Visual Odometry for Non-Overlapping Views Using Second-Order Cone Programming

Jae-Hak Kim[1], Richard Hartley[1], Jan-Michael Frahm[2] and Marc Pollefeys[2]

[1]Research School of Information Sciences and Engineering
The Australian National University.

[1]National ICT Australia (NICTA) *

[2]Department of Computer Science
University of North Carolina at Chapel Hill.

Abstract. We present a solution for motion estimation for a set of cameras which are firmly mounted on a head unit and do not have overlapping views in each image. This problem relates to ego-motion estimation of multiple cameras, or visual odometry. We reduce motion estimation to solving a triangulation problem, which finds a point in space from multiple views. The optimal solution of the triangulation problem in L-infinity norm is found using SOCP (Second-Order Cone Programming) Consequently, with the help of the optimal solution for the triangulation, we can solve visual odometry by using SOCP as well.

1 Introduction

Motion estimation of cameras or pose estimation, mostly in the case of having overlapping points or tracks between views, has been studied in computer vision research for many years [1]. However, non-overlapping or slightly overlapping camera systems have not been studied so much, particulary the motion estimation problem. The non-overlapping views mean that all images captured with cameras do not have any, or at most have only a few common points. There are potential applications for this camera system. For instance, we construct a cluster of multiple cameras which are firmly installed on a base unit such as a vehicle, and the cameras are positioned to look at different view directions. A panoramic or omnidirectional image can be obtained from images captured with a set of cameras with small overlap. Another example is a vehicle with cameras mounted on it to provide driving assistance such as side/rear view cameras.

An important problem is visual odometry – how can we estimate the tracks of a vehicle and use this data to determine where the vehicle is placed. There has been prior research considering a set of many cameras moving together as one camera. In [2] an algebraic solution to the multiple camera motion problem is presented. Similar research on planetary rover operations has been conducted to

* NICTA is funded by the Australian Government's Backing Australia's Ability initiative, in part through the Australian Research Council.

estimate the motion of a rover on Mars and to keep track of the rover [3]. Other research on visual odometry has been performed to estimate the motion of a stereo rig or single camera [4]. Prior work on non-overlapping cameras includes most notably the paper [5]. This differs from our work in aligning independently computed tracks of the different cameras, whereas we compute a motion estimate using all the cameras at once. Finally, an earlier solution to the problem was proposed in unpublished work of [6], which may appear elsewhere.

In this paper, we propose a solution to estimate the six degrees of freedom (DOFs) of the motion, three rotation parameters and three translation parameters (including scale), for a set of multiple cameras with non-overlapping views, based on L_∞ triangulation. A main contribution of this paper is that we provided a well-founded geometric solution to the motion estimation in non-overlapping multiple cameras.

2 Problem formulation

Consider a set of n calibrated cameras with non-overlapping fields of view. Since the cameras are calibrated, we may assume that they are all oriented in the same way just to simplify the mathematics. This is easily done by multiplying an inverse of the rotation matrix to the original image coordinates. This being the case, we can also assume that they all have camera matrices originally equal to $\mathtt{P}_i = [\mathtt{I}| - \mathbf{c}_i]$. We assume that all $\mathbf{c}_i$ are known.

The cameras then undergo a common motion, described by a Euclidean matrix

$$\mathtt{M} = \begin{bmatrix} \mathtt{R} & -\mathtt{R}\,\mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix} \tag{1}$$

where $\mathtt{R}$ is a rotation, and $\mathbf{t}$ is a translation of a set of cameras. Then, the i-th camera matrix changes to

$$\mathtt{P}'_i = \mathtt{P}_i\,\mathtt{M}^{-1} = [\mathtt{I} \mid -\mathbf{c}_i] \begin{bmatrix} \mathtt{R}^\top & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix} = [\mathtt{R}^\top \mid \mathbf{t} - \mathbf{c}_i]$$

which is located at $\mathtt{R}(\mathbf{c}_i - \mathbf{t})$.

