Relative Motion and Pose From Arbitrary Plane Curves

C.A. Rothwell, A. Zisserman, C.I. Marinos,
D.A. Forsyth and J.L. Mundy
Robotics Research Group
Department of Engineering Science
University of Oxford
Oxford OX1 3PJ

We show how to recover the position and orientation of a pair of known coplanar conics from a single perspective image. We describe and compare a number of methods for determining this relationship. One uses a simple four point back projection model, and the other utilizing transformation invariants. These results can be extended in a number of ways. Firstly, they can be applied to arbitrary plane curves using an invariant fitting technique. Secondly, the recovery methods are applicable to higher order algebraic curves.

We show examples of pose determination for both synthetic data and real images of conic and non-conic objects. For example we recognise and determine pose for a gasket containing non-conic curves. We assess the methods for stability and for accuracy against an accepted standard - Tsai's calibration method [16]. We conclude that using such methods enables accurate pose determination for arbitrary plane curves .

1 Introduction.

Two central issues in Model based vision are recognising objects in a scene and determining their pose. An object's pose is its position and orientation in a known coordinate frame (often the camera frame). Pose is important in a number of situations. For example, a robot, interacting with a scene, needs to know where objects are in order to manipulate them or avoid colliding with them. An Autonomous Guided Vehicle (AGV) needs pose both for obstacle avoidance and to use objects as "beacons" to determine its position and motion.

Existing polyhedral model based vision systems [15] conflate the two distinct problems of library indexing (recognition) and estimating transformation parameters (pose). They use local feature groups to estimate transformation parameters. An instance of an object is then confirmed by checking that other model features are correctly mapped to image features. Thus clustering on pose is used for recognition.

Recent work [4, 5] has shown that it is possible to compute shape descriptors of arbitrary plane objects that are unaffected by position, orientation and intrinsic parameters of the camera. Using these *invariant shape descriptors*, models can be found in a library without having to determine pose. The essential idea is that plane data (e.g. sets of points) can be *represented* by algebraic curves in a frame invariant manner [4]. This means that given an observation of a data set in a transformed frame, the representation computed for this set is exactly the original representation transformed according to the change of frame. This frame independence property means that we can associate an algebraic curve with the data set in a affine invariant manner. The algebraic curve becomes a affine invariant representation. Sets of algebraic curves admit *scalar projective invariants* and it is these that form the invariant shape descriptors. Our interest in pose arose when we realised that, by construction, the representing algebraic curves undergo the same affine distortion as the original data. Finding position and orientation of arbitrary plane curves is reduced to the question of determining the pose of known algebraic curves. In the sequel we concentrate on representation by pairs of conics, though the methods we develop are applicable to representation by any algebraic curves (e.g. a line and a conic, a cubic etc).

Previous methods for pose determination have concentrated on sets of points and lines [6, 3, 8, 15, 16]. However, apart from consideration of circles [2, 9], very little attention has been paid to pose recovery from plane curves. The novel aspects of this paper lie firstly in the method of using equiform invariants to recover pose for algebraic curves; and secondly in recovering pose for arbitrary plane curves using invariant algebraic curve representation[1].

To summarise: plane object curves are represented by a pair of conics. The invariants of the conic pair can be used to recognise the object from a single perspective view. Once an object has been positively identified, the

[1] The invariant fitting theorem [4] is formulated for *point sets* rather than continuous curves. However, in practice there has been no problem in applying it to curves.

extra constraints offered by its known identity can be exploited to determine transformation parameters. The problem can be stated as follows:

> Given a pair of coplanar model conics and their perspective image, determine the position and orientation of the model with respect to the camera coordinate frame.

The rest of this paper examines a number of methods which can be used to solve this problem. The paper is organised as follows: section 2 describes conic representation and transformation; section 3 deals with parameterising perspective transformation; section 4 develops and compares various methods of pose recovery; the application of these methods to real and synthetics data is covered in Section 5; finally, section 6 discusses the sensitivity of the methods to errors in conic fitting and camera calibration.

Earlier work has pointed out the instability of conic fitting when data only covers a small part of the conic [13]. In our work we only use conic representations for closed curves, and other representations in other circumstances.

2 Conic Representation.

A conic is the set of points (x, y) that satisfy:

$$ax^2 + bxy + cy^2 + dx + ey + f = 0 \qquad (1)$$

A more convenient representation of a conic uses homogeneous coordinates. Planar points are represented in homogeneous coordinates by 3-vectors. For example, the homogeneous point (X, Y, Z) corresponds to the point (x, y), where $x = \frac{X}{Z}$ and $y = \frac{Y}{Z}$. $(\lambda X, \lambda Y, \lambda Z)$ represents the same point. All vectors $\mathbf{x}$ and $\mathbf{X}$ will represent homogeneous 3-vectors in the rest of this paper.

Using homogeneous coordinates a conic C can be represented as a quadratic form:

$$\mathbf{x}^T C \mathbf{x} = 0 \qquad (2)$$

where:

$$C = \begin{pmatrix} a & \frac{b}{2} & \frac{d}{2} \\ \frac{b}{2} & c & \frac{e}{2} \\ \frac{d}{2} & \frac{e}{2} & f \end{pmatrix} \qquad (3)$$

From equation 2 it is clear that kC is the same curve as C $(k \in \mathcal{C} - \{0\})$[2]. When representing curves by conics we need to evaluate functions of C. To ensure that the function values for different conics are consistent we must specify the scaling[3] k. The scaling we use is $|C| = 1$ (where $|\ |$ is det). This means we cannot represent straight lines, where $|C| = 0$. However, each scale has some disadvantage [1], eg:

1. $A = 1$ prevents use of some parabolas, eg. $y^2 - 1 = 0$.
2. $B = 1$ makes it impossible to fit circles.
3. $F = 1$ makes it impossible to fit curves passing through the origin.

From now on, italic letters denote matrices, bold letters denote vectors, large letters denote world objects and small letters denote image objects (eg. C is a world conic, $\mathbf{X}$ a world point, and c and $\mathbf{x}$ their images). Transformations are represented by 3×3 matrices (see appendix A). A projective transformation matrix T mapping from the world to image planes is given by:

$$\mathbf{x} = T.\mathbf{X} \quad \text{where} \quad |T| \neq 0 \qquad (4)$$

If we consider two coplanar world conics C_1 and C_2, with their respective image conics c_1 and c_2, they map under T by:

$$c_1 = \kappa \ . \ (T^T)^{-1} C_1 \ T^{-1} \qquad (5)$$

$$c_2 = \kappa \ . \ (T^T)^{-1} C_2 \ T^{-1} \qquad (6)$$

In effect T^{-1} strips the effect of the coordinate transformation. The scaling $\kappa = |T|^{\frac{2}{3}}$ ensures the normalisation $|c_i| = 1$ when $|C_i| = 1$. Our aim in solving for pose is to determine a good approximation to T.

3 Perspective Transformation.

The matrix T represents a full projective mapping. It is a 3×3 matrix, therefore it has nine parameters, but as the scaling of the matrix does not change the mapping it has eight degrees of freedom. In machine vision we are normally concerned with plane to plane perspectivities. A perspectivity has six degrees of freedom[4] - 3 translation and 3 rotation. Thus, there are two extra constraints that a perspective T matrix must obey. These two extra

[2] $\mathcal{C}$ is the set of complex numbers

[3] For example the frame invariant fitting technique in [4] uses algebraic distance as an error metric for the fit. The algebraic distance for a conic C is defined as $q(C) = \sum(ax_i^2 + bx_iy_i + cy_i^2 + dx_i + ey_i + f)^2$ for all points (x_i, y_i) in the data set. Hence $q(2C) = 4q(C)$. To make the algebraic distance meaningful for all curves C we make $|C| = 1$.

[4] Assuming a fixed camera with known calibration.

constraints are that the first two columns form part of a scaled rotation matrix [6, 16] (see appendix A). This means that:

$$||\mathbf{u}|| = ||\mathbf{v}|| \quad \text{and} \quad \mathbf{u} \,.\, \mathbf{v} = 0 \tag{7}$$

where $\{\mathbf{u}, \mathbf{v}, \mathbf{t}\}$ are the columns of T. We use the equinorm condition on the first two columns rather than specifying unit norm (for a rotation matrix) to allow for improper scalings of T.

