OBD-II messages as v_f. To accurately estimate the vehicle's position and velocity, the IMU measurements are mechanized in the local- level-frame (LLF), more specifically in the Easting, Northing, and Up (ENU) navigational frame. Equations (2) through (4) describe the calculation of the pitch (p), roll (r), and azimuth (A) at epoch k within a time interval Δt_k.

$$p_k = \arctan\left(-\frac{f_{y_k}}{\sqrt{f^2_{x_k} + f^2_{z_k}}}\right), \tag{2}$$

$$r_k = -\arctan\left(\frac{f_{x_k}}{f_{z_k}}\right), \tag{3}$$

$$A_k = A_{k-1} + \Big[-\cos(p_k)\cos(r_k)\,\omega_{z_{c_k}} + \omega_e \sin(\varphi_{k-1}) + \frac{v_{e_{k-1}}\tan(\varphi_{k-1})}{R_N + h_{k-1}}\Big]\Delta t_k. \tag{4}$$

Calculating the vehicle's attitude enables the conversion of its forward speed to ENU velocities. This involves using a rotation matrix to transform the velocity vector from the body frame (b-frame) to the ENU frame as follows:

$$\boldsymbol{v}_k = \begin{bmatrix} v_{e_k} \\ v_{n_k} \\ v_{u_k} \end{bmatrix} = \begin{bmatrix} v_{f_k}\sin(A_k)\cos(p_k) \\ v_{f_k}\cos(A_k)\cos(p_k) \\ v_{f_k}\sin(p_k) \end{bmatrix}. \tag{5}$$

It then follows the relative position calculation in geodetic space as:

$$\begin{bmatrix} \Delta\varphi_k \\ \Delta\lambda_k \\ \Delta h_k \end{bmatrix} = \begin{bmatrix} 0 & \frac{1}{R_M + h_k} & 0 \\ \frac{1}{(R_N + h_k)\cos\varphi_k} & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} v_{e_k} \\ v_{n_k} \\ v_{u_k} \end{bmatrix}, \tag{6}$$

where R_M and R_N denote the Meridian and Normal radius of curvature of the earth's ellipsoid measured in meters. The position update equations for epoch k are then:

$$\boldsymbol{r}_k = \begin{bmatrix} \varphi_k = \varphi_{k-1} + \Delta\varphi_k \\ \lambda_k = \lambda_{k-1} + \Delta\lambda_k \\ h_k = h_{k-1} + \Delta h_k \end{bmatrix}. \tag{7}$$

2.1.4 GNSS/IMU Integration When GNSS is available, the GNSS/IMU integration model takes advantage of the additional data. It calculates the difference between the predicted RISS position and velocity (PV) state estimates and the actual PV state estimates obtained from the GNSS. This difference is used to improve the model's accuracy. In the absence of GNSS data, however, the RISS continues to operate in prediction mode, using the latest EKF-estimated bias and PVA corrections made prior to the GNSS outage.

2.2 Map Matching Algorithms

Map matching is a fundamental task in navigation systems, where the goal is to find the most likely path or road segment that corresponds to a given set of GNSS coordinates. In multi-sensor fusion positioning and navigation frameworks, it is expected that the integrity of each aiding system is reported to the navigation filter in order to either weight the filter's measurement covariances or discard measurements that may result in unreliable integrated navigation solutions (Noureldin and Elhabiby, 2023). In this section, we describe the topological map matching paradigms that are evaluated further in the paper all taking into account that the initial GNSS coordinate is accurate enough for integration with topological maps. The estimation of the initial GNSS position marking the beginning of a GNSS outage is beyond the scope of this paper and readers are encouraged to read the work of (Lee et al., 2021), which tackles such a problem via the infrastructure.

2.2.1 Classical Topological Map Matching Despite the existence of numerous variants and approaches to topological map matching in the literature, there has been no explicit definition of what constitutes the *classical* topological map matching procedure. We believe it is essential to have a benchmark or a generic representation, which we refer to as the *classical* algorithm. This sets the groundwork upon which refinements and advancements can be developed. Thus, we define *classical* topological map matching as the generic algorithm that associates a

given GNSS point with the most probable point on a road segment, based on positional and directional data. The first step of such a process is the loading of the map. For the *classical* approach, only the road centerlines are considered in the matching process.

