The Isoluminance Contour Model

Damian M. Conway and Marion S. Cottingham

Computer Science Department,
Monash University,
Victoria 3168.

Abstract.

A standard method of simplifying the task of obtaining a shaded image of a solid object is to represent it by a collection of surface patches or a polyhedron. A smooth shading technique is then applied to each patch/facet to restore smoothness to the image.

This paper introduces an entirely new approach to surface representation: the use of isoluminance contours to represent CSG-type primitive objects. The surface of an object is divided into strips by these contours, such that the area between two adjacent contours has a constant light intensity value. This method does not require any smooth shading technique and takes advantage of the polygon fill hardware facility available in many contemporary graphics devices.

Keywords: computer graphics, isoluminance contours, constructive solid geometry, surface representation.

Introduction.

A surface is often approximated by a collection of facets that are bounded by edges and vertices. Edges are straight line segments and vertices are normally defined in the Cartesian coordinate system, with the x, y and z-coordinates being sufficient to represent a 3-D point. Topological information is required to show how these facets, edges and vertices are connected. This information may be stored explicitly using Baumgart's Winged-edge data structure [Baum75] or can be implied by the order of storage [Cott85, Cott87].

Using this representation for convex surfaces, it is normal to cull backfacing polygons before the rendering step. This is achieved either by the calculation of normals or by determining if a facet has its vertices in a clockwise order. The remaining front-facing facets can then be generated using a smooth shading technique such as Gouraud's or Phong's [Gour71, Phon75].

This paper introduces an entirely new approach to approximating the surfaces of CSG-type primitive objects. These objects are represented by sets of isoluminance contours. Each contour is defined by a collection of 3D Cartesian points. Depending on the type of primitive being represented, these contours are computed by approximation or from exact geometric relationships

The contours are chosen such that the surface bounded by two sets of adjacent contour points has a constant light intensity value. To generate the image, an illumination model is applied to each contour to compute its intensity value and the area enclosed by each contour is rendered using the polygon fill facility which is available in the hardware of most current graphics devices. Hidden surface elimination is achieved by generating contours in order starting with the one furthest from the viewing position and overdrawing each contour, finishing with the nearest one.

This method has several advantages over the polyhedral method. An object can be represented by far fewer contours than facets and these contours can be generated quickly using hardware supported operations. The contours do not require any smooth shading techniques to be applied and there is no need for culling of back-facing facets.

Using CSG methods a complex solid is represented by a collection of simpler solids. These are combined by the Boolean set operations union, intersection and difference. These primitives are typically blocks, cylinders, spheres, and cones. Solid modeling systems use an underlying unambiguous representation of these primitive solids [Requ82]. Polygonal representations are used in at least three contemporary solid modeling systems: Matra's Euclid [Bern75], IBM's GDP [Wesl80] and Cambridge Interactive Systems' Medusa [Geis84]. The isoluminance contour representation is also unambiguous and could be used as an alternative to the polygonal representation.

Reflectance Models.

Non-specular reflection occurs when an object has a dull surface. For a non-specular reflector illuminated by a point light source, the intensity of light (I_d) reflected from a given point on its surface is proportional to the angle of incidence (θ_I) of a light ray at that point. This relationship, known as the Cosine Rule, is given by:

$$I_d \;=\; k_d \,.\, \cos(\theta_I)$$

where k_d is the diffuse reflection coefficient of the surface. The coefficient k_d is proportional to the frequency of light illuminating the object, but in practice this fact is often ignored.

Real world illumination is a vastly more complex phenomenon than is modeled by the Cosine Law. For example, almost all objects exhibit some degree of specular reflection. Specular reflection occurs when a light ray strikes a point on the surface of an object such that the angle of reflection of the ray (θ_R) is approximately equal to the angle between the sight ray and the surface normal (θ_S) at that point. For non-polished surfaces this effect is often approximated by the equation:

$$I_s \;=\; k_s \,.\, \cos^n(\theta_\Delta)$$

where k_d is the specular reflection coefficient of the surface, $\theta_\Delta = \theta_R - \theta_S$ and n is an arbitrary coefficient specifying the "shininess" of the surface. Because of this dependency on θ_S, specular reflection is a viewpoint dependent phenomenon.

