Constructive Solid Geometry Operators for the Isoluminance Contour Model

Damian Conway, Department of Computer Science,
Monash University, Melbourne,Victoria 3168, Australia

Abstract

Isoluminance contouring [Conw88a, Conw88b] is a representational scheme which supports a rendering technique combining the speed of flat shading and the visual realism of non-recursive ray-tracing. This paper presents extensions to the method which implement a full solid constructive geometry scheme on isoluminance contoured primitives.

1. Introduction

The contouring of luminous intensities within a scene produces a set of isoluminance contours [Conw88a]. An isoluminance contour is defined as a locus connecting adjacent points on the surface of an object, all of which have the same luminous intensity and chromaticity. For diffusely reflective objects this corresponds to finding sets of points on the surface of the object where the surface normal subtends the same angle to a vector in the direction from the point to the light source position $\underline{l}$. For a surface S defined by the equation $f(\underline{p}) = 0$ the k^{th} isoluminance contour for diffuse reflection is given by:

$$\left\{ \ \underline{p} \ : \ f(\underline{p}) = 0 \quad \wedge \quad \frac{\nabla f_{\underline{p}} \bullet (\underline{l} - \underline{p})}{|\nabla f_{\underline{p}}|| \underline{l} - \underline{p} |} \ = \ k \ \right\}$$

where k ranges between 1 (at the point(s) of maximum surface brightness) and 0 (at all points on the surface which are in shadow).

Of special interest are isoluminance contour sets which are planar and convex. Such sets may be rendered rapidly as flat-shaded polygons using hardware support, and exhibit spatial coherence properties which make them suitable for hidden surface removal techniques using simple sorting approaches such as the painter's algorithm [Newe72].

2. Fundamentals of Isoluminance Contouring

This section summarizes briefly the computation of the spatial distribution and intensity values of isoluminance contours for a sphere, polyhedron and the generalized truncus and outlines the method of rendering such objects. A more detailed description of the isoluminance contouring of these primitive objects, including the computation of more general objects such as ellipsoids and oblique frustra of skewed elliptical cones, is presented in [Conw90b].

For simplicity, the objects discussed are assumed to be unpatterned, purely diffuse or purely specular reflectors, illuminated by a single point source of white light. The construction and rendering of patterned objects with complex reflective behaviours and the use of multiple light sources and shadowing is discussed in [Conw90a], but is beyond the scope of this paper.

2.1. Primitives

To visualize the isoluminance contours of a diffusely reflective sphere, imagine the sphere as a planet with its north pole pointing directly towards the point light source. The isoluminance contours of this planetary sphere lie parallel to its lines of latitude and are distributed linearly along its north-south axis. See figure 1.

The surface intensities of the contours decrease linearly from a maximal value at the north pole to a minimal intensity at some terminator latitude in the northern hemisphere. The terminator latitude depends on the radius of the sphere and the distance to the light source. For an infinitely distant light source, the terminator is at the equator.

All contours south of the terminator share the same minimal surface intensity, since they lie in the shadow of the northern latitudes.

To visualize the isoluminance contours of the surface of a polyhedron, imagine a series of spherical shells of linearly increasing radii, concentric about the light source. The intersection of these shells with one face of the polygon creates a set of coplanar loci on the face. These loci are the isoluminance contours of the face.

The luminous intensities of the contours decrease from a maximum at the contour corresponding to the innermost spherical shell intersecting the face, to a minimum at the contour corresponding to the outmost intersecting shell.

The generalized truncus is a cylindrical solid object whose radius may vary linearly along its longitudinal axis. It represents a family of objects including all cones, cylinders, discs, rods and right frustra of cones.

When the dimensions of the generalized truncus are negligible with respect to the distance to the light source, the isoluminance contours of the curved sides of a truncus can be approximated by a set of quadrilaterals.

For a cylindrical truncus, these quadrilaterals are parallel to the axis of symmetry. They are distributed linearly across the cylinder from the edge of the curved side nearest the light source to the edge most distant from it. Hence they divide the cylinder lengthways. Contour intensities range from a maximum for the quadrilateral nearest the light source, to a minimum at some terminator quadrilateral.

