Sensors & Transducers, Vol. 273, Issue 2, June 2026, pp. 30-38

www.sensorsportal.com

Sensors & Transducers

Published by IFSA Publishing, S. L., 2026
http://www.sensorsportal.com

Vehicle Localization Based on Roadside Object Detection Using YOLOv8

Takayoshi YOKOTA
Center for Data Science and Artificial Intelligence Education, Institute of Science Tokyo,
2 Chome-12-1 Ookayama, Meguro-ku, Tokyo 152-8550, Japan
Tel.: + 81 3 5734 2651
E-mail: yokota@dsai.isct.ac.jp

Received: 7 April 2026 /Revised: 13 May 2026 /Accepted: 10 June 2026 /Published: 29 June 2026

Abstract: This article proposes a vehicle localization method that enhances MEMS sensor-based self-localization by incorporating roadside object detections obtained from on-board camera images. Although conventional MEMS-based approaches have achieved meter-level accuracy, their performance degrades on roads with limited surface variations, leading to reduced robustness. To overcome this limitation, roadside objects such as traffic lights, traffic signs, and poles are detected using a fine-tuned YOLOv8m model and utilized as visual landmarks. The spatial configurations of the detected bounding boxes are matched with a reference dataset associated with RTK-GNSS positions, enabling correction of accumulated localization errors. This approach reduces dependency on road surface features and improves localization stability in diverse urban environments. Experimental results using real-world driving data demonstrate that the proposed method achieves improved robustness and maintains localization errors within approximately 3.13 m (mean), 0.42 m (median), 12.6 m (RMSE), and 111.8 m (maximum) under challenging conditions.

Keywords: Vehicle localization, MEMS sensors. RTK-GNSS, Object detection, YOLOv8m, Bounding box.

1. Introduction

Among vehicle localization algorithms [5–19], previous studies have applied cross-correlation to signals obtained from MEMS sensors to identify the best match with reference data recorded together with precise RTK-GNSS measurements [1, 2]. Although such approaches can achieve meter-level accuracy, their performance may degrade on smooth roads with limited surface variations, occasionally resulting in position errors of up to 10 m.

To compensate for this limitation, camera images captured during driving are exploited to extract additional environmental information. In this study, YOLOv8m, developed by Ultralytics [3], is applied to video sequences to detect roadside objects. The spatial distribution of the detected objects within each frame is then used as supplementary information for vehicle localization. YOLOv8 provides several model variants (v8n, v8s, v8m, v8l, and v8x) with different model sizes and computational costs. Among them, YOLOv8m offers a good balance between accuracy and computational efficiency, making it suitable for on-board applications, which are the target scenario of this study.

Since the standard 80 classes provided in the COCO dataset are insufficient to represent roadside landmarks relevant to localization, additional training data were collected and annotated. These data include traffic lights, traffic signs, trees, and other roadside structures.

Although many studies have utilized camera or LiDAR data for vehicle localization [4], relatively few have focused on self-localization methods that rely primarily on on-board sensors without external infrastructure. This work aims to bridge this gap by

integrating visual landmark information with MEMS-sensor-based localization. This article is an extended version of the conference paper [1].

2. Vehicle Localization Algorithm Using Roadside Object Detection

Fig. 1 illustrates the overall processing flow. A video sequence captured by an on-board camera is processed using YOLOv8m, and the resulting object detections can be time-synchronized with measurements obtained from MEMS sensors, if available; however, no MEMS sensors are used in this study. The original COCO-80 label plays only a minor role in this study because many roadside objects suitable as landmarks are not included in the standard classes.

Therefore, YOLOv8m was fine-tuned using a custom dataset comprising seven classes of roadside objects: traffic lights, traffic signs, variable message signs (VMS), poles, trees, pedestrian bridges, and towers.