Real world objects also exhibit properties such as luminosity, translucency, transparency, surface texture and pattern, anisotrophism and shadowing. These effects may interact in complex and geometrically dependent ways and may also depend on the intensity and wavelength(s) of the incident light. Intensity is in turn determined by Lambert's (inverse square) Law and the physical properties of the medium through which the light is transmitted.

The full isoluminance contour model incorporates some of these illumination effects efficiently and research is continuing on others. However, in the interest of clarity, this paper addresses only diffuse and simple specular reflection.

To produce realistic images conventional CSG representation schemes involving polyhedral models require a smooth shading step in the rendering process. The most commonly used techniques for smooth shading are Gouraud's algorithm [Gour71], which involves linear interpolation of pixel intensities across each face of the polyhedron, or the more sophisticated Phong method [Phon75], which interpolates surface normals. Because isoluminance contours are determined by the actual shading of the object they represent, smooth shading is not required in the rendering process, allowing hardware fill facilities to be used. This results in a significant increase in the effective speed of rendering an object where smooth shading is not supported in hardware.

Isoluminance Contours.

An isoluminance contour is a set of planar points lying on the surface of an object, all of which have the same luminous intensity value. These contours store sufficient information about an object's shape and illumination (ie: reflectance properties) to enable it to be unambiguously rendered.

If the illumination is independent of the position of the viewer, such as for dull surfaces, the contour information is likewise viewpoint independent. Hence altering the viewing position only requires the transformation of the set of contour points to provide new screen coordinates; the illumination information will remain unchanged. The rendering of shiny or metallic surfaces requires specular reflection effects, which are dependent on the viewing position. This results in additional computation whenever the viewing position is changed.

Some applications, such as rendering of mechanical parts, do not require a high degree of realism. However, the speed with which an image can be generated is often an important consideration. Such applications do not require a sophisticated illumination model, and optical effects such as specular reflection and shadowing can be ignored. In such cases, the isoluminance contour model can be used to efficiently render CSG objects with or without specular reflection effects.

Representing a Sphere.

The isoluminance contours of a non-specularly reflective sphere, illuminated by a single point light source[1] positioned above the sphere's "north pole", correspond exactly to the "lines of latitude" on the sphere. Of course, the actual configuration of the light source and sphere is quite independent of any particular coordinate system; the "north" is purely local to the sphere.

The light source vector of an object is a vector specifying the direction of the light source from the object's centre or centroid. To compute the isoluminance contours of a sphere, the light source vector is defined to be "north" for the sphere. The "lines of latitude" on the sphere are then calculated relative to this "north" axis.

For purely diffuse reflection, the intensity assigned to each contour is computed using the Cosine Law. The (point) contour corresponding to the sphere's "north pole" is assigned the maximal intensity, which will generally be a function of the distance from the sphere to the light source. "Latitude lines" have a cosine distribution along the light source vector so the intensity of each subsequent contour can be computed by linear interpolation between the maximal intensity and the minimal (or ambient) intensity value at the "equator" of the sphere. All the contours corresponding to "southern" latitude lines represent points on the surface of the sphere which are not directly illuminated by the light source and hence are assigned the minimal intensity value.

Each contour defines a filled convex 3D polygon called an isoluminance lamina. These polygons are generated using the hardware supported polygon-fill operations of the display device. They are drawn in order of decreasing distance from the viewing point to accomplish hidden surface removal (a technique known as the Painter's Algorithm.) It is possible to find a suitable ordering of polygons to achieve this because no two isoluminance contours can intersect. The intersection of two or more contours would imply by definition that the object had more than one luminous intensity at the point of intersection.

Representing a Generalised Truncus.

The generalised truncus is a family of CSG objects which include all cones, cylinders, discs, rods and right truncii. Consider a generalised truncus with radii R_1 and R_2.

For the case where $R_1 = R_2$ the truncus represents a cylinder, disc or rod, depending on the height of the truncus relative to R_1 and R_2. For the case where one of the radii is zero, the truncus represents a cone. All other instances are truncii, which may resemble frustra of cones, beveled discs or tapered rods, depending on the relative height of the truncus.

