2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

L-VIWO: Visual-Inertial-Wheel Odometry based on Lane Lines

Bin Zhao[1], Yunzhou Zhang[1*], Junjie Huang[1], Xichen Zhang[1], Zeyu Long[1], Yulong Li[1]

Abstract— **To achieve precise localization for autonomous vehicles and mitigate the problem of accumulated drift error in odometry, this paper proposes L-VIWO, a Visual-Inertial-Wheel Odometry based on lane lines. This method effectively utilizes the lateral constraints provided by lane lines to eliminate and relieve the incrementally accumulated pose errors. Firstly, we introduce a lane line tracking method that enables multi-frame tracking of the same lane line, thereby obtaining multi-frame data of a lane line. Then, we utilize multi-frame data of the lane lines and the curvature characteristics of adjacent lane lines to optimize the positions of the lane line sample points, thus building a reliable lane line map. Finally, we use the built local lane line map to correct the position of the vehicle. Based on the corrected position and prior pose from the odometry, we build a graph optimization model to optimize the pose of the vehicle. Through localization experiments on the KAIST dataset, it has been demonstrated that the proposed method effectively enhances the localization accuracy of odometry, thus confirming the effectiveness of the method.**

Index Terms: Visual-inertial-wheel odometry, lane lines, factor graph optimization.

I. Introduction

With the continuous development of autonomous driving technology, localization technology has received extensive research as a critical technology for autonomous vehicles (AVs). Visual-inertial odometry (VIO) has garnered significant attention due to its low-cost advantages, such as ORB-SLAM3 [1] and VINS-Fusion [2]. However, there are specific issues when deploying VIO in AVs. For example, [3] has demonstrated that scale is unobservable for VIO when the vehicle moves with constant local linear acceleration. It has also been shown that the degradation of scale can be eliminated by incorporating wheel encoders. Therefore, Visual-Inertial-Wheel Odometry (VIWO) has been extensively studied [4]–[7]. This paper adopts VIWO as the fundamental framework for ease of application in AVs.

The introduction of wheel encoders can effectively improve the localization accuracy of VIO, but uncertainties in the environment and sensors can still introduce errors. For example, abrupt illumination changes or interference from dynamic objects can bring about errors in feature matching, resulting in pose estimation errors. Although these errors may not be very noticeable between adjacent frames, they still cannot be ignored when accumulated over a long time. The introduction of GNSS can effectively reduce cumulative errors [8]–[13], but GNSS signals can be highly unreliable in occlusion and multipath conditions.

Most of the operating scenarios for AVs are urban road scenarios. Road signs, as prominent features, can provide a stable constraint to reduce localization errors. For example, [14] proposed a joint Bundle Adjustment (BA) method that utilizes road signs to improve the localization accuracy of visual odometry effectively. However, the constraint effect of road surface signs was not considered. [15]–[17] located AVs by matching detected road signs with a high-precision prior road sign map. Associating high-precision prior information with the vehicle's environmental perception to reduce the cumulative drift error of the odometry. However, these methods still rely on high-precision odometry to provide prior poses during the map building.

In this paper, considering that lane lines are long-term and stable features, we propose L-VIWO, a Visual-Inertial-Wheel Odometry based on lane lines. This method utilizes the lateral constraint information from lane lines to correct the position of the vehicle and achieve pose optimization, effectively mitigating the accumulated drift issue of odometry. Additionally, as this paper aims to enhance the localization accuracy of odometry, the method performs lane line map building and pose optimization simultaneously without the need to build a prior map in advance. The main contributions of this paper are as follows:

- A method for lane line tracking and lane line map building. Without the need for a specific detection network, this method leverages the characteristics of lane lines to optimize the positions of lane line sample points, resulting in a reliable lane line map.
- A pose optimization method based on the lane line map that leverages the lateral constraints of lane lines to relieve accumulated errors in odometry.
- Localization and ablation experiments were conducted on a general dataset to validate the effectiveness of the proposed method.

II. Related Works

A. Visual SLAM

In recent years, vision-based localization systems have flourished. The most representative frameworks are ORB-SLAM [1] [18] [19] and VINS [2] [20]. ORB-SLAM3 [1] relies on a multi-map system and can run stably even when visual information is poor for a long time. VINS-Mono [20] is a robust and accurate VIO system. VINS-Fusion [2] extends upon it to support binocular and GPS data fusion.

*The corresponding author of this paper
[1]Bin Zhao, Yunzhou Zhang, Junjie Huang, Xichen Zhang, Zeyu Long and Yulong Li are with College of Information Science and Engineering, Northeastern University, Shenyang 110819, China zhangyunzhou@mail.neu.edu.cn

This work was supported by National Natural Science Foundation of China (No. 61973066, 61471110) and Major Science and Technology Projects of Liaoning Province(No. 2021JH1/10400049).

