sensors

MDPI

Article

SLAM-Based Self-Calibration of a Binocular Stereo Vision Rig in Real-Time

Hesheng Yin [1], Zhe Ma [2], Ming Zhong [2], Kuan Wu [3], Yuteng Wei [3], Junlong Guo [2] and Bo Huang [1,*]

1 State Key Laboratory of Robotics and System, Harbin Institute of Technology, Harbin 150001, China; 18b908038@stu.hit.edu.cn
2 Industrial Research Institute of Robotics and Intelligent Equipment, Harbin Institute of Technology, Weihai 264209, China; 171320515@stu.hit.edu.cn (Z.M.); zhongming@hit.edu.cn (M.Z.); junlongg@hit.edu.cn (J.G.)
3 Sphyrna Technology Company, Beijing 100096, China; kwu@shuangjisha.com (K.W.); ytwei@shuangjisha.com (Y.W.)
* Correspondence: huangboweihai@hit.edu.cn

Received: 23 December 2019; Accepted: 21 January 2020; Published: 22 January 2020

Abstract: The calibration problem of binocular stereo vision rig is critical for its practical application. However, most existing calibration methods are based on manual off-line algorithms for specific reference targets or patterns. In this paper, we propose a novel simultaneous localization and mapping (SLAM)-based self-calibration method designed to achieve real-time, automatic and accurate calibration of the binocular stereo vision (BSV) rig's extrinsic parameters in a short period without auxiliary equipment and special calibration markers, assuming the intrinsic parameters of the left and right cameras are known in advance. The main contribution of this paper is to use the SLAM algorithm as our main tool for the calibration method. The method mainly consists of two parts: SLAM-based construction of 3D scene point map and extrinsic parameter calibration. In the first part, the SLAM mainly constructs a 3D feature point map of the natural environment, which is used as a calibration area map. To improve the efficiency of calibration, a lightweight, real-time visual SLAM is built. In the second part, extrinsic parameters are calibrated through the 3D scene point map created by the SLAM. Ultimately, field experiments are performed to evaluate the feasibility, repeatability, and efficiency of our self-calibration method. The experimental data shows that the average absolute error of the Euler angles and translation vectors obtained by our method relative to the reference values obtained by Zhang's calibration method does not exceed 0.5° and 2 mm, respectively. The distribution range of the most widely spread parameter in Euler angles is less than 0.2° while that in translation vectors does not exceed 2.15 mm. Under the general texture scene and the normal driving speed of the mobile robot, the calibration time can be generally maintained within 10 s. The above results prove that our proposed method is reliable and has practical value.

Keywords: self-calibration; binocular stereo vision rig; extrinsic parameter; SLAM

1. Introduction

The practical application of binocular stereo vision (BSV) rig in the market of unmanned vehicles and mobile robots as sensing equipment has been greatly challenged [1,2]. There is still room for further improvement in the practicability and durability of the BSV rig. The lack of practicality is mainly reflected in its structural form. The positions of two cameras for most existing BSV rigs are relatively fixed, which means the operators can hardly adjust the baseline. This form makes it difficult for binocular vision to be installed in different sized spaces and cannot satisfy the measurement and

sensing range requirements when the robot faces different scale scenes. The drawback of durability is mainly reflected in the fact that BSV rig is often deformed due to temperature, vibration, etc., resulting in changes in the parameters calibrated at the factory. For example, to keep the calibration parameters from deviations, the structure of the rig is usually made of special materials and needs to be firmly connected with robots or cars. As a result, the university research team, autopilot company, and automotive aftermarket industry have to recalibrate the BSV rig frequently for the result accuracy. It is well known that calibration and recalibration of the rig have always been a burden for each user, which requires expertise, specialized equipment and many hours of work [3]. Consequently, it is necessary to study an automatic, robust, real-time calibration algorithm without prior information of the environment, human supervision and auxiliary equipment.

With the popular application of BSV rig in computer vision [4], many methods have been proposed to calibrate them. Most existing calibration methods are based on manual off-line calibration algorithms for specific reference targets or patterns, such as the traditional calibration methods, planar template methods [5], 3D-object-based calibration methods [6]. Typical algorithms of traditional calibration methods include the direct linear transformation algorithm (DLT) [7], the nonlinear optimization algorithm [8], and the Tsai's radial alignment constraint algorithm (RAC) [9]. Deng et al. [10] proposed a relational model for camera calibration, which takes into account the camera's geometric parameters and lens distortion effects. The combination of differential evolution and particle swarm optimization algorithm can effectively calibrate camera parameters. Batista et al. [11] used monoplane calibration points to achieve camera calibration. To avoid the singularity obtained by the calibration equation when using monoplane calibration points, a multi-step procedure combined with a nonlinear optimization iterative process is used to solve these parameters and improve the precision. Zhuang et al. [12] used an RAC method to calibrate the camera through a 2D planar calibration plate parallel to the camera plane. This method combines the advantages of traditional linear and non-linear optimization algorithms to simplify the parameter solving process, so the calibration results are relatively accurate. The traditional calibration method can accurately calibrate the camera parameters utilizing precisely fabricated planar or stereo targets [13]. However, the calibration algorithms and procedures are complex and time-consuming, so they are suitable for where camera parameters are not changed often.

To avoid the shortcomings of traditional methods, the planar template method has been extensively studied. Zhang [14] proposed to calibrate the camera with a checkerboard, which is widely used. The camera is required to view the displayed checkerboard pattern in several different directions. Yu et al. [15] proposed a robust recognition method of checkerboard pattern detection for camera calibration, which is based on Zhang's method. Chen et al. [16] described a novel camera calibration method that used only a single image of two coplanar circles with arbitrary radii to estimate the camera's extrinsic parameters and focal length. Kumar et al. [17] proposed a technique for camera calibration using a planar mirror. By allowing all cameras to see multiple fields of view through the mirror, it is possible to overcome the need for all cameras to see a common calibration object directly.

The 3D target-based method makes the calibration procedure more simplified by placing the target in the camera field of view once to obtain calibration parameters. Su et al. [18] used a spherical calibration object as a specialized reference pattern to calibrates the geometric parameters between any number of cameras on the network. Zhen et al. [19] proposed a method for calibrating extrinsic parameters of BSV rig based on double-ball targets. A target consisting of two identical spheres fixed at a known distance is freely placed in different positions and orientations. With the aid of markers, the calibration precision is high, and the calibration process is simpler, but the calibration method is still not flexible enough. Since it is difficult to machine 3D targets and keep the calibration images of all feature points at the same sharpness, therefore, their practical application is limited.

Unlike the above-mentioned calibration methods, the self-calibration method only requires a constraint from the image sequence without any special reference objects or patterns designed in advance, which may allow online calibration of camera parameters in real-time. Luong et al. [20] proposed earlier to calibrate the extrinsic parameters of BSV rig by point-matching methods in

general unknown situations. Using the SIFT [21] feature point correspondence and bundle adjustment (BA) algorithm, Tang et al. [22] proposed a local-global hybrid iterative optimization method for calibration of BSV rig. Wang et al. [23] proposed a self-calibration method using sea surface images, which can estimate the rotation matrix of a BSV rig with a wide baseline. Boudine et al. [24] proposed a self-calibration technology for BSV rig with variable intrinsic parameters, which is based on the relationship between two matching terms (i.e., the projection of two points representing the vertices of the triangle isosceles rectangular triangle) and the absolute cone image. Ji et al. [25] introduced a calibration model for the multiple fisheye camera rig which combines a generic polynomial and an equidistance projection model to achieve high-accuracy calibration from the fisheye camera to an equivalent ideal frame camera. Wang et al. [26] proposed a self-calibration method that uses a single feature point and converts the pitch and yaw imaging model into a quadratic equation of the pitch tangent value. However, the calibration process for the above work still requires complex user intervention. Some methods require the user to manually specify the point correspondence between the two camera views of BSV rig, which seems inconvenient in practice.

So far, we have found the work of Carrera et al. [27] is closer to ours. They used the improved monocular vision Mono-SLAM algorithm to process the video sequence of each camera separately, then robustly match and fuse the maps obtained by each camera based on the corresponding SURF [28] features. Finally, the extrinsic parameters between multiple cameras can be calibrated through matching the global map with the SURF features extracted from cameras. However, this method does not provide real-scale information on the surrounding environment, and its calibration accuracy and efficiency have yet to be further verified. Heng et al. [29] also used the visual SLAM-based self-calibration method to calibrate the BSV rig extrinsic parameters fixed on the aircraft and provide the scale information through the three-axis gyroscope. Heng et al. [30] further applied a similar idea and method to the calibration of the BSV rig mounted on car platforms and provided the scale information through the odometer to make the algorithm applicable to large scenes. However, the calibration efficiency of this method limits its practical application.

Based on the summary analysis of related research fields, there are still some problems in the calibration work of the BSV rig. The calibration process employing the auxiliary device is complicated and inconvenient for the users to personally operate. The self-calibration method is still in the research and exploration stage, and its calibration efficiency and accuracy have yet to be further verified. To the best of our knowledge, there is no existing self-calibration method that can be practically applied to the calibration of BSV rig in a wide range and achieve commercial value.

In this paper, we propose a novel SLAM-based self-calibration method for the BSV rig. The purpose is to achieve real-time, automatic and accurate calibration of the BSV rig's extrinsic parameters in a short period. The intrinsic parameters of the left and right cameras are estimated in advance using Zhang's method. And as far as possible, this calibration method can be used for large-scale practical applications on arbitrary BSV rigs, such as a car or mobile robot.

The main contributions of our work are reflected in the following aspects:

1. Our proposed SLAM-based self-calibration method can estimate the extrinsic parameters of the BSV rig without auxiliary equipment and special calibration markers.

2. Once the baseline is determined, a fully automatic calibration process can be implemented in real-time during the normal driving of the platform, without complex requirements for rig movement and prior information about the environment.

3. In the experimental part, the proposed self-calibration method was applied to the mobile platform where the BSV module is installed. Our method can accurately calibrate the BSV rig in a short period. As far as we know, there are few existing methods to perform the self-calibration of BSV rig in such high efficiency, thereby achieving the effect of practical application.

