Relative motion and pose from invariants

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

Projectively invariant shape descriptors efficiently identify instances of object models in images without reference to object pose. These descriptions rely on frame independent representations of planar curves, using plane conics.

We show that object pose can be determined from coplanar curves, given such a frame independent representation. This result is demonstrated for real image data.

The shape of objects in images changes as the camera is moved around. This extremely simple observation represents the dominant problem in model based vision. Nielsen [4, 5] first suggested using projectively invariant *labels* as landmarks for navigation. Recent papers [1, 2] have shown that it is possible to compute shape descriptors of arbitrary plane objects that are unaffected by camera position. These descriptors are known as transformational invariants. At no stage in this process, however, is the pose of the model determined. In this paper, we show that the available information does in fact determine the pose of the model. In particular, for complex planar objects, pose determination can be reduced to the simpler problem of pose determination for a pair of *known* planar conics.

For future reference we note the following results on the use of projective invariants in model based vision [1, 2]:

- Plane data can be *represented* by algebraic curves in a frame invariant manner [1]. 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 projectively invariant manner. The algebraic curve becomes a projectively invariant representation. In the sequel we concentrate on representation by conic curves.

- A pair of co-planar conic curves admit two *scalar projective invariants* [1]. These are two numbers computed from the conics in a particular frame (e.g. the world plane or the image plane) which are frame independent - their values are unaffected by projection. A pair of coplanar curves is represented by a pair of coplanar conics.

- These two numbers are an invariant shape descriptor. Image measurements of these descriptors can be matched to object properties regardless of position, orientation and intrinsic parameters of the camera.

- Existing polyhedral model based vision systems conflate the two distinct problems of library indexing and of estimating transformation parameters. 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. Using invariant shape descriptors models can be found in a library *without* having to determine transformation parameters.

Once an object has been positively identified, the extra constraints offered by its known identity can be exploited to determine transformation parameters. Since invariant fitting allows a pair of coplanar curves to be modeled by a pair of coplanar conics, and since, by construction, the modelling conics undergo the same projective distortion that the original curves do, finding position and orientation is reduced to the question of back-projecting a pair of conics. Consequently, the problem addressed in this paper is:

> Given a known pair of conics on the world plane, and their corresponding conics in the image, determine the transformation between the two planes.

The solution of this problem determines the object *pose*. That a solution is possible in principle follows from:

1. Two conics always intersect in four points (though the intersections may be complex). This gives four corresponding points on the image and world plane.

2. Apart from the combinatorics of matching these points, 4 points are sufficient to determine the projection between two planes [7].

BMVC 1990 doi:10.5244/C.4.4

The paper is organised as follows. First we outline the solution to the conic pair back-projection problem. Then we describe model acquisition and the application of the method to real data. Because conic fitting is notoriously ill-conditioned when data only covers a small part of the conic [6] the following discussion focuses on ellipses representing closed curves.

BACK PROJECTION OF A CONIC PAIR

Conic Notation

A conic curve is given by

$$Q(x,y) = Ax^2 + Bxy + Cy^2 + Dx + Ey + F = 0 \quad (1)$$

This can also be written:

$$Q(\mathbf{x}) = \mathbf{x}^T\mathbf{P}\mathbf{x} = 0, \text{ where } \mathbf{P} = \begin{bmatrix} A & \frac{B}{2} & \frac{D}{2} \\ \frac{B}{2} & C & \frac{E}{2} \\ \frac{D}{2} & \frac{E}{2} & F \end{bmatrix}$$

$\mathbf{P}$ is the coefficient matrix, and $\mathbf{x} = (x_1\ x_2\ x_3)^T$. Note that equation (1) above for the conic in Euclidean coordinates is obtained by performing the indicated matrix operations and then setting $x_3 = 1$. Unfortunately, if $Q(x,y) = 0$, then $kQ(x,y) = 0$, for k any real number. So although the *curves* are the same the polynomials are different. To avoid this problem we impose a normalising constraint on the polynomial, namely $\det(\mathbf{P}) = 1$.

