International Journal of Geo-Information

MDPI

Article

Drift-Aware Monocular Localization Based on a Pre-Constructed Dense 3D Map in Indoor Environments

Guanyuan Feng [1], Lin Ma [1,*], Xuezhi Tan [1] and Danyang Qin [2]

[1] School of Electronics and Information Engineering, Harbin Institute of Technology, Harbin 150080, China; fengguanyuan@126.com (G.F.); tanxz1957@hit.edu.cn (X.T.)

[2] Electronic Engineering College, Heilongjiang University, Harbin 150080, China; qindanyang@hlju.edu.cn

* Correspondence: malin@hit.edu.cn; Tel.: +86-451-8641-3513 (ext. 8202)

Received: 18 June 2018; Accepted: 19 July 2018; Published: 25 July 2018

Abstract: Recently, monocular localization has attracted increased attention due to its application to indoor navigation and augmented reality. In this paper, a drift-aware monocular localization system that performs global and local localization is presented based on a pre-constructed dense three-dimensional (3D) map. In global localization, a pixel-distance weighted least squares algorithm is investigated for calculating the absolute scale for the epipolar constraint. To reduce the accumulative errors that are caused by the relative position estimation, a map interaction-based drift detection method is introduced in local localization, and the drift distance is computed by the proposed line model-based maximum likelihood estimation sample consensus (MLESAC) algorithm. The line model contains a fitted line segment and some visual feature points, which are used to seek inliers of the estimated feature points for drift detection. Taking advantage of the drift detection method, the monocular localization system switches between the global and local localization modes, which effectively keeps the position errors within an expected range. The performance of the proposed monocular localization system is evaluated on typical indoor scenes, and experimental results show that compared with the existing localization methods, the accuracy improvement rates of the absolute position estimation and the relative position estimation are at least 30.09% and 65.59%, respectively.

Keywords: monocular localization; pre-constructed dense 3D map; absolute scale estimation; drift detection

1. Introduction

The emergence of wireless communication and the global positioning system (GPS) has ignited the idea of personal navigation systems (PNSs). PNSs have localization and navigation functions that provide users with positional information via a mobile terminal, such as a smartphone or a tablet personal computer (PC). Owing to the global navigation satellite system (GNSS) [1,2], position information is easy to obtain in outdoor environments, but in indoor environments, because of the shielding effect of structures, GNSS is incapable of providing reliable position services to a user. Therefore, the development of an accurate and stable indoor localization method that does not depend on satellite signals has become a research hotspot in recent years.

A variety of techniques are being investigated for indoor localization, and a typical category of these is radio signal-based methods, including WiFi-based [3,4], radio frequency identification (RFID)-based [5,6], and ultra wideband (UWB)-based methods [7,8]. However, an obvious drawback of signal-based localization systems is that moving objects or pedestrians affect signal strength or flight time, which makes localization results unreliable and even leads to localization failure. Moreover, some anchor points need to be installed and calibrated before localization is established,

in radio signal-based systems. Once the system is enabled, the anchor points cannot be moved. The installation and maintenance of the infrastructure at the service of these localization systems are costly. In consideration of these drawbacks, a stable and cost-efficient indoor localization method is desired for providing users with quality location-based services.

With the popularization of smart mobile terminals, cameras as vision sensors become standard equipment embedded into almost all smartphones and tablet PCs, which provides an opportunity to develop image-based localization, i.e., visual localization. For a visual localization system, the user merely needs to capture an image or an image sequence by a mobile terminal, and uploads the images to the server via a wireless network. Then, the positions of the user can be estimated by localization algorithms at the server. Finally, positional information is sent to the user's mobile terminal. The process of visual localization is realized without any infrastructure deployment. Before the implementation of visual localization, an off-line database that contains pose-tagged database images should be created, which has been studied in existing research [9,10].

Visual localization can be divided into two categories [11]: global localization and local localization. Typically, a visual localization system first estimates initial camera positions with global localization methods. Then, local localization methods are used to track the camera iteratively, namely, to perform subsequent localization.

In a monocular localization system, the key step of global localization is to compute the rotation matrix and translation vector between the query camera and the database camera. However, the absolute scale of the translation vector cannot be calculated based only on a query image and a database image, which is called the scale ambiguity problem [12]. There are many existing studies on solving this problem, which use for example, planar motion constraints and camera heights [13–15]. By taking advantage of database images, the absolute scale can be estimated by epipolar geometry or solving the Perspective-n-Points (PnP) problem. The epipolar-based method [16,17] can avoid computing the absolute scale by introducing multiple epipolar constraints between the query and database cameras. A prerequisite of this method is that, for a query image, more than one matched database images must be found to obtain multiple epipolar constraints. An alternative approach is based on solving the PnP problem that implicitly estimates the absolute scale by triangulation of the points in the real world [18]. However, this approach suffer from the errors that are caused by triangulation.

The development and popularization of RGB-D sensors makes it possible to conveniently measure the 3D positions of visual feature points. Then, the PnP-based method can be directly used for localization without triangulation [19,20]. In addition, a closed-form solution [21] for the scale ambiguity is studied based on the least squares approach, in which four available methods for calculating the absolute scale are involved. According to the moving direction of the query camera, one of the available methods should be chosen, which is a promising idea for absolute scale estimation. However, two deficiencies exist in this closed-form solution: the first is that there is no mention of how to obtain the 3D position of visual feature points, and the second is that the recognition method for the camera moving direction is not specified in [21].

For a monocular localization system, local localization is executed based on initial localization. Once the first two positions of the query camera have been acquired, subsequent query camera positions can be estimated by performing triangulation and solving the PnP problem. The EPnP [22] as a non-iterative solution to the PnP problem is commonly employed in local localization. However, the incremental frame-to-frame position estimation inherently accumulates errors, i.e., localization drifts. Because the user trajectory in monocular localization usually does not form a closed loop in most cases, the global optimization techniques, such as g2o (general graph optimization) [23] and global bundle adjustment [24], cannot be applied in drift correction. Thus, local optimization methods such as local bundle adjustment, are used to refine camera poses and 3D points that incorporate the information extracted from multiple query images [25,26]. However, the accumulative errors caused by local localization cannot be effectively detected and reduced by the existing approaches. With the increase of the trajectory length, the drifts caused by

position iteration significantly increase, which will ultimately lead to the result that accumulative errors exceed the maximum permissible value.

In consideration of the shortcomings of the existing approaches, an improved monocular localization system is proposed, which focuses on the absolute scale estimation and drift detection. The proposed system contains two modes: the global localization (i.e., the absolute position estimation) mode, and the local localization mode that contains the relative position estimation and drift detection. The main contributions of this paper are stated as follows:

(1) A pixel-distance weighted least squares (PWLS) algorithm is investigated to calculate the absolute scale of the translation vector between query and database cameras in global localization. The proposed PWLS algorithm fully considers the effect of the camera's direction of movement in scale estimation, and employs pixel-distance weights to control the contributions to the results of the absolute scale estimation in different moving directions.

(2) To effectively detect and reduce the drifts, a map interaction-based drift detection method is introduced in local localization. For the introduced drift detection method, the estimated feature points obtained by triangulation for relative position estimation are also used for drift detection, but unreliable estimated feature points are removed. With the aid of the linear characteristics of visual feature points that are stored in the pre-constructed dense 3D map, some line models are established. Taking advantage of line models and the estimator MLESAC [27], a line model-based MLESAC (LM-MLESAC) algorithm is proposed for the drift estimation. Because some estimated feature points contain significant errors caused by triangulation, the LM-MLESAC algorithm is executed to reject these estimated feature points to improve the drift estimation performance.

(3) By combining the global and local localization methods, a novel integrated monocular localization system is realized with the drift detection method. With the aid of the drift detection method, the average errors of the system are limited within an expected range.

The remainder of this paper is organized as follows: Section 2 discusses the global localization (i.e., the absolute position estimation). Section 3 studies the map interaction-based drift detection for local localization, including the relative position estimation, the line model establishment, and the line model-based MLESAC algorithm. In Section 4, the performances of the global and local localization methods are investigated. Finally, conclusions are presented in Section 5.

2. Absolute Position Estimation Based on the Dense 3D Map

The framework of the proposed monocular localization system is shown in Figure 1. The proposed system contains two modes: global localization and local localization. Local localization consisting of the relative position estimation and drift detection will be discussed in Section 3. The absolute position estimation, i.e., global localization, aims at acquiring query camera positions in the indoor coordinate system on the basis of pose-tagged database images which are stored in the pre-constructed 3D map. Global localization can be divided into two parts: scale-free localization using the epipolar constraint, and absolute scale estimation based on the proposed PWLS algorithm. The global localization method is utilized for both the initial position estimation and position correction, once the drift distance exceeds the threshold in local localization.

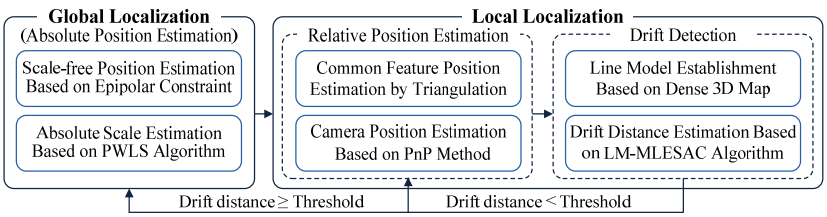


Figure 1. Framework of proposed monocular localization system.

2.1. Scale-Free Localization Using the Epipolar Constraint

Before presenting the monocular localization approach, it is necessary to review the dense 3D map, which is called the visual map in our previous works [10]. The visual map constructed by RGB-D (RGB-depth) sensors, e.g., the Microsoft Kinect, is an extended dense 3D map that contains essential elements for vision-based localization, such as 3D point clouds, database images, and the corresponding poses of the database camera. In this paper, the camera on the RGB-D sensor that is used to construct the visual map is called the database camera, and the camera on the user's smart terminal is called the query camera. In consideration of the computational complexity, only a subset of the images captured by the query camera are used for localization which are called query images. The keyframe selection method proposed in [25] is employed to select query images for localization.

In the visual map, each visual feature on the database image associates with a 3D point in the global coordinate system, i.e., the indoor coordinate system, which makes it realizable to obtain the 3D position of the visual feature. As shown in Figure 2, the indoor scene can be fully reconstructed by the visual map, and at the same time of the visual map construction, the poses of the RGB-D sensor (i.e., the database camera) are recorded and stored in the map. In the process of the visual map construction, the RGB-D sensor captures an RGB image and a depth disparity image, simultaneously. Then, the 3D point clouds are aligned using the visual map construction algorithm [10]. When the visual map construction has been completed, the 3D position of each visual feature point in indoor scenes is determined in the indoor coordinate system.

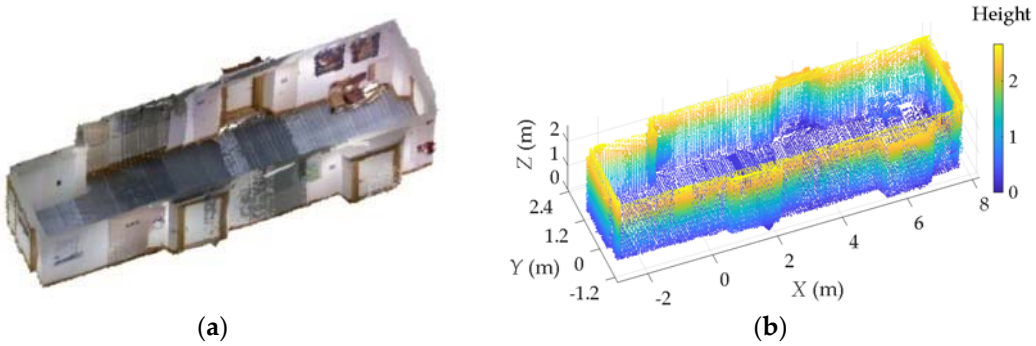


Figure 2. Example of pre-constructed visual map in corridor scene. (**a**) Visual feature points in visual map; (**b**) Aligned point clouds in visual map.

For global localization, the best-matched database image with the query image should firstly be retrieved in the pre-constructed visual map. During the process of visual map construction, the visual features on database images are described by ORB (oriented FAST and rotated BRIEF) descriptors. As an alternative to the SIFT (scale invariant feature transform) or the SURF (speeded up robust feature) descriptors, the ORB descriptor is tested to be rotation invariant and resistant to noise, and it is effective and stable for visual localization [28]. The visual features on the query image are also described by ORB descriptors. The Bag-Of-Features (BOF) [29] approach is used in the visual map construction to determine a closed loop of the database camera trajectory [10]. Therefore, the BOF approach can be utilized in this paper to retrieve the matched database images with the query image. Moreover, the best-matched database image can be determined by the 1-precision criterion [30] based on the ORB descriptors. The best-matched database image contains sufficiently matched visual features with the query image.