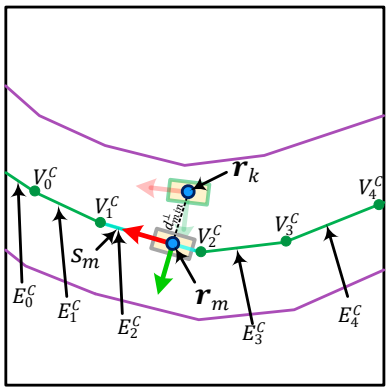


Figure 3. A visual representation of the TMM_P paradigm (not to scale).

We denote the set $\mathcal{C}$ contained within a bounding box with a predetermined size around a query GNSS coordinate $\boldsymbol{r}_k$ with direction A_k to match against. The map consists of elements $\boldsymbol{V^C}$, $\boldsymbol{E^C}$ representing vertices and edges of the road centerlines, respectively. Each segment s corresponds to the i-th edge, E_i^C, within the set $\boldsymbol{E^C}$, and is distinguished by its azimuth α_s (in degrees) and a unique identifier ID_s (integer). The steps of the algorithm are described in Algorithm 1 within a routine process called TMM_P. Figure 3 displays the TMM_P paradigm, showcasing road centerline elements as defined. A fundamental aspect of this algorithm is the process of associating a query GNSS point to a specific point on a road segment. This association is accomplished by considering the geographical coordinates and direction of the GNSS point, as well as the topology of the road network. For the sake of efficiency, especially when working with large-scale road networks, only the road data within a $\pm 30m$ bounding box around the query GNSS point is loaded in each epoch. This localized approach significantly reduces the computational overhead by limiting the road segments that need to be considered during each iteration. The algorithm systematically scans all the road segments within this subset, computing the perpendicular distance $d^{\perp}$ from the query GNSS point to each segment, as well as the difference in direction (azimuth) between the GNSS point and each road segment (line 5-10). The road segment with the smallest perpendicular distance and directional difference provided the direction difference does not exceed a certain threshold α_{th}, is selected as the matched segment s_m (line 11). The GNSS point is then associated with the corresponding point $\boldsymbol{r}_m$ on this matched segment (line 12-14). The process described, although straightforward, can encounter several challenges when applied to real-world scenarios. Key among these challenges is handling complex road intersections. In these situations, multiple road segments may present near-identical perpendicular distances from the query GNSS point, complicating the task of definitively identifying the correct match. Furthermore, after a turn or change in direction, there can be a delay before the moving vehicle's azimuth (heading direction) aligns with the correct road segment, which introduces further complexity to the map-matching process.

Algorithm 1 Generic TMM algorithm with a position-only update based on road centerlines

input: $\mathcal{C} \supseteq \{\boldsymbol{V^C}, \boldsymbol{E^C}, \alpha, \text{ID}\}$, $(\lambda_k, \varphi_k, A_k)$, α_{th}
output: s_m, $\boldsymbol{r}_m$, $d_{min}^{\perp}$