Wu *et al.* [3] proved that the VIO system has additional observability degradation in constant acceleration or speed movement, such as the scale being unobservable. They also demonstrated that scale degradation could be eliminated by using wheel encoders. Liu *et al.* [21] proposed a tightly coupled VIWO, which integrates IMU and wheel encoder data during the pre-integration stage. Then, the team proposed a bidirectional trajectory computation method [22], solving the problem of the unobservability of accelerometer bias and extrinsic parameters before the first turn. Lee *et al.* [23] developed a VIWO system based on MSCKF, which supports online calibration of external parameters. Zhang *et al.* [24] proposed a ground robot pose estimation method based on motion manifolds. This method utilizes wheel encoders to perform motion manifold-based 6-DOF integration and fuses the data with a monocular camera.

Cao *et al.* [8] proposed a nonlinear optimization-based system (GVINS) to fuse GNSS raw measurements with visual and inertial data tightly. Xiao *et al.* [9] proposed a tightly-coupled INS-centric GNSS-visual-inertial navigation system (IC-GVINS), which demonstrated excellent robustness and accuracy in complex environments.

B. Localization Methods based on Road Feature

With the continuous development of various detection networks, road feature-based localization methods have been widely used in AVs. Qin *et al.* [25] used robust semantic features to build maps and locate vehicles in parking lots, achieving centimeter-level localization. [16] extended this method to outdoor urban scenes and proposed a crowd-sourced mapping scheme. However, the mapping part of these two methods only involves simple concatenation of semantic point clouds without considering the structural features of the signs. Cheng *et al.* [26] used specially trained detection networks to extract critical point information from road signs and build compact semantic maps, effectively utilizing the structural features of the road signs. Wang *et al.* [17] proposed an accurate and robust localization algorithm based on visual semantic features and a lightweight HD map. They also introduced a robust data association method to solve the ambiguity caused by similar sign features. Zhou *et al.* [15] built an instance-level semantic road landmark map and considered the uncertainty of the IPM model during mapping and localization. [14] proposed a joint BA method based on traffic signs, effectively improving the localization accuracy of visual odometry.

III. System Overview

L-VIWO is mainly composed of five parts, as shown in Fig. 1. The first part performs lane line detection and tracking, providing multi-frame lane line data for map building. The second part calculates the lateral distance between the vehicle and the right lane line for each image frame, which is used for subsequent position correction. The third part is the VIWO, which provides a prior pose for the system. The fourth part optimizes the positions of the lane line sample points to build a reliable lane line map. The fifth part combines the lane line map and lateral distance to obtain the corrected vehicle position. It uses the corrected position and prior pose from odometry to build a pose graph optimization model, achieving pose optimization. The lane line detection and VIWO run in parallel, and the remaining part is performed after both are completed.

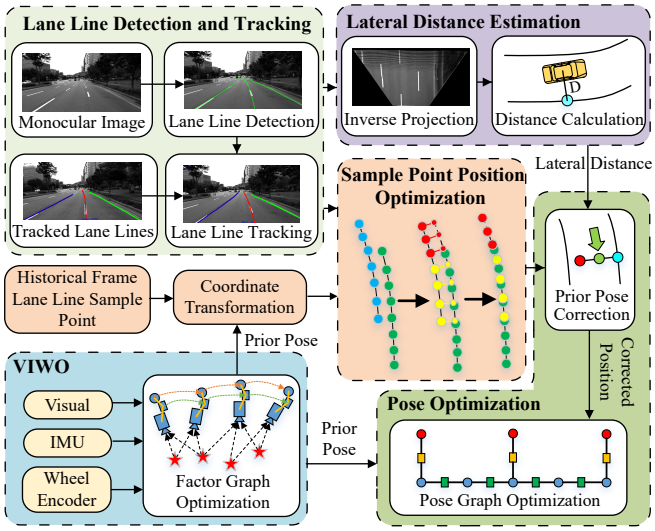


Fig. 1. An overview of L-VIWO. The lane line detection and VIWO run in parallel. After obtaining the prior pose from odometry and lane line detection results, lane line tracking, sample point position optimization, and lateral distance estimation are performed. The pose optimization is executed after detecting a lane change, ensuring the accuracy of the utilized lane line map.

IV. Methodology

A. Lane Detection and Representation

In this paper, we choose the lane line detection network LaneATT [27] to extract lane lines, which can achieve a maximum speed of 250FPS. At the same time, this network runs in parallel with odometry, further ensuring the system's real-time performance.

Unlike [26], which requires a specialized training detection network for keypoint extraction of road signs, this paper only uses simple sample points to represent lane lines, avoiding the system's dependence on a specialized detection network. The pixel coordinates of the sample points are provided by the detection network. The lane lines are represented as $L = \{id_l, M = (l^1, l^2 \cdots l^m)\}$, where id_l is the ID of the lane line, and M is the collection of sample point information in the detection frames. $l^m = \{id_c, P = (p_w^1, p_w^2 \cdots p_w^n)\}$ represents a lane line in an image frame, where id_c is the frame number of the image, and P is the collection of world coordinates of the lane line sample points.