The transformation between camera and world coordinate systems can be accomplished by a Euclidean 3D rotation (represented by a matrix) and a translation (represented by a vector). This more usual representation is given in appendix A. Here, we develop an alternative decomposition of the T matrix which makes explicit the dependence on plane orientation (a similar decomposition was used in [15]). The world plane (see figure 1), is expressed in camera coordinates as[5]:

$$z = px + qy + r \tag{8}$$

We define a rectangular world coordinate system on the world plane by X and Y axes. The position and orientation of these axes can be given relative to the camera coordinate frame using an induced rectangular coordinate frame also lying on the world plane. The induced frame has its $\tilde{X}$ and $\tilde{Y}$ axes lying in the world plane, centred at $(0, 0, r)$, and has the $\tilde{X}$ axis lying in the same plane as the camera x axis. Hence the induced frame position and orientation depends only on p, q and r. A rotation and translation within the plane maps the induced coordinate frame onto the world frame. The rotation is represented by θ and the translations by t_x and t_y (see figure 1). Hence the position and orientation of the world axes depends on θ, t_x, t_y and on p, q and r. This allows us to split the perspective mapping into two transformations, a planar Euclidean action from the world to induced frame, and a perspectivity from induced to camera frame. Alternatively we can separate the projection into three transformations:

- Planar Euclidean action from the world to induced frame. This depends on θ, t_x and t_y.
- Perspectivity from the plane $z = px + qy + r$ to the plane $z = px + qy + 1$. Both planes have induced frames which are parallel. This transformation depends on r, and is an isotropic scaling.
- Perspectivity from the plane $z = px + qy + 1$ to the image plane $z = 1$, depending on p and q. We assume that the image curves are normalised to lie on the plane $z = 1$ rather than the actual image plane $z = f$.

[5] This means that we cannot represent the family of planes $px + qy + r = 0$. These are the planes perpendicular to the image plane.

Choosing $z = 1$ is purely a matter of convenience. So is using the projection from the plane $z = px + qy + r$ to the plane $z = px + qy + 1$ (it amounts to moving "r" from one matrix to another - see below). We group the first two transformations into a single equiform transformation. An equiform mapping is a combination of a Euclidean transformation and an isotropic scaling. It preserves angles and ratio of lengths. The Euclidean mapping is the rotation and translation represented by θ, t_x and t_y. The scaling is represented by r. This means that the perspective mapping is represented by the six parameters p, q, r, θ, t_x and t_y.

The perspective mapping then, is a product of two transformation matrices, an equiform matrix and a central projection matrix. The equiform matrix is $H(r, \theta, t_x, t_y)$, and the perspective matrix $M(p, q)$. Derivation of the M matrix is shown in appendix F. Hence T may be written $T = M(p, q).H(r, \theta, t_x, t_y)$, where:

$$M = \begin{bmatrix} \frac{1}{\sqrt{1+p^2}} & \frac{-pq}{\sqrt{1+p^2+q^2}\sqrt{1+p^2}} & 0 \\ 0 & \frac{\sqrt{1+p^2}}{\sqrt{1+p^2+q^2}} & 0 \\ \frac{-p}{\sqrt{1+p^2}} & \frac{-q}{\sqrt{1+p^2+q^2}\sqrt{1+p^2}} & 1 \end{bmatrix} \tag{9}$$

$$H = \begin{bmatrix} c & -s & st_y - ct_x \\ s & c & -ct_y - st_x \\ 0 & 0 & r \end{bmatrix} \tag{10}$$

where $c = cos\theta$ and $s = sin\theta$

$$T = k \begin{bmatrix} (mc - pqs) & -(ms + pqc) & ([st_y - ct_x]m + ct_y pq + st_x pq) \\ sn^2 & cn^2 & -(ct_y + st_x)n^2 \\ -(mpc + sq) & (mps - qc) & ([ct_x - st_y]mp + [ct_y + st_x]q + mnr) \end{bmatrix} \tag{11}$$

where $m = \sqrt{1 + p^2 + q^2}$ and $n = \sqrt{1 + p^2}$

The remainder of this paper will show how the T can be recovered directly from a pair of world and image conics. We shall see that decomposing to the M and H matrices produces a more stable solution.

4 Pose Recovery.

Solving the ten independent equations given by equations 5 and 6 for the eight unconstrained parameters of

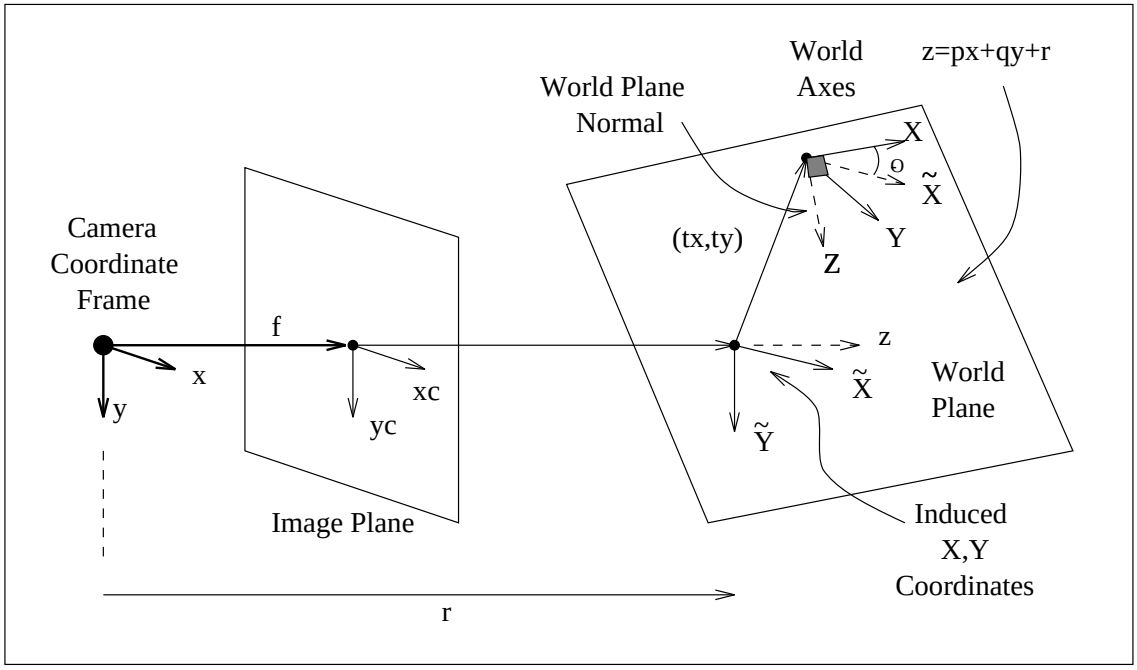

Figure 1: The camera and world plane geometry. The camera and induced x axes lie in the same plane. The induced $\tilde{X}\tilde{Y}$ plane and the world XY plane both lie in the plane $z = px + qy + r$. $\tilde{X}\tilde{Y}$ centred at $(0, 0, r)$. Hence a rotation θ and translation (t_x, t_y) in the world plane maps the induced coordinate axes onto the world axes.

T by least squares is impractical. An alternative scheme would be to solve eight of the equations (a system of eight polynomial equations in eight variables), yielding 256 solutions in the generic case (Bezout's Theorem [14],p.199). Most of these solutions could be eliminated by testing them against the other two equations. This again is impractical. In this paper we present a number of more subtle solutions. These come in two forms:

1. Solving for the T matrix directly and then decomposing the matrix to give the six pose parameters.

2. Decomposing the projective transformation into separate transformations each dependent on different pose parameters.

Each method has its own advantage. Solving for the T matrix directly involves matching world plane points to image points to give a system of solvable linear equations. The method is described in section 4.1.1. Decomposition of the T matrix relies on the use of the transformation invariants of conics. This is given in section 4.2.

Once we have derived a good approximation to the perspective transformation we can then use it as the starting point for an iterative scheme to progressively improve the estimate. The iterative scheme is given in section 4.3

4.1 Direct Solution.

4.1.1 Four Point Solution.

A projective mapping can be defined by four world points and their images [7]. The mapping we use is given in equation 4. Taking four world and image points sets up a system of eight linear simultaneous equations, in the elements of the T matrix (see appendix B), which are solved by Gaussian elimination. As we work with pairs of conics we need four curve points that are both preserved under projection and stable to numerical errors and noise. Such points are curve intersection points and tangencies.

4.1.2 Conic Intersections.

In general a pair of conics intersect at four points, though the intersections are not always real. Figure 2 shows how the number of real intersection points varies with position for a pair of ellipses.

Here we work through the complex case - but there is no difference in the methods for real or complex intersections, apart from combinatorics of matching. There are 24 ways (i.e. 4!) to match 4 image points to 4 ob-

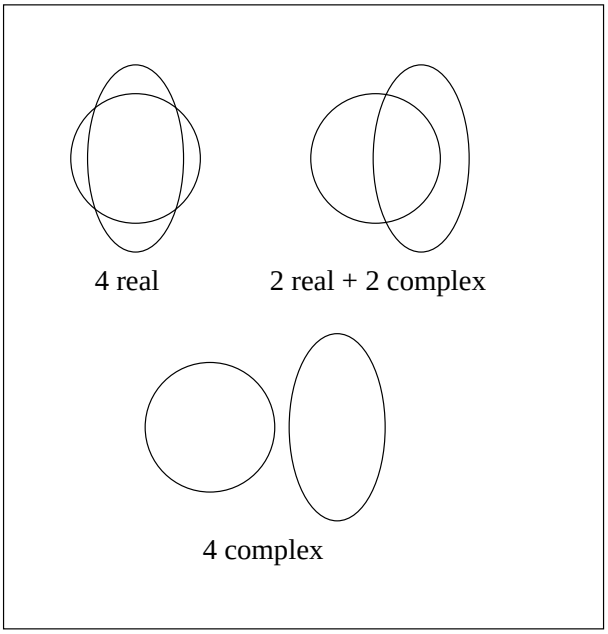

Figure 2: For a generic pair of conics there are always four points of intersection. These may all be real, or two real with a complex conjugate pair, or all complex as two complex conjugate pairs.

