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

Salience-guided Ground Factor for Robust Localization of Delivery Robots in Complex Urban Environments

Jooyong Park[1], Jungwoo Lee[2], Euncheol Choi[2] and Younggun Cho[2*]

***Abstract*— In urban environments for delivery robots, particularly in areas such as campuses and towns, many custom features defy standard road semantic categorizations. Addressing this challenge, our paper introduces a method leveraging Salient Object Detection (SOD) to extract these unique features, employing them as pivotal factors for enhanced robot loop closure and localization. Traditional geometric feature-based localization is hampered by fluctuating illumination and appearance changes. Our preference for SOD over semantic segmentation sidesteps the intricacies of classifying a myriad of non-standardized urban features. To achieve consistent ground features, the Motion Compensate IPM (MC-IPM) technique is implemented, capitalizing on motion for distortion compensation and subsequently selecting the most pertinent salient ground features through moment computations. For thorough evaluation, we validated the saliency detection and localization performances to the real urban scenarios. Project page: https://sites.google.com/view/salient-ground-feature/home.**

I. Introduction

In an era marked by the rapid integration of autonomous delivery robots into urban logistics systems, the robust and precise localization of these robots in complex urban environments has become an essential challenge. Accordingly, with advancements in Simultaneous Localization and Mapping (SLAM) [1–4] technology, last-mile delivery robots have been increasingly applied in various fields. For example, companies like Starship Technologies [5], Kiwibot [6], Nuro [7], and Neubility [8] provide delivery services based on environmental recognition through various sensors. These delivery robots have different sensor configurations (such as RGB camera, LiDAR, GPS, etc.). Using various sensors allows the robot to be more aware of the environment, but they also increase the cost of the delivery robot. Therefore, vision-based SLAM methods received significant attention because of the preference for low-cost sensor configurations for commercial-level products. However, such methods suffer from illumination, perspective, and appearance changes in the long term. Furthermore, delivery routes in urban environments such as campuses are mostly weak-GPS or GPS-denied regions.

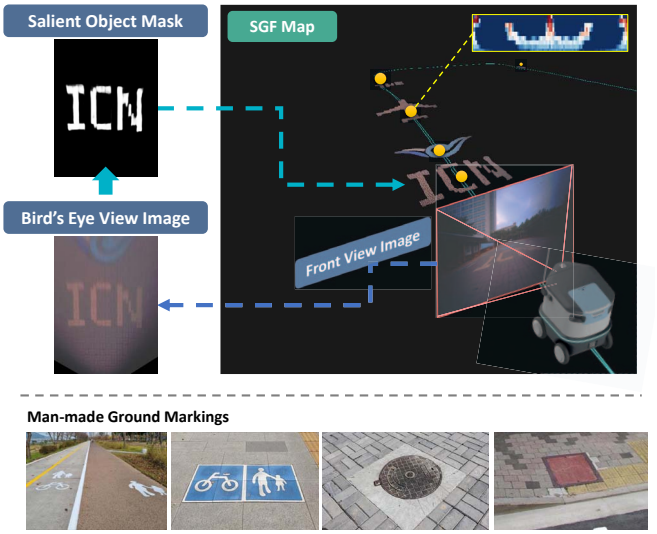

Fig. 1: Illustration of the proposed method. We extract and describe the salient feature on the ground from the Bird's Eye View (BEV). It can be used as a loop factor to perform localization.

Interestingly, most man-made ground markings exhibit high visibility, making them particularly salient to human observers. Human localization systems often utilize these features as clues for topological localization. Motivated by this characteristic, we propose a localization system that can capture custom ground markings as the Salient Ground Feature (SGF) based on SOD. Fig. 1 describes the illustration of the proposed method. Even though other delivery robots may utilize a variety of sensor configurations, a front view single camera is commonly used. The proposed method first converts the monocular image to the BEV with motion compensation. Then, we apply a SGF detector and describe the SGF in the descriptor vector. Finally, the delivery robot can correct the pose and localize on the map by utilizing SGF. The details will be explained in Section III.

The proposed localization pipeline is as shown in Fig. 2 and has the following contributions:

- We propose a new salience-guided factor for loop closure and localization of delivery robots in urban scenarios.
- We present a robust Inverse Perspective Mapping (IPM) to construct accurate ground features.
- The proposed method incorporates salient feature de-

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIP) (RS-2023-00302589), Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No.2022-0-00448) and Collabo R&D between Industry, University, and Research Institute funded by Korea Ministry of SMEs and Startups in 2023 (00224459).