B. Lane Line Tracking

To make full use of the continuity of lane lines and build a reliable lane line map, we need to track the detected lane lines in each frame to obtain multiple frame data for the same lane line. The lane lines detected in the current frame are denoted as $Line_D$, and the tracked lane lines in historical

frames are denoted as $Line_T$. Let the current frame be the k frame. The process of lane line tracking is shown in Fig. 2, and the detailed steps are as follows:

Step 1: Generate masks for $Line_D$ based on the position of the lane line sample points. Then, for the $Line_T$ observed in the $k-1$ frame, use the prior pose provided by the odometry to predict their positions in the current frame image and generate the masks.

Step 2: Calculate the 2D-IOU between two sets of masks:

$$IOU_i^j = \frac{mask_D^i \cap mask_T^j}{mask_D^i \cup mask_T^j} \tag{1}$$

where i and j are the lane line IDs for $Line_D$ and $Line_T$ respectively. The values of union and intersection are determined by the number of pixels in the corresponding region masks.

Step 3: Build the cost matrix for the Hungarian algorithm based on 2D-IOU:

$$Cost = \begin{bmatrix} cost_1^1 & cost_1^2 & \cdots & cost_1^n \\ cost_2^1 & cost_2^2 & \cdots & cost_2^n \\ \cdots & \cdots & \cdots & \cdots \\ cost_m^1 & cost_m^2 & \cdots & cost_m^n \end{bmatrix} \tag{2}$$

where m and n are the total number of lane lines in $Line_D$ and $Line_T$ respectively. Since the minimum cost Hungarian algorithm is used in this paper, the cost value is calculated as follows:

$$cost_i^j = 1 - IOU_i^j \tag{3}$$

After obtaining the cost matrix, the Hungarian algorithm is used to match the lane lines, and assignments with IOU values below the specified threshold are considered invalid.

Step 4: If there are matching failures, continue to match them with the unpaired $Line_T$ detected within the 100 frames. Assign a new ID to the still unmatched $Line_D$.

The reason we initially match with the $Line_T$ observed in the $k-1$ frame is that the lane lines observed in consecutive frames are basically the same set of lane lines. This helps improve tracking accuracy and reduce computational demands.

In particular, we found that when the vehicle undergoes large pitch angle changes, it causes significant position changes of the lane lines in the image, resulting in tracking failures. Therefore, when a substantial fluctuation in the vehicle's pitch angle is detected, the width of the mask will be increased, and the specified IOU threshold will be lowered during the tracking stage to ensure the accuracy of the tracking.

After tracking, matched lane line information is stored in the corresponding container L. For new observed lane lines, a new container is created to store their information. At the same time, we use the IPM model [28] to project the lane line sample points in the current frame onto the vehicle coordinate and use a cubic curve to fit the lane lines. Then, we draw a perpendicular line from the vehicle coordinate origin to the right lane line, and the distance between the origin and the perpendicular point is the lateral distance of the vehicle relative to the lane line, denoted as dis_b. The ID of the right lane line is also recorded. The dis_b and the ID will be used in the pose optimization part.

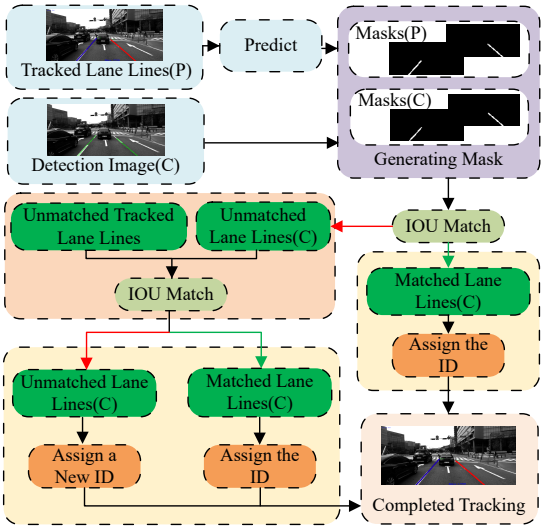


Fig. 2. The process of lane line tracking. The suffixes "P" and "C" represent the previous frame and current frame. "Predict" means predicting the position of the tracked lane lines in the current frame image.

C. Building of Lane Line Map

After lane line tracking, the lane line map can be built using the lane line data from multiple frames. However, directly concatenating the multi-frame data, as done in [16] [25], would introduce odometry errors into the map building. Therefore, we utilize the repetitively observed regions between adjacent frames to optimize the position of sample points to build a relatively accurate lane line map.

