Shadow Volume Generation from Free Form Surfaces *

Gregory Heflin and Gershon Elber
Computer Science Department
University of Utah
Salt Lake City, UT 84112 USA

Abstract

The generation of shadows has occupied the computer graphics community for some time. Several approaches have been successfully developed, but many except ray tracing assume a polygonal approximation of the model.

In this paper, an approach is presented that allows one to compute shadow volumes directly from free form models and exploit them for the generation of shadows using a Z-buffer based renderer. A polygonal approximation is not required for either the construction of the shadow volume or for the rendering process.

Key Words: Computer graphics, shadows, shadow volumes, parametric surfaces.

1 Introduction

Classifications of existing shadow rendering techniques have been presented by several authors [Bergeron 1986, Crow 1977, Max 1986]. The techniques can be categorized into the following six categories.

1. **Scan-line shadow generation.** Comparison is done between all models to determine pairs that can interact to produce shadows [Appel 1968, Bouknight 1970, Nishita 1991]. These precomputed relations are used to produce shadows during scan line rendering.

2. **Two-pass model-precision approach.** Models are divided into visible and hidden polygonal regions as viewed from the light source [Atherton 1978]. All the shadowed regions are tagged once and can then be rendered from any desired viewpoint.

3. **Shadow volume approach.** Volumes are generated that enclose shadowed regions of space [Bergeron 1986, Chin 1989, Crow 1977, Fuchs 1985, Max 1986, Nishita 1987]. The boundary surfaces of the shadow volumes are also processed by the Z buffer scan line renderer. Shadow surfaces in the z-list [Atherton 1981] in front of a visible surface are examined to determine if the visible surface is also in shadow. This is a common approach to shadow rendering. Alternatively, the shadow volumes may be used in the first pass of method 2, to classify and tag shadowed regions.

*This work was supported in part by DARPA (N00014-91-J-4123) and the NSF and DARPA Science and Technology Center for Computer Graphics and Scientific Visualization (ASC-89-20219). All opinions, findings, conclusions or recommendations expressed in this document are those of the authors and do not necessarily reflect the views of the sponsoring agencies.

4. **Z-buffer approach.** Depth information is computed and stored in a z-buffer for both the eye and light source viewpoints [Reeves 1987, Williams 1978]. The eye depth values are transformed to the light source view space and depths are compared. If the transformed depth of the point to be rendered is further from the light than the value recorded in the light source z-buffer, then it is in shadow.

5. **Radiosity.** Diffuse light is modeled using techniques from heat transfer theory [Cohen 1985]. Soft shadows are handled particularly well through accurate modeling of diffuse global illumination in polygonal environments.

6. **Ray Tracing.** Rays are traced from the eye through each pixel with shading calculations performed for each surface encountered [Cook 1984, Joy 1988, Kajiya 1982, Nishita 1990, Toth 1985, Whitted 1980, Woodward 1989]. Shadow rays are traced from the surfaces to each light source to determine shadowed regions.

Ray tracing currently provides the most realistic model for shadow generation in environments consisting of free form surfaces, but even direct ray tracing of free form surfaces remains an active and difficult area of research [Joy 1988, Kajiya 1982, Nishita 1990, Toth 1985, Woodward 1989]. In addition, the illumination calculations that ray tracing employs are computationally expensive. Techniques 2 through 5 have an additional computational advantage over ray casting approaches by providing view independent and global shadow representations. This allows one to render scenes from any viewpoint without the need to redetermine the underlying shadow representation, possibly exploiting the use of a hardware renderer. Different approaches for the direct rendering of free from surfaces with shadows, that do not use raytracing, have been examined elsewhere [Nishita 1991, Reeves 1987, Williams 1978].

1.1 Shadow Volume Applications

Shadow volume techniques have been used for near real-time rendering of shadows. Polygonal shadow volumes are represented using BSP trees in many of these approaches to improve performance [Chin 1989, Fuchs 1985]. The shadow volume technique fits well into existing scan line rendering methods and can be implemented with existing hardware approaches. Shadow volumes are also being used in the modeling of atmospheric effects using the notion of a light volume that is the complement of the shadow volume [Max 1986, Nishita 1987].

2 Background

Determining shadow volumes has applications in various fields. In computer graphics, it can make shadow computation a simpler task. In computer aided design and robotics, it can provide cues for accessibility and machinability when the light source direction is considered the direction of access.

