Available online at www.sciencedirect.com

ScienceDirect

Image and Vision Computing 26 (2008) 871–877

image AND vision COMPUTING

www.elsevier.com/locate/imavis

A minimal solution for relative pose with unknown focal length

Henrik Stewénius [a,*,1], David Nistér [b], Fredrik Kahl [c], Frederik Schaffalitzky [d,1]

[a] Center for Visualization and Virtual Environments University of Kentucky, USA
[b] Microsoft Live Labs, Microsoft Research, Redmond, USA
[c] Centre for Mathematical Sciences Lund University, Sweden
[d] Department of Engineering Science, University of Oxford, UK

Received 3 February 2006; accepted 9 October 2007

Abstract

Assume that we have two perspective images with known intrinsic parameters except for an unknown common focal length. It is a minimally constrained problem to find the relative orientation between the two images given six corresponding points. To this problem which to the best of our knowledge was unsolved we present an efficient solver. Through numerical experiments we demonstrate that the algorithm is correct, numerically stable and useful. The solutions are found through eigen-decomposition of a 15×15 matrix. The matrix itself is generated in closed form.

Keywords: Computer vision; Relative pose; Relative orientation; Camera calibration

1. Introduction

The task of computing a 3D reconstruction from a video sequence is central in computer vision. Different paradigms have been proposed for performing this task and the concept of RANSAC has been quite successful [8]. State-of-the-art real-time structure from motion uses the five-point method, e.g. [13], as a RANSAC engine. While this has proved to be efficient and stable, the cameras need to be pre-calibrated. For uncalibrated cameras, the seven-point method can be applied, e.g. [22], but it is not as stable due to projective degeneracies. We offer an attractive compromise having similar performance characteristics as the five-point method and still allowing for unknown focal lengths. By assuming constant and unknown focal lengths, but otherwise known intrinsics, a minimal problem arises for six points. This situation occurs if we assume the principal point to be in the middle of the image, that there is no skew and the aspect ratio is one. A detailed analysis of this case is given, showing the number of possible solutions, efficient ways to compute them and the stability of the solution with respect to measurement noise (Fig. 1).

Minimal case solvers have been built for a large number of camera models. The problem for two calibrated cameras and five points was first solved by Kruppa [11] who claimed 11 solutions (one too many) and the false root was then eliminated by Faugeras and Maybank [4]. A practical solution was given in [15] and improved in [13]. For three views and four points, the problem is not minimal but as it would be under-constrained with three points, the problem is still of interest and was solved in [14]. Minimal solutions also exist for uncalibrated perspective cameras [8], and uncalibrated affine cameras [9].

Given that the epipolar geometry has been computed in terms of the fundamental matrix, it is well known that it is possible to recover the focal length [8,21,10]. However, to the best of our knowledge the relative pose problem for minimal data is still unsolved. Here we present a solver for two cameras and six points. The solver constructs a 15×15 matrix in closed form. Solving the eigen-problem for this matrix gives the 15 (possibly complex) solutions to the relative pose problem.

[*] Corresponding author.
E-mail address: hstewenius@gmail.com (H. Stewénius).
[1] Present address: Google Switzerland.

0262-8856/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.imavis.2007.10.003

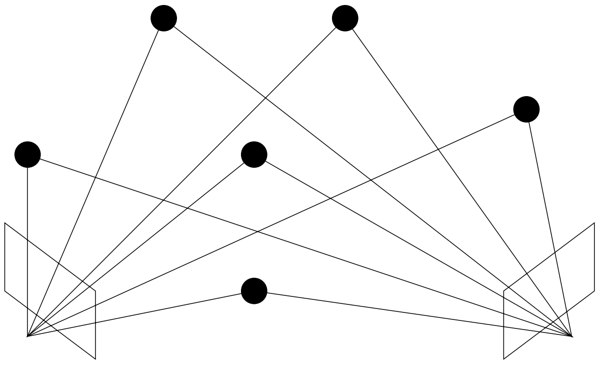

Fig. 1. The problem solved here: Relative orientation for two cameras, with a common but unknown focal length *f*, that see six unknown points.

We will first list the minimal cases for cameras with a common unknown focal length. Then the equations used will be introduced. The solver was found using Gröbner basis theory but this theory is not necessary to understand the solver. We will also give some numerical results.

