

Fast RANSAC hypothesis generation for essential matrix estimation

Tom Botterill

Department of Computer Science,
University of Canterbury, Christchurch, NZ.
Email: tom.botterill@grcnz.com.

Steven Mills

Areograph Ltd.,
90 Crawford St.,
Dunedin, NZ.

Richard Green

Department of Computer Science,
University of Canterbury, Christchurch, NZ.

Abstract—The RANSAC framework is often used to estimate the relative pose of two cameras from outlier-contaminated point correspondences, via the essential matrix, however this is computationally expensive due to the cost of computing essential matrices from many sets of five to seven correspondences. The leading contemporary 5-point solver (Nister, 2004) is slow because of the expensive linear algebra decompositions and polynomial solve which are required. To avoid these costs we propose to use Levenberg-Marquardt optimisation on a manifold to find a subset of the compatible essential matrices. The proposed algorithm finds essential matrices at a higher rate than closed-form approaches, and reduces the time needed to find relative poses using RANSAC by 25%.

The second contribution of this paper is to apply the optimisations used in 5-point solvers to the classic 7-point algorithm. RANSAC using the optimised 7-point algorithm is considerably faster than 5-point RANSAC (unless planar point configurations are common), despite the increased number of iterations necessary.

I. INTRODUCTION

A common task in computer vision is to compute the relative pose of two calibrated cameras, so that the scene’s 3D structure can be recovered and the images can be aligned. Relative pose can be estimated via the essential matrix, \mathbf{E} , a 3×3 matrix encoding the rotation and translation between two cameras [1]. \mathbf{E} can be computed from matches between features visible in both images (“correspondences”), however feature matching algorithms frequently match features incorrectly, introducing outlier correspondences. To estimate \mathbf{E} while simultaneously identifying outliers, the robust RANSAC framework is often used.

RANSAC (Random Sample Consensus [2]) works by repeatedly choosing small random subsets of correspondences (‘hypothesis sets’), fitting an essential matrix to each hypothesis set, then counting the total number of correspondences compatible with each essential matrix. Eventually an essential matrix compatible with many correspondences will be found, usually because the hypothesis set contained only inliers. The set of compatible correspondences consists mostly of inliers, and a robust global optimisation can then be applied to refine the inlier set and to obtain an accurate relative pose estimate. When outlier rates are high, RANSAC becomes computationally expensive, due to the large number of iterations needed before an all-inlier hypothesis set is found.

The minimum number of point-correspondences necessary to determine \mathbf{E} is five, but a simpler seven-point algorithm is frequently used (e.g. by [3], [4], [5]). To minimise the number of iterations needed by RANSAC to find an all-inlier hypothesis set, five point hypothesis sets should ideally be used, as the probability of these being contaminated by outliers is lower than for seven point sets, however the fastest contemporary 5-point solver, by Nister [6], is still computationally expensive, due to large linear algebra operations and a tenth-order polynomial solver. To avoid these costs we propose to sample from the space of possible solutions by repeatedly using the Levenberg-Marquardt (LM) optimisation algorithm from many different start points. Firstly, a four-dimensional basis containing up to ten essential matrices compatible with the five point correspondences is computed by modified Gram-Schmidt (MGS) orthonormalisation. Secondly, the LM algorithm is applied repeatedly from random start points on a manifold of linear combinations of these basis vectors to find a subset of the compatible essential matrices. This approach generates RANSAC hypotheses at a higher rate than an optimised implementation of Nister’s 5-point algorithm, and reduces the cost of RANSAC by typically 25%. The MGS orthonormalisation is also applied to the classic 7-point algorithm for \mathbf{E} ; RANSAC using this optimised 7-point algorithm is considerably faster than 5-point RANSAC, despite the increased number of iterations necessary, however is not suitable for environments where planar point configurations are common.

This paper is organised as follows: the following section introduces the essential matrix and its properties; Section III introduces the LM algorithm in the context of solving simultaneous equations on a manifold; Section IV reviews contemporary approaches for estimating essential matrices from point-correspondences; Section V describes how we propose to compute essential matrices efficiently using a gradient-descent procedure; Section VI validates the new approach against contemporary 5-point algorithms and an optimised 7-point algorithm; and Section VII presents our conclusions.

