

In Defence of the 8-point Algorithm

Richard I. Hartley,
GE-Corporate Research and Development,
Schenectady, NY, 12309,
email : hartley@crd.ge.com

Abstract

The fundamental matrix is a basic tool in the analysis of scenes taken with two uncalibrated cameras, and the 8-point algorithm is a frequently cited method for computing the fundamental matrix from a set of 8 or more point matches. It has the advantage of simplicity of implementation. The prevailing view is, however, that it is extremely susceptible to noise and hence virtually useless for most purposes. This paper challenges that view, by showing that by preceding the algorithm with a very simple normalization (translation and scaling) of the coordinates of the matched points, results are obtained comparable with the best iterative algorithms. This improved performance is justified by theory and verified by extensive experiments on real images.

1 Introduction

The 8-point algorithm for computing the essential matrix was introduced by Longuet-Higgins in a now classic paper ([7]). In that paper the essential matrix is used to compute the structure of a scene from two views with calibrated cameras. **The great advantage of the 8-point algorithm is that it is linear, hence fast and easily implemented.** If 8 point matches are known, then the solution of a set of linear equations is involved. With more than 8 points, a linear least squares minimization problem must be solved. The term 8-point algorithm will be used in this paper to describe this method whether only 8 points, or more than 8 points are used.

The essential property of the essential matrix is that it conveniently encapsulates the epipolar geometry of the imaging configuration. One notices immediately that the same algorithm may be used to compute a matrix with this property from uncalibrated cameras. In this case of uncalibrated cameras it has become customary to refer to the matrix so derived as

the *fundamental matrix*. Just as in the calibrated case, the fundamental matrix may be used to reconstruct the scene from two uncalibrated views, but in this case only up to a projective transformation ([3, 5]).

Unfortunately, despite its simplicity the 8-point algorithm has often been criticized for being excessively sensitive to noise in the specification of the matched points. Indeed this belief has become the prevailing wisdom. Consequently, because of its importance, many alternative algorithms have been proposed for the computation of the fundamental matrix. See [8] for a description and comparison of several algorithms for finding the fundamental matrix. Without exception, these algorithms are considerably more complicated than the 8-point algorithm. Other iterative algorithms have been described (briefly) in [6, 1].

It is the purpose of this paper to challenge the common view that the 8-point algorithm is inadequate and markedly inferior to the more complicated algorithms. The poor performance of the 8-point algorithm can probably be traced to implementations that do not take sufficient account of numerical considerations, most specifically the condition of the set of linear equations being solved. It is shown in this paper that a simple transformation (**translation and scaling**) of the points in the image before formulating the linear equations leads to an enormous improvement in the condition of the problem and hence of the stability of the result. The added complexity of the algorithm necessary to do this transformation is insignificant.

It is not claimed here that this modified 8-point algorithm will perform quite as well as the best iterative algorithms. However it is shown by thousands of experiments on many images that the difference is not very great between the modified 8-point algorithm and iterative techniques. Indeed the 8-point algorithm does better than some of the iterative techniques.

2 Outline of the 8-point Algorithm

Notation Vectors are represented by bold lower case letters, such as \mathbf{u} , and all such vectors are thought of as being column vectors unless explicitly transposed (for instance \mathbf{u}^\top is a row vector). Vectors are multiplied as if they were matrices. In particular, for vectors \mathbf{u} and \mathbf{v} , the product $\mathbf{u}^\top \mathbf{v}$ represents the inner product, whereas $\mathbf{u}\mathbf{v}^\top$ is a matrix. The norm of a vector \mathbf{f} is equal to the square root of the sum of squares of its entries, that is the Euclidean length of the vector. Similarly, for matrices, we use the Frobenius norm, which is defined to be the square root of the sum of squares of the entries of the matrix.

Linear solution for the fundamental matrix.

