EVALUATION OF RELATIVE POSE ESTIMATION METHODS FOR MULTI-CAMERA SETUPS

Volker Rodehorst*, Matthias Heinrichs and Olaf Hellwich

Computer Vision & Remote Sensing, Berlin University of Technology, Franklinstr. 28/29, FR 3-1,
D-10587 Berlin, Germany – (vr, matzeh, hellwich)@cs.tu-berlin.de

KEY WORDS: Camera motion estimation, multi-view geometry, relative orientation, algorithm comparison, performance analysis

ABSTRACT:

The fully automatic and reliable calculation of the relative camera pose from image correspondences is one of the challenging tasks in photogrammetry and computer vision. The problem has not been solved satisfactorily, e.g. in case of critical camera motions or when observing special point configurations. Furthermore, some methods provide multiple solutions for the relative orientation. In this paper we compare various techniques, analyze their difficulties and give results on synthetic and real data. We show that in case of noisy data pose estimation of a single camera remains difficult. The use of multiple calibrated cameras that are fixed on a rig leads to additional constraints, which significantly stabilize the pose estimation process.

1. INTRODUCTION

Camera pose estimation from image correspondences is one of the central tasks in photogrammetry and computer vision. The recovery of the position and orientation of one camera relative to another can be used for binocular stereo or ego-motion estimation. From an algebraic point of view the fundamental matrix describes the projective relation between two uncalibrated views. The essential matrix is important for motion analysis with a calibrated camera, as it contains the rotation and translation up to an unknown scale factor. The automated visual-based motion estimation using thousands of video frames requires an extremely accurate and reliable relative orientation method to handle extensive and intricate camera paths. We would like to point out that without the presence of noise all tested methods perform well. Nevertheless, it is very difficult to obtain stable results for all frames under real conditions with noisy image measurements. Therefore, we propose additional constraints to avoid false estimations. In literature several algorithms for direct relative orientation exist. In (McGlone et al., 2004) Förstner and Wrobel give an overview of various methods, which is updated in Table 1. A detailed description of recent developments can be found in (Stewénius et al., 2006) and degenerate configurations, like

a) coplanar object points and ruled quadric containing the projection centers or
b) orthogonal ruled quadric, especially cylinder containing the projection centers,

are discussed in (Philip, 1998).

Method	Points	Deg.	Solutions
Hartley, 1997	≥ 8	a)	1
Hartley & Zisserman, 2004	≥ 7	a)	≤ 3
Philip, 1996/98	≥ 6	a)	1
Pizarro et al., 2003	≥ 6	b)	≤ 6
Nister, 2004 Stewénius et al., 2006 Li & Hartley, 2006	≥ 5	b)	≤ 10

Table 1: Direct solvers for relative orientation

An interesting aspect of the three minimal 5-point algorithms and the 6-point algorithm of (Pizarro et al., 2003) is their stability, even if points from coplanar objects are observed. This is especially convenient in architectural environment, where many coplanar objects appear. In this paper we compare all direct solvers of Table 1 and a non-linear solver (Batra et al., 2007) to determine their strengths and weaknesses targeting an automatic approach for real world setups. Four implementations are based on the original MATLAB code provided by (Stewénius, 2004). This paper is organized as follows: First, the used methods are shortly introduced, followed by some notes on data conditioning. Section 3 deals with the selection of a unique solution from multiple results. The evaluation of the solver using synthetic data is described in section 4, followed by a discussion of the results. Section 6 introduces additional constraints for multiple cameras. The results using real data are shown in section 7. Finally, a discussion and conclusion of the results closes the paper.

2. RELATIVE POSE RECOVERY

In general, all methods analyze the motion or relative orientation of calibrated cameras using the essential matrix **E.** The main property of **E** is the coplanarity or *epipolar-constraint*

$$\mathbf{u}_i'^{\mathrm{T}}\mathbf{E}\mathbf{u}_i = 0 \tag{1}$$

in terms of the normalized coordinates

$$\mathbf{u}_i = \mathbf{K}^{-1}\mathbf{x}_i \quad \text{and} \quad \mathbf{u}_i' = \mathbf{K}'^{-1}\mathbf{x}_i' \tag{2}$$

of corresponding image points $\mathbf{x}_i \leftrightarrow \mathbf{x}_i'$ with known calibration matrices **K** and **K'**. This linear relation is also known as the Longuet-Higgins equation (Longuet-Higgins, 1981). The essential matrix has additional algebraic properties, e.g. the cubic *rank-constraint*

$$\det(\mathbf{E}) = 0 \tag{3}$$

and the cubic *trace-constraint* (Demazure, 1988)

$$2\mathbf{E}\mathbf{E}^{\mathrm{T}}\mathbf{E} - \mathrm{trace}(\mathbf{E}\mathbf{E}^{\mathrm{T}})\mathbf{E} = \mathbf{0} \tag{4}$$