Suppose that we compute all the essential matrices of the cameras independently, then decompose them into rotation and translation. We observe that the rotations computed from all the essential matrices are the same. This is true only because all the cameras have the same orientation. We can average them to get an overall estimate of rotation. Then, we would like to compute the translation. As we will demonstrate, this is a triangulation problem.

Geometric concept. First, let us look at a geometric idea derived from this problem. An illustration of a motion of a set of cameras is shown in Figure 1. A bundle of cameras is moved by a rotation $\mathtt{R}$ and translation $\mathbf{t}$. All cameras at $\mathbf{c}_i$ are moved to $\mathbf{c}'_i$. The first camera at position $\mathbf{c}'_1$ is a sum of vectors $\mathbf{c}_i$, $\mathbf{c}'_i - \mathbf{c}_i$ and $\mathbf{c}'_1 - \mathbf{c}'_i$ where $i = 1...3$. Observing that the vector $\mathbf{v}_i$ in Figure 1

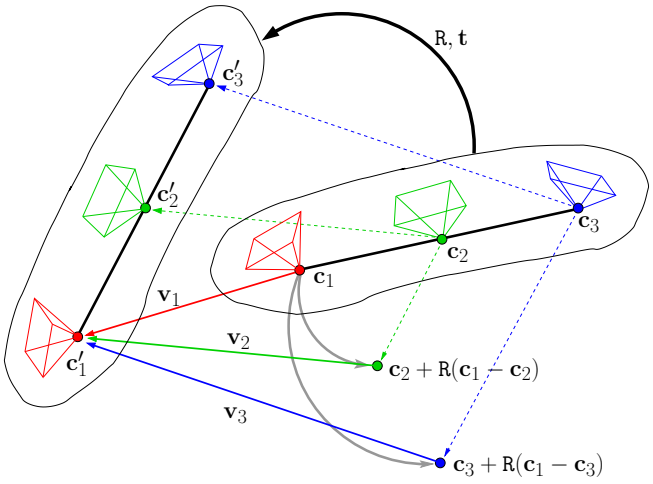


Fig. 1. *A set of cameras is moved by Euclidean motion of rotation* $\mathtt{R}$ *and translation* $\mathbf{t}$. *The centre of the first camera* $\mathbf{c}_1$ *is moved to* $\mathbf{c}'_1$ *by the motion. The centre* $\mathbf{c}'_1$ *is a common point where all translation direction vectors meet. The translation direction vectors are indicated as red, green and blue solid arrows which are* $\mathbf{v}_1$, $\mathbf{v}_2$ *and* $\mathbf{v}_3$, *respectively. Consequently, this is a triangulation problem.*

is the same as the vector $\mathbf{c}'_i - \mathbf{c}_i$ and the vector $\mathbf{c}'_1 - \mathbf{c}'_i$ is obtained by rotating the vector $\mathbf{c}_1 - \mathbf{c}_i$, the first camera at position $\mathbf{c}'_1$ can be rewritten as a sum of three vectors $\mathbf{c}_i$, $\mathtt{R}(\mathbf{c}_1 - \mathbf{c}_i)$ and $\mathbf{v}_i$. Therefore, the three vectors $\mathbf{v}_i$, colored solid arrows in Figure 1 meet in one common point $\mathbf{c}'_1$, the position of the centre of the first camera after the motion. It means that finding the motion of the set of cameras is the same as solving a triangulation problem for translation direction vectors derived from each view.

