Recent Developments on Direct Relative Orientation

Henrik Stewénius [1], Christopher Engels, David Nistér

Center for Visualization and Virtual Environments, Computer Science Department, University of Kentucky

Abstract

This paper presents a novel version of the five-point relative orientation algorithm given in Nistér (2004). The name of the algorithm arises from the fact that it can operate even on the minimal five point correspondences required for a finite number of solutions to relative orientation. For the minimal five correspondences the algorithm returns up to ten real solutions. The algorithm can also operate on many points. Like the previous version of the five-point algorithm, our method can operate correctly even in the face of critical surfaces, including planar and ruled quadric scenes.

The paper presents comparisons with other direct methods, including the previously developed five-point method, two different six-point methods, the seven-point method, and the eight-point method. It is shown that the five-point method is superior in most cases among the direct methods.

The new version of the algorithm was developed from the perspective of algebraic geometry and is presented in the context of computing a Gröbner basis. The constraints are formulated in terms of polynomial equations in the entries of the fundamental matrix. The polynomial equations generate an algebraic ideal for which a Gröbner basis is computed. The Gröbner basis is used to compute the action matrix for multiplication by a single variable monomial. The eigenvectors of the action matrix give the solutions for all the variables and thereby also relative orientation. Using a Gröbner basis makes the solution clear and easy to explain.

Key words: Relative Orientation, Fivepoint Method, Sixpoint Method, Gröbner Basis

1 Introduction

Relative orientation has always been and remains one of the most important operations in photogrammetry. A review of the history of relative orientation is given by Wrobel in Gruen and Huang (2001). The gold standard is to perform bundle adjustment McGlone et al. (2004); Triggs et al. (2000); Wolf (1983), which is widely used and has been intensively studied. Bundle adjustment is the way to achieve the best accuracy that the data affords and should preferably always be used as the final computational stage of a relative orientation, unless an application has prohibitive constraints, such as real-time requirements.

Bundle adjustment is typically started from an approximate configuration of the scene points and camera poses. With a sufficiently accurate initial configuration, this will lead to satisfactory convergence of the bundle adjustment. However, with an inaccurate initial configuration there is no guarantee that the bundle adjustment will find the globally optimal solution.

Recently there have been several advances on direct relative orientation methods Philip (1996); Nistér (2004). By direct we intend any method with guaranteed convergence to a global optimum, which includes closed form solutions and solutions using singular value decomposition or eigenvalue decomposition. A review of the various direct methods is given by Förstner and Wrobel in McGlone et al. (2004).

The direct methods, although not minimising the preferred image reprojection error, have the advantage of not getting stuck in a local minimum of the cost function. They can be used to obtain an initial configuration to use in bundle adjustment when no accurate initial configuration is available. They have also proven very effective as hypothesis generators in hypothesise-and-test algorithms such as RANSAC used to perform automatic relative orientation directly from the raw images, including automatic feature extraction and matching, see for example see for example Fischler and Bolles (1981); McGlone et al. (2004); Nistér (2003).

In such hypothesise-and-test algorithms, many minimal sets of point corre-

Email addresses: stewe@vis.uky.edu (Henrik Stewénius), engels@vis.uky.edu (Christopher Engels), dnister@vis.uky.edu (David Nistér).

URLs: http://www.vis.uky.edu/~stewe/ (Henrik Stewénius), http://www.vis.uky.edu/~engels/ (Christopher Engels), http://www.vis.uky.edu/~dnister/ (David Nistér).

[1] This work is partially funded by the project VISCOS (foundation for strategic research), by EU project SMErobot and by the Swedish Research Council.

spondences are used to generate hypotheses for the relative orientation. Thus, the direct relative orientation method is actually used on a very small number of points each time, which is done to avoid the effects of outliers. The best relative orientation hypothesis generated in this manner is then found according to a robust version of reprojection error using all the corresponding points.

The most well-known direct methods have been named based on the smallest number of points that they require in order to operate correctly and yield a finite number of possible solutions for the relative orientation. For the core task of performing relative orientation of two views with calibrated intrinsic parameters, which entails estimating the relative rotation and the translation direction between the two views, the minimal number of point correspondences required is five.

The five-point problem was first shown by Kruppa (1913) to have at most eleven solutions (here, the twisted pair ambiguity is not considered a real ambiguity since it can be resolved by requiring that the scene points are in front of the cameras). This was improved upon more recently Faugeras and Maybank (1990), showing that it has at most ten solutions. It has also been found Nistér (2004) that there are pathological cases with ten real solutions that are well separated in parameter space and have all the scene points in front of the cameras.

