iBTC: An Image-assisting Binary and Triangle Combined Descriptor for Place Recognition by Fusing LiDAR and Camera Measurements

Zuhao Zou[1], Chunran Zheng[1], Chongjian Yuan[1], Shunbo Zhou[2], Kaiwen Xue[2], Fu Zhang[1]

***Abstract*—In this work, we introduce a novel multimodal descriptor, the image-assisting binary and triangle combined (iBTC) descriptor, which fuses LiDAR (Light Detection and Ranging) and camera measurements for 3D place recognition. The inherent invariance of a triangle to rigid transformations inspires us to design triangle-based descriptors. We first extract distinct 3D key points from both LiDAR and camera measurements and organize them into triplets to form triangles. By utilizing the lengths of the sides of these triangles, we can create triangle descriptors, enabling the rapid retrieval of similar triangles from a database. By encoding the geometric and visual details at the triangle vertices into binary descriptors, we augment the triangle descriptors with richer local information. This enrichment process empowers our descriptors to reject mis-matched triangle pairs. Consequently, the remaining matched triangle pairs yield accurate loop closure place indices and relative poses.**

In our experiments, we conduct a thorough comparison of our proposed method with several SOTA methods across public and self-collected datasets. The results demonstrate that our method exhibits superior performance in place recognition and overcomes the limitations associated with the unimodal methods like BTC, RING++, ORB-DBoW2, and NetVLAD. Additionally, we perform a time cost benchmark experiment and the result indicates that our method's time consumption is reasonable, compared with baseline methods.

***Index Terms*—place recognition, loop detection, multimodal, descriptor, LiDAR SLAM, visual SLAM.**

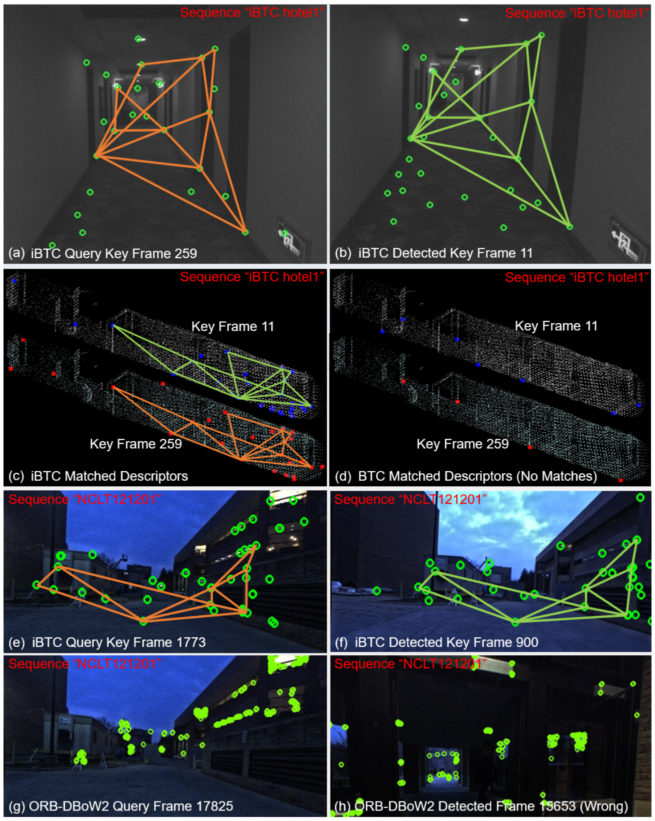
Sequence "iBTC hotel1"
(a) iBTC Query Key Frame 259
Sequence "iBTC hotel1"
(b) iBTC Detected Key Frame 11
Sequence "iBTC hotel1"
Key Frame 11
Key Frame 259
(c) iBTC Matched Descriptors
Sequence "iBTC hotel1"
Key Frame 11
Key Frame 259
(d) BTC Matched Descriptors (No Matches)
Sequence "NCLT121201"
(e) iBTC Query Key Frame 1773
Sequence "NCLT121201"
(f) iBTC Detected Key Frame 900
Sequence "NCLT121201"
(g) ORB-DBoW2 Query Frame 17825
Sequence "NCLT121201"
(h) ORB-DBoW2 Detected Frame 15653 (Wrong)

Fig. 1. (a)∼(h) show that iBTC overcomes the limitations of the unimodal methods like BTC [1] (LiDAR-based) and ORB-DBoW2 [2] (visual). (a)∼(d) demonstrate iBTC's success in identifying the correct loop closure in a structurally similar scene, specifically a long corridor in the "iBTC hotel1" sequence, while BTC fails. (a) and (b) depict matched triangle descriptors on query key frame 259 (orange lines) and correctly detected key frame 11 (green lines) on corresponding images, respectively. (c) confirms iBTC's successful matching of triangle descriptors presented with submap clouds, whereas (d) highlights BTC's inability, attributed to unstable key point extraction in such scenes. (e)∼(h) illustrate iBTC's accurate detection of key frame 800 for query key frame 1773 in an illumination-varying scene within the "NCLT121201" sequence. Conversely, ORB-DBoW2 produces an erroneous detection due to illumination variation. Note the representation of detected salient points by green circles in (a), (b), (e)∼(h), and detected 3D key points by red and blue dots in (c) and (d).

I. Introduction

PLACE recognition, also called loop closure detection, constitutes a fundamental technique within the realm of robotic navigation and exploration. As robotic platforms traverse diverse and expansive environments, simultaneous localization and mapping (SLAM) systems become indispensable for real-time, precise robot localization and mapping of the surroundings. Place recognition plays a pivotal role in recognizing previous place during robot exploration, thereby facilitating system relocalization and addressing drift issues inherent in SLAM systems. Moreover, it facilitates merging maps constructed across disparate robot exploration sessions into a unified map.

Given the diversity of sensors utilized in SLAM systems, encompassing cameras, LiDAR, or both, place recognition methods for different sensors have been developed. One cate-

Manuscript received: May 26, 2024; Accepted September 20, 2024. This paper was recommended for publication by Editor Sven Behnke upon evaluation of the Editor and Reviewers' comments.

This work is supported by Huawei Cloud Computing Technologies Co., Ltd. under the project 200010767, and the Grants Committee Early Career Scheme of The University of Hong Kong under the project 27202219.

