Synchronized Ego-Motion Recovery of Two Face-to-Face Cameras

Jinshi Cui[1], Yasushi Yagi[2], Hongbin Zha[1], Yasuhiro Mukaigawa[2], and Kazuaki Kondo[2]

[1] State Key Lab on Machine Perception, Peking University, China
{cjs,zha}@cis.pku.edu.cn
[2] Department of Intelligent Media, Osaka University, Japan
{yagi,mukaigawa,kondo}@am.sanken.osaka-u.ac.jp

Abstract. A movie captured by a wearable camera affixed to an actor's body gives audiences the sense of "immerse in the movie". The raw movie captured by wearable camera needs stabilization with jitters due to ego-motion. However, conventional approaches often fail in accurate ego-motion estimation when there are moving objects in the image and no sufficient feature pairs provided by background region. To address this problem, we proposed a new approach that utilizes an additional synchronized video captured by the camera attached on the foreground object (another actor). Formally we configure above sensor system as two face-to-face moving cameras. Then we derived the relations between four views including two consecutive views from each camera. The proposed solution has two steps. Firstly we calibrate the extrinsic relationship of two cameras with an AX=XB formulation, and secondly estimate the motion using calibration matrix. Experiments verify that this approach can recover from failures of conventional approach and provide acceptable stabilization results for real data.

Keywords: Wearable camera, synchronized ego-motion estimation, stabilization, two face-to-face cameras, extrinsic calibration.

1 Introduction

The goal of this work is to recover ego-motion of two face-to-face moving cameras simultaneously. This work aims at some situations where ego-motion with only one camera may fail and use another camera to provide additional information.

Ego-motion estimation of a moving camera is the task of recovering camera motion trajectory given a set of 2D image frames. It has many applications like stabilization in our application. Most existing methods take one of the following two cases. For the case of static scenes, the problem of fitting a 3D scene compatible with the images is well understood and essentially solved [1, 2]. The second case deals with dynamic scenes, where the segmentation into independently moving objects and the motion estimation for each object have to be solved simultaneously [3-4]. Above methods may fail in camera ego-motion estimation if: (1) the foreground occupies too much space in the image, (2) there are insufficient features in the background

Y. Yagi et al. (Eds.): ACCV 2007, Part I, LNCS 4843, pp. 544–554, 2007.

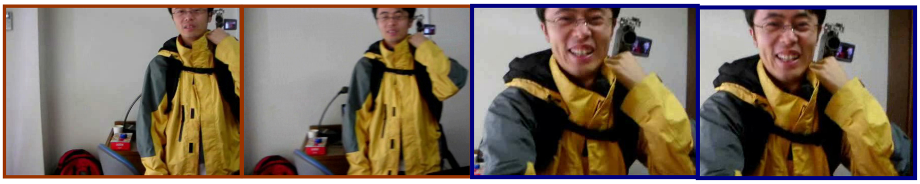


Fig. 1. Two image pairs captured by one wearable camera with a moving foreground. Left image pair: Camera motion can be computed using background region with enough feature point matches.Right image pair: There are very few feature matches in background region. It's impossible to estimate ego-motion without additional information. And motion of the foreground point matches is related with both camera motion and person's motion. If we know foreground person's motion, camera ego-motion can be estimated.

