A Multi-view Camera Tracking for Modeling of Indoor Environment

Kiyoung Kim and Woontack Woo

GIST U-VR Lab.
Gwangju 500-712, S.Korea
{kkim,wwoo}@gist.ac.kr

Abstract. In this paper, we propose a method to track a multi-view camera for modeling indoor environment without calibration patterns. A multi-view camera is more convenient for modeling background or objects in speed and usage than expensive 3D scanner. However it requires a good initial pose and motion of a multi-view camera because the initial pose has an effect on overall accuracy. Thus, we use structural constraints of a multi-view camera and coplanar calibration pattern to provide a good initial poses. Then, we estimate camera motion by calculating rigid-body transformation between corresponding 3D points in each point clouds set. Finally we perform bundle adjustment in order to optimize all poses of the camera. Since it gives absolute camera motion in a room without scene constraints, the proposed technique is more useful than conventional pose estimation for modeling indoor environment. The proposed method can be used to accurately augment virtual objects.

1 Introduction

Camera calibration is the process that determines the relation between a world coordinate system and a camera coordinate system [1]. The relation is represented by camera parameters, and enables us to rectify images and reconstruct 3D background [2]. Especially, the pose of a camera including position and orientation plays an important role in registering initial point clouds for modeling indoor environment with a multi-view camera [3].

Many researchers have proposed various camera calibration algorithms in order to obtain camera parameters accurately. Tsai proposed two-step optimization algorithm that exploits an accurate non-coplanar pattern called Tsai's grid [1]. It is known as the first algorithm to consider radial coefficient of a lens distortion. However, a precise right angle pattern is required to get the correct results. Zhang [4], Sturm [5], etc., overcame this constraint by using homograpies of several images (more than three). Zhang proposed a flexible calibration method with a coplanar pattern for a desktop vision system (DVS) [4]. Calibration can be done by capturing feature points in a coplanar pattern at different positions. However, it requires several images in order to obtain accurate results [4].

The process of obtaining extrinsic parameters, which determine pose of a camera, with calibration patterns is inconvenient in modeling of indoor environment. The accuracy of camera poses largely depends on how precise calibration

K. Aizawa, Y. Nakamura, and S. Satoh (Eds.): PCM 2004, LNCS 3331, pp. 288–297, 2004.

pattern is. Practically, it is difficult to make a calibration pattern with the required precision. In fact, it is more difficult than camera calibration because it uses only one image [6].

In this paper, we propose a method to track a multi-view camera for modeling of indoor environment without calibration patterns. Since the initial pose of a multi-view camera has an effect on overall accuracy, we use structural constraints of a multi-view camera to iteratively optimize an initial pose of a camera with known calibration pattern. This gives the absolute camera position in the indoor environment. Based on the initial position, we estimate camera motion by calculating rigid-body transformation between corresponding 3D points in each point clouds set. Finally, we apply bundle adjustment in order to optimize all poses of the camera. Thus, we can know the absolute motion of the camera in a room.

The proposed method provides a good initial pose with respect to a user-defined orientation in indoor environment. Since it only uses point clouds, it does not need calibration pattern to track a multi-view camera. It is useful for registration in an indoor environment object modeling, and it can be used to accurately augment virtual objects.

This paper is organized as follows: Camera model and the rigid-body transformation are introduced in Chapter 2. The method to find the initial pose and estimate motion by using 3D homographies is explained in Chapter 3. Then we show the experimental results of a proposed method in Chapter 4. We present the conclusion and future work in Chapter 5.

2 Camera Model and Rigid-Body Transformation

In a world coordinate system, a point, $M = [X, Y, Z]^T$, is projected onto $m = [u, v]^T$ in image plane by projection matrix P following simple pinhole camera model. Projection matrix consists of the product between intrinsic parameters and extrinsic parameters [2]. It is 3×4 matrix which has 11 degree of freedom.

$$\alpha \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = P \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}, P = A \begin{bmatrix} R\, t \end{bmatrix} \tag{1}$$

where, α is an arbitrary scale factor, R is a 3×3 rotation matrix and $t = [t_x, t_y, t_z]^T$ is 3×1 translation matrix. $[Rt]$ is a transformation matrix between a world coordinate system to a camera coordinate system. A is a special matrix composed of intrinsic parameters which shape is as follows.

$$A = \begin{bmatrix} -fk_u & fk_u \coth\theta & u_0 \\ 0 & \frac{-fk_v}{\sin\theta} & v_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2}$$

where, f is a focal length, k_u and k_v are horizontal and vertical scale factors, respectively. (u_o, v_o) is a principal point where z axis meets image plane. θ

represents the angle between X and Y axis in retinal plane. In practice, θ is almost 90 degree, thus $fk_u \cot\theta$ becomes 0 and $\sin\theta$ has 1 value, ideally. In the 3D space, camera motion can be represented as sequential transformations. Change of coordinate system in the 3D space can be rigid-body transformation, 4×4 homography. Its homogeneous representation is shown in equation (3).