Kruppa also gave an algebraic algorithm for solving the five-point problem, although it is not very amenable to efficient practical implementation. Efficient practical solutions to the five-point problem have been given more recently. Philip (1996) gave a method that finds the solutions for relative orientation by extracting the roots of a thirteenth-degree polynomial. This was improved upon by Nistér (2004). The solutions are found as the roots of a tenth-degree polynomial, which corresponds exactly to the intrinsic degree of difficulty of the problem.

In this paper, we present a novel version of this algorithm. The new version is solidly grounded in algebraic geometry and based on standard methods from this field. We compare it to the previous version and show that the new version is more numerically stable. The new version works by deriving a Gröbner basis for the ideal generated by the polynomial constraints on the essential matrix. This is done in a straightforward manner. The Gröbner basis can then be used to construct a 10×10 matrix whose eigenvalues and eigenvectors encode the solutions. The matrix is an action matrix for multiplication in the quotient ring of polynomials modulo the ideal generated by the polynomial constraints. This type of action matrix is a standard way of solving polynomials and we review some of this theory. For a more complete treatment of the tools used, see Cox et al. (1997, 1998).

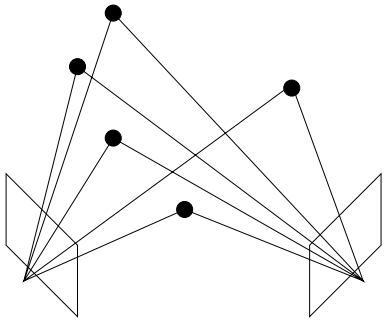

Fig. 1. We give a novel version of the direct method to solve for relative orientation of two cameras given five or more point correspondences.

The five-point method has the advantage that it works for planar scenes and also quadric critical surfaces. That is, it yields a number of possible solutions as usual, among which the true solution resides. The analysis of the algorithms (excepting the six-point method of Pizarro) under critical configurations is addressed in Philip (1998).

This is obviously useful if the scene is actually planar or a critical surface, but it also gives the five-point method better stability for near-planar and near-critical surfaces. There are direct methods that explicitly assume a planar scene, see for example McGlone et al. (2004); Faugeras (1993). However, these do not apply to near-planar scenes. The five-point method can operate both in the general case and seamlessly into a completely planar scene. To the best of our knowledge, there is only one other direct method apart from the five-point method that could claim this desirable property. This is the six-point method of Pizarro Pizarro et al. (2003), which selectively uses the constraints from six point correspondences and yields up to six solutions. For general scenes, six point correspondences can be used to solve linearly for relative orientation as described by Philip (1996). However, this breaks down for a planar scene Philip (1998).

In our experiments we compare the five-point method with the six-point methods of Pizarro and Philip. We also compare to the seven and eight-point methods and find that the five-point method is superior among all the direct methods for minimal cases. The seven and eight-point methods operate without using any assumptions about the intrinsic parameters of the cameras, except for assuming a pinhole camera model. Seven points then lead to up to three solutions Sturm (1869); Hartley and Zisserman (2000) and eight points lead to a linear solve for a unique solution von Sanden (1908); Longuet-Higgins (1981).

All of these methods can also operate with many points and not only with the minimal sets of points. This is done in a least squares fashion by finding an approximate null-vector through the use of singular value decomposition. However, the least squares measure minimised is not the preferred image repro-

jection error and it is not clear without experimentation that the algorithms can make efficient use of the additional points. We present extensive experiments with large point sets and find that the five-point method provides the most consistent results among the direct methods also at using many points.

The rest of the paper is organised as follows. First an introduction to the geometric constraints used will be given in Section 2 followed by an introduction to the tools of algebraic geometry in Section 3. In Section 4 our new method is explained and in Section 6 the new solver is compared both to the old version of the five-point algorithm as well as other direct methods.

2 The algebraic constraints

The methods for relative orientation of two views all use the well known coplanarity McGlone et al. (2004), or epipolar constraint, which is encoded by the 3×3 fundamental matrix F as

$$q'^{\top} F q = 0, \tag{1}$$

where q and q' are the homogeneous coordinates of the two points from a point correspondence. We will focus on how the various methods find the fundamental matrix. To obtain the relative rotation and translation direction from the fundamental matrix is a standard operation, see for example McGlone et al. (2004). Any rank-two matrix is a possible fundamental matrix and we have the well known single cubic constraint:

Theorem 1 *A real non-zero 3×3 matrix F is a fundamental matrix only if it satisfies the equation*

$$det(F) = 0. \tag{2}$$

If we have calibrated cameras and use the calibrated coordinate system, the fundamental matrix specialises to an essential matrix E. An essential matrix has the additional property that the two non-zero singular values are equal. This constraint is straightforward to verify on a given matrix. To express it algebraically, we need the following cubic constraints on the essential matrix, adapted from Stefanovic (1973); Demazure (1988); Maybank (1993); Philip (1996):

Theorem 2 *A real non-zero 3×3 matrix E is an essential matrix if and only if it satisfies the equation*

$$2EE^{\top}E - trace(EE^{\top})E = 0. \tag{3}$$

The constraints from Equations (1), (2) and (3) are are used in order to compute the essential matrix, it is possible to use a lower number of equations but this makes the problem harder and more unstable. For more on these constraints, see Philip (1996).