In the following the conics fitted in the image plane are $\mathbf{P'}_1, \mathbf{P'}_2$, and those in the world plane (the "model") $\mathbf{P}_1, \mathbf{P}_2$. Under a change of frame ($\mathbf{x'} = \mathbf{T}\mathbf{x}$) the conics transform as

$$\mathbf{P}_1 = k\mathbf{T}^T\mathbf{P'}_1\mathbf{T} \quad (2)$$
$$\mathbf{P}_2 = k\mathbf{T}^T\mathbf{P'}_2\mathbf{T} \quad (3)$$

and the problem here is to determine $\mathbf{T}$ which produces the "closest" match between the known $\mathbf{P}$ and those computed from the image conics transformed as above.

Perspective Transformation

A perspective projection between 2 planes is determined by 6 parameters (given a known projection point). After normalisation, each conic has 5 independent parameters, so by a simple degrees of freedom argument the solution is overdetermined (10 constraints on 6 unknowns). We have compared several methods for obtaining the back projection and report here only the current best choice. Other schemes may well exist which improve on this. The advantages of the current scheme are:

1. Ambiguities in the solutions are clearly visible.
2. It is tolerant of noise in the fitted conics (by using least squared costs).
3. No iteration is involved, each stage of the process has a closed form solution.

A perspective transformation projects points on the world plane to points on the image plane, and hence defines a mapping between coordinate systems on the two planes. It can be shown that this is a linear transformation in homogenous coordinates $\mathbf{x}_I = \mathbf{T}\mathbf{x}_w$ where $\mathbf{x}_w = (x_1\ x_2\ x_3)^T$ with world plane coordinates $x_w = x_1/x_3$, $y_w = x_2/x_3$; and $\mathbf{x}_I = (X\ Y\ f)^T$ with image coordinates (X, Y). The 6 parameters that specify the transformation can be interpreted as follows.

1. Three parameters $\{p, q, r\}$ specify the world plane in an image 3D coordinate frame with origin the focal point and z axis the camera optical axis. In this frame the world plane's equation is $z_I = px_I + qy_I + r$. $\{p, q\}$ specifies the orientation and r the intercept of the plane with the optical axis. There is a natural mapping between the image $z_I = f$ and world planes (this is shown in figure 1). The image coordinate system *induces* a coordinate system on the world plane: $\mathbf{x}_I = \mathbf{M}(p,q,r)\mathbf{x}_{\text{induced}}$.
2. Three parameters $\{t_x, t_y, \theta\}$ specify an in plane translation and rotation between the induced coordinate system and the actual coordinate system on the world plane, $\mathbf{x}_w = \mathbf{R}_2(\theta)\mathbf{x}_{\text{induced}} + \mathbf{t}$. This can be written $\mathbf{x}_{\text{induced}} = \mathbf{H}(t_x, t_y, \theta)\mathbf{x}_w$.

The matrices are given by:

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

$$\mathbf{H} = \begin{bmatrix} c & -s & st_y - ct_x \\ s & c & -ct_y - st_x \\ 0 & 0 & 1 \end{bmatrix}$$

where $c = \cos\theta$ and $s = \sin\theta$. Thus $\mathbf{T} = \mathbf{M}(p,q,r)\mathbf{H}(t_x, t_y, \theta)$. Under this perspective projection between the world and image plane the conic matrix transforms as $\mathbf{P}_w = k\mathbf{T}^T\mathbf{P}_I\mathbf{T}$.

The current scheme recovers these parameters in the following stages:

1. Orientation of the plane $\{p, q\}$.
2. Distance r.
3. In plane rotation and translation $\{\theta, \mathbf{t}\}$.

This partition is used because once the orientation has been determined there is no change of "shape" of the conic pair. The remaining parameters affect scaling and the position of the conics relative to the coordinate system on the world plane.

