6-DOF Image Localization From Massive Geo-Tagged Reference Images

Yafei Song, Xiaowu Chen, *Senior Member, IEEE*, Xiaogang Wang, Yu Zhang, and Jia Li, *Senior Member, IEEE*

***Abstract*—The 6-degrees of freedom (DOF) image localization, which aims to calculate the spatial position and rotation of a camera, is a challenging problem for most location-based services. In existing approaches, this problem is often tackled by finding the matches between 2D image points and 3D structure points so as to derive the location information via direct linear transformation algorithm. However, as these 2D-to-3D-based approaches need to reconstruct the 3D structure points of the scene, they may not be flexible enough to employ massive and increasing geo-tagged data. To this end, this paper presents a novel approach for 6-DOF image localization by fusing candidate poses relative to reference images. In this approach, we propose to localize an input image according to the position and rotation information of multiple geo-tagged images retrieved from a reference dataset. From the reference images, an efficient relative pose estimation algorithm is proposed to derive a set of candidate poses for the input image. Each candidate pose encodes the relative rotation and direction of the input image with respect to a specific reference image. Finally, these candidate poses can be fused together by minimizing a well-defined geometry error so that the 6-DOF location of the input image is effectively derived. Experimental results show that our method can obtain satisfactory localization accuracy. In addition, the proposed relative pose estimation algorithm is much faster than existing work.**

***Index Terms*—Image localization, one-sided radial fundamental matrix estimation, relative pose estimation.**

I. Introduction

LOCATION information of an image is important for various location-based services and applications, such as travel recommendation [1], image navigation [2], location guided image retrieval [3], [4], augmented reality [5], [6], and autonomous navigation [7], [8]. The 6-DOF image location information actually contains 3-DOF spatial position and 3-DOF rotation in the world coordinate. This information is difficult to be labeled manually as users have difficulty in adjusting the 6-DOF parameters simultaneously. Moreover, it will take huge time to manually label the rapidly increasing image data, the majority of which lack location information. Therefore, it has certain practical significance to localize images automatically.

To alleviate this problem, various methods have been proposed in existing literature. Some of them perform 6-DOF image localization by employing 3D point cloud model of the scene. The point cloud model is usually reconstructed from reference images via structure from motion (SfM) algorithm, e.g. [9], [10]. With the point cloud model, the localization problem can be formulated as a 2D-to-3D registration process [11]. Under this formulation, previous methods first find the matches between 2D image points and 3D structure points according to their features similarity. The point matches can be further utilized to calculate the location information using direct linear transformation (DLT) algorithm [12]. All of these steps can be embedded into a RANdom SAmple Consensus (RANSAC) [13] iteration process for robustness. Most researchers focus on how to find robust 2D-to-3D point matches efficiently and have proposed various methods, e.g. [7], [11], [14]–[17]. Benefitting from the pre-processed 3D point cloud model, these model-based methods can well localize the input image. However, as these methods need to reconstruct the 3D structure points of the scene, they are not flexible with massive and increasing geo-tagged data. Moreover, it is also time-consuming to reconstruct the 3D point cloud model.

Besides 3D point cloud model based methods, some methods [18]–[21] recognize the landmarks in the input image, then transfer the position of the landmarks to the input image. Some others [22]–[25] first retrieve or select the nearest neighbors of the input image from the reference dataset by measuring the visual similarity. Then the final position can be calculated by fusing the position of retrieved neighbors. These methods can be scalable benefiting from scalable image retrieval methods, e.g. [26], [27]. However, they mostly can only obtain the coarse position but lack the ability to calculate the accurate 6-DOF location, which limits their practical applicability.

In order to explore a more flexible method to obtain the 6-DOF location, we find that there have appeared massive geo-tagged images, e.g. google street view. As an alternate solution, an input image can be localized by exploiting these geo-tagged data directly. Inspired by recent image label transfer works [28], [29] on scene parsing, our basic idea is to transfer the 6-DOF location information to the input image by fusing its poses relative to the reference images, which is intuitively illustrated in Fig. 1. From this idea, given an input image, its nearest neighbors are retrieved from a large reference dataset using a content based

Manuscript received October 07, 2015; revised January 10, 2016, March 13, 2016, and April 23, 2016; accepted May 01, 2016. Date of publication May 13, 2016; date of current version July 15, 2016. This work was supported in part by the National Natural Science Foundation of China under Grant 61325011, Grant 61532003, and Grant 61421003. This paper was presented in part at the IEEE International Conference on Multimedia Big Data, Beijing, China, April 2015. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Shu-Ching Chen. *(Corresponding author: Xiaowu Chen.)*

Y. Song, X. Chen, X. Wang, and Y. Zhang are with the State Key Laboratory of Virtual Reality Technology and Systems, School of Computer Science and Engineering, Beihang University, Beijing 100191, China (e-mail: songyf@buaa.edu.cn; chen@buaa.edu.cn; wangxiaogang@buaa.edu.cn; octopus@buaa.edu.cn).

J. Li is with the State Key Laboratory of Virtual Reality Technology and Systems, School of Computer Science and Engineering, Beihang University, Beijing 100191, China, and also with the International Research Institute for Multidisciplinary Science, Beihang University, Beijing 100191, China (e-mail: jiali@buaa.edu.cn).

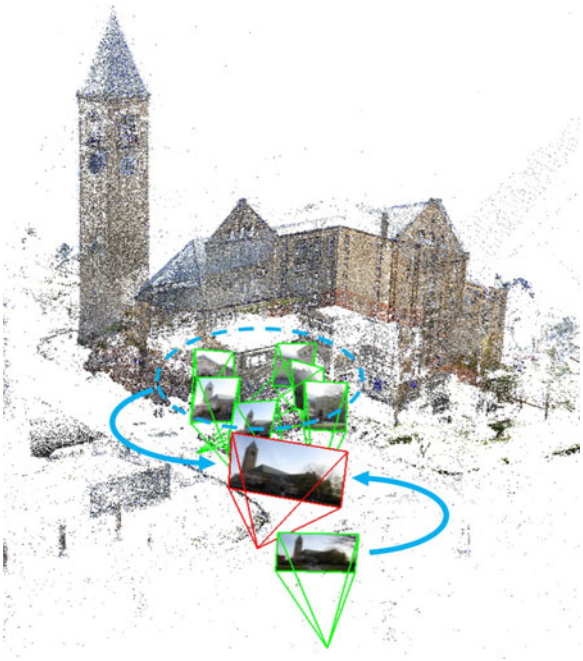


Fig. 1. Our basic idea is to localize an image by transferring the location information of reference images to it. To this end, we first estimate the relative pose between the input image (with red lines) and each of its reference images (with green lines), and then fuse all these candidate poses to obtain the final 6-DOF location. As a result, our method can flexibly exploit the incremental geo-tagged data.

image retrieval algorithm. An efficient algorithm is further proposed to estimate the pose of the input image relative to each of the nearest neighbors so as to obtain a set of candidate poses. At last, these candidate poses are fused together to figure out the final 6-DOF location by minimizing a well-defined geometry error.

Our contributions mainly include two aspects: 1) we propose an efficient algorithm to estimate the relative pose between a calibrated reference image and an uncalibrated input image so as to obtain several candidate poses of the input image; 2) in order to effectively figure out the 6-DOF location of the input image, we define and minimize a geometry error to fuse these candidate poses.

The rest of this paper is organized as follows. Section II reviews some related works. Section III first overviews the whole pipeline of our method and then explains each step in detail. Finally we show the experimental results in Section IV and conclude this paper in Section V.

II. Related Work

There are mainly three classes of methods related to our work, including image localization methods based on 3D point cloud model, localization via landmark recognition, and relative pose estimation.

A. Image Localization Based on 3D Point Cloud Model

These methods commonly formulate the localization task as a 2D-to-3D registration problem. They first find a set of point matches between 2D points in the input image and 3D structure points in the point cloud model by measuring the similarity among the features of each point. Before that, the point cloud model should be reconstructed via SfM systems, e.g. [9], [10]. Then the accurate 6-DOF location of the input image can be estimated via DLT algorithm [12]. The key challenge of these methods is how to efficiently find abundant and robust point matches. To this end, Irschara *et al.* [11] apply image retrieval techniques to find the nearest views of the input image so as to reduce searching space. The nearest views are generated from the 3D point cloud. The retrieval step of this work is similar to ours, which also verifies that it can be applied to accelerate 6-DOF image localization. However they use 3D point cloud to generate the visual documents of synthetic views while we directly use the original reference images. Sattler *et al.* [14] evaluate the performance of direct 2D-to-3D matching method by applying a direct matching framework based on visual vocabulary quantization and a prioritized correspondence search. This work explores the upper limit where 2D-to-3D methods can reach. Lim *et al.* [7] use inexpensive binary feature descriptors instead of scale-invariant features so as to enable real-time localization. Li *et al.* [15] utilize co-occurrence prior and bidirectional matching to efficiently find point matches which enables localizing images in worldwide scale. MiddelBerg *et al.* [16] develop a system under Client/Server framework. This system can simultaneously benefit from the scalability of a global localization server and the precision of a local pose tracker on a mobile device. Unlike previous works, Donoser and Schmalstieg [17] formulate the point matching process as a discriminative classification problem. These state-of-the-art methods can perform 6-DOF image localization task well by exploiting the 3D point cloud. However, it is time consuming to reconstruct the 3D point cloud. Moreover, as the geo-tagged data are usually dynamically growing in real world, e.g. Google Street View, these model-based methods are not flexible to immediately exploit the latest data.