2. Background

Suppose we are given m cameras, all calibrated except for a common unknown focal length and n corresponding image points. There are $6m + 3n + 1 - 7$ degrees of freedom (6 for each camera, 3 for each point, 1 for focal length and 7 for the unknown coordinate system) and $2mn$ equations, hence in total, there are $2mn + 6 - 6m - 3n$ excess constraints (Table 1).

The minimal case $(m, n) = (2, 6)$ will be solved here. The other possibility $(m, n) = (3, 4)$ is still an open problem.

2.1. Geometric constraints

The fundamental matrix F encodes the epipolar geometry of two views, and corresponding image points x and x' satisfy the coplanarity constraint

$$x^{\mathrm{T}}Fx' = 0. \tag{1}$$

Any rank-2 matrix is a possible fundamental matrix, i.e. we have the well known single cubic constraint, e.g. [8]:

Theorem 1. *If a real non-zero* 3×3 *matrix* F *is a fundamental matrix then*

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

Table 1
Number of excess constraints for m views and n points with unknown focal length f

m	n						
	1	2	3	4	5	6	7
1	−1	−2	−3	−4	−5	−6	−7
2	−5	−4	−3	−2	−1	**0**	1
3	−9	−6	−3	**0**	3	6	9
4	−13	−8	−3	2	7	12	17

An essential matrix has the additional property that the two non-zero singular values are equal. This leads to the following cubic constraints on the essential matrix, adapted from Maybank [12]:

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

$$2EE^{\mathrm{T}}E - \mathrm{tr}(EE^{\mathrm{T}})E = 0. \tag{3}$$

This constraint previously appeared in [19,3].

2.2. Gröbner bases

Here are only given some basic notions about algebraic geometry, the interested reader should consult [1,2].

The *ideal* generated by polynomials $f_1, \ldots, f_n \in \mathbb{C}[x_1, \ldots, x_n]$ is the set I of polynomials $g \in \mathbb{C}[x_1, \ldots, x_n]$ of the form:

$$g = \sum_{i=1}^{n} f_i p_i, \quad p_i \in \mathbb{C}[x_1, \ldots, x_n]. \tag{4}$$

We also say that the f_i *generate* the ideal I. A *Gröbner basis* of an ideal is a special set of generators, with the property that the leading term of every ideal element is divisible by the leading term of a generator. The notion of leading term is defined relative to a *monomial order*. The Gröbner basis exposes all leading terms of the ideal and leads to the useful notion of *remainder* with respect to (division by) the ideal. Gaussian elimination is a special case of *Buchberger's algorithm* which is a method for calculating a Gröbner basis from any generating set. Gröbner bases, monomial order and Buchberger's algorithm are explained in [1]. For ideals having a finite set of solutions ("zero-dimensional" ideals) the (vector space) dimension of the quotient ring $A = \mathbb{C}[x_1, \ldots, x_n]/I$ is also finite and the dimension equals the number of solutions, counted with multiplicity. Any polynomial f acts on the quotient ring A by multiplication ($f : g + I \mapsto fg + I$) and this is clearly a linear mapping from A to itself. A natural way to choose a (vector space) basis for A is to take all monomials that are not leading terms of any element of I. The action of a polynomial f is then described by a square matrix m_f called the *action matrix*. If R/I is finite and I has n solutions, the action matrix for multiplication by the polynomial f is computed by concatenating the $n \times 1$ vectors formed by taking the n elements in the monomial basis of R/I, reducing modulo $\mathrm{gb}(I)$ and representing them as $n \times 1$ vectors in the basis of R/I. The solutions to a zero-dimensional ideal can be read off directly from the left eigen-values and left eigen-vectors of appropriate action matrices [2].

The solver presented here was built by first computing a Gröbner Trace over $\mathbb{Z}_p$, please see [23,25]. The solver we build in this paper has some similarity with the matrix based Gröbner base algorithm proposed in [6]. A description of the machinery we used to build this solver is given in [20]. For the work in the finite field we used [7].

3. Solution procedure for $(m, n) = (2, 6)$

The inner calibration of the camera K is assumed to be $K = \mathrm{diag}([f \quad f \quad 1])$. With observations $\{x_i\}_{i=1}^{6}$ in the first image and $\{x'_i\}_{i=1}^{6}$ in the second image the epipolar constraint (1) gives six linear constraints on the fundamental matrix F. As a 3×3 matrix has 9 degrees of freedom this determines F up to 3 degrees of freedom, $F = l_0F_0 + l_1F_1 + l_2F_2$ for some scalars l_0, l_1, l_2. The fundamental matrix F can be computed only up to scale so we set $l_0 = 1$.