[1] Zuhao Zou (email: zuhao.zou@connect.hku.hk), Chunran Zheng (email: zhengcr@connect.hku.hk), Chongjian Yuan (email: ycj1@connect.hku.hk), and Fu Zhang (email: fuzhang@hku.hk) are with the Department of Mechanical Engineering, University of Hong Kong, Pokfulam, Hong Kong. Fu Zhang is the corresponding author.

[2] Shunbo Zhou (email: zhoushunbo@huawei.com) and Kaiwen Xue (email: xuekaiwen6@huawei.com) are with Huawei Cloud Computing Technologies Co.,Ltd..

Digital Object Identifier (DOI): see top of this page.

gory of place recognition methods relies on images captured by cameras, exemplified by FAB-MAP [3], DBoW2 [2], and SeqSLAM [4]. Another category leverages scans acquired by LiDAR sensors, such as ScanContext [5], RING++ [6], and BTC [1]. However, each method category exhibits its own set of limitations. Visual (camera-based) methods are susceptible to variations in illumination (e.g., plots (e)∼(h) of Fig. 1), whereas LiDAR-based approaches struggle with environments characterized by high structural similarity, such as long corridors (e.g., plots (a)∼(d) of Fig. 1) and large plazas. Consequently, multimodal methods incorporating both LiDAR and camera sensors, such as CoRAL [7], MinkLoc++ [8], and our proposed approach, emerge to mitigate the individual limitations of unimodal methods.

As the combination of LiDARs and cameras becomes widely used in autonomous vehicles, drones, and reconstruction sensor suites, optimizing their use for robust place recognition becomes crucial. Simply combining LiDAR and visual place recognition methods without fusion leads to two different output types: LiDAR methods provide precise loop closure transformations between query and historical point clouds, while visual methods offer relatively imprecise transformations between query and historical camera poses due to limited image resolution and incorrect visual descriptor matches. By designing a fusion framework, we can integrate these measurements to produce a unified output, facilitating the place recognition usage in SLAM systems.

Despite the potential advantages offered by multimodal methods for place recognition and relocalization, several challenges persist. Firstly, it is difficult to design a descriptor that effectively exerts the potential of both sensor types and conquers the limitations of unimodal methods. Secondly, it is a challenge to ensure the efficiency and robust performance of designed descriptors across different LiDAR (solid-state or spinning), camera (RGB or gray) types, and different sensor characteristics (e.g., Field Of View (FOV), resolution, and range).

Building upon a state-of-the-art (SOTA) LiDAR-based method - BTC, we introduce the image-assisting binary triangle combined descriptor (iBTC). More specifically, BTC extracts 3D key points from LiDAR point clouds and generates triangles descriptors using triplets of key points. BTC encodes the local geometrical distribution at triangle vertices as geometrical binary descriptors. In this work - iBTC, based on BTC, we propose to also obtain 3D key points associated with salient points (locally maximum intensity change pixels) on images and form corresponding triangle descriptors. Moreover, we propose to enrich the distinctiveness of triangle descriptors by encoding image details at triangle vertices with visual binary descriptors. In summary, our iBTC descriptors consist of triangle descriptors, geometrical binary descriptors, and visual binary descriptors.

Our method ensures that triangles extracted from both LiDAR scans and images seamlessly collaborate in a unified workflow. Moreover, our approach exerts the potential of both sensor types and overcomes their individual limitations for place recognition. The main contributions of this work are summarized as follows.

1). **iBTC descriptor**. We propose a compound descriptor, called iBTC, encoding global scene geometry, local point cloud geometry, and local image detail. As a result, in the experiment, our method performs reliably in structurally similar scenes (difficult for LiDAR-based methods) and against illumination changes (difficult for visual methods).

2). **Extensive experiments benchmarking iBTC's loop detection performance and efficiency**. We benchmark our place recognition method with self-collected and public datasets, different sensor types (solid-state and spinning LiDARs, RGB and gray cameras), different scenes (indoor and outdoor, structurally similar and illumination changing scenes), and different platforms (cars, robots, handheld platforms, and flying drones). The result shows that our method has superior performance compared with SOTA LiDAR-based (BTC [1], RING++ [6]), visual (ORB-DBow2 [2], [9], NetVLAD [10]), and multimodal (CoRAL [7] and MinkLoc++ [8]) place recognition methods. We also benchmark our method's time cost and prove that there is only a modest time cost increase for processing image information, approximately 10 milliseconds, compared to the original BTC time consumption of 13 milliseconds.

3). **Open source codes and dataset**. To share our findings with the community and make our work reproducible, we make our codes publicly available on our GitHub: github.com/hku-mars/iBTC. We also make our self-collected dataset publicly available and it is useful for peers to develop their place recognition methods against structurally similar scenes.

II. Related Works

A. Visual Place Recognition

The realm of handcrafted visual descriptors has reached maturity, with representations such as SIFT [11], SURF [12], and BRIEF [13] being prominent examples. These descriptors encode local image details surrounding 2D key points into floating-point or binary arrays, showcasing resilience against various transformations. BRIEF, particularly sought after in visual SLAM systems such as VINS-Mono [14] and ORB-SLAM [15], stands out due to its efficiency. While SIFT and SURF offer superior scale invariance, they entail significant computational overhead due to the calculation of the Difference of Gaussians (DoG) across scale levels. Despite their efficacy in capturing local image details, these descriptors inherently lack robustness against illumination changes. It's noteworthy that handcrafted visual local descriptors necessitate collaboration with other tools [2], [3] for loop closure retrieval. FAB-MAP [3] utilizes these descriptors to construct a probabilistic model and compute likelihood scores for candidate image frames. DBow2 [2] maintains an online vocabulary database to record words generated by handcrafted visual descriptors and scores candidate image frames based on a pre-loaded dictionary reflecting the uniqueness weights of appearing words. DBoW2's implementation is combined with ORB [9] by default and thus we denote it as ORB-DBoW2 in the rest of this paper standing out their combination.

