Comparison of different multiple camera calibration methods

Viktor Devecseri[1], Levente Raj, Dr. Róbert Tamás Fekete, *Budapest University of Technology and Economics* (**01.02.2012**, prof. Péter Korondi, DSc, *Budapest University of Technology and Economics*)

[1] MTA-ELTE Comparative Ethology Research Group

Abstract

Marker based optical motion capture systems use multiple cameras to determine the 3D position of markers. The precise knowledge of the position and orientation of the cameras plays a crucial role in precise marker recognition and position calculation. There are three camera calibration methods presented in this paper, including a new projector based method. The three calibration methods give different precision. Results of measurements will be presented and compared.

1. Introduction

Motion Capture systems are used to record the movement of actors and to translate the captured movement on to a digital model. These systems are used in the video game and movie industry. Other uses include military, entertainment, sports, medical applications, computer vision and robotics. There are multiple motion capture technologies available, but this article focuses on marker based optical motion capture systems.

The system used in this article is a custom system developed by us. There are 10 cameras placed around a 4.4x3.4x2 meters room. The exposure time of the cameras are lowered, so only the self-illuminated markers are visible on the images. The camera images are processed real-time on 5 computers by utilizing the GPUs to extract the marker positions. Based on that data the 3D position of the markers are reconstructed.

For high quality results precise knowing of the camera parameters are required, which can be determined with camera calibration. This article compares multiple camera calibration methods, and presents the results.

In section 2 the camera parameters and the 3D reconstruction method is presented. Section 3 details the calibration methods, and compares these by measurements.

2. Point reconstruction

2.1 Camera parameters

A camera can be described with two sets of parameters: intrinsic and extrinsic parameters.

The *intrinsic parameters* describe the internal structure of a camera: principle point (c_x, c_y), the focal length (f_x, f_y), and the distortion [2]. In the camera coordinate system the camera is placed at the origo and it faces towards the $\mathbf{+Z}$ axis. The (1) equations define the projection between a point (*X*, *Y*, *Z*) in the camera coordinate system, and the pixel coordinates (*x*, *y*) on the image plane (Fig.1).

$$x = f_x\left(\frac{X}{Z}\right) + c_x \qquad (1)$$
$$y = f_y\left(\frac{Y}{Z}\right) + c_y$$

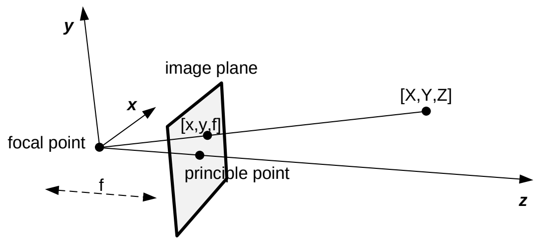


Fig.1. Projection in the camera coordinate system

The distortion has two main components: radial-distortion (k_1, k_2, k_3), and tangential distortion (p_0, p_1). Equations (2) define the transformation from pixel coordinates (x, y) to normalized pixel coordinates ($\hat{x}$, $\hat{y}$).

$$\hat{x} = \frac{x - c_x}{f_x} \qquad (2)$$
$$\hat{y} = \frac{y - c_y}{f_y}$$

The distortion can be described in normalized pixel coordinates, where $r = \sqrt{\hat{x} + \hat{y}}$. The offset of a point caused by the radial distortion is the following (3):

$$dr_x = \hat{x}(k_1 r^2 + k_2 r^4 + k_3 r^6)$$
$$dr_y = \hat{y}(k_1 r^2 + k_2 r^4 + k_3 r^6) \tag{3}$$

The offset of a point caused by the tangential distortion is defined by equations (4):

$$dp_x = 2p_1\hat{y} + p_2(r^2 + 2\hat{x}^2)$$
$$dp_y = p_1(r^2 + 2\hat{y}^2) + 2p_2\hat{x} \tag{4}$$

Finally, the position of a point after distortion in normalized pixel coordinates ($\hat{x}'$, $\hat{y}'$) can be calculated with equations (5):

$$\hat{x}' = \hat{x} + dr_x + dp_x$$
$$\hat{y}' = \hat{y} + dr_y + dp_y \tag{5}$$

The distortion of one of the cameras can be seen on Fig.2.

