Road-SLAM : Road Marking based SLAM with Lane-level Accuracy

Jinyong Jeong, Younggun Cho, and Ayoung Kim[1]

***Abstract*— In this paper, we propose the Road-SLAM algorithm, which robustly exploits road markings obtained from camera images. Road markings are well categorized and informative but susceptible to visual aliasing for global localization. To enable loop-closures using road marking matching, our method defines a feature consisting of road markings and surrounding lanes as a sub-map. The proposed method uses random forest method to improve the accuracy of matching using a sub-map containing road information. The random forest classifies road markings into six classes and only incorporates informative classes to avoid ambiguity. The proposed method is validated by comparing the SLAM result with RTK-Global Positioning System (GPS) data. Accurate loop detection improves global accuracy by compensating for cumulative errors in odometry sensors. This method achieved an average global accuracy of $1.098\ \mathrm{m}$ over $4.7\ \mathrm{km}$ of path length, while running at real-time performance.**

I. Introduction

Accurate real-time localization is one of the most fundamental technologies for autonomous vehicles in many road based environments. Many researchers have been developing algorithms for more accurate pose estimation using information from various sensors [1, 2, 3, 4, 5]. Among the many type of localization sensors, GPS is the most popular but has a critical weakness regarding availability in urban areas. GPS signals suffer from multipath and blackout issues, especially in highly complex urban canyons [6]. Furthermore, consumer-level single GPS merely provides meter-level accuracy, which is not sufficient for lane-level localization. Other sensors for estimating the position of vehicles include in-vehicle sensors such as an encoder and an Inertial Measurement Unit (IMU). The accuracy of these sensors is relatively high compared to GPS, but they can only estimate the local position and inevitably have cumulative errors. Despite researches on the pose estimation using these two sensor types [7], lane-level positional accuracy has hardly been achieved due to potential errors [8, 9].

In order to achieve accuracy in global localization, many studies have examined Simultaneous Localization and Mapping (SLAM) using additional sensor information. Recent research commonly uses aerial images provided by several companies. Using airborne imaging, these methods are capable of capturing large areas by extracting lane and road information for pose estimation [10, 11, 12, 13]. However, the accuracy of pose estimation based on aerial images depends on the accuracy of the image data. As addressed by [6], since the actual position measured by RTK GPS differs from the position in the aerial image, this error is reflected in the localization results.

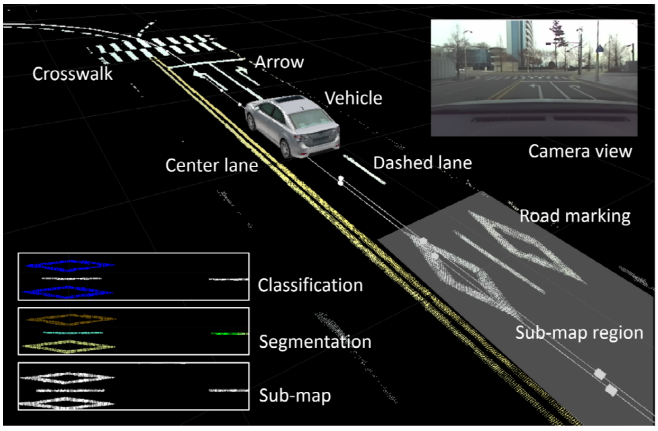


Fig. 1: Illustration of Road-SLAM. The road markings in images are transformed into 3D point clouds and classified into six classes through segmentation and classification processes. Using this information, sub-maps containing the relationships among markings are created and applied to the SLAM.

Many groups use on-board imaging sensors with prior 3D information. Schreiber et al., [14] obtained 3D data using a down-looking camera and a Light Detection and Ranging (LiDAR) sensor, and extracted lane information on the road manually. These information was used as prior information to estimate the lane level position of vehicle. Other groups [15, 5] used the 3D data generated from a camera and LiDAR sensor as the prior information. These papers, in particular, used mutual information between an image from synthetic 3D data and an image at the current frame to estimate vehicle pose. However, a relatively expensive sensor (e.g., 3D LiDAR) is required to create such prior information, and manual work is often necessary to obtain accurate labeled data.

To avoid the high cost of aerial sensors, studies have focused on vision-only systems and estimated poses from on-board camera images [16]. This line of research is most similar to ours, which detect road markings and lanes on pavement converted into features for lane-level localization

[1]J. Jeong, Y. Cho and A. Kim are with the Department of Civil and Environmental Engineering, KAIST, 291 Daehak-ro, Yuseong-gu, Daejeon 34141, Republic of Korea [jjy0923,yg.cho,ayoungk]@kaist.ac.kr

This material is supported by the Korea MOTIE under Industrial Technology Innovation Program (No.10051867), and is also supported by 'SLAM-based Lane Map Generation for Complex Urban Environment' project funded by NaverLabs Corporation. J. Jeong was financially supported by Korea MOLIT via 'U-City Master and Doctor Course Grant Program' and KAIST Institute for Robotics.