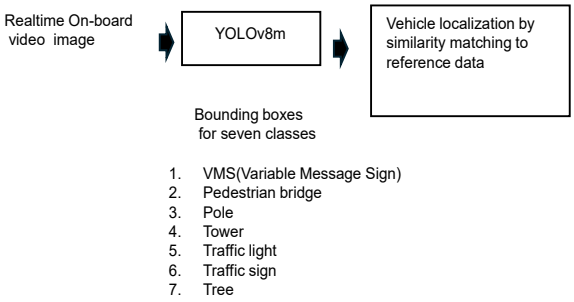

Fig. 1. Pipeline of the proposed vehicle localization method. YOLOv8m detects seven classes of roadside objects from real-time video images, and the vehicle position is estimated by similarity matching with reference data based on the detected bounding boxes.

$$\text{Similarity}(F_i, F_j) = \frac{1}{\max(N_i, N_j)} \times \sum_{\substack{l \le N_i \\ m \le N_j,\ c_l = c_m}} \max\left\{\alpha\, e^{-\beta_1(|x_l - x_m| + |y_l - y_m|)} + (1-\alpha)\, e^{-\beta_2(|w_l - w_m| + |h_l - h_m|)}\right\} \quad (1)$$

The dataset was constructed from video recordings captured for approximately 40 minutes along an arterial road in Edogawa-ku, Tokyo, on August 20, 2024. Frames were sampled at 30-second intervals, resulting in a total of 76 images (60 for training and 16 for validation).

To compensate for the limited dataset size, data augmentation techniques provided by YOLOv8m – such as flipping, rotation, noise injection, and contrast adjustment – were applied, expanding the training set to 660 images (i.e., each original image was augmented to approximately ten images).

The model was trained for 600 epochs with a batch size of 8 on a desktop PC equipped with an NVIDIA RTX 4060 GPU. The dataset consists of 660 images, resulting in approximately 83 iterations per epoch with a batch size of 8. The training data were shuffled at the beginning of each epoch. The trained YOLOv8m model was then applied to recorded video frames. For each detected object, the bounding box parameters (x, y, w, h) and the corresponding class label were obtained. The timestamp for each frame was computed from the video metadata based on a frame rate of 29.97 fps and the video start time.

3. Matching the Bounding Boxes to the Reference Data

Bounding-box matching was performed using the similarity criterion defined in Eq. (1), which is computationally more efficient than pixel-level image correlation. To estimate the position of vehicles, the similarity between the bounding-box layouts of a target frame and those stored in a reference dataset was evaluated. Each reference frame contained a set of bounding boxes detected by the fine-tuned YOLOv8m model, along with the corresponding RTK-GNSS position. For each pair of frames, bounding boxes with identical class labels were matched. The similarity score was computed based on the positional and size differences between corresponding bounding boxes. The reference frame with the highest similarity score was selected as the most probable vehicle position, and the associated RTK-GNSS coordinates were adopted as the estimated position.

Let F_i and F_j denote image frames with frame indices i and j, respectively. N_i and N_j represent the numbers of bounding boxes in frames F_i and F_j.

The class labels of the bounding boxes are denoted by c_l and c_m, and l and m are indices of bounding boxes in each frame. The similarity function is defined as follows: The parameters were empirically determined as $\alpha = 0.7$, $\beta_1 = 0.002$, and $\beta_2 = 0.005$. The variables (x_l, y_l) denote the coordinates of the bounding-box l's center, while w_l and h_l represent the width and height of the bounding box l. All values are normalized by the image resolution of 960×540(original resolution: 1920×1080).

Although other similarity measures, such as Intersection over Union (IoU), could be used, the proposed greedy similarity function was adopted for

computational efficiency, as it relies only on positional and size differences between bounding boxes.

4. Test Field and Experimental Setup

Field experiments were conducted on August 20, 2024, in Edogawa-ku, Tokyo, along National Route 14, also known as Shin-Ohashi-dori. A satellite image of the test course is shown in Fig. 2, where the evaluation section is indicated by a yellow arrow. The length of the evaluation section was approximately 1.75 km, and the travel time was about 300 s. The test vehicle was a Toyota Noah, a 2-liter, five-door minivan equipped with a GoPro HERO12 camera to capture video data during driving. Ground-truth vehicle positions were obtained using an RTK-GNSS receiver (u-blox ZED-F9P) [20], operating at 10 Hz with centimeter-level accuracy. Figs. 3 and 4 show the test vehicle and the RTK-GNSS receiver, respectively. The built-in electronic image stabilization function of the GoPro HERO12 was enabled.

