Adaptive Global Navigation for Sidewalk Robots: Integrating OSM Lanelets with Real-Time Pavement Segmentation-Based Correction

Abhishek Nannuri[1], Mohamed Wael Mohamed Lotfy Abdeltawab[2], Swaraj Tendulkar[1], Frank Schrödel[1]

Abstract—**This paper presents a novel approach for global path generation and waypoint correction for autonomous navigation of mobile robots in outdoor environments, with a focus on last-mile delivery. The study evaluates a test route from a set Start Location to two destinations: Hochschule Library (Goal Location 1) and the Hochschule Building H (Goal Location 2). The route includes mixed environments such as pavements and open spaces, which pose unique challenges for navigation. To address these, the paper extends the Lanelet2 framework commonly used for autonomous vehicles on roadways to pedestrian-shared spaces such as sidewalks. The OpenStreetMap (OSM) data is manually annotated according to the Lanelet2 framework to create global waypoints. While the original map data uses geographic coordinates (latitude/longitude) in decimal degrees (DD) format, this implementation converts these to a local metric coordinate system using the appropriate Universal Transverse Mercator (UTM) Zone. The paper also tackles the challenge posed due to the spatial constraints and precision required for navigation on pavements due to road boundaries. To tackle this challenge, the paper introduces a real-time vision-based correction mechanism. Using a depth camera, the sidewalks are semantically segmented to identify pavement boundaries and extract corresponding center points. This local perception is then used to dynamically correct the globally generated Lanelet waypoints, enabling safe and accurate robot navigation in constrained pedestrian environments like sidewalks. This paper provides the basis to scale the solution further to annotate pavements for entire cities through OSM maps enabling last mile global navigation for Autonomous Mobile Robots (AMR).**

I. Introduction

Autonomous vehicle (AV) navigation approaches span a spectrum from purely vision-based systems to those heavily dependent on high-definition maps. Companies like Tesla primarily leverage a vision-centric approach with neural networks processing camera data and minimal map dependencies, though currently still requiring human supervision [1]. On the other hand, companies like Waymo utilize detailed HD maps as a fundamental component of their navigation stack [2]. Among HD map frameworks, LibLanelet is considered as an influential structured map representation framework and laid foundation for modern HD maps. Lanelets are becoming increasingly popular since their definition is more lightweight compared to e.g., OpenDRIVE, yet powerful enough to fulfill all major needs in driving simulators and automated driving. A significant demonstration of this technology was the autonomous drive along the Bertha Benz memorial route [3]. Lanelet2, an improved version of the original LibLanelet framework, was introduced in 2018 and has since been widely adopted and applied in various academic and industrial projects [4]. Support has been demonstrated for many applications such as semantic localization, HD map creation and behavior planning [5].

Autonomous Mobile Robots (AMR) used for last-mile deliveries operate with a distinct Operational Design Domain (ODD) as compared to AVs. Global robot navigation can be divided into two main approaches: map-based and mapless. Map-based approaches rely on a comprehensive cost map for path planning [6], [7] and precise robot localization [8], [9] to ensure accurate path execution. However, these methods can be computationally expensive and require significant overhead to maintain up-to-date maps. To address these limitations, ViNT [10] and NoMaD [11] proposed generating topological maps and using vision-based images as subgoals for navigation, though they still require initial runs to collect subgoal images. On the other hand, mapless navigation techniques eliminate the need for maps entirely. AdaptiveON [12] focused on generating actions in a mapless fashion but was limited to local planning without addressing long-distance navigation. More recent advancements, such as MTG [13], enable long-distance navigation, while DTG [14] optimizes traversable trajectories in large-scale outdoor settings. In [15], a GNSS sensor was used to create a reference path on the sidewalk for the AMR. Most of the current approaches require test runs beforehand with onboard GPS/GNSS systems to generate the reference waypoints on pavements which is not feasible if AMR operation is to be scaled for cities. Taking inspiration from existing navigation approaches, this paper customizes an open-source map framework like Lanelet2, originally designed for AVs, to the AMR's ODD. This solution can be further scaled to enable global path generation that covers entire cities.