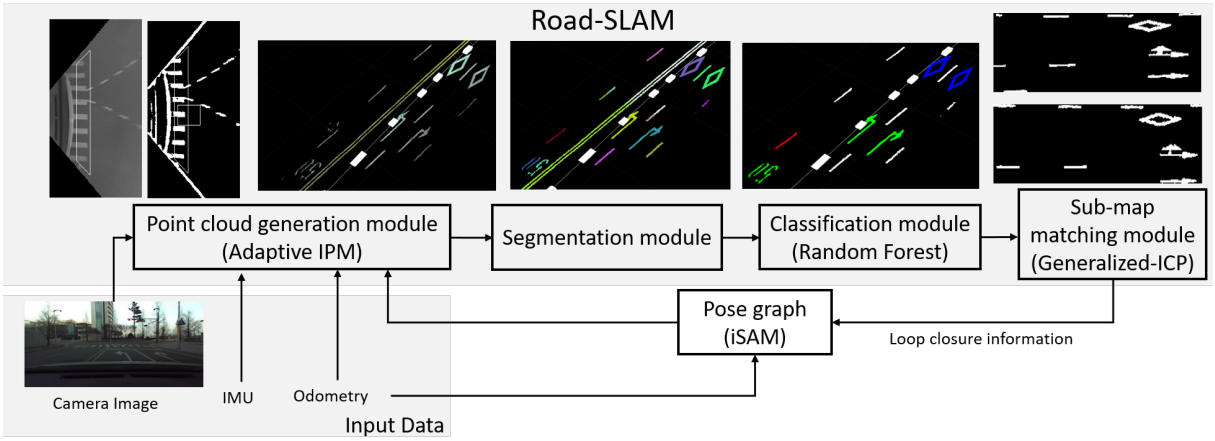


Fig. 2: Road-SLAM algorithm pipeline. Given a road image, each module is performed in a thread working in real-time.

and SLAM. Ranganathan et al., [17] extracted the corners from each road marking and estimated exact positions by comparing them with a lightweight prior map that was previously generated using the high-accuracy GPS. Rehder et al., [18] used the camera images and odometry to generate a local grid map and estimated the ego-motion of a vehicle by matching each map. In similar studies, road marking features are stored and used for pose error correction by comparing the current road features to the previously saved features [19, 20]. The main limitation of these methods is the high ambiguity caused when the markings and the lanes have similar shapes and repetitive patterns.

To tackle this ambiguity issue, the proposed method trains random forest trees to classify only the distinguishable road markings. This classification substantially improves matching performance by avoiding visual aliasing from markings with similar shapes. We then recognize a place by matching sub-maps constructed from these salient markings and the surrounding lanes. By doing so, we can recognize places using only visual road markings, which are less sensitive to environmental changes (e.g., lighting, time, and the surroundings). The overall SLAM implementation is as shown in Fig. 1 and has the following contributions:

- Robust matching using informative feature selection
- Real-time performance with fully automated match detection
- Accurate localization (cm-level) from visual loop-closure

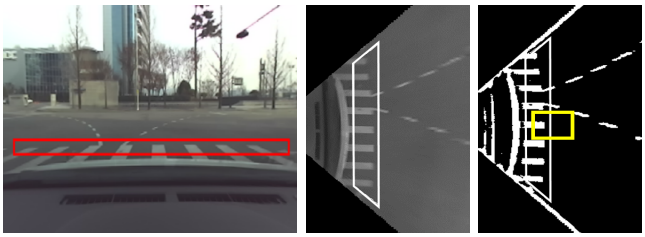
The rest of the paper is organized as follows. Section II describes our sensor system and provides an overview of the proposed method. The details of the proposed method are described in Section III. The evaluation of our results is shown in Section IV, and Section V finally concludes with a discussion.

II. System Overview

A mapping system on a car-like platform was used for the experiments. The platform is shown in Fig. 3 and equipped with a forward-looking ZED camera, an IMU, and two wheel encoders. Two in-vehicle sensors, an IMU and a wheel encoder, are used for navigation, and images are used for the road markings and lane detection. A RTK-GPS is mounted on the vehicle only to set the position and direction of the vehicle initially as well as to provide ground truth. Although the ZED camera also provides a high-precision depth image when used with Graphics Processing Unit (GPU), we use it as a mono camera (10 Hz) without using GPU in this paper.

The overall algorithm architecture is shown in Fig. 2 and `Road-SLAM.mp4`. First, the point cloud generation module creates an Inverse Perspective Mapping (IPM) image that removes the perspective effect from the image obtained from the camera. To extract the points belonging to the road marking, the IPM image is binarized with an adaptive binarization algorithm [21]. Using the generated point cloud, the segmentation module divides the sub-map obtained by accumulating a point cloud into several segments. Then, each segment constituting the sub-map is classified into six classes using a machine learning approach called random forest. To improve the matching accuracy of the sub-map, only the segments that can reduce the ambiguity of the sub-map are

Fig. 3: The sensor system is equipped with a ZED camera, a MTi IMU, two wheel encoders and a RTK GPS. The RTK GPS is only used to initially set the direction and position of the vehicle at the start of the algorithm.



(a) Camera Image (b) IPM Image (c) Binary Image

Fig. 4: Adaptive IPM and binarization applied for the proposed method. The red box in (a) is the ROI of the original image used to create a point cloud. The white box in (b) and (c) is the converted ROI from the camera image to the IPM image. The sub-map creation criteria is determined by evaluating the pixels in the yellow box (c).

selected and included in the final sub-map for the matching process. Lastly, the sub-map matching module detects the loop through point matching of the sub-maps determined in the previous pipeline process.

III. Road SLAM using Robust Sub-map Matching

This section illustrates the procedure for generating a sub-map. Pre-processed and binarized IPM are first segmented and classified to construct a sub-map. The sub-map is a matching candidate group for loop detection, and this sub-map generation module carefully selects only the informative markings and lanes in a sub-map.