The fundamental matrix is defined by the equation

$$\mathbf{u}'^\top F \mathbf{u} = 0 \quad (1)$$

for any pair of matching points $\mathbf{u}' \leftrightarrow \mathbf{u}$ in two images. Given sufficiently many point matches $\mathbf{u}'_i \leftrightarrow \mathbf{u}_i$, (at least 8) this equation (1) can be used to compute the unknown matrix F . In particular, writing $\mathbf{u} = (u, v, 1)^\top$ and $\mathbf{u}' = (u', v', 1)^\top$ each point match gives rise to one linear equation in the unknown entries of F . The coefficients of this equation are easily written in terms of the known coordinates \mathbf{u} and \mathbf{u}' . Specifically, the equation corresponding to a pair of points $(u, v, 1)$ and $(u', v', 1)$ will be

$$\begin{aligned} uu'f_{11} &+ uv'f_{21} + uf_{31} + vu'f_{12} + vv'f_{22} \\ &+ vf_{32} + u'f_{13} + v'f_{23} + f_{33} = 0. \end{aligned}$$

The row of the equation matrix may be represented as a vector $(uu', uv', u, vu', vv', v, u', v', 1)$. From all the point matches, we obtain a set of linear equations of the form

$$A\mathbf{f} = 0 \quad (2)$$

where \mathbf{f} is a 9-vector containing the entries of the matrix F , and A is the equation matrix. The fundamental matrix F , and hence the solution vector \mathbf{f} is defined only up to an unknown scale. For this reason, and to avoid the trivial solution \mathbf{f} , we make the additional constraint $\|\mathbf{f}\| = 1$ where $\|\mathbf{f}\|$ is the norm of \mathbf{f} ¹.

Under these conditions, it is possible to find a solution to the system (2) with as few as 8 point matches.

¹An alternative is to set $F_{33} = 1$ and solving a linear least squares minimization problem. The general conclusions of this paper are equally valid for this version of the algorithm.

With more than 8 point matches, we have an over-specified system of equations. Assuming the existence of a non-zero solution to this system of equations, we deduce that the matrix A must be rank-deficient. In other words, although A has 9 columns, the rank of A must be at most 8. In fact, except for exceptional configurations ([9]) the matrix A will have rank exactly 8, and there will be a unique solution for \mathbf{f} .

This previous discussion assumes that the data is perfect, and without noise. In fact, because of inaccuracies in the measurement or specification of the matched points, the matrix A will not be rank-deficient – it will have rank 9. In this case, we will not be able to find a non-zero solution to the equations $A\mathbf{f} = \mathbf{0}$. Instead, we seek a least-squares solution to this equation set. In particular, we seek the vector \mathbf{f} that minimizes $\|A\mathbf{f}\|$ subject to the constraint $\|\mathbf{f}\| = \mathbf{f}^\top \mathbf{f} = 1$. It is well known (and easily derived using Lagrange multipliers) that the solution to this problem is the unit eigenvector of $A^\top A$ corresponding to the smallest eigenvalue of A . Note that since $A^\top A$ is positive semi-definite and symmetric, all its eigenvectors are real and positive, or zero. For convenience, (though somewhat inexactly), we will call this eigenvector the *least eigenvector* of $A^\top A$. An appropriate algorithm for finding this eigenvector is the algorithm of Jacobi ([10]) or the Singular Value Decomposition ([10]).

The singularity constraint. An important property of the fundamental matrix is that it is singular, in fact of rank 2. Furthermore, the left and right null-spaces of F are generated by the vectors representing (in homogeneous coordinates) the two epipoles in the two images. Most applications of the fundamental matrix rely on the fact that it has rank 2. The matrix F found by solving the set of linear equations (2) will not in general have rank 2, and we should take steps to enforce this constraint. The most convenient way is to correct the matrix F found by the solution of (2). Matrix F is replaced by the matrix F' that minimizes the Frobenius norm $\|F - F'\|$ subject to the condition $\det F' = 0$. Thus, let $F = UDV^\top$ be the Singular Value Decomposition of F , where D is a diagonal matrix $D = \text{diag}(r, s, t)$ satisfying $r \geq s \geq t$. We let $F' = U\text{diag}(r, s, 0)V^\top$. This method was suggested by Tsai and Huang ([11]) and has been proven to minimize the Frobenius norm of $F - F'$, as required.

Thus, the 8-point algorithm for computation of the

fundamental matrix may be formulated as consisting of two steps, as follows.

Linear solution. Given point matches $\mathbf{u}'_i \leftrightarrow \mathbf{u}_i$, solve the equations $\mathbf{u}'_i{}^\top F \mathbf{u}_i = 0$ to find F . The solution is the least eigenvector, \mathbf{f} of $A^\top A$, where A is the equation matrix.

