

# EXPERIMENTAL EVALUATION OF RELATIVE POSE ESTIMATION ALGORITHMS

Marcel Brückner, Ferid Bajramovic, Joachim Denzler

*Chair for Computer Vision, Friedrich-Schiller-University Jena, Ernst-Abbe-Platz 2, 07743 Jena, Germany  
{brueckner,bajramov,denzler}@informatik.uni-jena.de; http://www4.informatik.uni-jena.de*

Keywords: relative pose, epipolar geometry, camera calibration

Abstract: We give an extensive experimental comparison of four popular relative pose (epipolar geometry) estimation algorithms: the eight, seven, six and five point algorithms. We focus on the practically important case that only a single solution may be returned by automatically selecting one of the solution candidates, and investigate the choice of error measure for the selection. We show that the five point algorithm gives very good results with automatic selection. As sometimes the eight point algorithm is better, we propose a combination algorithm which selects from the solutions of both algorithms and thus combines their strengths. We further investigate the behavior in the presence of outliers by using adaptive RANSAC, and give practical recommendations for the choice of the RANSAC parameters. Finally, we verify the simulation results on real data.

## 1 INTRODUCTION

Solving the relative pose problem is an important prerequisite for many computer vision and photogrammetry tasks, like stereo vision. It consists of estimating the relative position and orientation of two cameras from inter-image point correspondences, and is closely related to the epipolar geometry. It is generally agreed, that bundle adjustment gives the best solution to the problem (Triggs et al., 1999), but needs a good initial solution for its local optimization.

In this paper, we review and experimentally compare four non-local algorithms for estimating the essential matrix and thus relative pose, which can be used to initialize bundle adjustment: variants of the eight point and seven point algorithms (Hartley and Zisserman, 2003), which directly estimate the essential matrix, as well as a simple six point algorithm and the five point algorithm, which has been proposed recently (Stewénius et al., 2006).

In contrast to the experiments presented there, we add an automatic selection of the best of the multiple solutions computed by the five and seven point algorithms, as it seems practically more relevant to get exactly one solution. We also analyse the choice of the epipolar error measure required by the selection step.

As there is no single best algorithm, we propose the improvement of combining the best two algorithms followed by a selection step. To the best of our knowledge, this is a novel contribution.

Practically, point correspondences, which have been automatically extracted from images, always contain false matches. Estimating relative pose from such data requires a robust algorithm. The RANSAC scheme (Fischler and Bolles, 1981) gives robust variants of the algorithms mentioned above. In this paper, we analyse the optimal choice of the error measure, the threshold and the sample size for RANSAC, and give practical recommendations. We also investigate the improvement gained by our combination algorithm. Finally, we present results on real data.

The paper is structured as follows: In section 2, we give a repetition of important theoretical basics followed by a description of the algorithms in section 3. We present our experiments in section 4 and give conclusions in section 5.

## 2 THEORY

In this section, we introduce the camera model and some notation and give a short repetition

of theoretical basics of the relative pose problem. For further details, the reader is referred to (Hartley and Zisserman, 2003).

## 2.1 Camera Model

The pinhole camera model is expressed by the equation  $\mathbf{p} \cong \mathbf{K}\mathbf{p}_C$ , where  $\mathbf{p}_C$  is a 3D point in the camera coordinate system,  $\mathbf{p} = (p_x, p_y, 1)^T$  is the imaged point in homogeneous 2D pixel coordinates,  $\cong$  denotes equality up to scale and  $\mathbf{K} \stackrel{\text{def}}{=} ((f_x, s, o_x), (0, f_y, o_y), (0, 0, 1))^T$  is the camera calibration matrix, where  $f_x$  and  $f_y$  are the effective focal lengths,  $s$  is the skew parameter and  $(o_x, o_y)$  is the principal point. The relation between a 3D point in camera coordinates  $\mathbf{p}_C$  and the same point expressed in world coordinates  $\mathbf{p}_W$  is  $\mathbf{p}_C = \mathbf{R}\mathbf{p}_W + \mathbf{t}$ , where  $\mathbf{R}$  is the orientation of the camera and  $\mathbf{t}$  defines the position of its optical center. Thus,  $\mathbf{p}_W$  is mapped to the image point  $\mathbf{p}$  by the equation  $\mathbf{p} \cong \mathbf{K}(\mathbf{R}\mathbf{p}_W + \mathbf{t})$ . We will denote a pinhole camera by the tuple  $(\mathbf{K}, \mathbf{R}, \mathbf{t})$ .

## 2.2 Relative Pose

The relative pose  $(\mathbf{R}, \mathbf{t})$  of two cameras  $(\mathbf{K}, \mathbf{I}, 0)$ ,  $(\mathbf{K}', \mathbf{R}, \mathbf{t})$  is directly related to the essential matrix  $\mathbf{E}$ :

$$\mathbf{E} \stackrel{\text{def}}{\cong} [\mathbf{t}]_{\times} \mathbf{R} , \quad (1)$$

where  $[\mathbf{t}]_{\times}$  denotes the skew symmetric matrix associated with  $\mathbf{t}$ . The relative pose  $(\mathbf{R}, \mathbf{t})$  can be recovered from  $\mathbf{E}$  up to the scale of  $\mathbf{t}$  and a four-fold ambiguity, which can be resolved by the cheirality constraint. The translation  $\mathbf{t}$  spans the left nullspace of  $\mathbf{E}$  and can be computed via singular value decomposition (SVD).