The fundamental matrix F must fulfill (2), that is, $\det(F) = \det(F_0 + F_1l_1 + F_2l_2) = 0$, which is a third order polynomial equation in (l_1, l_2). The matrix F can be transformed into an essential matrix by correcting for the intrinsic calibration, $E = K^{\mathrm{T}}FK$. Set $P = f^{-1}K$ then Eq. (3) is equivalent to

$$2PFPPF^{\mathrm{T}}PPFP - \mathrm{tr}(PFPPF^{\mathrm{T}}P)PFP = 0. \tag{5}$$

$$\iff 2FP^2F^{\mathrm{T}}P^2F - \mathrm{tr}(FP^2F^{\mathrm{T}}P^2)F = 0. \tag{6}$$

Notice that f^{-1} only appears in even powers in the above set of polynomial equations and hence one can set $p = f^{-2}$. This is a set of nine fifth order equations in (l_1, l_2, p).

The 10 equations, (2) and (6), can be written

$$MX = 0, \tag{7}$$

where M is a 10×33 matrix of scalars and X is a vector of monomials

$$\begin{aligned} X = [&l_1^3p^2, l_1^3p^1, l_1^3, l_1^2l_2^1p^2, l_1^1l_2^2p^2, l_2^3p^2, l_1^2l_2^1p^1, l_1^1l_2^2p^1, l_2^3p^1, \\ &\times l_1^2l_2^1, l_1^1l_2^1p^2, l_1^2p^2, l_2^2p^2, l_1^1p^3, l_1^1l_2^2, l_2^3, l_1^2p^1, l_2^1p^3, l_1^1l_2^1p^1, \\ &\times l_2^2p^1, l_1^1p^2, l_2^1p^2, p^3, l_1^2, l_1^1l_2^1, l_2^2, l_1^1p^1, l_2^1p^1, p^2, l_1^1, l_2^1, p^1, 1]^{\mathrm{T}}. \end{aligned}$$

This ordering of the monomials is not a monomial order but it is quite close to the GrevLex [1] monomial order. The reason for not ordering the monomials in GrevLex is that the computations are easier to implement this way. The computed Gröbner basis is the same as we would get using GrevLex. The elimination order is still the one belonging to the GrevLex order.

From this point on all polynomials will be represented by rows in $n \times 33$ matrices. Addition of polynomials is now addition of rows. Multiplying a polynomial with a scalar α corresponds to multiplying the corresponding row with α. Multiplying a polynomial with the monomial p is implemented by shifting elements. If any non-zero number is in a position that is not shifted it means that multiplication with p was impossible for this vector within this representation. By writing pM_i we mean the polynomial represented by row i in M multiplied by the monomial p. Row indexing starts at 1. The operations that we do are similar to generating new S-polynomials as described in [1], our reason for working on matrices instead of individual polynomials is that our approach permits row-pivoting in the elimination step.

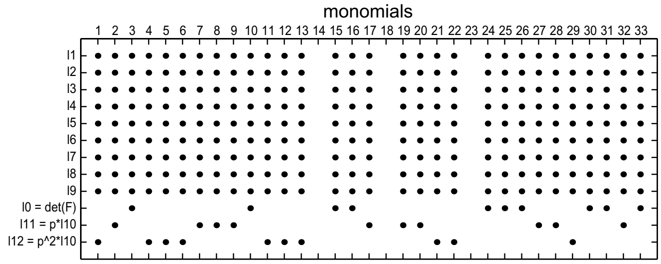

Fig. 2. Nine equations from Eq. (3) and $\det(F)$, $p\det(F)$ and $p^2\det(F)$.

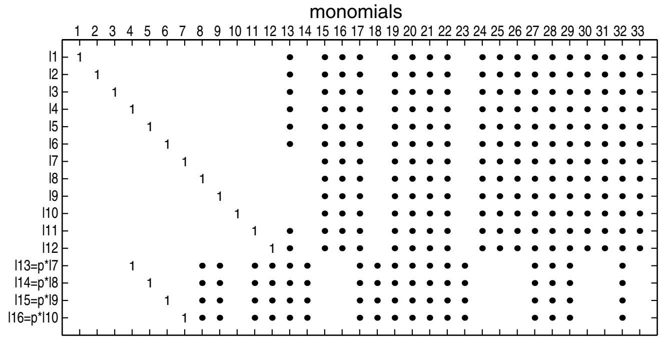

Fig. 3. The previous system after a Gauss–Jordan step and adding new equations based on multiples of the previous equations.