Secondly, let us derive detailed equations on this problem from the geometric concept we have described above. Let $\mathtt{E}_i$ be the essential matrix for the i-th camera. From $\mathtt{E}_1$, we can compute the translation vector of the first camera, $\mathtt{P}_1$, in the usual way. This is a vector passing through the original position of the first camera. The final position of this camera must lie along this vector. Next, we use $\mathtt{E}_i$, for $i > 1$ to estimate a vector along which the final position of the **first** camera can be found. Thus, for instance, we use $\mathtt{E}_2$ to find the final position of $\mathtt{P}_1$. This works as follows. The i-th essential matrix $\mathtt{E}_i$ decomposes into $\mathtt{R}_i = \mathtt{R}$ and a translation vector $\mathbf{v}_i$. In other words, $\mathtt{E}_i = \mathtt{R}[\mathbf{v}_i]_\times$. This means that the i-th camera moves to a point $\mathbf{c}_i + \lambda_i \mathbf{v}_i$, the value of λ_i being unknown. This point is the final position of each camera $\mathbf{c}'_i$ in Figure 1. We transfer this motion to determine the motion of the first camera. We consider the motion as taking place in two stages, first rotation, then translation. First the camera centre $\mathbf{c}_1$ is rotated by $\mathtt{R}$ about point $\mathbf{c}_i$ to point $\mathbf{c}_i + \mathtt{R}(\mathbf{c}_1 - \mathbf{c}_i)$. Then it is translated in the direction $\mathbf{v}_i$ to the point $\mathbf{c}'_1 = \mathbf{c}_i + \mathtt{R}(\mathbf{c}_1 - \mathbf{c}_i) + \lambda_i \mathbf{v}_i$. Thus, we see that $\mathbf{c}'_1$ lies on the line with direction vector $\mathbf{v}_i$, based at point $\mathbf{c}_i + \mathtt{R}(\mathbf{c}_1 + \mathbf{c}_i)$.

In short, each essential matrix $\mathtt{E}_i$ constrains the final position of the first camera to lie along a line. These lines are not all the same, in fact unless $\mathtt{R} = \mathtt{I}$, they are all different. The problem now comes down to finding the values of λ_i and $\mathbf{c}'_1$ such that for all i:

$$\mathbf{c}'_1 = \mathbf{c}_i + \mathtt{R}(\mathbf{c}_1 - \mathbf{c}_i) + \lambda_i \mathbf{v}_i \quad \text{for } i = 1, ..., n \ . \tag{2}$$

Having found $\mathbf{c}'_1$, we can get $\mathbf{t}$ from the equation $\mathbf{c}'_1 = \mathtt{R}(\mathbf{c}_1 - \mathbf{t})$.

Averaging Rotations. From the several cameras and their essential matrices $\mathtt{E}_i$, we have several estimates $\mathtt{R}_i = \mathtt{R}$ for the rotation of the camera rig. We determine the best estimate of $\mathtt{R}$ by averaging these rotations. This is done by representing each rotation $\mathtt{R}_i$ as a unit quaternion, computing the average of the quaternions and renormalizing to unit norm. Since a quaternion and its negative both represent the same rotation, it is important to choose consistently signed quaternions to represent the separate rotations $\mathtt{R}_i$.

Algebraic derivations. Alternatively, it is possible to show an algebraic derivation of the equations as follows. Given $\mathtt{P}_i = [\mathtt{I}| - \mathbf{c}_i]$ and $\mathtt{P}'_i = [\mathtt{R}^\top \mid \mathbf{t} - \mathbf{c}_i]$ (See (2)), an essential matrix is written as

$$\mathtt{E}_i = \mathtt{R}^\top [\mathbf{c}_i + \mathtt{R}(\mathbf{t} - \mathbf{c}_i)]_\times \mathtt{I} = [\mathtt{R}^\top \mathbf{c}_i + (\mathbf{t} - \mathbf{c}_i)]_\times \mathtt{R}^\top \ . \tag{3}$$

Considering that the decomposition of the essential matrix $\mathtt{E}_i$ is $\mathtt{E}_i = \mathtt{R}_i[\mathbf{v}_i]_\times = [\mathtt{R}_i\mathbf{v}_i]_\times \mathtt{R}_i$, we may get the rotation and translation from (3), namely $\mathtt{R}_i = \mathtt{R}^\top$ and $\lambda_i \mathtt{R}_i \mathbf{v}_i = \mathtt{R}^\top \mathbf{c}_i + (\mathbf{t} - \mathbf{c}_i)$. As a result, $\mathbf{t} = \lambda_i \mathtt{R}^\top \mathbf{v}_i + \mathbf{c}_i - \mathtt{R}^\top \mathbf{c}_i$ which is the same equation derived from the geometric concept.