A truncus can be decomposed into an infinite number of quadrilateral laminae each of which can be defined by specifying its four vertices (see

[1] Note that the method may be generalised to an arbitrary number of point or linear light sources by computing the set of composite illumination contours representing the cumulative illumination effect of the individual light sources, but the loss of symmetry often greatly increases the computation costs. Generally speaking a single point source has been found sufficient to produce easily recognizable and unambiguous renderings of CSG type objects.

appendix for calculation details.)

These laminae provide a suitable approximation to the isoluminance contours of a truncus, provided the light source is far enough away from the truncus. In addition to these laminae the surfaces of the ends of the truncus form two addition isoluminance contours.

The calculation of illumination intensities corresponding to each illumination contour is not as straightforward as in the case of the sphere (a reflection of the lower degree of rotational symmetry of the truncus.) For each contour, the cosine of the angle between the unit normal (N_i) of the portion of the truncus's surface represented by the contour[2] and the unit light source direction vector must be calculated. Given this cosine, an intensity between the maximal and minimal values can be interpolated using the Cosine Rule (see appendix.)

When the intensities have been computed for each contour, the truncus is rendered as filled polygons using the Painter's Algorithm, exactly as in the case of the sphere.

Isoluminance Contours for Specular Reflectors.

The isoluminance contour model can be used to efficiently render surfaces which exhibit a limited form of specular reflection. Although representation of "mirrorlike" surfaces is not feasible[3], the rendering of objects with metallic or glossy finishes is relatively straightforward. Specular reflection is dependent on the relative positions of the light source, the surface and the viewing position. The calculation of precise isoluminance contours for a specularly reflective surface can be an expensive point-by-point exercise if linear algebra is used. However, a good approximation of these contours can be made for all CSG primitives using a single construct: the "virtual" light source [cf: Blin76].

For simple objects such as the sphere, cone, cylinder and polyhedron, the isoluminance contours caused by specular reflection are approximately the same as the diffuse reflection contours for the same object illuminated from a different light source position. The position of this "virtual" light source depends on the viewing position, the position of the light source and the centre of the object.

When the position of the virtual light source is known, the isoluminance contours can be calculated exactly as if the object were diffusely reflective and illuminated from the virtual light source position. The resulting set of contours approximates the specular isoluminance contours for the object. The "true" isoluminance contours for an object will in general be foreshortened towards the

[2] Note that, except for the two ends of the truncus, N_i is not in general identical to the normal of the lamina itself.

[3] "Mirrorlike" surfaces often exhibit highly complex and asymmetrical intensity contours, the distribution of which is a function of the surface's shape and of the geometry and composition of its environment. Hence, except in the most trivial cases, computing the isoluminance contours of such a surface may easily become *less* efficient than ray-tracing it.

viewing position compared with those produced using this method. This effect is negligible except when the distance to the light source is small with respect to the size of the object.

As in the preceding diffuse reflection examples, an intensity is assigned to each contour. However instead of the Cosine Law, the $\cos^n$ equation (or some more sophisticated model) is used to produce a more realistic intensity distribution. When intensities have been assigned, the object is rendered in exactly the same manner as before.

Evaluation.

A CSG primitive rendering system based on the isoluminance contour model was implemented on an Apollo DN580 workstation. The following tables summarize the performance of the method:

Isoluminance Contour Model - Sphere				
area of image (pixels)	modeling of sphere (seconds)	display (seconds)	total time (seconds)	rendering speed (pixels/s)
550	1.3	1.8	3.1	180
4500	2.0	2.8	4.8	940
9000	3.2	5.5	8.1	1100
55000	16.2	26.2	52.4	1050
250000	109.2	177.3	286.5	900

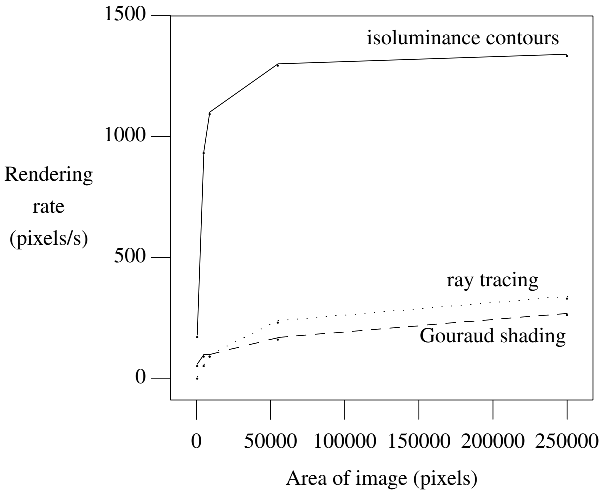
Isoluminance Contour Model - Truncus				
area of image (pixels)	modeling of truncus (seconds)	display (seconds)	total time (seconds)	rendering speed (pixels/s)
1000	1.0	1.2	2.2	450
2000	1.1	1.3	2.4	800
8000	1.4	2.1	3.5	2300
25000	1.9	3.1	5.0	5000
100000	3.0	5.7	8.7	11000