*We gratefully acknowledge the financial support for this research by the Thüringer Ministeriums für Bildung, Wissenschaft und Kultur for the project Hochschulübergreifende Forschungsgruppe Vernetztes und Kognitives Fahren.

Abhishek Nannuri[1], Swaraj Tendulkar[1] and Frank Schrödel[1] are with the Faculty of Mechanical Engineering, Schmalkalden University of Applied Sciences, 98574 Schmalkalden, Germany.

Mohamed Wael Mohamed Lotfy Abdeltawab[2] is with Faculty of Engineering and Material Science, German University in Cairo, New Cairo 1, Cairo Governorate, Egypt.

Unlike vehicles, which can tolerate shifts on the road, even minor offsets for delivery robots navigating on pavements can lead to potential hazards [15] i.e. AMR crossing pavement boundaries and falling onto the roads leading to accidents with vehicles. Hence, for precise AMR navigation on sidewalks, a segmentation based waypoint correction method is implemented as an additional safety mechanism.

II. Methodology

This section considers a 4 Wheel-Drive AMR and provides details of the test environment considered for this paper. Based on the test environment, a global navigation strategy based on Lanelet framework is implemented in the following subsection. The combination of IMU and encoders is used to localize the robot and track its movement in the local frame of reference. The robot aims to operate from the Start Location to the Hochschule campus as shown in Fig. 1. In this test track, there are two kinds of operational areas which the robot encounters i.e. Sidewalk/Pavement and Hochschule/University Campus. Initially, the AMR travels on the pavement which is denoted by the bold black line where a dedicated space and direction of movement for the robot is available. At the same time, it has limited space of movement due to the presence of roadways adjoining the pavement. Following the pavement, the AMR enters the Hochschule Schmalkalden campus, where plenty of operational space is available but has a lack of dedicated spaces of movement for the AMR. This is denoted by black dotted lines. This paper considers the AMR to be the only entity operating on the pavements since the paper tackles global path planning for AMRs in outdoor environments. This approach is a part of a 3 layered environment model presented in [19] which is inspired from the HD map structure developed for AVs. This model defines where further complexities like localization, dynamic obstacles detection/movement prediction [16], etc. will be tackled.

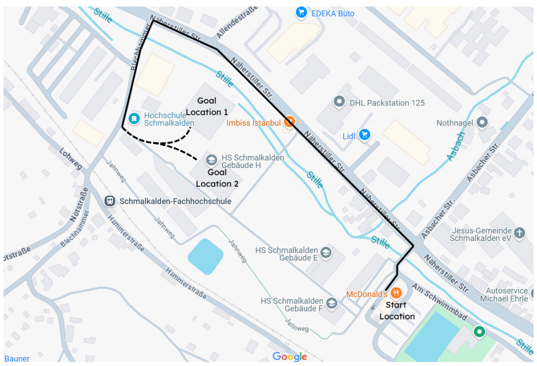

Fig. 1. Test Environment

A. Lanelet Based Global Waypoint Generation

For the implementation, OSM map data for the test environment is selected in the Java OpenStreetMap (JOSM) editor tool along with Lanelet2 presets provided by the Lanelet2 framework repository [17]. OpenStreetMap (OSM) maps compared to HD maps offer free, scalable, and quickly updateable global coverage with lower computational demands, making them well suited for many robotics and last-mile delivery applications [18]. The rules were customized with parameters relevant to AMRs such as speed limits and traversability factors and additional tags in the XML-based .osm map to identify crowded areas or specific Lanelets. These custom traffic rules and usage restrictions are then used by the trajectory planner to determine optimal and compliant paths. The test environment with Lanelet annotations is shown in Fig. 2. These customizations enable the global path planner to select from a number of paths between the start and goal location and consider factors beyond distance such as avoiding high population density areas while still optimizing traversal time which is a critical consideration for last-mile delivery applications. For Global Path Planner, OSM maps [20] were used whereas the JOSM tool [21] was used to annotate the maps in .osm format.

