Neurocomputing 495 (2022) 11–25

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Fast and robust active camera relocalization in the wild for fine-grained change detection

Qian Zhang, Wei Feng *, Yi-Bo Shi, Di Lin

College of Intelligence and Computing, Tianjin University, Tianjin, China
Key Research Center for Surface Monitoring and Analysis of Relics, State Administration of Cultural Heritage, Tianjin, China

ARTICLE INFO

Article history:
Received 10 September 2021
Revised 7 January 2022
Accepted 23 April 2022
Available online 28 April 2022

Keywords:
Active camera relocalization (ACR)
Fine-grained change detection (FGCD)
Point cloud alignment
Cultural heritage
Preventive conservation

ABSTRACT

Active camera relocalization (ACR) is an important and challenging task, whose feasibility and success highly depend on illumination consistency and convergence speed. If under varied lighting conditions in outdoor scenes, however, both the convergence and accuracy of ACR cannot be guaranteed. In this paper, we propose a fast and robust ACR scheme, namely rACR, that works well under highly varied illuminations. To achieve robustness to lighting variations, rather than using 2D feature matching, we rely on 3D point clouds, acquired by a visual SLAM engine (VSE), to register the current and reference camera coordinate frames. We present a scale-aware point cloud matching function that is minimized by a two-stage coarse-to-fine method, i.e., fast alignment considering only geometric error at first, followed by fine-grained alignment optimizing both geometric, photometric errors and the poses of VSE keyframes. The two aligned point clouds with equalized scales help to bridge current and reference observations, avoiding 2D feature matching that are sensitive to large lighting variances, and can directly generate effective camera pose adjustments. Moreover, to achieve fast convergence speed, we implement the above algorithm with a parallel scheme, which is specifically composed of an initialization procedure and three parallel threads, i.e., VSE thread, pose alignment thread, and pose adjustment thread. Extensive experiments show that, rACR has much higher robustness to lighting variations and 5× faster convergence rate over state-of-the-art methods, thus significantly improves its feasibility in real-world fine-grained change detection tasks in the wild.

1. Introduction

Active camera relocalization (ACR) aims to physically relocalize the *current* camera to the same 6D pose, producing the *reference* single or multiple images [1], which is a challenging task and widely applicable to fine-grained change detection [2–4] and urban environment monitoring [5]. Among all these applications, fine-grained change detection (FGCD) [3], aiming to find minute changes of a high-value scene by comparing twice observations within a proper time interval, maybe the most related problem whose success critically depends on ACR.

So far, most state-of-the-art ACR methods [6,3,1] rely on 2D feature matching to estimate the relative camera pose. Once the lighting conditions change a lot between twice observations, their convergence and accuracy can easily be undermined. In this paper, we use FDF (feature-point displacement flow) and AFD (average feature-point displacement) [1] to evaluate the camera relocalization accuracy. FDF uses a set of matched pairs to represent relocalization accuracy, the arrows in FDF visually show the pose difference between current camera and the target one. AFD indicates the average length of all arrows in FDF. The smaller the AFD score, the more accurate the camera relocalization. Besides, camera relocalization is successful if the corresponding AFD is less than 3. As shown in Fig. 1(a), when the lighting is consistent (see the 1st and 3rd columns in the first row), the state-of-the-art ACR method [1] works well. In contrast, under significant lighting variance (1st and 2nd columns), ACR fails since lighting difference influences the performance of 2D feature matching, see the feature matching result in the second row of Fig. 1(a). Therefore, the accuracy and feasibility of ACR in real-world highly depend on *illumination consistency*. That is, state-of-the-art ACR method [1] cannot work well under highly varied illuminations (see Fig. 7 for the experiment results of ACR [1] under varied illuminations), which significantly limits their application in outdoor scenes, because there are so many high-value scenes, e.g., cultural heritages or vital equipment, exist in outdoor where the lighting conditions certainly cannot be controlled. As verified by Fig. 1(b), due to lacking

* Corresponding author.
E-mail address: wfeng@ieee.org (W. Feng).

https://doi.org/10.1016/j.neucom.2022.04.102
0925-2312/

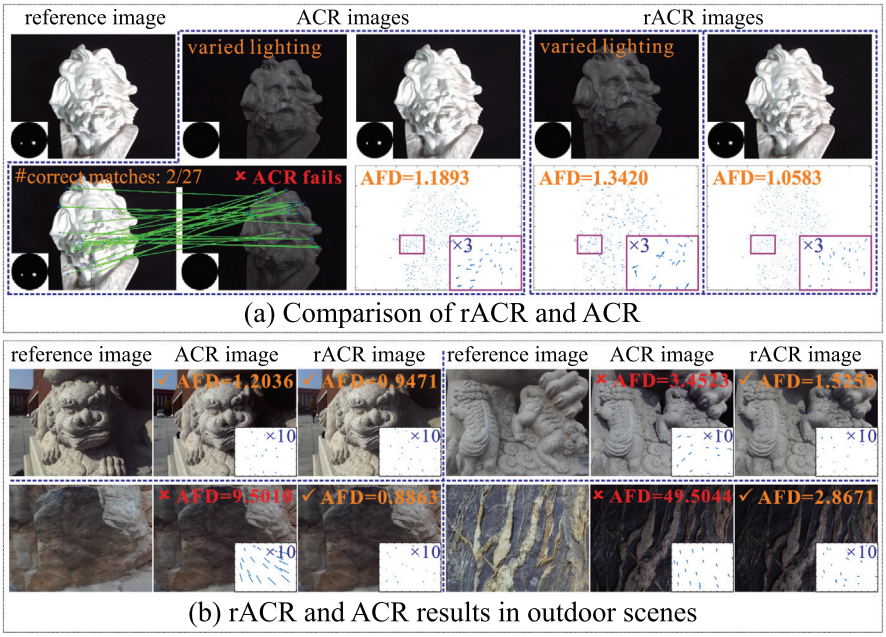


Fig. 1. (a) compares a state-of-the-art ACR method [1] and the proposed rACR under highly varied lightings (left) and similar lightings (right). Bottom-left subfigures are lighting probes indicating illumination distribution. Since highly varied illuminations lead to very different image appearances, 2D feature point matching inevitably fails, thus causing the ACR process to collapse; while rACR works well in both situations. (b) shows 4 examples of outdoor fine-grained change detection (FGCD) oriented ACR and rACR results. Under varied illuminations, ACR fails in 3 scenes (AFD $\geqslant$3). Note, we use AFD (average feature-point displacement) [1] to evaluate the camera relocalization accuracy. The smaller the AFD score, the more accurate the camera. relocalization.

robustness to lighting variations, there is no mature ACR solution to support FGCD of outdoor scenes with significant illumination variation between twice observations.

Besides, convergence speed is another critical factor for the feasibility of ACR. Since existing ACR methods cannot guarantee converge, i.e., easily fail, under highly varied illuminations, their speed in outdoor scenes is another issue.

In this paper, we propose a fast and robust ACR scheme, i,.e., rACR, that works well under highly varied illuminations, thus is much more feasible for outdoor scenes. The proposed rACR scheme relies on the correspondence of current and reference 3D point clouds, which can be easily acquired by a proper visual SLAM engine (VSE) [7] in real time, to register the two camera coordinate systems and circumvent 2D feature matching, thus avoids the negative effects of varied lightings to ACR. Specifically, the core of the proposed rACR is a scale-aware point cloud alignment method, which jointly considers both geometry, photometric alignment errors between current and reference point clouds, and the poses of corresponding VSE keyframes. We then present a two-stage coarse-to-fine optimization algorithm, *fast alignment* and *fine-grained alignment*, to effectively register twice observations and generate current camera adjustment motion. Besides, to achieve fast convergence speed, we present an efficient *algebraic estimation of real 3D scale factor* for 3D point cloud, avoiding the slow iterative bisection approaching process in the existing ACR method [1]. Finally, we compose a fast parallel rACR scheme, consisting of three parallel threads, i.e., the *adjustment thread*, *alignment thread*, *VSE thread*, and an initialization procedure, see Fig. 2 for details. Extensive experiments verify that the proposed rACR is able to averagely achieve 5× faster convergence speed and much better accuracy over the state-of-the-art ACR method [1] under varied illuminations. To our best knowledge, the proposed rACR, for the first time, successfully supports the outdoor FGCD task.

2. Related work

2.1. Active camera relocalization

Existing camera relocalization research can be divided into two categories [1], static camera relocalization (SCR) and active camera relocalization (ACR). SCR finds the closest image to the reference one from a large image database via appearance feature similarity [8,9], or use SfM and visual SLAM to register current camera of the input image into the global coordinate frame of an existing 3D scene [10]. Since SCR only conducts virtual camera alignment, state-of-the-art SCR methods can only achieve *cm*-level

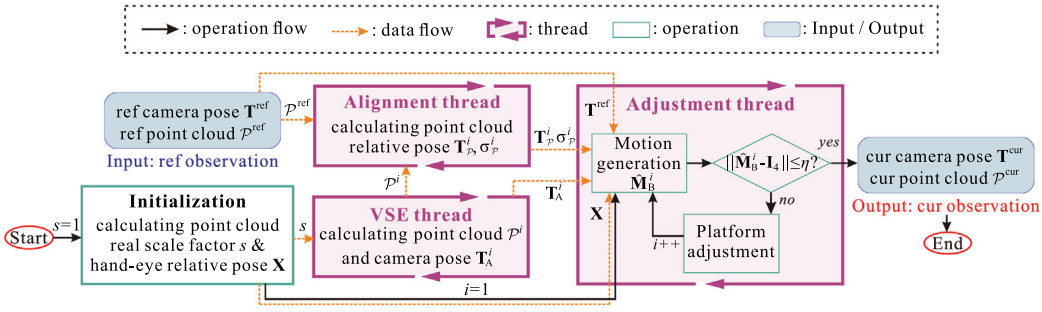


Fig. 2. Working flow of the proposed rACR scheme, which mainly consists of an initialization step and three parallel threads, i.e., the VSE thread, the alignment thread and the adjustment thread. See text for details.

translational relocalization accuracy and *degree*-level angular accuracy [11]. The second type is active camera relocalization (ACR), aiming to physically relocate current camera to the same pose of the reference image. Recently, using a precision robotic platform to adjust camera pose, Tian et al. [1] propose a hand-eye calibration free ACR algorithm from a single 2D reference image and achieve much higher 0.1 mm-level relocalization accuracy. Furthermore, based on the original ACR, depth camera based ACR [12] and ACR used in mobile device [13,6] are also proposed. Besides, low-cost hand-eye calibration [14,15] can effectively benefit the convergence of ACR. Except for camera pose, actively reproducing lighting condition has also been studied [16] recently. Despite the diversity and successes of previous SCR and ACR methods, their convergence highly depends on the consistent imaging conditions, without which the foundation of relocalization convergence and accuracy is undermined.

2.2. Visual SLAM

Simultaneous Localization and Mapping (SLAM) constructs the scene 3D map of an unknown environment and simultaneously acquires the camera pose in real time. RGB-based SLAM includes monocular SLAM [7] and stereo SLAM [17]. So far, successful RGBD-based SLAM systems are RGBD-SLAMv2 [18], ORB-SLAM2 [19], Stereo DSO [20], KinectFusion [21], Kintinuous [22], ElasticFusion [23] and BundleFusion [24]. The fusion-based SLAM methods focus on accurate 3D reconstruction than real-time camera pose estimation. Recently, the generalization power of convolution neural networks (CNN) has also been applied in visual SLAM [25]. In this paper, we use DSO SLAM [7] as the visual SLAM engine (VSE) in our rACR scheme.

2.3. 3D point cloud alignment

Our work is also highly related to 3D point cloud alignment. A typical workflow of point cloud alignment consists of global alignment and local refinement [26]. The former step computes an initial relative pose between two point clouds and the latter part further refines the initial estimation to obtain a better alignment. Classical global alignment methods find three pairs of corresponding points by RANSAC to align two point clouds [27,28]. However, they have worst case $O(n^3)$ time complexity. Later, Aiger et al. [29] propose a 4PCS algorithm to achieve a quadratic time complexity. Recently, based on 4PCS, a super4PCS method [30] with linear time complexity has been proposed. The most popular local alignment approaches are Iterative Closest Points (ICP) [31] and its variants [32,33]. To further improve the robustness of ICP algorithm, Bouaziz et al. [34] introduce a sparse ICP formulation to deal with outliers and incomplete data.

2.4. 2D and 3D feature representation

Feature representation for 2D or 3D data plays an important role in many computer vision tasks. In the aspect of 2D feature representation, previous learning based methods replace the descriptor [35] or detector [36] with a learnable alternative. After that, Yi et al. [37] first propose a fully learning based architecture to jointly solve description and detection problems. Luo et al. [38] introduce deformable convolution and local transformation to further encourage keypoints to be reliable and repeatable. Besides, some methods [39,40] employ attention mechanism for keypoint selection to improve the robustness for challenging situations such as background clutter, partial occlusion. Chen et al. [41] propose a separation training scheme to improve the matching accuracy under varied illuminations. As for 3D feature representation, early methods [42] extract 3D keypoint description from multi-view images. In contrast, Gojcic et al. [43] construct descriptors by converting 3D patches into smoothed density value representations. Liu et al. [44] propose a new RS-CNN to learn the geometric topology constraint among points, which leads to much shape awareness and robustness. Recently, Qiu et al. [45] try to exploit 3D point features from both the geometric and semantic information, which achieves excellent performance for semantic segmentation task.