B. Localization via Landmark Recognition

Some methods [18]–[21] localize an input image by recognizing the landmarks in it and transferring the landmarks' position to the input image. Before that, the location of each landmark should be annotated. Li *et al.* [18] formulate the task as a classification problem and recognize 500 categories of landmarks on a large dataset. Hao *et al.* [19] introduce the 3D visual phrase which is a triangular facet on the surface of a reconstructed 3D landmark model. The 3D visual phrase is further exploited to improve landmark recognition accuracy. Bergamo *et al.* [20] design a new discriminative codebook of local feature descriptors for scalable landmark classification. Zhu *et al.* [21] propose hierarchical multi-modal exemplar features to characterize landmark images so as to achieve low storage overhead and high recognition efficiency. Some others [23]–[25] retrieve several nearest neighbors of the input image from the reference database. Then the final position can be calculated by fusing the geo-information of all retrieved neighbors. Chen *et al.* [23] publish a city scale street view dataset and exploit user's

position priors to improve the recall rates on mobile devices. Cao and Snavely [24] embed the images in a graph so as to improve the bag-of-visual-words based location recognition method. Zamir *et al.* [25] query the input image's scale-invariant feature transform (SIFT) descriptors in the indexed tree of reference images. Then an associated voting scheme is utilized to determine the final position. Zhang and Kosecka [22] also localize an input image by estimating relative pose to each of its references. However, they mostly can only obtain the position of the input image. Benefitting from large-scale image retrieval algorithms, e.g. LIRe [27], these methods are always good at handling large-scale data. However, they are incapable of obtaining the accurate 6-DOF location of the input image. Besides these methods, Baatz *et al.* [30] can obtain the rotation and position information, however, their method assumes that the input image is calibrated and contains facade with grids of windows.

C. Relative Pose Estimation

In order to transfer the 6-DOF locations of neighbor images to the input image, we resort to estimate the relative pose between the input image and each of its neighbors. Given a pair of images, different algorithms have been proposed to estimate their relative pose under different configurations. For a pair of calibrated images, the relative pose can be estimated using 5-point algorithm, which has been well-studied. Nistér *et al.* [31] present an efficient algorithmic solution to tackle this problem. By solving a tenth degree polynomial in closed form, the algorithm is well suited for numerical implementation that also corresponds to the inherent complexity of the problem. For a pair of perspective images with known intrinsic parameters except for an unknown common focal length, Stewénius *et al.* [32] present an efficient solver given six corresponding points. However, in our situation, the input image is taken by users which should be assumed to be uncalibrated and usually have different focal length with reference images, thus the methods above can not be applied directly. To this end, we propose to estimate the intrinsic parameters first and transform the problem to the configuration of two calibrated images. Bujnak *et al.* [33] use Gröebner basis to address calibrated-uncalibrated setting and apply it to 3D reconstruction, who assume that the uncalibrated image only has one intrinsic parameter, i.e. focal length. However, as radial distortion obviously affects the relative pose estimation, it is also estimated in our method. Under this configuration, Brito *et al.* [34] solve a high-order polynomial system to obtain the fundamental matrix and minimize an algebraic error to extract focal length, which is time-consuming and slower than our algorithm.

III. Image Localization From Relative Poses

In this section we first overview our method in Section III-A, then present the details of each step, including nearest neighbors retrieval in Section III-B, fundamental matrix estimation in Section III-C, relative pose estimation in Section III-D and final location determination in Section III-E.

A. Overview

We propose to localize an input image by fusing candidate poses relative to reference images. The work-flow of our method is illustrated in Fig. 2. First of all, several nearest neighbors of the input image are retrieved from a reference dataset in Section III-B. This step can be performed using a bag-of-visual-words based image retrieval algorithm [27]. For each of these neighbors, the fundamental matrix between it and the input image is estimated in Section III-C. In our configuration, all the images in reference dataset have been calibrated during building the dataset, while the input image is not calibrated. Moreover, in order to robustly estimate the relative pose, we assume that the input image has two intrinsic parameters, including focal length and one radial distortion parameter. Under this configuration, the fundamental matrix is modified to form one-sided radial (OSR) fundamental matrix. A fast algorithm is further proposed to estimate the OSR fundamental matrix and extract the intrinsic parameters of the input image. The problem is then transformed to calibrated relative pose estimation problem. The pose of the input image relative to each of its neighbors can be obtained in Section III-D. At last, in order to fuse the candidate poses effectively, a geometry error is defined which encodes the candidate poses additionally with a regularization term. By minimizing this error, the final optimal 6-DOF location of the input image can be figured out in Section III-E.

B. Nearest Neighbors Retrieval

Our basic idea is to estimate the 6-DOF location of an input image based on the reference images. However, as the reference images may dynamically increase with time, it is time consuming to use all the reference images or construct a global model, e.g. a point cloud model. To this end, we propose first to retrieve some nearest neighbors of the input image according to the visual similarity. The similarity is usually measured by computing the distance between global features or local features along with bag-of-visual-words. In our case, the input image and its neighbors should have some duplicated areas, which implies that they are captured at the nearby places. That is exactly the objective of content based image retrieval algorithms.

Benefiting from the achievements of content based image retrieval techniques, there are many algorithms satisfying our requirement. Without loss of generality, we employ LIRe [27] which provides a library of basic and advanced functions for visual information retrieval. In consideration of the computing efficiency, SURF features [35] are extracted as the local features from the images in reference dataset. K-means is subsequently applied on the extracted local features to learn visual words. The learned visual words can be used to establish a histogram for each image. The histogram is usually called visual document of its corresponding image. The distance between visual documents can be used to measure the similarity between images. By ranking the similarity, several neighbors of the input image can be retrieved. Fig. 3 shows some input images and the retrieved neighbors. The retrieved nearest neighbors contain some real positive neighbors of the input image, and some false positive

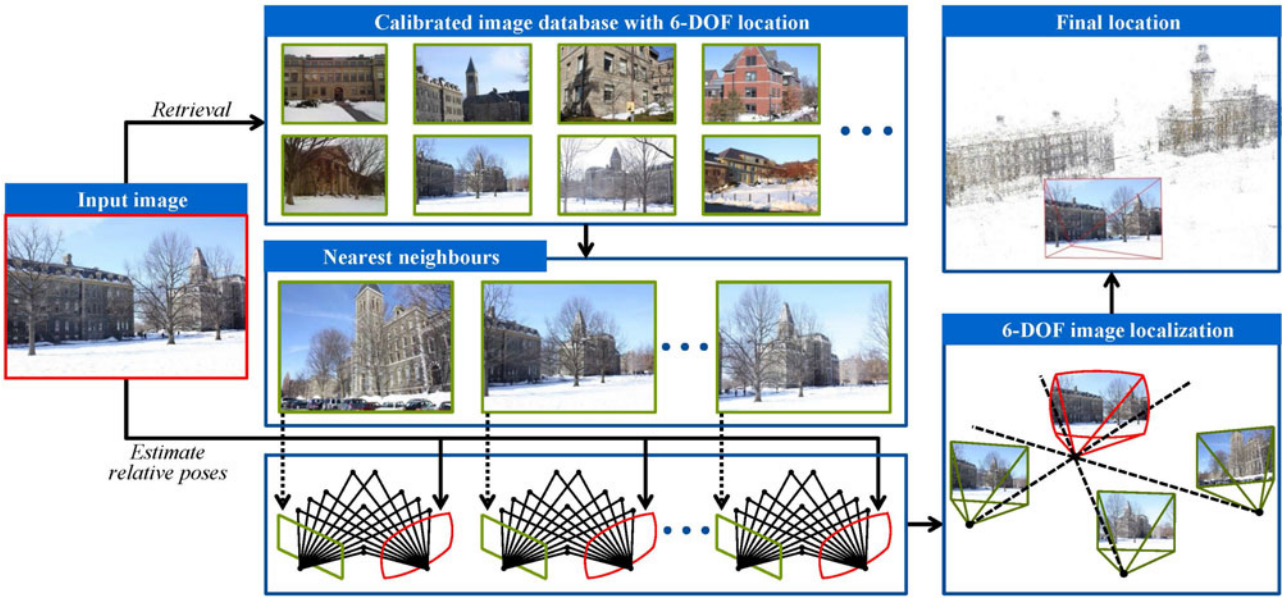


Fig. 2. Framework of our method. Given an input image, first, its nearest neighbors are retrieved from the reference database using a content-based image retrieval algorithm. Each neighbor has 6-DOF location information. Then, we estimate relative pose between the input image and each of its neighbors to obtain several candidate poses. At last, these candidate poses are integrated to get the final optimal 6-DOF location.

Fig. 3. Some image retrieval results. The retrieved images contain real neighbors of the input image, and some false positive cases as well. The false positive cases can be removed after applying epipolar constraints subsequently.

cases as well. However, most of the false positive cases can be removed after applying epipolar constraint [12] subsequently.