ject points. However, we know the transformation is a perspectivity so the *order* of points around the conic is preserved (this is not true of a general projectivity). This means that the combinatorics are reduced from 24 to 4 (the choice of which point to start from on the chosen conic). We do not know of any way to order complex points. However, the combinatorics can be reduced to 8 by the following argument: The intersections are found by solving a real quartic; hence any complex intersections will appear in complex conjugate pairs. All complex world points map to complex image points by the same T matrix that transforms real world points. Therefore if a real projection has occurred the T matrix found by mapping complex points to complex points is real. Furthermore, all complex conjugate pairs project to conjugate pairs[6]. That is if

$$(X_1, X_2, X_3) \rightarrow (x_1, x_2, x_3)$$

then

$$(\bar{X_1}, \bar{X_2}, \bar{X_3}) \rightarrow (\bar{x_1}, \bar{x_2}, \bar{x_3})$$

where $\bar{x_i}$ is the complex conjugate of x_i. Thus, the only possibilities for world points $[A, \bar{A}, B, \bar{B}]$ mapping to image points $[a, \bar{a}, b, \bar{b}]$, A and $\bar{A}$ are:

$$\begin{bmatrix} A \\ \bar{A} \\ B \\ \bar{B} \end{bmatrix} \rightarrow \left\{ \begin{bmatrix} a \\ \bar{a} \\ b \\ \bar{b} \end{bmatrix}, \begin{bmatrix} a \\ \bar{a} \\ \bar{b} \\ b \end{bmatrix}, \begin{bmatrix} \bar{a} \\ a \\ b \\ \bar{b} \end{bmatrix}, \begin{bmatrix} \bar{a} \\ a \\ \bar{b} \\ b \end{bmatrix}, \begin{bmatrix} b \\ \bar{b} \\ a \\ \bar{a} \end{bmatrix}, \begin{bmatrix} b \\ \bar{b} \\ \bar{a} \\ a \end{bmatrix}, \begin{bmatrix} \bar{b} \\ b \\ a \\ \bar{a} \end{bmatrix}, \begin{bmatrix} \bar{b} \\ b \\ \bar{a} \\ a \end{bmatrix} \right\}$$

Any correspondence outside this set will produce a complex projective transformation matrix.

[6] If the T matrix is real complex conjugate points project to complex conjugate points by linearity.

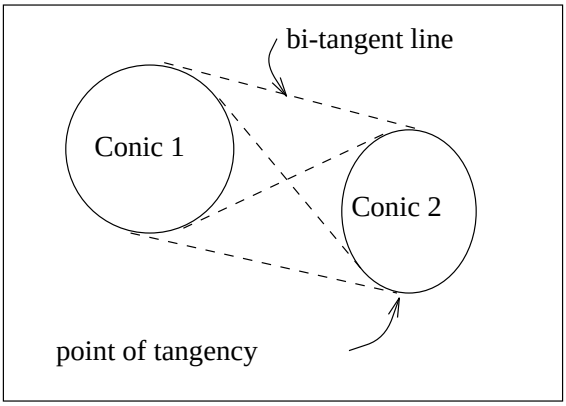

Figure 3: With a pair of conics there are four bi-tangent lines. These four lines are tangent to the conics at eight points. These points are preserved under projection.

4.1.3 Conic Tangencies.

Tangency is preserved under projection. For a pair of conics we use the bi-tangent lines as the tangents (see figure 3). Rather than use the points of tangencies of these lines to the conics as the points (which would be sensitive to noise) we use the actual bi-tangents (exploiting the duality of points and lines under projection, though such lines project by the transpose of T^{-1} rather than T). The bi-tangent lines are found by intersecting the inverse matrices of the conics. This amounts to solving a quartic (see appendix D).

Many real objects have co-planar features that are represented by one conic enclosed by another (eg. a pair of concentric circles). For such objects real bi-tangent lines do not exist. However, complex lines do exist and they can be used to find complex points of tangency. Again the points are in complex conjugate pairs and the treatment is the same as for intersection points. As in the conic intersection case, for real tangencies the correspondence is simpler as the combinatorics are reduced to 4. In the complex case there are 8 possibilities.

Using tangencies together with intersections yields eight points in the image and world planes. Although not all the points are functionally independent it would be possible to use all the points as part of a least squares technique to reduce noise (for example the Moore-Penrose pseudo inverse). The number of possible correspondences do however prevent an efficient implementation.

4.1.4 Reducing Ambiguity and Extracting Pose.

From each set of correspondences we calculate a T matrix. Only one matrix from the set of eight (or 4 in the real case) will correspond to the real perspectivity. It must satisfy:

1. The matrix must be real.
2. The matrix must correspond to a perspectivity.
3. The world plane must be in front of the camera.

We recognise a perspective transformation matrix as it must obey conditions given in equation (7) i.e. if the first two columns of the T matrix are not orthogonal, and do not have equal norms, the matrix can be removed from the solution set as it is not a perspectivity. Due to noise it is often difficult to distinguish a non-perspective matrix from a genuine perspectivity. In many cases the noise is sufficient to make the T matrix appear non-perspective even when it is. In these cases we cannot extract the pose parameters accurately and so the matrix is discarded.

If the estimated distance between the camera focal plane and the world plane is too small (less than a few millimetres), or negative, the solution can be discarded. These cases may be genuine perspectivities, but are obviously not the correct solutions.

The perspective transformation matrix can then be decomposed to give the perspective parameters p, q, r, θ, t_x and t_y by matching with the form given in equation 11, where we have used the expansion for T given by equations 9 and 10. The extraction of the pose parameters from T is outlined in appendix C. The results for finding pose by four point back projection appear in section 5.

4.2 Decomposing the T matrix into Two Distinct Transformations.

This section describes a method of determining pose by splitting the perspective transformation into two parts. It relies on the use of equiform invariants:

1. We define a plane to plane transformation which ensures a geometric similarity between the two pairs of conics. To do this we backproject the image conics onto a plane at $z = px + qy + 1$ with induced axes centred at $(0, 0, 1)$ and orientated with[7] $\tilde{X}_1$ and x axes co-planar (see figure 1). We know the back-projected conics are similar to the world conics as they have the same equiform invariants (see below). The plane $z = px + qy + 1$ is parallel to the world plane $z = px + qy + r$.
2. We determine r, a scaling parameter, by minimising the difference between the areas of the conics found by (1) when projected onto the plane $z = px+qy+r$ and the model conics. Then we perform an in plane rotation and translation to match the induced and model conics. The scaling and in plane transformation amount to an equiform transformation. Any equiform mapping applied to the conics will not change the values of their equiform invariants.

[7] $\tilde{X}_1$ denotes the x axis on the plane $z = px + qy + 1$, and $\tilde{X}$ the x axis on the plane $z = px + qy + r$. The $\tilde{X}_1$ and $\tilde{X}$ axes are also coplanar.

4.2.1 Equiform Invariants.

To find p and q we make use of equiform invariants. By definition $I(c)$ is an invariant for a conic c under the equiform transformation H if:

$$I(H(c)) \quad = \quad |H|^w \; I(c) \tag{12}$$

Where w is the weight of the invariant. If $w = 0$ we have a scalar invariant. The invariants we use are the trace and determinant of the upper left 2×2 submatrix of the conic matrix. Geometrically the invariants are related to the aspect ratio and area of an ellipse. $|H|$ is dependent only on the scaling of the equiform transformation (appendix E), and so the aspect ratio and area of a conic only change with scaling. The proof of their invariance is in appendix E. If we consider the conic C in equation 3, $trace^{2\times2}[C]$ and $det^{2\times2}[C]$ are:

$$trace^{2\times2} = a + c \tag{13}$$

$$det^{2\times2} = a.c - \frac{b^2}{4} \tag{14}$$

$trace^{2\times2}$ has weight $\frac{2}{3}$ and $det^{2\times2}$ weight $\frac{4}{3}$. We can see how the invariants reduce the problem of solving pose by considering equations 5 and 6 and letting $T = MH$:

$$\begin{aligned} c &= T(C) \\ &= M(H(C)) \\ \Rightarrow |M|^{-\frac{2}{3}} \; M^T c M &= H(C) \end{aligned}$$

and applying the operator $trace^{2\times2}$:

$$\begin{aligned} trace^{2\times2}[H(C)] &= trace^{2\times2}[|M|^{-\frac{2}{3}} \; M^T c M] \\ \Rightarrow |H|^{\frac{2}{3}} trace^{2\times2}[C] &= trace^{2\times2}[|M|^{-\frac{2}{3}} \; M^T c M] \\ & \quad \text{by the invariance property} \\ & \quad \text{of } trace^{2\times2} \end{aligned}$$

$$\Rightarrow trace^{2\times2}[C] = |H|^{-\frac{2}{3}}|M|^{-\frac{2}{3}}\ trace^{2\times2}[M^T c M]$$
$$= \kappa\ trace^{2\times2}[M^T c\ M]$$
$$= \kappa\ \frac{f_t(p,q,c)}{1+p^2+q^2} \quad (15)$$

where f_t is a conic in p and q with coefficients taken from the coefficients of c. Likewise for $det^{2\times2}$:

$$det^{2\times2}[C] = \kappa^2\ \frac{f_d(p,q,c)}{1+p^2+q^2} \quad (16)$$

where f_d is also a conic in p and q. f_t and f_d are given in appendix G. Hence we choose a scalar invariant which is independent of $r, \theta, t_x$and t_y (it is also independent of the choice of induced coordinate system):

$$I(c) = \frac{(trace^{2\times2}[C])^2}{det^{2\times2}[C]} = \frac{f_t(p,q,c)}{(1+p^2+q^2).f_d(p,q,c)} \quad (17)$$

and so:

$$\frac{(trace^{2\times2}[C_1])^2}{det^{2\times2}[C_1]} = \frac{(trace^{2\times2}[M^T.c_1.\ M])^2}{det^{2\times2}[M^T.c_1.\ M]} \quad (18)$$

$$\frac{(trace^{2\times2}[C_2])^2}{det^{2\times2}[C_2]} = \frac{(trace^{2\times2}[M^T.c_2.\ M])^2}{det^{2\times2}[M^T.c_2.\ M]} \quad (19)$$