A Triangulation Problem. Equation (2) gives us independent measurements of the position of point $\mathbf{c}'_1$. Denoting $\mathbf{c}_i + \mathtt{R}(\mathbf{c}_1 - \mathbf{c}_i)$ by $\mathbf{C}_i$, the point $\mathbf{c}'_1$ must lie at the intersection of the lines $\mathbf{C}_i + \lambda_i \mathbf{v}_i$. In the presence of noise, these lines will not meet, so we need find a good approximation to $\mathbf{c}'_1$. Note that the points $\mathbf{C}_i$ and vectors $\mathbf{v}_i$ are known, having been computed from the known calibration of the camera geometry, and the computed essential matrices $\mathtt{E}_i$.

The problem of estimating the best $\mathbf{c}'_1$ is identical with the triangulation problem studied (among many places) in [7, 8]. We adopt the approach of [7] of solving this under L_∞ norm. The derived solution is the point $\mathbf{c}'_1$ that minimizes the maximum difference between $\mathbf{c}'_1 - \mathbf{C}_i$ and the direction vector $\mathbf{v}_i$, for all i. In the presence of noise, the point $\mathbf{c}'_1$ will lie in the intersection of cones based at the vertex $\mathbf{C}_i$, and with axis defined by the direction vectors $\mathbf{v}_i$.

To formulate the triangulation problem, instead of $\mathbf{c}'_1$, we write $\mathbf{X}$ as the final position of the first camera where all translations derived from each essential matrix meet together. As we have explained in the previous section, in the presence of noise we have n cones, each one aligned with one of the translation directions. The desired point $\mathbf{X}$ lies in the overlap of all these cones, and, finding this overlap region gives the solution we need in order to get the motion

of cameras. Then, our original motion estimation problem is formulated as the following minimization problem:

$$\min_{\mathbf{X}} \max_{i} \frac{||(\mathbf{X}-\mathbf{C}_i)\times\mathbf{v}_i||}{(\mathbf{X}-\mathbf{C}_i)^\top\mathbf{v}_i} \ . \tag{4}$$

Note that the quotient is equal to $\tan^2(\theta_i)$ where θ_i is the angle between $\mathbf{v}_i$ and $(\mathbf{X}-\mathbf{C}_i)$. This problem can be solved as a Second-Order Cone Programming (SOCP) using a bisection algorithm [9].

3 Algorithm

The algorithm to estimate motion of cameras having non-overlapping views is as follows:

Given:

1. A set of cameras described in initial position by their known calibrated camera matrices $\mathtt{P}_i = \mathtt{R}_i[\mathtt{I}|-\mathbf{c}_i]$. The cameras then move to a second (unknown) position, described by camera matrices $\mathtt{P}'_i$.
2. For each camera pair $\mathtt{P}_i, \mathtt{P}'_i$, several point correspondences $\mathbf{x}_{ij} \leftrightarrow \mathbf{x}'_{ij}$ (expressed in calibrated coordinates as homogeneous 3-vectors).

Objective: Find the motion matrix of the form (1) that determines the common motion of the cameras, such that $\mathtt{P}'_i = \mathtt{P}_i\mathtt{M}^{-1}$.

