IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received December 2, 2018, accepted January 25, 2019, date of publication January 31, 2019, date of current version March 1, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2896096

Robust Precise Dynamic Point Reconstruction From Multi-View

DEGUI XIAO[1], JIANFANG LI[1], AND KEQIN LI[2], (Fellow, IEEE)
[1]College of Computer Science and Electronic Engineering, Hunan University, Changsha 410082, China
[2]Department of Computer Science, The State University of New York, New Paltz, NY 12561, USA

Corresponding author: Jianfang Li (lijianfang@hnu.edu.cn)

This work was supported by the National Natural Science Foundation of China, under Grant 61272062.

ABSTRACT Reconstructing precise dynamic points with multiple camera systems (MCSs) is a pivotal work in many computer vision applications, such as motion capture. However, the deviation of 2-D position leads to frequent mismatch when searching for correspondence from multi-view. This paper puts forward a two-stage framework based on passive optical motion capture system to reconstruct precise dynamic points with MCSs. Our proposed method improves the performance of calibration and matching simultaneously. In the calibration stage, the extrinsic parameters of numerous cameras are calibrated synchronously via an L-shaped frame, where the position of four reference points is optimized with multiple geometric constraints. Bundle adjustment occurs after calibration. In the reconstruction stage, we propose a novel sparse multi-view matching method called cyclical voting, which includes multiple pairs of global voting and in-group voting. Point residual method is proposed to exclude outliers in matching groups further. The experiments show that our proposed method can decrease mismatching significantly and achieve commendable reconstruction results compared with Cortex (one of the most successful commercial motion analysis software).

INDEX TERMS Stereo vision, dynamic point reconstruction, multi-camera calibration, sparse multi-view matching method.

I. INTRODUCTION

Recovering 3D structure and motion of non-rigid objects from sets of 2D points in multi-view is a challenging task in many computer vision applications,such as animation [1], biological [2], [3], medical diagnosis [4], and robot control [5]. To perform this work, precise dynamic point reconstruction is fundamental. Dynamic point reconstruction is accomplished mainly by non-rigid structure from motion (NRSFM) [6]–[9] or multiple camera systems (MCSs) [11], [12]. However, too much additional prior knowledge leads NRSFM to result in poor robustness, so the most common ways presently remain based on MCSs in real application, like passive optical motion capture systems [11], [12]. Instead of monocular images, cameras are fixed at multiple viewpoints in MCSs, ensuring that every camera captures each configuration of non-rigid objects. The ill-posed problem in NRSFM is thus avoided. The process of a typical dynamic point reconstruction by MCSs involves two stages, namely, calibration and reconstruction. On the one hand, the calibration of cameras distinctly affects the quality of reconstruction. On the other hand, the deviation of 2D position leads to interference when searching for correspondence in the reconstruction stage, as shown in Fig. 1.

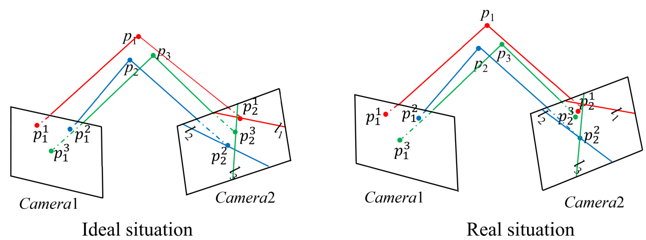


FIGURE 1. Comparison of ideal situation and real situation when searching for correspondence. p_1, p_2, and p_3 are three points in real world. p_1^1, p_1^2, p_1^3 and p_2^1, p_2^2, p_2^3 are projected points in cameras 1 and 2, respectively. l_1, l_2, and l_3 are the polar line of p_2^1, p_2^2, and p_2^3 in camera 2. The left figure shows the ideal situation, where p_2^1, p_2^2, and p_2^3 are located only on their corresponding polar line. The right figure illustrates the real situation, in which p_2^2 is a point located both on l_2 and l_3, p_2^3 and p_2^1 are located far away from their respective polar lines l_3 and l_1. In these cases, the matching of p_1^1, p_1^2 and p_1^3 is disturbed.

The associate editor coordinating the review of this manuscript and approving it for publication was Md. Moinul Hossain.

Therefore mismatching often occurs. Both reasons lead to erroneous point reconstruction and distorted model. In fact, the problem even exists in Cortex, which is one of the most successful commercial analysis and processing software of motion data. This study puts forward a framework for dynamic point reconstruction by MCSs, the framework is based on passive optical motion capture system. To lower the deviation of 2D position, markers are placed on key points, the 2D position of markers are exacted directly from images captured by MCSs. The overview of the presented framework is shown in Fig. 2. Our method improves the performance of calibration and matching at the same time. In the calibration stage, we consider multiple geometric constraints to optimize the position of calibration reference points. In the reconstruction stage, we consider multi views together instead of pairwise matching when searching for correspondence, and, thus, a sparse multi-view matching method is proposed. Our proposed approach exhibits high accuracy, without any hypothesis, and good robustness for numerous dynamic point reconstruction. Tests on standard and our own motion capture datasets demonstrate the excellence of our method.

Our study has two main contributions.

1) We propose an efficient calibration model for MCSs. Our method introduces Levenberg-Marquart(LM) algorithms [13] to take nonlinear geometric constraints into account, the result provides more accurate position of reference points for calibration. Experiments show that the treatment can improve the calibration performance comparing to Cortex.

2) We design a reconstruction model, which improves the quality of dynamic point reconstruction significantly. To search for correspondence, we propose a novel hierarchy cyclical voting (CV) method consisted by multiple global voting and in-group voting pairs. Point residual (PR) filtering strategy is then proposed to exclude outliers of matching groups during triangulation. Our approach considers all views together to correct mismatching successfully. Experiments show that our method performs well in motion capture application.

The rest of this paper is organized as follows. Section 2 introduces the related works. Section 3 discusses our calibration model for MCSs and reconstruction model in detail. Section 4 presents experiments and evaluation. Section 5 summarizes the conclusions.

II. RELATED WORK

In this section, we investigate the related work about calibration and 3D reconstruction of dynamic points based on stationary MCSs.

A. CALIBRATION OF MCSS

Calibration is the first step for most stereo reconstruction algorithms [14]. Intrinsic parameters can be read from cameras in certain situations, the challenge comes from calibrating extrinsic parameters. In general, cameras of stationary MCSs are fixed at a specified position, so many studies employ different types of calibration objects, such as markers, laser pointers, reference bars [15]–[17]. Active self-calibration provides another choice for calibration objects [18]. In select methods, extrinsic parameters are inferred by estimating the fundamental or essential matrix [19], [15], followed by bundle adjustment [20], [21]. The latter has been implemented in many types of research [22], [23]. Schneider *et al.* [24] proposed a general bundle adjustment with infinity scene points, and the process reduced the number of equations in [25] to avoid singular covariance matrices. Later, Schneider and Förstner [26] expanded his work to the calibration of extrinsic parameters. Our work is based on the theory introduced in [21], [27], and [28]. Zhang [27] proposed a classical and reliable calibration model, which has been used in Matlab and OpenCV. In his later work, Zhang [28] filled missing dimension with reference points on a line, and the method performed well especially for multiple cameras installed apart from each other.

B. 3D RECONSTRUCTION OF POINTS

Reconstructing 3D points from multi-view images is the most common method in real application presently. Higgins [29] first triangulated the position of stationary points by epipolar geometry. Later, the research on geometry makes great breakthrough in reconstructing static scenes, as summarized in [31] and [32]. The advance has wide application, including scene flow estimation [30] and motion capture [2], [3].

The real challenge comes from the 3D reconstruction of dynamic points with large displacement and fast move. Many types of research focus on dynamic point reconstruction from a series of monocular images. Avidan and Shashua [33] first proposed the term called trajectory triangulation, the research demonstrated that if a point moved along a straight line or a conic section, then reconstructing the point was possible. Enlightened by the work of Avidan and Shashua [33], Shashua and Wolf [34] demonstrated that the reconstruction of points moving along a polygon could be realized. Later, Kaminski *et al.* [35] introduced a polynomial representation to reconstruct dynamic points moving along the general trajectory. NRSFM is another research hotspot to reconstruct dynamic points from monocular images. The principal work was published by Bregler *et al*. [36]. They used linear shape models to represent non-rigid 3D structures, and the results showed the fitness within the factorization-based reconstruction paradigm in [37]. In subsequent research, e.g., [6], [8], To overcome inherent ambiguity of the non-rigid problem [10], substantial constraints and prior information were added for specified shape models. The shape models were used to represent facial expressions and the human body. However, these additional assumptions lead to difficulty in coping with complex movement.

Dynamic point reconstruction with MCSs has been proven to be an efficient method, and its core work is stereo matching. Most stereo matching algorithms generate disparity map by measuring the difference between pixels and patches in

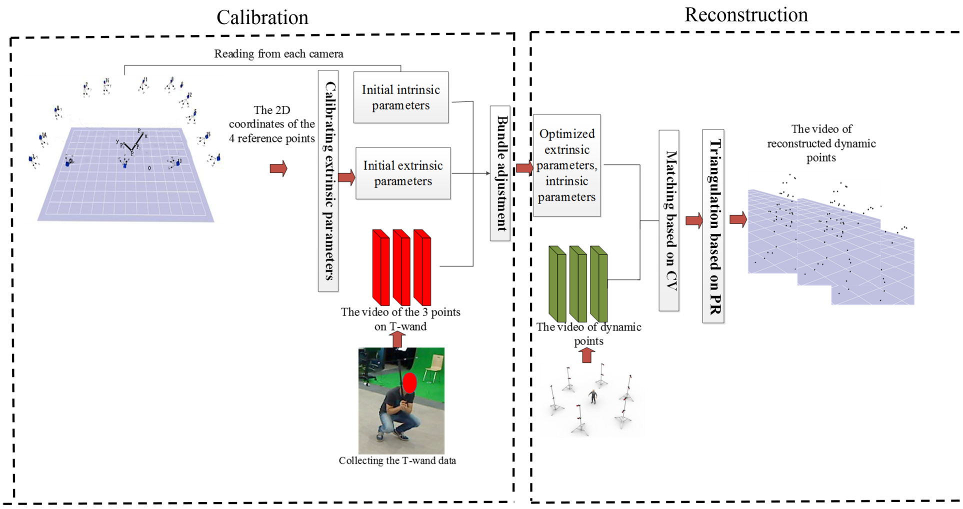


FIGURE 2. Overview of our framework.

multiple images, most literatures divide stereo matching algorithms into local and global methods [38]. With the development of deep learning, modern stereo matching employs CNN to predict disparity map [39], [40]. All these methods focus on feature point matching, which are used mainly for dense reconstruction.