Constraint Enforcement. Replace F by F' , the closest singular matrix to F under Frobenius norm.

The algorithm thus stated is extremely simple, and rapid to implement, assuming the availability of a suitable linear algebra library (for instance [10]).

3 Transformation of the Input

Image coordinates are sometimes given with the origin at the top-left of the image, and sometimes with the origin at the centre. The question immediately occurs whether this makes a difference to the results of the 8-point algorithm for computing the fundamental matrix. More generally, to what extent is the result of the 8-point algorithm dependent on the choice of coordinates in the image. Suppose, **for instance the image coordinates were changed by some affine or even projective transformation before running the algorithm.** Will this materially change the result? That is the question that we will now consider.

Suppose that coordinates \mathbf{u} in one image are replaced by $\hat{\mathbf{u}} = T\mathbf{u}$, and coordinates \mathbf{u}' in the other image are replaced by $\hat{\mathbf{u}}' = T'\mathbf{u}'$. Substituting in the equation $\mathbf{u}'^\top F \mathbf{u} = 0$, we derive the equation $\hat{\mathbf{u}}'{}^\top T'^{-\top} F T^{-1} \hat{\mathbf{u}} = 0$, where $T'^{-\top}$ is the inverse transpose of T' . This relation implies that $T'^{-\top} F T^{-1}$ is the fundamental matrix corresponding to the point correspondences $\hat{\mathbf{u}}' \leftrightarrow \hat{\mathbf{u}}$. An alternative method of finding the fundamental matrix is therefore suggested, as follows.

1. Transform the image coordinates according to transformations $\hat{\mathbf{u}}_i = T\mathbf{u}_i$ and $\hat{\mathbf{u}}'_i = T'\mathbf{u}'_i$.
2. Find the fundamental matrix \hat{F} corresponding to the matches $\hat{\mathbf{u}}'_i \leftrightarrow \hat{\mathbf{u}}_i$.
3. Set $F = T'^\top \hat{F} T$.

The fundamental matrix found in this way corresponds to the original untransformed point correspondences $\mathbf{u}'_i \leftrightarrow \mathbf{u}_i$. What choice should be made for the transformations T and T' will be left unspecified

for now. First, we need to determine whether carrying out this transformation has any effect whatever on the result.

As verified above, $\mathbf{u}'^\top F \mathbf{u} = \hat{\mathbf{u}}'{}^\top \hat{F} \hat{\mathbf{u}}$, where \hat{F} is defined by $\hat{F} = T'^{-\top} F T^{-1}$. So, if $\mathbf{u}'^\top F \mathbf{u} = \epsilon$, then also $\hat{\mathbf{u}}'{}^\top \hat{F} \hat{\mathbf{u}} = \epsilon$. Thus, there is a one-to-one correspondence between F and \hat{F} giving rise to the same error. It may appear therefore that the matrices F and \hat{F} minimizing the error ϵ (or more exactly, the sum of squares of errors corresponding to all points) will be related by the formula $\hat{F} = T'^{-\top} F T^{-1}$, and hence one may retrieve F as the product $T'^\top \hat{F} T$. This conclusion is **false** however. For, although F and \hat{F} so defined give rise to the same error ϵ , the condition $\|F\| = 1$, imposed as a constraint on the solution, is not equivalent to the condition $\|\hat{F}\| = 1$. In particular, there is no one-to-one correspondence between F and \hat{F} giving rise to the same error ϵ , subject to the constraint $\|F\| = \|\hat{F}\| = 1$. In fact, the two solutions for the fundamental matrix may be considerably different in the presence of moderate amounts of noise.

4 Condition Number

The linear method consists in finding the least eigenvector of the matrix $A^\top A$. This may be done by expressing $A^\top A$ as a product UDU^\top where U is orthogonal and D is diagonal. We assume that the diagonal entries of D are in non-increasing order. In this case, the least eigenvector of $A^\top A$ is the last column of U . Denote by κ the ratio d_1/d_8 (recalling that $A^\top A$ is a 9×9 matrix). The parameter κ is the condition number² of the matrix $A^\top A$, well known to be an important factor in the analysis of stability of linear problems ([4]). If κ is large, then very small changes to the data can cause large changes to the solution. The sensitivity of invariant subspaces is discussed in detail in [4], p413.