[1]Jooyong Park is with the HL Klemove Corporation, Gyeonggi-do, and Dept. Electrical and Computer Engineering, Inha University, South Korea. jooyong.park@hlcompany.com

Jungwoo Lee, [2]Euncheol Choi and [2*]Younggun Cho are with the Dept. Electrical and Computer Engineering, Inha University, Incheon, South Korea. [pihsdneirf, cheol_97]@inha.edu, yg.cho@inha.ac.kr

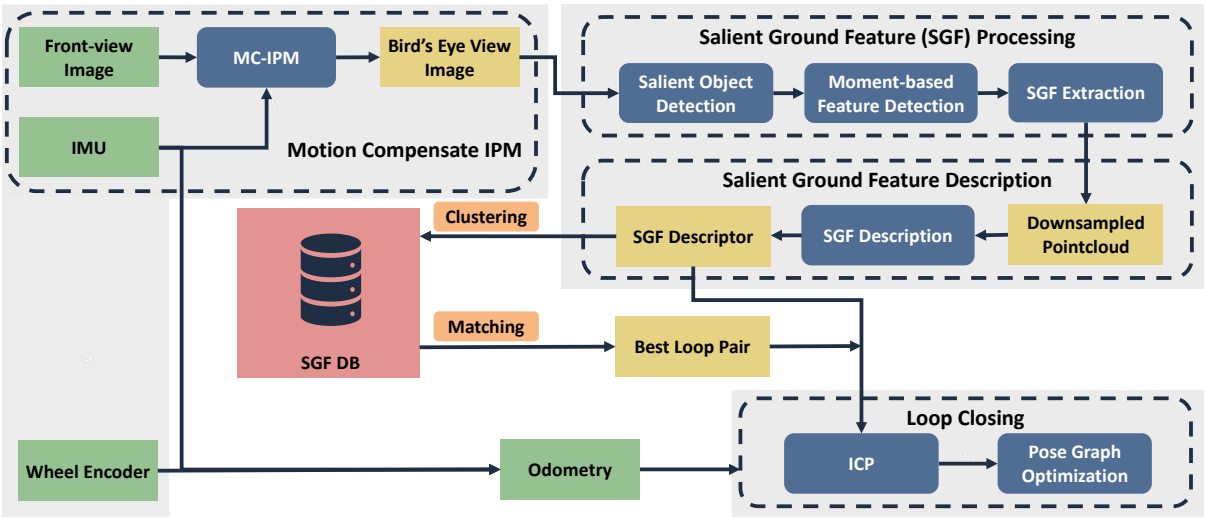

Fig. 2: The overall pipeline of the proposed system. The diagram describes utilizing SGF for localization and loop closure.

tection to capture custom man-made features into the ground factors.

- We validate our method in urban campus environments including various sequences, dynamic objects, and illumination changes.

II. Related Work

In the field of autonomous robots, precise localization is one of the most critical tasks. As mentioned above, vision-based SLAM methods are cost-competitive but suffer from illumination, perspective, and appearance changes in the long term. To address these challenges, various methodologies utilizing readily observable road markings such as lines and curb markers have been proposed [9–11]. However, these approaches still have the same limitations as relying on traditional features. Therefore, some approaches have been proposed using ground information as ground features [12, 13].

IPM The IPM algorithm, utilized for extracting robust ground features, is used in various applications [14–16]. Traditional IPM techniques do not work well on sloped or rough surfaces [17]. To address this issue, some approaches [18, 19] use estimated vanishing points. In other works, Jeong and Kim [20] leverage camera pose information to obtain precise IPM images, introducing an extended IPM algorithm. More recently, Zhu et al. [21] proposed a learning-based approach using synthetic datasets, while Bruls et al. [22] presented a method based on real-world datasets, ensuring that IPM works well in real-world environments.

Ground Feature-based SLAM Many works have leveraged ground features for robust localization in complex urban environments. Road-SLAM [12] exploits IPM image and informative six classes of road marking that can be obtained by the machine learning approach. With the advancements in CNN-based image segmentation networks [23, 24], many approaches use this network to utilize ground features. The AVP-SLAM [25], which leverages an image segmentation network [26] specifically trained for parking lots, enhances the accuracy of mapping and localization. Cheng et al. [27] achieve efficient and accurate localization via compact semantic map with CNN-based sparse semantic visual feature extracting front-end. Zhou et al. [28] proposed visual mapping and localization based on a compact instance-level semantic road marking parameterization. However, these methods, relying on semantic image segmentation, suffer from pre-defined class limitation, which is unsuitable for complex and diverse environments.

Delivery robots, operating in complex urban environments, face challenges such as unusual terrain (including paved roads, bumpy sidewalks, and obstacles like curbs) and dynamic occlusions from pedestrians and vehicles. They are also constrained by limitations in available sensors, including GNSS-denied scenarios and cheaper sensors. To address these problems, we introduce a Motion Compensate IPM (MC-IPM) method in Section III-A. We also demonstrate the extraction and utilization of undefined semantic ground features using a transformer-based SOD model [29] in Section III-B, Section III-C and Section III-D.

III. Method

A. Motion Compensate Inverse Perspective Mapping

We use the IPM method to facilitate obtaining ground features from a monocular camera. Since IPM assumes that the ground in front of the camera is flat, distortion compensation through local ground estimation is necessary in environments with uneven ground. To solve this problem, we introduce MC-IPM, a post-processing process that measures the relative motion changes of the robot and compensates for them.

