2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

Accurate and Efficient Loop Closure Detection With Deep Binary Image Descriptor and Augmented Point Cloud Registration

Jialiang Wang[1,2], Zhi Gao[1,3*], Zhipeng Lin[2], Zhiyu Zhou[1], Xiaonan Wang[4], Jianhua Cheng[4], Hao Zhang[2], Xinyi Liu[2] and Ben M. Chen[2]

***Abstract*— Loop Closure Detection (LCD) is an essential component of Simultaneous Localization and Mapping (SLAM), helping to correct drift errors, facilitate map merging, or both by identifying previously observed scenes. Despite its importance, traditional LCD algorithms based on single sensor such as camera or LiDAR exhibit degraded performance in challenging scenarios due to their inherent limitations. To address this issue, we propose a novel LCD method based on camera-LiDAR fusion, exploiting the rich textural information from cameras and the accurate geometric data from LiDAR to ensure robustness and speed in challenging environments. Specifically, we first employ deep hashing learning to encode deep image features into binary image descriptors for extremely fast loop candidate (LC) retrieval. Then, LiDAR points are augmented with image color for accurate geometric verification. Finally, we incorporate a spatial-temporal consistency check that mandates an LC to have consistently matched neighbors to be accepted as true. Our method is extensively verified and compared with the state-of-the-art methods on various datasets encompassing both indoor and outdoor environments. Experimental results demonstrate that our method obtains the best performance, increasing the maximum recall rate at 100% precision by a significant margin of 20% while operating in real-time at an average speed of 30 fps.**

I. Introduction

Loop Closure Detection (LCD) plays a critical role in both single-agent and multi-agent Simultaneous Localization and Mapping (SLAM) systems. Through its ability to recognize whether a robot has returned to a pre-visited area, LCD helps reduce accumulated errors and facilitates the merging of maps constructed by different robots. Despite significant progress made in LCD algorithms over the past few decades, there are still certain scenarios where their performance degrades. These scenarios are characterized by abrupt changes in lighting, environments with little or no texture, and perceptual aliasing, among others. Therefore, immediate efforts are needed to address these issues.

Traditionally, two major types of LCD methods have been studied: vision-based and LiDAR-based. Most vision-based methods [1]–[3] rely on extracting visual features to create global descriptors. Despite their high efficiency, global descriptors inevitably sacrifice image details and location information for the sake of descriptor compactness, leading to significant precision loss in the face of perceptual aliasing environments. Conversely, LiDAR-based methods focus on exploiting geometric characteristics, such as surface normal [4], height [5], angle and distance [6] to form descriptors. While these methods can precisely measure scene geometry, the lack of textural information in point clouds poses huge challenges when it comes to distinguishing places with similar geometric structures, such as corridors and tunnels.

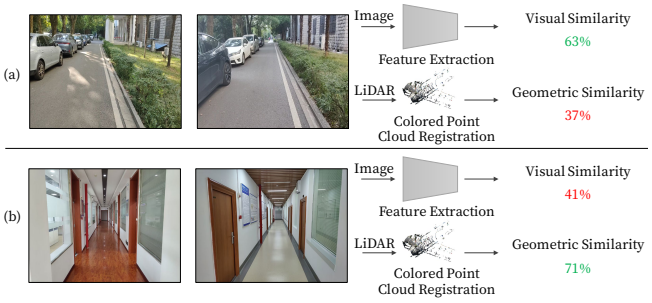


Fig. 1. Challenging scenarios encountered in traditional LCD. Both image pairs are taken from different places: (a) Visually similar places with noticeable structural differences (cars, building columns, etc.); (b) Geometrically indistinguishable corridors with evident texture and color distinctions. We reject these false loops by leveraging both visual and geometric information.

Recently, there has been significant interest in the fusion of modalities for LCD, resulting in notable achievements [7]–[11]. Specifically, Deep Neural Networks (DNNs) have been introduced in [7]–[9] to extract a shared embedding space across modalities and formulate a multi-modal descriptor. However, these methods are more likely enhanced versions of vision-based approaches rather than solid solutions that well preserve the internal structure of a scene. Additionally, these methods tend to be time-consuming as both modalities need to be processed with DNNs. Therefore, designing an LCD algorithm that is both fast and robust in complex real-life applications remains an open problem.

To tackle this challenge, we introduce a novel multi-stage LCD framework that leverages the rich textural information from cameras for fast retrieval and the precise geometric information from LiDAR for accurate verification. As shown in Fig. 1, only candidates with both visual and geometric similarity are accepted. To be specific, we first employ a deep hashing network to encode visual information into discriminative binary descriptors, which are indexed for efficient loop

*This work was supported in part by Hubei Province Natural Science Foundation under Grant 2021CFA088, and in part by Science and Technology Major Project under Grants 2021AAA010 and 2021AAA010-3 (Corresponding author: Zhi Gao gaozhinus@gmail.com).

[1]School of Remote Sensing and Information Engineering, Wuhan University, Wuhan, 430072, China.
[2]Department of Mechanical and Automation Engineering, The Chinese University of Hong Kong, Hong Kong SAR 999077, China.
[3]Hubei Luojia Laboratory, Wuhan University, Wuhan, 430079, China.
[4]ZG Technology Co., Ltd. Wuhan, 430000, China.

candidate (LC) retrieval. Subsequently, we augment LiDAR points with image color to perform colored Iterative Closet Point (ICP) registration, which effectively eliminates false loops by verifying their geometric similarity. Finally, we check whether the proposed loops have consistently matched spatial-temporal neighbors to ensure a precise selection of LC. Our method is therefore capable of reliable performance even in challenging environments. The contributions of our work can be summarized as follows:

1) We propose a novel LCD method based on camera-LiDAR fusion, leveraging deep image features and accurate point cloud geometries to achieve efficient and robust loop detection in challenging environments.
2) We incorporate a data-level fusion of visual and structural information by generating discriminative binary descriptors and RGB-colored point clouds, facilitating not only fast retrieval but also accurate verification.
3) We extensively evaluate our method on various indoor and outdoor datasets, outperforming the state-of-the-art algorithms by over 20% in terms of the maximum recall rate at 100% precision while operating at 30 fps.