1: **procedure** TMM_P($\mathcal{C}$, $\boldsymbol{r}_k$, A_k, α_{th})
2: $s_m \leftarrow NaN$ ▷ Matched segment ID
3: $\boldsymbol{r}_m \leftarrow (0,0,0)$ ▷ Matched position
4: $d_{min}^{\perp} \leftarrow \infty$ ▷ Matched position residual
5: **for** each segment $s \in \boldsymbol{E^C}$ with endpoints $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$ where $\boldsymbol{r}_1 = (\lambda_1, \varphi_1, h_1)$ and $\boldsymbol{r}_2 = (\lambda_2, \varphi_2, h_2)$, and associated α_s and ID_s **do**
6: $u = \frac{(\varphi_k - \varphi_1)(\varphi_2 - \varphi_1) + (\lambda_k - \lambda_1)(\lambda_2 - \lambda_1)}{\|\boldsymbol{r}_2 - \boldsymbol{r}_1\|^2}$
7: $u = \min(\max(u, 0), 1)$
8: $(\lambda_{\perp}, \varphi_{\perp}, h_{\perp}) = \boldsymbol{r}_1 + u(\boldsymbol{r}_2 - \boldsymbol{r}_1)$
9: $d^{\perp} = \sqrt{(\lambda_{\perp} - \lambda_k)^2 + (\varphi_{\perp} - \varphi_k)^2}$
10: $\Delta\alpha = \min(\min(|A_k - \alpha_s|, 360 - |A_k - \alpha_s|), \min(|A_k - (\alpha_s + 180)|, 360 - |A_k - (\alpha_s + 180)|))$
11: **if** $d^{\perp} < d_{min}^{\perp}$ and $\Delta\alpha < \alpha_{th}$ **then**
12: $d_{min}^{\perp} \leftarrow d^{\perp}$
13: $s_m \leftarrow \text{ID}_s$
14: $\boldsymbol{r}_m \leftarrow (\lambda_{\perp}, \varphi_{\perp}, h_{\perp})$
15: **end if**
16: **end for**
17: **return** s_m, $\boldsymbol{r}_m$, $d_{min}^{\perp}$
18: **end procedure**

2.2.2 Two-stage Kinematic Update TMM Topological maps of cities around the world are constantly evolving to include more features including but not limited to location and attributes of road edges, traffic lights, and so much more. This ongoing initiative aims to benefit public works, transportation, and related entities. In this study, we propose utilizing road edges and their associated area attributes stored in the map to determine whether a predicted position falls within an intersection. We define $\mathcal{P}$ as the set of polygons representing road edges, each accompanied by its attribute *subtype* that indicates whether it corresponds to a road segment or an intersection. In mathematical terms, $\mathcal{P} \supseteq \{\boldsymbol{V^P}, \boldsymbol{E^P}, subtype\}$. To identify the specific polygon containing a given point, we employ the ray-casting algorithm. The ray-casting algorithm is a method used to determine if a given point is inside or outside a polygon. Figure 4 illustrates an example of applying the ray-casting algorithm in the context of topological maps. In the figure, we have multiple locations of a vehicle represented by points scattered across the map, and various polygons that define the road edges and intersections. The purple-highlighted polygon represents an intersection. The algorithm starts by casting a ray from the vehicle's position, as depicted by the black arrows in the figure. It then counts the number of intersections between the ray and the edges of the selected polygon. In this case, as the ray in-

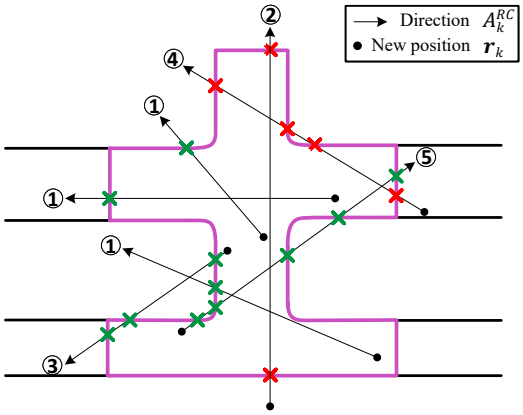


Figure 4. Ray-casting: Solving the point-in-polygon (PIP) problem for intersection checks in topological maps.

tersects the edges of the purple polygon an odd number of times, it indicates that the vehicle's position lies within the intersection. Conversely, if the ray intersects the edges of a different polygon an even number of times, the algorithm determines that the vehicle's position lies outside of that polygon. By leveraging the parity of the intersection count, the ray-casting algorithm provides an efficient way to determine whether a given point is inside a polygon. This valuable information enables us to make informed decisions on whether to utilize TMM updates or not. Additionally, the GNSS/IMU integration offers the advantage of uninterrupted and relatively accurate PVA estimates, particularly when integrating the vehicle's proprietary WSS readings. These benefits have led to the development of the proposed algorithm as illustrated in Algorithm 2, `TMM_Az`, which incorporates a two-stage kinematic update process (line 26 and line 1-7). Firstly, a prediction is generated through dead-reckoning, and then a correction is applied using TMM, but only if the predicted position falls within a road segment rather than an intersection (i.e., $\Phi = 0$ in Algorithm 2). Unlike the classical `TMM_P`, where the predicted position is snapped to the road centerline, `TMM_Az` outputs the matched azimuth A_m, which is subsequently used in a second stage of kinematic update to obtain the matched position $\boldsymbol{r}_m$. Figure 5 shows a visual representation of the proposed TMM paradigm applied at the beginning of a GNSS outage. It is important to note that the `TMM_Az` procedure assumes the initial position (representing the start of a GNSS outage) to be well within the road. The navigation framework is responsible for timing the switch to ensure uninterrupted and reliable positioning until the GNSS signal is recovered.

