MULTI-CAMERA PARAMETERS REFINEMENT FOR 3D RECONSTRUCTION

Anonymous ICME submission

ABSTRACT

With the development of virtual and augmented reality, indoor 3D reconstruction based on multi-camera system has become increasingly popular recently. This requires camera parameters to be as accurate as possible. We propose a new method to optimize and adjust the camera extrinsic parameters of the multi-camera system for high-quality 3D reconstruction. Our method consists of two parts: the global multi-camera calibration and the parameters refinement by point cloud registration. In global multi-camera calibration, we use checkerboard corners to optimize the camera extrinsic parameters globally obtained by pairwise camera calibration methods. In parameters refinement, we use ICP (Iterative Closest Point) to register the point clouds in different views reconstructed from the RGB and depth images. Then we use the transformation from ICP to adjust the extrinsic parameters to minimize the error caused by the depth estimation. The experimental results show our reconstructed results are precise and the effect of adjustment is promising.

***Index Terms*—** Multi-camera System, 3D reconstruction, camera calibration, four, five

1. INTRODUCTION

Real-time indoor 3D reconstructions from multiple views have become more and more prevalent recent years [1, 2]. Multi-camera systems using RGBD cameras have many advantages such as a wider horizon compared with the systems based on single camera, and are widely used in many kinds of applications. However, both the intrinsic and extrinsic parameters of the multi-camera systems are required to be calibrated accurately in order to achieve a high-quality and real-time 3D reconstruction of an entire object like people or furniture with less error.

For a single camera system, the accurate intrinsic parameters can be achieved by many internal calibration methods or toolboxes [3, 4]. Many efficient methods have also been developed for the calibration of different types of multi-camera calibration. One widely proposed approach for the multi-camera calibration is to use special calibration objects, such as patterns, rig or similar. Li [5] proposed a toolbox to calibrate the system which uses a feature descriptor based calibration pattern. The pattern can be automatically detected even if the pattern is partially seen in an image. The toolbox yields good results especially on the systems with few cameras (three or four) with minimal overlapping fields of view. Zhao [6] proposed an algorithm based on 1D objects which works well on a triple camera system. The algorithm integrates the rank-4 factorization with the standard 1D camera calibration method and is much more convenient than the plane-based algorithm. Svoboda [7] gives a method only required any bright-spot object like a laser pointer. Waving the object which can be easily detected in each image through the working space is the only work requested. Kalibr [8] is a free toolbox that solves the multiple camera calibration problems. It can get a good estimate but requires that neighbouring cameras have overlapping fields of view. However, if the system contains many cameras and they do not share the same overlapping fields of view, the calibration process should be simply repeated for each pair of cameras.

Another kind of methods which is commonly used is self-calibration. These methods do the calibration using the constraints and correspondences from the images without any special calibration objects. Bundler [9] is a structure-from-motion toolbox using the unordered images captured from different views. The pose of all cameras can be estimated by the 3D reconstruction. It uses SIFT keypoint detector [10] which works well on the outdoor scenery but weak on indoor cases because of the lack of the features. Vasconcelos [11] proposed a solution to calibrate a camera with two other calibrated cameras using independent pairwise point correspondences. Bushnevskiy [12] present a novel approach for the estimation of the geometry of the multi-camera system. The algorithm enforce constraints arising from the visible epipoles and is especially suitable for dome-like indoor cases.

All these methods use RGB images to calibrate the multi-camera system and achieve good results. However, 3D reconstruction system also use other kinds of sensors, like Near Infra-Red cameras (NIR). If there are such cameras within the system used to estimate the depth, the error of the final result of the 3D reconstruction not only comes from the calibration of the RGB cameras but also from the depth estimation. How to minimize the error of the whole reconstruction to achieve an accurate model is a problem.

In this paper, we present an efficient algorithm to optimize the extrinsic camera parameters for indoor 3D reconstruction. To avoid from accumulative error caused by the repetitive calibration for each pair of cameras, we use a global camera calibration method and optimize all the camera parameters consistently. Furthermore, we use the point cloud registration method to adjust the result of the calibration. This can effectively decrease the error result from the depth estimation and obtain a high-quality 3D reconstruction.

2. OVERVIEW

Our algorithm consists of two main parts. Firstly, we use the toolbox Kalibr [8] to calibrate our multi-camera system and achieve the intrinsic and initial extrinsic parameters of all our cameras. Then we use a checkerboard as the calibration object, detect the corners on it and do the global extrinsic parameters optimization, as described in Sec. 3.3. Secondly, we use the camera parameters and the input rgb and depth images to reconstruct the point clouds of the model and use ICP (Iterative Closest Point) [13] to align the point clouds of different views. Then the extrinsic parameters can be refined by the ICP transform to minimize the error caused by depth estimation, as described in Sec. 4. We compute the reprojection error and use a plaster model as ground truth to verify the effectivity of our algorithm, as described in Sec. 5.

