Comput. & Graphics Vol. 6, No. 1, pp. 35–37, 1982
Printed in Great Britain.

0097–8493/82/010035–03$03.00/0
Pergamon Press Ltd.

AN ORIENTATION METHOD FOR CENTRAL PROJECTION PROGRAMS

DAVID P. ANDERSON
Chemistry Department, University of Wisconsin, Madison, WI 53706, U.S.A.

(*Received* 16 *December* 1980)

Abstract—A centrally projected image is based on an object, a viewpoint and a viewer orientation. The programs reported to date which calculate centrally projected images require all of these as input. Instructions for the programs usually suggest that the hypothetical observer face toward the center of the object. There are two major problems with this: (1) there may be no clear way of defining the "center", and (2) an orientation chosen in this manner may result in parts of the object lying behind or to the side of the observer and hence being invisible or severely distorted under the projection. This paper describes an algorithm for calculating a viewing direction which will render visible the entire object whenever this is possible, and will furthermore minimize distortion in the projected image.

A FORMALIZATION OF THE PROBLEM

We will begin by describing central projection. Suppose an object in three-space is viewed by an observer with fixed position and orientation. Let **VP** be the viewpoint, described by a three-vector. Let **VA**, **VR** and **VU** be unit vectors which, in the orientation of the observer, point in the ahead, right and up directions respectively. (**VR**, **VA**, **VU**) must be orthonormal and right-handed (i.e. **VU** = **VR** × **VA**). **VA** is referred to as the "viewing direction". The "projection plane" or "image space" is the plane which is normal to **VA** and contains the point **VP** + **VA**. The image space has a coordinate system with origin **VP** + **VA** and basis vectors **VR** and **VU**.

In addition to the underlying (x, y, z) coordinates, three-space can be given a coordinate system in which **VP** is the origin and (**VR**, **VA**, **VU**) are basis vectors; these will be called "observer" coordinates. Let **P** be an object point; the observer coordinates (r, a, u) of **P** are the projections of **P-VP** onto **VR**, **VA** and **VU**, given by

$$\begin{aligned} r &= [\mathbf{P\text{-}VP}, \mathbf{VR}] \\ a &= [\mathbf{P\text{-}VP}, \mathbf{VA}] \\ u &= [\mathbf{P\text{-}VP}, \mathbf{VU}] \end{aligned} \tag{1}$$

where [,] denotes the scalar product.

P is visible iff it lies in front of the observer, i.e. iff $a > 0$. If **P** is visible, its "central projection" is the point **P′** where the ray which starts at **VP** and goes through **P** intersects the projection plane. The image space coordinates of **P′** are (px, py) where

$$\begin{aligned} px &= r/a \\ py &= u/a. \end{aligned} \tag{2}$$

This is shown by similar triangles, noting that the projection plane, in the observer coordinate system, is the plane $a = 1$.

Three-space is divided into visible and invisible parts by the plane $a = 0$. It is possible to compute the image of a simple object which is only partially visible without clipping it in three dimensions. This can be done as follows: if the segment from $\mathbf{P}_1$ to $\mathbf{P}_2$ is in the object, where $[\mathbf{P}_1\text{-}\mathbf{VP}, \mathbf{VA}] > 0$ and $[\mathbf{P}_2\text{-}\mathbf{VP}, \mathbf{VA}] < 0$, we can compute the images $\mathbf{P}_1'$ and $\mathbf{P}_2'$ by the above formulae, ignoring for the moment the invisibility of $\mathbf{P}_2$. The projection of the visible part of the segment is the ray in the image space which begins at $\mathbf{P}_1'$ and goes directly away from $\mathbf{P}_2'$ (that is, goes in the direction of $\mathbf{P}_1'\text{-}\mathbf{P}_2'$).

Most projection programs, however, are not capable of handling this case, and hence need a viewing direction which makes the entire object visible. Call a viewing direction "feasible" iff it has this property (the notion of feasibility is a function of object and viewpoint). For some viewpoints there is no feasible viewing direction, namely those points in the closed convex hull of the object. For any viewpoint the set of feasible viewing directions is a subset of the unit sphere which lies entirely in some hemisphere; it is convex in the sense that if $\mathbf{V}_1$ and $\mathbf{V}_2$ are feasible, so is any point on the smaller arc of the great circle through $\mathbf{V}_1$ and $\mathbf{V}_2$. Assuming that the object is closed and bounded, the set is open in the sphere topology.

