Fast Structural Representation and Structure-aware Loop Closing for Visual SLAM

Shuxiang Xie[1,2], Ryoichi Ishikawa[1], Ken Sakurada[2], Masaki Onishi[2] and Takeshi Oishi[1]

***Abstract*— Perceptual Aliasing is one of the main problems in simultaneous localization and mapping (SLAM). Wrong associations between different places may lead to failure of the whole map. Research on structure information is rarely investigated among existing solutions to this problem. In cases of visual SLAM without sensors, such as LiDAR or Inertial Measurement Unit (IMU), structure information can rarely be obtained due to the sparsity of 3D points, which also makes structure analysis complex. This study provides a spherical harmonics (SH) based fast structural representation (SH-FS) in visual SLAM using sparse point clouds, which extracts the structure information from sparse points into single vector. SH-FS was applied in conventional feature-based loop closing process. Furthermore, a structure-aware loop closing method in visual SLAM was proposed to improve the robustness of SLAM systems. Moreover, our methods show a favorable performance in extensive experiments on different large-scale real world datasets.**

I. Introduction

Simultaneous localization and mapping (SLAM) is a fundamental process that can be applied to several autonomous systems, where loop closing is used to correct the accumulation error during mapping process. However, loop closing might cause the problem of *perceptual aliasing*, which remains critical and makes SLAM fail.

Perceptual aliasing can be frequently observed in visual SLAM cases. Perceptual aliasing describes the phenomenon that different places generate similar visual/structural (or, in general, perceptual) footprints [1], such as staircases on different floors. The false association between different places will further cause a collapse in map structure and potentially affect future mapping process by the loop closing process, which is meant to correct the accumulation error in SLAM. Given SLAM has been widely adopted in several safety-critical applications, this kind of structure error that may cause safety problems should be avoided.

Many efforts have been made towards solving the problem of perceptual aliasing. Various latest open-source SLAM systems [2], [3], [4] generally try to solve it by carefully accepting loop closure candidates. Other existing methods, such as robust pose graph optimization [5], [6], [7], try to solve it by reducing the impact of residual errors from outliers. However, both approaches face some limitations and one of the challenges is the reliance on manually tuned parameters.

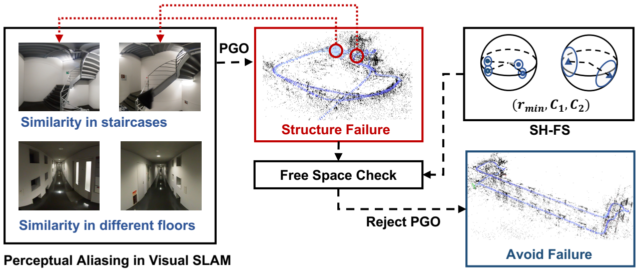

Fig. 1. Example of avoiding structure failure via free space check.

Since the problem of perceptual aliasing may lead to failure in structures, it might be possible to solve it from the perspective of structural consistency and healthiness. In order to detect and measure the structural consistency, a fast structural representation using sparse points is necessary. From our literature survey, there are no suitable methods that can abstract the structure information with sparse data.

Therefore, we propose the use of a novel spherical harmonics based fast structural representation (SH-FS), and apply it to give a structure-aware loop closing process to increase the robustness of SLAM system. Figure 1 shows a general picture of this process. Our method does not highly depend on parameter tuning and can easily be implemented in pose graph based SLAM systems.

Our contributions can be summarized as the following three points.

- Provision of SH-FS, which can extract structure information from sparse 3D points by considering the free space.
- Proposal of structure-aware loop closure detection by combining SH-FS with conventional feature-based detection methods.
- Proposal of the fast and robust loop closing failure detection method. This process does not highly rely on any manual setting and yields high robustness on different large-scale datasets.

II. Related Works

This section gives a brief review on the map and structural representation in SLAM, loop closing and perceptual aliasing, and robust pose graph optimization.

A. Structure in SLAM

Dense point clouds may describe the structure in an accurate way. Through further alignment using algorithms, such as ICP, the structure can be easily reconstructed [8].

[1] The authors are with The Institute of Industrial Science, The University of Tokyo, Japan. Emails: {shxxie,ishikawa,oishi}@cvl.iis.u-tokyo.ac.jp

[2] The authors are with The National Institute of Advanced Industrial Science and Technology (AIST), Tokyo, Japan. Emails: {k.sakurada,onishi-masaki}@aist.go.jp

Some methods also can create meshes from the feature points detected [9], [10]. However, additional data from IMU is required. As a result, constructing meshes based on landmarks faces some difficulties in some complicated indoor scenarios for visual SLAM.

Many state-of-art visual SLAM systems involve a graph of camera poses, while each pose is associated with a local 3D points cloud, which is usually sparse in visual SLAM cases. The concept of pose graph can help in map correction of global scale drift by optimizing on the basis of loop closure information. However, pose graph by itself cannot explicitly preserve the structure information. In a previous study [11], structure information was modeled by the distance in the descriptor space between nearest landmarks. Another study [12], used a virtual occupancy grid map to maintain the free space information and made it convenient in further map correction and path planning. Though this approach relies on dense point clouds alignment to perform the 3D reconstruction, the idea of free space may be a good attempt in structural representation.

Similar to the method mentioned in [11], our approach considers the relationship between landmarks and frames, however, instead of the distance in descriptor space, we study the Euclidean distance from frames to landmarks (free space information) for simplicity. Further, to reduce the calculation complexity, spherical harmonics (SH) is introduced, by which the free space information can be extracted and represented as a vector.