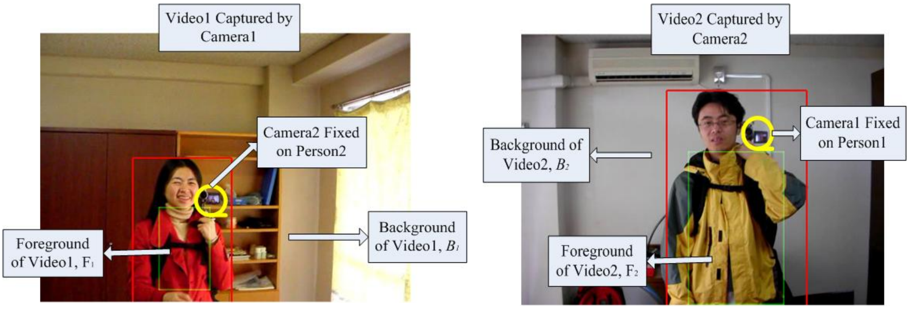

Video1 Captured by Camera1
Camera2 Fixed on Person2
Background of Video1, B_1
Foreground of Video1, F_1
Video2 Captured by Camera2
Camera1 Fixed on Person1
Background of Video2, B_2
Foreground of Video2, F_2

Fig. 2. Two face-to-face cameras in our application of "Dive into Movie". One camera is attached to the body of each person.

region of image pair, or (3) there is too much repeated structure for features to get a good match.

Fig. 1 showed the situation that almost the whole image is covered by moving foreground. It's impossible to estimate ego-motion in this case. Additional information can be utilized, such as inertial data [6] or synchronized image frames from another camera. In the case of using another camera, there can be two cases.

One is that the additional camera is fixed somewhere watching person1 or person2. In the case of watching person1, the motion of camera is directly estimated by pose estimation. In the case of watching person2, first the motion of person 2 is estimated, and then camera motion is estimated by eliminating the person's motion from the foreground motion of the camera. In both cases, it's necessary to make the fixed camera always watching the moving person. The other one is that the additional camera is just the one attached on the foreground object (i.e. another person's body). This configuration is very natural in our application (see Fig. 2).

Motivation for the above work is from a new application of computer vision technology in entertainment, so called "Dive into Movie". In this application, a movie captured by a wearable camera attached to the actor's body can give audiences the sense of "immerse in the movie". The raw movie captured by wearable camera needs to be stabilized due to jitters and ego-motion of the actor. And accurate ego-motion estimation of a moving camera is not easy when there are moving objects in the

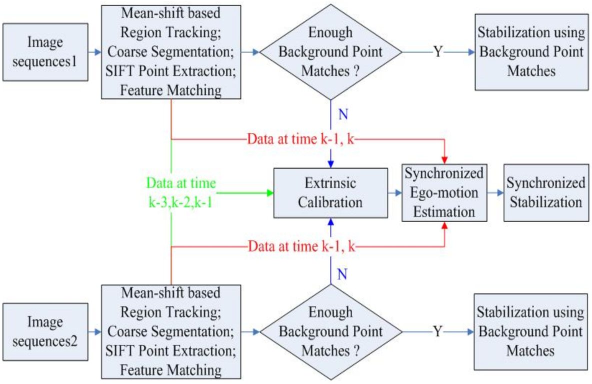


Image sequences1
Mean-shift based Region Tracking; Coarse Segmentation; SIFT Point Extraction; Feature Matching
Enough Background Point Matches ?
Y
Stabilization using Background Point Matches
N
Data at time k-1, k
Data at time k-3,k-2,k-1
Extrinsic Calibration
Synchronized Ego-motion Estimation
Synchronized Stabilization
Data at time k-1, k
N
Image sequences2
Mean-shift based Region Tracking; Coarse Segmentation; SIFT Point Extraction; Feature Matching
Enough Background Point Matches ?
Y
Stabilization using Background Point Matches

Fig. 3. Overview of the proposed approach at time *k*

image. In this application, there are at least two face-to-face interacting actors in a scene. The audience can choose anyone of the actors to see the movie from different views. One camera is attached to each actor. For simplicity, in this paper, we only consider the case of two actors in the scene. Then our goal is to recover ego-motion of two face-to-face cameras using information from both cameras.