First, IPM compensation uses the difference in roll and pitch between poses. In the previous task, we improved the Adaptive IPM[20] into an Extended and Adaptive IPM method that can compensate for both roll and pitch. The pixel coordinates (u, v) are converted into metric space (c, r)

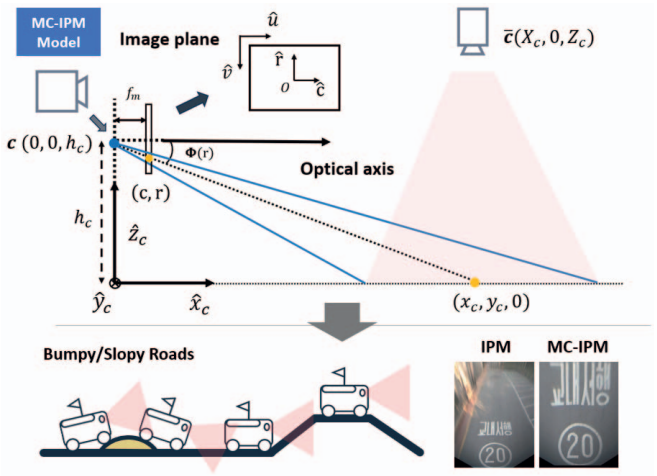

Fig. 3: Illustration of an MC-IPM model. A corrected BEV image is generated from a compensated 3D point projected through the MC-IPM. The lower right example is a compensation result of a situation crossing a speed bump.

and projected to the compensated 3D point $(x_c, y_c, 0)$ of the camera center coordinates as follows,

$$\begin{bmatrix} \mathrm{c}^{\psi} \\ \mathrm{r}^{\psi} \end{bmatrix} = \begin{bmatrix} \cos(-\boldsymbol{\psi}) & -\sin(-\boldsymbol{\psi}) \\ \sin(-\boldsymbol{\psi}) & \cos(-\boldsymbol{\psi}) \end{bmatrix} \begin{bmatrix} \mathrm{c} \\ \mathrm{r} \end{bmatrix}, \tag{1}$$

$$\begin{aligned} x_c &= h_c \cot(\alpha + \Phi(\mathrm{r}^{\psi}) + \boldsymbol{\theta}), \\ y_c &= -x_c \left(\frac{\mathrm{c}^{\psi}}{f_m}\right) \frac{\cos(\Phi(\mathrm{r}^{\psi}) + \boldsymbol{\theta})}{\cos(\alpha + \Phi(\mathrm{r}^{\psi}) + \boldsymbol{\theta})}. \end{aligned} \tag{2}$$

Eq. (1) is the roll compensation process before projecting from the image plane to 3D space, and $\boldsymbol{\psi}$ is the magnitude of roll change. In Eq. (2), h_c is the height of the front view camera (c), f_m is the focal length in metric units, α is the angle between the optical axis and the ground, $\boldsymbol{\theta}$ is the magnitude of pitch and $\Phi(\mathrm{r}^{\psi}) = \arctan(\mathrm{r}^{\psi}/f_m)$. The compensation formula for motion changes is detailed in [20]. After that, we can obtain the compensated BEV image by 2D projecting the 3D point to the virtual camera ($\bar{c}$) through the process as,

$$\begin{bmatrix} x_{\bar{c}} \\ y_{\bar{c}} \\ 1 \end{bmatrix} = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 0 & X_c \\ 0 & 0 & Z_c \end{bmatrix} \begin{bmatrix} x_c \\ y_c \\ 1 \end{bmatrix}, \tag{3}$$

$$\begin{bmatrix} u_{\bar{c}} \\ v_{\bar{c}} \\ 1 \end{bmatrix} = \mathbf{K}_{\bar{\mathbf{c}}} \begin{bmatrix} x_{\bar{c}} \\ y_{\bar{c}} \\ 1 \end{bmatrix}, \tag{4}$$

where $(X_c, 0, Z_c)$ and $\mathbf{K}_{\bar{\mathbf{c}}}$ are position (camera center coordinates) and intrinsic parameters of virtual camera, respectively. The IPM projection and the relationship between the front view camera and the virtual camera are shown in Fig. 3.

Compensation can be applied primarily in two cases: 1) temporary unevenness (e.g., speed bumps, cracks) and 2) uphill or downhill, as shown in Fig. 3. First, store N-poses in a queue and calculate the average. Since pitch motion change mainly occurs in driving scenarios, the average calculation does not include poses greater than the pitch threshold. Subsequently, the relative difference between the average pose in the area and the robot's pose that needs to be compensated is passed as IPM parameters $\boldsymbol{\psi}$ and $\boldsymbol{\theta}$. In this paper, we set the threshold for $N = 50$ and relative pitch differences, $d_{\boldsymbol{\theta}} = 0.025$ radians. The qualitative result of MC-IPM compensation as shown in Fig. 3.

B. Salient Ground Feature Detection

In the general computer vision field, SOD targets the segmentation of foreground objects in normal scenes. Motivated by this approach, we utilize general SOD networks to capture representative ground features. For SGF extraction, we adopt a SelfReformer[29] network, a transformer-based model. However, the general SOD networks show difficulties on ground features. Therefore we re-design the training sets with a partial mixture of well-known road markings[30]. To this end, we re-formulate SelfReformer into SelfReformer* that can capture the characteristics of SGF. The input of the SelfReformer* is the BEV image (3-channel) calibrated by MC-IPM and the SGF mask (1-channel) is obtained as the output.

After obtaining the binary SGF mask, we select the optimal SGF based on the image moment that can capture the nature of the binary image. Rather than performing extraction for all query images, we perform an efficient way that extracts the most significant features. This also avoids the ambiguity of defining a feature as an SGF that has not fully appeared.