3. Fast and robust ACR in the wild

3.1. Problem formulation and overview

Camera pose (or rigid body motion) $\mathbf{T} \in \mathrm{SE}(3)$ can be expressed as a 3D rotation $\mathbf{R} \in \mathrm{SO}(3)$ and a 3D translation $\mathbf{t} \in \mathbb{R}^3$, i.e., $\mathbf{T} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^{\mathrm{T}} & 1 \end{bmatrix}$. Following the notations in [1], we use a two-tuple form, i.e., $\mathbf{T} \simeq \langle \mathbf{R}, \mathbf{t} \rangle$, instead of the above representation in the rest of the paper for simplicity. We consider the illumination is *stable* during once observation[1], thus we can obtain an effective 3D point cloud by a proper visual SLAM engine (VSE). As shown in Fig. 3, we have $\mathbf{M}_{\mathrm{A}}\mathbf{X} = \mathbf{X}\mathbf{M}_{\mathrm{B}}$, where $\mathbf{X} \simeq \langle \mathbf{R}_{\mathrm{X}}, \mathbf{t}_{\mathrm{X}} \rangle$ is the hand-eye relative pose, $\mathbf{M}_{\mathrm{A}}$ and $\mathbf{M}_{\mathrm{B}}$ denote the motions of eye (camera) and hand (robotic platform), respectively. Let the reference 3D point cloud space be the world coordinate system. We seek the optimal hand motion $\hat{\mathbf{M}}_{\mathrm{B}}$ and physically execute it to actively relocalize the camera from initial pose $\tilde{\mathbf{T}}_{\mathrm{A}}^{0}$ to the reference pose $\mathbf{T}_{\mathrm{A}}^{\mathrm{ref}} \simeq \langle \mathbf{R}_{\mathrm{A}}^{\mathrm{ref}}, \mathbf{t}_{\mathrm{A}}^{\mathrm{ref}} \rangle$. Note, $\tilde{\mathbf{T}}_{\mathrm{A}}^{0}$ is the initial camera pose of current observation within the world coordinate system, and $\mathbf{T}_{\mathrm{A}}^{0}$ is the corresponding pose of $\tilde{\mathbf{T}}_{\mathrm{A}}^{0}$ within current 3D point cloud space. Thus, the rACR problem can be formulated as

$$\hat{\mathbf{M}}_{\mathrm{B}} = \arg\min_{\mathbf{M}_{\mathrm{B}}} \|\mathbf{X}\mathbf{M}_{\mathrm{B}}\mathbf{X}^{-1}\tilde{\mathbf{T}}_{\mathrm{A}}^{0} - \mathbf{T}_{\mathrm{A}}^{\mathrm{ref}}\|_{\mathrm{F}}^{2}, \tag{1}$$

where $\|\cdot\|_{\mathrm{F}}$ is the Frobenius norm. Generally, execute $\hat{\mathbf{M}}_{\mathrm{B}}$ in one shot to achieve the goal is impractical because of the platform's mechanical error or motion's calculation error. A practical scheme is to find a series of camera adjustment motions to asymptotically approach $\hat{\mathbf{M}}_{\mathrm{B}}$. Thus, we have

$$\begin{aligned} \hat{\mathbf{M}}_{\mathrm{B}}^{i} &= \arg\min_{\mathbf{M}_{\mathrm{B}}^{i}} \|\mathbf{X}\mathbf{M}_{\mathrm{B}}^{i}\mathbf{X}^{-1}\tilde{\mathbf{T}}_{\mathrm{A}}^{i} - \mathbf{T}_{\mathrm{A}}^{\mathrm{ref}}\|_{\mathrm{F}}^{2}, \\ \hat{\mathbf{M}}_{\mathrm{B}} &= \prod_{i=h}^{0} \hat{\mathbf{M}}_{\mathrm{B}}^{i}, \end{aligned} \tag{2}$$

where h is the camera adjustment times, $\tilde{\mathbf{T}}_{\mathrm{A}}^{i} \simeq \langle \tilde{\mathbf{R}}_{\mathrm{A}}^{i}, \tilde{\mathbf{t}}_{\mathrm{A}}^{i} \rangle$ and $\hat{\mathbf{M}}_{\mathrm{B}}^{i}$ denote the current camera pose in world coordinate system and optimal hand motion of the ith adjustment, respectively. Let $\mathbf{T}_{\mathrm{A}}^{i} \simeq \langle \mathbf{R}_{\mathrm{A}}^{i}, \mathbf{t}_{\mathrm{A}}^{i} \rangle$ be the current camera pose in current point cloud space. Let $\sigma_{\mathcal{P}}^{i}$ and $\mathbf{T}_{\mathcal{P}}^{i}$ be the relative scale factor and relative pose between reference point cloud $\mathcal{P}^{\mathrm{ref}}$ and current point clouds $\mathcal{P}^{i}$, respectively. Then, we have $\tilde{\mathbf{T}}_{\mathrm{A}}^{i} = \mathbf{T}_{\mathcal{P}}^{i} \begin{bmatrix} {\mathbf{R}_{\mathrm{A}}^{i}}^{-1} & -\sigma_{\mathcal{P}}^{i}{\mathbf{R}_{\mathrm{A}}^{i}}^{-1}\mathbf{t}_{\mathrm{A}}^{i} \\ \mathbf{0}^{\mathrm{T}} & 1 \end{bmatrix}$. Hence, refer to Eq. (2), the optimal hand motion $\hat{\mathbf{M}}_{\mathrm{B}}^{i}$ can be expressed as

$$\hat{\mathbf{M}}_{\mathrm{B}}^{i} = \mathbf{X}^{-1}\mathbf{T}_{\mathrm{A}}^{\mathrm{ref}} \begin{bmatrix} {\mathbf{R}_{\mathrm{A}}^{i}}^{-1} & -\sigma_{\mathcal{P}}^{i}{\mathbf{R}_{\mathrm{A}}^{i}}^{-1}\mathbf{t}_{\mathrm{A}}^{i} \\ \mathbf{0}^{\mathrm{T}} & 1 \end{bmatrix}^{-1} (\mathbf{T}_{\mathcal{P}}^{i})^{-1}\mathbf{X}. \tag{3}$$

[1] We consider the illumination is stable iff the VSE can work well. In fact, for outdoor scenes, except for the dawn and nightfall, the illumination of a day is generally stable and changes slowly within a short time period.

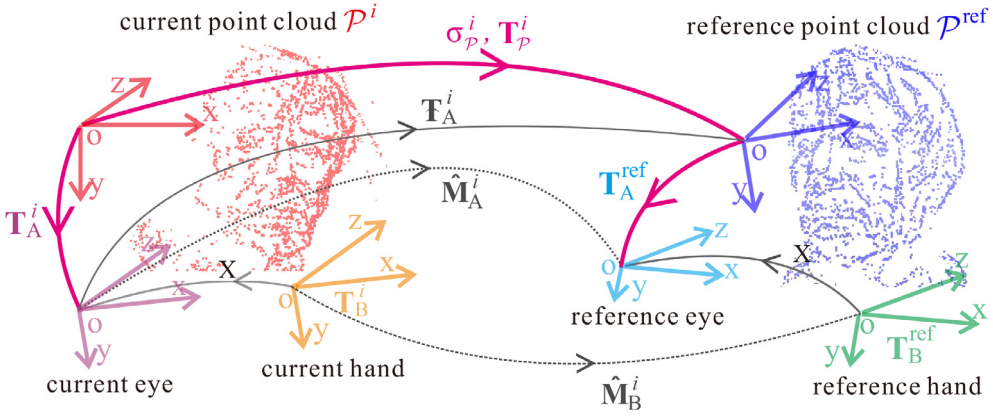


Fig. 3. Illustration of the key notations of rACR.

Fig. 2 shows the overview of the proposed rACR scheme. The input of rACR is the reference observation (point cloud $\mathcal{P}^{\text{ref}}$ and camera pose $\mathbf{T}_{\text{A}}^{\text{ref}}$). The output of rACR is the relocalization result, i.e., current observation (point cloud $\mathcal{P}^{\text{cur}}$ and camera pose $\mathbf{T}_{\text{A}}^{\text{cur}}$). The proposed rACR scheme consists of an initialization step and three parallel threads, i.e., the VSE thread, the alignment thread and the adjustment thread. The initialization step calculates the hand-eye relative pose $\mathbf{X}$ and real scale factor s of current point cloud, since the point cloud acquired from monocular VSE has no real scale. The VSE thread generates the latest current point cloud $\mathcal{P}$ and camera pose $\mathbf{T}_{\text{A}}$ in real time. The alignment thread focuses on high-quality scale-aware point cloud alignment, which calculates the relative pose $\mathbf{T}_{\mathcal{P}}$ and relative scale factor $\sigma_{\mathcal{P}}$ between $\mathcal{P}$ and $\mathcal{P}^{\text{ref}}$. The adjustment thread generates motion $\hat{\mathbf{M}}_{\text{B}}^i$ by Eq. (3) and execution it by robotic platform for ith adjustment.

3.2. Initialization

Let $\mathbf{M}_{\text{A}} \simeq \langle \mathbf{R}_{\text{A}}, \mathbf{t}_{\text{A}} \rangle$ and $\mathbf{M}_{\text{B}} \simeq \langle \mathbf{R}_{\text{B}}, \mathbf{t}_{\text{B}} \rangle$. We divide $\mathbf{M}_{\text{A}}\mathbf{X} = \mathbf{X}\mathbf{M}_{\text{B}}$ into rotation and translation components,

$$\mathbf{R}_{\text{A}}\mathbf{R}_{\text{X}} = \mathbf{R}_{\text{X}}\mathbf{R}_{\text{B}}, \tag{4}$$

$$\mathbf{t}_{\text{A}} + (\mathbf{R}_{\text{A}} - \mathbf{I}_3)\mathbf{t}_{\text{X}} = \mathbf{R}_{\text{X}}\mathbf{t}_{\text{B}}, \tag{5}$$

where $\mathbf{I}_3$ is 3×3 identity matrix. Note, $\mathbf{R}_{\text{A}}$ (acquired from VSE), $\mathbf{R}_{\text{B}}$ and $\mathbf{t}_{\text{B}}$ (acquired from robotic platform) are all known. Hence, using classical hand-eye calibration method [14], we can first solve $\mathbf{R}_{\text{X}}$ through at least twice platform movements. For a monocular VSE, the calculated camera translation $\mathbf{t}_{\text{VSE}}$ usually has no real scale. Let s be the real scale factor, then we have $\mathbf{t}_{\text{A}} = s\mathbf{t}_{\text{VSE}}$. Hence, Eq. (5) can be rewritten as

$$(\mathbf{R}_{\text{A}} - \mathbf{I}_3)\mathbf{t}_{\text{X}} = -s\mathbf{t}_{\text{VSE}} + \mathbf{R}_{\text{X}}\mathbf{t}_{\text{B}}. \tag{6}$$

Since $\mathbf{R}_{\text{A}}$ is a 3D rotation matrix, $\text{rank}(\mathbf{R}_{\text{A}} - \mathbf{I}_3) \leq 2$ [14]. There exists vector $\mathbf{k} = [k_1, k_2, k_3]^{\text{T}}$ that satisfies: 1) $\mathbf{k} \neq \mathbf{0}$, and 2) $\mathbf{k}^{\text{T}}(\mathbf{R}_{\text{A}} - \mathbf{I}_3) = \mathbf{0}^{\text{T}}$. We left multiply $\mathbf{k}^{\text{T}}$ on both sides of Eq. (6), yielding

$$\mathbf{k}^{\text{T}}(\mathbf{R}_{\text{A}} - \mathbf{I}_3)\mathbf{t}_{\text{X}} = \mathbf{k}^{\text{T}}(-s\mathbf{t}_{\text{VSE}} + \mathbf{R}_{\text{X}}\mathbf{t}_{\text{B}}). \tag{7}$$

According to Eq. (7) and $\mathbf{k}^{\text{T}}(\mathbf{R}_{\text{A}} - \mathbf{I}_3) = \mathbf{0}^{\text{T}}$, we have $\mathbf{k}^{\text{T}}(-s\mathbf{t}_{\text{VSE}} + \mathbf{R}_{\text{X}}\mathbf{t}_{\text{B}}) = 0$. Therefore,

$$s = \frac{\mathbf{k}^{\text{T}}\mathbf{t}_{\text{VSE}}}{\mathbf{k}^{\text{T}}(\mathbf{R}_{\text{X}}\mathbf{t}_{\text{B}})}. \tag{8}$$

That is, after solving $\mathbf{R}_{\text{X}}$, we can faithfully calculate s by Eq. (8). Then, through at least twice platform movements, we can easily solve $\mathbf{t}_{\text{X}}$ using Eq. (6).

3.3. VSE thread

As shown in Fig. 2, let $\mathbf{T}_{\text{VSE}} \simeq \langle \mathbf{R}_{\text{VSE}}, \mathbf{t}_{\text{VSE}} \rangle$ and $\mathcal{P}_{\text{VSE}}$ be the camera pose and point cloud taken from VSE respectively, where $\mathcal{P}_{\text{VSE}}$ and $\mathbf{T}_{\text{VSE}}$ have no real scale. Given the real scale factor s acquired from the initialization step, we have $\mathcal{P} = s\mathcal{P}_{\text{VSE}}$ and $\mathbf{T}_{\text{A}} \simeq \langle \mathbf{R}_{\text{A}}, \mathbf{t}_{\text{A}} \rangle = \langle \mathbf{R}_{\text{VSE}}, s\mathbf{t}_{\text{VSE}} \rangle$, where $\mathbf{T}_{\text{A}}$ and $\mathcal{P}$ are the camera pose and point cloud with real scale. In a word, the VSE thread maintains the latest current point cloud $\mathcal{P}$ and camera pose $\mathbf{T}_{\text{A}}$ in real time. In fact, it is not that any VSE would be suited to our rACR. Generally, existing VSEs can be divided into direct VSE and indirect VSE, respectively depending on photometric constraints and 2D feature matchings to reconstruct 3D point cloud. Direct VSE can generate semi-dense 3D point cloud but it is not robust to camera's rapid motion. In contrast, indirect VSE works well under camera's rapid motion but generates sparse 3D point cloud. For the rACR problem, first, we usually use a robotic platform to adjust camera, so camera's rapid motion usually would not happen. Second, semi-dense 3D point cloud encodes more scene structural information and it is obviously more conducive to 3D alignment than the sparse one. Hence, we use a state-of-the-art direct VSE, DSO [7], in this paper. Refer to the experiment part for the quantitative comparisons of direct VSE and indirect VSE to rACR accuracy.