To address this problem, we first configure the sensor system as two face-to-face moving cameras. And then we derived the relationship between four views that consist of two consecutive views of each camera. In estimation stage, two cameras are calibrated first, and then ego-motion is estimated using calibration result. The calibration problem is formed as AX=XB and refer to the solutions in traditional robotics hand-eye calibration [6-9]. Compared with the consistent motion of hand/eye in traditional hand-eye calibration, we deal with two independently moving cameras. To our knowledge, there is no other work reported on this problem. In [10], a similar configuration of stereo camera is proposed, which used two face-to-face static cameras. The epipolar geometry for these mutual cameras is studied and used to improve the performance of structure from motion approach. In contrast to [10], our approach tries to estimate the ego-motion of two moving face-to-face cameras.

The flowchart of the proposed system is showed in Fig. 3. Firstly input videos are pre-processed to segment out the background region and object region which moves consistently with the opposing camera. SIFT features are extracted and matched within consecutive two image pairs for background region and object respectively. If there are enough reliable point matches in background region, ego-motion is estimated and output stabilized frame. Above steps are processed for both cameras. Secondly, if estimation fails with background region, go to the synchronized estimation step, which includes two stages. Extrinsic parameters are calibrated in first stage for two cameras. Here, it’s necessary to get at least three consecutive images from each camera. And then, ego-motions are estimated with calibration result.

The following section provides the two-camera geometry. Section III describes estimation procedure. Finally, the evaluation of the experiments is given in Section IV.

2 Two-Camera Geometry

Our application of the ego-motion estimation is stabilization for "Dive into Movie". Cameras are affixed to the actors' body and move consistently with person (see Fig. 2). First of all, it is convenient to assign frames of reference.

W : a fixed frame of reference;
$C_1(k)$: the camera1 frame located at the optical center of camera1 with positive z axis along the optical axis at time k, attached on person 1, watching camera2, varying with camera1's motion;
$C_2(k)$: the camera2 frame located at the optical center of camera2 with positive z axis along the optical axis at time k, attached on person 2, watching camera1, varying with camera2 motion.

The relation between any two coordinates is represented by the rotation matrix $R_{a->b} \in SO(3)$ and a translation vector $t_{a->b} \in \Re^3$. $T_{a->b} = [R_{a->b} \quad t_{a->b}; \quad 0_{3\times 3} \quad 1]$ is the transformation from coordinates a to coordinates b. We express a point X_a with respect to the reference a, then $X_b = T_{a->b} X_a$.

We assume that the internal parameters of cameras are initialized as known. Given enough correct feature matches in two views (with static scenes) captured by the same camera, camera ego-motion can be computed easily.

In the following, we first recall two-view geometry of conventional static scene. And then foreground motion is taken into account. Finally, four-view (two from each camera) geometry is derived by 3D motion analysis on two moving cameras.

2.1 Two-View Geometry: Epipolar Constraint and Essential Matrix

As well known, the Essential matrix constrains the motion of points between two views from one camera. It encodes the epipolar constraint and motion matrix. The set of homogeneous image points $\{\mathrm{x}_i\}, i = 1,...,n$ in the first image is related to the set $\{\mathrm{x}'_i\}, i = 1,...,n$ in the second image by Essential matrix with the following equation:

$$x'_i E x_i = 0, E = \hat{T}R, \hat{T} = [0 \quad -t_3 \quad t_2; \quad t_3 \quad 0 \quad -t_1; \quad -t_2 \quad t_1 \quad 0] \qquad (1)$$

From above equation, given feature matches in two-view, Essential matrix can be determined, and then rotation matrix and translation vector can be computed up to a universal scale. We used RANSAC [1] for transformation matrix estimation.