A technique for direct generation of shadow volumes from a set of, possibly trimmed, free form surfaces is presented in this paper. A scan line Z buffer was enhanced to support shadow rendering using shadow volumes. All images in this paper were created using this renderer. Methods to render free form surfaces directly, without the need for a polygonal approximations, are under active research, and shadow volumes could easily fit into these approaches [Elber 1992b, Lane 1980, Nishita 1991, Schantz 1988]. In the derivation presented here, light sources are assumed to be point sources at either finite or infinite distances. The following definition of a *model* is used throughout this paper.

Definition 1 *A* **model** *is a set of, possibly trimmed, parametric surfaces with topological surface adjacency information stored explicitly or implicitly in the representation. Each surface of the model is*

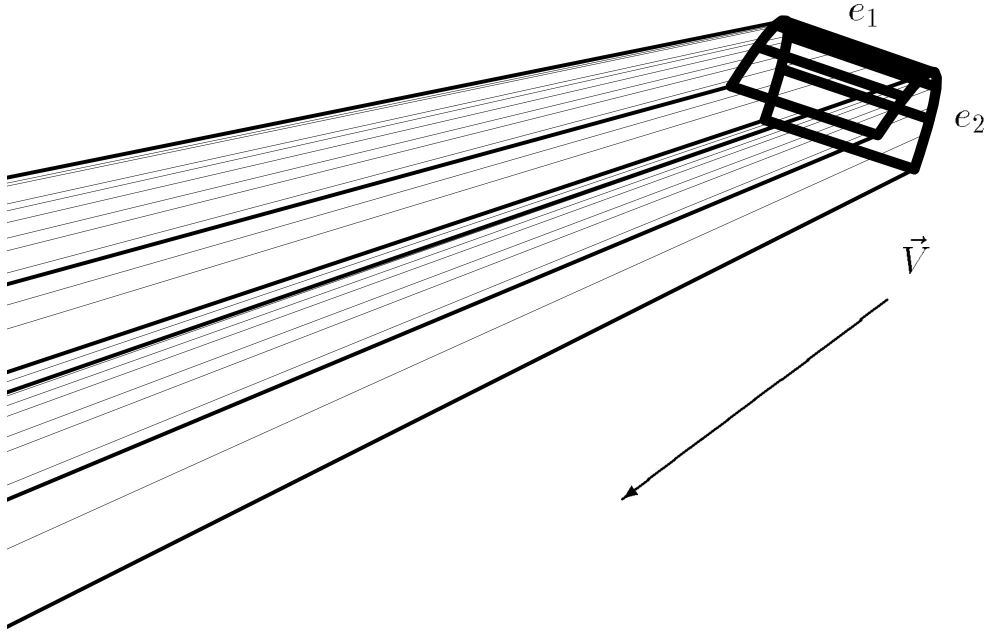

Figure 1: An open model consisting of a single surface with shadow volumes cast from all surface boundaries and silhouette edges. e_1 is a silhouette edge and e_2 is one of four boundary edges.

oriented so that the surface normals point outward. A model is considered closed if it dichotomizes the Euclidean space into regions that are inside and outside the model, and is considered open otherwise.

For example, models used in our implementation were developed within the Alpha_1 geometric modeling environment[EGS 1992] and consist of a set of, possibly trimmed, NURBs surfaces with surface adjacency information stored explicitly as the edges that are shared between two neighboring surfaces.

The addition of shadows to rendered images provides critical cues in determining relative positioning of models within a scene. The top left color image (Fig. 5) gives an example of a chess pawn in a scene rendered with shadows. Generating shadows within a scan line renderer increases the image quality considerably. In addition, the created shadow volumes are view independent as mentioned previously.

Direct generation of shadow volumes without polygonal tessellation has several advantages. It alleviates aliasing effects in the rendered images caused by polygonal approximations, and reduces the sometimes massive memory requirements associated with the polygonal technique. Furthermore, the accuracy of the shadow volume is not bound by a global polygonal approximation and can be adaptively computed. The polygonal shadow volume approach also requires the determination of adjacency information for locating silhouette edges. Surface topological adjacency information is assumed to be stored (Definition 1) in the model and need not be computed.

2.1 Shadow Volume

Our model for *shadow volumes* can now be formally defined.