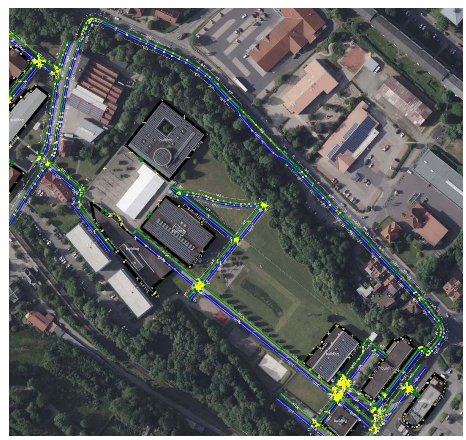

Fig. 2. Annotated HD Map Network

Fig. 2 illustrates the test environment annotated in the .osm format. The Lanelet2 framework was extended for the AMR's ODD to be operated on pavements and free spaces like college campus. Other lanelet2 parameters [17] like direction of travel, velocity limits etc. are also considered and are described in [19]. By using the AMR's current location (Start Location) from GPS as start and user prompted GPS Goal Locations 1 and 2 as inputs to the global path planner, a list of connections (Lanelet, areas or both) based on shortest distance is obtained. For this paper, the annotated test track selected is illustrated in Fig. 3. Throughout the implementation, UTM32 was established as the reference coordinate frame and converted all data into this common frame, including onboard GPS sensor data and data from the Lanelet map. This unified coordinate system ensures consistency across different data sources and processing modules. The test environment consists of both pavements and areas (university campus)—two fundamental primitives of the Lanelet2 framework as shown in Fig. 3. While Lanelet2 provides built-in functionality to extract centerline points

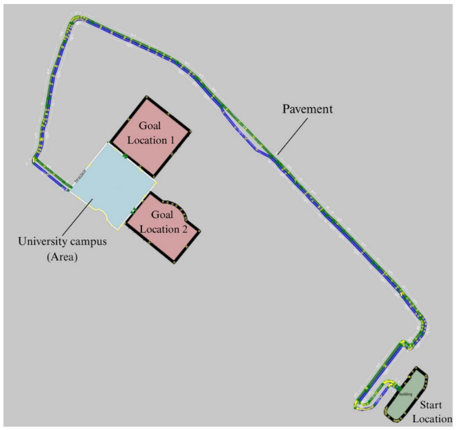

Fig. 3. Annotated Test Track

from Lanelets annotated on pavements when no custom centerline is annotated in the HD map, areas like campus present a unique challenge as they lack defined geometry conducive to automatic centerline extraction. This centerline extraction for Lanelets follows principles similar to those described by Yin et al. [22], who proposed efficient centerline extraction based on surveyed boundaries. To address the challenge of navigating through both Lanelets annotated on pavements and areas, a comprehensive centerline preprocessing approach using quintic Bezier curves was developed. Fig. 4 illustrates the results of the centerline preprocessing approach.

Fig. 4. Results of Centerline Preprocessing using Quintic Bezier Curves

Quintic (5th-order) Bézier curves were selected for centerline smoothing after evaluating several parametric curve families. Degree-5 provides C^2 continuity (up to curvature) at segment joints with six control points, which is the minimum order sufficient for jerk-free trajectories. The windowed Bézier formulation also ensures strict segment locality, unlike B-splines where basis function support can propagate annotation noise across distant regions. The global path planner provides a sequence of Lanelets and areas that constitute the complete path (e.g., L_1, L_2, L_3, …, L_n, A_1, L_{n+1}). The algorithm first extracts centerline points from all Lanelets, which inherently include the start and end points of each Lanelet (where the end of L_n is on the perimeter of area A_1, and the start of L_{n+1} is also on the area's perimeter). The centerline preprocessing algorithm employs a windowed approach using quintic Bezier curves.

