Pattern Recognition Vol. 17, No. 6, pp. 607–619, 1984.
Printed in Great Britain.

0031–3203/84 $3.00 + .00
Pergamon Press Ltd.

MATCHING WIRE FRAME OBJECTS FROM THEIR TWO DIMENSIONAL PERSPECTIVE PROJECTIONS

ROBERT M. HARALICK[1], YU HONG CHU[2], LAYNE T. WATSON[3] and LINDA G. SHAPIRO[1]

[1]Machine Vision International, Ann Arbor, MI 48104, [2]Department of Computer Science, Old Dominion University, Norfolk, VA 23508, and [3]Department of Computer Science, Virginia Polytechnic Institute and State University, Blacksburg, VA 24061, U.S.A.

(*Received* 13 *January* 1983; *in revised form* 15 *March* 1984; *received for publication* 11 *April* 1984)

Abstract—A wire frame object consists of a set of three dimensional arcs, each arc being a sequence of conics and line segments lying in the same plane, with different arcs being allowed to lie on different planes. Given a picture taken by a camera focusing on one wire frame object, we show how to determine what the object is and where it is situated relative to the camera when the camera viewing parameters are unknown.

To accomplish the object identification, we begin with a segmented picture. Then we construct a ray from the lens to each point on the boundary of every region. For each region, the collection of its associated rays is a cone. We show that by constructing cones, the two-dimensional to three-dimensional matching problem is transformed into an equivalent three-dimensional to three-dimensional matching problem.

This matching problem is expressed as a nonlinear optimization search procedure on the 6 camera viewing parameters: the 3 translation parameters and the 3 rotation parameters. A solution is found when a viewing position and optical axis is determined which is consistent with the world knowledge we have of possible curves and the observed image data.

Wire frame object matching   Object identification   Arc   Two-dimensional matching   Three-dimensional matching

1. INTRODUCTION

The problem of recognizing three-dimensional objects from their two-dimensional perspective projections is an important one in scene analysis. Here we treat a subset of this problem, namely the identification of a wire frame object from its two-dimensional perspective projection taken from an image produced by a camera.

Given for each wire frame object is the set of its three-dimensional arcs, each arc being a sequence of conic or line segments lying in the same plane. Different arcs may lie on different planes. The observed two-dimensional picture consists of a perspective projection of some wire frame object from the set of given wire frame objects. Our problem is to determine which wire frame object from the given set of wire frame objects produces the perspective projections recorded in the observed digitized image.[1]

As shown in Fig. 1, the camera is situated at a space point ($X1$, $Y1$, $Z1$), which we will call the center of projection. To allow for an arbitrary viewing direction, we rotate the camera coordinate systems. The pan angle θ is the first rotation, which is around the original z-axis. The tilt angle ϕ is the second rotation, which is around the new x-axis. The swing angle ψ is the third rotation, which is around the new y-axis. These rotations are shown in detail in Fig. 2. The image plane (perspective projection plane) lies F units of distance in front of the lens. The image plane is always perpendicular to the optical axis of the camera lens. In order to solve the wire frame identification problem, we must determine a set of camera viewing parameters ($X1$, $Y1$, $Z1$, θ, ϕ, ψ) which when applied to the identified wire frame object produces the observed image.

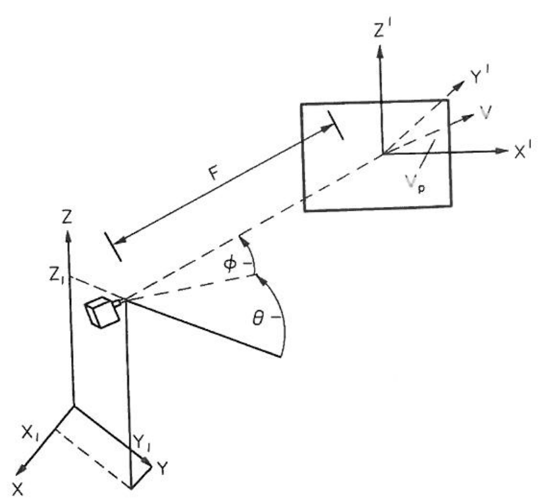

Fig. 1. The geometric relation between the camera and reference frame. θ is the rotation around the Z axis. ϕ is the rotation around the new X^1 axis. ψ is the rotation around the new Y^1-axis and is set to zero here. (X_1, Y_1, Z_1) is the camera lens position, (X^1, Y^1, Z^1) the rotated axes and F the position of the image phase in front of the lens.

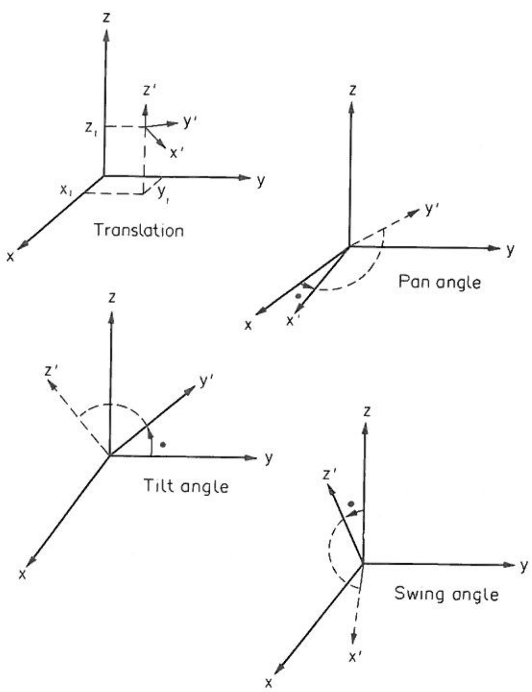

Fig. 2. The correspondence between the camera-oriented coordinate system and the object-oriented coordinate system. The pan angle is called θ, the tilt angle is called ϕ and the swing angle is called ψ.

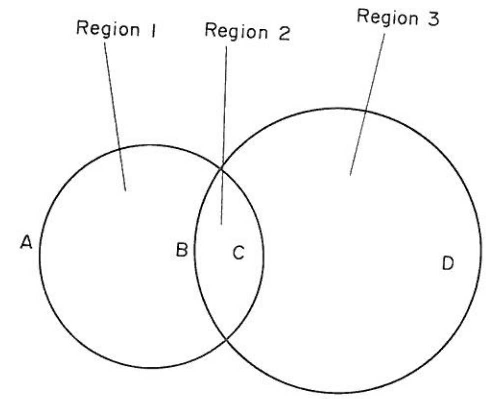

Fig. 3. Two curves crossing each other forming 4 arcs, A, B, C and D, and three regions.

The shape matching procedure discussed in the next section uses numerical nonlinear optimization to find the solution viewing parameter vector ($X1$, $Y1$, $Z1$, θ, ϕ, ψ) of the digitized image. The time complexity of this approach depends on the number of curve segments coming from all the objects in the given set and the time needed to perform the optimization for the arcs in the image. Details can be found in Section 2.

The optimization procedures and precise formulation of the problem are discussed in Section 3. Experimental results are given in Section 3.4. Some geometric facts, like a straight line is mapped to a straight line under perspective projection, are discussed and proved in the appendix.

2. THE MATCHING PROCEDURE

Each object consists of a set of three-dimensional arc primitives, each of which is a sequence of conic (ellipse, parabola, hyperbola) segments and/or line segments lying in the same plane. Different primitives may lie on different planes. The primitives are represented as a set of equations, for example $x^2 + y^2 = 1$ and $z = 1$ represents a circle in the three-dimensional world using a pre-chosen object-oriented coordinate system. Similarly, the intersection of four half planes: $0 \leq x \leq 2$ and $0 \leq z \leq 2$, with y coordinates equal to 1 represents a square of area 4.

The two-dimensional image consists of a perspective projection of one of the wire frame objects from a given set. Given such a perspective projection, we need to determine: (1) on what plane in the three-dimensional world do each of the arcs in the image lie; (2) what functional form each of the three-dimensional arcs has; (3) what are the values of the camera viewing parameters. We assume that the actual distance between the center of the lens and the image plane is known because the focus is fixed and we also assume that the size of the digitized image is known.

