3D Camera Tracking from Disparity Images

Kiyoung Kim* and Woontack Woo*

GIST U-VR Lab.
Gwangju 500-712, S.Korea

ABSTRACT

In this paper, we propose a robust camera tracking method that uses disparity images computed from known parameters of 3D camera and multiple epipolar constraints. We assume that baselines between lenses in 3D camera and intrinsic parameters are known. The proposed method reduces camera motion uncertainty encountered during camera tracking. Specifically, we first obtain corresponding feature points between initial lenses using normalized correlation method. In conjunction with matching features, we get disparity images. When the camera moves, the corresponding feature points, obtained from each lens of 3D camera, are robustly tracked via *Kanade-Lukas-Tomasi* (KLT) tracking algorithm. Secondly, relative pose parameters of each lens are calculated via Essential matrices. Essential matrices are computed from Fundamental matrix calculated using normalized 8-point algorithm with *RANSAC* scheme. Then, we determine scale factor of translation matrix by d-motion. This is required because the camera motion obtained from Essential matrix is up to scale. Finally, we optimize camera motion using multiple epipolar constraints between lenses and d-motion constraints computed from disparity images. The proposed method can be widely adopted in Augmented Reality (AR) applications, 3D reconstruction using 3D camera, and fine surveillance systems which not only need depth information, but also camera motion parameters in real-time.

Keywords: 3D camera, Tracking, Pose estimation, Disparity, Essential matrix

1. INTRODUCTION

These days, 3D cameras are gaining popularity in computer vision application area. 3D cameras are not replacing general cameras in Augmented Reality (AR), Human Computer Interaction (HCI), and surveillance systems. One obvious reason is that 3D camera provides depth and texture information of images in real time[1]. To satisfy the requirements of applications, we not only need camera calibration algorithm, but also robust pose estimation.

There have been many researches on 3D camera pose estimation in the traditional computer vision area. These can be classified roughly into two groups according to the constraints used. First, *Z.Zhang, et al.* optimized camera parameters using multiple epipolar constraints defined when a stereo camera moves around[4]. Second class is a method which uses depth information directly provided by 3D camera with the assumption that camera motion is the same as getting transformation between corresponding 3D point sets[2,3]. If the accuracy of disparity map obtained from 3D camera is guaranteed, camera position can be estimated from 3D-3D transformation between the corresponding feature points. *N.Ohta, et al.* derived mathematical properties and performed reliability experiments[5] for 3D-3D transformation, and *D.Demirdjian, et al*. proposed a method using d-motion[6] which is based on depth information and optical flow.

In a method that uses depth information obtained from 3D camera, however, overall pose estimation error largely depends on instability of the depth information and feature points tracking error[7]. This causes serious problems in real-time 3D camera motion tracking. On the other hand, a method which only uses geometric information and tracked feature points can only recover camera motion up to scale since we have no absolute measurements. Additionally, in a specific camera motion, we can not recover the motion due to triangulation uncertainty[8].

In this paper, we propose a robust camera tracking method that uses disparity images computed from known parameters of 3D camera and multiple epipolar constraints. We assume that baselines between lenses in 3D camera and intrinsic parameters are known. The proposed method reduces camera motion uncertainty encountered during camera tracking. Specifically, we first obtain corresponding feature points between initial lenses using normalized correlation method. In conjunction with matching features, we get disparity images. When the camera moves, the corresponding feature points, obtained from each lens of 3D camera, are robustly tracked via *Kanade-Lukas-Tomasi* (KLT) tracking algorithm[9].

* E-mail : {kkim, wwoo}@gist.ac.kr, Phone : +82-62-970-3157

Visual Communications and Image Processing 2005, edited by Shipeng Li, Fernando Pereira, Heung-Yeung Shum, Andrew G. Tescher, Proc. of SPIE Vol. 5960 (SPIE, Bellingham, WA, 2005) · 0277-786X/05/$15 · doi: 10.1117/12.632741

Secondly, relative pose parameters of each lens are calculated via Essential matrices. Essential matrices are computed from Fundamental matrix calculated using normalized 8-point algorithm with *RANSAC* scheme. Then, we determine scale factor of translation matrix by d-motion. This is required because the camera motion obtained from Essential matrix is up to scale. Finally, we optimize camera motion using multiple epipolar constraints between lenses and d-motion constraints computed from disparity images.