The moment-based SGF detector uses a sliding window approach to find the optimal SGF in 4-steps. 1) First, it determines the valid feature range, which is the range from when a feature appears to when it disappears. 2) Then, it calculates the Hu moment[31] for all features in the valid feature range. Each element of the Hu moment consists of η_{ij}, which has robust properties in translation and scale, as follows,

$$\mu_{pq} = \sum_{u \in U} \sum_{v \in V} (u - \bar{u})^p (v - \bar{v})^q I(u, v), \tag{5}$$

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^{\left(1 + \frac{p+q}{2}\right)}}, \tag{6}$$

where $I(\cdot, \cdot)$ represents intensity at pixel coordinates (u, v), and p,q is the order of moment. In addition, μ_{pq} is the central moment, which is invariant to translation, and η_{pq} is obtained by normalizing the central moment through the zero-order central moment. The Hu moment $\mathcal{H}$ is calculated to be invariant to translation, scale and rotation as follows.

$$\mathcal{H}_i = \{h_i^1, h_i^2, \ldots, h_i^7\}, \quad i \in \mathcal{V} \tag{7}$$

where $\mathcal{V}$ is all sets in a valid feature range, and the seven components of Hu moment follow [31]. 3) Next, we calculate the distance between the current feature and the previous feature by shape matching using the Hu moments as follows,

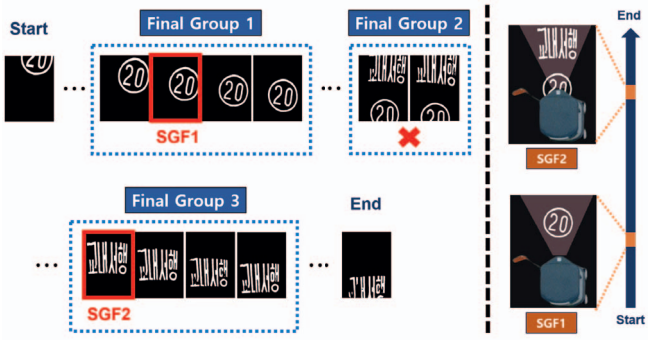

Fig. 4: An example of a SGF detected in a valid feature range. In the above case, two SGFs were detected in the three final groups, and not selected in final group 2 (boundary condition).

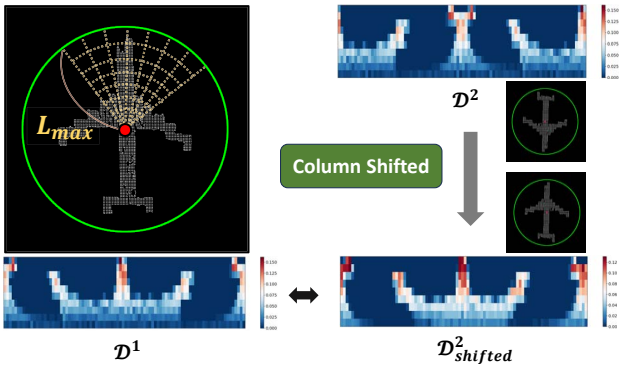

Fig. 5: Illustration of SGF description process and rotational invariance through column-shifted matching. The red dot is the center of the SGF points, and the green circle is the radius L_{max} boundary.

$$d_{hu}(\mathcal{H}_{i-1}, \mathcal{H}_i) = \sum_{j=1}^{7} \left| h_{i-1}^j - h_i^j \right|, \quad i \in \mathcal{V}. \tag{8}$$

Finally, 4) we select the the group's most salient feature whose distance is closer than the threshold (=0.005). If the feature region is outside the four boundaries of the image, we do not select it. According to the three invariant properties of Hu moment, it can catch consecutive features that are highly correlated. An example is shown in Fig. 4.

C. Salient Ground Feature Description

Once the optimal SGF is selected, those regions are extracted into 3D space as follows,

$$\begin{bmatrix} x_{sal} \\ y_{sal} \\ 1 \end{bmatrix} = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 0 & X_c \\ 0 & 0 & Z_c \end{bmatrix}^{-1} \cdot \mathbf{K_{\bar{c}}}^{-1} \begin{bmatrix} u_{sal} \\ v_{sal} \\ 1 \end{bmatrix}, \tag{9}$$

Afterward, downsampling is performed for SGF points. To use the SGF as a loop factor, we create a descriptor for the SGF points. We adopted the method proposed in scan context[32], which is a rotation robust description. Splits bins in azimuth and radial directions within the L_{max} around the centroid of the SGF points. In this paper, we used $L_{max} = 2$, considering the average size of the ground feature. To perform a reverse loop closing, we set $N_s = 90$, a central angle ($2\pi/N_s$), and $N_r = 10$, a radial gap ($L_{\max}/N_r$). The description process is shown in Fig. 5.

