In defence of the 8-point algorithm

Richard I. Hartley
G.E. CRD, Schenectady, NY, 12301.

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 ([1]). 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 ([2, 3]). Apart from scene reconstruction, the

fundamental matrix may also be used for many other tasks, such as image rectification ([4]), computation of projective invariants ([5]), outlier detection ([6, 7]) and stereo matching ([8]);

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 [9, 10] 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 [11, 12].

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 techiques. 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

$$\mathbf{u}'^\top F\mathbf{u} = \mathbf{0} \tag{1}$$

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

$$uu'F_{11} + uv'F_{21} + uF_{31} + vu'F_{12} + vv'F_{22} + vF_{32} + u'F_{13} + v'F_{23} + F_{33} = 0 \ . \quad (2)$$

The row of the equation matrix may be represented as a vector

$$(uu', uv', u, vu', vv', v, u', v', 1) \ . \quad (3)$$

From all the point matches, we obtain a set of linear equations of the form

$$A\mathbf{f} = 0 \quad (4)$$

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

$$||\mathbf{f}|| = 1 \quad (5)$$

where $||\mathbf{f}||$ is the norm of $\mathbf{f}$[1].

Under these conditions, it is possible to find a solution to the system (4) with as few as 8 point matches. With more than 8 point matches, we have an overspecified 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 ([13]) 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 corresponding to the smallest eigenvalue of $A^\top A$. 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 ([14]) or the Singular Value Decomposition ([14, 15]).

[1] An 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.

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 (4) will not in general have rank 2, and we should take steps to enforce this constraint. The most convenient way to enforce this constraint is to correct the matrix F found by the solution of (4). Matrix F is replaced by the matrix F' that minimizes the Frobenius norm $||F - F'||$ subject to the condition $\det F' = 0$. A convenient method of doing this is to use the Singular Value Decomposition (SVD). In particular, let $F = UDV^\top$ be the SVD of F, where D is a diagonal matrix $D = \mathrm{diag}(r, s, t)$ satisfying $r \geq s \geq t$. We let $F' = U\mathrm{diag}(r, s, 0)V^\top$. This method was suggested by Tsai and Huang ([16]) and has been proven to minimize the Frobenius norm of $F - F'$, as required.

Minimizing the difference between F and F' in Frobenius norm has little theoretical justification, and in fact there are other methods of enforcing the singularity constraint a posteriori which have more theoretical basis (for instance [17]). However, as will be seen this method gives good results.

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. This is done using the Singular Value Decomposition.

The algorithm thus stated is extremely simple, and rapid to implement, assuming the availability of a suitable linear algebra library (for instance [14]).

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 $\hat{F} = T'^{-\top} F T^{-1}$. Thus, 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 $||F|| = ||\hat{F}|| = 1$.

This is a crucial point, and so we will look at it from a different point of view. A set of point correspondences $\mathbf{u}'_i \leftrightarrow \mathbf{u}_i$ give rise to a set of equations of the form $A\mathbf{f} = 0$. If now we make the transformation $\hat{\mathbf{u}}_i = T\mathbf{u}_i$ and $\hat{\mathbf{u}}'_i = T'\mathbf{u}'_i$, then the set of equations will be replaced by a different set of equations of the form $\hat{A}\hat{\mathbf{f}} = 0$. One may verify, in particular that the matrix $\hat{A}$ may be written in the form $\hat{A} = AS$ where S is a 9×9 matrix that may be written explicitly in terms of the entries of T and T' (but it is not very important exactly how). Therefore one is led to consider the two sets of equations $A\mathbf{f} = 0$ and $AS\hat{\mathbf{f}} = 0$. One may guess that the least-squares solutions to these two sets of equations will be related according to $\hat{\mathbf{f}} = S^{-1}\mathbf{f}$. If this were so, then replacing $\hat{\mathbf{f}}$ by $S\hat{\mathbf{f}}$ one once more retrieves the original solution $\mathbf{f}$. The mapping $\hat{\mathbf{f}} \mapsto S\hat{\mathbf{f}}$ corresponds precisely to the matrix mapping $\hat{F} \mapsto T'^{\top} \hat{F} T$.

However, things are not that simple. Perhaps the least-squares solutions to the two sets of equations $A\mathbf{f} = 0$ and $AS\hat{\mathbf{f}} = 0$ are not so simply related. The solution $\mathbf{f}$ to the system $A\mathbf{f} = 0$ is the least eigenvector of the matrix $A^{\top}A$. Is it so that $\hat{\mathbf{f}} = S^{-1}\mathbf{f}$ is the least eigenvector of $(AS)^{\top}(AS)$? Letting λ be the

least eigenvalue of $A^\top A$, we verify :

$$\begin{aligned} S^\top A^\top AS\hat{\mathbf{f}} &= S^\top A^\top ASS^{-1}\mathbf{f} \\ &= S^\top A^\top A\mathbf{f} \\ &= S^\top \lambda \mathbf{f} \\ &= \lambda S^\top S\hat{\mathbf{f}} \\ &\neq \lambda \hat{\mathbf{f}} \ . \end{aligned}$$

Thus, in fact, $S^{-1}\mathbf{f}$ is not the least eigenvector of $(AS)^\top AS$. In fact it is not an eigenvector at all.

Let us see how significant this effect is. We take the example that T and T' are simply scalings of the coordinates, in fact, multiplication of the coordinates by a factor of 10. These transformations are represented by diagonal matrices of the form $T = T' = \mathrm{diag}(10, 10, 1)$ acting on homogeneous coordinates. In this case, the matrix S is also a diagonal matrix of the form $S = \mathrm{diag}(10^2, 10^2, 10, 10^2, 10^2, 10, 10, 10, 1)$, assuming that the vector $\mathbf{f}$ represents the elements of F in the row-major order
$f_{11}, f_{12}, f_{13}, f_{21}, f_{22}, f_{23}, f_{31}, f_{32}, f_{33}$. The matrix $S^\top S$ equals
$\mathrm{diag}(10^4, 10^4, 10^2, 10^4, 10^4, 10^2, 10^2, 10^2, 1)$. In this case, we see $(AS)^\top AS\hat{\mathbf{f}} = \lambda S^\top S\hat{\mathbf{f}}$, and so $\hat{\mathbf{f}}$ is very far from being an eigenvector of $(AS)^\top AS$.