3 Algebraic geometry

Algebraic geometry deals among other things with solving sets of multivariate polynomial equations such as the ones from Section 2. Our novel version of the five-point solver relies on results from this field, and we will briefly review the relevant material. For a more complete treatment of the tools used, see Cox et al. (1997, 1998).

A monomial in the unknowns $(x_1, \ldots, x_n)$ is a finite product of these unknowns and a polynomial is a finite sum of monomials. When computing and writing out polynomials the order of the monomials is very important and we will here first order them according to degree, highest degree first, and then within each degree according to phone-book (lexicographic) order. Ordering the monomials like this is usually more efficient than using a pure lexicographic order. The leading term of a polynomial is the first non-zero term.

For a set of polynomials $\{p_i\}$, the *ideal* $I(\{p_i\})$ is defined as the set of polynomials

$$\left\{ \sum \beta_i p_i \,|\, \beta_i \in \mathbb{C}[x_1, \ldots, x_n] \right\}. \tag{4}$$

All polynomials in the ideal $I(\{p_i\})$ evaluate to zero on the set of solutions to $\{p_i\}$. The *leading terms* of an ideal is the set of leading terms of all members of the ideal.

A *Gröbner basis* G of the ideal $I(\{p_i\})$ is a set of polynomials in $I(\{p_i\})$ such that every leading term of $I(\{p_i\})$ is divisible by the leading term of some member of G.

Since a Gröbner basis of an ideal I exposes all the leading terms of I, it gives a way of uniquely defining the remainders modulo I. We can thus consider the *quotient ring* $\mathbb{C}[x_1, \ldots, x_n]/I$, which is the set of distinct remainders modulo I. Intuitively, the quotient ring is the set of equivalence classes of polynomials that appear distinct when evaluated at the solutions to $\{p_i\}$. The dimensionality of the quotient ring is equal to the number of solutions. If the number of solutions is finite, then the dimensionality of the quotient ring is finite.

The monomials that are indivisible by the leading terms of G form a basis for

$\mathbb{C}[x_1, \ldots, x_n]/I$. If we label the basis monomials $u_1(\mathbf{x}), \ldots, u_N(\mathbf{x})$ and gather them in a vector

$$\mathbf{u}(\mathbf{x}) = \begin{bmatrix} u_1(\mathbf{x}) & \ldots & u_N(\mathbf{x}) \end{bmatrix}^\top, \tag{5}$$

we can use an N-dimensional coefficient vector v to represent any element $\mathbf{u}(\mathbf{x})^\top v$ in the quotient ring. Multiplication by a fixed polynomial f in the quotient ring is then a linear operator and the $N \times N$ matrix describing the operation is called the *action matrix* A_f. The action matrix works on the coefficient vectors in such a way that

$$f(\mathbf{x})\mathbf{u}(\mathbf{x})^\top v = \mathbf{u}(\mathbf{x})^\top A_f v, \quad \forall v \, \forall \mathbf{x} \in V, \tag{6}$$

which implies that

$$f(\mathbf{x})\mathbf{u}(\mathbf{x})^\top = \mathbf{u}(\mathbf{x})^\top A_f, \quad \forall \mathbf{x} \in V. \tag{7}$$

Note that this is the same as saying that if $\mathbf{x}$ is a solution to our polynomial equations, then $\mathbf{u}(\mathbf{x})$ is a left eigenvector to the action matrix A_f, corresponding to the eigenvalue $f(\mathbf{x})$. This means that for a general polynomial f, the N left eigenvectors of the action matrix A_f are up-to-scale versions of the monomial vector $\mathbf{u}(\mathbf{x})$ at the N solutions $\mathbf{x} \in V$.

Given a Gröbner basis for I, it is straightforward to compute the action matrix corresponding to any polynomial. With GrLex order, the monomials $x_1, \ldots, x_n$, consisting of the variables themselves, are typically present in the monomial basis. Hence, we can simply choose some simple polynomial f, for example $f = x_1$, compute the action matrix A_f, and then read off the values of the variables at all the solutions directly from the left eigenvectors of A_f. All we need is the Gröbner basis. We will find in Section 4 that in the case of the five-point problem this can be computed in a straightforward manner.