The epipolar geometry is a typical method for recovering the rigid transformation between two cameras by two-dimensional correspondences, i.e., feature correspondences. Let $\boldsymbol{K}_D$ and $\boldsymbol{K}_Q$ represent the calibration matrices of the database and query cameras, respectively. The position

coordinates $\boldsymbol{X}_D$ and $\boldsymbol{X}_Q$ of the matched descriptors on the database image and the query image can be normalized by:

$$\widetilde{\boldsymbol{X}}_D = \boldsymbol{K}_D^{-1}\boldsymbol{X}_D, \qquad \widetilde{\boldsymbol{X}}_Q = \boldsymbol{K}_Q^{-1}\boldsymbol{X}_Q \tag{1}$$

The relationship between the normalized coordinates of matched descriptors in the query image and the database image can be described as:

$$\widetilde{\boldsymbol{X}}_D \boldsymbol{E} \widetilde{\boldsymbol{X}}_Q = 0 \tag{2}$$

where $\boldsymbol{E}$ is the essential matrix. The essential matrix is used to present the camera transformation parameters up to an unknown scale between the database and query cameras in the following form:

$$\boldsymbol{E} \simeq [\boldsymbol{t}_E]_\times \boldsymbol{R}_E \tag{3}$$

where $\boldsymbol{t}_E$ represents the translation vector and $\boldsymbol{R}_E$ denotes the rotation matrix. $[\boldsymbol{t}_E]_\times$ is the skew-symmetric matrix of $\boldsymbol{t}_E = [t_x, t_y, t_z]^{\mathrm{T}}$.

The essential matrix $\boldsymbol{E}$ is determined by the normalized position coordinates of the matched ORB descriptors on the query and database images. The essential matrix can be computed by the five-point algorithm [31]. The translation vector $\boldsymbol{t}_E$ and the rotation matrix $\boldsymbol{R}_E$ can be extracted from the essential matrix $\boldsymbol{E}$ using singular value decomposition.

To clearly express spatial relationships, four coordinate systems are defined which are the indoor coordinate system $OXYZ$, the query camera coordinate system $O_Q X_Q Y_Q Z_Q$, the database camera coordinate system $O_D X_D Y_D Z_D$, and the query image coordinate system $O_I X_I Y_I$. As shown in Figure 3, the epipolar constraint reflects the relative transformation between the query and database cameras. The scale-free position relationship between the query camera and the database camera can be described as:

$$\boldsymbol{P}_D = \boldsymbol{R}_E \boldsymbol{P}_Q^R + \boldsymbol{t}_E \tag{4}$$

where $\boldsymbol{P}_D$ and $\boldsymbol{P}_Q^R$ denote the 3D positions of database and query cameras, respectively.

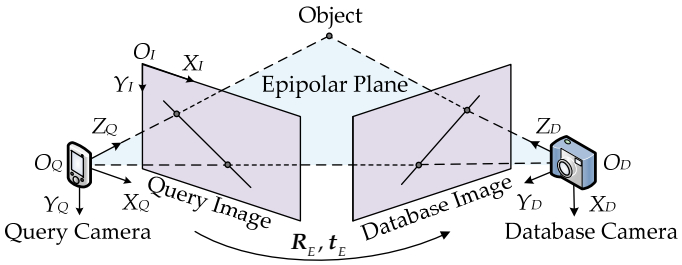


Figure 3. Epipolar constraint between query camera and database camera.

2.2. Absolute Scale Estimation Based on the PWLS Algorithm

Because the translation vector $\boldsymbol{t}_E$ that is exacted from the essential matrix is always a unit vector, the query camera position can only be recovered from the database camera position up to an absolute scale. This means that without the absolute scale, the definite position of the query camera cannot be obtained. As shown in Figure 4, a smile logo on the wall is captured by the query camera and database camera. Based on the epipolar constraint, the rotation matrix $\boldsymbol{R}_E$ and translation vector $\boldsymbol{t}_E$ between the query and database cameras can be computed from the correspondences of image features. Because the database camera positions are known and stored in the visual map, due to the projective principle, the relative positions of the query camera can be estimated, but only up to an unknown scale, which is the scale ambiguity problem. The number of possible query camera positions, such as the

positions $\boldsymbol{p}'_Q$, $\boldsymbol{p}''_Q$ and $\boldsymbol{p}'''_Q$, is infinite. However, the translation vectors that correspond to these possible positions are satisfied with:

$$\boldsymbol{t}_E = s_1\boldsymbol{t}'_E = s_2\boldsymbol{t}''_E = s_3\boldsymbol{t}'''_E \tag{5}$$

where s_1, s_2 and s_3 are different scales of the translation vector.

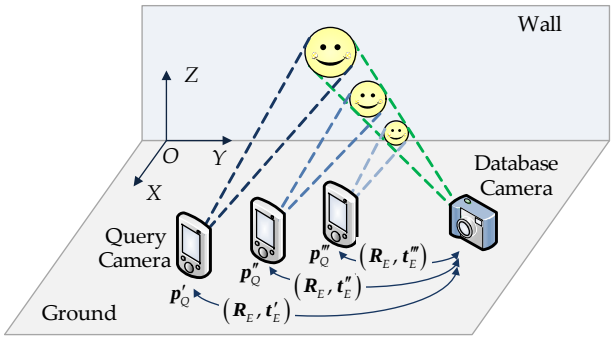


Figure 4. Illustration of scale ambiguity problem in monocular localization.

According to the above analysis, only if the absolute scale of the translation vector is determined can the definite position of the query camera be obtained. Therefore, in this paper, an absolute scale estimation method is proposed, which is based on the pre-constructed visual map, and more specifically, it is based on the best-matched database image and the corresponding point cloud. In the pre-constructed visual map, each pixel on the database image corresponds to a 3D point in the indoor coordinate system. Thus, the 3D positions of visual features in an indoor scene are known. As shown in Figure 5, a query image is captured by the user in an office room, and then the best-matched database image is retrieved in the visual map. In this way, the point cloud that corresponds to the best-matched database image can be found, and the 3D positions of the matched visual features on the query image are obtained.

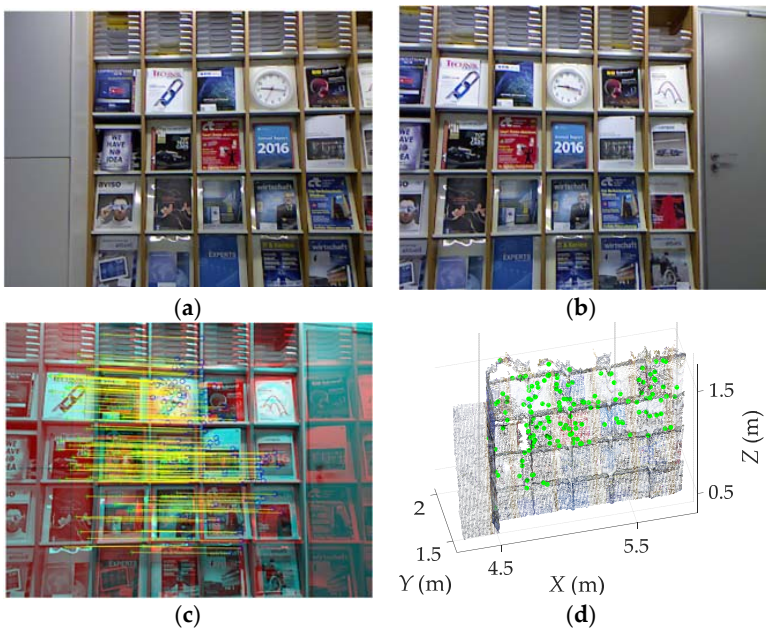


Figure 5. Matched visual features and corresponding 3D points. (**a**) Query image; (**b**) Best-matched database image in visual map; (**c**) Matched visual features between query and database images; (**d**) Matched visual features on a portion of visual map.

After finding the best-matched database image and the corresponding point cloud, it is convenient to obtain the image point position matrix $\boldsymbol{X}_Q = [\boldsymbol{u}_Q, \boldsymbol{v}_Q]$ of n matched features on the query image and the corresponding 3D point position matrix $\boldsymbol{X}_P = [\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}]$, where $\boldsymbol{u}_Q = [u_1, \cdots, u_n]^{\mathrm{T}}$, $\boldsymbol{v}_Q = [v_1, \cdots, v_n]^{\mathrm{T}}$, $\boldsymbol{x} = [x_1, \cdots, x_n]^{\mathrm{T}}$, $\boldsymbol{y} = [y_1, \cdots, y_n]^{\mathrm{T}}$, and $\boldsymbol{z} = [z_1, \cdots, z_n]^{\mathrm{T}}$. The position of a certain feature can be represented by a homogeneous coordinate $\widetilde{\boldsymbol{X}}_Q^i = [u_i, v_i, 1]^{\mathrm{T}}$, and the corresponding 3D point can be represented by a homogeneous coordinate $\widetilde{\boldsymbol{X}}_P^i = [x_i, y_i, z_i, 1]^{\mathrm{T}}$. According to the camera model [21], the relationship between the image point and the corresponding 3D point is defined as:

$$n_d \widetilde{\boldsymbol{X}}_Q^i = [\boldsymbol{R}_E | s\boldsymbol{t}_E] \widetilde{\boldsymbol{X}}_P^i \tag{6}$$

where n_d is the projective depth factor and s is the unknown absolute scale. The rotation matrix $\boldsymbol{R}_E$ and the translation vector $\boldsymbol{t}_E$ which is extracted from the essential matrix can be described as:

$$\boldsymbol{R}_E = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \tag{7}$$

$$\boldsymbol{t}_E = [t_1, t_2, t_3]^{\mathrm{T}} \tag{8}$$

Equation (6) also can be rewritten as:

$$n_d \begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} = \left[\begin{array}{ccc|c} r_{11} & r_{12} & r_{13} & st_1 \\ r_{21} & r_{22} & r_{23} & st_2 \\ r_{31} & r_{32} & r_{33} & st_3 \end{array}\right] \begin{bmatrix} x_i \\ y_i \\ z_i \\ 1 \end{bmatrix} \tag{9}$$

Based on the 2D-to-3D correspondence implied in Equation (9), three equality relationships can be obtained:

$$n_d u_i = r_{11} x_i + r_{12} y_i + r_{13} z_i + st_1 \tag{10}$$

$$n_d v_i = r_{21} x_i + r_{22} y_i + r_{23} z_i + st_2 \tag{11}$$

$$n_d = r_{31} x_i + r_{32} y_i + r_{33} z_i + st_3 \tag{12}$$

Substitute Equation (12) into Equation (10) and Equation (11) to remove the depth factor n_d:

$$(t_3 u_i - t_1)s = (r_{11} - u_i r_{31})x_i + (r_{12} - u_i r_{32})y_i + (r_{13} - u_i r_{33})z_i \tag{13}$$

$$(t_3 v_i - t_2)s = (r_{21} - v_i r_{31})x_i + (r_{22} - v_i r_{32})y_i + (r_{23} - v_i r_{33})z_i \tag{14}$$

According to Equation (13), it can be found that the absolute scale s depends on four aspects: the position u_i in the X_I-direction on the query image, the rotation matrix $\boldsymbol{R}_E$, the translation vector $\boldsymbol{t}_E$, and the coordinate $\widetilde{\boldsymbol{X}}_P^i$. However, different from Equation (13), Equation (14) demonstrates that the absolute scale s is determined by the position v_i instead of the position u_i. It indicates that the absolute scale depends on the image positions of visual features not only in the X_I-direction but also in the Y_I-direction. For a pair of the query image and the best-matched database image, in most cases, the motion of matched visual features occurs both in the X_I-direction and the Y_I-direction in the query image coordinate system $O_I X_I Y_I$. However, the direction in which the motion extends further depends on how the user moves and holds the smartphone.

When there is only one matched visual feature between the query image and the best-matched database image, the absolute scale can be determined by Equation (13) or (14). However, in practice, there is more than one matched visual feature. Thus, the absolute scale estimation is an over-determined problem. The most common method to solve this problem is the least squares

method [32], which is a traditional method in regression analysis for approximating the solution of an over-determined problem.

The least squares method assumes that all observations make an equal contribution to estimation results. However, for the absolute scale estimation, the moving distances of the matched visual features in different image directions make different contributions to the result of the scale estimation. Specifically, for one visual feature on the image, if the moving distance d_X in the X_I-direction is greater than the moving distance d_Y in the Y_I-direction, d_X should make more contributions to the scale estimation, and vice versa. This is because a greater moving distance is more robust to the noise caused by feature detection and matching, which is beneficial to the absolute scale estimation. Therefore, fully considering of camera moving directions and pixel moving distances, a pixel-distance weighted least squares method is proposed for estimating the absolute scale.

