MOTION ESTIMATION OF MULTIPLE DEPTH CAMERAS USING SPHERES

Xiaoming Deng[1], *Jie Liu*[1], *Feng Tian*[1], *Liang Chang*[2], *Hongan Wang*[1]

[1] Beijing Key Lab of Human-Computer Interaction, Institute of Software, Chinese Academy of Sciences
[2] College of Information Science and Technology, Beijing Normal University, China
Email: {xiaoming,tianfeng,hongan}@iscas.ac.cn, changliang@bnu.edu.cn

ABSTRACT

Automatic motion estimation of multiple depth cameras has remained a challenging topic in computer vision due to its reliance on the image correspondence problem. In this paper, spherical objects are employed to estimate motion parameters between multiple depth cameras. We move a sphere several times in the common view of depth cameras. We fit the spherical point clouds to get the sphere centers in each depth camera system, and then introduce a factorization based approach to estimate motions between the depth cameras. Both simulated and real experiments show the robustness and effectiveness of our method.

***Index Terms*—** Depth cameras, motion estimation, sphere object

1. INTRODUCTION

Multiple depth cameras is quite important in 3D reconstruction. The 3D data generated from a single depth camera comes from a single direction oriented around the center of the depth camera. This results in one-sided objects or people. We have to move the depth camera in order to generate the opposite side like in the Kinectfusion system[1], while Kinectfusion is mainly designed for static scene modeling. With multiple depth cameras, we can add the captured data together and produce a more complete and temporal synchronous scene.

In order to reconstruct a scene from the multiple depth cameras , the system must be calibrated[2][3][4]. This includes internal calibration of each camera as well as relative motion calibration between the cameras. Color camera calibration has been studied extensively. For depth sensors, different calibration methods have been developed depending on the technology used. Several internal calibration methods of depth camera calibration have been proposed in the Kinect community, they are all plane-based methods, and most of them are not fully automatic. Herrera made a comprehensive calibration of all parameters of the camera pair[2]. Using a similar formulation, Zhang and Mikhelson calibrated cameras with correspondences between the color and depth images[5][6]. The intrinsic parameter of depth cameras(such as Kinect) are calibrated during manufacturing. The calibrated parameters are stored in the devices internal memory and are used by the official drivers to perform the reconstruction.

3D reconstruction and visual surveillance using a depth camera network have imposed new challenges to camera motion calibration[7]. One essential problem is that current approaches for depth camera calibration using planes may not be feasible since these objects may not be simultaneously visible by all the conventional cameras. Although 1D object (line object with markers) based method has great advantage in the calibration of multiple perspective cameras[8][9], it is infeasible for the motion estimation of depth cameras due to the fact that they are often invisible in the depth image. The depth camera consists of an infrared laser projector combined with a monochrome imaging sensor, and 1D object is too thin for depth camera to recover its depth and the markers on it. Similarly, a popular method [10] in multiple camera self-calibration can not be used for depth cameras. Sphere is a widely available object, and a few calibration method using spheres were proposed in the field of omnidirectional and perspective cameras[7][11], which are flexible and accurate. In the field of depth camera, little attentions are paid to use sphere objects to recover camera motions.

In this paper, we propose a new method to calibrate extrinsic parameters of multiple depth cameras by using a sphere object, where minimal human interactions are required. Our calibration method has the following advantages: 1) Automatic motion estimation method. The method is automatic and suitable for practice use. Our calibration method does not need tedious correspondence selection as in previous depth camera calibration methods. 2) The method is linear and easy for implementation.

2. PRELIMINARY

The projection of a point from depth camera coordinates $\mathbf{x}_d = [x_d, y_d, z_d]^T$ to depth image coordinates $\mathbf{m}_d = [u_d, v_d]^T$ is obtained through the following equations[2][6]:

$$z_d \begin{bmatrix} u_d \\ v_d \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} f_{du} & s_d & u_{0d} \\ 0 & f_{dv} & v_{0d} \\ 0 & 0 & 1 \end{bmatrix}}_{\mathbf{K}} \begin{bmatrix} x_d \\ y_d \\ z_d \end{bmatrix} \quad (1)$$

where $\mathbf{K}$ is the intrinsic parameter matrix of depth camera, $\mathbf{f}_d = [f_{du}, f_{dv}]^T$ are the focal lengths, $\mathbf{p}_{0d} = [u_{0d}, v_{0d}]^T$ is the principal point and s_d is the skew parameter.

