IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received October 24, 2021, accepted November 15, 2021, date of publication November 25, 2021,
date of current version December 9, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3130928

The Relocalization of SLAM Tracking Based on Spherical Cameras

QINGLING CHANG[1], QIANG LIU[2], XIN YANG[1], HUANG YAJIANG[2], FEI REN[1], AND YAN CUI[1,2]

[1]China-Germany (Jiangmen) Artificial Intelligence Institute, Wuyi University, Jiangmen 529000, China
[2]Zhuhai 4Dage Network Technology, Zhuhai 519000, China

Corresponding author: Yan Cui (cuiyan@wyu.edu.cn)

This work was supported in part by school research projects of Wuyi University under Grant 2019AL032.

ABSTRACT This work proposes a novel solution to relocalize the SLAM tracking based on spherical cameras. It focuses on the imaging method of spherical camera, the feature extracting algorithm and the relocalization of SLAM tracking based on the 3D reconstruction. In the imaging method, we design a new camera containing eight fish-eye lenses, and then we propose a calibration method to calibrate the eight fish-eye lenses spherical camera; To get the high-performance feature points of panoramic image, we propose a network based on a separate network to extract local feature accurately and quickly. With the correct key points obtained by the feature extracting method, we reoptimize the SLAM tracking after the maximum posteriori estimation usually applied in common back-end SLAM to relocalize the SLAM tracking. The experiment results show that the calibration method achieved 0.973 reprojection error, lower than the common methods like Zhang's or DLT. The inlier rate and matching time of proposed SimpGeoDesc are all better than the reference models ContextDesc and GeoDesc. With the correct feature points, SLAM tracking is clearer and more steadily with our relocalization method. That is the solution of relocalization of SLAM tracking proposed in this work is effective. The AR application of the relocalization proves the feasibility of our propose relocalization method.

INDEX TERMS Camera relocalization, calibration, local feature descriptor, spherical camera, SLAM tracking.

I. INTRODUCTION

Camera relocalization refers to estimate the 6-DoF (Degree of Freedom) camera pose from an image with respect to a known environment. It is widely used in computer vision and robotics applications, like Simultaneous Localization and Mapping technology (SLAM), Augmented Reality (AR), and navigation. One convenient solution is to use advanced hardware, such as LIDAR (Light Detection And Ranging), Bluetooth, and GPS (Global Positioning System), etc., but this kind of method is affected by the weather, the light and the signal, etc. seriously. Another popular method just uses an RGB sensor to relocalize the camera pose, known as visual relocalization. This paper focuses on the relocalization of SLAM tracking with the RGB sensor. The traditional visual relocalization methods usually find similar images in the database through image retrieval [1], [2] first and then compute the relative poses of the retrieved images [3]–[5]. Though the performance is good, the retrieval process costs too much time to match the query image against all the database images. Some works are based on the structure to establish the correspondences between 2D image pixels and 3D scene points and then solve camera poses by PnP (Perspective-n-Point) algorithms [6]–[8]. And some recent researchers proposed CNNs-based approaches, they have shown great success in image-based localization, like PoseNet [9] and its variants [10]–[12]. These methods learn to regress the absolute camera poses from the input images through a CNN. Though they are efficient, the accuracy falls behind the structure-based methods. What's worse, researchers find that almost all of these existing methods cannot be used in SLAM tracking to relocalize the camera

The associate editor coordinating the review of this manuscript and approving it for publication was Turgay Celik.

well for the challenge of reducing the accumulated errors of SLAM tracking due to the local information during the back-end optimization.

On the other hand, SLAM [12] is used to describe the process of a carrier carrying a sensor to estimate its position and pose in an unknown environment. And it is widely used in many fields with the development trend of autonomous, miniaturized and intelligent unmanned systems represented by Micro Aerial Vehicle (MAV) and Automated Guided Vehicle (AGV), the application scenarios of the autonomous navigation system are also gradually expanding from military fields such as material delivery, battlefield reconnaissance, and cooperative operations to civil fields such as warehouse management, disaster relief, urban security, resource exploration and power line patrol. SLAM will become more and more important with the development of artificial intelligence (AI), and all these SLAM applications require it can automatic navigate correctly. However, the current SLAM systems are challenging at accuracy and stability. To solve these problems, besides improving the compute ability, reducing the accumulated errors during the movements of the camera (used in SLAM) is very important. The current back-end optimization can release some errors, but it still exists serious drift. Aiming to get a good performance of the SLAM system, we propose a new relocalization approach to reoptimizes the back-end after the maximum estimation. The contributions are summarized as:

(1) a new camera calibration approach for a spherical camera with multi-fisheyes, making sure the consistency of the panoramic images and the seamless splice of panoramic image stitching and providing the panoramic images to extract correct feature points;

(2) a local feature descriptor model SimpGeoDesc, which is suitable for panorama matching and can get accuracy feature point. The accuracy and informative feature points lay the foundation of relocalization of SLAM tracking;

(3) a new idea to relocalize the camera poses to correct the SLAM tracking error based on the local feature information from the SFM sparse point clouds and the global scene information

In the following parts, we discuss the relationships of our approaches with the previous works and then describe the technical details of our relocalization solution, finally claim the relocalization of SLAM tracking performance with experiments and practical application.

II. RELATED WORK

A. CAMERA CALIBRATION

Generally, there are two methods to calibrate the camera, per-pixel models and interpolation [13]–[17]. But the narrow field of view (FOV) lense' calibration methods cannot calibrate the wide FOV lenses, like fish-eye lenses, which consists of several refractions and reflections in the imaging formation process as a result of large number of optical elements. With the increase requirement of wide FOV cameras, works [18], [19] research the fisheye's calibration, [18], [19] implied an unrealistic constraint on the entrance pupil location, which impacted the parameter's sensitivity and correctness, then, in order to relax the single viewpoint assumption constraint, [20] proposed a formation model by taking into account the variation of the entrance pupil using thin lens modeling and presented a calibration procedure for the image formation to estimate the entrance pupil parameters using nonlinear optimization procedure with bundle adjustment. Furthermore, the work [21] described an approach for the self-calibration, the collinearity equations of the pinhole camera model are augmented with five radial lens distortion terms to correct the severe barrel distortion, weighted relative orientation stability constraints are added to the self-calibrating bundle adjustment solution to enforce the angular and positional stability of the camera. But all of them cannot calibrate our designed 8 fish-eye camera well. In order to calibrate correctly our spherical camera, we propose a new method by adding the distortion angular with regression mechanism.

B. LOCAL FEATURE

Local features have become a staple in computer vision with the introduction of SIFT [22]. Typically, they involve three distinct steps: key-point detection, orientation estimation, and descriptor extraction. Other renowned solutions are SURF [23], ORB [24], and AKAZE [25]. SIFT [22] has been proven to be the most robust among the other local invariant feature descriptors with respect to different geometrical changes [26]. Almost all these traditional methods are based on the cues designed artificially, and the cues do not always exist in the images. So, the learning-based methods are studied to learn the descriptor of features with CNNs (convolutional neural networks), and current descriptors are usually trained on precropped patches, typically from SIFT keypoints (i.e., Difference of Gaussians or DoG). They include Deep-desc [27], TFeat [28], L2-Net [29], Hard-Net [30], SOS-Net [31], and LogPolarDesc [32], and the majority of them are trained on the same dataset [33]. Furthermore, recent works use extra cues, such as geometry or global context, including GeoDesc [34] and ContextDesc [35]. And there also have been multiple seeks to learn keypoint detectors separately from the descriptor, including TILDE [36], TCDet [37], Quad-Net [38], and KeyNet [39]. A substitute is to treat this as an end-to-end learning problem, a trend that started with the introduction of LIFT [40] and also includes DELF [41], SuperPoint [42], LF-Net [43], D2-Net [44], and R2D2 [45]. But all these traditional and CNN-based methods are used to process perspective images, and they perform not well on panoramic images. On one hand, the existing models cannot deal with the wide FOV and the distortion in panorama, especially, in the two poles of panoramic images. On the other hand, the lack of labeled panoramic images unenabled the learning-based networks. In order to take use of the rich information and wide FOV of panoramic images, we do many works to extract the efficient feature descriptor of panoramas with learning methods. Inspired by the idea of adding geometry and global context information to the network in [34][35], we first extract the initial feature with SIFT [22] as the input of our framework, and then add the separate network after the input referring to mobileNet[46], thus, we propose the SimpGeoDesc to extract the panoramas' descriptor.

C. CAMERA RELOCALIZATION

Visual place recognition methods and 3D model based localization algorithms are the two main classes of camera relocalization. Visual place recognition methods find similar images in the database through image retrieval [1], [2], [47] and then computes the relative poses with the retrieved images [4], [5]. They have good generalization to unseen scenes, but the retrieval process needs to match the query image against all the database images, which can be costly for time-critical applications. Structure-based localization methods explicitly establish the correspondences between 2D image pixels and 3D scene points and then solve camera poses by PnP algorithms [6]–[8]. However, descriptor matching is expensive and time-consuming procedure making camera relocalization complicated problem for large scale scenes. In order to accelerate this stage, Active Search [48], [49] and its variants [50]–[53] eliminate correspondence search as soon as enough matches have been found. Recently, CNNs-based approaches PoseNet [9] and its variants [610]–[12] which learn to regress the absolute camera poses from the input images through a CNN have achieved great success. They are simple and efficient but generally fall behind the structure-based methods in terms of accuracy, as validated by [54]–[56]. and then researchers utilize a Scene Coordinate Regression model to predict pixels coordinate [57]–[59] or CNNs [54], [55], [60] with ground truth scene coordinates to improve the correspondence of structure-based methods. And the accuracy of camera relocalization is a very important factor in SLAM which attached more and more attentions recently. The first SLAM system MonoSLAM [61] used an extended Kalman filter (EKF) as the back-end and tracks very sparse feature points on the front-end. And PTAM [62] was the first solution to use nonlinear optimization instead of filters and the main solution for later SLAM systems. After that, Oriented FAST and Rotated BRIEF (ORB)-SLAM [63], [64] extended the system to three-thread structure (tracking, mapping, loop detection), since on, many SLAM systems were proposed, such as PL-SLAM [65]–[67], fisheye-SLAM [68], multicol-SLAM [69], VO-SF[70] combines visual odometry, k-means, and scene flow and reconstructs a 3D model of the rigid scene, and so on. Except these indirect methods, large-scale direct monocular SLAM (LSD-SLAM) [71] and direct sparse odometry (DSO) [72] are the representative direct works, and there is also semi-direct monocular visual odometry (SVO) [73]. Furthermore, in order to deal with the wide FOV spherical images, [74] proposed a panoramic image pose calculation method, after that, many works turn to research the panorama SLAM, like the early phase [75]–[77], and the recent works [78]–[80] focus on panoramas or omnidirectional visual SLAM. But drift or accumulate error in the optimization are still the challenge. In this work, we propose a new method to relocalization the camera pose based on SFM point clouds in the SLAM tracking.

III. METHODS AND MATERIALS

This section introduces the main three processes of relocalization, the imaging model of spherical cameras, the local feature based on deep learning and the relocalization of SLAM.And the processes are shown in FIGURE 1.

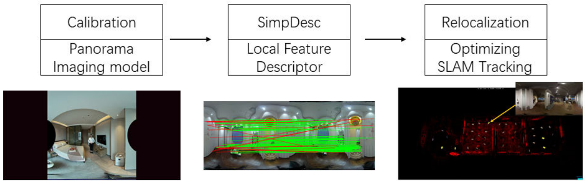


FIGURE 1. The process of our works.

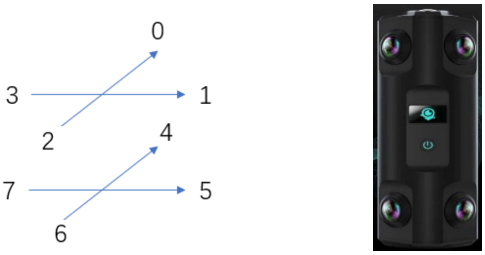
A. IMAGING SOLUTION OF SPHERICAL CAMERAS

The spherical cameras imaging model provides the fundamental of relocalization, which describes our spherical camera setup and its calibration method.