Algorithm:

1. Normalize the image coordinates to calibrated image coordinates by setting
$$\hat{\mathbf{x}}_{ij} = \mathtt{R}_i^{-1}\mathbf{x}_{ij} \ \text{ and } \ \hat{\mathbf{x}}'_{ij} = \mathtt{R}_i^{-1}\mathbf{x}'_{ij} \ ,$$
then adjust to unit length by setting $\hat{\mathbf{x}}_{ij} \leftarrow \hat{\mathbf{x}}_{ij}/\|\hat{\mathbf{x}}_{ij}\|$ and $\hat{\mathbf{x}}'_{ij} \leftarrow \hat{\mathbf{x}}'_{ij}/\|\hat{\mathbf{x}}'_{ij}\|$.
2. Compute each essential matrix $\mathtt{E}_i$ in terms of correspondences $\hat{\mathbf{x}}_{ij} \leftrightarrow \hat{\mathbf{x}}'_{ij}$ for the i-th camera.
3. Decompose each $\mathtt{E}_i$ as $\mathtt{E}_i = \mathtt{R}_i[\mathbf{v}_i]_\times$ and find the rotation $\mathtt{R}$ as the average of the rotations $\mathtt{R}_i$. Set $\mathbf{C}_i = \mathbf{c}_i + \mathtt{R}(\mathbf{c}_1 - \mathbf{c}_i)$.
4. Solve the triangulation problem by finding the point $\mathbf{X} = \mathbf{c}'_1$ that (approximately because of noise) satisfies the condition $\mathbf{X} = \mathbf{C}_i + \lambda_i\mathbf{v}_i$ for all i.
5. Compute $\mathbf{t}$ from $\mathbf{t} = \mathbf{c}_1 - \mathtt{R}^\top\mathbf{c}'_1$.

In our current implementation, we have used the L_∞ norm to solve the triangulation problem. Other methods of solving the triangulation problem may be used, for instance the optimal L_2 triangulation method given in [8].

Critical Motion. The algorithm has a critical condition when the rotation is zero. If this is so then, in the triangulation problem solved in this algorithm all the basepoints $\mathbf{C}_i$ involved are the same. Thus, we encounter a triangulation problem with a zero baseline. In this case, the magnitude of the translation can not be accurately determined.

4 Experiments

We have used SeDuMi and Yalmip toolbox for optimization of SOCP problems [10, 11]. We used a five point solver to estimate the essential matrices [12, 13]. We select the best five points from images using RANSAC to obtain an essential matrix, and then we improve the essential matrix by non-linear optimization.

An alternative method for computing the essential matrix based on [14] was tried. This method gives the optimal essential matrix in L_∞ norm. A comparison of the results for these two methods for computing $\mathtt{E}_i$ is given in Fig 6.

Real data. We used Point Grey's Ladybug$^{\text{TM}}$ camera to generate some test data for our problem . This camera unit consists of six 1024×768 CCD color sensors with small overlap of their field of view. The six cameras, 6 sensors with 2.5mm lenses, are closely packed on a head unit. Five CCDs are positioned in a horizontal ring around the head unit to capture side-view images, and one is located on the top of the head unit to take top-view images. Calibration information provided by Point Grey [15] is used to get intrinsic and relative extrinsic parameters of all six cameras.

A piece of paper is positioned on the ground, and the camera is placed on the paper. Some books and objects are randomly located around the camera. The camera is moved manually while the positions of the camera at some points are marked on the paper as edges of the camera head unit. These marked edges on the paper are used to get the ground truth of relative motion of the camera for this experiment. The experimental setup is shown in Figure 2. A panoramic image stitched in our experimental setup is shown in Figure 3.

In the experiment, 139 frames of image are captured by each camera. Feature tracking is performed on the image sequence by the KLT (Kanade-Lucas-Tomasi) tracker [16]. Since there is lens distortion in the captured image, we correct the image coordinates of the feature tracks using lens distortion parameters provided by the Ladybug SDK library. The corrected image coordinates are used in all the equations we have derived. After that, we remove outliers from the feature tracks by the RANSAC (Random Sample Consensus) algorithm with a model of epipolar geometry in two view and trifocal tensors in three view [17].

