This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSEN.2019.2958104, IEEE Sensors Journal

Accurate and Flexible Calibration Method for a Class of Visual Sensor Networks

Heng Deng, Kun Yang, Quan Quan and Kai-Yuan Cai

***Abstract*—Accurate and flexible calibration is a prerequisite for visual sensor networks to retrieve metric information from image data. This paper presents the design and implementation of an accurate and flexible calibration method for a class of visual sensor networks intended for 3D measurement and tracking in large volumes. The proposed method employs a generic camera model that is applicable for wide-angle lens cameras as well as for conventional cameras. It does not require all the employed cameras to share a common field of view, and only pairwise overlap is needed. In the calibration process, the poses between stereo cameras are first initialized using essential matrix decomposition and then optimized by the Levenberg-Marquardt algorithm. A weighted vision graph is proposed to select optimal transformation paths among cameras by using Dijkstra's shortest path algorithms for multi-camera calibration. Then, the global coordinates are constructed using a four-marker calibration triangle. Finally, a Unity3D-based virtual platform, in which the total number and configurations of cameras, as well as the environment scene, can be arbitrarily edited, is designed to test the proposed calibration algorithms. Extensive experiments based on synthetic and real data are performed to demonstrate the effectiveness of the proposed multi-camera calibration method. Experimental results show that the multi-camera calibration method is accurate and easy-to-implement in the presence of noise.**

***Index Terms*—Visual sensor networks, Multi-camera calibration, 1D calibration pattern, Vision graph, Bundle adjustment**

I. Introduction

Nowadays, Visual Sensor Networks (VSNs) have emerged as an important class of multi-camera networks for large-scale applications that can detect, monitor, and track events of interest areas with a number of potential applications, ranging from security to monitoring [1], [2], [3]. As pointed out in [4], current optical motion capture systems are a successful example of how useful the multi-camera system can be. In typical applications in these systems, some markers can be detected by cameras, and their positions are easily reconstructed by matching and triangulation algorithms.

Camera calibration is a necessary task, and a key part of the VSNs intended for applications involving accurate metric measurements, the major goal of which is to revive the positions and orientations of all the cameras in the network. It is essential for camera calibration to choose an appropriate camera model. Perspective camera model, i.e., the pinhole camera model, is the most common model, and it is applicable for most conventional cameras with a small Field Of View (FOV) and little distortion. A highly flexible technique to calibrate the perspective cameras has been proposed by Zhang [5] and widely used in a variety of applications [6], [7]. Nevertheless, some applications such as visual surveillance and monitoring require cameras with fish-eye lenses or wide FOV, in which perspective camera models are not suitable. For such a purpose, Kannala has proposed a generic camera model which is suitable and easily expandable for conventional, wide-angle and fish-eye lenses [8], and the generic camera model is employed and evaluated by our previous works [9], [10] as well as in this paper.

Usually, during the calibration process, the camera captures images from an object with known dimensions and shape (also considered as a calibration pattern). For 3D or 2D calibration [11], [12], it may require an expensive calibration apparatus and an elaborate setup, and the calibration objects might not be simultaneously observed by all the cameras due to self-occlusion [13]. Therefore, like the method employed in common optical motion capture systems, a 1D calibration pattern is used in this paper. 1D calibration was firstly proposed by Zhang [14] by using a 1D object consisting of three or more collinear points on a straight line. The main advantage of 1D calibration is that there is no self-occlusion problem. Thus, the 1D calibration object is especially suitable for a multi-camera system. The work of Wang et al. [15] was the first to propose a multi-camera calibration algorithm based on a 1D pattern that moves freely and without prior knowledge of the parameters of any of the cameras. Franca et al. [16] took the results of a normalized linear algorithm as initial values and performed nonlinear optimization with bundle adjustment. Authors in [17] proposed a new 1D calibration without imposing any restrictions on the movement of the pattern and without any prior information about the cameras or motion, and the method was shown to have a good accuracy suitable for practical applications.

Furthermore, most of the calibration methods are designed to calibrate the cameras with a common FOV, which are not applicable to solving the case of non-overlapping. Thus, accurate global calibration of cameras with a non-overlapping FOV is a very challenging task. Kurillo et al. [18] combined the idea of vision graphs for wide-area camera networks with small working volume overlap and calibration methods using virtual calibration object. Xia et al. [19] reviewed a variety of global calibration of non-overlapping multi-camera methods based on different types of techniques such as large-range measuring devices, large-scale calibration targets, optical mirrors, motion model, laser projection, visual measuring instruments, etc.

The authors are with School of Automation Science and Electrical Engineering, Beihang university, Beijing 100191, China. (e-mail: dengheng@buaa.edu.cn; yangkun_buaa@buaa.edu.cn; qq_buaa@buaa.edu.cn; kycai@buaa.edu.cn).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSEN.2019.2958104, IEEE Sensors Journal

However, most global calibration methods for non-overlapping multi-camera systems have a small effective range with a small number of cameras employed, which shortly limits the applications for a large-scale space [15], [20], [21], [22].

Our previous work [9] also employed a 1D calibration object, i.e., a freely-move wand with three collinear LEDs under general motions. However, the algorithm is demonstrated with at most three cameras. Besides, the point extraction method does not exhibit satisfactory behavior for LEDs, and the calibration results rely heavily on the moving of the wand. Therefore, this paper proposes an accurate and flexible approach to the geometric calibration of a VSN by using a 1D wand under general motions. The proposed method does not require all the cameras to share a common FOV, and only pairwise overlap is needed. The proposed calibration method is demonstrated and applied in real multi-camera based testbed for 3D tracking and control of UAVs [10]. As noted in [10], it is hopeful for the testbed to extend to a large-scale VSN in which a large number of cameras are employed. Based on our previous works, this paper aims to clarify the detailed calibration method for a class of VSNs with extensive simulation and real experimental results.

The contributions of this paper are: i) a flexible approach to calibrate multiple cameras of a large-scale visual sensor network by using a 1D wand under general motions is proposed; ii) a k-means++ based selection algorithm to optimized k clusters among a large number of point correspondences, which improves the effectiveness of employed points, is proposed; iii) no requirements for all the cameras to share a common FOV, and only pairwise overlap is needed; iv) a MATLAB-based Graphical User Interface (GUI) is developed to visualize and simplify the intrinsic calibration, and the source code is available at https://github.com/DengMark/CameraCalibrator; v) a virtual platform based on Unity3D technology [1] is designed to test the calibration method (it is the first time as far as we know).

The remainder of this paper is organized as follows. Section II gives a problem formulation of the proposed multi-camera calibration, including some preliminaries and the objective, followed by the main calibration algorithms described in Section III. Section IV shows the experimental results based on a synthetic and real multi-camera setup. Finally, Section V gives the conclusions and future research plan.

II. Problem Formulation

A. Preliminaries

1) Notations and Definitions: Some necessary notations and coordinate definitions are described in the following, which are the basic notions of perspective projective geometry, such as homogeneous coordinates and multi-view geometry for cameras. Let $\mathbb{R}^{m\times n}$ denote a real matrix with m rows and n columns while $\mathbb{R}^n$ an n-dimensional real column vector. Define $\mathbf{A}^{\mathrm{T}}$ and $\mathbf{A}^{-1}$ as transpose and inverse of matrix $\mathbf{A}$, respectively. Symbol $\bar{\mathbf{x}}$ is used to denote the homogeneous coordinate of $\mathbf{x}$ by adding 1 as the last element.

The overview diagram of the VSN is shown in Fig. 1. There are mainly two coordinate frames involved: Earth-Fixed Coordinate Frame (EFCF) and Camera Coordinate Frame (CCF). The EFCF $\{\mathrm{e}\}=\{o_{\mathrm{e}}x_{\mathrm{e}}y_{\mathrm{e}}z_{\mathrm{e}}\}$ denotes a right-hand frame with the coordinate origin o_{e} located on the horizontal ground plane. The CCF $\{\mathrm{c}_i\}=\{o_{\mathrm{c}_i}x_{\mathrm{c}_i}y_{\mathrm{c}_i}z_{\mathrm{c}_i}\},(i=0,1,\cdots,M-1)$ is attached to each external camera with its origin o_{c_i} located in the camera optical center and the $o_{\mathrm{c}_i}z_{\mathrm{c}_i}$ axis aligned with the optical axis. The system consisting of $M\in\mathbb{Z}_+$ cameras captures a 1D calibration pattern with three collinear markers, i.e., the points $\mathbf{A}_j,\mathbf{B}_j,\mathbf{C}_j$. The subscript j denotes the index of the image frame. When the calibration pattern is moved freely in the working volumes, the projections of each marker $(\mathbf{A}_j,\mathbf{B}_j,\mathbf{C}_j)$ in the jth frame from the ith camera are denoted as $(\mathbf{a}_{ij},\mathbf{b}_{ij},\mathbf{c}_{ij})$ in spherical coordinate. The 3D spherical coordinates are finally transformed into 2D image coordinates which are given by the generic camera model in the following.

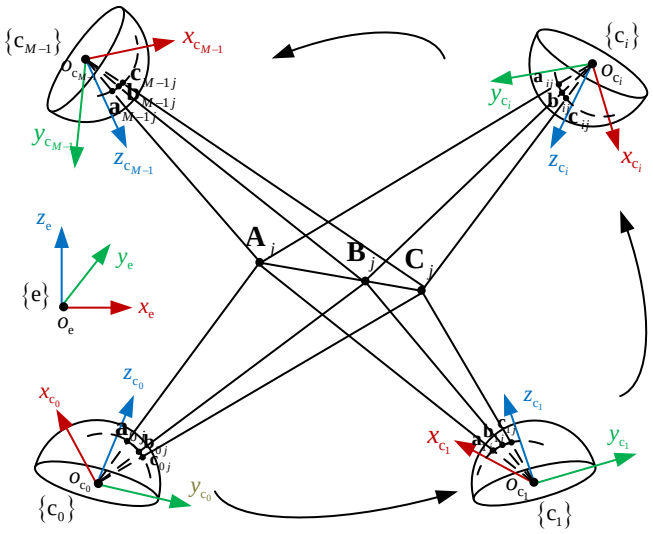


Fig. 1. Overview diagram of the visual sensor network.