And then, SGF descriptors are grouped into various features through online clustering. When the query SGF descriptor $\mathcal{D}^q$ is determined, it is assigned to the closest group via cosine distance to the previous SGF group's descriptor $\mathcal{D}^{sg}$ using column shifting as follows,

$$d(\mathcal{D}^q, \mathcal{D}^{sg}) = \frac{1}{N_c} \sum_{j=1}^{N_c} \left(1 - \frac{c_j^q \cdot c_j^{sg}}{\|c_j^q\| \|c_j^{sg}\|} \right), \tag{10}$$

where c_j^q and c_j^{sg} are the j-th columns of $\mathcal{D}^q$ and $\mathcal{D}^{sg}$, respectively, and N_c is the number of columns. $\mathcal{D}^{sg}$ is then replaced by the mean of $\mathcal{D}$ in the group and is then used for comparison with the new $\mathcal{D}^q$. If the distance with all $\mathcal{D}^{sg}$ is greater than 0.7, it is assigned as a new SGF group.

D. Loop Closing with SGF Factor

When the robot re-visits an area with the same SGF, loop closing can be performed via SGF association. Once SGF is selected and the SGF group is determined, perform the Iterative Closest Algorithm (ICP) matching with the closest descriptor within the group. In the reverse loop situation, we can provide the ICP initials using the best column key obtained when computing the nearest descriptor. SGF groups with symmetric properties (e.g., a circular manhole) are excluded from loop SGF because they cannot provide an accurate pose as a result of ICP. The global pose graph optimization with SGF loop factor is as follows,

$$\begin{aligned} \mathcal{X}^* = \arg\min_{\mathcal{X}} & \sum_t \| f(\mathbf{x}_t, \mathbf{x}_{t+1}) - \mathbf{z}_{t,t+1} \|^2_{\Sigma_t} \\ & + \sum_{i,j \in \mathcal{L}} \| f(\mathcal{S}_i, \mathcal{S}_j) - \mathbf{z}_{i,j} \|^2_{\Sigma_{ij}}, \end{aligned} \tag{11}$$

where $\mathbf{x_t} = [\mathbf{r}_t, \mathbf{t}_t]^T$ is the camera pose at t, $\mathcal{X} = [\mathbf{r}_0, \mathbf{t}_0, \ldots, \mathbf{r}_t, \mathbf{t}_t]^T$ are all sequences, and $\mathbf{z}_{t,t+1}$ are relative pose between camera frame at t and $t+1$. $\mathcal{S}_i$ and $\mathcal{S}_j$ are SGF pairs observed in different sequences and $\mathcal{L}$ is a set of all SGF pairs. The function $f(\cdot, \cdot)$ estimates the relative pose.

IV. Experimental Results

A. Experimental Setups

In the experiments, we utilized two mobile robots: A real delivery Robot from Neubility [8] and a 4-wheeled mobile robot. Fig. 6 represents the robots and sensors for evaluation. Both robots are equipped with wheel encoder, RGB camera, IMU and LiDAR. We constructed the ground-truth trajectories using well-known LiDAR SLAM methods [33, 34].

As described in Fig. 7, test sequences are composed of various driving scenarios on a campus because a campus environment includes many dynamic objects such as people

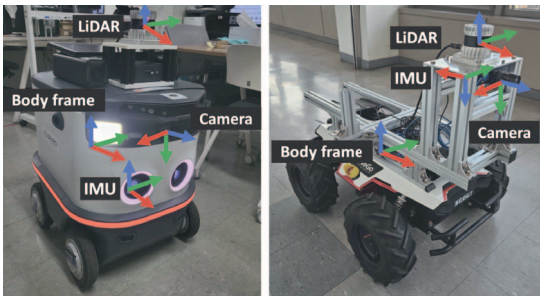

Fig. 6: Robots were used in the experiments. (Left) A delivery Robot from Neubility [8]. (Right) A 4-wheeled Mobile Robot.

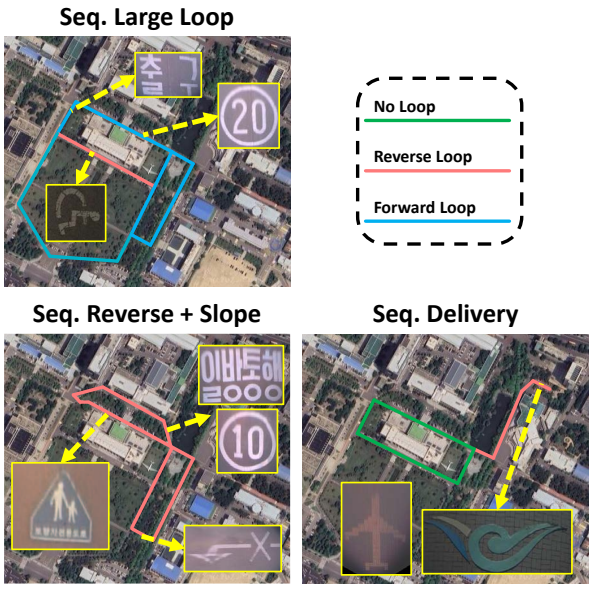

Fig. 7: Test environment containing several representative SGFs. The legend indicates the type of the sequence.

and vehicles. Also, to evaluate the long-term operation, we evaluated the proposed method for the night sequences. *Seq. Delivery* assumes a delivery route. In this scenario, the robot performs a delivery mission and then returns to its home. *Seq. Reverse+Slope* is a scenario where all loops are reverse loops only. *Seq. Large loop* covers about 2.7 kilometers, including both forward and reverse loops. For the concrete evaluation, we tested the proposed method in terms of saliency detection performance, feature detector score, and localization accuracy.