The essential matrix is closely related to the fundamental matrix which can be defined as:

$$\mathbf{F} \stackrel{\text{def}}{\cong} \mathbf{K}^{-T} \mathbf{E} \mathbf{K}'^{-1} . \quad (2)$$

The matrices  $\mathbf{F}$  and  $\mathbf{E}$  fulfill the following properties:

$$\mathbf{p}^T \mathbf{F} \mathbf{p}' = 0 \quad (3)$$

$$\hat{\mathbf{p}}^T \mathbf{E} \hat{\mathbf{p}}' = 0 , \quad (4)$$

where  $\mathbf{p}$  and  $\mathbf{p}'$  are corresponding points in the two cameras (i.e. images of the same 3D point), and  $\hat{\mathbf{p}} = \mathbf{K}^{-1} \mathbf{p}$  denotes camera normalized coordinates. Furthermore, both matrices are singular:

$$\det(\mathbf{F}) = 0 \quad \text{and} \quad \det(\mathbf{E}) = 0 . \quad (5)$$

The essential matrix has the following additional property (Nistér, 2004), which is closely related to the fact, that its two non-zero singular values are equal:

$$\mathbf{E} \mathbf{E}^T \mathbf{E} - \frac{1}{2} \text{trace}(\mathbf{E} \mathbf{E}^T) \mathbf{E} = 0 . \quad (6)$$

# 3 ALGORITHMS

## 3.1 Eight Point Algorithm

The well known eight point algorithm estimates  $\mathbf{F}$  from at least eight point correspondences based on equation (3). According to equation (2),  $\mathbf{E}$  can be computed from  $\mathbf{F}$ . As equation (4) has the same structure as (3), the (identical) eight point algorithm can also be used to directly estimate  $\mathbf{E}$  from camera normalized point correspondences  $(\hat{\mathbf{p}}, \hat{\mathbf{p}}')$ .

Equation (4) can be written as  $\tilde{\mathbf{a}}^T \tilde{\mathbf{e}} = 0$ , with

$$\tilde{\mathbf{a}} \stackrel{\text{def}}{=} (\hat{p}'_1 \hat{p}_1, \hat{p}'_2 \hat{p}_1, \hat{p}'_3 \hat{p}_1, \hat{p}'_1 \hat{p}_2, \hat{p}'_2 \hat{p}_2, \hat{p}'_3 \hat{p}_2, \hat{p}'_1 \hat{p}_3, \hat{p}'_2 \hat{p}_3, \hat{p}'_3 \hat{p}_3)^T \quad (7)$$

$$\tilde{\mathbf{e}} \stackrel{\text{def}}{=} (E_{11}, E_{12}, E_{13}, E_{21}, E_{22}, E_{23}, E_{31}, E_{32}, E_{33})^T . \quad (8)$$

Given  $n \geq 8$  camera normalized point correspondences, the according vectors  $\tilde{\mathbf{a}}_i^T$  can be stacked into an  $n \times 9$  data matrix  $\mathbf{A}$  with  $\mathbf{A} \tilde{\mathbf{e}} = 0$ . For  $n = 8$ ,  $\mathbf{A}$  has rank defect 1 and  $\tilde{\mathbf{e}}$  is in its right nullspace. Let  $\mathbf{A} = \mathbf{U} \text{diag}(s) \mathbf{V}^T$  be the singular value decomposition (SVD) of  $\mathbf{A}$ . Throughout the paper, the singular values are assumed in decreasing order in  $s$ . Then  $\tilde{\mathbf{e}}$  is the last column of  $\mathbf{V}$ . For  $n > 8$ , this gives the least squares approximation with  $\|\tilde{\mathbf{e}}\| = 1$ .

## 3.2 Seven Point Algorithm

The seven point algorithm is very similar to the eight point algorithm, but additionally uses and enforces equation (5). It thus needs only seven point correspondences. As in the eight point algorithm, the SVD of the data matrix  $\mathbf{A}$  is computed. For  $n = 7$ ,  $\mathbf{A}$  has rank defect 2, and  $\tilde{\mathbf{e}}$  is in its two dimensional right nullspace, which is spanned by the last two columns of  $\mathbf{V}$ . These two vectors are transformed back into the matrices  $\mathbf{Z}$  and  $\mathbf{W}$  according to equation (8). We get:

$$\mathbf{E} = z \mathbf{Z} + w \mathbf{W} , \quad (9)$$

where  $z, w$  are unknown real values. Given the arbitrary scale of  $\mathbf{E}$ , we can set  $w = 1$ . To compute  $z$ , substitute equation (9) into equation (5). This results in a third degree polynomial in  $z$ . Each of the up to three real roots gives a solution for  $\mathbf{E}$ . We use the companion matrix method to compute the roots (Cox et al., 2005). In case of  $n > 7$ , the algorithm is identical. The computation of the nullspace is a least squares approximation.

## 3.3 Six Point Algorithm

There are various six point algorithms (Philip, 1996; Pizarro et al., 2003). Here, we present a simple one.