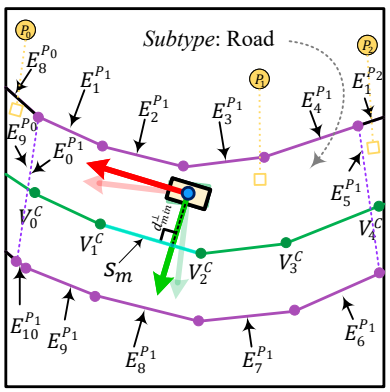


Figure 5. A visual representation of the `TMM_Az` paradigm (not to scale).

Algorithm 2 Two-Stage Kinematic Update TMM

1: **function** KINEMATICUPDATE($\boldsymbol{r}_{k-1}$, h_k, v_f, A_m, Δt_k)
2: $\Delta\varphi = \frac{v_f \cdot \sin(A_m) \cdot \Delta t_k}{R_M + h_k}$ ▷ Longitude update
3: $\Delta\lambda = \frac{v_f \cdot \cos(A_m) \cdot \Delta t_k}{(R_N + h_k) \cdot \cos(\varphi_{k-1})}$ ▷ Latitude update
4: $\varphi_m = \varphi_{k-1} + \Delta\varphi$ ▷ Longitude position update
5: $\lambda_m = \lambda_{k-1} + \Delta\lambda$ ▷ Latitude position update
6: **return** $(\lambda_m, \varphi_m, h_k)$
7: **end function**
8: **procedure** `TMM_Az`($\boldsymbol{\mathcal{C}}, \boldsymbol{\mathcal{P}}, \boldsymbol{r}_k, \boldsymbol{r}_{k-1}, A_k, \Delta t_k, \alpha_{th}$)
9: $\Phi \leftarrow$ RAYCASTING($\boldsymbol{r}_k, \boldsymbol{\mathcal{P}}$) ▷ Intersection flag
10: **if** $\Phi = 0$ **then**
11: $s_m \leftarrow NaN$ ▷ Matched segment ID
12: $A_m \leftarrow 0$ ▷ Matched heading/azimuth
13: $d^{\perp}_{min} \leftarrow \infty$ ▷ Matched position residual
14: **for** each segment $s \in \boldsymbol{E^C}$ with endpoints $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$ where $\boldsymbol{r}_1 = (\lambda_1, \varphi_1, h_1)$ and $\boldsymbol{r}_2 = (\lambda_2, \varphi_2, h_2)$, and associated α_s and ID_s **do**
15: $u = \frac{(\varphi_k - \varphi_1)(\varphi_2 - \varphi_1) + (\lambda_k - \lambda_1)(\lambda_2 - \lambda_1)}{\|\boldsymbol{r}_2 - \boldsymbol{r}_1\|^2}$
16: $u = \min(\max(u, 0), 1)$
17: $(\lambda_\perp, \varphi_\perp, h_\perp) = \boldsymbol{r}_1 + u(\boldsymbol{r}_2 - \boldsymbol{r}_1)$
18: $d^{\perp} = \sqrt{(\lambda_\perp - \lambda_k)^2 + (\varphi_\perp - \varphi_k)^2}$
19: $\Delta\alpha = \min(\min(|A_k - \alpha_s|, 360 - |A_k - \alpha_s|), \min(|A_k - (\alpha_s + 180)|, 360 - |A_k - (\alpha_s + 180)|))$
20: **if** $d^{\perp} < d^{\perp}_{min}$ and $\Delta\alpha < \alpha_{th}$ **then**
21: $d^{\perp}_{min} \leftarrow d^{\perp}$
22: $s_m \leftarrow \mathrm{ID}_s$
23: $A_m \leftarrow \alpha_s$
24: **end if**
25: **end for**
26: $\boldsymbol{r}_m \leftarrow$ KINEMATICUPDATE($\boldsymbol{r}_{k-1}, h_k, v_f, A_m, \Delta t_k$)
27: **else**
28: $\boldsymbol{r}_m \leftarrow \boldsymbol{r}_k$
29: **end if**
30: **return** $s_m, \boldsymbol{r}_m, A_m, d^{\perp}_{min}$
31: **end procedure**