3.4. Alignment thread

The alignment thread aims to calculate the relative pose $\mathbf{T}_{\mathcal{P}}$ and relative scale factor $\sigma_{\mathcal{P}}$ between reference and current point cloud. The essence is to solve a scale-aware point cloud alignment problem.

3.4.1. Scale-aware point cloud alignment

We formulate the scale-aware point cloud alignment as an optimization problem considering both geometric and photometric alignment errors. The geometric alignment error E_{g} can be written as

$$\text{E}_{\text{g}} = \sum_m (\|\mathbf{R}_{\mathcal{P}}(\sigma_{\mathcal{P}}\mathbf{p}_{\text{N}(m)}) + \mathbf{t}_{\mathcal{P}} - \mathbf{p}_m^{\text{ref}}\|_2^2, \tag{9}$$

where $\mathbf{p}_m^{\text{ref}}$ indicates the coordinate of mth 3D point in $\mathcal{P}^{\text{ref}}$, $\text{N}(m)$ denotes the index of the nearest 3D point of $\mathbf{p}_m^{\text{ref}}$ in current point cloud $\mathcal{P}$, $\mathbf{p}_{\text{N}(m)}$ is the coordinate of $\text{N}(m)$th 3D point in $\mathcal{P}$. On the other hand, VSE maintains multiple keyframes. For each 3D point in $\mathcal{P}^{\text{ref}}$, the intensities of its 2D projection pixels in all current keyframes should be the same. Let $\mathbf{T}_{\text{K}}^u \simeq \langle \mathbf{R}_{\text{K}}^u, \mathbf{t}_{\text{K}}^u \rangle$ be the camera pose of uth current keyframe image $\mathbf{D}^u$, then we have $\tilde{\mathbf{T}}_{\text{K}}^u = \mathbf{T}_{\mathcal{P}}^i \begin{bmatrix} \mathbf{R}_{\text{K}}^u & \sigma_{\mathcal{P}}^i \mathbf{t}_{\text{K}}^u \\ \mathbf{0}^{\text{T}} & 1 \end{bmatrix}$,

where $\tilde{\mathbf{T}}_{\mathrm{K}}^{u}$ is the camera pose of uth current keyframe in the world coordinate system. Hence, we can define the photometric alignment error $\mathrm{E_p}$ as

$$\mathrm{E_p} = \frac{1}{|\mathcal{K}|^2} \sum_{\substack{m \\ u,v \in \mathcal{K}}} \|\mathbf{D}^u(\pi(\tilde{\mathbf{T}}_{\mathrm{K}}^{u}, \mathbf{p}_m^{\mathrm{ref}})) - \mathbf{D}^v(\pi(\tilde{\mathbf{T}}_{\mathrm{K}}^{v}, \mathbf{p}_m^{\mathrm{ref}}))\|_\delta, \tag{10}$$

where $\|\cdot\|_\delta$ is the Huber norm, $\mathcal{K}$ indicates the index set of all keyframes, $|\mathcal{K}|$ denotes the keyframe number, $|\mathcal{K}|^2$ is a normalization factor, $\pi(\tilde{\mathbf{T}}_{\mathrm{K}}^{u}, \mathbf{p}_m^{\mathrm{ref}})$ indicates the corresponding 2D projection of 3D point $\mathbf{p}_m^{\mathrm{ref}}$ under camera pose $\tilde{\mathbf{T}}_{\mathrm{K}}^{u}$. Then, the scale-aware point cloud alignment function should be

$$\{\hat{\sigma}_{\mathcal{P}}, \hat{\mathbf{R}}_{\mathcal{P}}, \hat{\mathbf{t}}_{\mathcal{P}}\} = \arg\min_{\{\sigma_{\mathcal{P}}, \mathbf{R}_{\mathcal{P}}, \mathbf{t}_{\mathcal{P}}\}} \alpha \mathrm{E_g} + (1-\alpha)\mathrm{E_p}, \tag{11}$$

where α is the term weight. Directly solving Eq. (11) needs nonlinear optimization, which is time-consuming and easily falls into bad local minima if the initialization is not good. To solve this problem, we present an effective two-stage optimization strategy to conduct *fast alignment* at first, followed by *fine-grained alignment* when both poses and scales of twice observations are almost registered.

3.4.2. Fast alignment

we first carry out fast alignment by only optimizing the geometric alignment error to achieve quick registration of reference and current point clouds,

$$\{\hat{\sigma}_{\mathcal{P}}, \hat{\mathbf{R}}_{\mathcal{P}}, \hat{\mathbf{t}}_{\mathcal{P}}\} = \arg\min_{\{\sigma_{\mathcal{P}}, \mathbf{R}_{\mathcal{P}}, \mathbf{t}_{\mathcal{P}}\}} \mathrm{E_g}. \tag{12}$$

Specifically, we adopt a two-step strategy to alternately optimize the relative pose $\{\hat{\mathbf{R}}_{\mathcal{P}}, \hat{\mathbf{t}}_{\mathcal{P}}\}$ and scale factor $\hat{\sigma}_{\mathcal{P}}$,

$$\textbf{step1} : \{\hat{\mathbf{R}}_{\mathcal{P}}, \hat{\mathbf{t}}_{\mathcal{P}}\} = \arg\min_{\{\mathbf{R}_{\mathcal{P}}, \mathbf{t}_{\mathcal{P}}\}} \mathrm{E_g}, \tag{13}$$

$$\textbf{step2} : \hat{\sigma}_{\mathcal{P}} = \arg\min_{\sigma_{\mathcal{P}}} \mathrm{E_g}. \tag{14}$$

Note, Eq. (13) is a classical point cloud alignment problem that can be solved by the sparse ICP algorithm [34]. Besides, Eq. (14) can be directly solved in closed-form,

$$\hat{\sigma}_{\mathcal{P}} = \frac{\sum_{m=1} (\hat{\mathbf{R}}_{\mathcal{P}} \mathbf{p}_{\mathrm{N}(m)})(\mathbf{p}_m^{\mathrm{ref}} - \hat{\mathbf{t}}_{\mathcal{P}})^{\mathrm{T}}}{\sum_{m=1} (\hat{\mathbf{R}}_{\mathcal{P}} \mathbf{p}_{\mathrm{N}(m)})(\hat{\mathbf{R}}_{\mathcal{P}} \mathbf{p}_{\mathrm{N}(m)})^{\mathrm{T}}}. \tag{15}$$

We repeat such two-step optimization process until $||\triangle \mathbf{T}_{\mathcal{P}} - \mathbf{I}_4|| \leqslant \tau_{\mathbf{T}}$ and $\triangle\sigma_{\mathcal{P}} \leqslant \tau_\sigma$, where $\mathbf{I}_4$ is a 4×4 identity matrix, $\triangle\mathbf{T}_{\mathcal{P}}$ and $\triangle\sigma_{\mathcal{P}}$ denote the changes of $\mathbf{T}_{\mathcal{P}}$ and $\sigma_{\mathcal{P}}$ between two contiguous iterations, respectively. In our experiments, we set $\tau_{\mathbf{T}} = 10^{-4}$ and $\tau_\sigma = 10^{-5}$. Since both the above two steps can be solved in closed-form, $\sigma_{\mathcal{P}}$ and $\mathbf{T}_{\mathcal{P}}$ can quickly converge to a reasonably good solution.

3.4.3. Fine-grained alignment

After fast alignment, current and reference point clouds are almost registered. Using the calculated $\hat{\sigma}_{\mathcal{P}}$ and $\hat{\mathbf{T}}_{\mathcal{P}}$ (i.e., $\hat{\mathbf{R}}_{\mathcal{P}}, \hat{\mathbf{t}}_{\mathcal{P}}$) by fast alignment as the initial value, we can conduct fine-grained alignment by optimizing the full scale-aware point cloud alignment function, Eq. (11). In fact, to achieve finer scale alignment, the small estimation error of camera poses $\mathbf{T}_{\mathrm{K}}^{u}$ of current keyframes ($1 \leqslant u \leqslant |\mathcal{K}|$) done by the VSE thread, may still influence the alignment accuracy and cannot be ignored. Hence, we need first to optimize the camera poses of all current keyframes by minimizing the photometric alignment error of current point cloud, and then alternately optimize $\mathbf{T}_{\mathcal{P}}$ and $\sigma_{\mathcal{P}}$ in Eq. (11),

$$\textbf{step1} : \{\hat{\mathbf{T}}_{\mathrm{K}}^{1}, \ldots, \hat{\mathbf{T}}_{\mathrm{K}}^{|\mathcal{K}|}\} = \mathrm{argmin}_{\{\mathbf{T}_{\mathrm{K}}^{1}, \ldots, \mathbf{T}_{\mathrm{K}}^{|\mathcal{K}|}\}} \sum_{\substack{m \\ u,v \in \mathcal{K}}} \|\mathbf{D}^u(\pi(\mathbf{T}_{\mathrm{K}}^{u}, \mathbf{p}_m)) - \mathbf{D}^v(\pi(\mathbf{T}_{\mathrm{K}}^{v}, \mathbf{p}_m))\|_\delta, \tag{16}$$

$$\textbf{step2} : \{\hat{\mathbf{R}}_{\mathcal{P}}, \hat{\mathbf{t}}_{\mathcal{P}}\} = \arg\min_{\{\mathbf{R}_{\mathcal{P}}, \mathbf{t}_{\mathcal{P}}\}} \alpha \mathrm{E_g} + (1-\alpha)\mathrm{E_p}, \tag{17}$$

$$\textbf{step3} : \hat{\sigma}_{\mathcal{P}} = \arg\min_{\sigma_{\mathcal{P}}} \alpha \mathrm{E_g} + (1-\alpha)\mathrm{E_p}, \tag{18}$$

where $\mathbf{p}_m$ denotes the coordinate of mth 3D point in current point cloud $\mathcal{P}$. We use the LM algorithm to optimize Eqs. (16)–(18). Note, the optimization in fine-grained alignment thread is never-stopping. The alignment thread continuously provides the latest relative pose and relative scale between reference and current point clouds.

We consider both geometric and photometric alignment errors in Eq. (11). In fact, if mth reference 3D point $\mathbf{p}_m^{\mathrm{ref}}$ cannot be observed by both keyframes $\mathbf{D}^u$ and $\mathbf{D}^v$ due to scene's self-occlusion, the corresponding photometric constraint is inaccurate and may negatively affect the alignment accuracy. To solve this problem, for those reference 3D points that project to the same 2D position in a keyframe, we only reserve the corresponding photometric constraint term of the reference 3D point with the nearest distance to the camera center.

3.5. Adjustment thread

The adjustment thread achieves camera motion generation and execution. Specifically, in ith camera adjustment, we take the latest current point cloud $\mathcal{P}^i$ from the VSE thread, and take the latest point cloud relative pose $\mathbf{T}_{\mathcal{P}}^{i}$ and relative scale factor $\sigma_{\mathcal{P}}^{i}$ from the alignment thread. Then we can calculate the optimal hand motion $\hat{\mathbf{M}}_{\mathrm{B}}^{i}$ according to Eq. (3). We consider rACR is finished if the optimal hand motion is small enough. Let $g^i = ||\hat{\mathbf{M}}_{\mathrm{B}}^{i} - \mathbf{I}_4||$, where $\mathbf{I}_4$ indicates 4×4 identity matrix. If $g^i \leqslant \eta$, we consider rACR converges. Otherwise, we adjust the robotic platform by $\hat{\mathbf{M}}_{\mathrm{B}}^{i}$. We set $\eta = 10^{-4}$ in our experiments.

3.6. Synchronization

The three independent threads together achieve a fast and efficient rACR in practice. Specifically, as shown in Fig. 2, given the reference observation $(\mathbf{T}_{\mathrm{A}}^{\mathrm{ref}}, \mathcal{P}^{\mathrm{ref}})$, we first set $s = 1$ and start the VSE thread (note, we need to randomly move the camera around the objective scene to make the VSE work). After that, we conduct the initialization to update s and calibrate the hand-eye relative pose $\mathbf{X}$. Then VSE thread can generate latest current point cloud $\mathcal{P}$ and camera pose $\mathbf{T}_{\mathrm{A}}$ with real scale. Next, we start the alignment thread. The alignment thread takes as input the reference point cloud $\mathcal{P}^{\mathrm{ref}}$ and the latest current point cloud $\mathcal{P}$ taken from the VSE thread, and ceaselessly calculates the latest point cloud relative pose $\mathbf{T}_{\mathcal{P}}$ and relative scale factor $\sigma_{\mathcal{P}}$. We then start the adjustment thread. In ith camera adjustment, the adjustment thread takes the latest $\mathbf{T}_{\mathrm{A}}^{i}$ and latest $\mathbf{T}_{\mathcal{P}}^{i}, \sigma_{\mathcal{P}}^{i}$ from the VSE thread and the alignment thread respectively, then calculates the camera motion $\hat{\mathbf{M}}_{\mathrm{B}}^{i}$ and executes it by the robotic platform.