II. THE ESSENTIAL MATRIX

This section provides an overview of the important properties of the essential matrix, which is analysed in more detail in Hartley and Zisserman, Chapter 9 [7]. The essential matrix, \mathbf{E} ,

is a 3×3 matrix encoding the rotation and translation direction between two views. If the rotation is expressed as a matrix, \mathbf{R} , and the translation as a vector, \mathbf{t} , then \mathbf{E} is defined by

$$\mathbf{E} = [\mathbf{t}]_{\times} \mathbf{R} \quad (1)$$

where $[\mathbf{t}]_{\times}$ is the matrix-representation of the vector cross-product, with the property that $[\mathbf{t}]_{\times} \mathbf{x} \equiv \mathbf{t} \times \mathbf{x}$. As $[\mathbf{t}]_{\times}$ has rank 2 in general, \mathbf{E} also has rank 2. From two images alone, the length of \mathbf{t} cannot be determined, therefore \mathbf{E} is only determined up to scale.

If a 3D point \mathbf{X} is viewed in two images at locations \mathbf{x} and \mathbf{x}' (where \mathbf{x}, \mathbf{x}' are calibrated homogeneous image coordinates), then \mathbf{E} has the property that:

$$\mathbf{x}'^T \mathbf{E} \mathbf{x} = 0 \quad (2)$$

Expanding this equation gives a single linear constraint in the nine elements of \mathbf{E} for every correspondence. From N correspondences, these equations can be stacked to form a $9 \times N$ matrix, the singular value decomposition (SVD) of which gives a $(9 - N)$ -dimensional basis for the space in which \mathbf{E} lies [7], [8]. The points within this vector space which are essential matrices are those which can be decomposed into a rotation and a translation. \mathbf{E} can be decomposed in this way when its SVD has the form:

$$\mathbf{E} = \mathbf{U} \begin{pmatrix} s & 0 & 0 \\ 0 & s & 0 \\ 0 & 0 & 0 \end{pmatrix} \mathbf{V}^T \quad (3)$$

where \mathbf{U}, \mathbf{V} are orthonormal matrices. Due to the sign and scale ambiguity in \mathbf{E} , \mathbf{U}, \mathbf{V} can always be chosen to be rotation matrices, and s can be chosen to be 1.

Equation 3 is equivalent to

$$\mathbf{E} \mathbf{E}^T \mathbf{E} - \frac{1}{2} \text{trace}(\mathbf{E} \mathbf{E}^T) \mathbf{E} = \mathbf{0}, \quad (4)$$

which provides an efficient test of whether a matrix is approximately an essential matrix.

\mathbf{E} has five degrees of freedom, from the 3D rotation and 2D translation direction, and can be determined from five or more correspondences. Given five correspondences, up to ten (although typically about four) different essential matrices are possible [6]. Given seven correspondences, up to 3 are possible.

In summary, a basis containing \mathbf{E} can be computed from point correspondences via SVD, and any linear combination of these basis vectors satisfying Equation 3 is an essential matrix.

III. PRELIMINARIES

This section describes the Levenberg-Marquardt algorithm for nonlinear least-squares optimisation, then outlines how it can be applied on a manifold to solve a system of nonlinear equations.

A. The Levenberg-Marquardt gradient descent algorithm

The Levenberg-Marquardt (LM) algorithm, described in detail in Appendix 6 of Hartley and Zisserman [7], is a gradient-descent optimisation algorithm for minimising functions which are a sum of squared terms, i.e. to find \mathbf{x} minimising $\|\mathbf{f}(\mathbf{x})\|$. The LM algorithm is based on the Gauss-Newton algorithm, but uses an adaptive damping parameter which prevents instability away from minima, while allowing fast convergence close to minima.

When the system of equations being solved has an exact solution (i.e. when there are N equations in N unknowns, and we seek solutions \mathbf{x}^* where $\mathbf{f}(\mathbf{x}^*) = \mathbf{0}$), Newton-Raphson optimisation can be used. Newton-Raphson is closely related to Gauss-Newton optimisation of $\|\mathbf{f}\|^2$; both use a combination of \mathbf{f} and the Jacobian of \mathbf{f} to compute a step taking \mathbf{x} to where the minimum is predicted to be. Newton-Raphson is notoriously unstable away from roots [9], so many damped Newton-Raphson approaches have been proposed. One suitable approach is to apply the LM algorithm, which is stable away from minima, and which converges at a similar rate near to minima. LM iterations have approximately the same cost as Newton-Raphson iterations: \mathbf{f} and the Jacobian of \mathbf{f} are required and an $N \times N$ linear system is solved on each iteration.

