CVIU - COMPUTER VISION AND IMAGE UNDERSTANDING, VOL. 68, NO. 2, PP. 146-157, NOVEMBER 1997.
CAIP - 6TH INTERNATIONAL CONFERENCE ON COMPUTER ANALYSIS OF IMAGES AND PATTERNS, PRAGUE, CZECH REPUBLIC, PP. 190-197, SEPTEMBER 1995.
ARPA IMAGE UNDERSTANDING WORKSHOP, MONTEREY, CALIFORNIA, PP. 957-966, NOVEMBER 1994.

Triangulation

Richard I. Hartley* and Peter Sturm†
* GE-CRD, Rm K1-5C39, P.O. Box 8, Schenectady, NY, 12301
hartley@bunyip.crd.ge.com

†GRAVIR-IMAG & INRIA Rhône-Alpes*
655, Avenue de l'Europe, 38330 Montbonnot, France
Peter.Sturm@inrialpes.fr

Abstract

In this paper, we consider the problem of finding the position of a point in space given its position in two images taken with cameras with known calibration and pose. This process requires the intersection of two known rays in space, and is commonly known as triangulation. In the absence of noise, this problem is trivial. When noise is present, the two rays will not generally meet, in which case it is necessary to find the best point of intersection. This problem is especially critical in affine and projective reconstruction in which there is no meaningful metric information about the object space. It is desirable to find a triangulation method that is invariant to projective transformations of space. This paper solves that problem by assuming a Gaussian noise model for perturbation of the image coordinates. The triangulation problem then may be formulated as a least-squares minimization problem. In this paper a non-iterative solution is given that finds the global minimum. It is shown that in certain configurations, local minima occur, which are avoided by the new method. Extensive comparisons of the new method with several other methods show that it consistently gives superior results.

1 The Triangulation Problem

We suppose that a point x in R^3 is visible in two images. The two camera matrices P and P' corresponding to the two images are supposed known. Let $\mathbf{u}$ and $\mathbf{u}'$ be projections of the point $\mathbf{x}$ in the two images. From this data, the two rays in space corresponding to the two image points may easily be computed. The triangulation problem is to find the intersection of the two lines in space. At first sight this is a trivial

*This work has been done in the context of the MOVI project which belongs to CNRS, INPG, INRIA and UJF.

problem, since intersecting two lines in space does not present significant difficulties. Unfortunately, in the presence of noise these rays can not be guaranteed to cross, and we need to find the best solution under some assumed noise model.

A commonly suggested method [2] is to choose the mid-point of the common perpendicular to the two rays (the *mid-point method*). Perhaps a better choice would be to divide the common perpendicular in proportion to the distance from the two camera centers, since this would more closely equalize the angular error. Nevertheless, this method will not give optimal results, because of various approximations (for instance the angles will not be precisely equal in the two cases). In the case of projective reconstruction, or affine reconstruction however, the camera matrices, will be known in a projective frame of reference, in which concepts such as common perpendicular, or mid-point (in the projective case) have no sense. In this case, the simple mid-point method here will not work.

The importance of a good method for triangulation is clearly shown by Beardsley et al. who demonstrate that the mid-point method gives bad results. In [2, 3] they suggest an alternative method based on "quasi-Euclidean" reconstruction. In this method, an approximation to the correct Euclidean frame is selected and the mid-point method is carried out in this frame. The disadvantage of this method is that an approximate calibration of the camera is needed. It is also clearly sub-optimal.

This paper is an extended version of [9] which describes a new algorithm that gives an optimal global solution to the triangulation problem, equally valid in both the affine and projective reconstruction cases. The solution relies on the concepts of epipolar correspondence and the fundamental matrix [4]. The algorithm is non-iterative and simple in concept, relying on techniques of elementary calculus to minimize the chosen cost function. It is also moderate in computation requirements. In a series of experiments, the algorithm is extensively tested against many other methods of triangulation, and found to give consistent superior performance. No knowledge of camera calibration is needed.

The triangulation problem is a small cog in the machinery of computer vision, but in many applications of scene reconstruction it is a critical one, on which ultimate accuracy depends [2].

2 Transformational Invariance

In the last few years, there has been considerable interest in the subject of affine or projective reconstruction [4, 8, 10, 12, 16, 13, 15]. In such reconstruction methods, a 3D scene is to be reconstructed up to an unknown transformation from the given class. Normally, in such a situation, instead of knowing the correct pair of camera matrices P and P', one has a pair PH^{-1} and $P'H^{-1}$ where H is an unknown transformation of the considered class.

For instance, in the method of projective reconstruction given in [8] one starts with a set of image point correspondences $\mathbf{u}_i \leftrightarrow \mathbf{u}'_i$. From these correspondences, one can compute the fundamental matrix F, and hence a pair of camera matrices $\hat{P}$ and $\hat{P}'$. In the method of [8], the pair of camera matrices differ from the true ones by an unknown transformation H, and $\hat{P}$ is normalized so that $\hat{P} = (I \mid 0)$. Finally, the 3D space points can be computed by triangulation. If desired, the true Euclidean reconstruction of the scene may then be accomplished by the use of ground control points to determine the unknown transformation, H, and hence the true camera matrices, P and P'. Similarly, in the paper [7] one of the steps of a projective reconstruction algorithm is the reconstruction of points from three views, normalized so that the first camera matrix has the form $(I \mid 0)$. Given three or more views, an initial projective reconstruction may be transformed to a Euclidean reconstruction under the assumption that the images are taken all with the same camera [5].

A desirable feature of the method of triangulation used is that it should be invariant under transformations of the appropriate class. Thus, denote by τ a triangulation method used to compute a 3D space point $\mathbf{x}$ from a point correspondence $\mathbf{u} \leftrightarrow \mathbf{u}'$ and a pair of camera matrices P and P'. We write

$$\mathbf{x} = \tau(\mathbf{u}, \mathbf{u}', P, P') \ .$$

The triangulation is said to be invariant under a transformation H if

$$\tau(\mathbf{u}, \mathbf{u}', P, P') = H^{-1}\tau(\mathbf{u}, \mathbf{u}', PH^{-1}, P'H^{-1}) \ .$$

This means that triangulation using the transformed cameras results in the transformed point. If the camera matrices are known only up to an affine (or projective) transformation, then it is clearly desirable to use an affine- (resp. projective-) invariant triangulation method to compute the 3D space points.

3 The Minimization Criterion

We assume that the camera matrices, and hence the fundamental matrix, are known exactly, or at least with great accuracy compared with a pair of matching points in the two images. A formula is given in [6] for computing the fundamental matrix given a pair of camera matrices. The two rays corresponding to a matching pair of points $\mathbf{u} \leftrightarrow \mathbf{u}'$ will meet in space if and only if the points satisfy the familiar [11] relationship

$$\mathbf{u}'^{T} F \mathbf{u} = 0 \ . \tag{1}$$

It is clear, particularly for projective reconstruction, that it is inappropriate to minimize errors in the 3D projective space, $\mathcal{P}^3$. For instance, the method that finds the mid-point of the common perpendicular to the two rays in space is not suitable for projective reconstruction, since concepts such as distance and perpendicularity are not valid in the context of projective geometry. In fact, in projective reconstruction, this method will give different results depending on which particular projective reconstruction is considered – the method is not projective-invariant.

Normally, errors occur not in placement of a feature in space, but in its location in the two images, due to digitization errors, or the exact identification of a feature in the image. It is common to assume that features in the images are subject to Gaussian noise which displaces the feature from its correct location in the image. We assume that noise model in this paper.

