Refining essential matrix estimates from RANSAC

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

Steven Mills
Department of Computer Science,
University of Otago,
Dunedin, NZ.

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

***Abstract*—To estimate the relative pose of two cameras from outlier-contaminated feature correspondences, the essential matrix and inlier set is estimated using RANSAC, then this estimate is refined to minimise an error function on the correspondences. This paper evaluates several refinement methods which minimise functions of Sampson's error. All perform well on large sets of correspondences or when inlier rates are high, but many perform poorly or fail when the inlier set found by RANSAC is small; this is shown to be because the inlier sets contain remaining outliers, while missing some inliers.**

The most accurate solutions are given by minimising the robust Blake-Zisserman function of Sampson's error, although this provides only a minimal improvement in accuracy compared with least squares refinement. The most reliable results are given by nonlinear optimisation constrained to the essential manifold. An efficient parametrisation of the essential manifold as a quaternion and a unit vector is described; applying Iteratively Reweighted Least Squares combined with Levenberg-Marquardt optimisation on this manifold takes typically less than one millisecond.

I. Introduction

Many computer vision applications require the relative pose of two calibrated cameras to be computed from features matched between two images ('feature correspondences'). Normally some features will be incorrectly matched, so an estimation robust to these outliers must be used. This estimation is often done by estimating the essential matrix, $\mathbf{E}$, a 3×3 matrix encoding the relative orientation and translation direction between the two views [1].

$\mathbf{E}$ estimation is usually done in two stages: the first stage is to use RANSAC [2] (or a similar algorithm) to find an essential matrix which is approximately correct, together with a set of matched features which are mostly inliers, and the second stage is to refine the essential matrix to maximise a likelihood function, given the inlier correspondences which have been identified

Many schemes have been proposed for refining essential matrix estimates, however these schemes occasionally fail in the presence of outliers [3], [4], [5], and many have significant computational cost [6], [7], [8]. This paper reviews and evaluates some of these refinement algorithms and identifies the circumstances in which they fail. The most successful approach identified is to use Iteratively Reweighted Least Squares optimisation to minimise the Blake-Zisserman robust cost function of Sampson's error. An efficient parametrisation of the space of essential matrices in terms of a quaternion and a unit vector is described; using this parametrisation, $\mathbf{E}$ can be refined with an average of just 6 iterations, taking less than one millisecond in total.

This paper is organised as follows: the following section describes the essential matrix, its properties, and its estimation using RANSAC; Section III describes algorithms for refining $\mathbf{E}$; Section IV describes an efficient parametrisation of $\mathbf{E}$ as a point on a manifold of unit vectors and unit quaternions; Sections V and VI presents experimental results on simulated data and on synthetic images respectively, and Section VII discusses our findings. Source code associated with the paper is available online [9].

II. Background

This section gives a brief overview of the essential matrix and its properties, and its estimation using RANSAC. The properties of $\mathbf{E}$ are analysed in more detail in Hartley and Zisserman, Chapter 9 [10].

A. The essential matrix

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:

$$\mathbf{E} = [\mathbf{t}]_\times \mathbf{R}, \tag{1}$$

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. A matrix can be decomposed into a rotation and translation in this way when its Singular Value Decomposition (SVD; [10]) has the form:

$$\mathbf{E} = \mathbf{U} \begin{pmatrix} s & 0 & 0 \\ 0 & s & 0 \\ 0 & 0 & 0 \end{pmatrix} \mathbf{V}^T \tag{2}$$

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.

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:

$$\mathbf{x}'^T \mathbf{E} \mathbf{x} = 0 \tag{3}$$

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, with the essential matrix lying in the null space of this matrix. Orthogonal Least Squares Regression (OLSR), via the SVD, is used to find the least-squares fit [3]. Equation 3 is a biased measure of localisation error however, so results from OLSR are typically not accurate enough to be used alone [10].