to ensure that the two non-zero singular values are equal. A complete list of necessary constraints can be found in (Batra et al., 2007). The *linear 8-point* algorithm for the computation of the fundamental matrix (Hartley, 1997) can be used to estimate the essential matrix as well. First, the algorithm uses eight linear equations of (1) with normalized coordinates to estimate **E** and afterwards, the additional constraints (3) and (4) must be enforced. If the singular value decomposition is

$$\mathbf{E} = \mathbf{U} \cdot \mathrm{diag}(\sigma_1, \sigma_2, \sigma_3) \cdot \mathbf{V}^\mathsf{T} \quad \text{for} \quad \sigma_1 \geq \sigma_2 \geq \sigma_3 \tag{5}$$

then the closest essential matrix that minimizes $\left\|\mathbf{E} - \tilde{\mathbf{E}}\right\|$ can be obtained as follows

$$\tilde{\mathbf{E}} = \mathbf{U} \cdot \mathrm{diag}(\sigma, \sigma, 0) \cdot \mathbf{V}^\mathsf{T} \text{ with } \sigma = \frac{\sigma_1 + \sigma_2}{2}. \tag{6}$$

However, the later insertion of the constraints may provide wrong estimations. (Stewénius et al., 2006) and (Šegvić et al., 2007) mention that the 8-point algorithm has a forward bias, which leads to undesired camera motions. The *7-point* solver (Hartley & Zisserman, 2004) uses seven epipolar-constraints (1) on the nine components of the essential matrix. An orthogonal basis for the two-dimensional null-space of these constraints is computed using singular value decomposition. Thus **E** can be written as

$$\mathbf{E} = \alpha_1 \mathbf{E}_1 + \alpha_2 \mathbf{E}_2 \tag{7}$$

where $\mathbf{E}_i$ are the null-vectors for the epipolar-constraints row-wise in matrix form. Since **E** can only be solved up to scale, we are free to set the scalar multiplier $\alpha_2 = 1$. Subsequently, the solution space is reduced using the rank-constraint (3), which gives a third-order polynomial equation in α_1 with three possible solutions for the essential matrix. The *linear 6-point* solver (Philip, 1998) composes the nine third-order polynomial equations from the trace-constraint (4) into a 9×10 matrix and solves for the unknowns linearly. It provides a unique solution but is very sensitive to noise (Stewénius et al., 2006). The *6-point* method of (Pizarro et al., 2003) also composes the nine equations of (4) into a 9×10 matrix from which four rows corresponding to the largest singular values are selected. From these equations, a sixth-degree polynomial is computed with six possible solutions for **E**. The minimal *5-point* algorithms (Nister, 2004), (Stewénius et al., 2006), (Li & Hartley, 2006) need only five point correspondences. In general, the solution in the four-dimensional null-space derived from the epipolar-constraint (1)

$$\mathbf{E} = \sum_{i=1}^{4} \alpha_i \mathbf{E}_i \tag{8}$$

is found using the nine polynomial equations from the trace-constraint (4) and the polynomial equation from the rank-constraint (3). The real-valued zero crossings of this tenth-order polynomial indicate 10 possible solutions for the essential matrix **E**. They can be found using *Sturm sequences* to bracket the roots (Nister, 2004) or an *eigen-decomposition* (Stewénius et al., 2006), which produces slightly better results. The approach of (Li & Hartley, 2006) computes the unknown parameters *simultaneously* instead of back-substituting and solving all the unknowns sequentially. Finally, a *non-linear* solver from five points (Batra et al., 2007) was evaluated. This technique also extracts the four-dimensional null-space (8). To avoid eigen-decompositions, this approach suggests a non-linear optimization technique, e.g. Levenberg-Marquardt. This technique extracts the translation vector $\mathbf{t} = (t_x, t_y, t_z)^\mathsf{T}$ from the essential matrix **E** using singular value decomposition (Wang & Tsui, 2000)

$$\mathbf{t}^\mathsf{T} \mathbf{E} = \mathbf{0}. \tag{9}$$

Note, that **t** is related to the second projection center **C**' with

$$\mathbf{t} = -\mathbf{R}\mathbf{C}'. \tag{10}$$

The translation vector is used to parameterize a cost function, which enforces necessary constraints for the essential matrix. The state vector

$$\mathbf{b} = \left(\alpha_1, \alpha_2, \alpha_3, \alpha_4, t_x, t_y, t_z\right)^\mathsf{T} \tag{11}$$

consists of the scalars α_i defining the solution within the four-dimensional null-space and the three translation components. The cost function can be derived from the equation

$$\mathbf{E}\mathbf{E}^\mathsf{T} - [\mathbf{t}]_\times [\mathbf{t}]_\times^\mathsf{T} = \mathbf{0}, \tag{12}$$

where $[]_\times$ denotes the skew-symmetric matrix of vector **t**. The nine elements of **E** depending on seven elements of **b** are stored in a 7×9 matrix **A**. Overall, nine of those matrices can be formed, three from equation (9) and six from (12). The non-linear minimization task

$$\min_{\mathbf{b}} \sum_{i=1}^{9} \left\|\mathbf{b}^\mathsf{T} \mathbf{A}_i \mathbf{b}\right\|^2 \quad \text{with} \quad \|\mathbf{b}\| = 1 \tag{13}$$

starts with random values for α_i. Since there are up to 10 possible solutions and the null-space is generally non convex, this optimization should be iterated several times. For real-time applications this may not be suitable, but the technique can be used to improve the results obtained by direct solvers, which provide good approximation values.