In this paper, we use DSO SLAM [7] as our VSE. On average, the VSE thread can achieve 15 FPS on a commercial laptop with i7 CPU. In the alignment thread, the fast alignment stage achieves 5 FPS and the fine-grained alignment stage usually needs 5s to run one time. The platform adjustment is relatively slow, averagely spending 1.5s to move 1 cm for the robotic platform we used. Different from the widely used serial scheme of the existing ACR methods, the proposed parallel scheme has faster convergence rate. Under

the serial scheme, the overall system would come to a standstill when platform is moving. On the contrary, the parallel rACR scheme guarantees the VSE thread and alignment thread running continuously, no matter the platform is moving or not.

3.7. Convergence analysis

As shown in Fig. 3, the aim of rACR is to move the current camera (eye) to the pose of the reference one by multiple camera adjustments, i.e., rACR aims to make $\lim_{i\to\infty}\tilde{\mathbf{T}}_{\mathrm{A}}^{i}=\mathbf{T}_{\mathrm{A}}^{\mathrm{ref}}$. Since $\tilde{\mathbf{T}}_{\mathrm{A}}^{i}\simeq\langle\tilde{\mathbf{R}}_{\mathrm{A}}^{i},\tilde{\mathbf{t}}_{\mathrm{A}}^{i}\rangle$ and $\mathbf{T}_{\mathrm{A}}^{\mathrm{ref}}\simeq\langle\mathbf{R}_{\mathrm{A}}^{\mathrm{ref}},\mathbf{t}_{\mathrm{A}}^{\mathrm{ref}}\rangle$, rACR convergence is equivalent to the convergences of the rotation and translation components, i.e., $\lim_{i\to\infty}\tilde{\mathbf{R}}_{\mathrm{A}}^{i}=\mathbf{R}_{\mathrm{A}}^{\mathrm{ref}}$ and $\lim_{i\to\infty}\tilde{\mathbf{t}}_{\mathrm{A}}^{i}=\mathbf{t}_{\mathrm{A}}^{\mathrm{ref}}$. In the following analysis, we use the symbol with superscript '$\wedge$' and the original symbol to respectively denote the real and calculated ones. Besides, we follow the notations in [1] and use $\mathbf{R}\simeq\langle\theta,\bar{\mathbf{e}}\rangle\simeq\langle\cos\frac{\theta}{2},\bar{\mathbf{e}}\sin\frac{\theta}{2}\rangle$to indicate the equivalence of the rotation matrix, the angle-axis and the quaternion representations. For instance, $\hat{\mathbf{R}}_{\mathrm{X}}\simeq\langle\hat{\theta}_{\mathrm{X}},\hat{\bar{\mathbf{e}}}_{\mathrm{X}}\rangle\simeq\langle\cos\frac{\hat{\theta}_{\mathrm{X}}}{2},\hat{\bar{\mathbf{e}}}_{\mathrm{X}}\sin\frac{\hat{\theta}_{\mathrm{X}}}{2}\rangle$ and $\mathbf{R}_{\mathrm{X}}\simeq\langle\theta_{\mathrm{X}},\bar{\mathbf{e}}_{\mathrm{X}}\rangle\simeq\langle\cos\frac{\theta_{\mathrm{X}}}{2},\bar{\mathbf{e}}_{\mathrm{X}}\sin\frac{\theta_{\mathrm{X}}}{2}\rangle$ respectively denote the three kinds of representations of the real and calculated hand-eye relative pose.

We have $\lim_{i\to\infty}\tilde{\mathbf{R}}_{\mathrm{A}}^{i}=\mathbf{R}_{\mathrm{A}}^{\mathrm{ref}}$ if $|\hat{\theta}_{\mathrm{X}}-\theta_{\mathrm{X}}|\leqslant\frac{\pi}{3}$, where $\hat{\theta}_{\mathrm{X}}$ and θ_{X} are the angles of angle-axis representation of $\hat{\mathbf{R}}_{\mathrm{X}}$ and $\mathbf{R}_{\mathrm{X}}$. Specifically, let $\mathbf{T}^{i}\simeq\langle\mathbf{R}^{i},\mathbf{t}^{i}\rangle$ be the relative pose between the reference and current cameras in the world coordinate system, i.e., $\mathbf{T}^{i}=\tilde{\mathbf{T}}_{\mathrm{A}}^{i}(\mathbf{T}_{\mathrm{A}}^{\mathrm{ref}})^{-1}$, then we have $\mathbf{R}^{i}=\tilde{\mathbf{R}}_{\mathrm{A}}^{i}(\mathbf{R}_{\mathrm{A}}^{\mathrm{ref}})^{-1}$. Following the derivation of Theorem 1 in [1], we can easily get $\lim_{i\to\infty}\mathbf{R}^{i}=\mathbf{I}_{3}$ if $\theta\leqslant\frac{\pi}{3}$, where $\langle\theta,\bar{\mathbf{e}}\rangle$is the angle-axis representation of $\mathbf{R}_{\mathrm{X}}(\hat{\mathbf{R}}_{\mathrm{X}})^{-1}$. Therefore, we know $\lim_{i\to\infty}\tilde{\mathbf{R}}_{\mathrm{A}}^{i}=\mathbf{R}_{\mathrm{A}}^{\mathrm{ref}}$ if $\theta\leqslant\frac{\pi}{3}$. Next, under the quaternion representation, let $\mathbf{R}_{\mathrm{X}}(\hat{\mathbf{R}}_{\mathrm{X}})^{-1}\simeq\langle\cos\frac{\theta}{2},\bar{\mathbf{e}}\sin\frac{\theta}{2}\rangle$. According to the Rodrigues' formula, we further have

$$\begin{aligned}\cos\tfrac{\theta}{2}&=\cos\tfrac{\hat{\theta}_{\mathrm{X}}}{2}\cos\tfrac{\theta_{\mathrm{X}}}{2}+\sin\tfrac{\hat{\theta}_{\mathrm{X}}}{2}\sin\tfrac{\theta_{\mathrm{X}}}{2}\bar{\mathbf{e}}_{\mathrm{X}}\cdot\hat{\mathbf{e}}^{-}{}_{\mathrm{X}}\\&\leqslant\cos\tfrac{\hat{\theta}_{\mathrm{X}}}{2}\cos\tfrac{\theta_{\mathrm{X}}}{2}+\sin\tfrac{\hat{\theta}_{\mathrm{X}}}{2}\sin\tfrac{\theta_{\mathrm{X}}}{2}=\cos\tfrac{\hat{\theta}_{\mathrm{X}}-\theta_{\mathrm{X}}}{2}.\end{aligned}\tag{19}$$

It means that $\theta\geqslant|\hat{\theta}_{\mathrm{X}}-\theta_{\mathrm{X}}|$. Thus, we have $\lim_{i\to\infty}\tilde{\mathbf{R}}_{\mathrm{A}}^{i}=\mathbf{R}_{\mathrm{A}}^{\mathrm{ref}}$ if $|\hat{\theta}_{\mathrm{X}}-\theta_{\mathrm{X}}|\leqslant\frac{\pi}{3}$. In fact, Theorem 1 in [1] proves that the current camera pose would converge to the reference one if $\hat{\theta}_{\mathrm{X}}\leqslant\frac{\pi}{3}$ and $\theta_{\mathrm{X}}=0$, which is a particular case of rACR convergence process.

After $\hat{\mathbf{R}}_{\mathrm{A}}$ converges to $\mathbf{R}_{\mathrm{A}}^{\mathrm{ref}}$, we have $\lim_{i\to\infty}\tilde{\mathbf{t}}_{\mathrm{A}}^{i}=\mathbf{t}_{\mathrm{A}}^{\mathrm{ref}}$ if $|\hat{\theta}_{\mathrm{X}}-\theta_{\mathrm{X}}|<\frac{\pi}{4}$. Specifically, following the derivation of Eq. (28) in [1], we can get $\|\mathbf{t}^{i}\|^{2}-\|\mathbf{t}^{i+1}\|^{2}\geqslant(\frac{s}{\hat{s}})^{2}(2\cos^{2}\theta-1)$. When $\theta<\frac{\pi}{4},2\cos^{2}\theta-1>0$, i.e., $\|\mathbf{t}^{i}\|>\|\mathbf{t}^{i+1}\|$. Hence, if camera translation occurs infinite times, $\|\mathbf{t}^{i}\|$ certainly converges to zero, i.e., $\lim_{i\to\infty}\mathbf{t}^{i}=\mathbf{0}^{\mathrm{T}}$ if $\theta<\frac{\pi}{4}$. Besides, since $\mathbf{P}^{i}\simeq\langle\mathbf{R}^{i},\mathbf{t}^{i}\rangle=\tilde{\mathbf{T}}_{\mathrm{A}}^{i}(\mathbf{T}_{\mathrm{A}}^{\mathrm{ref}})^{-1}$, we have $\mathbf{t}^{i}=-\tilde{\mathbf{R}}_{\mathrm{A}}^{i}(\mathbf{R}_{\mathrm{A}}^{\mathrm{ref}})^{-1}\mathbf{t}_{\mathrm{A}}^{\mathrm{ref}}+\tilde{\mathbf{t}}_{\mathrm{A}}^{i}$. Since $\hat{\mathbf{R}}_{\mathrm{A}}$ has converged to $\mathbf{R}_{\mathrm{A}}^{\mathrm{ref}}$, then $\mathbf{t}^{i}=\tilde{\mathbf{t}}_{\mathrm{A}}^{i}-\mathbf{t}_{\mathrm{A}}^{\mathrm{ref}}$. Therefore, we know $\lim_{i\to\infty}\tilde{\mathbf{t}}_{\mathrm{A}}^{i}=\mathbf{t}_{\mathrm{A}}^{\mathrm{ref}}$ if $\theta<\frac{\pi}{4}$. Refer to the analysis of the rotation convergence, we know $\theta<\frac{\pi}{4}$ is equivalent to $|\hat{\theta}_{\mathrm{X}}-\theta_{\mathrm{X}}|<\frac{\pi}{4}$. Hence, we have $\lim_{i\to\infty}\tilde{\mathbf{t}}_{\mathrm{A}}^{i}=\mathbf{t}_{\mathrm{A}}^{\mathrm{ref}}$ if $|\hat{\theta}_{\mathrm{X}}-\theta_{\mathrm{X}}|<\frac{\pi}{4}$.

In a word, we have $\lim_{i\to\infty}\tilde{\mathbf{T}}_{\mathrm{A}}^{i}=\mathbf{T}_{\mathrm{A}}^{\mathrm{ref}}$ if $|\hat{\theta}_{\mathrm{X}}-\theta_{\mathrm{X}}|<\frac{\pi}{4}$. That is, the proposed rACR scheme converges if the angle difference of the real hand-eye relative pose and the calculated one is less than $\frac{\pi}{4}$. In fact, the above convergence condition can be easily satisfied in practice. Note, from the above analysis, we can find that the convergence is independent of the calculated real scale factor s. This is because the multi-adjustment strategy can guarantee that camera asymptotically approaches the target. Certainly, the more accurate the calculated camera relative pose and the real scale factor, the fewer iteration number the rACR needs. In contrast, the state-of-the-art ACR method [1] just guesses the hand-eye relative pose as identity matrix and uses a bisection strategy to handle the unknown scale problem, which obviously needs more iteration number than the proposed rACR.

4. Discussions

We then analyze why 2D matching based ACR is easy to fail under highly varied lightings. The reasons mainly have two aspects. First, previous research [36] reports that 2D matching (e.g., SIFT) is robust to lighting variation for distant near-planar scenes, e.g., buildings. But in this paper, we mainly focus on close-range high-value scenes, e.g., culture heritages, which usually have rich surface 3D micro-structures and very few pure-planar regions. Therefore, varied lightings easily cause local appearance differences of images, further lead to *higher wrong matching rate*. To verify it, we first capture a reference image under environment lighting, then we keep the camera still and capture current images under varied lighting conditions (three different side lightings and the night lighting). Table 1 shows the 2D feature matching accuracy (marked as "#match", "A/B" denotes right/total matching number) and pose estimation accuracy by 5-point algorithm [46] (angle value of rotation's angle-axis form, marked as

Table 1
2D matching performance under varied lightings for SIFT, TILDE [36]+SIFT, DELF [40] and ASLFeat [38]. See text for details.

		scene 1				scene 2			
		light 1	light 2	light 3	night	light 1	light 2	light 3	night
SIFT	#match	2/4	5/5	29/34	2/6	35/47	5/13	28/70	1/19
	a-rot	NaN	0.41	0.65	10.4	0.34	4.62	0.49	3
	AFD	NaN	10.1	13.3	136	5.72	67	10.7	54.7
TILDE + SIFT	#match	6/18	59/97	112/150	180/265	74/148	13/39	151/310	51/327
	a-rot	4.69	0.82	0.16	0.18	1.36	1.58	0.46	0.55
	AFD	57.4	17.5	**2.84**	3.61	23.7	19.9	8.59	11.2
DELF	#match	201/367	197/371	366/438	289/377	279/451	178/294	385/551	265/373
	a-rot	1.86	1.32	0.11	0.21	0.29	0.37	0.69	0.12
	AFD	26.1	23.5	**2.4**	3.95	6.53	8.3	10.5	**2.72**
ASLFeat	#match	11/14	107/145	241/330	794/1164	231/466	27/97	340/826	357/1705
	a-rot	0.14	2.05	2.23	0.68	0.22	1.53	5.05	0.78
	AFD	**2.87**	21.5	23.4	9.64	3.91	22.2	54.8	14.8

"a-rot") for SIFT, TILDE [36]+SIFT, DELF [40] and ASLFeat [38] for two scenes, where TILDE is a learning based keypoint detector, TILDE + SIFT means combining TILDE detector and SIFT descriptor, DELF and ASLFeat are two state-of-the-art learning based 2D feature detectors and descriptors. We use the official implementations of these methods to conduct the above experiments. We consider one matching is wrong if the Euclidean distance is larger than 1. From Table 1, we can find that nearly half of matchings are wrong for most cases. Higher wrong matching rate causes inaccurate pose estimation and finally results in ACR failure (AFD$\leq$ 3).