We conclude that the method of transformation leads to a different solution for the fundamental matrix. This is a rather undesirable feature of the 8-point algorithm as it stands, that the result is changed by a change of coordinates, or even simply a change of the origin of coordinates. A similar problem was observed by Bookstein ([18]) in the problem of fitting conics to sets of points. To correct this, it seems advisable to normalize the coordinates of the points in some way by expressing them in some fixed canonical frame, as yet unspecified.

4 Condition of the System of Equations

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 number[2] of the matrix $A^\top A$, well known to be an important factor in the analysis of stability of linear problems ([19]). Its relevance to the problem of finding the least eigenvector is briefly explained next.

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

The bottom right hand 2×2 block of matrix D is of the form $\begin{pmatrix} d_8 & 0 \\ 0 & 0 \end{pmatrix}$, assuming that $d_9 = 0$, which ideally will be the case. Now, suppose that this block is perturbed by the addition of noise to become $\begin{pmatrix} d_8 & \epsilon \\ \epsilon & 0 \end{pmatrix}$. In order to restore this matrix to diagonal form we need to multiply left and right by $V^\top$ and V, where V is a rotation through an angle $\theta = \arctan(2\epsilon/d_8)$ (as the reader may verify). If ϵ is of the same order of magnitude as d_8 then this is a significant rotation. Looking at the full matrix, $A^\top A = UDU^\top$, we see that the perturbed matrix will be written in the form $U\bar{V}D'\bar{V}^\top U^\top$ where $\bar{V} = \begin{pmatrix} I_{7\times 7} & 0 \\ 0 & V \end{pmatrix}$. Multiplying by $\bar{V}$ replaces the last column of U by a combination of the last two columns. Since the last column of U is the least eigenvector of the matrix, this perturbation will drastically alter the least eigenvector of the matrix $A^\top A$. Thus, changes to $A^\top A$ of the order of magnitude of the eigenvalue d_8 cause significant changes to the least eigenvector. Since multiplication by an orthogonal matrix does not change the Frobenius norm of a matrix, we see that $||A^\top A|| = \left(\sum_{i=1}^{9} d_i^2\right)^{1/2}$. If the ratio $\kappa = d_1/d_8$ is very large, then d_8 represents a very small part of the Frobenius norm of the matrix. A perturbation of the order of d_8 will therefore cause a very small relative change to the matrix $A^\top A$, while at the same time causing a very significant change to the least eigenvector. Since $A^\top A$ is written directly in terms of the coordinates of the points $\mathbf{u} \leftrightarrow \mathbf{u}'$, we see that if κ is large, then very small changes to the data can cause large changes to the solution. This is obviously very undesirable. The sensitivity of invariant subspaces is discussed in greater detail in [19], p413, where more specific conditions for the sensitivity of invariant subspaces are given.

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 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* ([19], 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 $\lambda_8(X_9) \leq \lambda_7(X_8) \leq \ldots \lambda_1(X_2) \leq 10^4 + 1$. 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 will be 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 $\{1001.5, 1002.3, 998.7, \ldots\}$. 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 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. This normalization has been implemented as a prior step in the 8-point algorithm with excellent results.

5.1 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. In the discussion of scaling, it was

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 $\mathbf{u} = (u, v, w)^\top$, with each of u, v and w having the same average magnitude. Rather than choose different scale factors for each point, an isotropic scaling factor is chosen so that the u and v coordinates of a point are scaled equally. To this end, we choose to scale the coordinates so that the average distance of a point $\mathbf{u}$ from the origin is equal to $\sqrt{2}$. This means that the "average" point is equal to $(1, 1, 1)^\top$. In summary the transformation is as follows

1. The points are translated so that their centroid is at the origin.
2. The points are then scaled so that the average distance from the origin is equal to $\sqrt{2}$.
3. This transformation is applied to each of the two images independently.

5.2 Non-isotropic Scaling

In non-isotropic scaling, the centroid of the points is translated to the origin as before. After this translation the points form a cloud about the origin. Scaling is then carried out so that the two principal moments of the set of points are both equal to unity. Thus, the set of points will form an approximately symmetric circular cloud of points of radius one about the origin.

Both translation and scaling can be done in one step as follows. Let $\mathbf{u}_i = (u_i, v_i, 1)^\top$ for $i = 1, \ldots, N$ and form the matrix $\sum_i \mathbf{u}_i {\mathbf{u}_i}^\top$. Since this matrix is symmetric and positive definite, we may take its Choleski factorization ([15, 14]) to get $\sum_{i=1}^{N} \mathbf{u}_i {\mathbf{u}_i}^\top = NKK^\top$, where K is upper triangular. It follows that $\sum_i K^{-1}\mathbf{u}_i {\mathbf{u}_i}^\top K^{-\top} = NI$, where I is the identity matrix. Setting $\hat{\mathbf{u}}_i = K^{-1}\mathbf{u}_i$, we have $\sum_i \hat{\mathbf{u}}_i {\hat{\mathbf{u}}_i}^\top = NI$. Consequently, the set of points $\hat{\mathbf{u}}_i$ have their centroid at the origin and the two principal moments are both equal to unity, as desired. Note that K^{-1} is upper triangular, and so it represents an affine transformation.

To summarize : the points are transformed so that

1. Their centroid is at the origin.
2. The principal moments are both equal to unity.

6 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. This may not be true if applied to specific classes of cameras, but it will be true for fundamental matrices computed from arbitrarily selected matched points, as the following argument shows.