Hartley proved (Hartley, 1997) that the linear 8-point algorithm performs significantly better, if the input data is conditioned. This insight should be still important for minimal solvers (Li & Hartley, 2006). The normalization is done by translating the centroid of the measured image points to the origin and scaling them to a mean Euclidean distance of $\sqrt{2}$, which can be combined into a similarity transformation **T** for the first and **T'** for the second image. Note, that the resulting relative orientation must be deconditioned before the essential constraints are enforced. In case of the minimum solvers or the non-linear algorithm, the four resulting null-vectors $\mathbf{E}_i$ must be deconditioned

$$\tilde{\mathbf{E}}_i = \mathbf{T}'^\mathsf{T} \mathbf{E}_i \mathbf{T} \tag{14}$$

before the root searching step (Šegvić et al., 2007). One might think that the data could be deconditioned as a final step, but this leads to false solutions as shown in (Hartley, 1997).

3. CHOOSING THE RIGHT SOLUTION

Most direct solvers provide multiple solutions for the relative orientation, except the linear solvers (Hartley, 1997) and (Philip, 1998). Up to 10 distinct physically valid solutions are possible (see Table 1). Although, in most cases the number of solutions varies between one and four, the correct solution is difficult to identify. Since the 7-point solver (Hartley & Zisserman, 2004) doesn't enforce the trace-constraint (4), the similarity of the two non-zero singular values may indicate the solution. Furthermore, the 6-point method of (Pizarro et al., 2003) doesn't employ the rank-constraint (4), so that the smallest determinant value maybe analyzed.

If no additional assumptions can be made, a possible criterion for choosing the right solution are the number of points, which lie in front of both cameras. A method to recover the twisted pair ambiguity and extract the projection matrices from **E** is described in (Nister, 2004; Hartley & Zisserman, 2004). Then, spatial object points are triangulated and their *cheirality* can be tested. Since several solutions may have all points in front of both cameras, this criterion is not sufficient. Furthermore, the cheirality-condition suffers from three disadvantages.

First, if there is no camera translation, the points can not be triangulated. To overcome this issue, a threshold t_{mov} for detecting enough motion (Weng et al., 1989) can be introduced:

$$\mathbf{P} = \left[\mathbf{I}|0\right], \quad \mathbf{P}' = \left[\mathbf{R}|\mathbf{t}\right], \quad \frac{\left\|\mathbf{u}' \times \mathbf{R}\mathbf{u}\right\|}{\left\|\mathbf{u}\right\| \cdot \left\|\mathbf{u}'\right\|} < t_{mov} \tag{15}$$

Second, the triangulation of object points can only be performed with a sufficient baseline to depth ratio. For example, noisy image points may cause the triangulated object point to flip behind the camera. This can be avoided, by testing points with a certain proximity to the camera. Third, if many points are used for a cheirality test, the triangulation is computationally intensive. If more than five correspondences are available, the additional information should be used to find the right solution. We compute the first-order geometric error (*Sampson-distance*) for all point correspondences $\mathbf{u}_i \leftrightarrow \mathbf{u}_i'$

$$d_{error} = \sum_i \frac{\left(\mathbf{u}_i' \mathbf{E} \mathbf{u}_i\right)^2}{\left(\mathbf{E}\mathbf{u}_i\right)_x^2 + \left(\mathbf{E}\mathbf{u}_i\right)_y^2 + \left(\mathbf{E}^\mathsf{T}\mathbf{u}_i'\right)_x^2 + \left(\mathbf{E}^\mathsf{T}\mathbf{u}_i'\right)_y^2} \tag{16}$$

where $(\,)_x^2$ represents the square of the vectors *x*-component. Finally, d_{error} should be minimal for the correct solution.

In our approach we used a combination of these criteria: First the translation is examined according to (15) and then the five points are tested to lie in front of both cameras. If there are multiple solutions with all points in front, the epipolar distance of all available correspondences is evaluated.

4. EVALUATION OF THE ALGORITHMS

Subsequently, we analyze all techniques with respect to their behavior under Gaussian noise, the selection strategy for multiple solutions, over-determined estimation and data conditioning. The evaluation was performed using synthetically generated data with ground truth. The camera motion between two views is randomly chosen from a uniform distribution. To generate the random numbers, we use the advanced mersenne twister (Matsumoto & Nishimura, 1998).