For solving the estimation problem with the least squares method, the general measurement model for the least squares method can be written as:

$$\boldsymbol{y} = \boldsymbol{H}\boldsymbol{x} + \boldsymbol{v} \tag{15}$$

where $\boldsymbol{y}$ represents a measurement vector that contains noise, $\boldsymbol{H}$ is a known observation matrix, $\boldsymbol{v}$ is some additional noise, and $\boldsymbol{x}$ is the state variable to be estimated. Given an estimated variable $\hat{\boldsymbol{x}}$ of $\boldsymbol{x}$, the error $\boldsymbol{\varepsilon}$ can be expressed as:

$$\boldsymbol{\varepsilon} = \boldsymbol{y} - \boldsymbol{H}\hat{\boldsymbol{x}} \tag{16}$$

For the absolute scale estimation, according to Equations (13) and (14), the image position errors of a certain matched visual feature can be calculated by:

$$\varepsilon_1 = \boldsymbol{M}_1\boldsymbol{X}_P^i - (t_3u_i - t_1)s \tag{17}$$

$$\varepsilon_2 = \boldsymbol{M}_2\boldsymbol{X}_P^i - (t_3v_i - t_2)s \tag{18}$$

where $\boldsymbol{X}_P^i = [x_i, y_i, z_i]^{\mathrm{T}}$, $\boldsymbol{M}_1 = [(r_{11} - u_ir_{31}), (r_{12} - u_ir_{32}), (r_{13} - u_ir_{33})]$, and $\boldsymbol{M}_2 = [(r_{21} - v_ir_{31}), (r_{22} - v_ir_{32}), (r_{23} - v_ir_{33})]$. $\boldsymbol{M}_1\boldsymbol{X}_P^i$ and $\boldsymbol{M}_2\boldsymbol{X}_P^i$ are treated as the measurement values, and s is the state variable that needed to be estimated by the measurement values. For the total number of n matched visual features between the query image and the best-matched database image, the error vector $\boldsymbol{\varepsilon}$ can be calculated by:

$$\boldsymbol{\varepsilon} = \begin{bmatrix} (r_{11} - u_1r_{31})x_1 + (r_{12} - u_1r_{32})y_1 + (r_{13} - u_1r_{33})z_1 - (t_3u_1 - t_1)s \\ (r_{21} - v_1r_{31})x_1 + (r_{22} - v_1r_{32})y_1 + (r_{23} - v_1r_{33})z_1 - (t_3v_1 - t_2)s \\ \vdots \quad\quad \vdots \quad\quad \vdots \quad\quad \vdots \\ (r_{11} - u_nr_{31})x_n + (r_{12} - u_nr_{32})y_n + (r_{13} - u_nr_{33})z_n - (t_3u_n - t_1)s \\ (r_{21} - v_nr_{31})x_n + (r_{22} - v_nr_{32})y_n + (r_{23} - v_nr_{33})z_n - (t_3v_n - t_2)s \end{bmatrix} \tag{19}$$

where the measurement vector is:

$$\boldsymbol{y} = \begin{bmatrix} (r_{11} - u_1r_{31})x_1 + (r_{12} - u_1r_{32})y_1 + (r_{13} - u_1r_{33})z_1 \\ (r_{21} - v_1r_{31})x_1 + (r_{22} - v_1r_{32})y_1 + (r_{23} - v_1r_{33})z_1 \\ \vdots \quad\quad \vdots \quad\quad \vdots \\ (r_{11} - u_nr_{31})x_n + (r_{12} - u_nr_{32})y_n + (r_{13} - u_nr_{33})z_n \\ (r_{21} - v_nr_{31})x_n + (r_{22} - v_nr_{32})y_n + (r_{23} - v_nr_{33})z_n \end{bmatrix} \tag{20}$$

and the observation vector is:

$$\boldsymbol{H} = [(t_3u_1 - t_1), (t_3v_1 - t_2), \cdots, (t_3u_n - t_1), (t_3v_n - t_2)]^{\mathrm{T}} \tag{21}$$

For a pair of the query image and the best-matched database image, the pixel-distance weight in the image direction X_I or Y_I is the ratio of the moving distance in this direction to the whole moving distance of all matched visual features in the image plane. As there are n matched features between the query image and the database image, and their positions on the query and database images are $\boldsymbol{X}_Q = \left[\boldsymbol{u}_Q, \boldsymbol{v}_Q\right]$ and $\boldsymbol{X}_D = [\boldsymbol{u}_D, \boldsymbol{v}_D]$, where $\boldsymbol{u}_Q = [u_1, \cdots, u_n]^{\mathrm{T}}$, $\boldsymbol{v}_Q = [v_1, \cdots, v_n]^{\mathrm{T}}$, $\boldsymbol{u}_D = \left[u'_1, \cdots, u'_n\right]^{\mathrm{T}}$ and $\boldsymbol{v}_D = \left[v'_1, \cdots, v'_n\right]^{\mathrm{T}}$, the pixel-distance weights for a matched visual feature can be calculated by:

$$w_1^i = \frac{\left|u_i - u'_i\right|}{\sum_{j=1}^{n}\left(\left|u_j - u'_j\right| + \left|v_j - v'_j\right|\right)} \tag{22}$$

$$w_2^i = \frac{\left|v_i - v'_i\right|}{\sum_{j=1}^{n}\left(\left|u_j - u'_j\right| + \left|v_j - v'_j\right|\right)} \tag{23}$$

where w_1^i and w_2^i are the weights for the scale estimation in the image directions X_I and Y_I, respectively. For all the n matched visual features, the weighted matrix can be represented by:

$$\boldsymbol{W} = \begin{bmatrix} w_1^1 & 0 & \cdots & 0 & 0 \\ 0 & w_2^1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & w_1^n & 0 \\ 0 & 0 & \cdots & 0 & w_2^n \end{bmatrix} \tag{24}$$

Then, the scale estimation problem is transformed into a weighted least squares form [33] in which the sum of the squared errors is minimized with the matrix $\boldsymbol{W}$ using the measurement vector $\boldsymbol{y}$ and the observation vector $\boldsymbol{H}$ according to the cost function:

$$\boldsymbol{J}(\hat{s}) = (\boldsymbol{y} - \boldsymbol{H}\hat{s})^{\mathrm{T}}\boldsymbol{W}(\boldsymbol{y} - \boldsymbol{H}\hat{s}) \tag{25}$$

where $\hat{s}$ is the estimated value of the absolute scale s. To minimize $\boldsymbol{J}$ with respect to $\hat{s}$, it is necessary to compute its partial derivative and set it to zero:

$$\frac{\partial \boldsymbol{J}(\hat{s})}{\partial \hat{s}} = -\boldsymbol{H}^{\mathrm{T}}\left(\boldsymbol{W} + \boldsymbol{W}^{\mathrm{T}}\right)(\boldsymbol{y} - \boldsymbol{H}\hat{s}) = -\boldsymbol{H}^{\mathrm{T}}\left(\boldsymbol{W} + \boldsymbol{W}^{\mathrm{T}}\right)\boldsymbol{y} + \boldsymbol{H}^{\mathrm{T}}\left(\boldsymbol{W} + \boldsymbol{W}^{\mathrm{T}}\right)\boldsymbol{H}\hat{s} = 0 \tag{26}$$

Since $\boldsymbol{W}$ is a diagonal matrix, its transpose can be obtained by:

$$\boldsymbol{W}^{\mathrm{T}} = \boldsymbol{W} \tag{27}$$

Then, the estimated value $\hat{s}$ of the absolute scale can be calculated by:

$$\hat{s} = \left(\boldsymbol{H}^{\boldsymbol{T}}\boldsymbol{W}\boldsymbol{H}\right)^{-1}\left(\boldsymbol{H}^{\boldsymbol{T}}\boldsymbol{W}\boldsymbol{y}\right) \tag{28}$$

Having performed the PWLM algorithm, the absolute position $\boldsymbol{p}_Q^A$ of the query camera can be calculated with the database camera position $\boldsymbol{p}_D$ by:

$$\boldsymbol{p}_Q^A = \boldsymbol{R}_E^{-1}(\boldsymbol{p}_D - \hat{s}\boldsymbol{t}_E) \tag{29}$$

The absolute position of the query camera is determined by three factors: (1) the position $\boldsymbol{p}_D$ of the database camera, which is stored in the visual map, (2) the rotation matrix $\boldsymbol{R}_E$ and translation vector $\boldsymbol{t}_E$ extracted from the essential matrix, and (3) the estimated value $\hat{s}$ of the absolute scale.

In the initial localization stage, the camera positions corresponding to the first two query images are estimated by global localization, and the two absolute positions are used for the relative position estimation. In addition, the absolute position estimation is implemented once the drifts caused by the relative position estimation exceed the given drift threshold, which will be discussed in Section 3.

3. Map Interaction-Based Drift Detection for Local Localization

In this section, a map interaction-based drift detection method is presented for local localization. The relative position estimation is achieved based on the triangulation and solving the PnP problem. During local localization, drift detection is applied to the keyframes to estimate the drift distance, which will be particularly discussed in this section.

3.1. Relative Position Estimation Based on Triangulation and PnP

After the initial positions of the query camera has been estimated by the first two query images, the subsequent camera positions can be acquired by the relative position estimation. The relative position estimation is implemented based on two technologies: (1) the position estimation of common features by triangulation, and (2) the relative position estimation of the query camera by solving the PnP problem. The process of the relative position estimation does not depend on the pre-built visual map, but it iteratively acquires the query camera positions step by step. Since no image retrieval process is involved in the relative position estimation, the time consumption is lower than that of the absolute position estimation.

The main advantage of global localization is that there are no accumulative errors because the current camera position does not depend on the previous camera positions. However, it is necessary to retrieve the best-matched database image to the query image, which increases the computing time on image retrieval. Therefore, in the proposed monocular localization system, only the first two positions of the query camera are estimated by global localization. From the third query image, the relative position estimation method is executed until the drift exceeds the given threshold.

The relative position estimation method is illustrated in Figure 6. The first two positions $\boldsymbol{p}_Q^1$ and $\boldsymbol{p}_Q^2$ of the query camera are calculated by the global position estimation method using the two query images, the best-matched database images, and the positions $\boldsymbol{p}_D^i$ and $\boldsymbol{p}_D^j$ of the database camera. From the third query image, the 3D positions of visual features in the query image cannot be directly obtained from point clouds. Therefore, these 3D positions must be estimated based on the previous two query images and the corresponding query camera positions. First, the common features, namely the matched visual features, are found in the first two query images. Then these features are matched with the visual features in the third query image. In this way, the common features that are contained within the first three query images are selected for triangulation. As shown in Figure 6, the database image interval l_{in} is defined as the distance between the two successive positions of the database camera.

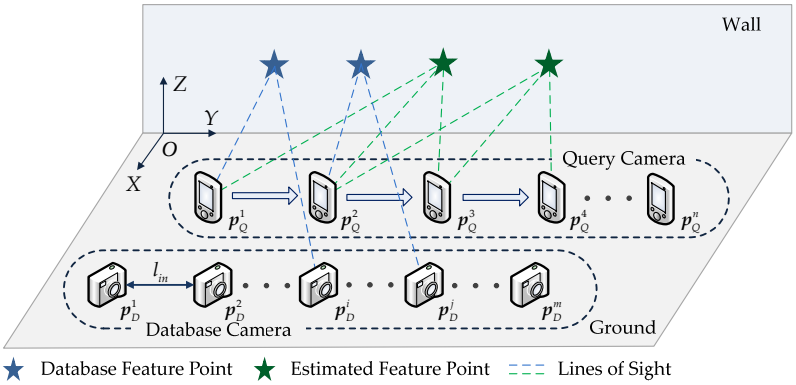


Figure 6. Illustration of relative position estimation method.

The estimated feature point is defined as a 3D point with the estimated position of the common feature, and the database feature point is a 3D point with the true position of the common feature that is stored in the visual map. Triangulation is a common method for finding the positions of common features in 3D space, which is selected as an appropriate approach for computing the 3D positions of estimated feature points. In this paper, the ray that starts from the camera lens center and passes through to the corresponding image point of the visual feature is called the line of sight. If the configuration of the cameras and the camera intrinsic parameters are known, the intersection of two lines of sight that correspond to the same visual feature theoretically can be computed, which is the 3D position of the visual feature in space. However, due to the noise that is caused by correspondence detection of visual features, the two lines of sight may not intersect at one point. Therefore, an optimization algorithm for estimating the 3D positions of visual features in triangulation is needed.