A. Point Cloud Generation

The first module generates the points belonging to the road marking using an IPM algorithm. IPM creates a bird's-eye image by removing the perspective effect from the image. Since the IPM equation is induced on the assumption that the road ahead of the vehicle is flat, the small pitch motion of a vehicle is likely to cause a large distortion in the IPM image. In order to overcome this problem, the adaptive IPM model [22] is applied, considering the pitch motion of the camera. The adaptive binarization algorithm [21] is then applied to filter only informative markings on the road. The results of the IPM and binarization process are shown in Fig. 4.

Followed by this image pre-processing, the binarized points are fed into the three-dimensional (3D) point cloud generation phase. In this conversion, we restrict the Region of Interest (ROI) near to the camera to avoid large perspective distortion caused by the IPM, as in Fig. 4. Projecting binarized points over the trajectory induced by navigational sensors (e.g., wheel encoder and IMU) results in a point cloud of lanes and markings.

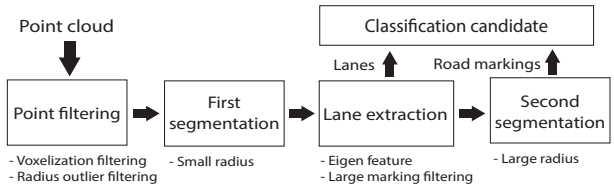


Fig. 5: Point segmentation process. Through the filtering and segmentation processes, each road marking is precisely segmented.

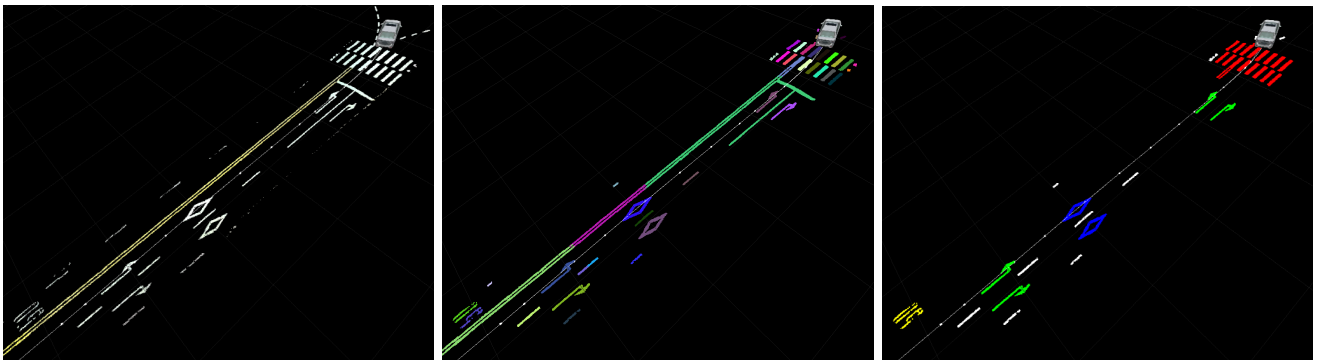
The sub-map is a map consisting of 3D road marking points in the vehicle local coordinate system for loop detection as shown in Fig. 8. The sub-map is generated when a road marking is detected in the vehicle's traveling direction. Therefore, the point cloud generation module detects the road marking by checking the number of points in the ROI, which is the yellow box in Fig. 4c. The algorithm, then, stores the accumulated 3D point cloud as one initial sub-map when it is detected.

B. Road Marking Segmentation

Given the binarized point cloud, the segmentation is followed focusing on the two aspects of the road markings. First, not all road markings are meaningful for pose estimation. For example, the center lane fails to capture motion along the line and is thus limited to estimating a full six Degree Of Freedom (DOF) transformation; a crosswalk has abundant feature points, but the repetitive patterns are confusing even under a small IPM distortion. In the proposed method, we have decided to remove these less informative elements and to increase the accuracy of the matching by applying segmentation. We found that dashed lanes, arrows, road markings, and numbers can considered informative for matching.

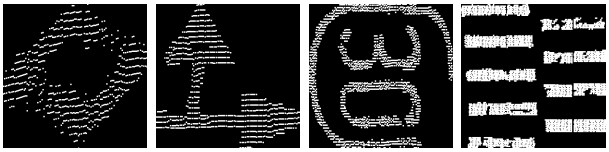
Second, several road markings and lanes are often captured in the same scene. As an example, Fig. 6a shows a sample road map consisting of points extracted from the binarized IPM image. As shown in the figure, the diamond-shaped road markings and the dashed lane are located very close to each other. In the case of numbers on the road, each number is spaced apart, and the dashed lane may be close to these numbers. When the road marking is near the lane element, we apply a two-stage segmentation process by adapting the coarse-to-fine concept. Small segments are all segmented to exclude lanes first. Then, using only segments with no lane elements, we apply a larger segmentation to merge road markings into a single cluster.