The cost of rendering a sphere using the isoluminance contour model is proportional to the size of the image and proceeds at approximately 1000 pixels per CPU second in this implementation.

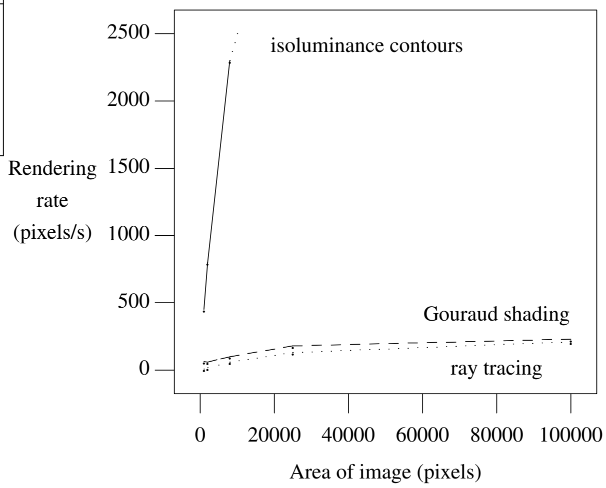
For the case of the cone, the cost of the isoluminance contour method becomes significantly more efficient as the size of the image is increased. This is because the cost of hardware scan conversion of polygons is bound by the number of vertices of the polygon, not the area it covers. Since the truncus is rendered using only quadrilateral polygons the cost of rendering is effectively constant regardless of the size of each polygon (cf: the case of the sphere where the number of vertices required to closely approximate each circular polygon, and hence the rendering cost, is kept proportional to the area that the polygon encloses.)

The following graphs compare the rendering rates for CSG sphere and cone primitives using the isoluminance contour method, Gouraud shading and ray tracing.

Rendering of a Sphere.



Rendering of a Cone.



Gouraud shading was performed using a 70 facet polyhedral approximation for the sphere and a 15 facet approximation for the cone. The ray tracing was set up to calculate only a single intersection per pixel. For all three methods the number of intensities was set at 200 and the cos^n shading model was used.

For the case of the sphere isoluminance contour modeling can render images approximately five times faster than Gouraud shading or simple ray tracing. For small images (area < 10000 pixels) the isoluminance contour method is twice as fast again. For a cone the improvement in speed over Gouraud's method or ray tracing is approximately proportional to the size of the image. For small images isoluminance contour modeling produces comparable images at least six times faster than either Gouraud's method or ray tracing.

Further Work.

Any representation of CSG primitives is, in itself, of limited use. The power of constructive solid geometry lies in the ability to combine such primitives in useful ways. Research is currently continuing into the implementation of Boolean operators (union, intersection and difference) applicable to the primitives presented in this paper, with a view to constructing a full CSG system based on isoluminance contour modeling.

One promising approach involves the clipping of each isoluminance lamina in various ways. For example, an operator to produce the intersection of two primitives would clip the laminae of the first primitive against the volume of the second, retaining those portions of the laminae which are internal to the second primitive, and then likewise clip the laminae of the second primitive against the volume of the first. The union of these two sets of clipped laminae represents the intersection of the two primitives. Union and difference operators can be derived in a similar fashion.

Preliminary investigations indicate that, while there are still some problems to be addressed, such clipping-based operations can be successfully implemented without loss of the significant speed advantages of the method over existing representation schemes.

Conclusions.

The isoluminance contour method provides an accurate and efficient means of computing and storing the information necessary to render CSG primitives when these primitives are illuminated by a single light source. The representation is unambiguous and produces realistic images quickly by taking advantage of hardware facilities common to many graphics devices. Advantage can be taken of the coherence of contour information within an object to enable hidden surface removal using a simple Painter's Algorithm. Storage costs and rendering speed can be accurately controlled by restricting the number of contours used.