II. Related Work

LCD is closely related to place recognition (PR) in that they both aim to identify similar places. However, there exist subtle differences between these two tasks. PR focuses on retrieving as many visited places as possible from a large database, with the primary objective of achieving high recall rates [12]. Conversely, LCD emphasizes precision since an incorrect loop may lead to catastrophic consequences for the entire system. Additionally, LCD favors low computational complexity and takes sequential data as input. Considering the close relationship between these two fields, PR methods that could be potentially adapted for LCD will also be discussed.

A. Vision-based LCD and Deep Hashing

Vision-based LCD commonly relies on global features extracted from images. One popular approach is the BoW method and its variants [1], [2], [13]. Such methods usually construct a vocabulary tree from visual words by clustering local features such as SIFT or ORB. This vocabulary tree is then used to generate a feature vector based on the distribution of visual words.

Similarly, another category of methods known as the Vector of Locally Aggregated Descriptors (VLAD), which aggregates handcrafted local features using K-means clustering [14], has reported competitive performance. Its deep-learning version, the NetVLAD family [3], [15]–[17], is later designed for visual place recognition (VPR) and LCD.

To boost computational and storage efficiency, many works have resorted to deep hashing methods, which encode high-dimensional features into compact binary codes using fully connected (FC) layers and carefully designed loss functions [18], [19]. These methods have been widely applied in large-scale image retrieval tasks [20], [21] and also explored in the context of VPR and LCD [22].

Despite their efficiency in candidate retrieval, LCD methods based on global features suffer from low precision in perceptual aliasing environments due to their inability to capture fine image details and accurate spatial structure. [23] This problem is largely mitigated in our method through geometric verification, which is discussed in Section III-B.

B. LiDAR-based LCD

Due to the strength of directly obtaining 3D structure, LiDAR-based LCD methods have also been proposed. In the early stage, the Normal Distribution Transform (NDT) histogram has been investigated to represent a 3D point cloud as a set of multivariate Gaussian distributions [4]. Another well-known method, called Scan Context, partitions the point cloud into bins and encodes the maximum point height of each bin as a feature, achieving lateral and rotational invariance [24]. Alternatively, LCD based on point cloud segmentation is explored in [25], [26] since segmented objects are noise-resistant and scale-invariant.

Similar to the vison-based counterpart, driven by DNN techniques, PointNetVLAD [27] has been proposed for PR based on point clouds, obtaining comparable performance with NetVLAD. More recently, transformer-based methods have been introduced to improve PR results by capturing short-range local features and long-range contextual features with attention mechanism [28], [29].

While LiDAR-based methods are robust to illumination change or viewpoint variation, they struggle to distinguish places with similar structures due to the lack of textural information, thereby limiting their practical applicability.

C. Multi-Modal LCD

Numerous methods have been proposed to leverage the complementary properties of cameras and LiDAR for improved performance [7]–[9]. In [7], a convolutional framework is introduced to transform image and LiDAR inputs into a shared representation space to enable joint detection and description of keypoints. However, adapting it for LCD will necessitate local matching of keypoints, which is exceedingly time-consuming. Similarly, the work [8] trains separate DNNs for each modality so that images and point clouds from the same location will have similar descriptors. However, this method only allows for matching images to high-definition LiDAR maps, which is not typically available in SLAM. In contrast, MinkLoc++ [9], conducts fusion in the final stage, processing each modality independently before concatenating the generated descriptors. Nonetheless, as this network tends to focus on the modality with a larger overfit to the training data, it often results in suboptimal performance. Although metric learning can potentially alleviate this issue, the network still lacks generalization ability.

In summary, the aforementioned methods adopt a descriptor-based detection strategy, which only partially addresses the challenges faced by vision and LiDAR based methods. Specifically, these methods fail to preserve the internal geometric structure of the environment, which is crucial for discriminating subtle differences between similar

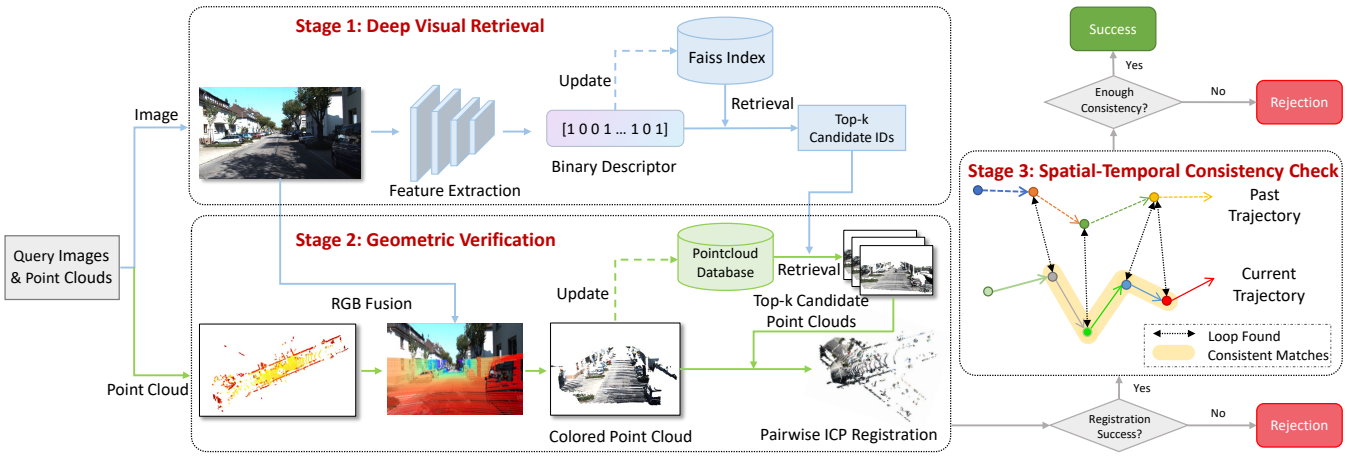


Fig. 2. Overview of the architecture of our proposed method.

places, in favor of descriptor compactness and computational efficiency. Consequently, the advantages offered by each sensor are not fully leveraged. In contrast, our proposed method combines deep visual features for a concise yet distinctive representation of the environment, while preserving the integral structure of scenes in the colored point clouds for geometric verification. This enables fast and robust performance in various challenging scenarios.