The end-to-end segmentation module is described in Fig. 5. Since accumulated points are obtained from multiple images, many points tend to be cluttered and gathered around one segment. To resolve this issue, the segmentation module starts with (*i*) voxelization to efficiently represent duplicated points, and (*ii*) radius outlier filtering to remove noise. Our aims of first segmentation is to exclude large clusters (e.g., center lines and stop lines) and dashed lanes in order to clearly detect salient road markings. The initial segmentation process divides the points by a small radius. By doing so, large features will be divided into a single segment, and dashed lanes will be clustered into several groups. These initial segments are checked using the linearity of the eigen feature to detect and classify lanes precedently. Then, the follow-up segmentation is performed using relatively large



(a) Colored Point Cloud (b) Segmented Point Cloud (c) Classified Point Cloud

Fig. 6: The segmentation and classification results. Resulting segments are color-coded by groups. The center lane and stop lane are divided into one segment as in (Fig. 6b). These large segments can be removed because we know the actual size and length of the road markings. In the classification result, the color of the group represents the class to which the segment belongs to (i.e., green: arrow, blue: road marking, yellow: number, red: cross walk).



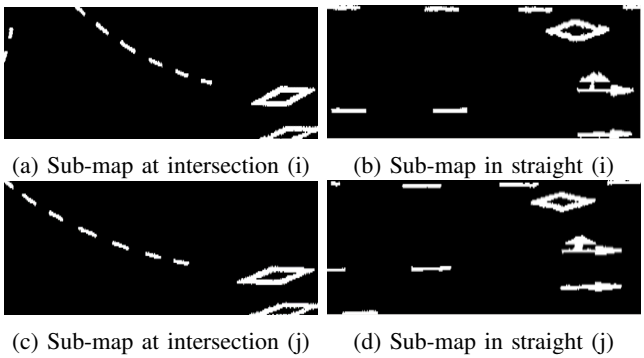
(a) Marking (b) Arrow (c) Number (d) Crosswalk

Fig. 7: Road marking segments. Each segment is classified into individual classes through a random forest method.

radius values for the points where the lanes and large segments are excluded. As a result, segments that are separated by a certain distance, such as numbers or crosswalks, are combined into one segment. These segments are also registered in the candidate group for classification.

C. Classification

Once segmented markings are established, the classification thread distinguishes segments that are less affected by IPM image distortion for robust matching. Using previously segmented road markings (Fig. 7), we extract a features via Ensemble of Shape Function (ESF) [23]. The ESF is a 640-tuple histogram of the shape function defined by three parameters, the distance between two randomly selected points, the area between the three points, and the angle between the three randomly selected points. The feature vectors constructed by ESF are used as inputs to the random forest to distinguish each segment. In the training phase, the maximum depth of the random forest is set to 100. Classification candidate segments obtained from the segmentation process are used for the training and testing of random forests. The output of the random forest consists of six classes: road markings, numbers, arrows, lanes, crosswalk, and others. Finally, among the segments classified as random forests, only informative classes (e.g., road markings, numbers, arrows, and lanes) are included in the sub-map for the matching process. Fig. 6c shows the results of the distinguishing of each segment using a random forest. Red, green, blue and yellow indicate crosswalk, arrow, road marking, and number respectively. In sorting appropriate segments for the sub-map, the random forest was selected due to its deterministic characteristic once trained.



(a) Sub-map at intersection (i) (b) Sub-map in straight (i)

(c) Sub-map at intersection (j) (d) Sub-map in straight (j)

Fig. 8: Matching candidate sub-maps pairs when a place is revisited. (a) and (b) are the previously obtained sub-maps, and (d) represents the sub-maps when revisiting.

D. GICP based Sub-map Matching

When a place is revisited, we perform sub-map matching given a loop-closure candidate pair. This candidate proposal is selected using the distance threshold from the current vehicle position. Examples of these pairs are shown in Fig. 8. The matching process uses the generalized Iterated Closest Point (ICP) algorithm [24] to select the sub-map with the minimum matching cost among candidates. If the ICP matching is successful, the relative position of the vehicle between two sub-maps is calculated using the ICP result. The calculated relative pose is passed to the Incremental Smoothing and Mapping (iSAM) pose graph as the constraint information to create a loop [25].

E. Road-SLAM

The proposed method, Road SLAM, is based on pose graph-based approach that minimizes the error function as below.

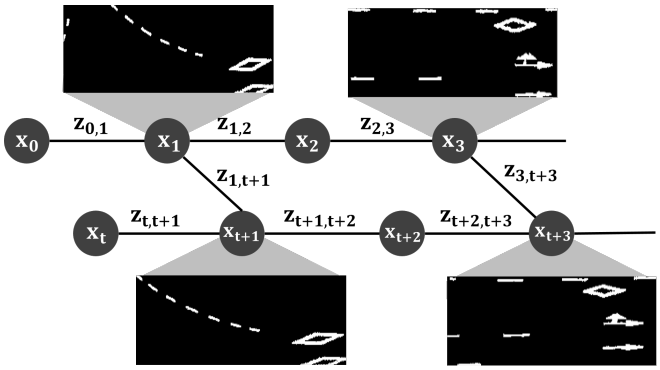


Fig. 9: Pose-graph SLAM illustration. Sub-maps used in Road-SLAM composed of road markings.

$$X^* = \arg\min_X \sum_t \|f(\mathbf{x}_t, \mathbf{x}_{t+1}) - \mathbf{z}_{t,t+1}\|^2_{\Sigma_t} + \sum_{i,j} \|f(\mathbf{x}_i, \mathbf{x}_j) - \mathbf{z}_{i,j}\|^2_{\Sigma_{i,j}} \quad (1)$$

