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

Visual Localization in Repetitive and Symmetric Indoor Parking Lots using 3D Key Text Graph

Joohyung Kim[1], Gunhee Koo[1], Heewon Park[2] and Nakju Doh[3]

***Abstract*— Indoor parking lots are the GPS-denied spaces to which vision-based localization approaches have usually been applied to solve localization problems. However, due to the *repetitiveness* and *symmetry* of the spaces, visual localization methods commonly confront difficulties in estimating precise 3D poses. In this study, we propose four novel modules that improve localization precision by imposing the existing methods with the spatial discerning ability. The first module constructs a key text graph that represents the topology of key texts in the space and becomes the basis for discerning repetitiveness and symmetry. Next, the orientation filtering module estimates the unknown 3D orientation of the query image and resolves spatial symmetric ambiguity. The similarity scoring module sorts out the top-scored database images, discerning the spatial repetitiveness based on detected key text bounding boxes. Our pose verification module evaluates the pose confidence of top-scored candidates and determines the most reliable pose. Our method has been validated in two real indoor parking lots, achieving new state-of-the-art performance levels.**

I. Introduction

Indoor parking lots are representative GPS-denied spaces where visual localization methods are often applied to solve 3D global localization problems for vehicles. However, their localization performances are limited because of spatial self-similarity, the major characteristic of indoor parking lots. For discerning the spatial difference among similar-looking images, pre-built maps with semantic information are adopted for precise visual localization [1], [2].

Among the pre-built maps, object-level high-definition maps (HD-map) are employed particularly for vehicle localization and autonomous parking, and they necessitate the use of a sliding window or surround-view cameras to capture sufficient object-level semantic features [1], [2], [3]. Additionally, these methods utilize both the query images and data from IMU or wheel encoders as input data to localize a vehicle while navigating. However, requiring an agent to navigate for even a short duration to achieve self-localization can be impractical and unsuitable in relocalization scenarios.

This work was supported by SAIT, Samsung Electronics Co., Ltd., National Research Foundation of Korea (No. NRF-2022R1F1A1073972), and TeeLabs Co., Ltd.

[1]Joohyung Kim and Gunhee Koo are equally contributed to this paper as first authors. They are with Korea University, 145 Anam-ro, Seoul, South Korea. kjh069@gmail.com; gunhee.koo@gmail.com

[2]Heewon Park is with Samsung Advanced Institute of Technology, 130 Samsung-ro, Yeongtong-gu, Suwon-si, Gyeonggi-do, South Korea. heewon7.park@samsung.com

[3]Nakju Doh is the corresponding author. He is a professor at the Institute of Convergence Science, Korea University, and CEO at TeeLabs, 131, Hwangeo-ro, Gyeyang-gu, Incheon, South Korea. nathan@teevr.com

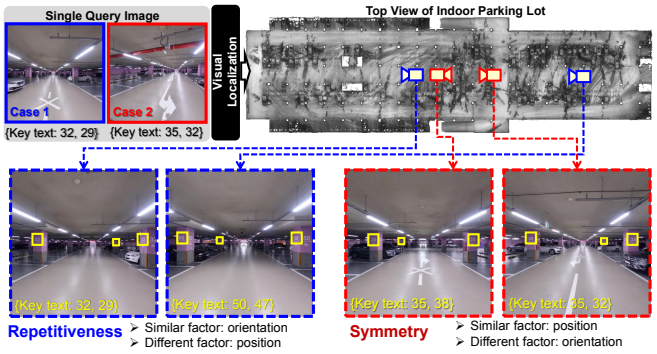


Fig. 1: **Two major spatial properties of indoor parking lots: *repetitiveness* and *symmetry*.** The properties cause images captured from varying positions and orientations to appear similar. Consequently, visual localization struggles to distinguish such similar-looking database images and correctly match their visual features with those of a query image.

To cover broad localization scenarios, including the relocalization case, the coarse-to-fine visual localization framework using only a single query image can be an alternative, [4], [5], [6]. The framework begins with a pre-built visual feature map. When a single query image is given, it retrieves the most similar database images by computing cosine distances of global descriptors, such as NetVLAD [7], and SFRS [8] during the image retrieval step. Next, 2D-3D correspondences are established based on the local feature matches between the retrieved database images and query image, and perspective-n-point (PnP) methods with a random sample consensus (RANSAC) loop are followed. Then, one or more pose candidates are yielded in the pose estimation step. Finally, the most reliable pose among the pose candidates is selected in the pose verification step. This coarse-to-fine framework shows the strengths of high adaptability and generalization for various spaces [4], [9].