The camera translation is scaled to 1 and the three rotation angles are constrained between -45 and 45 degrees. Then, 100 spatial object points are randomly generated in general position and projected into the images using the simulated cameras. If the known calibration matrices are applied (2), the normalized coordinates range from -1 to 1. The image coordinates are displaced with Gaussian noise. The standard deviation σ corresponds to an image with 1024×1024 pixels and the maximum Euclidean displacement is 2.4σ.

We ensure that the selected point correspondences are not collinear and avoid degenerate configurations of minimal sets with the constraints proposed by (Werner, 2003). For allowed configurations, the epipoles must lie in domains with piecewise-conic boundaries. An example for estimating the relative orientation is shown in Figure 1. To obtain statistically significant results, every technique is examined 100 times. The evaluation of the selection criteria is done with the 5-point algorithm of (Nistér, 2004). Here, additional 100 random 5-tuples are selected in each dataset and the best sample is taken as result. The experiments for over-determined computations are performed with the whole dataset of 100 points. Finally, the impact of data conditioning is evaluated for the 5- and 8-point algorithms with 100 random *n*-tuples in each dataset. The best solution according to the cheirality test and Sampson distance is selected. The deviation error of translation and rotation are measured in degrees. The included angle between the original and estimated translation direction gives an interpretable result. For rotation evaluation three unit vectors to the three axis directions $\mathbf{e}_x$, $\mathbf{e}_y$ and $\mathbf{e}_z$ are rotated using the original and the estimated rotation matrix. The error value is averaged over the three including angles of the resulting vectors:

$$r_{error} = \frac{1}{3} \sum_{i \in x,y,z} \operatorname{acos}\left(\left(\mathbf{R}_1 \mathbf{e}_i\right)^\mathsf{T} \mathbf{R}_2 \mathbf{e}_i\right) \tag{17}$$

We count all translations with an error less than 10 degrees and all rotations with an error less than 2 degrees (see Table 2).

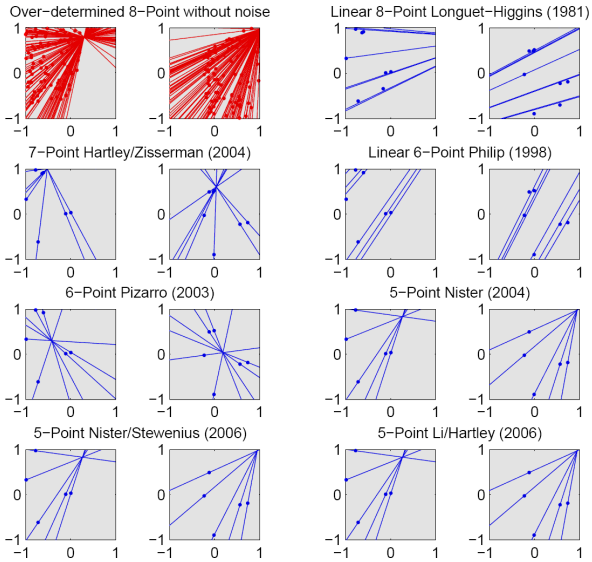

Figure 1: Estimated relative orientation using normalized image pairs with overlaid epipolar rays (correct reference solution in red).

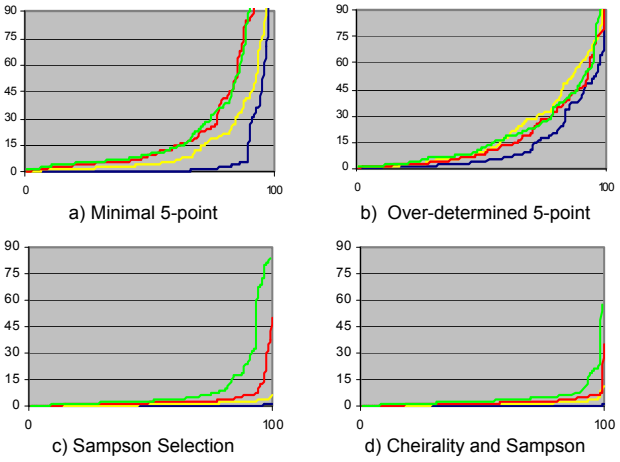

Figure 2: Translation error in degrees of the direct 5-point solver (Nistér, 2004) for 100 runs. blue: σ = 0.07, yellow: σ = 0.5, red: σ = 0.9, green: σ = 1.3.