4 Solving the problem

Below we will give a detailed description of the five-point algorithm. However, the steps can be summarised for the reader familiar with algebraic geometry as

(1) Use the linear equations from the epipolar constraint (1) to parametrise the essential matrix with three unknowns.
(2) Use the rank constraint (2) and the trace constraint (3) to build ten third-order polynomial equations in the three unknowns. The monomials

are ordered in GrLex order and represented in a 10×20 matrix, where each row corresponds to an equation and each column corresponds to a monomial.

(3) Compute the Gröbner basis. This turns out to be as simple as performing a Gauss-Jordan elimination on the 10×20 matrix.

(4) Compute the 10×10 action matrix for multiplication by one of the unknowns. This is a simple matter of extracting the correct elements from the eliminated 10×20 matrix and organising them to form the action matrix.

(5) Compute the left eigenvectors of the action matrix. Read off the values for the three unknowns at all the solution points and back-substitute to obtain the solutions for the essential matrix.

We now proceed to present the details. As the observations (q_i, q_i') are known, the epipolar constraint (1) gives five linear constraints on the essential matrix E. An orthogonal basis for the four-dimensional null-space of these constraints is then computed using singular value decomposition. In the case of more than five points, we generate the equations for all the points in the same way and compute the closest approximation to a four-dimensional null-space by using the four singular vectors corresponding to the four smallest singular values.

It is now possible to rewrite E as

$$E = xE_1 + yE_2 + zE_3 + wE_4, \tag{8}$$

where E_i are the components of the null-space for the epipolar constraint. It is only possible to solve for E up to scale, so we set $w = 1$ and there are then three (x, y, z) remaining unknowns. Inserting E into constraint (2) and (3) gives ten polynomial equations of degree three. There are twenty monomials up to degree three in the unknowns (x, y, z) and we choose to work in GrLex $(x > y > z)$. This gives the following monomials vector:

$$X = \left[x^3, x^2y, x^2z, xy^2, xyz, xz^2, y^3, y^2z, yz^2, z^3, x^2, xy, xz, y^2, yz, z^2, x, y, z, 1\right]^\top, \tag{9}$$

note that the first 10 monomials are cubic and the other are of degree two or lower. The ten equations can now be written

$$MX = 0, \tag{10}$$

where M is a 10×20 matrix. The rows of the matrix turn out to be linearly independent in general. After Gauss-Jordan elimination the system can be written

$$\begin{bmatrix} I & B \end{bmatrix} X = 0, \tag{11}$$

where I is the 10×10 identity matrix and B a 10×10 matrix. Knowing that there are ten solutions, it is easy to see that this set of equations must be a GrLex Gröbner-basis (since there are only ten monomials that are not divisible by any of the leading monomials from the equations).

The action matrix A_x for multiplication by x is then computed by considering the action of x on the basis monomials of $\mathbb{C}[x, y, z]/I$. The ten sets of solutions can then be read of out from the the eigenvectors of A_x^t after normalizing the last element to 1. The above can appear a bit complicated but given the matrix M the above steps are performed by the following piece of `Matlab` code:

```
B = M(:,1:10)\M(:,11:20);
At = zeros(10);
At = -B([1 2 3 5 6 8], :);
At(7,1) = 1;
At(8,2) = 1;
At(9,4) = 1;
At(10,7) = 1;
[V,D] = eig(At);
SOLS =   V(7:9,:)./(ones(3,1)*V(10,:));
```

A weakness in this scheme is that it is assumed that $w \neq 0$. For better stability it is possible to compute the solutions as normalized to 1 but then the number of solutions is doubled.

5 The other methods

In Section 6, the five-point method will be compared to some other direct methods. Common to all these methods is that they first use the N observed points in the epipolar constraint (1) to get linear constraints on the Fundamental/Essential matrix. The possible fundamental matrices are then parametrised as

$$F = x_0 F_0 + \sum_{i=1}^{8-N} x_i F_i, \tag{12}$$

where F_i are basis vectors for the null-space of the constraints. As F can only be determined up to scale, x_0 is set to zero. For $N < 8$ the remaining uncertainty is removed by inserting F into constraint (2), and when the intrinsic calibration is known, constraint (3). The resulting equations are used to com-

pute a polynomial in x_1 and solutions to the other variables are computed through back-substitution. The fundamental matrix is then computed from Equation (12).
The methods we compare are:

- The old version of the five-point method Nistér (2004). From the 10 equations in (2) and (3) a tenth degree polynomial in x_1 is computed. The roots of the polynomial are computed by eigen-decomposing a companion matrix. This method is more reliable than the faster option from Nistér (2004) of using Sturm sequences to bracket the roots.
- The six-point method of Pizarro et al. (2003). The 9 equations from (3) are composed into a 9×10 matrix from which the 4 rows corresponding to the four largest singular values are selected. From these four equations, a sixth degree polynomial in x_1 is computed.
- The linear six-point solver from Philip (1998). The 9 equations in (3) are composed into a 9×10 matrix and the unknowns are solved for linearly.
- The seven-point solver. The rank constraint (2) gives a third degree polynomial in x_1.
- The eight-point algorithm. A linear solver.