3. EXPERIMENTAL SETUP AND RESULTS

In this section, we present the equipment used and describe the road tests conducted to evaluate the effectiveness of the TMM algorithms. The tests were performed on a $13.8km$ long trajectory within the City of Toronto, covering downtown, suburban, and rural driving areas. Figure 6 displays the complete trajectory with overlaid topological map features on a satellite image. We provide a detailed discussion of the results obtained from the augmented `TMM_Az`-GNSS/RISS integration method, comparing them with the outcomes of the classical `TMM_P`-GNSS/RISS and the traditional GNSS/RISS integration.

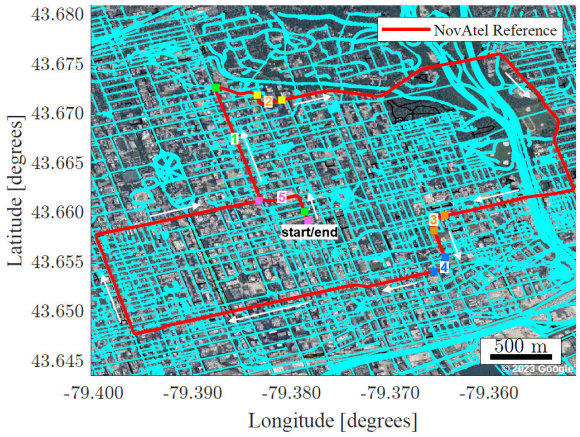


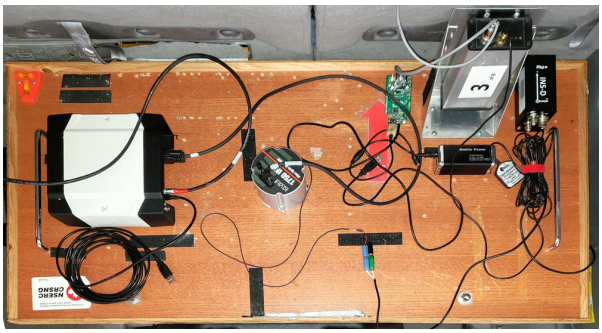
Figure 6. Toronto trajectory and topological map data overlaid on a satellite image generated using the © Google Maps API.

3.1 Experimental Setup

To test our algorithms, multiple road test trajectories were performed using a mini van equipped with a multi-sensor suite for positioning and navigation. Figure 7 shows the sensors as they were placed inside and around the vehicle.

(a) Mini van equipped with a multi-sensor suite.

(b) High-end (left) and low-cost (right) GNSS antennas.



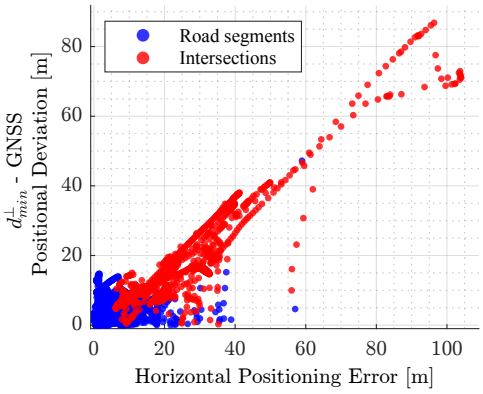
(c) Testbed with different grades of IMUs and GNSS receivers.

Figure 7. Experimental setup for advancing land vehicle positioning and navigation research and development.

