D-Calib: Calibration Software for Multiple Cameras System

Yuko Uematsu Tomoaki Teshima Hideo Saito
Keio University, Yokohama, Japan
{yu-ko, tomoaki, saito}@ozawa.ics.keio.ac.jp

Cao Honghua
Library Inc., Japan
cao@library-inc.co.jp

Abstract

This paper presents calibration software "D-Calib" for multiple cameras, which can calibrate all cameras at the same time with easy operation. Our calibration method consists of two steps: initialization of camera parameters using planar marker pattern, optimization of initial parameters by tracking two markers attached to a wand. By weak calibration and multi-stereo reconstruction by multiple cameras, the initial parameters can be optimized in spite of few markers. Even if the cameras are placed in a large space, the calibration can be achieved by swinging the wand around the space. This method is completed as commercial software which achieves 0.27 % of average errors against the marker pattern size.

1 Introduction

Estimation process of camera's unique parameters (such as focal length and skew parameters) is called as "camera calibration" [4]. This calibration process is very important issue for 3D reconstruction, 3D motion capture, and virtualized environment, etc. [3, 5].

DLT (Direct Linear Transformation) method is commonly used as the typical calibration method [2]. In DLT method, multiple markers whose 3D coordinates are already known are captured with a camera. Using the pairs of 2D coordinates in the captured images and known 3D coordinates of the markers, twelve camera parameters (DOF is eleven) are computed by least-square-method.

For using DLT method in large space, it is necessary to place the markers to cover the overall space. Increasing the number of markers is also necessary to improve the accuracy of the calibration. However, setting up the markers whose 3D coordinates is known in the large space takes amount of time and effort. Using a lot of cameras, there is also a big problem to place the markers so that all of them can be captured in field of view of the camera and they do not occlude each other.

Moreover, it often happens that a user has to click the display to assign 2D coordinates of markers. Therefore it involves a lot of effort and time when increasing the number of markers and cameras.

Zhang proposed a easy calibration system based on a moving plane where checker pattern is drawn [8]. This calibration method does not need 3D calibration objects with known 3D coordinates. However, since the calibration plane is not visible in all cameras, the user has to move the plane in front of each camera. Moreover the system can calibrate only part of the space. Therefore this system is not suite for using multiple cameras.

Svoboda et al. proposed a calibration method for multiple cameras using one bright point like a laser pointer [7]. The point in the captured images are detected and verified by a robust RANSAC. The structures of the projected points are reconstructed by factorization. These computations can be automatically done. Since one point is used as a calibration object, however, only re-projection errors are evaluated in their method.

In this paper, we introduce calibration software "D-Calib" designed for multiple cameras using in a large space [1]. In this software, we can estimate the camera parameters of all cameras at the same time by using a square pattern with six spheral markers and a wand with two spheral markers. After capturing one frame of the square pattern, the wand swinging in the whole of space is captured for a few hundreds of frames. Therefore we do not have to set up a lot of markers around the space. The spheral markers are made by retroreflective materials so that they can be successfully detected from the captured images. Since all the processes are automatically performed by image processing technique without manual operations, the user's task is extremely reduced.

In our method, initial parameters computed from the square pattern are optimized by using the two markers attached to the wand. Since the distance between the markers is known, we can also optimize 3D reconstruction errors in addition to 2D re-projection errors which are only accuracy evaluation in Svoboda's method.

14th International Conference on Image Analysis and Processing (ICIAP 2007)
0-7695-2877-5/07 $25.00 © 2007 IEEE

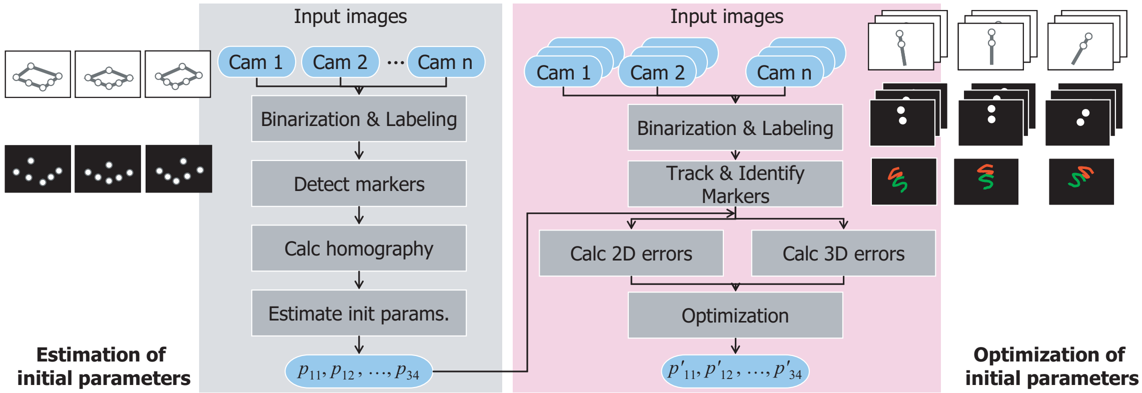


Figure 1. Flowchart of calibration process.