Definition 2 *A* **shadow volume** *is a sub-region of the Euclidean space that is occluded from a* light source *by a* model*. The volume is delineated by a boundary that partitions the Euclidean space into shadowed and unshadowed regions. The boundary is oriented so that its normals point outward from the shadowed region. The direction opposite the normal is called the* **occlusion direction** *and is denoted $\vec{O}$.*

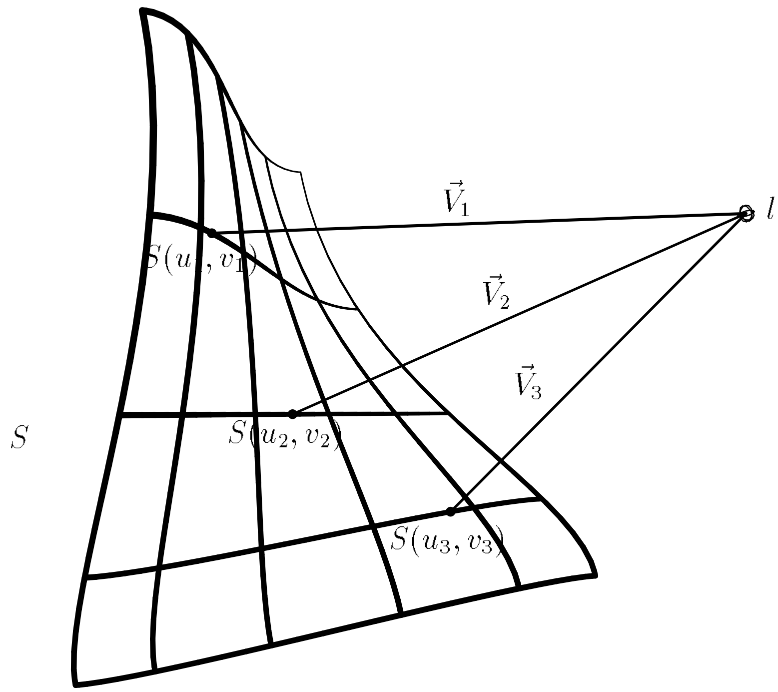

Figure 2: Let l be a point light source at a finite distance from surface S. Then $\vec{V}$, the viewing direction, varies across the surface S or $\vec{V}(u,v) = S(u,v) - l$.

Two types of curves are of interest when attempting to compute shadow volumes, the surface *silhouettes* and the surface boundaries that are not shared by any other surface, the *unshared surface boundary edges*. We refer to these two curve types as the surface *contours*. The shadow volume boundaries correspond directly to the contours of the model cast in the direction opposite the direction of the light source, forming ruled surfaces. The view direction from the light source is referred to as $\vec{V}$. Figure 1 shows an example of an open model consisting of a single surface with shadow volume boundaries cast from a silhouette curve (e_1 in Fig. 1), and from an unshared surface boundary edge (e_2 in Fig. 1). We assume that all surfaces are C^1 continuous. Those that are not can be subdivided in such a way that each resulting surface is C^1 continuous.

Let $S(u,v)$ be a regular [doCarmo 1976] C^1 continuous parametric surface. Then

$$\vec{n}(u,v) = \frac{\partial S}{\partial u} \times \frac{\partial S}{\partial v}, \tag{1}$$

is the unnormalized normal surface of S. The silhouettes of the surface, S, viewed from direction $\vec{V}$ are the solutions to the following equation in two variables,

$$\vec{V} \cdot \vec{n}(u,v) = 0. \tag{2}$$

If the light source is at a finite distance then the view direction varies across the surface and $\vec{V}$ becomes a function of u and v,

$$\vec{V}(u,v) = S(u,v) - l, \tag{3}$$

where l is the location of the light source (see Fig. 2). Equation (2) now becomes

$$\vec{V}(u,v) \cdot \vec{n}(u,v) = 0. \tag{4}$$

Silhouettes may also occur along edges that are shared by two surfaces resulting from a Boolean operation that trims the two intersecting surfaces. A silhouette along the shared edge occurs when one surface sharing the edge is front facing while the other is back facing.

3 Free Form Shadow Volume Computation

Construction of shadow volumes consists of two major tasks, contour extraction and casting. Closed models that partition the Euclidean space into inside and outside regions require only the detection of contours corresponding to silhouettes since surface boundaries are always shared. We first examine the generation of shadow volumes for a single surface and then examine their generation for arbitrary sets of surfaces that include the possibility of silhouettes along shared edges.

3.1 Intra-Surface Contour Extraction

Contours corresponding to silhouettes and surface boundaries must be extracted from the model. Extraction of the surface boundaries is straightforward. Silhouettes that are interior to a surface can be extracted by finding the zero set of Equation (2) or Equation (4) above. One approach [Elber 1992a] determines solutions within a desired tolerance, by symbolically computing the scalar field expressed in these equations and uses root finding techniques to find the zero set. Symbolically computing Equation (2) or (4), results in the need to find the zero set of a bivariate function. The zero sets are computed using a subdivision approach [Elber 1992a]. This approach was found to be extremely robust for silhouette extraction.