Second, highly varied lighting may lead to *fewer 2D matching number*. In this case, even though the wrong matching rate is lower, it also easily leads to inaccurate pose estimation and further causes ACR failure, see the results marked red in Table 1. In fact, if we can collect enough training data for all kinds of scenes and lighting conditions, the learning based 2D matching really has the possibility of supporting ACR under highly varied lightings. But this manner is impractical in real-world application. In summary, we can say that 2D matching is unstable and easy to fail under highly varied lighting and cannot support ACR in outdoor.

We then analyze why 3D alignment based rACR works well under varied lightings. First, 3D alignment takes as input two point clouds rather than the images, so 3D alignment itself is irrelevant to lighting variation between twice observation. Second, since lighting is usually stable during once observation, the point cloud taken from VSE is reliable. Hence, the only worry is the misalignment problem. Specifically, 3D alignment needs to find and align the corresponding 3D points between two point clouds (see Eq. (9)). However, we cannot guarantee the same 3D point can be faithfully reconstructed in both reference and current point clouds especially under significant lighting variation. Hence, the alignments of those "not repetitive" 3D points are obviously wrong and seemingly influence point cloud alignment accuracy. In fact, our rACR can always work well in extensive experiments. This is because we use DSO [7] (direct SLAM) as our VSE, which can generate semi-dense point cloud. Therefore, the "not repetitive" 3D point would be aligned to the 3D point that has very close distance to ground truth. Hence, these alignment errors are usually small and would not influence rACR accuracy.

To verify the above analysis, we add noises ($r\times$ point cloud width) of different magnitude r ($1\% \sim 16\%$) on reference point to simulate "not repetitive" 3D points. Then we run rACR 3 times for 3 different scenes. The relation of noise magnitude and AFD [1] (used to evaluate ACR accuracy) is shown in Fig. 4. Besides, Fig. 4 also shows the visual point cloud alignment results of one certain scene under different noise magnitudes. Note, in this paper, we consider camera relocalization is successful if the corresponding AFD is less than 3. Besides, see Fig. 5, the horizontal axis indicates varied lightings, the vertical axis denotes the offset magnitudes of point clouds after 3D alignment. Fig. 5 shows that the offset magnitude of 3D alignment is usually lower than 2% under varied lightings. In Fig. 4, we have verify that our rACR can work well (AFD $<$ 3) when the noise magnitude r of point cloud is less than 8%. Besides, Fig. 5 also shows that the noise magnitude of point cloud is usually less than 2% under varied lighting conditions. Therefore, combining the results of Fig. 4 and Fig. 5, we can say that r less than 8% is easy to meet in practice, and the lighting variations do not influence our rACR performance.

5. Experimental results

5.1. Setup

We build 10 indoor scenes (S1-10) and 10 outdoor scenes (S11-20) to benchmark our rACR method. For indoor scenes, we use an LED point light source mounted on a robotic arm to produce different side lighting conditions. For outdoor scenes, the lighting condition differences between twice observations usually come from the ambient illumination variations, e.g., weather differences or sunshine direction changes.

We compare the proposed rACR approach with homography-based camera relocalization (H-CR) [3], computational rephotography (CRP) [6] and the active camera relocalization method (ACR) [1]. H-CR and CRP are manual methods. ACR is the state-of-the-art camera relocalization method without hand-eye calibration and relies on a robotic platform to automatically adjust camera. All of these methods are based on SIFT feature matching. Besides, as shown in Table 1, we find TILDE + SIFT and DELF are better than SIFT feature for some cases. Thus, for more fair comparison, we replace SIFT with TILDE + SIFT and DELF in ACR method to generate two new baselines, i.e., ACR_TILDE and ACR_DELF.

Following the ACR method [1], we use *feature-point displacement flow* (FDF) (a sparse displacement vector field of matched feature points) and *average feature point displacement* (AFD) to quantitatively measure relocalization accuracy. Note, since feature point matchings easily fail under highly varied illuminations, which may impair the fairness of FDF and AFD measure. Therefore, after camera relocalization, we restore the reference illumination by controlling the robotic arm and capture an auxiliary current image to measure FDF and AFD for indoor scenes. For outdoor scenes, we uniformly divide the image into a regular grid and manually select 20–100 matching points between current and reference images to generate FDF and AFD.

5.2. Comparative study

5.2.1. Relocalization accuracy comparison

For indoor scenes, we capture the reference observation under front lighting and conduct our rACR and baselines under an arbitrary side lighting. As for the outdoor scenes, after capturing the reference observations, we conduct camera relocalization after several days. S14-15 are observed under day-night variations, S12, S16-17 are observed at the same time (similar sunshine direction) but different days, the others are observed at different time (different sunshine direction). Note, for all the comparison experiments, we place the camera to the same initial pose (via the robotic platform) to fairly compare the relocalization performance. The comparison results are shown in Table 2 and parts of the qualitative comparison results are shown in Fig. 14. The red color denotes camera relocalization failure (AFD $>$ 3). We can clearly see that the proposed rACR outperforms baselines in all cases. Moreover, CRP and HCR fail on all 20 scenes. ACR, ACR_TILDE and ACR_DELF fail for more than half of cases. The chief causes of the lower performance of the existing methods compared with our rACR, have the following two aspects. First, these baselines are more easily failed under extreme lighting difference (e.g., S3, S14-15), lighting difference may cause a large number of local shadow or shading variations, further influences the feature matching accuracy. Second, for weakly-textured scenes, 2D feature based method itself is prone to failure, even though lighting variation is not significant (e.g., S1-2). As shown in Table 2, considering that the original ACR method is just slightly worse than ACR_TILDE and ACR_DELF, which need high performance computing resources and are not suitable for outdoor, so we only compare our rACR with the original ACR in the rest of our experiments.

In this paper, we use a direct VSE (DSO SLAM [7]) in our rACR. We employ a state-of-the-art indirect VSE, ORB [19], and compare the ORB based rACR (rACR-ORB) with the original one. Fig. 6 shows the comparison results of the two methods for 5 scenes (S1–5) under varied lightings. We can see that the proposed rACR (rACR-DSO) achieves much more better relocalization accuracy

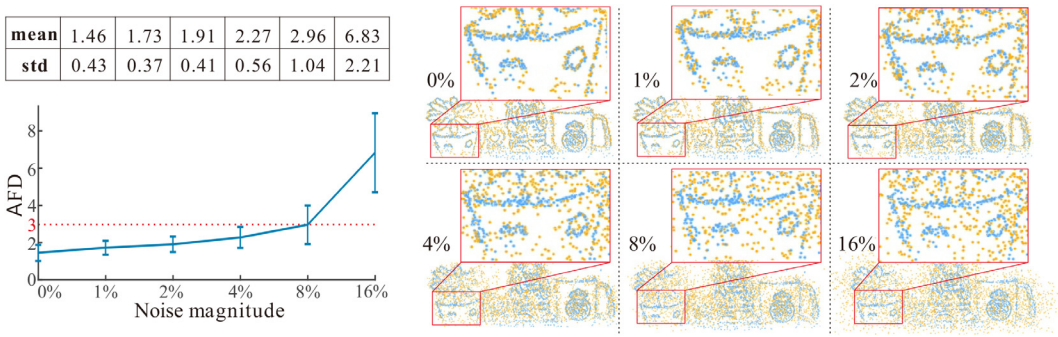


Fig. 4. Evaluation of our rACR under different noise magnitudes of point cloud. The orange and blue denote the reference and current point clouds, respectively.

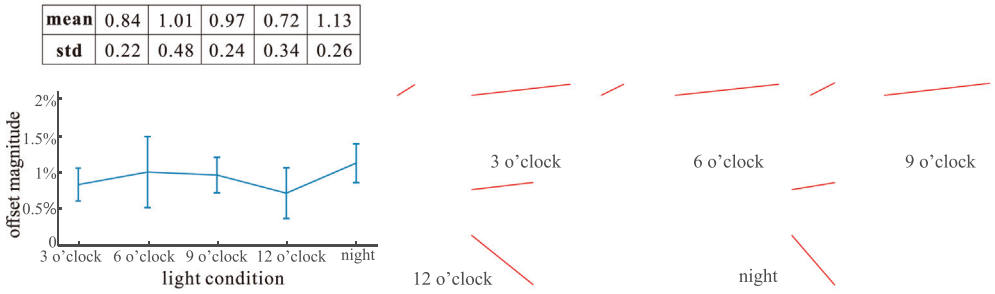


Fig. 5. Relation of the point cloud offset and different lighting conditions. Combining Fig. 4, we can find rACR is robust to varied lighting. The orange and blue denote the reference and current point clouds, respectively.

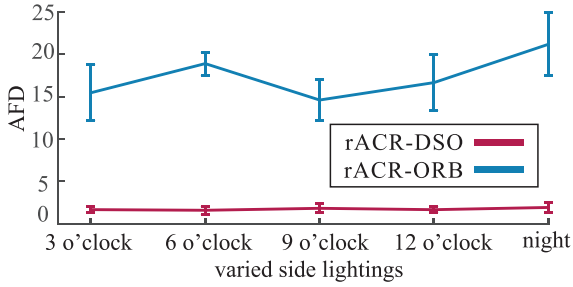
Table 2
Average AFD scores of rACR and the baselines under 20 scenes. The red color denotes camera relocalization failure (AFD > 3).

	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
CRP [6]	34.61	28.32	35.66	12.70	34.66	11.61	7.187	6.870	6.230	8.438
H-CR [3]	38.02	30.95	18.64	8.762	26.45	9.137	5.920	7.127	3.577	7.656
ACR [1]	45.37	11.02	5.485	6.886	8.391	3.192	2.528	1.318	1.573	4.599
ACR_TILDE [1,33]	13.82	50.36	19.30	16.99	16.84	1.508	4.847	6.800	2.831	10.05
ACR_DELF [1,37]	5.712	62.45	4.324	2.615	2.287	3.371	2.952	2.095	13.81	3.905
rACR (ours)	**1.261**	**1.637**	**0.856**	**1.909**	**1.098**	**1.146**	**1.879**	**0.984**	**0.976**	**1.650**
	S11	S12	S13	S14	S15	S16	S17	S18	S19	S20
CRP [6]	14.00	25.97	17.62	35.74	60.32	16.37	25.80	22.32	94.15	51.25
H-CR [3]	8.374	18.62	9.019	33.71	52.00	8.295	16.18	7.345	95.94	49.41
ACR [1]	3.452	1.204	2.546	49.50	88.53	1.859	9.501	5.581	79.91	70.53
ACR_TILDE [1,33]	2.551	1.147	2.529	30.91	45.72	1.471	3.923	2.313	57.23	60.12
ACR_DELF [1,37]	2.753	1.686	2.190	24.83	30.82	1.492	1.874	4.783	50.13	60.01
rACR (ours)	**1.526**	**0.947**	**1.548**	**2.867**	**1.869**	**1.120**	**0.886**	**1.338**	**1.147**	**2.535**

than rACR-ORB. The results also confirm the analysis (refer to Section 3.3) about why we choose direct VSE rather than indirect VSE in our rACR.

5.2.2. Comparison under varied illuminations

The state-of-the-art method ACR [1] employs 2D feature matching and 5-point algorithm to estimate the relative pose between reference and current camera. On the contrary, our rACR uses 3D point clouds to estimate the relative pose, which can greatly improve the robustness to highly varied illuminations between twice observations. To verify this, we quantitatively evaluate the robustness of our rACR and baseline ACR [1] to varied illuminations on the 10 indoor scenes S1–10. Specifically, we first capture the reference observation under the particular 12 o'clock lighting direction via controlling the robotic arm (mounted with the LED light source). After that, we conduct the two methods under 12 different current lighting directions ranging from 1 o'clock to 12 o'clock, respectively. Then we evaluate the relocalization accuracy using AFD score. For each indoor scene, we independently conduct such experiments 10 times and the results are shown in Fig. 7. We can see both AFD average and AFD variance of our rACR are small under all the illumination directions. It means rACR exhibits very good robustness to varied illuminations. In contrast, ACR method is very sensitive to lighting differences. The ACR method can only



rACR-DSO	avg	1.67	1.60	1.83	1.67	1.93
	std	0.34	0.48	0.52	0.38	0.60
rACR-ORB	avg	15.4	18.8	14.6	16.6	21.2
	std	3.32	1.35	2.41	3.29	3.71

Fig. 6. Quantitative comparisons of the direct VSE based rACR (rACR-DSO) and the indirect VSE based rACR (rACR-ORB) under varied side lightings.

