Exact visibility calculation for space-filling molecular models

Jean-Michel Cense

Ecole Nationale Supérieure de Chimie de Paris, 11, rue Pierre et Marie Curie, 75231 Paris Cedex 05, France

The visible portion of two intersecting atoms is exactly calculated to be bounded by a circular arc and an elliptic arc. The elliptic arc can be approximated either by a circular arc or by a cubic Bezier curve. Using one shaded template for each atom type and stamping the templates in a mask bitmap delineated by these curves, images are generated rapidly.

Keywords: raster graphics, space-filling molecular models, cubic Bezier curve, interactive drawing

INTRODUCTION

Shaded space-filling models are certainly the most informative representations of the surface of molecular structures. These models suffer from one major drawback: Only the front atoms are clearly visible, making it necessary to rotate the model to observe the back atoms. Furthermore, space-filling models are rather flat and do not convey depth relationships efficiently, even when cast shadows are displayed. Two methods are currently used to circumvent this deficiency, stereo pictures and animated pictures. Both methods make use of the brain's capacity to combine two or more two-dimensional (2D) pictures into one three-dimensional (3D) picture. The brain easily rectifies small defects; therefore animated shaded images of correctly delineated interpenetrating atoms are sufficient to initiate 3D perception. Up to now, animations were prepared by storing successive pictures in memory and playing the sequence rapidly. This procedure precluded the preparation of long animations because the storage of color pictures is memory demanding but, with the advent of accelerated graphic cards, real-time animation will be manageable in the very near future. In this paper we describe our efforts toward the exact and rapid determination of the visible portion of an atom.

In 1976, Knowlton and Cherry[1] published the first algorithm for rendering molecular models built from spheres and cylinders. The visible regions of space-filling models were bounded by circles approximating ellipses. This algorithm was later extended by Max,[2] who added shading and highlights, and by Gwillian and Max,[3] who added cast shadows.

In 1989, Cense[4] described a fast implementation of an area-based algorithm that makes use of the Macintosh Toolbox graphics (bitmaps and regions). This algorithm decomposes the drawing process into two separate units: calculation of shaded templates, and stamping into clipping regions delineated by curves. As pixels are always accessed in blocks, the drawing is very fast. The same principle has since been published by Schafmeister[5] for the Microsoft Windows System. As was recognized by Gwillian and Max,[3] the main advantage of these area-based methods is that the visible region of each atom can be calculated at any level of precision, depending only on the resolution of the output device, screen, laser printer, or slide maker.

Other algorithms using z-buffer,[6] ray-tracing,[7] or a context-free spheres algorithm[8] have also been published but they are usually inefficient with respect to drawing speed. Furthermore these methods generally require extensive antialiasing.

VISIBLE PORTION OF TWO INTERSECTING ATOMS

When two spheres intersect (Figure 1), a circle is formed that is projected onto the viewing plane as an ellipse, possibly reduced to a segment. When this intersection circle is

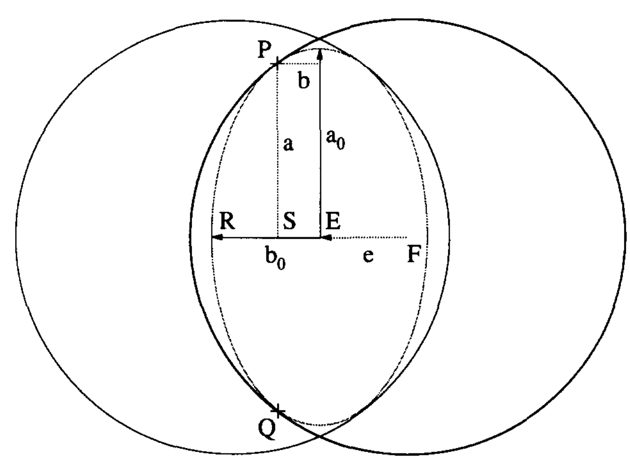

Figure 1. The circle of intersection between back sphere (left) and front sphere (right) is projected onto the viewing plane as ellipse E. Contact points P and Q between this ellipse and the disk F limit the visible portion of this circle of intersection

Address reprint requests to Dr. Cense at the Ecole Nationale Supérieure de Chimie de Paris, 11, rue Pierre et Marie Curie, 75231 Paris Cedex 05, France.
Received 7 February 1991; accepted 19 February 1991

partly visible, the visible part of the front atom is bounded by the corresponding arc of the projected ellipse. An exact calculation of this intersection is desirable if the algorithm is used to generate frames for animated sequences, as poorly approximated elliptic arcs are disturbing when the sequence is animated.

A common belief, reasserted by Max[3] and Schafmeister,[5] is that the exact solution of the intersection of the ellipse and the circle bounding the front sphere in the viewing plane requires solving a quartic equation. In fact the exact solution involves only simple geometry and trigonometry, as will be shown in this section.