The silhouette curves that are extracted do not lie along isoparametric curves of the surface, in general. To extract these silhouettes within a desired tolerance many implementations currently produce piecewise linear curves. This is undesirable since memory requirements can be large and the approximation can again result in aliasing effects in the rendered image. We are currently examining the use of data reduction techniques [Lyche 1987] to increase the order of the extracted silhouettes to significantly reduce their size and alleviate these problems.

The extracted contours are then cast in the view direction to produce ruled surfaces that form the boundaries of the shadow volume. If the light source is at infinity the ruled surfaces degenerate to a simple extrusion. Otherwise, $\vec{V}$ is a function of u and v (Equation (3)) and the casting direction varies along the contours. The resulting cast volume is infinite but can be clipped against the viewing frustum making it representable. The contours of a model always form closed loops. In our implementation contours are extracted from each surface of the model and would need to be connected before casting to yield true shadow volumes. Connecting the contours into closed loops may be necessary in some applications, but for rendering shadows a simpler approach can be taken. The notion of shadow surfaces can be introduced that parallels the use of shadow polygons in the polygonal shadow volume approach. Using this notion, a shadow surface is cast from each extracted contour, without the need to combine the contours into closed loops.

As discussed elsewhere [Bergeron 1986], silhouettes and boundaries produce different occlusion effects. That is, shadow surfaces corresponding to silhouettes delineate regions that are doubly in shadow while those corresponding to boundaries delineate single shadow regions. A 2-D example is shown in Fig. 3. Either each shadow surface corresponding to a silhouette can be dumped twice or it can be tagged with an occlusion count of two. Each shadow surface carries its origin information, including both the geometry surface that it was cast from and the light source to which it is associated.

3.2 Inter-Surface Contour Extraction

Shadow volume generation for geometrically closed models follows closely that of the intra-surface contour extraction discussed above. Contours along unshared surface edges no longer need to be detected since none exist, while contours corresponding to silhouettes along shared edges now need to be detected. Shared edges resulting from the application of boolean operations to free form surfaces are often piecewise linear approximations. This is owing to the complexity of the intersection problem. For example, the

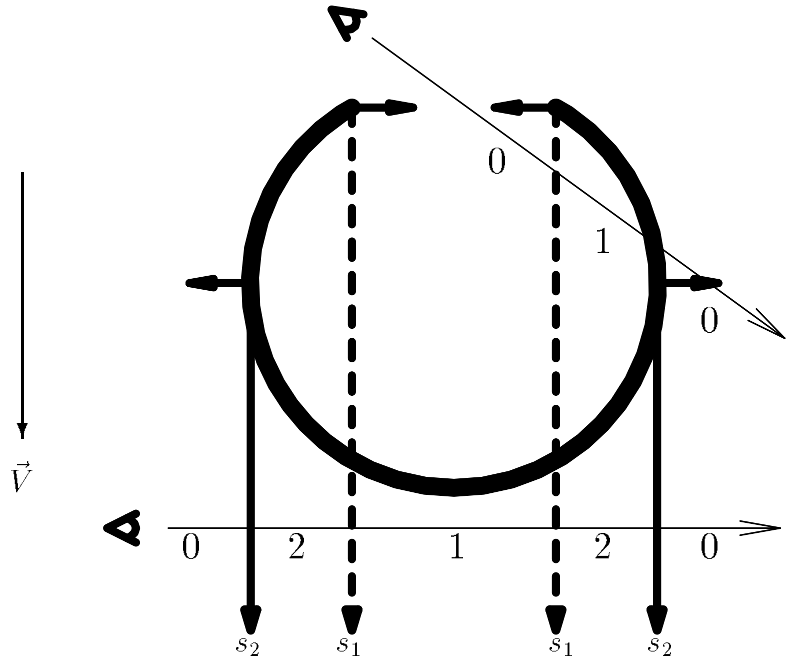

Figure 3: A 2-D example of occlusion counts for shadow surfaces. Shadow surfaces cast from surface unshared edge contours (s_1) delineate regions of space that are in single shadow while those cast from surface silhouette contours (s_2) delineate regions doubly in shadow. The numbers denote shadow occlusion amounts.

curve resulting from the intersection of two bicubic surfaces can be a polynomial with degree as high as 324 [Thomas 1984]. For these piecewise linear shared edges one can step along the vertices of the shared edge examining the normals of the two shared surfaces. Those with one surface normal that is back facing while the other is front facing lie along a silhouette.