First, the scale of the lane line sample points in the current frame is restored using the IPM model, obtaining their positions in the vehicle coordinate: $\left[X_v^i,\ Y_v^i,\ 0\right]^T$, where X_v^i and Y_v^i are provided by the IPM model. The plane formed by the X-axis and Y-axis in the vehicle coordinate is the ground plane, and the origin is located directly below the camera's optical center. Since the depth recovery accuracy of the IPM model attenuates with increasing distance, this paper selects a rectangular region of $13m \times 7m$ in front of the vehicle as the region of interest (ROI), and lane line sample points not within the ROI will be discarded.

The historical frame sample points of the lane line are transformed from the world coordinate into the current frame vehicle coordinate based on the prior pose output by the odometry:

$$P_v = T_I^V T_i^{w-1} P_w \tag{4}$$

where $P_w = \left[x_w,\ y_w,\ z_w,\ 1\right]^T$ and $P_v = \left[x_v,\ y_v,\ z_v,\ 1\right]^T$ are the homogeneous coordinates of the sample points in world and vehicle coordinates. T_i^w is the vehicle pose. T_I^V is the external matrix of the IMU concerning the vehicle coordinate origin. Currently, both the historical and current sample points are in the vehicle coordinate system of the current

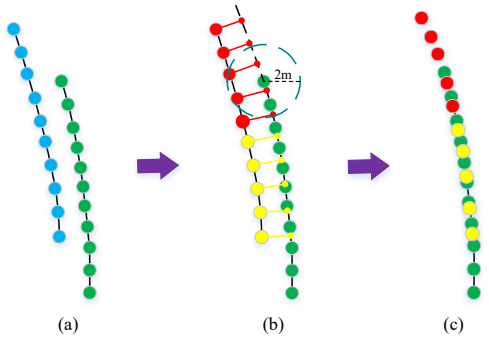


Fig. 3. The process of correcting the position of the lane line sample points. (a) and (c) show the positions of sample points before and after correction. (b) shows how to calculate the corrected positions of the sample points. The blue circle is the current frame sample points, the green circle is the historical frame sample points, the red circle is the new observed sample points, and the yellow circle is the repeated observed sample points.

frame, as shown in Fig. 3(a). It can be observed that the sample points of the current frame deviate from the curve formed by the sample points of the historical frames. Next, we use the repeated observation area to correct the position of the sample points.

Since this paper only considers lateral pose errors, we draw a perpendicular line from the current frame sample point to the lane line composed of the historical frame sample points. If the intersection point is not within the range of sample points from historical frames, it is considered a new observed point. Considering the error in scale recovery, intersection points within a distance of less than $2m$ from the range boundary are also considered new observed points. This process is illustrated in Fig. 3(b). The perpendicular point is taken as the corrected position for repeated observed sample points. For new observed sample points, the following method is used for correction:

$$\begin{bmatrix} X_v' \\ Y_v' \\ Z_v' \end{bmatrix} = \begin{bmatrix} X_v \\ Y_v \\ 0 \end{bmatrix} + \alpha_1 \begin{bmatrix} a_1 \\ b_1 \\ c_1 \end{bmatrix} + \alpha_2 \begin{bmatrix} a_2 \\ b_2 \\ c_2 \end{bmatrix} + \cdots + \alpha_n \begin{bmatrix} a_n \\ b_n \\ c_n \end{bmatrix} \quad (5)$$

$$\alpha_j = 1 - \frac{X_v^j}{\sum_{i=0}^{i=n} X_v^i} \quad (6)$$

where $\begin{bmatrix} X_v, & Y_v, & 0 \end{bmatrix}^T$ and $\begin{bmatrix} X_v', & Y_v', & Z_v' \end{bmatrix}^T$ are the position of the sample point before and after correction. $\begin{bmatrix} a_n, & b_n, & c_n \end{bmatrix}^T$ is the translation vector from the repeated observed sample point to the corrected position. Considering that the scale recovery accuracy of the IPM model decreases with distance, weights α_j are set. X_v^j is the X-axis coordinate of the repeated observed sample point in the vehicle coordinate. A larger value of X_v^j indicates a larger distance from the vehicle, resulting in a smaller value of α_j. The effect after correction is shown in Fig. 3(c).

In particular, to reduce the impact of IPM model errors, we utilize the characteristic of the same curvature of lane lines on both sides of the vehicle to correct the lane lines that are further away. The far and near lane lines are denoted as $Line_f$ and $Line_n$. First, we fix the position of the closest sample point in $Line_f$ relative to the vehicle and project this point onto $Line_n$ while recording the translation vector $\begin{bmatrix} a_l, & b_l \end{bmatrix}^T$. Then, we project the remaining sample points in $Line_f$ onto $Line_n$ and correct them using the aforementioned translation vector:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x_t \\ y_t \end{bmatrix} - \begin{bmatrix} a_l \\ b_l \end{bmatrix} \quad (7)$$

where $\begin{bmatrix} x_t, & y_t \end{bmatrix}^T$ is the position of the projected point, and $\begin{bmatrix} x', & y' \end{bmatrix}^T$ is the position of the corrected sample point. The z-coordinate of both lane lines' sample points is 0, and no correction is applied. It should be noted that this paper only corrects the lane lines on both sides of the vehicle. And, if the lane lines on both sides of the vehicle have significantly different curvatures, their extensions are considered different, and no corrections are made.