We now consider how the condition number of the matrix $A^\top A$ may be made small. We consider two sorts of transformation, translation and scaling. These methods will be given only an intuitive justification, since a complete analysis of the condition number of the matrix is too complex to undertake here.

The major reason for the poor condition of the matrix $A^\top A$ is the lack of homogeneity in the image coordinates. In an image of dimension 200×200 , a typical image point will be of the form $(100, 100, 1)$. If

²Strictly speaking, d_1/d_9 is the condition number, but d_1/d_8 is the parameter of importance here.

both \mathbf{u} and \mathbf{u}' are of this form, then the corresponding row of the equation matrix will be of the form $\mathbf{r}^\top = (10^4, 10^4, 10^2, 10^4, 10^4, 10^2, 10^2, 10^2, 1)$. The contribution to the matrix $A^\top A$ is of the form $\mathbf{r}\mathbf{r}^\top$, which will contain entries ranging between 10^8 and 1. For instance, the diagonal entries of $A^\top A$ will be $(10^8, 10^8, 10^4, 10^8, 10^8, 10^4, 10^4, 10^4, 1)$. Summing over all point correspondences will result in a matrix $A^\top A$ for which the diagonal entries are approximately in this proportion.

We may now use the *Interlacing Property* ([4], page 411) for the eigenvalues of a symmetric matrix to get a bound on the condition number of the matrix. Suppose that the diagonal entries of $X = A^\top A$ are equal to $(10^8, 10^8, 10^4, 10^8, 10^8, 10^4, 10^4, 10^4, 1)$. We denote by X_r the trailing $r \times r$ principal submatrix (that is the last r columns and rows) of the matrix $A^\top A$, and by $\lambda_i(X_r)$ its i -th largest eigenvalue. Thus, $X_9 = A^\top A$ and $\kappa = \lambda_1(X_9)/\lambda_8(X_9)$. First we consider the eigenvalues of X_2 . Since the sum of the two eigenvalues is $\text{trace}(X_2) = 10^4 + 1$, we see that $\lambda_1(X_2) + \lambda_2(X_2) = 10^4 + 1$. Since the matrix is positive semi-definite, both eigenvalues are non-negative, so we may deduce that $\lambda_1(X_2) \leq 10^4 + 1$. From the interlacing property, we deduce that $\lambda_8(X_9) \leq \lambda_7(X_8) \leq \dots \lambda_1(X_2) \leq 10^4 + 1$. On the other hand, also from the interlacing property, we know that the largest eigenvalue of $A^\top A$ is not less than the largest diagonal entry. Thus, $\lambda_1(X_9) \geq 10^8$. Therefore, the ratio $\kappa = \lambda_1(X_9)/\lambda_8(X_9) \geq 10^8/(10^4 + 1)$. Usually, in fact $\lambda_8(X_9)$ will be much smaller than $10^4 + 1$ and the condition number will be far greater.

This analysis shows that scaling the coordinate so that the homogeneous coordinates are on the average equal to unity will improve the condition of the matrix $A^\top A$.

Translation Consider a case where the origin of the image coordinates is at the top left hand corner of the image, so that all the image coordinates are positive. In this case, an improvement in the condition of the matrix may be achieved by translating the points so that the centroid of the points is at the origin. This claim was verified by experimentation, but can also be explained informally by arguing as follows. Suppose that the first image coordinates (the u -coordinates) of a set of points are $\{1001.5, 1002.3, 998.7, \dots\}$. By translating by 1000, these numbers may be changed to $\{1.5, 2.3, -1.3\}$. Thus, in the untranslated values,

the significant values of the coordinates are obscured by the coordinate offset of 1000. The significant part of the coordinate values is found only in the third or fourth significant figure of the coordinates. This has a bad effect on the condition of the corresponding matrix $A^\top A$. A more detailed analysis of the effect of translation is not provided here.

5 Effect of Scaling in Stage 2

So far we have discussed the effect of a normalizing transformation on the first stage of the 8-point algorithm, namely the solution of the set of linear equations to find F . The second step of the algorithm is to **enforce the singularity constraint that $\det F = 0$** .

The method described above of enforcing the singularity constraint gives the singular matrix \hat{F} nearest to F in Frobenius norm. The trouble with this method is that it treats all entries of the matrix equally, regardless of their magnitude. Thus, entries of F small in absolute value may be expected to undergo a perturbation much greater relative to their magnitude than the large entries.