In some situation, only the 2D positions are available, like passive optical motion capture system. In this case, only epipolar constraints are effective. Although epipolar geometric has been well developed in the application of pairwise matching [29], [32], but mismatching still happens very often even in commercial software [11], [12]. Considering all these related researches, our reconstruction model is based on the theory of epipolar geometry [29], [32].

III. PROPOSED FRAMEWORK

This study puts forward a framework for dynamic point reconstruction with MCSs, as shown in Fig. 2. The framework is divided into a calibration model and a reconstruction model. First, an L-shaped frame is placed in the center of MCSs to determine initial extrinsic parameters, and a T-wand is waved in the venues surrounded by MCSs. The video from each camera is then collected for bundle adjustment. After the preliminary work, 2D motion datasets are collected to reconstruct the dynamic points.

A. CALIBRATION MODEL

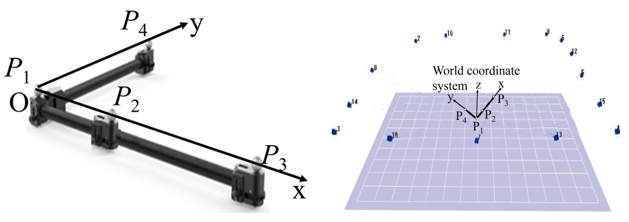
In stationary MCSs, cameras are fixed at a specific position before reconstruction, the intrinsic and extrinsic parameters of all the cameras must be calibrated as accurately as possible. Our presented calibration method is based on the calibration of passive optical motion capture system. The entire calibration process includes five steps: 1) determining the coordinates of the four reference points on an L-shaped frame in each camera coordinate system, 2) optimizing positions of the four reference points, 3) calculating the rotation parameters, 4) inferring the transformation parameters inversely, and 5) optimizing camera parameters by bundle adjustment.

Initial intrinsic parameters is read from cameras directly. (u_0, v_0) is the translation vector between the 2D points in the image plane and 2D points in the image; dx and dy are the change of units $(\frac{mm}{pixels})$ in the x and y axes of the image plane, respectively; f is the focus length, and $\mathbf{k} = [k_1, k_2]^T$ is the distortion coefficient, which is calculated according the calibration method proposed by Zhang [27]. In our study, all initial intrinsic parameters except for $\mathbf{k}$ are read from cameras directly. The extrinsic parameters are denoted as $exI = (t_x, t_y, t_z, r_x, r_y, r_z)$, where we denote $\mathbf{t} = [t_x, t_y, t_z]^T$ as the translation parameters and $\mathbf{r} = [r_x, r_y, r_z]^T$ as the rotation parameters of a camera.

1) DETERMINING THE COORDINATES OF REFERENCE POINTS IN TWO COORDINATE SYSTEMS

The initial extrinsic parameters are determined by the geometric relationship of four reference points on an L-shaped frame. Thus, the accuracy of position of reference points is crucial. As shown in Fig. 3, P_1, P_2, P_3 and P_4 represent the four reference points, respectively. The world coordinate system is established based on the right-hand coordinate system, where P_1P_4 is the x-axis, P_1P_3 is the y-axis, and the axis passing through P_1 and perpendicular to the plane of the L-shaped frame is z axis. In the world coordinate system, the coordinates of the four reference points are $P_1(0, 0, 0)$, $P_2(200, 0, 0)$, $P_3(600, 0, 0)$ and $P_4(0,400,0)$, and P_1 is the origin of the world coordinates system.

$P_{wi}(x_{wi}, y_{wi}, z_{wi})(i = 1, 2, 3, 4)$ represents the coordinates of point P_i in the camera coordinate system, and $P_{ci}(x_{ci}, y_{ci}, z_{ci})$ represents the projections of P_{wi} on the normalized image plane $(z = 1)$. Let $p_i(u_i, v_i)$ represent the pixel coordinate of the i^{th} reference point in a camera; thus, $P_{ci}(x_{ci}, y_{ci}, 1)$ can be easily calculated according to the



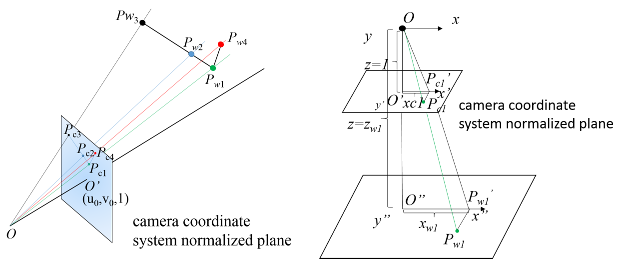
(a) (b)

FIGURE 3. **The presentation of the L-shaped frame. (a)Geometric relationship between the four reference points on the L-shaped frame, where $P_1P_4 = 400\ mm$, $P_1P_2 = 200\ mm$, $P_1P_3 = 600\ mm$, $P_2P_3 = 400\ mm$, and P_1P_3 is perpendicular to P_1P_4. In order to solve the 6 extrinsic parameters steadily, we chose four points. If we use 3 points only, these are 6 equations corresponding to 6 extrinsic parameters, obviously, that is unpractical in real application. Furthermore, 5 points will bring addition computation and limited benefit. (b)The world coordinate system determined by L-shaped frame. The L-shaped frame make it easy for us to determine the word coordinates system and the coordinate of the four reference points in world coordinates system. Our world coordinate system is established according to the right-hand coordinate system in MCSs.**

intrinsic parameters and p_i, shown as Eq. (1),

$$\begin{bmatrix} x_{ci} \\ y_{ci} \\ 1 \end{bmatrix} = \begin{bmatrix} dx & 0 & u_0 \\ 0 & dy & v_0 \\ 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix}. \tag{1}$$

At first, $P_{wi}(x_{wi}, y_{wi}, z_{wi})(i = 1, 2, 3)$ are calculated according to constrain 1, constrain 2.



(a) (b)

FIGURE 4. **Geometric relationship between the four referent points and their projections. (a)Projection of points P_{w1}, P_{w2}, P_{w3} and P_{w4} on normalized plane ($z = 1$) in camera coordinate system. OO' are the optical axes. (b) Detailed description of the projection relationship (taking P_{w1} as example). $x'O'y'$ is a normalized plane ($z = 1$), $x''O''y''$ is a plane that passes through P_{w1}, and parallel to the normalized plane. P'_{c1} and P'_{w1} are the foot of perpendicular from P_{c1} to x' and P_{w1} to x''.**

Constraint 1: As shown in Fig. 4(b), Eq. (2) can be derived according to similar triangle theorem,

$$\begin{aligned} \frac{x_{w1}}{x_{c1}} &= \frac{y_{w1}}{y_{c1}} = \frac{z_{w1}}{1}, \\ \frac{x_{w2}}{x_{c2}} &= \frac{y_{w2}}{y_{c2}} = \frac{z_{w2}}{1}, \\ \frac{x_{w3}}{x_{c3}} &= \frac{y_{w3}}{y_{c3}} = \frac{z_{w3}}{1}. \end{aligned} \tag{2}$$

Constraint 2: as shown in Fig. 5, $P'_{wi}(i = 1, 2, 3)$ is the projection from $P_{wi}(i = 1, 2, 3)$ to x axes, $P_{w1}{}'P'_{w2} = x_{w1} - x_{w2}$, $P'_{w1}P'_{w3} = x_{w1} - x_{w3}$, $P_{w1}P_{w2} = P_1P_2 = 200\ mm$,

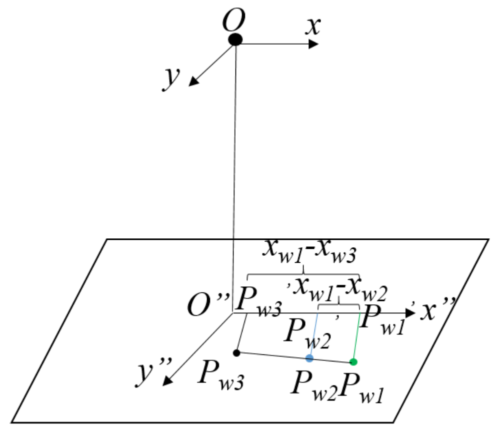


FIGURE 5. **The proportion relationship of similar polygons.**

$P_{w1}P_{w3} = P_1P_3 = 600\ mm$. According to the proportion relationship of similar polygons, $\frac{P'_{w1}P'_{w2}}{P'_{w1}P'_{w3}} = \frac{x_{w1}-x_{w2}}{x_{w1}-x_{w3}} = \frac{P_{w1}P_{w2}}{P_{w1}P_{w3}} = \frac{200}{600} = \frac{1}{3}$. Similar conclusion can be obtained as shown in Eq. (3),

$$\frac{x_{w1} - x_{w2}}{x_{w1} - x_{w3}} = \frac{1}{3}; \quad \frac{y_{w1} - y_{w2}}{y_{w1} - y_{w3}} = \frac{1}{3}; \quad \frac{z_{w1} - z_{w2}}{z_{w1} - z_{w3}} = \frac{1}{3}; \tag{3}$$

Transforming Eq. (2) and (3) together into a linear equation set, as shown in Eq. (4),

$$\begin{aligned} x_{w1} - x_{c1}z_{w1} &= 0; \quad y_{w1} - y_{c1}z_{w1} = 0, \\ x_{w2} - x_{c2}z_{w2} &= 0; \quad y_{w2} - y_{c2}z_{w2} = 0, \\ x_{w3} - x_{c3}z_{w3} &= 0; \quad y_{w3} - y_{c3}z_{w3} = 0, \\ 2x_{w1} - 3x_{w2} + x_{w3} &= 0, \\ 2y_{w1} - 3y_{w2} + y_{w3} &= 0, \\ 2z_{w1} - 3z_{w2} + z_{w3} &= 0. \end{aligned} \tag{4}$$

$P_{wi}(x_{wi}, y_{wi}, z_{wi})(i = 1, 2, 3)$ can be solved by SVD decomposition. P_{w4} is located on the ray OP_{c4}, shown as Fig. 3(a), according to the geometric relationship between P_{w4} and $P_{w1}P_{w2}$, the point on the ray OP_{c4} satisfying the following conditions is chosen as P_{w4}: 1) the length of $P_{w4}P_{w1}$ equals $400\ mm$, and 2) line $P_{w4}P_{w1}$ is perpendicular to line $P_{w1}P_{w2}$.