When multiplied out these two equations are both quartics in p and q. In general there should be sixteen solutions which satisfy the intersections of these quartics, but we find that there only ever exist four distinct real roots (see figure 4). This is a remarkable feature of the polynomials which reduces the ambiguity of the solution to reasonable size. Using these four values for p and q to determine the other pose parameters gives four distinct solutions. Solving for a unique pose is described later.

The quartics can not be solved in closed form. As there is no stable way of finding approximate roots[8] to initiate an iterative scheme we have to solve for all of the roots. We do this is by continuation [10]. Continuation is a robust method of finding all the roots of a system of polynomials.

We know that for a single conic under equiform transformation there is only one independent scalar invariant, but for a pair of conics there are a total of six, by a simple counting argument [5]. Most of these invariants lead to higher order forms than quartics with no easy solution, or to forms which are not stable (see section 6.1). However one exists which also provides a quartic in p and q. This is:

$$I_{12} = \frac{trace[c_1^{2\times2}\ .\ c_2^{2\times2}]}{trace^{2\times2}[c_1].trace^{2\times2}[c_2]} \quad (20)$$

Where the superscript 2×2 denotes the multiplication of the upper 2×2 submatrix. The third quartic is functionally independent of the first two, but it always has roots in the locality of the intersections of the first two quartics and so does not enable us to reduce the number of possible solutions (see figure 4). It could, however, be used to yield a least squares fit to our solution for p and q using all the intersection points found by the three possible pairings of the quartics. In practice though, we find that the intersections of the possible pairings of the three quartics tend to be sufficiently close together that we need only solve the first two quartics.

[8] eg. by finding an approximate solution to the pose parameters p and q

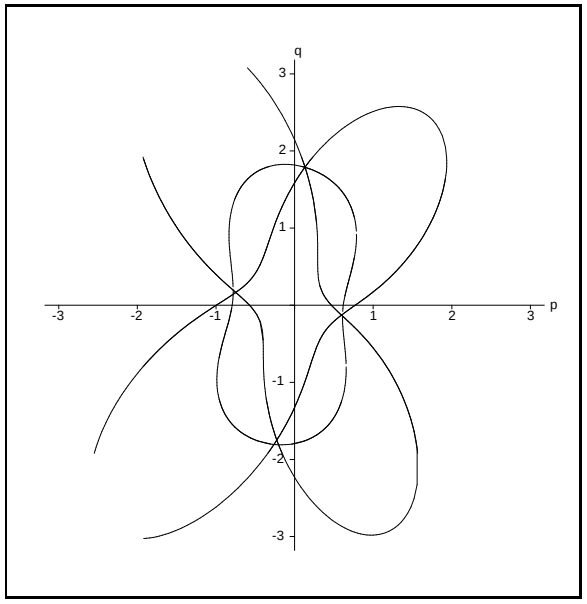

Figure 4: Graph showing the three quartics in p and q. There only ever appear to be four real solutions to the quartics compared to the possible sixteen. Note that the third quartic has roots coinciding with the roots of the first two, and so in general does not contribute any extra constraints to the solution.

4.2.2 Remaining Pose Parameters.

Once p and q are known the other four parameters can be found by equiform transformations on the conics. We find r by a least squares cost based on the areas of the image conics back projected by $M(p,q)$ and the model conics. The areas of the conics are affected only by the r term of H. If the area of conic c under the mapping $M(p,q)$ is A_{pq}, the area of the same conic projected onto the plane $z = px + qy + r$ is $r^2 A_{pq}$. The cost function we use is $\sum_i (A^i_{model} - r^2 A^i_{pq})^2$. Minimising this gives:

$$r^2 = \frac{\sum_i A^i_{model} \cdot A^i_{pq}}{\sum_i (A^i_{pq})^2}, \quad (21)$$

where the sum is over the two conics. The area of a conic is in fact a function of one of the equiform invariants used, $A^i = \pi/(det^{2\times2}[c_i])^{\frac{3}{2}}$.

The in-plane rotation within H can be found from the an-

gle between the line joining conic centroids for the model conic pair and the line joining centroids of the induced conic pair. Similarly the translation can be found from the displacement between the centroids. This method will fail for concentric conics, and will be poorly conditioned as the centres become close. For concentric conics the rotation is determined from the direction of the conic axes, though this is not so robust. In the case of concentric circles the rotation can not be determined.

4.2.3 Eliminating Ambiguous Solutions.

From each of the four quartic solutions to p and q we are able to derive solutions to the other pose parameters r, θ, t_x and t_y. We eliminate three of the solutions by choosing the one that produces the best match of the model conics to the image conics back projected onto the world plane. From equations 5 and 6 we can estimate the equations of the world conics by back projecting the image conics using the four estimate of T. Taking C_1^i and C_2^i to be the back projected conics:

$$\begin{aligned} C_1^i &= (T_i)^T .\ c_1\ .T_i \\ C_2^i &= (T_i)^T .\ c_2\ .T_i \quad i \in \{1,..,4\} \end{aligned} \tag{22}$$

We now compare each C_1^i and C_2^i to the actual models C_1 and C_2 to see which T_i provides the 'best' correspondence. There are a number of available cost functions to test the correspondence, eg. by finding the total Euclidean distance between the elements of the conic matrices. A good cost function gives a good measure of the difference between the model conics and the back projected image conics, but is also easy to calculate. The cost function we use does this. It is a combination of the Euclidean distance between the conic matrices and of the projective invariants I_{ab3} and I_{ab4} defined between two conics A and B by [4]:

$$\begin{aligned} I_{ab3}(A,B) &= Trace[A^{-1}.B] \\ I_{ab4}(A,B) &= Trace[B^{-1}.A] \end{aligned} \tag{23}$$

Obviously if $A = B$, $I_{ab3} = I_{ab4} = Trace[I] = 3$. We use this to define a cost function e_i.

$$\begin{aligned} e_i &= [I_{ab3}(C_1, C_1^i) - 3]^2 + [I_{ab4}(C_1, C_1^i) - 3]^2 \\ &+ [I_{ab3}(C_2, C_2^i) - 3]^2 + [I_{ab4}(C_2, C_2^i) - 3]^2 \end{aligned} \tag{24}$$

and we choose T from T_i by a tradeoff between the minimum Euclidean distance and minimum e_i.

4.3 Iteration on the Initial Estimate.

Once we have found an initial estimate for the pose parameters we iterate to improve the estimate. The scheme used is Newton's method over the six pose parameters with the invariant error e_i as the cost. We find that only one or two iterations are needed to make the cost negligible as the initial estimate is very close to the best solution.

method	p	q	r	θ	t_x	t_y
intersections	0.500	0.800	350.0	30.0^o	20.0	35.0
tangency	0.499	0.800	349.8	30.0^o	20.0	35.0
actual	0.500	0.800	350.0	30.0^o	20.0	35.0

Table 1: This table shows the pose parameters found using the two four point methods on synthetic data. Note the consistency between the estimated and actual pose parameters.

5 Results.

5.1 Synthetic Example.

Here we report on the results obtained by the different methods for the synthetic projection of a conic pair. Table 1 shows the pose results found using the different four point methods for one such synthetic projection. The pose parameters found from the intersection points and from the bi-tangent lines are similar to actual pose parameters.

The four pose solutions and their error costs found by T matrix decomposition method are shown in table 2. From the values of the invariant error and the Euclidean distance we deduce that the first solution gives the correct pose.

The results in the next section show how the different methods perform on real data where there is noise due to image localisation and due to error in the camera model. An indication of stability to error in the camera model for the decomposition method is given in section 6.2. In section 6.1 we show how the decomposition method is affected when noise is added to synthetic data.