If there is a feasible viewing direction there are infinitely many, and we need a criterion for selecting the best one. To this end we will analyze distortion in centrally projected images; our criterion will be lack of distortion.

DISTORTION FROM CENTRAL PROJECTION

Let us assume that a viewpoint **VP** has been used to calculate an image, and that we now look at the image from points other than **VP**. When viewed orthogonally, the image at a point **P′** appears locally to be stretched out in the direction of the line through **P′** and the image space origin. In particular, if the object is a small sphere whose center has projection **P′**, then the image is the intersection of the projection plane and the cone with vertex **VP** generated by the sphere, and hence is an ellipse. The major axis of the ellipse lies on a line through the image space origin. Let a be the angle between **P′-VP** and **VA**, b be half the angle subtended by the sphere from **VP**, and r be the distance from **VP** to **P′**.

$\mathbf{P}'$ divides the major axis into two segments with lengths

$$L_1 = r\sin(b)/\cos(a+b)$$
$$L_2 = r\sin(b)/\cos(a-b). \quad (3)$$

Hence the major axis has length

$$L_{\text{major}} = L_1 + L_2 = r\sin(b)(1/\cos(a+b) + 1/\cos(a-b)). \quad (4)$$

The minor axis has length

$$L_{\text{minor}} = 2r\sin(b). \quad (5)$$

As the size of the sphere (and hence of its image) goes to zero, the ratio of the lengths of the major and minor axes approaches a limit which is at least one and is equal to one iff $\mathbf{P}'$ is the image space origin. We adopt this limit as a measure of the local distortion at $\mathbf{P}'$; it is given by

$$D(\mathbf{P}') = \lim_{b\to 0} L_{\text{major}}/L_{\text{minor}} = \lim_{b\to 0}(1/\cos(a+b) + 1/\cos(a-b))/2 = 1/\cos(a). \quad (6)$$

As a is varied from 0 to $\pi/2$, the distortion increases strictly and without bound. Thus minimizing the maximum distortion is equivalent to minimizing the maximum value of a.

So far we have defined the notions of feasible and optimal viewing directions, and have found that the optimal direction is the vector **VA** which minimizes the maximum over object points **P** of the angle between **VA** and **P**-**VP**. We now present an algorithm to find this direction.

MOTIVATION FOR THE ALGORITHM

Assume that the convex hull of the object is a polyhedron. Let $\mathbf{P}_1, \ldots, \mathbf{P}_n$ be a set of object points which includes the set of vertices of the convex hull; e.g. if the object is a set of polygons, $\mathbf{P}_1, \ldots, \mathbf{P}_n$ could be taken to be the set of vertices of the polygons. Define the "image vectors" to be the vectors $\mathbf{P}_i$-**VP** normalized to unit length; these are directions from the observer to object points and should now be visualized as emanating from the origin. In what follows, we will deal with solid circular cones which are single and whose vertex is the origin. Such a cone will be called "feasible" iff it includes all the image vectors (which is equivalent to including the entire object). The axial direction of a feasible cone is a feasible viewing direction. The "optimal" cone is the narrowest feasible cone; the axial direction of the optimal cone is the optimal viewing direction.

The optimal cone satisfies one of the following conditions:

(1) the two most distant image vectors (in the sense of angular separation) both lie on the surface of the cone, and their midvector is its axis, or

(2) at least three image vectors lie on the surface of the cone.

This assertion can be proved by contradiction: suppose that the optimal cone satisfies neither (1) nor (2). If no image vectors lie on its surface, then we can shrink the cone slightly, keeping its axis fixed, and get a narrower feasible cone. If exactly one image vector, **V**, lies on the cone surface, we can find a narrower feasible cone by moving the axis slightly toward **V**, while constraining the cone to have **V** on its surface. If the cone has exactly two image vectors $\mathbf{V}_1$ and $\mathbf{V}_2$ on its surface, but its axis is not the midvector of $\mathbf{V}_1$ and $\mathbf{V}_2$, then we move the axis slightly toward the midvector, while constraining the cone to have $\mathbf{V}_1$ and $\mathbf{V}_2$ on its surface. In any case, we have found a narrower feasible cone, which contradicts the assumption of optimality.