The proposed method recovers camera motion with scale factor of translation matrix by using known 3D camera parameters. Additionally, the method is robust to noise. This is achieved by using epipolar geometry and robust known parameters of d-motion. And it can avoid critical motion problems. The proposed method can be widely adopted for Augmented Reality (AR) applications, 3D reconstruction using 3D camera, and fine surveillance systems which not only require depth information, but also camera motion parameters in real-time.

This paper is organized as follows. In chapter 2, we review previous researches on camera motion estimation using Essential matrix or depth information. In chapter 3, we briefly explain the proposed method based on previous researches. Experimental results and analysis of the proposed method are shown in Chapter 4. Finally, in Chapter 5, conclusion and future works are described.

2. CAEMRA POSE ESTIMATION

2.1 Relative pose estimation of calibrated camera up to scale

Geometric relationship between two images obtained from calibrated cameras is represented as 3×3 Essential matrix (E). E consists of camera motion parameters, rotation and translation matrix. Thus, if we know E, we can recover camera motions until up to scale. The definition of E is as follows[10].

$$E = SR = \begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix} R \tag{1}$$

where, S, skew-symmetric matrix, consists of translation matrix = $[t_x, t_y, t_z]^T$, and R is 3×3 rotation matrix. The rank of E is 2 since the rank of S matrix is 2. Thus, determinant of E should be 0.

If corresponding points $\{x_l \leftrightarrow x_r\}$ of two images are given, E can be calculated in two ways. First, we can reconstruct E by calculating Fundamental matrix (F) with given corresponding points. F of two images from different two cameras can be calculated robustly by applying *RANSAC* or *LMedS* methods based on 7-point or normalized 8-point algorithm[10].

$$F = K_r^{-T} E K_l^{-1} \tag{2}$$

where, F and E represents Fundamental matrix and Essential matrix, respectively, $\{K_l \leftrightarrow K_r\}$ represent intrinsic parameter matrix of corresponding cameras.We assume that intrinsic parameter matrix $\{K_l \leftrightarrow K_r\}$ of camera is given like equation (3).

$$K = \begin{bmatrix} \alpha & s & u_o \\ 0 & \beta & v_o \\ 0 & 0 & 1 \end{bmatrix} \tag{3}$$

where, α and β are focal lengths, s is skew parameter, and $(u_0, v_0)^T$ is the principal point. Second method is using epipolar constraint $x_r^T F x_l = 0$ of a point pair $\{x_l \leftrightarrow x_r\}$ which is corresponding left and right image. We can represent a relationship between E and corresponding points as a linear equation using the relationship described in equation (4).

$$\hat{x}_r^T E \hat{x}_l = x_r^T K_r^{-T} E K_l^{-1} x_l = 0 \tag{4}$$

In equation (4), in order to get E we set the relationship between e (=components of E) and 2D points to $Ae = 0$. From this equation, the last column of V^T becomes components of E after applying *Singular Value Decomposition* (SVD) of $A = UDV^T$.

After calculating E, we decompose E into rotation and translation matrix. As a result, we can get totally four different rotation and translation matrix after separating E with *SVD* method. Among them, we have to select correct rotation and translation matrix by applying geometrical relationship between a camera and triangulated 3D points.