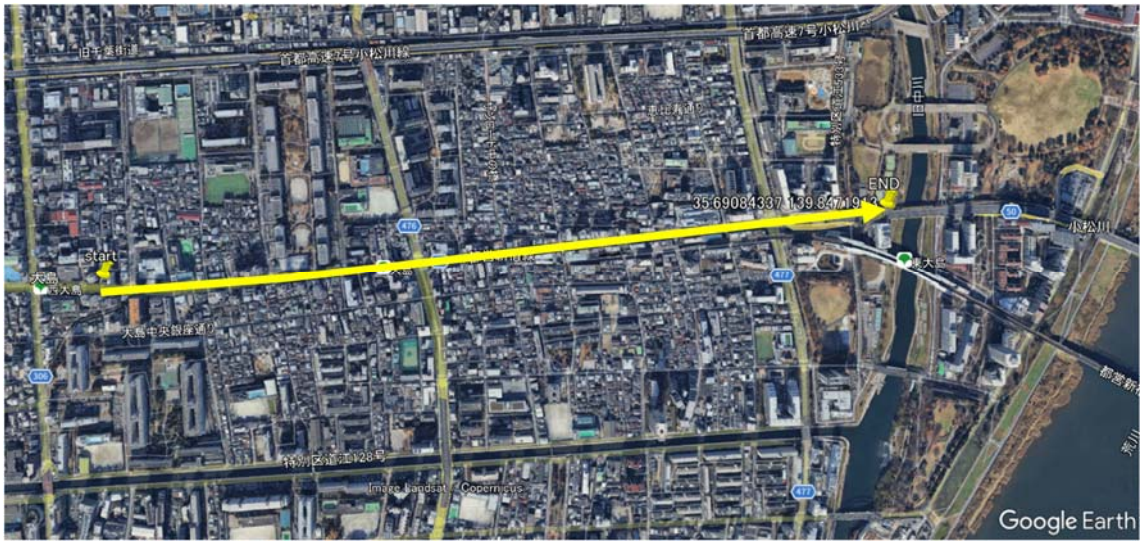

Fig. 2. Field test course. National road route 14 in Edogawa-ku, Tokyo, known as “Shin-ohashi-dori’. The length of the evaluation section is 1.75 km and is indicated by yellow arrow.

Fig. 3. Vehicle used for data collection (Toyota Noah, 2.0 L, 7-passenger capacity).

Fig. 4. RTK-GNSS receiver module (u-blox ZED-F9P) used for ground-truth position measurement in the field experiments.

Data acquisition for the section indicated by the yellow arrow was performed twice by driving along a circular course that included this section. The total travel time for the entire course was approximately 40 minutes. RTK-GNSS data, video images, and MEMS sensor data were acquired; however, the MEMS sensor data were not used in this article.

5. Details of Processing

5.1. Summary of YOLOv8m Training

Before presenting the vehicle localization results, a summary of the YOLOv8m training and validation process is provided. In the training phase of

YOLOv8m, still images were extracted from GoPro HERO12 video footage at 30-second intervals. The total number of images is 76. These images were divided into 60 training and 16 validation samples, and annotations were created using the cloud-based tool Label Studio [21].

Fig. 5 shows how the acquired images were used as training and validation data. The dataset was split into training (60 images) and validation (16 images) sets. Data augmentation increased the training dataset to 660 images.

Seven classes were defined as typical roadside objects: variable message signs (VMS), pedestrian bridges, poles, traffic lights, traffic signs, and trees. The frequency of each class in the training dataset is summarized in Fig. 6. As shown in the figure, the dataset is highly imbalanced, with the pole class being significantly overrepresented compared to the others. This imbalance is inherent to roadside environments and was therefore not corrected in this study.

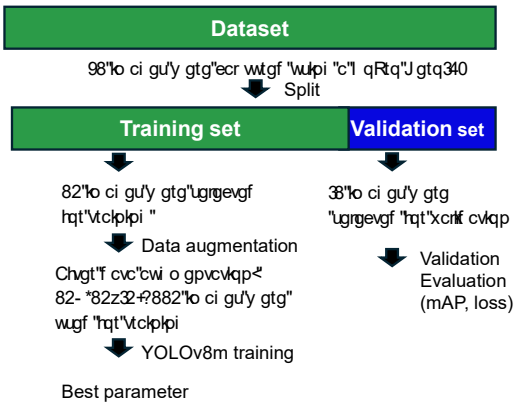

Fig. 5. Dataset preparation and training pipeline. The dataset was split into training (60 images) and validation (16 images) sets. Data augmentation increased the training data to 660 images. YOLOv8m was used for training and evaluation.