III. The Proposed Method

Figure 2 provides an overview of the proposed method, which consists of three stages: visual retrieval based on deep image feature, geometric verification, and spatial-temporal consistency check. The system input is continuous frames $\mathcal{F} = \{F_i\}_{i=1}^{n}$, with F_i containing aligned image I_i and point cloud $\mathbf{P}_i \in \mathbb{R}^{3\times n_p}$. For each frame F_i, the first stage retrieves a set of LCs $\mathcal{LC}_i = \{F_j\}_{j=1}^{k} \subset \mathcal{F}$ and the following stage iteratively refines the set to exclude false loops.

A. Deep Visual Retrieval With Binary Descriptor

Deep-learned image feature has been extensively studied in the context of VPR [12]. Inspired by one of its recent advancements, namely CosPlace [30], we incorporate a hashing layer to produce binary descriptors that largely accelerate the retrieval process. Generated descriptors are then encoded in a Faiss Index [31] for efficient similarity search. In the query stage, the feature vector for the query image is generated by the same network to retrieve the top-k similar candidates in the index, thus forming the initial LC set. The details of our network are elaborated below.

1) Dataset Preprocessing: LCD could be considered as a classification problem, where images taken at nearby locations are deemed to belong to a class. At training stage, we group dataset images into classes and assign a label to each class for supervised learning. For each image with known position x, y and heading α, we assign its label as a three-digit number l_{abc} by applying a spatial grid:

$$a = \lfloor \frac{x}{M} \rfloor, b = \lfloor \frac{y}{M} \rfloor, c = \lfloor \frac{\alpha}{\Delta} \rfloor, \tag{1}$$

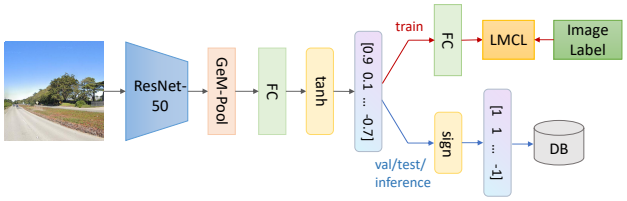


Fig. 3. Network architecture in the train and val/test/inference stage.

where $\lfloor \cdot \rfloor$ is the floor function, M is the length of the grid cell and Δ determines the extent of each class in terms of heading. However, due to the fact that Eq. 1 discretizes continuous space, it is possible for similar image pairs taken at nearly identical positions (but a few meters apart) to be wrongly assigned to different classes.

To address this issue, we reorganize non-adjacent classes into groups and train iteratively over them, so that there are no mislabelled image pairs within any single group. Formally, a group $\mathcal{G}_{uvw}$ is defined as a set of classes:

$$\begin{aligned}\mathcal{G}_{uvw} = \{l_{abc} \mid (a \bmod N = u) \wedge \\ (b \bmod N = v) \wedge (c \bmod L = w)\},\end{aligned} \tag{2}$$

where N and L are additional parameters controlling the spatial and directional separation between classes. For any two images belonging to different classes within a group, their positional and directional difference are guaranteed to be larger than $M \times (N-1)$ and $\Delta \times (L-1)$, respectively. This helps avoid confusion caused by discretization errors.

2) Network Design: Figure 3 illustrates the architecture of our network. We utilize ResNet-50 as the backbone, followed by a Generalized-mean (GeM) pooling and an FC layer. The network outputs a real-valued feature vector $\mathbf{x} \in \mathbb{R}^n$. During training stage, an additional FC layer is included after the output as a classifier, which maps the feature vector to the possibility distribution for each class, since the number of classes is usually unequal to the feature dimension. During val/test or inference stage, this classifier is discarded and we

binarize the output with a sign function: $\mathbf{h} = sgn(\mathbf{x}) \in \mathbb{Z}^n$. The final binary descriptor $\mathbf{h}$ is then used for retrieval.

Objective Function: To improve the separability of classes, we adopt Large Margin Cosine Loss (LMCL) [32], an improved version of Cross Entropy Loss, as our objective function. Given an image I_i with its feature vector $\mathbf{h}_i$, and label Y_i, the function is formulated as:

$$\mathcal{L}_{lmcl}(\mathbf{h}_i, Y_i) = \frac{1}{N}\sum_i -\log\frac{e^{s\left(\cos\left(\theta_{Y_i,i}\right)-m\right)}}{e^{s\left(\cos\left(\theta_{Y_i,i}\right)-m\right)}+\sum\limits_{j\neq Y_i} e^{s\cos\left(\theta_{j,i}\right)}}, \tag{3}$$

where N is the number of training samples, m is the separation margin, s is the norm of $\mathbf{h}_i$ and $\theta_{j,i}$ is the angle between $\mathbf{h}_i$ and the weight vector of the j-th class. The function enforces a large margin for different classes in the cosine space, which is beneficial for the network to learn more discriminative features.

Network Optimization: Directly optimizing the loss function above is difficult due to the vanishing gradient of binary descriptor $\mathbf{h}_i$. To enable end-to-end training, we first employ the hyperbolic tangent function to smooth the sign activation originally applied to the final FC layer. Then, we approximate the binary $\mathbf{h}_i$ in Eq. 3 by the real-valued $\mathbf{x}_i$ and add a quantization loss:

$$\mathcal{L}_{quant} = \left\|1 - |\mathbf{x}|\right\|_2. \tag{4}$$

This penalty term encourages the network to learn $sgn(\mathbf{x}) \approx \mathbf{x}$, reducing the approximation error caused by the sign function. The final loss function is the weighted sum of the LMCL loss and the quantization loss:

$$\mathcal{L} = \mathcal{L}_{lmcl}(\mathbf{x}_i, Y_i) + \lambda\mathcal{L}_{quant}. \tag{5}$$

B. Geometric Verification

While the initial LCs are visually similar to the query, their geometric similarity is not guaranteed. We hence perform a colored ICP registration [33] for accurate verification and to eliminate false loops.