The rest of the paper is organized as follows: Section 2 describes our SLAM-based self-calibration pipeline. Section 3 discusses the SLAM-based construction pipeline of a 3D scene point map that used to generate a calibration area map. Section 4 introduces the extrinsic parameter calibration algorithm

of BSV rig. Section 5 explains the experimental setup and verification results. Conclusions and future work are presented in Section 6.

2. SLAM-Based Self-Calibration Pipeline

This section describes our SLAM-based self-calibration pipeline as a whole, which can estimate the extrinsic parameters of the BSV rig. Here, the main contents of the pipeline are briefly summarized and described in the following sections.

Figure 1 shows our SLAM-based self-calibration pipeline. The method is mainly divided into two parts: SLAM-based construction of a 3D feature point map of the natural environment (also called 3D scene point map) and extrinsic parameter calibration. SLAM-based construction of the 3D scene point map is mainly introduced throughout Section 3. To improve the efficiency of calibration, a lightweight, real-time visual SLAM system is built in Section 3.1. The image sequence is captured in the process of BSV rig motion and converted to Grayscale image before being fed to the SLAM process described in Section 3.2. The monocular camera pose tracking step detailed in Section 3.3 estimates camera poses and track inlier feature points extracted from the image. Then, the tracked inlier feature points together with camera poses are used to construct the 3D scene point map of the natural environment described in Section 3.4. The scene point map is composed of a series of 3D feature points, so we also call it a 3D feature point map. To improve the precision of the algorithm, the BA is used to optimize the established 3D feature point map described in Section 3.5.

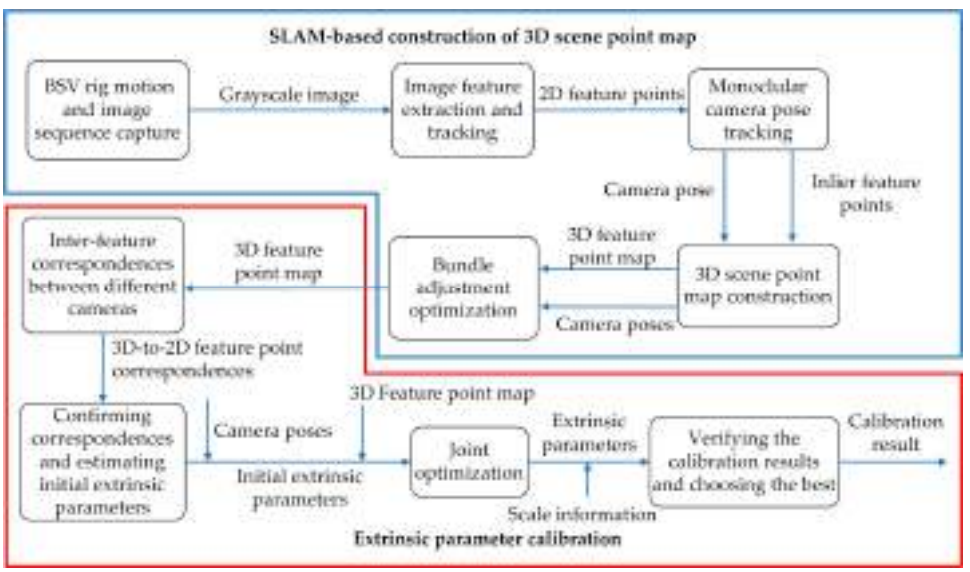


Figure 1. Simultaneous localization and mapping (SLAM)-based self-calibration pipeline which estimates the extrinsic parameter of the binocular stereo vision (BSV) rig.

At that time, an accurate 3D scene point map will be obtained, which can be used as a calibration area. The pipeline then goes into the extrinsic parameter calibration phase detailed in Section 4. The 3D-2D correspondences between the 3D scene points associated with the 2D feature points in the left camera and the 2D feature points in the right camera are obtained through inter-feature correspondences between two binocular cameras described in Section 4.1. Further, the Perspective-n-Point (PnP) method combined with the random sample consensus (RANSAC) algorithm is used to confirm correspondences and estimate the initial extrinsic parameters of BSV rig described in Section 4.2. Subsequently, this initial estimation is used together with the camera poses and 3D feature point map by the joint optimization described in Section 4.3 to acquire accurate extrinsic parameters. Finally, the scale information is given in Section 4.4 and the calibration results are verified to select the optimal result.

3. SLAM-Based Construction of 3D Scene Point Map

This section describes our SLAM-based construction of the natural environment that used to produce a map of the calibration area.

3.1. Lightweight Monocular Visual SLAM System

Garrigues et al. [31] proposed a features from accelerated segment test (FAST)-based optical flow method whose efficiency and performance are better compared to the feature corresponding methods based on slow corner points such as Harris [32], SIFT, and SURF. Therefore, the FAST-based optical flow method is chosen as the basic framework of our SLAM algorithm. However, the existing SLAM algorithm framework is more complicated, it is difficult to directly apply to solve our problems and meet the requirements for calibration efficiency. Therefore, a lightweight monocular vision SLAM system is built as shown in Figure 2. The system is already reflected in Figure 1, and the key points in the framework are detailed in the following sections. Therefore, only further supplements to the SLAM system are made here.

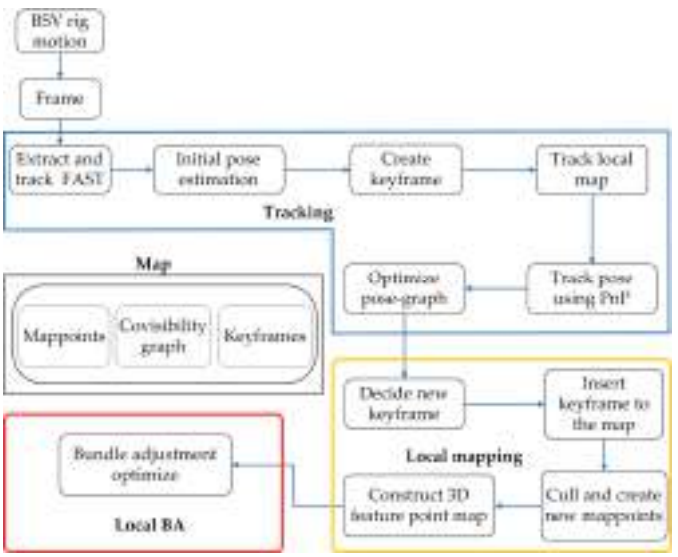


Figure 2. Lightweight monocular visual SLAM system.

The system is mainly composed of four parts: tracking, local mapping, local BA and map. The tracking part locates each frame by extracting and tracking feature points on the image and finds local map features for matching. The local mapping part determines the new keyframes, decides to delete or add new mappoints, and builds a 3D feature point map accordingly. The local BA part manages and optimizes the local map by performing BA. The map part is used to maintain all mappoints, covisibility graph, and keyframes.

The loop-closing detection part like most SLAM algorithms is not performed, because the BSV rig can be calibrated in a shorter path. If the loop-closing detection is added, the place recognition part also needs to be further added, which will increase the complexity of the algorithm.

3.2. BSV Rig Motion and Image Sequence Capture

In this step, the sequence of the monocular camera image is captured during camera motion and begins as a SLAM algorithm. To ensure the convenience and practicability, the self-calibration method can be fully automated in real-time during the normal driving of the platform, without complex requirements for rig movement. The movement form of BSV rig is to include as much translation movement as possible along the opposite direction of the camera's optical axis, and this kind of movement is also a common action on mobile platforms. Here, it is assumed that the two monocular cameras of the BSV rig has at most a minimum overlapping field of view. For example, the calibration

can be performed during the normal forward or backward movement of the car or mobile robot. Aircraft operating in tight spaces can also be calibrated during vertical movement. Of course, this form of movement is ideal, which can ensure the precision and efficiency of the calibration to the greatest extent. This point is analyzed during the initialization of the camera pose tracking in Section 3.3. But it is not limited to this way. This is mainly since the calibration is only performed when a keyframe is encountered. At this time, the local mapping part has already saved the feature point map and image feature points required for calibration.

The image sequences are captured by two monocular cameras of identical configuration while ensuring time synchronization. The algorithm does not require any prior information about the environment, nor does it need to set special markers or patterns in the scene. The image sequence should be converted from RGB to Grayscale before being fed to SLAM.

3.3. Monocular Camera Pose Tracking

The monocular camera poses and inlier feature points are tracked in the camera pose tracking step. The inlier feature points are required for 3D scene point map construction in Section 3.4.

To decrease the time to track the monocular camera pose, the optical flow method is employed to track the FAST detectors as the feature points. In this paper, optical flow information is used for feature corresponding, mappoint selection, outlier feature points culling and pose estimation. The calculation of partial pixel optical flow is referred to as a sparse optical flow, and the sparse optical flow is represented by a Lucas-Kanade optical flow [33]. The FAST detector is a kind of corner point, which mainly detects the obvious change of local pixel grayscale, and is known for its fast speed [34]. The 16-dimensional grayscale description, block optical flow estimation, and gradient descent search for each FAST corner point are the highlights of its success. Therefore, the LK optical flow method is used to track FAST detectors as feature points in this paper. Besides, this paper adopts the classic corner extraction and optical flow tracking strategies in [31]. However, they do not pay attention to the distribution of corners, but the uniform distribution of corners is intuitively important for us to obtain accurate and fast calibration results.

The feature points we extracted on the grayscale image are represented by green rectangles in Figure 3a. They are mainly located in areas where the grayscale values are relatively changed, and most of them are located in the corner points. However, they often appear to be "get together" as shown in Figure 3a. That is, the distribution of feature points is too crowded in a certain local area. This problem is easy to cause feature tracking errors, resulting in feature mismatch, which in turn affects camera pose tracking and 3D scene point map construction. Therefore, we added a sparse and uniform distribution strategy for feature points and hope to distribute the corner points as accurately as possible in all areas of the image.

(**a**) (**b**)

Figure 3. Feature points extraction in the grayscale image: (**a**) Original features from accelerated segment test (FAST) corner points distribution; (**b**) FAST corner points distribution after sparse processing.