The following observation makes the algorithm practical: a cone includes all the image vectors iff it includes the edges of the smallest convex solid pyramid whose vertex is the origin and which includes all the image vectors. Hence the optimal cone for the pyramid edges is the same as the optimal cone for the entire set of image vectors; hence we can discard all the image vectors except the pyramid edges. In addition, the image vectors which lie on the surface of the optimal cone are edges of the pyramid.

We combine the above facts to get an algorithm which is fast yet general: first, the subset of the image vectors consisting of the edges of the convex pyramid is constructed. Second, we find the two most distant edges and see if the cone centered at their midvector and containing them on its surface also contains the other edges. If so, this cone is optimal. Otherwise, for every set of three edges, we see if the (unique) cone whose surface passes through them contains the other edges. The optimal cone is then the narrowest one satisfying this condition.

It is interesting to consider the limit as the optimal cone becomes narrow, for then the geometry of the part of the sphere surface on which the image vectors lie becomes like that of the plane. The problem is then transformed to that of finding the smallest disc which contains a given set of points in the plane. Suitable translations of the above two assertions hold in the planar case: namely, that the optimal disc is determined by the vertices of the convex hull of the set of points, and its boundary either passes through three of the points or passes through two and is centered at their midpoint.

DETAILS OF THE ORIENTATION METHOD

Let us now examine the computational details of the spherical case. The convex pyramid defined earlier is constructed inductively, starting with the pyramid determined by any three non-coplanar image vectors, then adding the remaining image vectors one at a time. At each stage the pyramid is described by a circularly linked list of edge vectors, ordered clockwise around the pyramid surface when viewed from outside the pyramid. Suppose $\mathbf{V}_1$ immediately precedes $\mathbf{V}_2$ in the edge list;

then $\mathbf{V}_1 \times \mathbf{V}_2$ is normal to the pyramid face with edges $\mathbf{V}_1$ and $\mathbf{V}_2$. The pyramid consists of those points which have nonnegative scalar products with all the face normals. Since the face normals are used often, they are stored in a list with the same link structure as the edge list.

After the initial pyramid has been formed using three non-coplanar image vectors, each remaining image vector $\mathbf{V}$ is processed as follows: the scalar products between $\mathbf{V}$ and each of the face normals are found. These will be positive (negative) if $\mathbf{V}$ is on the right (wrong) side of the face plane. If all the scalar products are nonnegative then $\mathbf{V}$ lies within the current pyramid and is ignored. If all the scalar products are negative then $\mathbf{V}$ lies in the pyramid which is the reflection of the current pyramid through the origin; any cone which includes the current pyramid must exclude $\mathbf{V}$, hence there is no feasible cone and therefore no feasible viewing direction. If some of the scalar products are negative and some are not, then $\mathbf{V}$ becomes a new pyramid edge. The faces of the pyramid which gave negative scalar products necessarily are contiguous; the edges which are interior to this set of faces are removed from the list, and are replaced by $\mathbf{V}$. The list of face normals is updated accordingly.

When all the vectors have been processed, the pyramid is complete and we are assured of the existence of a feasible viewing direction. It is important to make the set of edges as small as possible. If two or more faces are found to be nearly coplanar they can be replaced by a single face; this is done by removing the edges interior to the set of faces.

We now find the two edge vectors $\mathbf{V}_1$ and $\mathbf{V}_2$ whose angular distance is greatest (i.e. whose scalar product is least). The midvector $\mathbf{V}_c$ is calculated as the normalized sum of $\mathbf{V}_1$ and $\mathbf{V}_2$. Let $d = [\mathbf{V}_c, \mathbf{V}_1]$; d is the cosine of the angle between $\mathbf{V}_c$ and $\mathbf{V}_1$ (or $\mathbf{V}_c$ and $\mathbf{V}_2$). The cone with axis $\mathbf{V}_c$ and which contains $\mathbf{V}_1$ and $\mathbf{V}_2$ on its surface is optimal iff it is feasible, which in turn is equivalent to $[\mathbf{V}_c, \mathbf{V}] \geq d$ for all edges $\mathbf{V}$.