The rows representing $p\det(F)$ and $p^2\det(F)$ are added to the matrix M, or in matrix formulation pM_{10} and p^2M_{10}. This system of 12 rows is seen in Fig. 2. This system is reduced using Gauss–Jordan elimination (row operations) and the rows $pM_{(7,8,9,10)}$ are added. This new system is seen in Fig. 3. Again, the system is reduced and the rows $pM_{(8,9)}$ are added. This new system is seen in Fig. 4, for future reference we call this system M_3. The system is reduced one last time, and the system now represents a Gröbner basis. This system is seen in Fig. 5.

Given the Gröbner basis from the previous step it is now possible to compute the action matrix m_{l_2}, see [2], for multiplication by l_2 by taking the last 15 columns from rows 8, 9, 11, 13, 18, 10, 15 and 16 and changing the sign of all

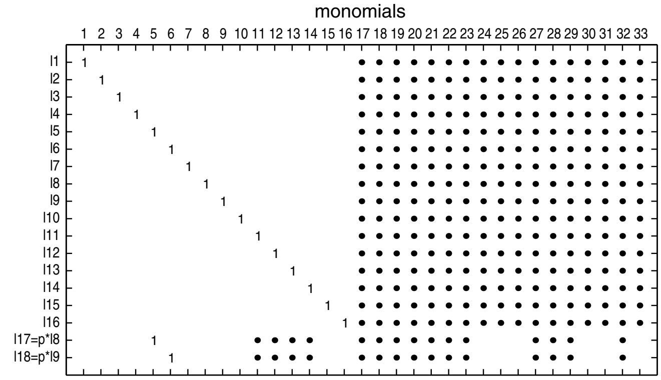

Fig. 4. The previous system after a Gauss–Jordan step and adding new equations based on multiples of the previous equations.

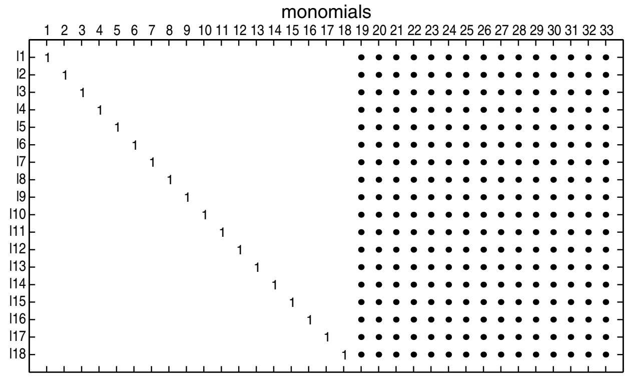

Fig. 5. Gauss–Jordan eliminated version of the previous system. This set of equations is a Gröbner basis.

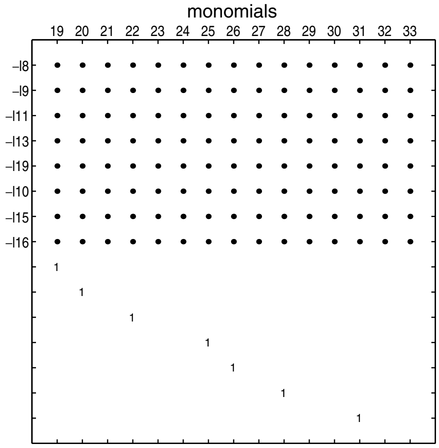

Fig. 6. Transpose of the Action-Matrix. Rows 1–8 are fetched from the previous matrix and rows 9–15 each contains a 1 and the other elements are 0.

coefficients and then putting ones in selected places, as described in Fig. 6.

Dividing each of the 15 eigen-vectors of the transposed action matrix by its last element and then selecting elements 12, 13 and 14 gives the solutions for (l_1, l_2, p). As there are 15 eigen-vectors there are 15 solutions. The fundamental matrices are then given by $F = F_0 + F_1 l_1 + F_2 l_2$, as $f^{-2} = p$ is known, the essential matrix can be computed and from the essential matrix the motion (R, t) can be computed.

4. Possible optimizations

4.1. Compute a monovariate polynomial

Solving the eigenvalue-problem is the most time-consuming part of the solver. Solving a monovariate polynomial equation using Sturm-sequences is faster.