Rotation Errors								
Method	σ = 0.07		σ = 0.5		σ = 0.9		σ = 1.3	
	cnt	mean	cnt	mean	cnt	mean	cnt	mean
Ground Truth	100	0.0860	100	0.7475	100	1.0872	100	2.0434
Evaluation of Different Algorithms								
8-Point	90	2.2589	52	5.0803	32	5.1768	18	6.3161
7-Point	82	1.7805	42	4.0002	33	5.2407	24	6.3662
6-Linear	73	2.4829	38	4.4165	29	4.6610	16	5.7286
6-Pizzaro	78	1.0380	49	4.1056	39	4.5695	24	4.7601
5-Nistér	89	0.9935	67	3.1534	52	4.5321	50	5.4091
5-Stewénius	89	0.9935	67	3.1534	52	4.5321	50	5.4091
5-Li	89	0.9935	67	3.1534	52	4.5321	36	5.6712
5-Non-linear	30	0.7699	24	1.0879	32	1.5902	32	3.8587
Selection Strategy for Multiple Solutions								
Sampson S	100	0.2826	100	1.4059	94	2.3680	81	3.1766
Cheirality C	38	2.3562	36	4.1240	19	3.9483	15	4.9022
C-5 + S	100	0.2610	99	1.5168	99	2.2784	91	3.1963
C-all + S	100	0.2420	100	1.2317	96	2.2333	88	2.7443
Over-determined Solution								
8-Point	99	0.3966	96	1.8661	81	3.1579	62	4.5419
5-Nistér	70	2.9331	51	3.5674	52	3.4531	48	4.3646
Data Conditioning								
8-Uncond	99	0.3197	94	2.0592	83	3.3274	61	4.1966
8-Cond	98	0.3433	93	2.0068	84	3.0240	64	3.6310
5-Uncond	100	0.2692	98	1.2245	95	2.2597	90	2.9573
5-Cond	100	0.2314	100	1.3404	97	2.1464	92	2.8859

Translation Errors								
Method	σ = 0.07		σ = 0.5		σ = 0.9		σ = 1.3	
	cnt	mean	cnt	mean	cnt	mean	cnt	mean
Ground Truth	100	0.0003	100	0.0030	100	0.0048	100	0.0086
Evaluation of Different Algorithms								
8-Point	95	0.3361	67	0.7335	57	0.9643	32	1.0265
7-Point	85	0.2191	69	0.6678	55	0.8858	45	0.9150
6-Linear	81	0.3507	51	0.7330	44	0.8677	33	1.0573
6-Pizzaro	79	0.1871	57	0.7736	44	0.7868	32	1.0351
5-Nistér *	88	0.1346	77	0.6184	64	0.7910	54	0.9737
5-Stewénius	88	0.1346	77	0.6184	64	0.7910	54	0.9737
5-Li	88	0.1346	77	0.6184	64	0.7910	38	1.0040
5-Non-linear	40	0.4367	31	0.4205	37	0.3403	37	0.4715
Selection Strategy for Multiple Solutions								
Sampson S *	100	0.0364	100	0.2069	95	0.3381	87	0.5475
Cheirality C	38	0.6003	40	0.7801	25	0.8582	21	1.1246
C-5 + S *	100	0.0350	100	0.2140	99	0.3184	96	0.4879
C-all + S	100	0.0280	100	0.1619	96	0.2800	90	0.4413
Over-determined Solution								
8-Point	97	0.0433	96	0.2529	89	0.4730	69	0.6653
5-Nistér *	84	0.5310	70	0.6350	73	0.5765	76	0.7148
Data Conditioning								
8-Uncond	94	0.0429	95	0.2776	92	0.4581	73	0.6194
8-Cond	98	0.0460	95	0.2817	86	0.3988	73	0.6678
5-Uncond	100	0.0354	99	0.1724	98	0.3231	95	0.5198
5-Cond	100	0.0292	100	0.1621	100	0.3037	94	0.4303

Table 2: Percentage of correct solutions and mean errors. (* indicates method in Figure 2)

5. DISCUSSION OF THE SIMULATION RESULTS

In general, all algorithms show one common effect: The estimation of camera rotation is much more stable and accurate, than the estimate of camera translation. To show the influence of noise, the camera pose is calculated from the object points and noisy image points via spatial resection and compared with the ground truth. Surprisingly, all direct 5-point solvers produce exactly the same results up to noise of $\sigma = 0.9$. The methods (Nistér, 2004) and (Stewenius et al., 2006) produce exactly the same results in all tests. The supposable higher accuracy of the second algorithm can not be verified by evaluating the first five significant digits.

The method of (Li & Hartley, 2006) has problems with large noise and can not reach the quality of the other two 5-point techniques at $\sigma = 1.3$. Opposed to (Batra et al., 2007), the numerical instability of the eigen-decomposition is not limiting the five-point algorithm. Even worse, the non-linear 5-point technique suffers from finding the solution and a perfect starting value does not ensure convergence of the non-linear optimization technique. This can be seen by the low number of correct solutions, even for small amount of noise. Nevertheless, if a solution was found, it is highly accurate.

Both 6-point algorithms produce results not as good as the 5-point techniques, especially if noise increases. The method of (Pizzaro, 2003) is a bit more reliable, than the linear one of (Philip, 1998). The 7- and 8-point algorithms show a similar behavior. If noise increases to realistic amounts, the results become even worse than the 6-point algorithms. In general, the 5-point algorithms outperform every other technique.