Despite these advantages, the performances of coarse-to-fine visual localization methods significantly decrease in spaces with high self-similarity, such as indoor parking lots. To address this spatial self-similarity property, we separate it into two properties: *repetitiveness* and *symmetry*, as in Fig.1. When two or more similar-looking images are captured from similar orientations but considerably different positions, we categorize this type of spatial similarity as *repetitiveness*. Conversely, if such images are captured in close positions but in significantly different orientations, we term this spatial

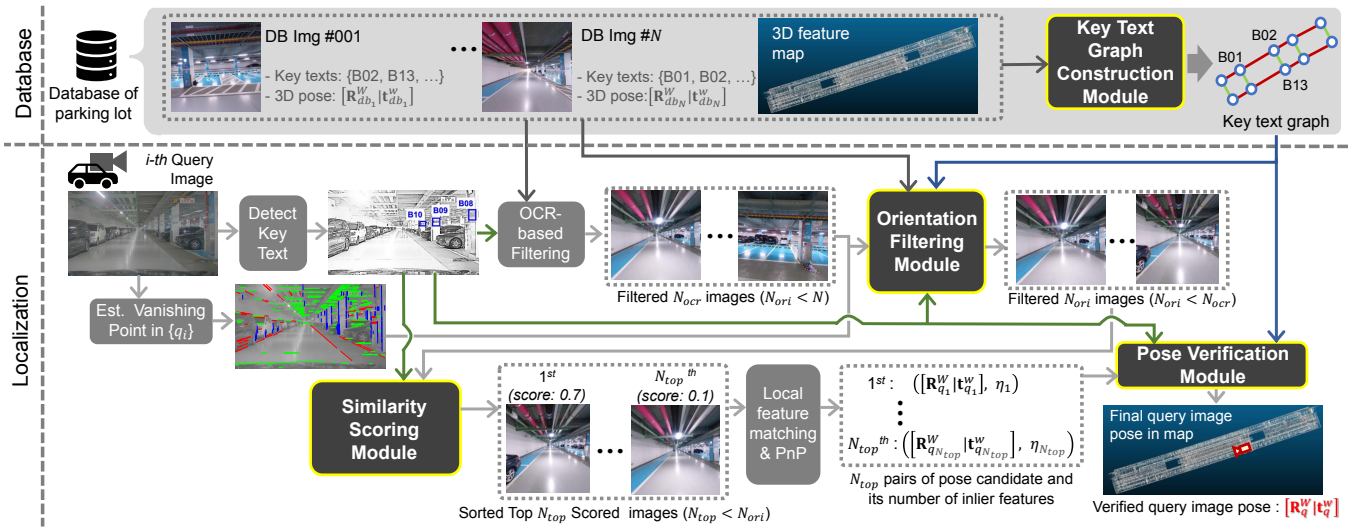


Fig. 2: **Overview of the proposed visual localization.** On top of coarse-to-fine frameworks, our method integrates four unique modules: (i) 3D key text graph construction module, (ii) orientation filtering module, (iii) similarity scoring module, and (iv) pose verification module.

similarity as *symmetry*. In these repetitive and symmetric indoor parking lots, humans primarily rely on surrounding textual information to deduce their locations. Therefore, the integration of text recognition techniques into localization frameworks is a feasible approach [10].

In this study, we propose four novel modules that actively utilize text information and seamlessly integrate the modules into a coarse-to-fine framework. Our contributions are summarized as follows:

- We first develop a key text graph that describes the topology of key texts.
- We propose an orientation filtering module and a similarity scoring module to address the challenges of repetitiveness and symmetry of indoor parking lots.
- We propose a novel pose verification module that outperforms the existing pose verification modules in precision and time.
- We validate our visual localization method, comparing it to existing methods in two real indoor parking lots.

II. Related Work

Recent studies on language models [11] have allowed the effective application of textual information to image retrieval tasks [10], [12], [13], [14]. While some approaches [12], [14] retrieve images related to a query phrase based on the embedded feature space, others [10], [13] directly investigate the presence of query texts within the images. Especially, [10] demonstrates the feasibility of employing OCR techniques for visual place recognition in urban spaces by harnessing scene texts such as shop signage, road signs, and street names. However, Levenshtein distance and Intersection-over-Union (IoU) of bounding boxes, the criteria employed in [10], are less suitable for discerning repetitive and symmetric database images of indoor parking lots. In contrast, our method is designed to retrieve correct database images from a set of repetitive and symmetric ones.

Based on the retrieved images, typical coarse-to-fine frameworks [4], [5], [6] estimate pose candidates and determine the most reliable pose in the pose verification step. Conventionally, the query pose is determined by the pose that has the largest number of inlier 2D-3D correspondences found in a PnP-RANSAC loop [6], [15], [16]. However, this approach has limitations in discerning repetitive environments and can be vulnerable to moved objects [4], [5], [17]. Other approaches [4], [5] involve comparing the query image with its synthetic counterpart generated from estimated viewpoints, but they demand a high computational cost. To address these limitations, our pose verification method utilizes a pre-constructed 3D key text graph to quantify the confidence of each pose candidate and finally determines the reliable pose of the query image.

III. Method

As a coarse-to-fine visual localization framework, the proposed method comprises four unique modules that finally estimate the optimal 3D pose of the query image using a spatial database, as in Fig.2. As an additional process for database generation, we construct a 3D key text graph using database images and a 3D feature map. In the localization process, we first retrieve the database images that contain key texts in common with those recognized in the query image. Next, we estimate the 3D orientation of the query image and subsequently filter out the database images whose 3D orientations are not roughly aligned with that of the query. Then, we directly score the similarity between the query and the remaining database images and choose the limited number of top-scored images. Finally, the 3D poses of the query image are estimated based on each top-scored database image, and the most reliable pose is determined. Each of the following subsections describes our four novel modules.