A permutation of the three homogeneous coordinates in either or both the images will result in another set of realizable matched points. The corresponding fundamental matrix will be obtained from the original one by permuting the corresponding rows and/or columns of the matrix. In doing this, any entry of F may be moved to any other position. This means that no entry of the fundamental matrix is qualitatively different from any other, and hence on the average (over all possible sets of matched points) all entries of F will have the same average 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.

$$F = \begin{pmatrix} 10^{-4} & 10^{-4} & 10^{-2} \\ 10^{-4} & 10^{-4} & 10^{-2} \\ 10^{-2} & 10^{-2} & 1 \end{pmatrix} \tag{6}$$

To verify this conclusion, here is the fundamental matrix for the pair of house images in Fig 1[3].

$$F = \begin{pmatrix} -9.796e-08 & 1.473e-06 & -6.660e-04 \\ -6.346e-07 & 1.049e-08 & 7.536e-03 \\ 8.107e-04 & -7.739e-03 & -2.364e-02 \end{pmatrix} \tag{7}$$

In comparing (7) with (6), one must bear in mind that F is defined only up to nonzero scaling. The imbalance of the matrix (7) is even worse than predicted by (6) 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

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

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.

7 Experimental Evaluation

The 8-point algorithm with prior transformation of the coordinates, as described here 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 8-point algorithm with pre-normalization as described above 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, whom I will acknowledge later. 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 that I have tested the algorithms on 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 [6, 7, 8]).

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. In the case of algorithms, such as the 8-point algorithm, that do not automatically enforce the singularity constraint (that is the constraint that $\det F = 0$) this constraint was enforced a posteriori by finding the nearest singular matrix to the computed fundamental matrix. This was done using the Singular Value Decomposition (as in [16, 20]).

4. 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, except outliers, and not just the ones that were used to compute F.

7.1 Other algorithms.

A brief description of the algorithms tested follows, but first some notation.

Given fundamental matrix F and point $\mathbf{u}_i$, the epipolar line in the second image corresponding to point $\mathbf{u}_i$ is $F\mathbf{u}_i$. Similarly, $F^\top \mathbf{u}'_i$ is the epipolar line corresponding to $\mathbf{u}'_i$. Point $\mathbf{u}'_i$ lies on epipolar line $F\mathbf{u}_i$ if and only if $\mathbf{u}_i'^\top F\mathbf{u}_i = 0$. However, the quantity $\mathbf{u}_i'^\top F\mathbf{u}_i$ does not correspond to any meaningful geometric quantity, certainly not to distance between the point $\mathbf{u}'_i$ and the epipolar line $F\mathbf{u}_i$. Writing $F\mathbf{u}_i = (\lambda, \mu, \nu)^\top$, the distance $d(\mathbf{u}'_i, F\mathbf{u}_i)$ is equal to $\mathbf{u}_i'^\top F\mathbf{u}_i / \sqrt{\lambda^2 + \nu^2}$, provided $\mathbf{u}'_i = (u'_i, v'_i, 1)^\top$. Similarly, denoting $F^\top \mathbf{u}'_i$ by (λ', μ', ν'), one has $d(\mathbf{u}_i, F^\top \mathbf{u}'_i) = \mathbf{u}_i'^\top F\mathbf{u}_i / \sqrt{\lambda'^2 + \nu'^2}$.

The 8-point algorithm In this algorithm, the points were used as is, without pre-transformation to compute the fundamental matrix. The algorithm minimizes the quantity $\sum_i \left(\mathbf{u}_i'^\top F\mathbf{u}_i\right)^2$. The singularity constraint was enforced.

The 8-point algorithm with isotropic scaling The 8-point algorithm was used with the translation and isotropic scaling method described in section 5.1. The singularity constraint was enforced.

The 8-point algorithm with non-isotropic scaling This is the same as the previous method, except that the non-isotropic scaling method described in section 5.2 was used.

Minimizing the epipolar distances In implementation by Zhengyou Zhang of an algorithm described in ([9, 6, 10]) 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. Thus this algorithm minimizes

$$\sum_i d(\mathbf{u}'_i, F\mathbf{u}_i)^2 + d(\mathbf{u}_i, F^\top \mathbf{u}'_i)^2 = (\mathbf{u}_i'^\top F\mathbf{u}_i)^2 \left(\frac{1}{\lambda_i^2 + \mu_i^2} + \frac{1}{\lambda_i'^2 + \mu_i'^2} \right) \quad .$$

Two versions of this algorithm were tested, in which respectively the unnormalized and normalized versions of the 8-point algorithm were used for initialization.

A gradient-based technique This algorithm is related to the previous method, but it minimizes a slightly different cost function, namely

$$\sum_i \frac{(\mathbf{u}_i'^\top F \mathbf{u}_i)^2}{\lambda_i^2 + \mu_i^2 + \lambda_i'^2 + \mu_i'^2} \ .$$

This cost function is a first order approximation to the point-displacement error discussed in the next method (below). The implementation tested was by Zhengyou Zhang, and the algorithm is discussed in ([9, 10]). Here also this algorithm was initialized using either the normalized or unnormalized 8-point 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 [11, 21]. 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. This algorithm is referred to as the "optimal algorithm" in this paper.

Approximate Calibration The results of an algorithm of Beardsley and Zisserman ([12]) were provided for comparison. This algorithm does an approximate normalization of the coordinates by selecting the origin of coordinates at the centre of the image, and by scaling by division by the approximate focal length of the camera (measured in pixels – that is, the scaling factor in the calibration matrix). Since this method employs a normalization similar to the isotropic scaling algorithm, one expects it to give similar results. It does, however rely on some approximate knowledge of camera calibration.

Iterative Linear Another algorithm provided by Beardsley and Zisserman is representative of a general approach to improving the performance of linear algorithms. This same approach can be applied to many different linear algorithms, such as camera pose and calibration estimation ([22]), projective reconstruction from lines ([23]) and reconstruction of point positions in space ([24]). In this approach, the 8-point algorithm is run a first time. From this initial solution a set of weights for the linear equations are computed. The set

of linear equations are multiplied by these weights and the 8-point algorithm is run again. This may be repeated several times. The weights are chosen in such a way that the linear equations express a meaningful measurable quantity. In this case, to minimize point-epipolar line distance, each equation $\mathbf{u}_i'^{\top} F \mathbf{u}_i = 0$ is multiplied by the weight

$$w_i = \left(\frac{1}{\lambda^2 + \mu^2} + \frac{1}{\lambda'^2 + \mu'^2} \right)^{1/2}$$

where the values λ_i, μ_i, λ_i' and μ_i' are computed from the previous iteration. The advantage of this type of algorithm is that it is simple to implement compared with iterative parameter estimation methods, such as Levenberg-Marquardt ([14]).