The least-squares fit to Equation 3 is only an essential matrix if it can be decomposed into a rotation and translation as per Equation 1. The closest (by L_2 norm) essential matrix to a given matrix, $\mathbf{M}$, is given by its SVD:

$$\mathbf{E}_{\text{closest}} = \mathbf{U} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \mathbf{V}^T \tag{4}$$

where $\mathbf{M} = \mathbf{U}\mathbf{D}\mathbf{V}^T$ is the SVD of $\mathbf{M}$ [10].

The OLSR algorithm, followed by this projection to the space of essential matrices gives an essential matrix which is approximately compatible with a set of correspondences.

$\mathbf{E}$ can be decomposed by SVD to give its corresponding rotation and translation direction, however two rotations and two (opposite) translation directions satisfy Equation 1 for any given $\mathbf{E}$. The correct $\mathbf{R}, \mathbf{t}$ pair is identified by reconstructing a 3D point for each possible $\mathbf{R}, \mathbf{t}$; the reconstructed point will fall in front of both cameras only for the correct $\mathbf{R}, \mathbf{t}$ [10].

Given the location of a feature in one image, $\mathbf{x}$, Equation 3 defines a line ('epiline') through the other image where any matching feature, $\mathbf{x}'$, must lie. Feature locations contain measurements errors however, so in practice features will not lie exactly on the corresponding epilines. The distance in the image between the epiline where a feature is known to lie, and where its reconstructed 3D point is projected back to is known as the reprojection error. An excellent approximation to the reprojection error [10], [5], Sampson's error, is given by:

$$r((\mathbf{x}, \mathbf{x}'); \mathbf{E}) = \frac{\mathbf{x}'^T \mathbf{E} \mathbf{x}}{\sqrt{(\mathbf{x}'^T \mathbf{E})_0^2 + (\mathbf{x}'^T \mathbf{E})_1^2 + (\mathbf{E}\mathbf{x})_0^2 + (\mathbf{E}\mathbf{x})_1^2}}$$

Errors from point localisation are approximately Gaussian, therefore, given N correctly matched features, $\{(\mathbf{x}_i, \mathbf{x}_i'), i = 1...N\}$, the MLE of $\mathbf{E}$ is approximately the point where

$$\sum_{i=1}^{N} r((\mathbf{x}_i, \mathbf{x}_i'); \mathbf{E})^2 \tag{5}$$

is minimised.

B. Estimating $\mathbf{E}$ using RANSAC

The RANSAC (Random Sample Consensus [2]) robust estimation framework enables $\mathbf{E}$ to be estimated from a set of point correspondences contaminated with outliers. RANSAC works by repeatedly choosing small random subsets of five correspondences ('hypothesis sets'), fitting an essential matrix to each hypothesis set, then counting the total number of correspondences where Sampson's error is below a threshold. Eventually an essential matrix compatible with many correspondences will be found, usually because the hypothesis set contained only inliers.

RANSAC effectively finds essential matrices which are approximately correct, and inlier sets consisting mostly of inliers (typically about 90%), but can be very slow to find more accurate solutions [12], because of the large number of iterations needed to find a hypothesis set containing only inliers, and because fivepoint solvers are sensitive to point localisation errors [11]. As a result, inlier sets tend to contain nearby outliers, and to miss some inliers. Raising the inlier/outlier threshold generally increases the numbers of both inliers and outliers, and reducing it reduces the number of both.

III. ESSENTIAL MATRIX REFINEMENT

The second stage of relative pose estimation is to refine the essential matrix (or the related fundamental matrix) to maximise some likelihood function given the inlier set. Several methods have been proposed; these each involve two important design decisions: firstly, a cost function to minimise is chosen (this is equivalent to choosing the distribution under which the estimate will be an MLE), and secondly, an appropriate optimisation algorithm and parametrisation is selected.

Cost functions in the form of a sum of squared residual errors (i.e. Equation 5) are appropriate when errors in feature localisation are approximately Gaussian, however this is not the case when features are incorrectly matched, when large residuals are often observed. Numerous alternative cost functions have been proposed which assign more appropriate likelihoods to correspondences with large residual errors; a selection of these are summarised in Figure III.

