

Animated Cartoons by Computer Using Ellipsoids

[Don Herbison-Evans](mailto:donherbisevans@yahoo.com) , donherbisevans@yahoo.com

Technical Report 94 (1974), Basser Department of Computer Science, University of Sydney, Australia

ABSTRACT

Most solid objects can be approximated as closely as required by a concatenation of interpenetrating three dimensional ellipsoids. This report describes a computer program that has been written using this principle to draw projections of human and other articulated figures.

DRAWING THE ELLIPSOIDS

Each ellipsoid may be specified by the coordinates of its centre : $\mathbf{u} = (u_x, u_y, u_z)$, the lengths of the three semi-axes : (v_1, v_2, v_3) , and a 3x3 rotation matrix : \mathbf{R} .

The surface of the unrotated ellipsoid can be described by the equation

$$\left(\frac{x' - u_x}{v_1} \right)^2 + \left(\frac{y' - u_y}{v_2} \right)^2 + \left(\frac{z' - u_z}{v_3} \right)^2 = 1$$

where

$$\mathbf{x}' = (x', y', z')$$

The equation of the surface may be written in matrix and vector form as :

$$(\mathbf{x}' - \mathbf{u})^+ \begin{pmatrix} 1/v_1^2 & 0 & 0 \\ 0 & 1/v_2^2 & 0 \\ 0 & 0 & 1/v_3^2 \end{pmatrix} (\mathbf{x}' - \mathbf{u}) = 1$$

and abbreviated to :

$$\mathbf{x}^+ \cdot \mathbf{V} \cdot \mathbf{x} = 1$$

where $\mathbf{x} = \mathbf{x}' - \mathbf{u}$

and \mathbf{x}^+ means the transpose of \mathbf{x} .

Rotation of the ellipsoid to some arbitrary orientation is effected by a similarity transformation, giving :

$$\mathbf{x}^+ \cdot \mathbf{R}^+ \cdot \mathbf{V} \cdot \mathbf{R} \cdot \mathbf{x} = 1$$

where \mathbf{R} is a rotation matrix. This \mathbf{R} can be generated as the product of a series of elementary rotations about the coordinate axes :

$$\mathbf{R} = \mathbf{R}_1 \cdot \mathbf{R}_2 \cdot \mathbf{R}_3 \dots \mathbf{R}_n$$

where

$$\mathbf{R}_i = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{pmatrix} \text{ or } \begin{pmatrix} \cos\phi & 0 & \sin\phi \\ 0 & 1 & 0 \\ -\sin\phi & 0 & \cos\phi \end{pmatrix} \text{ or } \begin{pmatrix} \cos\phi & \sin\phi & 0 \\ -\sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

representing an elementary rotation of an angle phi (ϕ) about the x, y, or z axis respectively.

Rotation axes can be specified either in fixed coordinates or in the axes of the ellipsoid by respectively either multiplying the next elementary matrix onto the right or left hand side of \mathbf{R} .

Ellipsoids, when projected orthographically onto the (x,y) plane, give an elliptical profile. The equation of the ellipse may be found from the matrix $\mathbf{R}^+ \cdot \mathbf{V} \cdot \mathbf{R}$ as follows :

writing

$$\mathbf{M} = \mathbf{R}^+ \cdot \mathbf{V} \cdot \mathbf{R} = \begin{pmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{pmatrix}$$

the equation of the ellipsoid may be written as a quadratic in z :

$$m_{33} \cdot z^2 + 2(m_{13}x + m_{23}y) \cdot z + (m_{11}x^2 + 2m_{12}xy + m_{22}y^2 - 1) = 0$$

where we have used the diagonal symmetry of \mathbf{M} to simplify the expression. It may be further abbreviated to :

$$A \cdot z^2 + 2B \cdot z + C = 0$$

where A, B, and C are the functions :

$$A = m_{33}$$

$$B = m_{13} \cdot x + m_{23} \cdot y$$

$$C = m_{11} \cdot x^2 + 2m_{12} \cdot x \cdot y + m_{22} \cdot y^2 - 1$$

This has solutions :

$$z = \{-B \pm \sqrt{B^2 - AC}\} / A$$

For any particular values of x and y, the two solutions of this equation for z represent the front and back of the ellipsoid surface. Where the front and back meet, we have the profile which will thus have the equation :

$$B^2 - AC = 0$$

which may be written as :

$$a.x^2 + 2b.x.y + c.y^2 = 1$$

where :

$$\begin{aligned} a &= m_{11} - m_{13}^2/m_{33} \\ b &= m_{12} - m_{13}m_{23}/m_{33} \\ c &= m_{22} - m_{23}^2/m_{33} \end{aligned}$$

This is the required equation of the elliptical profile.