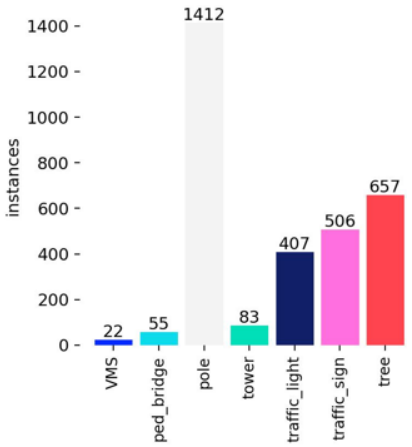

Fig. 6. Number of instances for each object class in the training dataset. The distribution is highly imbalanced, with the pole class being significantly more frequent than the other classes.

Despite this imbalance, the proposed method is applied to vehicle localization.

Table 1 shows the confusion matrix. Taking the pole class as an example, 21 out of 38 instances were correctly detected in the validation dataset, corresponding to a recall of approximately 55 %. In contrast, traffic signs were detected accurately (10 out of 10 instances). Overall, the object detection performance is limited, mainly due to the small dataset size and significant class imbalance. Although these issues are difficult to resolve, vehicle localization is performed using the detection results as they are.

Table 1. Confusion matrix of YOLOv8m object detection results on the validation dataset. A substantial number of objects, particularly poles, are misclassified as background, indicating a high false-negative rate. This degradation in performance is mainly attributed to dataset imbalance and the inherent difficulty of detecting small and thin objects.

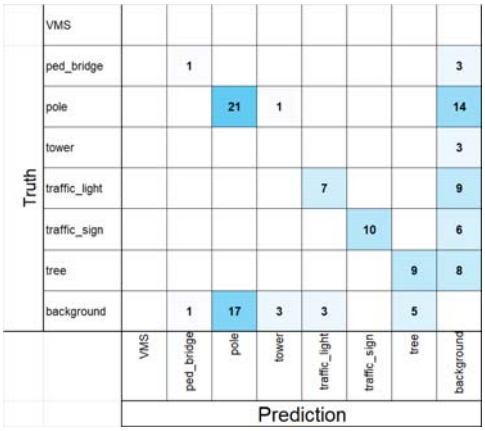

Fig. 7 shows the training and validation loss curves over 600 epochs. The box regression loss (box_loss), classification loss (cls_loss), and distribution focal loss (dfl_loss) are presented for both the training and validation datasets. The training losses decrease steadily, while the validation losses exhibit slight fluctuations and gradually stabilize, indicating reasonable generalization performance given the limited dataset size. To examine the convergence behavior of YOLOv8m, the model was trained for up to 600 epochs. However, slight overfitting was observed after approximately 400 epochs. Therefore, considering computational efficiency, the model parameters corresponding to the best validation performance within the first 400 epochs were used for the vehicle localization step, as they provide comparable performance while reducing training cost.

5.2. Object Detection

Examples of object detection are shown in Figs. 8a, 8b, and Figs. 9a, 9b. After fine-tuning, the COCO-80 classes are rarely detected, while the newly defined seven classes become dominant.

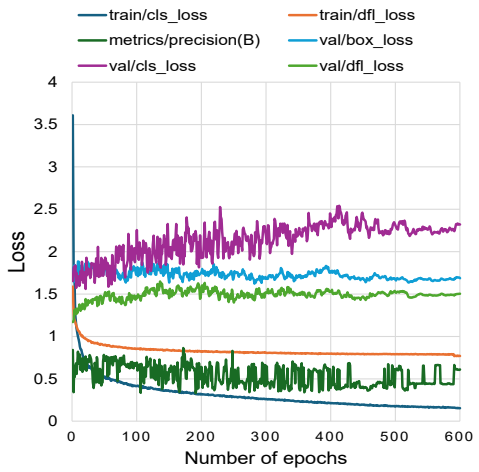

Fig. 7. Training and validation loss curves and precision during training. The training losses decrease steadily, while the validation losses remain higher and fluctuate, indicating possible overfitting. The precision gradually improves and stabilizes as training progresses.

Fig. 8a. YOLO detected image of the start point of the evaluation course. The first trip at 11:32:54.

Fig. 8b. YOLO detected image of the start point of the evaluation course. The first trip at 12:00:20.3.

Although the training is not perfect, the detection performance was moderately effective. The system outputs YOLOv8m object detection results in the form of bounding box attributes, including the class, width, height, and the x- and y-coordinates of the center position. YOLOv8m is sufficiently fast to detect objects in 29.97 fps HD video with a resolution of 1920 × 1080 pixels, while the images are downsampled to 960 × 540 for efficient processing.