C. OSR Fundamental Matrix Estimation

In this work, all images are assumed to be captured by pinhole camera. For a pair of images capturing some overlap contents, there exist many pairs of matched image points. For a pair of matched image points $\langle \mathbf{x}_L, \mathbf{x}_R \rangle$, they are both projected from an identical 3D point $\mathbf{X}$. The corresponding focal centers of the two cameras are denoted as $\mathbf{O}_L$ and $\mathbf{O}_R$ respectively. The epipolar constraint implies that the line $\mathbf{O}_L$-$\mathbf{x}_L$ and the line $\mathbf{O}_R$-$\mathbf{x}_R$ should intersect at $\mathbf{X}$. The right camera sees the line $\mathbf{O}_L$-$\mathbf{x}_L$ as a line in its image plane, which is called the epipolar line of the point $\mathbf{x}_R$. Symmetrically, the epipolar line of the point $\mathbf{x}_L$

Fig. 4. Fundamental matrix can encode epipolar constraint between a pair of images captured by ideal pinhole camera, e.g. reference and undistorted input image. If one image is assumed to have radial distortion, i.e. input image in this paper, the OSR fundamental matrix is further introduced.

also can be defined. The details can be found in [12]. Actually, the 3×3 fundamental matrix $\mathbf{F}$ encapsulates the epipolar constraint between the two images, which only depends on the intrinsic parameters of the cameras and their relative pose. In our situation, the reference images in dataset can be easily calibrated during collecting them. However, it is better to assume that the input image is uncalibrated as the intrinsic parameters are not easy to obtain for common users. As a result, the intrinsic parameters must be estimated first. According to the imaging process, the focal length certainty should be considered as it is the most important intrinsic parameter. Moreover, as shown in Fig. 4, the uncalibrated input image usually obviously suffer from radial distortion so as to influence the accuracy of relative pose estimation. To this end, we also assume that the intrinsic parameters contain one radial distortion parameter. Under this configuration, a OSR fundamental matrix is derived to simultaneously capture the epipolar constraint and the radial distortion. After estimating the OSR fundamental matrix, the intrinsic parameters of the input image can be extracted from it. Then the relative pose can be estimated the same as in traditional 5-point algorithm [31].

Given a pair of images which are composed of one calibrated reference image and one uncalibrated input image, we first find the initial point matches between the two images using *ratio test* algorithm by measuring Euclidean distance of SIFT features [36]. For each feature point $\mathcal{Q}$ in the input image, the algorithm first finds the closest point $\mathcal{P}$ and the second closest point $\mathcal{P}_s$ in its neighbor image. Then the closest point $\mathcal{P}$ is taken as a point that matches $\mathcal{Q}$ if the ratio of Euclidean distance from the closest point to the second closest point is less than a predefined threshold T_h

$$\frac{\boldsymbol{\mathcal{D}}\left(\mathcal{P},\mathcal{Q}\right)}{\boldsymbol{\mathcal{D}}\left(\mathcal{P}_s,\mathcal{Q}\right)} < T_h \tag{1}$$

where an empirical value of T_h is 0.5. The idea behind ratio test algorithm is that a real matched point should have distinct shorter distance from query point $\mathcal{Q}$ than others in feature space. Otherwise, the matched point is usually a false positive. As a result, this rule can pick out discriminative feature points and a set of robust point matches can be obtained.

A pair of matched points is denoted as $\langle \mathcal{P}, \mathcal{Q} \rangle$ and the set of matched points are denoted as $\mathcal{M} = \{\langle \mathcal{P}_i, \mathcal{Q}_i \rangle | i = 1, 2, ..., m\}$. For a specific pair of matched points $\langle \mathcal{P}, \mathcal{Q} \rangle$, we denote their homogeneous coordinates in image as $\mathbf{p} \propto (x_p, y_p, 1)^T$ and $\mathbf{q} \propto (x_q, y_q, 1)^T$, respectively. As we suppose that the input image has one radial distortion parameter λ, the undistorted image coordinate of point $\mathcal{Q}$ can be given as $\mathbf{q}_u \propto \left(x_q, y_q, 1 + \lambda r^2\right)^T$, where r is the distance between the feature point $\mathcal{Q}$ and the distortion center (u, v), which can be computed as

$$r^2 = (x_q - u)^2 + (y_q - v)^2 . \tag{2}$$

The distortion center (u, v) is assumed in image center.

According to the epipolar constraint, a pair of matched points $\langle \mathcal{P}, \mathcal{Q} \rangle$ satisfies the linear equation

$$\mathbf{p}^T \mathbf{F} \mathbf{q}_u = 0. \tag{3}$$

As $\mathbf{p} \propto (x_p, y_p, 1)^T$ and $\mathbf{q}_u \propto \left(x_q, y_q, 1 + \lambda r^2\right)^T$, the equation can be written as

$$\mathbf{p}^T \mathbf{F} \begin{pmatrix} x_q \\ y_q \\ 1 + \lambda r^2 \end{pmatrix} = \mathbf{p}^T [\mathbf{f}_1 \ \mathbf{f}_2 \ \mathbf{f}_3 \ \lambda \mathbf{f}_3] \begin{pmatrix} x_q \\ y_q \\ 1 \\ r^2 \end{pmatrix} = 0 \tag{4}$$

where $\mathbf{f}_i$ is the ith column of the fundamental matrix $\mathbf{F} = [\mathbf{f}_1 \ \mathbf{f}_2 \ \mathbf{f}_3]$. We denote the matrix $[\mathbf{f}_1 \ \mathbf{f}_2 \ \mathbf{f}_3 \ \lambda \mathbf{f}_3]$ as $\mathbf{V}$ and call it the OSR fundamental matrix in order to distinguish it from the original fundamental matrix $\mathbf{F}$. Furthermore, $\mathbf{V}$ is a matrix with rank two as $\mathbf{F}$ has rank two.

Inspired by the 8-point algorithm to estimate the fundamental matrix [12], as (4) is linear for each element of $\mathbf{V}$ and $\mathbf{V}$ is non-zero, we can first estimate $\mathbf{V}_{3\times4}$ using $11 = 3 \times 4 - 1$ linear equations. These equation can be deduced from 11 pairs of matched points. Then the low-rank constraint on $\mathbf{V}$ can be enforced using singular value decomposition (SVD) algorithm. According to the epipolar geometry constraint, given 11 pairs of matched points, 11 linear equations are correspondingly obtained as

$$\mathbf{A}\mathbf{v} = \mathbf{0} \tag{5}$$

where $\mathbf{A}$ is an 11×12 coefficient matrix and $\mathbf{v}$ is the vector version of $\mathbf{V}$ in row major order. Moreover, each row of $\mathbf{A}$ is $[x_{p_i} x_{q_i}, x_{p_i} y_{q_i}, x_{p_i}, x_{p_i} r_i^2, y_{p_i} x_{q_i}, y_{p_i} y_{q_i}, y_{p_i}, y_{p_i} r_i^2, x_{q_i}, y_{q_i}, 1, r_i^2]$ which is corresponding to a pair of matched points $\langle \mathcal{P}_i, \mathcal{Q}_i \rangle$. Then SVD algorithm can be applied to solve (5). Actually we decompose $\mathbf{A}$ via SVD and obtain the initial estimation of $\mathbf{v}$ by picking out the right-singular vector corresponding to the smallest singular value.

As the rank of $\mathbf{F}$ is two and $\mathbf{V} = [\mathbf{f}_1 \ \mathbf{f}_2 \ \mathbf{f}_3 \ \lambda \mathbf{f}_3]$, $\mathbf{V}$ also has rank two. However, the estimated matrix $\mathbf{V}$ may not satisfy these constraints, thus we should enforce them. As we know, SVD can be used to calculate the low-rank matrix which is closest to the original matrix measured by Frobenius norm. For a given matrix, its closest matrix with rank of c can be obtained by retaining the c largest singular value and setting the others as zero. Therefore the SVD algorithm can be applied to estimate $\mathbf{F}$ and $\mathbf{V}$. First, we get the matrix with rank one which is closest to the last two columns of $\mathbf{V}$ via SVD. As the two columns are linear dependent, the ratio between them is λ. Then, we figure out the matrix with rank two, which is closest to the first three columns of $\mathbf{V}$, as final estimated $\mathbf{F}$. As λ has been estimated, the final $\mathbf{V}$ can be obtained.

Moreover, the OSR matrix estimation algorithm is embedded in an RANSAC iteration process for robustness. Actually, in each iteration, 11 pairs of matched points are random selected to estimate an OSR fundamental matrix as a candidate. Each candidate matrix is evaluated by counting the inlier matched points. The matrix with most inliers is selected as final result. A pair of matched points is regarded as inlier if they satisfy the epipolar geometry constraint, actually the equation (3). For robust numerical computation, the constraint is adjusted as if the epipolar error is smaller than a threshold. The epipolar error is the distance between the image point and its corresponding epipolar line in the image plane. The threshold is empirically set as 9.0 in our experiments. As shown in Fig. 3, the retrieved results usually contain some false positives. As the false positives usually have less point matches, the neighbors with less than 20 inlier matched points are discarded.