2.2 Two-View Geometry with Moving Foreground

Let a set of homogeneous 3D space points $\{X_{F,i}(k)\}, i = 1,...,n$ be positions of foreground points at time k in the view of camera1 with a rigid motion independently from camera1's motion. Then, motion of these points in $C_1(k)$ can be represented as

$$\begin{aligned} X_{F_1,C_1}(k) &= T_{F_1,C_1}(k) X_{F_1,C_1}(k-1) \\ &= T_{C_1}^{-1}(k) T_{C_1<-W}(k-1) T_{F_1,W}(k) T_{W<-C_1}(k-1) X_{F_1,C_1}(k-1) \end{aligned} \tag{2}$$

where $T_{F,C_1}(k)$ represents 3D foreground motion in C_1's coordinates from time k-1 to k. $T_{C_1}(k)$ is C_1's motion and $T_{F_1,W}(k)$ is foreground motion in world coordinates. $T_{F,C_1}(k)$ and $T_{C_1}(k)$ can be computed with two-view geometry described in Section 2.1 using feature matches in foreground region and background region respectively. If there is no enough background feature matches for $T_{C_1}(k)$, and if $T_{F_1,W}(k)$ is given by some other way, $T_{C_1}(k)$ can be computed using Equation (2).

2.3 Four-View Geometry of Two Face-to-Face Cameras

In this case (see Fig.2), motion of camera1's foreground points F_1 in $C_2(k)$ coordinates is the same as C_2's motion $T_{C_2}(k)$, i.e. $T_{F_1,C_2}(k) = T_{C_2}(k)$. Then

$$T_{F_1,W}(k) = T_{W<-C_2}(k-1) T_{F_1,C_2}(k) T_{C_2<-W} = T_{W<-C_2}(k-1) T_{C_2}(k) T_{C_2<-W}(k-1) \tag{3}$$

Now, let's derive relations among four 3D motion transformation matrices: $T_{F_1,C_1}(k)$, $T_{C_2}(k)$, $T_{C_1}(k)$ and $T_{F_2,C_2}(k)$. With the relations, given any three matrices of these four, remaining unknown matrix can be computed. $T_{C_2}(k)$ and $T_{C_1}(k)$ are the target matrices in this paper. From Equation (2) and (3), we have

$$\begin{aligned} T_{F_1,C_1}(k) &= T_{C_1}^{-1}(k) T_{C_1<-W}(k-1) T_{W<-C_2}(k-1) T_{C_2}(k) T_{W<-C_1}(k-1) T_{C_2<-W}(k-1) \\ &= T_{C_1}^{-1}(k) T_{C_1<-C_2}(k-1) T_{C_2}(k) T_{C_1<-C_2}^{-1}(k-1) \end{aligned}$$

If we let $T_{C_2<-C_1} = T_{C2-1}$ for simplicity, then we have

$$T_{F_1,C_1}(k) = T_{C_1}^{-1}(k) T_{C1-2}(k-1) T_{C_2}(k) T_{C1-2}^{-1}(k-1) \tag{4}$$

Similarly, considering foreground points of Camera2, we can obtain:

$$T_{F_2,C_2}(k) = T_{C_2}^{-1}(k) T_{C2-1}(k-1) T_{C_1}(k) T_{C2-1}^{-1}(k-1) \tag{5}$$

Now let check relations between above matrices and image observations.

a) $T_{F_1,C_1}(k)$: motion of foreground points (belong to person2) in camera1;

b) $T_{F_2,C_2}(k)$: motion of foreground points (belong to person1) in camera2,

c) $T_{C_2}(k)$: computed from motion of background points in camera2

d) $T_{C_1}(k)$: computed from motion of background points in camera1;

e) $T_{C2-1}(k-1)$: Extrinsic calibration matrix between camera1 and camera2.

a)-d) can be computed using two-view relations described in Section 2.1. e) can not be directly computed, and to be determined in Section 3.1.

3 Synchronized Estimation

Recall the overview of the algorithm in Fig. 3. Synchronized estimation stage is divided into two steps: extrinsic calibration using frames at time k-3, k-2 and k-1 and motion estimation using frames at time k-1 and k.

3.1 Extrinsic Calibration of Two Face-to-Face Cameras

First we present an outline of our calibration procedure, and then the details of each step will be presented. The extrinsic calibration of two cameras is broken down into the following steps:

a) for time k-3, k-2 and k-1, compute T_{F_1,C_1}, T_{F_2,C_2}, T_{C_2} and T_{C_1} with steps in Section 2.3 ;

b) compute $T_{C2-1}(k-1)$ using Equation (6) in the following soon.