B. Salient Object Detection

To test the performance of SelfReformer*, we compared the method to the well-known SOD networks [35, 36] and the foundation model of semantic segmentation (GroundedSAM [37]). In Fig. 8, SelfReformer* shows the best performance on saliency mask prediction. This result reports the validity of the training sets for man-made ground markings. Also, we found an interesting point on the result of GroundedSAM. Although GroundedSAM is known for great generality on visual perception, only some parts of the custom man-made

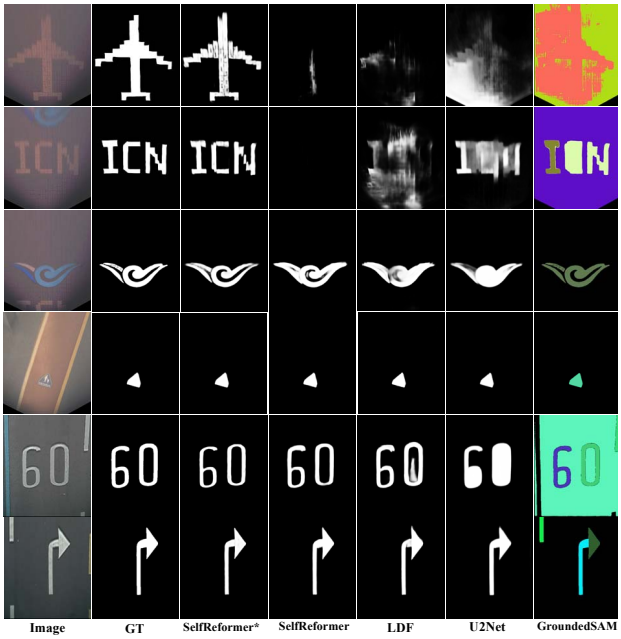

Fig. 8: SOD results. Quantitative comparisons between SelfReformer*(Ours) and other SOD networks.

markings can be represented by pre-defined semantics. On the other hand, SelfReformer* recognizes salient objects without pre-defined class information, so it can effectively recognize man-made features as well as standard markings in BEV images. For quantitative evaluation, we evaluated performance using well-known metrics MaxF (F_m), MAE (M), MeanF (F_A), S-measure (S_m) on the SOD task [35, 38–40]. As shown in Table 1, SelfReformer* outperforms all other models.

Methods	$F_m \uparrow$	M $\downarrow$	$F_A \uparrow$	$S_m \uparrow$
SelfReformer*	**.953**	**.032**	**.908**	**.920**
SelfReformer[29]	.463	.259	.310	.575
LDF [35]	.545	.179	.422	.664
U2Net [36]	.580	.140	.565	.679

TABLE 1: Quantitative comparisons of SOD performance.

C. Localization Evaluation

We present experimental results to validate the performance of the proposed auxiliary factor in the localization system. Since our algorithm is a vision-based localization method using both standard and man-made features in a complex urban environment, it is hard to compare directly with existing algorithms. Thus, we compared the results with Odometry and ORB-SLAM3. In ORB-SLAM3, we used a Monocular-IMU system to avoid scale ambiguity and manually perform scale correction on the outputs. We used evo [41] to estimate Absolute Trajectory Error (ATE).

Fig. 9 shows how the proposed SGF factor performs in campus scenario. The quantitative comparison can be found in Table 2. For all sequences, We can see that the proposed method achieves a good performance compared

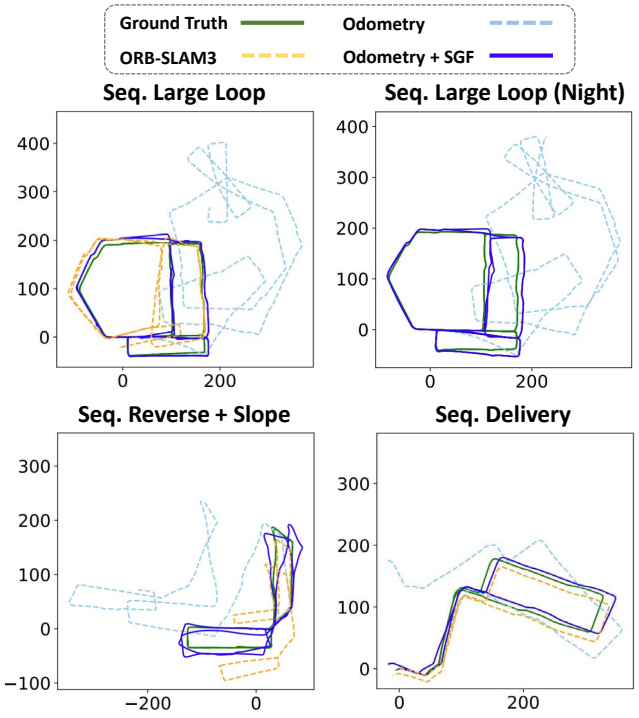

Fig. 9: Results of adding SGF factor with each sequence in campus scenario. Best viewed in color.