, where $\mathbf{x}_t = [x, y, z, \pi, \theta, \psi]^\top$ represents the pose of the vehicle at time t and the augmented representation n poses are presented as $X = [\mathbf{x}_1^\top, \cdots, \mathbf{x}_t^\top, \cdots, \mathbf{x}_n^\top]^\top$. The function $f(\cdot, \cdot)$ is the state transition model for two poses. Constraints of the pose graph represents the relative 6 DOF motion from odometry (temporal links) $\mathbf{z}_{t,t+1} = [x_{t,t+1}, y_{t,t+1}, z_{t,t+1}, \pi_{t,t+1}, \theta_{t,t+1}, \psi_{t,t+1}]^\top$ and ICP based matching (non-temporal links) $\mathbf{z}_{i,j} = [x_{i,j}, y_{i,j}, z_{i,j}, \pi_{i,j}, \theta_{i,j}, \psi_{i,j}]^\top$. The covariances of odometry and loop closure obtained from camera measurements are denoted as Σ_t and $\Sigma_{i,j}$.

For odometry constraints generation, the position of the vehicle is calculated by synchronizing all the sensors based on the time when the camera image is obtained. $\mathbf{x}_t$ is the position of the vehicle. The 6 DOF transformation ($\mathbf{z}_{t,t+1}$) between the nodes is obtained from the vehicle wheel encoder and the IMU sensor. The proposed method detects the loop through the matching process between the sub-maps when the vehicle re-visit the same place it passed before.

The overall SLAM framework is illustrate in Fig. 9. Black circle dots and edges indicate vehicle nodes (poses) and links (constraints). Temporal links are constructed by odometry measurements, and non-temporal links, which cross the sequential nodes, are generated by sub-map based links. Sub-maps for corresponding nodes are shown as image snapshots.

IV. Experiments and Results

We validate the proposed method using real-world data obtained from a car-like mapping platform. Our target environment is complex urban roads rather than a highway. An average vehicle speed of 50 to 60 km/h was used, which has very little impact on the performance of the algorithm. The code was implemented using an on-board PC (Intel i7-6700, 16G RAM). For incoming camera images (10 Hz) and navigational sensors (100 Hz), the algorithm runs in real-time without using a GPU. The overall trajectory and experimental area (600 m×400 m) are shown in Fig. 11a. From the experiment, the proposed algorithm was able to achieve mean error of 1.0987 m over 4.7 km of the travel distance.

A. Evaluation Criteria

For the evaluation, we use the vehicle RTK-GPS for a ground truth. Fig. 11a shows the experimental route of the vehicle. The green and red dots are the fixed and floating states of the RTK-GPS, respectively. The error in the fixed state is about 20 mm on average, and the error in the floating state ranged from 20 mm to 1 m. The blue dots are normal GPS conditions depending on the satellite condition and have meter-level error. In this paper, for accurate quantitative analysis, the accuracy was compared only when the RTK-GPS status was fixed.

B. Classification using Random Forest

This section evaluates the results of the random forest used to select the elements that constitute the sub-map. The training data for the random forest were obtained from about 25 km of the data collection using the same mapping system. The extracted data using ESF features were divided into six classes by hand-labeling (i.e., road marking (1), number (2), arrow (3), lane (4), crosswalk (5), and others (6)).

Table. I shows the classification result of the datasets. When examining individual classification error, the classifier accuracy is about 81.92%. For example, arrows are often recognized as lanes because the shape of the arrow that represents straightening is similar to a dashed lane. However, the purpose of our classification is not increased classification accuracy but robust matching. To achieve this goal, we use the classification mainly to determine sub-map inclusion. The classification results are largely divided into two classes and included or removed from sub-map. Specifically, the classifier of the proposed algorithm is used to remove crosswalk and other error-prone classes (e.g., others). The road markings in the blue area of table I are included in the sub-map, and the markings in the red areas are eliminated from the sub-map. With these criteria, the accuracy of the classifier is increased to about 98% because only six cases are misclassified (i.e., two crosswalks are classified as lanes and four lanes are classified as crosswalks).

TABLE I: Similarity matrix of random forest classifier. Marking and cross are abbreviations of road marking and crosswalk. Row indicates the actual class and column is the result of random forest classifier.

	Classification Result					
	marking	number	arrow	lane	cross	others
marking	33	7	2	0	0	0
number	0	12	8	0	0	0
arrow	0	0	87	30	0	0
lane	0	0	0	2	4	0
cross	0	0	0	2	85	0
others	0	0	0	0	37	189

TABLE II: Comparison of accuracy improvement in sub-map filtering by classification result. No filtering is the result of matching all points obtained from a binarized IPM image.

	Loop Detection				Mean
	candidate	deny	success	failure	error [m]
Rehder [18]	0	0	0	0	41.0293
No filtering	36	15	15	6	95.0159
Proposed	32	18	14	0	1.0984

C. Effect of Filtering

To evaluate the effects of eliminating unnecessary elements through overall filtering process, we compare the process to another road marking-based localization approach by Rehder [18]. Rehder [18] obtained lane information from a camera in an artificially created track and detected the loop. In their paper, the relative positions of the overlapping portions of the target point clouds were calculated using the ICP algorithm. For successful ICP measurements, very accurate odometry should be assumed when using only overlapping information of lanes, which is likely to fail in a larger urban environment due to the accumulated odometry error.