In this paper, a generic camera model is employed, which is applicable for conventional, wide-angle, and fish-eye lens cameras. The model description can be found in [9], [10]. To simplify, the mapping from a 3D world point ${}^{\mathrm{e}}\mathbf{P}=[\ p_{x_{\mathrm{e}}}\quad p_{y_{\mathrm{e}}}\quad p_{z_{\mathrm{e}}}\ ]^{\mathrm{T}}\in\mathbb{R}^3$ to a 2D pixel point $\mathbf{p}=[\ u\quad v\ ]^{\mathrm{T}}\in\mathbb{R}^2$ can be summarized as

$$\mathbf{p}=\mathbf{G}\left(k_1,k_2,m_u,m_v,u_0,v_0,k_3,k_4,k_5,\mathbf{R}_{\mathrm{e}}^{\mathrm{c}},\mathbf{t}_{\mathrm{e}}^{\mathrm{c}},{}^{\mathrm{e}}\mathbf{P}\right). \tag{1}$$

where (u_0,v_0) is the principal point and (m_u,m_v) are the total number of pixels per unit distance in horizontal and vertical direction, respectively. As for the parameters of the function $\mathbf{G}(\cdot)$, the first nine parameters $(k_1,k_2,m_u,m_v,u_0,v_0,k_3,k_4,k_5)$ are called the intrinsic parameters which describe the mapping from the point ${}^{\mathrm{c}}\mathbf{P}\in\mathbb{R}^3$ w.r.t. the CCF to the image point $\mathbf{p}$ while the remaining parameters $\left(\mathbf{R}_{\mathrm{e}}^{\mathrm{c}}\in\mathbb{R}^{3\times3},\mathbf{t}_{\mathrm{e}}^{\mathrm{c}}\in\mathbb{R}^3\right)$ are corresponding extrinsic parameters which are the rotation and translation transformation matrices from the EFCF to the CCF. Besides, the calibration patterns employed are the same as those in our previous work [10], including a 1D wand with three identical reflective markers and a 2D triangle board with four markers.

[1]Unity3D (https://unity3d.com) is a cross-platform game engine developed by Unity Technologies. The engine can be employed to create 2D, 3D, virtual reality, and augmented reality games, as well as simulations.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSEN.2019.2958104, IEEE Sensors Journal

B. Objective

Without loss of generality, the 0th camera is regarded as the reference camera. There are some necessary assumptions in the proposed method needed to be claimed.

Assumption 1. Only the radian distortion is considered in the camera model.

Assumption 2. Cameras are synchronized at fixed Frames Per Second (FPS).

Assumption 3. Camera intrinsic parameters remain unchanged during calibration.

Assumption 4. At least two cameras share a common overlap.

Based on the assumptions above, the objective is to calibrate the intrinsic parameters $\left(k_1^i, k_2^i, m_u^i, m_v^i, u_0^i, v_0^i, k_3^i, k_4^i, k_5^i\right)$ and extrinsic parameters $(\mathbf{R}_i, \mathbf{t}_i)$ of each camera provided the point correspondences $\left(\mathbf{p}_{ijk} \leftrightarrow {}^{\mathrm{e}}\mathbf{P}_{jk}\right)$, and $i = 0, 1, ..., M-1$ denotes the index of camera, $j = 1, 2, ..., N$ denotes the image frame index, while $k = 1, 2, 3$ denotes the three markers mounted on the calibration pattern. To be more specific, assume $\mathbf{p}_{ijk}$ as the actual projection of markers, the calibration is to estimate the complete parameters by solving the nonlinear minimization function defined as

$$\min_{\mathbf{R}_i, \mathbf{t}_i} \sum_{i=0}^{M-1} \sum_{j=1}^{N} \sum_{k=1}^{3} \left\| \mathbf{p}_{ijk} - \mathbf{G}\left(k_1^i, k_2^i, m_u^i, m_v^i, u_0^i, v_0^i, k_3^i, k_4^i, k_5^i, \mathbf{R}_i, \mathbf{t}_i, {}^{\mathrm{e}}\mathbf{P}_{jk}\right) \right\|^2. \quad (2)$$

C. Problem Analysis

For each observation, we have two independent equations for each marker according to (1). Thus, each observation of the calibration pattern with three markers will yield in total known six equations. Besides, considering that the three markers for each observation must be collinear, there are only five independent equations for each observation. Without loss of generality, the CCF is chosen to define the 1D calibration pattern. Then, we only need five parameters to define the positions of the three collinear markers with known distances. For example, we need three parameters to define the coordinates of one marker in the CCF and two parameters to define the orientation direction of the line. Thus, the nine parameters of the three markers are cut down to five parameters.

Since the points of the 1D calibration object are not simultaneously captured by all the cameras due to non-overlapping, For total N observations, define there are N_i observations for the ith camera. Note that $N_i \leq N$. Then, the total number of independent equations is $\sum_{i=1}^{M} 5N_i$. Since the three markers have different coordinates in each observation, the unknown point positions to be estimated is $5N$. Since there are nine intrinsic parameters and six extrinsic parameters (except the extrinsic parameters of the reference camera) to describe a camera, the total number of the camera parameters is $15M-6$. Thus, the total number of unknowns is $5N+15M-6$. Thus, the solution to the problem (2) exists if $\sum_{i=1}^{M} 5N_i \geq 5N+15M-6$ holds. Besides, the total number of observations is usually a big number such as 3000, then the solution exists. To speed up the optimization process, the nine intrinsic parameters need to be calibrated prior to deployment. Thus the number of unknowns is decreased to $5N+6M-6$, and the solution exists if $\sum_{i=1}^{M} 5N_i \geq 5N+6M-6$ holds.

Similarly, for situations with four or more collinear markers with known distances, there are more markers for each observation. It seems that there are more independent equations, but in fact it is not. As pointed in [14], the addition of the fourth marker or even more markers does not increase the total number of independent equations. It will always be $5N$ for four or more collinear markers since the collinearity and cross ratio are preserved under perspective projection. Since the total camera number is not changed, the number of unknowns remains the same. In practice, since three points with different distances are sufficient for marker detection and matching algorithms, we usually intend to employ three collinear markers to calibrate multiple cameras.

In spite of simplification above, the number of parameters to be optimized for the problem remains great. Furthermore, the structure of the visual sensor network is complicated with multiple cameras and overlapping. In this paper, we tend to divide the multi-camera calibration problem to many pairwise calibration problems. Subsection III-C in the following solves the optimization in pairwise calibration. Then, combined with a vision graph with the pairwise calibration results, Subsection III-D give the optimal path of each camera and a solution based on the bundle adjustment. The problem can be optimized by using Levenberg-Marquardt (LM) algorithm [23]. Due to the sparse nature of the problem, where is lack of interaction between cameras with great reprojection errors, spare bundle adjustment is applied [24].

III. Proposed Method

An outline of the flow diagram is shown in Fig. 2, and it is shown that the proposed method is mainly divided into three parts: intrinsic calibration, extrinsic calibration, and global coordinate determination. The intrinsic calibration is to calibrate the intrinsic parameters, including focal length, principal point, and lens distortion of each camera. Afterwards, the external calibration is performed to define how the local CCF is related to the EFCF, i.e., to determine the position and orientation of each camera. It can be roughly divided into pairwise calibration and multi-camera calibration. Furthermore, the proposed extrinsic calibration can be summarized in the following.

(a) Image acquisition and marker detection on multiple cameras.

(b) Marker selection based on the collinear principle and fixed-length structure, during which each marker is guaranteed to be captured by at least two cameras.

(c) Computation of the essential matrix between two cameras based on epipolar geometry.

(d) Initial extrinsic calibration by decomposing the essential matrix into rotational and translational terms and determining the scale factor through triangulation and optimization.

(e) Bundle adjustment to obtain the external parameters between two cameras.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSEN.2019.2958104, IEEE Sensors Journal

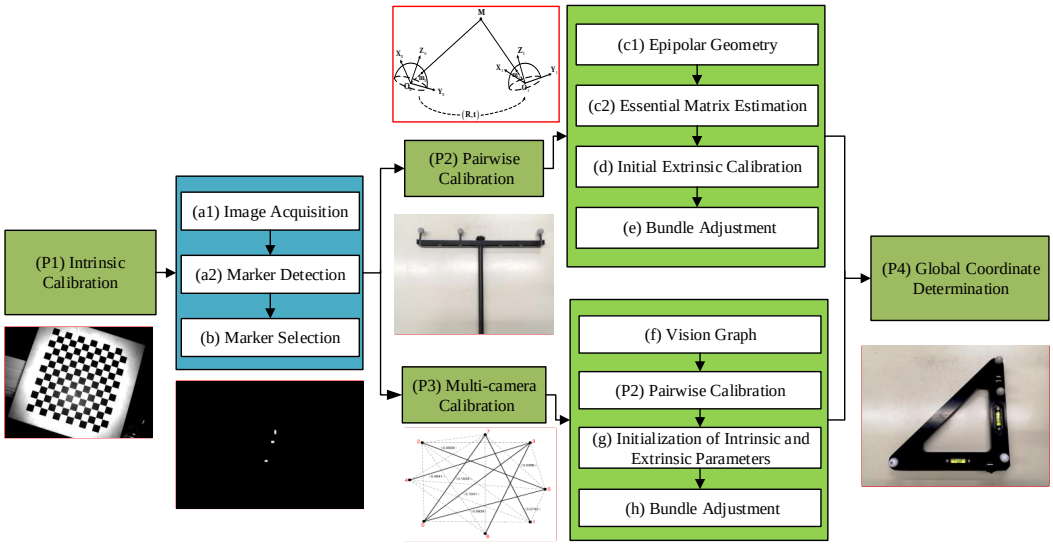


Fig. 2. Flow diagram of the proposed multi-camera calibration algorithm.

(f) Construction of adjacency matrix for vision graph describing interconnections among the cameras and optimal path determination using Dijkstra's algorithm.

(g) Initialization of intrinsic and extrinsic parameters based on the optimal path and pairwise calibration.

(h) Global optimization of all the extrinsic parameters based on bundle adjustment.

Different from most multi-camera calibration algorithms, the last step of our calibration is to determine the EFCF by using a triangle. This step is to transform the position and orientation of each camera including the reference camera to the EFCF, so that 3D measurement and tracking are realized easily.

A. Intrinsic Calibration

In this paper, it is assumed that the camera intrinsic parameters are needed to be separately estimated prior to calibration process since the parameters remain unchanged during calibration and the intrinsic parameters are not dependent on the EFCF. The goal of the intrinsic calibration is to estimate the nine parameters $\left(k_1^i, k_2^i, m_u^i, m_v^i, u_0^i, v_0^i, k_3^i, k_4^i, k_5^i\right), i = 0, 1, ..., M-1$ of each camera.