Methods	Sequence.			
	Large Loop (Day)	Reverse + Slope	Delivery	Large Loop (Night)
Odometry	292.966	140.532	71.147	295.106
ORB-SLAM3	28.422	48.546	19.828	-
Odometry + SGF	**9.482**	**14.384**	**12.365**	**11.540**

TABLE 2: Absolute Trajectory Error (ATE). (m)

to the traditional vision-based method, ORB-SLAM3. For *Seq. Delivery*, both ORB-SLAM3 and the proposed method performed well, but we can see that our proposed method achieved lower errors with reverse loop detection. The *Seq. Reverse + Slope* showed relatively low accuracy among all sequences because the SGFs were extracted from uneven ground. In the *Seq. Large Loop*, ORB-SLAM3 lost tracking while driving. Comparing the trajectory before the tracking loss, we can see that the error was not corrected at the reverse loop point.

Sequences	Detection	Find loop pairs (Rev.)	Loop closed (Rev.)
Seq. Large Loop	79/94	33/34 (3/3)	31 (1)
Seq. Reverse+Slope	37/46	11/16 (11/16)	7 (7)
Seq. Delivery	19/23	1/1 (1/1)	1 (1)

TABLE 3: Quantitative results for SGF detector, loop finding, and loop matching in each sequence.

In Table 3, each column quantitatively shows results for SGF detector, loop finding, and loop matching by all sequences. The first column is the number of SGFs found relative to the total number of SGFs that should be detected in the path, the second column is the number of pairs grouped in the same SGF group among all loop pairs and the last column is the number of SGF loop matching was successful among the pairs obtained in the second column. Parentheses indicate the reverse loop cases. The proposed SGF detector showed a detection success rate of nearly 80% or higher, and the SGF descriptor succeeded in loop matching in more than half, even in reverse loop cases.

We also tested the SGF for the night-time sequences. Each robot is equipped with LED lights, and we applied Contrast Limited Adaptive Histogram Equalization (CLAHE) to enhance the contrast of dark images. As in Fig. 9, the proposed method resulted in equivalent performance compared to the daytime. Note that ORB-SLAM 3 failed to optimize for both raw and enhanced night sequences.

D. Comparison of MC-IPM and IPM

To evaluate the performance of MC-IPM, we compare with conventional IPM. In uneven ground conditions, the IPM cannot provide consistent information to the SGF detector, resulting in poor performance. For a more explicit comparison, we selected *Seq. Large loop* and *Seq. Reverse+Slope* with varying ground conditions and a large number of SGFs. The results are shown in Table 4 and Fig. 10.

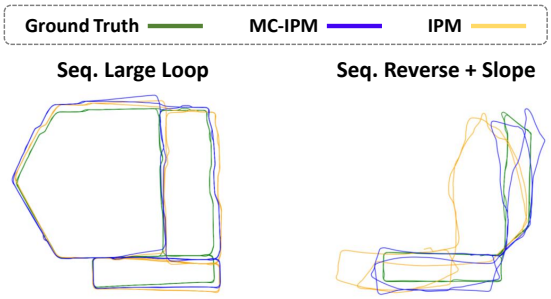

Fig. 10: Qualitative comparison results of MC-IPM and IPM.

Sequences	MC-IPM	IPM
Seq. Large Loop	**9.482** (m)	12.567 (m)
Seq. Reverse+Slope	**14.384** (m)	67.625 (m)

TABLE 4: Comparison ATE of MC-IPM and IPM.

V. Conclusion

In this paper, we propose a novel localization system that integrates standard and man-made features, one of the characteristics of complex urban structures, and utilizes them as SGF. To obtain more consistent SGF in uneven ground conditions, we applied MC-IPM, which utilizes the robot's motion. The SGF factor was applied to a delivery robot, performing well in the campus environment and solving the reverse loop case. It also validated strong performance against changes in lighting and appearance throughout the day and night. Our future work is to build a methodology that integrates SGF into vision-based SLAM algorithms to perform better.

REFERENCES

[1] C. Campos, R. Elvira, J. J. G. Rodríguez, J. 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, 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.

[3] J. Zhang and S. Singh, "Loam: Lidar odometry and mapping in real-time." in *Robotics: Science and systems*, vol. 2, no. 9. Berkeley, CA, 2014, pp. 1–9.

[4] T. Shan, B. Englot, D. Meyers, W. Wang, C. Ratti, and D. Rus, "Lio-sam: Tightly-coupled lidar inertial odometry via smoothing and mapping," in *2020 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2020, pp. 5135–5142.

[5] StarshipTechnologies, "Starship technologies: Automated robot delivery." https://www.starship.xyz/.

[6] Kiwibot, "Kiwibot autonomous delivery robots, revolutionizing the future." https://www.kiwibot.com/.

[7] Nuro, "Nuro - on a mission to better everyday life through robotics." https://www.kiwibot.com/.

[8] Neubility, "Neubility: Camera based autonomous robot-as-a-service," https://www.neubility.co.kr/.

[9] M. Schreiber, C. Knöppel, and U. Franke, "Laneloc: Lane marking based localization using highly accurate maps," in *2013 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2013, pp. 449–454.

[10] A. Ranganathan, D. Ilstrup, and T. Wu, "Light-weight localization for vehicles using road markings," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2013, pp. 921–927.

[11] Y. Lu, J. Huang, Y.-T. Chen, and B. Heisele, "Monocular localization in urban environments using road markings," in *2017 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2017, pp. 468–474.