Suppose a pair of image points $\boldsymbol{p}_I^1 = \left[x_I^1, y_I^1\right]^{\mathrm{T}}$ and $\boldsymbol{p}_I^2 = \left[x_I^2, y_I^2\right]^{\mathrm{T}}$ in the first two query images constitute a point correspondence, and the two image points are projected from the same visual feature in space. The homogeneous coordinate vectors $\overline{\boldsymbol{m}}_1$ and $\overline{\boldsymbol{m}}_2$ of the point correspondence can be calculated by:

$$\overline{\boldsymbol{m}}_1 = \left[x_I^1/f_0, y_I^1/f_0, 1\right], \qquad \overline{\boldsymbol{m}}_2 = \left[x_I^2/f_0, y_I^2/f_0, 1\right] \tag{30}$$

where f_0 is a scale constant, which is approximately equal to the image size [34,35]. The point correspondence relationship can be expressed in terms of the homogeneous coordinate vectors $\overline{\boldsymbol{m}}_1$ and $\overline{\boldsymbol{m}}_2$:

$$\overline{\boldsymbol{m}}_1 \boldsymbol{F} \overline{\boldsymbol{m}}_2^{\mathrm{T}} = 0 \tag{31}$$

where $\boldsymbol{F}$ is the fundamental matrix that can be computed based on matched visual features, i.e., the common features, in the first two query images [24]. Note that $\overline{\boldsymbol{m}}_1$ and $\overline{\boldsymbol{m}}_2$ represent the true positions of the matched features in the first and second query images, respectively.

However, in practice, the measured correspondence point positions $\boldsymbol{m}_1$ and $\boldsymbol{m}_2$, which are obtained by visual feature matching, may not coincide with $\overline{\boldsymbol{m}}_1$ and $\overline{\boldsymbol{m}}_2$ due to the noises caused by feature detection. Therefore, the measured correspondence points cannot strictly satisfy Equation (31). The optimal correction method, which was proposed in [35], is employed to estimate the 3D positions of common features between the first two query images. The optimal correction is an extended method of the first-order correction [36] for triangulation from two views.

In the relative position estimation, the 3D positions of common features for solving the PnP problem on the current query image are always triangulated based on the previous two query images and the corresponding positions of the query camera. The 3D positions of common features, i.e., the positions of estimated feature points, obtained by triangulation are used both to solve the PnP problem and detect drifts.

According to the result of the initial localization, the query camera positions $\boldsymbol{p}_Q^1$ and $\boldsymbol{p}_Q^2$ that correspond to the two query images are known. As shown in Figure 7, the 3D positions of common features, namely, the intersection of the two lines of sight that correspond to the same visual feature, such as $\boldsymbol{p}_1$, $\boldsymbol{p}_2$, $\boldsymbol{p}_3$, and $\boldsymbol{p}_t$, can be triangulated based on the query camera positions $\boldsymbol{p}_Q^1$ and $\boldsymbol{p}_Q^2$, and the image positions of the common features in the first two query images. With the image positions and the 3D positions of common features, the query camera position corresponding to the third query image can be computed by means of solving the PnP problem. In practice, the number of common features among the first three query images is always greater than three, which can satisfy the requirement for solving for the camera pose relationship by utilizing 3D-to-2D correspondences. In this paper, an effective method, namely, the EPnP method [22], is employed to acquire the query camera positions.

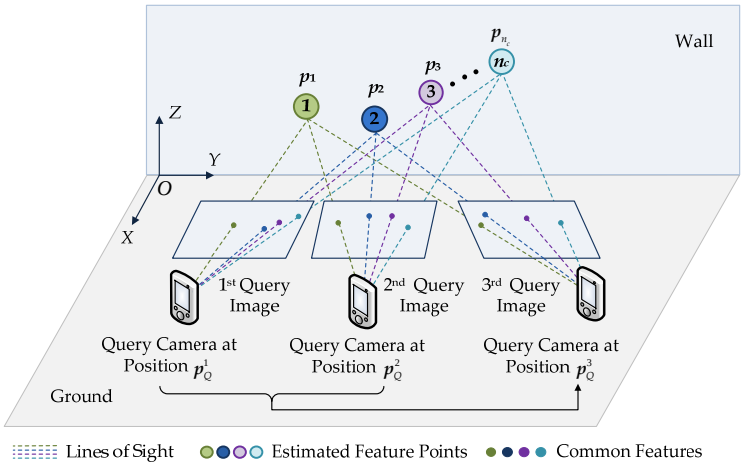


Figure 7. Schematic diagram of relative position estimation by solving the PnP problem.

As shown in Figure 7, a set of three ordered query images are selected as an example to illustrate the relative position estimation method. For the first two query images, query camera positions are estimated by global localization, as discussed earlier. Then, the common features between the first two query images and the third query images are found, and the 3D positions $\boldsymbol{p}_k$ $(k = 1, 2, \cdots, n_c)$ of the common features are acquired by triangulation, where k is the index of the common features and n_c is the total number of the common features. For the EPnP method, four virtual control points $\left\{ z_W^l : l = 1, 2, 3, 4 \right\}$ are required to express the 3D points in the form of a weighted sum. One of the virtual points is chosen as the centroid of $\boldsymbol{p}_k$, and the rest are chosen as the principal directions of the 3D points $\boldsymbol{p}_k$. The solution of the EPnP method is the 3D coordinates $\boldsymbol{z}_C^l$ of the virtual control points in the camera coordinate system. The transformation relationship, which is represented by the rotation matrix $\boldsymbol{R}_T$ and translation vector $\boldsymbol{t}_T$, between the indoor coordinate system and the camera coordinate system, can be solved by [37]:

$$\boldsymbol{z}_C^l = \boldsymbol{R}_T \boldsymbol{z}_W^l + \boldsymbol{t}_T \tag{32}$$

Taking advantage of $\boldsymbol{R}_T$ and $\boldsymbol{t}_T$, the camera position that corresponds to the third query image can be obtained. From the third query image, the camera positions are estimated by triangulation and solving the PnP problem until the drifts are detected and exceed the given threshold. Considering the cumulative errors that are generated in the process of position iterations, the local bundle adjustment [38,39] is utilized to optimize the poses of the camera in the relative position estimation.

3.2. Line Model Establishment for Drift Detection

Since the relative position estimation inevitably leads to cumulative errors due to position iterations, for the most visual SLAM systems, cumulative errors are commonly reduced by global optimization methods on the basis of closed loops. However, the global optimization methods cannot be employed in indoor localization scenarios, because user trajectories usually cannot form closed loops. Therefore, a drift detection method is introduced based on indoor map interaction for reducing cumulative errors (i.e., localization drifts). The main advantages of the introduced method are that there is no need to form closed loops, and that the drift correction is automatically triggered once the drift distance exceeds the given threshold. Based on the drift detection method, a switching strategy is achieved for monitoring cumulative errors. In the process of localization, when the estimated drift value is greater than the given threshold, the local localization mode will be interrupted and switch to the global localization mode to reduce accumulative errors.

There are three steps in the map interaction-based drift detection method: (1) establishing line models by database feature points; (2) removing unreliable features from the common feature set,

and (3) computing the drift distance by the proposed ML-MLESAC algorithm. Before introducing the drift detection method, some line models should be established to find the inliers of the estimated feature points for the drift distance estimation. Because the line segments in the line models are fitted by the database feature points that are with well linear characteristics, the line segments are applicable to exclude the outliers of the estimated feature points.

To reduce the time consumption, a subset of the query images are selected as the keyframes for drift detection. The keyframes are employed to determine whether the relative position estimation should be interrupted and the absolute position estimation should be activated. This decision is based on the interaction between the pre-constructed dense 3D map and the estimated feature points. The strategy for keyframe selection is to choose the frames from every t_{key} query images. The drift detection interval t_{key} depends on the accuracy and efficiency requirements of localization.

As shown in Figure 8, if the t^{th} query image is selected as a keyframe, the t^{th} position of the query camera is solved by the previous two camera positions. Specifically, the common features are selected among the current t^{th} query image, the previous $(t-1)^{\text{th}}$ and $(t-2)^{\text{th}}$ query images. In the relative position estimation, the estimated feature points, which have been used to solve the PnP problems, are triangulated from the previous two query images. That is, the positions of the estimated feature points are determined by the two factors: the image positions of the common features in the previous two query images, and the corresponding positions of the query camera.

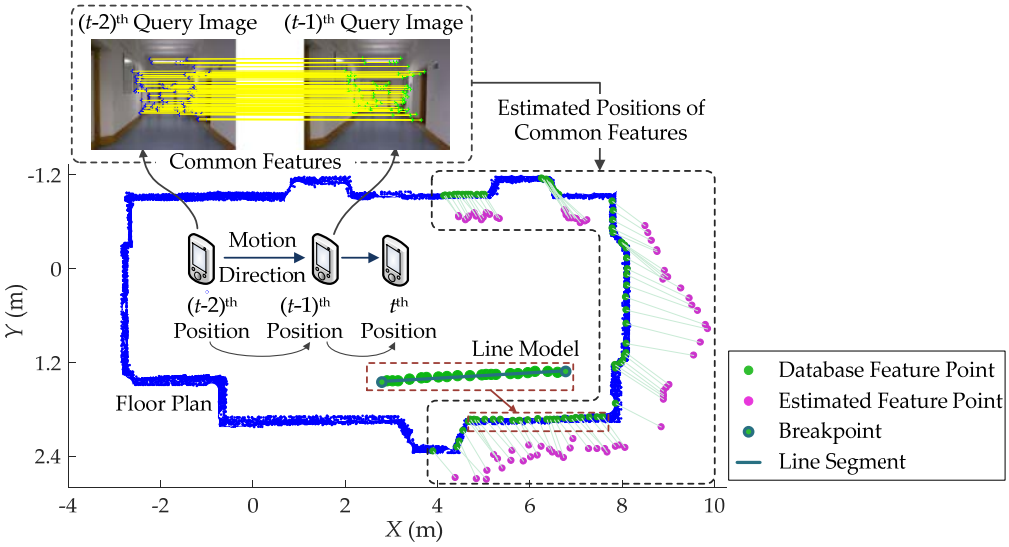


Figure 8. Schematic diagram of drift detection method based on dense 3D map interaction.

In practice, the pedestrian walks on the ground, and at the same time, positions of the pedestrian are estimated by the visual localization method. The drifts caused by the relative position estimation mainly appear and accumulate in the X- and Y-directions. Therefore, in drift detection, only the X- and Y-coordinates of the 3D feature points are considered, which means that the 3D feature points are represented by their projections (i.e. 2D points) on the XOY plane. Specifically, in this section, the estimated feature point is defined as a 2D point on the XOY plane, whose coordinates are same as the X- and Y-coordinates of the estimated 3D position of the common feature. In addition, the database feature points are defined as the 2D points with the same X- and Y-coordinates of the true 3D position of the common feature. The true positions of the common features can be found in the visual map by the same procedure as query image retrieval in the database (as described in Section 2.1). However, the search area is limited within a determined range near the keyframe. Each estimate feature point (shown as a purple dot in Figure 8) is associated with a database feature point (shown as a green dot in Figure 8) in the visual map.

As stated, the indoor scene is reconstructed by the visual map that contains a mass of 3D points in the $OXYZ$ coordinate system. If only the X- and Y-coordinates of the 3D points are considered, each 3D point stored in the visual map corresponds to a 2D point on the ground, i.e., the XOY plane. For all 3D points stored in the visual map, the corresponding 2D points form a floor plan of the indoor scene.

As shown in Figure 8, the points in the database feature point set S_D, as a subset of the 2D points used to form the floor plan, are associated with the common features. Considering all the points in the set S_D, breakpoints of the database feature points are detected by the covariance propagation method [40]. When the breakpoints are obtained, fitted line segments can be acquired by the least squares method using the database feature points in S_D between the breakpoints. Then, the line models are established based on the database feature points. A line model is composed of two elements: (1) a fitted line segment, and (2) a set of database feature points that are used for fitting the line segment. The direction of the line model is defined as the direction of the fitted line segment.

Before discussing the drift detection, it is necessary to analyze the effect of the user motion direction on the triangulation performance. For indoor localization scenarios, it often appears in a typical situation that a user who is holding a smartphone moves down a corridor and looks straight ahead. At the same time, user positions are estimated using the query images that are captured by the smartphone. As shown in Figure 9, two sequential query images are chosen to illustrate how to remove the unreliable features from the common feature set S_C. The database feature points that are stored in the visual map are shown as pentacles with marks 1, 2, and 3 in Figure 9. The line l_{jcc} joins the centers of the query camera, thereby indicating the moving direction of the user on the XOY plane. Each common feature in the set S_C relates to two projection points in the $(t-2)^{\text{th}}$ and the $(t-1)^{\text{th}}$ query images. For a common feature, when the corresponding database feature point is closer to the line l_{jcc}, the distance d_{dis}, as shown in Figure 9, between the projection points is shorter. This phenomenon is also mentioned in [41], whose author emphasized that when the visual feature are close to the line l_{jcc}, the depths of these features are difficult to triangulate. More seriously, if the features lie on the line l_{jcc}, their depths cannot be calculated by triangulation. Therefore, to improve the performance of drift detection, a strategy is proposed for removing the unreliable common features with inaccurately estimated positions.