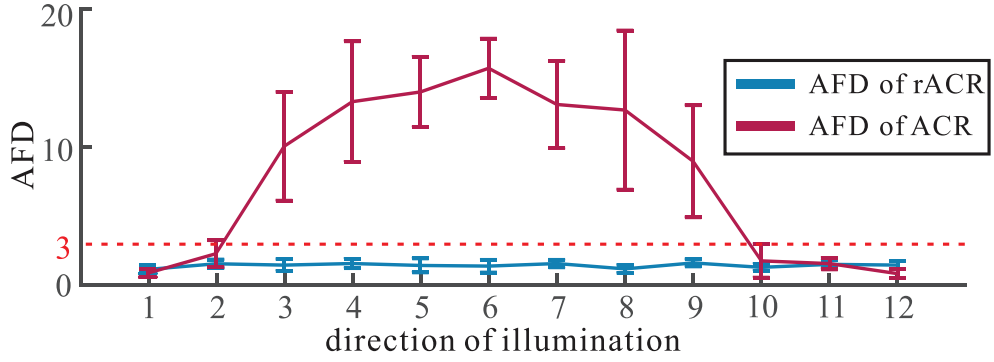


Fig. 7. AFDs of rACR and ACR under varied illumination directions. The reference image is captured under 12 o'clock direction.

achieve reliable relocalization results under similar illumination directions of the reference one, e.g., 1, 2, 10 and 11 o'clock, but the relocalization error rapidly grows if the current illumination directions are far away from the reference one. The dashed line in Fig. 7 denotes the line that AFD $= 3$ and we consider camera relocalization succeeds if AFD $\leqslant 3$.

5.2.3. Speed and convergence comparison

The existing ACR baseline [1] uses a bisection strategy to solve the scale unknown problem of the estimated relative pose by 2D feature matching. In short, the bisection strategy first guesses an initial translation scale, then the translation scale is reduced to half its original value if the translation directions of the estimated relative poses of successive two iterations are opposite, otherwise the translation scale remains the same. On the contrary, we propose a simple but effective point cloud real scale estimation method (see Section 3.2 for details) to avoid the slower bisection adjustment strategy. Therefore, our rACR theoretically has faster convergence speed than the ACR baseline. To verify this, we compare the convergence speed of the two methods. Since the ACR method only succeeds for scenes S7–9, S12–13 and S16, for a fair comparison, we only use the 6 scenes to compare the speed and convergence of ACR and rACR. Table 3 shows the average time spent comparisons of ACR and rACR, involving point cloud acquiring, camera pose estimation, and the overall process. First, due to the success of recent visual SLAM methods [7], we just need a dozen seconds to acquire reliable point clouds. Second, rACR directly gets the current camera pose from VSE, which takes very little time. Besides, the proposed parallel scheme also improves the convergence rate. In contrast, ACR needs 2D feature matching and pose estimation algorithm to calculate the camera relative pose, that process usually costs several seconds. Third, since the proposed rACR can also obtain the hand-eye relation and percept the real scale of scene, rACR needs not the bisection approaching strategy like the ACR method. Therefore, the iteration number of rACR is much fewer than ACR. In a word, although rACR needs additional point cloud acquirement time, the faster pose estimation step and fewer iteration number help rACR to achieve faster camera relocalization speed than ACR.

Besides, to further compare the convergence speed of our rACR and existing ACR method, we design a new version for each of the two methods. As shown in Table 3, "_w/X" and "_w/oX" denotes the method with or without the estimated hand-eye relative pose, respectively. That is, rACR_w/oX means we guess the hand-eye relative pose as identity matrix in our rACR. We can find that ACR_w/X and rACR respectively have faster convergence speed than ACR and rACR_w/oX, which shows that calculating hand-eye pose can effectively improve convergence speed. Besides, ACR_w/X is still much slower than rACR and rACR_w/oX. This is because our method avoids time-consuming bisection strategy by estimating the real scale of relative pose between reference and current images.

Table 3
Average time spent comparisons of ACR and rACR.

	pt cloud	camera pose	overall	#iteration
ACR	–	4.8s	316.7s	18.6
rACR	15.2s	0.2s	61.1s	5.3
ACR_w/X	–	4.8s	256.4s	14.3
rACR_w/oX	14.6s	0.2s	102.6s	8.1

To compare the convergence of ACR and rACR, we record all the intermediate images and corresponding camera poses during ACR and rACR processes for scenes S7–9, S12–13 and S16. We compute the AFD score for each intermediate image during the relocalization process. For each iteration, specifically, we also compare the average difference between current camera pose and the reference one in the world coordinate system, i.e., $\|\tilde{\mathbf{T}}_{\mathrm{A}}^{i} - \mathbf{T}_{\mathrm{A}}^{\mathrm{ref}}\|$. For fair comparison, we place the camera to a little farther position away from the reference one for rACR than ACR. As shown in Fig. 8, the proposed rACR can quickly converge to a very low AFD state after only 4–6 iterations, while ACR has much slower convergence rate. This is because ACR cannot estimate real translation scale [1]. It uses a slow bisection approaching strategy to guarantee convergence. Besides, ACR just guesses hand-eye relative pose as identity matrix, which also influences the relocalization speed. Moreover, since the ACR method relies on bisection approaching strategy, both $\|\tilde{\mathbf{T}}_{\mathrm{A}}^{i} - \mathbf{T}_{\mathrm{A}}^{\mathrm{ref}}\|$ and AFD may increase sometimes. In contrast, $\|\tilde{\mathbf{T}}_{\mathrm{A}}^{i} - \mathbf{T}_{\mathrm{A}}^{\mathrm{ref}}\|$ and AFD are always decreased for the proposed rACR.

5.3. Challenging situation study

We provide more comparison results of the proposed rACR and the state-of-the-art ACR to show the effectiveness of our method, especially for challenging situations when ACR is easy to fail. To this end, we additional build 9 different scenes that belong to 3 different types. The first type of scenes (W1-3) are weakly-textured, i.e., have fewer feature points. The second type of scenes (R1–3) have many repetitive patterns. For the third type of scenes (V1–3), we conduct rACR and ACR under highly varied illuminations between reference and current observations. For the 6 scenes of the first two types, we execute rACR and ACR under similar lighting conditions between twice observations. We conduct 5 independent camera relocalization for each method to avoid statistical bias. Table 4 shows the average AFD scores of the two methods under each scene. The corresponding visual relocalization results are shown in Figs. 9–11. we manually count the number of correct matching points. "A/B" means that there are "B" matching points

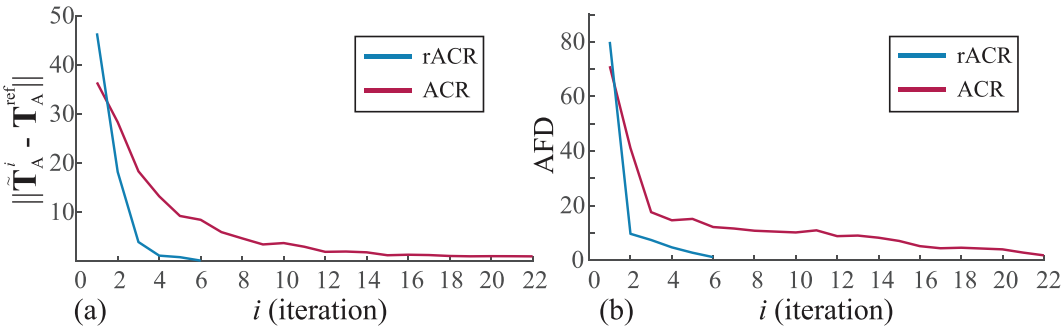


Fig. 8. Convergence comparison of rACR and ACR.

Table 4
Average AFD scores of the proposed rACR and ACR method under challenging situations.

	weak textures			repetitive patterns			varied illuminations		
	W1	W2	W3	R1	R2	R3	V1	V2	V3
ACR [1]	12.35	8.845	fail	4.263	4.186	7.069	16.17	19.24	11.74
rACR (ours)	**1.375**	**1.183**	**1.579**	**1.372**	**1.544**	**1.466**	**1.789**	**1.528**	**1.719**

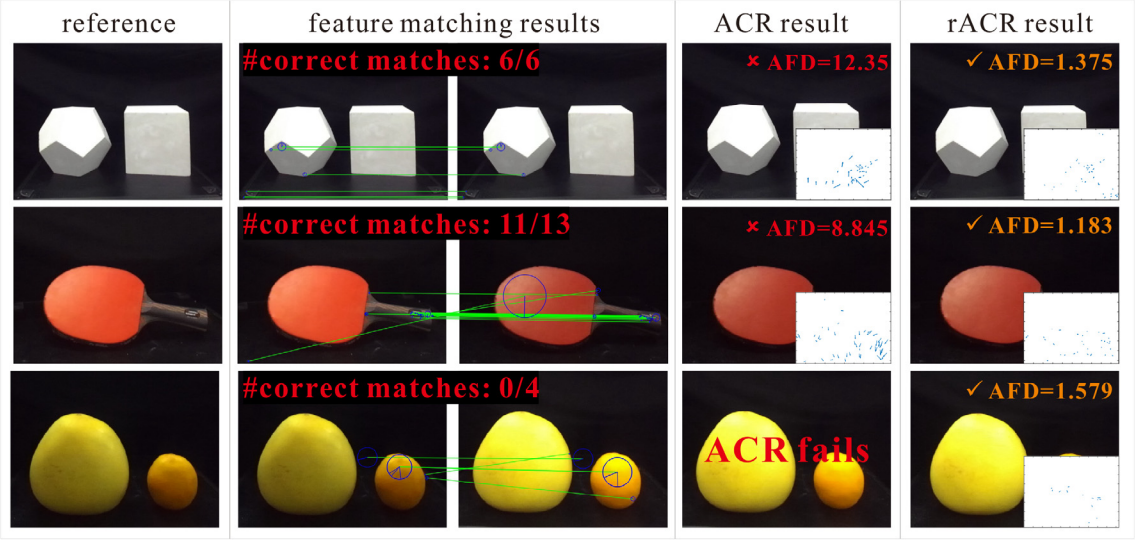


Fig. 9. rACR vs ACR for scenes with weak textures.

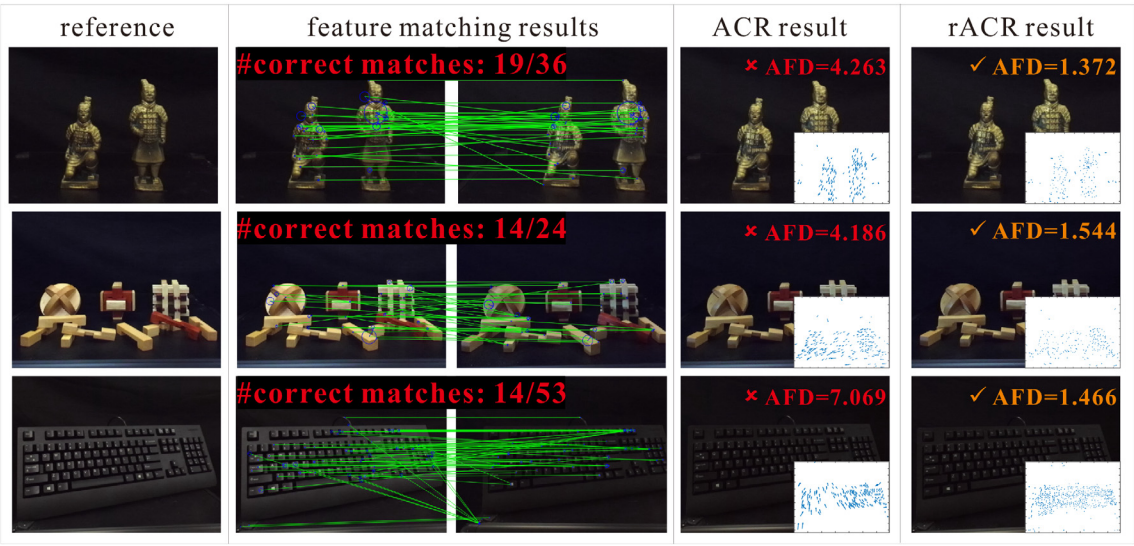


Fig. 10. rACR vs ACR for scenes with repetitive patterns.

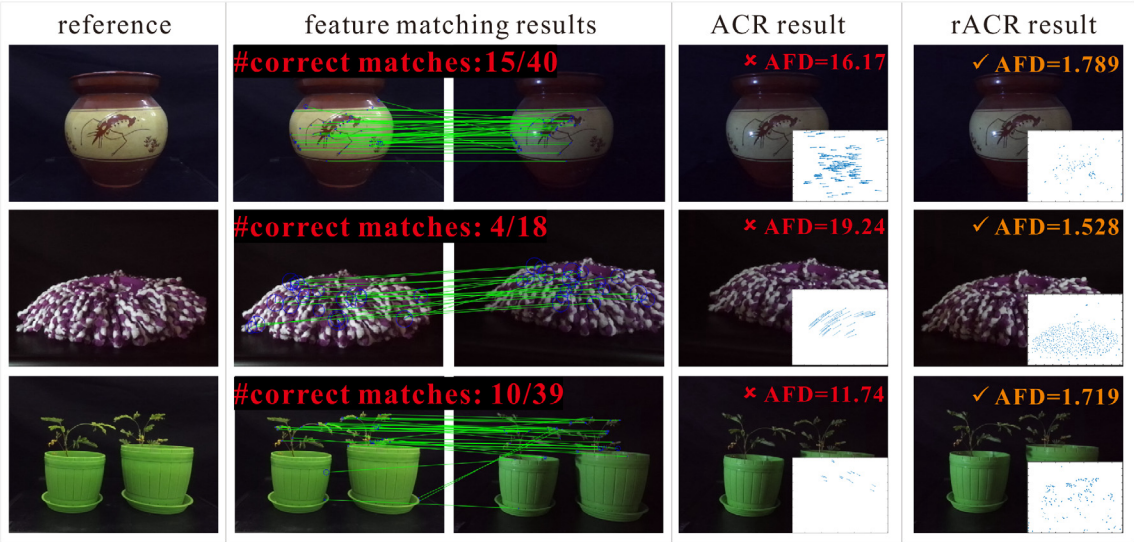


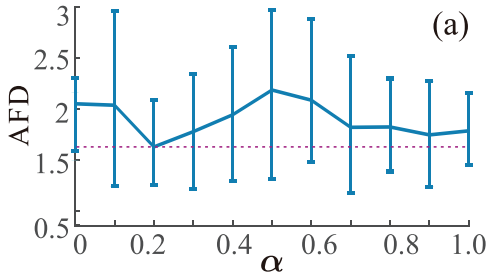
Fig. 11. rACR vs ACR for scenes under highly varied illuminations.

and "A" of them are correct. As shown in Fig. 9, we can find that the number of matching points is very few under weakly-textured scenes. In fact, even if most of the matchings are corrected, fewer feature matching constraints may lead to the pose estimation error of ACR, then causing inaccurate camera relocalization result. The ACR method uses 5-point algorithm [46] to estimate relative pose. Thus, ACR completely fails for the third scene since the matching point number is less than 5. Besides, as shown in Fig. 10 and Fig. 11, we can find that a lot of matching points are wrong and these erroneous matchings inevitable cause ACR failure. On the contrary, our rACR scheme uses 3D point clouds, rather than feature matching to bridge the current and reference observations. Therefore, the proposed rACR can outperform ACR by a large margin and achieve accurate camera relocalization result for all the three challenging situations.