By reordering the monomials the system can be written $[A \quad B]X$ where A is a 18×12 matrix (of scalars), B a matrix of polynomials in p with rows containing $[[2] \quad [2] \quad [2] \quad [3] \quad [3] \quad [3]]$ where $[n]$ is a polynomial in p of degree n, and X is the monomial vector

$$X = [l_1^3p^2, l_1^3p^1, l_1^3, l_1^2l_2^1p^2, l_1^1l_2^2p^2, l_2^3p^2, l_1^2l_2^1p^1, \ldots l_1^1l_2^2p^1, l_2^3p^1,$$
$$\times\, l_1^2l_2^1, l_1^1l_2^2, l_2^3, l_1^2, l_1^1l_2^1, l_2^2, l_1^1, l_1^0l_2^1, 1]^{\mathrm{T}}$$

Performing a Gauss–Jordan elimination on this system and extracting the last six rows gives a system

$$\underbrace{\begin{bmatrix} p^2 & 0 & [2] & [3] & [3] & [3] \\ p & 0 & [2] & [3] & [3] & [3] \\ 1 & 0 & [2] & [3] & [3] & [3] \\ 0 & p^2 & [2] & [3] & [3] & [3] \\ 0 & p & [2] & [3] & [3] & [3] \\ 0 & 1 & [2] & [3] & [3] & [3] \end{bmatrix}}_{C} \begin{bmatrix} l_1^2 \\ l_1 l_2 \\ l_2^2 \\ l_1 \\ l_2 \\ 1 \end{bmatrix}. \tag{8}$$

Existence of solutions for this system implies that $\det(C) = 0$. This is a polynomial of degree 15 in p. The solutions to this equation are solutions for p and can be used in Eq. (7) to get a 10×10 system of linear equations.

4.2. Efficient computation of M

When computing the system matrix M from the fundamental cubic Constraint (3) and the rank Constraint (2) an efficient way is to build a multiplication kernel for polynomials in known monomials, that is given vectors representing two polynomials the vector representing their product is computed. For monovariate polynomials this corresponds to convolution, for multivariate polynomials special code has to be written.

5. Numerical experiments

The error in the fundamental matrix is computed as $\min_i \|F \pm \widetilde{F}_i\|$ where F is the true fundamental matrix and $\{\widetilde{F}_i\}$ are the estimated fundamental matrices, all being normalized with Frobenius norm. The norm is the Frobenius norm.

5.1. Numerical precision of the solver

Fig. 7 shows the behavior for randomly chosen configurations of cameras and 3D scene points. There are two crit-

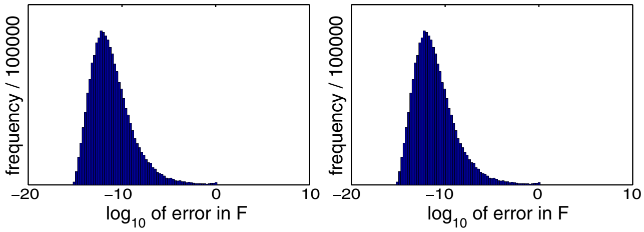

Fig. 7. $\log_{10}$ of error in determining f and F for perfect data.

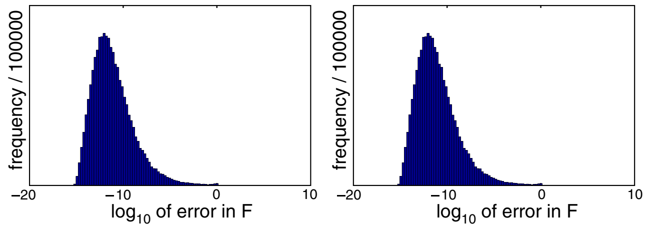

Fig. 8. $\log_{10}$ of error in determining f and F for perfect data when the axis intersect.

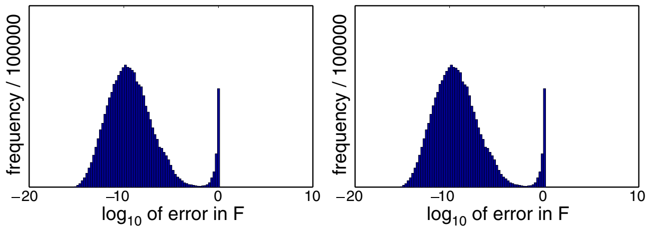

Fig. 9. $\log_{10}$ of error in determining f and F. The main axes of the cameras intersect and they have the same distance from principal point to intersection point (some noise added for stability).

ical configurations [21] for determining f from the fundamental matrix F:

- The main axes of the cameras intersect and both cameras have the same distance to the intersection point. Fig. 8 shows experimentally that if the distances from the cameras to the intersection point are not equal then f can be reliably estimated using our method. Fig. 9 shows that even when the distances are equal (and it is impossible to estimate f) out method still allows to estimate F.