2 Camera Calibration

The relationship between the 3D real world (X, Y, Z) and the 2D image (x, y) is represented as a following equation.

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \simeq \boldsymbol{P} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \simeq \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (1)$$

This 3×4 matrix $\boldsymbol{P}$ is called as a projection matrix, and the twelve elements of the matrix are called as camera parameters. Estimation of these camera parameters corresponds to camera calibration.

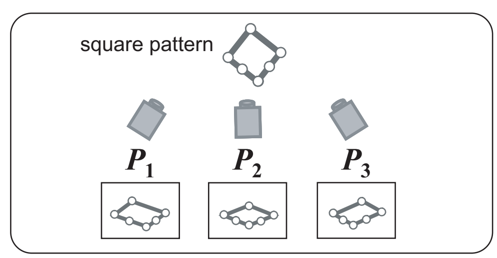
In our software, the twelve camera parameters are computed for each camera. Since we focus on computing accurate extrinsic parameters of multiple cameras, we assume that intrinsic parameters of the cameras are previously given by using other method. If using uncalibrated cameras, the intrinsic parameters are approximated by simple form without skew and distortion parameters as described in Sec. A.1.

3 Our Calibration Method

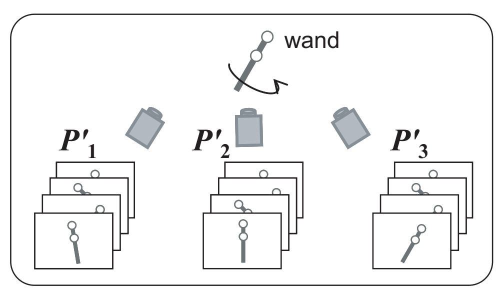
Fig. 1 shows the flowchart of the calibration method in our software. This method consists of two steps: (1) estimation of initial parameters, (2) optimization of initial parameters.

In the estimation of initial parameters, a square pattern with six spheral markers is placed on the floor and is captured by multiple cameras as shown in Fig. 2(a). The real positions of the spheral markers on the square pattern are previously known. Using the relationship between the 2D positions of the markers in captured images and the 3D positions on the markers, camera parameters are computed for each camera as initial parameters. These camera parameters represent twelve elements of a projection matrix $\boldsymbol{P}$ as described in Sec. 2. Actually, the extrinsic parameters of each camera are computed from the square pattern and combined with the intrinsic parameters obtained in advance into the camera parameters.

Since the initial parameters are computed from the square planar pattern, the accuracy against perpendicular direction to the plane is relatively small compared with the other directions. Moreover, high calibration accuracy can be obtained only in the area close to the square pattern. Therefore the initial parameters of every camera should be improved by the optimization at the same time in the next



(a) Capturing images for estimation of initial parameters.



(b) Capturing images for optimization of initial parameters.

Figure 2. System environments in our software.

step.

In the optimization step, we collect a hundreds of frames captured in the multiple cameras as shown in Fig. 2(b), while a wand that has two spheral markers is swung in the entire object space. The positions of the two spheral markers are detected and tracked from the captured image sequences of every camera. Since all the spheral markers are made by retroreflective materials, they can be easily detected from the captured images. By evaluating the estimated extrinsic parameters with epipolar constraints of tracked markers and the 3D distance between the two markers on the wand, the parameters are optimized.