For  $n = 6$ , the data matrix has rank defect 3, and  $\tilde{\mathbf{e}}$  is in its three dimensional right nullspace, which is spanned by the last three columns of  $\mathbf{V}$ . These three vectors are transformed back into the matrices  $\mathbf{Y}$ ,  $\mathbf{Z}$  and  $\mathbf{W}$  according to equation (8). Then we have:

$$\mathbf{E} = y\mathbf{Y} + z\mathbf{Z} + w\mathbf{W} , \quad (10)$$

where  $y, z, w$  are unknown real values. Given the arbitrary scale of  $\mathbf{E}$ , we can set  $w = 1$ . To compute  $y$  and  $z$ , substitute equation (10) into equation (6). This results in nine third degree polynomials in  $y$  and  $z$ :

$$\mathbf{B}\mathbf{v} = 0, \mathbf{v} \stackrel{\text{def}}{=} (y^3, y^2z, yz^2, z^3, y^2, yz, z^2, y, z, 1)^T , \quad (11)$$

where the  $9 \times 10$  matrix  $\mathbf{B}$  contains the coefficients of the polynomials. The common root  $(y, z)$  of the nine multivariate polynomials can be computed by various methods. As the solution is unique, we can choose a very simple method: compute the right nullvector  $\mathbf{b}$  of  $\mathbf{B}$  via SVD and extract the root  $y = b_8/b_{10}$ ,  $z = b_9/b_{10}$ . Note, however, that this method ignores the structure of the vector  $\mathbf{v}$ . According to equation (10), this finally gives  $\mathbf{E}$ . For  $n > 6$ , the same algorithm can be applied.

### 3.4 Five Point Algorithm

The first part of the five point algorithm is very similar to the six point algorithm. For  $n = 5$ ,  $\mathbf{A}$  has rank defect 4 and we get the following linear combination for  $\mathbf{E}$ :

$$\mathbf{E} = x\mathbf{X} + y\mathbf{Y} + z\mathbf{Z} + w\mathbf{W} , \quad (12)$$

where  $x, y, z, w$  are unknown scalars and  $\mathbf{X}, \mathbf{Y}, \mathbf{Z}, \mathbf{W}$  are formed from the last four columns of  $\mathbf{V}$  according to equation (8). Again, we set  $w = 1$ . Substituting equation (12) into the equations (5) and (6) gives ten third degree polynomials  $\mathbf{M}\mathbf{m} = 0$  in three unknowns, where the  $10 \times 20$  matrix  $\mathbf{M}$  contains the coefficients and the vector  $\mathbf{m}$  contains the monomials:

$$\mathbf{m} = (x^3, x^2y, x^2z, xy^2, xyz, xz^2, y^3, y^2z, yz^2, z^3, x^2, xy, xz, y^2, yz, z^2, x, y, z, 1)^T .$$

The multivariate problem can be transformed into a univariate problem, which can then be solved using the companion matrix or Sturm sequences (Nistér, 2004). A more efficient variant of the five point algorithm (Stewénus, 2005; Stewénus et al., 2006) directly solves the multivariate problem by using Gröbner bases. First, Gauss Jordan elimination with partial pivotization is applied to  $\mathbf{M}$ . This results in a matrix  $\mathbf{M}' = (\mathbf{I}\mathbf{B})$ , where  $\mathbf{I}$  is the  $10 \times 10$  identity matrix and  $\mathbf{B}$  is a  $10 \times 10$  matrix. The ten polynomials defined by  $\mathbf{M}'$  are a Gröbner basis and have the same common roots as the original system. Now, form the  $10 \times 10$  action matrix  $\mathbf{C}$  as follows: the first six rows of  $\mathbf{C}^T$  equal the first six rows of

$\mathbf{B}$ ,  $C_{1,7} = 1, C_{2,8} = 1, C_{3,9} = 1, C_{7,10} = 1$ , all remaining elements are zero. The eigenvectors  $\mathbf{u}_i$  corresponding to real eigenvalues of  $\mathbf{C}^T$  give the up to ten common real roots:  $x_i = u_{i,7}/u_{i,10}$ ,  $y_i = u_{i,8}/u_{i,10}$ ,  $z_i = u_{i,9}/u_{i,10}$ . By substituting into equation (12), each root  $(x_i, y_i, z_i)$  gives a solution for  $\mathbf{E}$ .

### 3.5 Normalization

According to (Hartley and Zisserman, 2003), point correspondences should be normalized before applying the eight or seven point algorithm to improve stability. The (inhomogeneous) points are normalized by translating such that their mean is in the origin and scaling by the inverse of their average norm. In homogeneous coordinates, the third coordinate (assumed to be 1 for all points) is simply ignored and not changed. The normalization is applied in each image independently. When using camera normalized coordinates, the same normalization can be used. For the six and five point algorithms, however, such a normalization is not possible, as it does not preserve equation (6).

### 3.6 Constraint Enforcement

Note that the solution computed by the eight and seven point algorithms does *not* respect all properties of an essential matrix as presented in section 2.2. This might also be the case for the six point algorithm because of the trick applied to solve the polynomial equations (ignoring the structure of the vector  $\mathbf{v}$ ). Thus, each resulting essential matrix should be corrected by enforcing that its singular values are  $(s, s, 0)$  with  $s > 0$  (we use  $s = 1$ ). This can be achieved by SVD and subsequent matrix multiplication using the desired singular values.