1) SPHERICAL CAMERA SETUP

We designed a camera called 4DKanKan Pro [81], composed of 8 fisheye lenses (4 in up group and 4 in down group). In each group, the angle of two adjacent lenses is 90 degrees, while the horizontal and vertical FOV of the fish-eye lens is 140 degrees 200 degrees. When stitching the eight fish-eyes panoramic images to a spherical image, the horizontal overlap FOV is 50 degrees, and the vertical overlap FOV is 40 degrees. For the uniform distribution of the four fisheyes in each group, the intersection of straight lines at the optical centers of four lenses is set as a virtual optical center of the four lenses. The distance from the virtual optical center to the lens is designed to be 35mm because of the physical size of the lens itself. The camera setup is shown in FIGURE 2.

In this paper, we use the spherical camera to obtain panoramic images, based on the panoramic images, we extract local feature for the 3D sparse and dense reconstruction [82]. In the 3D reconstruction, we computed the depth with the stereo vision, so we design our camera setup with up and down groups fish-eye lenses and the horizontal and vertical FOV should be 360 degrees and 180 degrees separately. So, it needs four lenses in each group. The reason is, the used fish-eye lenses' horizontal and vertical FOV is 140 degrees and 200 degrees, the vertical FOV is enough, but in horizontal direction, it need some overlaps to stitch the fish-eye images, and experiments find that when the overlap degree is equal or more 50 degree, the stitching result is best, so, the appropriate fish-eye lens number can be calculated by the formula 360/(140-50)=4, that is to say, it needs four



(a)the distribution of 8 fisheyes (b) the 8-fisheye camera

FIGURE 2. 4DKanKan Pro[81] structure.

fish-eye lenses and they can just be set on the coordinate axis. The resolution of fish-eye lens is 4608∗3456 and the spherical camera resolution is 8192∗4096.

2) SPHERICAL CAMERA CALIBRATION

a: THE CALIBRATION PATTERNS

The calibration object is a plurality of uniformly distributed black and white chessboard, see FIGURE 3, every nine chessboards compose a group, and they arrange a straight line, there are four groups in total, and the four groups are evenly distributed in a cylindrical space with a radius of 1.5 meters, that is, the angular of two adjacent groups is 90 degrees when calibration is performed, the camera is located at the center of the circle of the horizontal section of the cylinder on the ground to shoot, and a series of images are acquired. Each two adjacent images both in horizontal and vertical directions form a pair, with these pairs of images, we can perform the calibration.

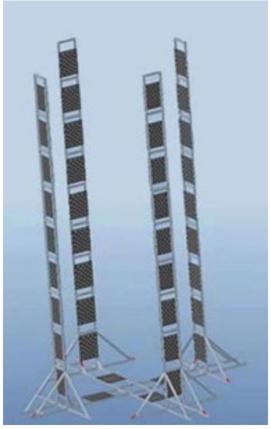


FIGURE 3. The calibration patterns.

b: CALIBRATION

When to calibrate the camera, we first detect the corner according to [83], then, the camera captures the image pairs of the chessboards, after that, it projects the chessboard corners from the current frame to the reference frame, and calculates the distance of each pair of corners, finally, optimizes the internal and external parameters through cost function. The cost function is defined as:

$$F = \min \sum_{i=1}^{N} [(u_{cur}^{i} - u_{ref}^{i})^{2} + (v_{cur}^{i} - v_{ref}^{i})^{2}] \tag{1}$$

where $\mathrm{u}_{\mathrm{cur}}^{\mathrm{i}}$ is the abscissa of the i^{th} point of the current frame projected on the reference frame, v_{cur}^{i} is the ordinate of the i^{th} point of the current frame projected on the reference frame, u_{ref}^{i} is the abscissa of the i^{th} point of the reference frame on the reference frame, and v_{ref}^{i} is the ordinate of the i^{th} point of the reference frame on the reference frame.

c: UNDISTORTION

With the distortion table of fish-eye lenses, we simulate a nonlinear model defined as:

$$\theta_{undistort} = \sum_{i=0}^{9} C_i * \theta_{distort}^{i} \tag{2}$$

where $\theta_{undistort}$ is the distortion angular, C_i is the coefficient of $\theta_{distort}^{i}, \theta_{distort}^{i}$ is the i exponential of distortion angular θ in the distortion table. Furthermore, we modify the cost function (equation 1), adding the distortion equation (2) to define the final cost:

$$\cos t = \min \sum_{i=0}^{N} (I_{ref}^{i} - \pi(T_{c2r}\pi^{-1}I_{cur}^{i})) \tag{3}$$

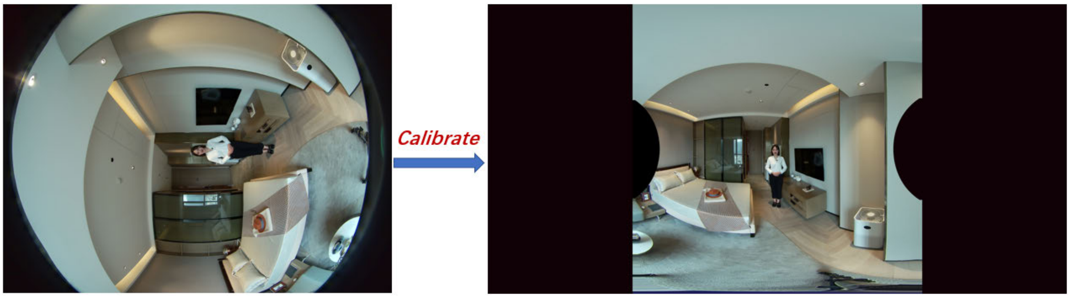
where I_{ref}^{i} is the reference frame of i^{th} pair features, and I_{cur}^{i} is the current frame of i^{th} pair features, T_{c2r} is the translation from the current frame to reference frame with the initial parameters (K, R, T). π is distortion function to project the image coordinate to camera coordinate with the internal parameters, and π^{-1} is in contrast with π, representing the projection from the camera coordinate to image coordinate.

In this work, we initialize the internal parameter as follows: focal is the theory value provided by the camera supporter, c_x is half of the width, and c_y is half of the heigh. R is 90 degrees according to the framework of the camera; T is set according to the distance of the optic center between 3.5cm and 15cm.

To get the minimal cost, we take about five times images of chessboards. For every pair of images left and right or up and down, we compute the cost, and i is iterative among each matching feature, and loop the process with the five times. For the 8 lenses, we get 12 minimal costs to get the accurate K, R and T. We have implemented the calibration, the raw and calibrated images can be seen in FIGURE 4, (a) is the raw image shot by one fish-eye lens with 140 degrees horizontal and 200 degrees vertical FOV. (b) is the calibrated image, it is an equirectangular image with arcs in the two sides.

3) PANORAMIC IMAGE STITCHING

The stitching process of panorama with the calibrated multi-fish eye camera shows in FIGURE 5, with which, we stitch all the 4 gaps of 4 fish-eye images in each group, and an indoor scene's images stitched result is shown in FIGURE 6, after the ‘‘imaging’’ step, we can get the stitched equirectangular



(a)The raw image taken by fisheye lens

(b) The image calibrated from (a)

FIGURE 4. **The raw image and calibrated image.**

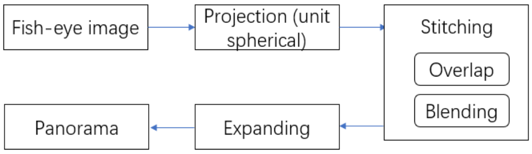


FIGURE 5. **The process of stitching panorama from multi-fish images.**

images, which is clear and fluent. Though there is a little discontinuous, eg. the 1/3 and 2/3 horizontal of the stitched image for some objective factors, such as the fish-eye lens structure and craft and the light. The structure causes the optical centers of the 8 fish-eye lenses camera cannot be completely coincident, leading to some position error of the stitched images. The craft causes the sensitivity of each lens is different, and the light of different views are also distinct, leading the color of splice may be inconsistent. In order to get continuous spherical image, we process the stitched image with optical flow shown in FIGURE 6.

B. LOCAL FEATURE BASED ON THE SEPARATE NETWORK FOR PANORAMIC IMAGES

The local feature descriptors based on learning have achieved significant improvement in recent years. However, most of these methods perform not well on panoramic images. To meet our image-based camera relocalization and 3D reconstruction requirements and other applications based on panoramic images, we propose an efficient local feature descriptor learning model based on CNN by transferring the model trained on perspective images to panoramic images.

To get more comprehensive and accurate feature descriptor, for lacking labeled panoramic image dataset, we started with studying global feature descriptor methods, such as the high-performance ContextDesc [35], a global augmentation model trained on the GL3D [34], a dataset generated from perspective images, and we transferred it to the panoramic images directly, it performed poorly. Analyzing discovers the reasons lie in: (1) in the geometric context augmentation model, the geometric context coordinates are encoded into the feature representation, but the coordinate of panoramic key-point is different a lot from the coordinate of perspective key-point. (2) in the visual context augmentation model, it extracts high-level visual information from perspective images with a ResNet [84] model, but for the panoramic image, the visual information is not distributed uniformly for its different distortions on different locations. That is, it is unreasonable to use the interpolation method proposed in ContextDesc [35] to get visual information and take it as a kind of augmentation on panoramic images. So, the augmentation model (ContextDesc [35]) trained on GL3D [34]cannot transfer to panoramic images seamlessly. Now that the efficient global model cannot suit the panoramic images for the above reasons, we turn to study the local feature methods. For the high performance of ContextDesc [35], we furthermore researched its one baseline model-GeoDesc [34], a model also trained on GL3D [34], and we transferred GeoDesc [34] to panoramic images dataset directly too, the experiment results show that though the accuracy can be acceptable, the cost time is too long, which cannot meet the real-time computing requirement in practical application. To speed up, we simplified the GeoDesc [34] using six separable convolution layers to replace the original feature descriptor and proposed a simply local feature descriptor network fitting panoramic images (called SimpGeoDesc), and the framework of SimpGeoDesc is shown in FIGURE 7. The SimpGeoDesc contains six layers, and in each layer, it consists of a MobileNet [46], a BN and a Relu layer. The input refers to GeoDesc [34], including an image, key-points, and the descriptor extracted by the SIFT [22] patch-based.

In practical application, before extracting the feature descriptor, we preprocess the panoramic images by just taking the middle part of the panoramic image avoiding the poles' serious distortion impact, but its horizontal angle is 360 degree which differs from the perspective image. Experiments show that our proposed local feature descriptor SimpGeoDesc model can extract the feature of panoramic image high-efficient, for the only six layers network, it is faster than GeoDesc [34] without losing accuracy, even in mobile phones, its computing cost can be accepted, e.g., with an image of 2048*1024 pixels, the feature extraction process costs about 1500 ms, and with an image of 640*480 pixels, the feature extraction process costs about 200 ms (including the SIFT [22] extraction process). Furthermore, we have

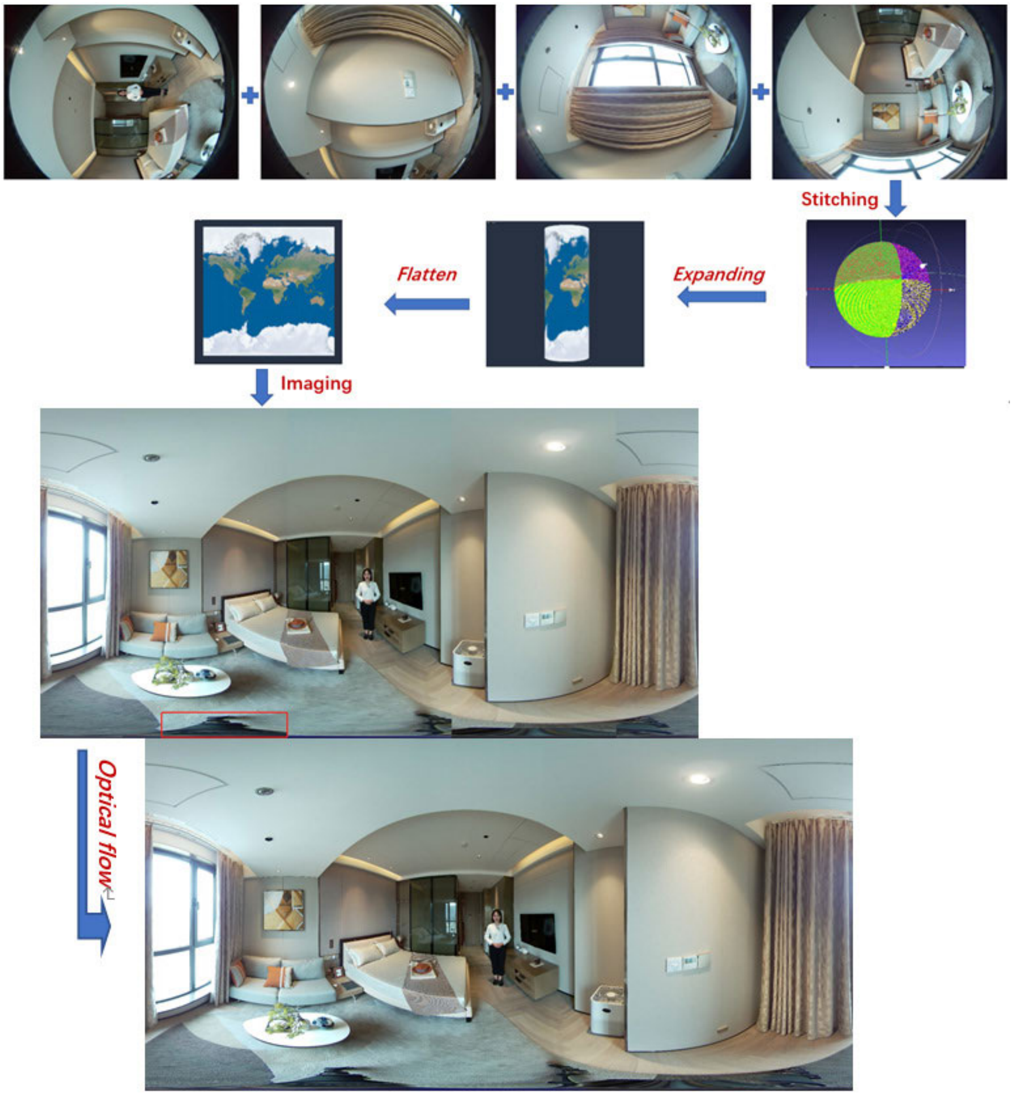


FIGURE 6. The process of stitching 4 fisheye images to a panoramic image and optical flow.