Subsequently, the corrected sample points are transformed from vehicle coordinates to world coordinates based on the prior pose from odometry, thus building an accurate lane line map. The lane line maps before and after correction are shown in Fig. 4. After optimizing the positions of the sample points, the lane line map exhibits enhanced continuity and effectively reduces mapping errors.

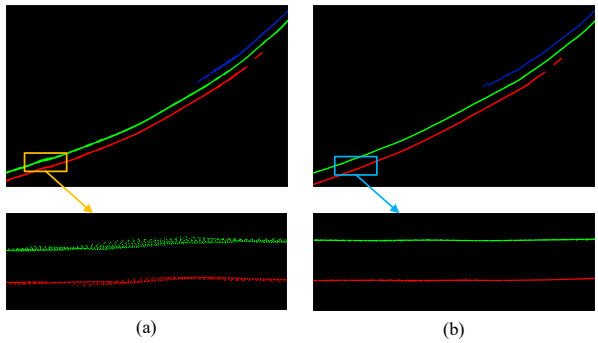


Fig. 4. The effect of sample point position correction. (a) and (b) represent the lane line maps before and after correction. These maps were built from the first 600 frames of the KAIST urban38 sequence.

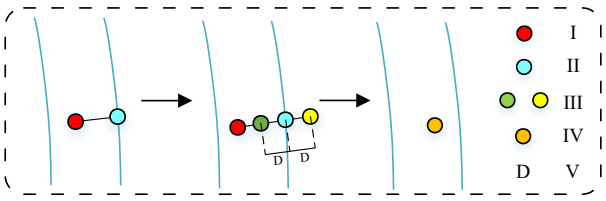


Fig. 5. The process of pose correction is based on lane lines. I is the prior position of odometry; II is the projection point of the vehicle on the lane line; III is the candidate correction position; IV is the corrected position of the vehicle; V is the lateral distance of the vehicle relative to the lane line.

D. Pose Optimization based on Lane Lines

When detecting lane changes by vehicle, we utilize the vehicle's lateral distance and lane line map for pose optimization. First, we utilize fourth-order curves to fit the lane lines. Next, we draw a perpendicular line from the prior position of the vehicle obtained from the odometry towards

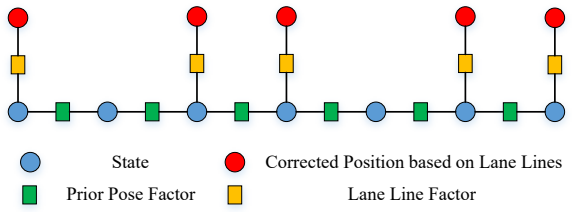
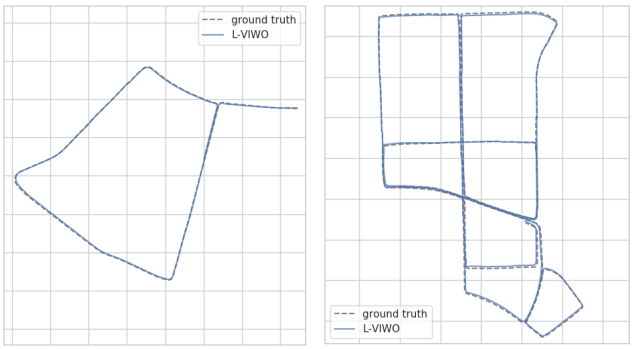


Fig. 6. Pose graph optimization model based on correction position.



(a) ubran26 (b) ubran38

Fig. 7. The ground truth and the estimated trajectories of L-VIWO on KAIST dataset.

the corresponding lane line and determine the corresponding intersection point. Then, along the perpendicular line, we search for positions that are at a distance equal to dis_b from the intersection point. We select the position on the left side of the lane line as the corrected position of the vehicle. The process of position correction is shown in Fig. 5.

Next, we build a pose graph optimization model using the corrected positions and the prior poses from odometry, as shown in Fig. 6. The red nodes are the corrected positions. The blue nodes are the state variables to be optimized. The yellow boxes denote the constraints provided by the corrected positions. It should be noted that we consider the pose transformations between adjacent frames outputted by the odometry to be relatively accurate. If the positional changes between the adjacent frames after correction deviate significantly from the positional changes from the odometry, the constraint is not introduced. The green boxes denote the constraints provided by the prior poses from odometry. The overall objective function and the residual factors are as follows:

$$\chi^* = arg\min_{\chi}\left\{\sum_{(i,j)\in O}\left\|O_{ij}\right\|^2 + \sum_{(i)\in L}\|L_i\|^2\right\} \tag{8}$$

$$O_{ij} = \begin{bmatrix}(q_i^o)^{-1}(p_j^o - p_i^o) - (q_i^l)^{-1}(p_j^l - p_i^l)\\ ((q_i^o)^{-1}q_j^o)^{-1} \otimes ((q_i^l)^{-1}q_j^l)\end{bmatrix} \tag{9}$$

$$L_i = p_{X_i}^l - p_i^l \tag{10}$$

where $\chi = [x_0, x_1, \cdots, x_n]$ is the optimization variable. $x_i = [q_i^l, p_i^l]$ is the 6-DOF pose of the vehicle. O is the set of poses from VIWO, and L is the set of corrected positions based on lane lines. O_{ij} is the residual function