Unlike [18], the proposed method targets larger scale urban environments with real-world image data, and it successfully achieved fully automated and robust sub-map matching for SLAM implementation. The comparison results are summarized in Table. II. We compare [18] and the proposed method both with and without filtering. In the table, *deny* denotes cases where matching does not occur in the matching candidate, *success* means that the matching is successful, and *failure* indicates registering a wrong matching. More matches occurs when not applying filtering but mostly for failure cases (i.e., 15 correct detections and 6 incorrect detections). We note that the matching candidate should be chosen carefully because the entire graph is distorted even if the wrong loop is detected only once. Without filtering, the pose graph was greatly distorted because six false loops occurred. In the case of the proposed method with the filtering process, since the elements that cause error were eliminated through classification, the loop was detected correctly, and the mean error was much smaller than those of the other two cases.

D. Road-SLAM Result

The SLAM result using the proposed method is shown in Fig. 11 in the comparison to odometry based map (Fig. 11c). This large improvement occurs when cumulative odometry errors are compensated for with accurate loop detection using only road markings. To qualitatively assess the accuracy of the localization, we present back-projected road map over the loop-closure area. By overlaying the road-marking points on every revisit, we evaluate localization accuracy from the consistency of the map points. Fig. 12a to Fig. 12d show the areas where the loop detection occurred (green area). To assess the accuracy of the loop detection, road marking was generated based on the corrected vehicle position without further processing. As can be seen in Fig. 12a to 12d, the road markings near the loop closure are overlapped correctly, even though the vehicle has repeatedly passed through the same area. On the other hand, Fig. 12e to Fig. 12h show the areas where the loop detection failed. Especially in the case of the intersection, the loop detection hardly took place, compared to other areas, because the crosswalks were removed for the accuracy of the loop detection. Also, in Fig. 12h, the road marking color was very cloudy, so the loop was not detected. Even for the area of Fig. 12e and Fig. 12f, the maximum error is below $2.0\ \mathrm{m}$ because the drift is corrected by nearby loop-closures.

Fig. 10 shows the error calculated from the comparison of the paths calculated through SLAM and RTK-GPS data. The blue line is the vehicle's travel path, and the colored dot indicates the RTK-GPS data at a fixed state with average error of about $2\ \mathrm{cm}$. In the northern part of the area where the data were acquired, the fixed state of RTK-GPS hardly occurred because there were many mountains and high-rise buildings. Based on the RTK-GPS data, the maximum error is about $3\ \mathrm{m}$ over $4.7\ \mathrm{km}$ of path length. We believe that adding other measurements such as in [6] will alleviate this issue. However, to clearly show the improvements from road marking sub-map matching, we intentionally used minimal navigation measurements in the tests and focused on the sub-map matching improvement over large loop-closures.

E. Computation Time

To verify the real-time performance of Road SLAM, Table. III summarizes the computation time for each module. Both average and maximum time taken for each module are listed in the table. Note that the adaptive IPM is performed whenever the image data is captured, whereas the other module only occurs when the sub-map is created.

TABLE III: Computation time for each pipeline of Road-SLAM. The number of points at the maximum cost was 54,457.

Time (sec)	IPM	Segmentation	Classification	Matching
Avg.	0.01917	0.11467	0.16203	0.02119
Max.	0.05234	0.18171	0.26312	0.06385

Although the maximum computation time for IPM is reported as $0.05234\ \mathrm{sec}$, the majority of frames (98.21%) revealed a computation time between $0.01\ \mathrm{sec}$ and $0.03\ \mathrm{sec}$. Computational cost for adaptive IPM is fairly constant but critical for the real-time performance, as it runs for every frame. On the other hand, frames in other modules vary with respect to the number of points in a sub-map. A sub-map consists of 80 frames on average, and this guarantees real-time operation with increased computational time. Overall, each process is performed in each thread in parallel, and the algorithm of the entire Road-SLAM has been proven to run at 30Hz.

V. Conclusion

In this paper, we present a SLAM algorithm that uses only a camera sensor and utilizes road marking information that

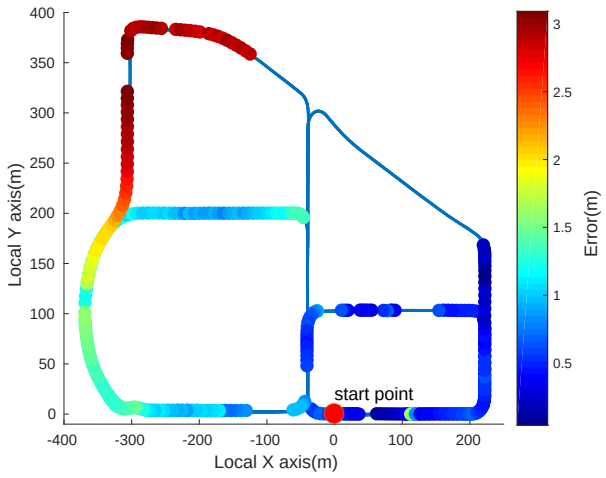


Fig. 10: Accuracy analysis of the proposed method. The blue line is the final result path of the algorithm, and the colored dots are the fixed RTK-GPS data. The error values between SLAM and RTK-GPS are color-coded by magnitude.

is robust to light and environmental changes. The accuracy of the SLAM algorithm can be improved by classifying and eliminating elements that increase the ambiguity of loop detection among various road markings using random forest. In addition, despite the use of a dead reckoning sensor without a global position sensor, high global position accuracy was achieved through very accurate loop detection. This result also suggests that localization can be achieved by using preliminary information about road markings. We found that the influence of the shadow of surrounding objects can be significant in some cases. Our future work is toward a light condition invariant algorithm following a similar line of research [20].