B. Loop Closing and Perceptual Aliasing

Some literatures have proposed several useful techniques to detect loop closure. Bag-of-words (BoW) model is widely used, which generates a vocabulary consisting of different visual words based on a pre-trained dictionary [13]. In some early studies, FAB-MAP [14], SURF [15], and SIFT [16] features were widely adopted for their high performance. Binary features, such as ORB [17], BRIEF [18], and BRISK [19] later yielded better performance in terms of the computational time. DBoW2 [20] introduces the idea of bag of binary words to recognize the place and is applied by ORB-SLAM [21] into SLAM algorithm for the first time.

Approaches on the basis of deep learning also play important roles in feature extraction and similar place recognition. Several deep-learned based local features like NetVLAD [22], SuperPoint [23], SuperGlue [24], and D2Net [25] have been extensively employed in visual SLAM and localization [26], [27], [28], [29]. Although involving deep learning techniques may greatly improve the performance of similar place recognition and loop closure detection in SLAM, repeated structures remain a significant challenge [30]. Moreover, deep-learned based methods might be slightly heavy in terms of computational power in cases where GPUs are not available.

Unfortunately, since the loop closing outlier pairs are mutually consistent, the problem of perceptual aliasing cannot be resolved by upgrading these detection algorithms. If two different places look the same, both places should be considered as loop closure candidates, for example, two identical rooms or staircases on different floors.

Most open-source SLAM systems, such as ORB-SLAM3 [3] and VINS-Mono [4] solve this problem by carefully tuning the detector and accepting the candidate. However, over cautiousness about the loop closing candidates cannot fundamentally solve perceptual aliasing problem.

C. Robust Pose Graph Optimization

Despite fine-tuning of detectors, many efforts have been made to solve perceptual aliasing. Most robust PGO algorithms attempt to reduce impacts from the outliers. Majority of studies start from modifying the cost function in PGO to make residual error scalable [5] and try to make optimizer able to adjust its performance based on the inputs. Leveraging these ideas, recent research developed cluster-based penalty scaling methods to further improve the performance [6].

Another category of robust PGO tends to select or remove outliers directly out of the optimization process based on the consistency between odometry measurement and loop closing result. [7], [31], [32]. In a previous study [1], the authors noticed that directly modeling the outliers using the co-relationships inside pose graph, instead of mitigating impacts from perceptual aliasing, can reveal the whole problem more clearly.

Nonetheless, the existing robust PGO approaches face various challenges. Some methods seem to highly depend on hyperparameter tuning, whereas different parameters selections, such as largest admissible residual, can result in totally different performance [1], [7], [31]. Some robust PGO methods rely on the complete map information and do not support incremental online SLAM process [1]. Methods related to convex relaxation are challenging in implementation, since normal optimizer cannot solve discrete problems, which makes their usage slightly inconvenient [32].

Most literatures acknowledge that the optimization result for a correct loop tends to be consistent with the odometry. However, due to the errors everywhere in tracking and mapping in visual SLAM, it is difficult to explicitly measure the global uncertainty of each frame using the existing methods [33], [34]. Thus, checking the consistency between optimized trajectory and the odometry seems really difficult in visual SLAM.

Different from robust PGO methods trying to mitigate the influence from outliers, our proposal tries to maintain the structure correctness by nullifying the false loop closure. If the loop closure is wrong, the safest solution is to neglect it and continue the tracking and mapping process. Leveraging the computational convenience of our structural representation, whether the loop closure is correct can also be easily checked.

III. Fast structure representation

In this study, we propose spherical harmonics based fast structural representation (SH-FS). Since spherical harmonics (SH) might not be a popular and familiar concept, in this section, we will explain some basics of SH briefly, followed by the details in SH-FS.

A. Spherical Harmonics (preliminary)

Spherical Harmonics (SH) form a complete set of orthogonal bases. Thus we can write any function that is defined on a spherical surface as a weighted summation of SH, which is very similar to the Fourier expansion.

SH $Y_l^m(\theta, \phi)$ are defined as

$$Y_l^m(\theta,\phi) = \frac{1}{\sqrt{2\pi}} \cdot N_l^m \cdot P_l^m(\cos\theta) \cdot e^{im\phi}, \tag{1}$$

where $P_l^m(x)$ represents the associated Legendre Polynomials, and

$$N_l^m = \sqrt{\frac{2l+1}{2}\frac{(l-|m|)!}{(l+|m|)!}}. \tag{2}$$

SH expansion can be described as, for any function $f(\theta,\phi)$ defined on surface of a sphere,

$$f(\theta,\phi) = \sum_{l=0}^{\infty}\sum_{m=-l}^{l} U_l^m Y_l^m(\theta,\phi), \tag{3}$$

where U_l^m are SH coefficients. SH coefficients can be obtained in a straightforward manner. Similar to Fourier series, we just integrate the target function $f(s)$ in the following way,

$$U_l^m = \int_S f(s) Y_l^m(s) ds. \tag{4}$$

One of the most important property of SH is the rotational invariance. For any rotation $R \in SO(3)$ applied on the function f on sphere S^2, it can be expressed by a linear transform on SH coefficients, which totally depends on R. The size of the transform depends on the orders of SH involved. Moreover, since SH bases form an orthonormal basis, the *dissimilarity* Q between two functions can be defined in a natural way. Let $\mathbf{a}$ and $\mathbf{b}$ be SH coefficients for two different functions, we can have $Q = \|\mathbf{a}-\mathbf{b}\|_2^2$ [35].

Spherical Harmonics have been applied to generate descriptors for spherical views and 3D shapes [36], [37]. Our proposed method also leverages the properties of SH as existing methods. However, instead of using omnidirectional images and 3D objects as input, we use sparse 3D points.