If this cone is not feasible, we then loop over every three-element subset $(\mathbf{V}_1, \mathbf{V}_2, \mathbf{V}_3)$ of the set of edge vectors. A vector $\mathbf{V}$ is found which makes equal acute angles with $\mathbf{V}_1$, $\mathbf{V}_2$ and $\mathbf{V}_3$. This is done by finding a non-zero solution to the underdetermined linear system $[\mathbf{V}, \mathbf{V}_1] = [\mathbf{V}, \mathbf{V}_2] = [\mathbf{V}, \mathbf{V}_3]$, negating if necessary to give a positive scalar product with $\mathbf{V}_1$, then normalizing. If $[\mathbf{V}, \mathbf{V}_1] \leq [\mathbf{V}, \mathbf{V}']$ for all edge vectors $\mathbf{V}'$, then $\mathbf{V}$ is feasible; the optimal direction is the $\mathbf{V}$ corresponding to some $(\mathbf{V}_1, \mathbf{V}_2, \mathbf{V}_3)$ which is feasible and for which $[\mathbf{V}, \mathbf{V}_1]$ is greatest.

This concludes the calculation of **VA**. Define the "image radius" to be the largest distance from the image space origin to an image point. A bonus of the algorithm is that it gives the image radius; this is $\tan(x)$, where x is the angle between the axis and surface of the optimal cone. This can be used for automatic scaling; that is, given a desired radius r of the plotted image, the program uses an appropriate scaling factor, namely $r/\tan(x)$. Note that this scale factor is also the distance (in the physical units of the output device) at which the picture should be viewed, orthogonally to the image space origin, to eliminate distortion. The image radius is minimized by the optimal viewing direction and is achieved by at least two image points.

CALCULATION OF THE OTHER ORIENTATION VECTORS

Having computed **VA**, we must still find **VR** and **VU** such that (**VR**, **VA**, **VU**) is an orthonormal right-handed basis. This condition does not uniquely determine **VR** and **VU**; the leeway corresponds to rotating the final image. We might ask that, in addition, the angle between **VU** and $(0, 0, 1)$ be as small as possible given that $[\mathbf{VU}, \mathbf{VA}] = 0$, for then the "up" direction in the image will correspond as closely as possible to the up (i.e. z) direction in three-space. This will be the case if **VR** is taken to be $(0, 0, 1) \times \mathbf{VA}$ and then, necessarily, **VU** is $\mathbf{VA} \times \mathbf{VR}$.

EFFICIENCY

Let n be the number of image vectors. A reasonable guess for the order of the average number of faces in the convex pyramid, as it is being constructed, is $n^{1/2}$. In constructing the pyramid we must find the scalar product of each image vector with each face normal of the pyramid at that stage, so the time to construct the pyramid is of order $n^{3/2}$. The remaining computation time depends on the case. Finding the two most distant vectors takes time of order n, so if the two-vector cone is feasible we're done in total time of order $n^{3/2}$. Otherwise, for every three-element subset of the edges we must see if the other edges lie in the cone determined by the three; there are about $n^{3/2}$ three-element subsets, so there are on the order of n^2 steps in this case.

It should be pointed out that n need not be of the same order of magnitude as the number of endpoints or vertices in the object. With a slight loss of accuracy and generality, a complex part of the object can be represented by the vertices of any convex polyhedron known to include it. In particular, the entire object could be represented by the eight points determined by its limits in the coordinate directions.

The algorithm has been implemented in structured FORTRAN on a Harris Slash 7 minicomputer, as part of a grid-point surface plotting program. For a 40 by 40 surface ($n = 1600$) between one and two CPU seconds are needed to find the viewing direction. Refinements of the algorithm could undoubtedly reduce this still further.

CONCLUSION

The problem of finding an optimal viewing direction has been formalized, and a solution given. The algorithm is not hard to program, and is efficient enough for interactive and real-time applications. One would hope that future graphics programs which use central projection will handle the orientation problem automatically, using this method or another like it.

Acknowledgements—The author wishes to thank Alex Strong for sharing his insight and the University of Wisconsin Chemistry Department for the use of its computing facility during the development of the program.