References

[1] Isaac Skog and Peter Handel. In-car positioning and navigation technologiesa survey. *IEEE Transactions on Intelligent Transportation Systems*, 10(1):4–21, 2009.

[2] Marcus A Brubaker, Andreas Geiger, and Raquel Urtasun. Map-based probabilistic visual self-localization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(4):652–665, 2016.

[3] Henning Lategahn and Christoph Stiller. Vision-only localization. *IEEE Transactions on Intelligent Transportation Systems*, 15(3):1246–1257, 2014.

[4] Ryan W Wolcott and Ryan M Eustice. Fast LiDAR localization using multiresolution gaussian mixture maps. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2814–2821. IEEE, 2015.

[5] Ryan W Wolcott and Ryan M Eustice. Visual localization within lidar maps for automated urban driving. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 176–183. IEEE, 2014.

[6] Hyunchul Roh, Jinyong Jeong, Younggun Cho, and Ayoung Kim. Accurate mobile urban mapping via digital map-based SLAM. *Sensors*, 16(8):1315, 2016.

[7] Karl Berntorp. Joint wheel-slip and vehicle-motion estimation based on inertial, GPS, and wheel-speed sensors. *IEEE Transactions on Control Systems Technology*, 24(3):1020–1027, 2016.

[8] Chris C Ward and Karl Iagnemma. A dynamic-model-based wheel slip detector for mobile robots on outdoor terrain. *IEEE Transactions on Robotics*, 24(4):821–831, 2008.

[9] Damrongrit Piyabongkarn, Rajesh Rajamani, John A Grogg, and Jae Y Lew. Development and experimental evaluation of a slip angle estimator for vehicle stability control. *IEEE Transactions on Control Systems Technology*, 17(1):78–88, 2009.

[10] Toni Heidenreich, Jens Spehr, and Christoph Stiller. LaneSLAM–simultaneous pose and lane estimation using maps with lane-level accuracy. In *Proceedings of the IEEE Intelligent Transportation Systems Conference*, pages 2512–2517. IEEE, 2015.

[11] Oliver Pink, Frank Moosmann, and Alexander Bachmann. Visual features for vehicle localization and ego-motion estimation. In *Proceedings of the IEEE Intelligent Vehicle Symposium*, pages 254–260. IEEE, 2009.

[12] Dixiao Cui, Jianru Xue, and Nanning Zheng. Real-time global localization of robotic cars in lane level via lane marking detection and shape registration. *IEEE Transactions on Intelligent Transportation Systems*, 17(4):1039–1050, 2016.

[13] Oliver Pink and Christoph Stiller. Automated map generation from aerial images for precise vehicle localization. In *Proceedings of the IEEE Intelligent Transportation Systems Conference*, pages 1517–1522. IEEE, 2010.

[14] Markus Schreiber, Carsten Knöppel, and Uwe Franke. Laneloc: Lane marking based localization using highly accurate maps. In *Proceedings of the IEEE Intelligent Vehicle Symposium*, pages 449–454. IEEE, 2013.

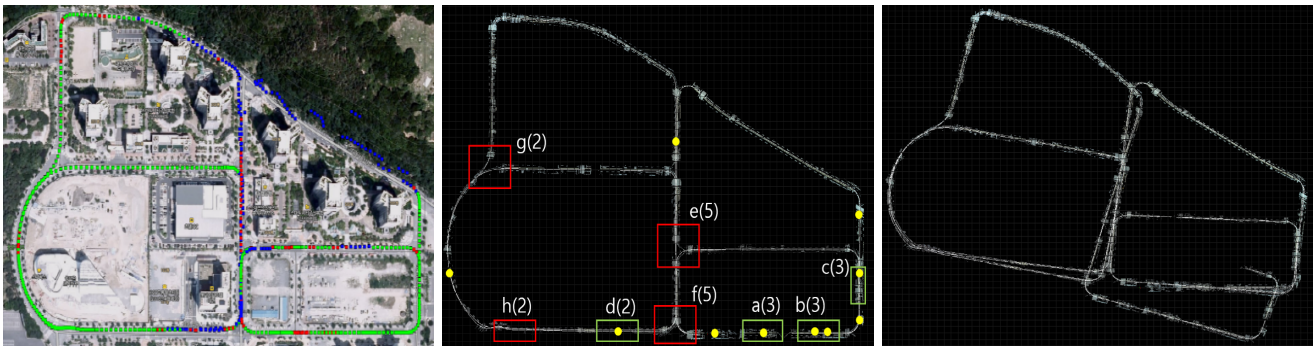
[15] Ashley Napier and Paul Newman. Generation and exploitation of local orthographic imagery for road vehicle localisation. In *Proceedings of the IEEE Intelligent Vehicle Symposium*, pages 590–596. IEEE, 2012.

[16] Xinxin Du and Kok Kiong Tan. Vision-based approach towards lane line detection and vehicle localization. *Machine Vision and Applications*, 27(2):175–191, 2016.

[17] Ananth Ranganathan, David Ilstrup, and Tao Wu. Light-weight localization for vehicles using road markings. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 921–927. IEEE, 2013.