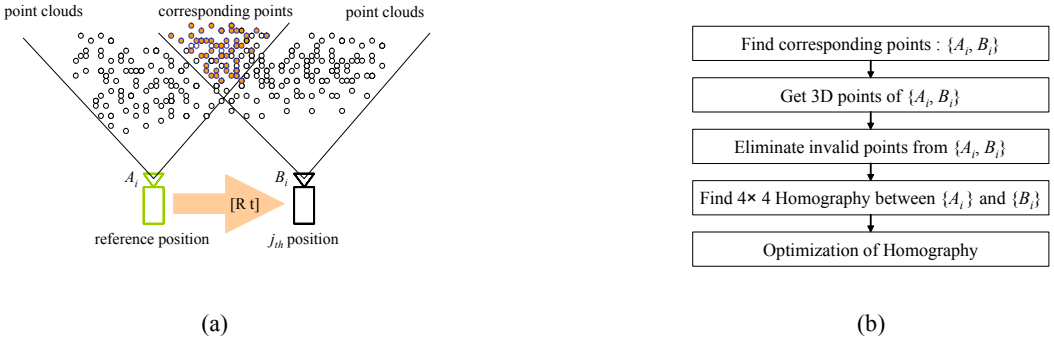
2.2 Direct pose estimation of 3D camera using depth information

The pose of 3D camera can be solved using calibrated camera theory explained in Sec. 2.1 and depth information. If the accuracy of depth information obtained from 3D camera is guaranteed, recovering camera motion is regarded as solving Absolute orientation problem. It is resulted in finding rotation matrix and translation matrix which optimize equation (5) with respect to corresponding feature points in two 3D point sets.

$$\sum_{i=0}^{n} \left\| B_i - (R \cdot A_i + t) \right\|^2 \tag{5}$$

where, $\{A \leftrightarrow B\}$ represents corresponding 3D point sets, R and t mean rotation matrix and translation matrix between two 3D point sets, respectively.

Figure 1 (a) shows a relationship between 3D camera motion and 3D point sets. Rotation matrix and translation matrix can be obtained using corresponding 3D points between 3D point set A on a reference position and 3D point set B on a *jth* position. In detail, the example of computing motion parameters is described in **Figure 1** (b).



(a) (b)

Figure 1. Direct pose computation from 3D corresponding point sets

It briefly consists of 4 steps. *K.Arun's* or *B.Horn's* method are widely adopted to get transformation between two 3D point sets. Following algorithm is a simple combination method of *Levenberg-Marquat* non-linear optimization and *K.Arun's* method.

Step 1. Compute centers of each point cloud and vectors

$$\overline{A} = 1/N \sum_{i=0}^{N} A_i, \quad \widetilde{A}_i = A_i - \overline{A} \quad , \quad \overline{B} = 1/N \sum_{i=0}^{N} B_i, \quad \widetilde{B}_i = B_i - \overline{B} \tag{6}$$

Step 2. Compute H matrix

$$H = \sum_i \begin{bmatrix} \widetilde{A}_{i,x}\widetilde{B}_{i,x} & \widetilde{A}_{i,x}\widetilde{B}_{i,y} & \widetilde{A}_{i,x}\widetilde{B}_{i,z} \\ \widetilde{A}_{i,y}\widetilde{B}_{i,x} & \widetilde{A}_{i,y}\widetilde{B}_{i,y} & \widetilde{A}_{i,y}\widetilde{B}_{i,z} \\ \widetilde{A}_{i,z}\widetilde{B}_{i,x} & \widetilde{A}_{i,z}\widetilde{B}_{i,y} & \widetilde{A}_{i,z}\widetilde{B}_{i,z} \end{bmatrix} \tag{7}$$

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 $t = \overline{B} - R \cdot \overline{A}$

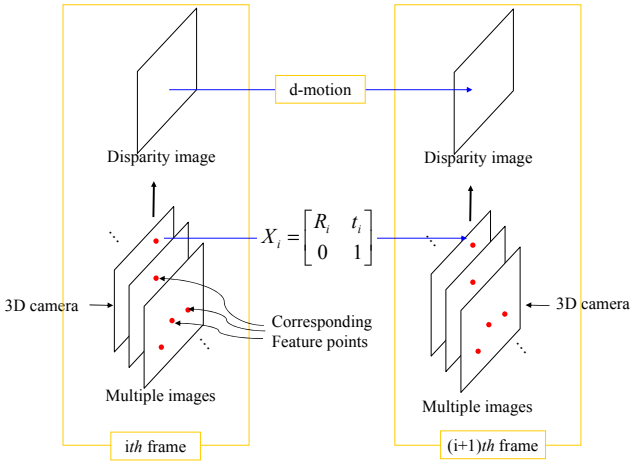
Step 6. Minimize R and t by using Levenberg-Marquat non-linear optimization