Orientation of the plane $\{p, q\}$

This is the most difficult of the stages and we devote most space to it here. To motivate the problem consider the case of a circle on the world plane. For a circle there are 2 constraints on the back-projection, namely:

1. The x^2 and y^2 terms in equation (1) have the same coefficient.
2. The coefficient of xy in equation (1) is zero.

This places 2 constraints on the 2 unknowns p and q which specify world plane orientation. These 2 constraints are sufficient to determine the 2 unknowns up to a 2 fold ambiguity. In the case of an ellipse however, similar constraints can be applied (aspect ratio and zero xy coefficient) but only in a special coordinate system - namely the natural frame of the ellipse. So there are in fact 3 unknowns the additional one being the angle (θ) between the coordinate system of the ellipse and the world coordinate frame. Thus, a single ellipse restricts the solution only to a curve (rather than a point set) in the (p, q) plane. Two ellipses, however, are sufficient because the 2 constraint curves will intersect in at most a finite number of points (counting constraints there are 4 constraints on 3 unknowns).

There are a number of relations that can be formed for a pair of conics from equations (2)-(3). We report here only the simplest. In a similar manner to the above 2 constraints for a circle we construct 2 functions of $\{p, q\}$ whose values are known on the world plane (from the model). We can then determine the values of $\{p, q\}$ which bring the function values back to the model values. The functions are:

$$\Theta_T(p,q) = \frac{\text{trace}_1}{\text{trace}_2} \qquad \Theta_D(p,q) = \frac{\det_1}{\det_2} \qquad (4)$$

where "trace" and "det" refer to the trace and determinant of the upper 2x2 of the conic matrix (equations (2)-(3)). These are quantities unaffected by in plane rotation and translation (the **H** part of **T**) the ratios are unaffected by scaling. They are given by:

$$\begin{aligned}
\text{trace} &= A' + C' + D'p + C'p^2 + F'p^2 \\
&\quad +E'q - B'pq + A'q^2 + F'q^2 \\
\det &= -(B')^2 + 4A'C' + 4C'D'p \\
&\quad -2B'E'p - (E')^2p^2 + 4C'F'p^2 \\
&\quad -2B'D'q + 4A'E'q + 2D'E'pq \\
&\quad -4B'F'pq - (D')^2q^2 + 4A'F'q^2
\end{aligned}$$

The known values in the world plane are

$$\Theta_T = \frac{A_1 + C_1}{A_2 + C_2} \qquad \Theta_D = \frac{4A_1C_1 - B_1^2}{4A_2C_2 - B_2^2}$$

Note: common factors are omitted; a prime indicates an image rather than model quantity; and trace and det involve all the coefficients of the image conic.

The functions each give a curve in the $\{p, q\}$ plane. The intersections of the curves determines $\{p, q\}$. Geometrically the constraints are related to the relative areas, and aspect ratios of the two ellipses, but *not* their relative orientation (counting constraints we have used 2 constraints, equation (4), to determine 2 unknowns, $\{p, q\}$).

Ambiguity of solutions

Equation (4) represents two *conic* curves in the $\{p, q\}$ plane, and thus intersect in at most 4 real points (if there are no real intersections then an iterative approach must be used). Figure 2a shows an example of the curves for the data of figure 3. In general then there is a four fold ambiguity. However, some of these solutions can be removed by judicious use of a *visibility constraint.* This is demonstrated in figures 2bc. It can be shown that points on the image line with equation $(p\ q)^T(X\ Y) = 1$ are back projected to an *ideal point*, also called a point at infinity (the back projection of this image line is a line in space parallel to the object plane - i.e. the intersection is at infinity). Thus for any point on an image contour there is a set of values of $(p\ q)$ (a line in p, q space) which can not occur since if they did the back-projected point would be at infinity. That would mean that the back-projected contour is an open curve which violates the assumption that the model is an ellipse. Moving around the image curve generates a forbidden region in p, q space which is bound by the envelope of the lines. It can be shown that if the image contour is a conic then the envelope is also a conic.