Algorithm 1 Centerline Preprocessing using Quintic Bezier Curves

Require: Sequence of Lanelets and Areas $\{L_1, L_2, \ldots, L_n, A_1, L_{n+1}, \ldots\}$
Ensure: Preprocessed centerline waypoints $W = \{w_1, w_2, \ldots, w_m\}$

$centerline\_points \leftarrow \{\}$
for each Lanelet L_i in sequence **do**
  $centerline\_points \leftarrow centerline\_points \cup ExtractCenterline(L_i)$
end for
for each Area A_j in sequence **do**
  $boundary\_points \leftarrow GetAreaBoundaryPoints(A_j, L_{prev}, L_{next})$
  $centerline\_points \leftarrow centerline\_points \cup boundary\_points$
end for
$W \leftarrow \{\}$, $window\_size \leftarrow 6$, $overlap \leftarrow 2$, $interval \leftarrow 1.5m$
$start\_idx \leftarrow 0$
while $start\_idx + window\_size \leq |centerline\_points|$ **do**
  $control\_points \leftarrow centerline\_points[start\_idx : start\_idx + window\_size]$
  $bezier\_curve \leftarrow FitQuinticBezier(control\_points)$
  $bisected\_points \leftarrow BisectCurve(bezier\_curve, interval)$
  $W \leftarrow W \cup bisected\_points$
  $start\_idx \leftarrow start\_idx + window\_size - overlap$
end while
return W

In Algorithm 1, the window size of 6 is dictated by the quintic Bézier degree ($n+1$ control points). An overlap of 2 points between successive windows enforces C^1 continuity at segment joins. The bisection interval of 1.5 m was chosen to match the robot's kinematic constraints. This approach is extended to handle transitions between Lanelets annotated on pavements and areas. At area boundaries, the common points between L_n and A_1, and between A_1 and L_{n+1} were included, then a quintic Bézier curve was fit and bisected to generate initial waypoints. While this preprocessing may occasionally result in paths that intersect with obstacles within areas (such as benches), these conflicts are resolved during trajectory generation using distance mapping and rProp-based optimization, as described by Lau et al. [23]. As illustrated in Fig. 4, the test track features the center points generated from the annotated Lanelets on the pavements connected to the center points generated on areas (university campus) from the above mentioned logic. For example, when the AMR navigates from point A (Start Location) to point B (Goal Location 2), it must traverse pavements (represented as Lanelets) and then navigate through university campus (represented as areas) in front of the destination. It is important to note that the quality of the generated map depends significantly on the human annotator's expertise since the approach does not utilize survey vehicles with LiDAR or extensive image collection systems. The reliance is solely on open-source map data such as OSM imagery or Bing aerial imagery which introduces challenges when aerial images contain occlusions from trees, clouds or shadows that might obscure pavements or relevant areas. In such cases, annotators must rely on intuition and contextual understanding to complete the map, potentially introducing inaccuracies. On open spaces like college campus, city centers, the operational space is available without hard restrictions.

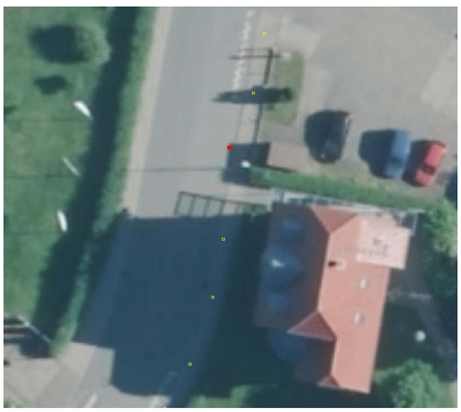

Fig. 5. Inaccuracy in Lanelet Annotation

Contrary to the open spaces, the pavements introduce limits to the robot operation with vehicles operating on roads adjoining. In this case, inaccuracies in annotations can generate innaccurate center points as shown by the red point in Fig. 5. The AMR trying to center itself on this waypoint can result in the AMR going off track leading to on road accidents. In such a case, one cannot rely only on manual Lanelet annotations where strict space restrictions are supposed to be complied with. Therefore, this paper introduces the use of segmentation based waypoint correction applied to the annotated Lanelets described in the next sections.