5.2 Real Data.

The pose parameters extracted from a real image (that of figure 5) are shown in table 3. The object was placed

sol	p	q	r	θ	t_x	t_y
1	0.500	0.800	350.0	30.0^o	20.0	35.0
2	-0.519	-0.477	317.2	36.8^o	21.7	31.4
3	-0.519	-0.477	172.8	91.8^o	49.7	30.4
4	-1.110	3.568	307.6	75.7^o	40.4	76.0
actual	0.500	0.800	350.0	30.00^o	20.0	35.0

sol	Inv_{err}	Euc dist
1	0.0000	0.0014
2	0.0011	33.55
3	10.51	75.27
4	11.05	28.02

Table 2: This table shows the four solutions found by T matrix decomposition on the same synthetic example as above. We choose solution 1 to be the correct solution as it has the smallest invariant error and Euclidean distance.

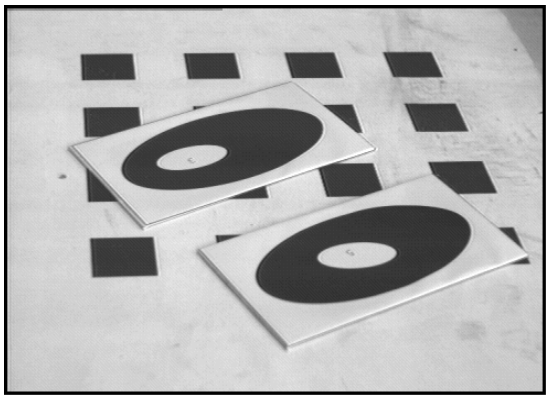

Figure 5: This image was used for the comparison of the different methods that we use. There are two objects in the image, we only report on the results for the nearer object. Under the objects the calibration grid used to find the pose by Tsai's method is visible.

method	σ	τ	r	θ	t_x	t_y
intersections	56.5^o	87.7^o	311.0	31.7^o	14.6	43.1
tangency	57.4^o	85.9^o	300.9	30.5^o	14.6	43.7
T decomposition	54.6^o	85.0^o	353.7	30.0^o	14.3	43.2
Tsai	55.6^o	86.2^o	365.0			

Table 3: This table shows the pose parameters found using all three methods. Angles are in degrees and distances in mm. The intersection and T matrix decomposition results are consistent with each other and with the Tsai result.

on a calibration grid so that the plane parameters could be compared to those found by Tsai's method. Instead of showing p and q we use spherical polar coordinates of the plane normal, namely slant (σ) and tilt (τ) as these give a better indication of the size of the measurement errors ($\sigma = cos^{-1}(\{1+p^2+q^2\}^{-1/2})$ and $\tau = tan^{-1}(q/p)$). We see that the results found using conic intersections, the bi-tangent lines and the matrix decomposition method are very similar, and comparable with the Tsai results. Note that the estimate of r found by the four point methods is smaller than the actual value; this is due to instability in the way that we determine r (see appendix C).

The T matrix recovered through use of the conic intersections has an angle of 88.1^o between the first two columns, and the ratio of the norms of the first two columns is 0.92. For the tangency method the angle between the first two columns of the T matrix is 85.8 and the ratio or norms 0.96. Therefore the T matrices found by both the four point methods represent reasonable approximations to perspectivities.

As we can measure absolute pose for two different objects within a scene, we can work out the differential pose for the two objects. Differential pose is the difference in pose for two objects. The difference we show is the angle between the planes for two models rigidly linked. Figure 6 shows an example of one such object. We are able to measure the actual angle between the planes by hand. The angles between planes for a number of such objects recovered by T matrix decomposition are shown in table 4. The results show how well the T matrix decomposition method works for recovering differential pose.

As an example of relative motion we can recover the angle that an object has been rotated through between consecutive images. The object used was a gasket which we can represent by two coplanar conics using the invariant fitting theorem [4]. The curves used are the outer boundary and the large circular hole (see figure 7). The gasket was rotated through 10^o within its plane each time. Therefore the plane parameters σ, τ and r should remain

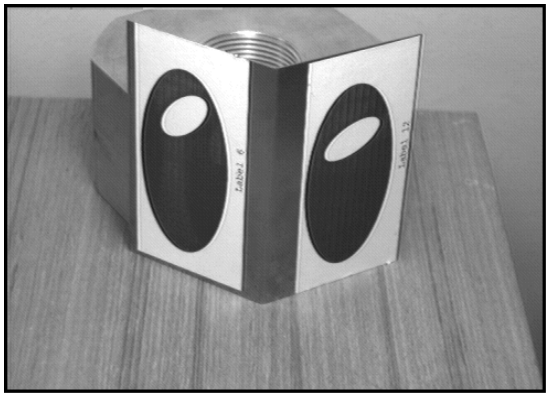

Figure 6: This image shows a typical object used to find differential pose. The object is composed of two planes set at 60^o to each other. Each plane contains a pair of ellipses each of which define a model in the model base.

object	a	b	c	d	e	f	g
image 1	0.6^o	16.2^o	33.2^o	47.1^o	60.3^o	78.8^o	90.6^o
image 2	3.4^o	12.8^o	28.4^o	45.1^o	61.8^o	74.9^o	92.1^o
actual	0.0^o	15.0^o	30.0^o	45.0^o	60.0^o	75.0^o	90.0^o

Table 4: This table shows differential pose found by T matrix decomposition for a number of objects containing two models on different planes. The angles shown are the angles between the two planes which contain the models.

constant. Table 5 shows the good agreement between actual and experimental estimation of relative motion.

There are situations where the T matrix method is unstable. If the gradients (ie. $\frac{dq}{dp}$) of the quartic curves are similar it is difficult to locate the pq roots accurately. Iteration on the initial pose solution can dramatically improve the result. A typical problem quartic plot is shown in figure 8.

Very rarely ambiguities in the solution can occur. In these cases the error costs are similar (within an order of magnitude of each other and both very small), and we

position	σ	τ	r (mm)	θ
1	35^o	94^o	280	182^o
2	35^o	90^o	277	163^o
3	36^o	88^o	277	144^o

Table 5: As the gasket is rotated in its plane by 10^o the estimated pose changes by a similar amount. The plane position, given by σ, τ and r remains fairly constant.

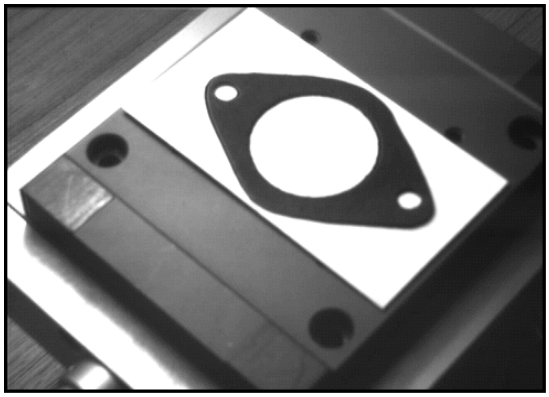

Figure 7: We are able to represent objects which do not have conic features using the frame invariant fitting theorem. The gasket curves represented by conics are the outer gasket curve and the large circular hole. The gasket was rotated on a calibration table and the estimated relative motion found by T matrix decomposition and the actual motion compared.

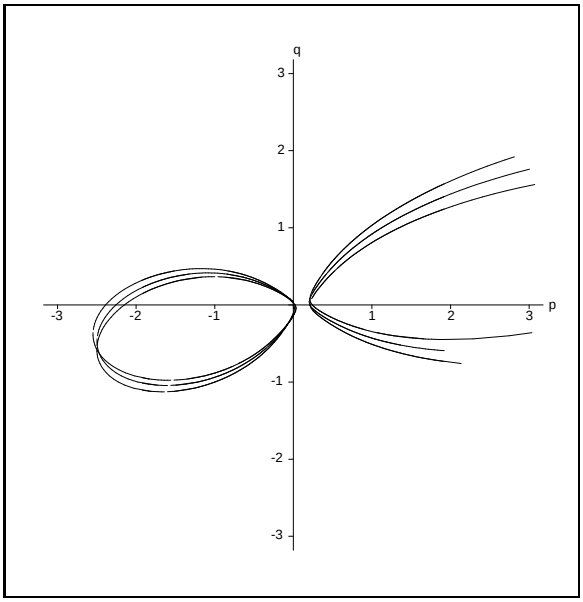

Figure 8: Graph showing how for some configurations it may be very difficult to obtain accurate estimates of the pq roots. In this case the angle between tangents at the point of intersections is very small. The first two quartics converge at 6^o at the root corresponding to the correct pose $((p, q) = (-0.062, -0.233))$.

have no way of distinguishing the correct pose. This is because the models we have used are the simplest possible, and there may not always be a unique projection from world conics to the image conics. Using richer model descriptions (ie. not just two coplanar conics), will overcome this problem.

5.3 Summary.

We have found that both conic intersections and bi-tangent lines of a conic pair can be used to recover pose, but for some objects the method is consistently inaccurate. If the intersection points are too close the method is obviously likely to fail. The same is true of the bi-tangent lines. Hence the four point methods do not produce reliable estimates of pose.

The results above show how decomposing the T matrix into M and H matrices yield stable solutions to pose. The results also highlight when instabilities occur and how we can spot them by monitoring the tangents to the quartics at intersections. The next section further highlights the stability of the T matrix decomposition method.