To get a unique solution, at least three views from one camera is necessary [6], with avoiding special configurations of view angles. In the following equation, matrices with underline denote that they can be calculated as known. Using Equation (4)

$$\begin{cases} \underline{T_{F_1,C_1}(k-1)} = \underline{T_{C_1}^{-1}(k-1)}T_{C1-2}(k-2)\underline{T_{C_2}(k-1)}T_{C1-2}^{-1}(k-2) \\ T_{C1-2}(k-1) = \underline{T_{C_1}^{-1}(k-1)}T_{C1-2}(k-2)\underline{T_{C_2}(k-1)} \end{cases} \Rightarrow$$

$$\underline{T_{F_1,C_1}(k-1)} = T_{C1-2}(k-1)\underline{T_{C_2}(k-1)}T_{C1-2}^{-1}(k-1)\underline{T_{C_1}^{-1}(k-1)} \tag{6}$$

$$\underline{T_{F_1,C_1}(k-1)}\underline{T_{C_1}(k-1)}T_{C1-2}(k-1) = T_{C1-2}(k-1)\underline{T_{C_2}(k-1)} \tag{7}$$

From Equation (6),

$$\begin{cases} \underline{T_{F_1,C_1}(k-2)} = T_{C1-2}(k-2)\underline{T_{C_2}(k-2)}T_{C1-2}^{-1}(k-2)\underline{T_{C_1}^{-1}(k-2)} \\ T_{C1-2}(k-2) = \underline{T_{C_1}(k-1)}T_{C1-2}(k-1)\underline{T_{C_2}^{-1}(k-1)} \end{cases} \Rightarrow$$

$$\begin{aligned} &\underline{T_{F_1,C_1}(k-2)} \\ &= \underline{T_{C_1}(k-1)}T_{C1-2}(k-1)\underline{T_{C_2}^{-1}(k-1)}\underline{T_{C_2}(k-2)}\underline{T_{C_2}(k-1)}T_{C1-2}^{-1}(k-1)\underline{T_{C_1}^{-1}(k-1)}\underline{T_{C_1}^{-1}(k-2)} \end{aligned} \tag{8}$$

$$\begin{aligned} &\underline{T_{C_1}(k-1)}\underline{T_{F_1,C_1}(k-2)}\underline{T_{C_1}(k-2)}\underline{T_{C_1}(k-1)}T_{C1-2}(k-1) \\ &= T_{C1-2}(k-1)\underline{T_{C_2}^{-1}(k-1)}\underline{T_{C_2}(k-2)}\underline{T_{C_2}(k-1)} \end{aligned} \tag{9}$$

In estimation of extrinsic motion, we decompose T into R and t. Then problem can be simplified as compute X that satisfies $AX = XB$ in Equation (7) and (9) for $X = R_{C1-2}(k-1)$. $t_{C1-2}(k-1)$ can be easily obtained from $R_{C1-2}(k-1)$ and Equation(7), (9). Here, both A and B are known, and X is unknown and has to be solved. While solutions to this question have been studied when A and B are general n $n \times n$ matrices, here we need solutions that belong to Euclidean group.

In the context of robot sensor calibration, [6] first motivate this equation, and provide a closed-form solution. Their approach is based on geometric interpretations of the eigen-values and eigenvectors of a rotation matrix. Both translation and orientation values are calculated simultaneously using least-square fitting. [7] used this formulation of the problem and developed a non-linear optimization technique to solve it. Martin and Park [8] derive a closed form solution as a linear least squares fit . [9] formulated the problem using canonical coordinates of the rotation group, which enables a particularly simple closed form solution.