A typical observation consists of a noisy point correspondence $\mathbf{u} \leftrightarrow \mathbf{u}'$ which does not in general satisfy the epipolar constraint (1). In reality, the correct values of the corresponding image points should be points $\hat{\mathbf{u}} \leftrightarrow \hat{\mathbf{u}}'$ lying close to the measured points $\mathbf{u} \leftrightarrow \mathbf{u}'$ and satisfying the equation $(\hat{\mathbf{u}}')^{\mathsf{T}} F\hat{\mathbf{u}}$ exactly. We seek the points $\hat{\mathbf{u}}$ and $\hat{\mathbf{u}}'$ that minimize the function

$$d(\mathbf{u}, \hat{\mathbf{u}})^2 + d(\mathbf{u}', \hat{\mathbf{u}}')^2 \ , \tag{2}$$

where $d(*, *)$ represents Euclidean distance, subject to the epipolar constraint

$$(\hat{\mathbf{u}}')^{\mathsf{T}} F\hat{\mathbf{u}} = 0 \ .$$

Assuming a Gaussian error distribution, the points $\hat{\mathbf{u}}'$ and $\hat{\mathbf{u}}$ are the most likely values for true image point correspondences. Once $\hat{\mathbf{u}}'$ and $\hat{\mathbf{u}}$ are found, the point $\mathbf{x}$ may be found by any triangulation method, since the corresponding rays will meet precisely in space.

4 An Optimal Method of Triangulation

In this section, we describe a method of triangulation that finds the global minimum of the cost function (2) using a non-iterative algorithm. If the Gaussian noise model can be assumed to be correct, this triangulation method is then provably optimal. This new method will be referred to as the **Polynomial** method, since it requires the solution of a sixth order polynomial.

4.1 Reformulation of the Minimization Problem

Given a measured correspondence $\mathbf{u} \leftrightarrow \mathbf{u}'$, we seek a pair of points $\hat{\mathbf{u}}$ and $\hat{\mathbf{u}}'$ that minimize the sum of squared distances (2) subject to the epipolar constraint $(\hat{\mathbf{u}}')^{\mathsf{T}} F\hat{\mathbf{u}} = 0$.

Any pair of points satisfying the epipolar constraint must lie on a pair of corresponding epipolar lines in the two images. Thus, in particular, the optimum point $\hat{\mathbf{u}}$ lies on an epipolar line $\hat{\lambda}$ and $\hat{\mathbf{u}}'$ lies on the corresponding epipolar line $\hat{\lambda}'$.

Now, we consider a pair of corresponding epipolar lines λ and λ'. Of all pairs of points on λ and λ' it is of course the pair of orthogonal projections of $\mathbf{u}$ on λ respectively $\mathbf{u}'$ on λ' which minimizes the sum of squared distances (2). Let ($\bar{\mathbf{u}}$,$\bar{\mathbf{u}}'$) be the pair of these orthogonal projections. We may write $d(\mathbf{u}, \bar{\mathbf{u}}) = d(\mathbf{u}, \lambda)$, where $d(\mathbf{u}, \lambda)$ represents the perpendicular distance from the point $\mathbf{u}$ to the line λ. A similar expression holds for $d(\mathbf{u}', \bar{\mathbf{u}}')$.

In view of the previous paragraph, we may reformulate the minimization problem as follows. We seek to minimize

$$d(\mathbf{u}, \lambda)^2 + d(\mathbf{u}', \lambda')^2 \tag{3}$$

where λ and λ' range over all choices of corresponding epipolar lines.

Suppose we have determined the pair of corresponding epipolar lines $\hat{\lambda}$ and $\hat{\lambda}'$ which minimize (3). The searched points $\hat{\mathbf{u}}$ and $\hat{\mathbf{u}}'$ are then just the orthogonal projections of $\mathbf{u}$ on $\hat{\lambda}$ respectively $\mathbf{u}'$ on $\hat{\lambda}'$.

Our strategy for minimizing (3) is as follows:

1. Parameterize the pencil of epipolar lines in the first image by a parameter t. Thus an epipolar line in the first image may be written as $\lambda(t)$.
2. Using the fundamental matrix F, compute the corresponding epipolar line $\lambda'(t)$ in the second image.
3. Express the distance function $d(\mathbf{u}, \lambda(t))^2 + d(\mathbf{u}', \lambda'(t))^2$ explicitly as a function of t.
4. Find the value of t that minimizes this function.

In this way, the problem is reduced to that of finding the minimum of a function of a single variable, t. It will be seen that for a suitable parameterization of the pencil of epipolar lines the distance function is a rational polynomial function of t. Using techniques of elementary calculus, the minimization problem reduces to finding the real roots of a polynomial of degree 6.

4.2 Details of Minimization

If both of the image points correspond with the epipoles, then the point in space lies on the line joining the camera centers. In this case it is impossible to determine the position of the point in space. If only one of the corresponding point lies at an epipole, then we conclude that the point in space must coincide with the other camera center. Consequently, we assume in the following that neither of the two image points $\mathbf{u}$ and $\mathbf{u}'$ corresponds with an epipole.

In this case, we may simplify the analysis by applying a rigid transformation to each image in order to place both points $\mathbf{u}$ and $\mathbf{u}'$ at the origin, $(0, 0, 1)^T$ in homogeneous coordinates. Furthermore, the epipoles may be placed on the x-axis at points $(1, 0, f)^T$ and $(1, 0, f')^T$ respectively. A value f equal to 0 means that the epipole is at infinity.

The details on how to determine these rigid transformations are given in 4.3. In the following, we assume that in homogeneous coordinates, $\mathbf{u} = \mathbf{u}' = (0, 0, 1)^T$ and that the two epipoles are at points $(1, 0, f)^T$ and $(1, 0, f')^T$.

Applying these rigid transformations has no effect on the sum-of-squares distance function (2), and hence does not change the minimization problem. However, the fundamental matrix must be adapted according to these transformations. Since $F(1, 0, f)^T = (1, 0, f')F = \mathbf{0}$, the fundamental matrix has a special form (how to compute this matrix from the original fundamental matrix is described in 4.3):

$$F = \begin{pmatrix} ff'd & -f'c & -f'd \\ -fb & a & b \\ -fd & c & d \end{pmatrix} \quad . \tag{4}$$

Consider an epipolar line in the first image passing through the point $(0, t, 1)^T$ (still in homogeneous coordinates) and the epipole $(1, 0, f)^T$. We denote this epipolar line by $\lambda(t)$. The vector representing this line is given by the cross product $(0, t, 1)^T \times (1, 0, f)^T = (tf, 1, -t)^T$, so the squared distance from the line to the origin is

$$d(\mathbf{u}, \lambda(t))^2 = \frac{t^2}{1 + (tf)^2} \quad .$$

Using the fundamental matrix to find the corresponding epipolar line in the other image, we see that

$$\lambda'(t) = F(0, t, 1)^T = (-f'(ct + d), at + b, ct + d)^T \quad .$$

This is the representation of the line $\lambda'(t)$ as a homogeneous vector. The squared distance of this line from the origin is equal to

$$d(\mathbf{u}', \lambda'(t))^2 = \frac{(ct + d)^2}{(at + b)^2 + f'^2(ct + d)^2} \quad .$$

The total squared distance is therefore given by

$$s(t) = \frac{t^2}{1 + (tf)^2} + \frac{(ct + d)^2}{(at + b)^2 + f'^2(ct + d)^2} \quad . \tag{5}$$

Our task is to find the minimum of this function.