The problem can be classified as a two-dimensional to three-dimensional planar curve matching problem with uncertainty on the viewing parameters and the identification of the wire frame object. Since the main concern here is to present the matching techniques, we will assume that we have a perfectly segmented picture consisting of several different simple arcs, each arc being represented as a string of connected pixels. Without loss of generality, we take the center of the digitized image to have coordinates (0, 0) and let half of the width of the image be a unit length in the Z-axis and half of the length of the image be a unit length in the X-axis. The coordinates of each pixel on the arc can then be determined on this basis.

2.1. *Three-dimensional cones from two-dimensional projections*

To construct the three-dimensional cones, as illustrated in Fig. 4, the image arcs have to be identified. The problem of what to do when distinct arcs touch must be solved.

To solve this problem, we locate the intersections of crossing arcs and do a linear or quadratic fit on each piece (maximal sequence of consecutive boundary pixels not crossing an intersection). We then merge some of the pieces on the basis of the similarity of the fitting coefficients. For example, suppose the three-dimensional world consists of 2 circles, one in front of the other. The perspective projection image looks like Fig. 3. Each region is bounded by a piecewise smooth arc. Each bounding arc is split into new smaller arcs at the intersections.

We then compute the equation of each arc in the

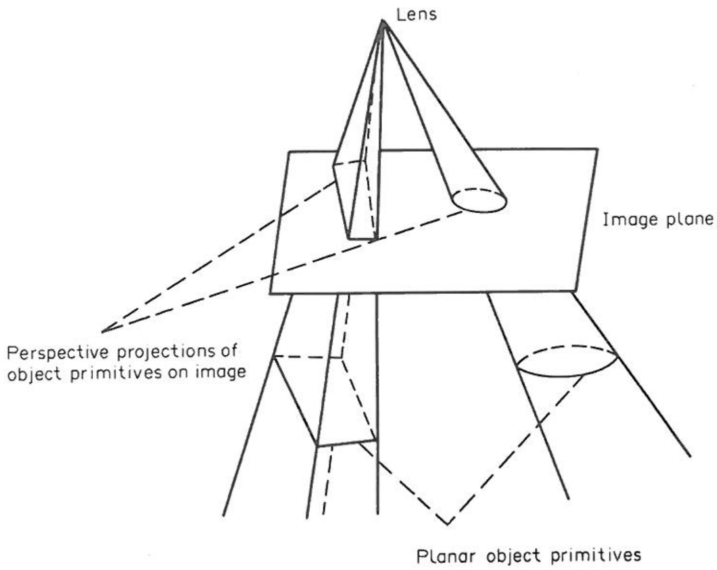

Fig. 4. How to construct cones out of the regions on the image.

image. For example, an observed arc will be initially represented as a sequence of n points stored in two arrays which contain its image (x, z) coordinates. Then we perform a quadratic or linear least squares fit of these points $\{x^i, z^i\}$. This is done by using a singular value decomposition to determine an X which minimizes $\|AX\|^2$, subject to the constraint $\|X\| = 1$, where the matrix A depends on the observed data points and X is the vector of unknown coefficients for the conic. The solution X is the right singular vector of A corresponding to the smallest singular value. If the conic is a circle with equation $ax^2 + bxz + cz^2 + dx + ez + f = 0$ then $X = (a, b, c, d, e, f)$,

$$A = \begin{vmatrix} (x^1)^2, & x^1z^1, & (z^1)^2, & x^1, & z^1, 1 \\ (x^2)^2, & x^2z^2, & (z^2)^2, & x^2, & z^2, 1 \\ \vdots & \vdots & & \vdots & \\ (x^n)^2, & x^nz^n, & (z^n)^2, & x^n, & z^n, 1 \end{vmatrix}$$

where (x^i, z^i) are the n points on the curve. The above procedure can be set up for general conics, the circle being a special conic.

After a least-squares fit on the segments between break points on the example of Fig. 3, there are 4 pieces A, B, C, D. By comparing their coefficients, we merge A with C, and then B with D.

Having identified the arcs on the image, we can construct their associated cones. The set of all possible three-dimensional arcs which have the same perspective projection on the image as a given image arc form a cone in three-dimensional space. The next step is to construct cones and compute their equations. The cones are constructed by connecting each point of the image arc with the apex (center of lens). The equations of these cones are quite easy to compute. We only have to replace the x, z coordinates in each image arc equation by (Fx/y), (Fz/y) where F is the distance between the apex (the lens) and the image plane. For example, if $x^2 + z^2 = 1$ and $F = 1$ is the equation for the image arc, the cone equation is $(x/y)^2 + (z/y)^2 = 1$ or, equivalently, $x^2 + z^2 = y^2$. The derivation of the equation of the cone can be found in Sommerville[(2)] (also discussed in the Appendix). The relation between these cones and the image in a camera-oriented coordinate system is shown in Fig. 4.

When a curve on the image is composed of a sequence of distinct arc primitives, the cone associated with the curve can be expressed as the union of equations of cones spanned by the apex and each distinct piece of the arc. For example, a triangular cone will be expressed as the union of cones from its three plane segments. Suppose the apex is P and the vertices of the triangle are A, B, C. From analytic geometry, we know each point between PA and PB can be expressed as a linear combination of the two vectors $\langle P, A\rangle$ and $\langle P, B\rangle$, where $\langle P, A\rangle$ designates the vector with origin P and end point A. Hence, the planar cone between PA and PB is expressed as $r1\langle P, A\rangle + r2\langle P, B\rangle$ with $r1$ and $r2$ nonnegative. The cone then consists of all points (x, y, z) satisfying $(x, y, z) \in \{r1\langle P, A\rangle + r2\langle P, B\rangle\} \cup \{r3\langle P, A\rangle + r4\langle P, C\rangle\} \cup \{r5\langle P, B\rangle + r6\langle P, C\rangle\}$ for some $r1, \ldots, r6$ which are nonnegative real numbers.

2.2. *Matching the image arcs to a three-dimensional arc*

Now, suppose we have the equations of the cones. We know that for each of these cones we must associate one arc from some object in the set of possible three-dimensional world objects. The associated arc must lie entirely in the cone. Otherwise, we would have an impossible image. Hence, the remaining problem is how to select arcs from the possible three-dimensional world objects, those arcs being repre-

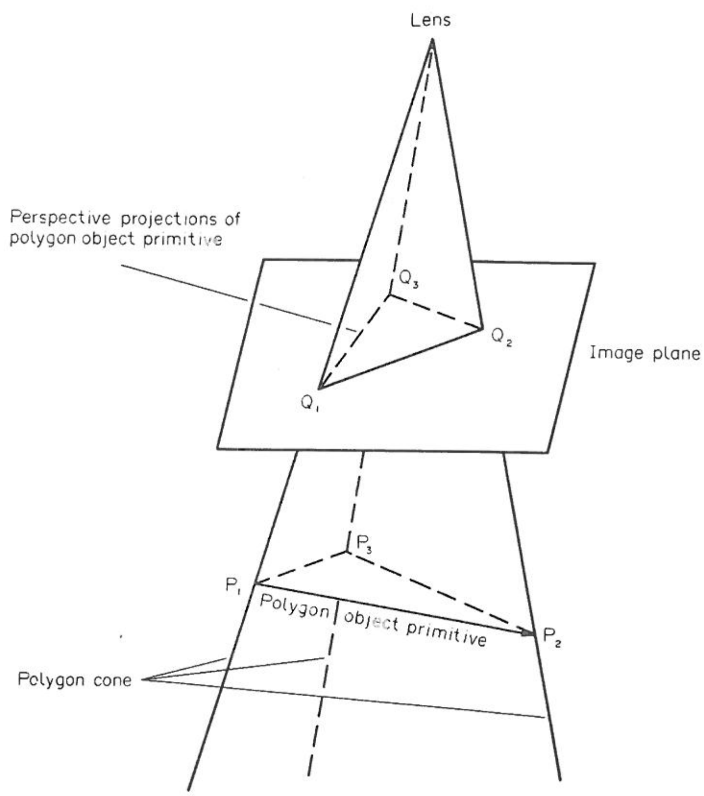

Fig. 5. Polygon cone point correspondences: ($Q1$, $P1$), ($Q2$, $P2$), ($Q3$, $P3$).

sented in the object-oriented coordinate system, and fit them into cones, which are represented in the camera-oriented coordinate system.