In [6], conditions for uniqueness of solutions are discussed. It is concluded that the solution can not be found with only one measurement, and the parameters can be uniquely estimated with two camera positions, but the orientations of the camera cannot be zero or π value. In this paper, we used the approach described in [8].

3.2 Ego-Motion Estimation

Given $T_{F_1,C_1}(k)$ - motion of foreground point in view of camera1 and $T_{C_2}(k)$ - motion of background points in view of camera2, and $T_{C1-2}(k-1)$ obtained in Section 3.2, $T_{C_1}(k)$ is computed using Equation (4).

$$\underline{T_{F_1,C_1}(k)} = T_{C_1}^{-1}(k)\underline{T_{C1-2}(k-1)T_{C_2}(k)T_{C1-2}^{-1}(k-1)} \tag{10}$$

4 Evaluations

Both simulated data and real data are used for evaluations. Using synthesis data, we check the accuracy of the approach and the sensitivity to various levels of noise. Using real data, the procedure outlined in Fig. 3. is implemented along with the proposed calibration and estimation approach. Furthermore, stabilization results using estimated ego-motion matrices are shown to prove the feasibility and accuracy of the approach.

4.1 Evaluations with Simulated Data

The simulated data was created using a set of known 3D points and transformations. Transformations between two cameras and ego-motions of both cameras are constructed with random rotation axis, angle and translation vector.

Additionally, in order to analysis the influence of noise, the data sets were defined by the radius of the Gaussian noise in the 2D pixel points. We create data sets with three different levels of noise. The resulting error in the calibration transformation is plotted in Fig.4. The error in the final estimation of transformation matrix, or residual error, is defined as $\left\| T_{true} - T_{estimated} \right\|_F$, where $\left\| \cdot \right\|_F$ is the Frobenious norm of the matrix.

Referring to the results (Fig.4 and 5), some interesting observations are made. The proposed approach produces results with low error. In Fig.5, we also showed the residual error resulted from noise in calibration matrix. We can find that the noise in calibration matrix doesn’t impact the final estimation result much. This is because the error in calibration stage might be eliminated by an inverse computation.

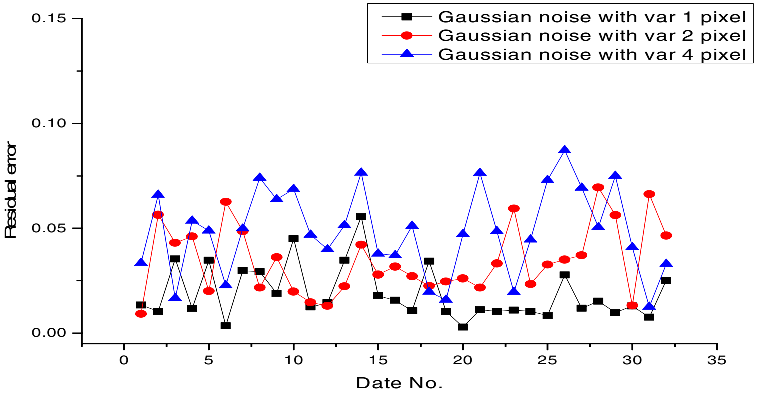


Fig. 4. Residual error of calibration result with different level of noise added to 2d image pixels

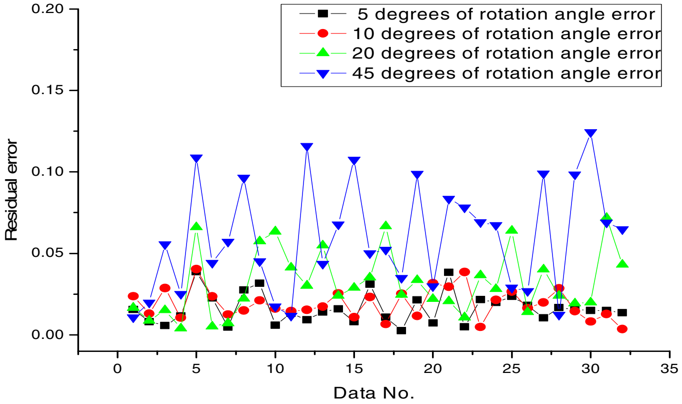


Fig. 5. Residual error of estimated matrices with different level of noise on calibration matrix