We may find the minimum using techniques of elementary calculus, as follows. We compute the derivative

$$s'(t) = \frac{2t}{(1 + (tf)^2)^2} - \frac{2(ad - bc)(at + b)(ct + d)}{((at + b)^2 + f'^2(ct + d)^2)^2} \quad . \tag{6}$$

Maxima and minima of $s(t)$ will occur when $s'(t) = 0$. Collecting the two terms in $s'(t)$ over a common denominator, and equating the numerator to 0 gives a condition

$$\begin{aligned} r(t) &= t((at + b)^2 + f'^2(ct + d)^2)^2 - (ad - bc)(1 + (tf)^2)^2(at + b)(ct + d) \\ &= 0 \quad . \end{aligned} \tag{7}$$

The minima and maxima of $s(t)$ will occur at the roots of this polynomial. This is a polynomial of degree 6, which may have up to 6 real roots, corresponding to 3 minima and 3 maxima of the function $s(t)$. The absolute minimum of the function $s(t)$ may be found by finding the roots of $r(t)$ and evaluating the function $s(t)$ given by (5) at each of the real roots. More simply, one checks the value of $s(t)$ at the real part of each root (complex or real) of $r(t)$, which saves the trouble of determining if a root is real or complex. One should also check the asymptotic value of $s(t)$ as $t \to \infty$ to see if the minimum distance occurs when $t = \infty$, corresponding to an epipolar line $u = 1/f$ in the first image.

4.3 Determining the Rigid Transformations

We first carry out a translation which takes the point $\mathbf{u}$ to the origin. If $\mathbf{u}$ is given by $\mathbf{u} = (u_1, u_2, 1)^T$, the translation is represented by

$$L = \begin{pmatrix} 1 & 0 & -u_1 \\ 0 & 1 & -u_2 \\ 0 & 0 & 1 \end{pmatrix} \quad .$$

Now, in order to place the epipole $\mathbf{e}$ on the x-axis, we rotate around the origin by an angle Θ which is determined as shown in the following. A rotation around the origin can be represented by a matrix

$$R = \begin{pmatrix} \cos\Theta & -\sin\Theta & 0 \\ \sin\Theta & \cos\Theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \ .$$

The (by L translated) epipole $\mathbf{e} = (e_1, e_2, e_3)^T$ is rotated on the x-axis, if

$$RL\mathbf{e} \approx (1, 0, f)^T$$

for some f. Developing the left side, we obtain the following equation

$$\sin\Theta(e_1 - e_3 u_1) + \cos\Theta(e_2 - e_3 u_2) = 0 \ ,$$

which allows us to determine the rotation angle Θ.

Let L' and R' be the corresponding transformations in the second image. The fundamental matrix for the transformed images (the same as in (4)) is then given by

$$F = R' L'^{-T} F_0 L^{-1} R^T \ ,$$

where F_0 here denotes the fundamental matrix before carrying out the transformations.

4.4 Local Minima

The fact that $r(t)$ in (7) has degree 6 means that $s(t)$ may have as many as three minima. In fact, this is indeed possible, as the following case shows. Setting $f = f' = 1$ and $a = 2, b = 3, c = 3, d = 4$ gives

$$F = \begin{pmatrix} 4 & -3 & -4 \\ -3 & 2 & 3 \\ -4 & 3 & 4 \end{pmatrix}$$

and a function

$$s(t) = \frac{t^2}{1+t^2} + \frac{(3t+4)^2}{(2t+3)^2 + (3t+4)^2}$$

with graph as shown in Fig 1 (a) (in this graph and also in (b) we make the substitution $t = tan(\theta)$ and plot for θ in the range $-\pi/2 \leq \theta \leq \pi/2$, so as to show the whole infinite range of t). The three minima are clearly shown.

As a second example, we consider the case where $f = f' = 1$, and $a = 2, b = -1, c = 1, d = 0$, i.e.

$$F = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 2 & -1 \\ 0 & 1 & 0 \end{pmatrix} \ .$$

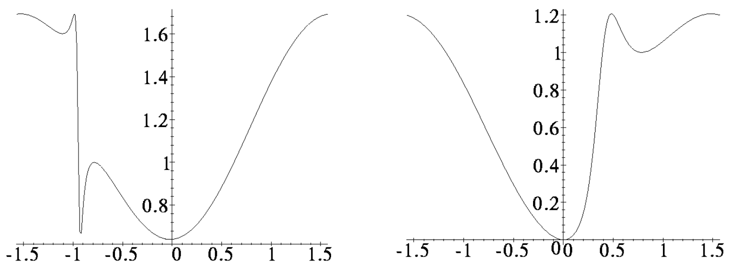


Figure 1: (a) Example of a cost function with three minima. (b) This is the cost function for a perfect point match, which nevertheless has two minima

In this case, the function $s(t)$ is given by

$$s(t) = \frac{t^2}{t^2+1} + \frac{t^2}{t^2+(2t-1)^2} \quad .$$

Both terms of the cost function vanish for a value of $t = 0$, which means that the corresponding points $\mathbf{u}$ and $\mathbf{u}'$ exactly satisfy the epipolar constraint. This can be verified by observing that $(\mathbf{u}')^\top F\mathbf{u} = 0$. Thus the two points are exactly matched. A graph of the cost function $s(t)$ is shown in Fig 1 (b). One sees apart from the absolute minimum at $t = 0$ also a local minimum at $t = 1$ ($\theta = \Pi/4$). Thus, even in the case of perfect matches local minima may occur. This example shows that an algorithm that attempts to minimize the cost function (2), or equivalently (3) by an iterative search beginning from an arbitrary initial point is in danger of finding a local minimum, even in the case of perfect point matches.

4.5 Optimality

Under the assumption of an unbiased Gaussian noise model, the most probable reconstruction is the one that minimizes the sum of squared distances between re-projected points and measured image points. In this sense, our method gives optimal results if the projection matrices (respectively the fundamental matrix) are exactly known: the constraint that re-projected points must lie on corresponding epipolar lines is automatically fulfilled for any reconstruction; here we only use it to obtain the parameterization in 4.2. Of course, in practice the projection matrices or fundamental matrix are not exactly known. Correcting these improves the accuracy of the reconstruction, but this requires iterative methods (*cf.* 5.5) and usually a good initialization.

5 Other Triangulation Methods

In this section, we discuss several other triangulation methods that will be compared with the polynomial method.

5.1 Linear Triangulation

The linear triangulation method is the most common one, described for instance in [8]. Suppose $\mathbf{u} = P\mathbf{x}$. We write in homogeneous coordinates $\mathbf{u} = w(u, v, 1)^T$, where (u, v) are the observed point coordinates and w is an unknown scale factor. Now, denoting by $\mathbf{p}_i^T$ the i-th row of the matrix P, the equation $\mathbf{u} = P\mathbf{x}$ may be written as follows:

$$wu = \mathbf{p}_1^T\mathbf{x} \quad , \quad wv = \mathbf{p}_2^T\mathbf{x} \quad , \quad w = \mathbf{p}_3^T\mathbf{x} \ .$$

Eliminating w using the third equation, we arrive at

$$\begin{array}{l} u\mathbf{p}_3^T\mathbf{x} = \mathbf{p}_1^T\mathbf{x} \\ v\mathbf{p}_3^T\mathbf{x} = \mathbf{p}_2^T\mathbf{x} \quad . \end{array} \tag{8}$$

From two views, we obtain a total of 4 linear equations in the coordinates of $\mathbf{x}$, which may be written in the form $A\mathbf{x} = \mathbf{0}$ for a suitable 4×4 matrix, A. These equations define $\mathbf{x}$ only up to an indeterminant scale factor, and we seek a non-zero solution for $\mathbf{x}$. Of course, with noisy data, the equations will not be satisfied precisely, and we seek a best solution.

The Linear-Eigen method. There are many ways to solve for $\mathbf{x}$ to satisfy $A\mathbf{x} = \mathbf{0}$. In one popular method, one finds $\mathbf{x}$ to minimize $||A\mathbf{x}||$ subject to the condition $||\mathbf{x}|| = 1$. The solution is the unit eigenvector corresponding to the smallest eigenvalue of the matrix $A^T A$. This problem may be solved using the Singular Value Decomposition, or Jacobi's method for finding eigenvalues of symmetric matrices [14, 1].