Comparison with polyhedral representations shaded using Gouraud's algorithm demonstrates that the isoluminance contour method is considerably cheaper in most cases and produces results of better quality. Where sophisticated optical effects such as transparency, shadowing or mirrorlike reflection are not required, the isoluminance contour method can produce results comparable to ray traced images at a fraction of the cost.

References.

[Baum75] Baumgart, B.G. *A Polyhedron Representation for Computer Vision.* Proceedings AFIPS National Computer Conference 1975, pp 589-596.

[Bern75] Bernascon, Y.J. & Brun, J.M. *Automated Aids for the Design of Mechanical Parts.* Tech. Paper MS75-508, Society of Manufacturing Engineers, 1975.

[Blin76] Blinn, J.F. & Newell, M.E. *Texture and Reflection in Computer Generated Images.* Comm. ACM, Vol. 19, no. 10 (Oct 1976), pp 542-547.

[Cott85] Cottingham, M.S. *A Compressed Data Structure for Surface Representation.* Computer Graphics Forum, Vol.4 (Sept 1985), no.3, pp 217-228.

[Cott87] Cottingham, M.S. *Compressed Data Structure for Rotational Sweep Method.* Ausgraph87 Conference, Perth, W.A., May 4-8, 1987.

[Geis84] Geisow, A.D. *The Medusa Modeling System.* ITCREST Conference, University of Glasgow, Scotland, Aug 27 - Sep 7, 1984.

[Gour71] Gouraud, H. *Computer Display of Curved Surfaces.* Tech. Rep. UTEC-CSC-71-113 (June 1971), Dept. Comp. Sci., University of Utah, Salt Lake City.

[Phon75] Phong, B.T. *Illumination for Computer Generated Pictures.* Comm. ACM, Vol. 18 (June 1975), pp 311-317.

[Requ82] Requicha, A.A.G. & Voelcker, H.B. *Solid Modeling: A Historical Summary and Contemporary Assessment.* IEEE Comp. Gr. & Appl., Vol.2, No.2 (March 1982), pp 9-24.

[Wesl80] Wesley, M.A. *Construction and Use of Geometric Models.* Computer Aided Design, J. Encarnacao ed., Springer-Verlag, N.Y., 1980, pp 79-136.

Appendix - Approximate Isoluminance Contours of a Truncus.

Let the axis of symmetry of the generalised truncus be parallel to the vector **a**. Let the height of the truncus be $|\mathbf{a}|$ and let **a** be "centred" at the point $\mathbf{c}_t$. That is, the centre of the end E_1 (radius R_1) is at the point $\mathbf{c}_t + \frac{1}{2}\mathbf{a}$ and the centre of the end E_2 (radius R_2) is at $\mathbf{c}_t - \frac{1}{2}\mathbf{a}$. Let the single point light source be at the point **l**.

Let $\mathbf{L}_E$ be the projection of the light source direction vector $\mathbf{L}_n$ onto the (parallel) planes containing E_1 and E_2. That is:

$$\mathbf{L}_E = \mathbf{L}_n - \frac{(\mathbf{a} \cdot \mathbf{L}_n) \, . \, \mathbf{a}}{|\mathbf{a}|^2}$$

If $\mathbf{L}_E = \vec{0}$ (ie: $\mathbf{L}_n$ is parallel to **a**), $\mathbf{L}_E$ is redefined to be an arbitrary vector perpendicular to **a**. $\mathbf{L}_E$ is then normalised to produce the unit vector $\mathbf{L}_{En}$.