Suppose that a set of matched points is normalized so that on the average all three homogeneous coordinates have the same magnitude. Thus, a typical point will look like $(1, 1, 1)^\top$. The fundamental matrix computed from these normalized coordinates may be expected to have all its entries approximately of the same magnitude.

Now, consider what happens if we scale the coordinates of points \mathbf{u}_i and \mathbf{u}'_i by a factor which we will assume is equal to 100. Thus, a typical coordinate will be of the order of $(100, 100, 1)^\top$. The corresponding fundamental matrix F will be obtained from original one by multiplying the first two rows, and the first two columns by 10^{-2} . Entries in the the top left 2×2 block will be multiplied by 10^{-4} . We conclude that a typical fundamental matrix derived from coordinates of magnitude $(100, 100, 1)^\top$ will have entries of the following order of magnitude.

$$F = \begin{pmatrix} 10^{-4} & 10^{-4} & 10^{-2} \\ 10^{-4} & 10^{-4} & 10^{-2} \\ 10^{-2} & 10^{-2} & 1 \end{pmatrix} \quad (3)$$

To verify this conclusion, here is the fundamental matrix for the pair of house images in Fig 1³.

³The notation -9.766e-08 means -9.766×10^{-8} .

$$F = \begin{pmatrix} -9.796e-08 & 1.473e-06 & -6.660e-04 \\ -6.346e-07 & 1.049e-08 & 7.536e-03 \\ 8.107e-04 & -7.739e-03 & -2.364e-02 \end{pmatrix}$$

In comparing this with (3), one must bear in mind that F is defined only up to nonzero scaling. The imbalance of the matrix is even worse than predicted by (3) because the image has dimension 512×512 . Now, in taking the closest singular matrix, all entries will tend to be perturbed by approximately the same amount. However, the relative perturbation will be greatest for the smallest entries. The question arises whether the small entries in the matrix F are important. Consider a typical point $\mathbf{u} \approx (100, 100, 1)^\top$. In computing the corresponding epipolar line $F\mathbf{u}$, we see that the largest entries in the vector \mathbf{u} are multiplied by the smallest, and hence least relatively stable entries of the matrix F . Thus, for computation of the epipolar line, the smallest entries in F are the most important. We have the following undesirable condition :

The most important entries in the fundamental matrix are precisely those that are subject to the largest relative perturbation when enforcing the singularity constraint without prior normalization.

This condition is corrected if normalization of the image coordinates is carried out first, for then all entries of the fundamental matrix will be treated approximately equally, and none is more important than another in computing epipolar lines.

6 Normalizing transformations

The previous sections concerned with the condition number of the matrix $A^\top A$ indicate that it is desirable to apply a transformation to the coordinates before carrying out the 8-point algorithm for finding the fundamental matrix.

6.1 Isotropic Scaling

As a first step, the coordinates in each image are translated (by a different translation for each image) so as to bring the centroid of the set of all points to the origin. The coordinates are also scaled. The previous sections suggested that the best results will be obtained if the coordinates are scaled, so that on the average a point \mathbf{u} is of the form $\mathbf{u} = (1, 1, 1)^\top$. Such a point will lie a distance $\sqrt{2}$ from the origin. Rather

than choose different scale factors for each direction, an isotropic scaling factor is chosen so that the u and v coordinates of a point are scaled equally. The transformation is as follows :

1. The points are translated so that their centroid is at the origin.
2. The points are then scaled isotropically so that the average distance from the origin is equal to $\sqrt{2}$.

Such a transformation is applied to each of the two images independently.

6.2 Non-isotropic Scaling

As an alternative to the isotropic scaling method just described, an affine transformation was tried, in which the centroid of the points was placed at the origin and the two principal moments of the set of points were both made equal to unity. Thus, the set of points are transformed to an approximately symmetric circular cloud of points of radius one about the origin. The results obtained using this type of transformation to the data were little different from those obtained using the isotropic scaling method.

7 Experimental Evaluation

The algorithm was coded, consisting of steps 1, 2 and 3 as given in section 3. The transformations used in step 1 are isotropic scalings as described in section 6.1. Step 2 consists of the two sub-steps of Linear solution and Constraint enforcement as described in section 2. This algorithm will be called the *normalized 8-point algorithm*.