3. GLOBAL CAMERA CALIBRATION

The cameras in our 3D reconstruction system are placed on the periphery of the room pointing inwards, only the neighbouring cameras have overlapping fields of view. We found that if we calibrate the cameras pairwisely, accumulative error will occur and can be seen obviously in the result of the reconstruction. After the initial calibration, we optimize the extrinsic parameters globally using a checkerboard.

3.1. Multi-camera System

We employ 8 camera pods around the working space looking inwards for a full capture as shown in Figure 1. Each pod consists of a color camera and 2 Near Infra-Red cameras (NIR). a laser pointer is used to produce special patterns, with it depth images can be achieved by the 2 NIR cameras using depth estimation methods.

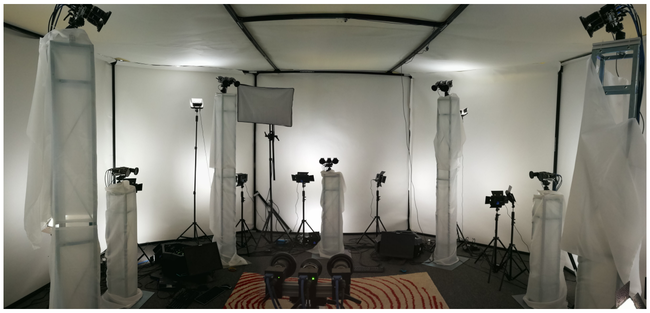


Fig. 1. Our multi-camera system with 8 camera pods pointing inwards.

3.2. Camera Model

Let $\mathbf{P}$ be the 3D space coordinates of a point P, and $\mathbf{p}$ the 2D coordinates of its projection in the image plane, the model of a pin-hole camera is usually described as

$$z_p\mathbf{p} = \mathbf{K_{in}}\mathbf{K_{ex}}\mathbf{P}, \tag{1}$$

where z_p is the projective depth of point P, $\mathbf{K_{in}}$ and $\mathbf{K_{ex}}$ are the intrinsic and extrinsic parameters of the camera.

3.3. Initial Calibration

The $\mathbf{K_{in}}$ and initial $\mathbf{K_{ex}}$ of each camera are calibrated by Kalibr [8] pairwisely because of the lack of the common overlapping fields of view, including the 8 color cameras' extrinsic parameters $\{\mathbf{K_{ex}}_i\}_{i=1,...,8}$. The extrinsic parameters from the 2 NIR cameras to the color camera in the same camera pod are also achieved, but these extrinsic parameters are fixed and used to estimate the depth in each view.

3.4. Global Optimization

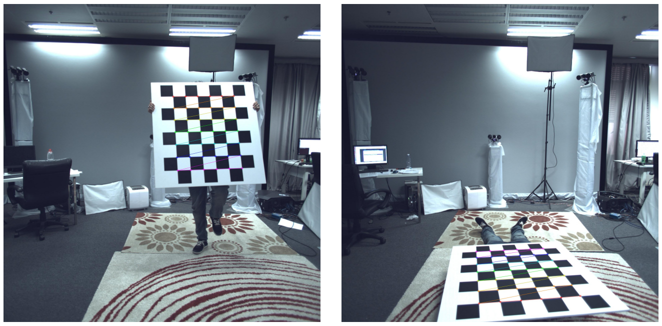
We use a 6 × 7 checkerboard with squares of 117 mm for the optimization. The user is requested to move the checkerboard in front of the cameras freely in the working space. The checkerboard can be simultaneously seen in three or four views usually. Then the 42 corners on the checkerboard can be detected in each group. We can estimate the 3D space coordinates $\mathbf{P_c}$ of each corner from the corresponding 2D coordinates $\{\mathbf{p_c}_j\}_{j=1,...,n}$, where n is the number of the views that the checkerboard can be seen in at the same time. To make the optimization result be more consistent to the whole system, we also use the images with checkerboard on the ground which allows the checkerboard corners can be seen in all the 8 cameras. This will help to reduce the accumulative error and inconsistence caused by the pairwise calibration. The aim of the optimization is to minimize the sum of all reprojection errors with respect to all 3D point and extrinsic parameters, which is described as

$$E(\{\mathbf{K_{ex}}_i\}) = \sum_i \sum_j ((\mathbf{K_{in}}_i \mathbf{K_{ex}}_i \mathbf{P_c}_j) - \mathbf{p_c}_{ij})^2. \tag{2}$$

We use an open source software package called sba to solve this problem [14].

4. REFINEMENT BY POINT CLOUD REGISTRATION

We achieve the depth data from the 2 NIR cameras using depth estimation methods. Although after the global optimization, we can get a result with less reprojection error, but it may not be the optimal solution to the 3D reconstruction beacause the quality of depth estimation also influence the final result. If the depth and camera parameters are both accurate



(a) (b)

Fig. 2. Results of the checkerboard corners detection. (a): Results when the user moves freely. (b): Results when the checkerboard is on the ground.

enough, the point clouds of different views should align very well. However, the error cannot be avoided completely. To minimize the error from depth data and get a high-quality reconsturction, We use ICP to refine the extrinsic parameters by point cloud registration.