As shown in Fig. 3, a MATLAB-based GUI of the intrinsic calibration is developed to visualize and simplify the whole process, in which we can easily modify the camera configuration and as well as the projection model type as described in [9]. Before running the intrinsic calibration algorithm, a high-precision checkerboard with a size of 30 mm square is used as the calibration plane, and a few images of the plane under different positions and orientations are captured by moving either the plane or the camera. Then, in the GUI, the camera configuration needs to be edited, including the focal length, the FOV, pixel size, and image resolution. Besides, we choose the calibration model only to consider the radial effects, and the projection type is chosen as the equidistance projection. After loading all the images, the detection and extraction of feature points are automatically executed and shown in the top right of the GUI. Finally, the calibration process is finished, and the nine internal parameters are shown as below. The reprojection errors of each view are shown in the below right bar, and it is found that the errors are so small that the calibrated camera intrinsic parameters can be employed in the following.

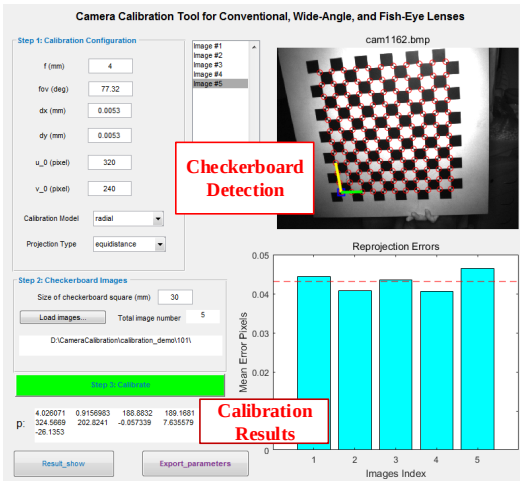


Fig. 3. MATLAB-based GUI of the intrinsic calibration process.

B. Marker Detection and Selection

Marker detection is to detect and extract the subpixel locations of the three collinear markers, which significantly influences calibration accuracy. In practice, the marker detection process is implemented in real-time on each camera client. Infrared reflective markers are employed as the calibration pattern, and they are easily observed and detected by the camera with an infrared-pass filter. The markers are first detected by thresholding, smoothing and segmenting, Then the centers of markers are extracted and synchronously transferred to the server as the main inputs of the calibration algorithm. To calculate the subpixel marker center precisely, image moment $M_{p,q}$ is employed with the definition as [25]

$$M_{pq} = \sum_u \sum_v u^p v^q I(u,v) \tag{3}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSEN.2019.2958104, IEEE Sensors Journal

where $p,q \in \mathbb{R}_+ \cup \{0\}$. $I(u,v)$ is the intensity of image point (u,v) after thresholding and smoothing procedure. Thus, the center of the marker is computed as

$$\begin{cases} \hat{u} = M_{10}/M_{00} \\ \hat{v} = M_{01}/M_{00}. \end{cases} \tag{4}$$

After obtaining the subpixel locations of the three markers, some noise points or outliers are eliminated based on colinearity. Besides, considering that there may be many points distributing mainly in the same area such as the center of the volume with common overlap, a *k*-means++ based selection algorithm is proposed to choose optimal matching points. The *k*-means++, an augmented algorithm of the classical *k*-means method, is a widely used clustering technique to minimize the average squared distance between points in the same cluster [26]. Note that in our instance, two sets of N matching points from two cameras are given defined as $\mathscr{X}_1 \subset \mathbb{R}^2$ and $\mathscr{X}_2 \subset \mathbb{R}^2$ and we need to select the optimal matching points for both point sets simultaneously. Thus, the augmented vector is proposed by combining the two point sets together to obtain a new set of data points $\mathscr{X} \subset \mathbb{R}^4$. Thus, the *k*-means++ is used to group the data points into fixed clusters. The procedure of the *k*-means++ based selection algorithm is summarized in **Algorithm 1**.

Algorithm 1 Procedure of the *k*-means++ based selection algorithm.

1. Initialization: fixed number $k \in \mathbb{Z}_+$, obtain the augmented point set $\mathscr{X}$ from the two points sets $\mathscr{X}_1$ and $\mathscr{X}_2$.
2. Choose one center c_1 uniformly at random from $\mathscr{X}$.
3. Compute the shortest distance from each data point to the closest center we have already chosen, denoted as $D(x)$; Then, compute the probability with $\frac{D(x)^2}{\sum_{x \subset \mathscr{X}} D(x)^2}$ for each data point $x \in \mathscr{X}$. Finally, choose a new center c_i based on a roulette-wheel selection [27].
4. Repeat Step 3 until k centers altogether have been chosen.
5. Refine the data points: obtain the 2D points w.r.t. each camera from the 4D points $c_i, i = 1,...,k$ chosen above.

We record the original data points and selected points based on *k*-means++ selection algorithm from two cameras in real experiments. The number k is fixed as 100, and the results are shown in Fig. 4. One test has been done with the number of original points being 14178. Results show that the selected points spread out as far as possible in the fixed resolution of 640×480 pixels.

C. Pairwise Calibration

Pairwise calibration is a fundamental step of multi-camera calibration. It is to calculate the relative pose between two cameras through epipolar geometry given two images from stereo cameras. First, the pixel coordinates of points from two cameras are transformed into normalized hemispherical coordinates. Then, it is found that an essential matrix can be estimated through epipolar geometry constraints between two cameras. Through decomposition of the essential matrix, an initial relative pose of the two cameras can be obtained up to a scale factor, and the factor is then computed from the actual

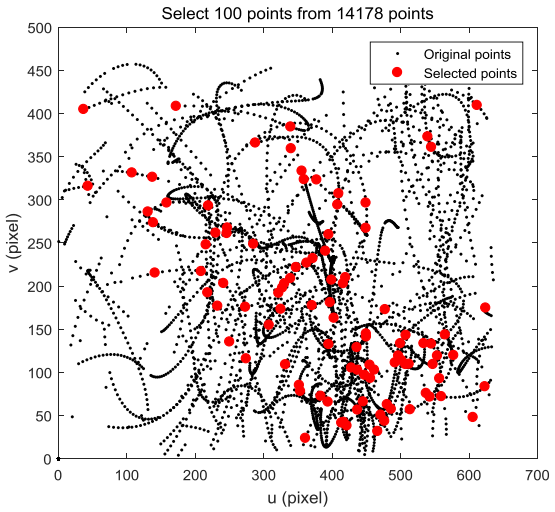


Fig. 4. Results of selecting markers based on the *k*-means++ based selection algorithm.

length constraints of the 1D wand based on 3D reconstruction algorithm. Finally, the solution is globally refined by bundle adjustment.

1) Epipolar Geometry and 3D Reconstruction:

a) Epipolar Geometry: Without loss of generality, assume that a 3D point $\mathbf{M}_j$ is projected to $\mathbf{m}_{0j}, \mathbf{m}_{1j} \in \mathbb{R}^3$ on the unit hemisphere centered at o_{c_0} and o_{c_1}, respectively. Thus, the epipolar geometric constraint is formulated as [9]

$$\mathbf{m}_{1j}^{\mathrm{T}} \mathbf{E}_{c_0}^{c_1} \mathbf{m}_{0j} = 0 \tag{5}$$

where $\mathbf{E}_{c_0}^{c_1} = \left[\mathbf{t}_{c_0}^{c_1}\right]_\times \mathbf{R}_{c_0}^{c_1}$ is defined as the essential matrix between the two cameras, $\mathbf{R}_{c_0}^{c_1} \in \mathbb{R}^{3\times3}, \mathbf{t}_{c_0}^{c_1} \in \mathbb{R}^3$ are the rotation and translation matrices from the left camera $\{c_0\}$ to the right camera $\{c_1\}$, respectively.

The point correspondences $(\mathbf{M}_j, \mathbf{m}_{0j}, \mathbf{m}_{1j})$ can be $(\mathbf{A}_j, \mathbf{a}_{0j}, \mathbf{a}_{1j})$, $(\mathbf{B}_j, \mathbf{b}_{0j}, \mathbf{b}_{1j})$ or $(\mathbf{C}_j, \mathbf{c}_{0j}, \mathbf{c}_{1j})$ for instances as shown in Fig. 1. According to equation (5), the epipolar geometric constraint depends only on the intrinsic parameters and relative pose of the cameras while it does not rely at all on the scene structure.

b) 3D Reconstruction: In this subsection, a linear reconstruction algorithm based on the epipolar geometric constraint above is proposed to solve the problem of how the position of a feature can be recovered from a set of point correspondences of two cameras. Without loss of generality, assume that the homogeneous coordinates of a 3D point $\mathbf{M}_j \in \mathbb{R}^3$ are

$$\bar{\mathbf{M}}_0 = \begin{bmatrix} X_0 \\ Y_0 \\ Z_0 \\ 1 \end{bmatrix}, \ \bar{\mathbf{M}}_1 = \begin{bmatrix} X_1 \\ Y_1 \\ Z_1 \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R}_{c_0}^{c_1} & \mathbf{t}_{c_0}^{c_1} \end{bmatrix} \bar{\mathbf{M}}_0$$

in $\{o_{c_0} x_{c_0} y_{c_0} z_{c_0}\}$ and $\{o_{c_1} x_{c_1} y_{c_1} z_{c_1}\}$, respectively. The 3D point is then projected into

$$\mathbf{m}_0 = \begin{bmatrix} \sin\theta_0 \cos\varphi_0 \\ \sin\theta_0 \sin\varphi_0 \\ \cos\theta_0 \end{bmatrix}, \mathbf{m}_1 = \begin{bmatrix} \sin\theta_1 \cos\varphi_1 \\ \sin\theta_1 \sin\varphi_1 \\ \cos\theta_1 \end{bmatrix}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSEN.2019.2958104, IEEE Sensors Journal

on the unit hemisphere centered at o_{c_0} and o_{c_1}, respectively. Then, one has equations as follows.