$$Error = \left\| (R \cdot A_i + t) - B_i \right\|^2 \tag{8}$$

3. MULTIPLE CAMERA GEOMETRY AND OPTIMIZATION CONSTRAINT

3.1 Overall algorithm procedure of combining epipolar and d-motion constraints

The proposed algorithm consists of two parts. One is to compute rotation and translation matrix up to scale from Essential matrix. The other is to compute d-motion from disparity images. Overall procedure to get camera pose is explained in **Figure 2**. **Figure 2** (a) shows the concept of 3D camera motion. The first part of the proposed algorithm is related to multiple images from 3D camera. And the second part is related to one disparity image obtained by using multiple images and known intrinsic parameters. **Figure 2** (b) shows the detail step to achieve the pose.



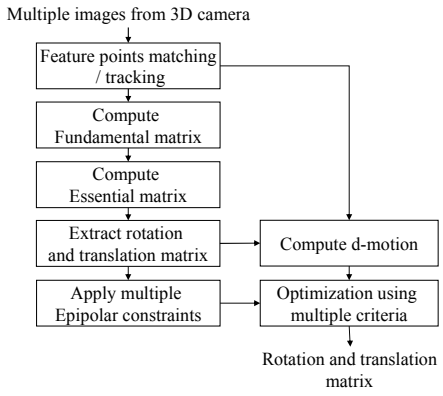
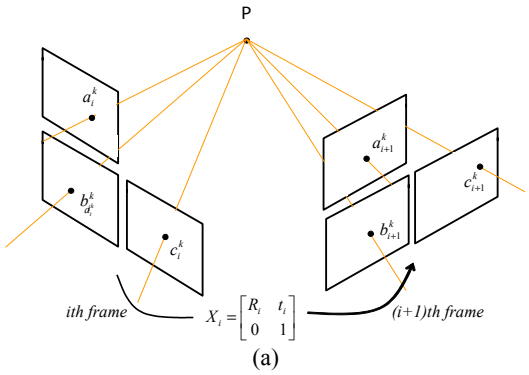


Figure 2. Overall procedure of proposed algorithm (a) concept of a 3D camera motion (b) detail procedure of algorithm

3.2 Multiple epipolar constraints

3D camera includes multiple lenses in a body. When the camera moves around, each lens has the same rigid body transformation. If there exist N lenses, lenses generate N^2 epipolar geometry. **Figure 3** (a) represents geometric information when the number of lenses in 3D camera is 3.



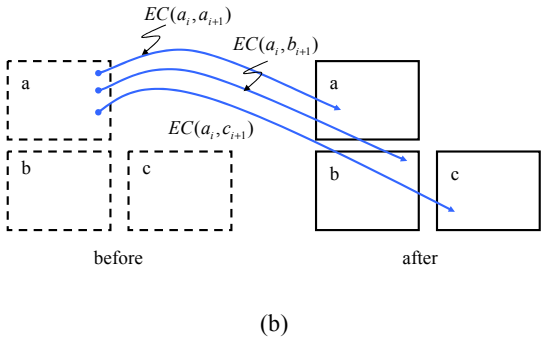


Figure 3. A 3D camera geometry which lenses number is 3 (a) motions of inner lenses (b) representation of multiple epipolar geometry constraints

Figure 3 (b) represents multiple epipolar constraints in case of 3 lenses. We extend the concept of 3 epipolar constraints described in Figure 3 to general case. If we have N lenses in 3D camera, we can represent multiple epipolar constraints as following equation.