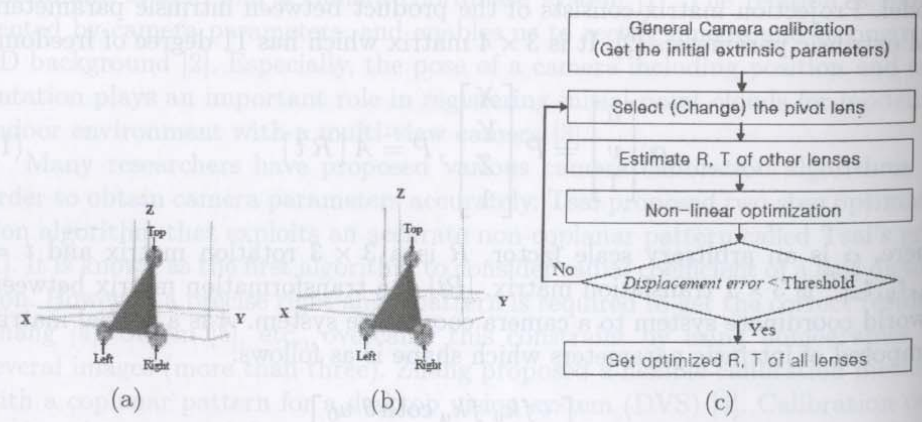
$$L_j = H_{i\to j} L_i = \begin{bmatrix} R_{i\to j} & t_{i\to j} \\ 0 & 1 \end{bmatrix} L_i \tag{3}$$

where, $H_{i\to j}$ indicates a homography including rotation and translation matrices which transform L_i coordinate system to L_j coordinate system.

3 Pose Optimization and Motion Estimation

3.1 Initial Pose Optimization

We introduce the pose optimization method of a multi-view camera. The proposed method uses structure constraints of a multi-view camera. Coplanar pattern is used in determining orientation of a world coordinate system. Practically, the pose estimation with general calibration algorithm using coplanar pattern makes considerable errors. We found that the reconstructed shape of a multi-view camera with general camera calibration methods is much different from its original one shown in Fig. 1 (a), (b). We propose the optimization algorithm reducing those errors and overall procedure is shown in Fig. 1 (c). A multi-view camera is an extension of a general stereo camera. It has more than 2 lenses in one camera body. Generally, the poses of those lenses are determined with a small displacement error through lens alignment process in the factory. Initial



(a) (b) (c)

Fig. 1. Calibration error and overall procedure for optimizing poses of inner lenses (a) original shape of camera (3 lenses: top, left, right) (b) reconstructed shape of camera by general camera calibration (c) overall procedure for reducing calibration errors

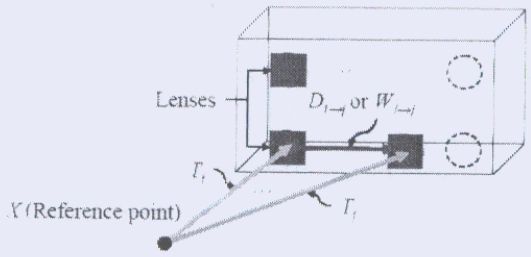


Fig. 2. Multi-view camera structure and displacement of inner lenses

displacements, then, were tested by using several calibration algorithms. Finally, manufacturers provide optimal intrinsic parameters and baselines so that it is possible to give disparity map. The important thing we focus on is that the distance among lenses are preserved whenever the camera is moving. We use displacement error including translation and rotation error as a cost function shown in equation (4).

$$\underbrace{R_{error} = \sum_{i=0, i\neq j}^{L-1} \|r_{i,j} - R_{i,j}\|, T_{error} = \sum_{i=0, i\neq j}^{L-1} \|G_{i,j} - S_{i,j}\|}_{E = R_{error} + \alpha T_{error}} \tag{4}$$

where, L denotes combination number between lenses. α is a scale factor. $G_{i,j}$ and $R_{i,j}$ are ideal translation and rotation matrix between i_{th} lens and j_{th} lens, respectively. $S_{i,j}$ and $r_{i,j}$ is experimental translation and rotation matrix obtained by camera calibration between i_{th} lens and j_{th} lens. R_{error} is quaternion operation.