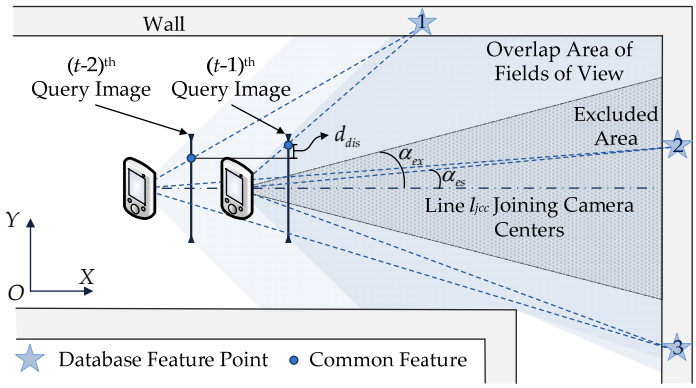


Figure 9. Illustration of removing unreliable key features for drift detection.

Supposing that there is a common feature contained in the keyframe (the t^{th} query image) and the previous two query images, the position of the database feature point that corresponds to the common feature is $\boldsymbol{p}_D = [x_D, y_D]$ on the XOY plane. The 2D positions $\boldsymbol{p}_C^{t-2} = \left[x_C^{t-2}, y_C^{t-2}\right]$ and $\boldsymbol{p}_C^{t-1} = \left[x_C^{t-1}, y_C^{t-1}\right]$ of the query camera are associated with the previous $(t-2)^{\text{th}}$ and $(t-1)^{\text{th}}$ query images. Based on the camera positions, the normal vector $\boldsymbol{n}_{\perp icc}$ of the line l_{jcc} can be computed by

$\boldsymbol{n}_{\perp jcc} = \left[y_C^{t-2} - y_C^{t-1}, x_C^{t-1} - x_C^{t-2}\right]$. Then, the distance $d_{\perp jcc}$ between the database feature point and the line l_{jcc} can be calculated by:

$$d_{\perp jcc} = \frac{\left|\left(y_C^{t-2} - y_C^{t-1}\right)\left(x_C^{t-1} - x_D\right) + \left(x_C^{t-1} - x_C^{t-2}\right)\left(y_C^{t-1} - y_D\right)\right|}{\left\|\boldsymbol{n}_{\perp jcc}\right\|_2} \tag{33}$$

The position relationship between the database feature point and line l_{jcc} can be represented by the angle α_{es}:

$$\alpha_{es} = \arcsin\left(\frac{d_{\perp jcc}}{\left\|\boldsymbol{p}_D - \boldsymbol{p}_C^{t-1}\right\|_2}\right), \left(\alpha_{es} \in [0, \frac{\pi}{2}]\right) \tag{34}$$

Setting an angle threshold α_{ex} for excluding unreliable features from the common feature set S_C, if $\alpha_{es} \leq \alpha_{ex}$, the feature that corresponds to α_{es} is removed from S_C. In this paper, the angle threshold α_{ex} is set to be $\pi/12$. According to the angle threshold, an excluded area can be obtained on the XOY plane, and for the database feature points in this area, such as the database feature point with mark 2 in Figure 9, the corresponding common features are removed from the set S_C.

In addition, if the database feature points are far from the query camera, the corresponding common features are also treated as unreliable features and excluded from the set S_C. The triangulation that is achieved by two query cameras can be treated as a stereo triangulation. Thus, the line between the query cameras can be regarded as the baseline. As mentioned in [26,42], only the nearby visual features whose associated depths are less than 40 times the baseline can be safely triangulated. Therefore, in this paper, this threshold (i.e., 40 times the distance between the query cameras) is chosen for defining distant features. If the distance between the database feature point the query camera exceeds the threshold, the corresponding common feature is removed from S_C. After removing these two types of common features, namely, the features that are close to the line l_{jcc} or far from the query camera, the remaining features in S_C are used for the drift distance estimation.

3.3. Drift Distance Estimation by the Line Model-Based MLESAC Algorithm

Each keyframe associates with one common feature set S_C and two point sets: the database feature point set S_D and the estimated feature point set S_E. For a common feature in the set S_C, there must be a corresponding database feature point in the set S_D whose position is stored in the visual map, and a corresponding estimated feature point in the set S_E whose position is acquired by triangulation. Based on different line models, the estimated feature point set S_E can be divided into several subsets which are called estimated feature point subsets. Specifically, a subset S'_E contains the estimated feature points that are associated with the database feature points that belong to the same line model. Therefore, there must be a line model that corresponds to an estimated feature point subset.

As shown in Figure 10, there are some estimated feature points (shown as purple dots) in the subset S'_E that are associated with the database feature points (shown as green dots) belonging to the same line model. The database feature points stored in the visual map are captured by the RGB-D sensor, and therefore the database feature points in the same line model have well linear characteristics. However, due to the errors caused by triangulation, the linear characteristics of estimated feature points are weakened. Therefore, it is feasible to find the inliers (i.e., the estimated feature points with less errors) by taking advantage of the well linear characteristics of database feature points.

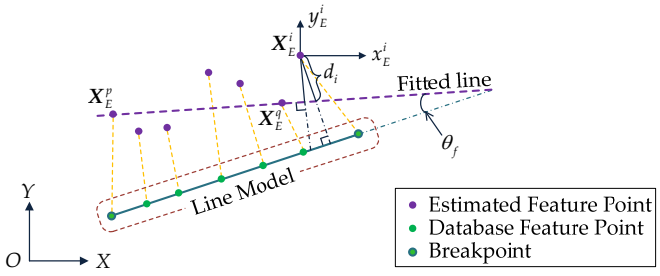


Figure 10. Illustration of Line Model-based MLESAC algorithm.

The aim of the proposed LM-MLESAC algorithm is to fit the points in each estimated feature point subset to a line, and then based on the fitted line, to find the inliers of the estimated feature points. The advantages of the proposed LM-MLESAC algorithm are (1) that a distance function is introduced for fitting the estimated feature points to a line, and (2) that a drift estimation algorithm is investigated based on the fitted line.

Suppose the subset S'_E contains n_e estimated feature points, i.e., $S'_E = \{\boldsymbol{X}_E^1, \boldsymbol{X}_E^2, \cdots \boldsymbol{X}_E^{n_e}\}$. The 2D coordinate $\boldsymbol{X}_E^i (1 \leq i \leq n_e)$ represents the position of the estimated feature point in the form of $\boldsymbol{X}_E^i = \left[x_E^i, y_E^i\right]$. For an estimated feature point, e.g., the point $\boldsymbol{X}_E^i$, it should be determined whether it is an inlier when two points $\boldsymbol{X}_E^p = \left[x_E^p, y_E^p\right]$ and $\boldsymbol{X}_E^q = \left[x_E^q, y_E^q\right]$ in the subset S'_E are randomly sampled to fit a line.

A function d_i is introduced based on the line model as a measurement for calculating the distance between the estimated feature point and the fitted line. As shown in Figure 10, the distance function d_i depends on the length of the line segment between the estimated feature point and the fitted line, and the direction of the segment is perpendicular to the direction of the line model. The distance function d_i for the estimated feature point $\boldsymbol{X}_E^i$ can be expressed as:

$$d_i = \frac{\left|\left(x_E^i - x_E^q\right)\left(y_E^q - y_E^p\right) + \left(y_E^i - y_E^q\right)\left(x_E^p - x_E^q\right)\right|}{\left\|\boldsymbol{X}_E^p - \boldsymbol{X}_E^q\right\|_2 \cdot \cos\theta_f} \tag{35}$$

where $\theta_f \in [0, \pi/2]$ is the intersection angle between the directions of the fitted line and the line model.

Following the idea of MLESAC [27], for an estimated feature point, the probability distribution functions of errors by an inlier and an outlier are expressed as an unbiased Gaussian distribution and a uniform distribution which can be described as follows:

$$p_{in}(d_i) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{d_i^2}{2\sigma^2}\right), \qquad p_{out}(d_i) = \frac{1}{v} \tag{36}$$

where σ is the standard deviation, and v is the search area of the given environment. All estimated feature points in S'_E are distributed in the search area v, which can be calculated by:

$$v = \left|\max_{j=1}^{n_e}(x_e^j) - \min_{j=1}^{n_e}\left(x_e^j\right)\right| \times \left|\max_{j=1}^{n_e}(y_e^j) - \min_{j=1}^{n_e}\left(y_e^j\right)\right| \tag{37}$$

where max(·) and min(·) are the functions for computing the maximum value and the minimum value of a given set of values, respectively.

Combining the Gaussian distribution $p_{in}(d_i)$ and the uniform distribution $p_{out}(d_i)$, a mixture model of the error probability distribution can be described as:

$$p(d_i) = \gamma \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{d_i^2}{2\sigma^2}\right) + (1-\gamma)\frac{1}{v} \tag{38}$$

where γ is the mixing parameter and its initial value is set to be 0.5. When two estimated feature points in S'_E are randomly sampled to fit a line, the remaining $(n_e - 2)$ points in S'_E form a new set S''_E. The specific steps for computing the mixing parameter γ and the standard deviation σ are summarized in Algorithm 1.

Algorithm 1: Iterative solution for the parameters γ and σ

1. Compute the likelihood p^i_{inlier} and $p^i_{outlier}$ of each feature point in the set S''_E by $p^i_{inlier} = \gamma p_{in}(d_i)$ and $p^i_{outlier} = (1-\gamma)p_{out}(d_i)$ with the parameters γ and σ (for initialization, $\gamma_0 = 0.5$ and $\sigma_0 = \sqrt{\sum_{i=1}^{n_e-2} \frac{d_i^2}{n_s}}$);
2. Estimate the expected value e^i_{inlier} of an inlier by $e^i_{inlier} = \frac{p^i_{inlier}}{p^i_{inlier}+p^i_{outlier}}$;
3. Update the mixture parameter γ and the standard deviation σ by $\gamma = \frac{1}{n_e-2}\sum_{i=1}^{n_e-2} e^i_{inlier}$ and $\sigma = \sqrt{\frac{\sum_{i=1}^{n_e-2}\left(e^i_{inlier} d_i^2\right)}{\gamma n_e}}$;
4. Repeat step 1 to step 3 until convergence.

Since the $(n_e - 2)$ points in the set S''_E are independent, the likelihood of these points satisfies:

$$p\left(S''_E\right) = \prod_{i=1}^{n_e-2}\left(p^i_{inlier} + p^i_{outlier}\right) \tag{39}$$

where $p^i_{inlier} = \gamma\left(1/\sqrt{2\pi}\sigma\right)\exp\left(-d_i^2/\left(2\sigma^2\right)\right)$ is the likelihood of an estimated feature if it is an inlier, and $p^i_{outlier} = (1-\gamma)/v$ is the likelihood if it is an outlier.

Because $p\left(S''_E\right)$ is usually a small real value, for computational feasibility, the negative log likelihood is used to represent the likelihood of the points as follows:

$$L_{nll} = -\log p\left(S''_E\right) = -\sum_{i=1}^{n_e-2}\log\left(\gamma\frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{d_i^2}{2\sigma^2}\right) + (1-\gamma)\frac{1}{v}\right) \tag{40}$$

The likelihood L_{nll} corresponds to the randomly sampled feature pair (e.g., $\boldsymbol{X}^p_E$ and $\boldsymbol{X}^q_E$) which is used to fit a line. To find the best-fitted line l^*, repeated random trials of feature pair sampling in S'_E are implemented until L_{nll} reaches L^*_{nll}, where:

$$L^*_{nll} = \underset{S''_E}{\operatorname{argmin}}(L_{nll}) \tag{41}$$

Taking advantage of the best-fitted line, inliers in S'_E can be selected by comparing the distance d_i of each estimated feature point with the threshold $d_{in} = 1.96\sigma$ [27]. An estimated feature point is an inlier if it satisfies $d_i \geq d_{in}$.

Based on different line models, every estimated feature point in the set S_E is checked for whether it is an inlier. All inliers are used to calculate the drift distance d_{drift}. The drift distance d_{drift} indicates accumulative errors that are caused by the relative position estimation, which is defined as:

$$d_{drift} = \frac{1}{n_{in}}\sum_{i=1}^{n_{in}}\left(\left\|\boldsymbol{X}^i_E - \boldsymbol{X}^i_D\right\|_2\right) \tag{42}$$

where n_{in} is the total number of the inliers, $\boldsymbol{X}^i_E$ is the 2D position of an estimated feature point, and $\boldsymbol{X}^i_D$ is the 2D position of the corresponding database feature point. The specific processes of the drift detection method are shown in Algorithm 2.

Algorithm 2: Drift detection based on map interaction

1. Select a keyframe (e.g., the t^{th} query image) from the query image sequence;
2. Find common features between the keyframe, the $(t-1)^{\text{th}}$ and the $(t-2)^{\text{th}}$ query images, and put them in the set S_C;
3. Remove unreliable common features from the set S_C;
4. Calculate the best-fitted lines by the line model-based MLESAC algorithm using the estimated feature points, and then select the inliers of the estimated feature points;
5. Compute the drift distance d_{drift} by the inliers.

A threshold d_{th} is given to monitor the localization drifts, where $d_{drift} \geq d_{th}$ indicates that the estimated drift distance exceeds the threshold, in which case the localization system must switch to the global localization mode from the local localization mode.