B. Spherical Harmonics based Fast Structural Representation

To briefly summarize, SH-FS consists of three parts:

- r_{min}, minimum radius
- $\mathbf{C}_{intra}$, free space between landmarks and camera
- $\mathbf{C}_{inter}$, free space between cameras

As stated in forementioned section, sparsity of data is the most challenging point in structure representation. We therefore, focus on the free space between landmarks and the camera. We record the landmark that is closest to the camera and denote the distance as *minimum radius* r_{min}. Then we use a sphere with radius as r_{min} to simulate the free space of this frame.

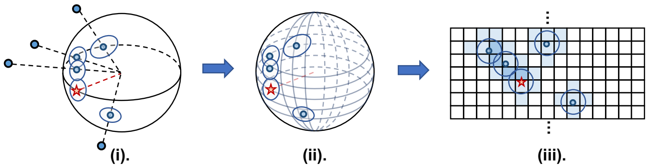

Fig. 2. (i). First we start from sparse landmarks, the closest landmark is indicated by the red star marker and r_{min} is indicated by the red dashed line. The circles represent effective range. (ii). Then we build a pixel map on the sphere surface for computational simplicity. (iii). The pixel map can be projected on a plane. Colored pixels stand for pixels that can be affected by landmarks.

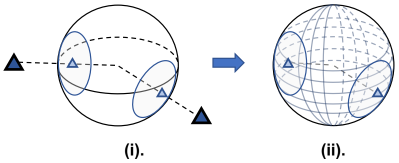

Fig. 3. We construct $\mathbf{C}_{inter}$ in a similar way as $\mathbf{C}_{intra}$. (i). Let the frame at the center be frame i, then triangles represent frame $i-1$ and $i+1$, the effective range of frame is larger than that of landmarks. (ii). We also construct $\mathbf{C}_{inter}$ on a same sized pixel map.

Another advantage of recording r_{min} is the approximation of normalized free space map. Due to sparsity, an accurate depth map is not available; however, the local distribution of landmarks with respect to the camera can also be considered as a structural footprint by considering the free space. The map is constructed by assigning each landmark a spherical Gaussian function on a spherical pixel map in the *local* coordinate system. Since Gaussian value will be close to 0 when the input is far from the center, only the surrounding range centered at the direction of the landmark will be considered, which is denoted as *effective range*.

Let the size of the pixel map be $M \times N$, and $\Xi_{(a,b)}$ $(0 \le a < M,\ 0 \le b < N)$ be the set of landmarks that can affect the (a,b)-th pixel. Define r_{ξ_i} as the distance from landmark $\xi_i \in \Xi_{(a,b)}$ to the camera. Let the parameter $\alpha_{\xi_i} = r_{\xi_i} - r_{min}$. If $|\Xi_{(a,b)}| \neq 0$, the value $p(a,b)$ on that pixel is defined as follows,

$$r_{min} + \frac{1}{|\Xi_{(a,b)}|}\sum_{\xi_i \in \Xi_{(a,b)}} G(\mathbf{v}_{\xi_\mathbf{i}}; \mu_{(a,b)}, \gamma, \alpha_{\xi_i}). \tag{5}$$

If $|\Xi_{(a,b)}| = 0$, we will have $p(a,b) = r_{min}$. For spherical Gaussian, the definition is as follows:

$$G(\mathbf{v}; \mu, \gamma, \alpha) = \alpha e^{\gamma(\mu\cdot\mathbf{v}-1)}, \tag{6}$$

where $\mathbf{v}$ and μ are both 3-dimensional unit vectors on a surface of a sphere. $\mathbf{v}$ represents the direction of the input vector and μ represents the center direction of this Gaussian. Figure 2 shows the generation of the pixel map.

Finally the whole pixel map is integrated together with SH base functions to obtain $\mathbf{C}_{intra}$. Define $\mathbf{C}_{intra}$ for frame i as $\mathbf{c}_{i,1}$,

$$\mathbf{c}_{i,1} \approx \frac{1}{r_{min}} \cdot \frac{1}{M \times N} \sum \frac{Y_l^m(a,b)}{p(a,b)} \sin(\frac{a}{M}\pi). \tag{7}$$

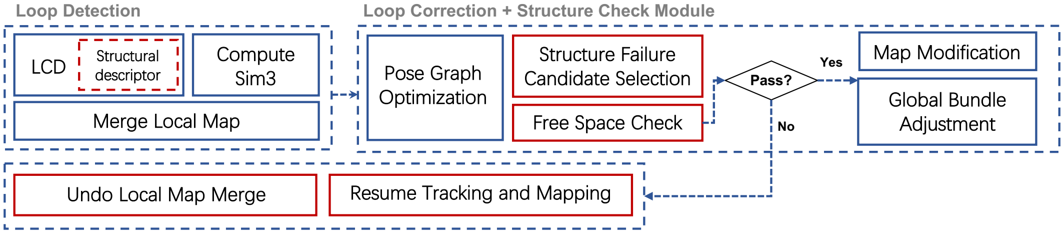

Fig. 4. Overview of the proposed loop failure detection process. Blocks marked as blue represents the original parts of conventional visual SLAM while blocks marked as red represents our proposed method.

An inverse transform is applied on the pixel map to smoothen the influence from very far landmarks.