D. Relative Pose Estimation

As stated before, the input image has two intrinsic parameters, focal length f and one radial distortion λ, where λ has been estimated in Section III-C during estimating the fundamental matrix. We will present the details to estimate the focal length f and relative pose from fundamental matrix $\mathbf{F}$. Since the images in database are fully calibrated, their intrinsic parameter matrix can be regarded as an identity matrix and the essential matrix $\mathbf{E}$ can be written as

$$\mathbf{E} = \mathbf{F}\mathbf{K} \tag{6}$$

where $\mathbf{K}$ is the intrinsic parameter matrix of the input image. As λ is known, $\mathbf{K}$ can be regarded as a diagonal matrix with diagonal elements $f, f, 1$ in turn.

An essential matrix has rank two and has two equal non-zero singular values [12]. That is to say, a real non-zero 3×3 matrix $\mathbf{E}$ is an essential matrix if and only if it satisfies the equation

$$2\mathbf{E}\mathbf{E}^T\mathbf{E} - \mathbf{tr}\left(\mathbf{E}\mathbf{E}^T\right)\mathbf{E} = \mathbf{0}. \tag{7}$$

(6) is substituted into (7) and obtain

$$2\mathbf{F}\mathbf{K}\mathbf{K}^T\mathbf{F}^T\mathbf{F} - \mathbf{tr}\left(\mathbf{F}\mathbf{K}\mathbf{K}^T\mathbf{F}^T\right)\mathbf{F} = \mathbf{0}. \tag{8}$$

(8) can be expanded to get nine linear equations of f^2 and correspondingly obtain nine estimated values of f. The first row and first column is taken as an example and the others are analogous. The corresponding equation is

$$f^2\left(F_{11}\mathbf{f}_1^T\mathbf{f}_1 + 2F_{12}\mathbf{f}_1^T\mathbf{f}_2 - F_{11}\mathbf{f}_2^T\mathbf{f}_2\right) = F_{11}\mathbf{f}_3^T\mathbf{f}_3 - 2F_{13}\mathbf{f}_1^T\mathbf{f}_3. \tag{9}$$

As focal length $f > 0$, f is

$$f = \sqrt{\frac{F_{11}\mathbf{f}_3^T\mathbf{f}_3 - 2F_{13}\mathbf{f}_1^T\mathbf{f}_3}{2F_{12}\mathbf{f}_1^T\mathbf{f}_2 + F_{11}\left(\mathbf{f}_1^T\mathbf{f}_1 - \mathbf{f}_2^T\mathbf{f}_2\right)}} \tag{10}$$

where F_{ij} is the ith row and jth column value of $\mathbf{F}$. The mean value of nine estimated f is taken as the final result. Moreover, we discard the equations in which the coefficient of f^2 is close to zero for computational stability.

After extract the intrinsic parameters, the essential matrix $\mathbf{E}$ can be obtained by applying essential matrix equation (6). Then the relative pose can be uniquely estimated same as in the 5-point algorithm [31]. We denote a projection matrix of an image as $\mathbf{P} = [\mathbf{R}, \mathbf{t}]$, where $\mathbf{R}$ is the 3×3 rotation matrix and $\mathbf{t}$ is the translation vector. For clarity, we denote $\mathbf{P}_n = [\mathbf{R}_n, \mathbf{t}_n]$ as the rotation matrix and translation vector of one neighbor relative to the world coordinate, and denote $\mathbf{P}_{rn} = [\mathbf{R}_{rn}, \mathbf{t}_{rn}]$ as the pose of the input image relative to its neighbor image. After applying our relative pose estimation algorithm, a set of $\mathbf{P}_{rn}$ can be obtained.

E. Final Location Determination

Based on the relative poses between the input image and its nearest neighbors, the final 6-DOF location information of the input image can be figured out by fusing all the candidate poses in this section. Here we first consider the rotation. Given the rotation matrix $\mathbf{R}_n$ of one neighbor relative to the world coordinate and the rotation matrix $\mathbf{R}_{rn}$ of the input image relative to the neighbor, we can get the rotation matrix $\mathbf{R}_r$ of the input image relative to the world coordinate as

$$\mathbf{R}_r = \mathbf{R}_{rn}\mathbf{R}_n. \tag{11}$$

For a given rotation matrix $\mathbf{R}_r$, it can be decomposed into multiplication of three basic rotation matrixes which are corresponding to Euler angles $\theta_z, \theta_x, \theta_y$ in turn. To obtain the final rotation matrix from several candidate poses, we can average all the candidate angles respectively to obtain final rotation angles. Then the final rotation matrix can be figured out via matrix multiplication of three basic rotation matrixes corresponding to $\theta_z, \theta_x, \theta_y$.

Now the last problem is that the relative pose estimation algorithm can only get the direction of $\mathbf{t}_i$ but no length. As a result, from a relative pose only a line can be obtained, where the input image should be on. Moreover, as two lines can determine a point, when there are more than one candidate relative poses, we can resort to triangulation theory and figure out the intersection point of these lines as the final position. If there are only one candidate relative pose unfortunately, which rarely occurs, we simply suppose that $\mathbf{t}_i$ has unit length. Given the pose of the input image relative to its one neighbor, although its 3D position cannot be obtained, the relative pose $[\mathbf{R}_{rn}, \mathbf{t}_{rn}]$ and the 3D position of one neighbor $\mathbf{l} = (x_l, y_l, z_l)^T$ determine a straight line $\mathbf{L}$, which can be denoted as

$$\frac{x - x_l}{x_d} = \frac{y - y_l}{y_d} = \frac{z - z_l}{z_d}. \tag{12}$$

The 3D position of the input image should be on this line. As $\mathbf{P}_n$ and $\mathbf{P}_{rn}$ are known, the position of the neighbor $\mathbf{l} = (x_l, y_l, z_l)^T$ can be obtained as

$$\mathbf{l} = \begin{bmatrix} x_l \\ y_l \\ z_l \end{bmatrix} = -\mathbf{R}_n^{-1}\mathbf{t}_n \tag{13}$$

and the direction of the line $\mathbf{d} = [x_d, y_d, z_d]^T$ can be obtained as

$$\mathbf{d} = \begin{bmatrix} x_d \\ y_d \\ z_d \end{bmatrix} = -\mathbf{R}_n^{-1}\mathbf{R}_{rn}^{-1}\mathbf{t}_{rn}. \tag{14}$$

One line $\mathbf{L}$ can only provide two different equations as

$$\begin{cases} z_d x - x_d z = z_d x_l - x_d z_l \\ z_d y - y_d z = z_d y_l - y_d z_l. \end{cases} \tag{15}$$

Therefore given $k\,(k \geqslant 2)$ relative poses, we can get $2k$ linear equations according to (15). These $2k$ linear equations can be solved by using least square algorithm to minimize the algebra error. Then the final position $\mathbf{x} = (x, y, z)$ of the input image can be obtained.

However, the algebra error has no geometry meaning and the solving method is likely influenced by the scale of coefficient. To this end, as shown in Fig. 5 we define the geometry error as the sum of square distance between the input image and each of the candidate line $\mathbf{L}_i$ as $G_l = \sum_{i=1}^{k} \boldsymbol{\mathcal{D}}^2(\mathbf{x}, \mathbf{L}_i)$, where

$$\begin{aligned} \boldsymbol{\mathcal{D}}^2(\mathbf{x}, \mathbf{L}_i) \;=\; & (x - x_{l_i})^2 + (y - y_{l_i})^2 + (z - z_{l_i})^2 \\ & - \frac{\left(x_{d_i}(x - x_{l_i}) + y_{d_i}(y - y_{l_i}) + z_{d_i}(z - z_{l_i})\right)^2}{x_{d_i}^2 + y_{d_i}^2 + z_{d_i}^2}. \end{aligned} \tag{16}$$

Moreover, as the direction of $\mathbf{L}_i$ is usually not accurate, we add a regularization term G_n which is the sum of square distance between the input image and each of its neighbors as $G_n = \sum_{i=1}^{k} \boldsymbol{\mathcal{D}}^2(\mathbf{x}, \mathbf{l}_i)$, where

$$\boldsymbol{\mathcal{D}}^2(\mathbf{x}, \mathbf{l}_i) = (x - x_{l_i})^2 + (y - y_{l_i})^2 + (z - z_{l_i})^2. \tag{17}$$

Finally, the final position $\mathbf{x}$ can be figured out by minimizing the geometry error $G = G_l + G_n$. As G is convex and quadratic, we can calculate its partial derivatives with respect to each variable

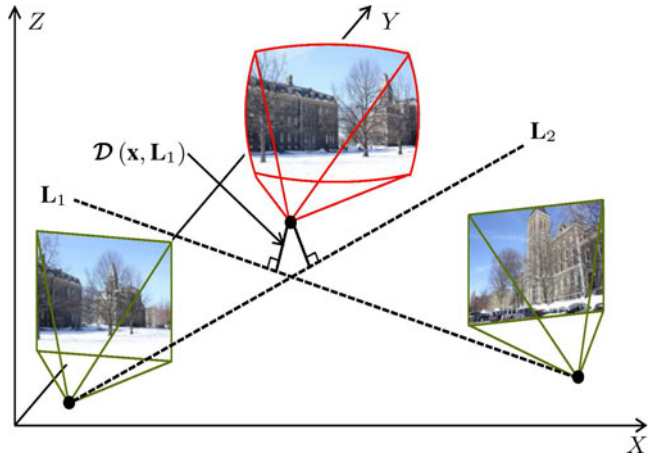


Fig. 5. To obtain the final position of the input image, we define a geometry error as the sum of distance quare between input image and each of the candidate line. Via minimizing it, the final position can be figured out.

of $\mathbf{x}$ respectively. $\frac{\partial G}{\partial x}$ is taken as an example and the derivation is

$$\frac{\partial G}{\partial x} = \sum_{i=1}^{k}(4\,(x - x_{l_i}) + \frac{2x_{d_i}\,(x_{d_i}\,(x - x_{l_i}) + y_{d_i}\,(y - y_{l_i}) + z_{d_i}\,(z - z_{l_i}))}{x_{d_i}^2 + y_{d_i}^2 + z_{d_i}^2}). \tag{18}$$

By setting the partial derivatives as zero, i.e. $\frac{\partial G}{\partial x} = 0$, $\frac{\partial G}{\partial y} = 0$, $\frac{\partial G}{\partial z} = 0$, a system of homogeneous linear equations is obtained. After solve this equation system, the final position $\mathbf{x}$ of the input image can be got.