7.2 The Images.

The various algorithms were tried with 5 different pairs of images. The images are presented in Figures 1 – 5 to show the diversity of image types, 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, as will be indicated later.

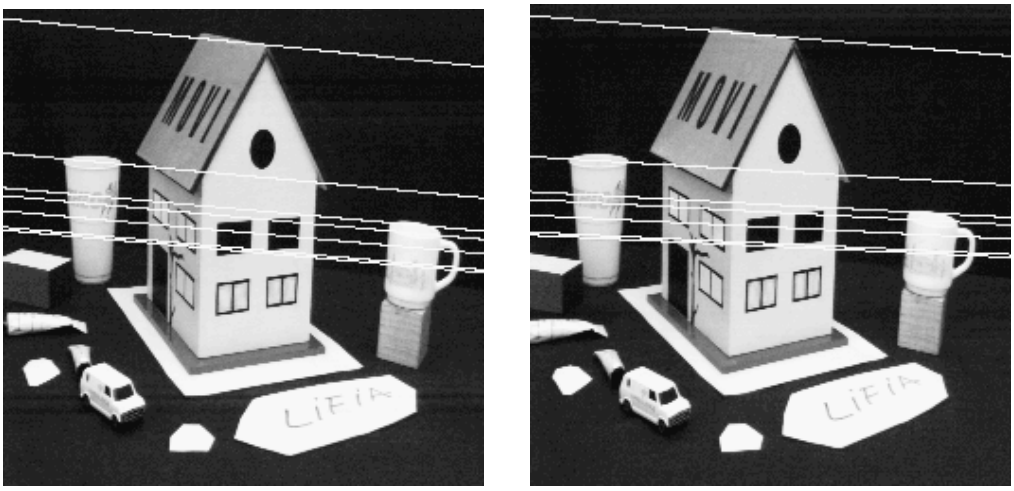


Figure 1: **Houses Images.** *The epipoles are a long way from the image centres.*

Figure 2: **Statue image** *An outdoor scene with the epipoles well away from the centre.*

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.

In the graph annotations the following notation is used.

method 0 represents the unnormalized 8-point algorithm

method 1 represents the 8-point algorithm with scaling in stage 1. (For an explanation, see below).

method 2 represents the 8-point algorithm with scaling in stage 2.

method 3 represents the normalized 8-point algorithm (mormalization in both stages.

method 4 represents the "optimal" algorithm (minimization of point displacement).

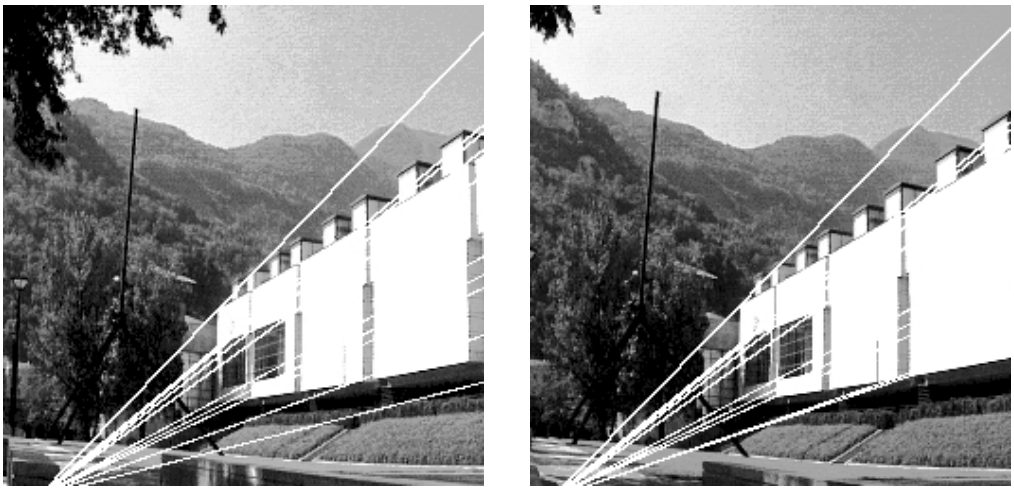
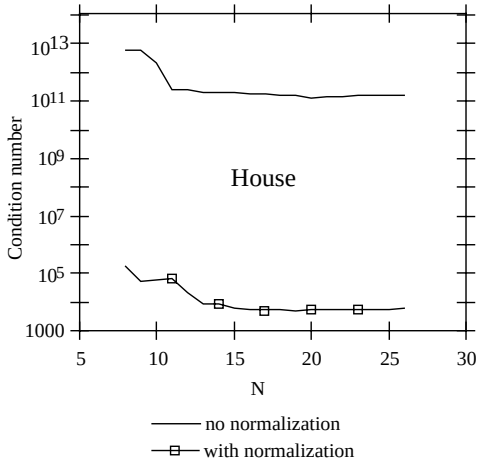


Figure 3: **Grenoble Museum** *The epipoles are close to the image.*

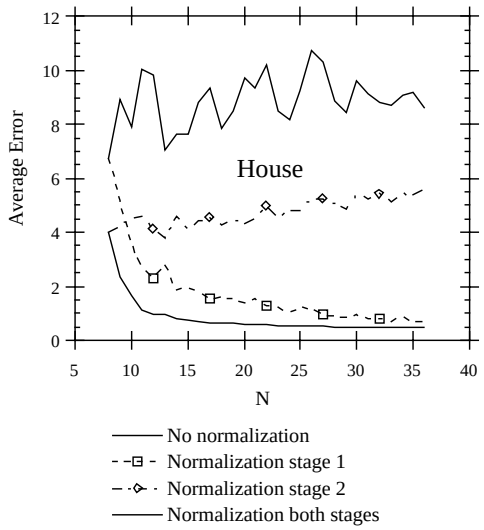
Graph 1 : Effect of Normalization on the Condition Number.



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 : Effect of normalization on the two stages of the algorithm.