In terms of feature points sparseness, the non-maximal suppression method [35] is used to filter local non-maximal points. The effect of the FAST corner distribution after sparse processing is shown in Figure 3b. To improve the feature corresponding performance, a gridded map management strategy similar to the method proposed in [36] is used to maximize the chance of detecting a significant number of corresponding features between the overlapping view of two monocular cameras. The strategy procedure is described as follows: 1. Mesh the feature point map; 2. Count the number of feature points in each grid; 3. Delete the feature points that are too close; 4. If there are no feature points in the grid, select the most significant FAST feature extracted in the region as the new feature point. Thereby, the density under sparseness can be maintained, and the desired number of seed feature points can be ensured for subsequent tracking. Figure 4a shows a feature point map managed by the gridded map, in which a blue rectangle represents an added feature point. The optical flow method is then used to track the feature points processed by the sparse and uniform distribution strategy. The tracking effect is shown in Figure 4b.

(**a**) (**b**)

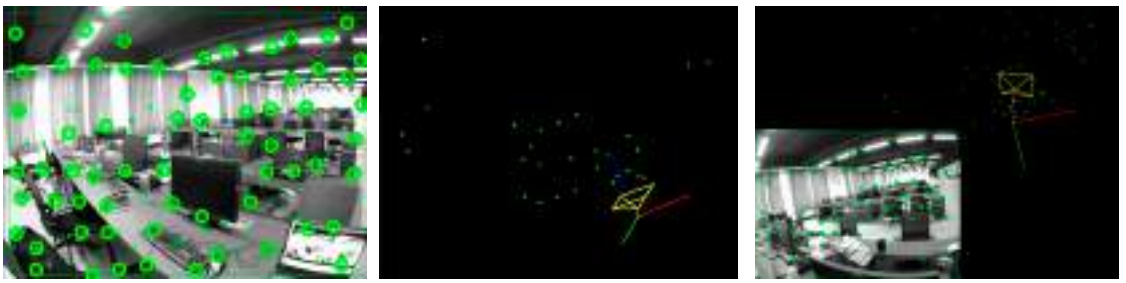
Figure 4. (**a**) FAST corner points distribution managed by the gridded map strategy; (**b**) Feature points tracked by the optical flow method.

Before tracking the monocular camera pose, the SLAM needs to be initialized first for tracking. The initialization task includes selecting two suitable monocular frames as the initial frames, estimating the relative pose of the initial frames, creating a local map, and creating a keyframe. According to our usage scenario, the initialization can be divided into the following three stages:

(1) Two initial frames that can be used as the first two frames are selected by feature point corresponding. When selecting the initial frames, it is necessary not only to ensure there are enough corresponding feature point pairs in the current and the last frames but also ensure there are as many co-vision feature points as possible on the two monocular cameras. The specific implementation algorithm is that when the BSV rig moves roughly in the opposite direction of the monocular camera's optical axis, the camera frame sequence is queried from the current frame in reverse order. When the motion of more than $N \in \mathbf{Z}$ corresponding feature point pairs is greater than $M \in \mathbf{Z}$ pixels, it is considered that the pixel moving distance is large enough. The frame that is queried in reverse order at this time is taken as a reference frame. The current frame and the reference frame are selected as the initial frames and are used for initialization. The M is related to image resolution, and there is a proportional relationship between them.

(2) The co-vision feature points set $\boldsymbol{P}_c$ of the current frame and the reference frame is taken out, and the corresponding feature point pairs set $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$ between the two frames are obtained. The co-vision feature points represented by circles in the current frame are shown in Figure 5a. Combining the Perspective-3-Points [37] with the random sample consensus (RANSAC) [38] algorithm, the essential matrix $\boldsymbol{E}$ between two frames is calculated. Singular Value Decomposition is then used to calculate the camera motion $\boldsymbol{T}_{Ini}$ between two frames. The reference frame is set to the world frame.

The camera pose estimation of the current frame is set to the initial frame pose. As shown in Figure 5b, the world frame is represented by three orthogonal axes (red, green, and blue axes), and the initial camera pose is represented by a yellow cube.



(**a**) (**b**) (**c**)

Figure 5. (**a**) Co-vision feature points in the current frame; (**b**) Initial frame pose of the monocular camera; (**c**) Inlier feature points and camera pose tracking.

(3) The initial feature point depth is obtained by triangulation, thereby obtaining an initial local map. The resulting local feature points map is represented in Figure 5b by a series of green dots.

After the initialization is completed, the current frame is used as the keyframe, and the initial frame pose and local map are optimized using BA optimization. Our algorithm will perform BA as soon as a new keyframe is obtained, which will be described in detail in Section 3.5. In the camera pose tracking process after obtaining the keyframes, the subsequent frames are corresponded with the local map to calculate the camera poses. The PnP method combined with the RANSAC algorithm is used to estimate camera pose. Here, the Levenberg–Marquardt algorithm [39] is used to minimize the distance between corresponding feature point pairs to calculate the pose of each frame.

However, the RANSAC uses only a few random points to determine the inliers, this method is susceptible to noise. To make the joint optimization more likely converge to the correct solution, the RANSAC solution is used as the initial value, and the camera poses are optimized using the pose-graph method to acquire a globally consistent camera pose estimation. In the pose graph optimization, nodes are used to represent the absolute poses to be optimized, expressed in $c_0, \ldots\ c_i, \ldots\ c_m$. $m \in \mathbf{Z}$ is the total number of nodes. An absolute pose c_i describes the transformation from the world frame to the camera frame $i \in \mathbf{Z}$. Edges are used to represent relative pose constraints between two pose nodes. A relative pose constraint c_{ij} between two camera frames i and $j \in \mathbf{Z}$ is defined as $c_{ij} := {c_i}^{-1} \cdot c_j$. Using ε as the set of all edges, the objective function in the optimization can be expressed as:

$$\min_{\boldsymbol{c}} \sum_{i,j \in \varepsilon} \Delta {c_{ij}}^T \boldsymbol{\Omega}_{ij} \Delta c_{ij}, \tag{1}$$

where $\boldsymbol{C}$ is the current set of all camera poses that need to be optimized; $\Delta c_{ij} = \ln(c_{ij} \cdot {c_j}^{-1} \cdot c_i)$ is the relative pose constraint error in the tangent space of SE(3); $\boldsymbol{\Omega}_{ij}$ is an information matrix, which is an inverse matrix of the covariance matrix of relative pose constraint. Rather than using appropriate marginalization to accurately estimate this uncertainty, it is better to approximate the information matrix roughly as a diagonal matrix in the way proposed in [40]:

$$\boldsymbol{\Omega}_{ij} = \omega_{ij} \begin{bmatrix} {\delta_t}^2 \boldsymbol{I}_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & {\delta_r}^2 \boldsymbol{I}_{3\times3} \end{bmatrix}, \tag{2}$$

where the δ_r and δ_t are the rotational component and the translational component of Δc_{ij}, respectively. It can be considered the camera pose constraints generated during the optimization are similar in accuracy [41]. Therefore, ω_{ij} is set to a constant $\omega_{ij} = 1$.

Figure 5c shows the results of inlier feature points and camera poses tracking. The brown line connected to the origins from the world frame to the current frame represents the tracked camera poses, and the yellow dot on the brown line represents the keyframe poses.

3.4. 3D Scene Point Map Construction

This step is used to construct the 3D feature point map of the natural environment which is used as a calibration area. The monocular camera poses and inlier feature points are input in this step, the output is 3D scene point map of the tracked inlier feature points in the keyframe. Note that the new keyframes are not determined due to the reduction of tracked feature points in consecutive frames. On the contrary, the new keyframe in the local map must have as many common views with other keyframes as possible to increase the number of 3D scene points required for calibration.

Figure 6 is a schematic diagram of our 3D feature point map construction. The Vector $\boldsymbol{C}$ represents the camera pose set of all co-view frames between the current keyframe and the previous keyframe; The vector $\boldsymbol{P}$ represents a 3D scene point map, which is represented by circles; The vector $\boldsymbol{p}$ represents the set of 2D feature points extracted and tracked in the image, and is represented by rectangles; The vector $\hat{\boldsymbol{p}}$ represents the set of 2D points projected from the local map onto the co-view frames, and is represented by triangles; The connection between the 3D feature points and the 2D feature points indicates the co-view relationship, and the yellow and green lines are used to distinguish them.

$$\begin{aligned} \boldsymbol{C} &= \{\boldsymbol{c}_0, \ldots \boldsymbol{c}_i, \ldots \boldsymbol{c}_m\}, \\ \boldsymbol{P} &= \{\boldsymbol{P}_0, \ldots \boldsymbol{P}_j, \ldots \boldsymbol{P}_n\}, \\ \boldsymbol{p} &= \{\boldsymbol{p}00, \ldots, \boldsymbol{p}0n, \ldots \boldsymbol{p}_{i0}, \ldots \boldsymbol{p}_{ij}, \ldots, \boldsymbol{p}_{in}, \ldots \boldsymbol{p}_{m0}, \ldots, \boldsymbol{p}_{mn}\}, \\ \hat{\boldsymbol{p}} &= \{\hat{\boldsymbol{p}}00, \ldots, \hat{\boldsymbol{p}}0n, \ldots \hat{\boldsymbol{p}}_{i0}, \ldots \hat{\boldsymbol{p}}_{ij}, \ldots, \hat{\boldsymbol{p}}_{in}, \ldots \hat{\boldsymbol{p}}_{m0}, \ldots, \hat{\boldsymbol{p}}_{mn}\}, \end{aligned} \tag{3}$$

where $\boldsymbol{c}_i$ is the absolute pose of co-view frame $i \in \mathbf{Z}$, m is the total number of co-view frames; $\boldsymbol{P}_j$ is the coordinate of 3D feature point $j \in \mathbf{Z}$ in the world frame, n is the total number of 3D feature points; $\boldsymbol{p}_{ij}$ is the observed image coordinate corresponding to 3D feature point j in the co-view frame i; $\hat{\boldsymbol{p}}_{ij}$ is the estimated image coordinate of 3D feature point j projected in co-view frame i.