Moreover, as there are some unstable candidate poses, we apply a maximum likelihood criterion and select two ($k = 2$) most stable candidate poses to calculate the final location of the input image. Actually the number of inlier point matches is used to evaluate each candidate pose. The more inlier point matches, the more robust the candidate pose.

IV. Experiments

In order to verify the feasibility and effectiveness of our proposed method, we test our image localization method on two public datasets, including Cornell Arts Quad dataset [10] and Dubrovnik dataset [37]. We also quantitatively compare our calibrated-uncalibrated relative pose estimation algorithm with previous works on synthetic and real data.

A. Image Localization

Cornell Arts Quad dataset and Dubrovnik dataset originally are created for 3D reconstruction and are also used by 3D point cloud model based localization algorithms. The dataset contains not only the position and rotation information of all images but also 3D structure point cloud. While in our experiments, we only use the location information of images.

TABLE I
Localization Accuracy (in Meters) of Li *et al.* [15], Our Method, Our Method Without Radial Distortion, Without Regularization Term, Without Maximum Likelihood Criterion, and Without Geometry Error

	localization accuracy (in meters)				
	1st quartile	median	3rd quartile	Mean	# localized images
Li *et al.* [15]	0.50	1.67	5.00	4.73	254
Our method	**0.46**	**1.59**	**4.54**	**4.08**	305
Our method w/o distortion	0.52	1.82	5.07	5.00	310
Our method w/o maximum	1.03	2.99	6.23	5.67	305
Our method w/o geometry	0.80	3.25	11.11	11.34	305
Our method w/o G_n	0.74	2.77	9.15	9.11	305

The Quad dataset contains 6514 images in total, including 348 query images and 6166 reference images. In the retrieval step, a visual vocabulary tree is first created via clustering SURF features which are extracted from reference images. As duplicated image areas are common in this dataset, most features should appear more than twice in all reference images. Therefore, about half of the reference images, i.e. 3000 images, are randomly selected for computational efficiency. The trained vocabulary tree contains 30 000 visual words. We further figure out the visual document for each reference image and query image. Via measuring the similarity between the visual documents, the top 20 nearest neighbor images are retrieved from reference images for each query image.

We compare our method with a representative 2D-to-3D based method [15]. As shown in Table I, our method not only successfully localizes more images, i.e. 305 *versus* 254, but also achieves more accurate localization results, i.e. 4.08 meters *versus* 4.73 meters in average. Fig. 6 visualizes our localization results on Quad dataset. Note that, the 3D point cloud is only used for visualization and not used in localization process.

We further perform an experiment to exploit the effectiveness of each module in our method. The Table I shows our localization results without radial distortion parameter λ (Our method w/o distortion), without maximum likelihood criterion(using all candidate poses), and without geometry error G (using algebra error), without regularization term G_n, respectively. The localization results show that, each module of our method is effective to improve the localization performance.

The ratio test algorithm is used to find the initial matched points between the input image and each of its neighbors. In order to exploit the influence of the ratio threshold parameter T_h, it ranges from 0.2 to 0.8 by step 0.1. Table II shows the localization accuracy when we use different T_h, with the T_h decreasing, the more accurate localized results, the less successfully localized images. As presented in Table II and Fig. 7, the value of T_h is a trade-off between localization accuracy and the proportion of successfully localized images. Without loss of generality, $T_h = 0.5$ in our experiments. Our method can localize most query images with high accuracy and about 89%

Fig. 6. We visualize our localization results on the Quad dataset. Note that the point cloud is only used for visualization.

TABLE II
LOCALIZATION ACCURACY ON QUAD DATASET (IN METERS) WHEN SET T_h AS DIFFERENT VALUE

	localization accuracy (in meters)				
T_h	1st quartile	median	3rd quartile	Mean	# localized images
0.2	**0.17**	**0.84**	**1.20**	**1.99**	249
0.3	0.37	1.06	3.29	3.24	289
0.4	0.45	1.47	4.23	3.83	302
0.5	0.46	1.59	4.54	4.08	305
0.6	0.41	1.58	4.79	4.26	306
0.7	0.38	1.48	4.79	4.50	310
0.8	0.53	1.77	4.89	5.29	311

TABLE III
LOCALIZATION ACCURACY ON QUAD DATASET (IN METERS) WHEN SET k AS DIFFERENT VALUE

	localization accuracy (in meters)				
k	1st quartile	median	3rd quartile	Mean	# localized images
2	**0.46**	**1.59**	**4.54**	**4.08**	305
3	0.88	2.45	5.43	4.79	305
4	0.88	2.28	5.38	4.75	305
5	0.94	2.35	5.36	4.93	305
6	0.97	2.42	5.50	4.99	305
20	1.03	2.99	6.23	5.67	305

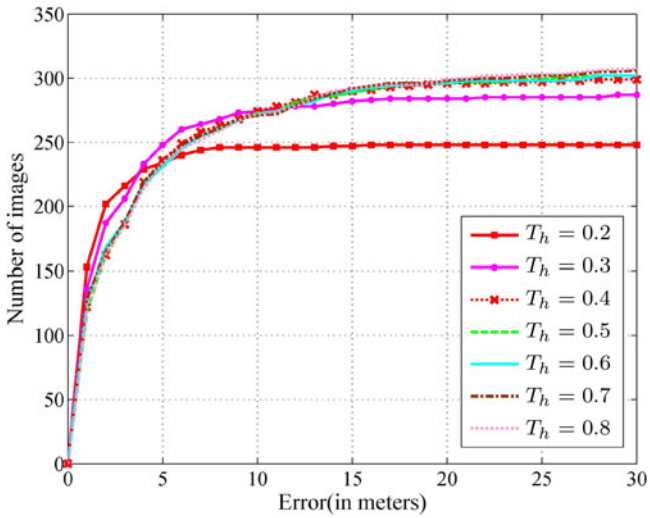


Fig. 7. Number of successful localized images along with error increasing on Quad dataset using different ratio value.

images can be localized in ten meters. The localization error has a median of 1.59 m, 1st quartile of 0.46 m and 3rd quartile of 4.08 m, which is more accurate compared with civilian GPS.

To determine the final location, we apply the maximum likelihood criterion and choose the most robust two candidate poses, instead of using all the candidate poses. In this experiment, its effectiveness is verified. We denote the maximum number of candidate poses as $k \in [2, 20]$. Note that, not every input image can find k candidate poses as some neighbors are removed due to the epipolar geometry constraint. As shown in Table III, the localization error increases along with the increasing k due to the additional coarse candidate poses. We set $k = 2$ in our experiments so as to achieve the most accurate localization results.

To verify the generalization of our method, we also perform the localization task on Dubrovnik dataset. There are 6044 reference images and 800 query images in this dataset. In the retrieval step, 3000 images are selected from all reference images to create a visual vocabulary tree with 30 000 visual words, which is the same as on Quad dataset. We also retrieve the top 20 nearest neighbors for each input image. As stated in [37], the ground truth of this dataset are noisy. Previous works, e.g. [15], usually only report the successful localization proportion. In our method, T_h controls the trade-off between localization accuracy and the number of successfully localized images as illustrated before. We set the ratio threshold $T_h = 0.5, 0.6, 0.7, 0.8$ respectively, and can successfully localized 75.5%, 88%, 96%, 100% images.