B. \mathbf{E} as a point on a manifold

Gradient descent optimisation algorithms like LM operate in \mathbb{R}^n , however it is often convenient to constrain the parameter sets to a manifold embedded in \mathbb{R}^n . Manifolds which can be locally approximated by a subspace of \mathbb{R}^m are differential manifolds. To use the LM algorithm to solve an equation where the solution \mathbf{x}^* lies on a manifold, the objective function $\mathbf{f}(\mathbf{x})$ is reparametrised on each iteration in terms of a basis of vectors tangent to the manifold at \mathbf{x} .

One convenient parametrisation of \mathbf{E} is in terms of its corresponding rotation and translation direction (Equation 1). Both the space of 3D rotations and the space of translation directions are differential manifolds; this parametrisation was used by Ma et al. [10], who represent rotations as rotation matrices (a Lie group), which are projected to and from the vector space of skew-symmetric matrices (their corresponding Lie algebra) on each iteration. These projection operations can considerably increase the cost of each LM iteration however.

IV. ALGORITHMS FOR COMPUTING \mathbf{E}

This section reviews contemporary approaches for estimating \mathbf{E} . Firstly, ‘‘closed form’’ algorithms for estimating all essential matrices compatible with sets of five to eight points are reviewed, then previous optimisation algorithms, which aim to estimate the most likely \mathbf{E} from many noisy correspondences, are described.

A. Computing \mathbf{E} from 7 or 8 correspondences

Given eight or more correspondences, a matrix \mathbf{F} minimising the sum

$$\sum_i (\mathbf{x}'_i{}^T \mathbf{F} \mathbf{x}_i)^2 \quad (5)$$

can be found by SVD LonguetHiggins-1981. If the correspondences are inliers then \mathbf{F} is usually approximately equal to \mathbf{E} (and is a suitable starting point for subsequent optimisation) however \mathbf{F} is often a poor estimate of \mathbf{E} [4], as Equation 5 is a biased measure of errors due to point localisation.

From seven correspondences, a basis of two vectors for \mathbf{F} can be computed. A set of either one or three linear combinations of these two basis vectors satisfy $\det \mathbf{F} = 0$; these linear combinations can be found by solving a cubic equation in the basis elements. The subset of these matrices which are also approximately essential matrices can be found efficiently by applying Equation 4. Chum [4] reports that the 7-point algorithm produces more accurate hypotheses than the 8-point algorithm.

B. Computing \mathbf{E} from 5 correspondences

The seven- and eight-point algorithms for \mathbf{E} are relatively fast, however ideally for RANSAC minimal subsets should be used, which for essential matrices is five correspondences.

Given five correspondences, four basis vectors satisfying Equation 2 can be computed by SVD. All linear combinations of these basis vectors satisfying Equation 4 are essential matrices. Equation 4 provides nine cubic constraints in the elements of \mathbf{E} ; the methods of Nister [6], and Stewenius et al. [11] both work by solving these nine equations.

Stewenius et al. solves these equations by constructing a Gröbner basis, after which the solutions are derived from the real eigenvectors of a 10×10 matrix. Nister converts the cubic equations to a single tenth-order equation, each real root of which corresponds to an essential matrix. Both methods first compute the coefficients of the system of polynomials to be solved by setting up and solving a 10-dimensional linear system.

C. Gradient-descent methods for computing \mathbf{E}

When outlier rates are low, or once most outliers have been removed by RANSAC, remaining errors in correspondences are from errors in image feature localisation, and can be assumed Gaussian and independent [7]. A variety of optimisation procedures have been proposed for fitting an accurate relative pose to these correspondences.

The ‘‘Gold-standard’’ for relative pose computation is to find R , \mathbf{t} , and the corresponding 3D structure which minimises the total squared reprojection error (the distance between the location of points projected into the image, and their measured locations). Hartley and Zisserman [7] recommend minimising this sum-of-squared errors using the LM algorithm, and several formulations of this minimisation have been proposed.

Ma et al. [10] refine \mathbf{E} by using a Newton-based method on the product of manifolds for the space of 3D rotations, and the space of unit translation vectors (equivalent to the manifolds described in Section III-B), however in general the method only converges locally due to the Newton method’s instability away from minima. Helmke et al. [12] propose a different manifold, in which \mathbf{E} is parametrised in terms of the

two rotation matrices in its SVD (Equation 3). Each Gauss-Newton iteration is computationally less expensive (having lower renormalisation costs) than Ma et al.’s approach, however again the method only converges locally in general. Both methods use the 8-point algorithm to initialise the optimisation.