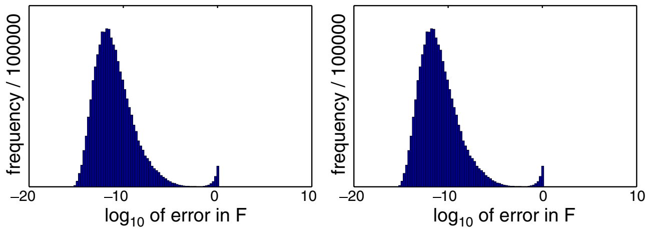

Fig. 10. $\log_{10}$ of error in determining f and F. The main axes of the cameras are parallel (some noise added for stability).

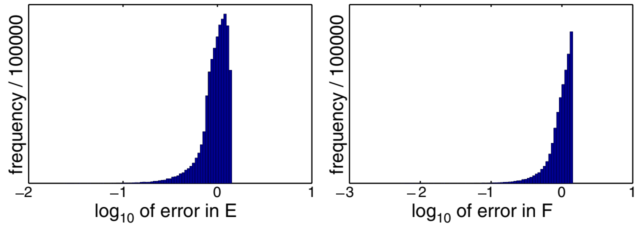

Fig. 11. $\log_{10}$ of error in determining E and F. All points on a plane.

- The main axes of the cameras are parallel. Similar to the other degenerate case, Fig. 10 shows that even though f can not be recovered it is again possible to compute F.

It also important to note that planar scenes are degenerate. This is demonstrated in Fig. 11. Here, neither f, E or F can be recovered by our method (the figure does not show a histogram for f because the range of errors is enormous whereas the range for E is not).

5.2. Stability of the solution compared to other methods

Our solver is compared with the five-point method of Nistér [13]. and the seven-point method. The comparison is slightly unfair since the five-point method is given the focal length.

In the numerical experiments we will use cameras with focal length $f = 1.0$ or 1.1. The world points are at depth 3.0 from camera one and are distributed with a normal distribution around this point. The standard deviation of the point cloud is 1.0, except when simulating a planar scene where we take one of the standard deviations to be much smaller, 0.01. Gaussian noise is added to image projections, assuming an image with a field of view of 53° and a resolution of 500 pixels. We do not test whether the 3D points are inside the field of view of the camera or not. The five-point method is implemented assuming that $f = 1.0$. All rotations indicate how much the second camera has been rotated relative to the first camera.

Figs. 12 and 13 show performance when the focal length is $f = 1.0$ (so the five-point method is given correct calibration). Out six-point method is narrowly beaten by the five-point method for sideways motion but performs much better for forward motion. This shows that, if the camera model is justified, it is always better to use the six-point solver than the seven-point method and generally no worse than the five-point solver.

For near-planar scenes (Fig. 14), the type of motion is important. For sideways motion our method is narrowly beaten by the five-point method but even the seven-point method is reasonable. For forward motion, only the five-point method is usable.

When the focal length is $f = 1.1$ (so the five-point solver has the wrong calibration) performance is similar to the case $f = 1.0$ for sideways motion (Fig. 15) but for forward motion the six-point solver comes out on top again (Fig. 16).

5.3. On the number of solutions

There are generally 15 solutions to the problem but some of these solutions can be complex or lead to negative values of $p = f^{-2}$. Fig. 17 shows the number of real solutions with positive values for f^2; these are the only solutions of practical interest.

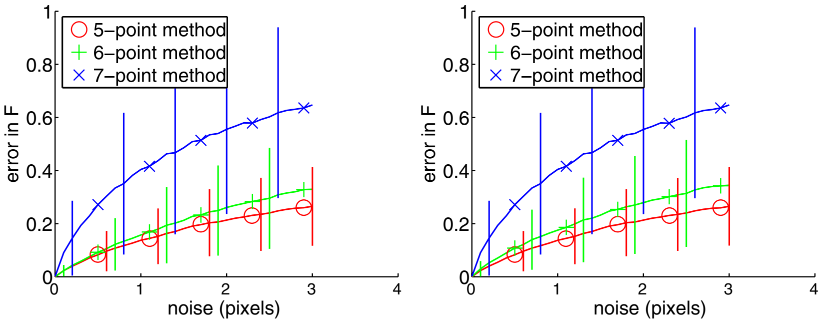

Fig. 12. Sideways motion, $f = 1.0$. Left: no rotation. Right: rotation 0.1 rad.