Figure 4: **Corridor Scene** *In the corridor scene the epipoles are right in the image.*



This plot shows the effect of normalization in the two stages of the 8-point algorithm. To explain this, four algorithmic steps may be identified :

Normalization Transformation of the image coordinates using transforms T and T'.

Solution Finding matrix F by solving a set of linear equations.

Constraint enforcement Replacing F by the closest singular matrix.

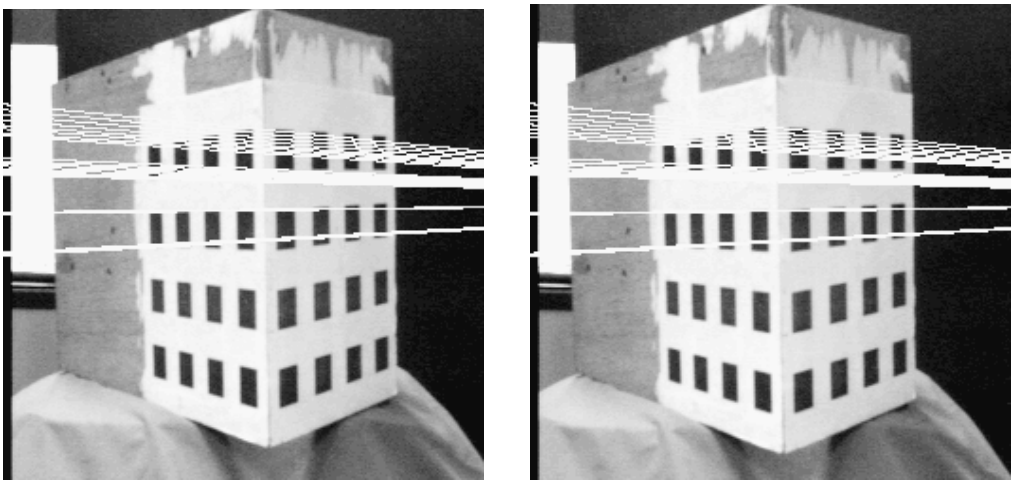


Figure 5: **Calibration Jig** *In this calibration jig, the matched points were known extremely accurately.*

Denormalization Replacing F by $T'^{\top}FT$.

It is possible to take these steps in a different order to show the effect of normalization on the Solution (stage 1) and Constraint enforcement (stage 2) steps of the algorithm. Thus, the four curves shown correspond to the following algorithm steps.

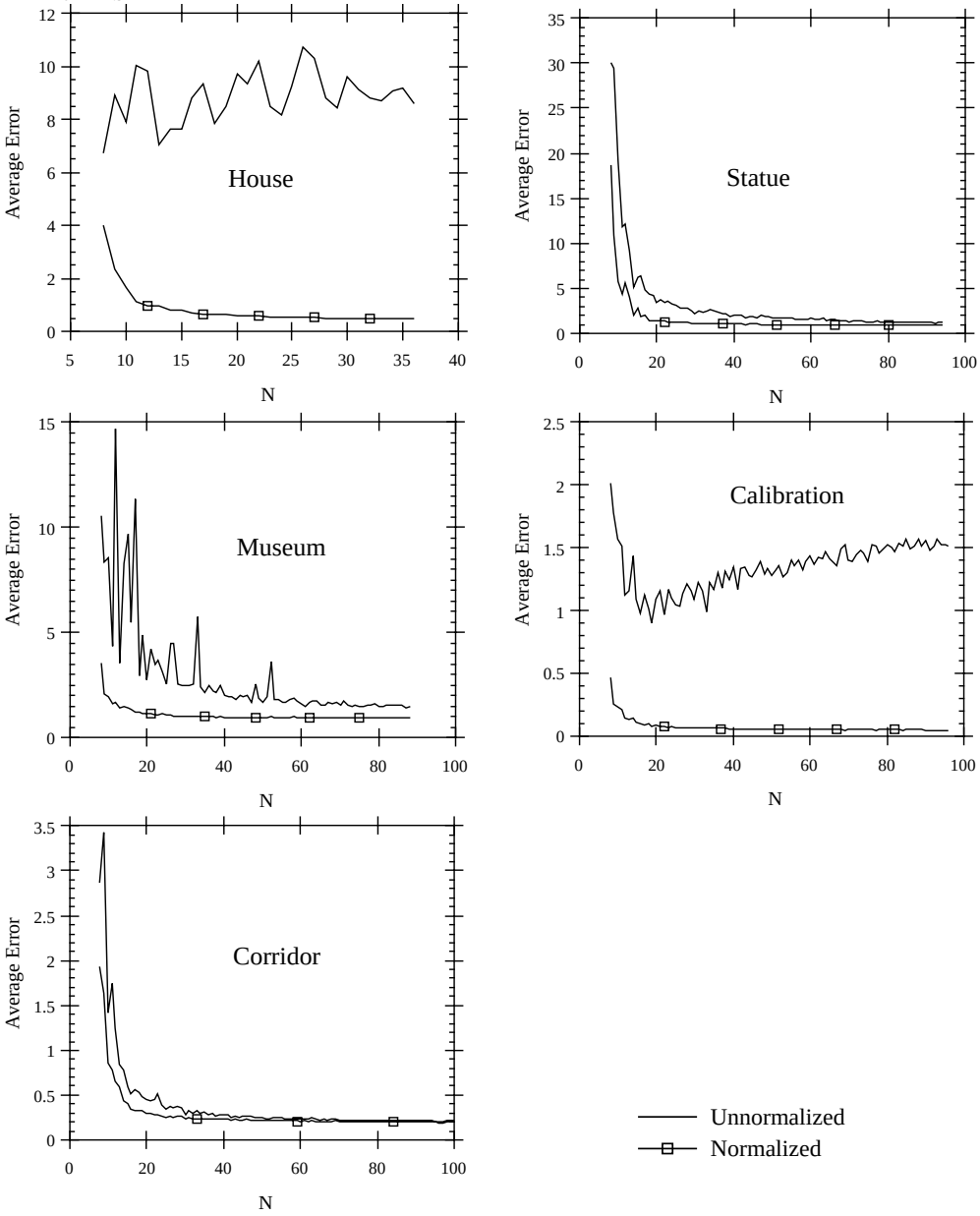
1. No normalization : Solution – Constraint enforcement.
2. Stage 1 normalization : Normalization – Solution – Denormalization – Constraint enforcement.
3. Stage 2 normalization : Solution – Normalization – Constraint enforcement – Denormalization.
4. Both stages of normalization : Normalization – Solution – Constraint enforcement – Denormalization.