If d is the distance between the centers of spheres F (front) and B (back), of radii r_f and r_b, the distance f between the center of the front sphere and the center E of the circle of intersection is

$$f = (d^2 + r_f^2 - r_b^2)/(2d) \tag{1}$$

If θ is the angle between the line joining the centers of the spheres and the viewing plane, the distance in the viewing plane between the center of the front sphere and the center of the circle of intersection is

$$e = f\cos(\theta) \tag{2}$$

When $\theta = 0$, the intersection circle is projected onto the viewing plane as a line. When $\theta \neq 0$, the intersection circle is projected onto the viewing plane as an ellipse of semiaxes a_0 and b_0, of equation

$$a^2/a_0^2 + b^2/b_0^2 = 1 \tag{3}$$

where a_0 is the radius of the circle of intersection

$$a_0^2 = r_f^2 - f^2 \tag{4}$$

$$b_0 = a_0 \sin(\theta) \tag{5}$$

If the circle of intersection is visible, this ellipse is tangent at P and Q to the circle bounding the sphere F in the viewing plane. For these two points

$$(b + e)^2 + a^2 = (b + f\cos(\theta))^2 + a^2 = r_f^2 \tag{6}$$

where a and b are found by solving the set of Equations (3)–(6). The quantity a^2 is extracted from Equations (3) and (5)

$$a^2 = a_0^2 - b^2/\sin^2(\theta) \tag{7}$$

Equation (7) is added to Equation (4):

$$a^2 = r_f^2 - f^2 - b^2/\sin^2(\theta) \tag{8}$$

Equation (8) is subtracted from Equation (6) and expanded:

$$b^2 + 2\,b\,f\cos(\theta) + f^2\cos^2(\theta) - f^2 - b^2/\sin^2(\theta) = 0 \tag{9}$$

Equation (9) is multiplied by $\sin^2(\theta)$:

$$b^2(\sin^2(\theta) - 1) - f^2\sin^4(\theta) + 2\,b\,f\cos(\theta)\sin^2(\theta) = 0 \tag{10}$$

$$f^2\sin^4(\theta) - 2\,b\,f\cos(\theta)\sin^2(\theta) + b^2\cos^2(\theta) = 0 \tag{11}$$

$$(f\sin^2(\theta) - b\cos(\theta))^2 = 0 \tag{12}$$

hence

$$b = f\sin^2(\theta)/\cos(\theta) \tag{13}$$

The circle of intersection will be visible if $a^2 > 0$:

$$r_f^2 - (f\cos(\theta) + b)^2 = r_f^2 - f^2/\cos^2(\theta) > 0 \tag{14}$$

APPROXIMATING THE VISIBLE REGION OF INTERSECTING ATOMS

Typically, the arc of ellipse PRQ is approximated by a circular arc running through P, R, and Q. This approximation fails to give a correct intersection when the front atom is a hydrogen atom bonded to a large atom, for example H–S. In this case the circular arc is made running through R and the extremities of the long axis of the ellipse E.

However, it is more appropriate to approximate the arc of ellipse by cubic Bezier curves. Cubic Bezier curves are used, for example, by PostScript[9] to draw any curve, including circles and ellipses. When control points are chosen judiciously, the cubic Bezier curve is virtually undistinguishable from the circle or the ellipse.

Four control points P, U, V, and Q are needed to define a cubic Bezier curve (Figure 2). The curve is tangent at the endpoints P and Q to the lines PU and QV. The two control points U and V are defined according to the following scheme:

A cubic Bezier curve can be divided into two half-curves PR and RQ, each defined by new control points. If p_0, p_1, p_2, and p_3 are the original control points P, U, V, and Q, the control points p'_i for the first half PR of the curve can be derived from the original control points p_i. The last control point of the first half is given by[10]

$$p'_3 = p_0/8 + 3p_1/8 + 3p_2/8 + p_3/8$$

Taking the axes of the ellipse as reference axes, one has for a symmetrical arc, $p_0 = p_3 = 0$ and $p_1 = p_2$. It follows that if $p_1 = 4 \times p'_3/3$ the cubic Bezier curve controlled by P, U, V, and Q will pass exactly through R.