Free space exists not only between landmarks and camera, but also between cameras. Let E be the set of frames inside the map when the loop closure happens. For the sake of simplicity, let us consider the free space of frame i between closest two frames $i-1, i+1 \in E$ in the *global* coordinate system. Let $\hat{\Xi}_{(a,b)} \subseteq \{i-1, i+1\}$ be the set of frames that can affect the (a,b)-th pixel. Let the parameter $\hat{a}_{\hat{\xi}_i} = d_{i,\hat{\xi}_i}$ which stands for the Euclidean distance between frame i and frame $\hat{\xi}_i \in \hat{\Xi}_{(a,b)}$. If $|\hat{\Xi}_{(a,b)}| \neq 0$, the value $\hat{p}(a,b)$ on that pixel is defined as,

$$r_{min} + \frac{1}{|\hat{\Xi}_{(a,b)}|} \sum_{\hat{\xi}_i \in \hat{\Xi}_{(a,b)}} G(\mathbf{v}_{\hat{\xi}_\mathbf{i}}; \hat{\mu}_{(a,b)}, \hat{\gamma}, \hat{\alpha}_{\hat{\xi}_i}). \quad (8)$$

If $|\hat{\Xi}_{(a,b)}| = 0$, we will have $\hat{p}(a,b) = r_{min}$. In the same way, we apply SH expansion to get $\mathbf{C}_{inter}$. Figure 3 shows the construction of the pixel map based on the free space between cameras.

IV. Loop Closure with SH-FS

We further explain the application of SH-FS in loop closure detection as well as loop failure detection in this section. Figure 4 gives an overview of the proposed method.

A. *Structural-aware Loop Closure Detection*

As stated in previous section, the local structure is represented by an SH coefficient vector $\mathbf{C}_{intra}$, say $\mathbf{c}_{i,1}$ for frame i. Then the $\mathbf{C}_{intra}$ difference between frame i and j can be written as $\|\mathbf{c}_{i,1} - \mathbf{c}_{j,1}\|_2$ on the basis of the property of SH. Therefore, the most straight forward application is to use it in loop closure detection.

Let Φ_i be the set of neighboring frames of frame i, and define the threshold $\tau_{i,1}$ of $\mathbf{C}_{intra}$ difference as follows:

$$\tau_{i,1} = \max_{\epsilon \in \Phi_i} \|\mathbf{c}_{i,1} - \mathbf{c}_{\epsilon,1}\|_2. \quad (9)$$

Further, in the loop closure detection, based on the original feature-based detector, we append one more criteria for candidate selection. For query frame i and candidate frame j, their $\mathbf{C}_{intra}$ difference should be restricted below $\tau_{i,1}$.

Generally speaking, combining structural-aware loop closure detection with conventional detection methods based on visual features implies higher standards for detection. Though methods such as BoW, can already yield good performance in most cases, involving structure information in detection will further improve the robustness and help to filter some outliers.

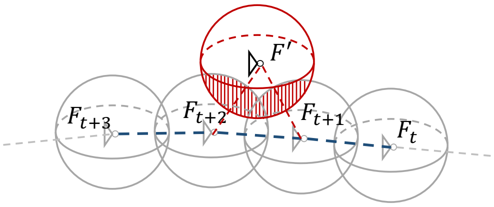

Fig. 5. Frames F_t, F_{t+1}, F_{t+2}, F_{t+3} are neighboring frames, thus it is normal for them to have overlapped parts in free space. F' is not neighboring frame of F_{t+1} and F_{t+2}, implying a violation of free space.

B. *Structural-aware Loop Failure Detection*

Consider E is the set of frames, for each frame $i \in E$, denotes the minimum radius as r_i. Furthermore, denote the euclidean distance between frames $i, j \in E$ as d_{ij}. For any frame $i \in E$, if there exists another frame $j \in E$, that makes $d_{ij} < r_i + r_j$. Meanwhile, if i and j are not adjacent to each other, then we consider i and j violates the correctness of free space. Figure 5 shows the general concept of checking free space violation.

Free space violation means part of the map is overlapping with other part and there is probably a structure failure. However, checking free space for all frames in E will be heavy in terms of computation time when N becomes large. In addition, in the case of a correct loop, checking the whole map in brute force is inefficient. Thus, we select some suspicious frames for structure failure check. By only checking these selected frames (failure candidates), the method can be greatly accelerated.

Due to the accumulated error in tracking and mapping parts of visual SLAM, global information is not reliable. Map correction process such as PGO can highly modify the map and the global information. Our concept is to measure the change before and after map correction. Note that initially $\mathbf{C}_{intra}$ is local information. In order to measure the structural change globally, we have to de-rotate $\mathbf{C}_{intra}$ back to world coordinate.

For frame i whose $\mathbf{C}_{intra}$ is $\mathbf{c}_{i,1}$, let the rotational pose of frame i be R_i. De-rotating $\mathbf{c}_{i,1}$ means casting a rotation R_i^{-1} on the spherical function described by $\mathbf{c}_{i,1}$. Thanks to

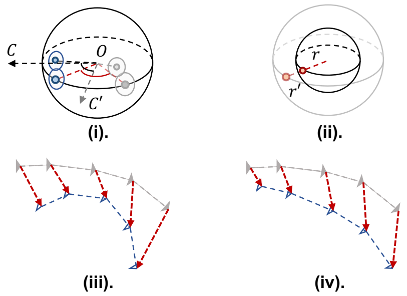

Fig. 6. (i). Structure change caused by rotation difference, OC and OC' stand for the orientation before and after PGO. (ii), Structure change in scale. (iii). Structural change caused by translation difference, it can also be considered as local structure distortion. (iv). Situation when there is no structure distortion.