1) Camera-LiDAR Fusion: The color information from the image is first projected into the point cloud to compensate for its lack of textural information. With a transformation matrix calibrated in advance, we acquire a mapping $g : \mathbb{R}^3 \mapsto \mathbb{Z}_+^2$ from the 3D point cloud to the 2D image. Those points with corresponding pixels outside of the image are cropped.

Since the mapping g is not bijective, some points in the point cloud may share the same pixel. We assign the color of the pixel to its nearest (projected) point and use a linearized function to approximate the colors of the remaining points [34]. Let $C_\mathbf{p}$ be a continuous color function defined in the vicinity of a point $\mathbf{p}$ with known color $C(\mathbf{p})$. For any nearby point $\mathbf{q}$, its color is approximated as:

$$C_\mathbf{p}(\mathbf{q}) \approx C(\mathbf{p}) + \mathbf{d}_\mathbf{p}^\top(\mathbf{f}(\mathbf{q}) - \mathbf{p}), \tag{6}$$

where $\mathbf{d}_\mathbf{p}$ is the gradient of $C_\mathbf{p}$ at $\mathbf{p}$ and $\mathbf{f}(\mathbf{q})$ is the projection of $\mathbf{q}$ onto the tangent plane of $\mathbf{p}$:

$$\mathbf{f}(\mathbf{q}) = \mathbf{q} - \mathbf{n}_\mathbf{p}(\mathbf{q} - \mathbf{p})^\top \mathbf{n}_\mathbf{p}. \tag{7}$$

To estimate the gradient $\mathbf{d}_\mathbf{p}$ at $\mathbf{p}$, we fit it to the colors of its local neighbors $\mathcal{N}_\mathbf{p}$ using least squares regression:

$$\mathbf{d}_\mathbf{p} = \underset{\mathbf{d}_\mathbf{p}}{argmin}\sum_{\mathbf{p}'\in\mathcal{N}_\mathbf{p}}\left\|C(\mathbf{p}) + \mathbf{d}_\mathbf{p}^\top(\mathbf{f}(\mathbf{p}') - \mathbf{p}) - C(\mathbf{p}')\right\|_2. \tag{8}$$

2) Colored Point Cloud Registration: Having obtained the colored point clouds, we measure their geometric similarity by calculating the ratio of inlier points after point cloud registration. The objective function for the optimal transformation $\mathbf{T}$ between two point clouds $\mathbf{P}$ and $\mathbf{Q}$ is formulated as:

$$E(\mathbf{T}) = (1-\sigma)E_C(\mathbf{T}) + \sigma E_G(\mathbf{T}), \tag{9}$$

where σ is a weight factor and E_C and E_G denote the photometric and geometric terms, respectively.

After i-th registration, a correspondence set $\mathcal{K} = \{(\mathbf{p}, \mathbf{q})\}$ is acquired, where $\mathbf{p}$ represents a point in $\mathbf{P}$ and $\mathbf{q}$ represents a point in $\mathbf{Q}$. Let $\mathbf{q}\prime = \mathbf{T}\mathbf{q}$ denote the transformed point of $\mathbf{q}$, the photometric term can be formulated as the sum of squared color differences $\mathbf{r}_C^{(\mathbf{p},\mathbf{q})}$:

$$E_C(\mathbf{T}) = \sum_{(\mathbf{p},\mathbf{q})\in\mathcal{K}}\mathbf{r}_C^{(\mathbf{p},\mathbf{q})} = \sum_{(\mathbf{p},\mathbf{q})\in\mathcal{K}}\left\|C_\mathbf{p}(\mathbf{f}(\mathbf{q}\prime)) - C(\mathbf{q})\right\|_2, \tag{10}$$

where $C_\mathbf{p}(\cdot)$ is the continuous color function calculated in Eq. 6 and $\mathbf{f}(\cdot)$ is the projection function defined by Eq. 7.

Similarly, the geometric term is calculated by accumulating the squared point-to-plane distances $\mathbf{r}_G^{(\mathbf{p},\mathbf{q})}$ between the transformed point $\mathbf{q}\prime$ and the tangent plane of $\mathbf{p}$:

$$E_G(\mathbf{T}) = \sum_{(\mathbf{p},\mathbf{q})\in\mathcal{K}}\mathbf{r}_G^{(\mathbf{p},\mathbf{q})} = \sum_{(\mathbf{p},\mathbf{q})\in\mathcal{K}}\left\|(\mathbf{q}\prime - \mathbf{p})^\top\mathbf{n}_\mathbf{p}\right\|_2. \tag{11}$$

The objective function is optimized following the Gauss-Newton method. For each iteration, $\mathbf{T}$ is linearized around the current estimate $\mathbf{T}^k$ by a vector $\xi = (\alpha, \beta, \gamma, x, y, z)$ which represents a rotational and a translational component:

$$\mathbf{T} = \begin{pmatrix} 1 & -\gamma & \beta & x \\ \gamma & 1 & -\alpha & y \\ -\beta & \alpha & 1 & z \\ 0 & 0 & 0 & 1 \end{pmatrix}\mathbf{T}^k, \tag{12}$$

We calculate the Jacobian matrix $\mathbf{J}_\mathbf{r}$ of the objective function with respect to ξ and solve the linear system:

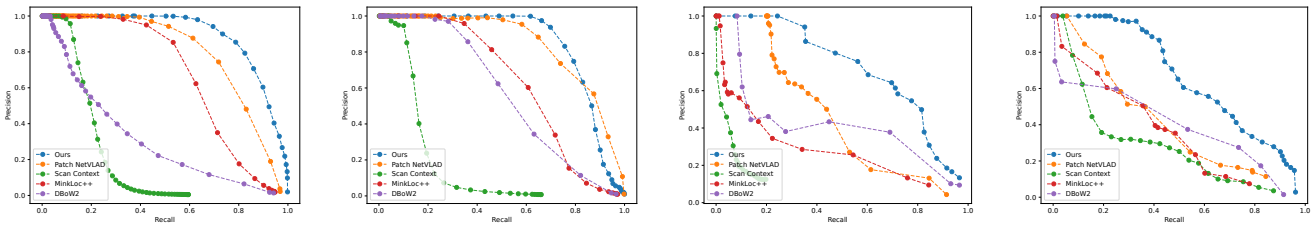
$$\mathbf{J}_\mathbf{r}{}^\top\mathbf{J}_\mathbf{r}\xi = -\mathbf{J}_\mathbf{r}{}^\top\mathbf{r}, \tag{13}$$