Matching then reduces to a mathematical nonlinear optimization problem, which is discussed in the next section. The procedure tries to find one wire frame object having a possible candidate arc for each arc in the image. The perspective projection of these candidates is exactly the picture we are analyzing. If this method fails for all objects in the given set of objects in the world, it implies that the image is of an object not in the set or that the given picture is impossible to obtain from one of the given objects. The picture must have included some perspective projections of some curves which were not part of any wire frame object.

3. THE OPTIMIZATION PROCEDURE

At this point, the object image of the wire frame object has been represented as a collection of arcs from which cones have been constructed. The problem whose solution we describe in this section is: find a wire frame object from the given set so that by an appropriate position and viewing direction of the camera, the wire frame object has one arc for each cone constructed out of the image. Each object arc must lie in the cone and in front of the image plane and project to an observed image arc. As we can see, by constructing these cones, the two-dimensional to three-dimensional curve matching problem is transformed into an equivalent three-dimensional cone to three-dimensional curve matching problem, which we solve as a nonlinear optimization problem. Once a ($X1$, $Y1$, $Z1$, θ, ϕ, ψ) is found which works for an object in the given set, the match is found as well.

There are two possible kinds of nonlinear optimizations. Section 3.1 discusses the case when the optimization is based on point correspondences. This case arises when we can identify three or more vertices on the three-dimensional object and three or more on the image. The point correspondences are not known and the optimization must solve for them. This is also the case that Roberts[3] and others have worked on. Watson and Ehrich[4] and Cohen[5] have applied more advanced minimization techniques to solve this problem. Section 3.2 discusses the case when there are no exact point correspondences; rather, what we hypothesize is that some three-dimensional object arcs are mapped to some arcs in the image. For example, we hypothesize that an ellipse on the object has been mapped to an ellipse on the image and part of a parabola on the object has been mapped to part of a hyperbola on the image.

3.1. *Solving the vertex correspondence problem*

First suppose we have a polygon cone with 3 edges. After such a cone is constructed, we have to determine the point correspondences, like those illustrated in,

Fig. 5. Once we have a correspondence, we can determine the camera viewing parameters consistent with the correspondence.

Let $T = (T^{ij})$, $1 \le i \le 3$, $1 \le j \le 4$, be the perspective projection matrix in a homogeneous coordinate system. The entries of the matrix depend on the 6 camera viewing parameters. Let P^i, $i = 1, 2, \ldots, M$ be the vertices of the object polygon expressed in homogeneous coordinates and Q^i, $i = 1, \ldots, M$ be the vertices of the image polygon, also expressed in homogeneous coordinates. The relationship between P^i and Q^i is given by the perspective transformation T; $T: P^i \Rightarrow Q^i$ where $P^i = (x, y, z, 1)^t$ and $Q^i = (x^*, z^*, t^*)^t$ and where the transpose of a vector V is denoted by V^t. The problem is: determine a permutation $J_1, J_2, \ldots, J_M$ of $1, 2, 3, \ldots, M$ which satisfies the vertices connection constraint (which prohibits certain types of matching because they are not consistent with the geometry, an example of which is shown later) and determine the camera viewing parameters $\langle X1, Y1, Z1, \theta, \phi, \psi \rangle$ which minimize

$$\sum \| TP^i - Q^{J^i} \|_2$$

over all perspective projection matrices T and permutations $\{J_1, J_2, \ldots, J_M\}$ of $\{1, 2, \ldots, M\}$. $\| \|_2$ is the Euclidean norm of a vector.

When we have more than three point correspondences, the system becomes an overly determined system. It is thus possible to use a least squares method to find the solution which minimizes the residual sum. Unfortunately, the least squares problem to solve for the camera parameters is a highly nonlinear problem due to the trigonometric dependencies of the entries on the viewing parameters. We handle this problem by solving for each entry of T independently and without any reference to the trigonometric dependence by setting up an overconstrained linear system whose solution is simply obtained by singular value decomposition.

$$\begin{vmatrix} T^{11} & T^{12} & T^{13} & T^{14} \\ T^{31} & T^{32} & T^{33} & T^{34} \\ T^{21} & T^{22} & T^{23} & T^{24} \\ 0 & 0 & 0 & 1 \end{vmatrix} * \begin{vmatrix} x^i \\ y^i \\ z^i \\ 1 \end{vmatrix} = \begin{vmatrix} x'^i \\ y'^i \\ z'^i \\ 1 \end{vmatrix}.$$

To solve for the entries of the first row

$$\begin{vmatrix} x^1 & y^1 & z^1 & 1 \\ x^2 & y^2 & z^2 & 1 \\ x^3 & y^3 & z^3 & 1 \\ x^4 & y^4 & z^4 & 1 \\ . & . & . & 1 \\ . & . & . & 1 \\ x^n & y^n & z^n & 1 \end{vmatrix} * \begin{vmatrix} T^{11} \\ T^{12} \\ T^{13} \\ T^{14} \end{vmatrix} = \begin{vmatrix} x'^1 \\ x'^2 \\ x'^3 \\ x'^4 \\ . \\ . \\ x'^n \end{vmatrix}.$$

A similar result can be derived for $(T^{21}, T^{22}, T^{23}, T^{24})$ and $(T^{31}, T^{32}, T^{33}, T^{34})$.

Having a value for each entry of T, we then determine if there exist viewing parameters $(X1, Y1, Z1, \theta, \phi, \psi)$ which produce, via the trigonometric dependencies, the values we already have for the entries of T and which, therefore, make T a perspective transformation. This latter part of translating the matrix entries of T to the viewing parameters is not trivial.

To find viewing parameters $\langle X1, Y1, Z1, \theta, \phi, \psi \rangle$ which minimize

$$\sum \| f^{ij} - T^{ij} \|_2$$

where $1 \le i \le 3$ and $1 \le j \le 4$ and

$$f^{11} = \cos\psi \cos\theta - \sin\psi \sin\phi \sin\theta,$$
$$f^{12} = \cos\psi \sin\theta + \sin\psi \sin\phi \cos\theta,$$
$$f^{13} = -\sin\psi \cos\phi,$$
$$f^{21} = \sin\psi \cos\theta + \cos\psi \sin\phi \sin\theta,$$
$$f^{22} = \sin\psi \sin\theta - \cos\psi \sin\phi \cos\theta,$$
$$f^{23} = \cos\psi \cos\phi,$$
$$f^{31} = -\cos\phi \sin\theta,$$
$$f^{32} = \cos\phi \cos\theta,$$
$$f^{33} = \sin\phi,$$
$$f^{14} = -X1 f^{11} - Y1 f^{12} - Z1 f^{13},$$
$$f^{24} = -X1 f^{21} - Y1 f^{22} - Z1 f^{23},$$
$$f^{34} = -X1 f^{31} - Y1 f^{32} - Z1 f^{33},$$

we use the program 'LMDIF1' in MINPACK, a package developed by the Argonne National Laboratory.[6] The algorithm employed by this program is a version of the Levenberg–Marquardt algorithm.[7] The Levenberg–Marquardt algorithm is a combination of the method of steepest descent (gradient search) and the classical Gauss–Newton method for nonlinear least squares problems. It is essentially steepest descent when the initial guess is far from the minimum point. Thus its global behavior is good and its ultimate convergence rate to the minimum point is also good. The method uses only first derivative information, yet typically has superlinear convergence.