The cheirality test alone is not sufficient to select a good solution, because many estimates of essential matrices have all points in front of both cameras. The Sampson distance of additional points is a better criterion, but with increasing noise it has a higher probability to select a wrong solution. Combining the two criterions can be done in two ways: First, the cheirality is tested only for the 5 points used for the computation. In this case for every set of solutions at least one has all points in front, but some solutions can be ignored. This makes the selection very robust and needs only moderate computation time. Second, the cheirality test can be performed over all available points. This lead to a bit more accurate results, but for larger noise the number of acceptable solutions decreases. In addition, the triangulation of all points is computationally intensive. The best trade off between robustness, accuracy and computational effort is to compare the Sampson distances of all points, which have the 5 points in front of both cameras.

We also investigated weather the algorithms can improve the accuracy of the essential matrix, if more than the minimal number of points is used. Surprisingly, both over-determined 5- and 8-point algorithms decrease in accuracy.

The comparison of results with and without data conditioning shows that an additional conditioning of the already normalized coordinates is not necessary for the computation of the essential matrix. The average values are so close to each other, that almost no influence can be measured.

6. MULTIPLE-CAMERA POSE ESTIMATION

As mentioned before, the main drawback of the 5-point algorithms are the 10 possible solutions. Selecting and detecting the right solution is not trivial, especially in the presence of noise. Since cheirality tests and epipolar distance filtering are instable, additional constraints must be imposed.

The situation becomes easier, if two or more cameras move in a fixed relation to each other, e.g. if they are mounted on the same vehicle. Their motion is not independent of each other (see Figure 3). This fixed relationship can be used to select the right solution pair of the two solution sets.

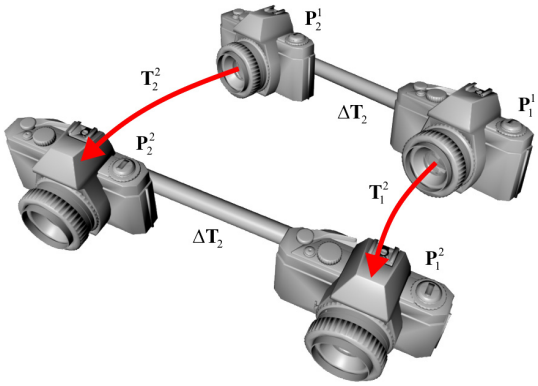

Figure 3: Constrained multiple-camera motion

We denote a camera i at time j with $\mathbf{P}_i^j$ and assume that the reference camera $\mathbf{P}_1^1$ is located at origin. As shown in Figure 8, the projection matrices $\mathbf{P}_1^1$ and $\mathbf{P}_2^1$ have a fixed relative orientation defined by $\Delta\mathbf{T}_2$. If the cameras move to the positions of $\mathbf{P}_1^2$ and $\mathbf{P}_2^2$, the essential matrices for both motions $\mathbf{T}_1^2$ and $\mathbf{T}_2^2$ are determined. Inspired by (Esquivel et al., 2007), the following constraint is imposed:

$$\mathbf{T}_1^2\Delta\mathbf{T}_2 = \Delta\mathbf{T}_2\mathbf{T}_2^2 \tag{18}$$

Each transformation can be separated into a translation vector $\mathbf{C}$ and a rotation matrix $\mathbf{R}$. Therefore, a correct pair of solutions must also fulfill:

$$\mathbf{R}_1^2\Delta\mathbf{R}_2 = \Delta\mathbf{R}_2\mathbf{R}_2^2 \tag{19}$$

Unfortunately, the essential matrix contains the translational component only up to scale and therefore, the ratio of the motion lengths $\mathbf{C}_1^j$ and $\mathbf{C}_2^j$ is unknown:

$$\mathbf{R}_1^2\Delta\mathbf{C}_2 + \mu\mathbf{C}_1^2 = \lambda\Delta\mathbf{R}_2\mathbf{C}_2^2 + \Delta\mathbf{C}_2 \tag{20}$$

To estimate the relative scale factors μ and λ, the linear equation system

$$\begin{bmatrix}\mathbf{C}_1^2 & -\Delta\mathbf{R}_2\mathbf{C}_2^2\end{bmatrix}\begin{pmatrix}\mu\\ \lambda\end{pmatrix} = \left(\Delta\mathbf{C}_2 - \mathbf{R}_1^2\Delta\mathbf{C}_2\right), \tag{21}$$

of the form $\mathbf{A}\,\mathbf{x} = \mathbf{b}$ can be used to solve for the unknown in $\mathbf{x}$. An extension for multiple cameras is straight forward:

$$\begin{bmatrix}\mathbf{C}_1^2 & -\Delta\mathbf{R}_2\mathbf{C}_2^2 & \mathbf{0}\\ \mathbf{C}_1^2 & \mathbf{0} & -\Delta\mathbf{R}_3\mathbf{C}_3^2\end{bmatrix}\begin{pmatrix}\mu\\ \lambda\\ \kappa\end{pmatrix} = \begin{pmatrix}\Delta\mathbf{C}_2 - \mathbf{R}_1^2\Delta\mathbf{C}_2\\ \Delta\mathbf{C}_3 - \mathbf{R}_1^2\Delta\mathbf{C}_3\end{pmatrix} \tag{22}$$