The Linear-LS method. By setting $\mathbf{x} = (x, y, z, 1)^T$ one reduces the set of homogeneous equations, $A\mathbf{x} = \mathbf{0}$ to a set of 4 non-homogeneous equations in 3 unknowns. One can find a least-squares solution to this problem by the method of pseudo-inverses, or by using the Singular Value Decomposition [14, 1].

Discussion. These two methods are quite similar, but in fact have quite different properties in the presence of noise. The **Linear-LS** method assumes that the solution point $\mathbf{x}$ is not at infinity, for otherwise we could not assume that $\mathbf{x} = (x, y, z, 1)^T$. This is a disadvantage of this method when we are seeking to carry out a projective reconstruction, when reconstructed points may lie on the plane at infinity. On the other hand, neither of these two linear methods is quite suitable for projective reconstruction, since they are non projective-invariant. To see this, suppose that the camera matrices P and P' are replaced by PH^{-1} and $P'H^{-1}$. One sees that in this case the matrix of equations, A becomes AH^{-1}. A point $\mathbf{x}$ such that $A\mathbf{x} = \epsilon$ for the original problem corresponds to a point $H\mathbf{x}$ satisfying $(AH^{-1})(H\mathbf{x}) = \epsilon$ for the transformed problem. Thus, there is a one-to-one correspondence between points $\mathbf{x}$ and $H\mathbf{x}$ giving the same error. However, neither the condition $||\mathbf{x}|| = 1$ nor the condition $\mathbf{x} = (x, y, z, 1)^T$ is invariant under application of the projective transformation H. Thus, in general the point $\mathbf{x}$ solving the original problem will not correspond to a solution $H\mathbf{x}$ for the transformed problem.

For affine transformations, on the other hand, the situation is different. In fact, although the condition $||\mathbf{x}|| = 1$ is not preserved under affine transformation, the condition $\mathbf{x} = (x, y, z, 1)^T$ is preserved, since for an affine transformation, $H(x, y, z, 1)^T = (x', y', z', 1)^T$. This means that there is a one-to-one correspondence between a vector $\mathbf{x} = (x, y, z, 1)^T$ such that $A(x, y, z, 1)^T = \epsilon$ and the vector $H\mathbf{x} = (x', y', z', 1)^T$ such that $(AH^{-1})(x', y', z', 1)^T = \epsilon$. The error is the same for corresponding points. Thus, the points that minimize the error $||\epsilon||$ correspond as well. Hence, the method **Linear-LS** is affine-invariant, whereas the method **Linear-Eigen** is not. These conclusions are confirmed by the experimental results.

5.2 Iterative Linear Methods

A cause of inaccuracy in the two methods **Linear-LS** and **Linear-Eigen** is that the value being minimized $||A\mathbf{x}||$ has no geometric meaning, and certainly does not correspond to the cost function (2). In addition, multiplying each of the equations (rows of A) by some weight will change the solution. The idea of the iterative linear method is to change the weights of the linear equations adaptively so that the weighted equations correspond to the errors in the image coordinate measurements.

In particular, consider the first of the equations (8). In general, the point $\mathbf{x}$ we find will not satisfy this equation exactly – rather, there will be an error $\epsilon = u\mathbf{p}_3^T\mathbf{x} - \mathbf{p}_1^T\mathbf{x}$. What we really want to minimize however, is the difference between the measured image coordinate value u and the projection of $\mathbf{x}$, which is given by $\mathbf{p}_1^T\mathbf{x}/\mathbf{p}_3^T\mathbf{x}$. Specifically, we wish to minimize $\epsilon' = \epsilon/\mathbf{p}_3^T\mathbf{x} = u - \mathbf{p}_1^T\mathbf{x}/\mathbf{p}_3^T\mathbf{x}$. This means that if the equation had been weighted by the factor $1/w$ where $w = \mathbf{p}_3^T\mathbf{x}$, then the resulting error would have been precisely what we wanted to minimize. The same weight $1/w$ is the correct one to apply to the second equation of (8). For a second image, the correct weight would be $1/w'$ where $w' = (\mathbf{p}'_3)^\mathsf{T}\mathbf{x}$. Of course, we can not weight the equations in this manner because the weights depend on the value of $\mathbf{x}$ which we do not know until after we have solved the equations. Therefore, we proceed iteratively to adapt the weights. We begin by setting $w_0 = w'_0 = 1$, and we solve the system of equations to find a solution $\mathbf{x}_0$. This is precisely the solution found by the linear method **Linear-Eigen** or **Linear-LS**, whichever is being used. Having found $\mathbf{x}_0$ we may compute the weights.

We repeat this process several times, at the i-th step multiplying the equations (8) for the first view by $1/w_i$ where $w_i = \mathbf{p}_3\mathbf{x}_{i-1}$ and the equations for the second view by $1/w'_i$ where $w'_i = \mathbf{p}'_3\mathbf{x}_{i-1}$ using the solution $\mathbf{x}_{i-1}$ found in the previous iteration. Within a few iterations this process will converge (one hopes) in which case we will have $\mathbf{x}_i = \mathbf{x}_{i-1}$ and so $w_i = \mathbf{p}_3^T\mathbf{x}_i$. The error (for the first equation of (8) for example) will be $\epsilon_i = u - \mathbf{p}_1^T\mathbf{x}_i/\mathbf{p}_3^T\mathbf{x}_i$ which is precisely the error in image measurements as in (2).

This method may be applied to either the **Linear-Eigen** or **Linear-LS** method. The corresponding methods will be called **Iterative-Eigen** and **Iterative-LS** respectively. The advantage of these methods over other iterative least-squares minimization methods such as a Levenberg-Marquardt (**LM**) iteration [14] is that they are very simple to program. In fact, they require only a trivial adaptation to the linear methods. There is no need for any separate initialization method, as is often required by **LM** (see 5.5).

Furthermore the decision on when to stop iterating (convergence) is simple. One stops when the change in the weights is small. Exactly when to stop is not critical, since the change in the reconstructed points $\mathbf{x}$ is not very sensitive to small changes in the weights. The disadvantage of this method is that it sometimes fails to converge. In unstable situations, such as when the points are near the epipoles, this occurs sufficiently often to be a problem (perhaps for 5% of the time). If this method is to be used in such unstable circumstances, then a fall-back method is necessary. In the experiments, we have used the optimal **Polynomial** method as a backup in case convergence has not occurred within 10 iterations. In this way the statistics are not negatively biased by occasional very bad results, due to non-convergence.

Despite the similarities of the properties of the **Iterative-LS** method with a direct non-linear least squares minimization of the goal function (2), it is not identical. Because the **Iterative-LS** method separates the two steps of computing $\mathbf{x}$ and the weights w and w', the result is slightly different. In fact the three methods **Iterative-LS**, **Iterative-Eigen** and **LM** are distinct. In particular, the methods **Iterative-LS** and **Iterative-Eigen** are not projective-invariant, though experiments show that they are quite insensitive to projective transformation. Of course, **Iterative-LS** is affine-invariant, just as **Linear-LS** is.

Experiments show that the iterative methods **Iterative-LS** and **Iterative-Eigen** perform substantially better than the corresponding non-iterative linear methods.