6 Experiments

We first compare the numerical error between the novel and previous five-point methods. We then compare the other methods to the new five-point method under various noisy conditions. Finally, we present a short real-world example.

6.1 Numerical precision

In order to investigate the precision of the five-point algorithms, we examine the numerical error on synthetic data in noise-free yet geometrically realistic conditions. For the purposes of numerical testing, we use the minimal set of five points. Note that more points only affect the computation of the initial null space, performed by singular value decomposition, which is known to be numerically stable. Since the algorithms produce multiple up-to-scale solutions for the essential matrix, we compute the error measure

$$\min_i \min(\| \frac{\hat{E}_i}{\| \hat{E}_i \|} - \frac{E}{\| E \|} \|, \| \frac{E}{\| E \|} + \frac{\hat{E}_i}{\| \hat{E}_i \|} \|), \tag{13}$$

where $\hat{E}_i$ are the ten estimated essential matrices, and E is the true essential matrix. In each case, we sample 50000 minimal sets of points and record the

resulting numerical errors. Note that we use double precision computations, so the maximum numerical precision of the machine is $2.2204e-16$.

We construct the configurations of cameras and points using the following parameters: number of points, scene depth, scene distance, scene width, baseline, and motion direction. We set the first camera at the origin and randomise scene points uniformly inside a block straight in front of it. The block starts at the scene distance, and the thickness of the block is the scene depth. The second camera is positioned according to the motion direction and baseline and is oriented toward the centroid of the scene points. As the default conditions for all the tests, the distance to the scene from the camera is 2, as is the width of the scene, which translates to a field of view of about 53 degrees. The default magnitude of the baseline to the second camera is 0.2. For non-planar scenes, the default scene depth is 2. We would expect a wider baseline to yield more accurate results, which we verify in Section 6.2

Figure 2 shows results for sideways camera motion of a non-planar scene. The results for a planar scene are shown in Figure 3, and the results for forward motion in a non-planar scene are in Figure 4. The results indicate only a minor increase in accuracy; note however that the histogram of the errors in the previous method has a tail consisting of a small number of substantial errors, whereas the new method eliminates the larger errors, and the largest error in that set is less than 10^{-5}. These errors result in a mean error for the previous method of approximately 10^{-5}, while the mean error of the revised method was less than 10^{-10}.

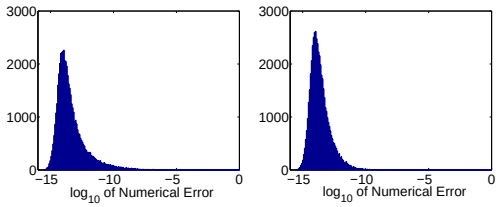

Fig. 2. Distribution of the numerical error of the five-point methods for sideways motion under normal conditions, 50000 samples. Left: Previous method. Right: New method. The median numerical error of the previous method is 1.9466e-14, while the median error for the new method is 1.6351e-14.

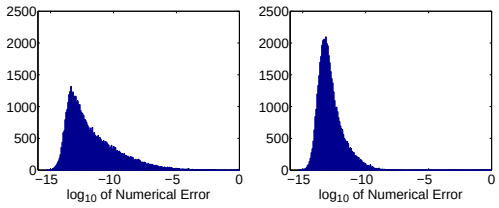

Fig. 3. Distribution of the numerical error of the five-point methods for sideways motion under planar conditions, 50000 samples. Left: Previous method. Right: New method.

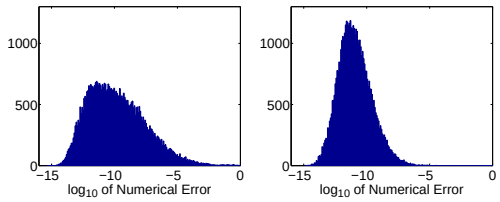

Fig. 4. Distribution of the numerical error of the five-point methods for forward motion under normal conditions, 50000 samples. Left: Previous method. Right: New method.

6.2 Performance under noise

In this section we compare the results of the new five-point method and other methods under noisy conditions. The default parameters are the same as those used in the numerical tests, except that we also simulate feature location errors by introducing a noise parameter into the image data. In the default case, we include a Gaussian perturbation in each image point with a standard deviation of one pixel in a 2000×2000 pixel image. To compare the methods, we measure the angle of translational error between the estimated solutions and the true motions in various situations. Each data point represents the median value of 5000 samples.