When the truncus is a frustrum of a cone, the geometry and distribution of the quadrilaterals is skewed so that they remain linearly distributed across each end of the frustrum, but are no longer parallel.

In both cases there are two other sets of isoluminance contours on the surface of a generalized truncus. These sets represent the planar ends of the truncus and are computed as if they were faces of a polyhedron.

For all purely diffuse primitives the intensity of each isoluminance contour is computed using Lambert's cosine law for diffuse reflection (that is: "the surface intensity of a point is proportional to the cosine of the angle of incidence of light at that point"). Any point on a contour can be chosen as the basis of this computation, since all points are equi-luminous. The resultant surface intensity is then assigned to the entire contour.

In the special case of the sphere, intensities can be linearly interpolated between contours, from a maximum at the point nearest the light source to a minimum for all contours beyond the terminator.

For purely specular objects the position of the light source is shifted to a "virtual" position [Blin76] prior to computation of the contours. This position lies along the bisector of the vectors from the object's centroid to the light source and viewing position respectively. The intensity of each contour is then calculated by application of some specular shading model (for example: [Phon75, Blin77, Cook82]) to any one point on the contour. As in the diffuse case the resultant intensity is assigned to the entire contour.

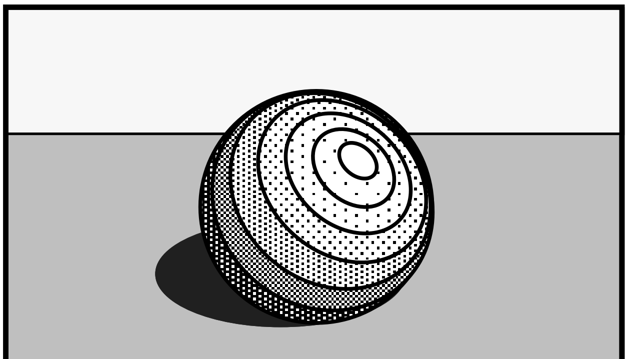

Figure 1: Isoluminance contours of a sphere

2.2. Rendering of Primitives

The rendering of isoluminance contoured primitives is accomplished in three stages. The isoluminance contours are first converted to flat-shaded planar filled polygons. That is, a polygonal approximation of each contour is constructed and the luminous intensity of the contour is mapped to some value in the colour-lookup table of the display device.

The polygons are then sorted by decreasing distance to the viewing position to ensure correct hidden surface removal by the painter's algorithm. This step is omitted if depth buffering is available.

None of the contours of a given primitive overlap or intersect. Each contour set is therefore well-ordered for hidden surface removal purposes.

If an object is viewed from the same side as the light source, the rendering sequence for its contours is always "brightest-to-darkest" (that is: the order in which they were originally generated.) If the object is viewed from the other side the order is reversed. Hence the sorting of the contours of a group of primitives may be accomplished in linear time using a merge sort.

Complications in ordering may arise due to cyclic overlap of individual contours produced by the union of three or more different primitives (see section 5). In such cases, one of the contours is split (as in [Newe72]) so as to resolve to unsortable geometry.

Finally the viewing transform is applied (in hardware if possible) and the sorted list of polygons is passed to the display device for scan-conversion and display.

The rendering algorithm is extremely fast (typically at least five times faster than non-recursive ray-tracing [Conw88b]) since the only display operation required is the scan-conversion of flat-shaded convex polygons, which is carried out in hardware. No point-by-point operations such as smooth-shading or ray-casting are performed.

3. Constructive Solid Geometry.

Isoluminance contouring can be extended to a full constructive solid geometry (CSG) model by introducing a set of Boolean operators over the domain of contour-representable objects. These operators are used to combine primitives in various ways to produce more complex structures.

Three operators are introduced:

- difference (**AND NOT**),
- union (**OR**)
- intersection (**AND**).

So that these operators may be applied to composite objects as well as primitives it is necessary that, given one or more sets of non-intersecting convex planar contours, these operators produce another set of contours with the same properties.

Given that all isoluminance contour primitives are composed exclusively of non-intersecting convex planar contours, the above condition enables scenes constructed using the CSG operators to be rendered using the same technique used for rendering individual primitives.