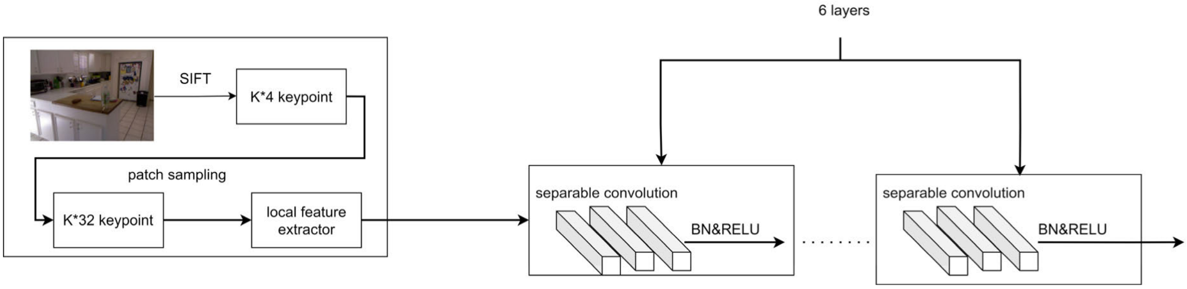


FIGURE 7. The framework of SimpGeoDesc.

proved that the performance decreased a lot with less than six layers in the network, and the experiment results can be seen in FIGURE 21.

C. RELOCALIZATION OF THE SLAM TRACKING BASED ON SFM

Generally, the camera pose is calculated in the common SLAM system with local feature, leading to the drift or error. What's worse, the error will cumulate with time going by, to correct the drift is still a challenge in SLAM. The fusion of spherical camera and IMU have improved the performance of SLAM tracking while cannot meet the requirement of some strict environment application. Inspirited by the SFM, we furthermore reoptimize the tracking with the global feature information from the SFM point clouds. It firstly reconstructs the sparse point clouds with SFM, then with

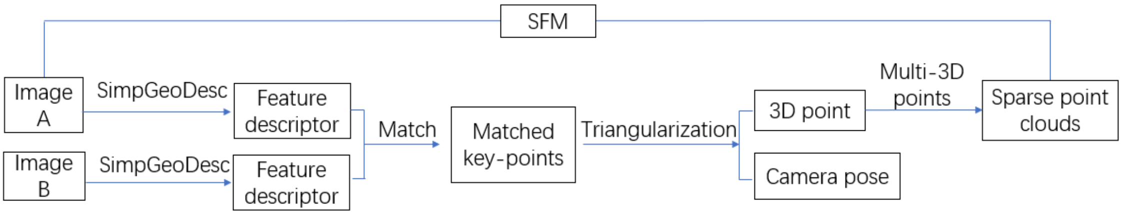


FIGURE 8. The SFM process.

the point clouds as the ground truth, it corrects the tracking camera poses through some extracted keyframes from SLAM to implement the relocalization. We test the proposed approach in practical application and get good visual performance.

1) SPARSE 3D RECONSTRUCTION OF SCENES WITH SFM

Based on SimpGeoDesc, the local feature descriptor extracted model proposed above, we get the matched key-points and then triangularine the matched key-points to get the 3D point and the camera pose. In the triangularization process, it first calculates the 3D point based on the 2D matched key-points extracted from the stereo images captured by 4DKanKan Pro [81] in the first location, and then, with the 3D point and the matched key-points of the subsequent images captured in the following-up locations, it computes the camera pose of the subsequent images. Also, many 3D points compose the sparse point clouds. The process of SFM is shown in FIGURE 8.

In order to explain the process of computing the 3D point and camera pose in detail, we take an example to describe the SFM process illustrated in FIGURE 9. FIGURE 9 shows that supposing we reconstruct a scene based on SFM with 4DKanKan Pro [81], for the three points A, B, and C of the scene, the camera begins to shoot at the first location to get the stereo images 1 and 2 (in FIGURE 10), and then, the camera shoots continue in subsequent locations to get the image 3, 4 and etc., with feature extracting model-SimpGeoDesc, we get the matched key-points, taking the blue lines (the view ray) to explain, 1a,2a,3a and 4a are the matched key-points from different images captured in different locations. With 1a and 2a, we can calculate the 3D point A based on the triangularization method. Furthermore, thanks to the calibrated stereo camera, we can compute the camera pose of the beginning shooting location, that is the camera pose of images 1 and 2. After that, with the 3D point (A), matched key-points (3a, 4a) and the begin shooting camera pose, we can compute the following-up camera poses of images 3 and 4, etc. The green and red lines represent the other points of the scene that are similar to point A. Many such 3D points (A, B and C) will compose the sparse point clouds. In the practical application, we take the first matched key-points feature descriptor as the 3D point's descriptor, e.g., taking the feature descriptor of 1a as the 3D point A's descriptor.

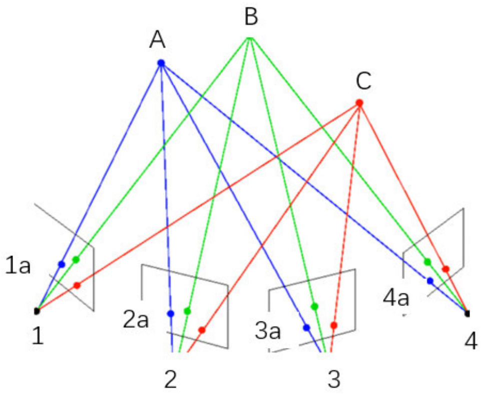


FIGURE 9. Triangularization the matching key-points to get 3D points and camera poses.

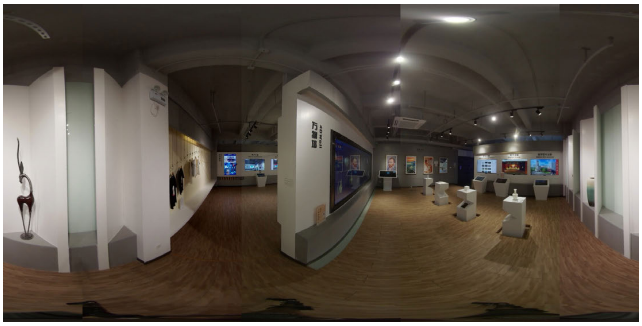


FIGURE 10. A panoramic image of an office.

With the above SFM process, we reconstruct an office scene (FIGURE 10). In order to reconstruct better, we take as many possible as images of the scene, e.g., about 600 images, and then, extract feature descriptors with SimpGeoDesc and match them to get the matched key-points, finally, calculate about 50000 3D points which compose the sparse point clouds of the office scene, the point clouds are shown in FIGURE 11. In FIGURE 11, the red points are the point clouds reconstructed from the yellow or colorful block (the images captured by 4DKanKan Pro [81]) with SFM. The yellow blocks stand for the shooting view is back to us and the colorful blocks are the contractible of the images captured facing us. From FIGURE 10 and FIGURE 11, we can see the point clouds are very similar to the scene, which proves the pose estimated by the feature points is accurate.

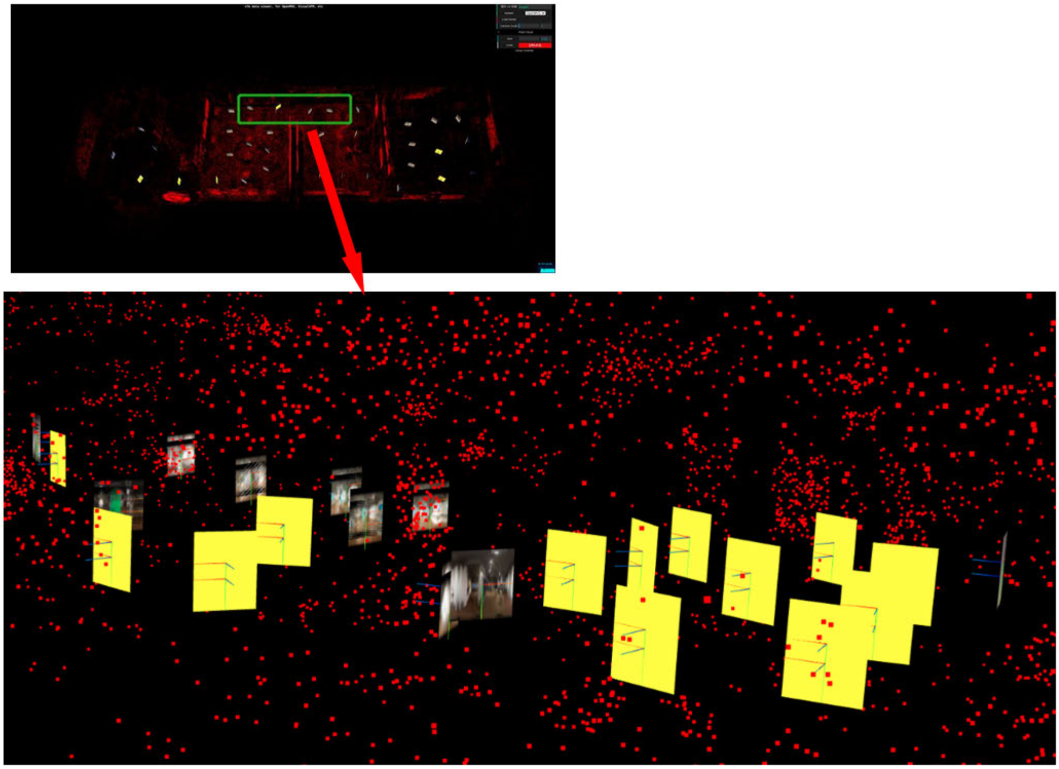


FIGURE 11. The SFM reconstruction based on panoramic images (FIGURE 10).

2) RELOCALIZATION TO OPTIMIZE THE SLAM TRACKING

The classical SLAM location is usually based on the local pose, and the local pose introduces error to the location of SLAM, what's worse, with the tracking increasing, the error will accumulate, leading to the tracking drift. In order to accurately locate the camera and break through the limitation of the local pose, we take the SFM reconstruction point clouds and camera poses based on triangularization as the global pose information. That is, we serve the point clouds and camera poses obtained by SFM as the ground truth in the SLAM tracking, and the process is shown in FIGURE 12. Adding the relocalization step (the green block) in the back-end optimization after maximum posteriori estimation based on the common SLAM framework.

In the relocalization process, there are two main steps, extracting the keyframe from the SLAM tracking and matching the key points to get the accurate camera pose of the keyframe, and the relocalization process is shown in FIGURE 13.