The 3D point can be recovered with its image coordinate and the depth z_d as follows:

$$\begin{bmatrix} x_d \\ y_d \\ z_d \end{bmatrix} = \mathbf{K}^{-1} z_d \begin{bmatrix} u_d \\ v_d \\ 1 \end{bmatrix} \quad (2)$$

3. ALGORITHM

We show how to recover motion parameters by aligning sphere-centers under multiple depth camera systems (See

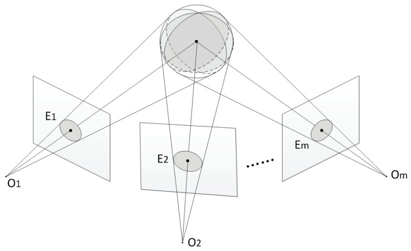


Fig. 1. Illustration of sphere calibration objects.

Fig. 1). We first detect sphere images, and then estimate the sphere center with the point cloud of sphere images. The sphere centers in each depth camera system are regarded as correspondences, and we choose a robust factorization method to get the motion parameters. We suppose that the captured scene has a dominant plane such as a floor or a desk, and the intrinsic parameters $\mathbf{K}$ of depth camera are known.

3.1. Sphere Image Detection and Sphere Center Estimation

The 2D image of a sphere contour is an ellipse, and Hough transform can be used for robust ellipse detection. In our problem, we use the method [12] for ellipse detection. This method takes the advantages of major axis of an ellipse to find ellipse parameters fast and efficiently. Based on the detected sphere contour images, we fit the sphere with the point clouds by lifting the pixels within the sphere contour images to 3D, and then recover the sphere center.

The procedures of our sphere image detection are as follows (See Fig. 2): 1) recover 3D point cloud from a depth image by (2); 2) fit the dominant plane with RANSAC, and keep the pixels in the depth image, the reconstructed points of which are on the same side of the plane as the depth camera and within a feasible distance range to the plane. The depths of the rest pixels are all set to zeros; 3) use the method in [12] to detect ellipses with the constraints that the length of major axis is within 50 and 200 pixels, and the aspect ratio is above 0.9. Finally, we use the recovered 3D points pixels within the detected sphere contour image to fit the sphere with the method [13] and get its sphere center.

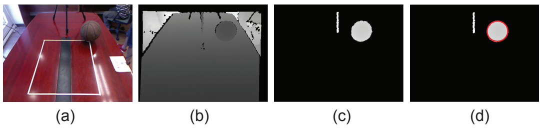


Fig. 2. Illustration of the sphere image detection. (a) RGB image. (b) Depth image. (c) Processed depth image by removing infeasible depth pixels. (d) Detected ellipse in the depth image.

Remark 1: If multiple ellipses are detected, we may choose the ellipse with the smallest sphere fitting error.

3.2. Motion Estimation

In this section, we show how to recover motion parameters of multiple depth cameras from spheres' centers by using a factorization method for 3D point sets. The factorization method is advantageous over other methods in that it treats each view equally and recover all the camera motion matrices simultaneously[14][15].

Denote $\mathbf{O}_{i,j}(i = 1, ..., m, j = 1, ..., n)$ to be the recovered j-th 3D coordinates of spherical centers in the i-th depth camera coordinate systems respectively.

Estimate translations. Each translation $\mathbf{t}_i$ is computed as the centroid of sphere centers in the i-th camera coordinate system

$$\mathbf{t}_i = \frac{1}{n}\sum_{j=1}^{n} \mathbf{O}_{i,j} \tag{3}$$

Centre the data. Centre the sphere centers in each depth image by expressing their coordinates with respect to the centroid:

$$\tilde{\mathbf{O}}_{i,j} = \mathbf{O}_{i,j} - \mathbf{t}_i. \tag{4}$$

Hereafter, we work with these centered coordinates.