5.3 Mid-point Method

A commonly suggested method for triangulation is to find the mid-point of the common perpendicular to the two rays corresponding to the matched points. This method is relatively easy to compute using a linear algorithm. However, ease of computation is almost its only virtue. This method is neither affine nor projective-invariant, since perpendicularity is not an affine and mid-point not a projective concept. It is seen to behave very poorly indeed under projective and affine transformation, and is by far the worst of the methods considered here in this regard. For the record, we outline an algorithm to compute this mid-point. Let $P = (M \mid -M\mathbf{c})$ be a decomposition of the first camera matrix. The center of the camera is $\begin{pmatrix} \mathbf{c} \\ 1 \end{pmatrix}$ in homogeneous coordinates. Furthermore, the point at infinity that maps to a point $\mathbf{u}$ in the image is given by $\begin{pmatrix} M^{-1}\mathbf{u} \\ 0 \end{pmatrix}$. Therefore, any point on the ray mapping to $\mathbf{u}$ may be written in the form $\begin{pmatrix} \mathbf{c} + \alpha M^{-1}\mathbf{u} \\ 1 \end{pmatrix}$ or in non-homogeneous coordinates, $\mathbf{c} + \alpha M^{-1}\mathbf{u}$, for some α. Given two images, the two rays must meet in space, which leads to an equation $\alpha M^{-1}\mathbf{u} - \alpha' M'^{-1}\mathbf{u}' = \mathbf{c}' - \mathbf{c}$. This gives three equations in two unknowns (the values of α and α') which we may solve using linear least-squares methods. This minimizes the squared distance between the two rays. The mid point between the two rays is then given by $(\mathbf{c} + \alpha M^{-1}\mathbf{u} + \mathbf{c}' + \alpha' M'^{-1}\mathbf{u})/2$.

5.4 Minimizing the Sum of the Magnitudes of Distances

Instead of minimizing the square sum of image errors, it is possible to adapt the polynomial method to minimize the sum of absolute values of the distances, instead of the squares of distances. This method will be called **Poly-Abs**.

The quantity to be minimized is $d(\mathbf{u}, \lambda) + d(\mathbf{u}^{'}, \lambda^{'})$ which, as a function of t, is expressed by

$$s_2(t) = \frac{|t|}{\sqrt{1+(tf)^2}} + \frac{|ct+d|}{\sqrt{(at+b)^2 + f'^2(ct+d)^2}} \ .$$

The first derivative is of the form

$$s_2'(t) = \omega_1 \frac{1}{(1+(tf)^2)^{3/2}} - \omega_2 \frac{(ad-bc)(at+b)}{((at+b)^2 + f'^2(ct+d)^2)^{3/2}} \tag{9}$$

where ω_1 and ω_2 are equal to -1 or 1, depending on the signs of t and $(ct+d)$ respectively.

Setting the derivative equal to zero, separating the two terms on opposite sides of the equal sign and squaring to remove the square roots gives

$$\frac{1}{(1+(tf)^2)^3} = \frac{(ad-bc)^2(at+b)^2}{((at+b)^2 + f'^2(ct+d)^2)^3}$$

which finally leads to a polynomial of degree 8 in t. We evaluate $s_2(t)$ at the roots of this polynomial to find the global minimum of $s_2(t)$.

5.5 Photogrammetry

In the photogrammetric community, "triangulation" means reconstruction from several, usually more than two images [17]. Most of the proposed methods are designed for calibrated cameras, i.e. are only applicable in the case of Euclidean reconstruction. The most general method however, the *bundle adjustment with self-calibration*, is easily adapted to the case of projective reconstruction. Here, the coordinates of the reconstructed points are estimated iteratively (usually by a Levenberg-Marquardt based method (**LM**)) with the objective of minimizing the sum of squared distances between measured image points and the re-projected 3D points. This is exactly the same minimization criterion as the cost function (2), and therefore the same results should be found.

An advantage of this method is that corrections of the projection matrices are easily incorporated into the reconstruction process. Since the given projection matrices are never exact in practice, this method may be used to enhance an initial reconstruction.

However, the major drawback is the need of a good initialization for the reconstruction. Thus, this method can not be considered as a stand-alone reconstruction technique. Therefore, we do not consider it in the experiments.

6 Experimental Evaluation of Triangulation Methods

A large number of experiments were carried out to evaluate the different methods described above. We concentrated on two configurations.

Configuration 1 The first configuration was meant to simulate a situation similar to a robot moving down a corridor, looking straight ahead. This configuration is shown in the left part of Fig 2. In this case, the two epipoles are close to the center of the images. For points lying on the line joining the camera centers depth can not be determined, and for points close to this line, reconstruction becomes difficult. Simulated experiments were carried out for points at several distances in front of the front camera.

Numerical values we used are as follows:

- The distance between the two cameras is 1 unit.
- The radius of the sphere of observed points is 0.05 units.
- The distance between the center of the point sphere and the projection center of the front camera is chosen as 0.15 or 0.55 units. The center of the sphere lies on the baseline of the two cameras.
- The cameras have the same calibration matrix

$$K = \begin{pmatrix} 700 & 0 & 0 \\ 0 & 700 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad .$$

Configuration 2 In the other configuration, the pair of cameras were almost parallel, as in an aerial imaging situation. The points were assumed to be approximately equidistant from both cameras, with several different distances being tried. This configuration is shown in the right-hand part of Fig 2. This was a fairly benign configuration for which most of the methods worked relatively well.

In each set of experiments, 50 points were chosen at random in the common field of view. For each of several noise levels varying from 1 to 10 pixels (in a 700×700 image), each point was reconstructed 100 times, with different instances of noise chosen from a Gaussian random variable with the given standard deviation (noise level). For each reconstructed point both the 3D reconstruction error, and the 2D residual error (after re-projection of the point) were measured. The errors shown are the median erros. Average errors were also computed. In this latter case the graphs (not shown in this paper) had the same general form and led to the same conclusions. However, they were a little less smooth than the graphs shown here, being more sensitive to occasional gross errors.

To measure the invariance to transformation, an affine or projective transformation was applied to each camera matrix. The projective and affine transformations were chosen so that one of the camera matrices was of the form $(I \mid 0)$. This is the normalized form of a camera matrix used in the projective reconstruction method of [8]. It represents a significant distortion, since the actual camera matrix was (by construction) of the form $(M \mid \mathbf{0})$, where M was a diagonal matrix $\mathrm{diag}(700, 700, 1)$.

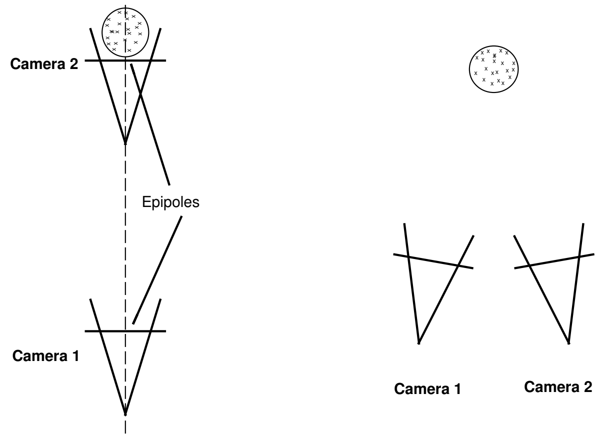
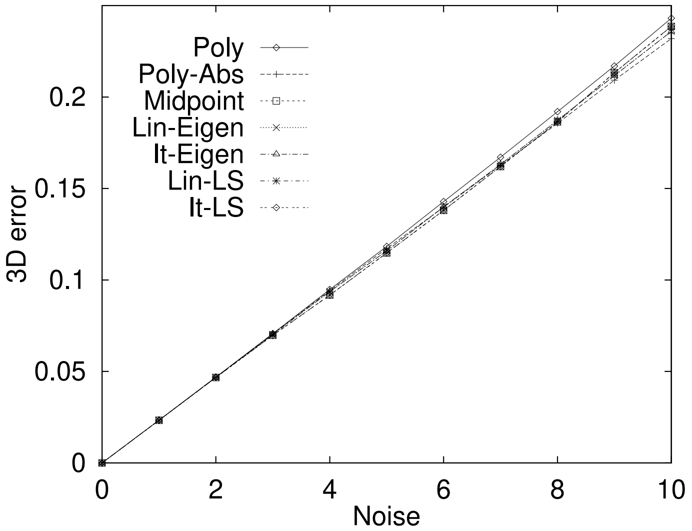
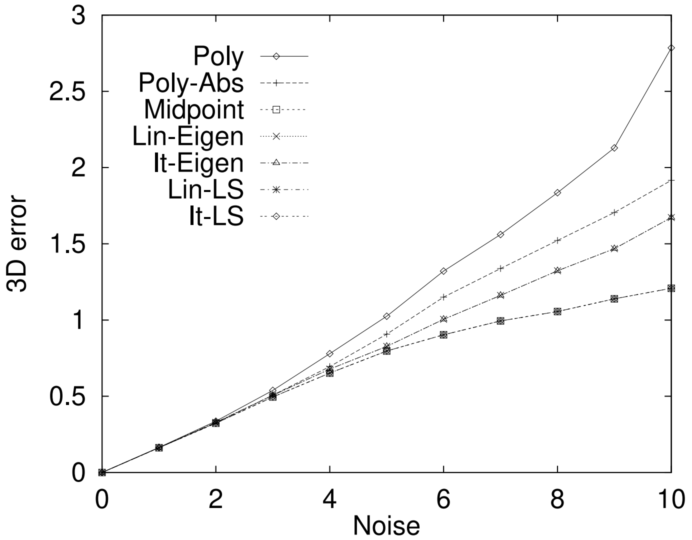


Figure 2: The two simulation configurations.