Even though the five point algorithm actually computes valid essential matrices, we also apply the constraint enforcement to them. This has the additional effect of normalizing the scale of the essential matrices, which appears desirable for some of the experiments.

### 3.7 Selecting the Correct Solution

The seven and five point algorithms can produce more than one solution. If there are additional point correspondences, the single correct solution can be selected. For each solution, the deviation of each correspondence from the epipolar constraint is measured and summed up over all correspondences. The solution with the smallest error is selected. There are various possibilities to measure the deviation from the epipolar constraint (Hartley and Zisserman, 2003):

1. The algebraic error:  $|\mathbf{p}^T \mathbf{F} \mathbf{p}'|$ .
2. The symmetric squared geometric error:

$$\frac{(\mathbf{p}^T \mathbf{F} \mathbf{p}')^2}{[\mathbf{F} \mathbf{p}' ]_1^2 + [\mathbf{F} \mathbf{p}' ]_2^2} + \frac{(\mathbf{p}'^T \mathbf{F} \mathbf{p})^2}{[\mathbf{F}^T \mathbf{p} ]_1^2 + [\mathbf{F}^T \mathbf{p} ]_2^2}, \quad (13)$$

where  $[\cdot]_i$  denotes the  $i$ th element of a vector,

3. The squared reprojection error  $d_2(\mathbf{p}, \mathbf{q})^2 + d_2(\mathbf{p}', \mathbf{q}')^2$ , where  $d_2$  denotes the Euclidean distance, and  $\mathbf{q}$  and  $\mathbf{q}'$  are the reprojections of the triangulated 3D point. For a suitable triangulation algorithm, the reader is referred to the literature (Hartley and Sturm, 1997; Hartley and Zisserman, 2003).
4. The Sampson error:

$$\frac{(\mathbf{p}^T \mathbf{F} \mathbf{p}')^2}{[\mathbf{F} \mathbf{p}' ]_1^2 + [\mathbf{F} \mathbf{p}' ]_2^2 + [\mathbf{F}^T \mathbf{p} ]_1^2 + [\mathbf{F}^T \mathbf{p} ]_2^2}. \quad (14)$$

### 3.8 RANSAC

To achieve robustness to false correspondences, the well known (adaptive) RANdom SAMple Conensus (RANSAC) algorithm (Fischler and Bolles, 1981; Hartley and Zisserman, 2003) can be applied:

Input: Point correspondences  $\mathcal{D}$ .

1. Iterate  $k$  times:
  - (a) Randomly select  $m$  elements from  $\mathcal{D}$ .
  - (b) Estimate the essential matrix from this subset.
  - (c) For each resulting solution  $\mathbf{E}$ :
    - i. Compute  $\mathcal{S} = \{(\mathbf{p}, \mathbf{p}') \in \mathcal{D} \mid d_E(\mathbf{p}, \mathbf{p}') < c\}$ , where  $d_E$  is an error measure from section 3.7.
    - ii. If  $\mathcal{S}$  is larger than  $\mathcal{B}$ : set  $\mathcal{B} := \mathcal{S}$  and adapt  $k$ .
2. Estimate  $\mathbf{E}$  from  $\mathcal{B}$  with automatic selection of the correct solution.

For details, the reader is referred to the literature. We will investigate the choice of the parameters  $m$  and  $c$ .

### 3.9 Combining Algorithms

There is no single best algorithm for all situations. This makes it difficult to choose a single one, especially if there is no prior knowledge about the camera motion (see section 4). Hence, we propose the novel approach of combining two or more algorithms, which exploits their combined strengths. We run several algorithms on the same data to produce a set of candidate solutions. The automatic selection procedure is applied to select the best solution. We call this procedure the *combination* algorithm.

It is straight forward to apply the combination in RANSAC. However, we also have the possibility to use a single algorithm during RANSAC iterations and a combination for the final estimation from the best support set  $\mathcal{B}$ . We will use the name *final combination* for this strategy. It has the advantage that the five point algorithm can be used during iterations with small sample size  $m = 5$  and the five and eight point algorithms can be combined for the final estimation.

## 4 EXPERIMENTS

### 4.1 Simulation

The simulation consists of two virtual pinhole cameras  $(\mathbf{K}, \mathbf{I}, 0)$  and  $(\mathbf{K}, \mathbf{R}_G, \mathbf{t}_G)$  with image size  $640 \times 480$ ,  $f_x = f_y = 500$ ,  $s = 0$ ,  $o_x = 320$ ,  $o_y = 240$ . The scene consists of random 3D points uniformly distributed in a cuboid (distance from first camera 1, depth 2, width and height 0.85). These 3D points can be projected into the cameras. Noise is simulated by adding random values uniformly distributed in  $[-\phi/2, \phi/2]$  to all coordinates. We choose  $\phi = 3$  for all experiments.