Determining the remaining 4 parameters

Distance r

Since $\{p, q\}$ are known, the conic pair can be back projected onto a plane parallel to the world plane. On any such plane the ellipses project to the same shape, but their scale varies. The distance r then, is simply a scaling parameter which can be recovered by minimising a cost based on "area" (a measure unaffected by the as yet undetermined t_x, t_y, θ). The area of an ellipse is calculated from the *diagonalised* conic matrix as area $= \pi\left(F - D^2/(4A) - E^2/(4C)\right)/\sqrt{(AC)}$.

The area on the world plane A_w is known, and the back-projected area scales as r^2, i.e. if A_1 is the area of $\mathbf{T}(p\ q\ 1)^T\mathbf{P}_I\mathbf{T}(p\ q\ 1)$, then the area of $\mathbf{T}(p\ q\ r)^T\mathbf{P}_I\mathbf{T}(p\ q\ r)$ is r^2A_1. The cost used is the squared difference in areas $\sum_i \left(\mathrm{A}^i_w - r^2\mathrm{A}^i_{pq1}\right)^2$ and the minimum is given by $r^2 = \sum_i \mathrm{A}^i_w.\mathrm{A}^i_{pq1} / \sum_i \mathrm{A}^i_{pq1}$, where the sum is over the two ellipses.

Rotation θ, Translation t

Having determined the back projected plane, all that remains is the in-plane rotation and translation between the induced and world coordinate systems. This is calculated by minimising the squared Euclidean distance between the centres of the back-projected conics, and the model conic centres. The translation is obtained from their centroids, and the rotation can be found in closed form. 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.

Using the matrix T

The parameters $\{p, q, r, \mathbf{t}, \theta\}$ determine the object pose relative to the camera coordinate system. They also define a transformation matrix $\mathbf{T}$. The information contained in $\mathbf{T}$ can be exploited in two ways:

1. To define a mapping between the 2D coordinate systems on the image and (model) world planes $\mathbf{x}_I = \mathbf{T}\mathbf{x}_w$. The inverse transformation is given by $\mathbf{T}^{-1}$.

2. To define a transform between camera and world 3D Euclidean systems: $\mathbf{x}_{\text{camera}} = \mathbf{R}_3\mathbf{x}_{\text{world}} + \mathbf{t_3}$ where $\mathbf{x}_{\text{camera}}$ and $\mathbf{x}_{\text{world}}$ are Euclidean 3-vectors, $\mathbf{R}_3 = \mathbf{O}_3(p,q)\mathbf{H}(0,0,\theta)$, where

$$\mathbf{O}_3(p,q) = \begin{bmatrix} \frac{1}{\sqrt{1+p^2}} & \frac{-pq}{\sqrt{1+p^2+q^2}\sqrt{1+p^2}} & \frac{p}{\sqrt{1+p^2+q^2}} \\ 0 & \frac{\sqrt{1+p^2}}{\sqrt{1+p^2+q^2}} & \frac{q}{\sqrt{1+p^2+q^2}} \\ \frac{-p}{\sqrt{1+p^2}} & \frac{-q}{\sqrt{1+p^2+q^2}\sqrt{1+p^2}} & \frac{1}{\sqrt{1+p^2+q^2}} \end{bmatrix}$$

and $\mathbf{t}_3 = \mathbf{M}(-p,-q,-r)\mathbf{H}(t_x,t_y,\theta)(0\ 0\ 1)^T$.

If the camera is moving then the above equation can be used to compute the camera's pose from each view relative to a fixed coordinate system on the object. It is then straightforward to recover ego-motion. Conversely, the object's relative motion can be determined. An example of this is given in the final section.

method	slant $\sigma/^o$	tilt $\tau/^o$	r/mm
conics	41.47	10.81	719.00
Tsai	44.45	13.13	701.49

Table 1: Comparison of recovered plane orientation and distance for the data of figure 3.