Fig. 2 shows a displacement of a multi-view camera. T_i and T_j is extrinsic parameters matrix of i_{th} and j_{th} lenses, respectively. And, $D_{i\to j}$ is transformation matrix between each lens in a camera coordinate system. $W_{i\to j}$ is transformation matrix between each lens in a world coordinate system.

Since the displacement of each lens is preserved in a world coordinate system, we can estimate extrinsic parameters of other lenses if we know the extrinsic parameter of a reference lens.

$$\begin{bmatrix} \acute{L_j} = W_{i\to j}\acute{L_i} \\ \widetilde{T_j} = T_i W_{i\to j} \end{bmatrix}, W_{i\to j} = \begin{bmatrix} R_i^T & -R_i^T t_i \\ 0 & 1 \end{bmatrix} D_{i\to j} \tag{5}$$

where, $D_{i\to j}$ is ideal transformation including $[Rt]$ in a camera coordinate system. $[R_i t_i]$ is extrinsic parameters matrix of i_{th} lens. We define the homography between i_{th} lens and j_{th} lens as $W_{i\to j}$.

When all lenses are located on the same plane and their Z-axis directions are parallel, $D_{i\to j}$ can be represented as the simple form which has only translation

values.

$$W_{i\to j} = \begin{bmatrix} R_i^T & -R_i^T t_i \\ 0 & 1 \end{bmatrix} D_{i\to j} = \begin{bmatrix} R_i^T & -R_i^T t_i \\ 0 & 1 \end{bmatrix} \begin{bmatrix} I_{3\times 3} & -I_{3\times 3}^T t_{i\to j} \\ 0 & 1 \end{bmatrix} \tag{6}$$

After estimating extrinsic parameters of other lenses, an overall optimization which minimizes calibration error must be performed. A cost function used in here is shown in equation (7).

$$\widetilde{E} = \sum_{i=0}^{L-1} \|T_i W_i X - T_j X\|, OptimizeT_i = \begin{bmatrix} R_i & t_i \\ 0 & 1 \end{bmatrix} \tag{7}$$

where, T_j is the transformation matrix of j_{th} lens obtained by general camera calibration. $T_i W_i$ represents estimated T_j, and X is the feature point in a world coordinate system. The proposed method is especially robust when the distance between the calibration pattern and a camera is far.

3.2 Direct Computation of Motion

The motion of a multi-view camera can be represented by rigid-body transformation. We calculate 4×4 homographies, called rigid-body transformation matrix, by using corresponding points of each point clouds without any calibration pattern. Overall flow of algorithm is shown in Fig. 3. In Fig. 3, A_i and B_i are the 2D corresponding points in each image. We can get depth values of A_i and B_i because we have several lenses. Since a multi-view camera cannot guarantee a perfect disparity map, we need the process to eliminate invalid points from the corresponding point set. Then, we get full 3D corresponding points from two images. Using these matched points, we can calculate rigid-body transformation. We exploit direct technique proposed by *K.Arun* to calculate homography [7].

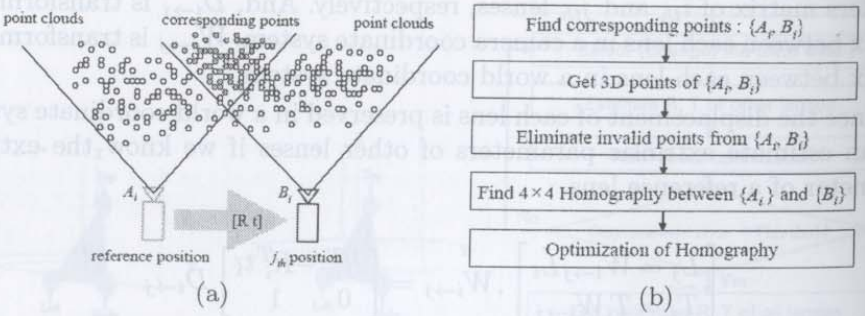


Fig. 3. Overall procedure to obtain 4×4 homography

We exploit direct technique proposed by *K.Arun* to calculate homography [7]. Given two point clouds $\{a_i, b_i\}$ in 3D, 4×4 homography including rotation

matrix and translation vector between each point cloud can be achieved through following steps. We use $Levenberg - Marquat$ non-linear optimization method to refine the results.