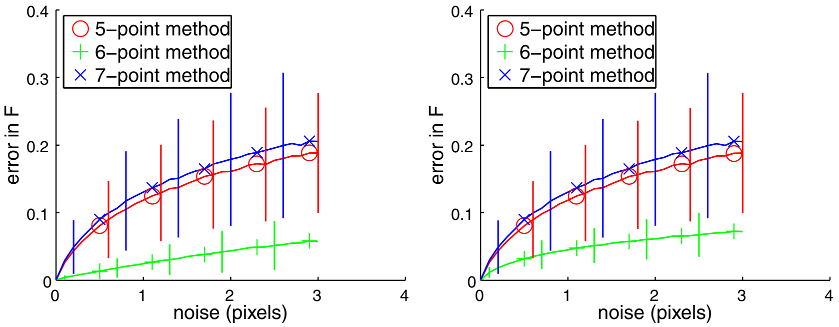

Fig. 13. Forward motion, $f = 1.0$. Left: no rotation. Right: rotation 0.1 rad.

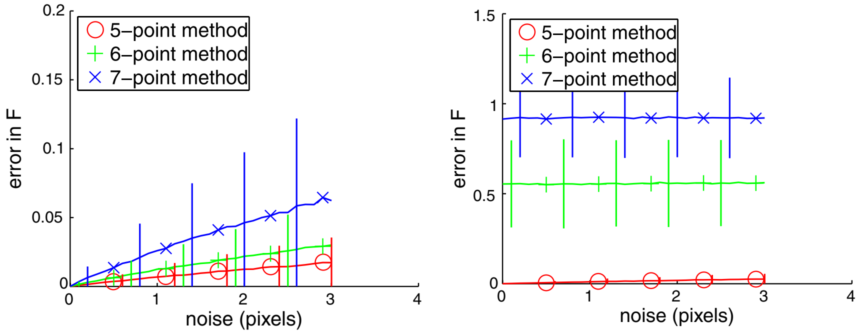

Fig. 14. Near-planar scene, no rotation. Left: sideways motion. Right: forward motion.

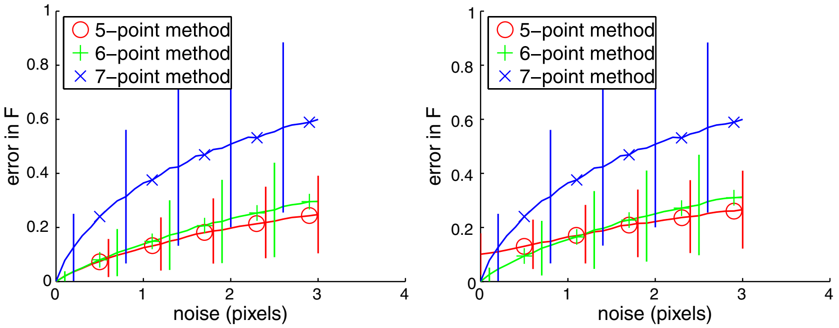

Fig. 15. Sideways motion, $f = 1.1$. Left: no rotation. Right: rotation 0.1 rad.

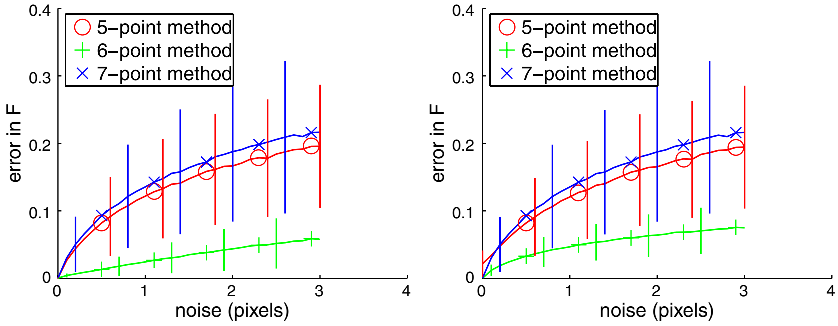

Fig. 16. Forward motion $f = 1.1$. Left: no rotation. Right: rotation 0.1 rad.