[12] J. Jeong, Y. Cho, and A. Kim, "Road-slam: Road marking based slam with lane-level accuracy," in *2017 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2017, pp. 1736–1473.

[13] 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)*. IEEE, 2021, pp. 11 248–11 254.

[14] S. Tuohy, D. O'Cualain, E. Jones, and M. Glavin, "Distance determination for an automobile environment using inverse perspective mapping in opencv," in *IET Irish Signals and Systems Conference (ISSC 2010)*, 2010, pp. 100–105.

[15] C. Guo, J.-i. Meguro, Y. Kojima, and T. Naito, "Automatic lane-level map generation for advanced driver assistance systems using low-cost sensors," in *2014 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2014, pp. 3975–3982.

[16] C.-C. Lin and M.-S. Wang, "A vision based top-view transformation model for a vehicle parking assistant," vol. 12, no. 4. Molecular Diversity Preservation International (MDPI), 2012, pp. 4431–4446.

[17] Z. Ying and G. Li, "Robust lane marking detection using boundary-based inverse perspective mapping," in *2016 ieee international conference on acoustics, speech and signal processing (icassp)*. IEEE, 2016, pp. 1921–1925.

[18] D. Zhang, B. Fang, W. Yang, X. Luo, and Y. Tang, "Robust inverse perspective mapping based on vanishing point," in *Proceedings 2014 IEEE International Conference on Security, Pattern Analysis, and Cybernetics (SPAC)*. IEEE, 2014, pp. 458–463.

[19] M. Nieto, L. Salgado, F. Jaureguizar, and J. Cabrera, "Stabilization of inverse perspective mapping images based on robust vanishing point estimation," in *2007 IEEE Intelligent Vehicles Symposium*. IEEE, 2007, pp. 315–320.

[20] 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)*. IEEE, 2016, pp. 38–41.

[21] X. Zhu, Z. Yin, J. Shi, H. Li, and D. Lin, "Generative adversarial frontal view to bird view synthesis," in *2018 International conference on 3D Vision (3DV)*. IEEE, 2018, pp. 454–463.

[22] T. Bruls, H. Porav, L. Kunze, and P. Newman, "The right (angled) perspective: Improving the understanding of road scenes using boosted inverse perspective mapping," in *2019 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2019, pp. 302–309.

[23] S. Lee, J. Kim, J. Shin Yoon, S. Shin, O. Bailo, N. Kim, T.-H. Lee, H. Seok Hong, S.-H. Han, and I. So Kweon, "Vpgnet: Vanishing point guided network for lane and road marking detection and recognition," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 1947–1955.

[24] T. Ahmad, D. Ilstrup, E. Emami, and G. Bebis, "Symbolic road marking recognition using convolutional neural networks," in *2017 IEEE intelligent vehicles symposium (IV)*. IEEE, 2017, pp. 1428–1433.

[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)*. IEEE, 2020, pp. 5939–5945.

[26] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*. Springer, 2015, pp. 234–241.

[27] 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.

[28] 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. 10 802–10 809, 2022.

[29] Y. K. Yun and W. Lin, "Selfreformer: Self-refined network with transformer for salient object detection," *arXiv preprint arXiv:2205.11283*, 2022.

[30] W. Jang, J. Hyun, J. An, M. Cho, and E. Kim, "A lane-level road marking map using a monocular camera," *IEEE/CAA Journal of Automatica Sinica*, vol. 9, no. 1, pp. 187–204, 2021.

[31] M.-K. Hu, "Visual pattern recognition by moment invariants," *IRE Transactions on Information Theory*, vol. 8, no. 2, pp. 179–187, 1962.

[32] G. Kim and A. Kim, "Scan context: Egocentric spatial descriptor for place recognition within 3d point cloud map," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 4802–4809.

[33] W. Xu and F. Zhang, "Fast-lio: A fast, robust lidar-inertial odometry package by tightly-coupled iterated kalman filter," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3317–3324, 2021.

[34] G. Kim and A. Kim, "Lt-mapper: A modular framework for lidar-based lifelong mapping," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 7995–8002.

[35] J. Wei, S. Wang, Z. Wu, C. Su, Q. Huang, and Q. Tian, "Label decoupling framework for salient object detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 13 025–13 034.

[36] X. Qin, Z. Zhang, C. Huang, M. Dehghan, O. R. Zaiane, and M. Jagersand, "U2-net: Going deeper with nested u-structure for salient object detection," *Pattern recognition*, vol. 106, p. 107404, 2020.

[37] "Grounded-sam: Marrying grounding dino with segment anything stable diffusion recognize anything - automatically detect , segment and generate anything," 2023.

[38] S. Chen, X. Tan, B. Wang, and X. Hu, "Reverse attention for salient object detection," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 234–250.

[39] Y. Pang, X. Zhao, L. Zhang, and H. Lu, "Multi-scale interactive network for salient object detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 9413–9422.

[40] J. Wei, S. Wang, and Q. Huang, "F^3net: fusion, feedback and focus for salient object detection," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, no. 07, 2020, pp. 12 321–12 328.

[41] M. Grupp, "evo: Python package for the evaluation of odometry and slam." https://github.com/MichaelGrupp/evo, 2017.