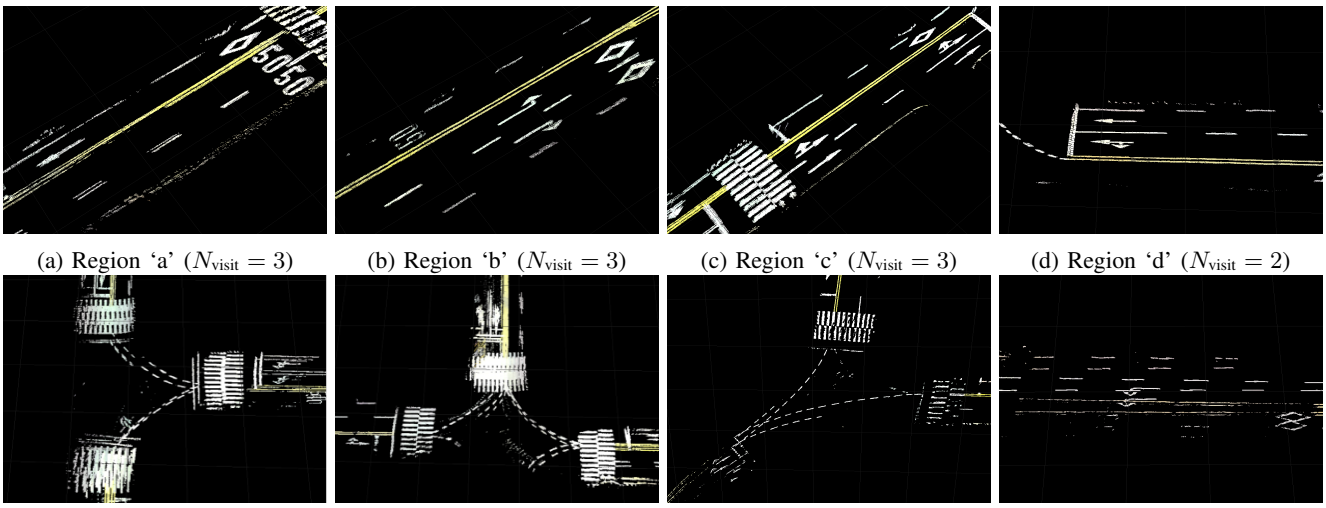
[18] Eike Rehder and Alexander Albrecht. Submap-based SLAM for road markings. In *Proceedings of the IEEE Intelligent Vehicle Symposium*, pages 1393–1398. IEEE, 2015.

[19] Marc Sons, Henning Lategahn, Christoph G Keller,



(a) Experiment environment. (b) Result of our method with loop detection. (c) Vehicle path without loop closure.

Fig. 11: Experimental results of the proposed method. (a) Aerial image of the mapped area, the depicted route was obtained using RTK-GPS. The experimental environment is $600\ \mathrm{m} \times 400\ \mathrm{m}$, and the total travel distance is $4.7\ \mathrm{km}$. (b) Resulting road map using the road marking loop detection. (c) Road map using only the odometry measurement. The yellow dots indicate the loop detection and graph correction position in (b). The green and red square regions in b indicate the area where the loop occurred and did not occur, respectively. Details of the rectangular part are shown in Fig. 12. The numbers in parentheses indicate the number of visits N_{visit} in the area.



(a) Region 'a' ($N_{\mathrm{visit}} = 3$) (b) Region 'b' ($N_{\mathrm{visit}} = 3$) (c) Region 'c' ($N_{\mathrm{visit}} = 3$) (d) Region 'd' ($N_{\mathrm{visit}} = 2$)

(e) Region 'e' ($N_{\mathrm{visit}} = 5$) (f) Region 'f' ($N_{\mathrm{visit}} = 5$) (g) Region 'g' ($N_{\mathrm{visit}} = 2$) (h) Region 'h' ($N_{\mathrm{visit}} = 3$)

Fig. 12: The road marking points overlaid with several vehicle passes in the same location. Consistent maps are generated when a loop closure is detected (a to d). If the loop is not detected or the distance from the loop detected area is increased, an error occurs in the map due to the accumulated error (e to h).

and Christoph Stiller. Multi trajectory pose adjustment for life-long mapping. In *Proceedings of the IEEE Intelligent Vehicle Symposium*, pages 901–906. IEEE, 2015.

[20] Tao Wu and Ananth Ranganathan. Vehicle localization using road markings. In *Proceedings of the IEEE Intelligent Vehicle Symposium*, pages 1185–1190. IEEE, 2013.

[21] Qingming Huang, Wen Gao, and Wenjian Cai. Thresholding technique with adaptive window selection for uneven lighting image. *Pattern recognition letters*, 26(6):801–808, 2005.

[22] Jinyong Jeong and Ayoung Kim. Adaptive inverse perspective mapping for lane map generation with SLAM. In *Proceedings of the International Conference on Ubiquitous Robots and Ambient Intelligence*, pages 38–41. IEEE, 2016.

[23] Walter Wohlkinger and Markus Vincze. Ensemble of shape functions for 3D object classification. In *Robotics and Biomimetics (ROBIO), 2011 IEEE International Conference on*, pages 2987–2992. IEEE, 2011.

[24] Aleksandr Segal, Dirk Haehnel, and Sebastian Thrun. Generalized-ICP. In *Robotics: science and systems*, volume 2, 2009.

[25] Michael Kaess, Ananth Ranganathan, and Frank Dellaert. iSAM: Incremental smoothing and mapping. volume 24, pages 1365–1378. IEEE, 2008.