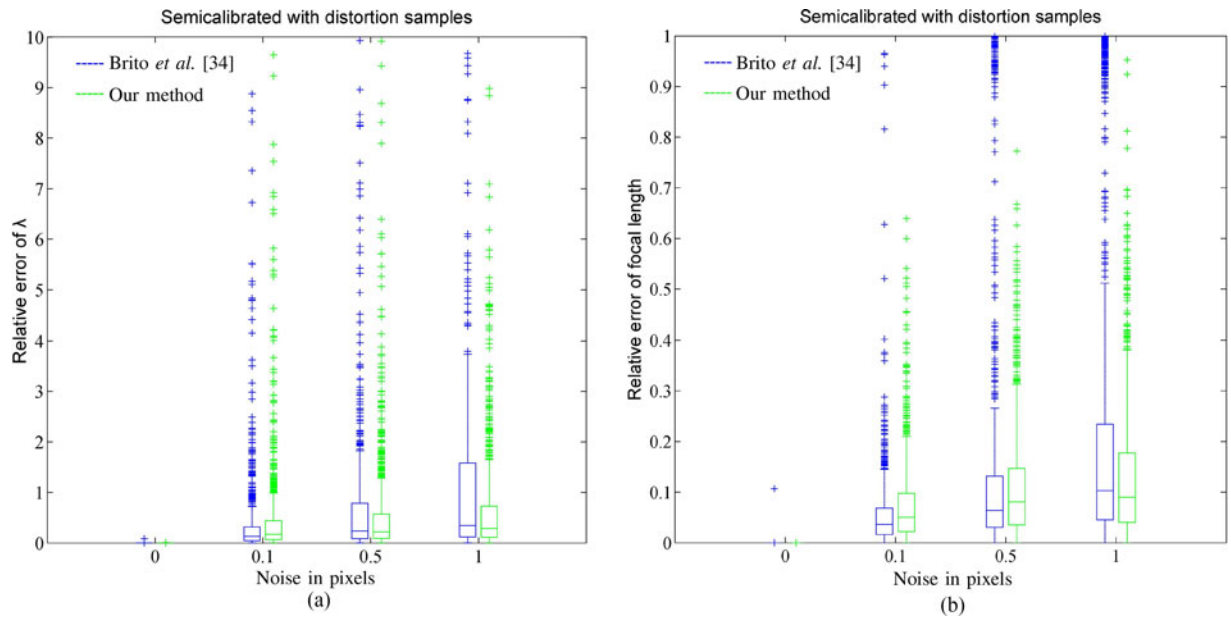


Fig. 8. Results of distortion parameter λ and focal length on synthetic data. (a) Relative error of distortion parameter λ compared with Brito *et al.* [34]. (b) Relative error of focal length compared with Brito *et al.* [34]. The horizontal axis represents the standard deviation of Gaussian noise added on points and the vertical axis represents relative error. The results show that, with the increasing noise, our method can achieve more accurate results.

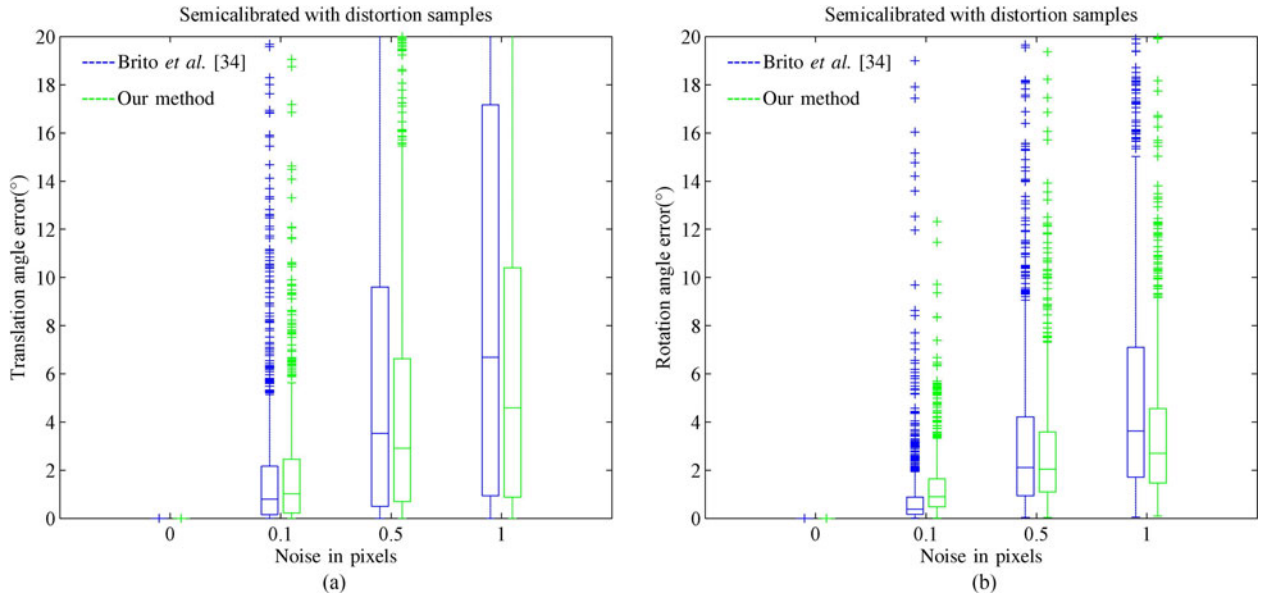


Fig. 9. Results of translation angle and rotation angle on synthetic data. (a) Translation angle error compared with Brito *et al.* [34]. (b) Rotation angle error compared with Brito *et al.* [34]. The horizontal axis represents the standard deviation of Gaussian noise added on points and the vertical axis represents angle error. The results show that, with the increasing noise, our method can achieve more accurate results.

To evaluate the efficiency of our method, we compare the running time of our method and Li *et al.* [15], including off-line pre-processing time and on-line localization time. In total, our method needs about 139 CPU hours, while Li *et al.* [15] need 277 CPU hours on Quad dataset. Specially, for on-line localization, our method averagely needs 30 seconds per image, while Li *et al.* [15] need a few seconds. Our method takes less time in total as the 3D reconstruction is time consuming for 2D-to-3D based methods. At the same time, our method takes more time than [15] in on-line localization. This is resulted from so many times of relative pose estimation, which can be parallelly accelerated in the future.

Overall, we perform 6-DOF localization task for input image via estimating and fusing its candidate poses relative to each of

its neighbors. The localization accuracy outperforms 2D-to-3D matching based method. Each module of our method also has been verified effective.

B. Relative Pose Estimation

Relative pose estimation is one of key steps in our method. In our configuration, the input image is uncalibrated and has two intrinsic parameters including focal length f and one order radial distortion parameter λ. Under this configuration, Brito *et al.* [34] estimate the fundamental matrix along with distortion parameter by solving a high-order polynomial system, and estimate the focal length by minimizing an algebraic error. For efficiency, we propose a 11-point algorithm based on SVD to estimate the fundamental matrix along with distortion parameter, and a closed form solver to calculate the focal length. We compare our algorithm with [34] on synthetic and real image data, respectively.

The first experiment is performed on synthetic data, which contain a calibrated camera, a set of random 3D points, and 1000 randomly generated uncalibrated cameras. Following the settings of Brito *et al.* [34], the calibrated camera is placed at the origin point and looks at the direction of z-axis. Its focal length is set as 1500. We randomly generate 1000 3D points $\{(x_i, y_i, z_i)|1 \leq i \leq 1000\}$, where $x_i \in (-100, 100), y_i \in (-100, 100), z_i \in (250, 350)$. Then the parameters of the uncalibrated cameras are also randomly generated. The camera location (x_c, y_c, z_c) is near the origin point, where $x_c \in (-100, 100), y_c \in (-100, 100), z_c \in (-50, 50)$. As an arbitrary rotation matrix can be divided into multiple multiplication of three basic rotation matrixes. Each basic rotation matrix is corresponding to a Euler angle. A rotation matrix is synthesized via randomly generating three Euler angles. The focal length is varying between $1/2$ and $2X$ of the focal length of the calibrated camera. The distortion parameter λ is between 5×10^{-7} and 0. The 3D points can be projected to form 2D virtual image points according to the parameters of each camera. Each image plane contains 1024×1024 pixels. Moreover, each camera should observe more than 500 3D points. If two 2D points in different images are from the same 3D points, they are taken as a pair of matched points. To simulate the noise in imaging process, we add some Gaussian noise on the coordinates of each 2D points.

The experimental results of distortion parameter λ and focal length f on synthetic data are shown in Fig. 8, where the horizontal axis represents the standard deviation of Gaussian noise added on 2D points and the vertical axis represents relative error of distortion parameter λ and focal length f. The relative error is calculated by dividing ground truth by absolute predicted error. Fig. 8(a) shows the relative error of distortion parameter and Fig. 8(b) shows the relative error of focal length compared with [34]. The results of translation angle and rotation angle on synthetic data are shown in Fig. 9, where the horizontal axis also represents the standard deviation of Gaussian noise added on 2D points and the vertical axis represents angle error of translation $\mathbf{t}$ and rotation matrix $\mathbf{R}$. The angle error is calculated by figuring out the angle between the estimated result and the ground truth. Fig. 9(a) shows the translation angle error and Fig. 9(b) shows the rotation angle error compared with Brito *et al.* [34]. Table IV shows the mean epipolar errors of inliers along with mean inlier ratios of our algorithm and [34] on synthetic data. The mean inlier ratio is the proportion of inliers in all matched points. Overall, the experiments show that the accuracy of our results are comparable with [34]. With the increasing noise, our method can achieve more accurate results, which means that our method is more robust than [34] when the coordinates of 2D points are noisy.