6 Stability.

6.1 Stability to Error in the Conic Coefficients.

The above decomposition using quartics can be compared to the method given in [17] which uses similar equiform invariants but derives conics in p and q. The conics are derived in appendix G. We have found that the method described above has greater stability to errors in the conic coefficients than that using conics. For example the quartics are totally immune to incorrect normalisation, whereas the conics can change considerably. The reason for this is that the each pq quartic depends on only one of the image conics, and errors cancel to some extent. Each pq conic depends on both image conics and errors are combined. Errors in the image conic coefficients are caused by both numerical error in normalisation and errors in the early vision and fitting. Figure 9 shows how the conics change when the scaling of the coefficients of one conic (that used in the synthetic example in section 5), is changed by $1, 5$ and 10%.

A more common problem is numerical inaccuracy in sin-

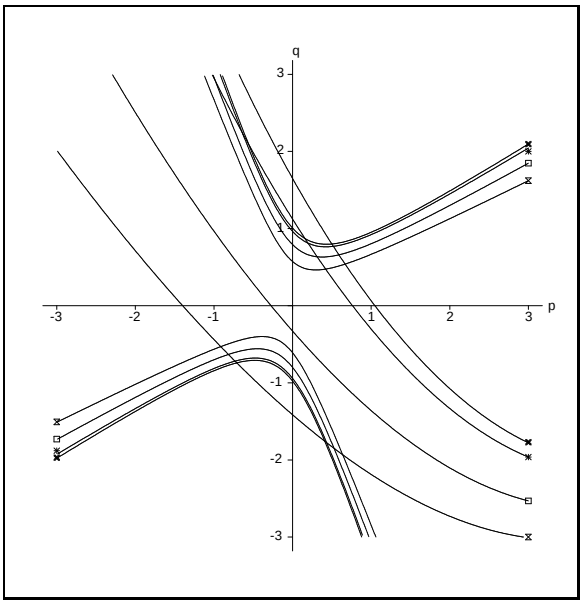

Figure 9: Graph showing how the pq conics change when the normalisation of one image conic is changed. We see that one of the conics changes rapidly for small changes in the normalisation. The unperturbed conics are denoted by an '**x**'. The change is scaling the coefficients of one image conic by 1% (asterisk), 5% (box) and 10% (timer). The unperturbed solution is at $p = 0.5, q = 0.8$. Any more than a 1% change dramatically changes the pq root, and for a 5% change there is no real root. The roots of the pq quartics are immune to changes in the normalisation.

% change in:	p	q	r	θ	t_x	t_y
changing c_x	1.8	2.2	0.8	0.0	4.2	6.4
changing f	0.0	0.7	0.6	0.0	0.0	0.0

Table 6: Table showing percentage change in pose parameter per percentage change in camera centre c_x or focal length f. The pose parameters are very stable to error in focal length, and moderately stable to errors in the camera centre.

gle image conic coefficients. Figure 10 shows how the pq conics change for 1, 5 and 10% changes made to the coefficient of x^2 in one conic. The right-hand graph shows the changes in the first two pq quartics for a 5 and 10% change in the same coefficient. Here we see that the system of quartics in p and q provide a far more stable solution than the conics. Changing other image conic coefficients provides similar results.

6.2 Stability to Camera Calibration.

The camera parameters were found using Tsai's technique [16]. The results in table 6 show how pose varies for small changes in the calibration parameters. The units we use are the percentage change in pose parameter p per percentage change in camera calibration parameter c. Hence the measure is $(\delta p/p)/(\delta c/c) \times 100\%$. The camera optical centre is at c_x, c_y and has focal length f. The results shown are for the image in figure 6.

We find that changing the vertical position of the camera centre has the same effect as changing the horizontal centre. The results show that the decomposition method is very stable with changes in the camera focal length and moderately stable to changes in the camera optical centre.

7 Conclusions.

We have demonstrated that pose and relative motion can be computed from two known coplanar curves. The method described which determines the world plane parameters p and q first, and then the other four pose parameters, is shown to be robust. We have demonstrated this technique for the case of conic pairs, however, the use of equiform invariants is applicable to any algebraic structures.

This technique enables model based vision systems which identify objects by using frame invariant properties of shapes, and then use the known object structure to determine pose. In turn, this means that projectively invariant labels such as those of [5] can be used as a direct source of positional information for a robot.

In tests on real objects, such as gaskets, where the curves are not conics, we have found the pose estimations to be as good as those for objects containing real conics. Being able to represent such objects by coplanar conic features allows us to build larger model bases which we can use for both recognition and pose determination.

7.1 Acknowledgements.

We are very grateful for discussions with Christopher Longuet Higgins in particular, and also Andrew Blake, Mike Brady, Steve Maybank, John Porrill and Gunnar Sparr. The support of GE (for CAR), SERC (for AZ), BP (for CIM), Magdalen College, Oxford (for DAF), General Electric Coolidge Fellowship (for JLM), and the University of Oxford are gratefully acknowledged.

References

[1] Bookstein, F. "Fitting Conic Sections to Scattered Data," CVGIP, 9, 56-91, 1979.

[2] Dhome, M., Lapreste J.T., Rives G., and Richetin M. "Spatial Localization of Modelled Objects of Revolution in Monocular Perspective Vision," Proceedings ECCV1, 1990.

[3] Faugeras, O.D., Lustman, F. and Toscanni, G. "Motion and Structure from Motion from Point and Line Matches," Proceedings ICCV1, 1987.

[4] Forsyth, D.A., Mundy, J.L., Zisserman, A.P. and Brown, C.M. "Projectively Invariant Representations Using Implicit Algebraic Curves," Proceedings ECCV1, 1989, also to appear in Image and Vision Computing, 1991.

[5] Forsyth, D.A., Mundy, J.L., Zisserman, A.P. and Brown, C.M. "Invariance :- a new framework for vision," Proceedings ICCV3, Osaka, 1990.

[6] Ganapathy, S. "Decomposition of Transformation Matrices for Robot Vision," Proceeding International Conference on Robotics and Automation, 1984.

[7] Hartshorne, R. *Foundations of Projective Geometry,* Benjamin, 1967.

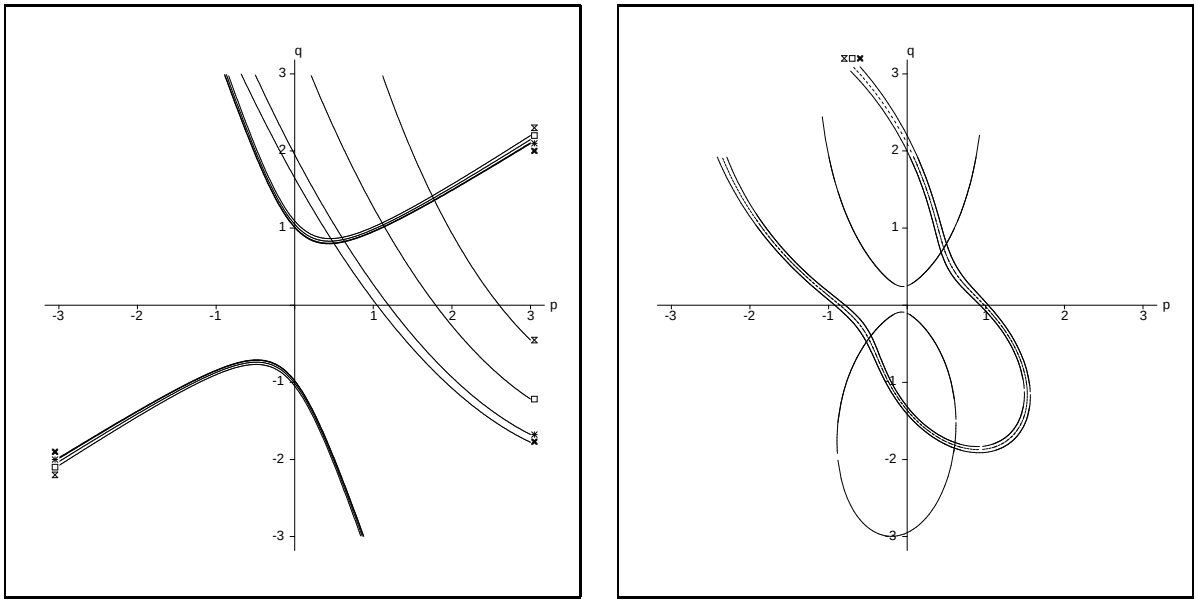

Figure 10: The left graph shows how the pq conics change for 1% (asterisk), 5% (box) and 10% (timer) changes in the coefficient of x^2. The unperturbed conics are denoted by an '**x**'. The right graph shows the changes to the pq quartics for 5% (box) and 10% (timer) changes. One of the quartics is unaffected by the changes. The original quartics are denoted by an '**x**'. The use of the quartics rather than the conics provides a far more stable solution to errors in the image conic coefficients

[8] Lee, C.H. and Huang, T. "Finding Four Point Correspondences and Determining Motion of a Rigid Object from two Weak Perspective Views," CVPR88, p398-403, 1988.