Note that both the Jacobian and the residual vector are accumulated over all the correspondences and consist of photometric and geometric parts:

$$\mathbf{J}_\mathbf{r}{}^\top\mathbf{J}_\mathbf{r} = \sum_{(\mathbf{p},\mathbf{q})\in\mathcal{K}}\mathbf{J}_{\mathbf{r}_C}^\top\mathbf{J}_{\mathbf{r}_C} + \mathbf{J}_{\mathbf{r}_G}^\top\mathbf{J}_{\mathbf{r}_G} \tag{14}$$

$$\mathbf{J}_\mathbf{r}{}^\top\mathbf{r} = \sum_{(\mathbf{p},\mathbf{q})\in\mathcal{K}}\mathbf{J}_{\mathbf{r}_C}^\top\mathbf{r}_C + \mathbf{J}_{\mathbf{r}_G}^\top\mathbf{r}_G \tag{15}$$

where $\mathbf{r}_C$ and $\mathbf{r}_G$ are the residuals defined in Eq. 10 and 11, respectively, with overscript $(\mathbf{p}, \mathbf{q})$ omitted for brevity.



(a) KITTI 00 (b) KITTI 05 (c) TUM (d) Self-recorded

Fig. 4. Precision-recall curves of our method and other state-of-the-art methods.

Specifically, the Jacobian matrices of the photometric term and the geometric term with respect to ξ are calculated as:

$$\underset{1\times 6}{\mathbf{J}_{\mathbf{r}_C}} = \sqrt{1-\sigma}[\mathbf{q}\times \underset{1\times 3}{\mathbf{d}_\mathbf{p}^\top(\mathbf{I}-\mathbf{n}_\mathbf{p}\mathbf{n}_\mathbf{p}^\top)}, \underset{1\times 3}{\mathbf{d}_\mathbf{p}^\top(\mathbf{I}-\mathbf{n}_\mathbf{p}\mathbf{n}_\mathbf{p}^\top)}], \quad (16)$$

$$\underset{1\times 6}{\mathbf{J}_{\mathbf{r}_G}} = \sqrt{\sigma}[\underset{1\times 3}{\mathbf{q}\times\mathbf{p}}, \underset{1\times 3}{\mathbf{n}_\mathbf{p}}]. \quad (17)$$

C. Spatial-Temporal Consistency Check

Different from PR, LCD operates on sequential data, which allows for further validation. Ideally, if an LC is successfully matched to the current frame, its spatial-temporal neighbors will also be matched. This effectively eliminates the occasionally detected false loops.

Given a sequence of frames with sample rate f Hz and average disposition d, we denote the timestamp, viewing angle and position for the frame F_i as $\{t_i, \alpha_i, \mathbf{X}_i\}$. *$F_i$ and F_j are defined as* spatial-temporal neighbors *(written as $F_i \sim F_j$) if they satisfy:*

$$(t_i - t_j < \Delta t) \wedge (|\alpha_i - \alpha_j| < \Delta\alpha) \wedge (\|\mathbf{X}_i - \mathbf{X}_j\| < \Delta x) \quad (18)$$

Following the definition given above, we may now give the criterion of the spatial-temporal consistency check:

Given a query frame F_q and m consecutive frames $\{F_{q-1}, F_{q-2}, \cdots, F_{q-m}\}$ before it, $\forall F_a \in \mathcal{LC}_q$, *the loop between F_q and candidate F_a is considered* valid *only if there exists $F_b \in \mathcal{LC}_{q-1}, F_c \in \mathcal{LC}_{q-2}, \cdots, F_m \in \mathcal{LC}_{q-m}$, where $F_a \sim F_b, F_b \sim F_c, \cdots, F_{m-1} \sim F_m$.*

The four hyperparameters $\Delta t, \Delta\alpha, \Delta x$ and m control the strictness of the check. Either decreasing the first three or increasing the last one will result in a more stringent check. We set $\Delta t = 20/f, \Delta\alpha = 25°, \Delta x = 10d$ and $m = 2$ for outdoor datasets but limiting $\Delta\alpha = 10°$ in indoor datasets since a large rotation of cameras in a confined space usually results in a significant change of the scene.

IV. Experimental Results

A. Implementation Details

Concerning the image feature extraction network, the dimension of the binary descriptor is empirically set to 8k to achieve the best results. We set m in Eq. 3 to 0.4, λ in Eq. 5 to 0.01, respectively. The network is trained for 100 epochs on two datasets: the San Francisco eXtra Large dataset [30] for outdoor imagery and the InLoc dataset [35] for indoor imagery.

In the visual retrieval stage, the top 10 candidates are proposed for further geometric verification. For all point clouds, we apply voxel downsampling and ground removal to improve registration robustness. The colored ICP registration is executed with a maximum of 30 iterations. Two point clouds pass the verification if the inlier ratio exceeds 60%.

B. Evaluation Datasets

We evaluate our method on two widely-used public datasets and one self-recorded dataset.

1) KITTI Dataset: The KITTI odometry dataset [36] consists of high-resolution stereo images, 64-line lidar point clouds, and precise ground truth poses. We use two sequences, KITTI 00 and 05, to examine algorithm performance in outdoor environments.

2) TUM RGB-D Dataset: The TUM RGB-D dataset [37] provides synchronized RGB-D sensor data, along with ground truth camera poses from a motion-capture system. We use the `fr2/large_with_loop` sequence, which is a long trajectory captured in a low-texture environment.

3) Self-recorded Dataset: We record another indoor sequence with Intel Realsense L515 in the city art gallery. The camera is mounted on a handheld device and moved through two exhibition halls with very similar layouts. The sequence is thus challenging and suitable for evaluating algorithm robustness against perceptual aliasing.

C. Loop Detection Performance

In this section, our method is benchmarked and compared with the state-of-the-art LCD or PR methods, which include `DBoW2` [13], a classic and widely used BoW method based on ORB feature; `Patch NetVLAD` [15], an advanced version of NetVLAD that incorporates patch-level features; `Scan Context` [24], a popular LiDAR-based LCD with handcrafted feature based on point height; `MinkLoc++` [9], a recent multi-modal PR method using deep-learned features.