TABLE IV
LEFT VALUES INDICATE THE MEAN INLIER EPIPOLAR ERROR (IN PIXELS ON UNDISTORTED IMAGE PLANE) AND RIGHT VALUES INDICATE THE MEAN INLIER RATIOS OF OUR ALGORITHM AND [34] ON SYNTHETIC DATA

Standard deviation	0.0	0.1	0.5	1.0
Our method	0.00/100%	1.20/99.4%	2.03/93.0%	2.80/80.8%
Brito *et al.* [34]	0.00/100%	0.94/96.7%	2.31/84.3%	3.03/73.0%

TABLE V
OUR METHOD NEEDS LESS THAN HALF TIME OF [34] TO ESTIMATE OSR FUNDAMENTAL MATRIX AND RELATIVE POSE (IN SECONDS)

	Our method	Brito *et al.* [34]
Estimate OSR fundamental matrix	**3.846**	9.768
Intrinsic parameters extraction and estimate relative pose	**0.415**	0.435
Total	**4.261**	10.203

Another problem is that as the configurations are randomly generated, how confident the results are. A more systematic approach is to uniformly sample all the feasible configurations. However, as one configuration is uniquely determined by 8 parameters (i.e. distortion parameter λ, focal length and 6-DOF relative pose), there is a really large number of potential configurations (e.g., 10^8 probable configurations if we quantize each parameter to 10 bins). As a result, the completely systematic approach is computationally infeasible. Inspired by the idea of the cross validation, we repeat the experiment ten times. The mean performance of our algorithm and its standard deviation are 0.722 ± 0.034 for relative error of distortion parameter λ, 0.063 ± 0.002 for relative error of focal length, 5.248 ± 0.242 for translation angle error, 3.016 ± 0.101 for rotation angle error. The mean performance of [34] and its standard deviation are 0.933 ± 0.042, 0.067 ± 0.002, 5.819 ± 0.273, 3.199 ± 0.117, respectively. The experimental results show that the performance stays stably in different trails using randomly generated configurations.

We also test our algorithm on 4 sets of real images which are used in [34]. The mean inlier epipolar error (in pixels on undistorted image plane) and mean inlier ratios of our results are $1.91/85.2\%$, $1.76/74.5\%$, $1.77/81.8\%$ and $1.57/90.7\%$ respectively on these 4 datasets, while the results of Brito *et al.* [34] are $1.96/84.5\%$, $1.78/77.4\%$, $1.62/83.8\%$ and $1.62/90.1\%$. Our mean inlier ratio is higher than Brito *et al.* [34] on dataset 1 and dataset 4, while Brito *et al.* [34] higher on dataset 2 and dataset 3.

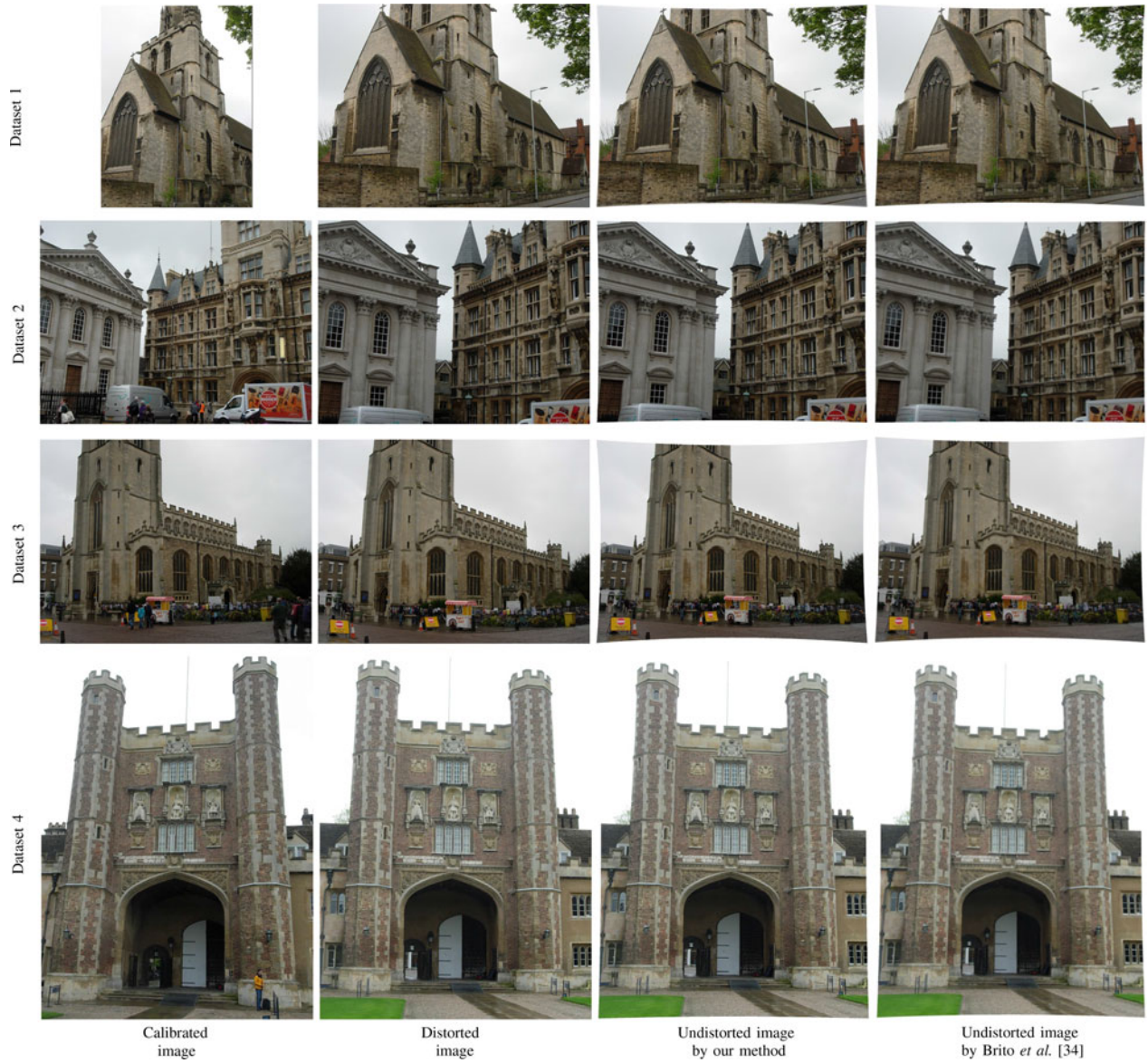


Fig. 10. Some undistorted results using the distortion parameter λ estimated by our method and Brito *et al.* [34].

Though the accuracy of our algorithm is not better than [34], our algorithm is faster. As shown in Table V, our algorithm needs less than half the time to estimate OSR fundamental matrix and relative pose averagely, i.e. 4.261 *versus* 10.20 seconds, excluding feature extracting and matching time. There are also some undistorted results using the λ estimated by our algorithm and Brito *et al.* [34] respectively in Fig. 10, which have little difference. Note that, we embed the fundamental matrix estimation algorithm in an RANSAC process with 200 iterations. All these experiments are done using Matlab 2013a on the same PC with I7-3770 CPU and 4 GB RAM. The reason for our algorithm's faster is that, our algorithm mainly applies SVD algorithm which can be performed fast, while Brito *et al.* [34] have to solve a higher-order polynomial system which needs more time.

Comparing our method with Brito *et al.* [34], all the formulation and computation differences are caused by using different number of point matches. Our method uses 11 point matches, while [34] uses 9 point matches. If there are only random noises on the coordinates of 2D point, theoretically, it is more robust to use more point matches. This is exactly why to usually refine the final result using all inlier point matches. This is also verified by the experiments, our results are better than [34] with the increasing noise on synthetic data. At the same time, there are also false positive point matches between real images. As a result, it is more difficult to select a set of good matches if using more. This maybe the reason why our method obtains lower inlier ratio than [34] on some real image datasets, i.e. real image dataset 2 and 3.

1) Failure Case: Lastly, we also discuss some failure cases. If there are few discriminative features in the input image, the image retrieval algorithm may fail to retrieve its real nearest neighbors. As a result, our method may lack the ability to localize such images. As shown in Fig. 11, there are two failure cases on Quad dataset. In the first case, the building is occluded

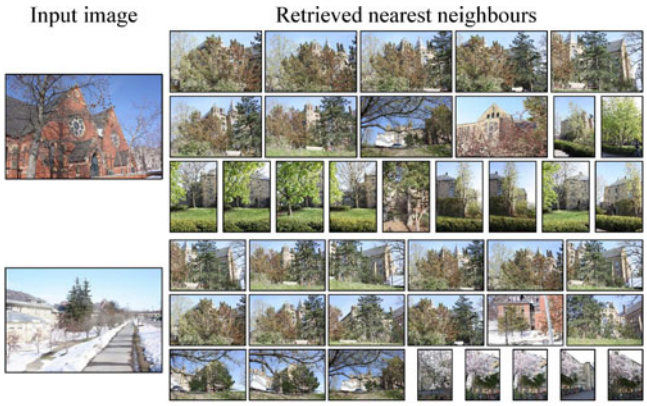


Fig. 11. Two failure cases on Quad dataset. If the input image contains few discriminative features, e.g., the building is occluded by a tree in top case, most areas are trees, road, or snow in bottow case, the retrieval results are all false positives. As a result, our method may fail to localize these images.

by a tree. In the second case, most areas of the image are undiscriminating, such as trees, road, or snow.

V. Conclusion

In this paper, we propose a flexible method to localize an input image by fusing its poses relative to its neighbors. The neighbors are retrieved from a reference image dataset. An efficient algorithm is further proposed to estimate the relative pose between an uncalibrated input image and a calibrated neighbor image. As a result, several candidate poses of the input image are obtained. To figure out the final location, we further define and minimize a geometry error to fuse these candidate poses. Each module of our method is verified effective in experiments. Our method can obtain satisfactory localization accuracy. Comparing with 2D-to-3D based methods, our method is more flexible to exploit the increasing geo-tagged data. At the same time, our method need more time in on-line localization process. This is resulted from so many times of relative pose estimation. As this process can be paralleled, we can accelerate our method using multiple CPU cores in the future.