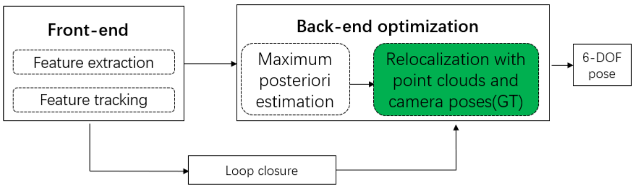


FIGURE 12. The framework of SLAM tracking with relocalization.

a: EXTRACTING FRAME

It refers to extracting keyframes from the SLAM tracking at intervals of time or distance. The first step is to extract a keyframe from the a marching camera of SLAM, then extract the feature descriptors with SimpGeoDesc, and match the key points of the keyframe from SLAM and the point clouds of SFM. The extracted keyframe served as the reference substance of the current SLAM tracking.

b: MATCHING KEY-POINTS

After getting the keyframe of SLAM tracking and extracting the key-points with SimpGeoDesc model from the keyframe, with the Fast Approximate Nearest Neighbor Search Library (FLANN) algorithm [47], the SLAM system matches the keyframe's key-points and the key-points of point clouds obtained from SFM, with the matched key-points, we get the camera pose based on the classical PNP algorithm [8]. For the ground truth camera pose, we can get the accurate camera pose of the keyframe from SLAM tracking. The matching and getting camera pose process is shown in FIGURE 14. The green blocks emphasize the matched key points and the accurate camera pose derivation.

IV. EXPERIMENTS

This section experiments three main parts of the relocalization of SLAM tracking based on spherical cameras. The

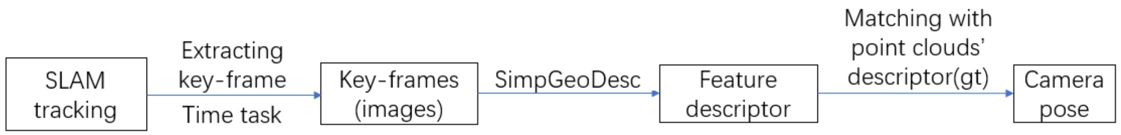


FIGURE 13. The process of relocalization of SLAM tracking.

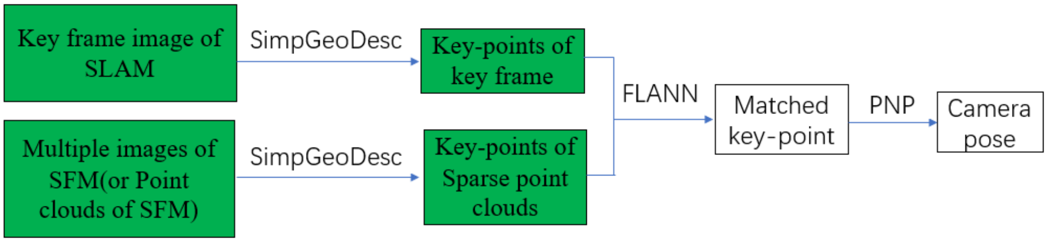


FIGURE 14. The detailed process of the relocalization based on extracted frame from SLAM.

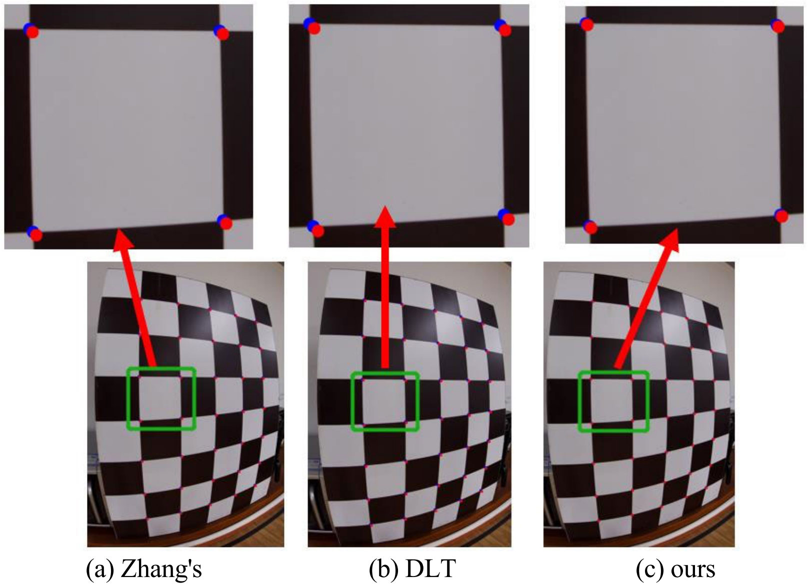


FIGURE 15. The calibration result of three methods.

TABLE 1. The reprojection error of 3 different calibrations (pixel: 5472∗3648).

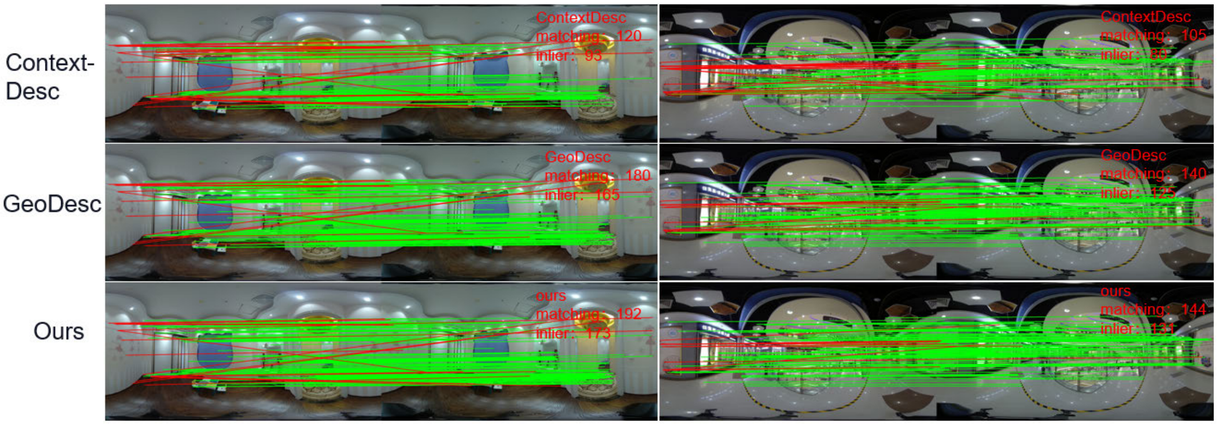
		Zhang's calibration	DLT calibration	ours
Focal/pixel	f	2538	2580	2558
Center point/pixel	cx	2730	2765	2747
	cy	1805	1852	1824
Reprojection error/pixel		2.564	1.871	0.973
Time		1.9s	2s	2.3s

spherical camera calibration, the local feature descriptor model and the relocalization solution.

A. CAMERA CALIBRATION

To prove the performance of our calibration method, we calibrate the multi-fish eyes of 4DKanKan Pro [81] with the traditional Zhang's [85], DLT [86] and our method separately with chessboards. Due to the spherical camera of 4DKanKan Pro [81] structure, the calibration pattern may not face us correctly, the image captured by 4DKanKan Pro [81] is 5472∗3648 pixels, and the calibration results are shown in FIGURE 15 and TABLE 1.

In FIGURE 15, the blue points distributed in the chessboard corner are the ground truth, and the red points are the calibration reprojection. (a) is the Zhang's [85] calibration reprojection, (b) is the DLT [86] calibration reprojection, and



(a) The office scene

(b) The market scene

FIGURE 16. The indoor office and market scenes.

(c) is our method's result. The first line images are the big resolution of the green boxes of second line images. From FIGURE 15, we find that neither Zhang's [85] nor DLT [86] cannot reproject the red points to the blue points correctly, what's worse, it deviates a lot from Zhang's [85] method. In contrast, our proposed calibration method can reproject the red points to blue points almost exactly. Furthermore, TABLE 1 also indicates the same conclusion. The calibration parameters are also shown in the table, and from TABLE 1, we can see the reprojection errors of Zhang's [85] and DLT [86] are obviously. And the reprojection error of our method is 0.973, which can meet our practical application well. The time is a little more than the other methods, for our solution considers various kind of illumination condition and the angle of inclination, leading to slightly time complexity.

B. LOCAL FEATURE DESCRIPTOR BASED ON PANORAMIC IMAGES

1) EXPERIMENT ENVIRONMENT

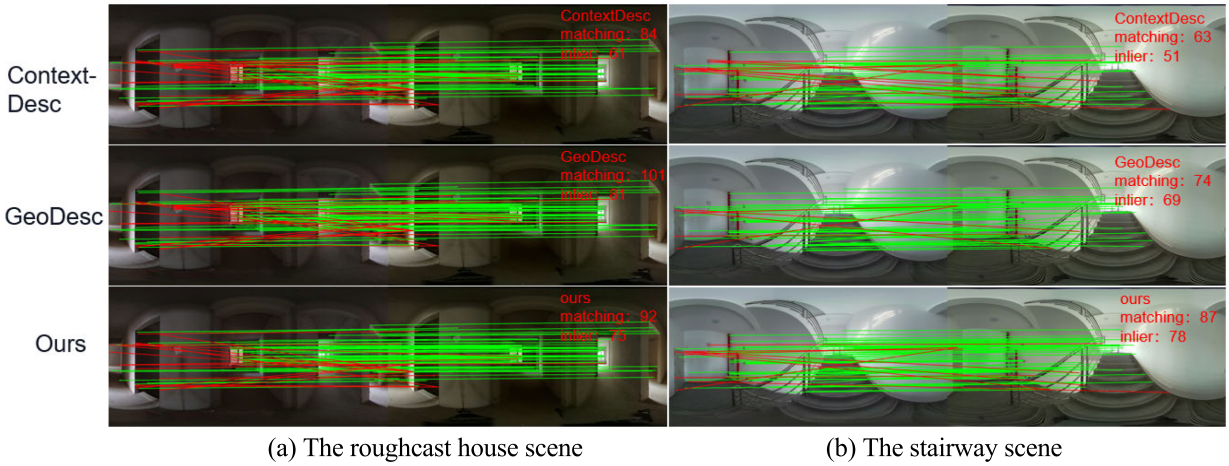
We provide the matched feature numbers and time cost of matching features between left and right image pairs. To compare the speed and accuracy of the three methods (ContextDesc [35], GeoDesc [34] and SimpGeoDesc), we set the number of patches as 512, and the image pixel is 2048*1024. From plenty of experiments, we select two group scenes, indoor and outdoor scenes. The indoor scenes contain office, market, stairway, and roughcast house. Generally, office and market scenes can represent the commonly used scenes, even the large scenes in computer vision, the scenes of stairway and roughcast house are always similar and contain whitewall or contexture less feature, which easily leads to wrong matches. The outdoor scenes contain two distant scenes and two near scenes, and the difference between two distant or near scenes is that the objects are or are not like the background. When the object is like the background, it increases the feature extraction difficulty. With these different scenes, we test the different models and compare the performances.

2) EXPERIMENT RESULTS

We test the three models trained on GL3D [34] on indoor and outdoor scenes separately, and in order to display the effects, we provide the matching results of the three models tested on each scene, and the experiment results tested on four indoor scenes are shown in FIGURE 16 and FIGURE 17. Moreover, the experiment results test on outdoor scenes is shown in FIGURE 18 and FIGURE 19. The red lines indicate the correct matching features, and the green lines indicate the wrong matching features. The evaluation of matching results for descriptors extracted with different models is listed in TABLE 2, TABLE 3. Both for the cases indoor and outdoor scenes, the matching feature number and inlier rate of SimpGeoDesc is similar with GeoDesc [34], better than ContextDesc [35]. However, the speed of SimpGeoDesc is about 1.5 times as fast as GeoDesc [34] and even three times that of ContextDesc [35]. It also indicates that ContextDesc [35] cannot be transferred to panoramic images directly though it performs well on perspective images. Additionally, we find the performance will drop rapidly when the layer number is less than 6, and the performance changes little when the layer number is 10∼6, and the experiment results are shown in FIGURE 20. So, in order to reduce the time cost, we utilize the 6 layers separate convolution in SimpGeoDesc.