This algorithm was tested on a large number of real images to evaluate its performance. In carrying out these tests, the normalized 8-point algorithm was compared with several other algorithms for finding the fundamental matrix. For the most part the implementations of these other algorithms were provided by other researchers. In this way the results were not biased in any way by my possibly inefficient implementation of competing algorithms. In addition, the images and matched points used as inputs have been supplied to me. Methods of obtaining the matched points therefore varied from image to image, as did methods for eliminating bad matches (outliers). In all cases, however, the matched points were found by automatic means, and usually some sort of outlier detection and removal was carried out, based on least-median squares techniques (see [2]).

The general procedure for evaluation was as follows.

1. Matching points were computed by automatic techniques, and outliers were detected and removed.
2. The fundamental matrix was computed using a subset of all points.
3. For each point \mathbf{u}_i , the corresponding epipolar line $F\mathbf{u}_i$ was computed and distance from the line $F\mathbf{u}_i$ from the matching point \mathbf{u}'_i was calculated. This was done in both directions, (that is starting from points \mathbf{u}_i in the first image and also from \mathbf{u}'_i in the second image). The average distance of the epipolar line from the corresponding point was computed, and used as a measure of quality of the computed Fundamental matrix. This evaluation was carried out using **all** matched points, and not just the ones that were used to compute F .

7.1 The algorithms

Here is a brief description of the algorithms tested.

The 8-point algorithm. In this algorithm, the points were used as is, without pre-transformation to compute the fundamental matrix. The singularity constraint was enforced.

The normalized 8-point algorithm. The 8-point algorithm was used with the translation and isotropic scaling method described in section 6.1. The singularity constraint was enforced.

Minimizing the epipolar distances. An implementation by Long Quan of an algorithm described by Luong ([8, 2]) was used. This is an iterative algorithm that uses a parametrization of the fundamental matrix with 7 parameters. Thus the singularity constraint is enforced as part of the algorithm. The cost function being minimized is the squared sum of distances of the points from epipolar lines. The point-line distances in both images are taken into account.

Optimal Algorithm (Minimizing point displacement). This algorithm (my own implementation) is an iterative algorithm. It finds the fundamental matrix F , and points $\hat{\mathbf{u}}_i$ and $\hat{\mathbf{u}}'_i$ such that $\hat{\mathbf{u}}_i^\top F \hat{\mathbf{u}}_i = 0$ exactly, $\det F = 0$ and the squared pixel error $\sum_i d(\hat{\mathbf{u}}_i, \mathbf{u}_i)^2 + d(\hat{\mathbf{u}}'_i, \mathbf{u}'_i)^2$ is minimized. The details

of how this is done are described in [6]. Under the assumption of gaussian noise in the placement of the matched points (an approximation to the truth), this algorithm gives the fundamental matrix corresponding to the most likely true placement of the matched points (the estimated points $\hat{\mathbf{u}}_i \leftrightarrow \hat{\mathbf{u}}'_i$). For this reason, I have generally considered this algorithm to be the best available. The experiments generally bear out this belief, but it is not the purpose of this paper to justify this point.

7.2 The Images.

The various algorithms were tried with 5 different pairs of images of diverse types and with different placements of the epipoles. Figures 1 and 2 show two of the different image sets and the placement of the epipoles. A few of the epipolar lines are shown in the images. The intersection of the pencil of lines is the epipole. There was a wide variation in the accuracy of the matched points for the different images.

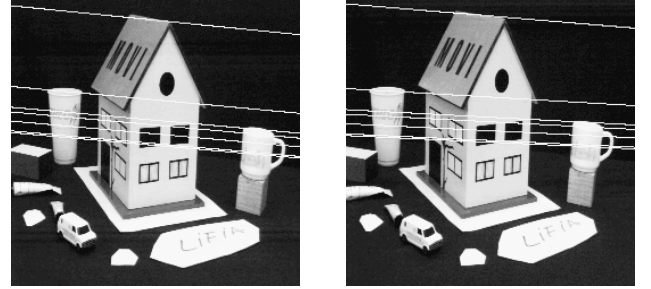


Fig 1. Houses Images.