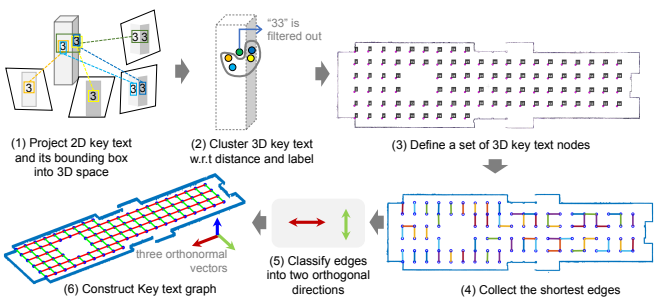


Fig. 3: **Key text graph construction module.** A 3D key text graph can be constructed automatically utilizing key text bounding boxes from the database images and their corresponding 3D poses in the pre-built map.

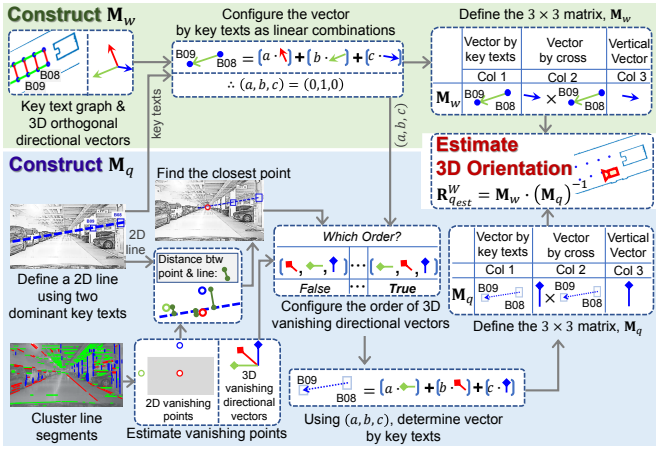


Fig. 4: **Estimation of the 3D orientation of a query image.** The core process of the orientation filtering module is to estimate the 3D orientation of the query image. Two individual directional matrices are constructed in the map, $\{w\}$, and the query camera, $\{q\}$, coordinate systems, respectively. From these two matrices, we can estimate the query image orientation $\mathbf{R}^w_{q_{est}}$.

A. Automatic 3D Key Text Graph Construction

The key text graph represents the topology of key text nodes, describing the connection of any two nodes as the superposition of edges. The automatic process for key text graph construction is described in Fig.3. Based on the database images and 3D map, we can define a set of key text nodes. During this process, misrecognized texts can be corrected, and false detections can be removed. Each key text node includes its approximate 3D position, key text, and a 3D bounding box of the key text. By connecting each node to its nearest adjacent node, we can gather the shortest edges and categor ize them into two or three orthogonal directions. Subsequently, we establish the edge connections among the nodes solely based on these orthogonal directions. Note that this automatic strategy is effective in the standard grid layout of indoor parking lots.

B. Orientation Filtering Module

The orientation filtering module specifies and leaves only the database images whose 3D orientations in the map can align with the orientation of the query image, as in Fig.2. Because the orientations of database images are already given, orientation filtering can work if the orientation value of the query image is guaranteed. This condition can be satisfied by constructing two individual 3×3 directional matrices, $\mathbf{M}_w$ and $\mathbf{M}_q$, as in Fig.4. These matrices consist of the three identical unit directional vectors but in different coordinate systems, such as the map $\{w\}$ and the query camera $\{q\}$ coordinate systems. Before these constructions, it is reasonably assumed that the third columns of $\mathbf{M}_w$ and $\mathbf{M}_q$ are configured as the vertical directional vector in each coordinate system, respectively.

Constructing the directional matrix in $\{w\}$, $\mathbf{M}_w$ requires the key text graph and only the two dominant key texts in the query image: the key text with the largest bound box and its nearest key text. Since the key text graph directly provides the 3D direction from the key text of the largest bounding box to the other key text, the directional vector passing the key texts can be simply quantified as the superposition of the orthogonal directional vectors of the key text graph. Then, the first column of $\mathbf{M}_w$, is configured as the directional vector passing the key texts. Successively, the second column of $\mathbf{M}_w$ is quantified by the cross product of its third and first columns.

Constructing the directional matrix in $\{q\}$, $\mathbf{M}_q$ requires the identical key texts used in constructing $\mathbf{M}_w$, the 2D vanishing points, and the 3D vanishing directional vectors [18] of the query image. We then define a 2D line passing the two dominant key texts in the query image. Next, we calculate the distances between the 2D line and each of the 2D vanishing points. Then, it can be understood that the 3D line generated by the key texts in $\{q\}$ similarly heads for the 3D vanishing direction corresponding to the closest 2D vanishing point. Based on the direction the 3D line heads for, we can configure the order of the 3D vanishing directional vectors in $\{q\}$ that corresponds to that of the 3D orthogonal directional vectors in $\{w\}$ used in constructing $\mathbf{M}_w$. Additionally, using the superposition coefficients derived in constructing $\mathbf{M}_w$, the first column of $\mathbf{M}_q$ is configured. Successively, the second column of $\mathbf{M}_q$ is quantified by the cross product of its third and first columns.