B. Segmentation based Pavement Limit and Center Point Generation

When AMR navigates on pavements, it is necessary to adhere to the pavement limits to avoid moving on to the roads. Therefore, a local navigation system which checks not only the limits of the pavement but also ensures the robot is centered on this pavement is necessary. To tackle this, a Realsense D435i camera is employed on the AMR. Convolutional Neural Networks (CNN) are used to detect and segment the pavement from the rest of the entities in the environment. YOLOv8 segmentation model is used to detect and segment the pavements. The model was trained on a custom dataset consisting of 190 original images for the selected test track, which were augmented using 2% rotation, brightness adjustment, and noise addition on Roboflow resulting in a total of 457 images. The dataset was split into 87% for training, 9% for validation, and 4% for testing on a local GPU. The segmentation solution was tested under different light conditions including occlusions/tree shadows. The center point (illustrated in red) is generated from the corresponding series of points (illustrated in dark blue) generated at the edge of the segmented mask which resemble the pavement limits in Fig. 6. Furthermore, the AMR getting closer to edge points triggers a warning to the control system.

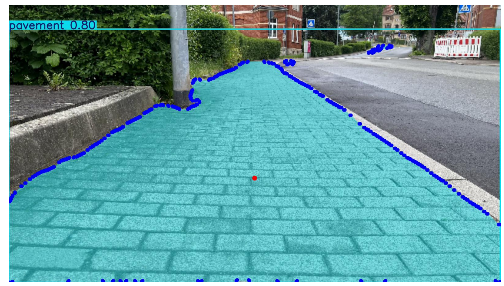

Fig. 6. Segmented Pavement Limit and Center Point Generation

The fundamental challenge in outdoor navigation is to track the center point throughout fluctuations due to the terrain or change of scene. The center point in this approach is tracked by Lucas-Kanade method [24]. Lucas–Kanade method assumes that the flow is essentially constant in a local neighbourhood of the pixel under consideration, and solves the basic optical flow equations for all the pixels in that neighbourhood, by the least squares criterion. The optical flow equation is fundamentally derived from the brightness constancy assumption, which states that the intensity of a given point in an image remains invariant under its motion over time. [25] Therefore, if $I(x, y, t)$ denotes the intensity of the image at pixel coordinates (x, y) and time t, then the brightness constancy assumption can be expressed as:

$$I(x, y, t) = I(x + \Delta x, y + \Delta y, t + \Delta t) \tag{1}$$

where Δx and Δy are small spatial displacements in the image plane over a small temporal interval Δt. To form a relation between these quantities analytically, the right-hand side of eq. (1) is approximated through the first-order Taylor series expansion around (x, y, t):

$$I(x + \Delta x, y + \Delta y, t + \Delta t) \approx I(x, y, t) + \frac{\partial I}{\partial x}\Delta x + \frac{\partial I}{\partial y}\Delta y + \frac{\partial I}{\partial t}\Delta t \tag{2}$$

By subtracting $I(x, y, t)$ from either sides and dividing by Δt throughout, the **optical flow constraint equation** yields:

$$I_x u + I_y v + I_t = 0 \tag{3}$$

where

- $u = \dfrac{\Delta x}{\Delta t}$ and $v = \dfrac{\Delta y}{\Delta t}$ illustrate the components of the velocity vector along the horizontal and vertical image axes, respectively.

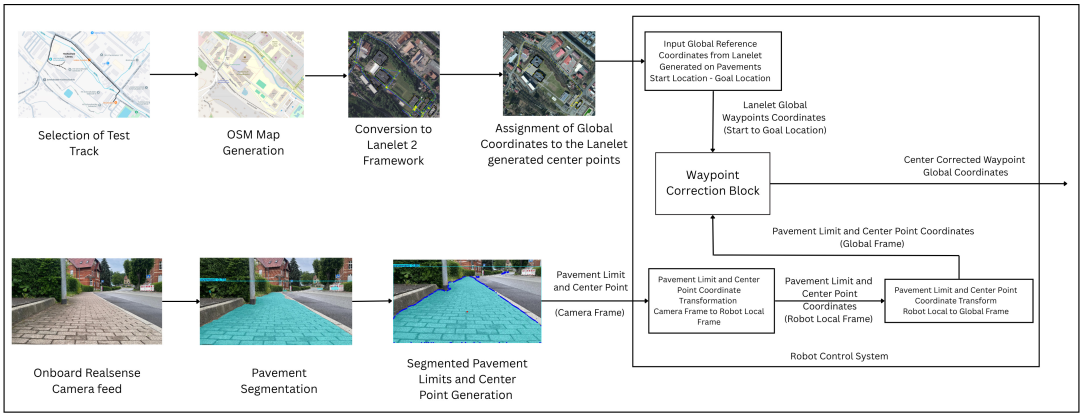

Fig. 7. Segmentation Based Waypoint Correction Architecture

- $I_x = \frac{\partial I}{\partial x}$, $I_y = \frac{\partial I}{\partial y}$, and $I_t = \frac{\partial I}{\partial t}$ represent the spatial and temporal partial derivatives of the image intensity function. [25]