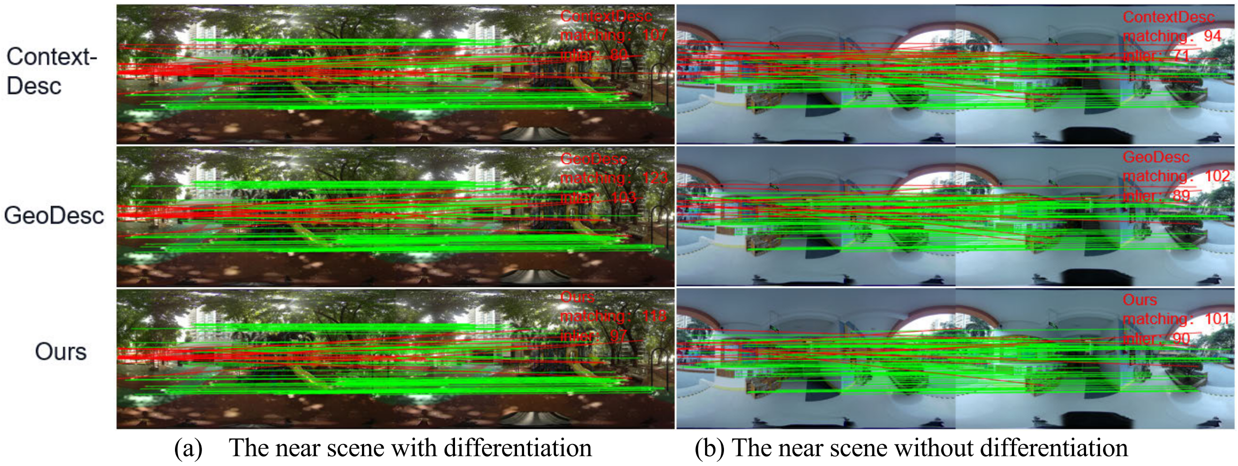
3) CAMERA LOCATION APPLICATION

In order to prove the performance of our proposed SimpGeoDesc model furthermore, we apply the model on camera location applications. We select three different scenes, including the indoor office, outdoor scenes, and stairway scene. The camera location results are shown in FIGURE 21. In FIGURE 21, the top line is the camera location results with descriptors extracted by ContextDesc [35] model, and the following line are the results of our SimpGeoDesc model. The



(a) The roughcast house scene (b) The stairway scene

FIGURE 17. **The indoor roughcast house and stairway scenes.**



(a) The near scene with differentiation (b) The near scene without differentiation

FIGURE 18. **The outdoors near scenes.**

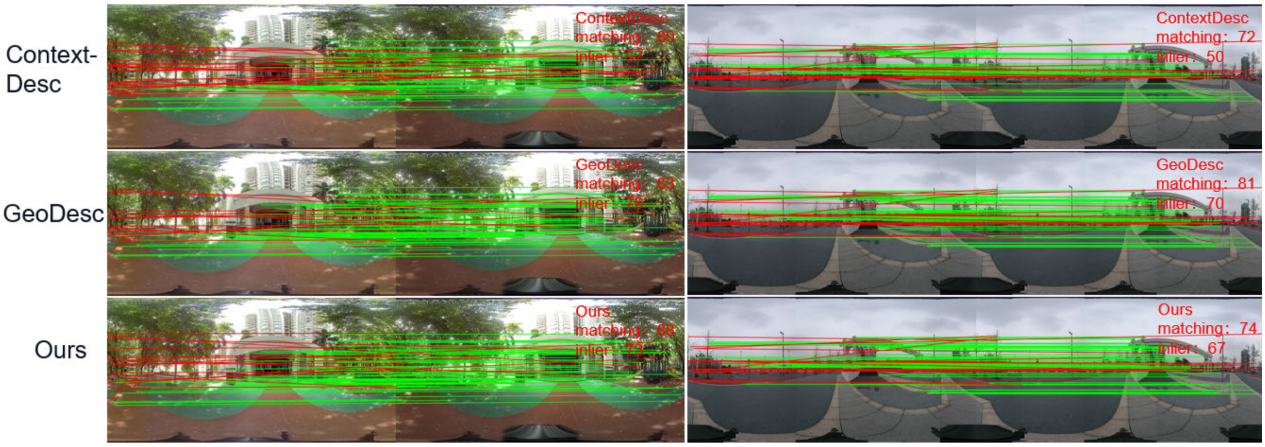
red rectangular in the top line labels the overlap location point in (a) and (b) groups, but in fact, the real distance between the overlap point is more than 1.5 meters. The blue rectangular in the group (c) distributes in two lines indicates the number of location points, and in the up line, it only calculates 34 points comparing the down line's 78 points and the real number of the stairway points is 78. Additionally, though there are overlaps both in two lines in the group (c), it thus illustrates the stairway scene structure. So, in the stairway scene, the number can prove the model's effects. In order to explain the performance clearly, we statistic the point number of the three scenes located by ContextDesc[35] and our SimpGeoDesc separately, and the result is shown in TABLE 4, from TABLE 4, we find there almost no overlaps points located by our modes in the group (a) and (b) comparing with ContextDesc [35], and the point number located by our model is nearly twice of the point number of ContextDesc [35]. Up to now, we claim that not only the matching key-point but also the camera location with feature descriptor both prove the effectiveness of our SimpGeoDesc model.

C. RELOCALIZATION BASED ON LOCAL FEATURE WITH SFM

In this part, we introduce the experiment environment and the relocalization results. In order to display the relocalization performance furthermore, we apply the relocalization on augmented reality (AR), so we also introduce the preliminary works of 3D dense reconstruction used in relocalization in AR in this section.

1) EXPERIMENT ENVIRONMENTS

To evaluate the performance of our relocalization method, we do many experiments, and at last, we select two types of classical scenes, an indoor scene (office) and an outdoor scene (technical park). The experiments prove that when the interval of time is 5 seconds, and the interval distance is 5 meters, the tracking fits the real road best. In the experiment, we have integrated the SLAM system into the mobile phone, and experiment results show that the system can support most cell phones on the current market. In this work, we select the Xiaomi 10 as the mobile set.



(a) The distant scene with differentiation (b) The distant scene without differentiation

FIGURE 19. The outdoors distant scenes.

TABLE 2. Evaluation of matching results for different models tested on the different indoor scenes.

Scene	Model	matching number	Inlier matching number	Inlier rate	Matching Time(S)
Scene 1 office	ContextDesc	120	93	0.78	0.91
	GeoDesc	180	165	0.92	0.33
	SimpGeoDesc	192	173	0.90	0.24
Scene 2 market	ContextDesc	105	80	0.76	0.87
	GeoDesc	140	125	0.89	0.30
	SimpGeoDesc	144	131	0.91	0.25
Scene 3 Roughcast house	ContextDesc	84	61	0.73	0.93
	GeoDesc	101	81	0.80	0.33
	SimpGeoDesc	92	75	0.82	0.25
Scene 4 stairway	ContextDesc	63	46	0.73	0.88
	GeoDesc	74	69	0.93	0.30
	SimpGeoDesc	87	78	0.90	0.23

On the other hand, we display the relocalization performance by applying it on AR, in the AR process, it utilizes the 3D dense reconstruction, and the dense reconstruction environment is ubuntu 18.04 with GTX 1080 i7-8700. 16GB. Before SLAM tracking, the 3D dense reconstruction has been done.

2) THE PRELIMINARY 3D RECONSTRUCTIONS

The relocalization is based on the SFM 3D reconstruction and in the AR application, we utilize the scenes' 3D dense reconstruction. So, before the relocalization experiments, we in advance reconstruct the 3D model, containing the SFM and 3D dense reconstruction.

a: SFM

Reconstructing the sparse point clouds with the classical SFM process but based on our feature descriptors extraction algorithm (SimpGeoDesc). The selected two scenes are indoor (FIGURE 10) and FIGURE 22 (outdoor), the reconstructed spare point clouds with SFM are shown in FIGURE 11 (indoor) and FIGURE 23 (outdoor). In FIGURE 11 and FIGURE 23, the red points compose the point clouds, and the yellow and colorful blocks are the images captured by the 4DKanKan Pro to reconstruct the point clouds. And in these images, colorful images are shot facing us and yellow images are shot back to us. In the two figures, we also provide the high-resolution images or magnify images of the green boxes shown in the following line in the same figure, which show the good performance of the SFM reconstruction with the extracted feature with SimpGeoDesc.

b: 3D DENSE RECONSTRUCTION

The dense reconstruction result is the digital model of a scene contrasting with sparse 3D reconstruction. The dense reconstruction process is shown in FIGURE 24 according to Cui [87]. In this paper, we mainly utilize the 3D dense model to AR application, and in order to prove our solution can suit

TABLE 3. Evaluation of matching results for different models tested on the different outdoor scenes.

Scene	Model	matching number	Inlier matching number	Inlier rate	Matching Time(S)
Scene1 near differ	ContextDesc	94	71	0.76	0.90
	GeoDesc	102	89	0.87	0.30
	SimpGeoDesc	101	90	0.89	0.25
Scene2 near similar	ContextDesc	107	80	0.75	0.94
	GeoDesc	123	103	0.84	0.33
	SimpGeoDesc	118	97	0.82	0.26
Scene3 distant differ	ContextDesc	72	50	0.69	0.91
	GeoDesc	81	70	0.86	0.32
	SimpGeoDesc	74	67	0.91	0.23
Scene4 distant similar	ContextDesc	80	57	0.71	0.92
	GeoDesc	83	70	0.84	0.30
	SimpGeoDesc	88	73	0.83	0.25

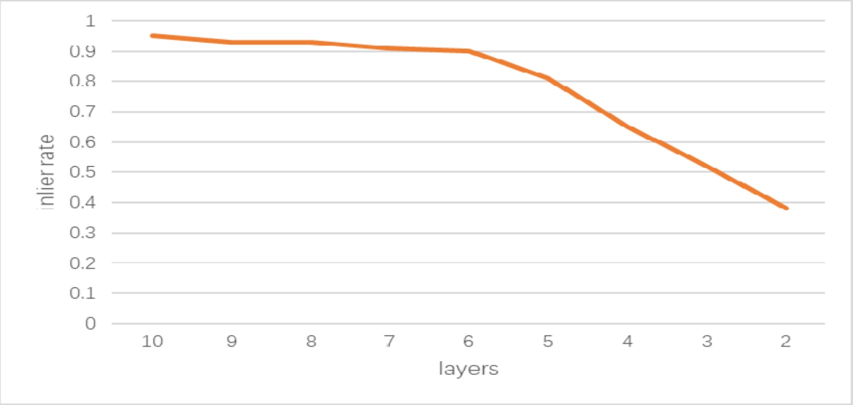
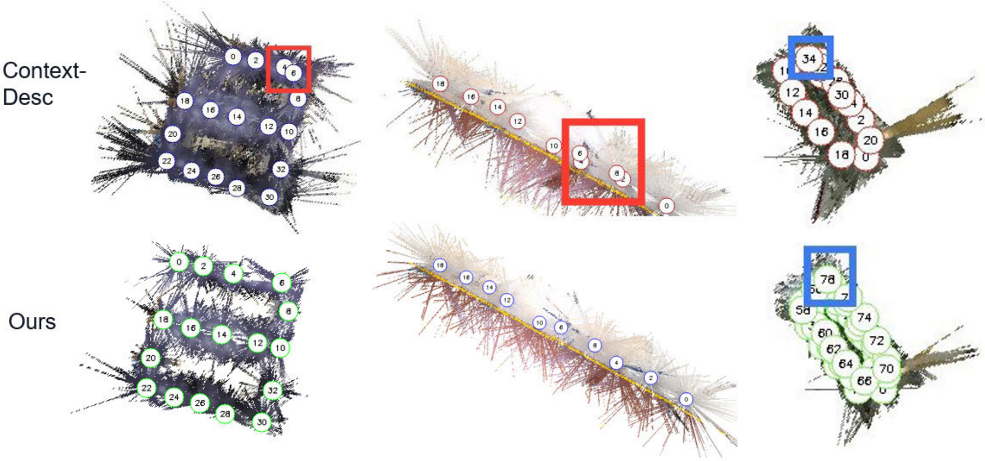


FIGURE 20. The inlier rates of different layer numbers.



(a) indoor office (b) outdoor (c) stairway

FIGURE 21. The camera location results with extracted descriptor by SimpGeoDesc (ours). From left to right are (a) indoor office scene, (b) outdoor scene and (c) stairway scene.

outdoor scene well, we choose the park (FIGURE 22) as the test scene. And the 3D dense reconstructions of the selected outdoor (FIGURE 22) is shown in FIGURE 25. Additionally, besides the depth estimation and surface reconstruction, we estimate the pose information firstly with SFM. When getting the dense point clouds, we fuse the dense point clouds with the camera poses obtained from SFM to get an effective 3D model.

TABLE 4. The location number with two models on the same scenes.