2) OPTIMIZING COORDINATES OF THE REFERENCE POINTS

Many nonlinear constraints are not considered in the above calculation, and, as such, the coordinates of the four reference points are not very accurate. The following constraints 3 to 6 are used to optimize the coordinates of the four reference points on the L-shaped frame:

Constraint 3: The lengths of $P_{w1}P_{w2}$, $P_{w1}P_{w3}$, $P_{w1}P_{w4}$, $P_{w2}P_{w3}$, $P_{w2}P_{w4}$, and $P_{w3}P_{w4}$.

Constraint 4: The reference points are located on the ray OP_1, OP_2, OP_3, and OP_4.

Constraint 5: P_1, P_2, and P_3 are proportional and collinear.

Constraint 6: P_4 is perpendicular to $P_{w1}P_{w2}$, $P_{w2}P_{w3}$, and $P_{w1}P_{w3}$. Equations formed by the above constraints are set as objective function, and $P_{wi}(x_{wi}, y_{wi}, z_{wi})(i = 1, 2, 3, 4)$ are optimized with LM algorithm.

3) CALCULATING THE ROTATION PARAMETERS

Let **R** represent the rotation matrix from the world coordinate system to the camera coordinate system. **R** can be written as the form of Eq. (5),

$$\mathbf{R} = \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix}. \tag{5}$$

R can also be represented as the form of r_x, r_y, r_z, as shown in Eq. (6):

$$\begin{aligned} R_{11} &= cos(r_y)cos(r_z) \\ R_{12} &= cos(r_y)sin(r_z) \\ R_{13} &= -sin(r_y) \\ R_{21} &= sin(r_x)sin(r_y)cos(r_z) - cos(r_x)sin(r_z) \\ R_{22} &= sin(r_x)sin(r_y)sin(r_z) + cos(r_x)cos(r_z) \\ R_{23} &= sin(r_x)cos(r_y) \\ R_{31} &= cos(r_x)sin(r_y)cos(r_z) + sin(r_x)sin(r_z) \\ R_{32} &= cos(r_x)sin(r_y)sin(r_z) - sin(r_x)cos(r_z) \\ R_{33} &= cos(rx)cos(ry) \end{aligned} \tag{6}$$

where $\mathbf{r} = [r_x, r_y, r_z]^T$ is the rotation parameters. The L-shaped frame is then translated to the position where P_1 coincides with the origin of camera coordinate system. The new coordinates of P_1, P_2, P_3 and P_4 are shown in Eq. (7),

$$\begin{aligned} P_{wc1}(x_{wc1}, y_{wc1}, z_{wc1}) &= P_{w1} - P_{w1}, \\ P_{wc2}(x_{wc2}, y_{wc2}, z_{wc2}) &= P_{w2} - P_{w1}, \\ P_{wc3}(x_{wc3}, y_{wc3}, z_{wc3}) &= P_{w3} - P_{w1}, \\ P_{wc4}(x_{wc4}, y_{wc4}, z_{wc4}) &= P_{w4} - P_{w1}. \end{aligned} \tag{7}$$

The relationship between $P_{wci}(x_{wci}, y_{wci}, z_{wci})$ and $P_{ci}(i = 2, 3, 4)$ on the normalized plane($z = 1$) is expressed as Eq. (8),

$$\begin{bmatrix} x_{ci} \\ y_{ci} \\ 1 \end{bmatrix} = \mathbf{R} \begin{bmatrix} x_{wci} \\ y_{wci} \\ z_{wci} \end{bmatrix}. \tag{8}$$

Let $\mathbf{S} = \begin{bmatrix} 0 & -c & -b \\ c & 0 & -a \\ b & a & 0 \end{bmatrix}$ represent an anti-symmetric matrix, where a, b, and c are independent of each other. According to the properties of anti-symmetric matrix and Rodriguez matrix in [41], $\mathbf{R} = (\mathbf{I} + \mathbf{S})(\mathbf{I} - \mathbf{S})^{-1}$, and **R** can be denoted as the form of a, b, and c, as shown in Eq. (9),

$$\mathbf{R} = \begin{bmatrix} \frac{1+a^2-b^2-c^2}{1+a^2+b^2+c^2} & \frac{-2c-2b}{1+a^2+b^2+c^2} & \frac{-2b+2ac}{1+a^2+b^2+c^2} \\ \frac{2c-2ab}{1+a^2+b^2+c^2} & \frac{1-a^2+b^2-c^2}{1+a^2+b^2+c^2} & \frac{-2a-2bc}{1+a^2+b^2+c^2} \\ \frac{2b+2ac}{1+a^2+b^2+c^2} & \frac{2a-2bc}{1+a^2+b^2+c^2} & \frac{1-a^2-b^2+c^2}{1+a^2+b^2+c^2} \end{bmatrix}. \tag{9}$$

At the same time, Eq. (8) can also be written as the form of Eq. (10),

$$\begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} = (\mathbf{I} + \mathbf{S})(\mathbf{I} - \mathbf{S})^{-1} \begin{bmatrix} x_{wci} \\ y_{wci} \\ z_{wci} \end{bmatrix}. \tag{10}$$

Substituting **S** with its full form and multiply with $(\mathbf{I} - \mathbf{S})$ on both sides of Eq. (10), then Eq. (10) can be written as Eq. (11),

$$\begin{bmatrix} 1 & c & b \\ -c & 1 & a \\ -b & -a & 1 \end{bmatrix} = \begin{bmatrix} 1 & -c & -b \\ c & 1 & -a \\ b & a & 1 \end{bmatrix} \begin{bmatrix} x_{wci} \\ y_{wci} \\ z_{wci} \end{bmatrix}. \tag{11}$$

Eq. (11) can be simplified as Eq. (12),

$$\begin{bmatrix} 0 & z_i + z_{wci} & y_i + y_{wci} \\ z_i + z_{wci} & 0 & x_i + x_{wci} \\ y_i + y_{wci} & x_i + x_{wci} & 0 \end{bmatrix} = \begin{bmatrix} a \\ b \\ c \end{bmatrix} \begin{bmatrix} x_{wci} - x_i \\ y_{wci} - y_i \\ z_{wci} - z_i \end{bmatrix}. \tag{12}$$

Thus the value of a, b, c are calculated by Eq. (13),

$$\begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} 0 & z_i + z_{wci} & y_i + y_{wci} \\ z_i + z_{wci} & 0 & x_i + x_{wci} \\ y_i + y_{wci} & x_i + x_{wci} & 0 \end{bmatrix}^{-1} \begin{bmatrix} x_{wci} - x_i \\ y_{wci} - y_i \\ z_{wci} - z_i \end{bmatrix}. \tag{13}$$

Given $P_{wci}(x_{wci}, y_{wci}, z_{wci})$ and $P_i(x_i, y_i, z_i)$, $(i = 1, 3, 3, 4)$, then the value of a, b, c are obtained by Householder orthogonal decomposition, and **R** is calculated according to Eq. (9).

According to Eq. (6), rotation parameters r_x and r_y are calculated by inverse trigonometric function, as shown in Eq. (14),

$$\begin{aligned} r_y &= -arcsin(R_{13}), \\ r_x &= -arccos(R_{33}/cos(r_y)). \end{aligned} \tag{14}$$

As cameras always face up to and look down at objects, r_x is always greater than 0. If the sign of $sin(r_x)cos(r_y)$ is different from that of R_{23}, r_y should be added or subtracted by π, thus r_x needs to be resolved with the adjusted r_y. The calculation of r_z is according to $r_z = -arcsin(R_{12}/cos(r_y))$, and the sign of r_z should be verified by similar means.

4) INFERRING THE TRANSLATION PARAMETERS INVERSELY

The translation parameters are greatly influenced by the deviation of pixel plane. Considering that the rotation parameters has high accuracy, the translation parameters are inferred inversely by the rotation matrix **R**. Let $\mathbf{R} = [\mathbf{R_1}\ \mathbf{R_2}\ \mathbf{R_3}]$, then the projection relationship from P_i to P_{ci} is shown as Eq. (15):

$$\lambda \begin{bmatrix} x_{ci} \\ y_{ci} \\ 1 \end{bmatrix} = \mathbf{R} \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} + \begin{bmatrix} t'_x \\ t'_y \\ t'_z \end{bmatrix} \tag{15}$$

where λ is the scale factor, and $\mathbf{t}' = [t'_x\ t'_y\ t'_z]^T$ denotes the translation parameters from word coordinate system to camera coordinate system. Eq. (16) can be obtained after eliminating λ,

$$\begin{aligned} t'_x - x_{ci}t'_z &= (\mathbf{R_1}^T - x_{ci}\mathbf{R_3}^T)\begin{bmatrix} x_i & y_i & z_i \end{bmatrix}^T, \\ t'_y - y_{ci}t'_z &= (\mathbf{R_2}^T - y_{ci}\mathbf{R_3}^T)\begin{bmatrix} x_i & y_i & z_i \end{bmatrix}^T. \end{aligned} \quad (16)$$