3.2. *Solving the arc correspondence problem*

In this optimization problem there is no exact point correspondence. We only know one arc is mapped to another arc. The actual point correspondence for these curves cannot be found until the transformation T is obtained. This is also the most important case, so we shall elaborate on it in detail.

Suppose we have n arcs $E^1, E^2, \ldots, E^n$ in the image after segmentation and merging. Let us also assume that there exist m three-dimensional arcs $E''^1, E''^2, \ldots, E''^m$ in a wire frame object. Assuming $n \le m$, our aim is to find the perspective transformation T and the n curves $E'^1, E'^2, \ldots, E'^n$ from $E''^1, E''^2, \ldots, E''^m$ of the wire frame object that minimizes the sum

$$\sum_{i=1}^{n} \| TE'^i - E^i \|.$$

The norm indicates the integral of the squared error function over the range of the three-dimensional curve.

We can see that a brute force approach to this problem is of complexity $m!/(m - n)!$ optimization time to find T. The actual complexity will be lower because certain geometric constraints, like the face connection topology, exist. The optimization time to find a T is proportional to $(V + 1)(V^3 + PV^2)$, where V is the number of degrees of freedom (i.e. the 6 camera parameters) and P is the number of points used to represent the three-dimensional curves. Details can be found in More.[6]

A better way to solve this problem is to find a T which minimizes $\| TE'^1 - E^1 \| + \| TE'^2 - E^2 \|$ and then use this T to determine whether the sum

$$\sum_{i=3}^{M} \| TE'^i - E^i \|$$

is small enough.

The reason we choose two arcs here is to make the solution unique. From Rogers and Adams,[8] we know six non-coplanar point correspondences determine a unique camera viewing parameter solution; by choosing two non-coplanar arcs we can have six or more such point correspondences.

The second part is done by applying T to E'^i $(i = 3, \ldots, m)$ to obtain a 'template'. Then, we match this template against the image to compute the error. If the match reports a small error, we report success. Otherwise, we will try the next pair of three-dimensional arcs. If all the three-dimensional arcs have been tried and no acceptable solution is found, we reject this object and repeat the whole procedure for a different three-dimensional arc pair of a different object.

The complexity of this algorithm is $m(m - 1)*$ [optimization time + computation time required to compute the product of T and points on curve i $(i = 3, \ldots, m)$ and check the consistency].

Now, let us examine how we compute $\| TE'^1 - E^1 \| + \| TE'^2 - E^2 \|$ even if we have no point correspondences. We know that the 3-D planar arc E'^1 can be represented by $f_1(x, y, z) = 0$ and $h_1 x + m_1 y + n_1 z + k_1 = 0$. Its 2-D perspective projection E^1 satisfies $g_1(x, F, z) = 0$, where F is a known constant. We define $T = (T^{1t}, T^{2t}, T^{3t})^t$, where each T^i is a row vector. If E'^1 is mapped to E^1, then we have $g_1(T^1P/T^3P, F, T^2P/T^3P) = 0$, where $P = (x, y, z, 1)^t$ is a point vector in the homogeneous coordinate system and P lies on E'^1, i.e. $f_1(x, y, z) = 0$.

Suppose 3-D planar arc E'^2 is represented by $f_2(x, y, z) = 0$ and $h_2 x + m_2 y + n_2 z + k_2 = 0$ and its perspective projection satisfies $g_2(x, F, z) = 0$.

We wish to determine a perspective transformation T to minimize

$$\int_{P \in E'^1} g_1(T^1P/T^3P, F, T^2P/T^3P)^2 + \int_{P \in E'^2} g_2(T^1P/T^3P, F, T^2P/T^3P)^2.$$

In other words, we want to compute the error sum over the whole three-dimensional arc if T is given.

In the polygon case, since each line on a plane can be represented by $ax + bz + c = 0$, we can replace each polygon by a product of $a(i)x + b(i)z + c(i) = 0$. So, we have

$$g(x, F, z) = \prod_i (a(i)x + b(i)z + c(i)) = 0.$$

Then we can apply the above method to the polygon cones, even if we do not know the specific order of the point correspondences.

This product approach can be used to derive a better initial guess when the guess is far off. It suffers the drawback that it tends to converge to some local minimum points. Hence, we use the arc to arc correspondence to set up the matching problem and only use this product technique to derive a better initial guess.

If we want to include the camera focal length F as an unknown, then we have to pick another arc which does not lie on the planes that contain E'^1 or E'^2. This is to make sure that we can determine all seven degrees of freedom. Focal distance is related to the scale factor of objects (or pictures).

3.3. *Solution algorithm summary*

1. Choose an object.
2. On the image, choose a pair of arcs E^i and E^j.
3. Go through all pairs of object arcs Cm, Cn. For each Cm, Cn find the six viewing parameters (x, y, z, a, b, c) which minimize the sum

$$\| TCm - E^i \| + \| TCn - E^j \|$$

(x, y, z, a, b, c).
The norm is computed by integrating over the ranges of Cm and Cn and the equations of E^i and E^j. (The details are described in Section 3.2.) The angle a is the pan angle θ, b is the tilt angle ϕ, c is the swing angle ψ and $(x, y, z) = (X1, Y1, Z1)$ are the coordinates of the lens.
4. Each arc correspondence chosen in step 2 induces a nonlinear constraint mapping on (x, y, z, a, b, c). The equations are exactly those we have described in Section 3.2. Input these equations to the MINPACK routine 'LMDIF1' and supply an initial guess of (x, y, z, a, b, c). If the guess is not too far off, the global minimum can be obtained. If we specify more constraints on (x, y, z, a, b, c), it will speed up the optimization. The minimum number of constraints to choose is the number of degrees of freedom of the nonlinear system.
5. Compute T from the solution (x, y, z, a, b, c). Apply T to all remaining object arcs Cu. Compare this computed image with the original image, by placing one 'template' over another. Determine the error sum of $\| TCu - E^i \|$. If it is less than a pre-chosen threshold, we report success and the curves on the image can be identified as having come from the corresponding object arcs.

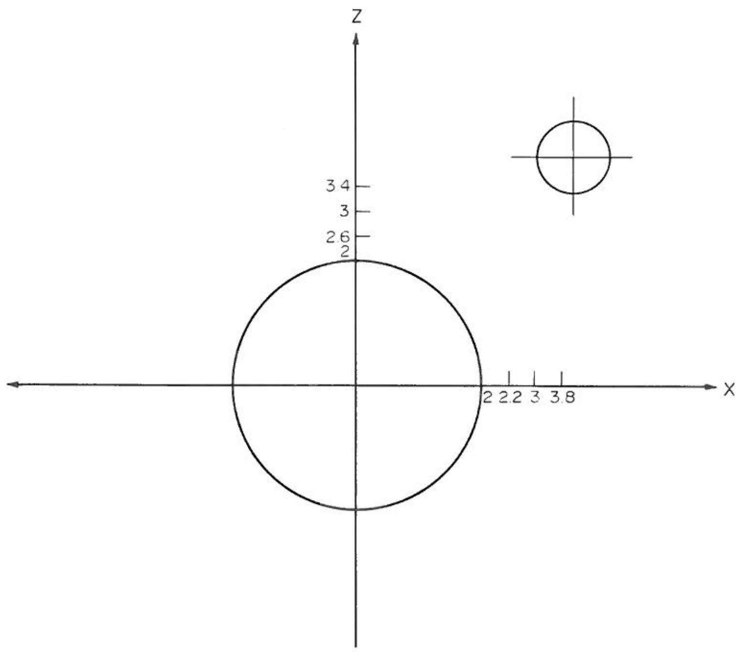

Fig. 6. Two circles on the plane $Y = 4$ used in the first test.

6. If not, change Cm and Cn and loop to step 3.
7. If we run out of curves from the wire frame object, report no possible solution with this object and try the next object.