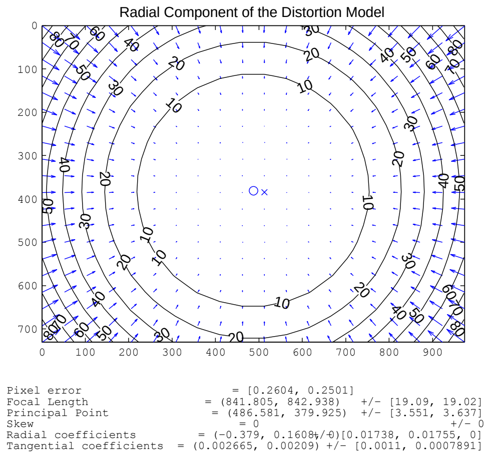


Fig.2. Distortion of one camera

The *extrinsic parameters* describe the camera pose in the world coordinate system with position and orientation [2]. By using these parameters a point in the world coordinate system can be transformed to the camera coordinate system, where the projection and distortion can be applied to the point to get the resulting pixel coordinates.

2.2 Point reconstruction

From a camera image, the marker position on the image plane can be determined with image processing. After that, the marker position must be undistorted.

By originating a ray ($\boldsymbol{d_1}$) from the camera focal point ($\boldsymbol{o_1}$) directed towards the 2D point ($\boldsymbol{p_1}$) on the image plane we get a half-line. All of the points of the half-line projects to the same point on the image plane.

The intersection of at least two different rays give the 3D position ($\boldsymbol{p}$) of the marker (Fig.3). However due to inaccurate measurements or noisy data, usually these rays do not intersect each other [1]. The point closest to all the rays approximates the 3D position of the marker. This can be calculated by solving the system of linear equations (6) in the least squares manner with SVD.

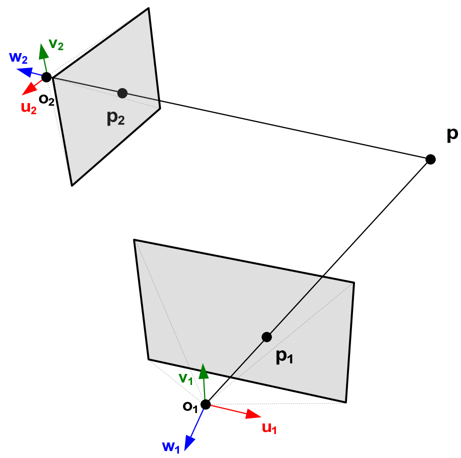


Fig.3. Point reconstruction

$$o_1 + t_1 \cdot \boldsymbol{d_1} = \boldsymbol{p}$$
$$o_2 + t_2 \cdot \boldsymbol{d_2} = \boldsymbol{p} \tag{6}$$

2.3 Point correspondence

When multiple markers are visible, the points must be matched on different image planes in order to successfully reconstruct the 3D positions.

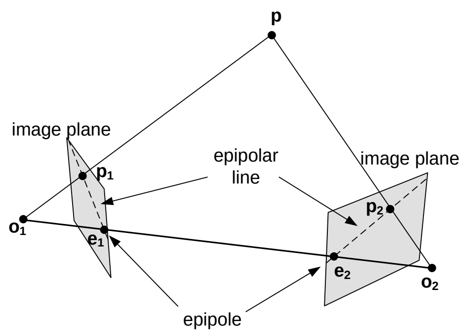


Fig.4. Epipolar geometry

Epipolar geometry simplifies this task (Fig.4). The focal point of a camera ($\boldsymbol{o_1}$, $\boldsymbol{o_2}$) projected onto the other image plane is the *epipole* ($\boldsymbol{e_1}$, $\boldsymbol{e_2}$). The line that connects the epipole with the marker projection ($\boldsymbol{p_1}$, $\boldsymbol{p_2}$) is the *epipolar line*. This is also the projection of the line connecting the marker ($\boldsymbol{p}$) and it's image ($\boldsymbol{p_1}$) on the other camera. The projection of $\boldsymbol{p_1}$ on the other camera lies on the corresponding epipolar line ($\boldsymbol{e_2}$, $\boldsymbol{p_2}$). This is the *epipolar constraint* [2].

Point matching is simplified into a 1D search by using the epipolar constraint. For a given point on one camera, the corresponding point on the other camera must lie on the corresponding epipolar line.

However due to inaccurate calibration the projected point and the epipolar line will not coincide, so the region of the epipolar line must be searched.