B. LiDAR-based Place Recognition

Initially inspired by sophisticated handcrafted 2D local descriptors, early research on handcrafted 3D descriptors primarily focused on encoding the local 3D geometry of point clouds. This trend was partly due to the versatility and low cost of short-range ($\leq$10m) point cloud scanners like Kinect 1 and 2. As LiDAR sensors became more affordable and versatile, the research topic shifted towards global descriptors to exert the potential LiDARs' long-range measurement capabilities ($\leq$300m). Global descriptors encapsulate all geometric information into a single descriptor, giving more promising place recognition performance. For instance, M2DP [16] projects point clouds onto specific planes with fixed azimuth and elevation angles and applies singular value decomposition (SVD) on projected 2D point matrices, concatenating the SVD results into a single descriptor. Similarly, Scan Context [5] discretizes projected 2D points along the z-plane and encodes the heights of discretized bins into a descriptor. Scan Context++ [17] introduces Cartesian coordinates for space discretization to improve Scan Context's lateral invariance. It also introduces sub-descriptors and topological place retrieval to highly accelerate scan context's loop retrieval. RING++ [6] removes ground planes and projects non-plane point clouds along with z-direction to generate bird's eye view (BEV) images. It generates a roto-translation-invariant gram descriptor from BEV images by Radom and Discrete Fourier Transformation. Compared to local descriptors, global descriptors offer more reliable detection performance due to encoding more scene information into a unified descriptor. However, these methods are sensitive to sensor view pose changes due to their projection-based nature. For example, when revisiting the same place, significant changes in the robot's pitch or roll angles will lead to a significant difference of projected images along the z-direction. BTC projects non-plane point clouds to adjacent planes to extract 3D key points corresponding to maximum counts of occupied voxel projection. BTC utilizes key point triplets to generate triangles and encodes triangle side lengths as triangle descriptors. Meanwhile, it encodes the triangle vertices' local geometrical information as binary descriptors to increase distinctiveness. Compared with projection-based methods, BTC has strong sensor view pose invariance across all 6 DoF (degrees of freedom) because its projection is based on the extract normals at the locally adjacent planes and its triangle descriptors are invariant to rigid transformations. BTC can handle not only 360 degrees LiDAR scan but also narrow FOV (70 degrees) scan provided by solid-state LiDAR. A recent work, SOLiD [18], pushes the loop detection methods' FOV limit down to 60 degrees by importing an new spatial representation encoding spatial occupancy along radial and azimuth-elevation directions, re-weighted using vertical direction information. Unlike Convolution Neuron Network(CNN)-based methods [19]–[21], BTC offers better adaptability to various LiDAR types and environmental conditions. However, in structurally similar scenes, such as long corridors, LiDAR-based place recognition methods (including BTC) struggle to extract unique geometrical information and generate distinct descriptors effectively.

C. Multimodal Place Recognition

CoRAL [7] constructs an elevation image from input point clouds and enhances it with projected RGB image features, aggregated using a deep neural network-based NetVLAD [10] layer. MinkLoc++ [8] utilizes designed point cloud and RGB image feature extraction blocks to construct feature maps, which are then aggregated into a unified multimodal descriptor for loop retrieval. Compared with CoRAL and MinkLoc++, CNN-based methods, our method has better adaptability to various sensor types and environments, and it does not need a training process and GPU acceleration for real-time execution.

III. Methodology

Our method's pipeline is shown in Fig. 2. The workflow is summarized as follows. The input of our method is registered LiDAR scans and images which can provided by LiDAR-visual-inertial SLAM systems [22], [23]. Subsequently, our method extracts key points and generates iBTC descriptors. After that, iBTC descriptors are used to retrieve matched triangle descriptors and candidate place indices. Most of the wrongly matched triangle descriptors are further rejected by validating the geometrical or visual binary descriptors encoded at the triangle vertices. Validated triangle matches provide 6 DoF loop closure pose guesses which will be subsequently verified in the geometrical verification module. The best candidate is the one with the highest score in the geometrical verification, and its place index and 6 DoF pose (w.r.t. the current submap pose) will be outputted as a result. Notably, the descriptor generation of our method requires proper LiDAR-camera extrinsic calibration (i.e., no significant mismatch in LiDAR point-to-image projection validation), which can be achieved by many existing calibration methods [24]–[26].

A. Key Points Extraction From LiDAR scans

The module of key point extraction from LiDAR scans is adapted from BTC and it involves several key steps. First, consecutive registered LiDAR scans are accumulated to create a submap. This submap is voxelized, and plane voxels are identified based on the ratio of on-plane points. These plane voxels are then merged to form larger planes. Next, non-plane voxels are projected onto adjacent planes using the plane normals to create projection images. Key points are extracted from these projection images by identifying pixels with maximum projection counts of occupied voxels among local image areas. We denote key points extracted from LiDAR scans as $\mathbf{P^g}$. We do not provide all the details of this module here, and readers are encouraged to refer to BTC [1].

B. Salient Visual Points Extraction

During accumulating registered LiDAR scans, M registered images are received (the corresponding camera poses estimated from LiDAR-visual-inertial SLAM systems). For each registered image, we need to extract salient visual points using the image intensity gradient. Specifically, the first step is to

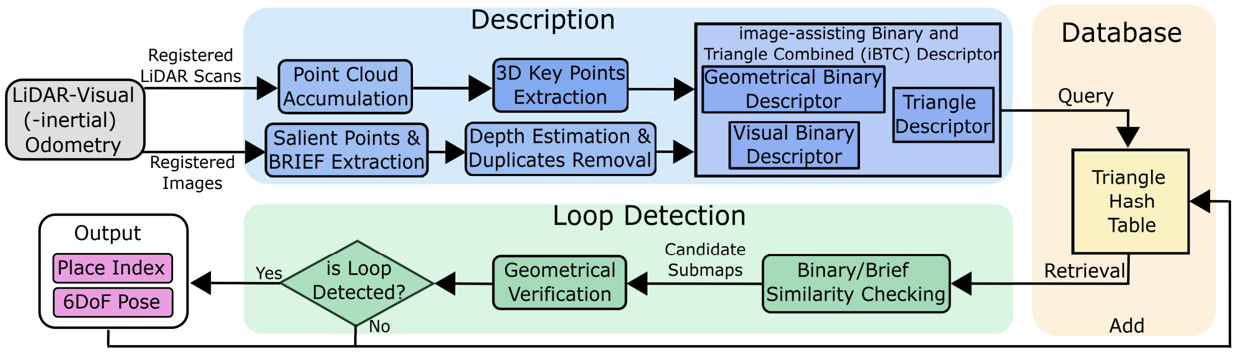

Fig. 2. iBTC pipeline.