Substituting P_i and P_{ci} into Eq. (16), the approximate solution of $\mathbf{t}' = \begin{bmatrix} t'_x & t'_y & t'_z \end{bmatrix}^T$ is calculated by least-squares method. However, the standard translation parameters $\mathbf{t} = [t_x, t_y, t_z]^T$ is from camera coordinate system to word coordinate system, and $\mathbf{t}'$ can be transformed to $\mathbf{t}$ by multiplying $\mathbf{R}^T$, as shown in Eq. (17),

$$\begin{bmatrix} t_x & t_y & t_z \end{bmatrix}^T = \mathbf{R}^T \begin{bmatrix} t'_x \\ t'_y \\ t'_z \end{bmatrix}. \quad (17)$$

5) BUNDLE ADJUSTMENT

Bundle adjustment is used to optimize the intrinsic and extrinsic parameters of all cameras. After filtering out the valid wand data, the core of the bundle adjustment is to design the objective functions. In this study, a T-wand is introduced in bundle adjustment, as shown in Fig. 6, where $T_1T_2 = 200\ mm$, $T_2T_3 = 300\ mm$ and $T_1T_3 = 500\ mm$. Each camera collects the video by waving the T-wand in the field surrounded by MCSs. The purpose of our object function is to minimize two errors: 1)the error between the actual position and the re-projected position of T_1, T_2 and T_3; 2)the error of Euclidean distance between reconstructed 3D points T_{1_3D}, T_{2_3D} and T_{3_3D}. Let x_{nij} and x'_{nij} represent the actual coordinate and re-projected coordinate of $T_i (i = 1, 2, 3)$, which is recorded in n^{th} frame of j^{th} camera, respectively. The optimized intrinsic and extrinsic parameters should satisfy the following objective functions in Eq. (18):

$$\begin{aligned} \min & \sum_{n=1}^{frameN} \sum_{i=1}^{3} \sum_{j=1}^{camN} ||x_{nij} - x'_{nij}|| \\ \text{st } & T_{1\_3D}T_{2\_3D} = 200 \\ & T_{2\_3D}T_{3\_3D} = 300 \\ & T_{1\_3D}T_{3\_3D} = 500 \end{aligned} \quad (18)$$

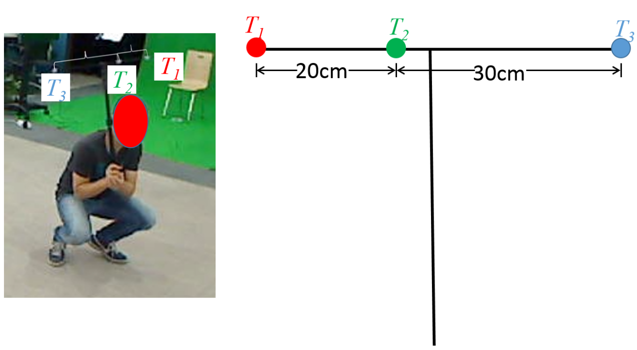


FIGURE 6. Collecting T-wand data for bundle adjustment.

where *frameN* denotes the total frames of the wand data, and *camN* denotes the total number of cameras. Initial camera parameters are optimized altogether using LM algorithm, the flow chat is shown as Fig. 7.

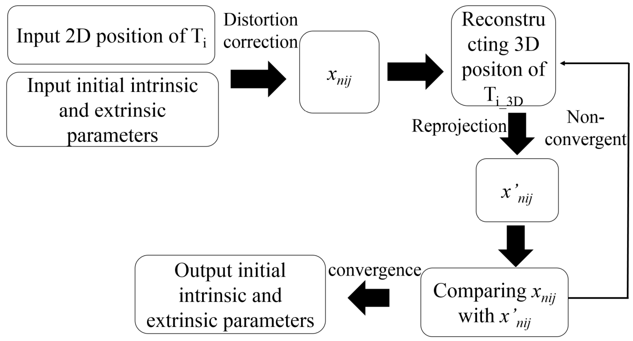


FIGURE 7. Flow chart of the bundle adjustment.

B. RECONSTRUCTION MODEL

Suppose there are n sets of 2D observation of dynamic points from n cameras, denoted as $\mathbf{C_1}, \mathbf{C_2}, \cdots, \mathbf{C_n}, \mathbf{C_i} = \{C_iP_1, C_iP_2, \cdots, C_iP_j, \cdots, C_iP_{k_i}\}$, where n represents the total number of cameras, C_iP_j denotes the j^{th} point in i^{th} camera, and k_i denotes the total number of points in i^{th} camera. Let $\mathbf{S} = \{\mathbf{S_1}, \mathbf{S_2}, \cdots, \mathbf{S_r}, \cdots, \mathbf{S_w}\}$ represent the set of matching groups, where $\mathbf{S_r}$ is the set of 2D observations of the r^{th} dynamic point in MCSs, and w is total number of dynamic points. Our purpose is to assign every 2D point C_iP_j to its corresponding matching group $\mathbf{S_r}$, and finally calculate the 3D coordinates of all dynamic points from set $\mathbf{S}$. Our proposed reconstruction method is based on a rigorous matching process, as shown in Fig. 8, which includes three stages: 1) coarse matching by determination of candidate points, 2) refined matching based on Cyclical Voting(CV), and 3) calculating the 3D coordinates. The code can be found in "https://github.com/Lijianfang6930/Robust-Precise-Dynamic-Point-Reconstruction-from-Multi-view."

1) COARSE MATCHING BY DETERMINING CANDIDATE CORRESPONDING POINTS

Coarse matching is accomplished by pairwise matching between points in different cameras, and the purpose is determining the candidate corresponding points for each single point. Let $(u_{i_1j_1}, v_{i_1j_1}, 1)$ and $(u_{i_2j_2}, v_{i_2j_2}, 1)$ represent the homogeneous coordinates of $C_{i_1}P_{j_1}$ and $C_{i_2}P_{j_2}$ on pixel plane, respectively. $\mathbf{F_{12}}$ represents the fundamental matrix from C_{i_1} to C_{i_2}, and l_0 represents the polar line of $C_{i_1}P_{j_1}$ from camera i_1 to camera i_2. According to epipolar geometry, point $C_{i_2}P_{j_2}$ is located on line l_0; thus, we obtain Eq. (19),

$$\begin{bmatrix} u_{i_2j_2} & v_{i_2j_2} & 1 \end{bmatrix} \mathbf{F_{12}} \begin{bmatrix} u_{i_1j_1} & v_{i_1j_1} & 1 \end{bmatrix}^T = 0. \quad (19)$$

In reality, point $C_{i_2}P_{j_2}$ is usually located near line l_0, sometimes even far away from l_0; therefore, bipolar constraint is introduced to determine the search area by a threshold θ,

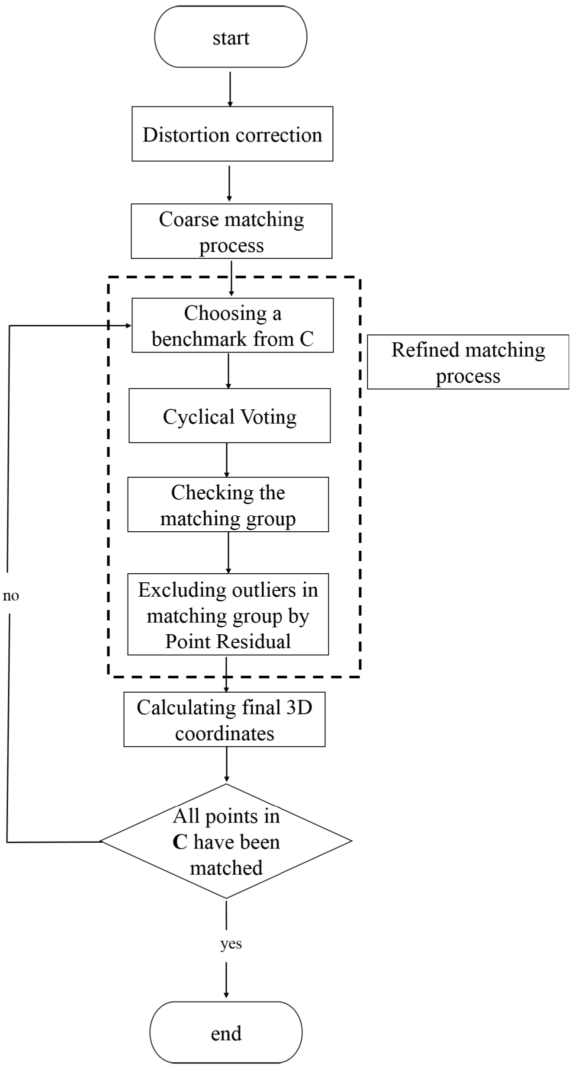


FIGURE 8. Flow chart of the proposed reconstruction method.

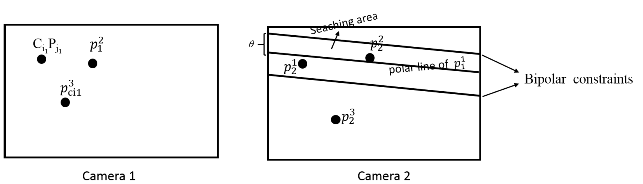
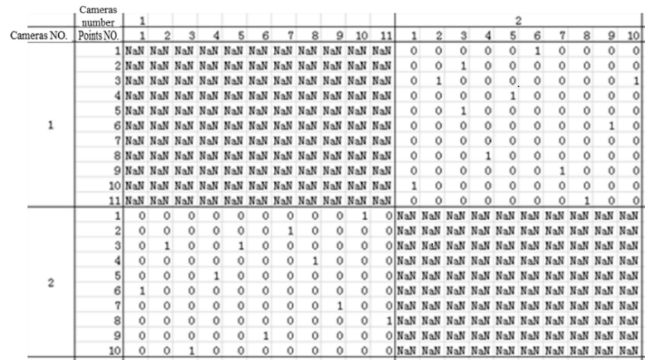


FIGURE 9. Bipolar constraint. A searching area is constrained by l_1 and l_2, and points within the area are considered as candidate corresponding points of $C_{i_1}P_{j_1}$.

as shown in Fig. 9. For any point $C_{i_2}P_{j_3}(u_{i_2j_3}, v_{i_2j_3})$ in C_{i_2}, if the distance d from $C_{i_2}P_{j_3}$ to line l_0 satisfies Eq. (20):

$$d = \left| \frac{\left[u_{i_2j_3}\ v_{i_2j_3}\ 1\right]\left[L_1\ L_2\ L_3\right]^T}{\sqrt{L_1^2 + L_1^2}} \right| \leq \theta \qquad (20)$$

where $\left[L_1\ L_2\ L_3\right]^T = \mathbf{F_{12}}\left[u_{i_1j_1}\ v_{i_1j_1}\ 1\right]^T$, then $C_{i_2}P_{j_3}$ is a candidate corresponding point of $C_{i_1}P_{j_1}$.