The missing constraint between $\mathbf{P}_2^2$ and $\mathbf{P}_3^2$ requires one camera as reference at origin. Nevertheless, we recommend this extra coordinate transformation, because the third constraint improves significantly the estimation of the translation factors.
The equation systems (21) and (22) are over-determined, since every camera introduces one unknown scale factor and each pair of cameras introduces three constraints. Therefore, the residual errors

$$\mathbf{r} = \mathbf{A}\mathbf{x} - \mathbf{b} \tag{23}$$

are used as quality measure for the pose estimation. We combined the conditions (19) and (21) into a cost function, which is computed for every possible pair of solutions.

As the rotation is much more stable than the translation (see section 4 and 5), some weighting factors w_{trans} and w_{rot} should be used to balance the cost function, e.g. $w_{trans} = 5w_{rot}$.

$$e_i^j = w_{trans}\cdot\|\mathbf{r}\| + w_{rot}\cdot r_{error} \tag{24}$$

If temporally tracked points in the camera pairs $(\mathbf{P}_1^1, \mathbf{P}_1^2)$ and $(\mathbf{P}_2^1, \mathbf{P}_2^2)$ are additionally matched between the cameras, the missing scale factors μ and λ can be computed according to the fixed camera pose $\Delta\mathbf{T}_2$. The triangulation of object points $\mathbf{X}$ using the spatial pair $(\mathbf{P}_1^1, \mathbf{P}_2^1)$ defines a reference scale.
Now the triangulated points $\mathbf{X}^1$ and $\mathbf{X}^2$ derived from the temporal pairs $(\mathbf{P}_1^1, \mathbf{P}_1^2)$ and $(\mathbf{P}_2^1, \mathbf{P}_2^2)$ respectively can be scaled to the reference points $\mathbf{X}$. The distance between the 3D-coordinates and the projection centers gives the relation of the scale factors used in $\Delta\mathbf{T}_2$, $\mathbf{T}_1^2$ and $\mathbf{T}_2^2$. In case of inappropriate motion the triangulation of temporally tracked points is less accurate than spatially matched points on the camera rig. Therefore, the average of the nearest five object points for each triangulation pair is used to compute the scale factors μ and λ:

$$\mu = 0.2\cdot\sum_{i=1}^{5}\frac{\|\mathbf{X}_i\|}{\|\mathbf{X}_i^1\|},\quad \lambda = 0.2\cdot\sum_{i=1}^{5}\frac{\|\mathbf{X}_i\|}{\|\mathbf{X}_i^2\|} \tag{25}$$

7. EXPERIMENTS WITH MULTIPLE CAMERAS

The multiple-camera setup is tested on a real data sequence. The sequence consists of three independent camera streams, which are mounted on a calibrated rig. Every frame has at least 100 tracked features. The essential matrices are computed with a RANSAC technique using the minimal 5-point solver. To ensure a certain motion of the cameras, frames with an average tracking disparity below 10 pixels are omitted.
The camera paths are reconstructed fully automatically. To compare the robustness of the proposed multi-camera technique, paths of the single track of the reference camera and the linked track are shown in Figures 4 and 5. The path in Figure 4 suffers from a miscalculation in the middle of the track, which results in orthogonal camera placement. After this discontinuity the scale is wrong, because the scale factor is calculated on the last camera pair. In Figure 5 the path is smooth and correctly scaled. The gaps in the path indicate skipped images with insufficient camera motion.
The whole multi-camera path consists of 900 frames. The reference position was manually set and the extracted path is shown as a red line in Figure 6. Please note, that the camera path has not been optimized by bundle adjustment or any further reference points.

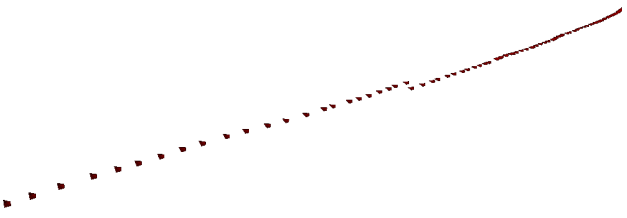

Figure 4: Reconstructed single-camera path over 110 frames

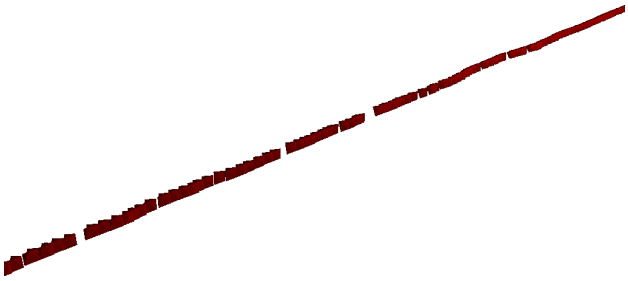

Figure 5: Reconstructed multi-camera path over 110 frames

Figure 6: Reconstructed multi-camera path over 900 frames.