4.2 Experimental Results with Real Data

For real experiments, recall the overview of the algorithm in Fig. 3. We used a real video data with cameras affixed on the body. Before estimation, synchronized input videos from both cameras are pre-processed to segment out the background region and object region. In this step, color distribution based mean-shift region tracking [11] is implemented for object region. SIFT features [12] are extracted and matched within consecutive two image pairs for background region and object respectively. In Fig.6, we showed the result of feature matches with SIFT features.

The synchronized estimation step has two stages. Extrinsic parameters are calibrated in first stage for two cameras with totally four image pairs (two for each camera using data at three time steps) using method in Section 3.1. Two-view transformation matrices for foreground and background region are computed using RANSAC [1]. Calibration matrix is computed using the approach described in [8]. Ego-motion is estimated using method in Section 3.2.

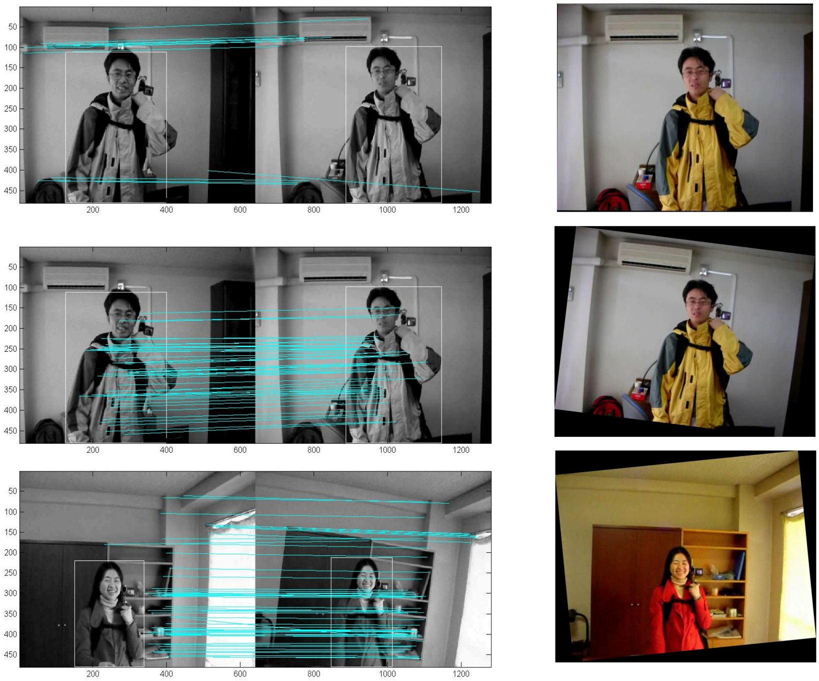
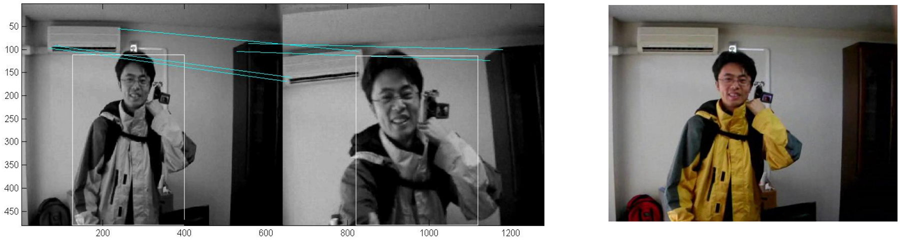


Fig. 6. One set of data for transformation computation (A and B) in calibration stage. Left column: Up) Point matches in background region of video2; Middle) Point matches in foreground region of video2; Down) Point matches in background region of video1. Right column: Up) stabilization result using background region in video2. Middle) stabilization result using foreground region in video2. Down) Stabilization result using background region in video1.