We use two different error measures to compare the estimate for  $\mathbf{E}$  to the ground truth for relative pose:

- The translation error  $e_t$  is measured by the angle (in degree,  $0 \leq e_t \leq 90$ ) between the ground truth translation  $\mathbf{t}_G$  and the estimate computed from  $\mathbf{E}$ .
- The rotation error  $e_r$  is measured by the rotation angle (in degree,  $0 \leq e_r \leq 180$ ) of the relative rotation  $\mathbf{R}_{\text{rel}}$  between the ground truth orientation  $\mathbf{R}_G$  and the estimate  $\mathbf{R}_E$  computed from  $\mathbf{E}$ :  $\mathbf{R}_{\text{rel}} = \mathbf{R}_G \mathbf{R}_E^T$ . The ambiguity resulting in two solutions for  $\mathbf{R}_E$  is resolved by computing the angle for both and taking the smaller one as error  $e_r$ .

All experiments are repeated at least 500 times each. Finally, the median  $e_T$  of  $e_t$  and  $e_R$  of  $e_r$  over all repetitions is computed. In the evaluation, we focus on the median translation error  $e_T$  and include results for the median rotation error  $e_R$  in the appendix. The rotation error  $e_R$  is much lower and gives structurally very similar results.

#### 4.1.1 Outlier-free Data

First, we analyse the performance of the automatic selection of the best solution for the five point algorithm. Figure 1 shows the results for sideways motion ( $\mathbf{t}_G = (0.1, 0, 0)^T$ ,  $\mathbf{R}_G = \mathbf{I}$ ). It also contains the error of the ideal selection which is computed by comparing all essential matrices to the ground truth. The

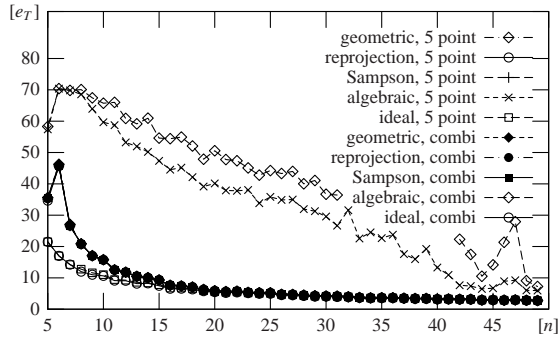


Figure 1: Comparison of error measures for automatic selection of the best solution in the five point algorithm, sideways motion. Median translation error  $e_T$  for varying number of point correspondences  $n$ . The plots for “euclidean”, “reprojection”, “Sampson” and their “combi” variants are almost identical, as are “ideal” and “ideal, combi”.

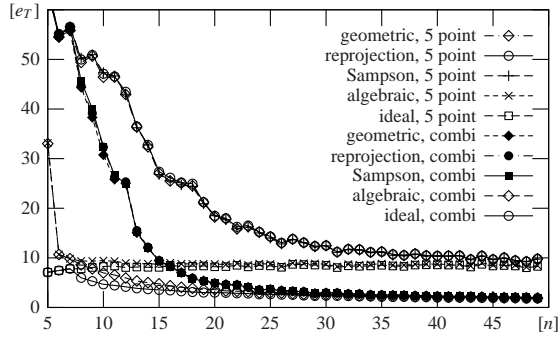


Figure 2: Comparison of error measures for automatic selection of the best solution in the five point algorithm, forward motion. Median translation error  $e_T$  for varying number of point correspondences  $n$ . The plots for “euclidean”, “reprojection” and “Sampson” are almost identical, as are their “combi” variants.

automatic selection works equally well with all error measures except for the algebraic one. Given enough points, the results almost reach the ideal selection.

In case of forward motion ( $t_G = (0, 0, -0.1)^T$ ,  $R_G = I$ ), the algebraic error is best (figure 2). Given enough points, the other error measures also give good results. For few points, however, the selection does not work well. Thus, if there is no prior knowledge about the translation, the Sampson or geometric error measure are the most reasonable choice. The reprojection error is also fine, but computationally more expensive.

The next experiment compares the various estimation algorithms. In contrast to the results presented by Stewénius, Engels and Nistér (Stewénius et al., 2006; Nistér, 2004), we apply automatic selection with the Sampson error measure for the five and seven point algorithms, which gives a more realistic comparison.

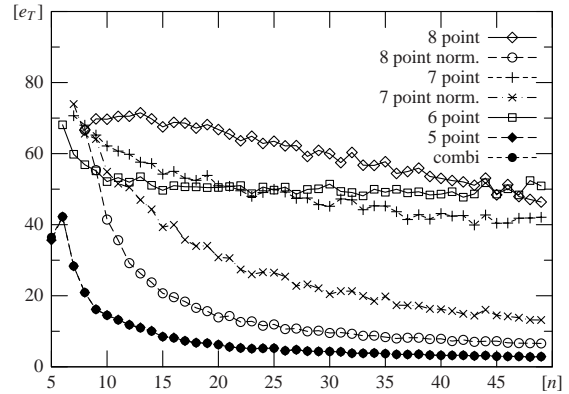


Figure 3: Comparison of algorithms with Sampson error for automatic selection, sideways motion. Median translation error  $e_T$  for varying number of point correspondences  $n$ . The plots for “5 point” and “combi” are almost identical.