3.1 Estimation of Initial Parameters

A square pattern with six spheral markers is placed on the floor. Then images of the square markers are captured by multiple cameras. From each captured image, the six markers' 2D coordinates are detected as $(x_1, y_1) \sim (x_6, y_6)$. We assume that the plane of the square pattern is X-Y plane in the 3D coordinate. Then we define 3D coordinates of the six markers as $(X_1, Y_1, 0) \sim (X_6, Y_6, 0)$. Based on this definition, the relationship between the 2D and 3D coordinates of the markers can be represented by using a 3×3 planar transformation matrix (Homography) $\boldsymbol{H}$ as following equation. $(1 \leq i \leq 6)$

$$\begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} \simeq \boldsymbol{H} \begin{bmatrix} X_i \\ Y_i \\ 1 \end{bmatrix} \simeq \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} X_i \\ Y_i \\ 1 \end{bmatrix} \quad (2)$$

This equation is equivalent to eq. (1) in Sec. 2, when $Z = 0$ in eq. (1). This means that $\boldsymbol{H}$ provides the first, second, and fourth column vectors of the projection matrix $\boldsymbol{P}$. $\boldsymbol{H}$ is a matrix which transfers points between the planes, and is easily computed by more than four corresponding points on both of the planes.

Accordingly, we compute $\boldsymbol{H}$ between the 3D plane (X-Y plane) and the 2D plane (x-y plane) from six corresponding points which are the spheral markers of the square pattern. Then the twelve camera parameters of each camera are computed from $\boldsymbol{H}$. The detail of the computing process will be described in Sec. A.

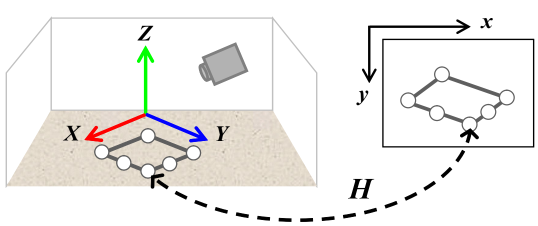


Figure 3. Corresponding of planes by homography.

3.2 Optimization of Initial Parameters

In this process, we define an error function consisting of 2D re-projection errors and 3D reconstruction errors, which is used for optimizing the initial parameters by minimizing the function.

As shown in Fig. 2(b), the images of the swinging wand with two spherical markers are captured for a hundreds frames by the multiple cameras. By detecting and tracking the two spheral markers from the captured image sequences, we can obtain a set of 2D-2D corresponding points between the cameras, which provides epipolar relationship among the cameras.

Using this data set, we can compute 2D re-projection errors of the initial camera parameters, which are evaluated by the distance between the point of the marker in the image and the re-projected epipolar line onto the other cameras. Since the 3D distance between the two markers on the wand are known, we also evaluate the accuracy of the initial camera parameters by comparing the 3D distance between two marker positions recovered from the input images with the real distance between them.

3.2.1 Tracking of Markers

After binarization and labeling are performed to the captured image sequence from each camera, two spheral markers' regions are detected from each image. For tracking the two markers, each marker must be identified through the image sequence.

The identification of each marker is performed by computing the distance of the positions between the previous frame and the next frame. The marker with the closest position to the position in the previous frame is identified as the same marker. Even though the two markers are occluded each other or are not detected in some frames, each marker can be continuously tracked by using direction of marker's movement. Therefore the cameras do not have to see all the markers through the captured images. Only the frames which are captured when the markers are visible in all cameras are automatically selected as a data set of 2D-2D correspondences.

3.2.2 Optimization

Using the 2D coordinates of the tracked markers, two kinds of errors are computed; (a) 2D re-projection errors ϵ_1, (b) 3D reconstruction errors ϵ_2. Error function is defined as eq. (3) and minimized by Down-hill simplex method to optimize the six extrinsic parameters included in the initial

14th International Conference on Image Analysis and Processing (ICIAP 2007)
0-7695-2877-5/07 $25.00 © 2007 IEEE

camera parameters; rotation (3 DOF: θ, ϕ, ψ), translation (3 DOF: t_x, t_y, t_z).

If all cameras have the same intrinsic parameters, it is possible to optimize seven parameters including the focal length f. This is because 2D re-projection errors are not absolute but relative values among the cameras.

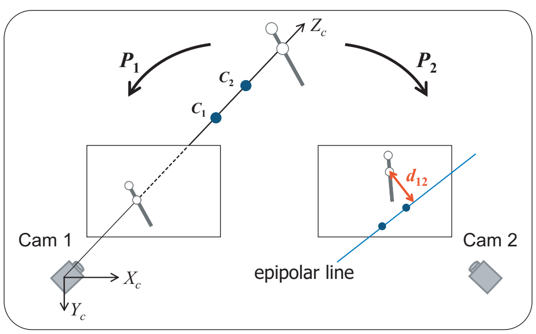
$$\begin{cases} cost(\theta, \phi, \psi, t_x, t_y, t_z) = (\epsilon_1 + \epsilon_2) \\ cost(\theta, \phi, \psi, t_x, t_y, t_z, f) = (\epsilon_1 + \epsilon_2) \end{cases} \tag{3}$$