Figs. 8a and 8b show snapshots of the video images captured at the same location but at different times of day and during different trips. The detected bounding boxes are overlaid on the images. Figs. 9a and 9b show the corresponding bounding-box-only representations. Since the images differ only in transient objects such as vehicles, the bounding-box representations remain highly similar. Consequently, the spatial layouts of the bounding boxes are also highly similar, leading to a high similarity in their spatial distributions. The proposed similarity matching procedure operates on these bounding-box representations rather than raw pixel data, eliminating the need for pixel-wise processing and enabling efficient computation.

Fig. 9a. YOLO detected bounding box only image of the start point of the evaluation course. The second trip at 11:32:54.

Fig. 9b. YOLO detected bounding box only image of the start point of the evaluation course. The second trip at 12:00:20.3.

To generate reference data, object detection results during the reference data acquisition phase are augmented with precise location information (latitude and longitude) and accurate time stamps, as illustrated in Fig. 10. The latitude and longitude are estimated by interpolating the 10 Hz RTK-GNSS data to match the 29.97 Hz video sampling rate. The interpolated positions are then associated with the corresponding YOLO-detected objects, forming a time-aligned reference dataset.

Since RTK-GNSS provides more accurate timestamps than the video camera, the alignment is performed using the GNSS timestamps. A precise calibration is subsequently applied to refine the alignment.

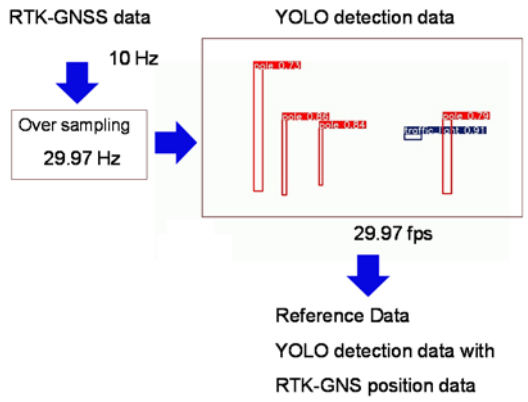

Fig. 10. Overview of the mapping process between object detection results and video frames. Detected roadside objects (left) are associated with corresponding bounding boxes in the video frames (center), which are sampled at 29.97 fps. This temporal alignment enables the conversion of detection results into a continuous time series for subsequent matching and localization.

6. Results and Discussion

6.1. Results of Similarity Matching

The results of bounding-box similarity matching are shown in Figs. 11a and 11b, which illustrate the relationship between the running times of the target vehicle and the reference vehicle. Each point represents a correspondence obtained by matching objects using the proposed similarity function, where the best similarity is selected for each time step of the target vehicle. The matching is performed frame by frame. The solid line indicates the estimated temporal alignment between the two vehicles derived from these correspondences. The step-like structure reflects differences in vehicle motion, such as stop-and-go behavior, and the dispersion of points indicates matching uncertainty. Despite the presence of outliers, the overall trend accurately captures the temporal alignment between the two vehicles. Fig. 11a shows the result using the best YOLOv8m weights after 100 epochs, whereas Fig. 11b presents the result using the best YOLOv8m weights after 400 epochs. The results are comparable, indicating that 100 epochs are sufficient to achieve reliable estimation of the temporal difference between the reference and target vehicles.

Fig. 11c presents the result after applying a median filter to the result of Fig. 11b.

Fig. 12 shows the ground truth of the temporal difference between the reference and target vehicles obtained using RTK-GNSS. Both vehicles were equipped with RTK-GNSS, and their precise positions were used to compute the temporal difference when they were at the same location. The sampling rate of RTK-GNSS was 10 Hz, whereas the video frame rate was 29.97 fps. Therefore, interpolation and oversampling were applied to obtain location data at 29.97 Hz.