Functions in the form of a sum of squares can be minimised efficiently using the Gauss-Newton algorithm, an iterative procedure requiring only first derivatives. In practice, the Levenberg-Marquardt (LM) algorithm [13], a dynamically damped version of Gauss-Newton, is frequently used. LM optimisation can also be used to minimise other cost functions, by weighting the residuals on each iteration so that their sum of squares is a local approximation to the desired cost function. This is known as Iteratively Reweighted Least Squares, or IRLS [10]. To minimise a function $\sum_{i=1}^{N} C(r_i)$ for an arbitrary cost function C using IRLS, weights $\{w_i\}$ are chosen so that $(w_i r_i)^2 = C(r_i)$. The function:

$$\sum_{i=1}^{N} (w_i r_i)^2 \tag{6}$$

is minimised by LM, with weights recomputed on each iteration.

Conventional gradient descent optimisation algorithms, such as LM, operate on parameter sets in Cartesian space ($\mathbb{R}^n$), 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 minimise an objective function where parameters $\mathbf{x}$ lie on

Cost function		Corresponding model
Least-squares	$C_{LS}(r) = r^2$	Gaussian errors.
Huber	$C_{Huber}(r) = \begin{cases} r^2, & \text{if } a < t \\ 2t\|r\| - t^2 & \text{otherwise.} \end{cases}$	Errors from heavy-tailed distribution, approx. Gaussian near minimum.
Pseudo-Huber	$C_{PH}(r) = 2t^2(\sqrt{1+(\frac{r}{t})^2} - 1)$	Smoothed version of Huber cost
Blake-Zisserman	$C_{BZ}(r) = \log(1+\epsilon) - \log(\exp(-(\frac{r}{\sigma})^2) + \epsilon)$ where $\epsilon \approx \exp(-(\frac{d}{\sigma})^2)$	Gaussian errors in inliers, outliers all equally likely.

Fig. 1. Cost functions evaluated (from [10]; Appendix 6.8), and the corresponding error distributions for which the model minimising the summed errors is an MLE. t is the inlier/outlier threshold, and σ^2 is the variance in feature localisation.

a manifold, the function 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. Ma et al. [6] use Newton's method to minimise an error similar to Equation 5, with updates constrained to the manifold of essential matrices. They report that false minima are found even with relatively small, and Gaussian, localisation errors however. Rosten et al. [14] use a Lie algebra to represent rotations, and estimate $\mathbf{E}$ by minimising a robust cost function by IRLS/LM. They report that the optimisation can be slow to converge, and can have cost comparable to the costs of RANSAC. Helmke et al. [15] propose an alternative manifold, in which $\mathbf{E}$ is parametrised in terms of the two rotation matrices in its SVD (Equation 2). 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.

An alternative parametrisation of rotations is as an axis-angle pair. Hartley and Kahl [8] use this parametrisation when minimising the (non-robust) L_∞ norm of Sampson's error. Run times are slow (several seconds or more), partly because of the parametrisation of $\mathbf{E}$ chosen. A starting point close to the true minimum is first found by searching the parameter space, as otherwise false minima are found.

A common alternative to estimating $\mathbf{E}$ is to estimate the fundamental matrix, $\mathbf{F}$, instead [4], [3], [16]. $\mathbf{F}$ is a 3×3 matrix which satisfies Equation 3, and is related to $\mathbf{E}$ via the two camera's calibration matrices $\mathbf{K}_1, \mathbf{K}_2$: $\mathbf{F} = \mathbf{K}_2^T \mathbf{E} \mathbf{K}_1$. Disadvantages of $\mathbf{F}$ estimation is that the sevenpoint hypothesis generation algorithm commonly used in RANSAC does not work for near-planar points [16], [11], [3], and that solutions can be found which are incompatible with the known camera calibration.