$$\begin{cases} \dfrac{n\|\boldsymbol{p}_{ij}-\hat{\boldsymbol{p}}_{ij}\|^2}{\sum_{j=1}^{n}\|\boldsymbol{p}_{ij}-\hat{\boldsymbol{p}}_{ij}\|^2} \le T \quad \text{Yes} & P_j \text{ is a 3D feature point that can be retained;} \\ \dfrac{n\|\boldsymbol{p}_{ij}-\hat{\boldsymbol{p}}_{ij}\|^2}{\sum_{j=1}^{n}\|\boldsymbol{p}_{ij}-\hat{\boldsymbol{p}}_{ij}\|^2} > T \quad \text{No} & \boldsymbol{P}_j \text{ is a 3D feature point to be deleted.} \end{cases} \tag{4}$$

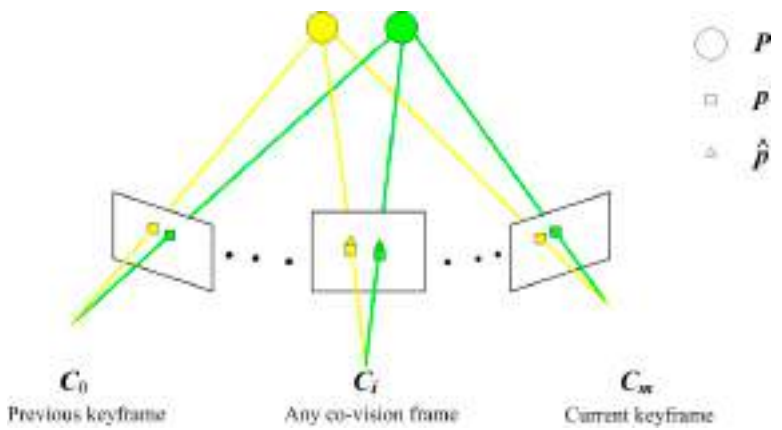


Figure 6. Schematic diagram of 3D feature points map construction.

When the SLAM creates a new keyframe, triangulation is used to build a local map. To reduce the inaccurate map caused by false tracking and noise, 3D feature points are projected onto any co-vision frame. When the reprojection error exceeds $T \in \mathbf{Z}$ times average error, the 3D feature point $\boldsymbol{P}_j$ is considered to be mismatched and deleted. This can improve the calibration precision to a certain extent.

3.5. Bundle Adjustment of Camera Pose and 3D Scene Point Map

The camera pose obtained in Section 3.3 and the 3D feature point map obtained in Section 3.4 are jointly optimized in this step for 3D scene point map refinement. The cost function of this optimization can be regarded as a nonlinear least-squares problem. In Formula (5), the cost function corresponds to the reprojection errors of the feature observations across all keyframes:

$$\min_{\boldsymbol{c}_i, \boldsymbol{P}_j} \sum_{i=1}^{m} \sum_{j=1}^{n} \rho\left(\left\| \boldsymbol{p}_{ij} - \pi\left(\boldsymbol{c}_i, \boldsymbol{P}_j\right) \right\|^2 \right), \tag{5}$$

where $\boldsymbol{c}_i$ is the absolute pose of keyframe $i \in \mathbf{Z}$, m is the total number of keyframes; $\boldsymbol{P}_j$ is the coordinate of 3D feature point $j \in \mathbf{Z}$ in the world frame, n is the total number of 3D feature points; $\boldsymbol{p}_{ij}$ is the observed image coordinate corresponding to 3D feature point j in the keyframe i; π is a standard camera projection function that predicts the image coordinates of the 3D feature point j given the camera's intrinsic parameters and the estimated camera pose $\boldsymbol{c}_i$; ρ is a robust kernel used for reducing the effects of outliers. The Huber kernel [42] is used here.

4. Extrinsic Parameter Calibration

This section describes how to calibrate the extrinsic parameters through the 3D scene point map created by the SLAM system.

4.1. Inter-Feature Correspondences between Different Cameras

In this step, the feature points in the overlapping fields of two monocular cameras are matched to obtain the 2D-2D feature correspondences.

Whenever the 3D scene point map is constructed by the left camera in Section 3.4, the optical flow method is used to perform feature correspondences on the left and right cameras. Compared with the use of descriptors for matching, the method based on optical flow tracking can obtain the feature correspondences faster. However, using this method will return some mismatched points.

To ensure the accuracy of the inter-feature correspondences between different cameras, the cross-correlation verification algorithm based on grayscale similarity is used to filter the feature correspondences. Assume the i-th ($i \in \mathbf{Z}$) 2D-2D feature correspondence in the left and right images is recorded as $\left(\boldsymbol{p}_i^l, \boldsymbol{p}_i^r\right)$. First, a search window block $K_{w \times w}$ with the size of w^2 ($w \in$ Z) pixels is defined with the seed point $\boldsymbol{p}_i^l$ as the center in the left image. In the right image, a search window block with the same size as $K_{w \times w}$ is also defined with the candidate point $\boldsymbol{p}_i^r$ as the center. Then, at the same starting point position, the grayscale sequences of the pixels in the window block are taken in the same order. These two sequences are recorded as $\boldsymbol{G}_l(i)$ and $\boldsymbol{G}_r(i)$, where $i = 0, 1, 2 \ldots w^2-1$. Based on these sequences, the cross-correlation coefficient in the two window blocks is calculated:

$$Corr(\boldsymbol{G}_l(i), \boldsymbol{G}_r(i)) = \frac{\sum_{i=0}^{w^2-1} \left[\left(\boldsymbol{G}_l(i) - m_l\right) * \left(\boldsymbol{G}_r(i) - m_r\right) \right]}{\sqrt{\sum_{i=0}^{w^2-1} \left(\boldsymbol{G}_l(i) - m_l\right)^2} \sqrt{\sum_{i=0}^{w^2-1} \left(\boldsymbol{G}_r(i) - m_r\right)^2}}, \tag{6}$$

where m_l and m_r are the arithmetic mean of the sequence $\boldsymbol{G}_l(i)$ and $\boldsymbol{G}_r(i)$, respectively.

The grayscale similarity of the feature correspondence is characterized by the cross-correlation coefficient. When the coefficient exceeds a threshold defined empirically, the gray sequence around the seed point and the candidate point should be highly cross-correlated. Now, the feature correspondence of the optical flow tracking is accurate. Otherwise, the feature correspondence is deleted as a mismatch. Figure 7 shows the result of feature correspondences between the left and right binocular cameras.

Figure 7. Feature correspondences between the left and right binocular cameras.

According to the 3D scene points associated with the 2D feature points in the left camera, the 3D-2D correspondences between the 3D scene points and the 2D feature points in the right camera are further obtained.

4.2. Confirming Correspondences and Estimating Initial Extrinsic Parameters

The goal of this step is to determine which feature correspondences between the two cameras are geometrically consistent and calculate an initial relative transformation between them.

The 3D-2D feature correspondences between the 3D scene points and the 2D feature points in the right camera have been found in the previous section. The RANSAC algorithm is still used to eliminate the outliers and the PnP method is used to solve the initial relative transformation:

$$s\boldsymbol{p}_i^r = \boldsymbol{K}_r \boldsymbol{T}_l^r \boldsymbol{P}_i, \tag{7}$$

where the $s \in \mathbf{R}^+$ is a local scale. $\boldsymbol{K}r$ is the intrinsic parameter matrix of the right camera. $\boldsymbol{P}_i$ is the 3D scene point in the world frame corresponding to the 2D feature point $\boldsymbol{p}_i^r$. $\boldsymbol{T}_l^r \in \mathbb{R}^{4\times4}$ is the relative transformation matrix denoted by the Lie group of euclidean transformation [43], and its specific form is as follows:

$$\boldsymbol{T}_l^r = \begin{bmatrix} \boldsymbol{R}_l^r & \boldsymbol{t}_l^r \\ 0_{1\times3} & 1 \end{bmatrix}, \tag{8}$$

where $\boldsymbol{R}_l^r \in \mathbb{R}^{3\times3}$ is the rotation transformation matrix between two cameras; $\boldsymbol{t}_l^r \in \mathbb{R}^3$ is the translation vector.

In the present study, three Euler angles [44] (represented r_x, r_y, and r_z, respectively) about the X-axis, Y-axis, and Z-axis are used to represent the rotation transformation matrix [26], as shown in Formula (9).

$$\boldsymbol{R}_l^r = \boldsymbol{R}_z \boldsymbol{R}_x \boldsymbol{R}_y, \tag{9}$$

where $\boldsymbol{R}_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(rx) & \sin(rx) \\ 0 & -\sin(rx) & \cos(rx) \end{bmatrix}$, $\boldsymbol{R}_y = \begin{bmatrix} \cos(ry) & 0 & -\sin(ry) \\ 0 & 1 & 0 \\ \sin(ry) & 0 & \cos(ry) \end{bmatrix}$, $\boldsymbol{R}_z = \begin{bmatrix} \cos(rz) & \sin(rz) & 0 \\ -\sin(rz) & \cos(rz) & 0 \\ 0 & 0 & 1 \end{bmatrix}$.

The reason why the Euler angles are used to further express the extrinsic parameters here is the Euler angles provide an intuitive way to describe rotation, which is convenient for evaluating the results obtained in the experimental part.

4.3. Joint Optimization

In this step, the monocular camera pose, the 3D scene point map, and the relative transformation between two monocular cameras are jointly optimized to obtain more accurate extrinsic parameters. The cost function is designed to achieve the following goals:

(1). The estimated camera pose of the monocular camera should be accurate.

(2). The 3D feature point map should be accurate.

(3). The relative transformation between the two monocular cameras should be accurate.