As a summary, Figure 11 shows an overall process of the proposed monocular localization system. The system starts by performing the absolute position estimation, which yields the initial camera positions. Then, the subsequent positions of the camera are acquired by the relative position estimation, in which process the keyframes are chosen from the query image sequence and used to compute the drift distance d_{drift}. If d_{drift} is greater than the threshold d_{th}, the absolute position estimation is activated to reduce the drifts, and after that the relative position estimation is resumed. As a switching strategy, drift detection is executed on the keyframes to determine whether the system should switch to the absolute position estimation to correct user positions or continue the relative position estimation. As a result, the cumulative errors of the localization system are limited within an expected range.

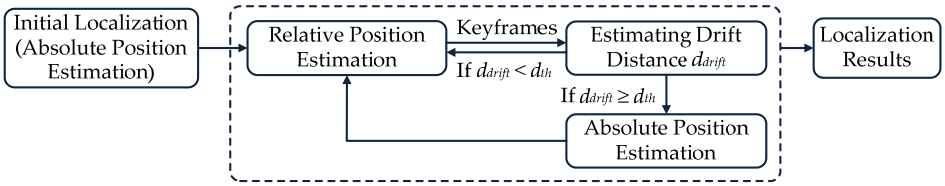


Figure 11. Overall process of proposed monocular localization system.

4. Simulation Results

To evaluate the performance of the proposed monocular localization system, an extensive experiment was conducted in different indoor scenes, such as an office, a corridor, and a gymnasium. The experiment was divided into two parts: the performance evaluation of global localization, and the performance evaluation of local localization. In addition, the performance of the drift detection method was evaluated in the local localization part.

For each experimental scene, the dense 3D map (namely, the visual map) was constructed using the method studied in [10]. RGB-D sensors, such as the Kinect, were applied to construct the map. The RGB-D sensor, whose poses were stored in the visual map, was used as the database camera for localization. In monocular localization, the RGB images that were captured by the RGB-D sensor served as database images, and the corresponding poses of the database camera were devoted to estimating the query camera positions. To precisely evaluate the accuracy of the proposed monocular localization system, a smartphone was placed on a tripod to capture query images at each test point, and the manually measured positions of the test points were used as ground truth to calculate localization errors.

The selected scenes for experiments were typical indoor scenes with different visual and structural complexities, which could fully evaluate the performance of the proposed monocular localization system.

All data processing was performed in MATLAB 2017A (Mathwork, Natick, MA, USA) with an Core i5 CPU (Intel, Santa Clara, CA, USA) and an 8 GB RAM (Kingston, Fountain Valley, CA, USA).

4.1. Performance Evaluation of Global Localization

Estimating the absolute query camera positions requires database images and the corresponding database camera positions, which are provided by the visual map. However, the position accuracy of the database camera has an effect on the global localization performance, i.e., if there are some errors in the database camera positions, these errors will be incorporated into the query camera positions. To avoid this effect, the true positions of the database camera were utilized to estimate the query camera positions. The true positions of the database camera were manually measured in the process of the visual map construction with a step distance of 10 centimeters. In the practical implementation of monocular localization, there is a phenomenon that the localization accuracy is related to the density of the database images. A more dense distribution of database images is conducive to an improvement of the localization accuracy. Therefore, in the global localization experiment, three cases, whose the database image intervals (l_{in}) were set to be 30 cm, 50 cm, and 100 cm, were selected for testing the localization accuracy. In each cases, fifty test points were selected in different environments (i.e., an office, a gymnasium, and a corridor) for evaluating the performances of the local localization methods. Some examples of database images in different experimental environments are shown in Figure 12.

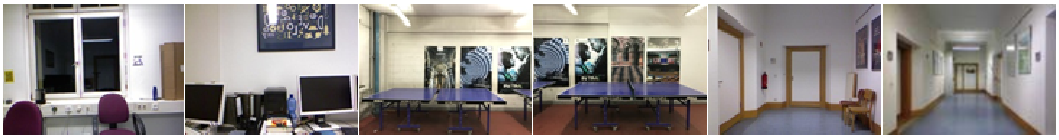


Figure 12. Examples of database images in different experimental environments (i.e., an office, a gymnasium and a corridor).

Before estimating the query camera positions, the most similar database images must be retrieved according to the query image. However, the correctness of the database image retrieval affects the localization accuracy. To eliminate the localization errors that are caused by mismatches in image retrieval, the most similar database images were selected by using the nearest ground-truth database images, which is a typical method for localization experiments [18]. In addition, taking advantage of this method, the running time of image retrieval was excluded from the processing time, and moreover, the scale of the indoor scene did not affect the performance of localization methods.

To effectively evaluate the performance of the proposed global localization method, some representative localization methods were also implemented under the same conditions, such as the PnP method [19,20], the least squares method [21], the 2DTriPnP (2D version of triangulation and perspective n-point) method [18], and the epipolar geometry method [16,17]. Each localization method was implemented in the three cases with different database image densities. By measuring the Euclidean distances between the true and estimated positions of the query camera on the XOY plane, the position errors by various localization methods were obtained. To demonstrate the performance improvement of the proposed method, an accuracy improvement rate i_{im} was introduced, which had the following form:

$$i_{im} = \left(\left|e_p - e_c\right| / e_c\right) \cdot 100\% \tag{43}$$

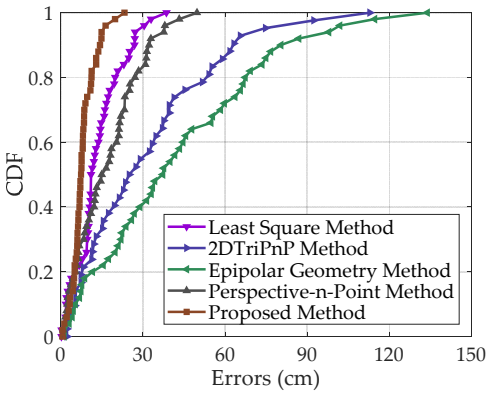
where e_p and e_c denote the average errors by the proposed method and the comparative method, respectively. The average errors, the maximum errors and the improvement rates of global localization are shown in Table 1. In this paper, the abbreviations, i.e., Avg., Max., and Impro., were used to denote the average errors, the maximum errors and the improvement rates, respectively.

Table 1. Position errors by global localization with various localization methods.

Methods	Case 1 (l_{in}=30 cm)			Case 2 (l_{in}=50 cm)			Case 3 (l_{in}=100 cm)		
	Avg. (cm)	Max. (cm)	Impro. (%)	Avg. (cm)	Max. (cm)	Impro. (%)	Avg. (cm)	Max. (cm)	Impro. (%)
Least Squares	13.63	38.59	39.03	19.18	45.96	39.21	22.50	56.72	30.09
2DTriPnP	26.89	112.89	69.10	30.85	121.44	62.20	38.87	151.13	59.53
Epipolar Geometry	43.11	133.85	80.72	49.74	141.13	76.56	60.55	193.19	74.02
Perspective n-Point	17.22	49.72	51.74	21.01	75.40	44.50	32.34	109.27	51.36
Proposed Method	8.31	23.28	-	11.66	28.64	-	15.73	35.41	-

As shown in Table 1, in terms of the average localization errors by different methods, the performance of the proposed method, the PnP method, and the least squares method evidently outperformed the other two methods in the three cases. The reason was that the three methods took advantage of both visual and depth information to estimate query camera positions, whereas only visual information, namely, the database images without depth information, was used in the 2DTriPnP and epipolar geometry methods. Compared with the PnP method, the least squares method and the proposed method achieved better accuracy, especially in the condition of a less dense distribution of database images, as in Case 3, which demonstrates that the two direct scale estimation methods have the advantage in global localization. Because the camera moving direction was fully considered in the absolute scale estimation, the accuracy improvements of the proposed method reached at least 30.09% in all experimental cases, compared with the least squares method.

Figure 13 shows the cumulative distribution functions (CDFs) of the localization errors by the PnP, least squares, 2DTriPnP, and epipolar geometry, and the proposed methods. According to the results shown in Figure 13 and Table 1, the proposed method outperformed the other methods in terms of maximum and average localization errors. Specifically, the maximum localization errors by the proposed method was limited within 0.4 meters under the condition of different database image densities. The accuracy improvement rates in the three cases were at least 39.03%, 39.21%, and 30.09%, compared with other localization methods. Moreover, in Case 1 (database images with an interval of 30 cm), the proposed method achieved the best performance, and the average localization error reached 8.31 cm, which satisfied the requirements of most location-based services.



(**a**)

Figure 13. *Cont.*

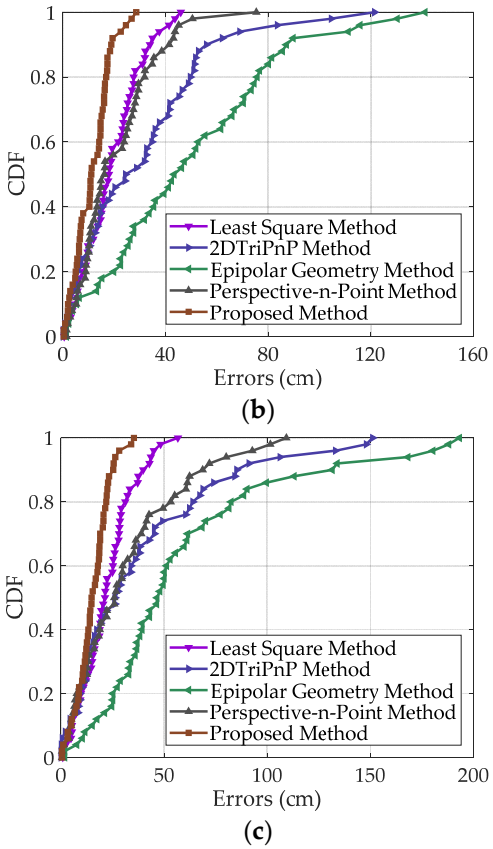


Figure 13. CDFs of average errors by various global localization methods. (**a**) CDFs of average errors in Case 1; (**b**) CDFs of average errors in Case 2; (**c**) CDFs of average errors in Case 3.

To comprehensively evaluate the performance of the proposed global localization method, position errors with respect to multiple confidence levels (25%, 50%, 78%, and 95%) are shown in Table 2. The position error with respect to the 50% confidence level was defined as a median error [18]. The median errors by the proposed global localization method were 7.40 cm, 11.10 cm, and 14.88 cm, which meant, in the three cases, that 50% of the position errors were less than 7.40 cm, 11.10 cm, and 14.88 cm, respectively. For the 95% confidence level, the position errors by the proposed method were limited within 30 cm in all cases. In practical localization applications, an user usually captures query images facing the wall and keeps a distance of more than 30 cm from the wall. When position errors are less than 30 cm, the user position will not be located in the other side of the wall. This is an advantage of the proposed method in that the user is able to identify which room he/she is in.

For the proposed global localization method, since the absolute positions of the query camera were estimated based on the dense 3D map, the localization performance was affected by the accuracy of 3D point clouds acquired by the Kinect (Microsoft, Redmond, USA). The accuracy of 3D point clouds depended on two main aspects: (1) the inherent errors of the sensors equipped on the Kinect, and (2) the errors of calibration between visual and depth cameras. According to existing research [43], the random errors of the depth sensor on the Kinect increase with measuring distances, and the maximum error of the depth sensor is 4cm. As mentioned previously, the study of visual map construction has been achieved in our previous works [10]. Before the map construction, the visual camera and the depth camera were calibrated by the method proposed in [44], and in this method, the accuracy of camera calibration was also investigated.

Table 2. Performance of various global localization methods.

Cases	Methods	Position Errors (cm) with Respect to Confidence Levels			
		25%	50%	78%	95%
Case 1 (l_{in} = 30 cm)	Least Squares	8.64	11.04	19.48	28.79
	2DTriPnP	11.46	25.00	51.74	74.76
	Epipolar Geometry	18.64	37.24	66.61	99.89
	Perspective n-Point	6.35	14.96	25.30	37.83
	Proposed Method	5.65	7.40	11.03	15.63
Case 2 (l_{in} = 50 cm)	Least Squares	9.34	17.96	27.57	39.47
	2DTriPnP	8.65	24.44	48.96	76.27
	Epipolar Geometry	23.16	43.32	74.52	113.51
	Perspective n-Point	9.84	16.16	29.48	44.43
	Proposed Method	6.00	11.10	16.42	23.58
Case 3 (l_{in} = 100 cm)	Least Squares	12.65	21.33	29.26	44.33
	2DTriPnP	12.13	26.23	62.32	119.69
	Epipolar Geometry	30.83	46.56	81.49	174.28
	Perspective n-Point	11.79	25.88	49.50	86.48
	Proposed Method	10.46	14.88	21.49	27.31

4.2. Performance Evaluation of Local Localization

The local localization experiments were executed in the following experimental scenes: an office, a gymnasium, and a corridor. According to the room areas, the user trajectories (l_u) were selected with different lengths of: 10 m, 15 m, and 20 m. For each trajectory, the test points were uniformly selected with a step distance of 10 centimeters. Thus, there were 100, 150, and 200 test points selected in the office, the gymnasium, and the corridor, respectively. The true positions of test points were manually measured for estimating distance errors by drift detection and position errors by local localization.