2D re-projection errors ϵ_1

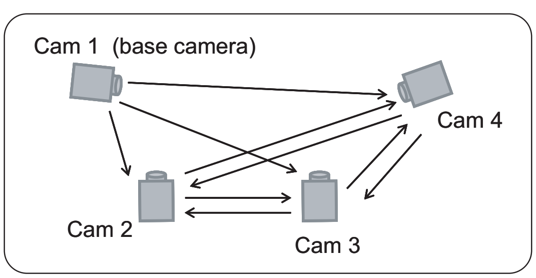
When a set of 2D-2D correspondences between two cameras is known, a point in the image captured by the camera 1 is projected onto the image captured by the camera 2 as a line and should exist anywhere on the line. This is called as "epipolar constraints" and the line is called as "epipolar line".

The marker in the image of the camera 1 is projected onto the image of the camera 2 as a line (epipolar line) by the initial camera parameters as shown in Fig. 4(a). The epipolar line should be projected on the position of the marker which is captured by the camera 2, however, it might be projected away from the marker because of the errors included in the initial parameters. Therefore the distance between the epipolar line and the marker in the image of camera 2 is defined as a 2D re-projection error from the camera 1 to the camera 2.

In the same way, 2D re-projection errors are computed mutually among all the cameras, and then the sum of them becomes ϵ_1. Here, the 2D re-projection errors are computed using epipolar constraints which represent relative geometry between the cameras. Therefore we define one camera as an absolute basis and do not compute the re-projection errors to the base camera as shown in Fig. 4(b).



(a) Errors from epipolar constraints.



(b) Relationship of projection between cameras.

Figure 4. Computation of 2D re-projection errors.

3D reconstruction errors ϵ_2

When camera parameters of two cameras are known, 3D coordinates of an object which is captured by the cameras can be reconstructed by using triangulation from 2D coordinates in the images. In our method, 3D reconstruction errors ϵ_2 are computed by multi-stereo reconstruction of multiple cameras.

When the two markers are captured by each camera as shown in Fig. 5, 3D coordinates of the markers can be obtained by triangulation from the 2D coordinates in each image (see Sec. B).

Computing the 3D coordinates of the markers from all the cameras, 3D distance D between the markers can be computed. Since the real distance s is known, the difference between D and s becomes a 3D reconstruction error at every frame, and then the sum of them becomes ϵ_2.

$$\epsilon_2 = \sum^{frame} (D - s)^2 \tag{4}$$

After computing ϵ_1 and ϵ_2, the initial parameters are optimized by minimizing the error function of eq. (3) by Down-hill simplex method.

4 Experimental Results

In this section, we will show two experimental results of our calibration software; four cameras configuration and

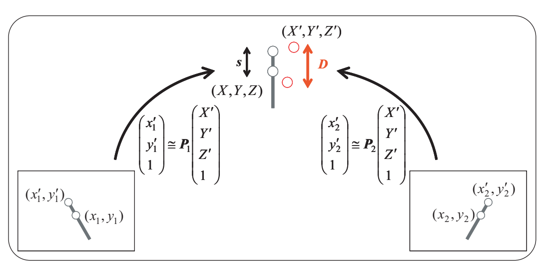


Figure 5. Computation of 3D reconstruction errors.

14th International Conference on Image Analysis and Processing (ICIAP 2007)
0-7695-2877-5/07 $25.00 © 2007 IEEE

(a) Square pattern of markers (b) Wand

Figure 6. Markers used in our software.

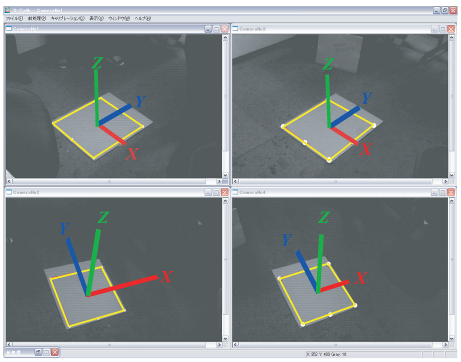


Figure 7. 3D coordinate system defined by initial parameters.