In Formula (10), the cost function contains a two-part weighted residual. The first partial residual is the same as Formula (5), corresponding to the reprojection errors of the feature observations for left monocular camera keyframes, while the second part corresponds to those for the right monocular camera frame. At this time, the right monocular camera frame is synchronized to the nearest keyframe in the left camera.

$$\min_{\boldsymbol{c}_i, \boldsymbol{P}_j, \boldsymbol{T}_l^r} \sum_{i=1}^{m} \sum_{j=1}^{n} w_1 \rho\left(\left\| \boldsymbol{p}_{ij} - \pi\left(\boldsymbol{c}_i, \boldsymbol{P}_j\right) \right\|^2 \right) + \sum_{k=1}^{l} w_2 \rho\left(\left\| \boldsymbol{p}_k - \pi\left(\boldsymbol{T}_l^r, \boldsymbol{P}_k\right) \right\|^2 \right), \tag{10}$$

where $\boldsymbol{P}_k$ is the coordinate of the 3D feature point $k \in \mathbf{Z}$ in the world frame, l is the total number of 3D feature points observed by the left and right cameras; $\boldsymbol{p}_k$ is the observed image coordinate in the right image frame corresponding to the 3D feature point $\boldsymbol{P}_k$; w_1 and w_2 are the weighting coefficients of the two residuals.

4.4. Verifying the Calibration Results and Choosing the Best

This step mainly provides the scale information based on the extrinsic parameters obtained in the previous step and uses this result to perform stereo rectification on the binocular images. Therefore, the optimal calibration result can be selected by the rectified error.

In the absence of the true coordinates of the marker as a reference, the encoder or IMU can be used to provide the scale information. But not all BSV rigs are equipped with the above sensors or sensor acquisition interfaces. To improve the practicability of the calibration algorithm as well as its application to general equipment, based on our BSV module, the baseline length between the left and right cameras can be conveniently measured. This parameter can provide the scale information of camera extrinsic parameters:

$$s = \frac{l_l^r}{\|\boldsymbol{t}_l^r\|}, \tag{11}$$

where l_l^r is the baseline length. According to the scale s, the translation vector $\boldsymbol{t}_l^r$ in a non-metric unit is converted into $s \cdot \boldsymbol{t}_l^r$ in a metric unit.

The method of stereo epipolar rectification [45] is used to select the optimal calibration result. When using extrinsic parameters to rectify two images, ideally the image coordinates of the feature points corresponding to the same 3D scene point should be on the same polar line. However, due to the imprecision of calibration, there may be some errors in the stereo rectification. According to this characteristic, the errors of stereo rectification are used as a criterion for judging whether the calibration is sufficiently accurate.

At present, the rectification algorithms are mainly divided into two categories, one is calibrated stereo rectification represented by Bouguet algorithm [45], and the other is non-calibrated stereo rectification represented by Hartley algorithm [46]. When the transformation matrix $\boldsymbol{T}_l^r$ between two cameras is obtained in Section 4.3, the rotation transformation matrix of epipolar rectification can be obtained using the Bouguet method, as shown in Formula (12):

$$\begin{aligned} \boldsymbol{R}_l &= \boldsymbol{R}_{rect} \boldsymbol{R}_l^{r\,\frac{1}{2}}, \\ \boldsymbol{R}_r &= \boldsymbol{R}_{rect} \boldsymbol{R}_l^{r\,-\frac{1}{2}}, \end{aligned} \tag{12}$$

where $\boldsymbol{R}_{rect}$ is the transformation matrix that makes the line of the camera's optical center parallel to the image plane. The method of constructing this matrix is completed by the translation vector $\boldsymbol{t}_l^r$ [45]. $\boldsymbol{R}_l$ and $\boldsymbol{R}_r$ are the rotation transformation matrices of the left and right cameras before and after rectification, respectively.

Suppose the 2D feature point coordinates on the left and right images corresponding to the same 3D scene point $\boldsymbol{P}_i = [XiYiZi]^T$ in the left camera frame are $\boldsymbol{p}_i^l = \left[u_i^l \; v_i^l\right]^T$ and $\boldsymbol{p}_i^r = \left[u_i^r \; v_i^r\right]^T$. According to the obtained extrinsic parameters, the 3D scene point in the right camera frame can be expressed as:

$$\boldsymbol{P}_i' = \left[X_i' \, Y_i' \, Z_i'\right]^T = \boldsymbol{R}_l^r \boldsymbol{P}_i + \boldsymbol{t}_l^r. \tag{13}$$

According to the pinhole camera model, the normalized coordinates of the feature points $\boldsymbol{p}_i^l$ can be obtained as:

$$\widetilde{\boldsymbol{p}_i^l} = \begin{bmatrix} \widetilde{u_i^l} \\ \widetilde{v_i^l} \end{bmatrix} = \begin{bmatrix} Xi/Yi \\ Yi/Zi \end{bmatrix}, \widetilde{\boldsymbol{p}_i^r} = \begin{bmatrix} \widetilde{u_i^r} \\ \widetilde{v_i^r} \end{bmatrix} = \begin{bmatrix} X_i'/Y_i' \\ Y_i'/Z_i' \end{bmatrix}. \tag{14}$$

The method of distortion rectification [47] is used to remap the positions of the feature points $\widetilde{\boldsymbol{p}_i^l}$ and $\widetilde{\boldsymbol{p}_i^r}$, and the undistorted expression can be written as:

$$\begin{aligned} U\left(\widetilde{\boldsymbol{p}_i^l}\right) &= \widetilde{\boldsymbol{p}_i^l} \cdot \left(1 + k_{l1}{r_l}^2 + k_{l2}{r_l}^4 + k_{l3}{r_l}^6\right) + \delta\left(\widetilde{\boldsymbol{p}_i^l}\right), \\ U\left(\widetilde{\boldsymbol{p}_i^r}\right) &= \widetilde{\boldsymbol{p}_i^r} \cdot \left(1 + k_{r1}{r_r}^2 + k_{r2}{r_r}^4 + k_{r3}{r_r}^6\right) + \delta\left(\widetilde{\boldsymbol{p}_i^r}\right), \end{aligned} \tag{15}$$

where k_{l1}, k_{l2}, k_{l3},p_{l1}, p_{l2} are the distortion coefficients of the left camera; k_{r1}, k_{r2},k_{r3}, p_{r1}, p_{r2} are the distortion coefficients of the right camera; ${r_l}^2 = \widetilde{u_i^l}^2 + \widetilde{v_i^l}^2$,${r_r}^2 = \widetilde{u_i^r}^2 + \widetilde{v_i^r}^2$; $\delta\left(\widetilde{\boldsymbol{p}_i^l}\right)$ and $\delta\left(\widetilde{\boldsymbol{p}_i^r}\right)$ both refer to the tangential distortion vector and are expressed as:

$$\delta\left(\widetilde{\boldsymbol{p}_i^l}\right) = \begin{bmatrix} 2p_{l1}\widetilde{u_i^l}\widetilde{v_i^l} + p_{l2}\left({r_l}^2 + 2\widetilde{u_i^l}^2\right) \\ 2p_{l2}\widetilde{u_i^l}\widetilde{v_i^l} + p_{l1}\left({r_l}^2 + 2\widetilde{v_i^l}^2\right) \end{bmatrix}, \; \delta\left(\widetilde{\boldsymbol{p}_i^r}\right) = \begin{bmatrix} 2p_{r1}\widetilde{u_i^r}\widetilde{v_i^r} + p_{r2}\left({r_r}^2 + 2\widetilde{u_i^r}^2\right) \\ 2p_{r2}\widetilde{u_i^r}\widetilde{v_i^r} + p_{r1}\left({r_r}^2 + 2\widetilde{v_i^r}^2\right) \end{bmatrix}. \tag{16}$$

The stereo epipolar rectification of $\widetilde{\boldsymbol{p}_i^r}$ and $\widetilde{\boldsymbol{p}_i^r}$ can be expressed as:

$$\begin{aligned} \boldsymbol{C}\left(\widetilde{\boldsymbol{p}_i^l}\right) &= \boldsymbol{R}lU\left(\widetilde{\boldsymbol{p}_i^l}\right), \\ \boldsymbol{C}\left(\widetilde{\boldsymbol{p}_i^r}\right) &= \boldsymbol{R}rU\left(\widetilde{\boldsymbol{p}_i^r}\right). \end{aligned} \tag{17}$$

Through the above rectification, the binocular images with coplanar line alignment can be obtained. The rectification error can be expressed as:

$$RectErr(\boldsymbol{R}_l^r, \boldsymbol{t}_l^r) = \frac{1}{\mathrm{N}} \sum_{i=1}^{\mathrm{N}} \left| C\left(v_i^l\right) - C\left(v_i^r\right) \right|. \tag{18}$$

Next, the stereo rectification error is used as the standard for verification of calibration results and to select the optimal calibration result.

$$\begin{cases} RectErr(\boldsymbol{R}_l^r, \boldsymbol{t}_l^r) \le T \text{ YES} & \left(\boldsymbol{R}_l^r, \boldsymbol{t}_l^r\right) \text{meets the requirements,} \\ RectErr(\boldsymbol{R}_l^r, \boldsymbol{t}_l^r) > T \text{ NO} & \left(\boldsymbol{R}_l^r, \boldsymbol{t}_l^r\right) \text{does not meet the requirements,} \end{cases} \tag{19}$$

where T is a threshold.

5. Experimental Verification

This section describes various experiments to verify the feasibility, repeatability, and efficiency of the calibration parameter estimation by our self-calibration method. The first experiment was to evaluate calibration feasibility. In this experiment, Zhang's checkerboard method was used to provide ground truth data. Then the results from our proposed self-calibration method were compared against those from Zhang's method. The second experiment focused on verifying the calibration repeatability of the extrinsic parameters and aimed at comparing experiments under different texture environments. The statistics of the calibration efficiency were also made during the experiment.

5.1. Experimental Setup

To verify the proposed self-calibration method, a BSV module was constructed as shown in Figure 8a. The two cameras can be rotated, and their baseline length can be adjusted arbitrarily. The image resolution of the left and right cameras is 640 pixels × 472 pixels and the captured images are transmitted to the computer through a network cable. Table 1 lists the intrinsic parameters of the left and right cameras, including focal length, principal point, and distortion coefficients obtained by Zhang's method.