As may be seen, normalization has the greatest effect on stage 1 (the Solution stage), but normalization for stage 2 has a significant effect as well. The best results are had by doing normalization in both stages.

Note, how for $N = 8$ the normalization has no effect on stage 1, since in this case we are finding the solution to a set of equations, and not a least-squares solution to a redundant set. This explains why the two pairs of curves show the same results for $N = 8$.

For these experiments, the house images were used.

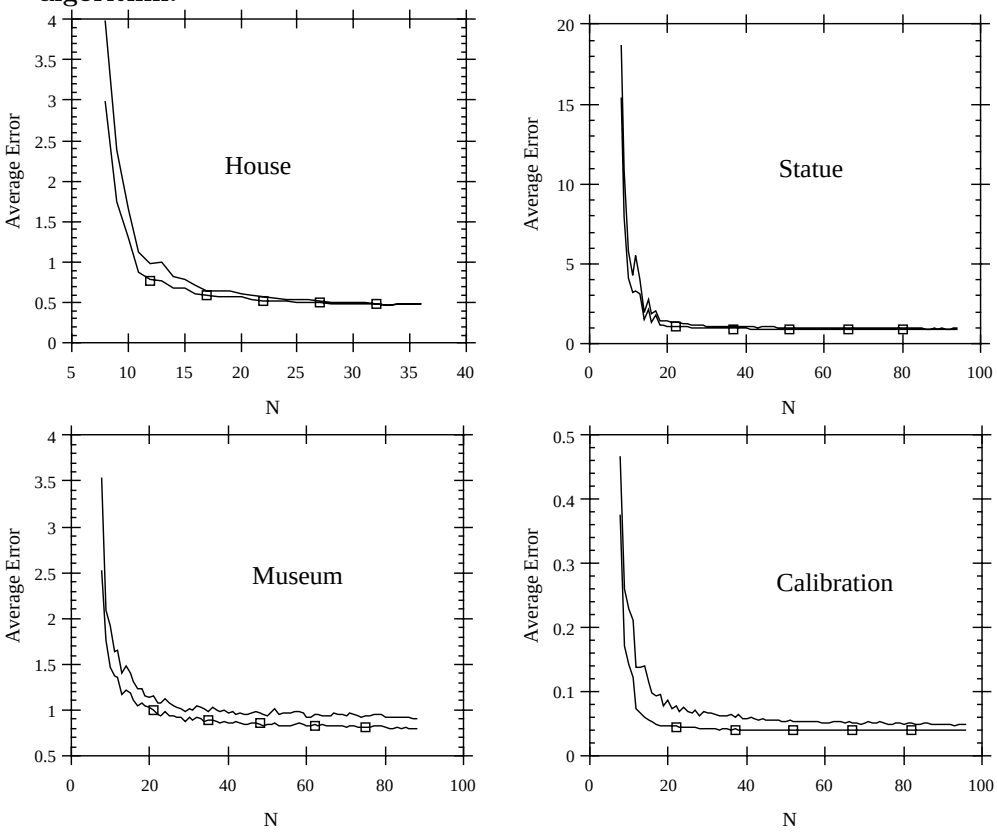
Graph 3 : Comparison of normalized and unnormalized 8-point algorithms.

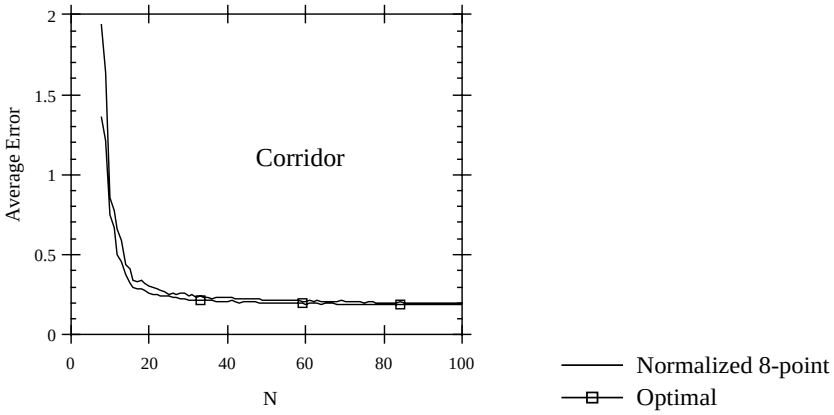


These set of graphs show the improvement achieved by normalization. The images used are from left-to-right and top-to-bottom : house, statue, museum, calibration, corridor. Note the differences in Y-scale for the different plots.

For some of the images the matched points were known with extreme accuracy (calibration image, corridor scene), whereas for others, the matches were less accurate (museum image). In all cases the normalized algorithm performs better than the unnormalized algorithm. In the cases of the calibration and corridor images the effect is not so great. In the case of the images with less accurate matches, the advantage of normalization is dramatic.

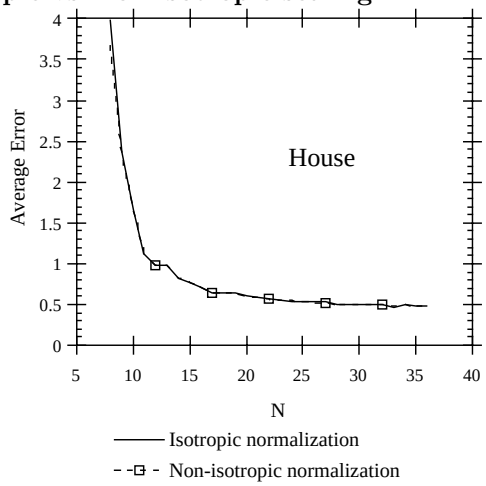
Graph 4 : Comparison of the 8-point algorithm with the optimal algorithm.