Estimate rotations. Construct the $3m \times n$ measurement matrix $\mathbf{W}$ from the centered data shown as in Eq. (5), compute its SVD $\mathbf{W} = \mathbf{U}\mathbf{\Lambda}\mathbf{V}^T$.

Since we have $\tilde{\mathbf{O}}_{i,j} = \hat{\mathbf{R}}_i\hat{\mathbf{O}}_j$, we get a factorization of $\mathbf{W}$ as follows.

$$\underbrace{\begin{bmatrix} \tilde{\mathbf{O}}_{11} & \tilde{\mathbf{O}}_{12} & \cdots & \tilde{\mathbf{O}}_{1n} \\ \tilde{\mathbf{O}}_{21} & \tilde{\mathbf{O}}_{22} & \cdots & \tilde{\mathbf{O}}_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ \tilde{\mathbf{O}}_{m1} & \tilde{\mathbf{O}}_{m2} & \cdots & \tilde{\mathbf{O}}_{mn} \end{bmatrix}}_{\mathbf{W}} = \underbrace{\begin{bmatrix} \hat{\mathbf{R}}_1 \\ \hat{\mathbf{R}}_2 \\ \vdots \\ \hat{\mathbf{R}}_m \end{bmatrix}}_{\mathbf{Q}} \underbrace{[\hat{\mathbf{O}}_1, \hat{\mathbf{O}}_1, ..., \hat{\mathbf{O}}_n]}_{\mathbf{S}} \tag{5}$$

where $\mathbf{Q}$ is a motion matrix, $\mathbf{S}$ is a shape matrix, and $\hat{\mathbf{O}}_j$ is an estimated sphere center in the world coordinate system.

From a measurement matrix $\mathbf{W}$, we can compute its rank-3 decomposition $\mathbf{W}_{3m\times n} = \mathbf{Q}_{3m\times 3}\mathbf{S}_{3\times n}$ by SVD. However, this decomposition is not unique as any nonsingular matrix $\mathbf{T}_{3\times 3}$ can be inserted between $\mathbf{Q}$ and $\mathbf{S}$ to obtain a new valid factorization as $\mathbf{W} = \hat{\mathbf{Q}}\hat{\mathbf{S}} = (\mathbf{Q}\mathbf{T})(\mathbf{T}^{-1}\mathbf{S})$.

Due to the orthogonality constraints of rotation matrix, we enforce $\mathbf{T}$ with the following constraints($k = 1, ..., m$):

$$\begin{aligned} \mathbf{Q}_{[3k+1:3k+3,:]}\mathbf{T}(\mathbf{Q}_{[3k+1:3k+3,:]}\mathbf{T})^T &= \\ \mathbf{Q}_{[3k+1:3k+3,:]}\underbrace{\mathbf{T}\mathbf{T}^T}_{\mathbf{\Omega}}\mathbf{Q}^T_{[3k+1:3k+3,:]} &= \mathbf{I}_{3\times 3} \end{aligned} \tag{6}$$

where $\mathbf{\Omega}_{3\times 3} = \mathbf{T}\mathbf{T}^T$. These equations are linear in terms of elements of the symmetric matrix $\mathbf{\Omega}_{3\times 3}$, and then we can get homogenous linear equations of $\mathbf{\Omega}_{3\times 3}$:

$$\mathbf{A}\mathrm{vec}(\mathbf{\Omega}) = \mathbf{0} \tag{7}$$

where $\mathrm{vec}(\mathbf{\Omega})$ can be estimated with singular vector decomposition(SVD).

After $\mathbf{\Omega}_{3\times3}$ is obtained, $\mathbf{T}$ can be recovered by SVD of $\mathbf{\Omega}_{3\times3}$ as follows

$$\mathbf{\Omega} = \mathbf{U\Lambda U}^T = \mathbf{U\Lambda}^{1/2}\mathbf{\Lambda}^{1/2}\mathbf{U}^T = \mathbf{U\Lambda}^{1/2}(\mathbf{U\Lambda}^{1/2})^T \quad (8)$$

where $\mathbf{T} = \mathbf{U\Lambda}^{1/2}$. Thus the rotation matrices $\{\hat{\mathbf{R}}_i\}_{i=1}^m$ can be recovered by $\mathbf{QT}$, and the sphere centers under multiple sphere motions can be recovered by $\mathbf{T}^{-1}\mathbf{S}$. The signs of $\text{vec}(\mathbf{T})$ are adjusted to ensure that the resulting matrices $\mathbf{QT}$ have positive determinant.