TABLE I

GENERATION TIME OF SH-FS PER FRAME IN DIFFERENT MAP SETTING

	Resolution $n \times m$, effective range l, (n, m, l)			
(n, m, l)	(10,20,1)	(30,60,3)	(50,100,5)	(50,100,10)
Time [ms]	0.632	1.022	1.586	3.935

TABLE II

AVERAGE FREE SPACE CHECKING TIME IN TWO REAL-WORLD DATASETS WITH 1011 AND 795 KEYFRAMES

	Average checking time [ms]	
	correct loop	wrong loop
Ours	1.950	1.757
Brute-force	56.067	4.615

the rotational invariance property held by SH, we need not rotate landmarks and reconstruct the pixel map again. There exists a linear transform $\Lambda_{R_i^{-1}}$, which makes the

$$\mathbf{c}^d_{i,1} = \Lambda_{R_i^{-1}} \cdot \mathbf{c}_{i,1}. \tag{10}$$

$\mathbf{c}^d_{i,1}$ is the de-rotated $\mathbf{C}_{intra}$, we may denote it as $\mathbf{C}^d_{intra}$. Thus r_{min}, $\mathbf{C}^d_{intra}$, and $\mathbf{C}_{inter}$ all represents global information.

We exploit the convenience of our proposed SH-FS to check the structure change caused by pose graph optimization (PGO) given the corresponding loop closing. We consider two categories of failure candidates: (1). Loop closing pairs; (2). Largely modified frames.

Loop closing pairs In most cases, loop closing points in trajectories are the most "dangerous" positions because PGO usually modifies the loop ends to a great extent. While the existence of scale drift makes it difficult to judge the correctness of loop closure before PGO, we should make sure of the correctness of the structure of the closing points after PGO. The first part of candidates is selected as the loop closing frame, current frame, and their neighbors.

Largely modified frames As we stated in abovementioned section, a healthy and correct loop tends to be consistent with the odometry. Therefore, those frames whose rotation, translation and scale are largely modified by PGO will be considered as suspicious frames (failure candidates). Figure 6 shows these three types of structural change. The change in rotation, translation and scale can be described by the change in $\mathbf{C}^d_{intra}$, $\mathbf{C}_{inter}$, and r_{min}.

$\mathbf{C}_{intra}$ threshold $\tau_{i,1}$ derived in Section IV-A is also used here to judge the change in $\mathbf{C}^d_{intra}$ caused by PGO. Define $\Delta\mathbf{c}^d_{i,1} = \|\mathbf{c}^d_{i,1} - \tilde{\mathbf{c}}^d_{i,1}\|_2$, where $\mathbf{c}^d_{i,1}$ and $\tilde{\mathbf{c}}^d_{i,1}$ are the de-rotated $\mathbf{C}^d_{intra}$ of frame i before and after PGO. If $\Delta\mathbf{c}^d_{i,1} > \tau_{i,1}$, then frame i is included as a failure candidate.

We also define the threshold $\tau_{i,2}$ for $\mathbf{C}_{inter}$ of frame i similarly, let Φ_i be the set of neighboring frames of frame i,

$$\tau_{i,2} = \max_{\epsilon\in\Phi_i} \|\mathbf{c}_{i,2} - \mathbf{c}_{\epsilon,2}\|_2. \tag{11}$$

Define $\Delta\mathbf{c}_{i,2} = \|\mathbf{c}_{i,2} - \tilde{\mathbf{c}}_{i,2}\|_2$, if $\Delta\mathbf{c}_{i,2}$ exceeds $\tau_{i,2}$, frame i is also included as a failure candidate.

Another reason for using 7-dof PGO is that the change of r_{min} can easily be reflected by the change in scale s_i. Following the setting in [38], scale s before PGO is initialized as 1, so we set the threshold $\tau_{i,r}$ of frame i as,

$$\tau_{i,r} = \max_{\epsilon\in\Phi_i} |\frac{r_i}{s_i} - \frac{r_\epsilon}{s_\epsilon}|, s_i = s_\epsilon = 1. \tag{12}$$

Define $\Delta r_i = |\frac{r_i}{s_i} - \frac{r_i}{\tilde{s}_i}|$, if Δr_i exceeds $\tau_{i,r}$, frame i is also included as a failure candidate.

After the selection of failure candidates, free space violation is checked between failure candidates and other frames in the map. If any violation is detected, the algorithm ends, thus result of PGO is rejected. In this case, the previous merge between loop pairs should be nullified and everything in the map should be restored to the same state as it was before PGO.

V. EXPERIMENTS

A. Experiment Setting

We carried out experiments on large-scale real-world datasets to evaluate the performance of our fast structural representation. We used an M1 chip Macbook Pro with 16 GB RAM. To show the practical usability, we evaluated the timings and precision recall of failure detection, which will be discussed in the following section.

Moreover, in our implementation, 7-dof PGO was implemented following the similarity transform setting in [38] and used g2o [39] as back-end.

B. Datasets

The problem of perceptual aliasing is less frequent in outdoor cases compared with indoor, due to the lack of duplicate structures. There should be enough similar places in the dataset to show the performance of the detector. New datasets were created since public benchmarks that are suitable for our method were not available.

Our datasets were taken indoors, covering several difficult scenarios for conventional loop detector including long corridor and staircases. The datasets are composed by synchronized video data (taken by Insta360 ONE X2) and LiDAR data (taken by Ouster OS0-128).

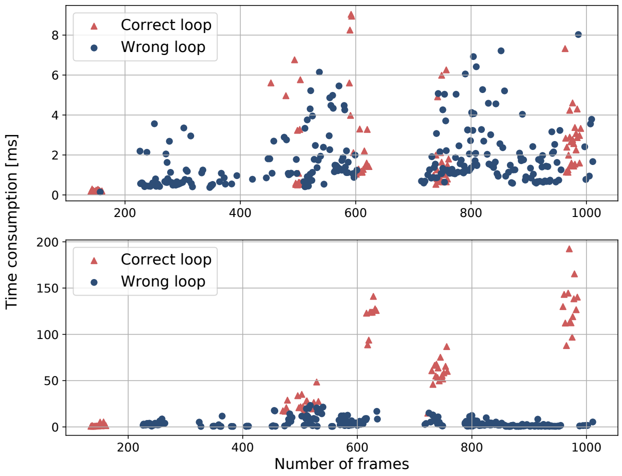

Fig. 7. **Top:** Checking time consumption when applying our detection method. **Bottom:** Checking time consumption when applying brute-force. Notice that the empty interval between 600 and 700 frames indicates that there were no loop closing detected in both datasets.