Let $\{\phi_i\}$ be an N-element subset of the infinite set of laminae of which the truncus is composed. Given $\mathbf{L}_{En}$, the vertices of each ϕ_i can be computed as follows. First $\{\gamma_{1i}\}$ and $\{\gamma_{2i}\}$, the distributions of laminae along $\mathbf{L}_{En}$ in the planes containing E_1 and E_2 respectively are computed using a cosine distribution:

$$\gamma_{1i} = \mathbf{c}_t + \tfrac{1}{2}\mathbf{a} + \cos(\frac{i\pi}{N}) \, . \, R_1 \, . \, \mathbf{L}_{En}$$

$$\text{for } 0 \le i < N$$

$$\gamma_{2i} = \mathbf{c}_t - \tfrac{1}{2}\mathbf{a} + \cos(\frac{i\pi}{N}) \, . \, R_2 \, . \, \mathbf{L}_{En}$$

Next $\{\delta_{1i}\}$, which defines the displacements from the centre of E_1 of the associated laminae vertices, is calculated:

$$\delta_{1i} = R_1 \,.\, \sin(\frac{i\pi}{N}) \,.\, \left(\mathbf{L}_{En} \times \frac{\mathbf{a}}{|\,\mathbf{a}\,|} \right) \quad \text{for } 0 \le i < N$$

The corresponding vectors δ_{2i} on E_2 are then calculated by substituting R_2 for R_1.

Finally the coordinates of the four vertices of the i^{th} lamina are computed:

$$\begin{aligned} \phi_i = \{\, \mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_4 \;:\; & \mathbf{p}_1 = \gamma_{1i} - \delta_{1i}, \\ & \mathbf{p}_2 = \gamma_{1i} + \delta_{1i}, \\ & \mathbf{p}_3 = \gamma_{2i} + \delta_{2i}, \\ & \mathbf{p}_4 = \gamma_{2i} - \delta_{2i} \;\} \end{aligned}$$

Since each lamina is planar, if the distance from the light source to the centre of the truncus is sufficiently large with respect to size of the truncus, the angle of incidence of light at any point on a given laminae is effectively constant. Thus if $|\,\mathbf{l} - \mathbf{c}_t\,| \gg max(\,|\,\mathbf{a}\,|\,, R_1, R_2\,)$, then each ϕ_i is an isoluminance lamina for the truncus.

Under the same assumption there are two other isoluminance contours on the surface of the truncus, namely the surface of the ends E_1 and E_2. The M-sided isoluminance lamina ϕ_{E1} corresponding to E_1 is:

$$\phi_{E1} = \{\; \mathbf{p}_i : \mathbf{p}_i = \gamma_{1i} \pm \delta_{1i}\,;\; 0 \le i < \tfrac{M}{2} \;\}$$

Note that the points $\mathbf{p}_i$ are generated in cyclic order around the edge of E_1 (that is: such that the set of edges $\overline{\mathbf{p}_i \mathbf{p}_{i+1}}$ forms a convex hull of ϕ_{E1}.) Lamina ϕ_{E2} is calculated in the same way.

To compute the luminance value for each isoluminance lamina the angle of incidence, θ_I, at the portion of the surface of the truncus represented by that lamina, must be determined. Let $\mathbf{p}_{14}$ be the midpoint of the line $\overline{\mathbf{p}_1 \mathbf{p}_4}$, where $\mathbf{p}_1$, $\mathbf{p}_4 \in \phi_i$. The unit normal $\mathbf{N}_i$ of that portion of the surface which is represented by the i^{th} lamina is given by the approximation:

$$\mathbf{N}_i = \mathbf{p}_{14} - (\,\mathbf{c}_t - h \,.\, \frac{\mathbf{a}}{|\,\mathbf{a}\,|}\,)$$

$$\text{where:} \quad h = \frac{(R_2)^2 - (R_1)^2}{2\,.\,|\,\mathbf{a}\,|}$$

For the special case of the end laminae ϕ_{E1} and ϕ_{E2}, the unit normals $\mathbf{N}_1$ and $\mathbf{N}_2$ are simply:

$$\mathbf{N}_1 = \frac{\mathbf{a}}{|\,\mathbf{a}\,|} \;,\; \mathbf{N}_2 = \frac{-\mathbf{a}}{|\,\mathbf{a}\,|}$$

Given the normal $\mathbf{N}_i$ of the truncus for the portion of the sphere represented by the i^{th} lamina, under the assumption $|\,\mathbf{l} - \mathbf{c}_t\,| \gg max(\,|\,\mathbf{a}\,|, R_1, R_2\,)$ the cosine of the angle of incidence for that lamina, θ_I, may be computed using the equation:

$$\cos(\theta_i) = \frac{\mathbf{N}_i \cdot \mathbf{L}_n}{|\,\mathbf{N}_i\,|\,.\,|\,\mathbf{L}_n\,|}$$