check if the image requires downsampling by computing a downsampling level as follows:

$$s = \text{floor}(\frac{w+h}{1000}) \tag{1}$$

Where w and h are image width and height. If $s > 0$, the image is down-sampled by a scale of 2^s and this step is important to ensure the real-time execution of salient points extraction module. The second step is to compute the image intensity gradient matrix and HARRIS response matrix. The HARRIS corner response function [27] is defined as follows:

$$R = a * c - b^2 - k * (a+c)^2 \tag{2}$$

Where $a = dx * dx$, $b = dx * dy$, and $c = dy * dy$, dx and dy denote Gaussian smoothed intensity gradient along horizontal and vertical directions respectively, k is an empirical parameter. Subsequently, we extract hundreds of raw visual points that have R values higher than a certain threshold for the image. We discretize the image space into certain grids and force each grid to impose no more than one salient point for efficiency and reliability purposes. As a result, the number of salient points decreases from hundreds to dozens for each image. Finally, we scale up the image coordinates of salient points in the down-sampled image by a scale of 2^s to roughly recover their coordinates in the original image. The local patches centering at the recovered coordinate are cropped, and we search the highest neighbouring R values within the patches. We replace the recovered coordinate with the image coordinate of the highest R value if they are not the same. In this way, we avoid computing the gradient matrix and R value matrix in the full resolution for salient point extraction, significantly improving efficiency. After that, we extract BRIEF descriptors for the extracted salient points.

C. Depth Estimation and Duplicate Removal

With the extracted salient points, we need to find their 3D positions in the submap coordinate. A salient point may be observed and extracted in multiple images of the current key frame, and thus we also need to further merge them using visual and spatial information to remove the duplicates. In detail, for each image, we project the down-sampled submap cloud to the registered image plane to generate a depth matrix. We search the salient point's depth within a local square area centering at each salient point. The depth value is selected as the one closest to the salient point. In this way, the 3d positions of salient points (denoted as $\mathbf{P^v}$) on all M images are found. We iterate over all $\mathbf{P^v}$ and use KD-tree to search near points with a range of 0.1 m. If the near point has a similar BRIEF (similarity $\leq$ 0.95 by default) to that of the query point, we increment the query point's supporter counter. Finally, we sort $\mathbf{P^v}$ according to their supporter counters in descending order and only keep the top 30 of $\mathbf{P^v}$ for efficiency.

D. iBTC Descriptor Generation

The full iBTC is designed as a triangle descriptor combined with three geometrical or visual binary descriptors at the triangle vertices. Each triplet of extracted key points ($\mathbf{P^g}$ or $\mathbf{P^v}$) forms a triangle. The triangle descriptor consists of three lengths representing the ascending edges of the triangle, denoted as $\mathbf{L} = [\mathrm{l}_1, \mathrm{l}_2, \mathrm{l}_3]$ $(\mathrm{l}_1 \leq \mathrm{l}_2 \leq \mathrm{l}_3)$.

The geometrical binary descriptor encodes the distribution of occupied voxel along the plane normals (designed by BTC [1]). The point cloud above the extracted 3D key points (see III-A) are divided into certain layers. Starting from the bottom layer, if a layer is occupied by points, the corresponding bit of the geometrical binary descriptor is set to one, otherwise zero (see Fig. 3).

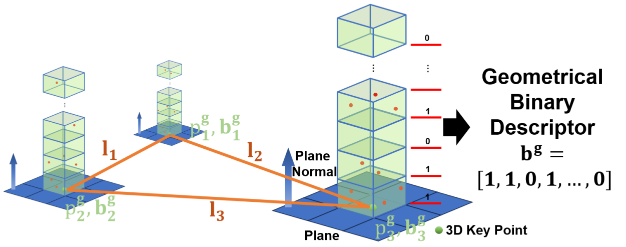
l_1
l_2
l_3
$\mathrm{p}^g_1, \mathrm{b}^g_1$
$\mathrm{p}^g_2, \mathrm{b}^g_2$
$\mathrm{p}^g_3, \mathrm{b}^g_3$
Plane Normal
Plane
0
1
0
1
1
Geometrical Binary Descriptor
$\mathbf{b^g} = [\mathbf{1}, \mathbf{1}, \mathbf{0}, \mathbf{1}, \ldots, \mathbf{0}]$
● 3D Key Point

Fig. 3. A triangle descriptor combined with geometrical binary descriptors.

The binary descriptor encodes the local image details. We employ the OpenCV [28] implementation of BRIEF generation to encode the image patches centering at the extracted salient points (see Fig. 4). We do not explain BRIEF generation process here due to limited space, and readers are encouraged to refer to BRIEF [13]. The visual binary (i.e., BRIEF) descriptor is set to 256 dimensions. Considering a salient point may involve multiple iBTC descriptors generation of current key frame, if each iBTC descriptor has one copy of the salient point's visual binary descriptor, there will exist extensive duplicate high dimensional descriptor copies causing the program to run out of memory quickly. Instead, we store all extracted visual binary descriptors in a vector and just store their corresponding vector indices in the iBTC descriptors.

Remark 1: We can also mix visual and LiDAR key points for triangle generation, significantly increasing the number of triangle descriptors. This method is beneficial when visual and LiDAR key points are sparse. However, the extraction accuracy levels of the two types of key points are significantly different, leading to inconsistencies in the edge length accuracy of the resultant mixed-type triangles. Finding reliable parameters for mixed-type triangles, such as quantization resolution for Hash table generation and triangle side length comparison thresholds, is still an open problem. In this work, we do not employ mixed-type triangles.