References

[1] Y.-Y. Chen, A.-J. Cheng, and W. Hsu, "Travel recommendation by mining people attributes and travel group types from community-contributed photos," *IEEE Trans. Multimedia*, vol. 15, no. 6, pp. 1283–1295, Oct. 2013.

[2] N. Snavely, S. M. Seitz, and R. Szeliski, "Photo tourism: Exploring photo collections in 3D," *ACM Trans. Graph.*, vol. 25, no. 3, pp. 835–846, Jul. 2006.

[3] J. Zhang, A. Hallquist, E. Liang, and A. Zakhor, "Location-based image retrieval for urban environments," in *Proc. IEEE 18th Int. Conf. Image Process.*, Sep. 2011, pp. 3677–3680.

[4] I.-H. Jhuo, T. Chen, and D. Lee, "Scene location guide by image-based retrieval," *Advances in Multimedia Modeling*, vol. 5916, pp. 196–206, 2010.

[5] D. Wagner, G. Reitmayr, A. Mulloni, T. Drummond, and D. Schmalstieg, "Real-time detection and tracking for augmented reality on mobile phones," *IEEE Trans. Vis. Comput. Graph.*, vol. 16, no. 3, pp. 355–368, May/Jun. 2010.

[6] G. Takacs *et al.*, "Outdoors augmented reality on mobile phone using loxel-based visual feature organization," in *Proc. 1st ACM Int. Conf. Multimedia Inform. Retrieval*, 2008, pp. 427–434.

[7] H. Lim, S. Sinha, M. Cohen, and M. Uyttendaele, "Real-time image-based 6-DOF localization in large-scale environments," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2012, pp. 1043–1050.

[8] L. Meier, P. Tanskanen, F. Fraundorfer, and M. Pollefeys, "PIXHAWK: A system for autonomous flight using onboard computer vision," in *Proc. IEEE Int. Conf. Robot. Autom.*, May. 2011, pp. 2992–2997.

[9] J.-M. Frahm *et al.*, " Building Rome on a cloudless day," in *Proc. 11th Eur. Conf. Comput. Vis.*, 2010, vol. 6314, pp. 368–381.

[10] D. Crandall, A. Owens, N. Snavely, and D. Huttenlocher, "SfM with MRFs: Discrete-continuous optimization for large-scale reconstruction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 12, pp. 2841–2853, Dec. 2013.

[11] A. Irschara, C. Zach, J.-M. Frahm, and H. Bischof, "From structure-from-motion point clouds to fast location recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2009, pp. 2599–2606.

[12] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. 2nd ed. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[13] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, Jun. 1981.

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

[15] Y. Li, N. Snavely, D. Huttenlocher, and P. Fua, "Worldwide pose estimation using 3D point clouds," in *Proc. 12th Eur. Conf. Comput. Vis.*, 2012, vol. 7572, pp. 15–29.

[16] S. Middelberg, T. Sattler, O. Untzelmann, and L. Kobbelt, " Scalable 6-DOF localization on mobile devices," in *Proc. 13th Eur. Conf. Comput. Vis.*, 2014, vol. 8690, pp. 268–283.

[17] M. Donoser and D. Schmalstieg, "Discriminative feature-to-point matching in image-based localization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2014, pp. 516–523.

[18] Y. Li, D. Crandall, and D. Huttenlocher, "Landmark classification in large-scale image collections," in *Proc. IEEE Int. Conf. Comput. Vis.*, Sep./Oct. 2009, pp. 1957–1964.

[19] Q. Hao *et al.*, "3D visual phrases for landmark recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2012, pp. 3594–3601.

[20] A. Bergamo, S. Sinha, and L. Torresani, "Leveraging structure from motion to learn discriminative codebooks for scalable landmark classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2013, pp. 763–770.

[21] L. Zhu, J. Shen, H. Jin, L. Xie, and R. Zheng, "Landmark classification with hierarchical multi-modal exemplar feature," *IEEE Trans. Multimedia*, vol. 17, no. 7, pp. 981–993, Jul. 2015.

[22] W. Zhang and J. Kosecka, "Image based localization in urban environments," in *Proc. 3rd Int. Symp. 3D Data Process., Vis., Transmiss.*, Jun. 2006, pp. 33–40.

[23] D. Chen *et al.*, "City-scale landmark identification on mobile devices," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2011, pp. 737–744.

[24] S. Cao and N. Snavely, "Graph-based discriminative learning for location recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2013, pp. 700–707.

[25] A. Zamir and M. Shah, "Image geo-localization based on multiplenearest neighbor feature matching using generalized graphs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 8, pp. 1546–1558, Aug. 2014.

[26] Y. Jing, M. Covell, D. Tsai, and J. Rehg, "Learning query-specific distance functions for large-scale web image search," *IEEE Trans. Multimedia*, vol. 15, no. 8, pp. 2022–2034, Dec. 2013.

[27] M. Lux and S. A. Chatzichristofis, "Lire: Lucene image retrieval: An extensible Java CBIR library," in *Proc. ACM Int. Conf. Multimedia*, 2008, pp. 1085–1088.

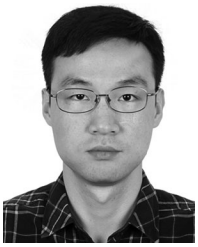
[28] C. Liu, J. Yuen, and A. Torralba, "Nonparametric scene parsing via label transfer," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 12, pp. 2368–2382, Dec. 2011.

[29] Q. Li *et al.*, "Geodesic propagation for semantic labeling," *IEEE Trans. Image Process.*, vol. 23, no. 11, pp. 4812–4825, Nov. 2014.

[30] G. Baatz, K. Köser, D. Chen, R. Grzeszczuk, and M. Pollefeys, " Leveraging 3D city models for rotation invariant place-of-interest recognition," *Int. J. Comput. Vis.*, vol. 96, no. 3, pp. 315–334, 2012.

[31] D. Nistér, "An efficient solution to the five-point relative pose problem," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 6, pp. 756–770, Jun. 2004.

[32] H. Stewénius, D. Nistér, F. Kahl, and F. Schaffalitzky, "A minimal solution for relative pose with unknown focal length," *Image Vis. Comput.*, vol. 26, no. 7, pp. 871–877, 2008.
[33] M. Bujnak, Z. Kukelova, and T. Pajdla, "3D reconstruction from image collections with a single known focal length," in *Proc. 12th IEEE Conf. Comput. Vis. Pattern Recog.*, Sep./Oct. 2009, pp. 1803–1810.
[34] J. H. Brito, C. Zach, K. Koeser, M. Ferreira, and M. Pollefeys, "One-sided radial-fundamental matrix estimation," in *Proc. Brit. Mach. Vis. Conference.*, 2012, pp. 96.1–96.12.
[35] H. Bay, T. Tuytelaars, and L. V. Gool, "SURF: Speeded up robust features," in *Proc. Eur. Conf. Comput. Vis.*, 2006, vol. 3951, pp. 404–417.
[36] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.
[37] Y. Li, N. Snavely, and D. Huttenlocher, "Location recognition using prioritized feature matching," in *Proc. Eur. Conf. Comput. Vis.*, 2010, vol. 6312, pp. 791–804.
[38] Y. Song, X. Chen, X. Wang, Y. Zhang, J. Li, "Fast Estimation of Relative Poses for 6-DOF Image Localization," *IEEE Int. Conf. Multimedia Big Data*, pp. 156–163, Apr. 2015.

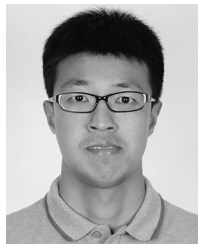


Yafei Song is currently working toward the Ph.D. degree at the State Key Laboratory of Virtual Reality Technology and System, School of Computer Science and Engineering, Beihang University, Beijing, China.

His research interests include computer vision and augmented reality.

Xiaowu Chen (M'09-SM'15) received the Ph.D. degree from Beihang University, Beijing, China, in 2001.

He is currently a Professor with the State Key Laboratory of Virtual Reality Technology and Systems as well as the School of Computer Science and Engineering, Beihang University. His research interests include computer vision, computer graphics, virtual reality, and augmented reality.



Xiaogang Wang is currently working toward the Ph.D. degree at the State Key Laboratory of Virtual Reality Technology and System, School of Computer Science and Engineering, Beihang University, Beijing, China.

His research interest include computer graphics.

Yu Zhang is currently working toward the Ph.D. degree at the State Key Laboratory of Virtual Reality Technology and System, School of Computer Science and Engineering, Beihang University, Beijing, China.

His research interests include computer vision and image processing.

Jia Li (M'12-SM'15) received the B.E. degree from Tsinghua University, Beijing, China, in 2005, and the Ph.D. degree from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, in 2011.

He is currently an Associate Professor with the State Key Laboratory of Virtual Reality Technology and Systems, School of Computer Science and Engineering, Beihang University, Beijing, China. His research interests include computer vision and image/video processing.