Batra et al. [13] avoid working on a manifold explicitly and instead use a constrained optimisation toolbox to solve a set of polynomials similar to Equation 4, subject to a nonlinear constraint on the parameters. The method is demonstrated to find solutions to the 5-point problem from random starting points, but is not computationally efficient.

[14] find \mathbf{E} by first finding the fundamental matrix which minimises a more complex robust error function. As the error function is only piecewise differentiable, the Gauss-Newton Hessian estimate is not suitable, so the Downhill Simplex method is used: a gradient descent method which does not require second-derivatives.

These gradient descent methods estimate essential matrices from many noisy point correspondences. Minimising an error function related to point localisation error is important to give an accurate solution, but convergence from distant start points is less important. In this paper, LM optimisation is applied to the problem of estimating \mathbf{E} from minimal sets of five correspondences for RANSAC hypothesis generation, as an alternative to contemporary ‘closed form’ 5-point solvers. It is important that each LM iteration is computationally cheap, and that solutions can be found without initial estimates being available, but unlike other gradient-descent based algorithms, the essential matrices found are exact solutions, so the function which is minimised does not have to relate to reprojection error. The following section describes a parametrisation of \mathbf{E} meeting these requirements.

V. FINDING \mathbf{E} IN TERMS OF BASIS VECTORS USING LM

To estimate \mathbf{E} efficiently from sets of five correspondences, we first compute the 4D basis in which essential matrices compatible with the five point correspondences lie, then apply the LM algorithm on a 3D manifold within the vector space to solve a minimal set of three equations satisfied by \mathbf{E} .

In Section II, we described how a four-dimensional basis $\mathbb{B} = \{\mathbf{b}_1, \dots, \mathbf{b}_4\}$ in which \mathbf{E} lies can be computed from five point correspondences. All linear combinations of this basis which satisfy Equation 4, and which have determinant zero, are essential matrices compatible with these five point correspondences. To find these essential matrices we use LM optimisation to find 4D vectors of coefficients, \mathbf{p} , so that $\mathbf{E}(\mathbf{p}) = \sum_{i=1}^4 p_i \mathbf{b}_i$ satisfies Equation 4 and has determinant zero.

As \mathbf{E} is only defined up to scale, the optimisation is constrained so that \mathbf{p} has length 1. The set of 4D parameter vectors with $\|\mathbf{p}\| \equiv 1$ is the unit sphere in 4D, \mathbb{S}^3 , which is a differential manifold.

As \mathbb{B} is orthonormal, $\mathbf{E}(\mathbf{p}) = \sum_{i=1}^4 p_i \mathbf{b}_i$ also has Frobenius norm 1 (i.e. $\text{trace}(\mathbf{E}\mathbf{E}^T) \equiv 1$). By constraining the search to this manifold, the constraint that the determinant is zero is

unnecessary: if $\det \mathbf{E} \neq 0$ then $\mathbf{E}\mathbf{E}^T\mathbf{E} - \frac{1}{2}\text{trace}(\mathbf{E}\mathbf{E}^T)\mathbf{E} = 0$ only when $\mathbf{E}\mathbf{E}^T = \frac{1}{2}\mathbf{I}$, which is false because $\text{trace}(\mathbf{E}\mathbf{E}^T) \equiv 1$.

We find $\mathbf{E}(\mathbf{p})$ minimising $\|\mathbf{f}(\mathbf{E}(\mathbf{p}))\|^2$, where

$$\mathbf{f}(\mathbf{E}) = \text{vec} \left(\mathbf{E}\mathbf{E}^T\mathbf{E} - \frac{1}{2}\text{trace}(\mathbf{E}\mathbf{E}^T)\mathbf{E} \right), \quad (6)$$

where vec denotes the nine elements of the 3×3 matrix interpreted as a 9D vector.