There are key frames where we marked the positions of the camera. They are frames 0, 30, 57, 80, 110 and 138 in this experiment. The estimated path of the cameras over the frames is shown in Figure 4. After frame 80, the essential matrix result was badly estimated and subsequent estimation results were erroneous.

A summary of the experimental results is shown in Tables 1 and 2. As can be seen, we have acquired a good estimation of rotations from frame 0 up to frame 80, within about one degree of accuracy. Adequate estimation of translations is reached up to frame 57 within less than 0.5 degrees. We have successfully tracked the motion of the camera through 57 frames. Somewhere between frame 57 and frame 80 an error occurred that invalidated the computation of the position of frame 80. Analysis indicates that this was due to a critical motion (near-zero rotation of the camera fixture) that made the translation estimation inaccurate. Therefore, we have shown the frame-to-frame rotations, over frames in Figure

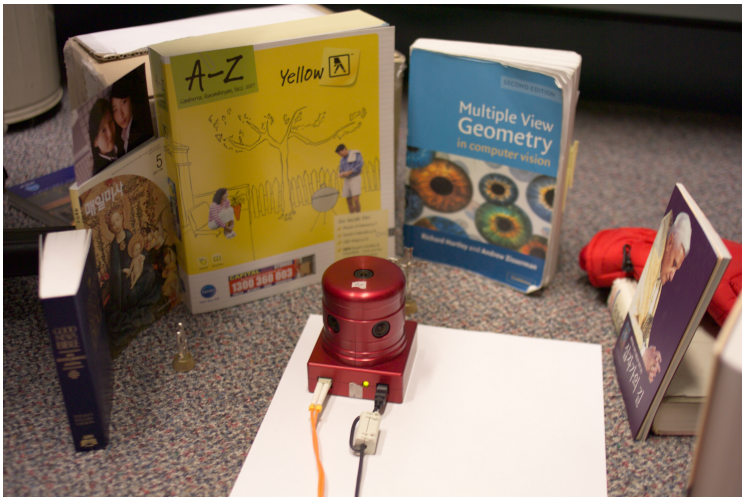


Fig. 2. *An experimental setup of the Ladybug™ camera on an A3 size paper surrounded by books. The camera is moved on the paper by hand, and the position of the camera at certain key frames is marked on the paper to provide the ground truth for the experiments.*

Rotation pair	True rotation		Estimated rotation	
	Axis	Angle	Axis	Angle
$(\mathtt{R}_0, \mathtt{R}_1)$	[0 0 -1]	85.5°	[-0.008647 -0.015547 0.999842]	85.15°
$(\mathtt{R}_0, \mathtt{R}_2)$	[0 0 -1]	157.0°	[-0.022212 -0.008558 0.999717]	156.18°
$(\mathtt{R}_0, \mathtt{R}_3)$	[0 0 -1]	134.0°	[0.024939 -0.005637 -0.999673]	134.95°

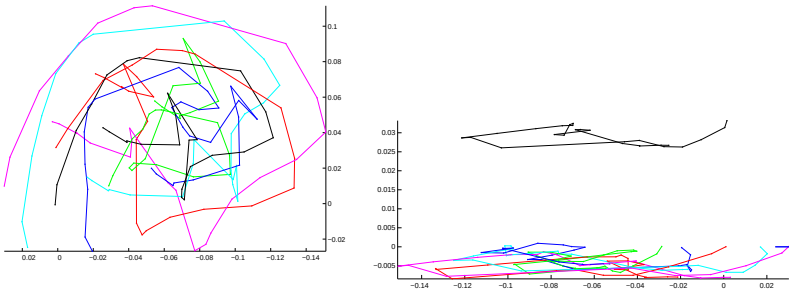
Table 1. *Experimental results of rotations at key frames 0, 30, 57 and 80, which correspond to the position number 0–3, respectively. For instance, a pair of rotation* $(\mathtt{R}_0, \mathtt{R}_1)$ *corresponds to a pair of rotations at key frame* 0 *and* 30*. Angles of each rotation are represented by the axis-angle rotation representation.*

5-(a). As can be seen, there are frames for which the camera motion was less than 5 degrees. This occurred for frames 57 to 62, 67 to 72 and 72 to 77.