$$\begin{cases} s_0\mathbf{m}_0 = \mathbf{P}_0\bar{\mathbf{M}}_0 \\ s_1\mathbf{m}_1 = \mathbf{P}_1\bar{\mathbf{M}}_0 \end{cases} \tag{6}$$

where s_0, s_1 are scale factors and $\mathbf{P}_0 = [\mathbf{I}_3\ \mathbf{0}_{3\times1}] \in \mathbb{R}^{3\times4}$, $\mathbf{P}_1 = [\mathbf{R}_{c_0}^{c_1}\ \mathbf{t}_{c_0}^{c_1}] \in \mathbb{R}^{3\times4}$. Suppose the detailed form of the matrices are

$$\mathbf{P}_0 = \begin{bmatrix} P_{11}^0 & P_{12}^0 & P_{13}^0 & P_{14}^0 \\ P_{21}^0 & P_{22}^0 & P_{23}^0 & P_{24}^0 \\ P_{31}^0 & P_{32}^0 & P_{33}^0 & P_{34}^0 \end{bmatrix}, \mathbf{P}_1 = \begin{bmatrix} P_{11}^1 & P_{12}^1 & P_{13}^1 & P_{14}^1 \\ P_{21}^1 & P_{22}^1 & P_{23}^1 & P_{24}^1 \\ P_{31}^1 & P_{32}^1 & P_{33}^1 & P_{34}^1 \end{bmatrix}$$

For each image point on the unit hemisphere, the scale factors in equation (6) can be eliminated through a cross product. Thus, one has the abstract form as

$$\mathbf{A}\mathbf{M}_0 = \mathbf{b}.$$

Therefore, given $\mathbf{m}_0, \mathbf{m}_1, \mathbf{R}_{c_0}^{c_1}, \mathbf{t}_{c_0}^{c_1}$, the homogeneous coordinates of a 3D point $\mathbf{M}_j$ w.r.t. $\{o_{c_0}x_{c_0}y_{c_0}z_{c_0}\}$ is reconstructed by a least square method

$$\mathbf{M}_0 = \left(\mathbf{A}^{\mathrm{T}}\mathbf{A}\right)^{-1}\mathbf{A}^{\mathrm{T}}\mathbf{b}. \tag{7}$$

It is noted that equation (7) provides a linear solution, which is subject to noises. Thus, the solution could be refined by minimizing the reprojection errors or Sampson errors. Furthermore, the linear reconstruction algorithm could be extended to N-view ($N > 2$) triangulation for the calibration of multiple cameras.

2) Essential Matrix Estimation: Given a set of point correspondences between two cameras, $\{\mathbf{m}_{0j} \leftrightarrow \mathbf{m}_{1j}\}, j = 1,...,N$, the objective is to estimate the essential matrix $\mathbf{E}_{c_0}^{c_1} = \left[\mathbf{t}_{c_0}^{c_1}\right]_\times \mathbf{R}_{c_0}^{c_1}$ such that the epipolar geometry constraints (5) is fulfilled. The essential matrix has only five degrees of freedom: both the rotation matrix and the translation vector have three degrees of freedom, but there is a single unknown scale factor. These properties makes the essential matrix have two equal singular values and a third singular value equal to zero. The problem of estimating the essential matrix is then formulated as

$$\mathbf{H}\mathbf{e} = \mathbf{0} \tag{8}$$

where $\mathbf{H} \in \mathbb{R}^{N\times9}$ contains the spherical coordinates of the N measurement point correspondences and $\mathbf{e} \in \mathbb{R}^9$ contains the essential matrix coefficients by stacking operation. Because of the scale ambiguity, the last term of the coefficient vector $\mathbf{e}$ is set to 1 to determine $\mathbf{e}$ up to scale. Thus the unknown number of terms is reduced to eight. Thus, equation (8) can be solved by the 8-point algorithm [28]. Considering the existence of estimation error, the derived matrix does not meet the properties of an essential matrix, it is proposed to compute the closest matrix $\mathbf{E}_1$ to the obtained matrix $\mathbf{E}$ in the sense of Frobenius norm subject to the condition $\det(\mathbf{E}_1) = 0$.

Remark 1. Given at least 8 point correspondences, it is possible to solve linearly for the solving of $\mathbf{e}$ up to scale, but the solution is sensitive to image noises. With more than 8 point correspondences, the coefficient matrix $\mathbf{H}$ is usually with column rank due to noises, and the linear solution does not exist. Then, the least-squares solution for $\mathbf{e}$ is the singular vector corresponding to the smallest singular value of $\mathbf{H}$. Consider one extreme situation where all the points are imaged onto the center of the image plane, and the corresponding angle $\theta_0 = 0$ for an instant. It is calculated that the matrix $\mathbf{H}$ has six zero columns, and the rank is only three. Therefore, there are infinitely many solutions to equation (8). However, this situation is almost impossible. On the other hand, when most points are imaged onto the edges of the image plane, the values of the angles θ are close to half of FOV, and the solution still exists. In all, when the 1D object moves freely, the solution to equation (8) always exists as long as the three markers of the object are accurately captured by synchronous cameras.

3) Initial Extrinsic Calibration: Once the essential matrix $\mathbf{E}_{c_0}^{c_1}$ is obtained by solving the problem (8), the rotation matrix and translation vector of the two cameras may be retrieved through an essential matrix decomposition. This decomposition is implemented by means of a Singular Value Decomposition (SVD). It is supposed that the rotation matrix of the reference camera equals a unit matrix while the translation vector remains zero. The decomposition of the essential matrix has four possible solutions with a scale ambiguity of the translation vector. The solution can be concluded in the following theorem.

Theorem 1. *Recovering the camera matrix from the essential matrix. Suppose that the essential matrix is defined as* $\mathbf{E} = [\mathbf{t}]_\times \mathbf{R}$. *Let the singular value decomposition of the essential matrix be* $\mathbf{E} = \mathbf{U}\mathrm{diag}(1,1,0)\mathbf{V}^{\mathrm{T}}$, *and assume*

$$\mathbf{W} = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Then, the four possible solutions for the camera matrix $\mathbf{P} = [\mathbf{R}\ \mathbf{t}]$ *are:*

$$\mathbf{P} = \begin{cases} \mathbf{P}_1 = \left[\ \mathbf{U}\mathbf{W}\mathbf{V}^{\mathrm{T}} \quad \mathbf{U}\left[\ 0 \quad 0 \quad 1\ \right]^{\mathrm{T}}\ \right] \\ \mathbf{P}_2 = \left[\ \mathbf{U}\mathbf{W}\mathbf{V}^{\mathrm{T}} \quad -\mathbf{U}\left[\ 0 \quad 0 \quad 1\ \right]^{\mathrm{T}}\ \right] \\ \mathbf{P}_3 = \left[\ \mathbf{U}\mathbf{W}^{\mathrm{T}}\mathbf{V}^{\mathrm{T}} \quad \mathbf{U}\left[\ 0 \quad 0 \quad 1\ \right]^{\mathrm{T}}\ \right] \\ \mathbf{P}_4 = \left[\ \mathbf{U}\mathbf{W}^{\mathrm{T}}\mathbf{V}^{\mathrm{T}} \quad -\mathbf{U}\left[\ 0 \quad 0 \quad 1\ \right]^{\mathrm{T}}\ \right] \end{cases}$$

It is noted that only one of the four solutions satisfy the condition that a 3D point reconstructed from both camera views must be in front of both cameras, namely, the value of the third term of the reconstructed coordinates must be positive. Thus, the solution is found by testing with a single point to determine whether it is in front of both cameras. Finally, the scale ambiguity λ of the essential matrix is simultaneously resolved from the known geometry of the 1D calibration pattern, defined as

$$\lambda = \frac{1}{N}\sum_{j=1}^{N}\frac{L_{AC}}{\left\|\mathbf{A}_j^r - \mathbf{C}_j^r\right\|} \tag{9}$$

where N is the total number of frames, L_{AC} is the actual length of the markers $\mathbf{A}, \mathbf{C}$, and $\mathbf{A}_j^r, \mathbf{C}_j^r$ represent the reconstructed coordinates of the markers $\mathbf{A}, \mathbf{C}$ based on the 3D linear reconstruction solution (7) from two views of cameras with the intrinsic and extrinsic parameters obtained above. Finally, the translation vector is initialized as $\mathbf{t}_{c_0}^{c_1} = \lambda\tilde{\mathbf{t}}_{c_0}^{c_1}$.

4) Bundle Adjustment: Up to now, the relative position and orientation of the two cameras are obtained in a closed-form solution which is therefore prone to errors. This subsection is to refine the extrinsic parameters using bundle adjustment

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSEN.2019.2958104, IEEE Sensors Journal

algorithm.Given the initial pose of cameras based on the **Theorem** 1, define $\mathbf{x}$ as the calibrated extrinsic parameters between two cameras, one can minimize the following reprojection error

$$\min_{\mathbf{x},\mathbf{A}_j,\mathbf{B}_j,\mathbf{C}_j} \sum_{i=0}^{1}\sum_{j=1}^{N} \left(\Psi_a^2+\Psi_b^2+\Psi_c^2\right) \tag{10}$$

Here,

$$\begin{aligned}\Psi_a &= \left\|\mathbf{a}_{ij}-\mathbf{G}\left(k_1^i,k_2^i,m_u^i,m_v^i,u_0^i,v_0^i,k_3^i,k_4^i,k_5^i,\mathbf{x},\mathbf{A}_j\right)\right\|\\ \Psi_b &= \left\|\mathbf{b}_{ij}-\mathbf{G}\left(k_1^i,k_2^i,m_u^i,m_v^i,u_0^i,v_0^i,k_3^i,k_4^i,k_5^i,\mathbf{x},\mathbf{B}_j\right)\right\|\\ \Psi_c &= \left\|\mathbf{c}_{ij}-\mathbf{G}\left(k_1^i,k_2^i,m_u^i,m_v^i,u_0^i,v_0^i,k_3^i,k_4^i,k_5^i,\mathbf{x},\mathbf{C}_j\right)\right\|.\end{aligned}$$

Before the nonlinear optimization, the reconstructed distance between the two 3D points $\mathbf{A},\mathbf{C}$ is compared with the known length L_{AC} to removed outliers. If

$$\left|\frac{L_{AC}-\left\|\mathbf{A}_j^r-\mathbf{C}_j^r\right\|}{L_{AC}}\right|>1\%$$

holds, then the jth image pair is removed from the measurements.