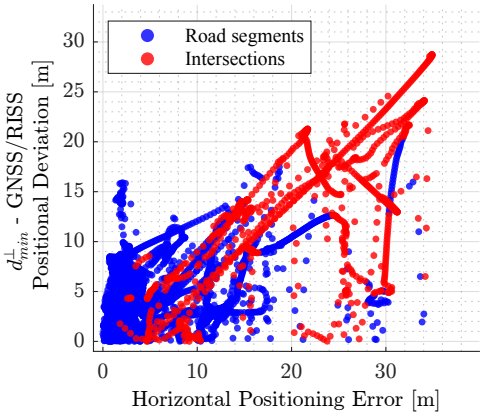
The navigation system used to validate our positioning and navigation solutions comprises a high-end NovAtel triple-frequency GNSS ProPak6^{TM} integrated with the tactical-grade FOG-based KVH1750 IMU through NovAtel's SPAN® technology. This advanced navigation system played a crucial role in verifying the accuracy and performance of our solutions especially when dealing with GNSS challenging environments such as downtown Toronto. As for the low-cost solutions, a simple u-blox LEA-6 GNSS unit was used (placed as shown in Figure 7b) along with the MEMS-based VTI-IMU SCC1300-D04 (housed within the grey box, illustrated in Figure 7c).

3.2 Results and Discussions

To begin, we conducted an initial analysis by running the standalone low-cost GNSS solution through the classical TMM algorithm (referred to as `TMM_P`). Additionally, we compared the resulting positioning residual $d_{min}^{\perp}$ versus the actual horizontal positioning error. We then employed the ray-casting algorithm to classify each position obtained from the reference NovAtel unit as either within the road or an intersection. By applying this classification to both the standalone GNSS solution and its integration with the inertial sensor system, we gained insights into the impact of the area type on the accuracy of the classical TMM. The findings are visualized in Figure 8, indicating that the area type (road or intersection) significantly affects the accuracy of the classical TMM. Notably, a substantial decrease in positioning accuracy was observed in intersections. It is worth noting that the directional difference tolerance (α_{th}) was set at 20° for the standalone GNSS and LC integrated solutions. In the standalone approach, the matching process relied on the COG azimuth, while the integrated solution determined the azimuth through the mechanization process.



(a) GNSS



(b) GNSS/RISS

Figure 8. Discrepancies between estimated positions obtained via classical map matching approach `TMM_P` and the true horizontal positioning error.

To provide quantitative results, Table 1 presents the horizontal positioning error statistics of the TMM_P paradigm when applied on each navigation system, illustrating the effect of the area type on the end map-matching accuracy.

NAV System	Polygon *subtype*	Avg	Max	Std
GNSS	Intersections	20.72	84.74	16.09
	Road	3.60	46.35	3.09
	Any	6.65	86.86	9.83
GNSS/RISS	Intersections	12.57	28.73	7.09
	Road	4.40	19.91	3.30
	Any	5.87	34.79	5.27

Table 1. Horizontal positioning error (in meters) of TMM_P when applied on standalone GNSS and on integrated GNSS/RISS.

For the map-matched standalone GNSS system, higher positioning error was exhibited at intersections, averaging $20.72m$, compared to $3.60m$ on road segments. On the other hand, the map-matched integrated GNSS/RISS navigation system demonstrated improved performance, achieving lower error values at both intersections and overall. These findings highlight the impact of area type on the accuracy of the TMM algorithm, with intersections posing a challenge in achieving reliable positioning. To address this issue, we introduced the two-stage kinematic update TMM_Az algorithm and compared it against the classical TMM_P algorithm and the traditional GNSS/RISS integrated navigation solution. We evaluated these methods in five distinct GNSS outages, ranging from approximately $\sim 30sec$ to $\sim 6.5mins$ in duration. Outages are shown in Figure 6, delimited by colored squares. It is worth noting that the vehicle's starting and ending positions in the trajectory coincide. Table 2 presents a comparative analysis of the RMSE performance for the positioning solutions during simulated GNSS outages along the Toronto trajectory.

RMSE [m]	G/R	TMM_P-G/R	TMM_Az-G/R
Outage 1 (380 sec)	35.20	18.5989	**15.6123**
Outage 2 (55 sec)	3.694	4.26665	**2.94331**
Outage 3 (31 sec)	**1.2896**	8.82372	1.55655
Outage 4 (32 sec)	1.3923	3.91953	**1.27334**
Outage 5 (110 sec)	11.693	6.11657	**3.28319**

Table 2. Comparative analysis of RMS Error performance within simulated GNSS outages along the Toronto trajectory. Note: G (GNSS) and R (RISS).