TABLE I
COMPARISON OF TRAJECTORY ACCURACY ON KAIST (RMSE ATE IN M). BEST IN BOLD.

KAIST-urban	[22]	[21]	[24]	[5]	[30]	[1]	L-VIWO
26(4.0km)	12.0	11.9	14.8	16.1	32.8	7.3	**3.9**
28(11.5km)	15.4	27.8	25.0	33.1	34.7	12.3	**7.4**
38(11.4km)	11.8	16.0	33.5	43.0	55.5	32.9	**6.8**
39(11.0km)	7.5	8.0	21.3	24.0	33.4	15.6	**6.2**

TABLE II
COMPARISON OF LOCALIZATION ACCURACY BETWEEN VIWO AND L-VIWO ON KAIST (RMSE ATE IN M). BEST IN BOLD.

Method	kaist-26	kaist-28	kaist-35	kaist-36	kaist-38	kaist-39
VIWO	4.08	7.97	8.67	49.53	7.72	7.32
L-VIWO	**3.85**	**7.44**	**8.06**	**45.94**	**6.79**	**6.19**

for inter-frame relative poses, where the first row is the error in relative position, and the second row is the error in relative rotation. $(q_i^o,\ p_i^o)$ and $(q_j^o,\ p_j^o)$ are the position and orientation from odometry. L_i is the residual function constructed using the corrected position based on lane lines, and $p_{X_i}^l$ is the corrected position obtained using the lateral distance of the vehicle. Pose optimization based on lane lines can be achieved by utilizing the pose graph optimization model.

V. EVALUATION

In order to validate the effectiveness of the proposed method, we evaluated it on the KAIST Complex Urban Dataset [29]. The proposed method uses the left camera, one IMU, and one wheel encoder. The captured images, IMU measurements, and wheel encoder measurements have frequencies of 10 Hz, 100 Hz, and 100 Hz, respectively. In our experiments, the LaneATT lane line detection network was trained on an NVIDIA GeForce RTX 3080 10 GB GPU. All experiments were conducted on a laptop with an Intel(R) Core(TM) i7-10870H CPU @ 2.20GHz, 16GB RAM, and an NVIDIA GeForce RTX 3060 6GB GPU.

A. Pose Estimation Performance

The localization experiments in this paper were conducted on the urban26, urban28, urban38, and urban39 sequences of the KAIST dataset. The root mean square error (RMSE), which is the absolute trajectory error (ATE) after 6-DoF trajectory alignment, is adopted for comparison in our experiments. In addition to the proposed L-VIWO algorithm, the evaluated algorithms included VIWO [5], [21], [22], [24], VIO [30], and VO [1]. ORB-SLAM3 [1] using the stereo camera. The experimental results of [5], [21], [22], [24], [30] were provided by [22]. All the algorithms are evaluated without performing loop closure. The experimental results are shown in Table I. It can be observed that the proposed method in this paper achieves significantly better localization accuracy compared to other algorithms in the four sequences. Fig. 7 shows the estimated trajectories of L-VIWO in urban26 and urban38.

TABLE III
RESULTS OF ABLATION EXPERIMENT I (RMSE ATE IN M).

Method	kaist-26	kaist-28	kaist-35	kaist-36	kaist-38	kaist-39
w/o A,B	4.15	7.73	8.41	49.03	7.47	7.31
w/o A	4.08	7.54	8.35	48.63	7.46	7.29
w/o B	3.87	7.45	8.13	46.57	6.87	6.50
L-VIWO	**3.85**	**7.44**	**8.06**	**45.94**	**6.79**	**6.19**

Note: L-VIWO represents the complete system. "A" represents the correction method based on multi-frame data of lane lines, and "B" represents the correction method based on left-right lane line curvature constraint. "w/o" stands for removing the corresponding method.

TABLE IV
RESULTS OF ABLATION EXPERIMENT II (RMSE ATE IN M).

Method	kaist-26	kaist-28	kaist-35	kaist-36	kaist-38	kaist-39
w/o P	4.08	7.71	8.25	46.82	7.46	6.27
L-VIWO	**3.85**	**7.44**	**8.06**	**45.94**	**6.79**	**6.19**

Note: "P" represents using the degree of change in corrected position to determine whether to introduce lane line constraints.