For our comparison convenience of the simulated experiments Section 4.1, we choose the rotation matrix and translation vector of the first camera as $[\mathbf{I}_{3\times3}, \mathbf{0}_{3\times1}]$. Then the rotation matrices and translation vectors can be transformed as follows:

$$\begin{aligned}\bar{\mathbf{R}}_i &= \hat{\mathbf{R}}_i\hat{\mathbf{R}}_1^T \\ \bar{\mathbf{t}}_i &= -\hat{\mathbf{R}}_i\hat{\mathbf{R}}_1^T\mathbf{t}_1 + \mathbf{t}_i\end{aligned}$$

If the sphere centers may not be detected in all images, some of the elements of $\mathbf{W}$ may be unknown, which is the missing data problem. This problem can be handled with the methods[10].

Remark 2: If the sphere centers during movements are collinear, there are infinitely many rotations and reflections solutions. Therefore, this is the degeneracy configuration of our method.

3.3. Complete Algorithm

An outline of our algorithm is summarized as follows:
Input: images of sphere objects by at least three motions.
Output: motion parameters of multiple depth cameras, $\{\bar{\mathbf{R}}_i, \bar{\mathbf{t}}_i\}_{i=1}^m$.

1. For each sphere movement j, detect the sphere image $\mathbf{E}_{i,j}$ on the image of depth camera i, and lift the pixels within each sphere contour image to 3D point clouds $\mathbf{S}_{i,j}(i = 1, ..., m, j = 1, ..., n)$;

2. Estimate the sphere centers under a depth camera coordinate system i with $\{\mathbf{S}_{i,j}\}_{j=1}^n$, use RANSAC to remove outliers of sphere center correspondences, and then put the extraction into a set $\mathbf{O}_{i,j}(i = 1, ..., m, j = 1, ..., n)$;

3. Use the sphere center sets $\mathbf{O}_{i,j}(i = 1, ..., m, j = 1, ..., n)$ to estimate the depth camera motions $\{\bar{\mathbf{R}}_i, \bar{\mathbf{t}}_i\}_{i=1}^m$.

4. EXPERIMENTS

The proposed algorithm has been tested on both simulated data and real image data.

4.1. Simulated Experiments

We perform a lot of simulations with six cameras as shown in Fig. 3. In the simulation, the intrinsic and extrinsic parameters of the simulated depth cameras are known, and the image resolutions are of 640 × 480. The rotation matrix and translation vector of the first camera is set to $[\mathbf{I}_{3\times3}, \mathbf{0}_{3\times1}]$. We move a 3D sphere with 162 points of uniform grid ten times, and project it on six depth image planes. The value of each pixel in the depth image is its depth value. Gaussian noise with mean 0 and standard derivations σ are added to the depth values, and the noise level σ increases from 0mm to 10mm in steps of 2mm. At each noise level, 100 independent trials are performed. The estimated camera motion parameters are compared with the ground truth, and RMS errors are measured. The errors of the cameras' Euler angles and translations relative to the first camera are shown in Fig. 4. The errors increase almost linearly as the noise level increases.