3. Calibration methods

In section 2 the 3D point correspondence and 3D reconstruction was presented. It can be seen that the camera calibration highly influences both the correct point matching between different image planes, and the 3D reconstruction of the points.

3.1 Basic method

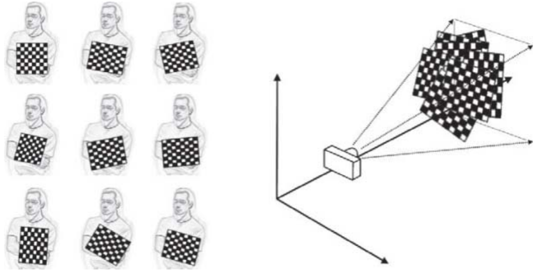


Fig.5. Intrinsic camera calibration with OpenCV

OpenCV [3] has a built in method for finding both the intrinsic and extrinsic parameters of a camera.

The intrinsic parameters can be found from multiple images of a chessboard pattern (Fig.5). The algorithm requires known 3D points of the calibration object, which are the inner corners of the chessboard. Also it requires the pixel coordinates of these points. The 3D points and the 2D points are matched with each other. The outputs are the intrinsic parameters, and the extrinsic parameters for every image [3].

The algorithm first estimates the initial values of the intrinsic parameters. For every image it computes the camera pose, or extrinsic parameters based on the initial values of intrinsic parameters. Finally it uses the Levenberg-Marquardt optimization algorithm to minimize the reprojection error by adjusting both the intrinsic and extrinsic parameters. In this step the algorithm projects the 3D points of the calibration pattern to the image by using both the estimated intrinsic and extrinsic parameters. It calculates the distance between the projected point and the input of the 2D points. Then both the intrinsic and extrinsic parameters are adjusted to decrease the reprojection error [3]. The Levenberg-Marquardt algorithm [5] is a non-linear optimization method, it numerically finds the local minimum and uses both the Gauss-Newton and gradient descent methods.

Technically at least 2 different views are required with a 5x5 chessboard. However, in practice for high quality results 10 different images of a 7x8 chessboard is required due to noise and numerical stability [2]. On Fig.6 a camera image can be seen before and after the intrinsic calibration.

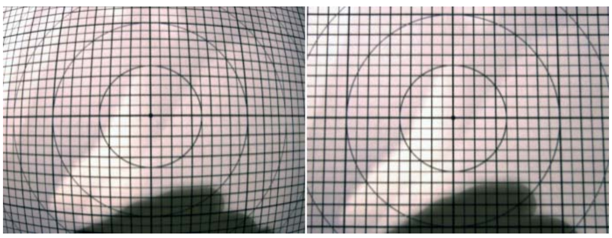


Fig.6. Camera distortion before and after calibration

After the intrinsic parameters are known the extrinsic parameters can be found similarly. By giving at least 4 different points both in pixel coordinates and in the world space the extrinsic parameters can be computed by using OpenCV. The same algorithm is used as the algorithm for the extrinsic parameter estimation in the intrinsic calibration method.

3.2 Projector based method

The problem with the previous method is that it is easy to introduce noise into measurements. This is especially true for the extrinsic calibration where it is hard to click on the exact pixel. So the calibration algorithm works with noisy data.

Both the intrinsic and the extrinsic calibration are automated. For intrinsic calibration a big dot is shown on the monitor, which is captured by a camera which faces the monitor. From the captured image the point of gravity of the shown blob is extracted. By taking multiple images the extracted positions are averaged. Both the points of the calibration object (3D positions) and their pixel coordinates in sub-pixel precision are known exactly. After this points are gathered the same algorithm can be used to calculate the intrinsic parameters.

The extrinsic parameters are calibrated in the same way. A projector was installed on the ceiling which projects the circles on the floor in known positions. The same way both the 3D positions and their 2D positions are known. The extrinsic parameters are calculated with the same algorithm used in the previous method, but less noise was added to the measurements due to automation.

However, the extrinsic calibration algorithm minimizes the error in the known positions which are at the floor level. The distance between the rays used for reconstructing the marker positions are minimal at this level of height. But at higher positions, e.g. at 1.5 meters up from the ground, the distance between the rays grows. This can result in wrong point correspondence where the markers are near to each other. To solve this problem a 1 meter height table was put in the active space and the projected points were measured both at the floor and at the table level. That way the camera parameters were optimized to minimize the error at both level of heights. This decreased the average distance between the rays in the whole active area.