Building conic models of general plane curves

If the model curves are known conics then there is no difficulty. One needs to choose a coordinate system within which to express these conics, and a sensible choice is the natural frame of one of the conics, i.e. the centre of the ellipse as origin and coordinate axes aligned with the ellipse axes.

However, if the model curves are more general or are conics of unknown form, it is necessary to fit conics to them, using the invariant fitting techniques of [1]. If the curves are known in the world plane coordinate system then conic fitting directly produces the model. If the curves are not known in the world frame then they can be obtained by fitting curves in the image (using the invariant fitting method [1]) and back projecting to the world plane. This requires knowledge of the relation between the camera and the world plane. The easiest way to obtain this is to image the object together with a known *calibration* pair of conics (or single circle) such that it is coplanar with the model curve. The calibration curves are used to compute the transformation between image and world plane. This transformation can then be used to back-project the representing conics to the model plane. This will give the representing conics that would have been found if the fitting had been carried out in the model plane.

APPLICATION TO REAL DATA

We include two examples here. In the first the curves are actually ellipses and we assess the accuracy of the back-projection transform for "ideal" data. In the second the curves are not conics and we measure relative motion between two views.

In the first case the model is two ellipses which are generated via PostScript, so their equations are known. The accuracy of transform obtained from the back-projection method can be assessed by firstly computing the discrepancy between the model and back-projected curves (see figure 3b); and secondly comparing the transformation parameters with those obtained from Tsai's calibration technique [8]. The results are given in table 1. The values are very close. In practice it is found that the values

of p and q are less stable when the slant is small (less than 20^o). However, if the model is known to be circular then two constraints can be applied to the curve and the results are better conditioned.

Finally, an example of object motion. The object is a SUN mouse modeled by the button (non-conic) curves. The model was acquired using the back-projection method described above. The mouse was rotated by $\sim 90^o$ remaining approximately in the same plane (see figure 4). Results are given in table 2. The

view	slant $\sigma/^o$	tilt $\tau/^o$	r/mm	$\theta/^o$
A	44.12	97.27	339.98	0.00
B	48.00	93.21	316.62	92.36

Table 2: Pose results for the two views of the mouse shown in figure 4. The mouse was rotated by $\sim 90^o$ between the views. The plane is the same in both cases.

computed orientation of the plane is constant to within 4^o of slant and 4^o of tilt. The computed in plane rotation is 92^o, which is an extremely good agreement with the actual motion.

Relative motion from unknown coplanar curves

It is possible to use these techniques, without reference to a model base, to obtain relative motion (rotation and translation direction) from two *unknown* coplanar curves. Again, this follows in principle by considering the 4 intersection points of the two representing conics. As is well known [3] 4 coplanar points are sufficient to determine the 5 parameters of relative motion. Furthermore, correct correspondence and coplanarity of the curves can be tested by exploiting the projectively invariant shape descriptors. If the descriptors vary, the curves have either been incorrectly matched or are not coplanar.

CONCLUSIONS

We have demonstrated that pose and relative motion can be computed from two known coplanar curves. These techniques enable model based vision systems which identify objects by using projectively invariant properties of shapes to determine object pose. In turn, this means that projectively invariant labels such as those of [2] can be used as a direct source of positional information by autonomous guided vehicles.

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 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] **Forsyth, D.A, Mundy, J.L., Zisserman, A.P. and Brown, C.M.** "Projectively invariant representations using implicit algebraic curves, " *Proc. 1st European Conference on Computer Vision*, Springer Verlag Lecture Notes in Computer Science, 1990.

[2] **Forsyth, D.A., Mundy, J.L. Zisserman, A.P. and Brown, C.M.** "Invariance- a new framework for vision," *To appear in 3rd International Conference on Computer Vision, 1990.*

[3] **Longuet-Higgins H.C.**, "The reconstruction of a plane surface from two perspective projections", *Proc. R. Soc. Lond.*, B 227, 399-410, 1986.