(**a**) (**b**)

Figure 8. Experiment platform: (**a**) The BSV rig; (**b**) Mobile robot platform.

Table 1. Camera intrinsic parameters.

Camera	f_u/Pixels	f_v/Pixels	u_0/Pixels	v_0/Pixels	k_1, k_2, k_3, p_1, p_2
Left	561.86	556.47	318.26	217.06	−0.53, 0.28, 0.003, −0.002, 0.0002
Right	617.91	611.58	312.89	242.26	−0.56, 0.31, 0.004, −0.001, 0.0004

Initially, to verify the performance of our calibration algorithm, we performed experiments on a computer, with an Intel Core i7-8750 processor, 8 GB of RAM. To verify the actual application effect of our algorithm in the later period, a mobile robot was used as our experimental platform as shown in Figure 8b. A set of calibration software based on Android system was developed for the convenience of testing. The mobile robot was equipped with a CortexTM-A7 processor, 2 GB of RAM. This processor is currently used in most cars.

5.2. Experimental 1-Calibration Feasibility

It is well known that calibration algorithms are difficult to evaluate because it is hard to obtain ground truths of the estimated calibration parameters. Here, direct and indirect methods were used to evaluate the feasibility of the proposed algorithm. In the direct method experiment, the calibration result of Zhang's method was used as the ground truth data. Zhang's checkerboard method is widely used in the calibration of the BSV rig with the advantages of low cost, convenience, and high feasibility [26]. Therefore, the calibration results of the proposed method were compared with those produced by Zhang's method.

Fifteen groups of experiments were performed by adjusting the baseline position and rotation angle between the two cameras of the BSV rig. The three Euler angles transformed from the rotation matrix obtained by the proposed and Zhang's methods in each experiment were shown in Figure 9, namely r_x, r_y, and r_z. By comparison, it is found that the trends of r_x, r_y, and r_z obtained by two methods were consistent. In Figure 10, the results obtained by Zhang's method were used as references, and the absolute error of the angles obtained by our proposed method concerning the references was calculated. As shown in Figure 10, the average absolute error of the three angles did not exceed 0.5°.

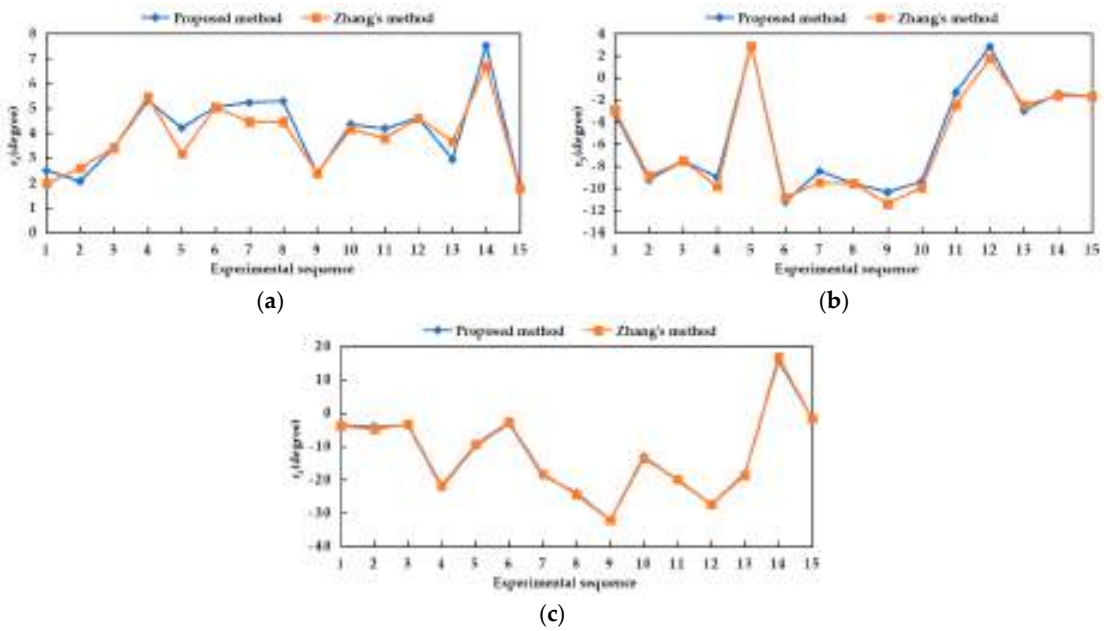


Figure 9. Euler angles between two cameras obtained by our proposed and Zhang's methods: (**a**) r_x obtained by two methods; (**b**) r_y obtained by two methods; (**c**) r_z obtained by two methods.

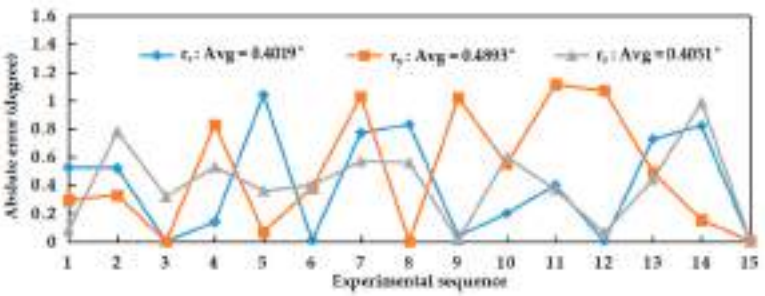


Figure 10. Absolute error of the Euler angles.

The results of the translation vector obtained by two methods were shown in Figure 11, namely t_x, t_y and t_z. By comparison, it was also found that the trends of t_x, t_y and t_z obtained by two methods were consistent. The absolute error of the translation vector relative to the reference values is shown in Figure 12. The average absolute error of the translation vector did not exceed 2 mm. Among them, the minimum error of t_y was 1.15 mm.

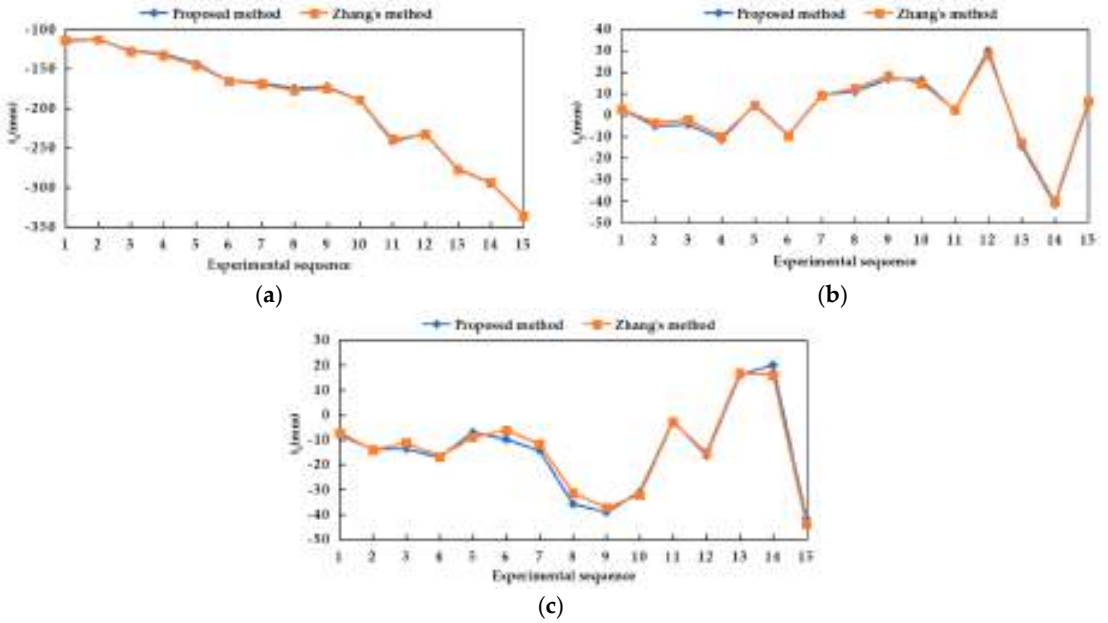


Figure 11. Translation vector between two cameras obtained by our proposed and Zhang's methods: (**a**) t_x obtained by two methods; (**b**) t_y obtained by two methods; (**c**) t_z obtained by two methods.

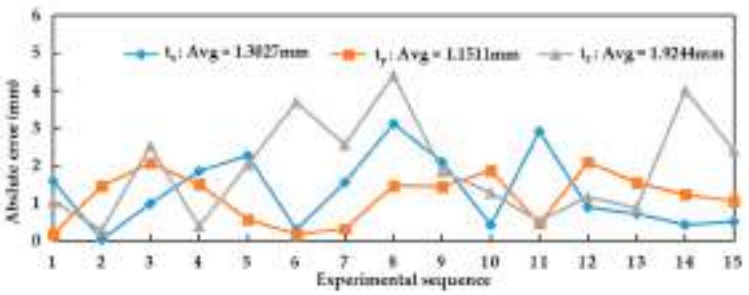


Figure 12. Absolute error of the translation vector.

At the same time, the RMSEs of our and Zhang's calibration methods were recorded in 15 groups of experiments to evaluate the precision of re-projection. The average RMSE was 0.29 pixel when using Zhang's method, and 0.51 pixel when using our method. Although Zhang's method showed higher precision, it can be seen the results obtained by our proposed and Zhang's method were close, which can prove that our method was reliable.

In the indirect method experiment, the depth image obtained by calibration parameters was used for qualitative evaluation. If the calibration parameters are accurate, the result of the stereo epipolar rectification should be ideal and the depth map formed by the BSV rig will also be dense and accurate. Based on this assumption, the results of the first experimental calibration were selected to perform depth calculations on the original images. Figure 13 showed the depth maps calculated using the parameters estimated by Zhang's and our method, respectively.

It can be seen from the experimental results that the parameters we estimated would generate a depth map with a uniform density similar to Zhang's method. This result proved that our estimated parameters have high precision, which can be attributed to the fact: 1. our pipeline filtered and optimized a large number of the tracked feature points; 2. our pipeline verified the results after given the scale parameter and selected the optimal calibration result.