[9] Masciangelo, S. "3-D Cues from a Single View: Detection of Elliptical Arcs and Model-based Perspective Backprojection", Proceedings BMVC90, Oxford, 1990.

[10] Morgan, A.P. *Solving Polynomial Systems Using Continuation for Scientific and Engineering Problems*, Prentice-Hall, Englewood Cliffs, N.J., 1987.

[11] Nielsen, L. "Automated guidance of vehicles using vision and projectively invariant marking," Automatica, 24, 2, 135-148, 1988.

[12] Nielsen, L. and Sparr, G. "Projective Area-Invariants as an Extension of the Cross-Ratio" The 6th Scandinavian Conference on Image Analysis, Oulu, 969-986, 1989.

[13] Porrill, J. "Fitting ellipses and predicting confidence envelopes using a bias corrected Kalman filter," Image and Vision Computing, 8, 37-41, 1990.

[14] Shafarevich, *Basic Algebraic Geometry,* Springer, 1977.

[15] Thompson, D.W. and Mundy, J.L. "3D model matching from an unconstrained viewpoint," Proceedings IEEE Conference on Robotics and Automation, 1987.

[16] Tsai, R.Y. "An Efficient and Accurate Camera Calibration Technique for 3D Machine Vision," Journal of Robotics and Automation, RA-3 no. 4, 1987.

[17] Zisserman, A.P., Marinos, C.I., Forsyth, D.A., Mundy, J.L. and Rothwell, C.A. "Relative Motion and Pose from Invariants," Proceedings BMVC90, Oxford, 1990.

A Construction of the T matrix from a homogeneous 4×4 transformation matrix.

The 3×3 rotation matrix R, and a 3×1 translation vector $\mathbf{t}$ can be included in a homogeneous 4×4 matrix G, which will represent a general 3-space Euclidean action:

$$G = \begin{bmatrix} R & \mathbf{t} \\ 0 & 1 \end{bmatrix}$$

Let G represent the world to camera frame transformation with world frame points $\mathbf{X_4}$ and camera frame points $\mathbf{x_4}$:

$$\begin{aligned} \mathbf{x_4} &= G\mathbf{X_4} \\ &= \begin{bmatrix} \mathbf{u} & \mathbf{v} & \mathbf{w} & \mathbf{t} \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned}$$

where $\mathbf{u}$, $\mathbf{v}$ and $\mathbf{w}$ are 3×1 vectors all of unit norm and orthogonal to each other which form the columns of R. Under perspective projection for a camera of focal length f, the camera frame point $\mathbf{x_4}$ projects to the image plane point $\mathbf{x}$ by a matrix P where:

$$P = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{1}{f} & 0 \end{bmatrix}$$

Hence $\mathbf{x} = PG\mathbf{X_4}$, where:

$$PG = \begin{bmatrix} u_x & v_x & w_x & t_x \\ u_y & v_y & w_y & t_y \\ \frac{u_z}{f} & \frac{v_z}{f} & \frac{w_z}{f} & \frac{t_z}{f} \end{bmatrix}$$

All curve points lie in the world XY plane, and so $Z = 0$. Hence we may drop the third column of the transformation matrix. We also normalise the image points so that they lie on the plane $z = 1$ rather than $z = f$. To do this we must have calibrated the camera to determine its parameters (focal length f, and optical centre (c_x, c_y)). Hence if x' and y' are the new image points, x and y the image coordinates about the camera centre on the actual image plane, we let $x' = x/f$ and $y' = y/f$. This can be achieved by scaling the third element of homogeneous image vector by f. Hence:

$$\begin{aligned} \mathbf{x} &= \begin{bmatrix} u_x & v_x & t_x \\ u_y & v_y & t_y \\ u_z & v_z & t_z \end{bmatrix} \mathbf{X} \\ \Rightarrow T &= \begin{bmatrix} \mathbf{u} & \mathbf{v} & \mathbf{t} \end{bmatrix} \end{aligned}$$

Hence the first two columns of T are orthogonal and of equal norm.

B Forming 8 linear simultaneous equations by matching 4 world and image points.

We can represent a world point (X_i, Y_i) by the homogeneous vector $\mathbf{X_i} = (X_i, Y_i, 1)^T$. Likewise an image point (x_i, y_i) may be represented $\mathbf{x_i} = (x_i, y_i, 1)^T$. Under a projective mapping T:

$$\begin{aligned} k\mathbf{x_i} &= T\mathbf{X_i} \\ &= \begin{bmatrix} A & B & C \\ D & E & F \\ G & H & 1 \end{bmatrix} \mathbf{X_i} \end{aligned}$$

where $\{A, B, C, ..., H\} \in \mathcal{C}$ and $k \in \mathcal{C}$ is a scaling. Hence:

$$\begin{aligned} kx_i &= AX_i + BY_i + C \\ ky_i &= DX_i + EY_i + F \\ k &= GX_i + HY_i + 1 \end{aligned}$$

Eliminating k and taking $i \in \{1, .., 4\}$ gives eight linear equations in the eight unknowns $\{A, B, C, ..., H\}$ which are solvable by Gaussian elimination:

$$\begin{aligned} AX_i + BY_i + C - GX_ix_i - HY_ix_i &= x_i \\ DX_i + EY_i + F - GX_iy_i - HY_iy_i &= y_i \end{aligned}$$

C Extracting the pose parameters from the T matrix.

Extracting pose and camera parameters from a transformation matrix is a commom problem in machine vision. Similar methods are given in [6, 16]. We find that good results can only be obtained when the T matrix is a good approximation to a perspectivity (equal norms and orthogonality of the first two columns of T). Using this assumption means that the pose parameters can be estimated directly from the T matrix. As the T matrix has 8 constraints, and there are only 6 pose parameters it would be possible to use a least squares or pseudo-inverse method to account for all of the constraints. We find this method unnecessary

The T matrix found by 4 point back projection is of the form:

$$T = \begin{bmatrix} A & B & C \\ D & E & F \\ G & H & 1 \end{bmatrix}$$

When we compare this to equation 11 we find that:

$$\frac{D^2 + E^2}{DG + EH} = \frac{1 + p^2}{q} \quad (25)$$

$$(\frac{EG - DH}{DG + EH})^2 = (1 + p^2 + q^2)\frac{p^2}{q^2} \quad (26)$$

Eliminating q from equations 25 and 26 gives a cubic in p^2, with only one positive real root, which we can solve apart from a sign ambiguity in p (the sign of p is equal to the sign of $EG - DH$). Substituting this into equation 25 then solves for q. We can then solve for k using a number of constraints, for example:

$$k = +\sqrt{\frac{D^2 + E^2}{(1 + p^2)^2}}$$

Knowing p, q and k allows us to solve for r (though we find this method of finding r does not always give good results):

$$\begin{aligned} |T| &= k|M||H| \\ &= k(1 + p^2 + q^2)^{-1/2}r \\ \Rightarrow r &= \frac{|T|}{k}(1 + p^2 + q^2)^{1/2} \end{aligned}$$

We find θ from $\theta = tan^{-1}(D/E)$, and finally t_x and t_y are found by solving the following simultaneous equations:

$$\begin{aligned} AEt_x + BEt_y &= -CE \\ BDt_x + BEt_y &= -BF \end{aligned}$$

D Derivation of the bi-tangent lines of a conic pair.

Consider a conic $F_1 = 0$ where:

$$F_1 = \mathbf{x}^T C_1 \mathbf{x}$$

The normal to F_1 at $\mathbf{x_1}$ is $\nabla F_1 = (2\mathbf{x_1}^T C_1)^T$. The tangent line at $\mathbf{x_1}$ is the set of points $\mathbf{r}$ such that:

$$\begin{aligned} (\nabla F)^T|_{x1} \cdot \mathbf{r} &= 0 \\ \Rightarrow 2\mathbf{x_1}^T C_1 \cdot \mathbf{r} &= 0 \\ \Rightarrow \mathbf{x_1}^T C_1 \cdot \mathbf{r} &= 0 \end{aligned}$$

Let $\mathbf{l_1}^T = \mathbf{x_1}^T C_1$ then $\mathbf{l}_1^T \cdot \mathbf{r}$ is a line. Now,

$$\begin{aligned} \mathbf{l_1}^T C_1^{-1} \mathbf{l_1} &= \mathbf{x_1}^T C_1 C_1^{-1} \mathbf{l_1} \\ &= \mathbf{x_1}^T \mathbf{l_1} \\ &= \mathbf{x_1}^T C_1 \mathbf{x_1} \\ &= 0 \end{aligned}$$

Likewise for the second curve:

$$\mathbf{l_2}^T . C_2^{-1} . \mathbf{l_2} = 0.$$

If $\mathbf{l}$ is a bi-tangent line (i.e. $\mathbf{l} = \mathbf{l_1} = \mathbf{l_2}$) then it obeys both $\mathbf{l}^T . C_1^{-1} . \mathbf{l} = 0$ and $\mathbf{l}^T . C_2^{-1} . \mathbf{l} = 0$. Hence $\mathbf{l}$ is given by the intersections of C_1^{-1} and C_2^{-1}. Using the duality of points and lines the bi-tangent line $\mathbf{l}$ may be thought of as a point. This point is preserved under projection.