Besides, the number of the parameters to be optimized could be reduced if the collinear constraints within the three points are taken into account. The positions of $\mathbf{B}_j,\mathbf{C}_j$ can be expressed by the starting point $\mathbf{A}_j$ with the normalized direction vector $\mathbf{n}_j$ and their fixed distance. They have the relation as follows

$$\begin{cases}\mathbf{B}_j=\mathbf{A}_j+L_{AB}\mathbf{n}_j\\ \mathbf{C}_j=\mathbf{A}_j+L_{AC}\mathbf{n}_j\end{cases} \tag{11}$$

where $\mathbf{n}_j=\left[\begin{array}{ccc}\sin\phi_j\cos\theta_j & \sin\phi_j\sin\theta_j & \cos\phi_j\end{array}\right]^{\mathrm{T}}\in\mathbb{R}^3$ denotes the normalized direction of the 1D wand, and it is known that the (ϕ_j,θ_j) are the spherical coordinates centered at $\mathbf{A}_j$. Substituting equation (11) into the optimization problem (10) will reduce the number of input parameters from nine to five. Since $\mathbf{A}_j^r,\mathbf{B}_j^r,\mathbf{C}_j^r$ is known from the linear reconstruction algorithm, the normalized direction vector $\mathbf{n}_j$ can be obtained based on equation (11). Thus, the initialization of all the necessary parameters is finished. Finally, the nonlinear minimization problem (10) can be solved by bundle adjustment using the sparse LM algorithm.

D. Multi-camera Calibration

Multi-camera calibration in this section involves more than two cameras.In this paper, the structure of the VSN is represented by a weighted vision graph as in [18], the determination of the vision graph will be described in detail first, followed by the initialization of intrinsic and extrinsic parameters of multiple cameras in the VSN. Finally, the total parameters are refined by bundle adjustment.

1) Vision Graph Determination: In terms of vision graph theory, the layout of M cameras can be represented by a graph G consisting of M vertices $V_i, i=0,...,M-1$ which represents individual cameras. The edges of the graph G represent the overlap between different camera pairs, and the weights a_{ij} are assigned to the edges corresponding to the reprojection error of stereo calibration between the ith and jth camera. If the reprojection error is too large since there is a small number of common points, the corresponding vertices are not connected. To describe the graph clearly, adjacency matrix $\mathbf{A}(G)=\left\{a_{ij}\right\}\in\mathbb{R}^{M\times M}$ is defined. It is known that $\mathbf{A}(G)$ is a symmetric matrix. The weights then represent the accuracy of the internal calibration between two cameras. The adjacency matrix is updated after finishing pairwise calibration between each camera pair and obtaining the corresponding reprojection error. Note that using the reprojection errors as the weights is a different method from that in [9] where the number of common points is related to the weights. Next, to find the optimal path for transformation from the reference camera to other cameras, the Dijkstra's shortest path algorithm [29] on the weighted vision graph is employed. The algorithm solves the single-source shortest path problem for a graph with positive weights, and it will succeed as long as the graph is connected [18].

2) Initialization of Intrinsic and Extrinsic Parameters: Assume that i,j,k are indices of consecutive cameras on the shortest path of the vision graph. According to pairwise calibration in Subsection III-C, the transformations from the ith camera to the jth camera and from the jth camera to the kth camera are $[\mathbf{R}_{\mathrm{c}_i}^{\mathrm{c}_j}\ \mathbf{t}_{\mathrm{c}_i}^{\mathrm{c}_j}]$ and $[\mathbf{R}_{\mathrm{c}_j}^{\mathrm{c}_k}\ \mathbf{t}_{\mathrm{c}_j}^{\mathrm{c}_k}]$, respectively. Then, the transformation from the ith camera to the kth camera can be obtained as follows.

$$\begin{cases}\mathbf{R}_{\mathrm{c}_i}^{\mathrm{c}_k}=\mathbf{R}_{\mathrm{c}_j}^{\mathrm{c}_k}\mathbf{R}_{\mathrm{c}_i}^{\mathrm{c}_j}\\ \mathbf{t}_{\mathrm{c}_i}^{\mathrm{c}_k}=\mathbf{R}_{\mathrm{c}_j}^{\mathrm{c}_k}\mathbf{t}_{\mathrm{c}_i}^{\mathrm{c}_j}+\mathbf{t}_{\mathrm{c}_j}^{\mathrm{c}_k}.\end{cases} \tag{12}$$

If a path from the reference camera has a length longer than two, equation (12) is employed sequentially to cover the entire path. Therefore, combined with the intrinsic parameters already calibrated in Subsection III-A, the initialization of intrinsic and extrinsic parameters of all cameras in the VSN is finished.

3) Bundle Adjustment: Similarly, the objective is to refine all parameters using bundle adjustment by minimizing the reprojection. Given the initial pose of cameras based on vision graph and pairwise calibration described above, define $\mathbf{y}$ as the extrinsic parameters of all the cameras except the reference camera, one can minimize the following reprojection error

$$\min_{\mathbf{y},\mathbf{A}_j,\mathbf{B}_j,\mathbf{C}_j} \sum_{i=0}^{M-1}\sum_{j=1}^{N} \left(\Psi_a^2+\Psi_b^2+\Psi_c^2\right) \tag{13}$$

Here,

$$\begin{aligned}\Psi_a &= \left\|\mathbf{a}_{ij}-\mathbf{G}\left(k_1^i,k_2^i,m_u^i,m_v^i,u_0^i,v_0^i,k_3^i,k_4^i,k_5^i,\mathbf{y},\mathbf{A}_j\right)\right\|\\ \Psi_b &= \left\|\mathbf{b}_{ij}-\mathbf{G}\left(k_1^i,k_2^i,m_u^i,m_v^i,u_0^i,v_0^i,k_3^i,k_4^i,k_5^i,\mathbf{y},\mathbf{B}_j\right)\right\|\\ \Psi_c &= \left\|\mathbf{c}_{ij}-\mathbf{G}\left(k_1^i,k_2^i,m_u^i,m_v^i,u_0^i,v_0^i,k_3^i,k_4^i,k_5^i,\mathbf{y},\mathbf{C}_j\right)\right\|.\end{aligned}$$

Besides, the positions of $\mathbf{B}_j,\mathbf{C}_j$ can be expressed by the point $\mathbf{A}_j$ with the normalized direction vector $\mathbf{n}_j$ and their fixed distance in equation (11). Similarly, $\mathbf{n}_j$ is obtained from the reconstructed points $\mathbf{A}_j^r,\mathbf{B}_j^r,\mathbf{C}_j^r$ by N-view triangulation algorithm extended from equation (7). After all the optimization variables are initialized, the multi-camera calibration problem (13) can be effectively optimized by using the sparse LM algorithm which simultaneously refines the camera parameters and the 3D structure w.r.t. the reference camera.

E. Global Coordinate Determination

In general, we can obtain the intrinsic parameters $\left(k_1^i,k_2^i,m_u^i,m_v^i,u_0^i,v_0^i,k_3^i,k_4^i,k_5^i\right), i=0,1,...,M-1$of each camera, and their extrinsic parameters $[\mathbf{R}_{\mathrm{c}_0}^{\mathrm{c}_i}\ \mathbf{t}_{\mathrm{c}_0}^{\mathrm{c}_i}], i=1,...,M-1$ with regards to the 0th camera. Different from most literature,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSEN.2019.2958104, IEEE Sensors Journal

one more step is added after multi-camera calibration to determine the EFCF $\{\mathrm{e}\} = \{o_\mathrm{e} x_\mathrm{e} y_\mathrm{e} z_\mathrm{e}\}$ defined by a triangle. In practice, the triangle board is placed horizontally on the ground floor (it is better to be at the center of the volume of multiple cameras), where the four markers are employed to retrieve the transformations of each camera w.r.t. the EFCF, denoted as $[\mathbf{R}_\mathrm{e}^{\mathrm{c}_i}\ \mathbf{t}_\mathrm{e}^{\mathrm{c}_i}], i = 0, \cdots, M-1$. Therefore, given the spatial position relation of the four markers and the extrinsic parameters of each camera w.r.t. the reference camera based on multi-camera calibration algorithm described above, the objective is to calculate

$$\begin{cases} \mathbf{R}_\mathrm{e}^{\mathrm{c}_i} = \mathbf{R}_{\mathrm{c}_0}^{\mathrm{c}_i} \mathbf{R}_\mathrm{e}^{\mathrm{c}_0} \\ \mathbf{t}_\mathrm{e}^{\mathrm{c}_i} = \mathbf{R}_{\mathrm{c}_0}^{\mathrm{c}_i} \mathbf{t}_\mathrm{e}^{\mathrm{c}_0} + \mathbf{t}_{\mathrm{c}_0}^{\mathrm{c}_i}. \end{cases} \tag{14}$$

According to equation (14), $\mathbf{R}_{\mathrm{c}_0}^{\mathrm{c}_i}$ and $\mathbf{t}_{\mathrm{c}_0}^{\mathrm{c}_i}$ are already calibrated according to the multi-camera calibration describe above. Thus, the problem is transformed into the determination of the rotation matrix and the translation vector of the 0th camera w.r.t. the EFCF, i.e., $[\mathbf{R}_\mathrm{e}^{\mathrm{c}_0}\ \mathbf{t}_\mathrm{e}^{\mathrm{c}_0}]$. The main idea is as follows. Given the global coordinates of the four markers, denoted as $\mathbf{M}_j, j = 1,2,3,4$, combined with the transformations of each camera w.r.t. the 0th camera, the corresponding CCF on ith camera is built as

$${}^{\mathrm{c}_i}\mathbf{M}_j = \mathbf{R}_\mathrm{e}^{\mathrm{c}_i} \cdot \mathbf{M}_j + \mathbf{t}_\mathrm{e}^{\mathrm{c}_i} \tag{15}$$

where the form of $[\mathbf{R}_\mathrm{e}^{\mathrm{c}_i}\ \mathbf{t}_\mathrm{e}^{\mathrm{c}_i}]$ are given by equation (14). Then, the estimated variables become only the $[\mathbf{R}_\mathrm{e}^{\mathrm{c}_0}\ \mathbf{t}_\mathrm{e}^{\mathrm{c}_0}]$. Thus, combining equation (15) with the camera model, the reprojected pixels of the points $\mathbf{M}_j$ can be obtained. And the parameters are optimized by minimizing the reprojection error by using the LM algorithm.

Note that the global coordinate determination process involves the accurate matching of each marker on the triangle. According to the special angular relation, the four markers can be easily detected and classified into two groups, one with only a single point like the marker $\mathbf{D}$, the other group with three collinear markers $\mathbf{E}, \mathbf{F}, \mathbf{G}$. Then, the matching of the three collinear markers is the same as that of the 1D calibration wand described in Subsection III-B.