3.4. *Results*

All experiments used a camera with a focal length $F = 1$. In our first experiment, the input image consists of the perspective projections of two circles on a plane with the following parameters $(X1, Y1, Z1, \theta, \phi, \psi) = (0, 0, 0, 0, 0, 0)$ (refer to Fig. 6 for details). We first compute the equations of these two circles on the image then follow the process outlined in Section 3.3. The program finds the solution with an initial guess of

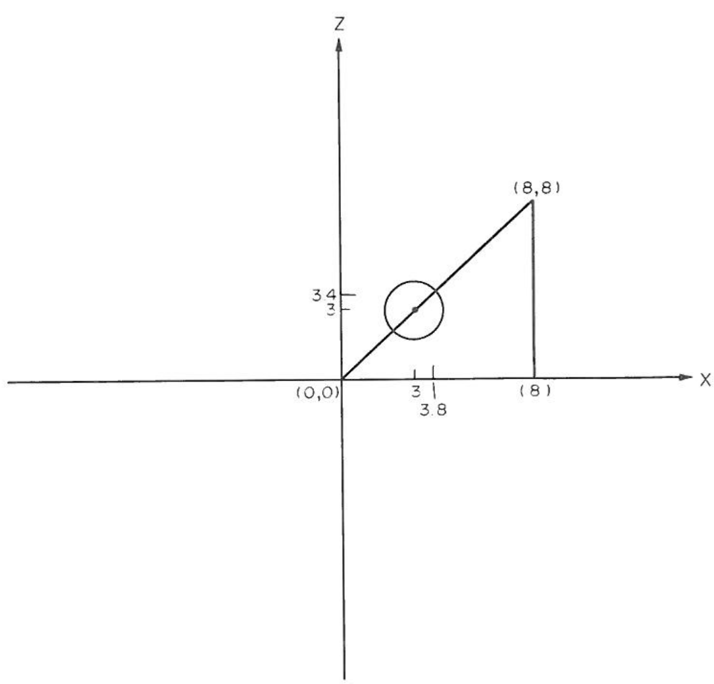

Fig. 7. Triangle and circle on the plane $Y = 4$ used in the second test.

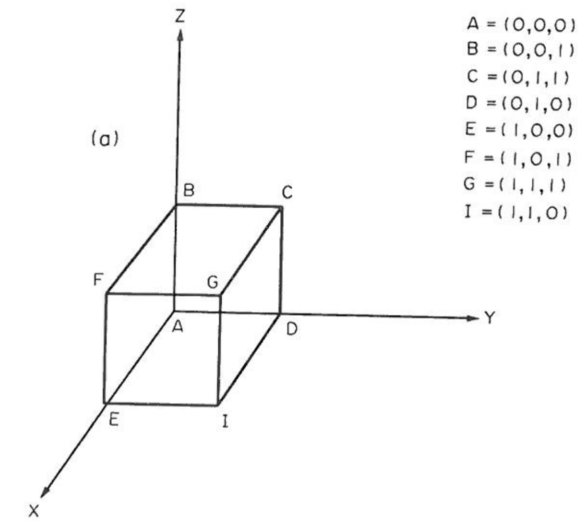

Fig. 8a. A wire frame cube used in the third experiment.

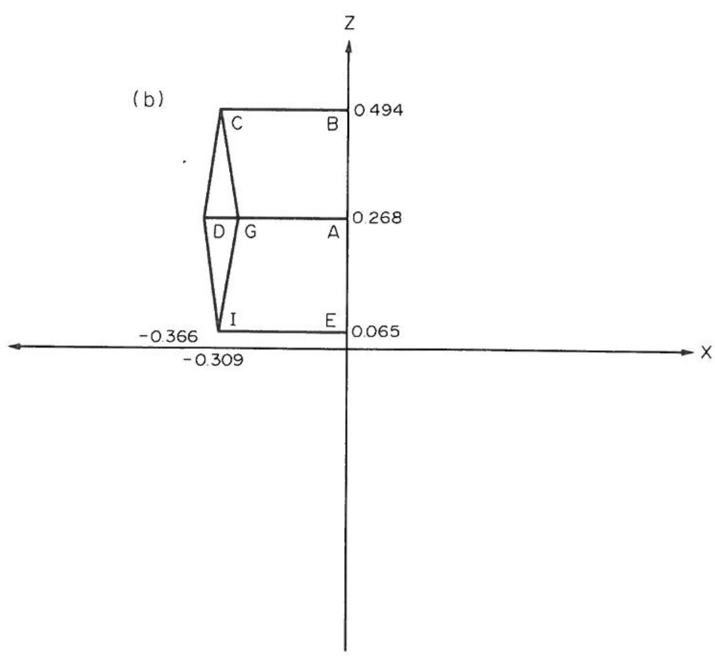

Fig. 8b. Perspective projection of wire frame cube from Fig. 8a.

(1, 1, 1, 1, 1, 1). (Remember that angles are measured in radians.)

In our second experiment, we change one ellipse in the above test to a triangle (Fig. 7). The triangle is expressed as a product of the equations of its 3 boundaries. The convergence behavior is about the same. The point we want to emphasize is that this approach uses a product of equations to get around the combinatorial problem involved in the vertex-to-vertex matching approach that has been used by Roberts.[3] Another advantage of this approach is that it will tolerate the noise introduced in the segmentation procedure. This approach can be used even if we are unable to find the vertices (corners) in a two-dimensional image.

In our next experiment, the object is the wire frame cube of Fig. 8a. The cube has vertices (0, 0, 0), (0, 0, 1), (0, 1, 1), (0, 1, 0), (1, 0, 0), (1, 0, 1), (1, 1, 1) and (1, 1, 0). The camera sits at $(-2, 0, -2)$ and the viewing axis is (cos (0.5235), 0, sin (0.5235)) (see Fig. 8b). Occlusion is allowed here, hence we only see 9 edges. The 2 arcs we choose to start the process are the boundaries of the

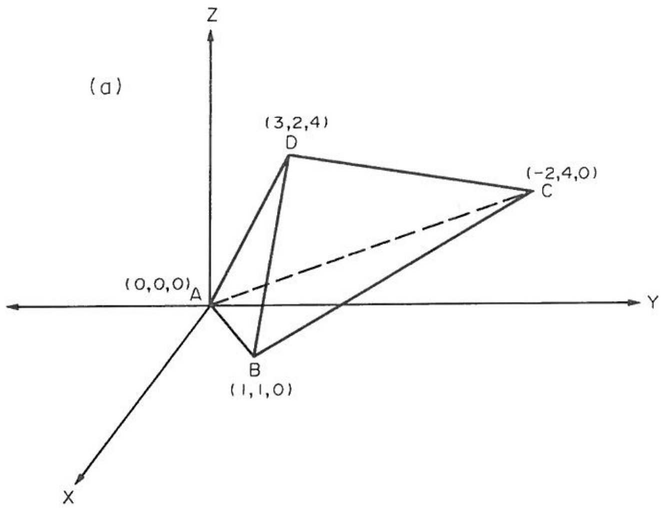

Fig. 9a. A wire framed triangular pyramid used in the last experiment.

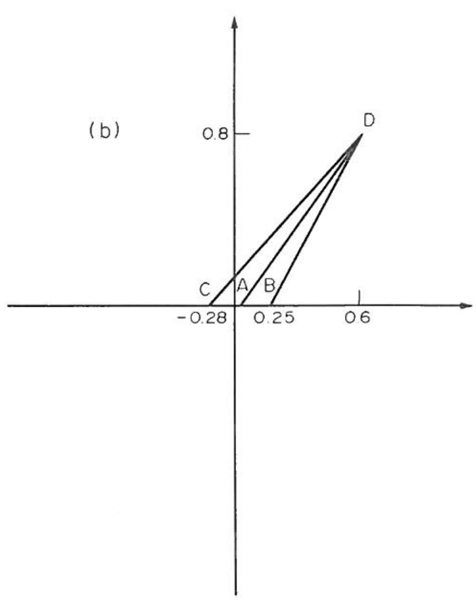

Fig. 9b. Perspective projection of triangular pyramid of Fig. 9a.

two outside larger squares which share a common edge. The viewing solution is (−2, 0, −2, −1.5708, 0.5235, 0). The procedure converges with initial guesses like (−1, 1, −1, −1.57, 0.5235, 0), (−1, 0.2, −1, −1.4, 0.5, 0.01) and (−1.5, 0.1, −1.7, −1.4, 0.5, 0). This test shows that we can extend the matching technique to identify objects.