Unlike other CSG approaches [Requ77, Athe83, Laid86] , isoluminance contour operators are applied in object space (that is, during definition of the object) rather that in image space (during rendering.) This has the advantage that the operators need only be applied once, regardless of the number of times the object is eventually rendered.

All isoluminance contour CSG operators are based on three-dimensional clipping of the individual contours in each object against the volume of another object.

To clip the polygonal approximation ($\mathcal{P}_\phi$) of a contour ϕ against a volume $\mathcal{V}$, the volume is first projected into the plane of ϕ and a polygonal approximation $\mathcal{P}_\mathcal{V}$ is constructed. The 2D Sutherland-Hodgman clipping algorithm for convex polygons [Suth74] is then applied as follows:

Each edge of $\mathcal{P}_\phi$ is clipped against the lines containing the edges of $\mathcal{P}_\mathcal{V}$. For the union and difference operators the parts of $\mathcal{P}_\phi$ *outside* $\mathcal{P}_\mathcal{V}$ are retained. For the intersection operator, those parts of $\mathcal{P}_\phi$ *within* $\mathcal{P}_\mathcal{V}$ are retained.

The appropriate concatenation of contour sets clipped in this way forms a composite object, which retains the required property that none of its contours intersect. Sections 4 to 6 give specific techniques for the three constructive operators.

4. The Difference Operator

The isoluminance contour difference operation, $\mathcal{A} - \mathcal{B}$, is implemented by clipping the contours of object $\mathcal{A}$ against the volume of object $\mathcal{B}$. Note that this clipping volume is the total theoretical volume of the primitive components of $\mathcal{B}$, not merely the region enclosed by its set of isoluminance contours. Since the contours of $\mathcal{A}$ are planar, this problem may be reduced, as described above, to that of finding the difference of two 2D polygons: a contour from $\mathcal{A}$ and the projection of $\mathcal{B}$ into the same plane.

Once every contour of $\mathcal{A}$ has been clipped (retaining that portion of the contour external to the volume of $\mathcal{B}$), the set could be rendered as outlined in section 2.2, to produce the composite object $\mathcal{A} - \mathcal{B}$.

Note however that, although the geometry of the composite object is now correct, the shading of its surface (as encoded by its isoluminance contours) may be wrong, since the contours comprising the composite object $\mathcal{A} - \mathcal{B}$ were originally computed for object $\mathcal{A}$, which had a different geometry (assuming the difference operation was non-trivial.)

An additional set of contours must therefore be computed to correctly shade the new surface created by the "removal" of $\mathcal{B}$. This surface may be divided into sections which are purely concave, planar or convex:

4.1. Concave difference

In the concave case, the additional isoluminance contours required may be computed directly, by considering those portions of the contours of object $\mathcal{B}$ which lie on the new face of $\mathcal{A} - \mathcal{B}$. These portions of $\mathcal{B}$*'s* contour set will map the isoluminant points on the new surface, forming a collection of isoluminance contours. These contours are constructed by clipping the contours of $\mathcal{B}$ against the volume of $\mathcal{A}$, retaining those portions of $\mathcal{B}$'s contours *internal* to $\mathcal{A}$.

The intensities of these new contours may be found by inverting the surface normal of the relevant portion of $\mathcal{B}$'s surface at some point on each contour . The luminous intensity at that point is then recomputed using the inverted surface normal and the surface characteristics of object $\mathcal{A}$. Figure 2 illustrates the technique.

Note that, unlike other isoluminance contours, these contours are rendered as *unfilled* polygons, since they bound a concavity rather than filling a convex region.