POLYGON APPROXIMATION TO AN ELLIPSE

A polygomal approximation to this profile may be generated using the fast method of Cohen (1970). An approximation to a circle centred on the origin may be generated as a series of p straight lines by joining a sequence of points generated by the equation :

$$\begin{pmatrix} x \\ y \end{pmatrix}_{i+1} = \begin{pmatrix} \cos\alpha & \sin\alpha \\ -\sin\alpha & \cos\alpha \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}_i$$

or

$$s_{i+1} = P.s_i$$

where

$$\begin{aligned} s_1 &= (x,y)_1 \text{ is some initial point on the circle, and} \\ \alpha &= 2.\pi/p \quad (\alpha = 2.\pi/p \text{ if your browser supports Greek letters!}) \end{aligned}$$

An ellipse with its axes parallel to the coordinate axes and an axial ratio of $f = a_1/a_2$ can be similarly generated using :

$$s_{i+1} = E^{-1}.P.E.s_i$$

where **E** lengthens one axis, i.e.

$$E = \begin{pmatrix} f & 0 \\ 0 & 1 \end{pmatrix}$$

An ellipse with its axis a_1 at an angle θ to the x axis can be generated using :

$$s_{i+1} = Q^{-1}.E^{-1}.P.E.Q.s_i$$

where

$$Q = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix}$$

This may be abbreviated to :

$$\mathbf{s}_{i+1} = \mathbf{T} \cdot \mathbf{s}_i$$

where the matrix \mathbf{T} has elements :

$$\begin{aligned} t_{11} &= \cos\alpha - (f-1)/f \cdot \sin\alpha \cdot \sin\theta \cdot \cos\theta \\ t_{12} &= (f \cdot \cos^2\theta + (\sin^2\theta)/f) \cdot \sin\alpha \\ t_{21} &= -(f \cdot \sin^2\theta + (\cos^2\theta)/f) \cdot \sin\alpha \\ t_{22} &= \cos\alpha + (f-1)/f \cdot \sin\alpha \cdot \sin\theta \cdot \cos\theta \end{aligned}$$

This may be related back to the quadratic in x and y :

$$a \cdot x^2 + 2 \cdot b \cdot x \cdot y + c \cdot y^2 = 1$$

through the eigenvalues :

$$\lambda_{\pm} = \sqrt{(a+c) \pm \sqrt{4b^2 + (a-c)^2}}$$

which give an axial ratio

$$f = \lambda_+ / \lambda_-$$

and an orientation

$$\theta = \arctan(2b/(a-c))/2$$

A simple initial point on the ellipse is

$$\mathbf{s}_1 = (1/a^{1/2}, 0)$$

This method of drawing an ellipse has the added advantage of putting an increased density of points where the curvature is greatest.

It does not require any square roots inside the loop, as the direct solution of a quadratic would.

The sequence of points on the elliptical profile obtained by this method refer to a coordinate system with the origin at the ellipse centre. They must be added to the coordinates of the ellipse centre before drawing them.

PROJECTING THE ELLIPSOIDS

The equation of the orthographically projected profile of each ellipsoid is obtained by Cohen's method. This is then shifted to the appropriate position in three dimensional space by translation by an amount \mathbf{u} (the coordinates of the ellipsoid centre). The resulting coordinates(\mathbf{x}) are perspective-projected assuming that the observer is at the origin, onto a plane window parallel to the (x , y) plane, centred on the position (d_x , d_y , d_z).

Then

$$\begin{aligned} p_x &= (x - d_x) \cdot d_z / z \\ p_y &= (y - d_y) \cdot d_z / z \end{aligned}$$

where (p_x , p_y)

REPRESENTATION OF FIGURES

A series of ellipsoids was fitted by eye to the required figure, and a series of joints fitted where the figure was to articulate.

Each joint connected two ellipsoids. Each ellipsoid could have any number of joints.

The data for the program specifies the number of ellipsoids, their semi-axis lengths, the number of joints, a list of which ellipsoids connect at each joint, and the vector distance of each joint from the centres of each of the two ellipsoids that they connect.

From this data, the three dimensional positions of each of the joints and ellipsoid centres can be deduced. A set of rotation matrices, one for each ellipsoid, are also set up as unit matrices.