Now, suppose we do not hypothesize the right arc correspondences to start the matching procedure. For example, if we switch the two faces (arcs) used in the above experiment and start with a 'close' initial guess, like (−2, 0, −2, −1.57, 0.5235, 0), the procedure converges to a local minimum point (0, 0, *, −1.57, 1.113, 0) which projects the cube to a quadrilateral. This local minimum has a small error because one face of the cube has been mapped to the line $X = 0$. We then change our three-dimensional candidate to a pyramid and choose two triangular faces of it to start the matching and the procedure converges to infinity, i.e. we have ($X1$, $Y1$, $Z1$) becoming larger and larger, which means we have the camera moving further away from the two faces.

The last experiment is a projection of a triangular pyramid which has two triangles in the image, described in Figs. 9a and 9b. The viewing parameters are (0, −4, 0, 0, 0, 0) and the vertices of the pyramid are (0, 0, 0), (3, 2, 4), (1, 1, 0) and (−2, 4, 0). It converges with initial guesses (5, −3, 5, 0, 0, 0), (0.9, −3.05, −0.1, 0.2, 0, 0), (0, −9, 0, 0, 0, 0), (0, −3.5, 0, 0.2, 0.2, 0.2) and (0, −0.3, 0, 0, 0, 0). We then switch the two faces or change the two faces (triangles) to two squares and the procedure either converges to infinity or converges to a local minimum point with rather large error norm.

Finally, we point out that this procedure uses an edge-to-edge matching approach instead of a point-to-point approach to find the camera position and matching error. The first two experiments show that the matching works for conics. The last two experiments show that we can extend this technique to identify objects in the real world if we have an acceptable model of the objects.

The optimizing procedure is iterative. Like most iterative methods, if we have a bad initial guess, the method will in general diverge or converge to a local minimum point.

4. CONCLUSION

The construction of three-dimensional cones from a two-dimensional perspective projection really is a transformation which transforms the two-dimensional to three-dimensional matching problem into a three-dimensional to three-dimensional matching problem. We showed how to find proper curves (faces of wire frame models) which lie on some cones, then put all these correspondences together to compute the viewing parameter vector $(X1, Y1, Z1, \theta, \phi, \psi)$ by using nonlinear optimization.

It does turn out that polygon cones are easier to solve than conics, because straight lines are first degree equations, while conics are second degree equations. By lemma 4 of the Appendix, it seems proper to say that we can extend the three-dimensional world to N-ics (equations of N-th degree planar curves). Further work is needed.

We also point out that one of our approaches uses a product of equations of the edges of the face of an object to get around the combinatorial problem in the vertex-to-vertex matching approach used by Roberts.[3] Another advantage of doing so is that it will tolerate the noise introduced in the segmentation procedure. This approach can be used even if we are unable to find the vertices (corners) in the image.

Our recent work has concentrated on applying this proposed technique to a planar surfaced model. Each object in the model is a solid consisting of planar faces. We will allow more than three faces to meet at one vertex. Curved surfaced objects may also be allowed, but the difficulties involved in processing hidden lines are considerable.

REFERENCES

1. R. M. Haralick, Y. H. Chu, L. T. Watson and L. G. Shapiro, Identification of wire frame objects from their two dimensional perspective projections, *IEEE Pattern Recognition and Image Processing Conference*, pp. 572–579 (1982).
2. D. M. Y. Sommerville, *Analytic Geometry of Three Dimensions*, p. 94. Cambridge University Press (1959).
3. L. G. Roberts, Machine perception of three-dimensional solids, *Optical and Electro-Optical Information Processing*, J. T. Tippet, ed, pp. 159–167. M.I.T. Press, Cambridge, MA (1965).
4. L. T. Watson and R. W. Ehrich, private communication.
5. N. Cohen, Vertices correspondence between model object and image, M.S. Thesis, Department of Computer Science, Virginia Polytechnic Institute and State University (1979).
6. J. J. More, *MINPACK Documentation*. Argonne National Laboratory, Argonne, IL (1979).
7. K. M. Brown and J. E. Dennis, Derivative free analogues of the Levenberg–Marquardt and Gauss algorithms for nonlinear least square approximations, *Numer. Math.* **18**, 289–297 (1972).
8. D. T. Rogers and J. A. Adams, *Mathematical Elements for Computer Graphics*, p. 80. McGraw-Hill, New York (1976).
9. J. A. Hummel, *Vector Geometry*. Addison-Wesley, Reading, MA (1965).
10. D. Hilbert, *Geometry and Imagination*, p. 9. Chelsea, New York (1952).
11. A. D. Campbell, *Advanced Analytic Geometry*, p. 121. John Wiley and Sons, New York (1938).

APPENDIX

Definition of a cone

From Sommerville's *Analytic Geometry of Three Dimensions*,[2] we know that a general equation of a cone of the second order is

$$f(x, y, z) = ax^2 + by^2 + cz^2 + 2fyz + 2gxz + 2hxy = 0.$$

If we had a cone with vertex (x_0, y_0, z_0) and knew that its projection on the $z = 0$ plane is

$$g(x, y) = ax^2 + by^2 + 2hxy + 2gx + 2fy + c = 0,$$

then we have the equation of the cone

$$F(x', y', w') = ax'^2 + by'^2 + 2hx'y' + 2gx'w' + 2fy'w' + cw'^2 = 0,$$

where $x' = z_0x - x_0z$, $y' = z_0y - y_0z$, $z' = z_0w - w_0z$ and w_0 is a constant which makes (x_0, y_0, z_0, w_0) a homogeneous coordinate of (x_0, y_0, z_0), or in nonhomogeneous coordinates,

$$(z_0)^2g(x, y) - z_0z(xD(F)/(Dx_0) + yD(F)/(Dy_0) + D(F)/(Dw_0)) + z^2g(x_0, y_0) = 0.$$

The projecting cone of a planar curve can be defined as the set of all lines which connects (x_0, y_0, z_0), the camera lens coordinates, and (x, y, z), which is any point in the planar curve.

From theorem 7-3 of Hummel's *Vector Geometry*,[9] the discriminant of a quadratic equation $f(x, y) = ax^2 + bxy + cy^2 + dx + ey + f = 0$ is DISC $= b^2 - 4ac$. If DISC is less than zero then the function is an ellipse. If DISC is equal to zero then the function is a parabola. If DISC is greater than zero then the function is a hyperbola, excluding those degenerate cases.

Lemma 1. Ellipses are mapped into ellipses.

Proof. An ellipse (including a circle) can, by perspective projection, become an ellipse or parabola or hyperbola and vice versa. These facts can be found in Hilbert's *Geometry and Imagination*.[10] Basically, the proofs given there are geometrical; we shall present an algebraic proof in lemma 3.

Under perspective transformation (i.e. projection with a fixed center), in general, a closed curve will project to a closed curve, an open curve to an open curve. We will consider some degenerate cases later. Note here that the image plane of the camera is always perpendicular to the optical axis of the lens.

If a plane cuts through a projection cone, its intersection will be an ellipse or a parabola or a hyperbola depending on the angle α between the plane and the axis of the projecting cone.

If it is greater than the angle of the projecting cone, it is an ellipse. If they are equal, we have a parabola. If there is no intersection with the axis or the angle is smaller, we have a hyperbola. The angle of the cone is defined as the angle between the axis and any generating line of the cone. We assume that we have a quadratic cone here.

In our application, we know $\alpha = 90°$ and β is always less than 90°. So, when we have an ellipse in the lower cone (it is equivalent to say it comes from the boundary of some objects), we are guaranteed we will see an ellipse in the photo if the object is not occluded by others.