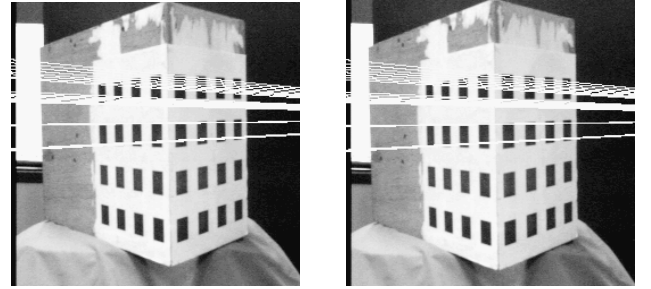


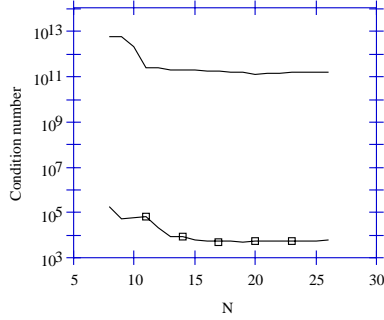
Fig 2. Calibration Jig.

7.3 Graphical Presentation of the Results.

The following graphs show the results of several runs of the algorithms, with different numbers of points being used. The number of points used to compute the fundamental matrix ranged from 8 up to three-quarters of the total number of matched points. For each value of N , the algorithms were run 100 times using randomly selected sets of N matching points. The average error (point – epipolar line distance) was

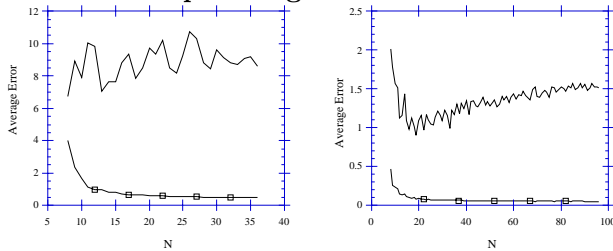
computed using all available matched points. The graphs show the average error over the 100 runs for each value of N . The error shown is the average point-epipolar line distance measured in pixels.

Graph 1 : Effect of Normalization on the Condition Number.



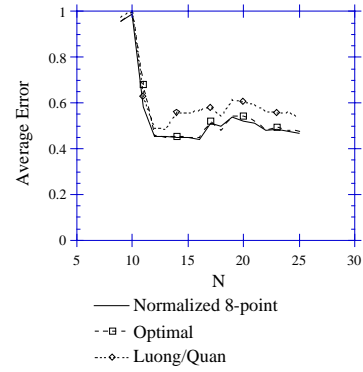
This graph shows a plot of the base-10 logarithm of the condition number of the linear equation set in the case of the house images, for varying numbers of points (the x -axis). The upper curve is without normalization, the lower one with normalization. The improvement is approximately 10^8 .

Graph 2 : Comparison of normalized and unnormalized 8-point algorithms.



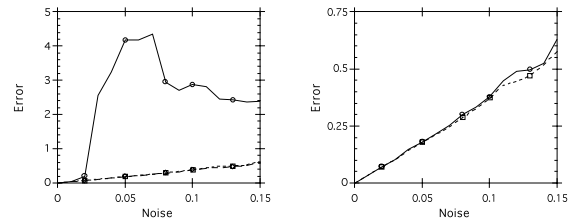
These set of graphs show the improvement achieved by normalization. On the left is the data for the houses. On the right is the data for the calibration jig. Note the differences in Y-scale for the different plots. For the calibration jig the matched points were known with extreme accuracy (about 0.1 pixels accuracy) whereas for the houses the accuracy was about 1 pixel.

Graph 3 : Comparison with minimized epipolar distance



This graph compares the normalized 8-point algorithm, the optimal algorithm and the epipolar distance minimization algorithm. In this case data was gathered for only one run for each value of N . Nevertheless, the results seem to be consistent. The normalized 8-point and optimal algorithms perform best, and the point-epipolar distance algorithm slightly less well. The graphs start with 9 points. Only the optimal algorithm performed well with 8 points (1.2 pixels error), and the other algorithms were off the graph.

Graph 4 : Reconstruction Error.