Data reduction techniques can significantly reduce the size of the piecewise linear approximation by representing them as higher order curves [Lyche 1987]. Furthermore, special surface–surface intersection cases, such as those for quadric surfaces, have closed form representations as high order curves. Unfortunately, for shared edges represented as higher order curves, one cannot simply step along the curve as done for the piecewise linear case. The following approach has been developed to handle shared edges of arbitrary order. Let $S_1(u, v)$ and $S_2(r, s)$ be two surfaces intersecting along a shared edge e. The edge e can be expressed in the domain of each surface as $e(t) = S_1(u(t), v(t)) = S_2(r(t), s(t))$. Let $\vec{n}_1(u, v)$ and $\vec{n}_2(r, s)$ be the unnormalized normal surfaces of S_1 and S_2 as expressed in Equation (1). Then

$$g(t) = (\vec{n}_1(u(t), v(t)) \cdot \vec{V})(\vec{n}_2(r(t), s(t)) \cdot \vec{V}), \tag{5}$$

is positive if both normals are front facing or back facing, and is negative if one is front facing and the other is back facing. Since Equation (5) is C^0 continuous if the surfaces S_1 and S_2 are regular C^1 continuous surfaces, the zero set of Equation (5) provides the domain along the shared edge that is a silhouette viewed along $\vec{V}$. $g(t)$ in Equation (5) can be computed symbolically and represented as a single scalar curve [Elber 1992a].

The extracted contours are again cast into ruled shadow surfaces. The approach is summarized in Algorithm 1. We discuss the orientation component of the algorithm in the next section.

Algorithm 1. Shadow volume generation.

```
For each lightSrc
    For each model in scene
    DO
        contourCrvs = extractContours( model, lightSrcPos )
        shadowSrfs = createRuled( contourCrvs, lightSrcPos )
        orientShadSurfs( shadowSrfs )
    END
```

4 Shadow Surface Orientation

The shadow surface boundaries must be oriented according to Definition 2. Let p be a point on a silhouette extracted from surface S. The shadow volume cast from the silhouette has a normal at p that is identical to the normal of S at p. This is because the orientation of the silhouette is maintained when forming the ruled surface. Somewhat counter intuitively, the shadow surface orientation should sometimes be reversed so that its normals point in the opposite direction. Such a case is now discussed.

4.1 The Torus Anomaly

A torus viewed at a oblique angle is presented in Fig. 4. The star shaped inner silhouette contains four cusps. A cusp in a silhouette can be formed at a point where the tangent of the silhouette curve is collinear with the viewing direction, $\vec{V}$. Inspecting the surface normals of the torus along each of the four regions we see that in two regions the normals point into the silhouette loop while in two others, the normals are pointing outward (see Fig. 4a). Here, it would be impossible to construct a single shadow volume for the star shaped silhouette curve that is orientable [doCarmo 1976]. Two of the shadow surface orientations must be reversed so that their normals point in the direction of $-\vec{O}$ as specified in Definition 2. The correct shadow surface orientations, for the torus example, with shadow surface normals pointing in the direction of $-\vec{O}$, are shown in Fig. 4b. A method for determining $\vec{O}$ and orienting the shadow surfaces is discussed in the following section.

4.2 Occlusion Direction Determination

We now present a method for orienting shadow surfaces cast from silhouettes based on determining $\vec{O}$. Orientation of the other contour types can be performed similarly. Let $p = S(u_0, v_0)$ be a point on a silhouette of surface S. To orient the corresponding shadow surface correctly we compare the shadow surface normal, $\vec{n}(u_0, v_0)$, with $\vec{O}$. If they point in the same direction the shadow surface orientation is reversed to satisfy Definition 2.