Based on the fact that the coordinate system of a 3D directional vector can be transformed only by 3D rotation, the 3D orientation of the query image in the map coordinate system, $\mathbf{R}^w_{q_{est}}$, is estimated as

$$\mathbf{R}^w_{q_{est}} = \mathbf{M}_w \cdot (\mathbf{M}_q)^{-1}. \tag{1}$$

Therefore, the module filters out the database images that are not roughly aligned to the derived $\mathbf{R}^w_{q_{est}}$.

C. Similarity Scoring Module

Our similarity scoring module quantifies the similarity between the query image and database images and sorts out top-scored database image candidates. Different from

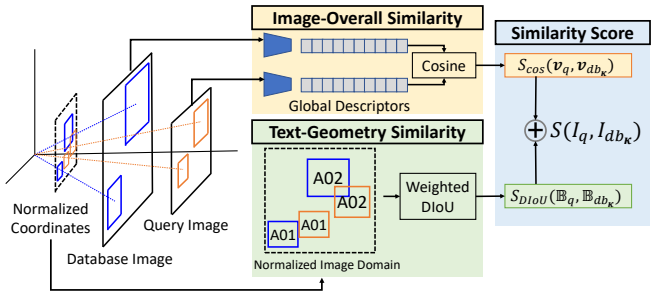


Fig. 5: **Similarity Scoring Module.** The similarity scoring module combines scores based on the image-overall similarity and text-geometry similarity. The image-overall similarity is computed from the cosine distance of global descriptors of a query and database image. The text-geometry similarity is computed based on weighted DIoU of bounding boxes in the normalized coordinates.

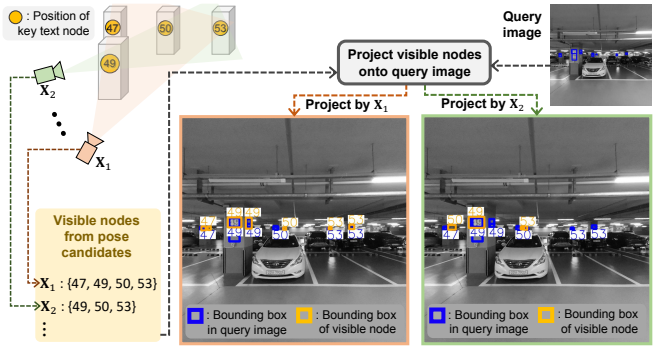


Fig. 6: **Key Text Projection Verification.** To quantify pose confidence for each pose candidate, our verification module projects visible key text nodes onto the estimated image plane and evaluate bounding box overlaps.

the existing scoring methods [7], [8], which only score the image-overall similarity, our scoring method simultaneously considers the image-overall and text-geometry similarities, as in Fig.5. The combination of the two terms strengths in distinguishing the repetitive spaces.

Our similarity score between a query image I_q and the κ-th database image I_{db_κ} is formulated as follows:

$$S(I_q, I_{db_\kappa}) = w_{cos} \cdot S_{cos}(\mathbf{v}_q, \mathbf{v}_{db_\kappa}) + w_{DIoU} \cdot S_{DIoU}(\mathbb{B}_q, \mathbb{B}_{db_\kappa}) \tag{2}$$

where w_{cos} and w_{DIoU} are weighting factors, $S_{cos}(\mathbf{v}_q, \mathbf{v}_{db_\kappa})$ indicates the cosine distance between $\mathbf{v}_q$ and $\mathbf{v}_{db_\kappa}$, the global descriptor vectors of I_q and I_{db_κ}, respectively. Next, $S_{DIoU}(\mathbb{B}_q, \mathbb{B}_{db_\kappa})$ indicates the weighted average of distance-IoU (DIoU) [19] between $\mathbb{B}_q$ and $\mathbb{B}_{db_\kappa}$. Here, $\mathbb{B}_q$ indicates a set of n_q key text bounding boxes in I_q, and $\mathbb{B}_{db_\kappa}$ indicates a set of n_{db_κ} key text bounding boxes in I_{db_κ}.

Specifically, the weighted average of DIoU, the second term of (2), is formulated as follows:

$$S_{DIoU}(\mathbb{B}_q, \mathbb{B}_{db_\kappa}) = \sum_{B_i^q \in \mathbb{B}_q} r_i \cdot \max\left(\left\{f_D\left(B_i^q, B_1^{db_\kappa}\right), \ldots, f_D\left(B_i^q, B_{n_{db_\kappa}}^{db_\kappa}\right)\right\}\right) \tag{3}$$

where B_i^q and $B_j^{db_\kappa}$ are the i-th and j-th elements of $\mathbb{B}_q$ and $\mathbb{B}_{db_\kappa}$, respectively, and r_i represents the ratio of the size of B_i^q to the sum of sizes of all elements in $\mathbb{B}_q$, assigning greater weights to larger bounding boxes. Additionally, $f_D(\cdot)$ indicates a conditional DIoU function using two bounding boxes as input, and it is formulated as follows:

$$f_D\left(B_i^q, B_j^{db_\kappa}\right) = \begin{cases} \mathrm{DIoU}(B_i^q, B_j^{db_\kappa}) & \text{if } \mathbf{k}^q(i) = \mathbf{k}^{db_\kappa}(j), \\ -1 & \text{otherwise} \end{cases} \tag{4}$$

where $\mathbf{k}^q(i)$ and $\mathbf{k}^{db_\kappa}(j)$ indicate i-th key text and j-th key text in I_q and I_{db_κ}, respectively. For matched key texts, in I_q, $\max\{\cdot\}$ in (3) outputs the highest DIoU value that is achieved by the most similar bounding box of $\mathbf{k}^q(i)$ among the bounding boxes of all the key texts in I_{db_κ} in text-geometry aspect. For unmatched key texts in I_q, the conditional DIoU as (4) results in a penalty of -1, thereby reducing the complete similarity score as (2). Additionally, $\mathrm{DIoU}(\cdot)$ operates not in the original image domains [19] but in the normalized image domain to cover the different camera specifications, as in Fig. 5.

Consequently, N_{ori} database images are sorted based on the similarity scores, (2), and N_{top} database images are selected for the pose estimation. Successively, the conventional PnP-RANSAC operates in the local feature matching and PnP step as in Fig. 2; the N_{top} pairs of 3D pose candidate and its corresponding numbers of inlier features are yielded.

D. Pose Verification Module

The final query pose is determined among the N_{top} pose candidates based on pose confidence computation in our pose verification module. Our module, key text projection verification (KPV), computes pose confidence based on the number of inlier points and the overlaps of key text bounding boxes. To compute overlaps of bounding boxes, KPV projects 3D bounding boxes of key texts visible at the estimated viewpoint onto the image plane, as in Fig. 6. For the bounding box projection, we utilize the 3D bounding box of key texts, which is contained in the node information of the key text graph. The pose confidence, $C_{\mathbf{X}_\kappa}$, is computed as follows:

$$C_{\mathbf{X}_\kappa} = \alpha \cdot \frac{\eta_\kappa}{\sum_{\tau=1}^{N_{top}} \eta_\tau} + (1-\alpha) \cdot \frac{\zeta_\kappa}{\sum_{\tau=1}^{N_{top}} \zeta_\tau} \tag{5}$$

where η_κ is the number of inliers used to estimate a pose candidate $\mathbf{X}_\kappa$, and ζ_κ is normalized weighted DIoU score as $\zeta_\kappa = (1 + S_{DIoU}(\mathbb{B}_q, \mathbb{B}_\kappa^{proj}))/2$, ranging from 0 to 1. Here, we compute $S_{DIoU}(\mathbb{B}_q, \mathbb{B}_\kappa^{proj})$ as (3), only substituting $\mathbb{B}_{db_\kappa}$ in (3) with $\mathbb{B}_\kappa^{proj}$, a set of bounding boxes projected from the 3D space onto the image plane, as in Fig. 6. The two terms in (5) are modulated by the weight factor $\alpha = \frac{1}{1+n_q}$. The weight factor ensures that the KPV prioritizes the number of inliers in the absence of key texts in the query image. As the number of key texts increases, the KPV weighs more on the bounding box overlaps.

IV. Experiments

A. Implementation Details

For OCR-based filtering, we utilized off-the-shelf models provided in [20]. Among the models, Mask-RCNN model [21] pre-trained on a CTW1500 dataset [22] for text detection and ABINet [23] for text recognition were employed. The weighting factors w_{cos} and w_{DIoU} in (2) were configured as 2 and 1, respectively. Next, the number of top-scored images, N_{top}, was configured as 10.

In the pose estimation step, we utilized state-of-the-art learned feature extractors, such as SFRS [8] for the global descriptors and SuperPoint [24] for the local features. Additionally, we utilized a graph neural network called SuperGlue [15] for local feature matching. The models and networks we employed for our experiment were the pre-trained models.

B. Experiment Datasets

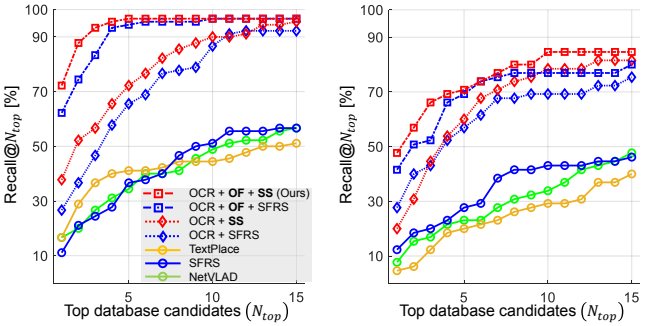
There were two actual large indoor parking lots for our experimental validation: parking lot A, whose area was about 8,750m^2, and parking lot B, whose area was about 6,250m^2. Following the database [4], we adopted a highly precise scanning sensor, Leica RTC360, to acquire images and a point cloud map of parking lot A. To cover the whole space, the sensor scanned the parking lot at 104 different positions, yielding 416 database images and a 3D point cloud map. Next, the query image data were acquired by using a low-cost calibrated hand-held scanner of TeeLabs. When each query image was captured, its corresponding LiDAR scan data were also captured. Thus, the reference pose of each query image could be defined by using the generalized iterative closest point (G-ICP) [25] to align the scan data with the constructed 3D map.