In the following experiments we consider both solutions from the minimal sets of points necessary for each method, i.e. 5,6,7, or 8 points, as well as nonminimal sets of 50 points. Figure 5 shows the results of increasing the number of points used in each algorithm beyond the minimal cases for both sideways and forward motion. Note that the five-point, six-point, and eight-point methods benefit the most from additional points, although the seven-point method also benefits somewhat. The seven-point, eight-point, and six-point linear methods also have a bias towards forward motion Nistér (2004), which partially accounts for the improvement of those methods in the forward case. The linear six-point solver does not perform well using additional points, so we preclude it from further non-minimal case tests.

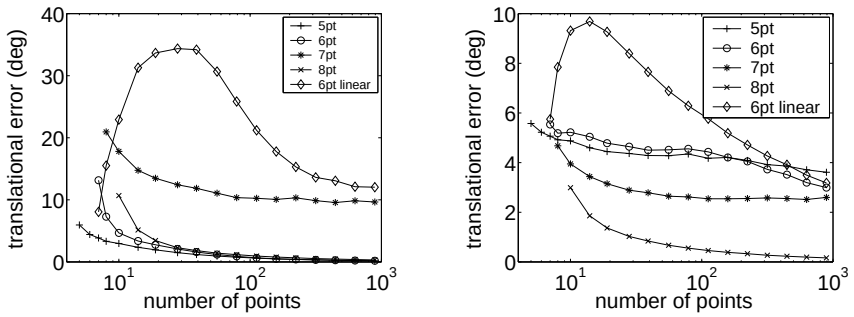

Fig. 5. Translational error relative to number of points. Left: sideways motion. Right: forward motion

We also measure the failure rate of the various algorithms. We define the failure rate at a specific error threshold to be the percentage of instances solved that give no solutions where the translational error is less than the specified error. The distributions of the error levels are shown in Figure 6 and the failure rates are shown in Figure 7. Note that in the minimal case, which is useful for creating hypotheses for RANSAC Fischler and Bolles (1981), the five-point method is clearly the best choice. In the overdetermined case, the performances of the five-point, six-point, and eight-point methods are comparable, while the seven-point also improves somewhat.

Figure 8 shows the stability of the algorithms under increasing noise. In each case, we create Gaussian error that simulates up to a one pixel error in a 2000×2000 image. Note that the five-point method is generally least affected

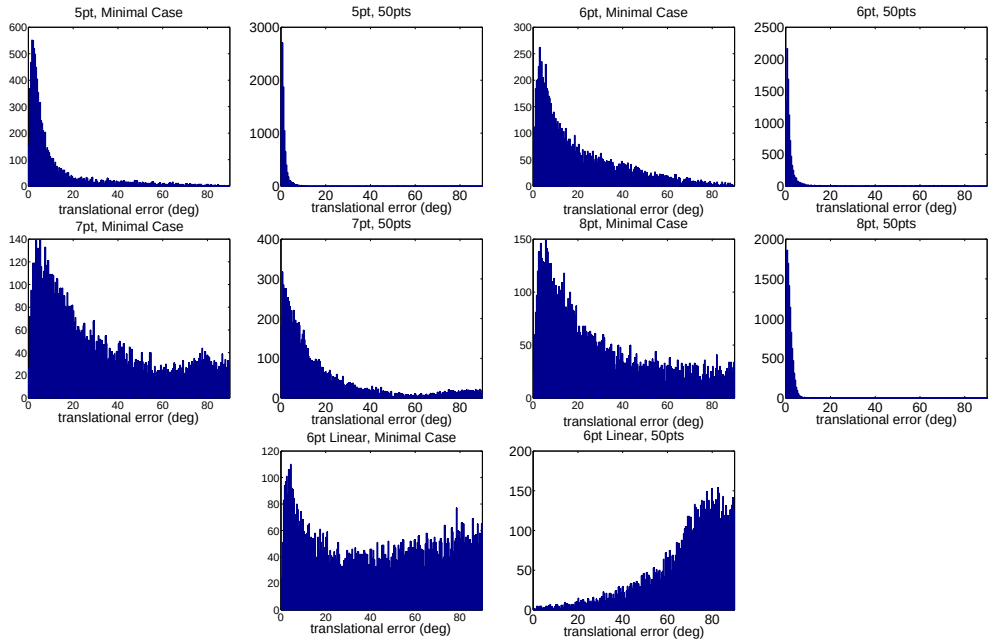

Fig. 6. Distribution of translational error over 10000 samples of the methods using minimal cases (left) and 50 points (right) for sideways motion. From top to bottom: five-point, six-point, seven-point, eight point, and linear six-point methods.