The normal of S at p is orthogonal to the view direction, $\vec{V}$, by definition (Equation (2) and Equation (4)). The view direction, $\vec{V}$, then lies in the tangent plane of the surface and can be expressed as a linear combination of the surface partial derivatives, provided S is regular,

$$\vec{V} = a\frac{\partial S}{\partial u} + b\frac{\partial S}{\partial v}. \tag{6}$$

The scalars, a and b, can be determined since Equation (6) is a set of three equations in x, y, and z. It degenerates into two equations and two unknowns (a and b) since $\vec{V}$ is known to lie in the tangent plane

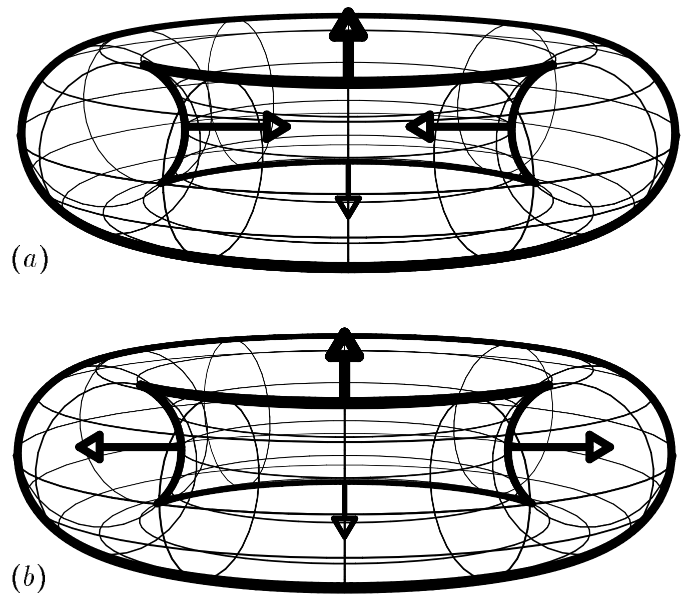

Figure 4: View is from the light source. Contours are shown in bold. (a) Shadow surface normals aligned with the normals of the surface (pointing outside the model) from which they were extracted. (b) Shadow surface normals properly oriented to point outside the shadow volume in the direction of $-\vec{O}$.

spanned by $\frac{\partial S}{\partial u}$ and $\frac{\partial S}{\partial v}$. The values of a and b give the direction in parametric space that corresponds to $\vec{V}$ in Euclidean space, at p. Let $c(t)$ be a curve in S through p such that $c'(t)$ at p is parallel to $\vec{V}$. By examining the component of the second derivative of $c(t)$ at p, in the direction $\vec{n}$, $\vec{O}$ can be expressed analytically.

$$c(t) = S(u(t), v(t)), \tag{7}$$

where $u(t) = at + u_0$ and $v(t) = bt + v_0$.

The first derivative of the curve,

$$c'(t) = \frac{\partial S}{\partial u}\frac{du}{dt} + \frac{\partial S}{\partial v}\frac{dv}{dt}, \tag{8}$$

corresponds directly to Equation (6) above, with $a = \frac{du}{dt}$ and $b = \frac{dv}{dt}$. The second derivative is then

$$c''(t) = \frac{\partial^2 S}{\partial u^2}\frac{du}{dt}^2 + 2\frac{\partial^2 S}{\partial u \partial v}\frac{du}{dt}\frac{dv}{dt} + \frac{\partial^2 S}{\partial v^2}\frac{dv}{dt}^2 + \frac{\partial S}{\partial u}\frac{d^2 u}{dt^2} + \frac{\partial S}{\partial v}\frac{d^2 v}{dt^2} \tag{9}$$

The component of $c''(t)$ in the direction of $\vec{n}$ can be found by computing the dot product of $c''(t)$ and $\vec{n}$. The last two components of Equation (9) contribute only in the direction of the tangent plane (and are zero when $u(t) = at + u_0$ and $v(t) = bt + v_0$). Taking this into account and substituting $a = \frac{du}{dt}$ and $b = \frac{dv}{dt}$,

$$c''(t) \cdot \vec{n} = \left(\frac{\partial^2 S}{\partial u^2}a^2 + 2\frac{\partial^2 S}{\partial u \partial v}ab + \frac{\partial^2 S}{\partial v^2}b^2 \right) \cdot \vec{n}. \tag{10}$$

Examining the sign of Equation (10), $\vec{O}$ can be determined. If it is positive then $\vec{O} = \vec{n}$, otherwise $\vec{O} = -\vec{n}$. It is unnecessary to explicitly determine $\vec{O}$ to orient the surface. If Equation (10) is positive then the shadow surface normal, $\vec{n}$, points in the direction of occlusion and the shadow surface orientation must be reversed.

Algorithm 2. Shadow volume orientation.

```
Evaluate shadow surface normal, n⃗, at p.
Determine a and b from Equation (6).
If ( c''(t) · n⃗ > 0 )
    Reverse shadow surface orientation.
```

The orientation process is summarized in Algorithm 2. In practice, a finite difference method was found sufficient to approximate $c''(t)$. The remainder of the procedure follows that above. This technique has been found to be robust over a variety of models, including the ones in this papers.