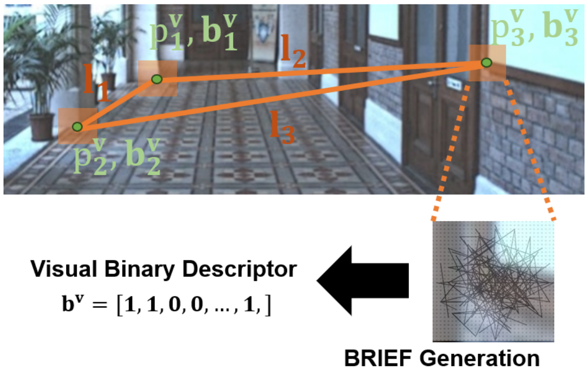

Fig. 4. A triangle descriptor combined with visual binary descriptors.

E. Database Maintenance

In the current key frame, we insert all generated iBTC descriptors into a database (a Hash table) using the Hash key generated from triangle side lengths $\mathbf{L}$. We first quantize $\mathbf{L}$ as follows:

$$\bar{\mathrm{l}}_1 = \frac{\mathrm{round}(\mathrm{l}_1)}{\Delta l}, \bar{\mathrm{l}}_2 = \frac{\mathrm{round}(\mathrm{l}_2)}{\Delta l}, \bar{\mathrm{l}}_3 = \frac{\mathrm{round}(\mathrm{l}_3)}{\Delta l} \quad (3)$$

Where Δl is a fixed resolution. Using quantized triangle side lengths $\bar{\mathrm{l}}_1$, $\bar{\mathrm{l}}_2$, and $\bar{\mathrm{l}}_3$, we generate the Hash key for the iBTC descriptor.

$$\mathrm{Hash}(\mathbf{L}) = \mathrm{Mod}(\mathrm{Mod}((\bar{\mathrm{l}}_3 * p + \bar{\mathrm{l}}_2) * p, B) + p, B) \quad (4)$$

Where p is a large prime number to alleviate Hash collisions, and B is a maximum value set to prevent out-of-bounds indices.

The advantage of employing a hash key and hash table lies in their ability to offer constant time complexity $\mathcal{O}(n)$ for both inserting and retrieving n descriptors in the new submap. This efficiency contrasts sharply with tree-based databases, such as KD-trees, which necessitate frequent tree rebalancing for insertions. The time required for tree rebalancing is linearly proportional to the database size, making hash tables a more time-efficient choice for managing large sets of descriptors.

F. Loop Detection

Given the current iBTC descriptors, we need to first find the top K most voted candidates (previous key frame indices). Specifically, we initialize a zeros vote vector of the size of existing key frames and iterate over current iBTC descriptors. If the Hash value of an iBTC descriptor's triangle $\mathrm{Hash}(\mathbf{L})$ successfully locates its position in the database, we will iterative over all stored iBTC descriptors at the position and increment the vote of the matched key frame by one. At the same time, store the matched iBTC descriptor pairs. After all current iBTC descriptors are iterated over, we only keep the top K most voted candidates and corresponding matched iBTC descriptor pairs.

There exist many wrongly matched iBTC descriptor pairs requiring further validation. We validate the descriptor pairs by comparing their geometrical or visual binary descriptors. If the binary descriptors of a pair of matched iBTC descriptors are not the same type, this pair will be rejected. If the same type and the average similarity between three binary descriptors is larger than a fixed similarity threshold (0.7 for both geometrical and visual binary descriptors by default), this pair will be accepted.

A pair of matched triangles naturally can provide a rough transformation that aligns the query submap to the candidate submap ${}^C_Q\mathbf{T}_r \in SE(3)$ which brings our iBTC descriptor the property of sensor pose invariance (mathematical detail is provided in [1]). Among all estimated ${}^C_Q\mathbf{T}_r$ using matched triangles, we find out the most accurate one by employing RANSAC [29] to find the transformation with the maximum number of correctly matched vertices. At the final stage, we examine ${}^C_Q\mathbf{T}_r$ with plane-plane overlap ratio between transformed query submap using ${}^C_Q\mathbf{T}_r$ and candidate submap. We find out how many planes are overlapping by checking through all planes in the transformed query submap to see if they have near (a fixed radius) and similar normal direction (a fixed normal difference threshold) planes in the candidate submap. The overlap ratio is defined as the ratio between the overlapping plane number and the overall plane number of the query submap. If one of the candidate frames scores an overlap ratio larger than a pre-defined threshold δ_p, its place index (i.e., frame index) and validated transformation ${}^C_Q\mathbf{T}$ will be the output result. If multiples satisfy the condition, the place index and ${}^C_Q\mathbf{T}$ of the one with the highest overlap ratio is output as the result.

IV. Experiment

In this section, we benchmark our method's detection performance and efficiency across a range of public datasets such as Newer College [30], NCLT [31], MARS-LVIG Dataset [32], and KITTI [33], as well as our self-collected dataset. It is worth noting that these datasets have very different sensor platform setups as shown in Tab. I.

The requisite inputs of our method are the undistorted LiDAR scans and images, and the estimated poses of LiDAR and camera by LiDAR-visual-inertial SLAM systems (e.g., R3LIVE and FAST-LVIO [22], [23]). Our experiment platform is a 2.90 GHz 16-cores Intel i7-10700 CPU and 15.5 GB RAM.

A. Benchmarking Against Unimodal Methods

To evaluate our method, we change the overlap threshold δ_p from 0.05 to 1 with an increment of 0.05 for each