Additionally, to further validate the effectiveness of the method, we conducted localization experiments on the KAIST dataset using the VIWO algorithm employed in this paper. The results are shown in Table II. It can be observed that the introduction of lane line constraints resulted in a substantial improvement in the system's localization accuracy. In particular, it achieves a 13% and 15% improvement in the urban38 and urban39 sequences, respectively.

B. Ablation Study

In the process of lane map building, we utilize multiple frame data of the same lane line and left-right lane line curvature constraints to correct the positions of sample points. To observe the effectiveness of these two correction methods, we conducted an ablation study. The corresponding correction methods were removed, and localization experiments were conducted again on the KAIST dataset. The results are shown in Table III. It can be observed that regardless of removing any correction method, the localization accuracy of the system will decrease. Especially in urban26, the localization accuracy of L-VIWO without the two correction methods is even lower than VIWO. The result demonstrates that the correction methods used in this paper contribute to improving the localization accuracy of the system.

Furthermore, in the pose graph optimization process, we determine whether to introduce lane line constraints by comparing position changes after correction with odometry-estimated changes. We removed this judgment process and conducted the localization experiments again, as shown in Table IV. It can be observed that not including this judgment process leads to a decrease in localization accuracy, demonstrating the effectiveness of the method.

C. Method Universality Testing

To further validate the universality and effectiveness of the proposed method, we incorporate the pose optimization method into VINS-Fusion [2] and ORB-SLAM3 [1] and observe the improvement effect in localization accuracy. VINS-Fusion using the stereo camera and IMU. ORB-SLAM3 using the stereo camera. The results are shown in Table V. It can be observed that by incorporating lane line constraints, both algorithms demonstrate improved localization accuracy across four sequences. The result demonstrates that the proposed pose optimization method is equally applicable to both VIO and VO, highlighting its universality and effectiveness.

TABLE V
RESULTS OF UNIVERSALITY EXPERIMENTS (RMSE ATE IN M).

Method	kaist-26	kaist-28	kaist-38	kaist-39
VINS-Fusion	20.92	24.33	51.21	32.83
L-VINS-Fusion	**20.14**	**23.31**	**48.27**	**30.72**
ORB-SLAM3	7.34	12.31	32.93	15.58
L-ORB-SLAM3	**6.98**	**11.58**	**30.47**	**14.19**

Note: "L" represents the introduction of lane line constraints. All the algorithms are evaluated without performing loop closure.

TABLE VI
AVERAGE RUN-TIME OF RACH MODULE.

Lane line detection	Lane line tracking	Lane line optimization	Odometry	System
31.69ms	2.45ms	0.96ms	70.26ms	74.32ms

Note: All results are the average of 30 experimental results in the KAIST dataset.

D. Time Analysis

To test the real-time performance of the system, we conducted time-consuming tests on each part of the system, and the results are shown in Table VI. It should be noted that the pose graph optimization based on lane lines will only be performed when a lane change is detected. It processes multiple frames of data together and runs in parallel with the system. Therefore, its impact was only considered within the overall system timing, and no separate time testing was conducted for it.

The odometry part takes 70.26ms per frame, and after introducing the lane line constraints, the system takes 74.32ms per frame, only increasing by 4.06ms. At the same time, the lane line detection part runs in parallel with the odometry part and takes much less time than the odometry part, so it does not affect the overall algorithm's run time. The camera capture rate is 10Hz, which means the system can complete the data processing before obtaining the next frame image, indicating that the system can run in real time.

VI. CONCLUSIONS

This paper proposes L-VIWO, a Visual-Inertial-Wheel Odometry based on lane lines, that utilizes the lateral constraints of lane lines to reduce accumulated errors in odometry. Localization experiments on a general dataset have demonstrated the effectiveness of the proposed method. In future work, we will consider constraints from other road signs to improve our system further.

REFERENCES

[1] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. M. Montiel, and J. D. Tardós, "Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam," *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 1874–1890, 2021.

[2] T. Qin, S. Cao, J. Pan, and S. Shen, "A general optimization-based framework for global pose estimation with multiple sensors," *arXiv preprint arXiv:1901.03642*, 2019.

[3] K. J. Wu, C. X. Guo, G. Georgiou, and S. I. Roumeliotis, "Vins on wheels," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5155–5162, 2017.

[4] D. Li, K. Eckenhoff, K. Wu, Y. Wang, R. Xiong, and G. Huang, "Gyro-aided camera-odometer online calibration and localization," in *2017 American Control Conference (ACC)*, pp. 3579–3586, 2017.

[5] M. Zhang, Y. Chen, and M. Li, "Vision-aided localization for ground robots," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2455–2461, 2019.

[6] R. Kang, L. Xiong, M. Xu, J. Zhao, and P. Zhang, "Vins-vehicle: A tightly-coupled vehicle dynamics extension to visual-inertial state estimator," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pp. 3593–3600, 2019.