	Indoor office	outdoor	stairway
ContextDesc	16	8	18
SimpGeoDesc (ours)	17	10	40

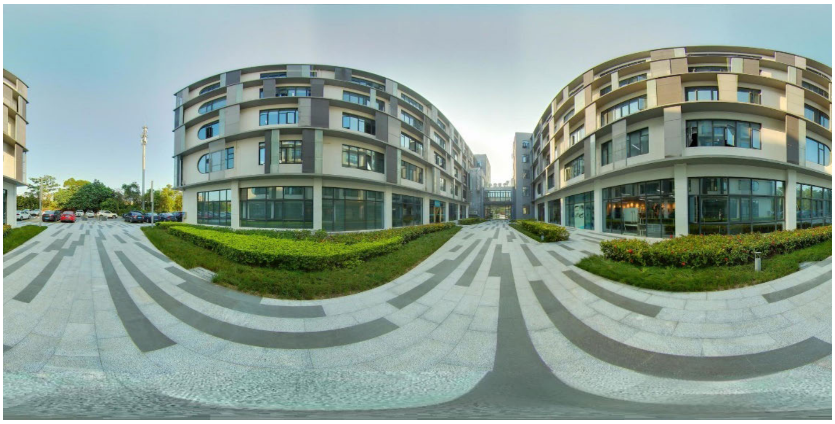


FIGURE 22. The outdoor scene (park).

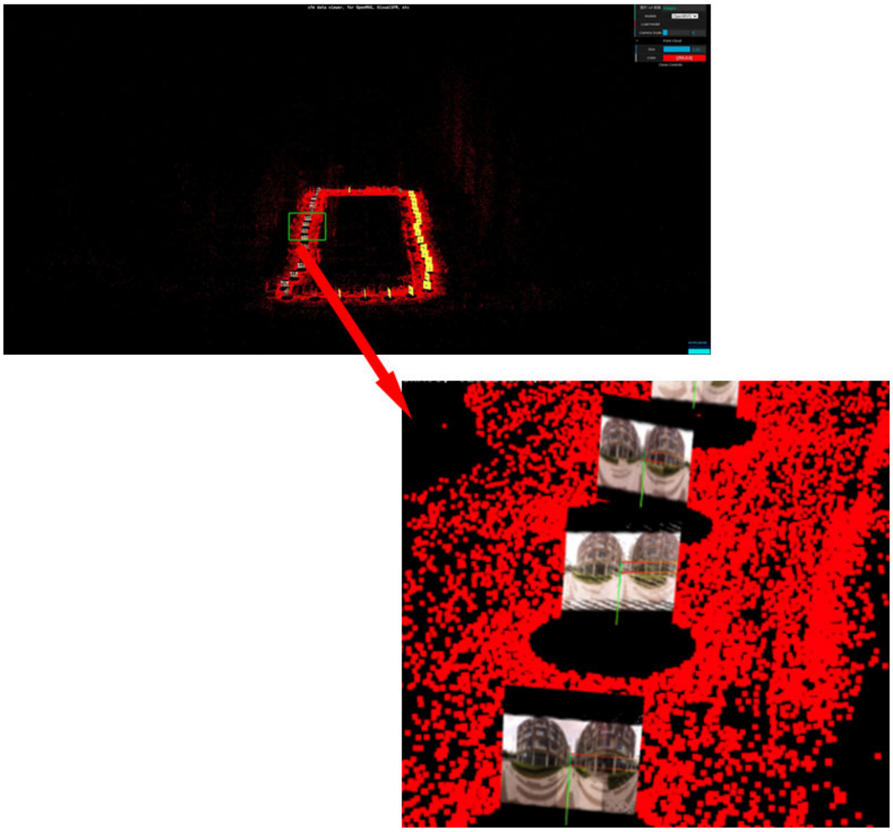


FIGURE 23. The SFM reconstruction of outdoor scene based on panoramic images.

3) RELOCALIZATION BASED ON SFM

a: EXPERIMENT RESULTS

With the SFM reconstruction point clouds obtained in advance and the extracted keyframe from the SLAM tracking, we match the key points of the keyframe and the point clouds of SFM to get the correct pose of the current keyframe. The relocalization results of the selected two scenes are shown in FIGURE 26 (indoor) and FIGURE 27 (outdoor).

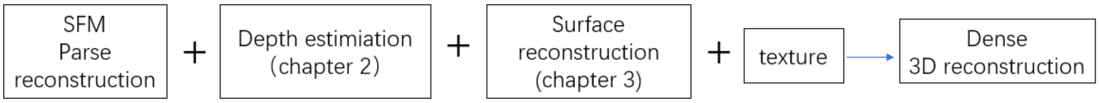


FIGURE 24. The process of dense 3D reconstruction of our solution.

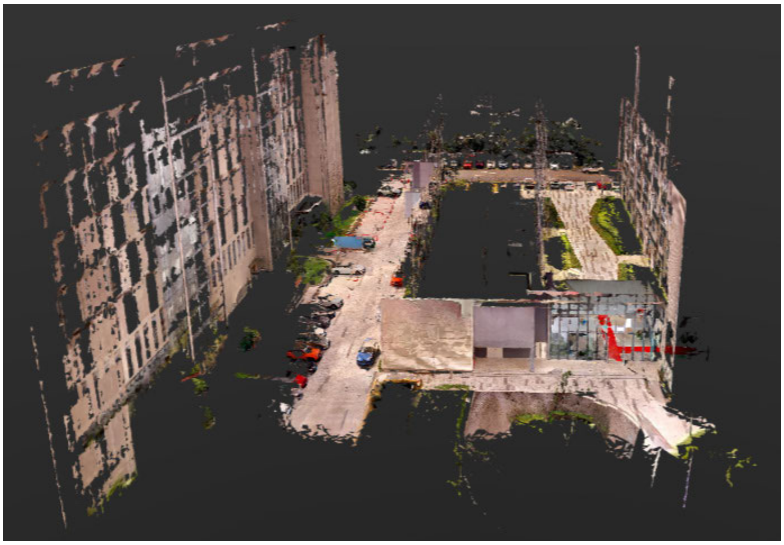


FIGURE 25. The 3D dense reconstruction model of outdoor scene (park).

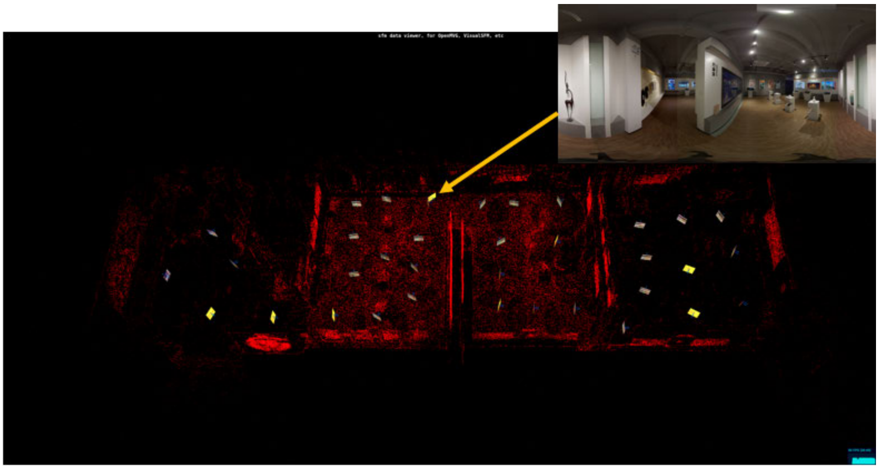


FIGURE 26. The indoor scene relocalization results based on a frame.

In FIGURE 26 and FIGURE 27, the red points are the sparse point clouds reconstructed with SFM, the colorful and yellow blocks are the images captured by 4DKanKan Pro [81] during SFM reconstruction, the image on the top right corner is the extracted keyframe from SLAM, and the high resolution of the point clouds in FIGURE 26 and FIGURE 27 are shown in FIGURE 11 and FIGURE 23 separately, (see the green box parts in FIGURE 11 and FIGURE 23). From FIGURE 26 and FIGURE 27, we find that the keyframes are relocalized to the sparse point clouds correctly both in the indoor and outdoor scenes.

b: COMPARATIONS

In order to illustrate the performance of our proposed relocalization solution, we compare our model with the VO-SF [70] on the outdoor scene, and the results are shown in FIGURE 28 and FIGURE 29, FIGURE 28 is the far point result and FIGURE 29 is the near point result. Both are the one relocalized pose from the frame (FIGURE 22), whose shoot view is some of left, so, when to relocalize, it matches the extracted frame with the point clouds come from the image captured at some of left view. In FIGURE 28 and FIGURE 29, the first line is the relocalization result of VO-SF [70], the second line

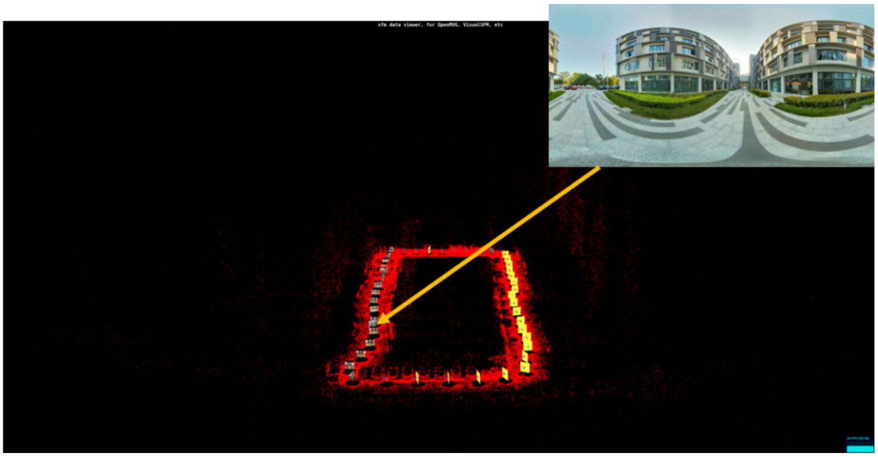


FIGURE 27. The outdoor scene relocalization results based on 3D model with an extracted frame (FIGURE 22).

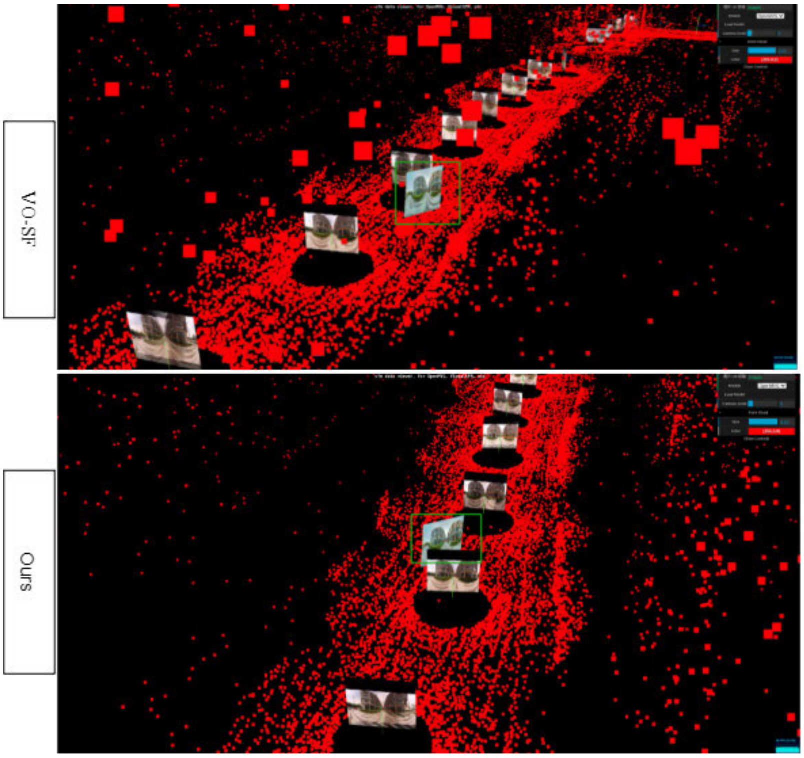


FIGURE 28. The relocalization results (far point) before and after based on the 3D model with extracted frame (FIGURE 22).

is the relocalization result of our solution. From FIGURE 28, in the second line, we find that the matched image's rotation angle is almost the same as the extracted frame (FIGURE 22) in our solution, what's more, the matched image's position is on the tracking of the SLAM. But in contrast, in the first line, the relocalization result of VO-SF [70], the rotation angle is large, nearly vertical direction, what's worse, it is offtrack from the SLAM tracking. The near point FIGURE 29 can prove these consequences more clearly. And the similar result in the indoor scenes. Additionally, plenty of experiments find that the average positional error and angular error is about 3∼5cm and 1 degree compared with 30∼50cm and 10 degrees error of before relocalization. Thus, we can draw the conclusion that our solution can relocalize the SLAM tracking well by reducing the drift (or accumulate errors) with the global information to reoptimize the back-end of SLAM.