Torr and Murray [3] estimate $\mathbf{F}$ using an OLSR-based method to minimise a robust function of Sampson's error (e.g. Huber's cost function). Each iteration applies the OLSR algorithm (Section II-A), but weights each constraint (each row in the matrix) as for IRLS, with the weights calculated from both the robust cost function and the bias correction needed for Equation 3. After each iteration, the solutions are projected to the space of fundamental matrices (those with determinant zero) by SVD. This *Reweighted OLSR* method is compared to a gradient-descent-based minimisation of the robust cost on the output of a RANSAC-like algorithm. The gradient-descent-based minimisation is more accurate, although false minima are often found. The authors recommend the combination of a RANSAC-like prior estimate, *Reweighted OLSR* to improve the solution, then nonlinear optimisation to further refine the solution. Zhang [17] also uses a nonlinear optimisation to minimise a robust cost, and find that results are accurate only when gross outliers are first removed by a RANSAC-like algorithm. $\mathbf{F}$ is parametrised to preserve constraints that $\mathbf{F}$ is singular and has unknown scale; by constraining the optimisation, more accurate results are obtained than with *Reweighted OLSR*.

Trivedi [18] describes a relative pose refinement scheme where correspondences are assumed to be a mixture of inliers, with Gaussian localisation errors, or outliers, where any localisation error has the same likelihood. This leads to a robust cost function similar in shape to the Blake-Zisserman cost shown in Figure III. To minimise this cost, the Downhill Simplex method is used; a gradient descent-like method which makes few implicit assumptions about the shape of the objective function. Lacey et al. [12] show this method to give more accurate results than RANSAC for relative pose estimation in the presence of outliers, given a suitable prior estimate of the relative pose.

In summary, numerous robust and least-squares optimisation methods have been proposed for refinement of the essential or fundamental matrix. These methods often perform poorly in the presence of gross outliers; these outliers should be removed first by RANSAC [3], [17]. Of the methods reviewed, many involve computationally expensive parametrisations of $\mathbf{E}$, and some converge to false minima if not initialised appropriately [3], [8].

IV. Comparison of Refinement Algorithms

This paper aims to identify the combination of cost function, optimisation algorithm, and parametrisation most appropriate for refining the essential matrix and inlier set estimated by RANSAC. The combination must be robust to the outliers which remain following RANSAC, however these incorrect correspondences have low residual errors, so will not necessarily degrade results. The three criteria used to evaluate these algorithms are firstly, the 'Success rate', the proportion of times when an approximately-correct solution is found; secondly the median absolute error in the solution found; and thirdly the computational cost, as RANSAC is frequently used in applications requiring real-time performance.

The optimisation algorithms evaluated are *Reweighted OLSR*, and a nonlinear optimiser. *Reweighted OLSR*, described in Section III, solves an OLSR problem on each iteration, then projects the solution to the space of fundamental matrices following each iteration. We have also modified this algorithm so that the solution is projected to the space of essential matrices instead, using the projection defined in Equation 4 (marked *Reweighted OLSR for* $\mathbf{E}$).

The nonlinear optimiser evaluated is IRLS, combined with LM optimisation (labelled *IRLS/LM*). Although different optimisation algorithms could be used, those which are constrained to the set of essential matrices should all find the same minima; the difference between different algorithms is their computational efficiency for this problem. The different cost functions evaluated are listed in Table III.

For nonlinear optimisation, $\mathbf{E}$ is parametrised as a function $\mathbf{E}(\mathbf{q}, \mathbf{t})$ of a unit translation vector $\mathbf{t}$, and a rotation, which is expressed as a quaternion $\mathbf{q}$. Unit quaternions concisely represent 3D rotations as 4D unit vectors. 4D unit vectors form the differential manifold $\mathbb{S}^3$ (the unit sphere in 4D), on which optimisation algorithms can be applied (Section III). This manifold is well-suited to the problem of $\mathbf{E}$ estimation, as it is a continuous representation of rotations, with distances in $\mathbb{S}^3$ corresponding to the difference in orientation between corresponding rotations, and because the computational costs of converting quaternions to rotation matrices, and of normalising quaternions following updates, are low compared to other representations of rotations (i.e. axis-angle or Lie group representations).