For each point reconstructed by the depth image, we consider an estimation

$$\tilde{\mathbf{P}}_{ij} = f_i(\mathbf{P}_j), \qquad (3)$$

where $\mathbf{P}_j$ is the true 3D coordinates in the camera coordinate system. f_i is a nonlinear function for view i, represent the depth influence. $\tilde{\mathbf{P}}_{ij}$ is the 3D coordinates we get from the depth data and intrinsic parameters in view i. With the extrinsic parameter, we can transform all the points into the world coordinate system. The estimation can be written as

$$\hat{\mathbf{P}}_{gj} = g(\mathbf{K}_{\mathbf{ex}i}) f_i(\mathbf{P}_j), \qquad (4)$$

where $\hat{\mathbf{P}}_{gj}$ is the 3D coordinates in the world coordinate system we reconstruct from the depth and camera parameters. ICP can be replaced as a rigid transform to all views

$$\tilde{\hat{\mathbf{P}}}_{gj} = M_i g(\mathbf{K}_{\mathbf{ex}i}) f_i(\mathbf{P}_j) = g(\mathbf{K}_{\mathbf{ex}i}) M_i f_i(\mathbf{P}_j), \qquad (5)$$

where $\tilde{\hat{\mathbf{P}}}_{gj}$ is the coordinates of the point after the alignment. The transform M_i can refine the error caused by f_i, and improve the quality of the result of 3D reconstruction.

We choose the point cloud of view 1 as the reference, align the point cloud of view 2 to it, combine the result of the two views, then align the point cloud of view 3 to the combined result and so on. Each ICP process produce a transformation matrix, then we can refine the extrinsic parameters using the rigid transformation.

5. EXPERMENTS

5.1. reprojection error

5.2. plaster model

6. CONCLUSION

7. REFERENCES

[1] Mingsong Dou, Sameh Khamis, Yury Degtyarev, Philip Davidson, Sean Ryan Fanello, Adarsh Kowdle, Sergio Orts Escolano, Christoph Rhemann, David Kim, Jonathan Taylor, et al., "Fusion4d: Real-time performance capture of challenging scenes," *ACM Transactions on Graphics (TOG)*, vol. 35, no. 4, pp. 114, 2016.

[2] Sergio Orts-Escolano, Christoph Rhemann, Sean Fanello, Wayne Chang, Adarsh Kowdle, Yury Degtyarev, David Kim, Philip L Davidson, Sameh Khamis, Mingsong Dou, et al., "Holoportation: Virtual 3d teleportation in real-time," in *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*. ACM, 2016, pp. 741–754.

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

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

[5] Bo Li, Lionel Heng, Kevin Koser, and Marc Pollefeys, "A multiple-camera system calibration toolbox using a feature descriptor-based calibration pattern," vol. 8215, no. 2, pp. 1301–1307, 2013.

[6] Zijian Zhao and Yuncai Liu, "Practical multi-camera calibration algorithm with 1d objects for virtual environments," in *Multimedia and Expo, 2008 IEEE International Conference on*. IEEE, 2008, pp. 1197–1200.

[7] Tomáš Svoboda, Daniel Martinec, and Tomáš Pajdla, "A convenient multicamera self-calibration for virtual environments," *PRESENCE: teleoperators and virtual environments*, vol. 14, no. 4, pp. 407–422, 2005.

[8] J Maye, P Furgale, and R Siegwart, "Self-supervised calibration for robotic systems," in *Intelligent Vehicles Symposium*, 2013, pp. 473–480.

[9] Noah Snavely, Steven M Seitz, and Richard Szeliski, "Photo tourism: exploring photo collections in 3d," in *ACM transactions on graphics (TOG)*. ACM, 2006, vol. 25, pp. 835–846.

[10] David G Lowe, “Distinctive image features from scale-invariant keypoints,” *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[11] Francisco Vasconcelos, João Pedro Barreto, and Edmond Boyer, “A minimal solution for camera calibration using independent pairwise correspondences,” in *European Conference on Computer Vision*. Springer, 2012, pp. 724–737.

[12] Andrey Bushnevskiy, Lorenzo Sorgi, and Bodo Rosenhahn, “Multicamera calibration from visible and mirrored epipoles,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3373–3381.

[13] Paul J. Besl and Neil D. Mckay, *A Method for Registration of 3-D Shapes*, IEEE Computer Society, 1992.

[14] M.I. A. Lourakis and A.A. Argyros, “SBA: A Software Package for Generic Sparse Bundle Adjustment,” *ACM Trans. Math. Software*, vol. 36, no. 1, pp. 1–30, 2009.