At each iteration, a basis of three vectors tangent to \mathbb{S}^3 at \mathbf{p} , $\{\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3\}$ is computed. Each 4D tangent vector \mathbf{a}_j corresponds to a 3×3 matrix \mathbf{A}_j , which is the linear combination of the basis elements \mathbb{B} with coefficients \mathbf{a}_j . When the parameter set \mathbf{p} is adjusted by a small amount δ in the direction \mathbf{a}_j , $\mathbf{E}(\mathbf{p} + \delta\mathbf{a}) \approx \mathbf{E}(\mathbf{p}) + \delta\mathbf{A}_j$. The vector of derivatives of the components of \mathbf{f} in the direction of tangent vector \mathbf{a}_j is given by

$$J_{\text{col}j} = \frac{\partial}{\partial \mathbf{a}_j} \mathbf{f} = \text{vec}[(\mathbf{A}_j\mathbf{E}^T + \mathbf{E}\mathbf{A}_j^T - \text{trace}(\mathbf{E}\mathbf{A}_j^T)\mathbf{I})\mathbf{E} + (\mathbf{E}\mathbf{E}^T - \frac{1}{2}\text{trace}(\mathbf{E}\mathbf{E}^T)\mathbf{I})\mathbf{A}_j] \quad (7)$$

LM is now applied, with $\{\mathbf{a}_j\}$, \mathbf{p} renormalised, and $\{\mathbf{A}_j\}$ recomputed at each iteration.

\mathbf{r} has nine elements, defining nine constraints on \mathbf{E} . The manifold \mathbb{S}^3 has only three degrees of freedom however, and any three equations from \mathbf{r} are zero at only a finite set of points on the manifold, some of which correspond to essential matrices. For efficiency, the LM optimisation uses just the first three elements of \mathbf{r} . These three equations have additional solutions, which are eliminated by checking Equation 4. In simulated data, 80% of minima where $\mathbf{r}_{1..3} = \mathbf{0}$ correspond to essential matrices; others are rejected.

A. Important optimisations

This parametrisation of \mathbf{E} has many false minima, and the same solutions are often found multiple times. Optimisation is stopped if the solution approaches one which has already been found, or if the rate of descent is so low that the residuals are unlikely to converge to zero.

When optimising on a manifold, parameter and basis vectors are renormalised after each update, e.g. $q \leftarrow \frac{\mathbf{p}}{\|\mathbf{p}\|}$. This normalisation is relatively expensive if the norm is computed using a square root, however after the first iteration or two, updates are small and the binomial approximation $\frac{1}{\|\mathbf{p}\|} \approx (\frac{3}{2} - \frac{1}{2}\|\mathbf{p}\|^2)$ is sufficiently accurate that the square root and division can be eliminated.

B. Implementation details

Our proposed algorithm (labelled *LM Basis*) together with the contemporary 5-point solvers by Nister et al. [6] and Stewenius et al. [11] are each implemented in C++ using the Eigen 3 matrix library [15] and compiled using gcc. Eigen is designed to generate efficient code for the small fixed-size matrix expressions common to all 5-point algorithms, and includes efficient implementations of popular linear algebra routines.

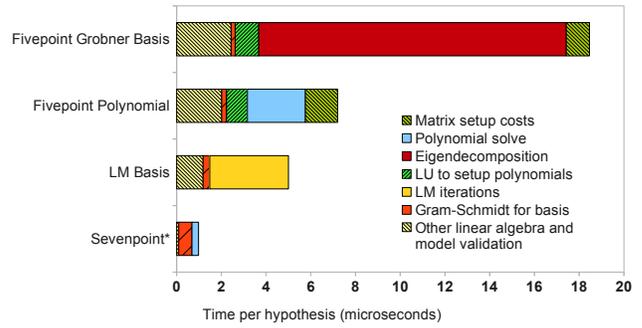


Fig. 1. Time per-hypothesis for different 5-point solvers. *The time for Sevenpoint is per hypothesis set, as the average number of solutions depends on the inlier rate.

Our C++ implementations of Nister’s and Stewenius’ 5-point solvers are based on the MATLAB code available from [16]. Significant costs common to both approaches are an LU decomposition to rearrange basis vectors into a system of polynomials, and matrix setup costs. Stewenius’ approach (labelled *Fivepoint Gröbner Basis*) computes the eigendecomposition of a 10×10 action matrix, for which we use Eigen’s Francis-QR implementation [9]. Nister’s approach (labelled *Fivepoint Polynomial*) solves a 10th order polynomial instead. For this we use the Jenkins-Traub [9] implementation from [17]; for the 10th order polynomials encountered this is considerably faster than the QR-based polynomial solver in Eigen.

To find bases for \mathbf{E} , a SVD is often used, e.g. by [8], [7]. For problems this small (9×9) however, a faster Modified Gram-Schmidt (MGS) orthonormalisation is considerably faster, and has comparable numerical accuracy [18].

All times given refer to code running on a single core of a 2.93GHz Intel i7 processor. Source code for all implementations is available on the author’s web page [19].