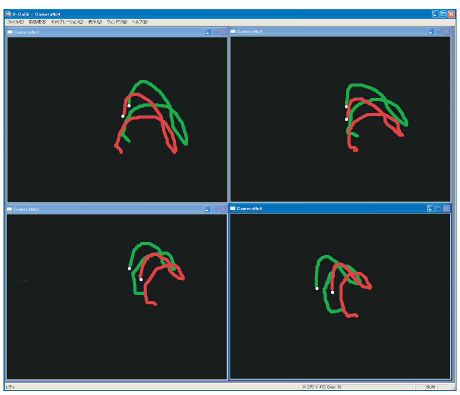


Figure 8. Tracking results of two markers of wand.

six cameras configuration. In both of the experiments, the square pattern with six markers is captured at one frame by each camera to estimate initial camera parameters as shown in Fig. 6(a). Next, the wand swinging in the space is captured for 200 frames by each camera as shown in Fig. 6(b). The size of the space where the wand is swinging is about 2m$\times$2m$\times$2m. In both of the experiments, the size of square pattern marker is 500×500 [mm], the distance between the two markers on the wand is 250 [mm]. The resolution of the captured image is 640×480.

Fig. 7 shows the results of X-Y-Z coordinate systems defined by initial parameters estimated by four cameras. Fig. 8 shows the tracking results of two markers which are attached to the wand by the four cameras. We can see that the two markers are successfully detected and tracked through the captured image sequence without confusing the markers by our tracking method.

Table 1. Evaluation results from optimized parameters.

numbers of cameras		4	6
Minimum error [mm]		0.001	0.069
Maximum error [mm]		4.613	2.008
Average error [mm]		1.345	0.992
Standard deviation		0.895	0.379
Size of calibrated space	X [mm]	789.251	620.695
	Y [mm]	1368.694	755.244
	Z [mm]	735.965	323.614

Table 2. Comparison of results between initial and optimized parameters.

	initial	optimized
Minimum error [mm]	2.816	0.069
Maximum error [mm]	5.965	2.008
Average error [mm]	4.671	0.992
Standard deviation	0.564	0.379

Table 1 shows the evaluation results of errors obtained by optimization of the initial parameters using the tracked markers. These errors represent 3D reconstruction errors, which are computed using optimized camera parameters instead of the initial parameters, in the same way described in Sec. 3.2.2. The error is computed at every frame in the captured image sequence, and then minimum, maximum and average values and standard deviation of the computed errors are listed in Table 1. Table 2 shows each error computed by using the initial and the optimized parameters.

From Tab. 1, we can find that the minimum errors are less than 0.1 [mm] and the maximum error is also about 4.6 [mm]. These results can be considered small enough compared to the size of the space. The errors included in the initial parameters can be reduced by our optimization method as shown in Tab. 2. Moreover, the standard deviation shows that the space can be impartially calibrated by swinging the wand around the space.

5 Conclusions

In this paper we proposed calibration software for multiple cameras. The camera parameters of all the cameras can be estimated at the same time. Since all the processing of this software can be automatically performed without user's manual operations like a assigning the marker's positions by clicking the display, the user can easily use this software. This software can calibrate a lot of cameras which are placed in large space without placing many markers around the space in advance. Therefore it is well suited for the task which needs large space like 3D motion capture.

A Calculation P from H

A projection matrix consists of intrinsic and extrinsic parameters. Therefore we obtain a projection matrix by computing the intrinsic and the extrinsic parameters from the homography and combining them [6].

We define intrinsic parameters, a rotation matrix and a translation vector of extrinsic parameters as $\boldsymbol{A}$, $\boldsymbol{R}$ and $\boldsymbol{t}$. Then a projection matrix $\boldsymbol{P}$ will be

$$\boldsymbol{P} = \boldsymbol{A}\,[\boldsymbol{R} \mid \boldsymbol{t}] = \boldsymbol{A}\,[\boldsymbol{r_1}\ \boldsymbol{r_2}\ \boldsymbol{r_3}\ \boldsymbol{t}] \tag{5}$$

$$\boldsymbol{A} = \begin{bmatrix} f & 0 & c_x \\ 0 & f & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{6}$$

$\boldsymbol{r_1}$, $\boldsymbol{r_2}$, $\boldsymbol{r_3}$ are column vectors of the rotation matrix $\boldsymbol{R}$, f is the focal length, (c_x, c_y) is a center of the image. From eq. (2),

$$\boldsymbol{P} = \boldsymbol{A}\,[\boldsymbol{r_1}\ \boldsymbol{r_2}\ \boldsymbol{t}] = \boldsymbol{H} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \tag{7}$$

$$\boldsymbol{A}^{-1}\boldsymbol{H} = [\boldsymbol{r_1}\ \boldsymbol{r_2}\ \boldsymbol{t}] \tag{8}$$

Using the homography $\boldsymbol{H}$ computed from six corresponding points of spheral markers, the intrinsic and extrinsic parameters are computed from eq. (8). When both the intrinsic and extrinsic parameters are obtained, the projection matrix $\boldsymbol{P}$ is obtained by (5).