	Cameras number	1											2									
Cameras NO.	Points NO.	1	2	3	4	5	6	7	8	9	10	11	1	2	3	4	5	6	7	8	9	10
	1	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0	0	0	0	0	1	0	0	0	0
	2	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0	0	1	0	0	0	0	0	0	0
	3	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0	1	0	0	0	0	0	0	0	1
	4	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0	0	0	0	1	0	0	0	0	0
	5	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0	0	1	0	0	0	0	0	0	0
1	6	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0	0	0	0	0	0	0	0	1	0
	7	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0	0	0	0	0	0	0	0	0	0
	8	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0	0	0	1	0	0	0	0	0	0
	9	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0	0	0	0	0	0	1	0	0	0
	10	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	1	0	0	0	0	0	0	0	0	0
	11	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0	0	0	0	0	0	0	1	0	0
	1	0	0	0	0	0	0	0	0	0	1	0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
	2	0	0	0	0	0	0	1	0	0	0	0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
	3	0	1	0	0	1	0	0	0	0	0	0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
	4	0	0	0	0	0	0	0	1	0	0	0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
	5	0	0	0	1	0	0	0	0	0	0	0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2	6	1	0	0	0	0	0	0	0	0	0	0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
	7	0	0	0	0	0	0	0	0	1	0	0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
	8	0	0	0	0	0	0	0	0	0	0	1	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
	9	0	0	0	0	0	1	0	0	0	0	0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
	10	0	0	1	0	0	0	0	0	0	0	0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

FIGURE 10. Storage form of TP for 2 cameras.

In this study, we introduce a 0-1 matrix to represent the corresponding relation for any pair of points in pairwise matching. If a point is filtered out by bipolar constraint, then it is marked as 1, otherwise, it is recorded as 0. The matrix is denoted as **TP**, whose storage form is shown as Fig. 10. **TP** records the camera number and point number of the candidate corresponding point for any point C_iP_j. Our subsequent matching process is all based on **TP**, and it greatly facilitates the retrieval of candidate correspondence.

2) REFINED MATCHING PROCESS BASED ON CYCLICAL VOTING

The objective of matching is to sign every point to a specific matching group, where the points are the 2D observation of the same 3D dynamic in multiple views. If we ignore the noise and interference among numerous points, it is a simple task by epipolar geometry in pairwise matching situation, and the process of course matching is enough. But noise and interference may cause significant mismatching in reality application, as shown in Fig. 11. To address the problem, we design a refined matching process, which considers all views together when searching for a pair correspondence. Our designed matching algorithm can decrease mismatching significantly comparing to Cortex, and can be generalized to engineering application too.

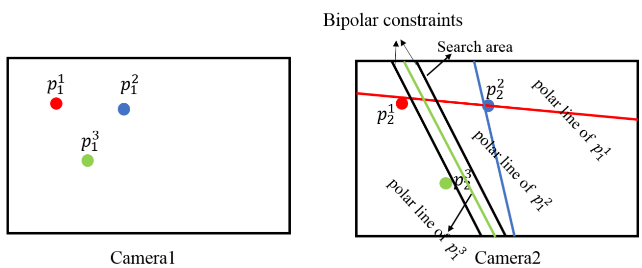


FIGURE 11. Issue of pairwise matching. Usually, p_{21}, p_{22} and p_{23} in Camera 2 are the corresponding points of p_{11}, p_{12}, p_{13} in Camera 1, respectively. In real situation, the following mismatching may exist: 1) Both p_{22} and p_{21} are the candidate corresponding points of p_{11}, but only p_{21} is the right one. 2) p_{23} is outside of the searching scope and far away from the polar line of p_{13}, so p_{23} is not chosen as the candidate corresponding point of p_{13}.

Algorithm 1 The Matching Process Based on Cyclical Voting

```
Input: TP, ipa
Output: S
r = 1
/☆ Traverse all points in TP ☆/
for ipa = 1 : m do
    if P_ipa has been matched then
        continue
    end if
    Determine the initial S_r
    if length(S_r) ≤ 2 then
        continue
    end if
    while S_r is not stable do
        /☆ Global Voting: traverse every point in TP ☆/
        if the votes of a point in TP is over half of total
        number of FPs then
            Incorporate the point into S_r
        end if
        /☆ In-group Voting: traverse every point in S_r ☆/
        if the votes of a point in S_r is less than half of total
        number of FPs then
            Kick the point out of S_r
        end if
        Deal with the situation that two or more points
        belong to the same camera in S_r.
    end while
    Mark the points in S_r as matched in TP
    r = r + 1
end for
```

In an ideal situation, points in the same matching group are corresponded to each other. If a point belongs to a specified matching group, then it must correspond to the majority points in the matching group. According to this idea, points in the matching group are set as fiducial points (*FPs*) in every step. If a point outside the matching group obtains majority votes from *FPs*, then the point is added into the matching group. For the definition of voting in this study, if point a is *FP*, and point b is a candidate corresponding point of point a, then point b receives a vote from *FP*. At the same time, if a point within the matching group receives majority votes from *FPs*, then the point is retained in the matching group; otherwise, it will be kicked out of the matching group. Algorithm 1 shows the matching process for a single frame in MCS, where **TP** is a 0-1 matrix of $m \times m$, $m = \sum_{i=1}^{n} k_i$ is the total number of points in all cameras, and k_i denotes the total number of points in i^{th} camera. *ipa* represents the serial number of points from 1 to m in **TP**. Our purpose is to assign every point C_iP_j to its corresponding matching group $\mathbf{S_r}$. Algorithm 2 describes how to determine initial $\mathbf{S_r}$, where **pp** is a set of candidate corresponding points of P_{ipa} in **TP**, **tcp** is a set of points that receives two votes from $pp(i)$ and P_{ipa} in **TP**.

Algorithm 2 Determine the Initial $\mathbf{S_r}$

```
Input: TP, ipa
Output: initial S_r
Add P_ipa into S_r
if pp == [] then
    return S_r
else
    for i = 1:length(pp) do
        if tcp ==[] then
            continue
        else
            Deal with the situation that two or more points
            belong to the same camera in tcp;
            Add point pp(i) and points in tcp into S_r
        end if
    end for
end if
Delete the repetitive points in S_r
if votes of a point in S_r is less than 2/3 of total number of
FPs then
    Kick the point out S_r
end if
return S_r
```

Later, we describe our method based on an instance including 15 cameras and 40 dynamic points, each camera captures 3600-frames motion capture data. Here, C_1P_7 in the 60^{th} frame is chosen as the initial *FP* randomly. We must find a matching group $\mathbf{S_r}$ containing the 2D observations of the r^{th} dynamic point.

TABLE 1. Candidate corresponded points of C_1P_7.

Camera NO.	**2**	4	5	6	7	8	10	10	11	11	**13**	15
Point NO.	**10**	7	7	5	7	8	5	7	7	11	**8**	6

TABLE 2. Candidate corresponded points of C_2P_{10}.

Camera NO.	**1**	3	4	5	6	7	9	10	11	12	**13**	14
Point NO.	**7**	11	8	10	10	8	10	8	2	9	**8**	9

TABLE 3. The initial matching group $\mathbf{S_r}$.

Camera NO.	1	**2**	**13**
Point NO.	7	**10**	**8**

The first step is determining the initial matching group $\mathbf{S_r}$. Twelve candidate corresponding points of C_1P_7 can be found in **TP**, as shown in Table 1. Only C_2P_{10} have two or more candidate corresponding points that similar with Table 1, as shown in Table 2. The intersection of Table 1 and 2 are selected as the initial points in matching group $\mathbf{S_r}$, as shown in Table 3.

TABLE 4. The updated $\mathbf{S_r}$ after the first round global voting.

Camera NO.	1	2	3	**4**	**6**	**9**	**12**	**13**	**14**
Point NO.	7	10	11	**8**	**10**	**10**	**9**	**8**	**9**

The second step is determining the final matching group by CV, which includes multiple rounds of global voting and in-group voting. In the first round of global voting, if votes of the point P_{ipa} in **TP** is greater than a certain value, which is determined as half of the total points in $\mathbf{S_r}$, then P_{ipa} will be added into matching group P_{ipa}. Here, votes of 6 points are more than half of the total number of *FPs*, as shown in Table 4. In the first round of in-group voting, if the votes of a point in $\mathbf{S_r}$ are less than half of the total number of *FPs*, then the point is kicked out of $\mathbf{S_r}$. Except for points C_2P_{10} and $C_{13}P_8$, no other *FP* votes for the original point C_1P_7. This means that C_1P_7 only gets two votes, and will be kicked out of $\mathbf{S_r}$. Table 5 shows the results of the first round voting.

TABLE 5. The updated $\mathbf{S_r}$ after the first round in-group voting.

Camera NO.	2	3	4	6	9	12	13	14
Point NO.	10	11	8	10	10	9	8	9

The results of the second-round voting is shown in Table 6, which indicates that a new point C_7P_8 is added into the matching group $\mathbf{S_r}$. After the third-round voting, $\mathbf{S_r}$ stays stable, and the final matching group $\mathbf{S_r}$ is shown in Table 6. Points C_2P_{10}, C_3P_{11}, C_4P_8, C_6P_{10}, C_7P_8, C_9P_{10}, $C_{12}P_9$, $C_{13}P_8$, and $C_{14}P_9$ are the 2D observations of the r^{th} dynamic point.

TABLE 6. The updated $\mathbf{S_r}$ after the second round global voting.