At each iteration, the manifold $\mathbb{S}^3$ is parametrised as three vectors $\boldsymbol{\tau}_1, \boldsymbol{\tau}_2, \boldsymbol{\tau}_3$ tangent to the sphere $\mathbb{S}^3$ at $\mathbf{q}$. For each correspondence $(\mathbf{x}, \mathbf{x}')$, the derivative of the cost function in the direction $\boldsymbol{\tau}_i$ is computed by finite differences:

$$\begin{aligned}&\Delta_{\boldsymbol{\tau}_i} C((\mathbf{x}, \mathbf{x}'), \mathbf{E}(\mathbf{q}, \mathbf{t})) \\ &= \tfrac{1}{\delta}\left[C((\mathbf{x}, \mathbf{x}'), \mathbf{E}(\mathbf{q} + \delta\boldsymbol{\tau}_i, \mathbf{t})) - C((\mathbf{x}, \mathbf{x}'), \mathbf{E}(\mathbf{q}, \mathbf{t}))\right]\end{aligned} \tag{7}$$

for some small step size δ.

Unit translation directions similarly form the differential manifold $\mathbb{S}^2$ (the unit sphere in 3D), and derivatives are computed in the same way. After each iteration, $\mathbf{q}$ and $\mathbf{t}$ are updated and renormalised.

V. Results on simulated data

This section presents experimental results using simulated data. For each run, 3D points are generated randomly, and are projected into two cameras. The cameras have a field of view of 0.8 radians, and a relative pose with random translation direction, and random relative orientation with an angle of up-to 0.75 radians (to ensure that points stay in front of the cameras). Outlier correspondences are introduced by mismatching features at random, and simulated localisation errors with standard deviation 0.0025 radians (2 pixels at 640x480) are added to each feature location. After each run, the relative pose from the algorithm being tested is compared to the known relative pose. Figures given are averages over thousands of runs, and have 95% confidence bounds smaller than 1% of their values, however values vary with different parametrisations of the algorithms and of input data.

The inlier/outlier threshold in RANSAC is set to 0.01 radians. For robust cost functions, the inlier/outlier threshold is set to 0.005 radians; this is determined empirically to correctly separate inliers and outliers when the correct relative pose is found. Following RANSAC, and following each algorithm when sequences of algorithms are considered, inliers and outliers are re-assessed, and correspondences with reconstructed points not lying in front of both cameras are marked as outliers.

The robust norms are first tested by refining $\mathbf{E}$ estimated by RANSAC on all correspondences. The optimisation diverges from the true solution in most cases, even for outlier rates as low as 10%. Even the most robust cost function, Blake-Zisserman, often converges to an inaccurate solution. All subsequent experiments use only the current inlier set.

The next experiment conducted is to identify which algorithms are most often successful. A successful run is defined to be one where the orientation is recovered to within 0.25 radians of the true value; and where a majority of the inlier set are in front of the same pair of cameras. Results are shown in Table I. When inlier sets are large (e.g. 25% of 500 correspondences), all algorithms successfully recover the relative pose over 90% of the time. When fewer correspondences are available however, e.g. with only 50 or 100 correspondences and a 25% inlier rate, the performance of all of the algorithms deteriorates, in particular the *Reweighted OLSR* algorithms. When *Reweighted OLSR* fails, the solution often alternates between a matrix with low residual errors, and an essential or fundamental matrix with considerably higher residual errors. There is no significant difference between the success rates for different robust cost functions, or for *Reweighted OLSR* for $\mathbf{E}$ rather than $\mathbf{F}$.

When simulated points are approximately planar, the performance of all algorithms falls substantially (even though RANSAC is still correctly identifying inlier sets containing over 80% inliers). While RANSAC for $\mathbf{F}$ is known to perform poorly in this situation [16], this is not generally the case when estimating $\mathbf{E}$ [11]. The small numbers of outliers remaining are having a large effect on accuracy in this situation (although given 500 planar inliers, robust optimisation still achieves only 79% success).