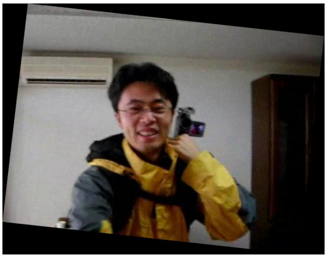


Fig. 7. Stabilization results. Left-top: point matches in background region of video2. No enough features and the conventional approach failed in this case. Right-top: original image before motion. Down: stabilization result using the proposed approach. Compared with original image on the right-top, we can see our approach can provide acceptable stabilization result.

Finally, a 2D affine transformation is derived from motion matrix for stabilization only considering effect of rotation: $x' = sRx$, where we set $s = 1/R_{33}$ for simplicity. x and x' are homogeneous image points before and after motion. Since the main purpose of this paper is ego-motion recovery, stabilization has not been carefully considered, which can be our future work.

In Fig. 7, we showed the stabilization result on background region and foreground region, they all failed. Stabilization result with the proposed approach is given. Compared with original image, we can see it provides acceptable result.

5 Conclusion

Accurate estimation of ego-motion is not easy when there is moving foreground. Especially in some special situations it's almost impossible. To address the problem, we proposed a new approach that utilizes additional video captured by the camera attached on the foreground object (i.e. another actor in our application).

We first configure the sensor system as two face-to-face moving cameras. And then we derived the relationship between four views from two cameras. In estimation stage, two cameras are calibrated firstly, and then ego-motion is estimated. We calibrate the extrinsic relationship of two cameras with an AX=XB formulation. Experiments with simulated data and real data verify that this approach can provide acceptable ego-motion estimation and stabilization results.

Acknowledgment

This work was supported in part by the NKBRPC (No. 2006CB303100), NSFC Grant (No. 60333010), NSFC Grant (No. 60605001) and Key grant Project of Chinese Ministry of Education (No. 103001).

References

1. Hartley, R.I., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press, Cambridge (2000)
2. Faugeras, O., Luong, Q.T., Papadopoulo, T.: The geometry of multiple images. MIT Press, Cambridge (2001)
3. Schindler, K., Suter, D.: Two-view multibody structure-and-motion with outliers through model selection. IEEE T-PAMI 28(6), 983–995 (2006)
4. Wolf, L., Shashua, A.: Two-body segmentation from two perspective views. In: Proc. CVPR, pp. 263–270 (2001)
5. Makadia, A., Daniilidis, K.: Correspondenceless Ego-Motion Estimation Using an IMU. In: Proceedings of the IEEE International Conference on Robotics and Automation (2005)
6. Shiu, Y.C., Ahmad, S.: Calibration of wrist-mounted robotic sensors by solving homogenous transform equations of the form AX = XB. IEEE Transactions on Robotics and Automation 5(1), 16–29 (1989)
7. Li, M.: Kinematic calibration of an active head-eye system. IEEE Transactions on Robotics and Automation 14(1), 153–157 (1998)

8. Park, F.C., Martin, B.J.: Robot sensor calibration: Solving AX = XB on the Euclidean group. IEEE T-RA 10(5), 717–721 (1994)
9. Neubert, J., Ferrier, N.J.: Robust active stereo calibration. In: Proceedings of the IEEE International Conference on Robotics and Automation, vol. 3, pp. 2525–2531 (2002)
10. Sato, J.: Recovering Multiple View Geometry from Mutual Projections of Multiple Cameras. Int. J. Comput. Vision 66(2), 123–140 (2006)
11. Comaniciu, D., Ramesh, V., Meer, P.: Kernel-Based Object Tracking. IEEE Trans. Pattern Analysis Machine Intell. 25(5), 564–575 (2003)
12. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 60(2), 91–110 (2004)