5 Rendering

Rendering shadow volumes can be done in two ways. The first technique follows the common scan line approach [Crow 1977]. The shadow surfaces are added to the free form surface database before scan line rendering. These surfaces are not treated as renderable data but are added only to provide shadowing information. Traversing the z-list [Atherton 1981], L, of surfaces at a given pixel of the Z-buffer, we examine the normals of the shadow surfaces that we encounter. While traversing L, occlusion counters for each light source are incremented for each corresponding front facing shadow surface encountered and decremented for each that is back facing. If open surfaces are part of the database, then the counter increments and decrements are by the occlusion count associated with the shadow surface type, with those corresponding to silhouettes having an occlusion count of two. When calculating shading information for a surface, those light sources with a negative occlusion counter are not included in the calculations.

An alternate approach, that corresponds to the two pass model precision approach to shadow generation [Atherton 1978], represents shadows as trimmed regions of the model through a preprocess. The shadow volumes are intersected with models in the scene to trim the model into shadowed and unshadowed regions. This approach provides accurate shadow generation for free form surfaces and the rendering of such models can be done using existing rendering hardware to achieve near real-time shadow generation. The use of Boolean operations requires the grouping of extracted contours curves into closed loops to form true volumes. In addition, data reduction techniques [Lyche 1987] may be needed to decrease the data size and raise the order of the extracted contours to increase the robustness and efficiency of the Boolean operations. The resulting trimmed shadowed and unshadowed regions are desirable if accessibility and visibility is to be explicitly solved.

6 Examples

The included rendered examples were produced by incorporating the above techniques into an existing scan line renderer in the Alpha_1 modeling system [EGS 1992]. The color images show shadow renderings of several free form surface models. The corresponding shadow surfaces for the top left image of the floating torus and sphere (Fig. 6) are shown below it (Fig. 8). Timings of shadow volume generation and rendering are shown in table 1 for the examples.

Table 1: Rendering and shadow volume generation times on DEC 5000/240 for 500x350, one sample per pixel, images.

Scene	Without Shadows	Shadow Vol. Generation	With Shadows
Pawn	68 Sec.	23 Sec.	792 Sec.
Torus/sphere	58 Sec.	15 Sec.	625 Sec.
Teapot/mug	94 Sec	38 Sec.	202 Sec.

7 Conclusion

The shadow volume technique we have described provides a method for creating accurate shadow representations directly from free form surface models, without the need for a polygonal approximation. Robust and efficient techniques for trimming the surfaces into shadowed and unshadowed regions using shadow volumes need to be further explored. The regions that result can provide useful accessibility information for manufacturing of parts and robot path planning. Generation of shadow volumes from free form surfaces for area light sources should also be further explored. We plan to investigate the application of shadow volumes in accessibility determination.

8 Acknowledgments

We would like to thank Elaine Cohen, Jamie Painter, and Mike Blum for their valuable comments on the material in this paper.

References

[EGS 1992] Engineering Geometry Systems (1992) *Alpha_1 User's Manual.*

[Appel 1968] Appel A (1968) *Some Techniques for Shading Machine Renderings of Solids.* SJCC Proceedings 1968, Thompson Books, Washington, DC, pp 37-45

[Atherton 1978] Atherton P, Weiler K, Greenberg D (1978) *Polygon Shadow Generation.* SIGGRAPH Proceedings 1978, Computer Graphics 12(3): 275-281

[Atherton 1981] Atherton P (1981) *A Method of Interactive Visualization of CAD Surface Models on a Color Video Display.* SIGGRAPH Proceedings 1981, Computer Graphics 15(3)

[Bergeron 1986] Bergeron P (1986) *A General Version of Crow's Shadow Volumes.* CG&A 6(9): 17-28

[Bouknight 1970] Bouknight J, Kelly K (1970) *An Algorithm for Producing Half-Tone Computer Graphics Presentations with Shadows and Movable Light Sources.* AFIPS Proceedings 1970, AFIPS Press, Reston, VA, 36: 1-10

[Chin 1989] Chin N, Feiner S (1989) *Near Real-Time Shadow Generation Using BSP Trees.* SIGGRAPH Proceedings 1989, Computer Graphics 23(3): 99-106

[Cohen 1985] Cohen M, Greenberg D (1985) *The Hemi-Cube: A Radiosity Solution for Complex Environments.* SIGGRAPH Proceedings 1985, Computer Graphics 19(3): 31-40