The performance of LCD is evaluated in terms of maximum recall rate at 100% precision and precision-recall (P-R) curves. However, literature adopts two ways in P-R calculation, which we define as *frame-wise* or *pair-wise*. Consider a simple scenario where F_q is the only query frame and is a loop to F_l, F_{l+1}, F_{l+2}, and the algorithm only retrieves F_l as a candidate.

The frame-wise method would consider the recall rate as 100% since the total amount of query frames is 1 and

TABLE I

MAXIMUM RECALL RATE AT 100% PRECISION BY FRAME-WISE (FW) AND PAIR-WISE(PW) EVALUATION (%)

Methods	KITTI 00		KITTI 05		TUM		Self-recorded	
	PW	FW	PW	FW	PW	FW	PW	FW
DBoW2	2.35	39.6	17.4	33.8	8.32	27.3	0.24	6.21
Patch NetVLAD	28.5	**91.7**	23.1	**85.9**	20.5	83.3	5.42	37.3
Scan Context	8.12	82.8	3.84	78.3	0.03	0.03	3.84	12.7
MinkLoc++	15.1	84.2	24.3	77.2	1.22	19.4	1.4	10.1
Ours	**50.3**	91.1	**48.9**	83.0	**24.3**	**84.7**	**22.6**	**42.7**

F_q is correctly identified. However, the pair-wise method calculates it as 33.3% because it aims to retrieve all loop pairs. While the frame-wise method is most commonly used, we argue that the pair-wise method is rather important as the quality of F_l, F_{l+1}, F_{l+2} could be different in terms of optimizing the SLAM system, which will be proved in the following section. This suggests that a higher pair-wise recall helps to find more reliable loops. We present both methods in Table I but only the pair-wise method in Fig. 4 since the trend of the curve is similar for both methods.

The results show that our method outperforms the existing methods on both indoor and outdoor datasets in terms of Pair-wise evaluation, increasing the maximum recall rate by a significant margin of 20%. In frame-wise evaluation, our method also achieves the highest recall rate on indoor datasets but is slightly inferior to Patch NetVLAD on outdoor datasets. We attribute this to the fact that Patch-NetVLAD is specially designed for outdoor VPR and has very high top-k recalls, which suits the frame-wise evaluation better.

However, our method focuses more on robustness and accuracy. Figure 5(a) illustrates how Patch NetVLAD, despite utilizing patch-level features to enhance image details, consistently generates false loops due to perceptual aliasing. While applying a stricter threshold could eliminate these false loops, it would also compromise stability.

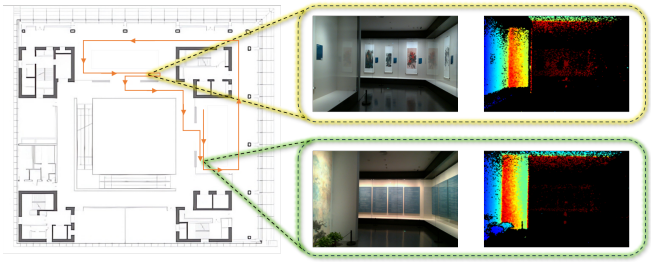
It is also noteworthy that both Scan Context and MinkLoc++ perform poorly in the indoor datasets. This is attributed to their primary design or training for outdoor environments characterized by open scenes and sparse point clouds. However, as depicted in Fig. 5(b), the indoor point clouds are dense and relatively homogeneous, posing a substantial challenge for these methods. Our method achieves comparable performance in both environments due to our utilization of visual cues for primary retrieval and the applicability of colored ICP registration to both data types.

TABLE II

ABSOLUTE TRAJECTORY ERROR OF OUR METHOD AND ORB SLAM2 ON TWO DATASETS (UNIT: M)

Methods	KITTI 00			TUM		
	MIN	RMSE	MAX	MIN	RMSE	MAX
ORB SLAM2	0.145	1.363	3.738	0.044	0.226	0.438
Ours	**0.072**	**1.291**	**3.577**	**0.019**	**0.087**	**0.186**

(a) False loop detected by Patch NetVLAD in KITTI 00



(b) False loop detected by Scan Context in the self-recorded dataset

Fig. 5. Examples of false loops detected by other methods. The retrieved candidate in (a) shares similar image features on the left side, but the right part is different. Our method rules it out by geometric verification. Indoor scenes in (b) have almost identical geometric structures, failing the LiDAR-based methods. The image features are nevertheless different so the candidate is rejected by our method in the first stage.

D. Quality of Loop Candidates

While higher recall rates suggest better performance, detected loops are utilized for trajectory optimization and generally, a single good loop is adequate to correct accumulated errors, rendering additional candidates unused and justifying frame-wise evaluation. However, we argue that the qualities of LCs are nevertheless different, and the additional candidates are important as they may provide better optimization opportunities. Therefore, we calculate the absolute trajectory error (ATE) after loop closure, which is defined as the positional difference between the ground truth and the estimated trajectory, to illustrate the quality of LCs.

Given the fact that our method can detect more loops than others, it would be impractical to expect them to outperform us. Therefore, we choose to exclusively evaluate our method against DBoW2, and the performance of other methods should be somewhere in between. We replace the LCD module of ORB SLAM2 by DBow2 and our method while keeping the rest of the system unchanged. Representative results are listed in Table II.

Due to the optimization mechanism of ORB SLAM2, only a fixed number of global adjustments are performed (4 times in KITTI 00 and 1 time in TUM) even though our method provides more loop opportunities. Nonetheless, our method's ATE is up to 60% lower than that of ORB SLAM2, indicating the superior quality of our loops. Note that our method performs significantly better in the low-texture indoor dataset, where handcrafted ORB features are less effective than deep-learned counterparts.