The following is the proof. Use the lower ellipse to construct a cone, then let the image plane cut through the cone. By definition, we will have an ellipse as the intersection, because α is greater than β.

Properties of perspective transformations

Most functions we treat in this paper are planar functions. Recall that a two dimensional function is expressed as $f(x, y) = 0$ and $ax + by + cz + d = 0$. The perspective transformation T is the product of rotation and translation and perspective projection matrix. For the derivation see Rogers and Adams.[8]

$$T = \begin{vmatrix} T^{11}, & T^{12} & T^{13}, & T^{14} \\ T^{21}, & T^{22}, & T^{23}, & T^{24} \\ T^{31}, & T^{32}, & T^{33}, & T^{34} \end{vmatrix}$$

$$X = \begin{vmatrix} x \\ y \\ z \\ 1 \end{vmatrix}$$

$$B = \begin{vmatrix} x^* \\ z^* \\ t^* \end{vmatrix}$$

$$TX = B$$

where X is the three-dimensional coordinates of a point and B is the 'homogeneous' perspective projection coordinates. If the coefficient c does not equal 0 in the plane equation, $z = -1/c(d + ax + by)$.

So,

$$x^* = T^{11}x + T^{12}y + T^{13}(-1)/c(d + ax + by) + T^{14}$$
$$= (T^{11} + (-a)/cT^{13})x + (T^{12} + (-b)/cT^{13})y + (T^{14} + (-d)/cT^{13})$$
$$= a1x + b1y + c1$$
$$z^* = T^{21}x + T^{22}y + T^{23}(-1)/c(d + ax + by) + T^{24}$$
$$= (T^{11} + (-a)/cT^{23})x + (T^{22} + (-b)/cT^{23})y + (T^{24} + (-d)/cT^{23})$$
$$= a2x + b2y + c2$$
$$t^* = (T^{31} + (-a)/cT^{33})x + (T^{32} + (-b)/cT^{33})y + (T^{34} + (-d)/cT^{33})$$
$$= a3x + b3y + c3$$
$$s = x^*/t^* = (a1x + b1y + c1)/(a3x + b3y + c3)$$
$$t = z^*/t^* = (a2x + b2y + c2)/(a3x + b3y + c3)$$

(s, t) is the coordinate of $(x, y, z, 1)$ in the image plane. Which is exactly a bilinear (or linear fractional) transformation defined in Campbell's *Advanced Analytic Geometry*.[11]

Define.

$$G = \begin{vmatrix} a1 & b1 & c1 \\ a2 & b2 & c2 \\ a3 & b3 & c3 \end{vmatrix}$$

If $G \neq 0$, we have

$$x = (A^1s + A^2t + A^3)/(C^1s + C^2t + C^3)$$
$$y = (B^1s + B^2t + B^3)/(C^1s + C^2t + C^3)$$

where A^i, B^j, C^k $(i, j, k, = 1, 2, 3)$ are the cofactors of ai, bj, ck respectively in G.

So, the following results can be applied from a two-dimensional world to a three-dimensional world and vice-versa.

Lemma 2. Straight lines are mapped to straight lines.

Proof. The curve $f(x, y) = 0$ is $ax + by + c = 0$. Substituting x and y by equations derived above, we get

$$a(A^1s + A^2t + A^3) + b(B^1s + B^2t + B^3) + c(C^1s + C^2t + C^3) = 0.$$

Rearranging the terms, we get

$$(aA^1 + bB^1 + cC^1)s + (aA^2 + bB^2 + cC^2)t + (aA^3 + bB^3 + cC^3) = 0,$$

which is still an equation of a straight line. The degenerate case is when we have a line mapped into a point if the line is parallel to the viewing axis.

Lemma 3. Conics are mapped into conics.

Proof. The equation of the conic section is

$$f(x, y) = ax^2 + by^2 + c + 2fy + 2gx + 2hxy = 0.$$

Substituting x, y in terms of s and t and simplify it, we get

$$a(A^1s + A^2t + A^3)^2 + b(B^1s + B^2t + B^3)^2$$
$$+ c(C^1s + C^2t + C^3)^2 + 2g(A^1s + A^2t + A^3)$$
$$+ (C^1s + C^2t + C^3) + 2f(B^1s + B^2t + B^3)$$
$$+ (C^1s + C^2t + C^3) + 2h(A^1s + A^2t + A^3)$$
$$+ (B^1s + B^2t + B^3) = 0.$$

Grouping terms with same variables together, we get

$$s^2(a(A^1)^2 + b(B^1)^2 + c(C^1)^2 + 2gA^1C^1 + 2fB^1C^1$$
$$+ 2hA^1B^1) + t^2(a(A^2)^2 + b(B^2)^2 + c(C^2)^2 + 2gA^2C^2$$
$$+ 2fB^2C^2 + 2hA^2B^2)' + 1(a(A^3)^2 + b(B^3)^2 + c(C^3)^2$$
$$+ 2gA^3C^3 + 2fB^3C^3 + 2hA^3B^3) + t(a(2A^2A^3) + b(2B^2B^3)$$
$$+ c(2C^2C^3) + 2g(A^3C^2 + A^2C^3) + 2f(B^2C^3 + B^3C^2)$$
$$+ 2h(A^2B^3 + A^3B^2)) + s(a(2A^1A^3) + b(2B^1B^3) + c(2C^1C^3)$$
$$+ 2g(A^3C^1 + A^1C^3) + 2f(B^1C^3 + B^3C^1)$$
$$+ 2h(A^1B^3 + A^3C^1)) + st(a(2A^1A^2) + b(2B^1B^2)$$
$$+ c(2C^1C^2) + 2g(A^1C^2 + A^2C^1) + 2f(B^1C^2 + B^2C^1)$$
$$+ 2h(A^1B^2 + A^2B^1)) = 0$$

which is still a conic.

From Theorem 7-3 of Hummel,[9] we can determine which kind of curve it is by rotating and testing the discriminant.

For equations like

$$AX^2 + BXY + CY^2 + DX + EY + F = 0,$$

we can eliminate XY terms by rotating this curve by θ, where

$$\sin(2\theta) = B/\sqrt{(B^2 + (C - A)^2)}$$

and

$$\cos(2\theta) = (C - A)/\sqrt{(B^2 + (C - A)^2)}.$$

Lemma 4. N-ics are mapped into N-ics.

Proof. N-ics are planar curves which are polynominals of N-th degree in x and y. Following the proof of Lemma 3, we see immediately that N-ics are mapped into N-ics because the order of the polynomial is preserved. The degenerate cases arise when the plane of the curve is parallel to the viewing axis, then the curve is mapped into a straight line.

A N-ics is $f(x, y) = 0$, where

$$f(x, y) = \sum a^{ij}x^i y^i,\ i + j \leq N.$$

Substituting x, y in both sides and multiplying both sides by $(C^1 s + C^2 t + C^3)^N$, we can see the highest term is still to the N-th power. So, we have N-ics mapped to N-ics, in general.

Heuristics

Lemma 5. Suppose we have a quadratic equation $ax^2 + cy^2 + dx + ey + f = 0$, the eccentricity a/c (or c/a) is independent of d, e, f.

Proof. Complete the squares of x and y. We get

$a(x + d/2/a)^2 + c(y + e/2/c) = d^2/(4/a) + e^2/(4/c) - f(x + d/(2a))^2/(k/a) + (y + e/(2c))^2/(k/c) = 1$, where $k = d^2/(4/a) + e^2/(4/c) - f$.
Defining eccentricity as major axis length/minor axis length, we have major/minor $= (k/a)/(k/c) = a/c$ or c/a if the major axis lies on the y axis.
If a or c is 0, which happens when we have a parabola, $y^2 = 4px$ (or $x^2 = 4py$), we can derive similar constraints for p.

Lemma 6. If two conics on a cone lie on planes which are parallel, if the top one has equation $ax + by + cz = d$ and $Ax^2 + Bxz + Cz^2 + Dx + Ez + F = 0$, then, the lower one will have equation $ax + by + cz = d + k$ and $Ax^2 + Bxz + Cz^2 + D'x + E'z + F' = 0$. Since A, B, C in the two equations are the same, they both have the same eccentricity.