Points U and V are located on the tangent to the ellipse, which is also the tangent to the circle F, at a distance from P and Q such that $ST = 4 \times SR/3$. As the tangents at P

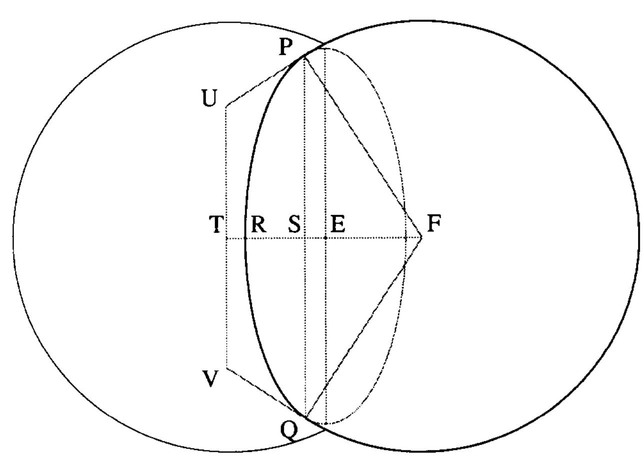

Figure 2. The arc of ellipse PRQ can be approximated by a cubic Bezier curve using control points P, U, V, and Q

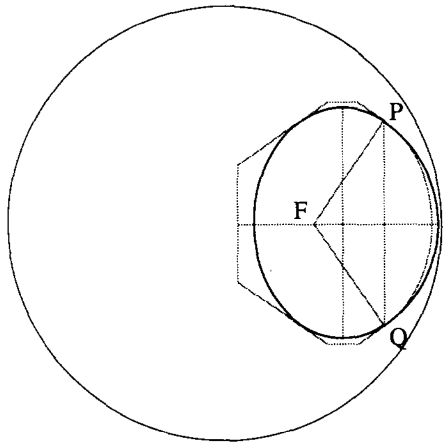

Figure 3. If the front atom is a hydrogen atom bonded to a large atom, the visible portion of the circle of intersection can be approximated by three cubic Bezier curves

and Q to the circle F are simply found from the location of P, Q, and F, and as the distance SR is $b_0 - b$, the points U and V are easily calculated.

$$\begin{aligned} PU &= ST \times PF/PS \\ &= 4\,SR \times PF/3\,PS \\ &= 4\,(b_0 - b)r_f/3\,a \end{aligned}$$

When atom F is a hydrogen atom bonded to a large atom (Figure 3), the arc of the ellipse is approximated by three cubic Bezier curves.

IMPLEMENTATION

To accelerate the drawing, we assume parallel projection and neglect cast shadows. Therefore only one template for each atom type needs be calculated. Since the clipping region is not calculated pixel-by-pixel but rather delineated by a series of curves, the image-generation bottleneck is the transfer operation between the source bitmap (the shaded template) and the destination bitmap (the screen), according to the clipping region.

As bitmap transfer operations are usually coded in ROM or hardware, the drawing speed is largely independent of the exact algorithm used to approximate the visible part of the circle of intersection. If cubic Bezier curves are not available, fast algorithms to draw these curves can be easily implemented.[10]

The algorithm using a circular arc to approximate the ellipse of intersection has been implemented in THINK C for the Macintosh and the results are very encouraging. On a Macintosh-IIfx with a standard 256-color screen (32 shades for each atom) and with a standard display card, a 120-atom molecule containing C, H, O, N, and S (a peptide) is displayed in 1.7 sec. The shading process itself takes 0.2 sec and is no longer required as long as the model is not rescaled. Under the same conditions, a 500-atom molecule is displayed in 5.5 sec.

CONCLUSIONS

The principle detailed in this paper—exact determination of the visible portion of intersecting atoms—allows for rapid generation of space-filling molecular models. It makes use of shaded templates stamped into clipping regions delineated by circular arcs or cubic Bezier curves. As no pixel depth comparison is necessary, all bit transfers are rapidly performed in blocks. The implementation described here is a first step toward truly interactive space-filling molecular models imaging.

REFERENCES

1 Knowlton, K. and Cherry, L. ATOMS—A 3-D opaque molecule system for color pictures of space-filling or ball-and-stick models. *Comp. Chem.* (1977) **1**, 161–166
2 Max, N. L. ATOMLLL—ATOMS with shading and highlights. *Comp. Graphics* (1979) **13**, 165–173
3 Gwillian, M. and Max, N. Atoms with shadows—An area-based algorithm for cast shadows on space-filling molecular models. *J. Mol. Graphics* (1989) **7**, 54–59
4 Cense, J. M. MolDraw: Molecular graphics for the Macintosh. *Tetrahedron Comp. Methodology* (1989) **2**, 65–71
5 Schafmeister, C. Fast algorithm for generating CPK images on graphics workstations. *J. Mol. Graphics* (1990) **8**, 201–206
6 Porter, T. Spherical Shading. *Comp. Graphics* (1978) **12**, 282–285
7 Lauher, J. W. Chem-Ray: A molecular graphics program featuring an umbra and penumbra shadowing routine. *J. Mol. Graphics* (1990) **8**, 34–38
8 Palmer, T. C. and Hausher, F. H. Context-free spheres: A new method for rapid CPK image generation. *J. Mol. Graphics* (1988) **6**, 149–154
9 Adobe Systems Incorporated. *Postscript language reference manual*. Addison-Wesley, Reading, 1985, 117
10 Newman, W. M. and Sproull, R. F. *Principles of Interactive Computer Graphics*. McGraw-Hill, New York, 1983, 329