$$f_{\cdot\cdot\cdot} = \sum^{N} \sum^{N} EC(i_{\cdot}, k_{\cdot\cdot}) \tag{9}$$

$$EC(j_i, k_{i+1}) = \sum_{q=0}^{M} (\hat{x}_{k_{i+1}}^{q})^T E \hat{x}_{j_i}^{q} \tag{10}$$

where, $EC(j_i, k_{i+1})$ represents the epipolar constraint between two images. One image is the *jth* lens in *ith* frame, the other image is the *kth* lens in the *(i+1)th* frame. *N* and *M* is the number of lenses in a single 3D camera and the number of corresponding feature points, respectively. *E* is an Essential matrix. Ideally, if there are no noise in the images, then $f_{i,i+1}$ should be 0. That is, all corresponding points satisfy epipolar constraint, perfectly.

In general, it is required to parameterize Fundamental matrix before we apply optimization algorithm. Fundamental matrix can be parameterized by using two epipoles. If we assume that the coordinates of two epipoles are $(\alpha, \beta, -1)^T$ and $(\alpha', \beta', -1)^T$, respectively, *F* can be represented as following equation (11)[10].

$$F = \begin{bmatrix} a & b & \alpha a + \beta b \\ c & d & \alpha c + \beta d \\ \alpha' a + \beta' c & \alpha' b + \beta' d & \alpha' \alpha a + \alpha' \beta b + \beta' \alpha c + \beta' \beta d \end{bmatrix} \tag{11}$$

3.3 d-motion of 3D camera constraint

3D camera motion is related to baselines, disparity images, and intrinsic parameters. We can combine these parameters. d-motion is the representation of combined these parameters[6]. When we have corresponding feature points between arbitrary *ith* frame and *(i+1)th* frame, we can define the disparity of one 3D point as d_i or d_{i+1}. Then, its d-motion is described as equation (12).

$$\omega_{i+1} = H_d \omega_i = \begin{bmatrix} x \\ y \\ d \\ 1 \end{bmatrix}_{i+1} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 0 & fB \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 0 & fB \\ 0 & 0 & 1 & 0 \end{bmatrix}^{-1} \begin{bmatrix} x \\ y \\ d \\ 1 \end{bmatrix}_i \tag{12}$$

where, *f* represents focal length of a camera, and *B* represents baseline of a camera. And the origin of point (x, y) in the image plane is the principal point. Thus, we can use equation (12) as one constraint which includes disparity and feature points at the same time. As a result, to get rotation and translation matrix is to minimize equation (13).

$$\sum_{j=0}^{N} \left\| H_d \omega_i^j - \omega_{i+1}^j \right\|^2 \tag{13}$$

where, *N* is the number of corresponding points between reference *ith* frame and *(i+1)th* frame. If we have at least six corresponding points (3 for translation and 3 for rotation) and initial *R* and *t*, then we can utilize *Levenberg-Marquat* non-linear optimization.

4. EXPERIMENTS

We used *Digiclops*[11], IEEE 1394 3D camera, to obtain images and disparity images. *Digiclops* exploits CCD sensor, ICX084AK, and its focal length is 6 mm. And it provides 640×480 image resolution. Especially, it has 3 lenses on the same plane, shown in **Figure 4**, and its Z axis direction is parallel. Initially, we performed *Tsai*'s camera calibration with the non-coplanar pattern to achieve intrinsic parameters of each lens exactly[12]. And the baselines between 3 lenses are given as shown in **Figure 4**. We implemented the proposed method by using OpenCV[14] beta 4.0 library and OpenGL.