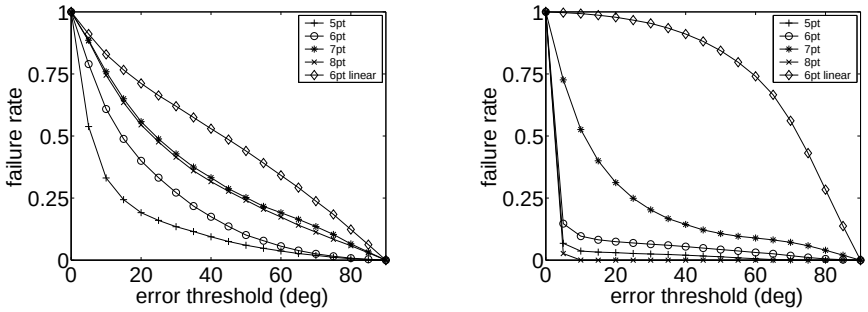

Fig. 7. Failure rate relative to a required error threshold for sideways motion. Left: minimal cases. Right: 50 points.

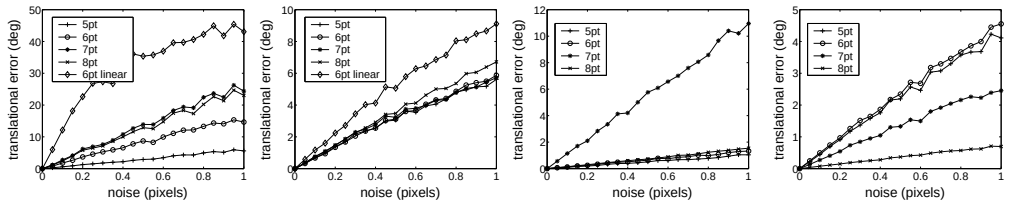

Fig. 8. Translational error relative to Gaussian noise equivalent to one pixel in a 2000×2000 pixel image. From left to right: sideways motion, minimal cases; forward motion, minimal cases; sideways motion, 50 points; forward motion, 50 points

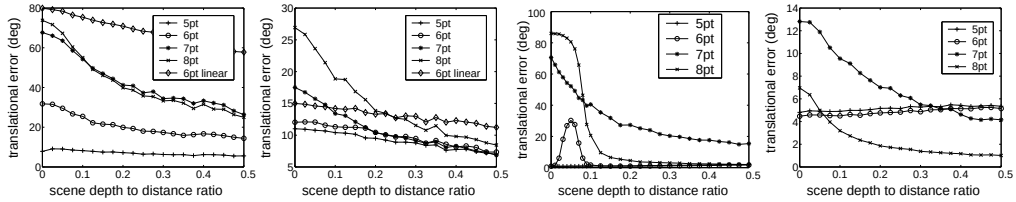

Fig. 9. Translational error relative to scene depth. From left to right: sideways motion, minimal cases; forward motion, minimal cases; sideways motion, 50 points; forward motion, 50 points. Note the instability of the six-point algorithm in the sideways motion 50 point case.

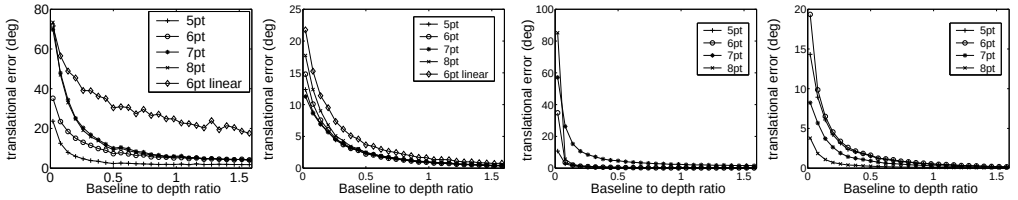

Fig. 10. Translational error relative to increasing baseline. From left to right: sideways motion, minimal cases; forward motion, minimal cases; sideways motion, 50 points; forward motion, 50 points

by the noise, although the seven-point and eight-point methods have a forward bias that becomes particularly apparent in the forward nonminimal case. As seen in Figure 9, the five-point and six-point algorithms maintain stability while the seven-point and eight-point algorithms fail in the degenerate case of a planar scene. Note that in the sideways nonminimal case, the six-point method seems to lack stability in the near-planar range, even though it works as expected for the true planar case. Figure 10 shows the results of increasing the baseline between the cameras in both sideways and forward motion. Once again, the five-point generally produces the least translational error, except in the forward nonminimal case. From Figure 10, we can see that the differences between the procedures become smaller for more favorable base to depth ratios which are common in photogrammetry. Note again that the average base to depth ratio used in the other experiments is about 0.1 to 1, while in many photogrammetric applications one tries to reach a ratio of 0.5 to 1.