During these GNSS outages, the TMM_Az algorithm consistently outperforms the other methods, demonstrating improved positioning accuracy, particularly in complex maneuvers. For example, during Outage 1, TMM_Az achieved an RMSE of $15.61m$, resulting in a percentage improvement of 15.69% compared to TMM_P and 55.59% compared to RISS. Similarly, in Outage 2, TMM_Az achieved an RMSE of $2.94m$, resulting in a percentage improvement of 31.06% compared to TMM_P and 20.30% compared to RISS. During Outage 3, there was an improvement of 82.33% observed with respect to TMM_P. However, no significant difference was observed compared to RISS. This can be attributed to the outage being limited to a brief turn, and the inertial sensor biases and PVA errors were effectively reset at the start of the outage. Figure 9 provides

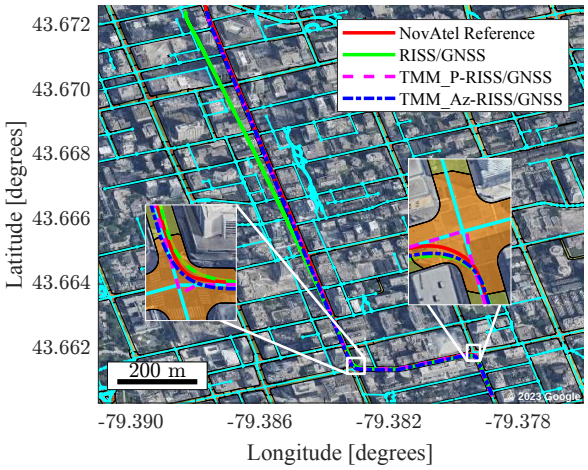


Figure 9. Positioning performance in Outage 1.

a visual representation of the positioning and navigation performance, focusing on two distinct intersections where the classical TMM algorithm exhibits degraded accuracy compared to the proposed TMM paradigm. To distinguish between the road areas, intersections were shaded in orange, while road polygons were presented in a green hue. The horizontal positioning error plot, depicted in Figure 10, reflects the performance, with enclosed peaks indicating increased errors within intersections. Additionally, the

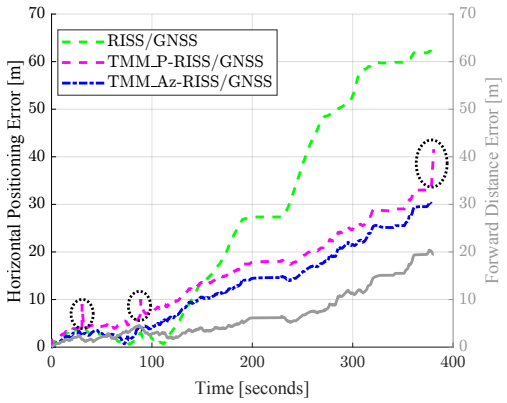


Figure 10. Horizontal positioning error for Outage 1.

accumulation of WSS-derived forward distance (i.e, along-track distance) error over time is presented to illustrate its impact on the solutions, as it was employed by all the algorithms. In terms of maximum error performance, the classical TMM algorithm resulted in a maximum error of $41.99m$, whereas the proposed TMM algorithm achieved a maximum error of $30.42m$, signifying an improvement of 33.71% during this relatively long GNSS outage.

4. CONCLUSIONS

This paper introduces classical topological map matching and its integration benefits with GNSS and inertial navigation systems. It addresses the limitation of relying solely