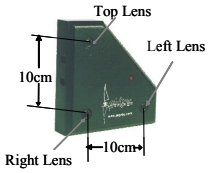


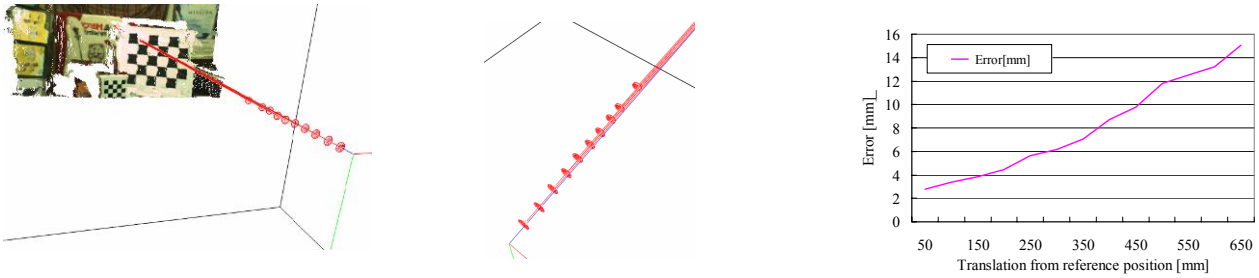
Figure 4. Structure of 3D camera : *Digiclops* has 3 lenses and its baselines are almost equal

4.1 Direct pose computation from 3D to 3D point sets

We tested a method that computes camera pose directly using 3D-3D point sets obtained from 3D camera. To show experimental results intuitively, we moved a 3D camera by using special cart which enables us to move the camera along Z-axis exactly. We checked a camera position per 50 mm movement. Figure 5 (a), (b) show the camera positions in 3D virtual space. Virtual cameras are represented as a circle. We measured the rotation and translation error which definition is as shown in equation (14). Figure 5 (c) shows the rotation and translation errors in millimeters.

$$Error = \frac{1}{M}\sum_{i=0}^{M} \left\| (R \cdot A_i + t) - B_i \right\|^2 \tag{14}$$

where, M represents the number of corresponding feature points. And $\{A_i, B_i\}$ are the 3D point sets of corresponding feature points.



(a) (b) (c)

Figure 5. Direct pose computation using 3D-3D point sets (a) scene with cameras and point cloud (b) top view of motion (c) errors with translation along Z-axis from reference position

Feature points used in this case were 120 points. As shown in the results, error of one pixel is below 16 mm when the distance is 650 mm. However, we often met the serious problems when we tried to rotate and translate the camera together. That is, the experiments of general motions showed unstable results.

4.2 Epipolar constaints and d-motion

We applied proposed method to compute the absolute pose in general positions. First, we obtained original images. In case of *Digiclops*, we got 3 images at once. And also we got the disparity image calculated from those 3 images. And we moved the camera to another position. Original image pair we used in this experiment is shown in Figure 6. We added some rotation and translation values together to the camera movement. We captured 3 images at *ith* frame and at *(i+1)th* frame from top, right, and left lens, respectively.

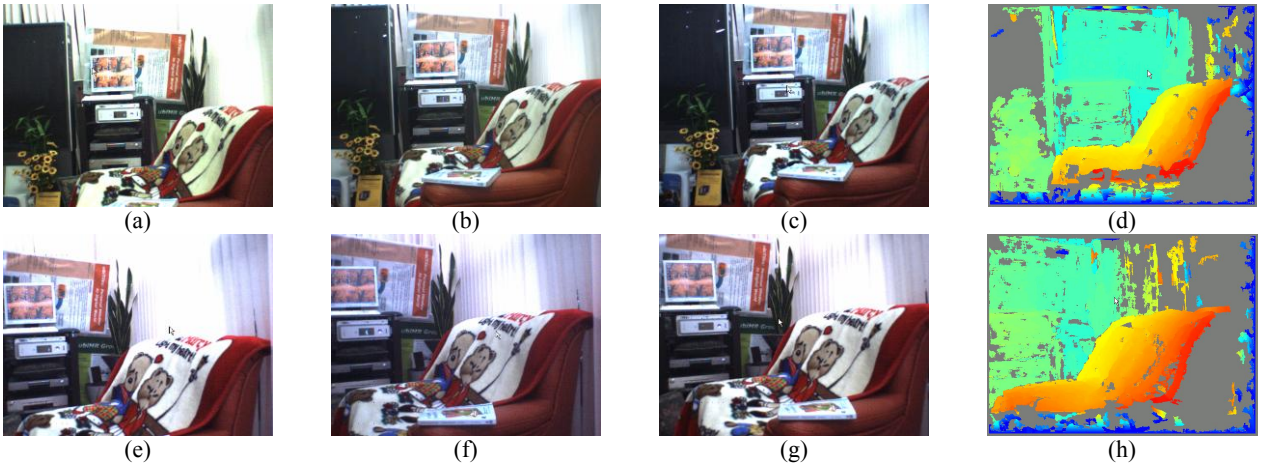


Figure 6. Original images taken from 3D camera, ith frame : (a) Top (b) Right (c) Left (d) Disparity, (i+1)th frame : (e) Top (f) Right (g) Left (h) Disparity