IV. Experimental Results

In this section, extensive experiments are conducted to validate the proposed multi-camera calibration algorithm including simulation experiments using synthetic data and experiments with real data. The algorithm is evaluated by the calibration accuracy, i.e., the root mean square reconstructed errors and the root mean square reprojection errors defined as follows. A video of thorough experimental tests is available at https://youtu.be/XmglR4HAsEw and our Reliable Flight Control Group website http://rfly.buaa.edu.cn.

A. Calibration Accuracy

It is necessary to define suitable error criteria to evaluate the performance of the multi-camera calibration method.

Let $\mathbf{A}_j^r, \mathbf{B}_j^r, \mathbf{C}_j^r$ be the reconstructed Euclidean coordinates of the three collinear markers of the 1D calibration wand in jth ($j = 1, \ldots N$) frame based on 3D reconstructed algorithm. Then, the root mean square reconstructed distance error is defined as follows.

$$\varepsilon_\mathrm{d} = \sqrt{\frac{1}{3N}\sum_{j=1}^{N}\left[\begin{array}{l}\left(L_{AB} - \left\|\mathbf{A}_j^r - \mathbf{B}_j^r\right\|\right)^2 + \left(L_{AC} - \left\|\mathbf{A}_j^r - \mathbf{C}_j^r\right\|\right)^2 \\ + \left(L_{BC} - \left\|\mathbf{B}_j^r - \mathbf{C}_j^r\right\|\right)^2\end{array}\right]} \tag{16}$$

Let $\mathbf{D}_j^r, \mathbf{E}_j^r, \mathbf{F}_j^r, \mathbf{G}_j^r$ be the reconstructed Euclidean coordinates of the four markers of the triangle in jth ($j = 1, \ldots N$) frame based on 3D reconstructed algorithm. Then, the root mean square reconstructed Euclidean error of each marker is defined as follows.

$$\varepsilon_\mathrm{e}(\mathbf{M}) = \sqrt{\frac{1}{N}\sum_{j=1}^{N}\left(\left\|\mathbf{M}_j^r - \mathbf{M}_j\right\|\right)^2}, \mathbf{M} = \mathbf{D}, \mathbf{E}, \mathbf{F}, \mathbf{G} \tag{17}$$

Let $\mathbf{a}_{ij}, \mathbf{b}_{ij}, \mathbf{c}_{ij}$ be the projected pixels of the three markers in jth ($j = 1, \ldots N$) frame captured by the ith ($i = 0, \ldots M-1$) camera, $\hat{\mathbf{a}}_{ij}, \hat{\mathbf{b}}_{ij}, \hat{\mathbf{c}}_{ij}$ are denoted as the reprojected pixels computed with the camera model and the known global coordinates. Then, the root mean square reprojection error is defined as follows.

$$\varepsilon_\mathrm{r} = \sqrt{\frac{1}{6MN}\sum_{i=0}^{M-1}\sum_{j=1}^{N}\left((\mathbf{a}_{ij} - \hat{\mathbf{a}}_{ij})^2 + \left(\mathbf{b}_{ij} - \hat{\mathbf{b}}_{ij}\right)^2 + (\mathbf{c}_{ij} - \hat{\mathbf{c}}_{ij})^2\right)} \tag{18}$$

Fig. 5. Experiment settings of the proposed visual sensor network.

B. Simulation Experiments

In order to fulfill various testing requirements and solve the practical problem that the visual sensor network is difficult to be installed and adjusted arbitrarily, a Unity3D-based virtual visual platform, which enables users to modify camera configurations and scene settings, is developed to produce synthetic visual data.

1) Simulation Setting: As shown in Fig. 6, eight cameras are arranged in the Unity3D scene to cover an area of $10 \times 10 \times 2.5$ m, in which the transforms and rotations of the cameras can be modified arbitrarily. In the simulation, all the cameras have the same internal parameters including the image resolutions of 640×480 pixels, the focal lengths of 4 mm, the pixel sizes of 5.3×5.3 μm, and the FOV of 58.6°.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSEN.2019.2958104, IEEE Sensors Journal

The configurations of the calibration wand and the calibration triangle are the same as those in real experiments described as in [10]. In the simulation scene, some wall structures are added to occlude the vision view of some cameras. The horizontal layout of the VSN with the FOV of each camera is shown in Fig. 7. It is clear that the lines of sight of Cam3 and Cam8 are occluded by the wall.

Fig. 6. Overview of the virtual scene with an eight-camera arrangement. All the eight cameras are arranged around the view volume, and the right side is the top view of the main camera in a god's perspective. Some point lights are added to the scene to supply lighting in simulation.

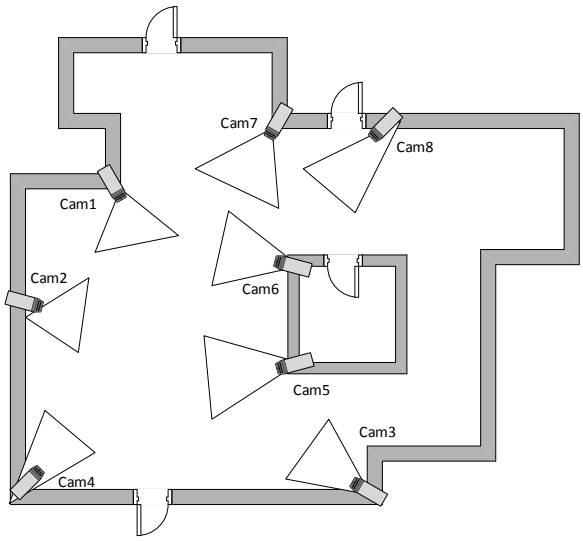


Fig. 7. The horizontal layout of the eight cameras in the VSN. The triangle symbols represent the FOV of each camera as an abstract description.

Corresponding to the multi-camera scene created above, a simple User Interface (UI) is designed to send connection request commands to the VSN and to visualize the calibration process as shown in Fig. 8. The top side of the UI shows the real-time FPS. The main operation panel is arranged on the right side, in which we can choose the number of cameras and connect any camera with the VSN, and then set or remove the wand, triangle and rigid objects.

As described above, it is necessary to calibrate the intrinsic parameters of the cameras, and the parameters remain unchanged during the whole calibration process. Therefore, 15 raw images of a checkerboard are captured by one camera in simulation and used as the inputs of the intrinsic calibration algorithm in Subsection III-A. The internal parameters are then obtained as $k_1 = 3.992, k_2 = 1.530, m_u = 189.406, m_v = 189.345, u_0 = 320.642, v_0 = 240.745, k_3 = -3.82091, k_4 = 25.2191, k_5 = -31.1443$.

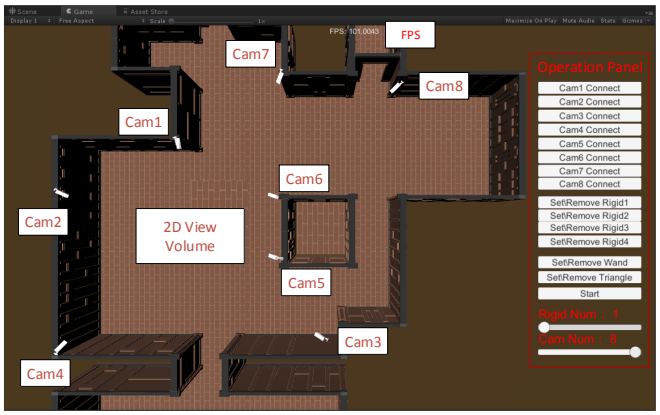


Fig. 8. UI of the virtual platform to make a connection and to visualize the calibration process.

2) Simulation Results: Based on the camera arrangement described in Fig. 6, we have generated 6000 positions of the calibration wand with no image noises to implemented the whole calibration process. The calibration errors are small with the mean reconstructed error of 0.31 mm and the mean reprojection error of 0.2158 pixels. The vision graph and the optimal path of the eight cameras are shown in Fig. 9. Simulation results show that the reprojection errors are very small, indicating that the calibration method is accurate. Besides, the the vision graph can simplify the structure of the sensor network. Furthermore, the comparison of cameras positions between the truth and calibration results in simulation is conclude in Table I. Results shown that our calibration method has an accurate estimation.

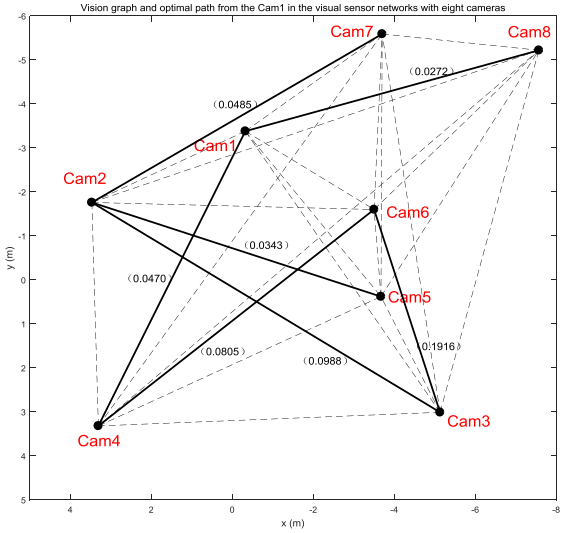


Fig. 9. Vision graph and the optimal path from the reference Cam1 obtained based on the Dijkstra's shortest path algorithm in simulation. Numbers indicate the reprojection error of the pairwise calibration of the two cameras.