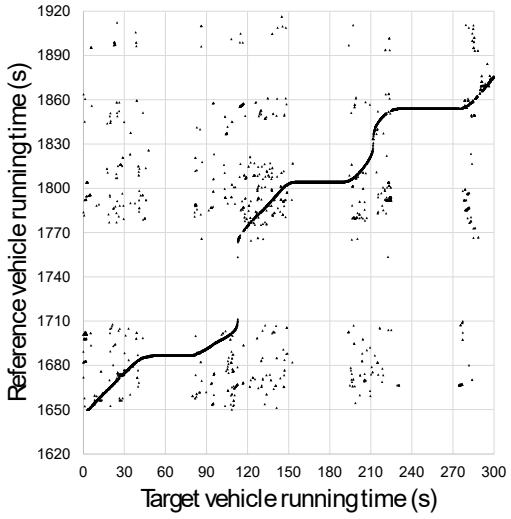

Fig. 11a. Results of bounding-box similarity matching using the best model obtained after 100 epochs.

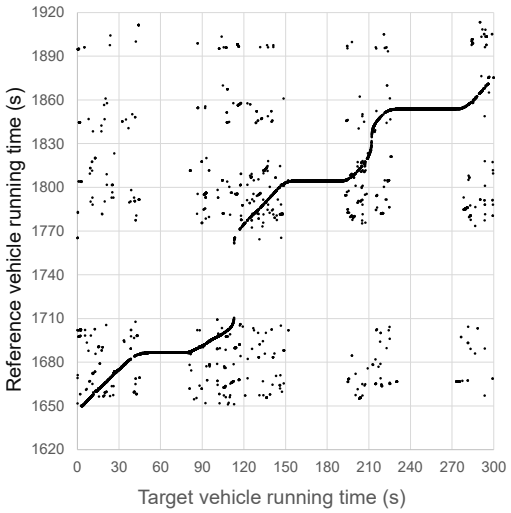

Fig. 11b. Results of bounding-box similarity matching using the best model obtained after 400 epochs.

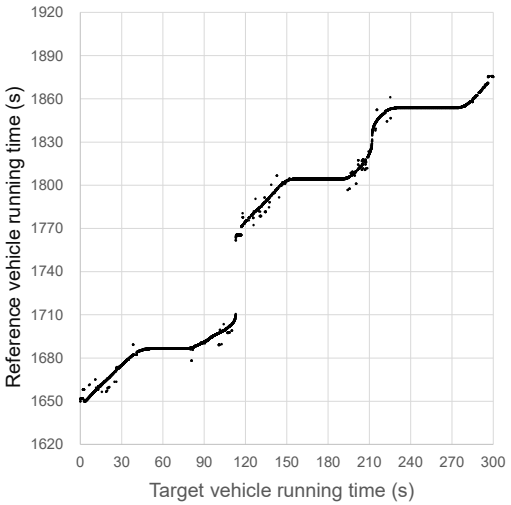

Fig. 11c. Results after median filtering of the bounding-box similarity matching using the best model obtained after 400 epochs.

By comparing Figs. 11a and 11b with Fig. 12, it can be observed that the temporal difference is estimated with high accuracy, although some spike noise remains.

6.2. Results of Location Estimation

Fig. 13 shows the results of position estimation. Fig. 14 shows the cumulative distribution of the position error. The error distribution exhibits higher kurtosis than a Gaussian distribution, with approximately 84.4 % of the errors within 2 m, 92.7 % within 5 m, and 95 % within 10 m, while a small number of outliers exceeding 20 m are observed.

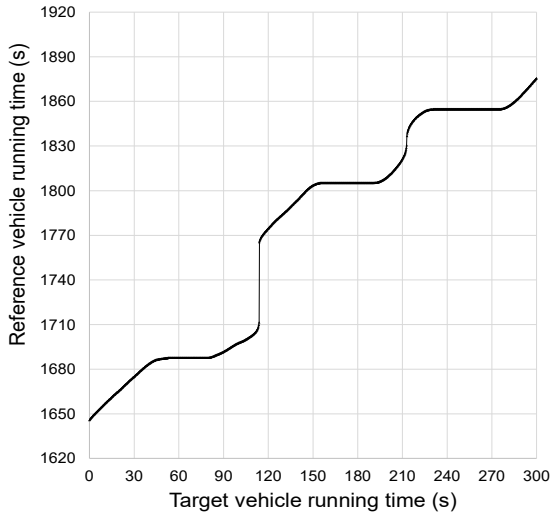

Fig. 12. Ground truth of the temporal difference between the target and reference vehicles.

Table 2 summarizes the error statistics. Although occasional large-error spikes are observed, the overall error distribution remains narrow.