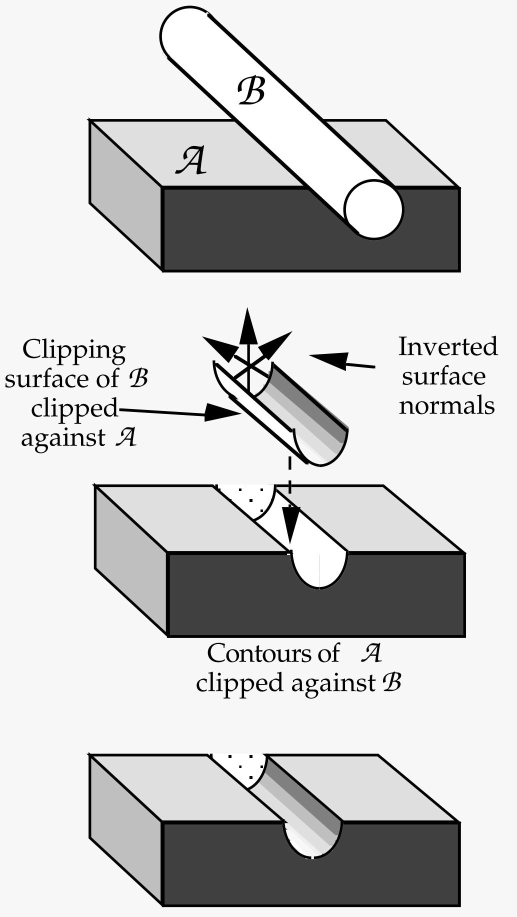

Figure 2: Concave difference

4.2. Planar difference

New planar faces result when object $\mathcal{B}$ is a polyhedron. As for the concave case, the additional contours required to correctly render the new planar surface are computed using the contours of $\mathcal{B}$ and the surface characteristics of $\mathcal{A}$. Those portions of $\mathcal{B}$'s contours interior to the volume of $\mathcal{A}$ form the contours of the new planar surface and are once again found by clipping against the volume of $\mathcal{A}$.

The luminous intensity of these clipped contours is found, as before, by inverting the normal of the associated isoluminance contours and recomputing their brightness using the surface characteristics of object $\mathcal{A}$. This is summarized in Figure 3.

Note that, unlike the preceding concave case, in the case of planar difference, the new surface is rendered as a *filled* polygon.

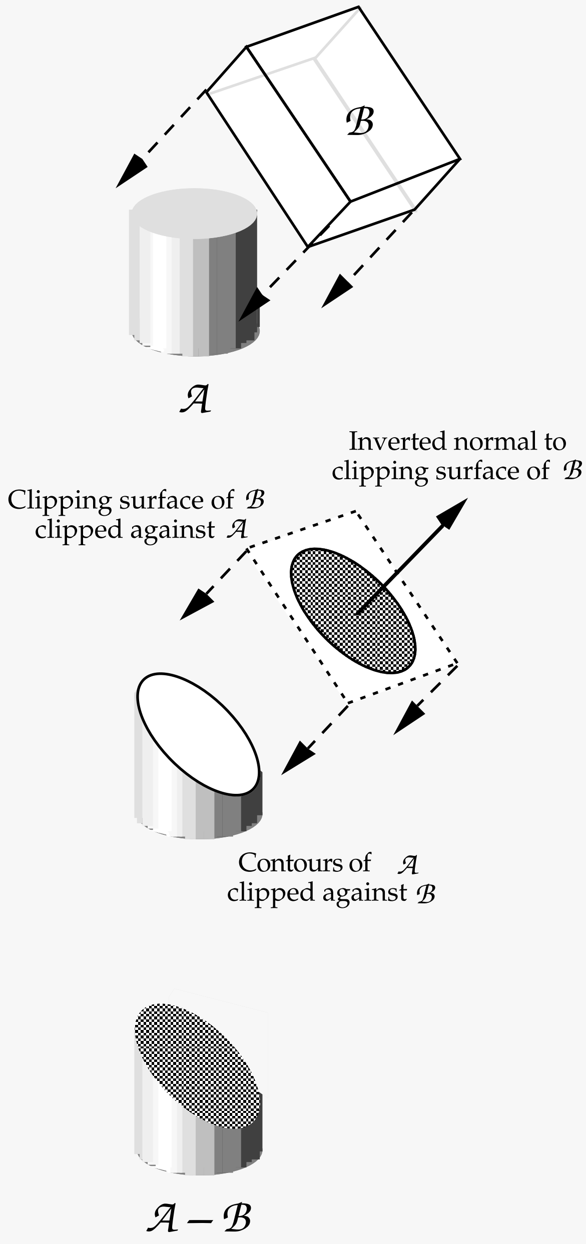

Figure 3: Planar difference

4.3. Convex difference

In any difference operation resulting in a new convex face, the volume $\mathcal{B}$ being subtracted must itself be concave. Since all primitives in the isoluminance contour model are convex, $\mathcal{B}$ must be a compound object, resulting from the subtraction of a convex volume $\mathcal{C}$ from some object $\mathcal{D}$. Hence:

$$\begin{aligned}\mathcal{A} - \mathcal{B} &= \mathcal{A} - (\mathcal{D} - \mathcal{C}) \\ &= (\mathcal{A} - \mathcal{D}) \cup (\mathcal{D} \cap \mathcal{C})\end{aligned}$$

In other words, a difference operation resulting in a new convex surface is equivalent to a difference operation resulting in a concave or planar surface, followed by a union with some convex object which resulted from an intersection (see Figure 4.) The computation of concave and planar differences is covered in sections 4.1 and 4.2 above, and the union and intersection of objects in sections 5 and 6 below.

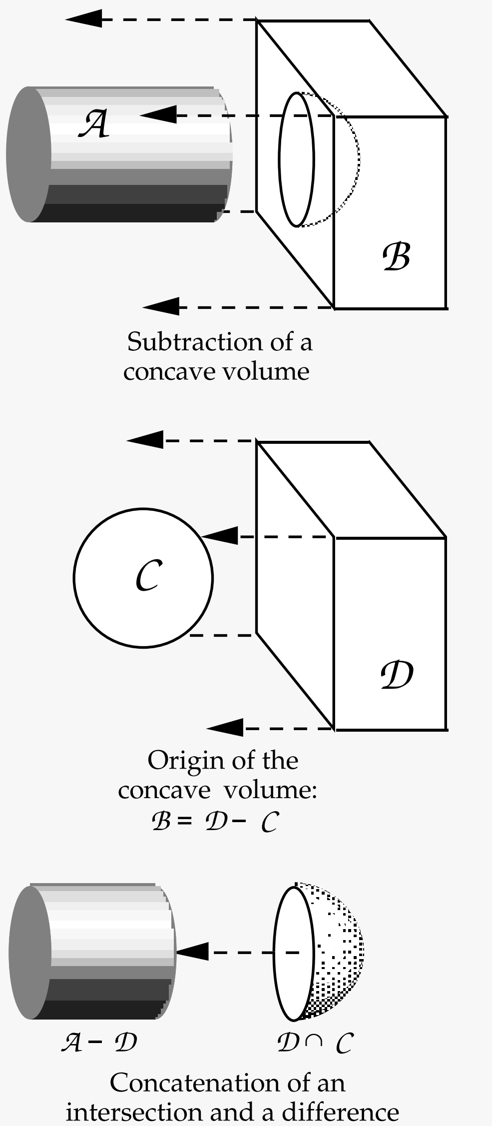

Figure 4: Convex difference

In practice however, not all the steps in the concave/planar difference are required. It suffices to perform the initial clipping of $\mathcal{A}$'s contours. The contours of the new surface created by the difference $\mathcal{A} - \mathcal{D}$ need not be computed, since they will later be totally obscured by the intersection component $\mathcal{D} \cap \mathcal{C}$.

Likewise, no consideration need be given to the portion of the surface of the intersection which adjoins the difference, since it too will be invisible.

Finally, since the volumes $\mathcal{A} - \mathcal{D}$ and $\mathcal{D} \cap \mathcal{C}$ are guaranteed to be disjoint, it suffices to concatenate the two contour sets in order to produce their union.