To test the performance of the various algorithms for reconstruction accuracy experiments were done to measure the degradation of accuracy as noise levels increase. The Calibration images (Fig 2) were used for this purpose. Since reconstruction error is most appropriately measured in an Euclidean frame, a Euclidean model was built for the calibration cube, initially by inspection and then by refinement using the image data. This model served as ground truth. Next, the image coordinates were corrected (by an average of 0.02 pixels) to agree exactly with the Euclidean model. Varying amounts of zero-mean gaussian noise were added to the image coordinates, a projective reconstruction was carried out, and a projective transformation was computed to bring the projective reconstruction most nearly into agreement with the model. The average 3D displacement of the reconstructed points

from the model was measured. The plotted values are the result average over all points (128 in all) for 10 trials. The reconstruction error is measured in units equal to the length of the side of one of the black squares in the image.

At the left are the results of three algorithms : at the top is unnormalized 8-point algorithm, whereas at the bottom almost overlapped are the results of the normalized 8-point algorithm and the optimal algorithm. In the right hand graph, only the normalized 8-point and optimal algorithms are shown. The result shows that the results of the normalized 8-point algorithm is almost indistinguishable from the optimal algorithm, but that the unnormalized algorithm performs very much worse.

8 Conclusions

With normalization of the coordinates in order to improve the condition of the problem, the 8-point algorithm performs almost as well as the best iterative algorithms. On the other hand, it runs about 20 times faster and is far easier to code. There seems to be little advantage in choosing the non-isotropic scaling scheme for the normalization transform, since the simpler isotropic scaling performs just as well. Without normalization of the inputs, however, the 8-point algorithm performs quite badly, often with errors as large as 10 pixels, which makes it virtually useless. It would seem to follow that the reason that other researchers have had such poor results with the 8-point algorithm is that they have not carried out any preliminary normalization step as discussed here. An exception is the algorithm of Zisserman and Beardsley ([1]) which does do some normalization. This algorithm was also tested and found to perform almost as well as our normalized 8-point algorithm, but space limitations forbid presentation of more details of these tests.

9 Acknowledgements

I wish to thank all those people who supplied data and algorithm results to me for the running of these tests. This includes most particularly Andrew Zisserman, Paul Beardsley, Jean-Claude Cottier, Long Quan and Boubakeur Boufama gave me images and data, and Long Quan, Andrew Zisserman and Paul Beardsley who let me use the results of their algorithms. Finally, thanks to Roger Mohr for making possible my sojourn in Grenoble allowing me the possibility to do this work.

References

- [1] P. A. Beardsley, A. Zisserman, and D. W. Murray. Navigation using affine structure from motion. In *Computer Vision - ECCV '94, Volume II, LNCS-Series Vol. 801, Springer-Verlag*, pages 85–96, 1994.
- [2] R. Deriche, Z. Zhang, Q.-T. Luong, and O. Faugeras. Robust recovery of the epipolar geometry for an uncalibrated stereo rig. In *Computer Vision - ECCV '94, Volume I, LNCS-Series Vol. 800, Springer-Verlag*, pages 567–576, 1994.
- [3] O. D. Faugeras. What can be seen in three dimensions with an uncalibrated stereo rig? In *Computer Vision - ECCV '92, LNCS-Series Vol. 588, Springer-Verlag*, pages 563 – 578, 1992.
- [4] Gene H. Golub and Charles F. Van Loan. *Matrix Computations, Second edition*. The Johns Hopkins University Press, Baltimore, London, 1989.
- [5] R. Hartley, R. Gupta, and T. Chang. Stereo from uncalibrated cameras. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 761–764, 1992.
- [6] Richard I. Hartley. Euclidean reconstruction from uncalibrated views. In *Proc. of the Second European Workshop on Invariance, Ponta Delgada, Azores*, pages 187–202, October 1993.
- [7] H.C. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 293:133–135, Sept 1981.
- [8] Quang-Tuan Luong, Rachid Deriche, Olivier D. Faugeras, and Theodore Papadopoulos. On determining the fundamental matrix: analysis of different methods and experimental results. Report RR-1894, INRIA, 1993.
- [9] S. J. Maybank. The projective geometry of ambiguous surfaces. *Phil. Trans. R. Soc. Lond., A* 332:1 – 47, 1990.
- [10] William H. Press, Brian P. Flannery, Saul A. Teukolsky, and William T. Vetterling. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, 1988.
- [11] R. Y. Tsai and T. S. Huang. Uniqueness and estimation of three dimensional motion parameters of rigid objects with curved surfaces. *IEEE Trans. Patt. Anal. Machine Intell.*, PAMI-6:13–27, 1984.