5.4. Ablation study

To quantitatively measure the influence of geometric and photometric terms in Eq. (11), we conduct rACR under the 10 indoor scenes for different parameter α taken from range $[0, 1]$. For each value of α, we conduct 5 times for 3 scenes (S1–3). The corresponding AFD scores are shown in Fig. 12(a). We can see that rACR achieves the best performance when $\alpha = 0.2$. Hence, we set α to 0.2 in our experiments.

Besides, we compare various versions of rACR to verify the effectiveness of the scale-aware point cloud alignment. Specifically, we use "rACR" to denote the proposed rACR method. "rACR-P", "rACR-F" and "rACR-S" indicates the rACR without optimizing current keyframe poses (Eq. (16)), without fine-grained alignment (Eqs. (16)–(18)) and only optimizing the relative pose $\mathbf{T}_{\mathcal{P}}$ but not the relative scale factor $\sigma_{\mathcal{P}}$ in Eq. (9), respectively. We independently test each version 5 times for 3 scenes (S1–3). The "average + std" AFD scores are shown in Fig. 12(b). Comparing rACR-F with rACR-S, we can find that scale-aware point cloud alignment plays a very important role for better relocalization accuracy, since the scale difference between reference and current point clouds may severely affect the accuracy of current camera pose estimation in the world coordinate system. Furthermore, with the help of optimizing current keyframe poses, the proposed rACR achieves better camera relocalization accuracy than rACR-P.



(b)

	S1	S2	S3
rACR-S	3.22+0.52	3.98+0.48	4.13+0.32
rACR-F	1.74+0.19	2.42+0.13	1.87+0.20
rACR-P	1.61+0.06	2.38+0.35	2.07+0.29
rACR	1.59+0.04	1.95+0.08	1.69+0.09

Fig. 12. (a) The influence of α to rACR accuracy. (b) AFD average + std of each version of rACR for 3 scenes.

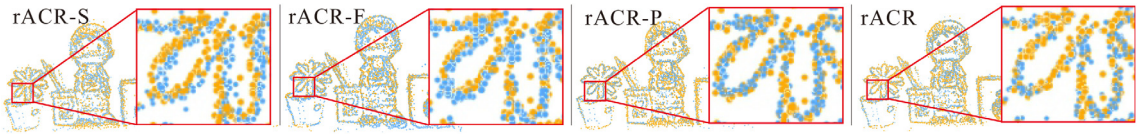


Fig. 13. Comparison of different versions of rACR in 3 scenes. In each case, the first column shows the reference image. The visual point cloud alignment results of different versions of rACR are listed on the right.

Table 5
Speed comparison of serial rACR (alternately conducting point cloud alignment and motion execution in one thread) and parallel rACR.

	mean	std
serial rACR	84.6s	9.25s
parallel rACR	59.5s	5.70s

Note, it is reasonable that the performances of rACR-F and rACR-P are close in Fig. 12(b). Specifically, our point cloud alignment function (Eq. (11)) has two terms, the geometric term E_g and the photometric term E_p. The fast alignment stage only optimizes E_g and the fine-grained alignment stage optimizes both E_g and E_p. Since there exit inevitable estimation errors of current keyframe poses (estimated by SLAM), these errors would influence the estimation accuracy of $\mathbf{R}_{\mathcal{P}}$, $\mathbf{t}_{\mathcal{P}}$ and $\sigma_{\mathcal{P}}$ if we only optimize Eqs. 17,18 in the fine-grained alignment stage. Therefore, if we do not simultaneously optimize the current keyframe poses (i.e., Eq. (16)), conducting fine-grained alignment may not bring a remarkable accuracy improvement. This is why rACR-F and rACR-P have close

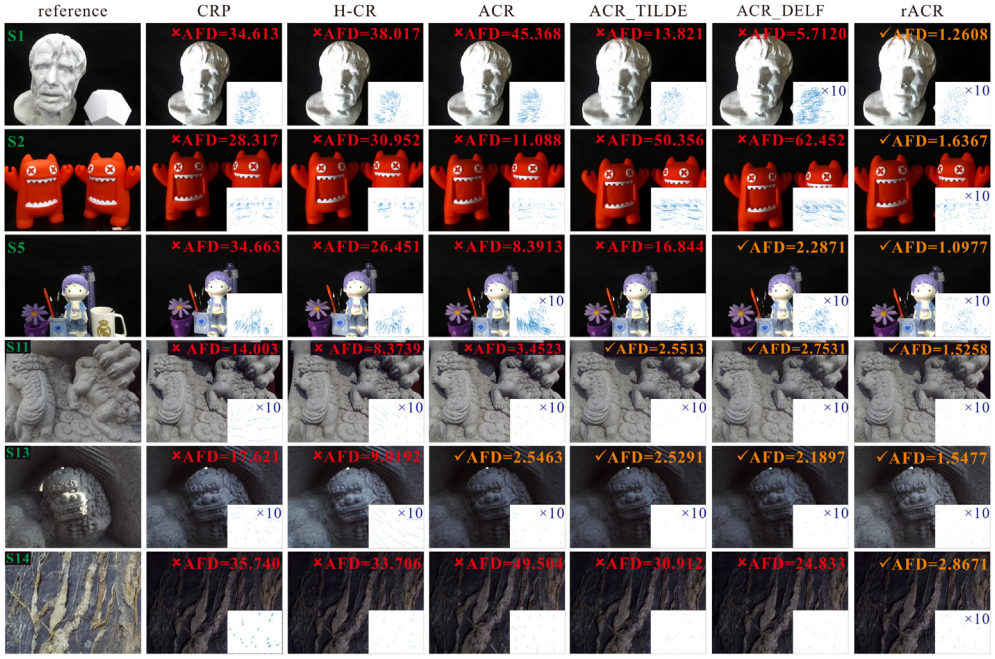


Fig. 14. Visual comparison of our rACR and baselines for parts of the scenes. Bottom-right subfigure is the FDF map. We consider camera relocalization is successful if AFD $\leqslant$ 3.0.

performance. Furthermore, when we optimize both Eqs. (16)–(18), rACR can achieve a noticeable performance improvement, see the results of the last row in Fig. 13(b). Besides, Fig. 13 shows the visual point cloud alignment results for one case. We can see that the proposed rACR achieves the best 3D alignment effect (e.g., see the zoom-in parts in Fig. 13).

To validate the efficiency of the proposed parallel rACR scheme, we also implement a serial rACR version, that alternately conducts point cloud alignment and motion execution within a single thread. We compare the serial rACR and parallel rACR on 3 different scenes (S1-3), for each of which we independently run two versions of rACR 10 times. For fairness, in each test, we place the camera to the same initial pose (via the robotic platform). As shown in Table 5, we can clearly see that parallel rACR is almost 1.5 times faster.

5.5. Real-world applications for FGCD

An important real-world application of the propose rACR is fine-grained change detection (FGCD) [3] in the wild. Different from the traditional change detection (CD) problem [47], which mainly focuses on finding large-scale changes, FGCD [3] aims to discover and locate minute changes by comparing reference and current observations within proper time interval for the same scene. In

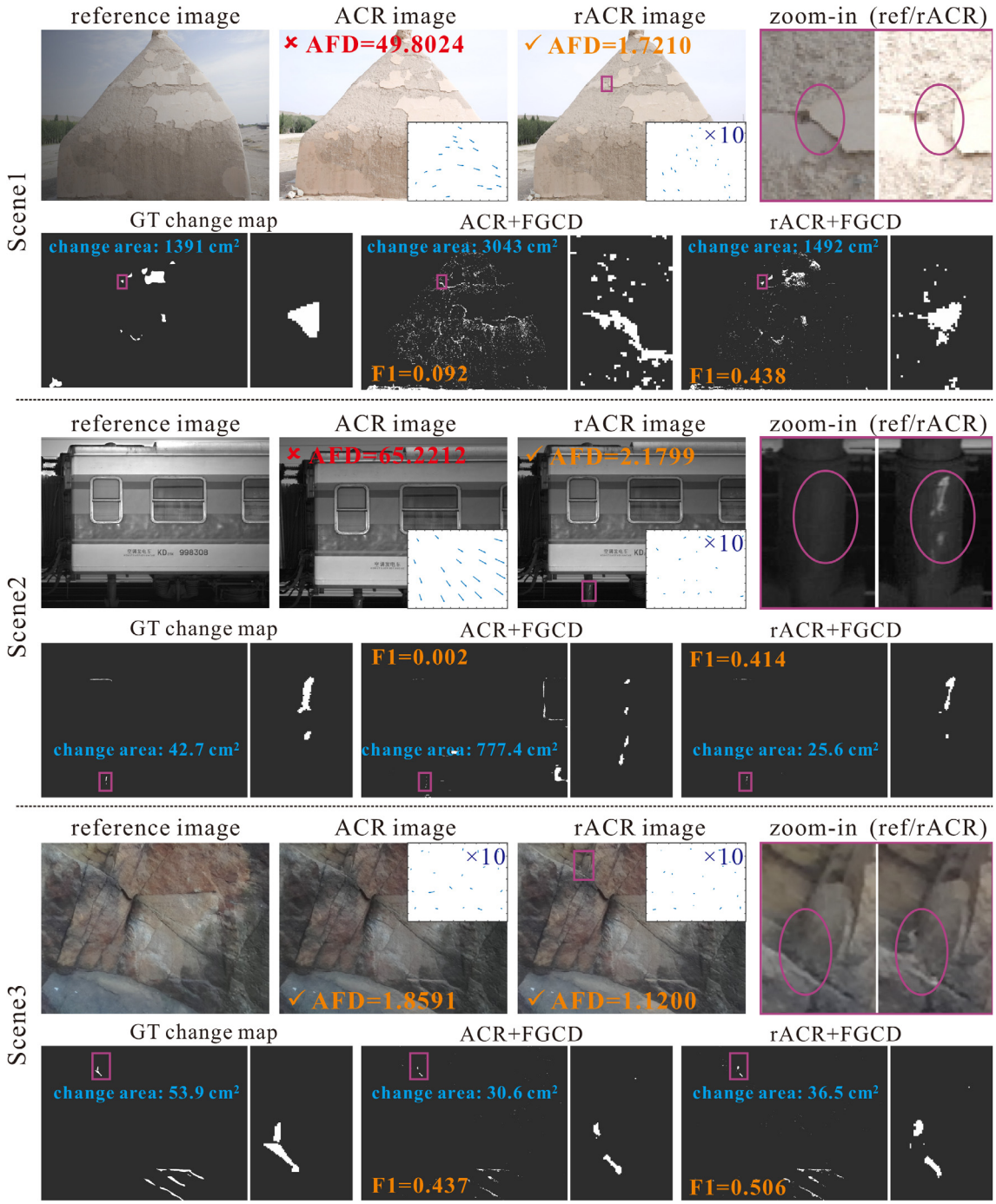


Fig. 15. Applications: FGCD of high-value outdoor scenes. Scene1: Yearly FGCD of an ancient stupa of Yuan Dynasty (1271–1368); Scene2: Running status inspection of high speed train; Scene3: Quarterly FGCD of stone inscription in The Summer Palace.

fact, FGCD plays a fundamental and important role in high-value object monitoring tasks. For instance, preventive conservation of cultural heritage needs long-term tiny change monitoring and measurement to analyze the causes of cultural relics deterioration [48,49]. Other examples include biomedical diagnosis and all important buildings (e.g., dam) monitoring, which all require reliable detection of fine-grained changes. FGCD usually has two stages, data capturing stage and change detection stage. The former aims to capture twice observation images with similar camera pose, which is just the objective of active camera relocalization (ACR). The later aims to accurately detect the scene changes from twice captured images. Obviously, accurate camera relocalization can effectively reduce the difficulty of detecting scene tiny changes.