4) AR APPLICATION WITH THE RELOCALIZATION OF SLAM TRACKING

a: APPLICATION ENVIRONMENT

To display the relocalization performance, besides the relocalization results, we also apply the relocalization in the AR

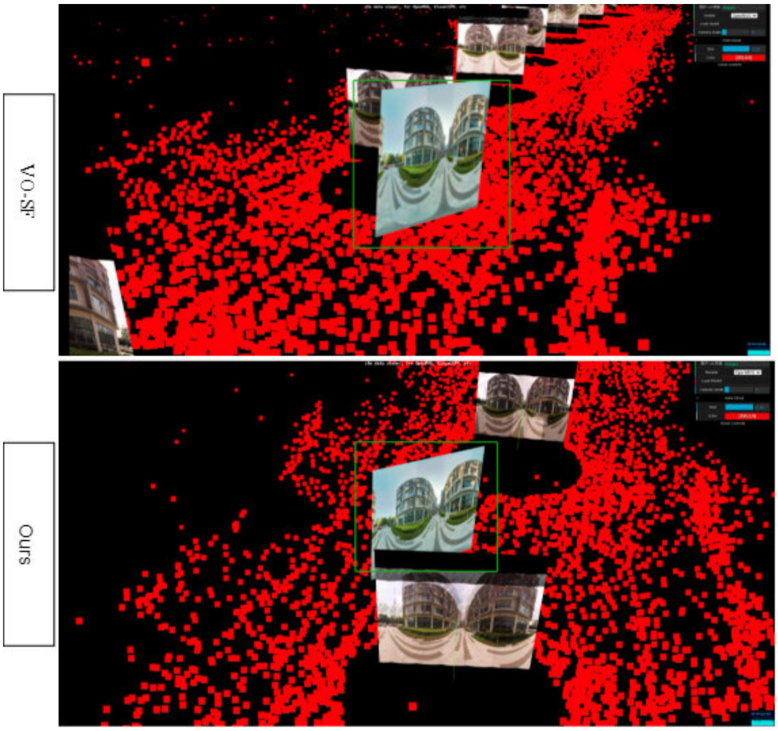
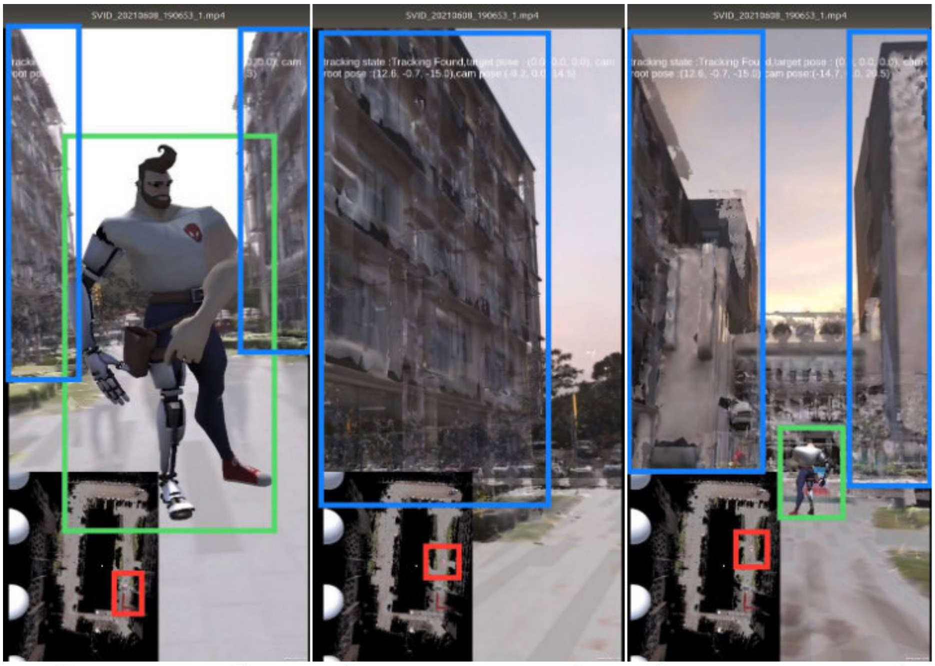


FIGURE 29. The relocalization results (near point) before and after based on the 3D model with extracted frame (FIGURE 22).


SVID_20210608_190653_1.mp4
SVID_20210608_190653_1.mp4
SVID_20210608_190653_1.mp4

(a) tracking point 1

(b) tracking point 2

(c) tracking point 3

FIGURE 30. The relocalization result based on SLAM with spherical camera.

application. From several experiments, we selected the AR effects of the outdoor scene (FIGURE 22). To implement the AR effect well, we have reconstructed the 3D sparse and dense model in the preliminary works. And every 5 seconds or 5 meters, we extract one keyframe from SLAM tracking.

b: EXPERIMENT AND THE RESULTS

The detail process is that it first extracts a keyframe from the SLAM tracking every 5 seconds or 5 meters and then extracts the key-point from the keyframe, after that, it matches those key-points of the keyframe and sparse point clouds to get the correct camera pose of the keyframe, finally, with the correct camera pose to amend the SLAM tracking whether there have been accumulated errors or not, which makes sure the SLAM tracking validity. In order to validate the effects of AR application with our proposed relocalization solution, we utilize it on the outdoor scene (FIGURE 22), and in the AR application, it renders a virtual robot placing in the reconstructed 3D model which fused with the real building nicely, and the results are shown in FIGURE 30, we take 3 SLAM tracking points to show in FIGURE 30: (a) the tracking point 1, (b) the tracking point 2, (c) the tracking point 3. In the figure, the red box is the moving camera (or mobile set), the green box is the virtual robot role rendered out, and the blue box is part of the dense 3D reconstruction of the scene in FIGURE 22. As the camera (or mobile set) moves, the view of robot changed naturally and the movements of the robot are fluent. And the whole virtual scene is stability during the movement process. The video of the SLAM tracking is available at https://github.com/qlchang/research.

c: ANALYZATIONS

From FIGURE 30, we find that when the camera (mobile set) moves (red rectangular), the virtual robot's (green rectangular) relative position is invariable, and the 3D model fused with the real scene (building) (blue rectangular) is almost a unified entity. What'more, from the vedio, we find that, even though the SLAM drifts, the relocalization can correct it. The results prove that our proposed relocalization solution is highly efficient and robust.

V. CONCLUSION

The proposed relocalization of SLAM tracking mainly contains three parts, the imaging model of spherical camera, the local feature extracting network SimpGeoDesc and the relocalization method based on 3D reconstruction. First, in the imaging model, we not only design a new spherical camera with eight fish-eye lenses but also propose a new calibration method to deal with the distortion according to the structure of the camera, and the experiment results prove that our calibration is applicative for our eight fish-eye lenses spherical camera, which provides the fundamental for the relocalization. Second, based on GeoDesc [34] and ContextDesc [35], we propose a local feature extraction model, which transfers the model trained on the perspective image dataset to panoramic image dataset, though its performance dropped a little compared with ContextDesc [35], and consistent with GeoDesc [34], it is much more computing effective compared with ContextDesc [35] and GeoDesc [34]. Finally, based on the effective SimpGeoDesc, we can get the correct point clouds and 3D reconstruction, and the point clouds provide the ground truth for relocalization. In order to prove the performance of our relocalization solution, we provide the "AR application", the stable and clear AR results proves the performance of our relocalization method again. Additionally, our relocalization solution can not only optimize the SLAM tracking, but also can be used in many scenes which need to location accurately, our solution has strong universality.

Though the performance of our proposed relocalization solution is good, there is still room to improve. In future work, we will first focus on the fundamental feature descriptor, and improve the accuracy computational efficiency of keypoints extraction and matching on panoramic images.

REFERENCES

[1] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "NetVLAD: CNN architecture for weakly supervised place recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 5297–5307.

[2] A. Torii, R. Arandjelovic, J. Sivic, M. Okutomi, and T. Pajdla, "24/7 place recognition by view synthesis," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1808–1817.

[3] V. Balntas, S. Li, and V. Prisacariu, "RelocNet: Continuous metric learning relocalisation using neural nets," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 751–767.

[4] Z. Laskar, I. Melekhov, S. Kalia, and J. Kannala, "Camera relocalization by computing pairwise relative poses using convolutional neural network," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops (ICCVW)*, Oct. 2017, pp. 929–938.

[5] S. Saha, G. Varma, and C. V. Jawahar, "Improved visual relocalization by discovering anchor points," 2018, *arXiv:1811.04370*.

[6] X.-S. Gao, X.-R. Hou, J. Tang, and H.-F. Cheng, "Complete solution classification for the perspective-three-point problem," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 8, pp. 930–943, Aug. 2003.

[7] L. Quan and Z. Lan, "Linear N-point camera pose determination," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 8, pp. 774–780, Aug. 1999.

[8] V. Lepetit, F. Moreno-Noguer, and P. Fua, "EP*n*P: An accurate O(*n*) solution to the P*n*P problem," *Int. J. Comput. Vis.*, vol. 81, no. 2, p. 155, 2009.

[9] A. Kendall, M. Grimes, and R. Cipolla, "PoseNet: A convolutional network for real-time 6-DOF camera relocalization," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 2938–2946.

[10] A. Kendall and R. Cipolla, "Modelling uncertainty in deep learning for camera relocalization," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2016, pp. 4762–4769.

[11] A. Kendall and R. Cipolla, "Geometric loss functions for camera pose regression with deep learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5974–5983.

[12] F. Walch, C. Hazirbas, L. Leal-Taixe, T. Sattler, S. Hilsenbeck, and D. Cremers, "Image-based localization using LSTMs for structured feature correlation," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 627–637.

[13] A. K. Dunne, J. Mallon, and P. F. Whelan, "Efficient generic calibration method for general cameras with single centre of projection," *Comput. Vis. Image Understand.*, vol. 114, no. 2, pp. 220–233, Feb. 2010.

[14] M. D. Grossberg and S. K. Nayar, "A general imaging model and a method for finding its parameters," in *Proc. 8th IEEE Int. Conf. Comput. Vis. (ICCV)*, vol. 2, Jul. 2001, pp. 108–115.

[15] F. Bergamasco, A. Albarelli, E. Rodola, and A. Torsello, "Can a fully unconstrained imaging model be applied effectively to central cameras?" in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 1391–1398.

[16] F. Bergamasco, L. Cosmo, A. Gasparetto, A. Albarelli, and A. Torsello, "Parameter-free lens distortion calibration of central cameras," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 3847–3855.

[17] S. Ramalingam and P. Sturm, "A unifying model for camera calibration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 7, pp. 1309–1319, Jul. 2016.

[18] J. Kannala and S. S. Brandt, "A generic camera model and calibration method for conventional, wide-angle, and fish-eye lenses," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 8, pp. 1335–1340, Aug. 2006.

[19] P. Fasogbon and L. Fan, "Generic calibration of cameras with non-parallel optical elements," in *Proc. 24th Int. Conf. Pattern Recognit. (ICPR)*, Aug. 2018, pp. 1875–1881.

[20] P. Fasogbon and E. Aksu, "Calibration of fisheye camera using entrance pupil," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2019, pp. 469–473.

[21] D. D. Lichti, D. Jarron, W. Tredoux, M. Shahbazi, and R. Radovanovic, "Geometric modelling and calibration of a spherical camera imaging system," *Photogramm. Rec.*, vol. 35, no. 170, pp. 123–142, Jun. 2020.

[22] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.

[23] H. Bay, T. Tuytelaars, and L. Van Gool, "SURF: Speeded up robust features," in *Proc. Eur. Conf. Comput. Vis.* Berlin, Germany: Springer, 2006, pp. 404–417.

[24] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 2564–2571.

[25] P. F. Alcantarilla and T. Solutions, "Fast explicit diffusion for accelerated features in nonlinear scale spaces," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 7, pp. 1281–1298, Oct. 2011.

[26] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 10, pp. 1615–1630, Oct. 2005.

[27] E. Simo-Serra, E. Trulls, L. Ferraz, I. Kokkinos, P. Fua, and F. Moreno-Noguer, "Discriminative learning of deep convolutional feature point descriptors," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 118–126.

[28] V. Balntas, E. Riba, D. Ponsa, and K. Mikolajczyk, "Learning local feature descriptors with triplets and shallow convolutional neural networks," in *Proc. Brit. Mach. Vis. Conf.*, 2016, vol. 1, no. 2, p. 3.