By combining information from several nearby pixels, the Lucas–Kanade method can resolve the inherent ambiguity of the optical flow equation. The algorithm described in this section is combined with the global navigation solution and is explained in the next section.

III. Segmentation Based Waypoint Correction

This section integrates the two approaches mentioned above and forms a robust navigation system that can compensate for annotation inaccuracies while maintaining efficient operation. The overall architecture illustrated in Fig. 7 operates by first generating a global path using the annotated Lanelet2 map. The Lanelet2 map is annotated based on the OSM map generated for the test track selected. This path consists of a sequence of Lanelets with GPS coordinates assigned to each generated center points from the Start Location to the Goal Locations as shown in Fig. 4. The input global reference coordinates for the center points are sent to the robot's control system. In the control system, these intermediate waypoint coordinates generated are sent to the Waypoint Correction Block as shown in Fig. 7. Simultaneously, the onboard camera performs real-time segmentation of pavements, extracting pavement limit points that are used to calculate the center points of the detected pavement. The coordinates of the pavement limit points and the generated centre point is then converted from the camera frame to the robot local frame. Subsequently, the coordinates are converted further to the global frame based on the robot's GPS. The data from both sources, map-based global waypoints and vision-based local center points along with the pavement limits are transformed into the common UTM32 coordinate frame and are sent to the Waypoint Correction Block.

The Waypoint Correction Block illustrated in Fig. 8 takes the Lanelet global waypoints from the start to the goal location and the current location of the AMR. Both the inputs are compared and the next waypoint as per the current location of the AMR is selected. The global coordinates of the waypoint selected are compared with the pavement limit coordinates generated as a result of the camera segmentation. This comparison is performed to check whether the annotated waypoint lies within the segmented pavement limits. Once the condition is satisfied, the waypoint is then compared to the center point generated by the pavement segmentation. If a waypoint is generated away from the center of the pavement near its edge as shown in Fig. 5, the system employs a weighted averaging method that favors the more reliable data source. For instance, if the segmented pavement has a high confidence score of 75%, the system combines the data using:

$$P_{\text{corrected}} = \alpha \cdot P_{\text{vision}} + (1 - \alpha) \cdot P_{\text{map}} \tag{4}$$

Where α represents the confidence score of the vision system (e.g., 0.75), P_{vision} is the center point extracted from image segmentation, P_{map} is the corresponding waypoint from the Lanelet2 map, and $P_{\text{corrected}}$ is the resulting waypoint used for navigation. This approach is particularly valuable in the test environment, where the AMR must navigate both pavements with limited operational space and open areas with more freedom of movement. When the robot travels on pavements, the combined approach provides precise centerline guidance despite potential map inaccuracies. Upon entering campus areas, the system can intelligently reduce its dependence on the pavement detection model, specifically trained for sidewalks. This adaptive approach allows the system to maintain accuracy even when map data contains errors due to annotation limitations or environmental changes not reflected in the map.