When it came to parking lot B, we adopted the same hand-held scanner of TeeLabs for database generation. For the database, a total of 1,569 images and a 3D point cloud map were obtained. Meanwhile, the query images were captured by a 52° field of view camera (e-con Systems See3CAM CU22), and their corresponding LiDAR (VLP-32C) scan data were acquired at the same time, where the sensors were calibrated by [26]. The ground truth poses of query images were defined by using the 3D transformation between LiDAR scan and 3D map by G-ICP [25] and the extrinsic calibration result between LiDAR and camera by [26].

The totals of 90 query images for parking lot A and 65 query images for parking lot B were used to compare the performance of our method to those of the existing methods such as [7], [8], [10] in Sec. IV-C and [4], [5], [6] in Sec. IV-D. Capturing the query images at different times when databases were constructed, we ensured the condition for long-term localization.

C. Image Retrieval Evaluation

Image retrieval techniques limit the upper bounds of visual localization performances [27]. Hence, it is the fundamental validation to compare our method and the existing methods in the retrieval aspect. Our image retrieval method is equivalent to a series of processes including the OCR-based filtering, the orientation filtering module, denoted as OF, and the similarity scoring module, denoted as SS, as in Fig. 2.



(a) Parking Lot A (b) Parking Lot B

Fig. 7: **Image retrieval performance.** The percentage of correctly localized queries is plotted against varying values of N_{top}, the number of retrieved database images.

We followed the standard place recognition evaluation metric [7], [8], [28]. A query image is regarded as correctly localized if at least one of the N_{top} retrieved database images is within specific thresholds of position and orientation from the ground truth query pose. Here, we set the threshold values as 15m and 40°. These values effectively constrain the domain of DB image poses, ensuring accurate feature matching and pose estimation in the subsequent localization process.

The recall, the percentage of correctly localized queries, was plotted against varying values of N_{top}, as in Fig. 7. The lines with circle markers showed the recall performance of the original existing methods [7], [8], [10], and the red dotted line with rectangle markers showed the recall performance of our method. In two different parking lots, our method totally outperformed the existing methods.

The existing methods [7], [8], [10] achieved relatively low performance due to their inability to distinguish the self-similarity within parking lots. If the repetitiveness and symmetry can be distinguished during the process of image retrieval, there is the potential that the recall performances of these methods will be improved. This potential could be observed by comparing the original SFRS [8], SFRS with OCR-based filtering, and SFRS with OCR-based filtering and our OF, which were shown as the blue line with circles (SFRS), the blue dotted line with diamonds (OCR + SFRS), and the blue dotted line with rectangles (OCR + OF + SFRS), respectively, in Fig. 7. As the filtering processes were gradually applied to the original SFRS [8], its recall performance was also gradually improved. These results verify the effectiveness of our filtering approaches addressing challenges due to the self-similarity in indoor parking lots.

The effectiveness of our SS was also verified by comparing the red and blue dotted lines in Fig. 7. Specifically, the red dotted line with diamonds (OCR + SS) represented the recall performances of SS with the OCR-based filtering, and the blue dotted line with diamonds (OCR + SFRS) represented that of SFRS [8] with the OCR-based filtering. Although (2)