(**a**) (**b**) (**c**)

Figure 13. (**a**) The original image; (**b**) The depth map calculated using the parameters estimated by Zhang's method; (**c**) The right depth map calculated using our estimated parameters.

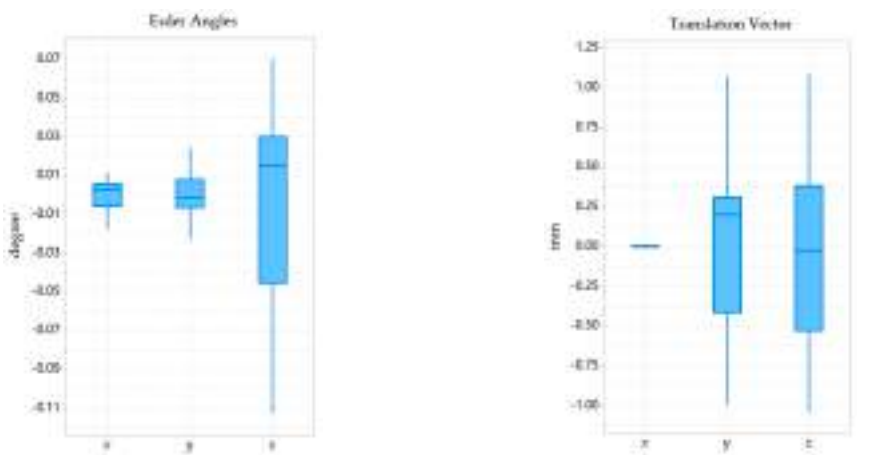
5.3. Experimental 2-Calibration Repeatability and Efficiency

This experiment focused on verifying the calibration repeatability of the extrinsic parameters and aimed at comparing experiments under different texture environments. The statistics of the calibration efficiency were also made during this experiment.

The experiment was mainly divided into two groups. The two sets of experiments were performed in the same way under the same camera configuration. In the first set of experiments, a mobile robot was first used to perform 20 experiments under general texture conditions, with approximately 250 feature points extracted. In the second set of experiments, 20 experiments were also performed under weak texture environments, with approximately 80 feature points extracted. The mean and standard deviation of the Euler angles and translation vectors were listed in Table 2. Besides, the distributions of Euler angles and translation vector obtained from the two sets of experiments were recorded in the form of boxplots in Figures 14 and 15. For comparison, the mean was subtracted from the boxplot.

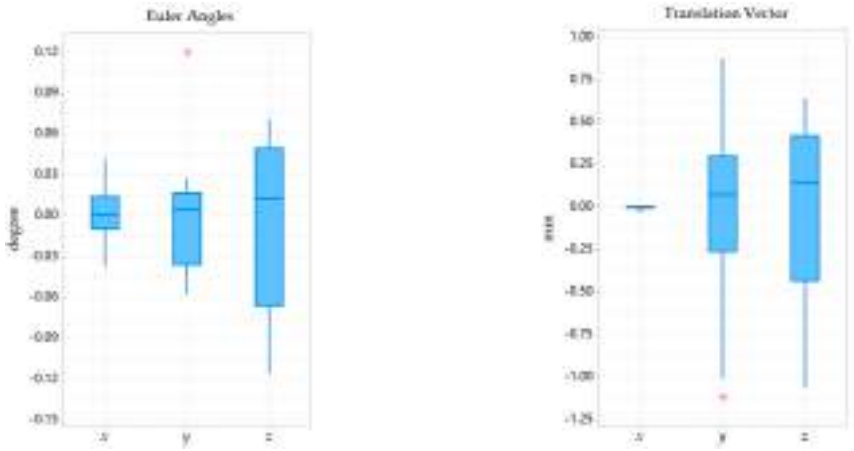
Table 2. The mean and standard deviation of Euler angles and translation vectors.

First Set of Experiments				Second Set of Experiments			
Euler Angles (degree)		**Translation Vector (mm)**		**Euler Angles (degree)**		**Translation Vector (mm)**	
$\begin{bmatrix} 5.2788 \\ 2.0636 \\ 1.0381 \end{bmatrix}$	$\pm \begin{bmatrix} 0.0080 \\ 0.0111 \\ 0.0456 \end{bmatrix}$	$\begin{bmatrix} -99.9931 \\ -0.5070 \\ 0.7027 \end{bmatrix}$	$\pm \begin{bmatrix} 0.0039 \\ 0.5850 \\ 0.5618 \end{bmatrix}$	$\begin{bmatrix} 5.2782 \\ 2.1244 \\ 0.9770 \end{bmatrix}$	$\pm \begin{bmatrix} 0.0196 \\ 0.0418 \\ 0.0579 \end{bmatrix}$	$\begin{bmatrix} -99.9849 \\ -0.6226 \\ 0.8459 \end{bmatrix}$	$\pm \begin{bmatrix} 0.0050 \\ 0.5448 \\ 0.5106 \end{bmatrix}$



(**a**) (**b**)

Figure 14. Box plot of the Euler angles and translation vector minus the mean in the first set of experiments: (**a**) Distribution of Euler angles; (**b**) Distribution of translation vector.



(**a**) (**b**)

Figure 15. Box plot of the Euler angles and translation vector minus the mean in the second set of experiments: (**a**) Distribution of Euler angles; (**b**) Distribution of translation vector.

By observing Table 2, it can be found that the largest standard deviation in Euler angles appeared at r_z in the second set of experiments, which is 0.0579°. The largest standard deviation in the translation vector appeared in t_y of the first set of experiments, which was 0.5850 mm. Therefore, the standard deviation of each parameter in the two sets of experiments was small.

Looking further at Figures 14 and 15, in the first set of experiments, the parameter r_z, which was the most widely distributed parameter in the Euler angles, had a normal interval length of approximately 0.18° and the interquartile range of approximately 0.075°. The most widely distributed parameter in the translation vector was t_z, whose normal interval length was about 2.125 mm and the interquartile range was about 0.9 mm. In the second set of experiments, the parameter r_z, which was the most widely distributed parameter in Euler angles, had a normal interval length of less than 0.2° and the interquartile range about 0.1°. The most widely distributed parameter in the translation vector was t_z, whose normal interval length was less than 2 mm and the interquartile range was about 0.6 mm. Therefore, the distribution of each parameter in the two sets of experiments was concentrated, which showed that our calibration algorithm was repeatable.

At the same time, by comparing the mean and data distribution in the two texture environments, it was found that no major changes occurred, and repeatable results could be obtained. This was mainly due to the re-projection in 3D scene point map construction, all of which were in the range of 0.5–1.0-pixel RMSE. However, through statistical analysis of the calibration time, they were found to be different. The running speed of the mobile robot in both experiments was set to 0.3 m/s. The calibration average time of the first group of experiments was about 7.5 s, while that of the second group was about 2.3 min. Therefore, although the calibration algorithm still maintained certain repeatability under weak texture conditions, it took more time to finish calibration. This was mainly due to our verification and selection of the calibration results because it would take more time to select the required calibration results among the few feature points. However, in practical applications in general scenes, such as outdoor street scenes, the conditions with weak textures are rarely encountered. At the same time, it can be seen that under the general texture scene and the normal driving speed of the robot, the calibration time can be generally maintained within 10 s, which has a higher calibration efficiency. Experimental demonstrations can be found in the supplementary materials.

6. Conclusions and Future Work

In this paper, a novel SLAM-based self-calibration method for the BSV rig was proposed. The proposed method was enabled to estimate the extrinsic parameters of the BSV rig without auxiliary equipment and special calibration markers. Further, the calibration process can be fully automated in

real-time during the normal driving of the platform, without complex requirements for rig movement and prior information about the environment.

Field experiments were performed to evaluate the feasibility, repeatability of the calibration parameter estimation by our self-calibration method. In terms of feasibility experiments, it can be seen from the experimental data that the results obtained by our proposed and Zhang's calibration method were close, which could prove that our method was reliable. In the repeatability experiment, 20 experiments performed under general and weak texture environments respectively showed that our calibration method can both obtain relatively compact distribution results. Under the general texture scene and the normal driving speed of the robot, the calibration time can be generally maintained within 10 s, which was more efficient. In short, the experiment results showed that the precision and efficiency of the proposed SLAM-based self-calibration method have reached a relatively high level. This calibration method can be used for large-scale practical applications on arbitrary BSV rigs, such as a car or mobile robot.

However, the calibration method still has shortcomings in the following cases. Firstly, incorrect feature correspondences in a dynamic environment could cause an erroneous 3D scene point map and further lead to unreliable calibration results. Secondly, the baseline length obtained by measurement equipment is adopted as the scale information of the method currently. Although in a more accurate situation, the calibration results can meet the requirements of the advanced driving assistant system. However, calibration precision is still affected by external human factors. Therefore, a suitable method that can automatically calculate the baseline will make the calibration more convenient. In the future, the solution in the dynamic environment and other scale information is the next work we need to focus on.

Supplementary Materials: The following are available online at http://www.mdpi.com/1424-8220/20/3/621/s1, The experimental demonstration video is available online at https://youtu.be/__wgcoX3jKw.

Author Contributions: The work presented here was carried out in collaboration among all authors. H.Y. and B.H. conceived the article, conducted the experiments, and wrote the paper. K.W. and Y.W. helped design the algorithms and software. Z.M., M.Z. and J.G. contributed to data collection and analysis. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National key Research and Development Program of China (Grant No. 2018YFB1308000), the Special Program for Science and Technology of National Regional Innovation Center of Weihai City (Grant No. 2017QYCX10), the National Natural Science Foundation of China (Grant No. 51905119) and the Fundamental Research Funds for Central Universities (Grant No. HIT.NSRIF.2020090).

