Conic sections

Dear Editor:

I read with interest Theo Pavlidis's article, "Scan Conversion of Regions Bounded by Parabolic Splines," in the June issue.

I have devised a similar technique that will work for all conic sections, not just parabolas. This is incorporated into a computer-aided design system to perform hatching, cross-hatching, and also area-fill.

A conic section may be parameterized in the form

\[ x = \frac{a t^2 + b t + c}{d(t)} \]

with \( t \) as the parameter, and with these vectors:

- \( x = (x, y, z); \) Three-Dimensional Coordinates
- \( a = (a_x, a_y, a_z); \) Second-order Coefficients
- \( b = (b_x, b_y, b_z); \) First-order Coefficients
- \( c = (c_x, c_y, c_z); \) Zero-order Coefficients

and this scalar function:

\[ d(t) = 1 \text{ for a line } a = 0 \]
\[ = t \text{ for a hyperbola}; \]
\[ = 1 + t^2 \text{ for an ellipse}. \]

For each scan-line or hatch line, we set the appropriate coordinate to a constant. As an example, suppose we have an ellipse, and we wish to find where it intersects \( y = 5 \). Our equation is:

\[ 5 = \frac{a_t^2 + b t + c_y}{1 + t^2} \]

or:

\[ (a_t^5) t^2 + b t + (c_y - 5) = 0 \]

which can easily be solved, for \( t \), as a quadratic equation. Substituting \( t \) into the original equations for the curve yields the intersection point(s).

If there is only an arc of an ellipse, then the arc is defined as being between two parameter limits, and the parameter values \((t)\) can be tested to see if they fall within these limits.

These techniques may be applied to higher-order rational curves, generating higher-order polynomial equations, of course.

More information about this technique may be found in my doctoral dissertation or in its companion paper.\(^{12}\)

Yours truly,

Joshua Zev Levin
Aydin Computer Systems
Horsham, Penn.


The author replies:

In regards to the comments by Professor Levin, it is well-known that any conic can be expressed in a rational parametric form

\[ P(t) = \frac{A t^2 + B t + C}{a t^2 + b t + c} \]

and therefore it is simple in principle to find its intersection with the line \( y = \) constant

I limited the treatment in the paper to parabolas only because the formulas become quite messy in the general case. One must have formulas not only for the intersections but also for subdivisions into monotonic arcs in order to apply a scan conversion algorithm that is of the order \( N \log N \) rather than order \( N^2 \).

The main point of the paper is not that one can do scan conversion for parabolas (I think that is obvious) but that this can be done by an algorithm similar to the one used for the scan conversion of polygons.

Theo Pavlidis
LETTERS TO THE EDITOR

(CONTINUED)

Don't forget twixt

Dear Editor:

I would like to commend the Doctors Thalmann on their extensive bibliography of computer animation. However, I want to point out the omission of the work by Dr. Julian Gomez on the development of an interactive, event-driven animation system called twixt, which, for several years, has been the central animation control system in use at the Ohio State University Computer Graphics Research Group. A number of animated sequences, including “Snoot and Muttley,” by Susan Van Baerle and Douglas Kingsbury, have been generated using twixt.

The system is based on a direct, visual system of interaction with graphical objects and their attributes, as opposed to, say, animation scripts. Experience with the system has shown that twixt provides the user with a flexible and powerful set of animation tools.

Sincerely,
David Zeltzer
Assistant Professor
Massachusetts Institute of Technology


The authors reply:

Yes, that was an omission. At the time the bibliography was written, the proceedings of Eurographics 84 were not yet published. However, we did reference Dr. Gomez's work in our October 1985 IEEE CG&A article, “Three-Dimensional Computer Animation: More an Evolution than a Motion Problem.”

Nadia Magnenat-Thalmann and Daniel Thalmann

Commercial animation

Dear Editor:

Although I enjoyed the Thalmann's article on 3D computer animation and their inventory of evolution techniques, I feel compelled to dispel the notion that most commercial computer animation results from translation, scaling, and rotation, or from keyframe interpolation.

Indeed, speaking as a working professional, I must confess that much of our time and energies are spent not modeling changes in the shape or position of objects or the camera, but in lighting, surface properties, and a host of parameters that might be called “graphic art variables” and which often have little relationship to physics.

Little is known about those factors outside the sphere of working professionals, but in general graphic art factors include a host of techniques used to visually enhance the static as well as moving image. The purpose of these techniques is to give the image styling and emotional attitude and, especially with factors that involve action, to help lead the eye and direct attention.

I could list many examples here; classical ones includes glows, rimlight, kicks, diagonal highlights, beveled edge normals, and filters. All complement basic shape deformations, transformations, and lighting. Rendering in a commercial environment is seldom automatic or completely algorithmic.

Respectfully,
Judson Rosebush
The Judson Rosebush Co.
New York, N.Y.

The authors reply:

We quite agree that most people involved in computer animation spend most of their time in lighting and surface properties. What we have tried to say is that the same effort has not been made in the design of evolution laws that drive the movement of objects, cameras, lights, and colors.

Daniel Thalmann and
Nadia Magnenat-Thalmann