[29] Y. Tian, B. Fan, and F. Wu, "L2-Net: Deep learning of discriminative patch descriptor in Euclidean space," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 661–669.

[30] A. Mishchuk, D. Mishkin, F. Radenovic, and J. Matas, "Working hard to know your neighbor's margins: Local descriptor learning loss," 2017, *arXiv:1705.10872*.

[31] Y. Tian, X. Yu, B. Fan, F. Wu, H. Heijnen, and V. Balntas, "SOSNet: Second order similarity regularization for local descriptor learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 11016–11025.

[32] P. Ebel, E. Trulls, K. M. Yi, P. Fua, and A. Mishchuk, "Beyond Cartesian representations for local descriptors," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 253–262.

[33] M. Brown, G. Hua, and S. Winder, "Discriminative learning of local image descriptors," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 1, pp. 43–57, Jan. 2010.

[34] Z. Luo, T. Shen, L. Zhou, S. Zhu, R. Zhang, Y. Yao, T. Fang, and L. Quan, "GeoDesc: Learning local descriptors by integrating geometry constraints," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 168–183.

[35] Z. Luo, T. Shen, L. Zhou, J. Zhang, Y. Yao, S. Li, T. Fang, and L. Quan, "ContextDesc: Local descriptor augmentation with cross-modality context," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 2527–2536.

[36] Y. Verdie, K. M. Yi, P. Fua, and V. Lepetit, "TILDE: A temporally invariant learned DEtector," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 5279–5288.

[37] X. Zhang, F. X. Yu, S. Karaman, and S.-F. Chang, "Learning discriminative and transformation covariant local feature detectors," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6818–6826.

[38] N. Savinov, A. Seki, L. Ladicky, T. Sattler, and M. Pollefeys, "Quad-networks: Unsupervised learning to rank for interest point detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1822–1830.

[39] A. B. Laguna, E. Riba, D. Ponsa, and K. Mikolajczyk, "Key.Net: Keypoint detection by handcrafted and learned CNN filters," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 5836–5844.

[40] K. M. Yi, E. Trulls, V. Lepetit, and P. Fua, "LIFT: Learned invariant feature transform," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 467–483.

[41] H. Noh, A. Araujo, J. Sim, T. Weyand, and B. Han, "Large-scale image retrieval with attentive deep local features," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 3456–3465.

[42] D. DeTone, T. Malisiewicz, and A. Rabinovich, "SuperPoint: Self-supervised interest point detection and description," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2018, pp. 224–236.

[43] Y. Ono, E. Trulls, P. Fua, and K. M. Yi, "LF-Net: Learning local features from images," 2018, *arXiv:1805.09662*.

[44] 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 *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 8092–8101.

[45] J. Revaud, P. Weinzaepfel, C. De Souza, N. Pion, G. Csurka, Y. Cabon, and M. Humenberger, "R2D2: Repeatable and reliable detector and descriptor," 2019, *arXiv:1906.06195*.

[46] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient convolutional neural networks for mobile vision applications," 2017, *arXiv:1704.04861*.

[47] D. A. Suju and H. Jose, "FLANN: Fast approximate nearest neighbour search algorithm for elucidating human-wildlife conflicts in forest areas," in *Proc. 4th Int. Conf. Signal Process., Commun. Netw. (ICSCN)*, Mar. 2017, pp. 1–6.

[48] T. Sattler, B. Leibe, and L. Kobbelt, "Fast image-based localization using direct 2D-to-3D matching," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 667–674.

[49] T. Sattler, B. Leibe, and L. Kobbelt, "Efficient & effective prioritized matching for large-scale image-based localization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 9, pp. 1744–1756, Sep. 2016.

[50] Y. Li, N. Snavely, and D. P. Huttenlocher, "Location recognition using prioritized feature matching," in *Proc. Eur. Conf. Comput. Vis.* Berlin, Germany: Springer, 2010, pp. 791–804.

[51] L. Liu, H. Li, and Y. Dai, "Efficient global 2D-3D matching for camera localization in a large-scale 3D map," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2372–2381.

[52] S. Choudhary and P. Narayanan, "Visibility probability structure from SfM datasets and applications," in *Proc. Eur. Conf. Comput. Vis.* Berlin, Germany: Springer, 2012, pp. 130–143.

[53] P.-E. Sarlin, C. Cadena, R. Siegwart, and M. Dymczyk, "From coarse to fine: Robust hierarchical localization at large scale," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 12716–12725.

[54] E. Brachmann, A. Krull, S. Nowozin, J. Shotton, F. Michel, S. Gumhold, and C. Rother, "DSAC—Differentiable RANSAC for camera localization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6684–6692.

[55] E. Brachmann and C. Rother, "Learning less is more–6D camera localization via 3D surface regression," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4654–4662.

[56] T. Sattler, Q. Zhou, M. Pollefeys, and L. Leal-Taixe, "Understanding the limitations of CNN-based absolute camera pose regression," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3302–3312.

[57] J. Shotton, B. Glocker, C. Zach, S. Izadi, A. Criminisi, and A. Fitzgibbon, "Scene coordinate regression forests for camera relocalization in RGB-D images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 2930–2937.

[58] J. Valentin, M. Niebner, J. Shotton, A. Fitzgibbon, S. Izadi, and P. Torr, "Exploiting uncertainty in regression forests for accurate camera relocalization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 4400–4408.

[59] D. Massiceti, A. Krull, E. Brachmann, C. Rother, and P. H. S. Torr, "Random forests versus neural networks—What's best for camera localization?" in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 5118–5125.

[60] X. Li, J. Ylioinas, J. Verbeek, and J. Kannala, "Scene coordinate regression with angle-based reprojection loss for camera relocalization," in *Proc. Eur. Conf. Comput. Vis. (ECCV) Workshops*, 2018.

[61] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "MonoSLAM: Real-time single camera SLAM," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 6, pp. 1052–1067, Jun. 2007.

[62] G. Klein and D. Murray, "Parallel tracking and mapping for small AR workspaces," in *Proc. 6th IEEE ACM Int. Symp. Mixed Augmented Reality*, Nov. 2007, pp. 225–234.

[63] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM: A versatile and accurate monocular SLAM system," *IEEE Trans. Robot.*, vol. 31, no. 5, pp. 1147–1163, Oct. 2015.

[64] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras," *IEEE Trans. Robot.*, vol. 33, no. 5, pp. 1255–1262, Oct. 2017.

[65] A. Pumarola, A. Vakhitov, A. Agudo, A. Sanfeliu, and F. Moreno-Noguer, "PL-SLAM: Real-time monocular visual SLAM with points and lines," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 4503–4508.

[66] F.-A. Moreno, D. Zuñiga-Noël, D. Scaramuzza, and J. Gonzalez-Jimenez, "PL-SLAM: A stereo SLAM system through the combination of points and line segments," *IEEE Trans. Robot.*, vol. 35, no. 3, pp. 734–746, Jun. 2019.

[67] C. Campos, R. Elvira, J. J. G. Rodriguez, J. M. M. Montiel, and J. D. Tardos, "ORB-SLAM3: An accurate open-source library for visual, visual–inertial, and multimap SLAM," *IEEE Trans. Robot.*, early access, May 25, 2021, doi: 10.1109/TRO.2021.3075644.

[68] Liu, Guo, Feng, and Yang, "Accurate and robust monocular SLAM with omnidirectional cameras," *Sensors*, vol. 19, no. 20, p. 4494, Oct. 2019.

[69] S. Urban and S. Hinz, "MultiCol-SLAM—A modular real-time multi-camera SLAM system," 2016, *arXiv:1610.07336*.

[70] M. Jaimez, C. Kerl, J. Gonzalez-Jimenez, and D. Cremers, "Fast odometry and scene flow from RGB-D cameras based on geometric clustering," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 3992–3999.

[71] J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM: Large-scale direct monocular SLAM," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2014, pp. 834–849.

[72] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 3, pp. 611–625, Mar. 2017.

[73] C. Forster, M. Pizzoli, and D. Scaramuzza, "SVO: Fast semi-direct monocular visual odometry," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2014, pp. 15–22.

[74] F. Kangni and R. Laganiere, "Orientation and pose recovery from spherical panoramas," in *Proc. IEEE 11th Int. Conf. Comput. Vis.*, Oct. 2007, pp. 1–8.

[75] C. Geyer and K. Daniilidis, "Catadioptric camera calibration," in *Proc. 7th IEEE Int. Conf. Comput. Vis.*, vol. 1, Sep. 1999, pp. 398–404.

[76] S. Ikeda, T. Sato, and N. Yokoya, "High-resolution panoramic movie generation from video streams acquired by an omnidirectional multi-camera system," in *Proc. IEEE Int. Conf. Multisensor Fusion Integr. Intell. Syst. (MFI)*, Aug. 2003, pp. 155–160.

[77] J. A. Parian and A. Gruen, "A sensor model for panoramic cameras," in *Proc. 6th Opt. 3D Meas. Techn.*, Zurich, Switzerland, 2003, pp. 22–25.

[78] J. Li, X. Wang, and S. Li, "Spherical-model-based SLAM on full-view images for indoor environments," *Appl. Sci.*, vol. 8, no. 11, p. 2268, Nov. 2018.

[79] H. Seok and J. Lim, "ROVO: Robust omnidirectional visual odometry for wide-baseline wide-FOV camera systems," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 6344–6350.

[80] Y. Liu and J. Miura, "RDMO-SLAM: Real-time visual SLAM for dynamic environments using semantic label prediction with optical flow," *IEEE Access*, vol. 9, pp. 106981–106997, 2021.

[81] *4DKankan Pro*.

[82] L. Nan and P. Wonka, "PolyFit: Polygonal surface reconstruction from point clouds," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2353–2361.

[83] A. Geiger, F. Moosmann, O. Car, and B. Schuster, "Automatic camera and range sensor calibration using a single shot," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2012, pp. 3936–3943.

[84] S. Targ, D. Almeida, and K. Lyman, "Resnet in Resnet: Generalizing residual architectures," 2016, *arXiv:1603.08029*.

[85] Z. Zhang, "Flexible camera calibration by viewing a plane from unknown orientations," in *Proc. 7th IEEE Int. Conf. Comput. Vis.*, vol. 1, Sep. 1999, pp. 666–673.

[86] F. Barone, M. Marrazzo, and C. J. Oton, "Camera calibration with weighted direct linear transformation and anisotropic uncertainties of image control points," *Sensors*, vol. 20, no. 4, p. 1175, Feb. 2020.

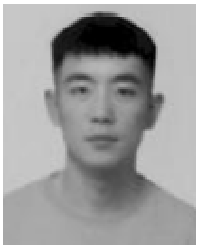
[87] Y. Cui, Q. Chang, Q. Liu, X. Yang, Y. Huang, S. Chen, F. Ren, and D. Stricker, "3D reconstruction with spherical cameras," *IEEE Access*, early access, Oct. 11, 2021, doi: 10.1109/ACCESS.2021.3119367.

QINGLING CHANG received the Ph.D. degree from the Institute of Computing Technology, Chinese Academy of Sciences, in 2015. She is currently a Master Supervisor and an Associate Professor at Wuyi University and the Sub Decanal of the China-German Artificial Intelligence Institute. Her research interests include artificial intelligence, computer vision, and knowledge graph.

QIANG LIU is currently a Senior Researcher with the China-German Artificial Intelligence Institute (CGAII) and a Master Supervisor with Sun Yat-sen University. In this article, he is mainly responsible for the system design. His main research interests include SLAM, SfM, and depth estimation.

XIN YANG received the B.S. degree in network engineering from the Guangdong Petrochemical College, in 2019. He is currently pursuing the master's degree in electronics and communication engineering with Wuyi University. His research interests include depth estimation and deep learning.

HUANG YAJIANG received the B.S. degree from Jilin University, in 2014, and the master's degree from Xi'an Jiaotong University, in 2017. His main research interests include SfM, SLAM, deep learning, and image enhancement.



FEI REN received the B.S. degree in mechanical design and manufacture and automation from the Hunan Institute of Traffic Engineering, in 2020. He is currently pursuing the master's degree in electronic message with Wuyi University.

YAN CUI is currently a Professor with the Faculty of Computer Science, Wuyi University, and the Dean of the Faculty of Intelligent Manufacturing, Wuyi Universtiy, and the China-Germany Artificial Intelligence Institute. His research interests include computer vision and computer graphic.