Camera NO.	2	3	4	6	**7**	9	12	13	14
Point NO.	10	11	8	10	**8**	10	9	8	9

3) CALCULATING THE 3D COORDINATES

The 3D coordinates are triangulated by the DLT algorithm from matching groups. In this step, a method called Point Residual, expressed as Algorithm 3, is proposed to exclude outliers further. At the end of the last round of in-group voting, the votes of every point in matching group $\mathbf{S_r}$ are obtained, denoted as $\mathbf{V_r}$. After setting a 2D *FP* and a 3D *FP*, if the Manhattan distance between 3D *FP* and 3D point reconstructed by 2D *FP* and point in {$\mathbf{S_r}$-2D *FP*} is larger than a threshold, then the point in {$\mathbf{S_r}$-2D *FP*} is excluded from $\mathbf{S_r}$. The entire process is shown as Algorithm 3, where k_1 and k_2 are the serial numbers of points in $\mathbf{S_r}$ corresponding to $V_r'(1)$ and $V_r'(2)$.

Algorithm 3 Point Residual

Input: $\mathbf{S_r}$, $\mathbf{V_r}$

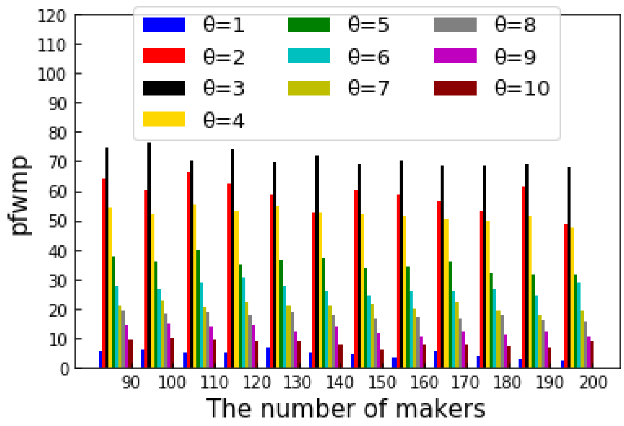
Output: Refined $\mathbf{S_r}$

```
V'_r = Sort(V_r) /☆ Sort V_r from largest to smallest. ☆/
Triangulate the 3D FP by S_r(k_1) and S_r(k_2).
for i = 1:length(S_r) do
    if i = k_1 then
        continue
    else
        if Manhattan distance between 3D FP and 3D point
        reconstructed by S_r(k_1) and S_r(i) is larger than a
        threshold then
            Delete S_r(i)
        end if
    end if
end for
return S_r(i)
```

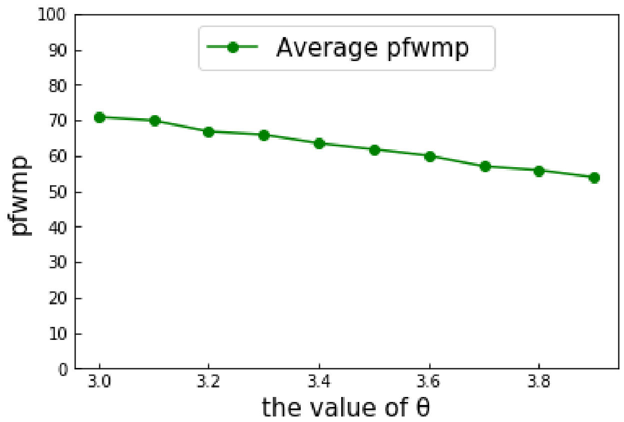
IV. EXPERIMENTS AND EVALUATION

In this section, we provide our evaluation based on the standard and our own datasets. The standard datasets are used to compare our method with NRSFM methods [6], [8], [42], [43] in precision evaluation, they include Drink, Pick-up, Yoga, Stretch, and Dance. As 2D observations of standard datasets are unavailable directly, true 3D points are projected to synthetic cameras every 24 degrees to generate 15 sets of 2D observations. Gaussian noise is then added to all these 15 sets of 2D observations, as [6], [8], [42], and [43] done. Our own datasets are used to compare our framework to Cortex, we collect multiple 2D motion datasets using the MCS provided by Motion Analysis. The MCS includes 15 cameras, and each 2D motion dataset contains 3600 frames and 90, 100, 110, 120, 130, 140, 150, 160, 170, 180, 190 and 200 dynamic points. In addition, as one of the most successful commercial analysis and processing software of motion data, Cortex [11] is chosen as our benchmark for the evaluation on our own datasets, it has been widely used to reconstruct dynamic points in motion capture application. Our experiments include: 1) calculation of the value of θ, 2) evaluation of calibration, 3) evaluation of reconstruction results, and 4)visualization of sample reconstruction results.

The position error metric is the same as that reported in [6], [8], [42], and [43], where $e_{3D} = \frac{1}{\sigma FN}\sum_{f=1}^{F}\sum_{n=1}^{N} e_n^f$ represents the normalized mean 3D error between the reconstructed 3D points and the ground truth; e_n^f is used to denote the 3D error of the n^{th} point in frame f; and $\sigma = \frac{1}{3F}\sum_{f=1}^{F}(\sigma_x^f + \sigma_y^f + \sigma_z^f)$, σ_x^f, σ_y^f, and σ_z^f are the standard deviations of error in frame f for x, y, and z coordinates. When evaluating on our own datesets, directly comparing the 3D position of the reconstructed dynamic points is unpractical in each frame, since Cortex can not output the coordinates of reconstructed dynamic points. Thus, we use the metric of the percentage of frames, whose number of reconstruction points is equal to the number of markers, the error metric is denoted as *pfwmp*, the higher the *pfwmp*, the better the result of reconstruction quality.



(a)



(b)

FIGURE 12. (a): Variation of *pfwmp* with the value of θ from 1-10. (b): Variation of *pfwmp* with the value of θ from 3.0-3.9.

A. DETERMINING THE VALUE OF θ

We first test the value of θ from 1 to 10, and results are shown in Fig. 12(a). When θ increases from 1 to 3, The figure shows that *pfwmp* increases at the same time on all datasets, and reaches the peak at $\theta = 3$ (shown as the black bar), The reason is that when the value of θ is small, some correct corresponding points are excluded by bipolar constraint. With the continuous increase of θ, *pfwmp* continues to declines. The increase of θ lead increase of the number of points in the search area to increase, finally resulting in much mismatch. To further refine the value of θ, we test the value of θ from 3.0 to 3.9. The statistical result is shown in Fig. 12(b), indicating that *pfwmp* is decremented when θ is from 3.0 to 3.9. Therefore, we determine $\theta = 3.0$ for our selected MCS.

B. CALIBRATION EVALUATION

We test the calibration method based on our own datasets. Utilizing the same reconstructing method, we use the camera parameters calibrated by our own method and Cortex respectively. Fig. 13 shows the results. When using our calibration method, *pfwmp* (blue bar) shows an average of 2.7% higher compared with Cortex (the red bar) on all of our datasets. Therefore, our calibration method leads to better reconstruction results compared with Cortex. In an additional test, when the position of the four reference points is not optimized, *pfwmp* drops by an average of 8.5% using our calibration method, as the black bar shows. Optimizing the position of the four reference points with multiple geometric constraints improves the calibration quality significantly.

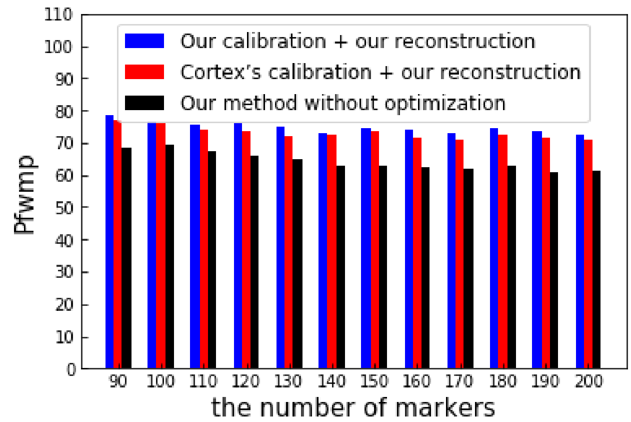


FIGURE 13. Comparison of *pfwmp* by using our calibration (the blue bar) + our reconstruction, Cortex (the red bar) + our reconstruction, method without optimization (the black bar).

C. RECONSTRUCTION EVALUATION

In this subsection, we divide our evaluation into three parts: 1) compare the normalized mean 3D error e_{3D} with NRSFM method on the standard datasets, 2) compare the reconstruction quality with Cortex on our own datasets, and 3) compare the matching quality with Cortex on our own datasets.

1) COMPARING WITH NRSFM METHOD

To evaluate position precision, we compare our proposed method with the state-of-art NRSFM method on standard datasets. e_{3D} are quoted from [6], [8], [42], and [43]. As shown in Table 7, the e_{3D} of our reconstruction performs lower than all the latest state-of-art NRSFM method except for [6], which only performs better in the dataset of Drink. In another test, we use the 15 synthetic 2D datasets with noise during matching and the 15 synthetic datasets without noise during triangulation. We find that the reconstructed points are

TABLE 7. Comparison of performance on standard datasets.

Dataset	SPM	EM-PND	LSMLF	I Khan	Ours
Yoga	0.022	0.014	0.102	0.003	**0.002**
Stretch	0.029	0.016	0.152	0.009	**0.000**
Pick-up	0.036	0.037	0.083	0.019	**0.001**
Drink	0.0286	0.004	0.085	**0.001**	0.003
Dance	0.145	0.183	0.135	0.010	**0.005**

almost coincident with the ground truth and that the e_{3D} of each dataset is much close to 0. This result means that the deviation of the reconstructed points in Table 7 is mainly caused by the additional noise. Our experiments demonstrate that the proposed method can reach a reliable position precision.

2) COMPARING RECONSTRUCTION RESULTS WITH CORTEX

To evaluate our method in real application, we compare our method with the commercial software Cortex on our own datasets. In Fig. 14, as the number of points increases, regardless of our method or Cortex, *pfwmp* shows a slight downward trend. However, our method (blue bar) performs better than Cortex (black bar) on each dataset, and *pfwmp* is 6.2% higher on average. In addition, based on Cortex's calibration results, our reconstruction method (red bar) performs better than Cortex's reconstruction method (black bar), and the *pfwmp* of the former is 3.7% higher on average. Moreover, our method has a standard deviation of 2.0, whereas Cortex has a standard deviation of 2.7, indicating that the former is more stable as the number of points increases. Our experiments prove that our reconstruction method can achieve better results than Cortex.