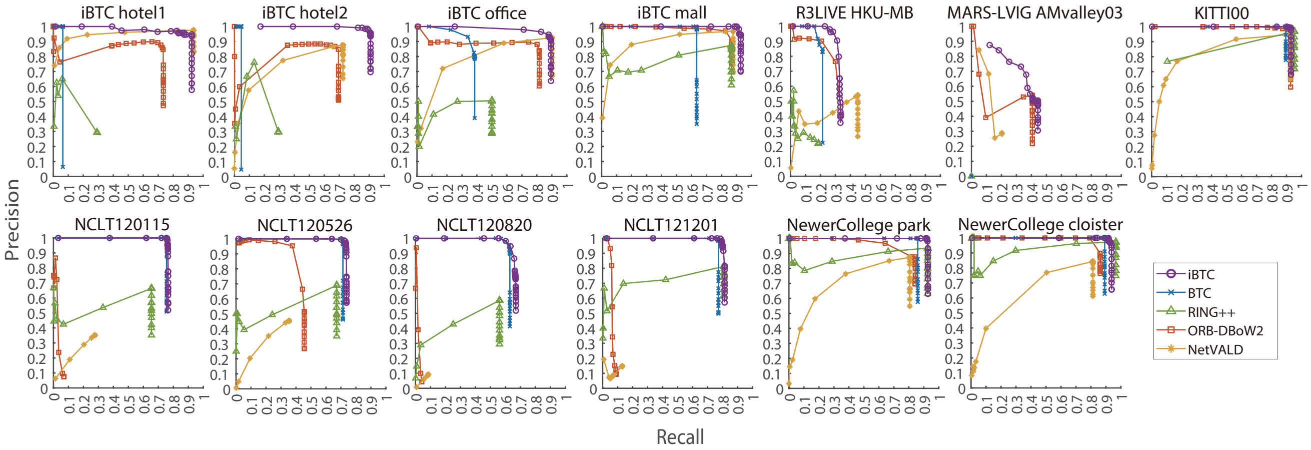

Fig. 5. Precision-Recall curve benchmark result.

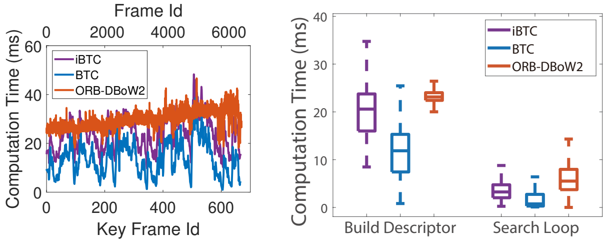

Fig. 6. Time benchmark result on sequence "iBTC mall". The left plot shows the overall time cost trends of methods against key frame ID or frame ID. The right plot shows separate time costs of methods for building descriptor and searching loop closure.

TABLE I
DATASET SENSOR PLATFORM SETUPS

Name	Camera Detail	LiDAR Detail	Platform Detail
iBTC & R3LIVE	RGB	Solid-state	Handheld platform
MARS-LVIG	RGB	Solid-state	Flying drone
NewerCollege	Gray (left)	64 lines spinning	Handheld platform
NCLT	RGB (front)	16 lines spinning	Mobile robot
KITTI	Gray (cam0)	64 lines spinning	Car

sequence to generate the Precision-Recall (PR) curve. We benchmark our method with BTC, RING++[1], ORB-DBoW2[2], and NetVLAD[3], which are SOTA LiDAR-based and visual loop detection methods. The result is shown in Fig. 5 and it shows that our method outperforms the other four methods in 13 sequences. Sequence information is shown as Tab. II. BTC's performance degenerates severely in structurally similar scenes (e.g. the corridors in "iBTC hotel1", "hotel2", "office", and "R3LIVE HKU-MB"; the valley in MARS-LVIG AMvalley03). The scenes lack local geometrical distribution variation, making it inherently challenging for BTC to robustly extract key points (see Fig. 1) and to distinguish triangles with similar geometrical binary descriptors. This also poses challenges for RING++ in robustly extracting distinct descriptors. In contrast, our method employs image information in key point extraction, descriptor generation, and triangle validation, leading to better performance in these scenes. In the four NCLT sequences, the illumination variation make the visual loop detection methods ORB-DBoW2 and NetVLAD struggle to detect loop closure robustly (see Fig. 1). Relying on not only visual measurement but also LiDAR measurement, our method has robust performance against illumination.

Additionally, we use the Recall@1 metric, defined as the ratio of the top 1 true positive number to the total ground truth positive number (also employed in [6]), to benchmark our method against others. This metric differs from the PR-curve as it focuses on recall aspect: the number of loops correctly detected when the robot revisits historical places. The results, shown in Tab. III, indicate that our method outperforms or is very close to the best-performing methods in 13 sequences, except R3LIVE HKU-MB. In this sequence, our method's Recall@1 is lower than NetVLAD. However, according to the PR curve plot (Fig. 5), our method offers a better precision-recall trade-off in this sequence.

TABLE II
DATASET SEQUENCES INFORMATION

Sequence Name	Indoor or Outdoor	Structurally Similar Scene	Illumination Varying	Self-collected or Public Dataset
iBTC hotel1	Indoor	Yes	No	Self-collected
iBTC hotel2	Indoor	Yes	No	Self-collected
iBTC office	Indoor	Yes	No	Self-collected
iBTC mall	Both	No	No	Self-collected
R3LIVE HKU-MB	Both	Yes	Yes	Public
MARS-LVIG AMvalley03	Outdoor	Yes	No	Public
NewerCollege park	Outdoor	No	No	Public
NewerCollege cloister	Both	No	No	Public
NCLT120115	Both	No	Yes	Public
NCLT120526	Both	No	Yes	Public
NCLT120820	Both	No	Yes	Public
NCLT121201	Both	No	Yes	Public

For a fair comparison, for the same type of LiDAR, we employ the same configuration file to test iBTC. Meanwhile, BTC uses the same configuration files as iBTC. However, if the camera type varies, we will alter the camera-relevant parameters for iBTC, i.e., image size and camera intrinsic parameters. In addition, we directly use the policy mentioned in BTC experiment context to generate ground truth loop closure information. The policy is based on the overlap percentages

[1] retrieved from https://github.com/lus6-Jenny/RING on Aug. 2024. The provided "evaluate.py" script serves as our base. We modify it to use the same local submap accumulation policy and point cloud downsampling rate as iBTC and BTC. The PR curve for RING++ is generated by altering the correlation score between the query and matched TIRING vectors.

[2] retrieved from https://github.com/dorian3d/DBoW2 on Mar. 2024. The provided "demo.cpp" is based on ORB. We modify it to load ORBSLAM's ORB vocabulary in advance and skip recent frames for loop detection. ORB-DBoW2's PR curve is generated by altering the threshold of QueryResults.begin().Score.

[3] retrieved from https://www.di.ens.fr/willow/research/netvlad/ on Aug. 2024. The provided "demo.m" is our base script. We modify it to load the pre-trained network called "Off-the-shelf on Pitts30k + AlexNet + NetVLAD" in advance and skip recent frames for loop detection. NetVLAD's PR curve is generated by altering the threshold of euclidean distance between query and matched NetVLAD feature vectors.