5. The Union Operator

Given the general difference operator described above, it is simple to construct a union operation which produces non-intersecting contour sets, using the observation that the union of $\mathcal{A}$ and $\mathcal{B}$ is the sum of the volume of $\mathcal{A}$ and that part of the volume of $\mathcal{B}$ outside $\mathcal{A}$:

$$\mathcal{A} \cup \mathcal{B} = \mathcal{A} \textbf{ OR } (\mathcal{B} - \mathcal{A})$$

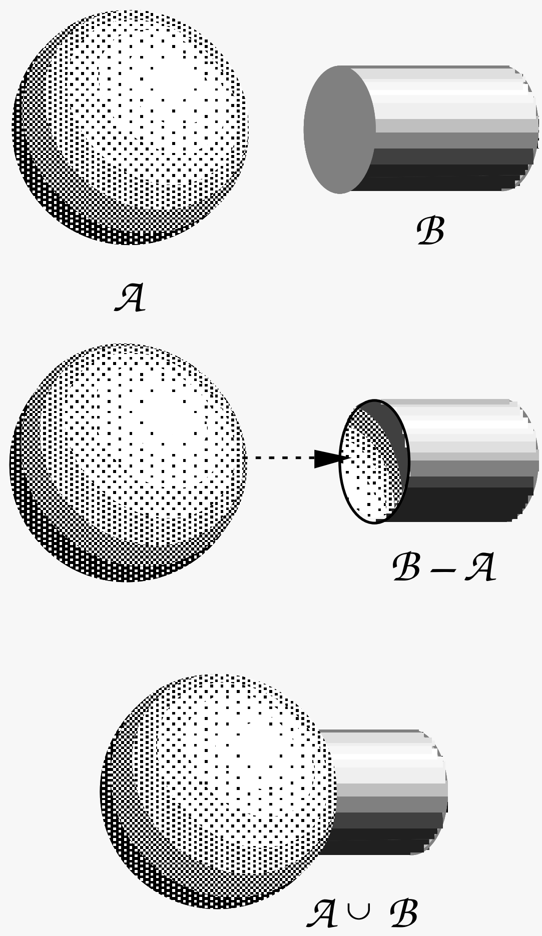

Figure 5: Union via difference

Since the volumes $\mathcal{A}$ and $\mathcal{B} - \mathcal{A}$ are mutually exclusive, their concatenation forms a union satisfying the requirement that no contours in the object intersect. Figure 5 summarizes the technique.

It should also be noted that the new face created by the difference operator is always internal to the composite object and therefore never visible. Hence the calculation of additional contours for that surface may be omitted from the difference operation within a union operation.

6. The Intersection Operator

The intersection operator $\mathcal{A} \cap \mathcal{B}$ may be derived from the observation that intersection creates no new surfaces. Every point on the surface of $\mathcal{A} \cap \mathcal{B}$ derives either from the surface of object $\mathcal{A}$ or from the surface of object $\mathcal{B}$.

The intersection of objects $\mathcal{A}$ and $\mathcal{B}$ is the union of that portion of $\mathcal{A}$ internal to the volume of $\mathcal{B}$ and that portion of $\mathcal{B}$ internal to the volume of $\mathcal{A}$.

To form the intersection of contour sets representing objects $\mathcal{A}$ and $\mathcal{B}$, each contour in $\mathcal{A}$ is clipped against the volume of $\mathcal{B}$, retaining those portions of the contours which are interior to $\mathcal{B}$'s volume.

Those clipped contours which do not touch the surface of volume $\mathcal{A}$ are discarded, since they will not contribute to the surface of the intersection. That is, only those contours of $\mathcal{A}$ which retain some of their original edges after the clipping phase are used to construct the intersection. These contours represent the portion of the surface of $\mathcal{A}$ which is interior to the volume of $\mathcal{B}$.

The clipping and discard phases are repeated, clipping contours from $\mathcal{B}$ against the volume of $\mathcal{A}$. These contours represent the portion of the surface of $\mathcal{B}$ which is interior to the volume of $\mathcal{A}$.

When the two resultant contours sets are concatenated, they correctly encode the geometric and illumination properties of the surface of the composite object because all visible surface portions of the contours sets $\mathcal{A}$ and $\mathcal{B}$ are retained. This is summarized in Figure 6.