[Cook 1984] Cook R, Porter T, Carpenter L (1984) *Distributed Ray Tracing.* SIGGRAPH Proceedings 1984, Computer Graphics 18(3): 137-144

[Crow 1977] Crow F (1977) *Shadow Algorithms for Computer Graphics.* SIGGRAPH Proceedings 1977, Computer Graphics 11(3): 242-248

[doCarmo 1976] doCarmo M (1976) *Differential Geometry of Curves and Surfaces.* Prentice Hall

[Elber 1992a] Elber G (1992) *Free From Surface Analysis using a Hybrid of Symbolic and Numeric Computation.* Ph.D. thesis, University of Utah, Computer Science Department

[Elber 1992b] Elber G, Cohen E (1992) *Adaptive Isocurves Based Rendering for Freeform Surfaces.* Technical Report UUCS-92-040, University of Utah

[Fuchs 1985] Fuchs H, Goldfeather J, Hultquist J, Susan S, Austin J, Brooks F, Eyles J Jr., Poulton J (1985) *Fast Spheres, Shadows, Textures, Transparencies, and Image Enhancements in Pixel-Planes.* SIGGRAPH Proceedings 1985, Computer Graphics 19(3): 111-120

[Joy 1988] Joy K, Bhetanabhotla M (1988) *Ray Tracing Parametric Patches Using Numerical Techniques and Ray Coherence.* SIGGRAPH Proceedings 1986, Computer Graphics 20(4): 279-285.

[Kajiya 1982] Kajiya J (1982) *Ray Tracing Parametric Patches.* SIGGRAPH Proceedings 1982, Computer Graphics 16(3): 245-254.

[Lane 1980] Lane J, Carpenter L, Whitted T, Blinn J (1980) *Scan Line Methods for Displaying Parametrically Defined Surfaces.* Communications of the ACM 23(1): 23-34

[Lyche 1987] Lyche T, Morken K (1987) *Knot Removal for Parametric B-spline Curves and Surfaces.* Computer Aided Design 4: 217-230

[Max 1986] Max N (1986) *Atmospheric Illumination and Shadows.* SIGGRAPH Proceedings 1986, Computer Graphics 20(4): 117-124

[Nishita 1987] Nishita T, Miyawaki Y, Nakame E (1987) *A Shading Model for Atmospheric Scattering Considering Luminous Intensity Distribution of Light Sources.* SIGGRAPH Proceedings 1987, Computer Graphics 21(4): 303-310

[Nishita 1990] Nishita T, Sederberg T, Kakimoto M (1990) *Ray Tracing Trimmed Rational Surface Patches.* SIGGRAPH Proceedings 1990, Computer Graphics 24(4): 337-345

[Nishita 1991] Nishita T, Kaneda K, Nakamae E (1991) *A scanline algorithm for displaying trimmed surfaces by using Bézier clipping.* The Visual Computer 7(5-6): 269-279

[Reeves 1987] Reeves W, Salesin D, Cook R (1987) *Rendering Antialiased Shadows with Depth Maps.* SIGGRAPH Proceedings 1987, Computer Graphics 21(4): 283-291

[Schantz 1988] Schantz M, Chang S (1988) *Rendering Trimmed NURBS with adaptive Forward Differencing.* SIGGRAPH Proceedings 1988, Computer Graphics 22(4): 189-198

[Thomas 1984] Thomas S (1984) *Modelling Volumes Bounded by B-Spline Surfaces.* Ph.D. thesis, University of Utah, Computer Science Department, 1984

[Toth 1985] Toth D (1985) *On Ray Tracing Parametric Surfaces.* SIGGRAPH Proceedings 1985, Computer Graphics 19(3): 171-179

[Watt 1992] Watt A, Watt M (1992) *Advanced Animation and Rendering Techniques.* Addison-Wesley Publishers, pp 155-177

[Whitted 1980] Whitted T (1980) *An Improved Illumination Model for Shaded Display.* Communications of the ACM 23(6): 343-349

[Williams 1978] Williams L (1978) *Casting Curved Shadows on Curved Surfaces.* SIGGRAPH Proceedings 1978, Computer Graphics 12(3): 270-274

[Woodward 1989] Woodward C (1989) *Ray Tracing Parametric Surfaces by Subdivision in Viewing Plane.* in *Theory and Practice of Geometric Modeling* Strasser W, Seidel HP editors, Springer-Verlag, pp 273-290