Then, the performance of the calibration algorithm w.r.t. the image noises is tested. Gaussian noises with mean value 0 and standard deviation $\sigma \in \mathbb{R}_+ \cup \{0\}$ are added to the image points of each camera. The noise level σ varies from 0 to 2 pixels in delta step of 0.2 pixels. $\sigma = 0$ means the ideal

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSEN.2019.2958104, IEEE Sensors Journal

TABLE I
COMPARISON OF CAMERA POSITIONS BETWEEN THE TRUTH AND CALIBRATION RESULTS IN SIMULATION.

	t_x (m)		t_y (m)		t_z (m)	
Method	Truth	Result	Truth	Result	Truth	Result
Cam1	-0.32	-0.3224	-3.38	-3.3931	2.52	2.5329
Cam2	3.47	3.4872	-1.76	-1.766	2.5	2.5055
Cam3	-5.13	-5.1516	3.01	3.0163	2.5	2.5061
Cam4	3.31	3.3279	3.32	3.3302	2.725	2.7263
Cam5	-3.67	-3.6869	0.38	0.3788	2.7	2.7066
Cam6	-3.5	-3.5132	-1.6	-1.6056	2.5	2.5046
Cam7	-3.7	-3.7185	-5.59	-5.6037	2.5	2.5085
Cam8	-7.6	-7.6316	-5.22	-5.2342	2.5	2.5012

condition without image noise. At least 6000 positions of the calibration wand are generated, and then we obtain the calibration accuracy for each test at different noise levels. The results of the noise analysis are depicted in Fig. 10 and Fig. 11. Based on the results, it is shown that the calibration errors increase with the noise levels increase. When the noise level is under 1 pixel for each camera, the calibration results are acceptable. When the noise level raises to 2 pixels, the reconstructed Euclidean errors are 10 mm, which is considered to be large for the multi-camera tracking system. Besides, it is noted that the total FPS will decrease greatly during the simulation for eight cameras since all the algorithm and rendering are implemented in a single PC.

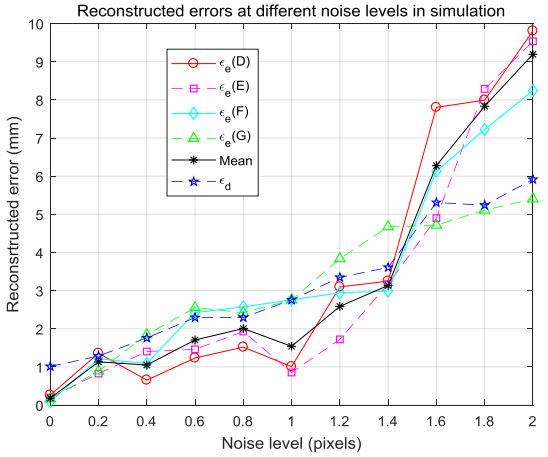


Fig. 10. Reconstructed errors at different noise levels in simulation. Black star curve represents the mean reconstructed error of the four markers.

Finally, we have compared the calibration results of the proposed method with the method of our previous work [9]. The reprojection errors of each camera are shown in Table II. Based on the results, our proposed method outperforms the previous work.

C. Real Experiments

1) Experiment Setting: The real experiments with real data are performed on five CMOS smart cameras (type: SCZE130M-GEHD) with the image resolution of 640×480 pixels to cover an area of $5 \times 5 \times 2.5$ m. The cameras are equipped with the infrared light source and infrared-pass filters to capture marker images, and are synchronized by external triggers (type: CBAT328-IO602) at 100 FPS. All the cameras have 4 mm lens (type: AZURE-0420MM) installed with the

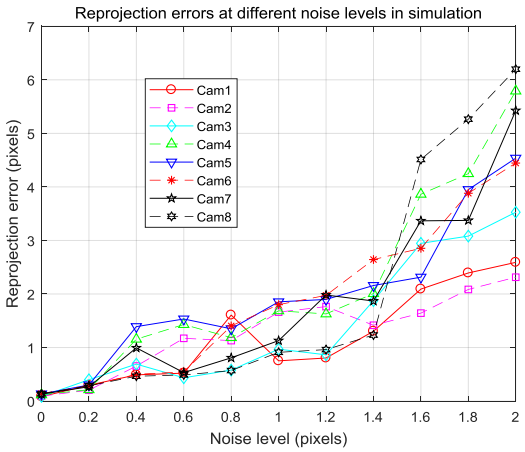


Fig. 11. Reprojection errors at different noise levels in simulation.

FOV of 77.32°. Image processing to obtain the marker points is executed using an FPGA module inside the smart cameras. The main calibration algorithm is run on a PC with Intel Core i7 processor, 3.6 GHz and 8 GB RAM. The configurations of the calibration objects are shown in [10]. Before arranging the VSN, the internal parameters of each camera needs to be obtained by intrinsic calibration in Subsection III-A.

2) Real Experiment Results: First, we demonstrate the proposed calibration method for the five-camera VSN. Results show that the calibration errors are small with the mean reconstructed error of 0.83 mm and the mean reprojection error of 0.228 pixels. The vision graph and the optimal path of the five cameras are shown in Fig. 12. Note that the reprojection error of calibration between Cam1 and Cam3 is relatively large so that the optimal path of the Cam3 includes an intermediate Cam2.

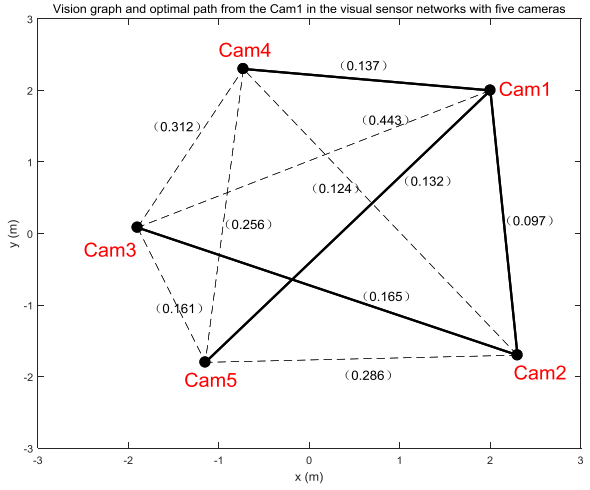


Fig. 12. Vision graph and the optimal path from the reference Cam1 obtained based on the Dijkstra's shortest path algorithm. Numbers indicate the reprojection error of the pairwise calibration of the two cameras.

Then, the effect of the total number of frames to the calibration algorithm is evaluated. Results are shown in Fig. 13 and Fig. 14. Based on these results, the calibration algorithm is accurate, and the mean reconstructed Euclidean error is within 1 mm while the mean reprojection error is smaller than 0.3 pixels. The error is acceptable and reasonable with the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSEN.2019.2958104, IEEE Sensors Journal

TABLE II
REPROJECTION ERRORS ε_R (PIXELS) OF EACH CAMERA AT DIFFERENT NOISE LEVELS IN SIMULATION.

Noise	0.2		0.6		1.0		1.4		1.8	
Method	Proposed	Fu [9]	Proposed	Fu [9]	Proposed	Fu [9]	Proposed	Fu [9]	Proposed	Fu [9]
Cam1	0.2959	0.2851	0.5217	0.5415	0.7517	1.9110	1.3041	2.6545	2.3928	3.3240
Cam2	0.2038	0.6237	1.1722	1.5193	1.6616	1.7412	1.4161	2.2650	2.0849	3.4158
Cam3	0.3986	0.2470	0.4350	0.5215	0.9692	1.9938	1.8756	2.1571	3.0811	4.2381
Cam4	0.2126	0.4984	1.4295	1.5149	1.6857	2.5088	2.0075	2.7714	4.2477	4.5202
Cam5	0.3009	0.4672	1.5329	1.5808	1.8535	1.2796	2.1584	2.7148	3.9495	4.5333
Cam6	0.2830	1.2456	0.5459	0.5742	1.8006	0.9161	2.6434	2.2214	3.8815	5.4853
Cam7	0.2677	0.2822	0.5314	0.7613	1.1291	1.0985	1.8692	2.2758	3.3746	3.5416
Cam8	0.2864	0.3191	0.4884	1.5541	0.9104	3.4591	1.2335	2.1087	5.2648	5.8616
Mean	0.2811	0.4960	0.8321	1.0710	1.3452	1.8635	1.8135	2.3961	3.5346	4.3650

TABLE III
REPROJECTION ERRORS ε_R (PIXELS) OF EACH CAMERA WITH A DIFFERENT NUMBER OF FRAMES IN REAL EXPERIMENTS.

Num	2180		3360		4844		5554		6792	
Method	Proposed	Fu [9]	Proposed	Fu [9]	Proposed	Fu [9]	Proposed	Fu [9]	Proposed	Fu [9]
Cam1	0.1776	0.3128	0.1667	0.3126	0.2107	0.6643	0.1664	0.2407	0.1768	0.3352
Cam2	0.2167	0.3216	0.223	0.3168	0.2296	0.2611	0.2347	0.2607	0.2318	0.3393
Cam3	0.192	0.3180	0.1645	0.3287	0.2129	0.2501	0.2006	0.2691	0.1855	0.2879
Cam4	0.2071	0.3229	0.1966	0.3179	0.2139	0.6838	0.2422	0.2832	0.2418	0.3290
Cam5	0.2989	0.4318	0.2364	0.4110	0.3054	0.5544	0.2855	0.3149	0.3181	0.5760
Mean	0.2185	0.3414	0.1974	0.3374	0.2345	0.4827	0.2259	0.2737	0.2308	0.3735

results of the simulation. As the number of captured images is increasing, the error does not change much and will converge to a stable value. The conclusion is consistent with that in simulation. Finally, we have compared the calibration results

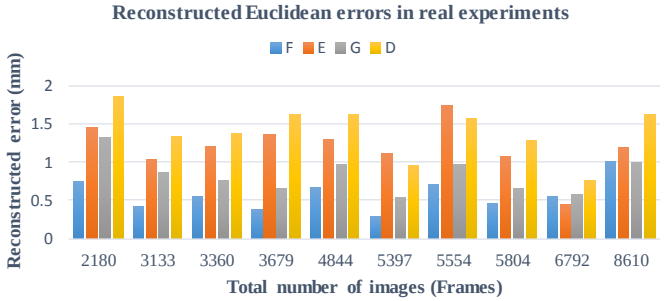


Fig. 13. Reconstructed Euclidean errors of each marker on the triangle with a different number of frames in real experiments .

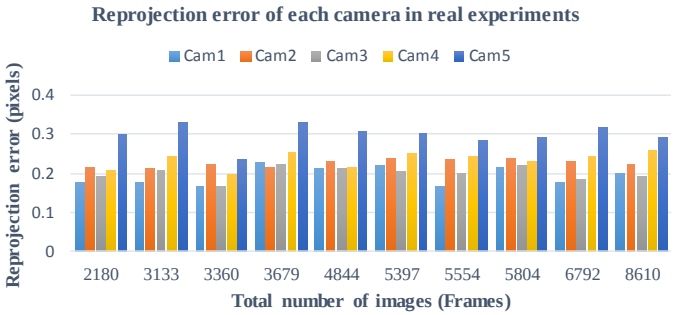


Fig. 14. Reprojection errors of each camera with a different number of frames in real experiments.

of the proposed method with the method of our previous work [9]. The reprojection errors of each camera are shown in Table III. Results show that our proposed method outperforms the previous work.