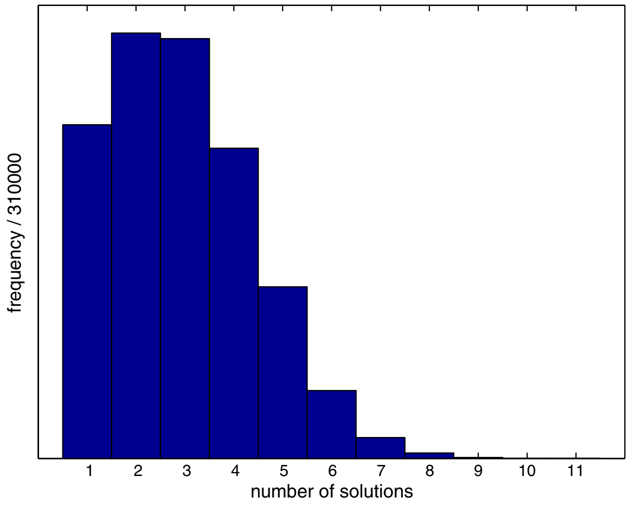

Fig. 17. The number of solutions (real with positive f^2) for the left-hand geometry in Fig. 12.

6. Conclusions

We have presented a solution to the minimal problem of six points in two views with unknown focal length. Using Gröbner basis techniques we show how the problem can be solved in an efficient manner, making it an attractive solution for semi-calibrated structure and motion computations.

Apart from computing the solutions to the eigen-problem, all the steps are in closed form and can be heavily optimized. For a fully optimized version the eigen-problem is likely to be the bottleneck, the current version takes about 2 ms.

We have also shown that the method gives very good stability under noise, competitive even with the five-point method and more stable than the seven-point method.

References

[1] D. Cox, J. Little, D. O'Shea, Ideals, Varieties, and Algorithms, Springer-Verlag, 1997.

[2] D. Cox, J. Little, D. O'Shea, Using Algebraic Geometry, Springer-Verlag, 1998.

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

[4] O. Faugeras, S. Maybank, Motion from point matches: multiplicity of solutions, International Journal of Computer Vision 4 (3) (1990) 225–246.

[6] J.C. Faugère, A new efficient algorithm for computing Gröbner bases (f4), Journal of Pure and Applied Algebra 139 (1–3) (1999) 61–88.

[7] D. Grayson, M. Stillman, Macaulay 2, <http://www.math.uiuc.edu/Macaulay2/>, 1993–2002.

[8] R. Hartley, A. Zisserman, Multiple View Geometry in Computer Vision, Cambridge University Press, 2000.

[9] J. Koenderink, A. van Doorn, Affine structure from motion, Journal of the Optical Society of America 8 (2) (1991) 377–385.

[10] F. Kahl, B. Triggs, Critical motions in euclidean structure from motion, IEEE Conference on Computer Vision and Pattern Recognition 2 (1999) 366–372.

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

[12] S. Maybank, Theory of Reconstruction from Image Motion, Springer-Verlag, 1993.

[13] D. Nistér, An efficient solution to the five-point relative pose problem, IEEE Transactions on Pattern Analysis and Machine Intelligence 26 (6) (2004) 756–770.

[14] D. Nistér, F. Schaffalitzky, What do four points in two calibrated images tell us about the epipoles? Proceedings of the 8th European Conference on Computer Vision 2 (2004) 41–57.

[15] J. Philip, A non-iterative algorithm for determining all essential matrices corresponding to five point pairs, Photogrammetric Record 15 (88) (1996) 589–599.

[19] P. Stefanovic, Relative orientation – a new approach, I. T. C. Journal 3 (1973) 417–448.

[20] H. Stewénius, Gröbner Basis Methods for Minimal Problems in Computer Vision, PhD Thesis, Lund University, 2005. <http://www.maths.lth.se/matematiklth/personal/stewe/THESIS/>.

[21] P. Sturm, On focal length calibration from two views, IEEE International Conference on Computer Vision and Pattern Recognition 2 (2001) 145–150.

[22] P. Torr, A. Fitzgibbon, A. Zisserman, The problem of degeneracy in structure and motion recovery from uncalibrated image sequences, International Journal of Computer Vision 32 (1) (1999) 27–44.

[23] C. Traverso, Gröbner trace algorithms, Lecture Notes in Computer Science, ISAAC'88: Proceedings of the International Symposium ISSAC'88 on Symbolic and Algebraic Computation, 1989.

[25] F. Winkler, A p-adic approach to the computation of Gröbnerbases, Journal of Symbolic Computation 6, 287–304.