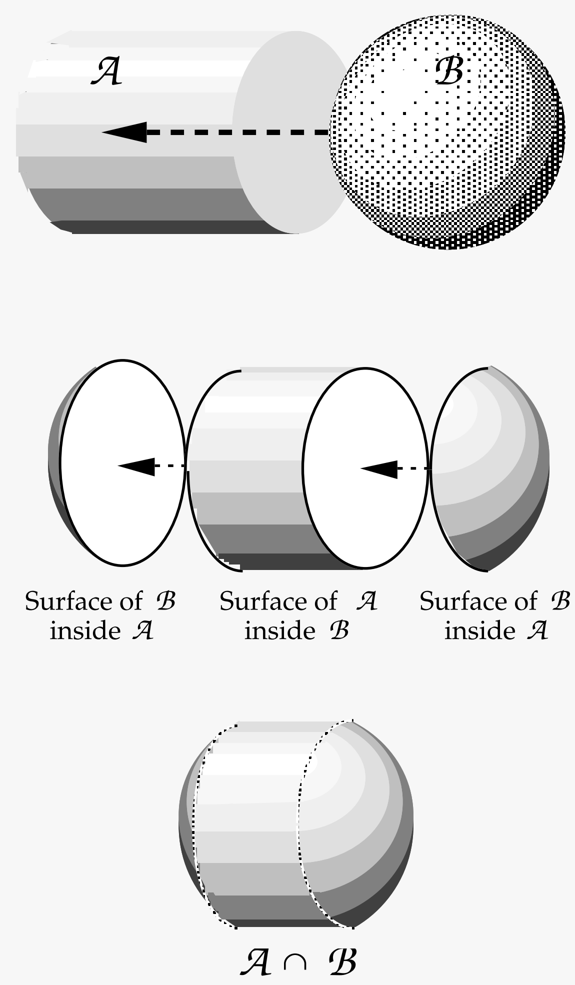

Figure 6: Intersection via union

7. Conclusion

Isoluminance contours provide a means of accelerated rendering in a constructive solid geometry scheme. The speed of the rendering method derives from the use of a representation of primitives based on flat-shaded polygons which nevertheless encode a smooth intensity distribution across the surface of the object. This representation eliminates the need for smooth-shading such as in Gouraud or Phong shading, or for point-by point computation, as in ray-tracing.

The representational power of the scheme is greatly enhanced by the definition of Boolean constructive solid geometry operators based on polygon clipping. The operators are designed such that they operate in object space, producing only sets of planar, non-intersecting, flat-shaded polygons which may be rendered quickly using the same algorithm as for isoluminance contoured primitives.

References

[Athe83] Atherton, P. R., *A Scan-line Hidden Surface Removal Procedure For Constructive Solid Geometry,* Computer Graphics, vol. 17, no. 3, pp. 73-82, ACM, July 1983.

[Conw88a] Conway, D.M. & Cottingham, M.S., *The Isoluminance Contour Model*, Proc. AUSGRAPH '88, Melbourne Australia, July 4-8, 1988, pp. 43-50.

[Conw88b] Conway, D.M. & Cottingham, M.S., *A Method of Rendering CSG-type Solids Using Isoluminance Contours*, Technical Report 88/111, Dept. Computer Science, MonashUniversity, Victoria Australia, June 1988

[Conw90a] Conway, D.M., *An Improved Shading Model for Isoluminance Contouring,* Submitted to IEEE Computer Graphics and Applications.

[Conw90b] Conway, D.M., *Accelerated 3D Rendering using Isoluminance Contouris,* PhD thesis.

[Laid86] Laidlaw, D. H., W. B. Trumbore, and J. F. Hughes, *Constructive Solid Geometry for Polyhedral Objects,* Computer Graphics, vol. 20, no. 4, pp. 161-170, ACM, 1986.

[Newe72] Newell, M. E., R. G. Newell, and T. L. Sancha, *A Solution to the Hidden Surface Problem,* Proceedings ACM National Conference, pp. 443-450, ACM, 1972.

[Suth74] Sutherland, I.E. and Hodgman, G.W., *Reentrant Polygon Clipping,* Commumications of the ACM, vol. 17, no. 1, pp 32-42, ACM, January 1974

[Requ77] Requicha, A.A.G. & Voelcker, H.B., *Constructive Solid Geometry,*' Technical Memo 25, Production Automation Project, University of Rochester, Rochester N.Y., November 1977