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

References

1. Li, S. Binocular spherical stereo. *IEEE Trans. Intell. Transp. Syst.* **2008**, *9*, 589–600.
2. Tu, J.; Zhang, L. Effective data-driven calibration for a galvanometric laser scanning system using binocular stereo vision. *Sensors* **2018**, *18*, 197. [CrossRef] [PubMed]
3. Heng, L.; Bürki, M.; Lee, G.H.; Furgale, P.; Siegwart, R.; Pollefeys, M. Infrastructure-based calibration of a multi-camera rig. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China, 31 May–7 June 2014; pp. 4912–4919.
4. Gil, G.; Savino, G.; Piantini, S.; Pierini, M. Motorcycles that see: Multifocal stereo vision sensor for advanced safety systems in tilting vehicles. *Sensors* **2018**, *18*, 295. [CrossRef] [PubMed]
5. Chai, X.; Gao, F.; Hu, Y. Mirror binocular calibration method based on sole principal point. *Opt. Eng.* **2019**, *58*, 094109. [CrossRef]
6. Semeniuta, O. Analysis of camera calibration with respect to measurement accuracy. *Procedia Cirp* **2016**, *41*, 765–770. [CrossRef]
7. Abdel-Aziz, Y.; Karara, H.; Hauck, M. Direct linear transformation from comparator coordinates into object space coordinates in close-range photogrammetry. *Photogramm. Eng. Remote Sens.* **2015**, *81*, 103–107. [CrossRef]

8. Rohac, J.; Sipos, M.; Simanek, J. Calibration of low-cost triaxial inertial sensors. *IEEE Instrum. Meas. Mag.* **2015**, *18*, 32–38. [CrossRef]
9. Wu, D.; Chen, T.; Li, A. A high precision approach to calibrate a structured light vision sensor in a robot-based three-dimensional measurement system. *Sensors* **2016**, *16*, 1388. [CrossRef]
10. Deng, L.; Lu, G.; Shao, Y.; Fei, M.; Hu, H. A novel camera calibration technique based on differential evolution particle swarm optimization algorithm. *Neurocomputing* **2016**, *174*, 456–465. [CrossRef]
11. Batista, J.; Araújo, H.; de Almeida, A.T. Iterative multistep explicit camera calibration. *IEEE Trans. Rob. Autom.* **1999**, *15*, 897–917. [CrossRef]
12. Zhuang, H.; Wu, W.-C. Camera calibration with a near-parallel (ill-conditioned) calibration board configuration. *IEEE Trans. Rob. Autom.* **1996**, *12*, 918–921. [CrossRef]
13. Wang, Y.; Liu, L.; Cai, B.; Wang, K.; Chen, X.; Wang, Y.; Tao, B. Stereo calibration with absolute phase target. *Opt. Express* **2019**, *27*, 22254–22267. [CrossRef] [PubMed]
14. Zhang, Z. A flexible new technique for camera calibration. *IEEE Trans. Pattern Anal. Mach. Intell.* **2000**, *22*, 1330–1334. [CrossRef]
15. Yu, A.C.; Peng, Q. Robust recognition of checkerboard pattern for camera calibration. *Opt. Eng.* **2006**, *45*, 1173–1183. [CrossRef]
16. Chen, Q.; Wu, H.; Wada, T. Camera calibration with two arbitrary coplanar circles. In Proceedings of the European Conference on Computer Vision, Berlin, Germany, 11–14 May 2004; pp. 521–532.
17. Kumar, R.K.; Ilie, A.; Frahm, J.M.; Pollefeys, M. Simple calibration of non-overlapping cameras with a mirror. In Proceedings of the Proc IEEE Conference on Computer Vision & Pattern Recognition, Anchorage, AK, USA, 23–28 June 2008; pp. 1–7.
18. Su, P.C.; Shen, J.; Xu, W.; Cheung, S.S.; Luo, Y. A Fast and Robust Extrinsic Calibration for RGB-D Camera Networks. *Sensors* **2018**, *18*, 235. [CrossRef]
19. Wei, Z.; Zhao, K. Structural Parameters Calibration for Binocular Stereo Vision Sensors Using a Double-Sphere Target. *Sensors* **2016**, *16*, 1074. [CrossRef]
20. Luong, Q.-T.; Faugeras, O.D. Self-calibration of a stereo rig from unknown camera motions and point correspondences. In *Calibration and Orientation of Cameras in Computer Vision*; Springer: Berlin, Germany, 2001; pp. 195–229.
21. Cheung, W.; Hamarneh, G. N-SIFT: N-dimensional Scale Invariant Feature Transform. *IEEE Trans. Image Process.* **2009**, *18*, 2012–2021. [CrossRef]
22. Zhang, Z.; Tang, Q. Camera self-calibration based on multiple view images. In Proceedings of the Nicograph International (NicoInt), Hangzhou, China, 6–8 July 2016; pp. 88–91.
23. Wang, H.; Mou, W.; Mou, X.; Yuan, S.; Ulun, S.; Yang, S.; Shin, B.-S. An automatic self-calibration approach for wide baseline stereo cameras using sea surface images. *Unmanned Systs.* **2015**, *3*, 277–290. [CrossRef]
24. Boudine, B.; Kramm, S.; El Akkad, N.; Bensrhair, A.; Saaidi, A.; Satori, K. A flexible technique based on fundamental matrix for camera self-calibration with variable intrinsic parameters from two views. J. *Visual Commun. Image Represent.* **2016**, *39*, 40–50. [CrossRef]
25. Ji, S.; Qin, Z.; Shan, J.; Lu, M. Panoramic SLAM from a multiple fisheye camera rig. *ISPRS J. Photogramm. Remote Sens.* **2020**, *159*, 169–183. [CrossRef]
26. Wang, Y.; Wang, X.; Wan, Z.; Zhang, J. A Method for Extrinsic Parameter Calibration of Rotating Binocular Stereo Vision Using a Single Feature Point. *Sensors* **2018**, *18*, 3666. [CrossRef] [PubMed]
27. Carrera, G.; Angeli, A.; Davison, A.J. SLAM-based automatic extrinsic calibration of a multi-camera rig. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Shanghai, China, 9–13 May 2011; pp. 2652–2659.
28. Bay, H.; Ess, A.; Tuytelaars, T.; Van Gool, L. Speeded-up robust features (SURF). *Comput. Vision Image Understanding* **2008**, *110*, 346–359. [CrossRef]
29. Heng, L.; Lee, G.H.; Pollefeys, M. Self-calibration and visual SLAM with a multi-camera system on a micro aerial vehicle. *Auton. Rob.* **2015**, *39*, 259–277. [CrossRef]
30. Heng, L.; Furgale, P.; Pollefeys, M. Leveraging Image-based Localization for Infrastructure-based Calibration of a Multi-camera Rig. *J. Field Rob.* **2015**, *32*, 775–802. [CrossRef]
31. Garrigues, M.; Manzanera, A.; Bernard, T.M. Video Extruder: A semi-dense point tracker for extracting beams of trajectories in real time. *J. Real-Time Image Proc.* **2016**, *11*, 785–798. [CrossRef]

32. Derpanis, K.G. The harris corner detector. Available online: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.482.1724&rep=rep1&type=pdf (accessed on 22 January 2020).
33. Antonakos, E.; Alabortimedina, J.; Tzimiropoulos, G.; Zafeiriou, S.P. Feature-based Lucas-Kanade and active appearance models. *IEEE Trans. Image Process.* **2015**, *24*, 2617–2632. [CrossRef]
34. Krig, S. Interest point detector and feature descriptor survey. In *Computer vision metrics*; Springer: Berlin, Germany, 2016; pp. 187–246.
35. Bailo, O.; Rameau, F.; Joo, K.; Park, J.; Bogdan, O.; Kweon, I.S. Efficient adaptive non-maximal suppression algorithms for homogeneous spatial keypoint distribution. *Pattern Recognit. Lett.* **2018**, *106*, 53–60. [CrossRef]
36. Bian, J.; Lin, W.-Y.; Matsushita, Y.; Yeung, S.-K.; Nguyen, T.-D.; Cheng, M.-M. Gms: Grid-based motion statistics for fast, ultra-robust feature correspondence. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 4181–4190.
37. Gao, X.-S.; Hou, X.-R.; Tang, J.; Cheng, H.-F. Complete solution classification for the perspective-three-point problem. *IEEE Trans. Pattern Anal. Mach. Intell.* **2003**, *25*, 930–943.
38. Chum, O.; Matas, J.; Kittler, J. Locally optimized RANSAC. In Proceedings of the Joint Pattern Recognition Symposium, Berlin, Germany, 10–12 September 2003; pp. 236–243.
39. Shawash, J.; Selviah, D.R. Real-Time Nonlinear Parameter Estimation Using the Levenberg–Marquardt Algorithm on Field Programmable Gate Arrays. *IEEE Trans. Ind. Electron.* **2013**, *60*, 170–176. [CrossRef]
40. Strasdat, H.; Davison, A.J.; Montiel, J.M.M.; Konolige, K. Double window optimisation for constant time visual SLAM. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Barcelona, Spain, 6–13 November 2011; pp. 2352–2359.
41. Yang, S.; Scherer, S.A.; Yi, X.; Zell, A. Multi-camera visual SLAM for autonomous navigation of micro aerial vehicles. *Rob. Autom. Syst.* **2017**, *93*, 116–134. [CrossRef]
42. Härdle, W.; Gasser, T. On robust kernel estimation of derivatives of regression functions. *Scand. J. Stat.* **1985**, 233–240.
43. Strasdat, H.; Montiel, J.; Davison, A.J. *Scale drift-aware large scale monocular SLAM. Robotics: Science and Systems VI*; The Mit Press: London, England, 2010.
44. Wang, Y.; Rajamani, R. Direction cosine matrix estimation with an inertial measurement unit. *Mech. Syst. Sig. Process.* **2018**, *109*, 268–284. [CrossRef]
45. Fusiello, A.; Trucco, E.; Verri, A. A compact algorithm for rectification of stereo pairs. *Mach. Vision Appl.* **2000**, *12*, 16–22. [CrossRef]
46. Hartley, R.; Gupta, R. Computing matched-epipolar projections. In Proceedings of the IEEE Computer Society Conference on Computer Vision & Pattern Recognition (CVPR), New York, NY, USA, 15–17 June 1993; pp. 549–555.
47. Chakraborty, D.P. Image intensifier distortion correction. *Med. Phys.* **1987**, *14*, 249–252. [CrossRef]