ANIMATION

A series of subroutines was written to perform the various actions. Data cards are read to specify in sequence which actions to perform and with what parameters. The subroutines are :

a. **DRAW**

This draws the approximate perspective projection of the figure.

b. **SHIFT** (x, y, z)

This shifts the whole figure by an amount (x,y,z), i.e. adds (x,y,z) to the coordinates of all the joints and to the centres of all the ellipsoids.

c. **ROTATE** (θ , a, x, y, z)

This shifts the origin to point (x,y,z), and then rotates the whole figure (i.e. multiplies the joints, the ellipsoid centres, and the ellipsoid rotation matrices by the appropriate three-dimensional elementary rotation matrix according to the value of axis 'a'). It then shifts the origin back again to where it was. As an option, the centre of a specified ellipsoid can be nominated as the centre of rotation (x,y,z).

d. **GROUND**

This finds the bottom points of all the ellipsoids, and then the lowest one of these. The whole figure is then shifted up by this amount, so that the lowest point is just touching the ground (the y=0 plane). The y coordinate of the bottom, b_y , of an ellipsoid with centre at \mathbf{u} , semi-axis lengths \mathbf{v} , and rotation matrix \mathbf{R} is

$$b_y = u_y - (r_{12}.v_x)^2 - (r_{22}.v_y)^2 - (r_{32}.v_z)^2$$

assuming the positive y direction is 'up'.

e. **VIEW** (x, y, z)

This moves the centre of the viewing window to (x,y,z) for use in the projection process when drawing the figure.

f. **GROW** (f)

This multiplies all semi-axis lengths, joint positions, and ellipsoid centre coordinates by the value 'f'. It then calls GROUND to reset the figure to touch the plane y=0 at the figures lowest point.

g. **BEND** (j, e, θ , a)

This shifts the origin to joint 'j', and then rotates using the appropriate elementary rotation matrix those joints, and the matrices and centres of those ellipsoids NOT connected through ellipsoid 'e' to joint 'j'. The origin is then shifted back again, and the figure re-grounded.

HIDDEN LINE SUPPRESSION

Each point on the profile of each ellipsoid is tested for visibility.

Suppose the coordinates of a point (x_i, y_i, z_i) on the profile of the i^{th} ellipsoid have been calculated. These are checked against all other ellipsoids. For the j^{th} , the values of x_i and y_i can be substituted into the quadratic for z_j of the j^{th} ellipsoid :

$$z_j = (-B \pm \sqrt{B^2 - AC}) / A$$

If the discriminant $B^2 - A.C$ is negative, then (x_i, y_i, z_i) is to the side of ellipsoid j and cannot be obscured by it.

If the discriminant is positive, then the root with the negative sign gives the point nearer the observer on the j^{th} ellipsoid which has the same x and y values as the point on the outline of the i^{th} . If $z_j < z_i$, then the point (x_i, y_i, z_i) is obscured by the nearer face of ellipsoid j , otherwise it is visible. Note that this test also works for the case where the ellipsoids interpenetrate.

The number of tests to be done in total is equal to $p.n.(n-1)$ where there are 'p' points around each ellipse and 'n' ellipsoids. The number of tests can be considerably diminished by some initial pre-testing to find out which ellipsoids cannot possibly obscure which.

Thus if the maximum semi-axes of the i^{th} and j^{th} ellipsoids are h_i and h_j , and the centres of the ellipsoids are at \mathbf{u}_i and \mathbf{u}_j , then if

$$(h_i + h_j)^2 < (u_{ix} - u_{jx})^2 + (u_{iy} - u_{jy})^2$$

then the ellipsoids cannot obscure any part of each other.

Before drawing each ellipsoid, this test is done between it and all the others, and tags put on those that can obscure it. Then as the ellipsoid is being drawn, only the tagged ellipsoids are tested for obscuration.

ERRORS

This method of testing hidden lines is only rigorously valid for orthographic projection. In using it for perspective projection, it leads to some errors. However, the use of polygonal approximations to the ellipse outlines has already introduced some errors. The projection error can be kept to less than the polygon approximation error if

$$z_m \gg (p \cdot x_a) / (\pi \cdot u_a)$$

where

z_m = mean distance of the figure from the observer

x_a = maximum value of $(x^2 + y^2)^{1/2}$ of any part of the figure

u_a = longest axis of the ellipsoids composing the figure

p = number of points around the polygonal outline.

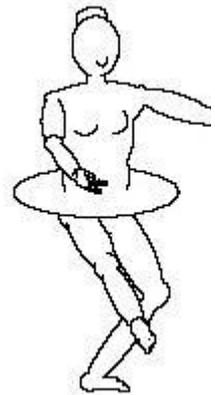
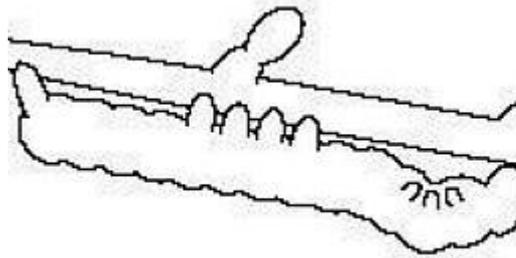
RESULTS

The illustrations show some stills from a number of animated sequences.

REFERENCE

Cohen, D. (1970)

"Linear difference Curves", Proceedings of the International Symposium *Computer Graphics 1970*, Brunel University.



(updated 1 November 2009)