This is the same as the previous set of graphs, except that it compares the normalized 8-point algorithm with the optimal (minimized point displacement) algorithm. In all cases the normalized 8-point algorithm performs almost as well as the optimal algorithm.

Graph 5 : Isotropic vs. non-isotropic scaling.



The 8-pt algorithm with isotropic and non-isotropic scaling was compared. The two graphs are almost indistinguishable.

Graph 6 : Comparison with other algorithms

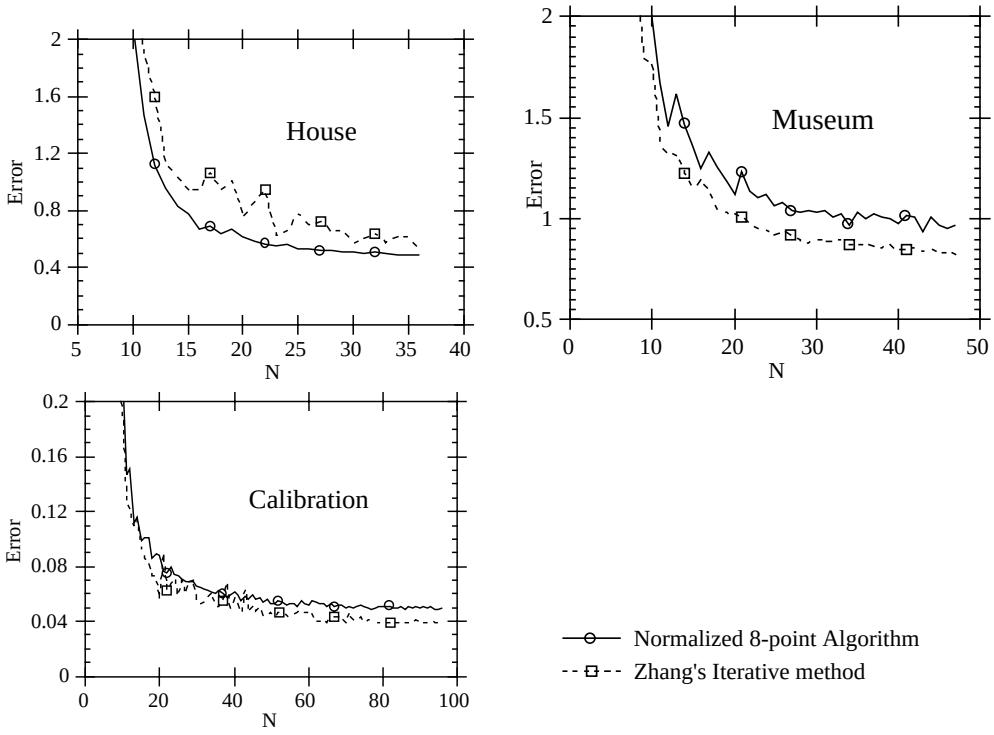
The papers ([9, 10]) gives details of several good algorithms, and the normalized 8-point algorithm was carefully compared with some of these. Two algorithms were tried :

1. The iterative algorithm minimizing the symmetric point-epipolar line distance in the two images.

2. The gradient-based method.

See section 7.1 for more details. For the tests, mmplementations of these algorithms supplied by Zhang in executable format were used. These are among the best algorithms available for computing the fundamental matrix.

On theoretical grounds, the second of these methods may be preferable, but in our experiments they performed almost identically. This is confirmed by [10]. Consequently, only only the results of the comparisons with the gradient-based method are shown in the following graphs, which compare the normalized 8-point algorithm with the gradient-based method.



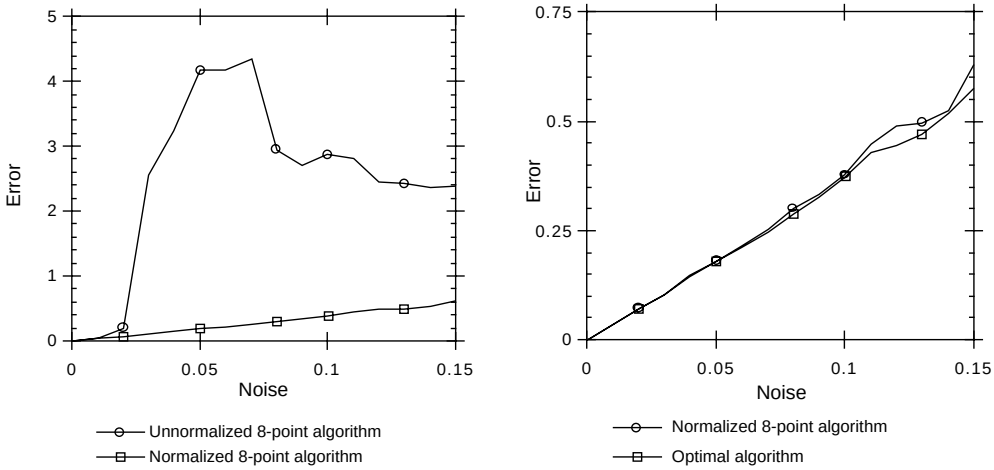
Results are shown for three of the data sets. In the other two cases (statue and corridor) the results of the two algorithms were almost indistinguishable. In fact, it is a curious thing that all algorithms (even the unnormalized 8-point algorithm) give very similar performance on these two data sets. In the three graphs shown, the normalized 8-point algorithm performs distinctly better than the iterative algorithms on the house data set, worse on the museum data set and just slightly worse on the calibration set. In this comparison the iterative algorithms were initialized using the unnormalized 8-point algorithm. Com-

parison with Graph 4 shows that they do not perform as well as the optimal algorithm. If the normalized 8-point algorithm is used for initialization, then the results improve and are not significantly different from those of the optimal algorithm. Once more Zhang's implementation was used for this test. Thus, in carrying out an iterative algorithm to find the fundamental matrix, good initialization seems to be more important than exactly which cost function is being minimized.

The normalized 8-point algorithm was also compared with the Least Median of Squares algorithm of Zhang, but the latter algorithm did not perform so well on our tests. This is probably because it is weeding out outliers. Outlier rejection has already been performed on the data sets using the techniques of ([7]) and all remaing points are used in evaluating the fit, including points that Zhang's LMedSq algorithm may have rejected.