Unfortunately, the state-of-the-art ACR method [1] cannot do well under highly varied lightings, especially for the outdoor environment, since the lighting conditions cannot be controlled. Therefore, reliable high-quality outdoor FGCD forms the practical motivation of this work. As shown in Fig. 15, we compare the FGCD performances of the proposed rACR and ACR baseline [1] under three real scenes, including an ancient stupa of Yuan Dynasty (1271–1368), a high speed train and a stone inscription in the Summer Palace. Fo each scene, we first capture the reference image, then conduct ACR or rACR to capture the current images after about one year later. Next, we employ a state-of-the-art fine-grained change detection method [3] to detect the minute changes from the reference and current images. We carefully annotate the ground truth changes of the three cases and use widely-used F1-Score (the larger the better) to evaluate the FGCD accuracy. The quantitative and visual comparison results are shown in Fig. 15. The FDF maps and AFD scores are shown in the bottom-right of ACR/rACR images. The white regions in the change map indicate the scene real changes, which are also circled in the zoom-in images. Besides, the areas of scene changes are marked in the change maps. We can see that the ACR baseline [1] fails (AFD$>$ 3) for the first two cases where lightings have changed a lot, further leading to inaccurate change detection results. On the contrary, our rACR works well under varied lightings, so the minute changes occurred in scenes can be faithfully detected. Due to the inaccurate camera relocalization, the results of ACR + FGCD obviously have more false alarm than our ours.

6. Conclusion

In this paper, we have addressed illumination robustness and convergence speed, two critical issues of active camera relocalization (ACR), by proposing rACR, a new fast and robust ACR scheme, using the correspondence of current and reference point clouds to align the two camera coordinate systems. Specifically, rACR is a two-stage procedure. Fast alignment is the first stage, where an efficient algebraic estimation of current 3D point cloud scale is proposed to physically equalize current and reference observations, and effectively generate camera adjustment motions. After the camera pose is almost aligned by fast alignment, the second stage fine-grained alignment launches by jointly optimizing the relative scale factor and pose displacement in a finer level, via scale-aware 3D point cloud alignment. Since rACR circumvents 2D feature matching and avoids slow bisection approaching, it achieves 5$\times$ faster convergence speed and better relocalization accuracy over state-of-the-art ACR competitors. Besides, the fast marching stage can also estimate the unknown hand-eye relative pose, which, as a byproduct, may further benefit the ACR accuracy and speed.

Compared to previous 2D matching based ACR strategy, the proposed rACR adopts a 3D way to achieve robustness to lighting variations. As verified by our experiment, due to the reliability of current visual SLAM methods, the relative complexity of acquiring current and reference observations using point clouds and 2D images is almost comparable. In most cases, for two arbitrarily different *stable* lighting conditions, rACR is able to relocalize the camera to the target pose. It may only fail for purely homogeneous 3D scene shapes, e.g., a ball or a plane surface. In this case, 3D point cloud has no discriminative power to establish effective matching from current and reference observations. For this situation, 3D and 2D fused strategy may be a better choice to generate some equalized virtual lighting for better ACR performance, which is our future work.

CRediT authorship contribution statement

Qian Zhang: Writing - original draft. **Wei Feng:** Methodology, Writing - review & editing. **Yi-Bo Shi:** Visualization. **Di Lin:** Data curation, Validation.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work is financially supported in part by the National Key Research and Development Program of China under Grant No. 2019YFC1520904, and the National Natural Science Foundation of China (NSFC) under Grant No. 62072334.

References

[1] F.-P. Tian, W. Feng, Q. Zhang, X. Wang, J. Sun, V. Loia, Z.-Q. Liu, Active camera relocalization from a single reference image without hand-eye calibration, IEEE Trans. Pattern Anal. Mach. Intell. 41 (12) (2019) 2791–2806.
[2] K. Sakurada, T. Okatani, K. Deguchi, Detecting changes in 3D structure of a scene from multi-view images captured by a vehicle-mounted camera, CVPR (2013).
[3] W. Feng, F.-P. Tian, Q. Zhang, N. Zhang, L. Wan, J. Sun, Fine-grained change detection of misaligned scenes with varied illuminations, in: ICCV, 2015..
[4] R. Huang, W. Feng, Z. Wang, M. Fan, L. Wan, J. Sun, Learning to detect fine-grained change under variant imaging conditions, in: ICCVW, 2017..
[5] A. Taneja, L. Ballan, M. Pollefeys, Image based detection of geometric changes in urban environments, in: ICCV, 2011..
[6] S. Bae, A. Agarwala, F. Durand, Computational rephotography, ACM Trans. Graphics 29 (3) (2010) 1–15.
[7] J. Engel, V. Koltun, D. Cremers, Direct sparse odometry, IEEE Trans. Pattern Anal. Mach. Intell. 40 (3) (2018) 611–625.
[8] T. Sattler, M. Havlena, K. Schindler, M. Pollefeys, Large-scale location recognition and the geometric burstiness problem, CVPR, 2016.
[9] A.R. Zamir, M. Shah, Image geo-localization based on multiple nearest neighbor feature matching using generalized graphs, IEEE Trans. Pattern Anal. Mach. Intell. 36 (8) (2014) 1546–1558.
[10] G.R. Abner, K. Pushmeet, G. Ben, S. Jamie, S. Toby, F. Andrew, I. Shahram, Multi-output learning for camera relocalization, CVPR (2014).
[11] A. Kendall, R. Cipolla, Modelling uncertainty in deep learning for camera relocalization, ICRA, 2016.
[12] D. Miao, F.-P. Tian, W. Feng, Active camera relocalization with RGBD camera from a single 2D image, ICASSP (2018).
[13] Y.-B. Shi, F.-P. Tian, D. Miao, W. Feng, Fast and reliable computational rephotography on mobile device, in: ICME, 2018..
[14] R.Y. Tsai, R.K. Lenz, A new technique for fully autonomous and efficient 3D robotics hand/eye calibration, IEEE Trans. Robot. Autom. 5 (3) (1989) 345–358.
[15] J. Heller, M. Havlena, T. Pajdla, Globally optimal hand-eye calibration using branch-and-bound, IEEE Trans. Pattern Anal. Mach. Intell. 38 (5) (2016) 1027–1033.
[16] Q. Zhang, W. Feng, L. Wan, F.-P. Tian, P. Tan, Active recurrence of lighting condition for fine-grained change detection, in: International Joint Conference on Artificial Intelligence, 2018..
[17] J. Engel, J. Stückler, D. Cremers, Large-scale direct SLAM with stereo cameras, IROS, 2015.
[18] F. Endres, J. Hess, J. Sturm, D. Cremers, W. Burgard, 3-D mapping with an RGB-D camera, IEEE Trans. Rob. 30 (1) (2014) 177–187.

[19] R. Mur-Artal, J.D. Tardos, ORB-SLAM2: An open-source SLAM system for monocular, stereo and RGB-D cameras, IEEE Trans. Rob. 33 (5) (2017) 1255–1262.
[20] R. Wang, M. Schwrer, D. Cremers, Stereo DSO: Large-scale direct sparse visual odometry with stereo cameras, in: ICCV, 2017..
[21] R.A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A.J. Davison, P. Kohi, J. Shotton, S. Hodges, A. Fitzgibbon, KinectFusion: Real-time dense surface mapping and tracking, in: ISMAR, 2011..
[22] T. Whelan, M. Kaess, H. Johannsson, M. Fallon, J.J. Leonard, J. McDonald, Real-time large-scale dense rgb-d slam with volumetric fusion, Int. J. Robot. Res. 34 (4–5) (2015) 598–626.
[23] T. Whelan, R.F. Salas-Moreno, B. Glocker, A.J. Davison, S. Leutenegger, ElasticFusion: Real-time dense slam and light source estimation, Int. J. Robot. Res. 35 (14) (2016) 1697–1716.
[24] A. Dai, M. Niener, M. Zollhfer, S. Izadi, C. Theobalt, BundleFusion: Real-time globally consistent 3D reconstruction using on-the-fly surface reintegration, ACM Trans. Graphics 36 (4) (2017) 76a.
[25] A. Kendall, M. Grimes, R. Cipolla, PoseNet: A convolutional network for real-time 6-DoF camera relocalization, in: ICCV, 2015..
[26] G. Tam, Z.Q. Cheng, Y.K. Lai, F.C. Langbein, Y. Liu, D. Marshall, R. Martin, X.F. Sun, P.L. Rosin, Registration of 3D point clouds and meshes: A survey from rigid to nonrigid, IEEE Trans. Visual Comput. Graphics 19 (7) (2013) 1199–1217.
[27] S. Irani, P. Raghavan, Combinatorial and experimental results for randomized point matching algorithms, Comput. Geometry 12 (1–2) (1999) 17–31.
[28] C.S. Chen, Y.P. Hung, J.B. Cheng, Ransac-based darces: A new approach to fast automatic registration of partially overlapping range images, IEEE Trans. Pattern Anal. Mach. Intell. 21 (11) (1999) 1229–1234.
[29] D. Aiger, N.J. Mitra, D. Cohen-Or, 4-points congruent sets for robust pairwise surface registration, ACM Transactions on Graphics..
[30] N. Mellado, D. Aiger, N.J. Mitra, Super 4pcs fast global pointcloud registration via smart indexing, Comput. Graphics Forum 33 (5) (2014) 205–215.
[31] P.J. Besl, N.D. McKay, Method for registration of 3-D shapes, International Society for Optics and Photonics, 1992.
[32] Y. Chen, G. Medioni, Object modelling by registration of multiple range images, Image Vis. Comput. 10 (3) (1992) 145–155.
[33] S. Rusinkiewicz, M. Levoy, Efficient variants of the ICP algorithm, in: 3DIM, 2001..
[34] S. Bouaziz, A. Tagliasacchi, M. Pauly, Sparse iterative closest point, SGP (2013).
[35] Y. Tian, B. Fan, F. Wu, L2-net: Deep learning of discriminative patch descriptor in euclidean space, CVPR (2017) 661–669.
[36] Y. Verdie, K.M. Yi, P. Fua, V. Lepetit, Tilde: A temporally invariant learned detector, CVPR, 2015.
[37] K.M. Yi, E. Trulls, V. Lepetit, P. Fua, Lift: Learned invariant feature transform, in: ECCV, Springer, 2016, pp. 467–483..
[38] Z. Luo, L. Zhou, X. Bai, H. Chen, J. Zhang, Y. Yao, S. Li, T. Fang, L. Quan, Aslfeat: Learning local features of accurate shape and localization, CVPR..
[39] H. Noh, A. Araujo, J. Sim, T. Weyand, B. Han, Large-scale image retrieval with attentive deep local features, in: ICCV, 2017, pp. 3456–3465..
[40] M. Teichmann, A. Araujo, M. Zhu, J. Sim, Detect-to-retrieve: Efficient regional aggregation for image search, CVPR, 2019.
[41] P.H. Chen, Z.X. Luo, Z.K. Huang, C. Yang, K.W. Chen, If-net: An illumination-invariant feature network, in: ICRA, IEEE, 2020, pp. 8630–8636..
[42] H. Su, S. Maji, E. Kalogerakis, E. Learned-Miller, Multi-view convolutional neural networks for 3d shape recognition, in: ICCV, 2015, pp. 945–953..
[43] Z. Gojcic, C. Zhou, J.D. Wegner, A. Wieser, The perfect match: 3d point cloud matching with smoothed densities, CVPR, 2019, pp. 5545–5554.
[44] Y.C. Liu, B. Fan, S.M. Xiang, C.H. Pan, Relation-shape convolutional neural network for point cloud analysis, CVPR (2019) 8895–8904.
[45] S. Qiu, S. Anwar, N. Barnes, Semantic segmentation for real point cloud scenes via bilateral augmentation and adaptive fusion, in: in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 1757–1767.
[46] D. Nistér, An efficient solution to the five-point relative pose problem, IEEE Trans. Pattern Anal. Mach. Intell. 26 (6) (2004) 756–770.
[47] A. Taneja, L. Ballan, M. Pollefeys, City-scale change detection in cadastral 3D models using images, CVPR, 2013.
[48] S. Staniforth (Ed.), Historical perspectives on preventive conservation, Getty Conservation Institute, 2013..
[49] H. Wirilander, Preventive conservation: A key method to ensure cultural heritage's authenticity and integrity in preservation process, E-Conservation Magazine 6 (24)..

Qian Zhang received the B.E. degree in software engineering from School of Computer Software, Xidian University, China, in 2013, and is currently working toward the Ph.D. degree at the School of Computer Science and Technology, College of Intelligence and Computing, Tianjin University, China. His research interests include computer vision and machine learning, especially on active vision, image-based 3D reconstruction, and fine-grained change detection.

Wei Feng received the BS and MPhil degrees in computer science from Northwestern Polytechnical University, China, in 2000 and 2003, respectively, and the PhD degree in computer science from City University of Hong Kong in 2008. From 2008 to 2010, he was a research fellow at the Chinese University of Hong Kong and City University of Hong Kong. He is now a full professor in the School of Computer Science and Technology, Tianjin University, China. His major research interests are active robotic vision and visual intelligence, specifically including active camera relocalization and lighting recurrence, general Markov Random Fields modeling, energy minimization, active 3D scene perception, SLAM, and generic pattern recognition. Recently, he focuses on solving preventive conservation problems of cultural heritages via computer vision and machine learning. He is the Associate Editor of Neurocomputing and Journal of Ambient Intelligence and Humanized Computing.

Yi-Bo Shi received the B.E. and M.E. degrees in school of computer science and technology, Tianjin University in 2017 and 2020. His research interests focus on active vision, image-based 3D reconstruction, visual SLAM and related topics.

Di Lin received the bachelor's degree in software engineering from Sun Yatsen University, Guangzhou, China, in 2012, and the Ph.D. degree from The Chinese University of Hong Kong, Hong Kong, in 2016. He is currently an Associate Professor with the College of Intelligence and Computing, Tianjin University, Tianjin, China. His research interests are computer vision and machine learning.