Step 1. Compute centers of each point cloud and vectors.

$$\overline{a}_i = \frac{1}{N}\sum_{i=0}^{N} a_i, \widetilde{a}_i = a_i - \overline{a}_i, \overline{b}_i = \frac{1}{N}\sum_{i=0}^{N} b_i, \widetilde{b}_i = b_i - \overline{b}_i \tag{8}$$

Step 2. Compute H matrix.

$$H = \sum_i \begin{bmatrix} \widetilde{a}_{i,x}\widetilde{b}_{i,x} & \widetilde{a}_{i,x}\widetilde{b}_{i,y} & \widetilde{a}_{i,x}\widetilde{b}_{i,z} \\ \widetilde{a}_{i,y}\widetilde{b}_{i,x} & \widetilde{a}_{i,y}\widetilde{b}_{i,y} & \widetilde{a}_{i,y}\widetilde{b}_{i,z} \\ \widetilde{a}_{i,z}\widetilde{b}_{i,x} & \widetilde{a}_{i,z}\widetilde{b}_{i,y} & \widetilde{a}_{i,z}\widetilde{b}_{i,z} \end{bmatrix} \tag{9}$$

Step 3. Compute the SVD(Singular Value Decomposition) of $H = USV^T$.
Step 4. Find $R = VU^T$ and Compute $Det(R) = 1$.
Step 5. Find $t = (b - R \cdot \widetilde{a})$.
Step 6. Minimize R and t by using $Levenberg - Marquat$

$$Error = \|(R \cdot a_i + t) - b_i\|^2 \tag{10}$$

We obtain the $[Rt]$ matrix through above procedure without calibration patterns. The last step is called bundle adjustment [10]. In this step, we optimize all poses of cameras using the cost function defined in equation (11).

$$\widetilde{E} = \sum_{i=0}^{L-1} \|T_i W_i X - T_j X\|, OptimizeT_i = \begin{bmatrix} R_i & t_i \\ 0 & 1 \end{bmatrix} \tag{11}$$

where, M is the total number of images taken from a multi-view camera, $a_{j,i}, b_{j,i}$ is corresponding points between i_{th} and j_{th} point clouds.

4 Experimental Results and Analysis

We use *digiclops*, $IEEE1394$ multi-view camera, to obtain images and point clouds. *Digiclops* exploits CCD sensor, $ICX084AK$, and its focal length is $6mm$ [8]. Especially, it has 3 lenses on the same plane, shown in Fig. 4(a), and its Z axis direction is parallel. Initially, we perform a Tsai's camera calibration with the non-coplanar pattern to achieve intrinsic parameters of each lens exactly. Fig. 4 shows the structure of *digiclops* and the calibration pattern. The calibration pattern gives orientation of a world coordinate system and feature points.

The displacement error function of *digiclops* defined in Chapter 3.1 is shown in (12). If we get optimized extrinsic parameters, then error is almost 0.

$$\begin{cases} T_{error} = \|S_{r,t} - G_{r,t}\| + \|S_{t,l} - G_{t,l}\| + \|S_{r,l} - G_{r,l}\| \\ R_{error} = \|R_t - R_r\| + \|R_r - R_l\| + \|R_l - R_t\| \end{cases} \tag{12}$$