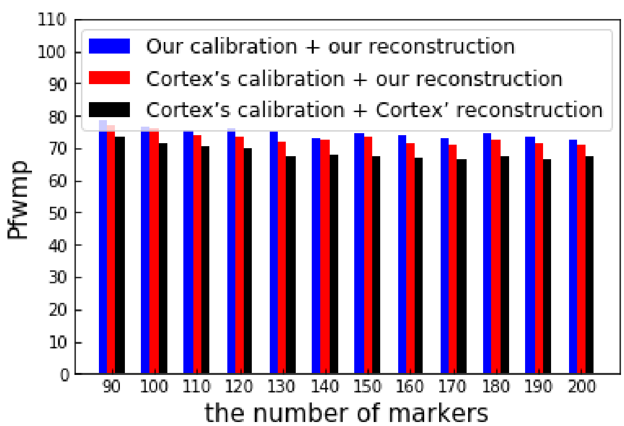


FIGURE 14. Comparison results of our reconstruction method and Cortex.

3) COMPARING THE MATCHING RESULTS WITH CORTEX

In 3.3.3, we introduce how our method works. The same motion data is input into Cortex, where the camera number minus 1 corresponding to the camera number in our method. In addition to the 60^{th} frame, we record the matching group of the r^{th} dynamic point in the 1060^{th} frame, 2060^{th} frame, and 3060^{th} frames. Table 8 shows the results of our method, and Fig. 15 shows the results of Cortex. In the 60^{th} frame, seven cameras can capture the r^{th} dynamic point in Cortex. In fact, the left camera (camera 3) and middle camera (camera 7) should see the point, but they fail to capture the point in Fig. 15(a). These two cameras correspond to C_2P_{10} and C_6P_{10} in our matching group. In the 1060^{th} frame, the r^{th} point faces to camera 7 and cameras 4, these two

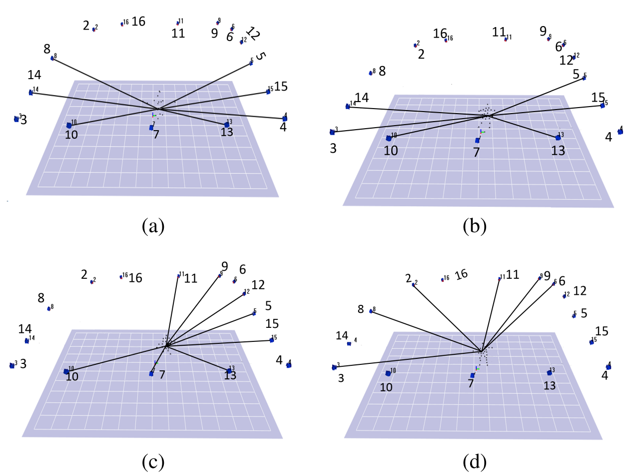


FIGURE 15. Cameras capturing the r^{th} dynamic point in Cortex. (a) 60^{th} frame. (b) 1060^{th} frame. (c) 2060^{th} frame. (d) 3060^{th} frame.

cameras should see the point in Fig. 15(b), but they miss the point. On the contrary, our matching group contains these two cameras, which are denoted as C_6P_2 and C_3P_7 in Table 8. Our method also finds that Camera 12 is likely to see the point in cortex. In the 2060^{th} frame, camera 4 in Fig. 15(c) does not capture the r^{th} point, but the corresponding point C_3P_5 can be found in Table 8. Both our method and Cortex miss camera 6(C_5 in Table 8). In the 3060^{th} frame, Cortex misses cameras 16 and 14, but the corresponding points of $C_{15}P_6$ and $C_{13}P_{10}$ can be found in our matching group. Therefore, if we only consider the correct cameras included in the matching group, our matching method performs much better than Cortex.

TABLE 8. The matching groups of the r^{th} dynamic point in our method.

Frame NO.	Matching groups
60^{th} frame	C_7P_8, $C_{13}P_8$, C_2P_{10}, C_9P_{10}, C_6P_{10}, $C_{12}P_9$, C_3P_{11}, $C_{14}P_9$, C_4P_8
1060^{th} frame	$C_{13}P_2$, C_2P_4, C_9P_{11}, C_6P_9, $C_{12}P_{10}$, C_3P_7, $C_{14}P_6$, C_4P_{11}, $C_{11}P_9$
2060^{th} frame	C_9P_8, C_6P_6, $C_{12}P_{11}$, C_3P_5, $C_{14}P_8$, C_4P_2, $C_{11}P_8$, C_8P_8, $C_{10}P_5$
3060^{th} frame	C_5P10, C_8P_7, $C_{10}P_4$, $C_{15}P_6$, C_1P_5, C_7P_6, $C_{13}P_{10}$, C_2P_9

D. VISUALIZATION OF SAMPLE RECONSTRUCTION RESULTS

To visualize our reconstruction results, we present the comparison between the reconstruction results and ground truth on standard datasets in Fig. 16. We also provide a visualization for reconstruction results on our own dataset from 1695^{th} frame to 1721^{th} frame, the datasets include two humans, 100 dynamic points, and 27 frames. Although many points on the two humans almost overlap, and the interference

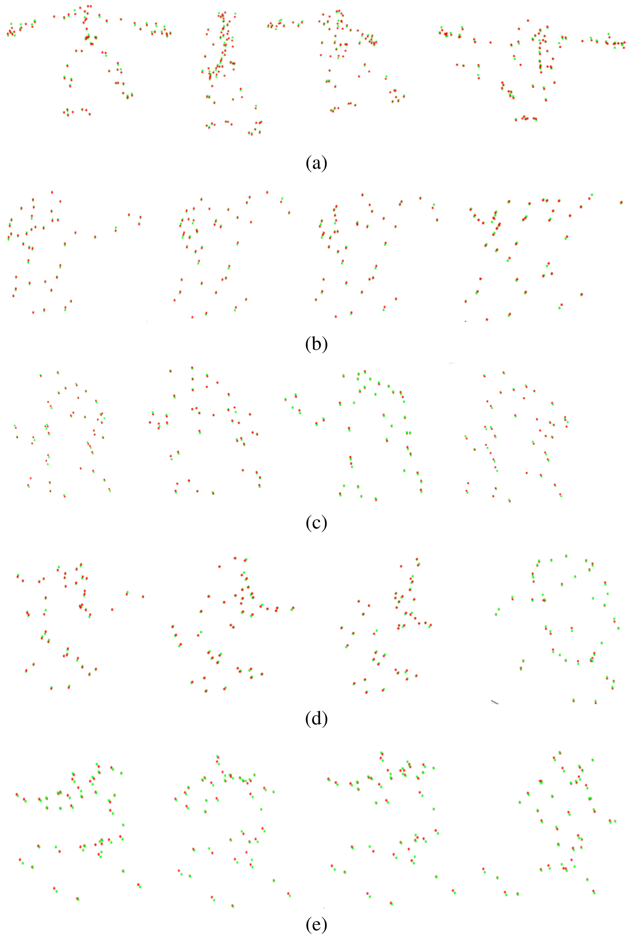


FIGURE 16. Visualization of sample reconstruction results on standard dataset. The ed points are the ground truth, and the green points are the reconstruction results by our method.) (a)1^{th}, 34^{th}, 148^{th} and 210^{th} frame of Dance dataset, $e_{3D} = 0.005$. (b)103^{th}, 374^{th}, 662^{th} and 960^{th} frames of Drink dataset, $e_{3D} = 0.003$. (c)1^{th}, 138^{th}, 342^{th} and 357^{th} frame of Pickup dataset, $e_{3D} = 0.001$. (d)44^{th}, 162^{th}, 240^{th} and 357^{th} frame of Stretch dataset, $e_{3D} = 0.000$. (e)1^{th}, 43^{th}, 206^{th} and 357^{th} frame of Yoga dataset, $e_{3D} = 0.002$. (a) Dance. (b) Drink. (c) Pick up. (d) Stretch. (e) Yoga.

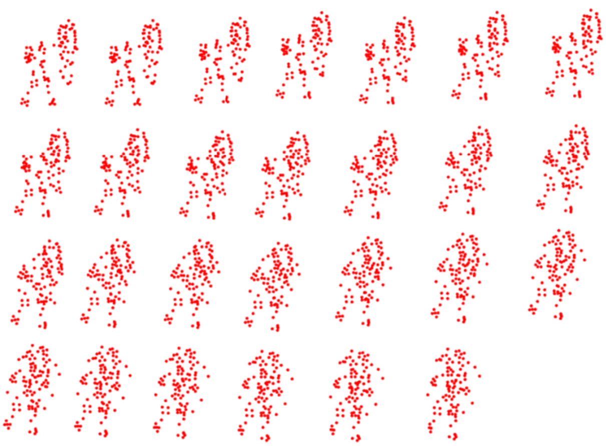


FIGURE 17. Recovering 100 3D dynamic points in our own datasets from 1695^{th} frame to 1721^{th} frame.

between the points is quite serious, our method still works well to reconstruct these points. The number of reconstruction points is 100 in all frames but one, as shown in Fig. 17. However, in Cortex, we find eight frames whose reconstruction points are less or more than 100.

V. CONCLUSION

This study puts forward a complete framework to reconstruct precise dynamic points only with their 2D positions in MCSs. Our method focuses on decreasing mismatch when searching for correspondence in multi-view. In the application of the motion capture system, we introduce multiple constraints to optimize the position of reference points, and we find that the treatment improves the performance of calibration. During matching, basing on epipolar geometry, we propose a novel sparse multi-view matching method, which consists of CV and PR. Experiments prove that our method can achieve outstanding performance on standard and our own datasets. Compared with commercial software Cortex, our method exhibits better reconstruction quality and decrease mismatching significantly. In the future, we intend to develop a method to determine the search area automatically, and improve our computation speed by parallel computing.

ACKNOWLEDGMENT

The authors are grateful to Beijing Nokov Technology Co., LTD for providing Cortex, Multiple Cameras Systems (MCSs) and 2D motion capture datasets.