TABLE III

ABLATION STUDY ON FOUR DATASETS

Methods	KITTI 00				KITTI 05				TUM				Self-Recorded			
	LC	FL	Prec.	Rec.	LC	FL	Prec.	Rec.	LC	FL	Prec.	Rec.	LC	FL	Prec.	Rec.
Baseline (pure visual retrieval)	1656	426	74.2	62.4	810	440	44.7	92.8	275	20	92.7	16.1	125	19	84.8	20.3
Ours (visual retrieval + consistency check)	1474	288	80.4	60.2	680	320	52.9	92.3	238	14	94.1	14.2	111	4	95.4	20.3
Ours (visual retrieval + vanilla ICP)	287	4	98.2	14.3	37	1	97.2	9.23	53	2	96.2	3.23	57	0	100	10.9
Ours (visual retrieval + colored ICP)	335	9	97.3	16.5	52	3	94.2	12.5	143	6	95.8	8.68	72	0	100	13.7
Ours (complete method)	**330**	**4**	**98.7**	**16.5**	**49**	**1**	**97.9**	**12.3**	**94**	**3**	**96.8**	**5.70**	**72**	**0**	**100**	**13.7**

E. Ablation Study

To evaluate the effectiveness of each component in our method, we conduct an ablation study on four datasets. We compare the following variations of our method: `Baseline`, which includes only the visual retrieval stage without any further verification; `Visual retrieval with consistency check`, which adds the spatial-temporal consistency check module but employs no geometric data for verification; `Visual retrieval with vanilla ICP`, which sets $\sigma = 1$ in Eq. 9 and reduces the objective function to mere geometric terms (equivalent to vanilla ICP); `Visual retrieval with colored ICP`, which sets $\sigma = 0.96$ to add photometric terms; `Complete method`, which includes all three stages elaborated in Section III. For all variations, the threshold for geometric verification is set to 40%.

Since both geometric verification and spatial-temporal consistency check are performed on previously retrieved results, it will inevitability increase precision and reduce recall. To provide a comprehensive evaluation, we have thus chosen to present the total number of loop candidates (LC) and the number of false loops (FL) included in addition to precision and recall rates. The results are shown in Table III.

It is notable that adding geometric verification significantly reduces the number of false loops. The colored ICP yields slightly better results than vanilla ICP by providing dozens more loop opportunities at the cost of including only few false loops. The colored ICP performs particularly well in indoor sequences where geometric constraints are relatively insufficient, which helps improve the recall rate without any precision loss in our self-recorded dataset. With spatial-temporal consistency check enabled, the complete method achieves the highest precision while preserving a fairly large number of correct loops for trajectory optimization, demonstrating the effectiveness of each component.

F. Computational Efficiency

The binary descriptor in our method is extremely efficient for storage and query. Compared to its real-valued counterpart (produced without sign activation, equivalent to original CosPlace descriptor), it is 32x smaller (boolean vs float32) in size and up to 20x faster to query (Hamming distance vs Euclidean distance). Although binary descriptors are less accurate than real-valued ones with the same dimension, a higher dimension can compensate for this while preserving superior efficiency. Table IV shows that the 8k-dimensional binary descriptor outperforms 2k-dimensional real-valued descriptors in terms of both speed and accuracy.

TABLE IV

COMPARISON OF BINARY AND REAL-VALUED DESCRIPTORS

Type	Rec.@100%Prec.[1]	Add to index (ms)[2]	Query by index (ms)[2]
2k Float	50.3	31.7	5.85
8k Bool	51.9	2.91	1.99

[1] measured on KITTI 00; [2] on an index size of 10k

TABLE V

AVERAGE EXECUTION TIME OF DIFFERENT METHODS (UNIT: MS)

Methods	Feature Extraction	Query	Verification	Total
DBoW2	**12.51**	3.82	**2.44**	**18.77**
Patch NetVLAD	42.13	37.89	N/A	80.02
Scan Context	113.29	8.07	N/A	121.36
MinkLoc++	77.63	**0.01**	N/A	77.64
Ours	15.91	**0.01**	13.36	29.28

We further measure the time consumption of the whole system on KITTI 00 and compare it against the aforementioned methods. We split a typical LCD framework into two or three stages: feature extraction, candidate query, and optionally, candidate verification. The average time consumption for each stage is shown in Table V.

Due to the lightweight network and efficient descriptor, our method achieves real-time performance at an average frame rate of 30 Hz, which is 2-3 times faster than other deep-learning-based methods. Although DBoW2 is the fastest method so far, it is also the least accurate. Our approach strikes a favorable balance between speed and accuracy. Furthermore, the execution time does not significantly increase as the database grows larger due to the non-exhaustive nature of the Faiss Index and the fixed number of proposed candidates for verification. Additionally, the colored ICP registration can be easily parallelized since each candidate is verified independently.

V. CONCLUSIONS

In this paper, we propose a novel algorithm for fast and robust LCD via synergizing deep visual retrieval and accurate geometric verification. The proposed method extracts deep-learned binary descriptors from images for efficient candidate retrieval before fusing RGB information with structural data from point clouds to verify geometric similarity, enabling

reliable performance in challenging environments. Experimental results demonstrate that our method outperforms existing methods in both indoor and outdoor environments by a significant margin. The detected loop candidates are proved to be of better quality to reduce trajectory errors. The whole system achieves real-time performance and could be hopefully integrated into SLAM systems soon. Future work will focus on algorithm robustness in extremely dynamic conditions and optimization for different sensor types and hardware configurations.

REFERENCES

[1] E. Garcia-Fidalgo and A. Ortiz, "ibow-lcd: An appearance-based loop-closure detection approach using incremental bags of binary words," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3051–3057, 2018.

[2] H. Yue, J. Miao, W. Chen, W. Wang, F. Guo, and Z. Li, "Automatic vocabulary and graph verification for accurate loop closure detection," *Journal of Field Robotics*, vol. 39, no. 7, pp. 1069–1084, 2022.

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

[4] J. Saarinen, H. Andreasson, T. Stoyanov, and A. J. Lilienthal, "Normal distributions transform monte-carlo localization (ndt-mcl)," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 382–389, IEEE, 2013.

[5] M. Bosse and R. Zlot, "Place recognition using keypoint voting in large 3d lidar datasets," in *2013 IEEE International Conference on Robotics and Automation*, pp. 2677–2684, IEEE, 2013.

[6] H. Zhao, M. Tang, and H. Ding, "Hoppf: A novel local surface descriptor for 3d object recognition," *Pattern Recognition*, vol. 103, p. 107272, 2020.