Fig. 15 shows the directional distribution of the position error. The position errors in the east–west and north–south directions are concentrated near the origin, while larger errors exhibit a correlated trend in both directions, reflecting the constraint of the reference trajectory along the road geometry.

The maximum error is mainly caused by occasional spike errors in the similarity matching of detected object frames. In most cases, the error remains within several meters, demonstrating the effectiveness of the proposed method. These spike errors can be mitigated by integrating a dead-reckoning approach.

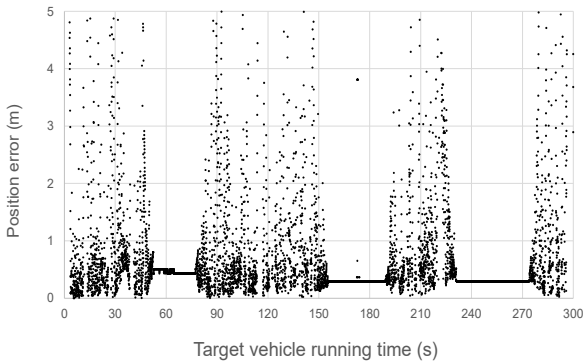

Fig. 13. Position error over time for the target vehicle. The horizontal axis represents the running time of the target vehicle, while the vertical axis shows the position error with respect to the reference trajectory. The error remains within several meters for most of the duration, with occasional spikes caused by mismatches in object detection.

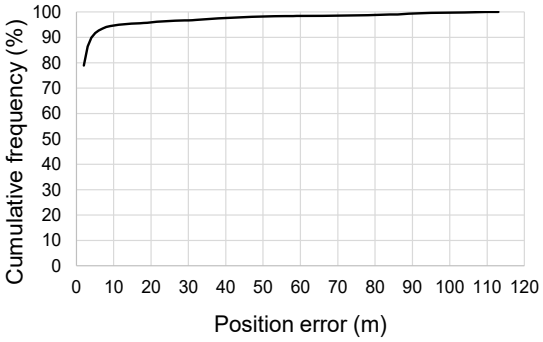

Fig. 14. Cumulative distribution of position error. Approximately 84.4 % of the errors are within 2 m, 92.7 % within 5 m, and 95 % within 10 m, while a small number of outliers exceeding 20 m are observed.

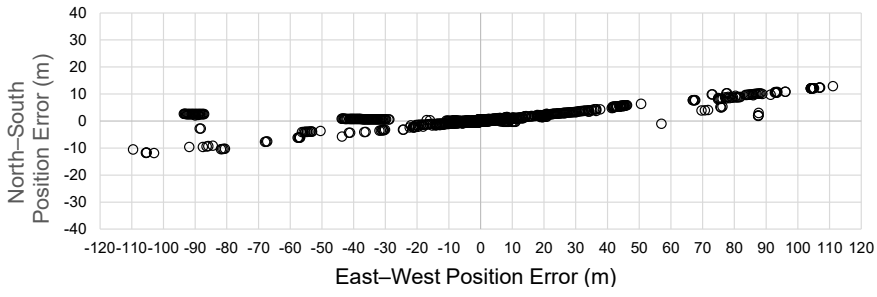

Fig. 15. Scatter plot of east–west and north–south position errors. Most errors are concentrated near the origin, while larger errors exhibit a correlated trend in both directions, reflecting the constraint of the reference trajectory along the road geometry.

Although dead reckoning inherently accumulates errors over time, its combination with the proposed method, which provides reliable absolute position estimates, can significantly enhance localization robustness and stability. Moreover, integrating a MEMS sensor-based localization algorithm can further compensate for the limitations of the proposed approach.

Table 2. Summary of the position estimation error.

Average error	3.13 m
Median error	0.42 m
RMSE	12.6 m
Max error	111.8 m

7. Conclusions

This article presents a vehicle localization algorithm based on the spatial arrangement of bounding boxes obtained from on-board camera images. In an experiment conducted on a 1.7 km arterial road section, the results demonstrate that incorporating roadside object detections improves localization robustness under challenging conditions. The estimated location error was 3.13 m on average, with approximately 84.4 % of errors within 2 m, 92.7 % within 5 m.

The proposed approach can also be extended to pedestrian localization scenarios where only camera images are available. Consequently, the method has the potential to support navigation in GNSS-denied environments, such as urban canyons and mountainous regions, provided that suitable visual landmarks are available.

Future improvements in YOLO training and model optimization are expected to further enhance the reliability and accuracy of the position estimation.