In existing works, two methods are typically used to perform local localization: the EPnP method [22] and the EPnP in combination with local bundle adjustment (BA) method [25,26]. In this paper, the two typical methods and the proposed method were tested under the same conditions to evaluate the performance of the proposed method. In addition to the position errors by local localization, the distance error e_d of the drift estimation was calculated at each test point by the following equation:

$$e_d = \left| d_{drift} - d_{cam} \right| \tag{44}$$

where d_{drift} is the estimated drift distance, and d_{cam} is the distance between the estimated and true positions of the query camera. For the drift estimation, d_{drift} is computed by the LM-MLESAC algorithm. A general idea of the drift distance estimation is that, without removing the unreliable common features and outliers of the estimated feature points, all features in the set S_C are employed to calculate localization drifts. Based on this idea, a general drift estimation method was implemented as a comparative method.

The distance errors by the two drift estimation methods in different scenes are shown in Table 3. In the three scenes, the average and maximum distance errors by the proposed method were significantly less than those of the general method. The reason is that the proposed method removes the unreliable common features and the outliers of the estimated feature points. Therefore, the inliers yield a better performance of the drift estimation. A greater distance error in the drift estimation will falsely trigger global localization, especially when the drift threshold is set to be a small value, which will lead to an increase in the time consumption and poor user experience.

The cumulative distribution functions of distance errors by the proposed drift estimation algorithm are shown in Figure 14. In different experimental scenes, the performance of the proposed drift estimation algorithm was less affected by the lengths of the user trajectories. In the gymnasium scene, the drift estimation performance was better than in the other two scenes, because the gymnasium had a simple structure, and distinctive visual features were distributed on the wall. Based on the experiment, we conclude that, for the drift estimation, a simple building structure will yield a better result.

Table 3. Distance errors by drift estimation in different scenes.

Scenes	Evaluation Criterions	General Method	Proposed Method
Office (Number of test points = 100)	Avg. (cm)	29.11	8.95
	Max. (cm)	137.66	24.73
	Impro. (%)	69.25	-
Gymnasium (Number of test points = 150)	Avg. (cm)	24.36	6.14
	Max. (cm)	113.75	19.61
	Impro. (%)	74.79	-
Corridor (Number of test points = 200)	Avg. (cm)	28.93	9.42
	Max. (cm)	142.17	28.29
	Impro. (%)	67.44	-

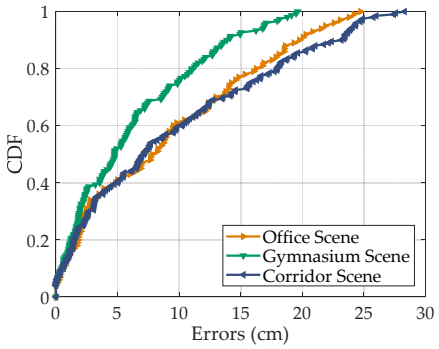


Figure 14. CDFs of distance errors by proposed drift estimation algorithm in different indoor scenes.

To further evaluate the performance of the proposed drift estimation algorithm, the distance errors with respect to different confidence levels (25%, 50%, 78%, and 95%) are shown in Table 4. Since the distance errors by the drift estimation were limited within 30 cm in most situations, the recommended threshold d_{th} should be equal to or greater than 30 cm. If the threshold d_{th} was less than 30 cm, global localization would be frequently triggered due to the drift estimation errors.

Table 4. Performance of proposed drift estimation algorithm.

Scenes	Distance Errors (cm) with Respect to Confidence Levels			
	25%	50%	78%	95%
Office (Number of test points = 100)	2.06	7.80	15.35	21.87
Gymnasium (Number of test points = 150)	1.73	4.76	10.69	16.93
Corridor (Number of test points = 200)	2.17	7.12	17.16	24.32

The local localization experiment of the proposed method was carried out based on the initial (global) localization. In the process of local localization, drift detection was implemented on the keyframes. Once the drift distance exceeded the given threshold, global localization was triggered on the condition that the interval of database images was set to be 30 cm (i.e., $l_{in} = 30$ cm). For the office scene, the gymnasium scene, and the corridor scene, the drift thresholds (d_{th}) were set to be 30 cm, 40 cm, and 60 cm, respectively. In each experimental scene, the same initial positions obtained by the proposed global localization method were offered to various local localization methods. The drift detection interval t_{key} was set to be 10 frames.

The performances of various local localization methods in different experimental scenes are shown in Table 5. The local localization experiment was also a comprehensive experiment on the proposed monocular localization system, because global (initial) localization and drift detection were included in the local localization experiment. With the aid of drift detection and global localization, the position errors by the local localization method were significantly reduced and the accumulative errors did not increase along with the lengths of the trajectories. For local localization, the query camera positions were iteratively calculated, which inevitably led to accumulative errors. For example, in terms of the average

errors by the PnP method, there was an increase from 58.86 cm in the office scene to 122.80 cm in the corridor scene. If the length of the trajectory was further extended, the position errors would continue to increase, which would ultimately leads to an unacceptable position error for indoor localization. Owing to the drift detection method, in the three experimental scenes, the maximum position errors were limited within 35 cm, 46 cm, and 68 cm, which obviously outperformed the comparative methods.

Table 5. Position errors by local localization with various localization methods.

Methods	Office (Number of Test Points=100, d_{th}=30 cm)			Gymnasium (Number of Test Points = 150, d_{th}=40 cm)			Corridor (Number of Test Points=200, d_{th}=60 cm)		
	Avg. (cm)	Max. (cm)	Impro. (%)	Avg. (cm)	Max. (cm)	Impro. (%)	Avg. (cm)	Max. (cm)	Impro. (%)
PnP Method	58.86	107.63	69.01	106.07	179.70	79.76	122.80	230.96	73.83
PnP + BA Method	53.01	104.18	65.59	93.46	163.35	77.03	95.91	195.21	66.49
Proposed Method	18.24	34.73	-	21.47	45.22	-	32.14	67.69	-

The position errors and the CDFs of the position errors by various local localization methods are shown in Figure 15. With the drift thresholds of 30 cm, 40 cm, and 60 cm in the office scene, the gymnasium scene, and the corridor scene, the maximum errors by the proposed method were limited within 34.73 cm, 45.22 cm, and 67.69 cm, respectively. In these scenes, the maximum errors slightly exceeded the drift threshold. There are two main factors that led to this problem. First, not every query image is used for drift detection. Therefore, the drifts cannot be strictly limited within the given threshold. Second, the estimated drift distance contains errors, which results in the unsuccessful activation of global localization when the true drift distance is larger than the threshold. However, this problem did not seriously affect the performance of local localization, as demonstrated by the average errors by the proposed method. In addition, the experimental results also indicated that the local bundle adjustment algorithm contributed to the increase in the localization accuracy to an extent.

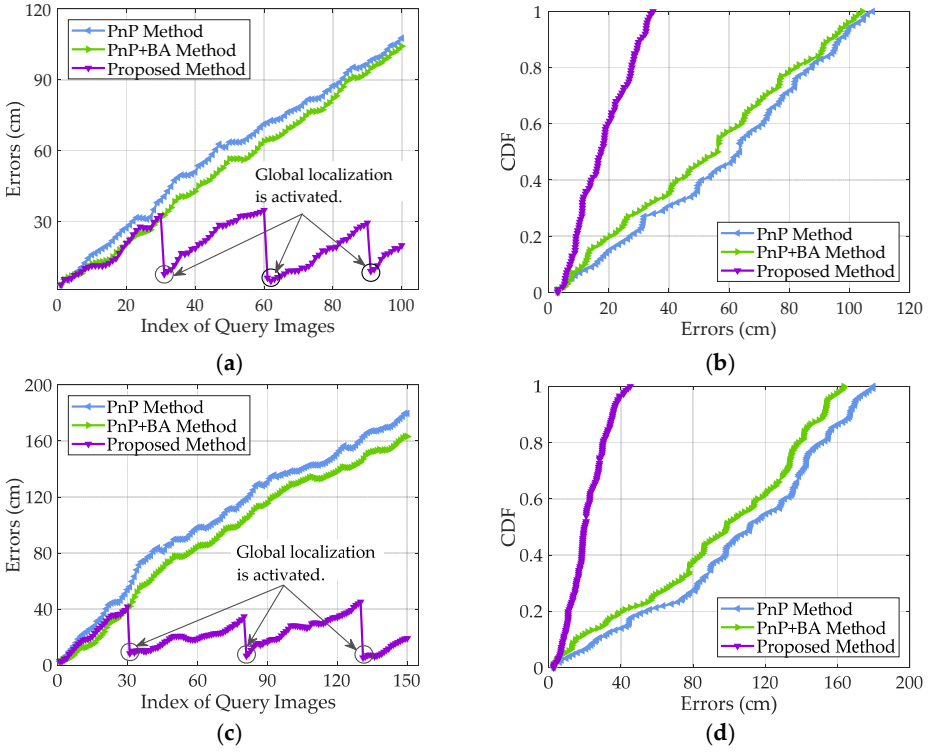


Figure 15. *Cont.*

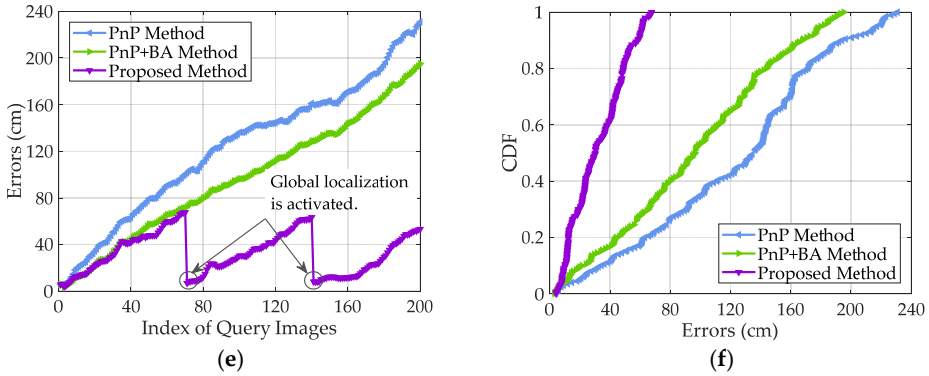


Figure 15. Position errors and CDFs of position errors by various local localization methods. (**a**) Position errors in office scene; (**b**) CDFs of position errors in the office scene; (**c**) Position errors in the gymnasium scene; (**d**) CDFs of position errors in the gymnasium scene; (**e**) Position errors in the corridor scene; (**f**) CDFs of position errors in the corridor scene.

To clearly illustrate the performance of the proposed local localization method, position errors with respect to multiple confidence levels (25%, 50%, 78%, and 95%) are shown in Table 6. According to the experimental results, the median errors by the proposed method were 17.46 cm, 19.99 cm, and 30.19 cm, corresponding to the 30-cm, 40-cm, and 60-cm drift thresholds, respectively. The experimental results demonstrated that the accumulative errors could be effectively reduced by the proposed localization method. Moreover, in all experimental scenes, the average and maximum errors by the proposed method were limited within an acceptable range that satisfied the requirements of most indoor localization and navigation applications.

Table 6. Performance of local localization with various methods.

Scenes	Methods	Position Errors (cm) with Respect to Confidence Levels			
		25%	50%	78%	95%
Office (Number of test points = 100, d_{th} = 30 cm)	PnP	31.48	63.51	84.97	101.15
	PnP + BA	25.21	56.07	79.26	98.49
	Proposed Method	10.34	17.46	27.33	32.54
Gymnasium (Number of test points = 150, d_{th} = 40 cm)	PnP	75.75	111.56	146.78	170.42
	PnP + BA	57.74	98.39	135.73	154.55
	Proposed Method	13.71	19.99	29.85	38.37
Corridor (Number of test points = 200, d_{th} = 60 cm)	PnP	78.48	135.43	164.94	217.17
	PnP + BA	56.66	96.49	137.38	179.80
	Proposed Method	14.73	30.19	47.96	61.51

For the proposed localization system, the average computation time of the absolute position estimation, drift detection, and the relative position estimation were 0.647 s, 0.234 s, and 0.217 s, as shown in Table 7. As mentioned previously, database image retrieval was not applied in the experiments, so there was no computation time for image retrieval included in the system. It was obvious that the absolute position estimation needed more time to compute the epipolar relationship and estimate the absolute scale. Therefore, for the keyframes in local localization, drift detection was first employed to decide whether the absolute position estimation should be activated, which was a time-saving strategy for the localization system.

Table 7. Computation time of the proposed localization system.