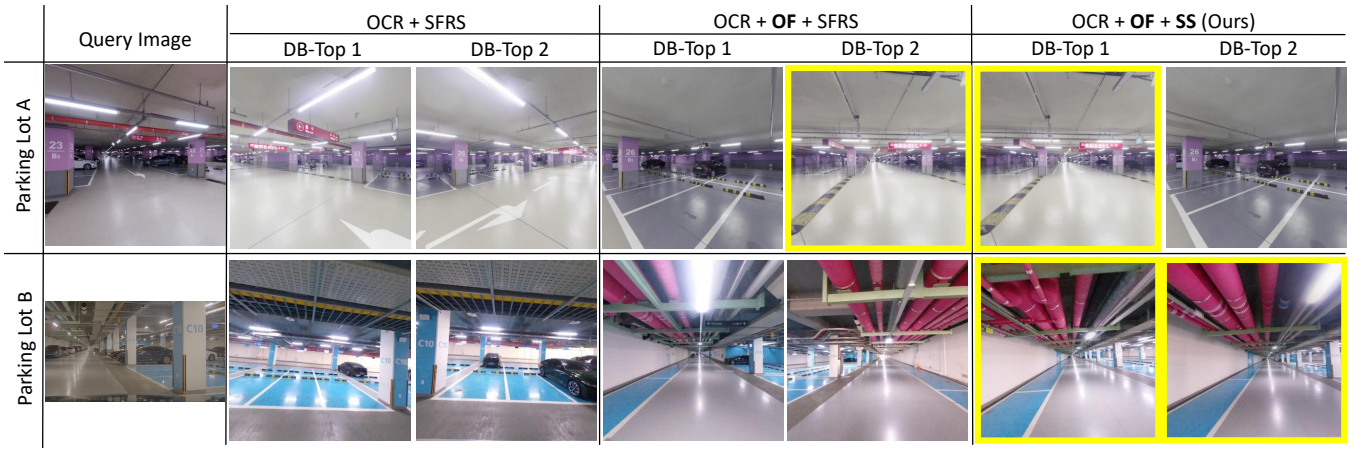


Fig. 8: **Visualization of image retrieval performance.** The baseline method, utilizing OCR-based filtering and SRFS [8]-based scoring, struggles with correct database image retrieval in self-similar parking lots. Our OF imposes the image retrieval process to discern the symmetric images and improves the effectiveness of retrieving the correct image. Our SS precisely addresses the repetitive problem, which outperforms SFRS. Yellow boxes highlight the correctly retrieved image.

TABLE I: Localization accuracy

Methods	[%] within evaluation thresholds (m, deg)		Average time $[ms]$
	Parking lot A	Parking lot B	
	(0.5, 5) / (1, 10) / (4, 15)	(1, 10) / (2, 10) / (4, 15)	
#Inliers [6]	57.8 / 62.2 / 62.2	70.3 / 71.9 / 78.1	$<$ **1.0**
PV [4]	67.8 / 68.9 / 68.9	32.8 / 39.1 / 45.3	2,199
MPV [5]	71.1 / 73.3 / 73.3	50.0 / 54.7 / 57.8	9,441
KPV (Ours)	**72.2 / 74.4 / 76.7**	**76.6 / 79.7 / 84.4**	2.3

contains the text-geometry similarity as (3), it was difficult to affirm that our scoring module outperformed the SFRS in all the N_{top} cases of the parking lots. However, if our OF was additionally guaranteed, SS outperformed the SFRS [8] in all the cases. This superiority was observed by comparing the red dotted line with rectangles (OCR + OF + SS) and the blue dotted line with rectangles (OCR + OF + SFRS) in Fig. 7. These results validated that our SS could operate effectively if the OCR-based filtering process and our OF were supported.

We also visualized the results by using a query image for each parking lot as in Fig. 8. Configuring the SFRS [8] with the OCR-based filtering as the baseline, we visualized the top 2 candidate database images derived from the baseline case, the baseline with our OF case, and our complete method, including OF and SS.

D. Localization Evaluation

Our pose verification module, KPV, was validated in terms of localization accuracy and computational time. For each query image, we compared KPV with representative pose verification methods such as PV [4], MPV [5], and the number of inlier features used in PnP, #Inliers [6]. We used the same top-10 pose candidates estimated with database images retrieved by our method, as our image retrieval method can output the most reliable pose candidates, as in Sec. IV-C. We set three accuracy evaluation thresholds for fine-level, medium-level, and coarse-level accuracy for each parking lot dataset. Considering the differences in sensors used for each parking lot dataset in Sec. IV-B, we configured three threshold sets as follows: $(0.5m, 5^\circ)$, $(1m, 10^\circ)$, and $(4m, 15^\circ)$, for parking lot A, and $(1m, 10^\circ)$, $(2m, 10^\circ)$, and $(4m, 15^\circ)$, for parking lot B, respectively.

Table I presents the percentages of correctly localized query images within each evaluation threshold set. In both parking lots and across all accuracy-evaluation thresholds, KPV consistently outperformed PV [4], MPV [5], and #Inliers [6]. In contrast to PV [4] and MPV [5], which rely on comparing a query image with its synthetic counterpart created by projecting colored point clouds, KPV evaluates the pose confidence utilizing the projection of 3D key text bounding boxes. This unique approach contributes to performance improvements.

Computation times are also presented in Table. I evaluated using Python on an Intel i7-9700K CPU. The average time cost of KPV was also superior to that of PV [4] and MPV [5] and roughly comparable to #Inliers [6]. This is because while PV and MPV necessitate pixel-wise computation [29], [30], KPV requires instance-wise computation, and #Inliers requires only sorting. These results confirm that KPV effectively selects the most reliable one among the candidates without incurring significant computational costs like PV or MPV.

V. Conclusion

Our study has addressed the challenges of visual localization in indoor parking lots characterized by self-similarity properties, namely repetitiveness and symmetry. Our method has been validated by using the two real indoor parking lot datasets in the aspects of image retrieval and localization accuracy performances. The results have demonstrated the effectiveness of key text graph construction and its applications for resolving spatial self-similarity challenges. We anticipate that our approach can be adapted to other environments where 3D key text graphs can be utilized.

REFERENCES

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

[2] L. Cui, C. Rong, J. Huang, A. Rosendo, and L. Kneip, "Monte-carlo localization in underground parking lots using parking slot numbers," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2021, pp. 2267–2274.

[3] J. Lv, C. Meng, Y. Wang, J. Sun, R. Xiong, and S. Pu, "So-pfh: Semantic object-based point feature histogram for global localization in parking lot," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2022, pp. 4431–4438.