REFERENCES

[1] T. Karras, T. Aila, S. Laine, and A. Herva, "Audio-driven facial animation by joint end-to-end learning of pose and emotion," *ACM Trans. Graph.*, vol. 36, no. 4, pp. 94–108, 2017.

[2] T. Schubert, A. Gkogkidis, F. Ball, and W. Burgard, "Automatic initialization for skeleton tracking in optical motion capture," in *Proc. IEEE Conf. Robot. Autom.*, May 2015, pp. 734–739.

[3] T. Schubert, K. Eggensperger, A. Gkogkidis, F. Hutter, T. Ball, and W. Burgard, "Automatic bone parameter estimation for skeleton tracking in optical motion capture," in *Proc. IEEE Conf. Robot. Autom.*, May 2015, pp. 734–739.

[4] M. Balazia and P. Sojka, "Gait recognition from motion capture data," *ACM Trans. Multimedia Comput.*, vol. 14, no. 22, pp. 22:2–22:18, Apr. 2018.

[5] T. Dabóczi, "Analysis of the distortion of marker-Based optical position measurement as a function of exposure time," *IEEE Trans. Instrum. Meas.*, vol. 65, no. 9, pp. 2023–2034, Sep. 2016.

[6] I. Khan, "Robust sparse and dense nonrigid structure from motion," *IEEE Trans. Multimedia*, vol. 20, no. 4, pp. 841–850, Apr. 2018.

[7] A. Agudo and F. Moreno-Noguer, "DUST: Dual union of spatio-temporal subspaces for monocular multiple object 3D reconstruction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 6262–6270.

[8] M. Lee, J. Cho, C.-H. Choi, and S. Oh, "Procrustean normal distribution for non-rigid structure from motion," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2013, pp. 1280–1287.

[9] Y. Gao and A. L. Yuille, "Symmetric non-rigid structure from motion for category-specific object structure estimation," in *Proc. Comput. Vis.-ECCV*. New York, NY, USA: Springer, 2016, pp. 408–424, 2016.

[10] J. Xiao, J. Chai, and T. Kanade, "A closed-form solution to non-Rigid shape and motion recovery," in *Computer Vision—ECCV*, New York, NY, USA: Springer, 2004, pp. 573–587.

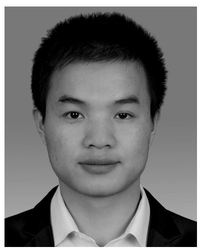
[11] *Motion Analysis Corp.* [Online]. Available: http://www.tp-ontrol.hu/index.php/TP_Toolbox

[12] *Motion Capture Systems Vicon*. [Online]. Available: https://www.vicon.com

[13] J. J. Moré, "The Levenberg-Marquardt algorithm: Implementation and theory," in *Numerical Analysis* (Lecture Notes in Mathematics). 1978, pp. 105–116.

[14] J. Dehais, M. Anthimopoulos, S. Shevchik, and S. Mougiakakou, "Two-view 3D reconstruction for food volume estimation," *IEEE Trans. Multimedia*, vol. 19, no. 5, pp. 1090–1099, May 2017.
[15] G. Kurillo, Z. Li, and R. Bajcsy, "Wide-area external multicamera calibration using vision graphs and virtual calibration object," in *Proc. 2nd ACM/IEEE Int. Conf. Distrib. Smart Cameras*, Sep. 2008, pp. 1–9.
[16] J. Barreto and K. Daniilidis, "Wide area multiple camera calibration and estimation of radial distortion," in *Proc. 5th Workshop Omnidirectional Vis.*, Jan. 2004, pp. 1–12.
[17] H. G. Maas, "Image sequence based automatic multi-camera system calibration techniques," *ISPRS J. Photogram. Remote Sens.*, vol. 54, nos. 5–6, pp. 352–359, 1999.
[18] M. Brückner, F. Bajramovic, and J. Denzler, "Intrinsic and extrinsic active self-calibration of multi-camera systems," *Mach. Vis. Appl.*, vol. 25, no. 2, pp. 389–403, 2014.
[19] P. Baker and Y. Aloimonos, "Complete calibration of a multi-camera network," in *Proc. IEEE Workshop Omnidirectional Vis.*, Jun. 2000, pp. 134–141.
[20] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 3rd ed. Cambridge, U.K.: Cambridge Univ. Press, 2004.
[21] B. Triggs, P. F. Mclauchlan, R. I. Hartley, and A. W. Fitzgibbon, "Bundle adjustment: A modern synthesis," in *Proc. Int. Workshop Vis. Algorithms, Theory Pract.*, 1999, pp. 298–372.
[22] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "G^2o: A general framework for graph optimization," in *Proc. IEEE Conf. Robot. Autom.*, May 2011, pp. 3607–3613.
[23] M. I. A. Lourakis and A. A. Argyros, "SBA: A software package for generic sparse bundle adjustment," *ACM Trans. Math. Softw.*, vol. 36, no. 1, p. 2, 2009.
[24] J. Schneider, F. Schindler, T. Läbe, sand W. Förstner, "Bundle adjustment for multi-camera systems with points at infinity," *ISPRS Ann. Photogram., Remote Sens. Spatial Inf. Sci.*, vol. I-3, pp. 75–80, Aug. 2012.
[25] W. Rstner, "Minimal representations for uncertainty and estimation in projective spaces," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 619–632.
[26] J. Schneider and W. Förstner, "Bundle adjustment and system calibration with points at infinity for omnidirectional camera systems," *Photogrammetrie-Fernerkundung-Geoinformation*, vol. 2013, no. 4, pp. 309–321, Aug. 2013.
[27] Z. Zhang, "A flexible new technique for camera calibration," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Dec. 1999.
[28] Z. Zhang, "Camera calibration with one-dimensional objects," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 7, pp. 892–899, Jul. 2004.
[29] L. Higgins, "A computer algorithm for reconstructing a scene from two projections," *Nature*, vol. 293, pp. 133–135, Sep. 1981.
[30] D. Xiao, Q. Yang, B. Yang, and W. Wei, "Monocular scene flow estimation via variational method," *Multimedia Tools Appl.*, vol. 76, no. 8, pp. 10575–10597, 2017.
[31] T. Basha, Y. Moses, and N. Kiryati, "Multi-view scene flow estimation: A view centered variational approach," *Int. J. Comput. Vis.*, vol. 101, no. 1, pp. 6–21, 2013.
[32] Q.-T. Luong, *The Geometry of Multiple Images*. Cambridge, MA, USA: MIT Press, 2001.
[33] S. Avidan and A. Shashua, "Trajectory triangulation: 3D reconstruction of moving points from a monocular image sequence," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 4, pp. 348–357, Apr. 2000.
[34] A. Shashua and L. Wolf, "Homography tensors: On algebraic entities that represent three views of static or moving planar points," in *Computer Vision—ECCV*. New York, NY, USA: Springer, 2000, pp. 507–521.
[35] J. Y. Kaminski and M. Teicher, "A general framework for trajectory triangulation," *J. Math. Imag. Vis.*, vol. 21, no. 1, pp. 27–41, Jul. 2004.
[36] C. Bregler, A. Hertzmann, and H. Biermann, "Recovering non-rigid 3D shape from image streams," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2000, pp. 690–696.
[37] C. Tomasi and T. Kanade, "Shape and motion from image streams under orthography: A factorization method," *Int. J. Comput. Vis.*, vol. 9, no. 2, pp. 137–145, 1992.
[38] E. Tola, V. Lepetit, and P. Fua, "DAISY: An efficient dense descriptor applied to wide-baseline stereo," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 5, pp. 815–830, May 2010.
[39] A. Shaked and L. Wolf, "Improved stereo matching with constant highway networks and reflective confidence learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 6901–6910.
[40] J. Zbontar and Y. LeCun, "Computing the stereo matching cost with a convolutional neural network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit*, Jul. 2015, pp. 1592–1599.
[41] J. Li and Y. Fei, "Applications of Lodrigues matrix in 3D coordinate transformation," *Geo-Spatial Inf. Sci.*, vol. 10, no. 3, pp. 173–176, 2007.
[42] Q. Dong and H. Wang, "Latent-smoothness nonrigid structure from motion by revisiting multilinear factorization," *IEEE Trans.*, to be published.
[43] Y. Dai, H. Li, and M. He, "A simple prior-free method for non-rigid structure-from-motion factorization," *Int. J. Comput. Vis.*, vol. 107, no. 2, pp. 101–122, 2014.
[44] M. Lee, J. Cho, and S. Oh, "Consensus of non-rigid reconstructions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 4670–4678.
[45] Z. Zhang, *Computer Vision Fundamentals of Computer Algorithms*. Beijing, China: Science Press, 1998.

DEGUI XIAO received the B.E. degree in industrial automation from the Wuhan University of Textile, Wuhan, China, in 1994, and the Ph.D. degree in computer science and technology from the Huazhong University of Science and Technology, Wuhan, in 2003. He is currently an Associate Professor with the College of Computer Science and Electronic Engineering, Hunan University, Changsha, China. His current research interests include image and video processing, computer vision, and edge computing.



JIANFANG LI is currently pursuing the Ph.D. degree with the College of Computer Science and Electronic Engineering, Hunan University, Changsha, China. His research interests include stereoscopic evaluation algorithms and motion capture.

KEQIN LI is currently a Distinguished Professor of computer science with SUNY. He has published over 560 journal articles, book chapters, and refereed conference papers. His current research interests include parallel computing and high-performance computing, distributed computing, energy-efficient computing and communication, heterogeneous computing systems, cloud computing, big data computing, CPU–GPU hybrid and cooperative computing, multicore computing, storage and file systems, wireless communication networks, sensor networks, peer-to-peer file sharing systems, mobile computing, service computing, the Internet of Things, and cyber-physical systems. He is a Fellow of the IEEE. He was a recipient of several best paper awards. He currently serves or has served on the editorial boards for the IEEE Transactions on Parallel and Distributed Systems, the IEEE Transactions on Computers, the IEEE Transactions on Cloud Computing, the IEEE Transactions on Services Computing, and the IEEE Transactions on Sustainable Computing.