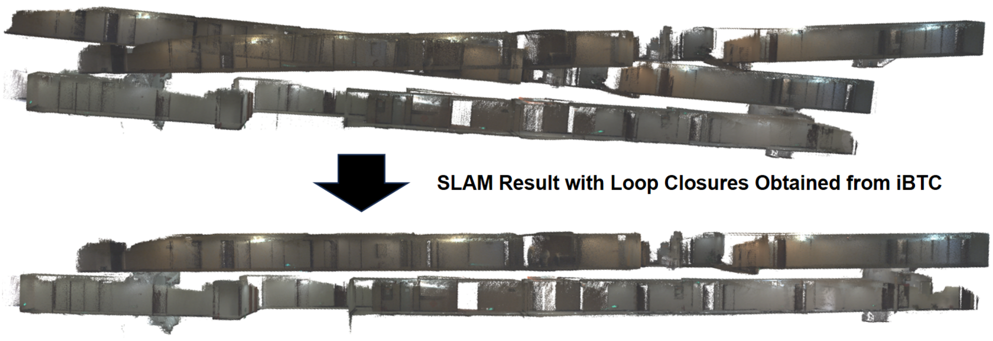

Fig. 7. SLAM result with loop closures obtained from iBTC on sequence "iBTC hotel1".

TABLE III
RECALL@1 BENCHMARK RESULT

Sequence Name	iBTC	BTC	ORB-DBoW2	NetVLAD	RING++
iBTC hotel1	92.4	6.4	73.4	**93.6**	29.2
iBTC hotel2	**90.9**	4.6	69.6	72.5	29.4
iBTC office	89.4	38.5	81.6	**90.14**	50.0
iBTC mall	**92.8**	63.4	86.8	87.7	86.6
R3LIVE HKU-MB	33.3	21.4	34.1	**44.9**	18.6
MARS-LVIG AMvalley03	**44.4**	0.0	40.6	20.3	0.0
NewerCollege park	**98.6**	95.3	84.2	80.6	92.5
NewerCollege cloister	93.8	89.2	86.2	81.3	**96.6**
NCLT120115	**76.5**	75.5	7.0	27.4	65.4
NCLT120526	**73.7**	71.4	45.5	35.4	67.2
NCLT120820	**66.9**	63.0	4.2	8.5	56.1
NCLT121201	**81.4**	77.3	9.0	13.0	80.9
KITTI00	93.0	89.5	92.8	92.6	**95.2**

between ground truth submap point clouds registered in the global coordinate frame. For the generation policy details, readers can refer to BTC's publication [1]. Unlike LiDAR-based or hybrid methods, in the visual method, we only consider the point cloud within the camera's FOV for ground truth generation.

B. Benchmarking Against Multimodal Methods

We benchmark our method with two SOTA multimodal methods based on convolutional neural networks - CoRAL and MinkLoc++. For a fair comparison, rather than self-adapting and self-training multimodal methods for the sequences used in the PR curve benchmark experiment, we directly compare our result against the stated result in CoRAL's and MinkLoc++'s publication [7], [8] on sequence KITTI00. We use only the KITTI00 sequence, following the practice of CoRAL and MinkLoc++ [7], [8], as it revisits the same locations repeatedly, unlike other KITTI sequences. We directly use the stated ground truth generation policy in [7], [8] to generate ground truth and compute our method's average recall within top K candidates ($K = 1\%$ of total key frame number). The metric is also called AR@1%. The result (see Tab. IV) shows that our method outperforms other multimodal methods as well as BTC in the KITTI00 sequence with AR@1% metric.

TABLE IV
AR@1% ON KITTI00

	Modality	**AR@1%**
CoRAL	LiDAR+Camera	76.4
MinkLoc++	LiDAR+Camera	82.1
BTC	LiDAR	83.8
iBTC	**LiDAR+Camera**	**91.6**

C. Time Consumption

To evaluate the efficiency of our method, we record the time costs of iBTC, BTC, and ORB-DBoW2 for building descriptors and searching loop closure in "iBTC mall" sequence. The record time costs are plotted against each key frame or each frame and also plotted using a box plot (see Fig. 6). These results show that, compared with BTC's overall time cost, our method increases the overall time cost due to the processing of visual information. The introduced time cost increment for processing visual information is approximately 10 milliseconds, which is reasonable compared with the overall time cost of BTC, about 13 milliseconds. Furthermore, the overall time cost of our method is smaller than that of ORB-DBoW2.

D. Application in SLAM systems

Our method allows SLAM systems to construct the map of challenge scenes for LiDAR-based methods. An example map is shown in Fig. 7. This example map is generated using FAST-LVIO [23] and iBTC on "iBTC hotel1" sequence. This sequence is structurally similar scene where LiDAR-based methods like BTC and Scan Context struggle to provide reliable loop closures for SLAM system.

Furthermore, we assess the map consistency difference between the SLAM results (see Fig. 7) with and without loop closure using iBTC, employing a conventional approach [34]–[36]. Specifically, we voxelize the constructed maps and count the voxel numbers. The resultant voxel numbers of the SLAM results with and without loop closure using iBTC are 2,280,174 and 3,292,374, respectively. The smaller voxel number indicates that loop closure using iBTC significantly improves the map consistency of SLAM results.

V. CONCLUSION

This paper proposes iBTC, a robust and efficient multimodal 3D place recognition method. iBTC is designed as a triangle descriptor combined with three geometrical or visual binary descriptors at the triangle vertices. In this way, iBTC is encoded with global scene geometry, local point cloud geometry, and local image detail. In the experiment, iBTC shows that it overcomes the limitations of the unimodal methods and it outperforms other multimodal methods. Additionally, we perform a time cost benchmark experiment to prove that iBTC's time consumption is reasonable.

Future improvements for iBTC include several planned enhancements. First, mixing visual and LiDAR key points for triangle generation to increase robustness when both key points are sparse. Second, using point cloud normals to warp local image patches, thereby increasing the viewpoint invariance of the visual binary descriptors.

REFERENCES

[1] C. Yuan, J. Lin, Z. Liu, H. Wei, X. Hong, and F. Zhang, "Btc: A binary and triangle combined descriptor for 3-d place recognition," *IEEE Transactions on Robotics*, vol. 40, pp. 1580–1599, 2024.