The next experiment examines why the methods fail when they do; results from various experiments to determine why are given in Figure 2. One possibility is that false minima are being found, so the optimisation is artificially started from

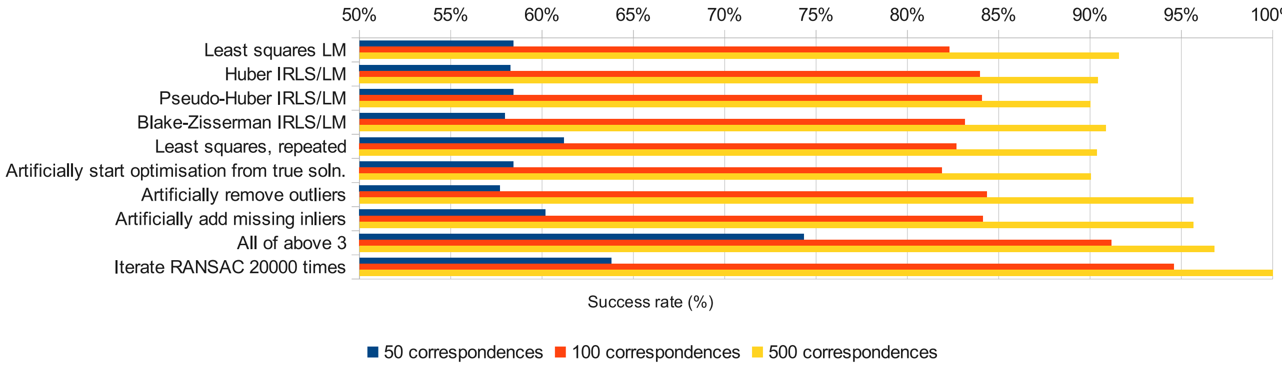

Fig. 2. Experiments to identify when the IRLS/LM method fails. 25% inliers, least-squares cost unless otherwise stated.

TABLE I
SUCCESS RATES FOR DIFFERENT $\mathbf{E}$ REFINEMENT ALGORITHMS, SIMULATED DATA WITH 25% INLIERS BEFORE RANSAC. A RUN IS SUCCESSFUL IF THE ORIENTATION ESTIMATED IS WITHIN 0.25 RADIANS OF THE CORRECT ORIENTATION.

	# correspondences			
Algorithm	50	100	500	500, planar points
Least squares/LM	58%	82%	92%	58%
Huber IRLS/LM	58%	84%	90%	58%
Pseudo-Huber IRLS/LM	58%	84%	90%	58%
Blake-Zisserman IRLS/LM	58%	83%	91%	58%
Reweighted OLSR for $\mathbf{E}$	40%	76%	90%	58%
Reweighted OLSR for $\mathbf{F}$	39%	75%	92%	58%
Reweighted OLSR Pseudo-Huber	40%	75%	92%	58%
RANSAC alone	37%	71%	85%	54%

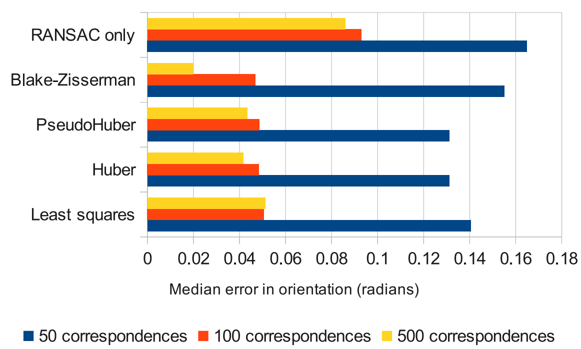

Fig. 3. Median absolute errors in orientation, radians.

the relative pose which is known to be correct. Success rates and residual errors are identical, indicating that the same minima are being found, and that false minima are not a problem. Another reason may be the outliers included in RANSAC's inlier set. Artificially removing these outliers gives an improvement in accuracy for large point sets, but not for smaller point sets. Alternatively, artificially adding missing inliers improves results for small and large point sets, and both removing outliers and adding inliers improves results substantially, indicating that missed inliers and additional outliers are both important for recovering correct relative poses. Larger inlier sets can be found by iterating RANSAC more times, this is illustrated by iterating RANSAC 20000 times. The inlier sets with largest support still contain outliers, and for small correspondence sets, these outliers cause success rates to be lower than the 74% which would be possible if the correct inliers/outliers were found.