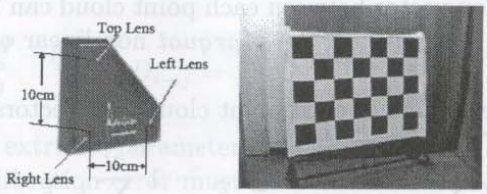


Fig. 4. Experimental setup (a) *digiclops* (b) calibration pattern

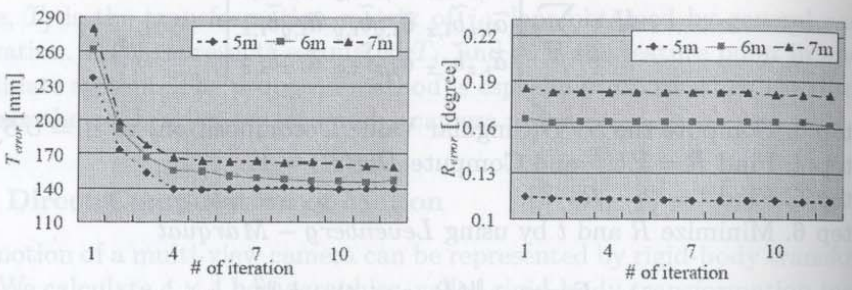


Fig. 5. Results of optimizing pose with increasing distance (a) T_{error} (b) R_{error}

where, R_{error} is quaternion operation. $G_{r,l}$ is ideal values related to the distance between $left$ lens and $right$ lens shown in Fig. 4. $S_{r,l}$ is experimental values related to the distance between $left$ lens and $right$ lens calculated by Tsai's camera calibration. Experimental results are shown in Fig. 5. Proposed method reduces calibration error effectively.

In order to measure an accuracy of camera motion, we perform experiments using synthetic data implemented in *OpenGL* space. Equation (13) is errors we measure.

$$Error = \sum_{i=0}^{M} \|(R \cdot a_i + t) - b_i\|^2, NormalizedError(NE) = \frac{Error}{M} \tag{13}$$

where, $Error$ is the sum of all pixel errors which are differences between destination points $\{b_i\}$ and transformed source points $\{R \cdot a_i + t\}$. $NormalizedError(NE)$ is the mean error of each pixel. The simulation results are shown in Fig. 6. Fig. 6 (a) shows initial two point clouds. Fig. 6 (b) shows the results after applying $[Rt]$ to source points. $Error$ is under $0.001mm$. Fig. 6 (c) shows $Error$ and NE when the number of points is increasing.

We also apply our algorithm to real environment. We use well-known *RANSAC* (Random Sample Consensus) method to find out 2D corresponding points from two images [9].

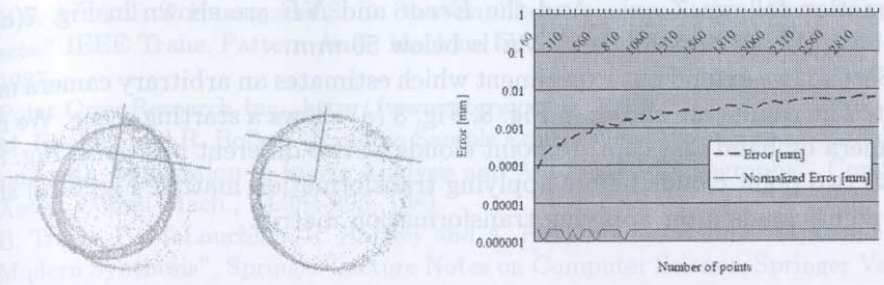


Fig. 6. Accuracy of rigid-body transformation (a) initial two point clouds (b) after fitting (c) errors as number of points is increasing

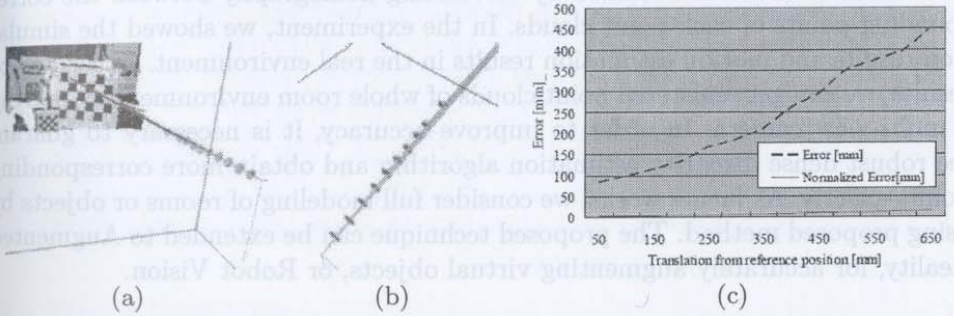


Fig. 7. Camera moving along Z axis (a) scene with cameras and point cloud (b) top view of motion (c) errors with translation along Z-axis from reference position