[4] H. Taira, M. Okutomi, T. Sattler, M. Cimpoi, M. Pollefeys, J. Sivic, T. Pajdla, and A. Torii, "Inloc: Indoor visual localization with dense matching and view synthesis," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7199–7209.

[5] J. Hyeon, J. Kim, and N. Doh, "Pose correction for highly accurate visual localization in large-scale indoor spaces," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 15 974–15 983.

[6] P.-E. Sarlin, C. Cadena, R. Siegwart, and M. Dymczyk, "From coarse to fine: Robust hierarchical localization at large scale," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 12 716–12 725.

[7] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "Netvlad: Cnn architecture for weakly supervised place recognition," in *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, 2016, pp. 5297–5307.

[8] Y. Ge, H. Wang, F. Zhu, R. Zhao, and H. Li, "Self-supervising fine-grained region similarities for large-scale image localization," in *Proceedings of the European Conference on Computer Vision*, 2020, pp. 369–386.

[9] T. Sattler, W. Maddern, C. Toft, A. Torii, L. Hammarstrand, E. Stenborg, D. Safari, M. Okutomi, M. Pollefeys, J. Sivic, *et al.*, "Benchmarking 6dof outdoor visual localization in changing conditions," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8601–8610.

[10] Z. Hong, Y. Petillot, D. Lane, Y. Miao, and S. Wang, "Textplace: Visual place recognition and topological localization through reading scene texts," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 2861–2870.

[11] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[12] N. Vo, L. Jiang, C. Sun, K. Murphy, L.-J. Li, L. Fei-Fei, and J. Hays, "Composing text and image for image retrieval-an empirical odyssey," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 6439–6448.

[13] H. Wang, X. Bai, M. Yang, S. Zhu, J. Wang, and W. Liu, "Scene text retrieval via joint text detection and similarity learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 4558–4567.

[14] L. Wang, Y. Li, and S. Lazebnik, "Learning deep structure-preserving image-text embeddings," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2016, pp. 5005–5013.

[15] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superglue: Learning feature matching with graph neural networks," in *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, 2020, pp. 4938–4947.

[16] J. L. Schonberger and J.-M. Frahm, "Structure-from-motion revisited," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4104–4113.

[17] H. Taira, I. Rocco, J. Sedlar, M. Okutomi, J. Sivic, T. Pajdla, T. Sattler, and A. Torii, "Is this the right place? geometric-semantic pose verification for indoor visual localization," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 4373–4383.

[18] X. Lu, J. Yaoy, H. Li, Y. Liu, and X. Zhang, "2-line exhaustive searching for real-time vanishing point estimation in manhattan world," in *Proceedings of the IEEE Winter Conference on Applications of Computer Vision*, 2017, pp. 345–353.

[19] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, and D. Ren, "Distance-iou loss: Faster and better learning for bounding box regression," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 07, 2020, pp. 12 993–13 000.

[20] Z. Kuang, H. Sun, Z. Li, X. Yue, T. H. Lin, J. Chen, H. Wei, Y. Zhu, T. Gao, W. Zhang, *et al.*, "Mmocr: A comprehensive toolbox for text detection, recognition and understanding," in *Proceedings of the 29th ACM International Conference on Multimedia*, 2021, pp. 3791–3794.

[21] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2017, pp. 2961–2969.

[22] L. Yuliang, J. Lianwen, Z. Shuaitao, and Z. Sheng, "Detecting curve text in the wild: New dataset and new solution," *arXiv preprint arXiv:1712.02170*, 2017.

[23] S. Fang, H. Xie, Y. Wang, Z. Mao, and Y. Zhang, "Read like humans: Autonomous, bidirectional and iterative language modeling for scene text recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 7098–7107.

[24] D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superpoint: Self-supervised interest point detection and description," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 224–236.

[25] A. Segal, D. Haehnel, and S. Thrun, "Generalized-icp." in *Robotics: Science and Systems*, vol. 2, no. 4, 2009, p. 435.

[26] G. Koo, J. Kang, B. Jang, and N. Doh, "Precise camera–lidar extrinsic calibration based on a weighting strategy using analytic plane covariances," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–13, 2022.

[27] M. Humenberger, Y. Cabon, N. Pion, P. Weinzaepfel, D. Lee, N. Guérin, T. Sattler, and G. Csurka, "Investigating the role of image retrieval for visual localization: An exhaustive benchmark," *International Journal of Computer Vision*, vol. 130, no. 7, pp. 1811–1836, 2022.

[28] R. Arandjelović and A. Zisserman, "Dislocation: Scalable descriptor distinctiveness for location recognition," in *Proceedings of the Asian Conference on Computer Vision*, 2014, pp. 188–204.

[29] R. Arandjelović and A. Zisserman, "Three things everyone should know to improve object retrieval," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2012, pp. 2911–2918.

[30] C. Liu, J. Yuen, and A. Torralba, "Sift flow: Dense correspondence across scenes and its applications," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 5, pp. 978–994, 2010.