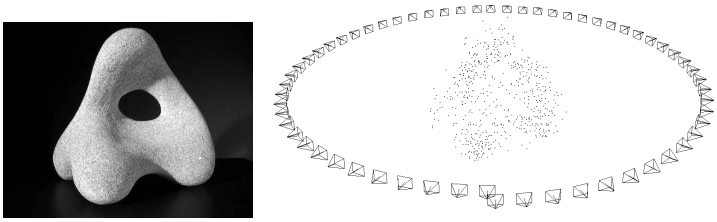

Fig. 11. Computed camera motion using a turntable sequence using the five-point method. No constraints were placed on the system enforcing the circular structure.

6.3 Real Example

Figure 11 shows an example of the capabilities of the five-point method. Here, we use the method within a RANSAC framework, followed by local bundle adjustment.

7 Conclusions

In the experimental section of this paper we have shown that the novel five-point method gives more consistent results than the other direct methods in all tested situations. The only other method that can handle both planar and non-planar cases is the six-point method. However, we found in the experimental section that the practical performance of the six-point method is not seamless into the planar case. We have also shown that the new method is numerically more stable than the previous five-point method.

Finally, we have shown the advantages of using the framework of algebraic geometry in order to get a systematic and straight-forward and easily explained solution to the five-point problem. The existing solver to the fivepoints problem was quite difficult to implement and in order to remove this barrier from trying the fivepoints algorithm we also provide a matlab version Stewénius (2005b).

References

Cox, D., Little, J., O'Shea, D., 1997. Ideals, Varieties, and Algorithms. Springer-Verlag New York.

Cox, D., Little, J., O'Shea, D., 1998. Using Algebraic Geometry, Springer-Verlag New York.

Demazure, M., 1988. Sur Deux Problemes de Reconstruction, Technical Report No 882, INRIA, Rocquencourt, France.

Faugeras, O., Maybank S., 1990. Motion from Point Matches: Multiplicity of Solutions, International Journal of Computer Vision, 4(3):225-246.

Faugeras, O., 1993. Three-Dimensional Computer Vision: a Geometric Viewpoint, MIT Press, ISBN 0-262-06158-9.

Fischler, M., Bolles, R., 1981. Random Sample Consensus: a Paradigm for Model Fitting with Application to Image Analysis and Automated Cartography, Commun. Assoc. Comp. Mach. 24:381-395.

Gruen, A., Huang, T. S., 2001. Calibration and Orientation of Cameras in Computer Vision, Springer-Verlag, ISBN 3-540-65283-3.

Hartley R., Zisserman, A., 2000. Multiple View Geometry in Computer Vision, Cambridge University Press, ISBN 0-521-62304-9.

Kruppa, E., 1913. Zur Ermittlung eines Objektes aus zwei Perspektiven mit Innerer Orientierung, Sitz.-Ber. Akad. Wiss., Wien, Math. Naturw. Kl., Abt. IIa., 122:1939-1948.

Longuet-Higgins, H., 1981. A Computer Algorithm for Reconstructing a Scene from Two Projections, Nature, 293(10):133-135.

Maybank, S., 1993. Theory of Reconstruction from Image Motion, Springer-Verlag, ISBN 3-540-55537-4.

Nistér, D., 2003. Preemptive RANSAC for Live Structure and Motion Estimation, Proc. IEEE International Conference on Computer Vision, pp. 199-206.

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.

Philip, J., 1998. Critical Point Configurations of the 5-, 6-, 7-, and 8-point Algorithms for Relative Orientation, TRITA-MAT-1998-MA-13.

Pizarro, O., Eustice, R., Singh, H., 2003. Relative Pose Estimation for Instrumented, Calibrated Platforms, VIIth Digital Image Computing: Techniques and Applications, Sydney.

McGlone, J.C., Mikhail, E.M., Bethel, J., Mullen, R. (Eds.), 2004. Manual of Photogrammetry, fifth ed. American Society of Photogrammetry and Remote Sensing ISBN 1-57083-071-1.

Stefanovic, P., 1973. Relative Orientation - a New Approach, I. T. C. Journal, 1973-3:417-448.

Stewénius, H., 2005. Gröbner Basis Methods for Minimal Problems in Computer Vision, Ph.D. Thesis, Lund, Sweden.

Stewénius, H. Matlab code for the for solving the fivepoint problem. `http://www.maths.lth.se/~stewe/FIVEPOINT/`.

Sturm, R. 1869. Das Problem der Projektivität und seine Anwendung auf die Flächen zweiten Grades, Math. Annalen 1, 533-573.

Triggs, B., McLauchlan, P., Hartley, R., Fitzgibbon, A., 2000. Bundle Adjustment - a Modern Synthesis, Springer Lecture Notes on Computer Science,

Springer Verlag, 1883:298-375.
von Sanden, H., 1908. Die Bestimmung der Kernpunkte in der Photogrammetrie. Doct. Thesis, Univ. Göttingen.
Wolf, P. R, 1983. Elements of Photogrammetry. McGraw-Hill, New York.