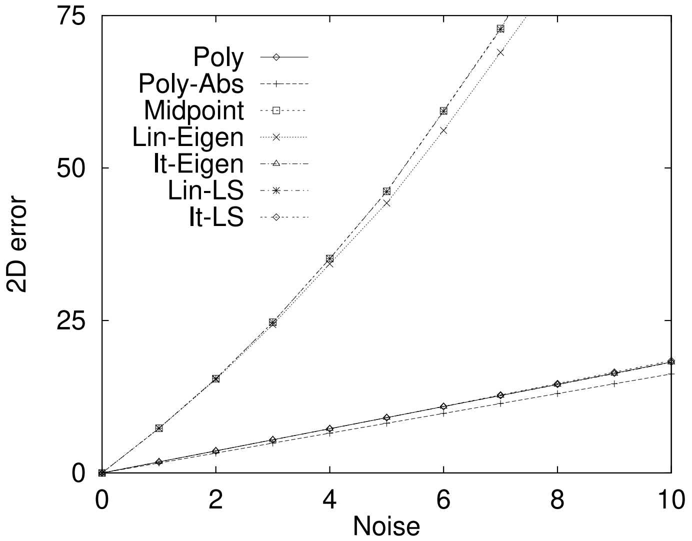
The most unstable situation is Configuration 1, in which the epipoles are in the center of the two images, and points lie close to the epipoles. Since this situation gave the most severe test to the algorithms, we will give the results for that configuration. Results of two cases are presented. In one case the points are at a distance of 0.15 units in front of the first camera (near points case) and in the other case, they are at 0.55 units distance (far points case). The results will be presented in the form of graphs with a commentary for each graph. The measured error is denoted either as 2D error (meaning error of measured compared with the re-projected points), or 3D error, meaning the error compared with the correct values of the points in space. In addition, we talk of Euclidean, affine and projective reconstruction errors. For affine or projective reconstruction, the camera matrices were transformed by a transformation of the given sort, the triangulation was carried out, and finally the reconstructed points were re-transformed into the original frame to compare with the correct values. For Euclidean reconstruction, no transformation was carried out. 50 points were reconstructed in 100 trials. Every data point in the graph expresses the average of the 100 median errors for each 50 points. The horizontal axis of each graph is the noise level (between 0 and 10 pixels RMS in each axial direction), and the vertical axis measures the error, in pixels for 2D error, or in space units for 3D error.



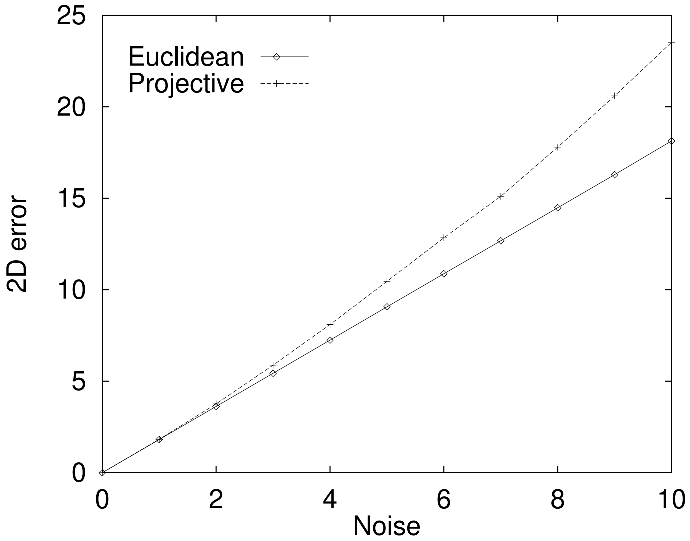
Graph 1 : **3D error for Euclidean reconstruction (near points).** *This graph shows all methods. All perform almost equally. The Polynomial method performs marginally worse than the others. It is designed to minimize 2D error, which explains why optimal in this regard, it is not quite optimal for 3D errors. Euclidean reconstruction is the only instance in which Mid-point performed even marginally well, and the only case in which Polynomial was beaten.*



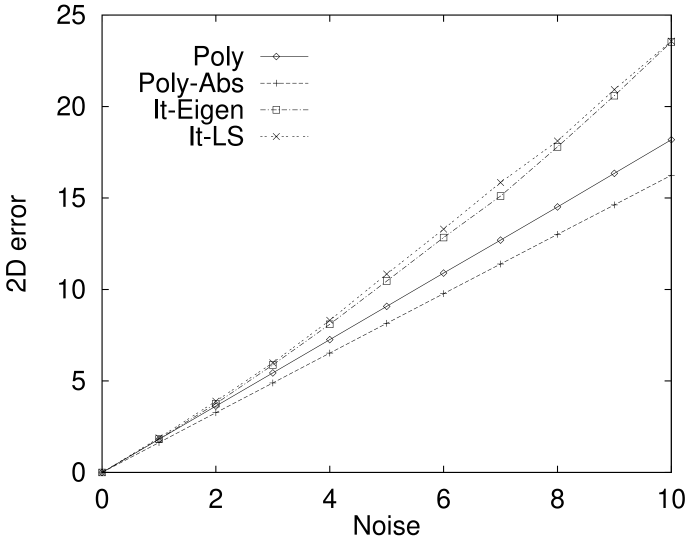
Graph 2 : **3D error for Euclidean reconstruction (far points).** *The configuration is the same as for Graph 1, except that the points are further from both cameras. The curves from the bottom are Linear-LS, Iterative-LS and Mid-point, which are almost indistinguishable. The curves for Linear-Eigen and Iterative-Eigen are also identical. Then follow Poly-Abs and Polynomial.*



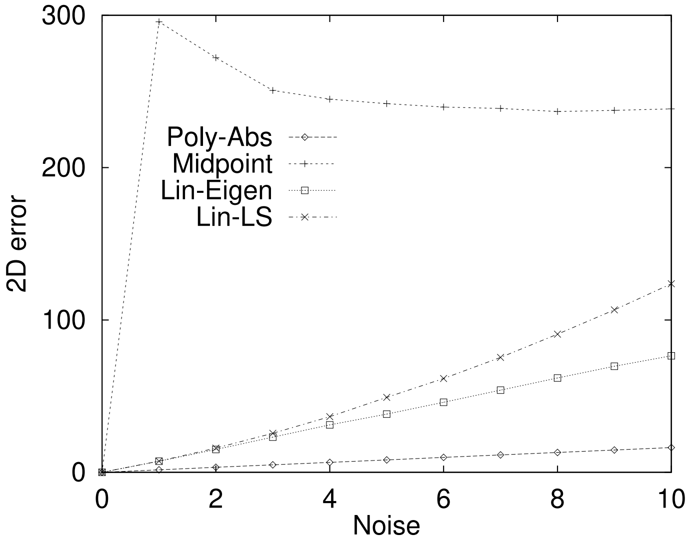
Graph 3 : **2D error for Euclidean reconstruction (near points).** *The configuration is the same as for Graph 1, except that the median 2D error is measured. Of course Poly-Abs performs best (since it is optimized for this task) but Polynomial, Iterative-LS and Iterative-Eigen are almost indistinguishable. The three very bad performers are Linear-Eigen, Linear-LS and Mid-Point. The maximum Y-scale is 75 pixels. Hence, this graph shows that 2D error and 3D error are not well correlated, since despite large 2D errors, these methods perform well in terms of 3D error.*