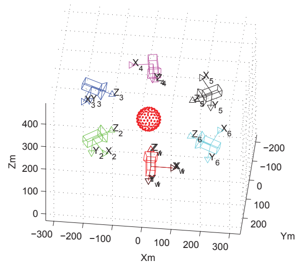
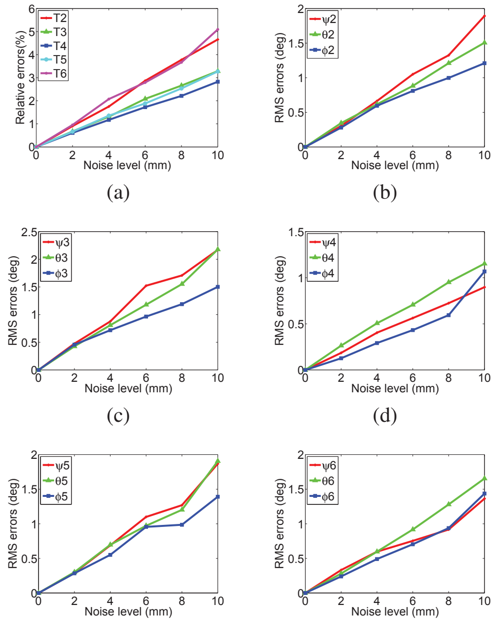


Fig. 3. The synthetic experimental setup with six cameras and a sphere object.



(a) (b)

(c) (d)

(e) (f)

Fig. 4. The errors of six simulated cameras extrinsic parameters.

4.2. Real Experiments

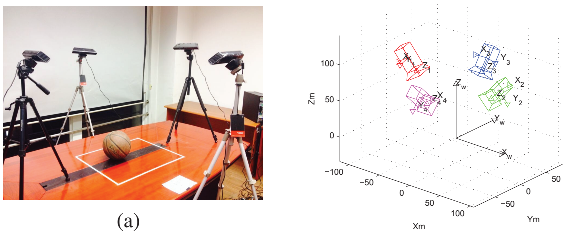
In this section, we describe a real dataset consists of four Kinect cameras as shown Fig. 5(a), and the effective capture volume is approximately 70 × 60 × 50 cm (a virtual cube above the white square in Fig. 5(a)). The depth cameras are with an image resolution of 640 × 480 pixels. Although many delicate approaches could be applied, we simply synchronize

Table 1. Reconstruction errors with the estimated camera motion results by our method and the planar-based method.

Method	θ(degrees)	σ_θ (degrees)
Our method	89.8	1.1
Planar-based method	89.7	1.5

the cameras by multiple thread techniques. The four depth cameras locate on a rough circle, looking inward at a common free space. The angle of nearby two Kinects is about 90 degrees, so that the active lights of Kinects are not interfering severely with each other.

We use a basketball as the calibration object and move the sphere 27 times in the common view of depth cameras. The camera pose estimation results are shown in Fig. 5(b).



(a)

(b)

Fig. 5. (a) Camera setup. (b) Camera pose estimation results.

We use 3D cube reconstruction to verify the motion estimation results (See Fig. 6). Each planes of the cube(black&white patterns on a glass cube) are fitted with reconstructed 3D points, and we compute the angles between each two nearby planes, $\{\theta_i\}_{i=1}^{7}$ (the ground truth is 90 degrees). The average angle θ and standard deviation σ_θ are computed as shown in Table 1. We see that the results with our method are a little better than those with planar-based method.

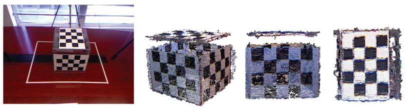


Fig. 6. Cube reconstruction results. Each row shows an reconstruction example with color images and three views of reconstructed point clouds.

We also use the calibrated multiple depth cameras to reconstruct 3D objects, in which only the points within the effective capture volume are kept. Our datasets include ”kettle”, ”milk box”, ”shoe”, ”plant”, ”arm” and ”teabox”. The reconstructed results are shown in Fig. 7.

5. CONCLUSION

We propose a new method to calibrate multiple depth cameras by moving a sphere. It is automatic to recover the depth camera motions, linear, and easy for implementation. Both statistical evaluation and real data verify the feasibility and power of this calibration approach. We also show that our method is suitable for 3D reconstruction application with multiple depth cameras.

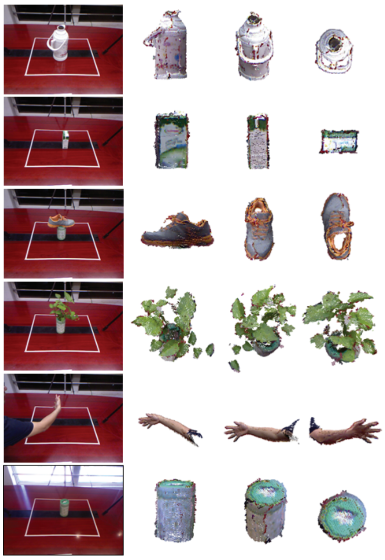


Fig. 7. Reconstruction results. Each row shows an reconstruction example with color images and three views of reconstructed point clouds.