	Global Localization Mode	Local Localization Mode	
	Absolute Position Estimation (s)	Relative Position Estimation (s)	Drift Detection (s)
Computation Time	0.647	0.217	0.234

5. Conclusions

In this paper, a drift-aware monocular localization system is proposed, which is based on a pre-constructed dense 3D map for indoor environments. The proposed system can be divided into two modes: the global localization mode and the local localization mode. Considering the impact of the camera moving direction on the scale estimation, a pixel-distance weighted least squares algorithm is investigated in global localization for computing the absolute scale, which is used to acquire the absolute positions of the query camera. Because the accumulative errors caused by the relative position estimation are inevitable, a drift detection method is introduced, and the drift distance is estimated by the proposed line model-based MLESAC algorithm. In the process of localization, with the aid of drift detection, the system switches from the local localization mode to the global localization mode when the estimated drift distance is greater than the given threshold, which effectively reduces the accumulative errors that are caused by position iteration. According to the experimental results, under the condition of different database image densities, the average and maximum position errors by the proposed global localization method are limited within 16 cm and 36 cm, respectively, which outperforms the comparative localization methods. In the three experimental scenes, taking advantage of the proposed drift detection method, the maximum position errors by local localization are limited within 35 cm, 46 cm, and 68 cm, corresponding to the 30-cm, 40-cm, and 60-cm thresholds, respectively. Compared with the comparative methods, the position accuracies of global and local localization in the proposed monocular localization system are improved by at least 30.09% and 65.59%, respectively. The experimental results also show that the introduced drift detection method achieves a higher accuracy of the drift estimation, compared with the general method. In the future, how to manage a mass of pre-constructed 3D dense maps will be studied, and efficient image retrieval methods for indoor visual localization will be further investigated.

Author Contributions: All authors contributed to this paper. G.F. conceived the idea and designed the experiments, and wrote the paper. L.M. supervised and guided this study. D.Q. gave comments and suggestions for this work. X.T. reviewed the manuscript and provided comments that enhanced the quality of this paper.

Funding: National Natural Science Foundation of China (61571162, 61771186), Ministry of Education-China Mobile Research Foundation (MCM20170106), Heilongjiang Province Natural Science Foundation (F2016019), and University Nursing Program for Young Scholars with Creative Talents in Heilongjiang Province (UNPYSCT-2017125).

Acknowledgments: Many thanks are given to the editors and the anonymous reviewers for their insightful comments on this paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Grewal, M.S. Global navigation satellite systems. *Wiley Interdiscip. Rev. Comput. Stat.* **2011**, *3*, 383–384. [CrossRef]
2. Li, X.; Zhang, X.; Ren, X.; Fritsche, M.; Wickert, J.; Schuh, H. Precise positioning with current multi-constellation global navigation satellite systems: GPS, GLONASS, Galileo and BeiDou. *Sci. Rep.* **2015**, *5*, 8328. [CrossRef] [PubMed]
3. Yang, C.; Shao, H.R. WiFi-based indoor positioning. *IEEE Commun. Mag.* **2015**, *53*, 150–157. [CrossRef]
4. Du, X.; Yang, K. A map-assisted WiFi AP placement algorithm enabling mobile device's indoor positioning. *IEEE Syst. J.* **2017**, *11*, 1467–1475. [CrossRef]
5. Huang, C.H.; Lee, L.H.; Ho, C.C.; Wu, L.L.; Lai, Z.H. Real-time RFID indoor positioning system based on Kalman-filter drift removal and Heron-bilateration location estimation. *IEEE Trans. Instrum. Meas.* **2015**, *64*, 728–739. [CrossRef]
6. Wang, C.; Shi, Z.; Wu, F. Intelligent RFID indoor localization system using a gaussian filtering based extreme learning machine. *Symmetry* **2017**, *9*, 30. [CrossRef]

7. De, A.G.; Moschitta, A.; Carbone, P. Positioning techniques in indoor environments based on stochastic modeling of UWB round-trip-time measurements. *IEEE Trans. Intell. Transp. Syst.* **2016**, *17*, 2272–2281. [CrossRef]
8. Monica, S.; Ferrari, G. UWB-based localization in large indoor scenarios: Optimized placement of anchor nodes. *IEEE Aerosp. Electron. Syst. Mag.* **2015**, *51*, 987–999. [CrossRef]
9. Huitl, R.; Schroth, G.; Hilsenbeck, S.; Schweiger, F.; Steinbach, E. TUMindoor: An extensive image and point cloud dataset for visual indoor localization and mapping. In Proceedings of the 2012 19th IEEE International Conference on Image Processing, Lake Buena Vista, FL, USA, 30 September–3 October 2012; IEEE: Piscataway, NJ, USA, 2012.
10. Feng, G.; Ma, L.; Tan, X. Visual map construction using RGB-D sensors for image-based localization in indoor environments. *J. Sens.* **2017**, *99*, 1–18. [CrossRef]
11. Se, S.; Lowe, D.; Little, J. Local and global localization for mobile robots using visual landmarks. In Proceedings of the 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems, Maui, HI, USA, 29 October–3 November 2001; IEEE: Piscataway, NJ, USA, 2001.
12. Scaramuzza, D.; Fraundorfer, F. Visual odometry [tutorial]. *IEEE Robot. Autom. Mag.* **2011**, *18*, 80–92. [CrossRef]
13. Choi, S.; Joung, J.H.; Yu, W.; Cho, J.I. What does ground tell us? Monocular visual odometry under planar motion constraint. In Proceedings of the 2011 11th International Conference on Control, Automation and Systems, Gyeonggi-do, Korea, 26–29 October 2011; IEEE: Piscataway, NJ, USA, 2001.
14. Scaramuzza, D.; Siegwart, R. Appearance-guided monocular omnidirectional visual odometry for outdoor ground vehicles. *IEEE Trans. Robot.* **2008**, *24*, 1015–1026. [CrossRef]
15. Choi, S.; Park, J.; Yu, W. Resolving scale ambiguity for monocular visual odometry. In Proceedings of the 2013 10th International Conference on Ubiquitous Robots and Ambient Intelligence, Jeju, Korea, 30 October–2 November 2013; IEEE: Piscataway, NJ, USA, 2013.
16. Sadeghi, H.; Valaee, S.; Shirani, S. A weighted KNN epipolar geometry-based approach for vision-based indoor localization using smartphone cameras. In Proceedings of the 2014 IEEE 8th Sensor Array and Multichannel Signal Processing Workshop, A Coruna, Spain, 22–25 June 2014; IEEE: Piscataway, NJ, USA, 2014.
17. Zhang, Y.; Ma, L.; Tan, X. Smart phone camera image localization method for narrow corridors based on epipolar geometry. In Proceedings of the 2016 International Wireless Communications and Mobile Computing Conference, Paphos, Cyprus, 5–9 September 2016; IEEE: Piscataway, NJ, USA, 2016.
18. Sadeghi, H.; Valaee, S.; Shirani, S. 2DTriPnP: A robust two-dimensional method for fine visual localization using Google streetview database. *IEEE Veh. Technol. Mag.* **2016**, *66*, 4678–4690. [CrossRef]
19. Deretey, E. Visual Localization in Underground Mines and Indoor Environments Using PnP. Master's Thesis, Queen's University, Kingston, ON, Canada, January 2016.
20. Deretey, E.; Ahmed, M.T.; Marshall, J.A.; Greenspan, M. Visual indoor positioning with a single camera using PnP. In Proceedings of the 2015 International Conference on Indoor Positioning and Indoor Navigation, Banff, AB, Canada, 13–16 October 2015; IEEE: Piscataway, NJ, USA, 2015.
21. Esteban, I.; Dorst, L.; Dijk, J. Closed form solution for the scale ambiguity problem in monocular visual odometry. In Proceedings of the International Conference on Intelligent Robotics and Applications, Shanghai, China, 10–12 November 2010; Springer: Berlin, Germany, 2010.
22. Lepetit, V.; Moreno, N.F.; Fua, P. EPnP: An accurate o(n) solution to the PnP problem. *Int. J. Comput. Vis.* **2009**, *81*, 155. [CrossRef]
23. Kümmerle, R.; Grisetti, G.; Strasdat, H.; Konolige, K.; Burgard, W. G^2o: A general framework for graph optimization. In Proceedings of the 2011 IEEE International Conference on Robotics and Automation, Shanghai, China, 9–13 May 2011; IEEE: Piscataway, NJ, USA, 2011.
24. Hartley, R.; Zisserman, A. *Multiple View Geometry in Computer Vision*, 2nd ed.; Cambridge University Press: Cambridge, UK, 2000; pp. 344–357. ISBN 9780521540513.
25. Mur-Artal, R.; Montiel, J.M.M.; Tardos, J.D. ORB-SLAM: A versatile and accurate monocular SLAM system. *IEEE Trans. Robot.* **2015**, *31*, 1147–1163. [CrossRef]
26. Mur-Artal, R.; Tardós, J.D. ORB-SLAM2: An open-source slam system for monocular, stereo, and RGB-D cameras. *IEEE Trans. Robot.* **2017**, *33*, 1255–1262. [CrossRef]

27. Torr, P.H.S.; Zisserman, A. MLESAC: A new robust estimator with application to estimating image geometry. *Comput. Vis. Image Underst.* **2000**, *78*, 138–156. [CrossRef]
28. Rublee, E.; Rabaud, V.; Konolige, K.; Bradski, G. ORB: An efficient alternative to SIFT or SURF. In Proceedings of the 2011 IEEE International Conference on Computer Vision, Barcelona, Spain, 6–13 November 2011; IEEE: Piscataway, NJ, USA, 2011.
29. Nowak, E.; Jurie, F.; Triggs, B. Sampling strategies for bag-of-features image classification. In Proceedings of the European Conference on Computer Vision 2006, Graz, Austria, 7–13 May 2006; Springer: Berlin, Germany, 2006.
30. Mikolajczyk, K.; Schmid, C. A performance evaluation of local descriptors. *IEEE Trans. Pattern Anal. Mach. Intell.* **2005**, *27*, 1615–1630. [CrossRef] [PubMed]
31. Nistér, D. An efficient solution to the five-point relative pose problem. *IEEE Trans. Pattern Anal. Mach. Intell.* **2004**, *26*, 756–770. [CrossRef] [PubMed]
32. Levenberg, K. A method for the solution of certain non-linear problems in least squares. *Q. Appl. Math.* **1944**, *2*, 164–168. [CrossRef]
33. Strutz, T. *Data Fitting and Uncertainty: A Practical Introduction to Weighted Least Squares and Beyond*, 1st ed.; Vieweg and Teubner: Wiesbaden, Germany, 2011; pp. 25–46. ISBN 9783834810229.
34. Hartley, R.I. In defense of the eight-point algorithm. *IEEE Trans. Pattern Anal. Mach. Intell.* **1997**, *19*, 580–593. [CrossRef]
35. Kanatani, K.; Sugaya, Y.; Niitsuma, H. Triangulation from two views revisited: Hartley-Sturm vs. optimal correction. In Proceedings of the 19th British Machine Vision Conference, Leeds, UK, 1–4 September 2008; British Machine Vision Association: Durham, UK, 2008.
36. Kanatani, K. *Statistical Optimization for Geometric Computation: Theory and Practice*; Courier Corporation: Milton Keynes, UK, 2005; pp. 131–170. ISBN 9780444824271.
37. Arun, K.S.; Huang, T.S.; Blostein, S.D. Least-squares fitting of two 3-D point sets. *IEEE Trans. Pattern Anal. Mach. Intell.* **1987**, *5*, 698–700. [CrossRef]
38. Mouragnon, E.; Lhuillier, M.; Dhome, M.; Dekeyser, F.; Sayd, P. Real time localization and 3D reconstruction. In Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, New York, NY, USA, 17–22 June 2006; IEEE: Piscataway, NJ, USA, 2006.
39. Klein, G.; Murray, D. Parallel tracking and mapping for small AR workspaces. In Proceedings of the 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality, Nara, Japan, 13–16 November 2007; IEEE: Piscataway, NJ, USA, 2007.
40. Ji, Q.; Haralick, R.M. Breakpoint detection using covariance propagation. *IEEE Trans. Pattern Anal. Mach. Intell.* **1998**, *20*, 845–851. [CrossRef]
41. Hartley, R.I.; Sturm, P. Triangulation. *Comput. Vis. Image Underst.* **1997**, *68*, 146–157. [CrossRef]
42. Paz, L.M.; Piniés, P.; Tardós, J.D.; Neira, J. Large-scale 6-DOF SLAM with stereo-in-hand. *IEEE Trans. Robot.* **2008**, *24*, 946–957. [CrossRef]
43. Khoshelham, K.; Elberink, S.O. Accuracy and resolution of kinect depth data for indoor mapping applications. *Sensors* **2012**, *12*, 1437–1454. [CrossRef] [PubMed]
44. Herrera, D.; Kannala, J.; Heikkilä, J. Joint depth and color camera calibration with distortion correction. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *34*, 2058–2064. [CrossRef] [PubMed]