8. CONCLUSIONS

In this paper we have shown that robust camera pose estimation on real data is still a difficult task. In general, the estimation of camera rotation is more reliable than the translation. The minimal 5-point solvers produce better results than all other methods, especially in presence of noise. In case of multiple solutions, the best selection criterion is a combination of a preceding cheirality test with minimal points followed by the computation of the Sampson distance over all available points.
Furthermore, using over-determined variants of the minimal solver not necessarily increase the accuracy of the essential matrix. Instead, the result should geometrically be improved with standard techniques like bundle adjustment.
Finally, data conditioning for the computation of the essential matrix is not necessary. However, the essential matrix computation produces wrong estimates from time to time. If a camera path over several hundred frames needs to be reconstructed, one miscalculation corrupts the whole path. We have shown that additional multi-camera constraints can be imposed to gives stable results for extensive camera path reconstructions.

ACKNOWLEDGEMENTS

This work was partially supported by grants from the German Research Foundation DFG.

REFERENCES

Batra, D., Nabbe, B. and Hebert, M., 2007. An alternative formulation for five point relative pose problem, *IEEE Workshop on Motion and Video Computing* WMVC '07, 6 p.

Demazure, M., 1988. Sur deux problemes de reconstruction, Technical Report No 882, INRIA, Rocquencourt, France.

Esquivel, S., Woelk, F., Koch, R., 2007. Calibration of a multi-camera rig from non-overlapping views, *DAGM Symposium*, LNCS 4713, Heidelberg, pp. 82-91.

Hartley, R., 1997. In defense of the eight-point algorithm, *IEEE Trans. on Pattern Analysis and Machine Intelligence,* vol. 19, no. 6, pp. 580-593.

Hartley, R. and Zisserman, A., 2004. Multiple view geometry in computer vision, Cambridge University Press, 2. edition, 672 p.

Li, H.D. and Hartley, R.I., 2006. Five-point motion estimation made easy, *Int. Conf. on Pattern Recognition*, vol. 1, pp. 630-633.

Longuet-Higgins, H.C., 1981. A computer algorithm for reconstructing a scene from two projections, *Nature*, vol. 293, pp. 133–135.

Matsumoto, M. and Nishimura, T., 1998. Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator, *ACM Transactions on Modeling and Computer Simulation*, vol. 8, no. 1, pp 3-30.

McGlone, J.C., Mikhail, E.M., Bethel, J., Mullen, R. (Eds.), 2004. Manual of Photogrammetry, 5th edition, American Society of Photogrammetry and Remote Sensing.

Nister, D., 2004. An efficient solution to the five-point relative pose problem, *IEEE Conf. on Computer Vision and Pattern Recognition*, vol. 2, pp. 195-202.

Philip, J., 1996. A non-iterative algorithm for determining all essential matrices corresponding to five point pairs, *Photogrammetric Record*, vol. 15(88), pp. 689-599.

Philip, J., 1998. Critical point configurations of the 5-, 6-, 7-, and 8-point algorithms for relative orientation, Technical Report TRITA-MAT-1998-MA-13, Dept. of Mathematics, Royal Inst. of Tech., Stockholm.

Pizarro, O., Eustice, R., Singh, H., 2003. Relative pose estimation for instrumented, calibrated platforms, 7th Digital Image Computing: Techniques and Applications.

Šegvić, S., Schweighofer, G. and Pinz A., 2007. Influence of numerical conditioning on the accuracy of relative orientation, *IEEE Conf. on Computer Vision and Pattern Recognition, 2007*, 8 p.

Stewénius, H., 2004. Matlab code for solving the fivepoint problem, *http://www.maths.lth.se/~stewe/FIVEPOINT/*, (accessed 18. Nov. 2007).

Stewénius, H., Engels, C. and Nistér, D., 2006. Recent developments on direct relative orientation, *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 60, no. 4, June 2006, pp. 284-294.

Wang, W. and Tsui, H., 2000. An SVD decomposition of the essential matrix with eight solutions for the relative positions of two perspective cameras, *Int. Conf. on Pattern Recognition,* vol. 1, pp. 362-365.

Weng, J.Y., Huang, T.S. and Ahuja, N., 1989. Motion and structure from two perspective views: algorithms, error analysis, and error estimation, *Trans. on Pattern Analysis and Machine Intelligence*, vol. 11, no. 5, pp. 451-476.

Werner, T., 2003. Constraint on five points in two images, *IEEE Conf. on Computer Vision and Pattern Recognition*, vol. II, pp. 203-208.