In Figure 5-(c), we have shown the difference between the ground truth and estimated position of the cameras in this experiment. As can be seen, the position of the cameras is accurately estimated up to 57 frames. However, the track went off at frame 80. A beneficial feature of our method is that we can avoid such bad condition for the estimation by looking at the angles between frames and residual errors on the SOCP, and then we try to use other frames for the estimation.

Using the L_∞ optimal E-matrix. The results so-far were achieved using the 5-point algorithm (with iterative refinement) for calculating the essential matrix. We also tried using the method given in [14]. Since this method is quite new, we did not have time to obtain complete results. However, Fig 6 compares the

Fig. 3. *A panoramic image is obtained by stitching together all six images from the Ladybug™ camera. This image is created by LadybugPro, the software provided by Point Grey Research Inc.*



(a) Top view (b) Front view

Fig. 4. *Estimated path of the Ladybug™ camera viewed from (a) top and (b) front. The cameras numbered 0, 1, 2, 3, 4 and 5 are indicated as red, green, blue, cyan, magenta and black paths respectively.*

angular error in the direction of the translation direction for the two methods. As may be seen, the L_∞-optimal method seems to work substantially better.

5 Discussion

We have presented a solution to find the motion of cameras that are rigidly mounted and have minimally overlapping fields of view. This method works equally well for any number of cameras, not just two, and will therefore provide more accurate estimates than methods involving only pairs of cameras. The method requires a non-zero frame-to-frame rotation. Probably because of this, the estimation of motion through a long image sequence significantly went off track.

The method geometrically showed good estimation results in experiments with real world data. However, the accumulated errors in processing long sequences of images made the system produce bad estimations over long tracks. In

Translation pair	Scale ratio		Angles	
	True value	Estimated value	True value	Estimated value
$(\mathbf{t}_{01}, \mathbf{t}_{02})$	0.6757	0.7424	28.5°	28.04°
$(\mathbf{t}_{01}, \mathbf{t}_{03})$	0.4386	1.3406	42.5°	84.01°

Table 2. *Experimental results of translation between two key frames are shown in scale ratio of two translation vectors and in angles of that at the two key frames. The translation direction vector* $\mathbf{t}_{0i}$ *is a vector from the centre of the camera at the starting position, frame number 0, to the centre of the camera at the position number* i*. For example,* $\mathbf{t}_{01}$ *is a vector from the centre of the camera at frame 0 to the centre of the camera at frame 30.*

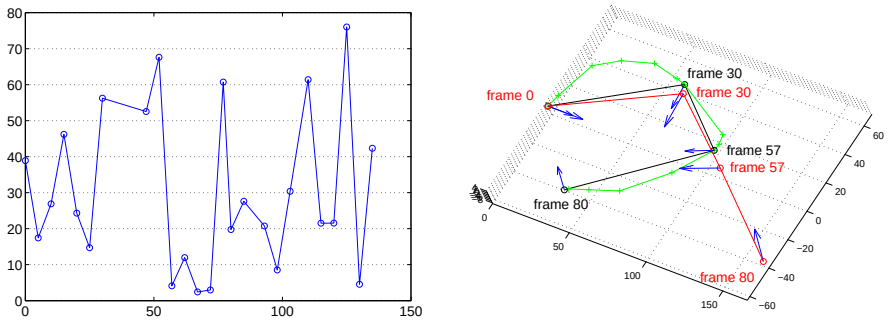
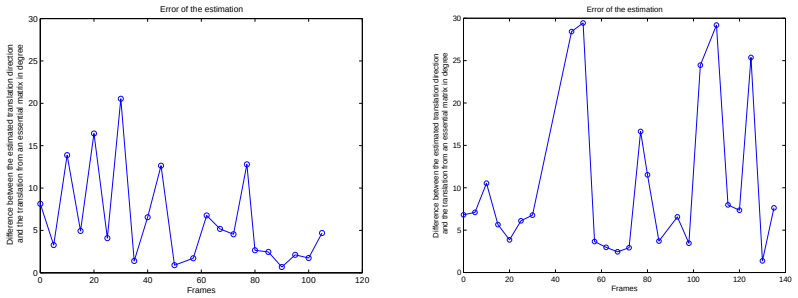