C. Discussion

An LM-based scheme for finding essential matrices from sets of five correspondences has been described. Unlike contemporary 5-point solvers, which solve a small number of large (10D) linear or polynomial systems, the proposed scheme use a large number of comparatively cheap LM iterations, where systems sized 3×3 are solved on each iteration. The LM algorithm is used so that optimisation can start far away from minima, and \mathbf{E} is parametrised in terms of a linear combination of basis vectors, so that each LM iteration is computationally inexpensive.

The authors also investigated 5-point solvers operating on the 5D manifold of 3D rotations and translation directions; this approach was found to be considerably more expensive due to the 5D system which was set up and solved on each iteration. Alternatively, when \mathbf{E} was parametrised directly in terms of \mathbb{R}^5 , convergence was considerably slower.

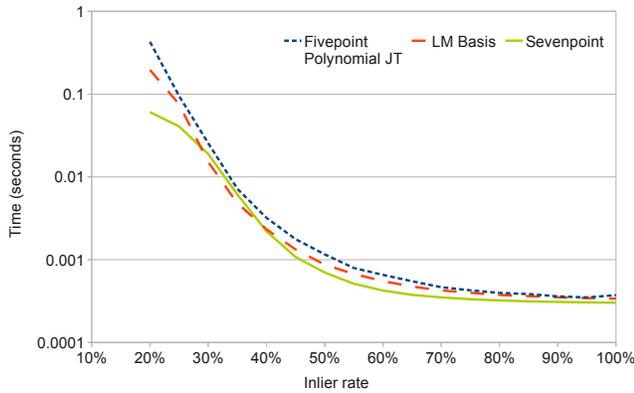


Fig. 2. Time taken to find \mathbf{E} in RANSAC with different inlier rates: *LM Basis* is up to 25% faster than the *Fivepoint Polynomial*, but an optimised 7-point solver is generally faster.

D. Optimised 7-point algorithm

The MGS orthonormalisation used to compute a 4D basis from five points is also ideal for the 7-point algorithm, where a 2D basis is computed from 7 point correspondences. A cubic equation is then solved directly to find linear combinations of these basis vectors with determinant zero. In the presence of outliers these matrices are often not essential matrices, and even with all-inlier hypothesis sets point localisation errors result in matrices which are not exactly essential matrices. Equation 4 is used to identify which of these matrices are approximately essential matrices, and Equation 3 is then applied to find the closest (by L2 norm [7]) essential matrix.

VI. RESULTS

Firstly, we compare the relative speeds of the different approaches at hypothesis generation. Each 5-point solver is evaluated on sets of five randomly generated correspondences. The mean time per-hypothesis for each algorithm is shown in Figure 1. The *LM Basis* approach outperforms all other 5-point algorithms in terms of the rate at which hypotheses are generated. Table I shows that while *LM Basis* often finds the same solutions multiple times, and also finds many false minima, these can be detected in typically just one or two iterations. Secondly, Nister’s 5-point algorithm and the *LM*

| Algorithm | Solutions | False minima | Repeated solutions |
|----------------------|-----------|--------------|--------------------|
| <i>LM Basis</i> | 2.4 (5.7) | 3.8 (3.7) | 3.9 (1.3) |
| Fivepoint Polynomial | 3.9 | - | - |

TABLE I
AVERAGE NUMBERS OF SOLUTIONS FOUND (AND ITERATIONS REQUIRED) PER FIVE-POINT SET. *LM BASIS* IS STARTED FROM 10 RANDOM POINTS.

Basis 5-point algorithm are tested in RANSAC, to verify that *LM Basis* solutions are as good as those from a contemporary 5-point solver. In addition, these 5-point algorithms are compared to the optimised 7-point solver. The probability of a 7-point hypothesis set being outlier-contaminated is considerably higher than for 5-point hypothesis sets, and the 7-point algorithm also fails when point sets are approximately planar (so

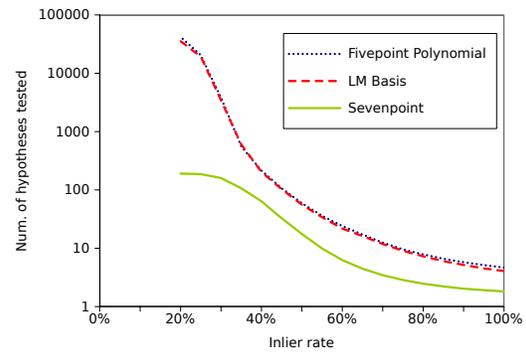


Fig. 3. The number of hypotheses generated per RANSAC run is the same for *LM Basis* and *Fivepoint Polynomial*. *Sevenpoint* generates fewer hypotheses per hypothesis set (often none for outlier-contaminated hypothesis sets).

should not be used when environments contain many planar point configurations), however the 7-point algorithm has the advantage of being considerably faster.