E Derivation of the equiform invariants.

Under the equiform mapping H, conic P maps to conic p by $p = H(P)$. Using equations 5 or 6, this may be written:

$$P = |H|^{-\frac{2}{3}} . H^T \; p \; H, \qquad |p| = 1$$

where H is a 3×3 matrix represented by a 2×2 rotation matrix R, a 2×1 translation vector $\mathbf{t}$, and a scaling r:

$$H = \begin{bmatrix} R & \mathbf{t} \\ 0 & r \end{bmatrix}, \qquad |H| = r$$

Let p be written as a combination of a 2×2 matrix A_1, a 2×1 vector A_2 and a scalar A_3:

$$p = \begin{bmatrix} A_1 & A_2 \\ A_2^T & A_3 \end{bmatrix}$$

Hence:

$$\begin{aligned} P &= r^{-\frac{2}{3}} . H^T p H \\ &= r^{-\frac{2}{3}} \begin{pmatrix} R^T A_1 R & R^T A_1 \mathbf{t} + R^T A_2 r \\ \mathbf{t}^T A_1 R + r A_2^T R & \mathbf{t}^t A_1 \mathbf{t} + \mathbf{t}^T A_2 r + r A_2^T \mathbf{t} + r A_3 r \end{pmatrix} \end{aligned}$$

Considering only the upper 2×2 submatrix, and writing R and A_1 as:

$$R = \begin{bmatrix} c_\theta & -s_\theta \\ s_\theta & c_\theta \end{bmatrix} \text{ and } A_1 = \begin{pmatrix} A & B/2 \\ B/2 & C \end{pmatrix}$$

where $c_\theta = cos\theta$ and $s_\theta = sin\theta$, gives:

$$\begin{aligned} {}^{2\times2}[P] &= r^{-\frac{2}{3}} . R^T A_1 R \\ &= r^{-\frac{2}{3}} \begin{pmatrix} A c_\theta^2 + B c_\theta s_\theta + C s_\theta^2 & \frac{B}{2}(c_\theta^2 - s_\theta^2) - A c_\theta s_\theta + C c_\theta s_\theta \\ \frac{B}{2}(c_\theta^2 - s_\theta^2) - A c_\theta s_\theta + C c_\theta s_\theta & C c_\theta^2 - B c_\theta s_\theta + A s_\theta^2 \end{pmatrix} \end{aligned}$$

Hence:

$$\begin{aligned} trace^{2\times2}[P] &= r^{-\frac{2}{3}} (A + C) \\ &= r^{-\frac{2}{3}} \; trace^{2\times2}[p] \\ det^{2\times2}[P] &= r^{-\frac{4}{3}} (AC - B^2/4) \\ &= r^{-\frac{4}{3}} \; det^{2\times2}[p] \end{aligned}$$

Writing $r = |H|$ and $p = H(P)$:

$$\begin{aligned} trace^{2\times2}[H(P)] &= |H|^{\frac{2}{3}} \; trace^{2\times2}[P] \\ det^{2\times2}[H(P)] &= |H|^{\frac{4}{3}} \; det^{2\times2}[P] \end{aligned}$$

These compare to the definition of an invariant $I(H(P)) = |H|^w \; I(P)$ under the mapping H, and so $trace^{2\times2}$ and $det^{2\times2}$ are equiform invariants of weights $\frac{2}{3}$ and $\frac{4}{3}$.

F Derivation of the M matrix.

The M matrix represents the projection from the plane $\Pi : z = px + qy + 1$ to the image plane $z = 1$ through the point $(0, 0, 0)$. The $\tilde{X}$ axis of Π lies in the same plane as the camera x axis. The $\tilde{Z}$ axis lies parallel to the camera vector $(p, q, 1)^T$. First we derive the transformation G taking points in the Π frame to camera frame. This is a rotation R followed by a translation $(0, 0, 1)$.

To determine the rotation we derive the inverse rotation by rotating the surface normal of Π so that it is parallel to the optical axis $(0, 0, 1)^T$. The inverse rotation is made up of two rotations:

1. A rotation θ about the image y axis which rotates the $\tilde{X}$ axis so that is parallel to the x axis.
2. The new $\tilde{Y}$ and y axes are now co-planar. A rotation ϕ about the x axis will make them parallel, and so make the two planes parallel.

The normal to the plane is $\mathbf{n} = (p, q, 1)^T$. A rotation θ about the y axis is represented by:

$$R_y(\theta) = \begin{bmatrix} c_\theta & 0 & s_\theta \\ 0 & 1 & 0 \\ -s_\theta & 0 & c_\theta \end{bmatrix}$$

$$\text{where} \quad c_\theta = cos_\theta \quad \text{and} \quad s_\theta = sin_\theta$$

and a rotation ϕ about the x axis is represented by:

$$R_x(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_\phi & -s_\phi \\ 0 & s_\phi & c_\phi \end{bmatrix}$$

$$\text{where} \quad c_\phi = cos_\phi \quad \text{and} \quad s_\phi = sin_\phi$$

Hence $\mathbf{n}' = R_x(\phi)R_y(\theta)\ \mathbf{n}$:

$$\mathbf{n}' = \begin{pmatrix} pc_\theta + s_\theta \\ qc_\phi + ps_\theta s_\phi - c_\theta s_\phi \\ c_\theta c\phi + qs_\phi - ps_\theta c_\phi \end{pmatrix}$$

$\mathbf{n}'$ is parallel to the vector $(0, 0, 1)$, so $pc_\theta + s_\theta = 0$. Therefore:

$$c_\theta = \frac{1}{\sqrt{1+p^2}} \qquad \text{and} \qquad s_\theta = \frac{-p}{\sqrt{1+p^2}}$$

$$\Rightarrow \mathbf{n}' = (0, qc_\phi - \sqrt{1+p^2}s_\phi, qs_\phi + \sqrt{1+p^2}c_\phi)^T$$

and so $qc_\phi - \sqrt{1+p^2}s_\phi = 0$. Hence:

$$c_\phi = \frac{\sqrt{1+p^2}}{\sqrt{1+p^2+q^2}} \qquad \text{and} \quad s_\phi = \frac{q}{\sqrt{1+p^2+q^2}}$$

Therefore we write R:

$$R = R_y^{-1}(\theta).R_x^{-1}(\phi) = \begin{bmatrix} \frac{1}{\sqrt{1+p^2}} & \frac{-pq}{\sqrt{1+p^2+q^2}\sqrt{1+p^2}} & \frac{p}{\sqrt{1+p^2+q^2}} \\ 0 & \frac{\sqrt{1+p^2}}{\sqrt{1+p^2+q^2}} & \frac{q}{\sqrt{1+p^2+q^2}} \\ \frac{-p}{\sqrt{1+p^2}} & \frac{-q}{\sqrt{1+p^2+q^2}\sqrt{1+p^2}} & \frac{1}{\sqrt{1+p^2+q^2}} \end{bmatrix}$$

The translation of G is $\mathbf{t} = (0, 0, 1)^T$. From appendix A,

$$G = \begin{bmatrix} R & \mathbf{t} \\ 0 & 1 \end{bmatrix}$$

and so using the result of appendix A, the projection matrix M mapping from the Π frame to the camera frame linked by G is:

$$M = \begin{bmatrix} \frac{1}{\sqrt{1+p^2}} & \frac{-pq}{\sqrt{1+p^2+q^2}\sqrt{1+p^2}} & 0 \\ 0 & \frac{\sqrt{1+p^2}}{\sqrt{1+p^2+q^2}} & 0 \\ \frac{-p}{\sqrt{1+p^2}} & \frac{-q}{\sqrt{1+p^2+q^2}\sqrt{1+p^2}} & 1 \end{bmatrix}$$

G Derivation pq conics used in BMVC90.

From equation 15 for two world conics C_1 and C_2 with images c_1 and c_2:

$$trace^{2\times 2}[C_1] = \kappa \frac{f_t(p, q, c_1)}{1 + p^2 + q^2}$$
$$trace^{2\times 2}[C_2] = \kappa \frac{f_t(p, q, c_2)}{1 + p^2 + q^2}$$

where f_t is a conic in p and q with coefficients taken from the relevant image conic:

$$f_t(p, q, c) = a + c - dp + cp^2 + fp^2 - eq - bpq + aq^2 + fq^2$$

Hence:

$$\frac{trace^{2\times 2}[C_1]}{trace^{2\times 2}[C_2]} = \frac{f_t(p, q, c_1)}{f_t(p, q, c_2)}$$

Which when multiplied out gives a conic in p and q. The other conic is derived using equation 16. This gives:

$$\frac{det^{2\times 2}[C_1]}{det^{2\times 2}[C_2]} = \frac{f_d(p, q, c_1)}{f_d(p, q, c_2)}$$

where:

$$f_d(p, q, c) = (4ac - b^2 - 4cdp + 2bep - e^2p^2 + 4cfp^2 + 2bdq - 4aeq + 2depq - 4bfpq - d^2q^2 + 4afq^2)/4$$