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

[5] **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.

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

[7] **Semple, J.G. and Kneebone, G.T.** *Algebraic Projective Geometry.* Oxford University Press, Oxford (1952).

[8] **Tsai, R.** "An efficient and accurate camera calibration technique for 3D machine vision.", *Journal of Robotics and Automation*, RA-3 4, pp. 323-344, 1987.

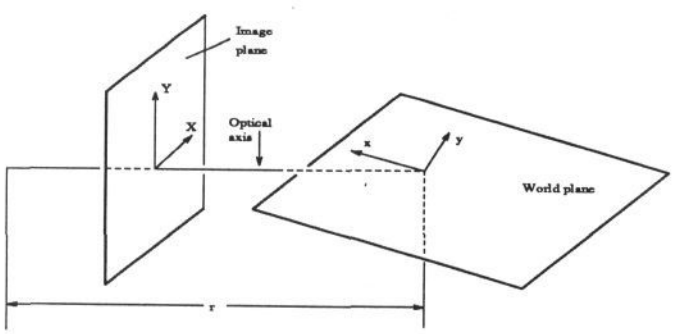

Figure 1: The image cartesian coordinate system $\mathbf{x}_I$ has origin the focal point and z axis the camera optical axis. The image plane is at $z = f$ as usual, and the world plane is $z_I = px_I + qy_I + r$.

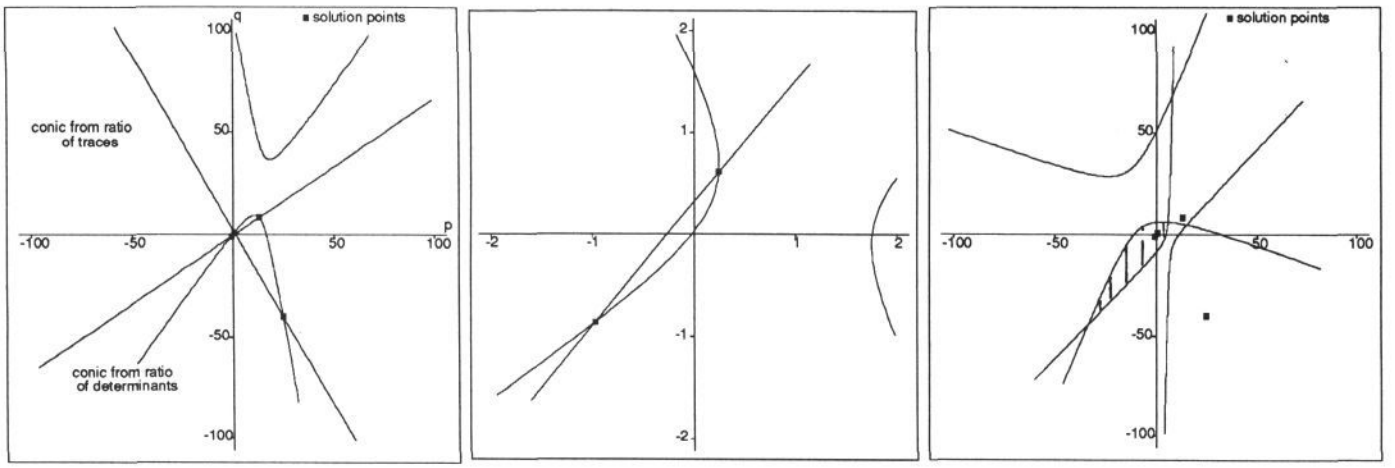

Figure 2: (a) The orientation p, q is obtained from the intersection of two conics (in this case hyperbolae). There are four intersection points in general. The curves are for the image shown in figure 3. (b) Central region of (a) in greater detail. (c) When the restrictions imposed by the visibility constraint are included there are only two solutions remaining. The feasible region is the region inside the envelopes (which are hyperbolae in this case).