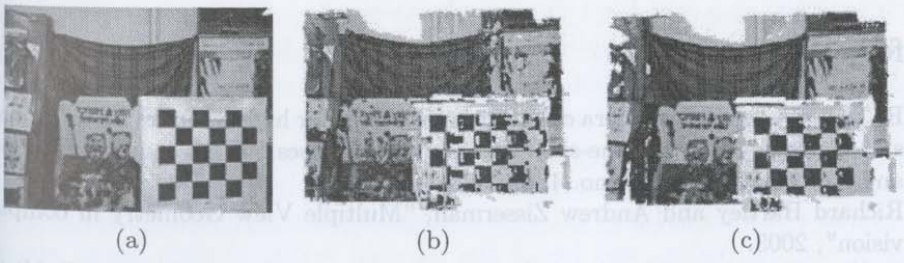


Fig. 8. Real scene experiment (a) original 2D image (b) before fitting two point clouds (c) after fitting two point clouds by obtained transformation

First, we move a multi-view camera along Z-axis (back-ward movement) by using accurate cart that enables us to move the camera along Z-axis exactly. We check a camera position per 50 mm movement. Fig. 7 (a), (b) show the camera motion in 3D space. Virtual cameras are represented as a circle. As expected,

the motion follows Z axis. And the $Error$ and NE are shown in Fig. 7(c). In this case, M is 120 points and NE is below 50 mm.

Second, we extend our experiment which estimates an arbitrary camera movement. The results are shown in Fig. 8. Fig. 8 (a) shows a starting scene. We move a camera by hand and capture point clouds at two different positions. Fig. 8 (b) shows two point clouds before applying transformation matrix. Fig. 8 (c) shows two point clouds after applying transformation matrix.

5 Conclusions and Future Work

We proposed a method to optimize camera poses with coplanar pattern and estimate motion for modeling indoor environment. The proposed method provides optimal starting position of a multi-view camera by using structural constraints. And camera motion is obtained by calculating homography between the corresponding points of each point clouds. In the experiment, we showed the simulation results and motion estimation results in the real environment. According to results, we can get registered point clouds of whole room environment by moving a multi-view camera. In order to improve accuracy, it is necessary to guarantee robust dense disparity estimation algorithm and obtain more corresponding points exactly. As future works, we consider full modeling of rooms or objects by using proposed method. The proposed technique can be extended to Augmented Reality, for accurately augmenting virtual objects, or Robot Vision.

Acknowledgements. This work was supported in part by the Ministry of Information and Communication (MIC) through the Realistic Broadcasting Research Center at GIST, and in part by MIC through Next Generation PC Project at GIST.

References

1. R. Tsai, "A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses," IEEE Journal of Robotics and Auto-mation, vol. 3, no. 4, pp. 323-344, 1987.
2. Richard Hartley and Andrew Zisserman, "Multiple View Geometry in computer vision", 2003.
3. S. Kim, E. Chang, C. Ahn and W. Woo, "Image-based Panoramic 3D Virtual Environment using Rotating Two Multi-view Cameras," IEEE Proc. ICIP2003, vol. 1, pp. 917-920, 2003.
4. Z. Zhang, "Flexible camera calibration by viewing a plane from unknown orientations," ICCV99, vol. 1, pp. 666-673, 1999.
5. Peter Sturm, Steve Maybank, "On Plane-Based Camera Calibration: A General Algorithm, Singularities, Applications", Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Fort Collins, USA, pp. 432-437 June, 1999. 6.
6. P. Sturm, "Algorithms for Plan-Based Pose Estimation", Proc. Computer Vision and Pattern Recognition, vol.1, pp.706-711, 2002.

7. K.S. Arun, T. S. Huang, and S. D. Blostein, "Least-squares fitting of two 3-D point sets," IEEE Trans. Pattern Anal. Machine Intell., vol. PAMI-9, no. 5, pp.698-700, 1987.
8. Point Grey Research Inc., http://www.ptgrey.com, 2002.
9. M. Fischler and R. Bolles, "Random Sample Consensus: a Paradigm for Model Fitting with Application to Image Analysis and Automated Cartography", Commun. Assoc. Comp. Mach., 24:381-395, 1981.
10. B. Triggs, P. McLauchlan, R. Hartley and A. Fitzgibbon, "Bundle Adjustment - a Modern Synthesis", Springer Lecture Notes on Computer Science, Springer Verlag, 1883:298-375, 2000.