VI. Results

A. Timings

We measure the total generation time of $\mathbf{C}_{intra}$, $\mathbf{C}_{inter}$, r_{min}, and average free space checking time to show the real-time performance of our method.

The generation time is directly affected by the resolution of the pixel map and the effective range of landmarks. Let the pixel map size be $n \times m$ and effective range be l pixels in the map, Table I shows the generation time consumption measurement for each frame.

As stated in the previous section, the purpose of failure candidate selection method is to reduce the time needed for free space checking. Otherwise, the free space checking will become a brute force process for all frames in maps which can be inefficient in cases of the correct loop. Figure 7 and Table II give the time consumption performance of our method and brute-force in two datasets with 1011 and 795 keyframes. Since there is no free space overlapping in the correct loop, the brute-force method will not meet break points and will make time consumption large. When map size becomes larger, time consumption with brute-force will grow quadratically.

Further, structure-aware loop detection stated in Section IV-A also helps in reducing number of loop closure outliers and thus reduce the checking time.

B. Precision Recall of Failure Detection

We use precision and recall (PR) to evaluate the performance of our failure detector. We define the concept that "the loop closure result is wrong" as *Positive*. Table III gives the results. We compare our method against the following methods:

(1). Visual SLAM system [40], which is based on Visual Odometry (VO) and BoW without failure detection methods (denoted as Baseline).

TABLE III

Precision and Recall of false loop closure detection in real-world datasets

	Resolution $n \times m$, effective range l, (n, m, l)			
	(10,20,1)	(30,60,3)	(50,100,5)	(50,100,10)
Baseline	0/0			
Ours	0.89/**0.95**	0.90/**0.99**	0.91/**0.98**	0.87/**0.98**
NC(0.5)	0.79/0.52	0.75/0.66	0.83/0.65	0.80/0.70
NC(0.8)	0.84/0.32	0.81/0.62	0.86/0.73	0.85/0.61
Without1	0.81/0.83	0.85/0.80	0.85/0.78	0.81/0.76
Without2	0.87/0.74	0.84/0.85	0.90/0.81	0.81/0.88

(2). Failure rejection based on percentage of failure candidates without checking free space. Reject the loop closure when the number of failure candidates exceeds 50% of frames in the loop (denoted as NC(0.5)).

(3). Failure rejection based on the percentage of failure candidates with 80% percentage (denoted as NC(0.8)).

(4). Our proposed method removing Category 1 candidate selection in Section IV-B (denoted as Without1).

(5). Our proposed method removing Category 2 candidate selection in Section IV-B (denoted as Without2).

In our case, *recall* is critical which means how many wrong loop closures are detected.

C. Qualitative Evaluation

We tested our method on several large-scale indoor real world datasets, which contains multiple places sharing highly alike structures. To show the effectiveness of our methods, we compare our methods qualitatively with some open-source systems including Fast-LIO [8], Fast-LIO-SLAM with the application of SC-A-LOAM [41], and the baseline method. Figure 8 and 9 show the comparison between our proposal and baseline method as well.

In order to check whether IMU data can help to resolve the problem of perceptual aliasing, we also planned to test the performance of visual-inertial-odometry based SLAM systems. However, since our indoor data consists some narrow corridors and staircases, it can cause failure in tracking due to the lack of features if it is handled by the Pinhold camera model. Then panoramic data has to be applied to ensure a successful tracking process. Our literature research indicates that open-source visual inertial SLAM that supports panoramic data currently is not available. Therefore, we choose a more robust and stable LIO-based method to show whether IMU data can help in our case.

D. Results Interpretation

Table I and II show that the time taken for either the generation of structural representation or free space checking process are relatively small and have almost no influence on the original SLAM algorithm. The experimental performance shows the average frame rate of our implementation was around 27[fps] which can be defined as *real-time*.

Table III shows the precision recall performance under different parameter settings. Our method can successfully detect 98% of false loop closing results.