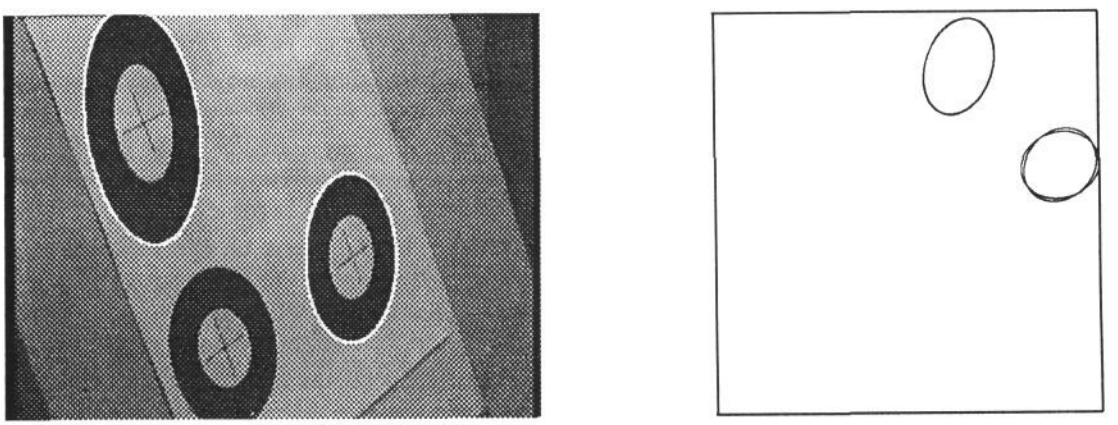

Figure 3: (a) The "model" in this case is the indicated pair of ellipses (produced in PostScript so their equations are known). (b) The back-projection of this image onto the model outline. The curves are almost indistinguishable.

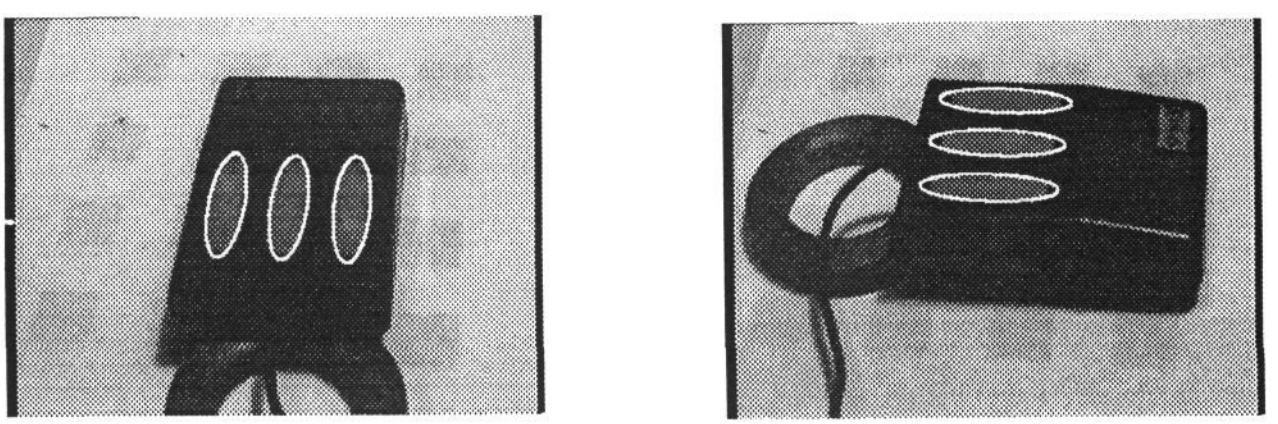

Figure 4: Images of a mouse with the representing conics superimposed. The motion between the views is a 90° rotation and small translation of the mouse with the camera static. The plane of the mouse buttons is approximately the same in both images. Results are given in table 2.