Fig. 5. *The angles between pairs of frames used to estimate the motion are shown in (a). Note that a zero or near-zero rotation means a critical condition for estimating the motion of the cameras from the given frames. (b) Ground truth of positions (indicated as red lines) of the cameras with orientations at key frames 0, 30, 57 and 80, and estimated positions (indicated as black lines) of the cameras with their orientations at the same key frames. Orientations of the cameras are marked as blue arrows. Green lines are the estimated path through all 80 frames.*

the real experiments, we have found that a robust and accurate essential matrix estimation is a critical requirement to obtain correct motion estimation in this problem.

References

1. Hartley, R.I., Zisserman, A.: Multiple View Geometry in Computer Vision. Second edn. Cambridge University Press, ISBN: 0521540518 (2004)
2. Pless, R.: Using many cameras as one. In: CVPR03. (2003) II: 587–593
3. Cheng, Y., Maimone, M., Matthies, L.: Visual odometry on the mars exploration rovers. In: Systems, Man and Cybernetics, 2005 IEEE International Conference on. (2005)
4. Nister, D., Naroditsky, O., Bergen, J.: Visual odometry. Conf. Computer Vision and Pattern Recognition (2004) 652–659
5. Caspi, Y., Irani, M.: Aligning non-overlapping sequences. International Journal of Computer Vision **48**(1) (2002) 39–51



(a) optimal (b) 5-point

Fig. 6. *Comparison of the angular error in the translation direction for two different methods of computing the essential matrix.*

6. Clipp, B., Kim, J.H., Frahm, J.M., Pollefeys, M., Hartley, R.: Robust 6dof motion estimation for non-overlapping, multi-camera systems. Technical Report TR07-006, (Department of Computer Science, The University of North Carolina at Chapel Hill)
7. Hartley, R., Schaffalitzky, F.: L_∞ minimization in geometric reconstruction problems. In: Conf. Computer Vision and Pattern Recognition. Volume I., Washington DC, USA (2004) 504–509
8. Lu, F., Hartley, R.: A fast optimal algorithm for L_2 triangulation. In: Asian Conf. Computer Vision. (2007)
9. Kahl, F.: Multiple view geometry and the L_∞-norm. In: Int. Conf. Computer Vision, Beijing, China (2005) 1002–1009
10. Sturm, J.: Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. Optimization Methods and Software **11–12** (1999) 625–653 Special issue on Interior Point Methods (CD supplement with software).
11. Löberg, J.: Yalmip : A toolbox for modeling and optimization in MATLAB. In: Proceedings of the CACSD Conference, Taipei, Taiwan (2004)
12. Stewénius, H., Engels, C., Nistér, D.: Recent developments on direct relative orientation. ISPRS Journal of Photogrammetry and Remote Sensing **60** (2006) 284–294
13. Li, H., Hartley, R.: Five-point motion estimation made easy. In: ICPR (1), IEEE Computer Society (2006) 630–633
14. Hartley, R., Kahl, F.: Global optimization through searching rotation space and optimal estimation of the essential matrix. In: Int. Conf. Computer Vision. (2007)
15. Point Grey Research Incorporated: Ladybug$^{\mathrm{TM}}$2 camera. http://www.ptgrey.com (2006)
16. Lucas, B., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: IJCAI81. (1981) 674–679
17. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun. ACM **24**(6) (1981) 381–395