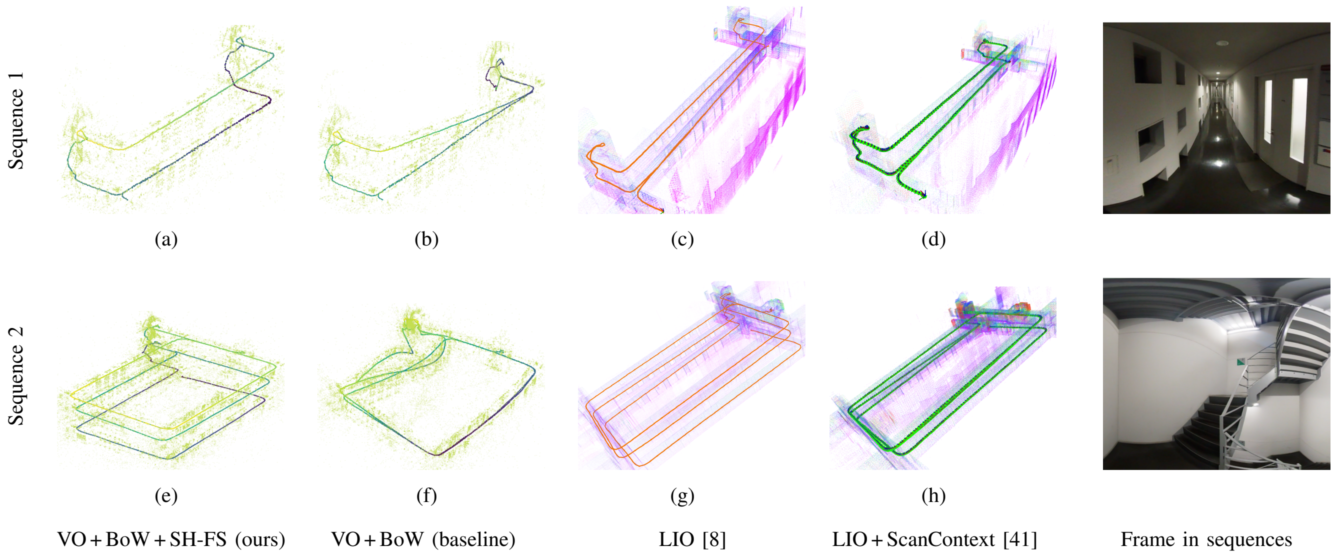

Fig. 8. (a) to (d) show the performance of different methods in a simple sequence that covers the structure of two floors. (e) to (h) show the performance of different methods in a relatively harder sequence that covers the structure of three floors with large loops. We can see that in (h), even with IMU and LiDAR data, the structure still collapsed due to wrong loop closures between different floors. The rightmost images show typical places where false loop closing happens. The intuitive reason is also the mutual consistency between different floors, causing ScanContext detection to make mistakes.

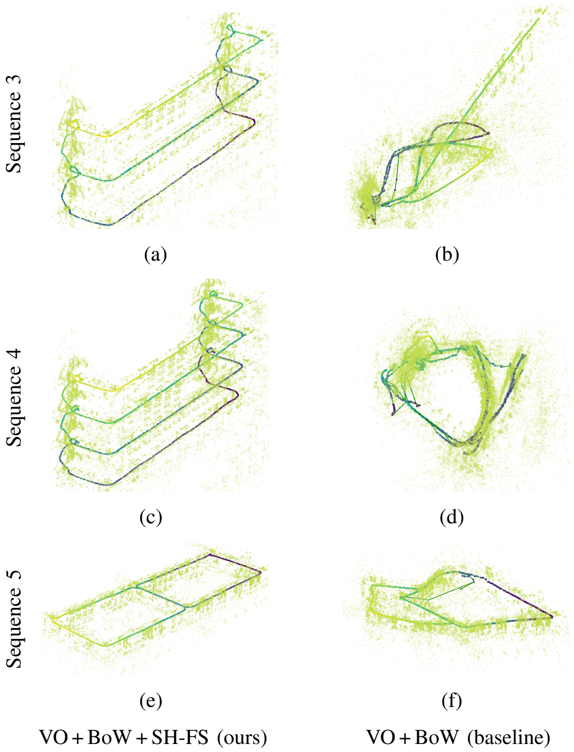

Fig. 9. Harder cases with more floors and more potential false loop closings were also tested. When there was no failure checking process, the baseline method can make huge mapping errors and lead to structural failure.

Furthermore, Table III results indicate that our method is robust and does not rely on parameter tuning. Small sized map might fail to cover all the structure information thus the performance might be degraded. Hence, the map size should be made *enoughly* large, in our case $(30, 60, 3)$, to keep good performance.

Figure 8 confirms the wide existence of perceptual aliasing in different SLAM systems. In certain indoor cases, IMU and LiDAR data might also not help to avoid it. Figure 8 and 9 also gives a clear picture of how our method can help to avoid structure failure caused by perceptual aliasing and further indicate the effectiveness of our failure detection and free space checking methods.

Our method does not rely on PGO algorithms as well. It can be applied to methods based on pose graph and landmarks, which makes it very easy to be implemented.

The results also reveal the limitations of our method. First, our method does not ensure 100% failure detection since theoretically our method can only detect those failures that cause free space overlapping. Wrong loop closure without structure failure can occur in either very small or very large loops, in this case our method might be unsuccessful. Second, though our method can be rapid, when the map grows larger, the time cost by PGO becomes the bottleneck. In such cases, the time consumed by PGO might be even hundreds of times longer than the time taken by free space checking. Thus applying our checking method will hardly improve time efficiency in those cases.

VII. Conclusion

In this paper, we proposed spherical harmonics based fast structural representation (SH-FS) in visual SLAM framework that can help to improve the accuracy and robustness of loop closing and loop correction. We demonstrate high performance in several large-scale real-world datasets, with substantial improvement comparing baseline method. Furthermore, different from normal robust PGO methods, we do not try to mitigate influence of outliers during optimization; nevertheless, we reject outliers based on the structure healthiness after optimization. Thus our method is easier to implement and takes lesser time.

Future work will further generalize the fast structural representation and application in different part of SLAM, such as tracking and mapping modules. We also plan to implement our method in both visual inertial SLAM and LiDAR inertial SLAM systems. In addition, we aim to extend our fast structural representation into an environment reconstruction system, leveraging the fast computation, which can be useful in other tasks such as path planning for robots.

REFERENCES

[1] P. Lajoie, S. Hu, G. Beltrame, and L. Carlone. "Modeling perceptual aliasing in SLAM via discrete-continuous graphical models," *IEEE Robotics and Automation Letters*, 2019.

[2] R. Mur-Artal and J. D. Tardos, "ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras," *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.

[3] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. Montiel, and J. D.Tardós, "Orb-slam3: An accurate open-source library for visual, visual-inertial and multi-map slam," *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 1874-1890, 2020.

[4] 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.

[5] Y. Latif, C. Cadena, and J. Neira, "Robust loop closing over time for pose graph SLAM," *International Journal of Robotics Research*, vol. 32, no. 14, pp. 1611–1626, 2013.