[7] B. Wang, C. Chen, Z. Cui, J. Qin, C. X. Lu, Z. Yu, P. Zhao, Z. Dong, F. Zhu, N. Trigoni, *et al.*, "P2-net: Joint description and detection of local features for pixel and point matching," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 16004–16013, 2021.

[8] D. Cattaneo, M. Vaghi, S. Fontana, A. L. Ballardini, and D. G. Sorrenti, "Global visual localization in lidar-maps through shared 2d-3d embedding space," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4365–4371, IEEE, 2020.

[9] J. Komorowski, M. Wysoczańska, and T. Trzcinski, "Minkloc++: lidar and monocular image fusion for place recognition," in *2021 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, IEEE, 2021.

[10] A. J. Lee, S. Song, H. Lim, W. Lee, and H. Myung, "$(lc)^2$: Lidar-camera loop constraints for cross-modal place recognition," *IEEE Robotics and Automation Letters*, vol. 8, no. 6, pp. 3589–3596, 2023.

[11] M. Chghaf, S. R. Flórez, and A. El Ouardi, "A multimodal loop closure fusion for autonomous vehicles slam," *Robotics and Autonomous Systems*, vol. 165, p. 104446, 2023.

[12] X. Zhang, L. Wang, and Y. Su, "Visual place recognition: A survey from deep learning perspective," *Pattern Recognition*, vol. 113, p. 107760, 2021.

[13] D. Gálvez-López and J. D. Tardos, "Bags of binary words for fast place recognition in image sequences," *IEEE Transactions on Robotics*, vol. 28, no. 5, pp. 1188–1197, 2012.

[14] H. Jégou, M. Douze, C. Schmid, and P. Pérez, "Aggregating local descriptors into a compact image representation," in *2010 IEEE computer society Conference on Computer Vision and Pattern Recognition*, pp. 3304–3311, IEEE, 2010.

[15] S. Hausler, S. Garg, M. Xu, M. Milford, and T. Fischer, "Patch-netvlad: Multi-scale fusion of locally-global descriptors for place recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14141–14152, 2021.

[16] Y. Xu, J. Huang, J. Wang, Y. Wang, H. Qin, and K. Nan, "Esa-vlad: A lightweight network based on second-order attention and netvlad for loop closure detection," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 6545–6552, 2021.

[17] K. Zhang, J. Ma, and J. Jiang, "Loop closure detection with reweighting netvlad and local motion and structure consensus," *IEEE/CAA Journal of Automatica Sinica*, vol. 9, no. 6, pp. 1087–1090, 2022.

[18] R. Xia, Y. Pan, H. Lai, C. Liu, and S. Yan, "Supervised hashing for image retrieval via image representation learning," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 28, 2014.

[19] L. Yuan, T. Wang, X. Zhang, F. E. Tay, Z. Jie, W. Liu, and J. Feng, "Central similarity quantization for efficient image and video retrieval," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 3083–3092, 2020.

[20] X. Luo, H. Wang, D. Wu, C. Chen, M. Deng, J. Huang, and X.-S. Hua, "A survey on deep hashing methods," *ACM Trans. Knowl. Discov. Data*, vol. 17, feb 2023.

[21] W. Song, Z. Gao, R. Dian, P. Ghamisi, Y. Zhang, and J. A. Benediktsson, "Asymmetric hash code learning for remote sensing image retrieval," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–14, 2022.

[22] L. Wu and Y. Wu, "Deep supervised hashing with similar hierarchy for place recognition," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3781–3786, 2019.

[23] K. A. Tsintotas, L. Bampis, and A. Gasteratos, "The revisiting problem in simultaneous localization and mapping: A survey on visual loop closure detection," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 11, pp. 19929–19953, 2022.

[24] G. Kim and A. Kim, "Scan context: Egocentric spatial descriptor for place recognition within 3D point cloud map," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, (Madrid), Oct. 2018.

[25] R. Dubé, D. Dugas, E. Stumm, J. Nieto, R. Siegwart, and C. Cadena, "Segmatch: Segment based place recognition in 3d point clouds," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5266–5272, IEEE, 2017.

[26] Y. Fan, Y. He, and U.-X. Tan, "Seed: A segmentation-based egocentric 3d point cloud descriptor for loop closure detection," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5158–5163, IEEE, 2020.

[27] M. A. Uy and G. H. Lee, "Pointnetvlad: Deep point cloud based retrieval for large-scale place recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4470–4479, 2018.

[28] Z. Hou, Y. Yan, C. Xu, and H. Kong, "Hitpr: Hierarchical transformer for place recognition in point cloud," in *2022 International Conference on Robotics and Automation (ICRA)*, pp. 2612–2618, IEEE, 2022.

[29] T.-X. Xu, Y.-C. Guo, Z. Li, G. Yu, Y.-K. Lai, and S.-H. Zhang, "Transloc3d: point cloud based large-scale place recognition using adaptive receptive fields," *Communications in Information and Systems*, vol. 23, no. 1, pp. 57–83, 2023.

[30] G. Berton, C. Masone, and B. Caputo, "Rethinking visual geo-localization for large-scale applications," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4878–4888, 2022.

[31] J. Johnson, M. Douze, and H. Jégou, "Billion-scale similarity search with GPUs," *IEEE Transactions on Big Data*, vol. 7, no. 3, pp. 535–547, 2019.

[32] H. Wang, Y. Wang, Z. Zhou, X. Ji, D. Gong, J. Zhou, Z. Li, and W. Liu, "Cosface: Large margin cosine loss for deep face recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5265–5274, 2018.

[33] J. Park, Q.-Y. Zhou, and V. Koltun, "Colored point cloud registration revisited," in *Proceedings of the IEEE international conference on computer vision*, pp. 143–152, 2017.

[34] Z. Lin, Z. Gao, B. M. Chen, J. Chen, and C. Li, "Accurate lidar-camera fused odometry and rgb-colored mapping," *IEEE Robotics and Automation Letters*, 2024.

[35] 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 *CVPR*, 2018.

[36] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[37] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of rgb-d slam systems," in *Proc. of the International Conference on Intelligent Robot Systems (IROS)*, Oct. 2012.