Proof. Suppose the equation of the cone is $Ax^2 + Bxz + Cz^2 Dxz + Ezy + Fy^2 = 0$. Parallel planes implies that their equations are $ax + by + cz = d$ and $ax + by + cz = d + k$ or, equivalently, $y = lx + mz + n$ and $y = lx + mz + n + k$. Substituting in y, we get

$$x^2(A + Dl + Fl^2) + xz(B + Dm + El + 2Flm)$$
$$+ z^2(C + Em + Fm^2) + x(Dn + 2Fl) + z(En + 2Fm)$$
$$+ Fn^2 = 0$$

and

$$x^2(A + Dl + Fl^2) + xz(B + Dm + El + 2Flm)$$
$$+ z^2(C + Em + Fm^2) + x(D(n + k) + 2Fl)$$
$$+ z(E(n + k) + 2Fm) + F(n + k)^2 = 0.$$

The coefficients of x^2, xz, z^2 are the same in the above equations, so we have the same shape, the same eccentricity by lemma 5.

Lemma 7. Eccentricity constraint of a curve on a quadratic cone is an eighth order polynominal in the plane normals $1, n$: $F(1, n) = 0$.

Proof. Substituting $y = k/m - lx/m - nz/m$ into the cone equation defined in lemma 6, we get

$$x^2(A - Dl/m) + xz(B - Dn/m - El/m + F2l/mn/m)$$
$$+ z^2(C - En/m + F(n/m)^2) + x(Dk/m - 2Fl/mk/m)$$
$$+ z(Ek/m - 2Fn/mk/m) + F(k/m)^2 = 0.$$

Use $a^1x^2 + b^1xz + c^1z^2 + d^1x + e^1z + f^1 = 0$ to represent the above equation. Rotating the curve to eliminate the xz term, and carrying out the coordinate transformation, we get coefficients of

$$x^2 = a^1(1/2 + 1/2(c^1 - a^1)/\sqrt{((b^1)^2 + (c^1 - a^1)^2)})$$
$$+ b^1(b/2/\sqrt{((b^1)^2 + (c^1 - a^1)^2)})$$
$$+ c^1(1/2 - (c^1 - a^1)/2/\sqrt{((b^1)^2 + (c^1 - a^1)^2)}),$$

coefficients of

$$y^2 = c^1(1/2 + 1/2(c^1 - a^1)/\sqrt{((b^1)^2 + (c^1 - a^1)^2)})$$
$$-b^1(b/2/\sqrt{((b^1)^2 + (c^1 - a^1)^2)})$$
$$+ a^1(1/2 - (c^1 - a^1)/2/\sqrt{((b^1)^2 + (c^1 - a^1)^2)}).$$

Applying lemma 5 and assuming the ratio is r we get this equation of $F(l, m, n)$ as

$$(1 + r)^2B^2 + 4rA^2 - 4(1 + r^2)AC + 4rC^2$$
$$+ n/m^{(1+r)^2(-2BD)+4(1+r^2)AE-4r(2CE)}$$
$$+ l/m^{(1+r)^2(-2BE)+4(1+r^2)CD+4r(-2AD)}$$
$$+ ln/m^{2(1+r)^2 4BF+(1+r)^2 2DE-4(1+r^2)DE}$$
$$+ 1^2n/m^{3(1+r)^2(-4FE)+4(1+r^2)FE}$$
$$+ n^2l/m^{3(1+r)^2(-4FD)+4(1+r^2)FD}$$
$$+ l^2/m^{2(1+r)^2E^2-4(1+r^2)CF+4rD^2+4r(2AF)}$$
$$+ n^2/m^{2(1+r)^2D^2-4(1+r^2)AF+4rE^2+4r(2CF)}$$
$$+ l^3/m^3(4r(-2DF))$$
$$+ n^3/m^3(4r(-2EF))$$
$$+ l^4/m^4(4rF^2)$$
$$+ n^4/m^4(4rF^2)$$
$$+ l^2n^2/m^{4(1+r)^2F^2-4(1+r^2)F^2)}$$
$$= 0.$$

Recalling that $m = \pm\sqrt{(1 - l^2 - n^2)}$, if we substitute it in both sides, rearrange terms and square both sides, we will have an eighth order polynominal in l and n. So, the search on l, m, n is still one-dimensional if this constraint is true.

About the Author—ROBERT M. HARALICK was born in Brooklyn, New York, on 30 September 1943. He received a B.S. degree from the University of Kansas in 1966. He has worked with Autonetics and IBM. In 1965 he worked for the Center for Research, University of Kansas, as a Research Engineer and in 1969, when he completed his Ph.D. at the University of Kansas, he joined the faculty of the Electrical Engineering Department, where he served as a Professor from 1975 to 1978. In 1979 Dr. Haralick joined the faculty of the Electrical Engineering Department at Virginia Polytechnic Institute and State University, where he was a Professor and Director of the Spatial Data Analysis Laboratory. In 1984 Dr. Haralick joined Machine Vision International, where he is Vice President of Research. Dr. Haralick has done research in pattern recognition, multi-image processing, remote sensing, texture analysis, data compression, clustering, artificial intelligence and general systems theory. He is responsible for the development of GIPSY (General Image Processing System), a multi-image processing package which runs on a minicomputer system.

He is a member of the Institute of Electrical and Electronic Engineers, the Association for Computer Machinery, Sigma Xi, the Pattern Recognition Society and the Society for General Systems Research.

About the Author—Y. H. CHU received a B.S. degree in Mathematics from the National Central University in

Chungli, Taiwan, R.O.C., an M.S. in Mathematics from Virginia Tech, Blacksburg, VA, and a Ph.D. program in Computer Science, also from Virginia Tech, in 1983.

He is currently with the Department of Computer Science of Old Dominion University, Norfolk, VA. His research interests include computer vision, expert system construction, numerical analysis, natural and programming languages and image processing.

About the Author—LAYNE T. WATSON was born in Vanderburg County, IN, on 24 December 1947. He received a B.A. degree (magna cum laude) in Psychology and Mathematics from the University of Evansville, Evansville, IN, in 1969 and a Ph.D. degree in Mathematics from the University of Michigan, Ann Arbor, in 1974.

While at the University of Michigan, he studied numerical analysis under C. Moler, C. deBoor and G. Fix, and worked with K. Murty on complementarity problems. He served on the faculties of the University of Michigan and Michigan State University before coming to Virginia Polytechnic Institute and State University, Blacksburg. His current interests are fluid dynamics, structural mechanics, homotopy algorithms and image processing.

Dr. Watson is a member of Phi Kappa Phi, Blue Key, Psi Chi, Kappa Mu Epsilon, Phi Beta Chi, SIAM, SIAM AGLA, IEEE, IEEE Computer Society, ACM and ACM SIGNUM.

About the Author—LINDA G. SHAPIRO was born in Chicago, IL, in 1949. She received a B.S. degree in Mathematics from the University of Illinois at Urbana-Champaign in 1970 and M.S. and Ph.D. degrees in Computer Science from the University of Iowa, Iowa City, in 1972 and 1974, respectively.

She was an Assistant Professor of Computer Science at Kansas State University, Manhattan, from 1974 to 1978 and an Associate Professor at Virginia Polytechnic Institute and State University, Blacksburg from 1979 to 1984, when she joined Machine Vision International as Director of Intelligent Systems. Her research interests include computer vision, pattern recognition, intelligent spatial information systems, computer graphics and data structures. She has completed an undergraduate textbook on data structures with R. Baron.

Dr. Shapiro is currently editor of *Computer Vision, Graphics, and Image Processing*. She is a senior member of the IEEE Computer Society and a member of the Association for Computing Machinery, the Pattern Recognition Society and the American Association for Artificial Intelligence. She is also co-editor of the newsletter for the IEEE Technical Committee on Machine Intelligence and Pattern Analysis.