In our first simulated RANSAC setup, 200 3D points are generated randomly, projected into two 640 pixel-wide images with a FOV of 1 radian, then simulated localisation errors with mean 1 pixel are added. To simulate outliers, points are mismatched at random. The classic RANSAC algorithm is used, terminating only when an inlier set containing 20% of correspondences is found (this termination condition ensures every run is successful, and that RANSAC is stopped as soon as a large inlier set is found). Figure 2 shows that the average time taken to find an inlier set is consistently lower with *LM Basis* than the contemporary 5-point algorithm. Figure 3 shows that on average the same number of hypotheses are generated by the two 5-point solvers on each RANSAC run, indicating that the subset of hypotheses found by *LM Basis* are just as good as the complete sets of hypotheses found by a closed-form 5-point solver. In this experiment rotations of up to 0.75 radians, and translations of up to half the mean point depth are generated at random. Similar results are obtained for the cases of pure rotation, forwards and sideways motion, a narrow FOV, and planar scenes.

Figure 2 also shows that the 7-point algorithm outperforms the 5-point algorithms except where outlier rates are high. Each run of the 7-point algorithm takes just one microsecond, compared with about 10 microseconds for the *LM Basis* algorithm (or 14 microseconds for OpenCV’s SVD-based 7-point implementation), so many more iterations are possible in the same time. The 7-point algorithm has the added advantage of generating fewer candidate hypotheses (often none when a hypothesis set is outlier-contaminated; Figure 3), greatly reducing the time needed to test whether points are compatible.

Numerous variations and optimisations have been proposed for every stage of RANSAC (some are reviewed by [20], [4]), however in most of these, the primary cost is the cost of hypothesis generation, and it is likely that these schemes can also operate efficiently with the subsets of hypotheses found using *LM Basis*.

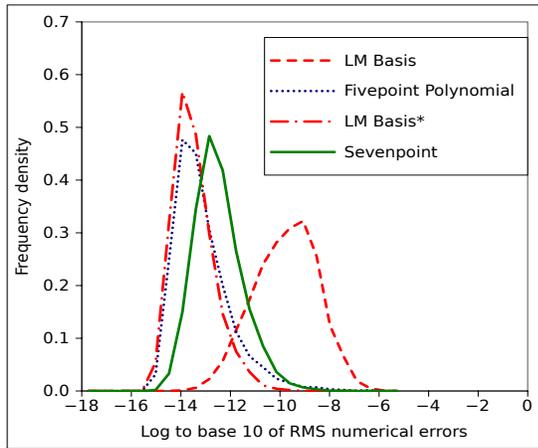


Fig. 4. Numerical errors on simulated noise-free data. The accuracy of *LM Basis* can be improved by changing its convergence criteria (marked *), which increases its computational cost by 15%.

A. Analysis of errors

Firstly numerical errors are compared (Figure 4). Errors are measured as the RMS difference between the essential matrix used to simulate noise-free data, and the essential matrix found. *LM Basis* can be parametrised to iterate until errors are as low as with *Fivepoint Polynomial*, however this level of accuracy is not necessary for RANSAC hypothesis generation—as shown by Figure 5, numerical errors for all methods are insignificant compared to errors from point localisation.

Secondly, errors arising from point localisation errors are evaluated (Figure 5) for a planar and non-planar scene. There is no detectable difference in errors between the different 5-point solvers, as the same equations are being solved exactly. Significant errors in orientation are observed when point localisation is poor, and errors are higher for a planar scene. Errors are lower for the 7-point solver, as this problem is overdetermined (although the extra information is not used optimally), but the 7-point solver fails for planar scenes.

VII. CONCLUSIONS

This paper described a novel scheme for computing essential matrices for RANSAC hypothesis generation. The new scheme uses multiple randomly-initialised runs of the Levenberg-Marquardt algorithm to sample from the set of essential matrices compatible with sets of five correspondences. The new scheme can generate hypotheses at a rate comparable with the fastest contemporary 5-point solver, and the subsets of hypotheses found are just as suitable for RANSAC.