We computed fundamental matrices and extracted the initial rotation and translation parameters from Essential matrices. The Fundamental matrices computed by normalized 8-point algorithm with non-linear optimization scheme are visualized in epipolar lines. We used Harris' corner detector[13] with normalized correlation algorithm[15] to match the feature points in 3 images. One results of correlation matching and Fundamental matrix are shown in the Figure 7. In Figure 7 (a), 182 corresponding feature points are found and tracked. However, still outliers exist.

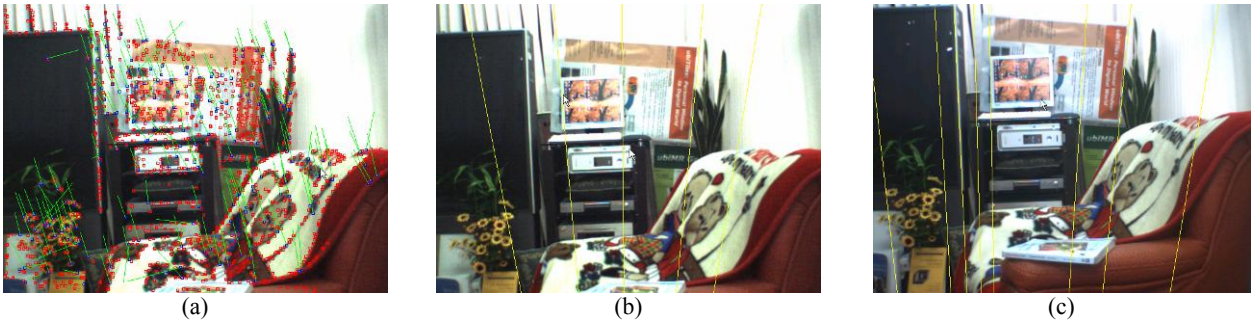


Figure 7. Results of correlation matching and Fundamental matrix, (a) feature matching: correlation window size = 13×13 and size of search window = (image size / 6)×(image size / 6) (b) top image and epipolar lines (c) right images and epipolar lines: residual error of fundamental matrix is under 0.5 pixels

To show the effectiveness of proposed method, we registered two point clouds by using obtained motion parameters. The point clouds were computed by combining original images and disparity images.

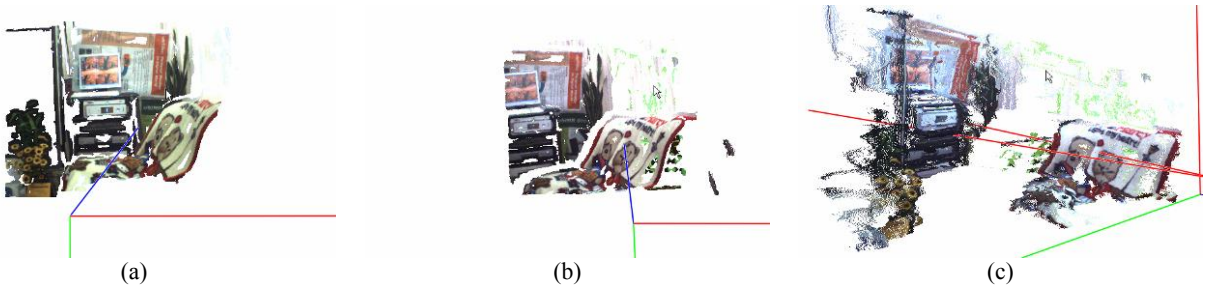


Figure 8. Register point clouds using translation and rotation parameters (a) ith frame (b) (i+1)th frame (c) registered point clouds