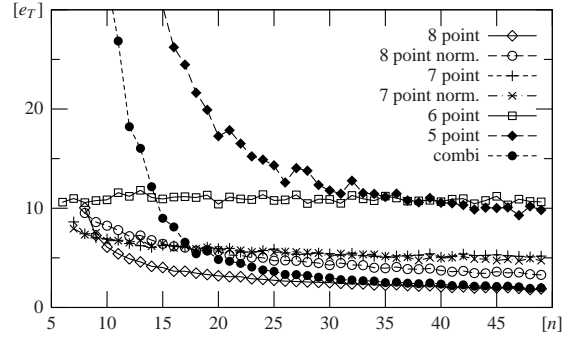


Figure 4: Comparison of algorithms, forward motion. Median translation error  $e_T$  for varying number of point correspondences  $n$ . The plots for “7 point” and “7 point norm.” are mostly identical.

Figures 3 and 4 show the results for sideways and forward motion, respectively. For sideways motion, the five point algorithm with automatic selection still gives superior results. For forward motion, however, the eight point algorithm is best. Surprisingly, in this case, the eight point algorithm with data normalization is *worse* than without normalization.

Given this situation, we add a combination of the five point and the *unnormalized* eight point algorithms to the comparison (“combi”). For sideways motion (figures 1 and 3), the results of the combination are almost identical to the five point results (except for selection with the algebraic error). For forward motion (figures 2 and 4), the automatic selection works better than with the five point algorithm alone, but still needs enough points to produce good results. Then, however, the combination reaches the results of the unnormalized eight point algorithm, which is the best single algorithm in this situation.



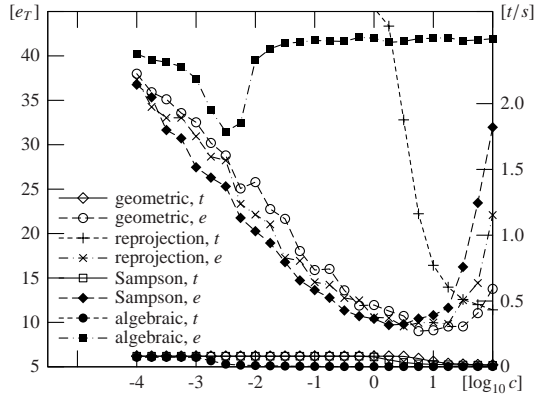


Figure 5: Five point RANSAC with all four error measures. Median translation error  $e_T$  and mean computation times for varying values of the threshold  $c$ . Outlier probability  $r = 29.44\%$  ( $r_s = 16\%$ ).

The consequence of the simulation results is that our combination with the Sampson error measure for automatic selection is the best choice for outlier-free data without prior knowledge about the translation.

#### 4.1.2 RANSAC

Next, we analyse the best choice of the threshold  $c$  and also the choice of the error measure for the RANSAC variant of the five point algorithm. In this experiment, we use a different camera setup:  $t_G = (0.1, 0, 0.1)^T$  and  $R_G$  is a rotation about the  $y$  axis by  $0.01$  ( $\approx 5.7^\circ$ ). Outliers are generated by replacing each projected image point by a randomly generated point within the image with probability  $r_s$ . The probability of a point *pair* being an outlier is thus  $r = 1 - (1 - r_s)^2$ .

Figure 5 shows the median translation error as well as the mean computation times for 29.44% outliers. The geometric, reprojection and Sampson error measures give good results. However, the computation time for the reprojection error is at least 10 times higher. Given an optimal threshold  $c_{opt}$ , the geometric error gives the best results, even though the difference is small. However, as further experiments show,  $c_{opt}$  depends on the amount of outliers  $r$  and is thus difficult to guess. For the Sampson error,  $c_{opt}$  is much less affected by  $r$ , and is roughly equal to the noise level.

In the next experiment, we analyse the choice of the sample size  $m$ . We use the Sampson error with threshold  $c = 1.5$ . Figure 6 shows that increasing the sample size decreases the median translation error. However, the computation time increases drastically (figure 7). In case of sample size  $m = 8$ , we also include the combination of the five point and the unnor-

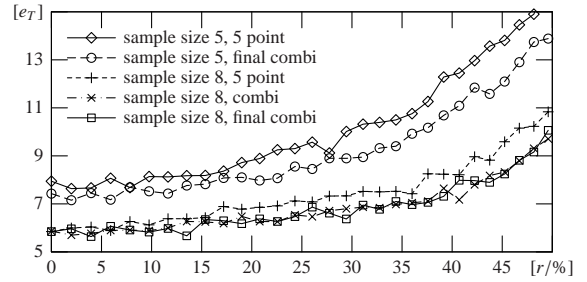


Figure 6: Median translation error  $e_T$  for RANSAC algorithms with various sample sizes  $m$  on data with varying amounts of outliers  $r$ .

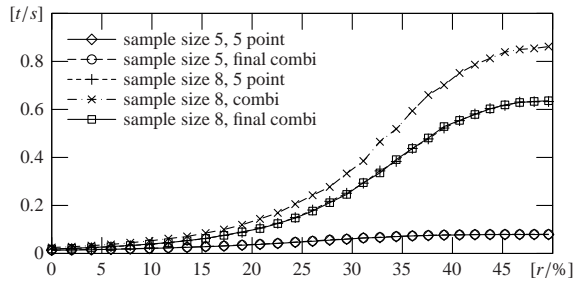


Figure 7: Mean computation times for RANSAC algorithms with various sample sizes  $m$  on data with varying amounts of outliers  $r$ .

malized eight point algorithm (“combi”), which gives better results than the five point algorithm, but also further increases the computation time. Note, however, that the implementation could be optimized by exploiting that the first part of both algorithms is identical (SVD of data matrix). For sample sizes  $m = 5$  and  $m = 8$ , we apply the final combination algorithm using the five point algorithm during RANSAC iterations and the five point and unnormalized eight point algorithms only for the final estimation from the best support set (“final combi”). In case of  $m = 8$ , this approach gives comparably good results to the previous case, but without the additional computation time. Furthermore, we also get the final combination benefit for sample size  $m = 5$ .