The normalized 8-point algorithm was also compared with two algorithms supplied by Andrew Zisserman and Paul Beardsley. These are respectively the algorithms referred to as "Approximate Calibration" and "Iterative Linear" in section 7.1. The results of all 3 algorithms were roughly comparable, though insufficiently many test were run to reach a firm conclusion. The results of this test are reported in [25].

Graph 7 : 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 (5) were used for this purpose. Since reconstruction error is most appropriately measured in a 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 is a comparison of the unnormalized and the normalized 8-point algorithms. In the right hand graph, 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.

Even if extra accuracy is needed and an iterative algorithm is used, it is best to use the normalized, rather than the unnormalized 8-point algorithm to provide a starting point for iteration. Difficulties with stopping criteria, as well as the risk of finding a local minimum mean that the quality of the iteratively estimated result depends on the initial estimate.

The technique of data normalization described here is widely applicable to other problems. Among others it is directly applicable to the following problems: computing the projective transformations between point sets; estimating the trifocal tensor ([26]) and determining the camera matrix of a projective camera using the DLT algorithm ([27]).

9 Acknowledgements

I wish to thank all those people who supplied data and algorithms to me for the running of these tests. This includes most specifically Andrew Zisserman and Paul Beardsley who gave me the corridor and calibration jig image sets and

matched points; Zhengyou Zhang supplied implementations (currently available at http::/www.inria.fr/robotvis/personnel/zzhang/zzhang-eng.html) of other methods which were used for comparison; Jean-Claude Cottier gave me the museum and statue images and matched points and Long Quan and Boubakeur Boufama gave me the house images and matched points, and the use of their algorithm. In addition, Gerard Medioni supplied a coding of Berthold Horn's algorithm ([28] for reconstruction in the calibrated case. Evaluation of these methods in the calibrated case is a project for possible future work. Finally, thanks to Roger Mohr for making possible my sojourn in Grenoble allowing me the possibility to do this work.

References

[1] H.C. Longuet-Higgins, "A computer algorithm for reconstructing a scene from two projections," *Nature*, vol. 293, pp. 133–135, Sept 1981.

[2] 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*, 1992, pp. 563 – 578.

[3] R. Hartley, R. Gupta, and T. Chang, "Stereo from uncalibrated cameras," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 1992, pp. 761–764.

[4] Richard Hartley and Rajiv Gupta, "Computing matched-epipolar projections," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 1993, pp. 549 – 555.

[5] Stefan Carlsson, "Multiple image invariants using the double algebra," in *Proc. of the Second Europe-US Workshop on Invariance, Ponta Delgada, Azores*, October 1993, pp. 335–350.

[6] 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*, 1994, pp. 567–576.

[7] P. H. S. Torr and D. W. Murray, "Outlier detection and motion segmentation," in *Sensor Fusion VI*, P. S. Schenker, Ed. 1993, pp. 432–443, SPIE volume 2059, Boston.

[8] Z. Zhang, R. Deriche, O. Faugeras, and Q.-T. Luong, "A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry," *Artificial Intelligence Journal*, vol. 78, pp. 87–119, October 1995.

[9] Quang-Tuan Luong, Rachid Deriche, Olivier D. Faugeras, and Theodore Papadopoulo, "On determining the fundamental matrix: analysis of different methods and experimental results," Report RR-1894, INRIA, 1993.

[10] Zhengyou Zhang, "Determining the epipolar geometry and its uncertainty : A review," Report RR-2927, INRIA, 1996.

[11] Richard I. Hartley, "Euclidean reconstruction from uncalibrated views," in *Proc. of the Second Europe-US Workshop on Invariance, Ponta Delgada, Azores*, October 1993, pp. 187–202.

[12] 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*, 1994, pp. 85–96.

[13] S. J. Maybank, "The projective geometry of ambiguous surfaces," *Phil. Trans. R. Soc. Lond.*, vol. A 332, pp. 1 – 47, 1990.

[14] 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.

[15] K.E. Atkinson, *An Introduction to Numerical Analysis, 2nd Edition*, John Wiley and Sons, New York, 1989.

[16] 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.*, vol. PAMI-6, pp. 13–27, 1984.

[17] Richard I. Hartley, "Minimizing algebraic distance," in *Proc. DARPA Image Understanding Workshop*, 1997.

[18] Fred L. Bookstein, "Fitting conic sections to scattered data," *Computer Graphics and Image Processing*, vol. 9, pp. 56 – 71, 1979.

[19] Gene H. Golub and Charles F. Van Loan, *Matrix Computations, Second edition*, The Johns Hopkins University Press, Baltimore, London, 1989.

[20] R. I. Hartley, "Estimation of relative camera positions for uncalibrated cameras," in *Computer Vision - ECCV '92, LNCS-Series Vol. 588, Springer-Verlag*, 1992, pp. 579 – 587.

[21] Richard I. Hartley, "Projective reconstruction and invariants from multiple images," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 16, pp. 1036–1041, October 1994.

[22] I.E. Sutherland, "Sketchpad: A man-machine graphical communications system," Technical Report 296, MIT Lincoln Laboratories, 1963, Also published by Garland Publishing Inc, New York, 1980.

[23] Richard I. Hartley, "Projective reconstruction from line correspondences," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 1994, pp. 903–907.

[24] Richard I. Hartley and Peter Sturm, "Triangulation," in *Proc. ARPA Image Understanding Workshop*, 1994, pp. 957–966.

[25] R. I. Hartley, "In defence of the 8-point algorithm," in *Proc. International Conference on Computer Vision*, 1995, pp. 1064 – 1070.

[26] R. I. Hartley, "A linear method for reconstruction from lies and points," in *Proc. International Conference on Computer Vision*, 1995, pp. 882 – 887.

[27] I.E. Sutherland, "Three dimensional data input by tablet," *Proceedings of IEEE*, vol. Vol. 62, No. 4, pp. 453–461, April 1974.

[28] B. K. P. Horn, "Relative orientation," *International Journal of Computer Vision*, vol. 4, pp. 59 – 78, 1990.