One possible method of improving the inlier set's accuracy would be to repeat the nonlinear optimisation several times, with the inlier set reassessed after each run. The results of two least-squares iterations (labelled *Least squares, repeated*) show that this provides no consistent improvement in results.

While the choice of robust cost function makes no significant difference to success rates, it does affect the accuracy of the solutions, as shown in Figure 3. For large correspondence sets, the robust cost functions all give more accurate solutions than least-squares cost. The Blake-Zisserman cost function, which is designed to correctly model the inlier/outlier distribution, gives very accurate results for large point sets, but poorer-than least-squares results for small point sets. This is because for small point sets, the RANSAC inlier set still contains many outliers. As the residual errors in these outliers are as low as for inliers, the robust cost functions do nothing to avoid them.

Table II shows the run times for the various algorithms, and the numbers of iterations required. Each is implemented in C++ using the Eigen matrix library [19], and compiled using gcc (source code is available online [9]). Times are given for code running on a single core of an Intel i7 processor, running at 2.93GHz. Times are dominated by the computation of residuals (and corresponding weights for IRLS); the costs of the LM algorithm itself are minimal. All algorithms are substantially faster than the efficient RANSAC implementation used [11], demonstrating the suitability of the IRLS/LM optimisation algorithm, and quaternion/unit translation vector parametrisation chosen. Convergence is fast for all cost functions, indicating that IRLS/LM performs well at this optimisation. The Blake-Zisserman cost function requires more iterations than the other cost functions, however this optimisation is still fast compared with RANSAC.

VI. RESULTS ON SYNTHETIC IMAGES

The SLAMDUNK system [20] simulates images from a moving camera by using a ray tracer. The images provided by the authors include high levels of self-similarity, and large planar surfaces; these images are used to test the essential

TABLE II
TYPICAL RUN TIMES AND NUMBERS OF ITERATIONS REQUIRED (ALL VARY WITH DIFFERENT PARAMETRISATIONS/CONVERGENCE CRITERIA).

	50 correspondences		500 correspondences	
Algorithm	Time (ms)	# Iterations	Time (ms)	# Iterations
Least squares/LM	0.10	6	0.30	6
Huber IRLS/LM	0.11	7	0.40	7
Pseudo-Huber IRLS/LM	0.12	7	0.50	7
Blake-Zisserman IRLS/LM	0.19	12	1.6	11
All Reweighted OLSR	0.06-0.08	3	0.1	3
RANSAC, until 20% support found	19	2175	21	2305
RANSAC, 20000 iterations	180	20000	210	20000

TABLE III
RESULTS ON SYNTHETIC IMAGES (SLAMDUNK DATASET).

		Median error (radians)	
Algorithm	Success	Orientation	Translation dir.
RANSAC	98%	0.078	0.286
IRLS/LM Pseudo-Huber	99%	0.072	0.213
IRLS/LM Blake-Zisserman	99%	0.071	0.200
Least-squares LM	99%	0.073	0.222
Least-squares LM, repeated	99%	0.072	0.217
Reweighted OLSR for E	96%	0.072	0.222

matrix refinement. Patches of the image centred on FAST corners [21] are matched between frames, using the sum of squared differences to compare patches. 269 pairs of frames captured 0.67 seconds apart are used. Relative orientations of 0.5 radians are typical between pairs of frames, typically about 300 features are matched, and inlier rates are typically 50%. Table III shows the errors in recovered rotations and translations, and the success rate of the various algorithms.

All algorithms achieve at least 96% success, although again the nonlinear optimisation outperforms Reweighted OLSR. As before, the robust norms provide only a very small improvement in accuracy.