IV. Conclusion

This paper extends the Lanelet2 framework which is traditionally used for road vehicles to pedestrian environments like sidewalks. To address the spatial and safety constraints inherent in these areas, a real-time camera-based pavement segmentation module that locally corrects global waypoints is introduced. The resulting system fuses global map-based planning with local scene understanding, creating a robust

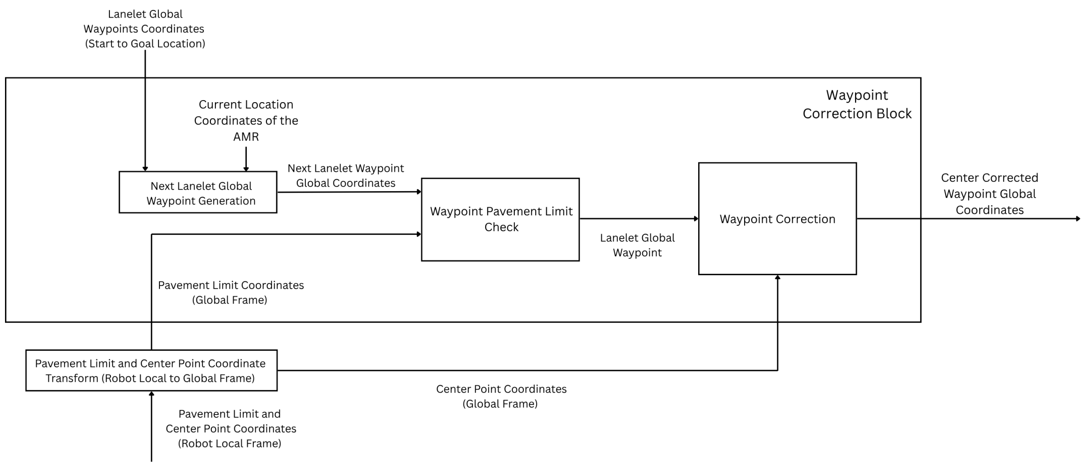

Fig. 8. Waypoint Correction Block

and adaptable global navigation solution well-suited for common last-mile delivery scenarios. This groundwork opens the path for scalable deployment of sidewalk-based delivery robots for entire cities/towns and lays the foundation for further automation in map annotation and combination with dynamic obstacle avoidance and semantic understanding in complex urban settings.

REFERENCES

[1] B. Cheng, H. Liu, Y. Wang, and Y. Liu, "Application Prospect of High-Precision Map Mode and Pure Vision Mode in Automatic Driving," in Proc. Int. Conf. on Artificial Intelligence and Security (ICAIS), Lecture Notes in Computer Science, vol. 1586, Springer, 2022.

[2] L. Mi, H. Zhao, C. Nash, X. Jin, J. Gao, C. Sun, C. Schmid, N. Shavit, Y. Chai, and D. Anguelov, "HDMapGen: A Hierarchical Graph Generative Model of High Definition Maps", IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Nashville, USA, Jun. 2021.

[3] J. Ziegler, P. Bender, M. Schreiber, H. Lategahn, T. Strauss, C. Stiller, T. Dang, U. Franke, N. Appenrodt, C. G. Keller, E. Kaus, R. G. Herrtwich, C. Rabe, D. Pfeiffer, F. Lindner, F. Stein, F. Erbs, M. Enzweiler, C. Knoppel, J. Hipp, M. Haueis, M. Trepte, C. Brenk, A. Tamke, M. Ghanaat, M. Braun, A. Joos, H. Fritz, H. Mock, M. Hein, E. Zeeb, "Making Bertha drive – an autonomous journey on a historic route," IEEE Intelligent Transportation Systems Magazine, 2014.

[4] F. Poggenhans et al., "Lanelet2: A high-definition map framework for the future of automated driving," 2018 21st International Conference on Intelligent Transportation Systems (ITSC), Maui, HI, USA, 2018.

[5] F. Immel, R. Fehler, F. Bieder, C. Stiller. (2024). Generation of Training Data from HD Maps in the Lanelet2 Framework. 10.48550/arXiv.2407.17409.

[6] S. J. Fusic, G. Kanagaraj, K. Hariharan, and S. Karthikeyan, "Optimal path planning of autonomous navigation in outdoor environment via heuristic technique," Transportation research interdisciplinary perspectives, 2021.

[7] J. Li, H. Qin, J. Wang, and J. Li, "Openstreetmap-based autonomous navigation for the four wheel-legged robot via 3d-lidar and ccd camera," IEEE Transactions on Industrial Electronics, 2021.

[8] N. Akai, L. Y. Morales, E. Takeuchi, Y. Yoshihara, and Y. Ninomiya, "Robust localization using 3d ndt scan matching with experimentally determined uncertainty and road marker matching," in IEEE Intelligent Vehicles Symposium (IV), 2017.