6. ACKNOWLEDGMENT

This work is supported by National Natural Science Foundation of China (Nos. 61232013, 61005039, 61170182), National 973 Project (No.2013CB329305), Beijing Higher Education Young Elite Teacher Project(No. YETP0231), NLPR open project (No. 20090096).

7. REFERENCES

[1] Richard A. Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J. Davison, Pushmeet Kohli, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon, “Kinectfusion: Real-time dense surface mapping and tracking,” in *Proc. International Symposium on Mixed and Augmented Reality. Mixed Reality*, 2011.

[2] Daniel Herrera C., Juho Kannala, and Janne Heikkila, “Joint depth and color camera calibration with distortion correction,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, pp. 2058–2064, 2012.

[3] Li Guan and Marc Pollefeys, “A unified approach to calibrate a network of camcorders and tof cameras,” in *Proc. ECCV Workshop on Multi-camera and Multi-modal Sensor Fusion Algorithms and Applications*, 2008.

[4] Mitsuru Nakazawa, Ikuhisa Mitsugami, Yasushi Makihara, Hozuma Nakajima, Hitoshi Habe, Hirotake Yamazoe, and Yasushi Yagi, “Dynamic scene reconstruction using asynchronous multiple kinects,” in *Proc. International Conference on Pattern Recognition*, 2012, pp. 469–472.

[5] Cha Zhang and Zhengyou Zhang, “Calibration between depth and color sensors for commodity depth cameras,” in *Proc. International Conference on Multimedia and Expo*, 2011.

[6] Ilya Mikhelson, Philip Greggory Lee, Alan V. Sahakian, Ying Wu, and Aggelos K. Katsaggelos, “Automatic, fast, online calibration between depth and color cameras,” *J. Visual Communication and Image Representation*, vol. 25, pp. 218–226, 2014.

[7] Xianghua Ying and Zhanyi Hu, “Spherical objects based motion estimation for catadioptric cameras,” in *Proc. International Conference on Pattern Recognition*, 2004, pp. 231–234.

[8] Liang Wang and Fuqing Duan, “Zhang’s one-dimensional calibration revisited with the heteroscedastic error-in-variables model,” in *Proc. International Conference on Image Processing*, 2011, pp. 857–860.

[9] Xiaoming Deng, Fuchao Wu, Yihong Wu, Liang Chang, Wei Liu, and Hongan Wang, “Calibration of central catadioptric camera with one-dimensional object undertaking general motions,” in *Proc. International Conference on Image Processing*, 2011, pp. 637–640.

[10] T. Svoboda, D. Martinec, and T. Pajdla, “A convenient multi-camera self-calibration for virtual environments,” *PRESENCE: Teleoperators and Virtual Environments*, vol. 14, pp. 407–422, 2005.

[11] Guoqiang Zhang and Kwan-Yee Kenneth Wong, “Motion estimation from spheres,” in *Proc. International Conference on Computer Vision and Pattern Recognition*, 2006.

[12] Yonghong Xie and Qiang Ji, “A new efficient ellipse detection method,” in *Proc. International Conference on Pattern Recognition*, 2002, pp. 957–960.

[13] Gabriel Taubin, “Estimation of planar curves, surfaces and nonplanar space curves defined by implicit equations, with applications to edge and range image segmentation,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 13, pp. 1115–1138, 1991.

[14] Peter F. Sturm and Bill Triggs, “A factorization based algorithm for multi-image projective structure and motion,” in *Proc. ECCV*, 1996, pp. 709–720.

[15] Yuchao Dai, Hongdong Li, and Mingyi He, “A simple prior-free method for non-rigid structure-from-motion factorization,” in *Proc. International Conference on Computer Vision and Pattern Recognition*, 2012, pp. 2018–2025.