## 4.2 Real Data

To verify the results presented above, we also perform experiments with a calibrated camera (Sony DFW-VL500) mounted onto a robotic arm, which provides us with ground truth data for relative pose. We record two different sequences: motion on a sphere around the scene in  $10^\circ$  steps with the camera pointing to the center (five image pairs), and forward motion (four image pairs). The scenes are shown in figure 8. We



Figure 8: Scenes used for the experiments with real data. Top: sequence 1, image pair 2. Bottom: sequence 2, pair 1.

Table 1: Median translation errors  $e_T$  on scene 1.

image pair	1	2	3	4	5
5 point	<b>0.8</b>	<b>1.9</b>	<b>0.7</b>	<b>0.4</b>	<b>1.0</b>
final combi	<b>0.8</b>	<b>1.9</b>	<b>0.7</b>	<b>0.4</b>	<b>1.0</b>
6 point	40.0	61.3	65.1	26.9	4.9
7 point	59.0	62.6	69.0	2.4	2.0
7 point norm.	23.7	58.2	39.0	6.8	16.3
8 point	62.7	65.4	66.5	20.5	21.6
8 point norm.	65.9	36.0	29.9	6.6	15.4

use SIFT (Lowe, 2004) to detect 200 point correspondences. These are fed into the RANSAC variants (using the Sampson error with  $m = 8$  and  $c = 1$ ) of all algorithms presented in section 3, and also the “final combi” algorithm as in the synthetic experiments.

The results are shown in tables 1 and 2. On the first scene, only the five point and the “final combi” algorithms give good results, which may be caused by the dominantly planar distribution of the SIFT features (Nistér, 2004). On the second scene, most algorithms work well. In contrast to the synthetic experiments with forward motion, the eight point algorithm with normalization is better than without. It gives the best results for this scene. The five point algorithm has problems with the second image pair, but “final combi” works well and is close to the eight point algorithm. Overall, these experiments show that the “final combi” algorithm is the best choice if there is no prior knowledge about the relative pose.

## 5 CONCLUSIONS

We have shown that the five point algorithm with automatic selection of the single best solution gives very good estimates for relative pose. Due to its problems with forward motion, we proposed a combination with the eight point algorithm and showed that this gives very good results. In presence of outliers,

Table 2: Median translation errors  $e_T$  on scene 2.

image pair	1	2	3	4
5 point	2.1	13.4	<b>1.2</b>	1.5
final combi	1.1	1.6	<b>1.2</b>	1.4
6 point	1.0	<b>0.5</b>	6.4	1.2
7 point	<b>1.0</b>	1.2	1.2	1.8
7 point norm.	5.9	17.8	13.0	2.7
8 point	1.2	1.7	1.4	1.6
8 point norm.	1.0	1.2	1.2	<b>1.2</b>

RANSAC provides the necessary robustness. Our (final) combination is also beneficial in this case.

Finally, we summarize our recommendations for cases without prior knowledge about the motion. We suggest using RANSAC with the Sampson error, the five point algorithm during iterations, and the five point and normalized eight point algorithms for the final estimation. We called this approach *final combination*. The RANSAC threshold should be chosen similar to the noise level. The sample size has to be at least 5, but should be increased to 8 or 10 (or even more) if computation time permits. Furthermore, it is advantageous to use as many points as possible.

## REFERENCES

- Cox, D. A., Little, J., and O’Shea, D. (2005). *Using Algebraic Geometry*. Graduate Texts in Mathematics. Springer, 2nd edition.
- Fischler, M. A. and Bolles, R. C. (1981). Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the Association for Computing Machinery*, 24(6):381–395.
- Hartley, R. and Sturm, P. (1997). Triangulation. *Computer Vision and Image Understanding*, 68(2):146–157.
- Hartley, R. and Zisserman, A. (2003). *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2nd edition.
- Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110.
- Nistér, D. (2004). An efficient solution to the five-point relative pose problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(6):756–770.
- Philip, J. (1996). A non-iterative algorithm for determining all essential matrices corresponding to five point pairs. *Photogrammetric Record*, 15(88):589–599.
- Pizarro, O., Eustice, R., and Singh, H. (2003). Relative pose estimation for instrumented, calibrated imaging platforms. In *Proceedings of Digital Image Computing Techniques and Applications*, pages 601–612.
- Stewénus, H. (2005). *Gröbner Basis Methods for Minimal Problems in Computer Vision*. PhD thesis, Centre for Mathematical Sciences LTH, Lund Univ., Sweden.

Stewénius, H., Engels, C., and Nistér, D. (2006). Recent Developments on Direct Relative Orientation. *ISPRS Journal of Photogrammetry and Remote Sensing*, 60(4):284–294.