[2] 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.

[3] M. Cummins and P. Newman, "Fab-map: Probabilistic localization and mapping in the space of appearance," *The International journal of robotics research*, vol. 27, no. 6, pp. 647–665, 2008.

[4] M. J. Milford and G. F. Wyeth, "Seqslam: Visual route-based navigation for sunny summer days and stormy winter nights," in *2012 IEEE international conference on robotics and automation*. IEEE, 2012, pp. 1643–1649.

[5] 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*, Madrid, Oct. 2018.

[6] X. Xu, S. Lu, J. Wu, H. Lu, Q. Zhu, Y. Liao, R. Xiong, and Y. Wang, "Ring++: Roto-translation-invariant gram for global localization on a sparse scan map," *IEEE Transactions on Robotics*, 2023.

[7] Y. Pan, X. Xu, W. Li, Y. Cui, Y. Wang, and R. Xiong, "Coral: Colored structural representation for bi-modal place recognition," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 2084–2091.

[8] 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)*. IEEE, 2021, pp. 1–8.

[9] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *2011 International conference on computer vision*. Ieee, 2011, pp. 2564–2571.

[10] 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*, 2016, pp. 5297–5307.

[11] P. Scovanner, S. Ali, and M. Shah, "A 3-dimensional sift descriptor and its application to action recognition," in *Proceedings of the 15th ACM international conference on Multimedia*, 2007, pp. 357–360.

[12] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (surf)," *Computer vision and image understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[13] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "Brief: Binary robust independent elementary features," in *Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part IV 11*. Springer, 2010, pp. 778–792.

[14] 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.

[15] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "Orb-slam: a versatile and accurate monocular slam system," *IEEE transactions on robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[16] L. He, X. Wang, and H. Zhang, "M2dp: A novel 3d point cloud descriptor and its application in loop closure detection," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 231–237.

[17] G. Kim, S. Choi, and A. Kim, "Scan context++: Structural place recognition robust to rotation and lateral variations in urban environments," *IEEE Transactions on Robotics*, 2021.

[18] H. Kim, J. Choi, T. Sim, G. Kim, and Y. Cho, "Narrowing your fov with solid: Spatially organized and lightweight global descriptor for fov-constrained lidar place recognition," *IEEE Robotics and Automation Letters*, vol. 9, no. 11, pp. 9645–9652, 2024.

[19] D. Cattaneo, M. Vaghi, and A. Valada, "Lcdnet: Deep loop closure detection and point cloud registration for lidar slam," *IEEE Transactions on Robotics*, vol. 38, no. 4, pp. 2074–2093, 2022.

[20] L. Hui, M. Cheng, J. Xie, J. Yang, and M.-M. Cheng, "Efficient 3d point cloud feature learning for large-scale place recognition," *IEEE Transactions on Image Processing*, vol. 31, pp. 1258–1270, 2022.

[21] C. Shi, X. Chen, J. Xiao, B. Dai, and H. Lu, "Fast and accurate deep loop closing and relocalization for reliable lidar slam," *IEEE Transactions on Robotics*, vol. 40, pp. 2620–2640, 2024.

[22] J. Lin and F. Zhang, "R 3 live: A robust, real-time, rgb-colored, lidar-inertial-visual tightly-coupled state estimation and mapping package," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 10 672–10 678.

[23] C. Zheng, Q. Zhu, W. Xu, X. Liu, Q. Guo, and F. Zhang, "Fast-livo: Fast and tightly-coupled sparse-direct lidar-inertial-visual odometry," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 4003–4009.

[24] C. Yuan, X. Liu, X. Hong, and F. Zhang, "Pixel-level extrinsic self calibration of high resolution lidar and camera in targetless environments," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 7517–7524, 2021.

[25] L. Zhou, Z. Li, and M. Kaess, "Automatic extrinsic calibration of a camera and a 3d lidar using line and plane correspondences," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 5562–5569.

[26] D. Tsai, S. Worrall, M. Shan, A. Lohr, and E. Nebot, "Optimising the selection of samples for robust lidar camera calibration," in *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, 2021, pp. 2631–2638.

[27] C. Harris, M. Stephens, *et al.*, "A combined corner and edge detector," in *Alvey vision conference*, vol. 15, no. 50. Citeseer, 1988, pp. 10–5244.

[28] G. Bradski and A. Kaehler, *Learning OpenCV: Computer vision with the OpenCV library*. "O'Reilly Media, Inc.", 2008.

[29] P. J. Huber, "Robust regression: asymptotics, conjectures and monte carlo," *The annals of statistics*, pp. 799–821, 1973.

[30] M. Ramezani, Y. Wang, M. Camurri, D. Wisth, M. Mattamala, and M. Fallon, "The newer college dataset: Handheld lidar, inertial and vision with ground truth," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 4353–4360.

[31] N. Carlevaris-Bianco, A. K. Ushani, and R. M. Eustice, "University of michigan north campus long-term vision and lidar dataset," *The International Journal of Robotics Research*, vol. 35, no. 9, pp. 1023–1035, 2016.

[32] H. Li, Y. Zou, N. Chen, J. Lin, X. Liu, W. Xu, C. Zheng, R. Li, D. He, F. Kong, *et al.*, "Mars-lvig dataset: A multi-sensor aerial robots slam dataset for lidar-visual-inertial-gnss fusion," *The International Journal of Robotics Research*, p. 02783649241227968, 2024.

[33] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *2012 IEEE conference on computer vision and pattern recognition*. IEEE, 2012, pp. 3354–3361.

[34] A. Filatov, A. Filatov, K. Krinkin, B. Chen, and D. Molodan, "2d slam quality evaluation methods," in *2017 21st Conference of Open Innovations Association (FRUCT)*, 2017, pp. 120–126.

[35] Z. Liu, X. Liu, and F. Zhang, "Efficient and consistent bundle adjustment on lidar point clouds," *IEEE Transactions on Robotics*, vol. 39, no. 6, pp. 4366–4386, 2023.

[36] Z. Zou, C. Yuan, W. Xu, H. Li, S. Zhou, K. Xue, and F. Zhang, "Lta-om: Long-term association lidar–imu odometry and mapping," *Journal of Field Robotics*, 2024.