Acknowledgements

This work was supported in part by JSPS KAKENHI Grant Number JP25K15360.

References

[1]. T. Yokota, Roadside object detection-based vehicle localization, in *Proceedings of the 6th Winter IFSA Conference on Automation, Robotics & Communications for Industry 4.0/5.0/6.0 (ARCI)*, 2026, pp. 20-23.

[2]. T. Yokota, Vehicle localization algorithm with lane discrimination based on inertial and geomagnetic sensor data for GNSS-denied environments and its evaluation for different vehicles and drivers, *Sensors & Transducers*, Vol. 268, Issue 1, 2025, pp. 19-27.

[3]. Ultralytics, YOLOv8, https://github.com/ultralytics/ultralytics

[4]. S. Li, H. S. Yoon, Vehicle localization in 3D world coordinates using a single camera at traffic intersection, *Sensors*, Vol. 23, Issue 7, 2023, 3661.

[5]. T. Yokota, Network-wide vehicle localization algorithm based on MEMS sensor data, *Sensors & Transducers*, Vol. 265, Issue 12, 2024, pp. 17-26.

[6]. T. Yokota, Vehicle localization by correlated MEMS sensor data with velocity compensation, in *Proceedings of the 26th IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2023, pp. 2999-3004.

[7]. T. Yokota, T. Yamagiwa, Vehicle localization by optimally weighted use of MEMS sensor data, in *Proceedings of the 3rd IFSA Winter Conference on Automation, Robotics & Communications for Industry (ARCI)*, 2023, pp. 79-84.

[8]. T. Yamagiwa, T. Yokota, Vehicle localization method based on MEMS sensor data comprising pressure, acceleration and angular velocity, in *Proceedings of the IEEE International Conference on Mechatronics and Automation (ICMA)*, 2022, pp. 748-753.

[9]. T. Yokota, Vehicle localization by altitude data matching in spatial domain and its fusion with dead reckoning, *International Journal of Mechatronics and Automation*, Vol. 8, Issue 4, 2021, pp. 208-216.

[10]. T. Yokota, Vehicle localization by dynamic programming from altitude and yaw rate time series acquired by MEMS sensor, *SICE Journal of Control, Measurement, and System Integration*, Vol. 14, Issue 1, 2021, pp. 78-88.

[11]. T. Yokota, Vehicle localization based on MEMS sensor data, in *Proceedings of the 60th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE)*, 2021, pp. 1105-1110.

[12]. T. Yokota, Localization algorithm based on altitude time series in GNSS-denied environments, in *Proceedings of the 59th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE)*, 2020, pp. 985-990.

[13]. T. Yokota, Y. Hashimoto, T. Yamamoto, H. Yoshii, Fast and robust map-matching algorithm based on a global measure and dynamic programming for sparse probe data, *IET Intelligent Transport Systems*, Vol. 13, Issue 11, 2019, pp. 1613-1623.

[14]. J. Tsurushiro, T. Nagaosa, Vehicle localization using its vibration caused by road surface roughness, in *Proceedings of the IEEE International Conference on Vehicular Electronics and Safety (ICVES)*, 2015, pp. 164-169.

[15]. A. J. Dean, R. D. Martini, S. N. Brennan, Terrain-based road vehicle localization using particle filters, *Vehicle System Dynamics*, Vol. 49, Issue 8, 2011, pp. 1209-1223.

[16]. E. Laftchiev, C. Lagoa, S. Brennan, Terrain-based vehicle localization from real-time data using dynamical models, in *Proceedings of the 51st IEEE Annual Conference on Decision and Control (CDC)*, 2012, pp. 3366-3371.

[17]. J. Gim, C. Ahn, Ground feature-based vehicle positioning, in *Proceedings of the 59th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE)*, 2020, pp. 983-984.

[18]. J. Gim, C. Ahn, IMU-based virtual road profile sensor for vehicle localization, *Sensors*, Vol. 18, Issue 10, 2018, 3477.

[19]. X. Qu, B. Soheilian, N. Paparoditis, Landmark-based localization in urban environment, *ISPRS Journal of Photogrammetry and Remote Sensing*, Vol. 140, 2018, pp. 90-103.

[20]. u-blox, ZED-F9P GNSS module, https://www.u-blox.com/en/product/zed-f9p-module

[21]. HumanSignal, Label Studio, https://labelstud.i