[9] S. Lowry, N. Sunderhauf, P. Newman, J. J. Leonard, D. Cox, P. Corke, and M. J. Milford, "Visual place recognition: A survey," IEEE Transactions on Robotics, 2015.

[10] D. Shah, A. Sridhar, N. Dashora, K. Stachowicz, K. Black, N. Hirose, and S. Levine, "Vint: A foundation model for visual navigation," arXiv preprint arXiv:2306.14846, 2023.

[11] A. Sridhar, D. Shah, C. Glossop, and S. Levine, "Nomad: Goal masked diffusion policies for navigation and exploration," in Proc. IEEE Int. Conf. Robot. Autom. (ICRA), May 2024.

[12] J. Liang, K. Weerakoon, T. Guan, N. Karapetyan, and D. Manocha, "Adaptiveon: Adaptive outdoor local navigation method for stable and reliable actions," IEEE Robotics and Automation Letters, 2022

[13] J. Liang, P. Gao, X. Xiao, A. J. Sathyamoorthy, M. Elnoor, M. C. Lin, and D. Manocha, "Mtg: Mapless trajectory generator with traversability coverage for outdoor navigation," in Proc. 2024 IEEE Int. Conf. Robot. Autom. (ICRA), Yokohama, Japan, May 13–17, 2024.

[14] J. Liang, A. Payandeh, D. Song, X. Xiao, and D. Manocha, "Dtg: Diffusion-based trajectory generation for mapless global navigation," in Proc. 2024 IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS), Abu Dhabi, UAE, Oct. 14–18, 2024.

[15] M. Y. Khandelwal, S. Tendulkar, G. A. Kolbai and F. Schroedel, "Coaction between Automobiles and Mobile Robots - Interoperability for Affordable Last Mile Delivery Solutions," AmEC 2024 – Automotive meets Electronics & Control; 14. GMM Symposium, Germany, 2024.

[16] S. Tendulkar, M. Y. Khandelwal and F. Schrödel, "V2X based Vehicle Environment Perception and Occupancy Analysis for Dynamic Pedestrian Behaviour," 2024 European Control Conference (ECC), Stockholm, Sweden, 2024.

[17] "Lanelet2," GitHub repository, FZI Forschungszentrum Informatik, [Online]. Available: https://github.com/fzi-forschungszentrum-informatik/Lanelet2. [Accessed: 01-Jun-2025].

[18] J. Y. Liao, P. Doshi, Z. Zhang, D. Paz and H. Christensen, "OSM vs HD Maps: Map Representations for Trajectory Prediction," 2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Abu Dhabi, United Arab Emirates, 2024.

[19] S. Tendulkar, A. Nannuri, S. Nagdev, and F. Schrödel, "Multilayer Environment Model for Outdoor Autonomous Mobile Robot Navigation," preprint, 2026, doi: 10.13140/RG.2.2.11843.36640, [Online].

[20] OpenStreetMap contributors, "OpenStreetMap," [Online]. Available: https://www.openstreetmap.org/. [Accessed: 01-Jun-2025]. Data licensed under the Open Data Commons Open Database License.

[21] JOSM team, "JOSM - Java OpenStreetMap Editor," [Online]. Available: https://josm.openstreetmap.de/. [Accessed: 01-Jun-2025].

[22] C. Yin, M. Cecotti, D.J. Auger, A. Fotouhi, and H. Jiang, "Lane Centerline Extraction Based on Surveyed Boundaries: An Efficient Approach Using Maximal Disks," Sensors, vol. 25, p. 2571, 2025.

[23] B. Lau, C. Sprunk and W. Burgard, "Kinodynamic motion planning for mobile robots using splines," 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, St. Louis, MO, USA, 2009.

[24] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in Proc. 7th Int. Joint Conf. Artif. Intell. (IJCAI'81), vol. 2, Vancouver, Canada, Aug. 1981.

[25] B.K. Horn, B. G. Schunck, "Determining optical flow", Artificial Intelligence, Volume 17, Issues 1–3, 1981.