[7] Z. Dang, T. Wang, and F. Pang, "Tightly-coupled data fusion of vins and odometer based on wheel slip estimation," in *2018 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 1613–1619, 2018.

[8] S. Cao, X. Lu, and S. Shen, "Gvins: Tightly coupled gnss–visual–inertial fusion for smooth and consistent state estimation," *IEEE Transactions on Robotics*, vol. 38, no. 4, pp. 2004–2021, 2022.

[9] X. Niu, H. Tang, T. Zhang, J. Fan, and J. Liu, "Ic-gvins: A robust, real-time, ins-centric gnss-visual-inertial navigation system," *IEEE Robotics and Automation Letters*, vol. 8, no. 1, pp. 216–223, 2023.

[10] M. Schreiber, H. Königshof, A.-M. Hellmund, and C. Stiller, "Vehicle localization with tightly coupled gnss and visual odometry," in *2016 IEEE Intelligent Vehicles Symposium (IV)*, pp. 858–863, 2016.

[11] K. Jo, K. Chu, and M. Sunwoo, "Interacting multiple model filter-based sensor fusion of gps with in-vehicle sensors for real-time vehicle positioning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 1, pp. 329–343, 2012.

[12] R. Jin, J. Liu, H. Zhang, and X. Niu, "Fast and accurate initialization for monocular vision/ins/gnss integrated system on land vehicle," *IEEE Sensors Journal*, vol. 21, no. 22, pp. 26074–26085, 2021.

[13] L. Xiong, R. Kang, J. Zhao, P. Zhang, M. Xu, R. Ju, C. Ye, and T. Feng, "G-vido: A vehicle dynamics and intermittent gnss-aided visual-inertial state estimator for autonomous driving," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 8, pp. 11845–11861, 2022.

[14] Y. Zhang, H. Zhang, G. Wang, J. Yang, and J.-N. Hwang, "Bundle adjustment for monocular visual odometry based on detections of traffic signs," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 1, pp. 151–162, 2020.

[15] Y. Zhou, X. Li, S. Li, and X. Wang, "Visual mapping and localization system based on compact instance-level road markings with spatial uncertainty," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 10802–10809, 2022.

[16] T. Qin, Y. Zheng, T. Chen, Y. Chen, and Q. Su, "A light-weight semantic map for visual localization towards autonomous driving," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 11248–11254, 2021.

[17] H. Wang, C. Xue, Y. Zhou, F. Wen, and H. Zhang, "Visual semantic localization based on hd map for autonomous vehicles in urban scenarios," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 11255–11261, 2021.

[18] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós, "Orb-slam: A versatile and accurate monocular slam system," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[19] R. Mur-Artal and J. D. Tardós, "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras," *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.

[20] T. Qin, P. Li, and S. Shen, "Vins-mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 1004–1020, 2018.

[21] J. Liu, W. Gao, and Z. Hu, "Visual-inertial odometry tightly coupled with wheel encoder adopting robust initialization and online extrinsic calibration," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5391–5397, 2019.

[22] J. Liu, W. Gao, and Z. Hu, "Bidirectional trajectory computation for odometer-aided visual-inertial slam," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1670–1677, 2021.

[23] W. Lee, K. Eckenhoff, Y. Yang, P. Geneva, and G. Huang, "Visual-inertial-wheel odometry with online calibration," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4559–4566, 2020.

[24] M. Zhang, X. Zuo, Y. Chen, Y. Liu, and M. Li, "Pose estimation for ground robots: On manifold representation, integration, reparameterization, and optimization," *IEEE Transactions on Robotics*, vol. 37, no. 4, pp. 1081–1099, 2021.

[25] T. Qin, T. Chen, Y. Chen, and Q. Su, "Avp-slam: Semantic visual mapping and localization for autonomous vehicles in the parking lot," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5939–5945, 2020.

[26] W. Cheng, S. Yang, M. Zhou, Z. Liu, Y. Chen, and M. Li, "Road mapping and localization using sparse semantic visual features," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 8118–8125, 2021.

[27] L. Tabelini, R. Berriel, T. M. Paixão, C. Badue, A. F. De Souza, and T. Oliveira-Santos, "Keep your eyes on the lane: Real-time attention-guided lane detection," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 294–302, 2021.

[28] J. Jeong and A. Kim, "Adaptive inverse perspective mapping for lane map generation with slam," in *2016 13th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, pp. 38–41, 2016.

[29] J. Jeong, Y. Cho, Y.-S. Shin, H. Roh, and A. Kim, "Complex urban dataset with multi-level sensors from highly diverse urban environments," *The International Journal of Robotics Research*, vol. 38, no. 6, pp. 642–657, 2019.

[30] M. Li and A. I. Mourikis, "Optimization-based estimator design for vision-aided inertial navigation," in *Robotics: Science and Systems*, vol. 8, pp. 241–248, Berlin Germany, 2013.