[6] F. Wu and G. Beltrame, "Cluster-based penalty scaling for robust pose graph optimization," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6193–6200, 2020.

[7] P. Agarwal, G. D. Tipaldi, and L. Spinello, "Robot Map Optimization using Dynamic Covariance Scaling," *2013 IEEE International Conference onRobotics and Automation (ICRA)*, 2013.

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

[9] A. Rosinol, T. Sattler, M. Pollefeys, and L. Carlone. "Incremental Visual-Inertial 3D Mesh Generation with Structural Regularities". In *IEEE International Conference on Robotics and Automation*, 2019.

[10] A. Rosinol, A. Violette, M. Abate, N. Hughes, Y. Chang, J. Shi, A. Gupta, and L. Carlone, "Kimera: From SLAM to spatial perception with 3D dynamic scene graphs," *International Journal of Robotics Research*, vol. 40, no. 12-14, pp. 1510-1546, 2021.

[11] E. Eade and T. Drummond, "Unified loop closing and recovery for real time monocular SLAM," in *Proc. 19th British Machine Vision Conference*, Leeds, UK, September 2008.

[12] B.-J. Ho, P. Sodhi, P. Teixeira, M. Hsiao, T. Kusnur, and M. Kaess, "Virtual occupancy grid map for submap-based pose graph SLAM and planning in 3D environments," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2175–2182, 2018.

[13] D. Nister and H. Stewenius, "Scalable recognition with a vocabulary tree," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 2161-2168, 2006.

[14] 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.

[15] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (SURF)," *Computer Visions and Image Understanding*, vol. 10, no. 3, pp. 346–359, 2008.

[16] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[17] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: an efficient alternative to SIFT or SURF," *IEEE International Conference on Computer Vision*, pp. 2564–2571, 2011.

[18] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "Brief: Binary robust independent elementary features," In *European Conference on Computer Vision*, pp. 778–792, 2010.

[19] S. Leutenegger, M. Chli, and R. Y. Siegwart, "BRISK: Binary robust invariant scalable keypoints," In *European Conference on Computer Vision*, pp. 2548–2555, 2011.

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

[21] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM: a versatile and accurate monocular SLAM system," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[22] R. Arandjelović, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "NetVLAD: CNN architecture for weakly supervised place recognition," In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 6, pp. 1437-1451, 2018.

[23] D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superpoint: Selfsupervised interest point detection and description," In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 224-236, 2018.

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

[25] M. Dusmanu, I. Rocco, T. Pajdla, M. Pollefeys, J. Sivic, A. Torii, and T. Sattler, "D2-Net: A Trainable CNN for Joint Description and Detection of Local Features," In *IEEE/CVF conference on computer vision and pattern recognition*, pp. 8092-8101, 2019.

[26] D. Li, X. Shi, Q. Long, S. Liu, W. Yang, F. Wang, Q. Wei, and F. Qiao, "DXSLAM: A robust and efficient visual SLAM system with deep features," In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4958-4965, 2020.

[27] J. Tang, L. Ericson, J. Folkesson, and P. Jensfelt, "GCNv2: Efficient correspondence prediction for real-time SLAM," In *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3505–3512, 2019.

[28] T. Sattler et al., "Benchmarking 6DOF outdoor visual localization in changing conditions," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8601–8610, 2018.

[29] C. Toft, W. Maddern, A. Torii, L. Hammarstrand, E. Stenborg, D. Safari, M. Okutomi, M. Pollefeys, J. Sivic, T. Pajdla, et al., "Long-term visual localization revisited," In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[30] A. Torii, J. Sivic, T. Pajdla, and M. Okutomi, "Visual place recognition with repetitive structures," In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 883-890, 2013.

[31] N. Sunderhauf and P. Protzel, "Switchable constraints for robust pose graph SLAM," *IEEE International Conference on Intelligent Robots and Systems*, pp. 1879–1884, 2012.

[32] L. Carlone and G. C. Calafiore, "Convex Relaxations for Pose Graph Optimization with Outliers," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 1160–1167, 2018.

[33] A. Eudes and M. Lhuillier, "Error propagations for local bundle adjustment," in *Computer Vision and Pattern Recognition*, pp. 2411–2418, 2009.

[34] M. Polic, W. Forstner, and T. Pajdla, "Fast and accurate camera covariance computation for large 3d reconstruction". In *European Conference on Computer Vision*, 2018.

[35] H. Friedrich, D. Dederscheck, K. Krajsek, and R. Mester. "Viewbased robot localization using spherical harmonics: Concept and first experimental results". In *Joint Pattern Recognition Symposium*, vol. 4713, pp. 21–31. Springer, 2007.

[36] C. Alexandre, R. Patrick, and F. David. "Appearance-based segmentation of indoors/outdoors sequences of spherical views". In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1946-1951, 2013.

[37] K. Michael, F. Thomas, and R. Szymon. "Rotation invariant spherical harmonic representation of 3 d shape descriptors". In *Symposium on geometry processing*, vol. 6, pp. 156-164, 2003.

[38] H. Strasdat, J. M. M. Montiel, and A. J. Davison, "Scale drift-aware large scale monocular slam," *Robotics: Science and Systems*, vol. 2, no. 3, pp 7, 2010.

[39] G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "g2o: A general framework for graph optimization," in *IEEE International Conference on Robotics and Automation*, pp.9-13, 2011.

[40] S. Sumikura, M. Shibuya, and K. Sakurada. 2019. "OpenVSLAM: A Versatile Visual SLAM Framework," *ACM International Conference on Multimedia*, 2019.

[41] G. Kim, S. Choi, and A. Kim, "Scan Context++: Structural Place Recognition Robust to Rotation and Lateral Variations in Urban Environments," In *IEEE Transactions on Robotics*, 2021.