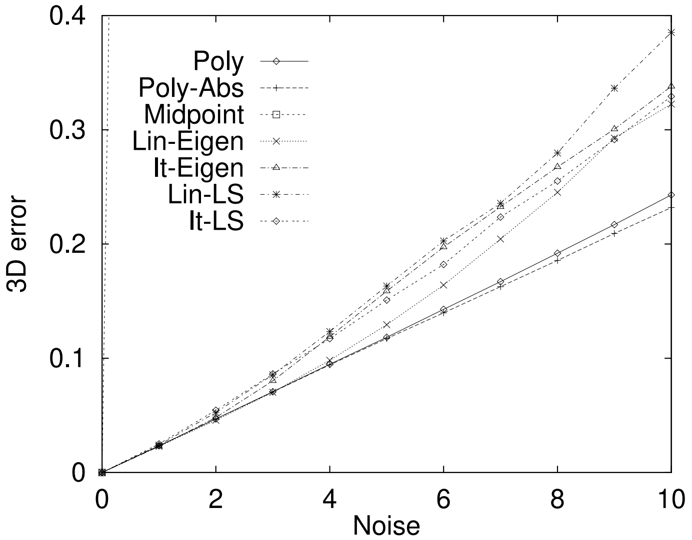
Graph 4 : **Comparison of Euclidean (lower curve) and projective 2D errors.** *The method shown is Iterative-Eigen. The graph shows that this method is almost projective-invariant (that is the two curves are almost the same). This would be an excellent method, except for its failure to converge in very unstable situations (about 1% of trials with noise above 2 pixels). The non-converging cases are ignored in this graph. In cases where the points are not near the epipoles non-convergence is not a problem. The method Iterative-LS (not shown) performs similarly, but just slightly worse, whereas Polynomial is exactly projective-invariant (the two curves are superimposed).*



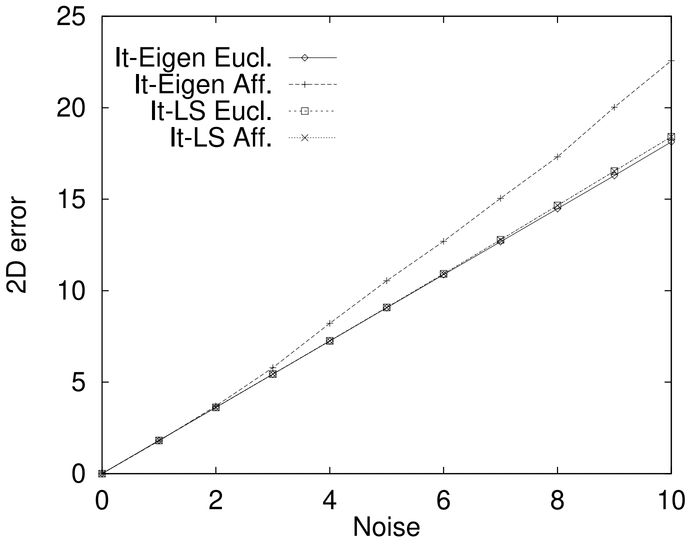
Graph 5 : **2D error for projective reconstruction (near points).** *This is the case for which all methods performed well in the Euclidean case. This graph shows the results for methods (from the bottom) Poly-Abs, Polynomial, Iterative-Eigen, and Iterative-LS. This graph shows that Polynomial, or Poly-Abs is the best method for projective reconstruction, whereas Iterative-Eigen and Iterative-LS (except for occasional non-convergence) perform almost as well.*



Graph 6 : **2D error for projective reconstruction (near points), continued.** *This shows the bad performers for the same configuration as for Graph 5. The graphs shown are (from the bottom), Poly-Abs (as reference), Linear-Eigen, Linear-LS and Mid-point. This shows how serious a problem non-invariance under transforms can be.*



Graph 7 : **3D error for projective reconstruction (near points).** *This is the same as Graphs 5 and 6 except that we show the 3D error. Poly-Abs performs marginally better than Polynomial. Then follow Linear-Eigen, Iterative-LS, Iterative-Eigen and Linear-LS. The graph for the Mid-point method goes off scale already for a noise of 1 pixel.*



Graph 8 : **Affine Invariance.** *The three curves shown are from the bottom Iterative-Eigen (Euclidean) Iterative-LS (Euclidean and affine superimposed) and Iterative-Eigen (affine). Thus, as predicted by theory, the Iterative-LS method is precisely affine-invariant, but Iterative-Eigen is not (but almost). Once more we remark that except for occasional non-convergence, these would be good methods.*

7 Evaluation with Real Images

The algorithms were also carried out with the pair of real images shown in Figures 3. These images were the images used for one set of experiments in [2].

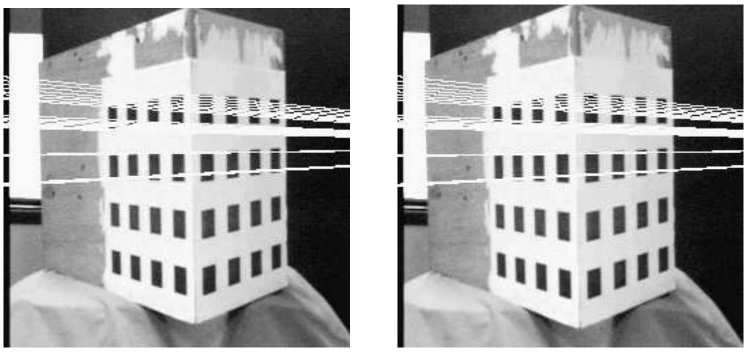
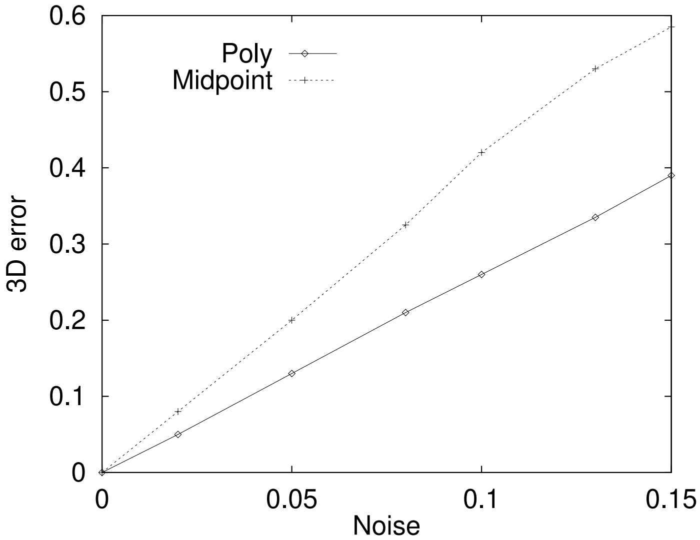


Figure 3: Pair of images used for reconstruction experiments, showing matching epipolar lines.

The goal of these experiments was to determine how the triangulation method affects the accuracy of reconstruction. Since it makes sense to measure the accuracy of reconstruction in a Euclidean frame where distance has a meaning, a close approximation to a correct Euclidean model for the object was estimated by eye and refined using the measured image locations of the corners of the dark squares. The Euclidean model so obtained was used as ground truth.