5. CONCLUSIONS AND FUTURE WORKS

In this paper, we proposed the method which optimizes the pose of 3D camera using multiple epipolar constraints and disparity images with known camera parameters. As shown in experimental results, we first showed the pose estimation of utilizing transformations between 3D-3D point sets. We rendered virtual camera positions in OpenGL space as well. However, we found that 3D-3D point sets algorithm often caused serious problems when we moved a camera in general. To improve this problem, we applied multiple epipolar constraints which are defined between all lenses in 3D camera. As a result, we overcame the problems of 3D-3D point sets algorithm and two problems of one camera case. We could recover the scale factor of translation matrix which is not recovered when we just use single camera. And we could avoid ambiguous motions which make triangulations problems. As a future work, we will analyze the error source and study about reliability of the algorithm theoretically. The proposed method can be applied to augmented reality applications which need not only depth information, but also accurate camera motions. And also it can be useful in 3D reconstruction or surveillance systems.

ACKNOWLEDGMENTS

This research is supported by Immersive Contents Research Center (ICRC) in Gwangju Institute of Science and Technology.

REFERENCES

1. W. Woo, N. Kim, K. Wong, and M. Tadenuma, "Sketch on Dynamic Gesture Tracking and Analysis Exploiting Vision-based 3D Interface," Proc. SPIE PW-EI-VCIP'01, 4310, pp. 656-666, 2001.
2. B. Horn, “Closed-form solution of absolute orientation using unit quaternions,” J. Opt. Soc. Amer. A, vol. 4, no. 4, pp. 629–642, 1987.
3. K. Arun, T. Huang, and S. 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.
4. Z. Zhang, Q.-T. Luong, and O. Faugeras, “Motion of an uncalibrated stereo rig: Self-calibration and metric reconstruction,” IEEE Trans. Robot. Automat., vol. 12, pp. 103–113, 1996.
5. N. Ohta and K. Kanatani, “Optimal estimation of three-dimensional rotation and reliability evaluation,” Proceedings of the 5th European Conference on Computer Vision (ECCV'98), June 2-6, 1998, University of Freiburg, Freiburg, Germany, Vol. 1, Hans Burkhardt and Bernd Neumann (Eds.), Computer Vision--ECCV'98 Volume 1, LNCS, No. 1406, Springer-Verlag, Berlin, 1998, pp. 175-187.
6. D. Demirdjian and T. Darrell, “Using Multiple-Hypothesis Disparity Maps and Image Velocity for 3-D Motion Estimation,” International Journal of Computer Vision 47(1-3): pp.219-222, 2002.
7. K. Kim and W. Woo, "A multi-view camera tracking for modeling of indoor environment," *LNCS (PCM)*, 3331, pp. 288-297, 2004.
8. P. Sturm, “Critical Motion Sequences for the Self-Calibration of Cameras and Stereo Systems with Variable Focal Length,” BMVC, Nottingham, England, pp. 63-72, 1999.
9. C. Tomasi and T. Kanade, “Detection and Tracking of Point Features,” Carnegie Mellon Univ., TR CMU-CS-91-132, 1991.
10. R. Hartley and A. Zisserman, “Multiple View Geometry in computer vision”, 2003.
11. Point Grey Research Inc., http://www.ptgrey.com, 2002.
12. 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 Automation, vol. 3, no. 4, pp. 323-344, 1987.
13. C. Harris and M. Stephens, "A combined corner and edge detector,” In Fourth Alvey Vision Conference, pp. 147-151, 1988.
14. Intel OpenCV Library, http://www.intel.com/research/mrl/research/openCV
15. Z. Zhang, R. Deriche, O. Faugeras and Q.-T. Luong, “A Robust Technique for Matching Two Uncalibrated Images Through the Recovery of the Unknown Epipolar Geometry,” Artificial Intelligence Journal, Vol.78, pages 87-119, October 1995.