5-point solvers are also compared to an optimised 7-point algorithm based on modified Gram-Schmidt orthonormalisation. For scenes without planar point configurations, the 7-point algorithm is so much faster than any 5-point algorithm that RANSAC finds an inlier set sooner, despite the increased likelihood of 7-point hypothesis sets being outlier-contaminated.

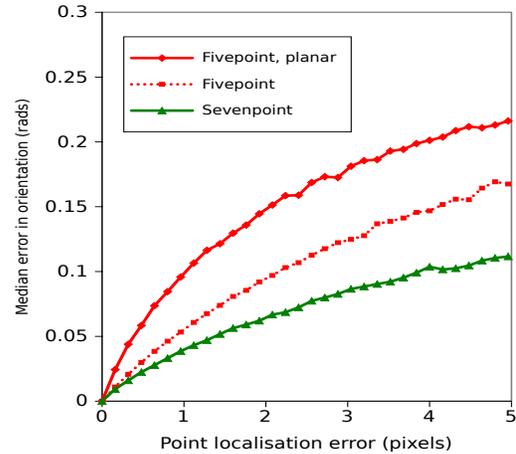


Fig. 5. Errors in recovered rotation vs. point localisation errors. There is no significant difference between different 5-point solvers. The Sevenpoint solver is more accurate for a point cloud, but does not work at all for a planar scene.

REFERENCES

- [1] H. Longuet-Higgins, “A computer algorithm for reconstructing a scene from two projections,” *Nature*, vol. 293, pp. 133–135, 1981.
- [2] M. A. Fischler and R. C. Bolles, “Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography,” *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981.
- [3] A. Ogale, “Google street view from a computer vision perspective,” Presentation available online, 2010. [Online]. Available: http://vision.stanford.edu/teaching/cs223b/lecture/google_streetview_slides.pdf
- [4] O. Chum, “Two-view geometry estimation by random sample and consensus,” Ph.D. dissertation, Czech Technical University, 2005.
- [5] G. Bradski and A. Kaehler, *Learning OpenCV: Computer Vision with the OpenCV Library*. O’Reilly Media, 2008.
- [6] D. Nistér, “An efficient solution to the five-point relative pose problem,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 6, pp. 756–777, 2004.
- [7] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge, UK: Cambridge University Press, 2003.
- [8] Z. Zhang, “Determining the epipolar geometry and its uncertainty: A review,” *IJCV*, vol. 27, no. 2, pp. 161–195, 1998.
- [9] W. Press, *Numerical recipes: the art of scientific computing*. Cambridge University Press, 2001.
- [10] Y. Ma, J. Košecká, and S. Sastry, “Optimization criteria and geometric algorithms for motion and structure estimation,” *IJCV*, vol. 44, 2001
- [11] H. Stewénius, C. Engels, and D. Nistér, “Recent developments on direct relative orientation,” *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 60, pp. 284–294, 2006.
- [12] U. Helmke, K. Hper, P. Lee, and J. Moore, “Essential matrix estimation using gauss-newton iterations on a manifold,” *IJCV*, vol. 74, 2007
- [13] D. Batra, B. Nabbe, and M. Hebert, “An alternative formulation for five point relative pose problem,” in *IEEE Workshop on Motion and Video Computing*. IEEE, 2007, pp. 21–21.
- [14] H. Trivedi, “Estimation of stereo and motion parameters using a variational principle,” *Image and Vision Computing*, vol. 5, 1987.
- [15] G. Guennebaud and B. Jacob, “Eigen 2 matrix library,” n.d. [Online]. Available: <http://eigen.tuxfamily.org/>
- [16] H. Stewénius, “Calibrated fivepoint solver source code,” online, 2006, retrieved April 2011. [Online]. Available: <http://www.vis.uky.edu/~stewe/FIVEPOINT/>
- [17] “C++ port of netlib jenkins-traub algorithm,” online, retrieved April 2011. [Online]. Available: <http://www.akiti.ca/PolyRootRe.html>
- [18] A. Björck, “Numerics of gram-schmidt orthogonalization,” *Linear Algebra and its Applications*, vol. 197–198, pp. 297 – 316, 1994.
- [19] T. Botterill, Essential matrix estimation source code [Online]. Available: <http://www.hilandtom.com/tombotterill/code>, 2011.
- [20] S. Choi, T. Kim, and W. Yu, “Performance evaluation of ransac family,” in *Proceedings of the British Machine Vision Conference*, 2009.