3.3 Bundle Adjustment

In the previous methods the cameras were calibrated separately, not as a system.

By using the results of the previous methods the camera parameters can be further refined. A marker is moved around in the whole active area (*wanding*) and the 2D and estimated 3D positions were captured. The camera parameters can be further refined with the Levenberg-Marquardt algorithm [5] by minimizing the reprojection error on all the cameras. That way all the cameras are considered as a system and the whole system is optimized globally. When the whole active area is covered by the marker both the intrinsic and extrinsic camera parameters are tuned that the projection rays distance is the smallest in the area. For bundle adjustment we used the SBA library [6].

3.4 Comparison

For measurements a marker was moved around in the whole active area for about 2 minutes. Then both the reprojection error, and the distance between the reconstructed point and the projection rays were calculated for every sample, and the summed squared error was calculated. The results are shown on table 1.

Tab.1.

Comparison of the calibration methods

	reprojection error [px]	average ray-point distance [cm]
manual	15,598	4,066
floor	4,077	1,324
floor + table	2,282	0,780
bundle adj.	0,705	0,233

Intrinsic calibration have given very good results with the first method. The other methods significantly didn't influence the intrinsic parameters, so the errors come from the inaccurate extrinsic calibration.

The first, manual method gives poor results. This is due that the user must manually select the pixels with the known positions. The second method, where a projector projected the calibration pattern, gives better results especially at the level of the floor. However for the whole active area the average ray distance is about 1.3 cm. By using the same method with a table at a known height, the error is halved.

Finally with bundle adjustment the best result can be achieved. The average ray-point distance is under 0.5 cm in the whole active area.

4. Conclusion

The paper have covered four different calibration methods for optical motion systems. We use the combination of the first manual method and bundle adjustment to fully calibrate the camera system with less than 0.5 cm error.

However, camera calibration can be thought also as art besides science, because some unexpected data can badly influence the bundle adjustment. At that step local minimization is done, and if the starting values are bad, the algorithm can drift into a local minimum far from the global minimum. Also some noise during sampling can influence badly the error of the whole measurement, so the results are again far from the global minimum.

Accurate camera calibration requires some manual labor as well beside the automatic methods.

Bibliography

[1] Dr. Szirmay-Kalos László and Antal György and Csonka Ferenc: *Háromdimenziós grafika, animáció és játékfejlesztés*, ComputerBooks 2003.

[2] Gary Bradski and Adrian Kaehler: *Learning OpenCV: Computer Vision with the OpenCV Library*, O'Reilly, 2008.

[3] Bradski, G.: *The OpenCV Library*, Dr. Dobb's Journal of Software Tools, 2000. URL: http://opencv.org/

[4] Bouguet, J. Y.: Camera calibration toolbox for Matlab, 2008, URL: http://www.vision.caltech.edu/bouguetj/calib_doc/

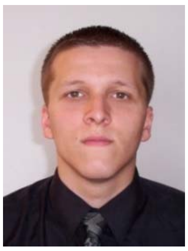
[5] Kenneth Levenberg: *A method for the solution of certain non-linear problems in least squares*, Quarterly Journal of Applied Mathmatics 2, 164-168, 1944.

[6] M.I. A. Lourakis and A.A. Argyros: SBA: *A Software Package for Generic Sparse Bundle Adjustment*, ACM Trans. Math. Software, 36(1): pp. 1-30, 2009. URL: http://users.ics.forth.gr/~lourakis/sba/

Authors:

MSc. Viktor Devecseri
Budapest University of Techno-logy and Economics
Bertalan Lajos Street 4-6
1111 Budapest
tel. (36) 1 463-2602
fax (36) 1 463-3787
email: *devi@mogi.bme.hu*

MSc. Levente Raj
Budapest University of Techno-logy and Economics
Bertalan Lajos Street 4-6
1111 Budapest
tel. (36) 1 463-2602
fax (36) 1 463-3787
email: *raj@mogi.bme.hu*



Dr. Róbert Tamás Fekete
Budapest University of Techno-logy and Economics
Bertalan Lajos Street 4-6
1111 Budapest
tel. (36) 1 463-2602
fax (36) 1 463-3787
email: *frt@mogi.bme.hu*