on road centerlines in map-matching algorithms. Our proposed solution utilizes the ray-casting algorithm to determine the predicted position's area and employs a two-stage kinematic update process for improved positioning. We conducted tests on five simulated GNSS outages, demonstrating significant improvements in accuracy compared to the classical approach. Our method achieved an 82.33% reduction in RMS positioning error and a 33.71% improvement in maximum positioning error during a GNSS outage lasting around 6.5 minutes. Moreover, the system's effectiveness scales with an increase in the number of turns and distances, particularly during GNSS outages. However, our method depends on automotive WSS measurements, which accumulate errors over time. This cumulative error along with the drift caused by the gyro bias can adversely affect the system during prolonged GNSS outages, potentially causing an erroneous predicted position outside the road boundary, encouraging the need to correct position in a single kinematic update stage. Despite these challenges, our approach has demonstrated promising results. As part of our future work, we plan to bolster the system's capabilities by integrating perception systems, such as cameras. This would enable accounting for lane changes, provide longitudinal position updates through visual map matching, and allow for further monitoring of vehicle dynamics. Lastly, we foresee the potential of our approach's usage of road edges - as opposed to merely centerlines - in augmenting particle filtering algorithms, potentially delivering substantial improvements in overall navigation performance.

REFERENCES

Chao, P., Xu, Y., Hua, W. and Zhou, X., 2020. A Survey on Map-Matching Algorithms. In: R. Borovica-Gajic, J. Qi and W. Wang (eds), Databases Theory and Applications, Vol. 12008, Springer International Publishing, Cham, pp. 121–133. Series Title: Lecture Notes in Computer Science.

City of Toronto Open Data, 2017. Archive Location: Toronto, Ontario, Canada. https://www.toronto.ca/city-government/data-research-maps/open-data/ (10 July 2023).

Cossaboom, M., Georgy, J., Karamat, T. and Noureldin, A., 2012. Augmented Kalman Filter and Map Matching for 3D RISS/GPS Integration for Land Vehicles. International Journal of Navigation and Observation vol. 2012, pp. Article ID 576807, 16 pages.

Dawson, E., Rashed, M. A., Abdelfatah, W. and Noureldin, A., 2022. Radar-Based Multisensor Fusion for Uninterrupted Reliable Positioning in GNSS-Denied Environments. IEEE Transactions on Intelligent Transportation Systems 23(12), pp. 23384–23398.

Georgy, J., Noureldin, A., Syed, Z. and Goodall, C., 2010. Nonlinear filtering for tightly coupled RISS/GPS integration. In: IEEE/ION Position, Location and Navigation Symposium, pp. 1014–1021. ISSN: 2153-3598.

Kasmi, A., Denis, D., Aufrere, R. and Chapuis, R., 2018. Map Matching and Lanes Number Estimation with Openstreetmap. In: 2018 21st International Conference on Intelligent Transportation Systems (ITSC), pp. 2659–2664. ISSN: 2153-0017.

Lee, E. S., Vora, A., Parchami, A., Chakravarty, P., Pandey, G. and Kumar, V., 2021. Infrastructure Node-based Vehicle Localization for Autonomous Driving. arXiv:2109.10457.

Liu, R., Wang, J. and Zhang, B., 2020. High Definition Map for Automated Driving: Overview and Analysis. Journal of Navigation 73(2), pp. 324–341.

Ma, W.-C., Tartavull, I., Bârsan, I. A., Wang, S., Bai, M., Mattyus, G., Homayounfar, N., Lakshmikanth, S. K., Pokrovsky, A. and Urtasun, R., 2019. Exploiting Sparse Semantic HD Maps for Self-Driving Vehicle Localization. In: 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 5304–5311. ISSN: 2153-0866.

Noureldin, A. and Elhabiby, M., 2023. A Framework for Multi-Sensor Positioning and Mapping for Autonomous Vehicles. In: The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. XLVIII-1-W1-2023, pp. 339–343. ISPRS TC I (WG I/2), 12th International Symposium on Mobile Mapping Technology (MMT 2023), 24-26 May 2023, Padua, Italy.

Noureldin, A., Karamat, T. B. and Georgy, J., 2013. Fundamentals of Inertial Navigation, Satellite-based Positioning and their Integration. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-30466-8.

Uhlemann, E., 2020. Every Effort Toward Traffic Safety Counts [Connected and Automated Vehicles]. IEEE Vehicular Technology Magazine 15(2), pp. 144–148.

Wei, X., Barsan, I. A., Wang, S., Martinez, J. and Urtasun, R., 2019. Learning to Localize Through Compressed Binary Maps. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Long Beach, CA, USA, pp. 10308–10316.