A.1 Computation of Intrinsic Parameters

When using uncalibrated cameras, intrinsic parameters of the cameras have to be computed. Since we define $\boldsymbol{A}$ as eq. (6), we only have to obtain a focal length f. Using the property of the rotation matrix $\boldsymbol{R}$ that inner product of $\boldsymbol{r_1}$ and $\boldsymbol{r_2}$ is 0, f is obtained by developing eq. (8) as follows.

$$f^2 = \frac{(h_{11} - c_x h_{31})(h_{12} - c_x h_{32})}{-h_{31}h_{32}} + \frac{(h_{21} - c_y h_{31})(h_{22} - c_x h_{32})}{-h_{31}h_{32}} \tag{9}$$

A.2 Computation of Extrinsic Parameters

The first and second column vectors $\boldsymbol{r_1}$, $\boldsymbol{r_2}$ of the rotation matrix $\boldsymbol{R}$ and the translation vector $\boldsymbol{t}$ are already obtained when $\boldsymbol{H}$ is computed from eq. (8). Therefore we only have to compute the third column vector $\boldsymbol{r_3}$ of $\boldsymbol{R}$. Also using the property of $\boldsymbol{R}$ that the cross product of $\boldsymbol{r_1}$ and $\boldsymbol{r_2}$ becomes $\boldsymbol{r_3}$, we obtain it as follows.

$$\boldsymbol{R} = [\boldsymbol{r_1}\ \boldsymbol{r_2}\ \boldsymbol{r_3}] = [\boldsymbol{r_1}\ \boldsymbol{r_2}\ (\boldsymbol{r_1} \times \boldsymbol{r_2})] \tag{10}$$

B Multi-Stereo Reconstruction

When camera parameters of multiple cameras are known, a 3D coordinate of a point (X, Y, Z) can be computed from following equation using 2D coordinates of the corresponding points (x_i, y_i) in each camera.

$$\begin{bmatrix} (p^1_{31}x_1 - p^1_{11}) & (p^1_{32}x_1 - p^1_{12}) & (p^1_{33}x_1 - p^1_{13}) \\ (p^1_{31}y_1 - p^1_{11}) & (p^1_{32}y_1 - p^1_{12}) & (p^1_{33}y_1 - p^1_{13}) \\ \vdots & & \\ (p^n_{31}x_n - p^n_{11}) & (p^n_{32}x_n - p^n_{12}) & (p^n_{33}x_n - p^n_{13}) \\ (p^n_{31}y_n - p^n_{11}) & (p^n_{32}y_n - p^n_{12}) & (p^n_{33}y_n - p^n_{13}) \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} p^1_{14} - p^1_{34}x_1 \\ p^1_{14} - p^1_{34}y_1 \\ p^n_{14} - p^n_{34}x_n \\ p^n_{14} - p^n_{34}y_n \end{bmatrix} \tag{11}$$

References

[1] http://www.library-inc.co.jp/.

[2] Y. I. Abdel-Aziz and H. M. Karara. Direct linear transformation from comparator coordinates into object space coordinates in close-range photogrammetry. In *Proc. of Symposium on Close-Range Photogrammetry*, pages 1–18, 1971.

[3] G. K. M. Cheung, S. Baker, and T. Kanade. Shape-from-silhouette of articulated objects and its use for human body kinematics estimation and motion capture. In *Proc. of Computer Vision and Pattern Recognition*, pages 77–84, June 2003.

[4] R. Hartley and A. Zisserman. *Multiple View Geometry in computer vision*. CAMBRIDGE UNIVERSITY PRESS, 2000.

[5] H. Saito, S. Baba, and T. Kanade. Appearance-based virtual view generation from multi-camera videos captured in the 3d room. *IEEE Transactions on Multimedia*, 5(3):303–316, September 2002.

[6] G. Simon, A. Fitzgibbon, and A. Zisserman. Markerless tracking using planar structures in the scene. In *Proc. of the ISAR*, pages 120–128, 2000.

[7] T. Svoboda, D. Martinec, and T. Pajdla. A convenient multi-camera self-calibration for virtual environments. *Presence*, 4(14):407–422, August 2005.

[8] Z. Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(22):1330–1334, 2000.