Furthermore, eight smart cameras with an image resolution of 1280×1024 pixels to cover a bigger area of $5 \times 8 \times 2.7$ m. Results shown that the calibration errors are small with the mean reconstructed error of 0.34 mm and the mean reprojection error of 0.139 pixels. Similarly, the vision graph and the optimal path of the eight cameras are depicted in Fig. 15. The top table shows the results of the adjacency matrix, and it is clear that the matrix is symmetric with the elements being positive. Fig. 15(b) shows the vision graph is based on the adjacency matrix, and the edges between two vertices are not connected if the weights are too large, such as the edges between vertex 1 and vertex 5 since the weight a_{15} is too large. The paths in thick line represent the optimal transformation path based on the vision graph and the Dijkstra's shortest path algorithm. Supplemental results are available at https://youtu.be/Eu3kRMH5n-A.

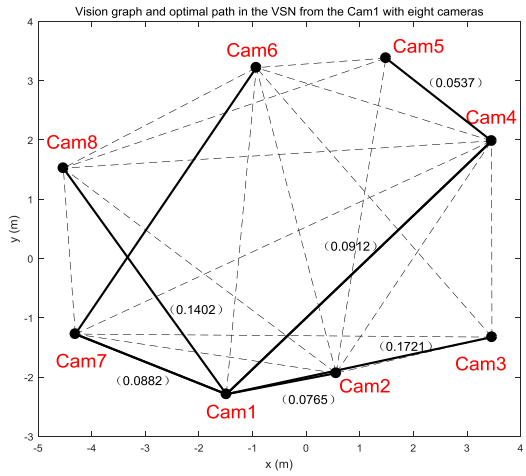
V. CONCLUSION

In this paper, an accurate and flexible calibration method for a class of visual sensor networks is proposed and implemented. The proposed method does not require all the cameras to share a common FOV, and only pairwise overlap is needed. Based on the experimental results with synthetic and real data, the feasibility and accuracy of the proposed multi-camera calibration method in the presence of noise are demonstrated. We have made some improvements based on our previous work [9]. First, infrared reflective markers (passive vision) are employed as the target points instead of LEDs (active vision), thus the detection and extraction of markers are more accurate and robust. Second, a k-means++ based selection algorithm is proposed to efficiently choose optimal matching points. Third, extensive experiments with more cameras (e.g., eight cameras in simulation) are conducted in this paper to validate the calibration algorithms. Besides, the proposed Unity3D-based virtual platform is an effective tool to test vision-based algorithms for different applications.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSEN.2019.2958104, IEEE Sensors Journal

a_{ij}	1	2	3	4	5	6	7	8
1	N/A	0.0765	0.1721	0.0912	INF	0.1959	0.0882	0.1402
2	0.0765	N/A	0.1158	0.4919	0.1259	0.1687	0.1415	0.1252
3	0.1721	0.1158	N/A	0.0931	INF	0.1139	0.0968	INF
4	0.0912	0.4919	0.0931	N/A	0.0537	0.2481	0.1581	0.0775
5	INF	0.1259	INF	0.0537	N/A	0.0768	INF	0.1062
6	0.1959	0.1687	0.1139	0.2481	0.0768	N/A	0.0937	0.3588
7	0.0882	0.1415	0.0968	0.1581	INF	0.0937	N/A	0.127
8	0.1402	0.1252	INF	0.0775	0.1062	0.3588	0.127	N/A

(a)



(b)

Fig. 15. Vision graph results of eight cameras: (a) Weighting matrix of each pairwise cameras, a_{ij} represents the reprojection of stereo calibration of the ith camera and the jth camera. "INF" means the value is too large. (b) Vision graph and the optimal path from the reference camera "Cam1" obtained from the above setup based on the Dijkstra's shortest path algorithm. Numbers indicate the reprojection error of the calibration of the two cameras.

However, the proposed method requires all the cameras are to be synchronized, and the camera internal parameters are assumed to be unchanged and need to be calibrated prior to deployment. Besides, the camera arrangement is found to make a great difference to the final results. Therefore, in future research, the multi-camera calibration with asynchronous cameras in the visual sensor network needs to be studied further. A good camera configuration is needed to extend the coverage as well as decrease the calibration error. Based on the fundamentals of this paper, it is hopeful to construct a large-scale VSN to implement more essential applications with more cameras and target objects.

REFERENCES

[1] S. Soro and W. Heinzelman, "A survey of visual sensor networks," *Advances in multimedia*, vol. 2009, 2009.

[2] B. Tavli, K. Bicakci, R. Zilan, and J. M. Barcelo-Ordinas, "A survey of visual sensor network platforms," *Multimedia Tools and Applications*, vol. 60, no. 3, pp. 689–726, 2012.

[3] A. Amjad, M. Patwary, A. Griffiths, and A.-H. Soliman, "Characterization of field-of-view for energy efficient application-aware visual sensor networks," *IEEE Sensors Journal*, vol. 16, no. 9, pp. 3109–3122, 2016.

[4] H. Aghajan and A. Cavallaro, *Multi-camera networks: principles and applications*. Academic press, 2009.

[5] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 22, no. 11, pp. 1330–1334, 2000.

[6] T.-W. Hui and R. Chung, "Determining shape and motion from monocular camera: A direct approach using normal flows," *Pattern recognition*, vol. 48, no. 2, pp. 422–437, 2015.

[7] H. Deng, U. Arif, Q. Fu, Z. Xi, Q. Quan, and K.-Y. Cai, "Visual–inertial estimation of velocity for multicopters based on vision motion constraint," *Robotics and Autonomous Systems*, vol. 107, pp. 262–279, 2018.

[8] J. Kannala and S. S. Brandt, "A generic camera model and calibration method for conventional, wide-angle, and fish-eye lenses," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 8, pp. 1335–1340, 2006.

[9] Q. Fu, Q. Quan, and K.-Y. Cai, "Calibration of multiple fish-eye cameras using a wand," *IET Computer Vision*, vol. 9, no. 3, pp. 378–389, 2014.

[10] H. Deng, Q. Fu, Q. Quan, K. Yang, and K.-Y. Cai, "Indoor Multi-Camera Based Testbed for 3D Tracking and Control of UAVs," *IEEE Transactions on Instrumentation and Measurement*, 2019.

[11] E. Shen and R. Hornsey, "Multi-camera network calibration with a non-planar target," *IEEE Sensors Journal*, vol. 11, no. 10, pp. 2356–2364, 2011.

[12] C. Ricolfe-Viala, A.-J. Sanchez-Salmeron, and A. Valera, "Efficient lens distortion correction for decoupling in calibration of wide angle lens cameras," *IEEE Sensors journal*, vol. 13, no. 2, pp. 854–863, 2012.

[13] L. Wang, F. Duan, and K. Lu, "An adaptively weighted algorithm for camera calibration with 1d objects," *Neurocomputing*, vol. 149, pp. 1552–1559, 2015.

[14] Z. Zhang, "Camera calibration with one-dimensional objects," *IEEE transactions on pattern analysis and machine intelligence*, vol. 26, no. 7, pp. 892–899, 2004.

[15] L. Wang, F. Wu, and Z. Hu, "Multi-camera calibration with one-dimensional object under general motions," in *2007 IEEE 11th International Conference on Computer Vision*, pp. 1–7, IEEE, 2007.

[16] J. A. de França, M. R. Stemmer, M. B. d. M. França, and E. G. Alves, "Revisiting Zhang's 1D calibration algorithm," *Pattern Recognition*, vol. 43, no. 3, pp. 1180–1187, 2010.

[17] J. A. De FrançA, M. R. Stemmer, M. B. d. M. França, and J. C. Piai, "A new robust algorithmic for multi-camera calibration with a 1d object under general motions without prior knowledge of any camera intrinsic parameter," *Pattern Recognition*, vol. 45, no. 10, pp. 3636–3647, 2012.

[18] G. Kurillo, Z. Li, and R. Bajcsy, "Wide-area external multi-camera calibration using vision graphs and virtual calibration object," in *2008 Second ACM/IEEE International Conference on Distributed Smart Cameras*, pp. 1–9, IEEE, 2008.

[19] R. Xia, M. Hu, J. Zhao, S. Chen, Y. Chen, and S. Fu, "Global calibration of non-overlapping cameras: State of the art," *Optik*, vol. 158, pp. 951–961, 2018.

[20] Z. Liu, G. Zhang, Z. Wei, and J. Sun, "Novel calibration method for non-overlapping multiple vision sensors based on 1d target," *Optics and Lasers in Engineering*, vol. 49, no. 4, pp. 570–577, 2011.

[21] R. Xia, M. Hu, J. Zhao, S. Chen, and Y. Chen, "Global calibration of multi-cameras with non-overlapping fields of view based on photogrammetry and reconfigurable target," *Measurement Science and Technology*, vol. 29, no. 6, p. 065005, 2018.

[22] T. Yang, Q. Zhao, X. Wang, and D. Huang, "Accurate calibration approach for non-overlapping multi-camera system," *Optics & Laser Technology*, vol. 110, pp. 78–86, 2019.

[23] J. J. Moré, "The Levenberg-Marquardt algorithm: implementation and theory," in *Numerical analysis*, pp. 105–116, Springer, 1978.

[24] M. Lourakis and A. Argyros, "The design and implementation of a generic sparse bundle adjustment software package based on the levenberg-marquardt algorithm," tech. rep., Technical Report 340, Institute of Computer Science-FORTH, Heraklion, Crete, Greece, 2004.

[25] P. Corke, *Robotics, vision and control: fundamental algorithms In MATLAB® second, completely revised*, vol. 118. Springer, 2017.

[26] D. Arthur and S. Vassilvitskii, "k-means++: The advantages of careful seeding," in *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pp. 1027–1035, Society for Industrial and Applied Mathematics, 2007.

[27] A. Lipowski and D. Lipowska, "Roulette-wheel selection via stochastic acceptance," *Physica A: Statistical Mechanics and its Applications*, vol. 391, no. 6, pp. 2193–2196, 2012.

[28] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge university press, 2003.

[29] J.-C. Chen, "Dijkstra's shortest path algorithm," *Journal of Formalized Mathematics*, vol. 15, no. 9, pp. 237–247, 2003.