Triggs, B., McLauchlan, P. F., Hartley, R. I., and Fitzgibbon, A. W. (1999). Bundle Adjustment — A Modern Synthesis. In *Proc. of the Int. Workshop on Vision Algorithms: Theory and Practice*, pages 298–373.

## APPENDIX

In figures 9–13 and tables 3–4, we present additional results for the rotation error  $e_R$ . Each figure refers to the corresponding figure for the translation error  $e_T$ .

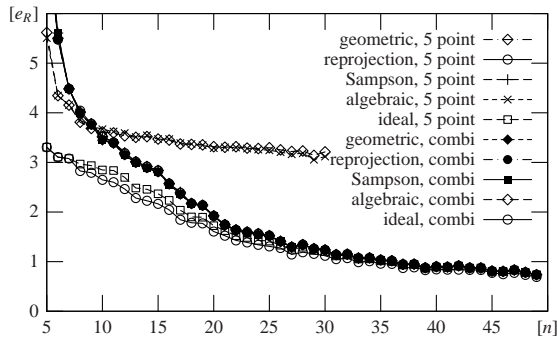


Figure 9: As figure 1, but using median rotation error  $e_R$ .

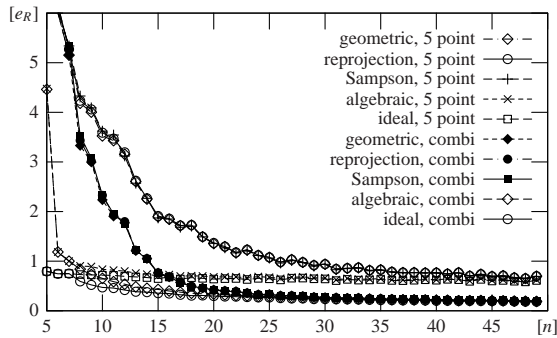


Figure 10: As figure 2, but using median rotation error  $e_R$ .

Table 3: Median rotation errors  $e_R$  on scene 1.

image pair	1	2	3	4	5
5 point	<b>0.3</b>	<b>0.6</b>	<b>0.2</b>	0.2	<b>0.3</b>
final combi	0.6	3.5	0.2	0.2	<b>0.3</b>
6 point	10.5	7.8	10.6	2.4	1.0
7 point	6.7	11.6	8.7	<b>0.1</b>	0.3
7 point norm.	8.8	8.8	10.1	9.9	20.0
8 point	12.7	11.7	11.1	3.1	8.8
8 point norm.	16.7	11.5	14.8	18.5	20.0

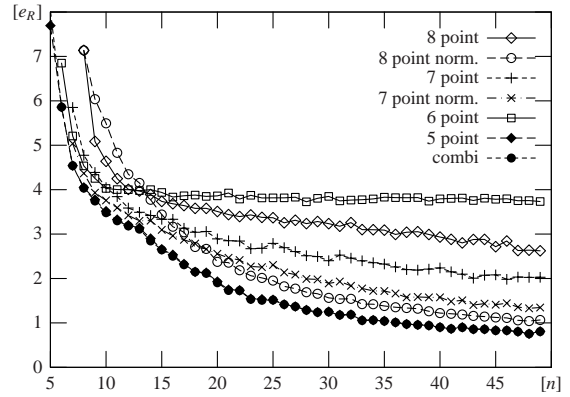


Figure 11: As figure 3, but using median rotation error  $e_R$ .

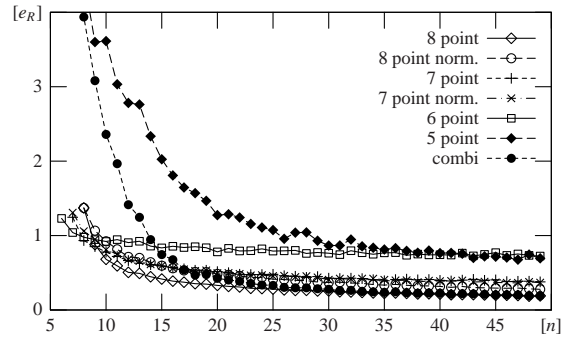


Figure 12: As figure 4, but using median rotation error  $e_R$ .

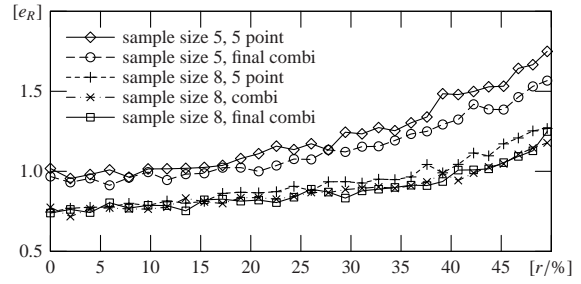


Figure 13: As figure 6, but using median rotation error  $e_R$ .

Table 4: Median rotation errors  $e_R$  on scene 2.

image pair	1	2	3	4
5 point	0.13	1.90	0.03	0.10
final combi	0.05	0.14	0.03	0.14
6 point	0.05	0.18	0.16	0.07
7 point	<b>0.03</b>	0.17	0.04	0.21
7 point norm.	0.23	3.04	0.56	0.89
8 point	0.05	<b>0.12</b>	0.04	0.12
8 point norm.	0.20	0.18	<b>0.03</b>	<b>0.01</b>