VII. DISCUSSION

This paper has evaluated several different options for the refinement of essential matrices estimated by RANSAC. The most reliable and accurate approach is to use Iteratively Reweighted Least Squares, combined with Levenberg-Marquardt optimisation, to minimise a robust cost function based on Sampson's approximation to the reprojection error. Essential matrices are parametrised by a unit quaternion and unit translation vector; this efficient and minimal parametrisation ensures that computational costs are low compared with the cost of RANSAC. This approach is shown to outperform alternative methods based on iteratively solving a weighted linear system.

All algorithms occasionally fail to compute an accurate solution, particularly with small correspondence sets and when inlier rates are low. Failures are caused by both remaining outliers, and missing inliers, in the inlier set found by RANSAC. Increasing the number of RANSAC iterations only partially addresses this problem, as the essential matrix with the largest support still leads to an inlier set containing many outliers.

Different robust cost functions are evaluated. Improvements in accuracy compared with a least-squares approach are minimal, although the Blake-Zisserman cost function provides a small improvement in accuracy for large correspondence sets. For smaller correspondence sets, the minimal robust cost often incorrectly identifies remaining outliers as inliers.

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] P. H. S. Torr and D. W. Murray, "The development and comparison of robust methods for estimating the fundamental matrix," *Int. J of Computer Vision*, vol. 24, no. 3, pp. 271–300, 1997.

[4] Z. Zhang, "Determining the epipolar geometry and its uncertainty: A review," *Int. J of Computer Vision*, vol. 27, no. 2, pp. 161–195, 1998.

[5] T. Botterill, "Visual navigation for mobile robots using the bag-of-words algorithm," Ph.D. dissertation, University of Canterbury, 2010.

[6] Y. Ma, J. Košecká, and S. Sastry, "Optimization criteria and geometric algorithms for motion and structure estimation," *Int. J of Computer Vision*, vol. 44, no. 3, pp. 219–249, 2001.

[7] E. Rosten, R. Porter, and T. Drummond, "Faster and better: A machine learning approach to corner detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, pp. 105–119, 2010.

[8] R. Hartley and F. Kahl, "Global optimization through searching rotation space and optimal estimation of the essential matrix," in *ICCV*, 2007.

[9] T. Botterill, "Computer vision c++ source code," Online, 2011. [Online]. Available: http://hilandtom.com/tombotterill/code

[10] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge, UK: Cambridge University Press, 2003.

[11] T. Botterill, S. Mills, and R. Green, "Fast RANSAC hypothesis generation for essential matrix estimation," in *Proc. Digital Image Computing: Techniques and Applications (DICTA)*, 2011.

[12] A. J. Lacey, N. Pinitkarn, and N. A. Thacker, "An evaluation of the performance of RANSAC algorithms for stereo camera calibration," in *Proc. British Machine Vision Conference*, 2000.

[13] D. Marquardt, "An algorithm for least-squares estimation of nonlinear parameters," *SIAM Journal on Applied Mathematics*, vol. 11, 1963.

[14] E. Rosten, G. Reitmayr, and T. Drummond, "Improved RANSAC performance using simple, iterative minimal-set solvers," *Arxiv preprint arXiv:1007.1432*, 2010.

[15] U. Helmke, K. Hper, P. Lee, and J. Moore, "Essential matrix estimation using Gauss-Newton iterations on a manifold," *Int. J of Computer Vision*, vol. 74, pp. 117–136, 2007.

[16] O. Chum, "Two-view geometry estimation by random sample and consensus," Ph.D. dissertation, Czech Technical University Prague, 2005.

[17] Z. Zhang, "Flexible camera calibration by viewing a plane from unknown orientations," in *Proc. Int. Conf on Computer Vision*, 1999.

[18] H. Trivedi, "Estimation of stereo and motion parameters using a variational principle," *Image and Vision Computing*, vol. 5, 1987.

[19] G. Guennebaud and B. Jacob, "Eigen 2 matrix library," n.d. [Online]. Available: http://eigen.tuxfamily.org/

[20] J. Funke and T. Pietzsch, "A framework for evaluation visual SLAM," in *Proc. British Machine Vision Conference*, 2009.

[21] E. Rosten and T. Drummond, "Machine leaning for high-speed corner detection," in *Proc. ECCV*, 2006, pp. 430–443.