We desired to measure how the accuracy of the reconstruction varies with noise. For this reason, the measured pixel locations were corrected to correspond exactly to the Euclidean model. This involved correcting each point coordinate by an average of 0.02 pixels. The correction was so small, because of the very great accuracy of the provided matched points. At this stage we had a model and a set of matched points corresponding exactly to the model. Next, a projective reconstruction of the points was computed by the method of [8, 5], and a projective transform H was computed that brought the projective reconstruction into agreement with the Euclidean model. Next, controlled zero-mean Gaussian noise was introduced into the point coordinates, triangulation was carried out in the projective frame, the transformation H was applied, and the error was measured in the Euclidean frame. Graph 9 shows the results of this experiment for two triangulation methods. It clearly shows that the optimal method gives superior reconstruction results. Note that for these experiments, the projective frame was computed only once, with noiseless data, but triangulation was carried out for data with added noise. This was done to separate the effect of noise on the computation of the projective frame from the effect of noise in the triangulation process. The graph shows the average reconstruction error over all points in 10 separate runs at each chosen noise level.



Graph 9 : **Reconstruction Error.** *This graph shows the reconstruction error for the Mid-point (above) and Polynomial methods. On the horizontal axis is the noise, on the vertical axis the reconstruction error. The units for reconstruction error are relative to a unit distance equal to the side of one of the dark squares in the image. The methods Linear-LS, Linear-Eigen, Iterative-LS and Iterative-Eigen gave results close to the Polynomial method. Even for the best method the error is large for higher noise levels, because there is little movement between the images. However, for the actual coordinate error in the original matched points (about 0.02 pixels), the error is small.*

In this pair of images, the two epipoles are distant from the image. For cases where the epipoles are close to the images, the results on synthetic images show that the advantage of the Polynomial methods will be more pronounced.

8 Timing

The following table shows approximate relative speeds for the different algorithms.

Poly	14
Linear-Eigen	3
Iterative-Eigen	5
Mid-point	2
Poly-Abs	30
Linear-LS	2
Iterative-LS	3

Since these are relative measurements only no units appear, but all these algorithms will process several thousands of points per second. In most applications, speed of computation will not be an issue, since it will be small compared with other parts of the computation, such as point matching, or camera model computation.

9 Discussion of Results

All the methods performed relatively for Euclidean reconstruction, as measured in terms of 3D error. In the case of 2D error, only the methods Polynomial, Poly-Abs, Iterative-LS and Iterative-Eigen perform acceptably, and the last two have the disadvantage of occasional non-convergence. The **Poly-Abs** method seems to give slightly better 3D error performance than **Polynomial** but both of these seem to be excellent methods, not susceptible to serious failure and giving the best overall 3D and 2D error performance. The only distinct disadvantage is that they are not especially easily generalizable to more than two images, in contrast to the other proposed methods. They are a bit slower than the other methods, but by a factor of 2 or 3 only, which is probably not significant.

The **Iterative-LS** method is a good method, apart from the problem of occasional non-convergence. Its advantage is that it is about 3 times as fast as the polynomial method and is nearly projective-invariant. In general **Iterative-LS** seems to perform better than **Iterative-Eigen**, but not very significantly. The big problem, however, is non-convergence. This occurs frequently enough in unstable situations to be a definite problem. If this method is used, there must be a back-up method, such as the polynomial method to use in case of non-convergence.

We summarize the conclusions for the various methods.

Poly This is the method of choice when there are only two images and time is not an issue. It is clearly superior to all other methods, except perhaps **Poly-Abs**. In fact, it is optimum under the assumption of a Gaussian noise model. It is affine and projective-invariant.

Poly-Abs This is guaranteed to find the global minimum of sum of magnitude of image error. This may be a better model for image noise, placing less emphasis on larger errors. It seems to give slightly better 3D error results. Otherwise it does not behave much differently from **Poly** and it is affine and projective-invariant.

Mid-point This is not a method that one could recommend in any circumstances. Even for Euclidean reconstruction it is no better than other linear methods, such as Linear-LS, which beats it in most other respects. It is neither affine nor projective-invariant.

Linear-Eigen The main advantage is speed and simplicity. It is neither affine nor projective-invariant.

Linear-LS This has the advantage of being affine-invariant, but should not be used for projective reconstruction.

Iterative-Eigen This method gives very good results, markedly better than **Linear-Eigen**, but not quite as good as **Poly**. It may easily be generalized to several images, and is almost projective-invariant. The big disadvantage is occasional non-convergence, which occurs often enough to be a problem. It must be used with a back-up method in case of non-convergence.

Iterative-LS This method is similar in performance and properties to **Linear-Eigen**, but should not be used for projective reconstruction, since it does not handle points at infinity well. On the other hand it is affine-invariant.

In summary, the **Polynomial** or **Poly-Abs** method is the method of choice for almost all applications. The **Poly-Abs** method seems to give slightly better 3D reconstruction results. Both these methods are stable, provably optimal, and relatively easy to code. For Euclidean reconstruction, the linear methods are a possible alternative choice, as long as 2D error is not important. However, for affine or projective reconstruction situations, they may be orders of magnitude inferior.

Acknowledgement

Thanks to Paul Beardsley and Andrew Zisserman for making the calibration images and data available to us.

References

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

[2] P.A. Beardsley, A. Zisserman, and D.W. Murray, Navigation using affine structure from motion, in *Computer Vision - ECCV '94*, LNCS-Series **801**, pp. 85–96, Springer-Verlag, 1994.

[3] P.A. Beardsley, A. Zisserman, and D.W. Murray, Sequential updating of projective and affine structure from motion. Technical Report, Oxford University, 1994.

[4] O.D. Faugeras, What can be seen in three dimensions with an uncalibrated stereo rig ?, in *Computer Vision - ECCV '92*, LNCS-Series **588**, pp. 563–578, Springer-Verlag, 1992.

[5] R.I. Hartley, Euclidean reconstruction from uncalibrated views, in *Applications of Invariance in Computer Vision*, LNCS-Series **825**, pp. 237–256, Springer-Verlag, 1994.

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

[7] R.I. Hartley, Lines and points in three views - an integrated approach, in *Proc. of the ARPA Image Understanding Workshop 1994, Monterey, CA*, 1994, pp. 1009–1016.

[8] 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.

[9] R.I. Hartley and P. Sturm, Triangulation, in *Proc. of the ARPA Image Understanding Workshop 1994, Monterey, CA*, 1994, pp. 957–966.

[10] J.J. Koenderink and A.J. van Doorn, Affine structure from motion, *Journal of the Optical Society of America, A* **8** (2), 1992, 377–385.

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

[12] R. Mohr, F. Veillon, and L. Quan, Relative 3D reconstruction using multiple uncalibrated images, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 1993, pp. 543–548.

[13] J. Ponce, D.H. Marimont, and T.A. Cass, Analytical methods for uncalibrated stereo and motion reconstruction, in *Computer Vision - ECCV '94*, LNCS-Series **800**, pp. 463–470, Springer-Verlag, 1994.

[14] W.H. Press, B.P. Flannery, S.A. Teukolsky, and W.T. Vetterling, *Numerical Recipes in C: The Art of Scientific Computing*, Cambridge University Press, 1988.

[15] L.S. Shapiro, A. Zisserman, and M. Brady, Motion from point matches using affine epipolar geometry, in *Computer Vision - ECCV '94*, LNCS-Series **801**, pp. 73–84, Springer-Verlag, 1994.

[16] A. Shashua, Projective depth: A geometric invariant for 3D reconstruction from two perspective/orthographic views and for visual recognition, in *Proc. International Conference on Computer Vision*, pp. 583–590, 1993.

[17] C.C. Slama (Ed.), *Manual of Photogrammetry*, 4th Edition. American Society of Photogrammetry and Remote Sensing, 1980.