Efficient Algorithms for Computing the Shadow Volumes from an Area Light Source

Kwan-Hee Yoo[1,2], Dae Seoung Kim[2], Kyung-Yong Chwa[2] and Sung Yong Shin[2]

[1]Multimedia Lab., DACOM Corporation R&D Center
34 Gajeong-dong, Yusung-gu Taejon 305-350, KOREA
khyoo@halla.dacom.co.kr

[2]Department of Computer Science, KAIST
Kusung-dong 373-1, Yusung-gu Taejon 305-701, KOREA
{khyoo, dskim, kychwa, syshin}@jupiter.kaist.ac.kr

Suppose that an area light source in the 3D space shines past a scene polygon. The area light source generates the two types of shadow volumes for each scene polygon, *i.e.*, one with partial occlusion and the other with complete occlusion. These are called, *penumbra* and *umbra*, respectively.

In this paper, we propose two algorithms for computing the penumbra and the umbra of a scene polygon with n vertices from an area light source with m vertices in $O(m+n)$ time and $O(m+n\log n)$ time, respectively.

Efficient Algorithms for Computing the Shadow Volumes from an Area Light Source

Kwan-Hee Yoo[1,2], Dae Seoung Kim[2], Kyung-Yong Chwa[2] and Sung Yong Shin[2]

[1]Multimedia Lab., DACOM Corporation R&D Center
34 Gajeong-dong, Yusung-gu Taejon 305-350, KOREA
khyoo@halla.dacom.co.kr

[2]Department of Computer Science, KAIST
Kusung-dong 373-1, Yusung-gu Taejon 305-701, KOREA
{khyoo, dskim, kychwa, syshin}@jupiter.kaist.ac.kr

3次元空間中に面光源と多角形遮蔽物があるとする．すると2種類の影，すなわち半影と本影ができる．本文では m 頂点の多角形光源と n 頂点の多角形遮蔽物が与えられたときに，その半影を求める $O(m+n)$ 時間のアルゴリズムと本影を求める $O(m+n\log n)$ 時間のアルゴリズムを与える．

1 Introduction

Suppose that an area light source in the 3D space shines past the scene polygons which represent walls, doors, windows, tables and so on. The polygons cast shadows, and in general attenuate or eliminate the light reaching regions of the 3D space. In this environment, consider a rendering system which generates a realistic picture of the scene from a view point. The system involves shadow testing at every point on the visible surfaces. To efficiently perform the shadow testing, many researchers [Cam91, Hec92, NN85] proposed algorithms for computing the shadow volume cast by each scene polygon. The shadow testing for a point in the 3D space is done by checking if the point is within a shadow volume. An area light source generates two types of shadow volumes for each scene polygon, *i.e.*, one with partial occlusion and the other with complete occlusion. These are called, *penumbra* and *umbra*, respectively.

For a convex area light source with m vertices and a convex scene polygon with n vertices, Nishita and Nakamae [NN85] proposed an algorithm for computing the penumbra and umbra in $O(mn \log mn)$ time, respectively. Campbell [Cam91] found the bounding planes of the penumbra and those of the umbra in $O(mn)$ time, respectively. In the latter method, an additional processing is required to construct the umbra and the penumbra from the bounding planes. He used a 3D dimensional binary space partition tree [PY90]. However, its construction takes cubic time in the worst case [PY90].

In this paper, we propose two algorithms for constructing the penumbra and the umbra in $O(m+n)$ time and $O(m+n \log n)$ time, respectively. The details of the algorithm are described in Section 2 and Section 3. Finally, Section 4 concludes this paper.

2 Computing the Penumbra

In this section, we present an algorithm for computing the penumbra cast by a scene polygon from an area light source.

Let P be a convex polygon with n vertices and S a convex area light source with m vertices. We denote the vertices of P by v_0, v_1, $\cdots$, and v_{n-1} and those of S by u_0, u_1, $\cdots$, and u_{m-1}. The area light source S generates a shadow volume for the polygon P such that the visibility between the interior of the volume and S is partially or completely blocked by P. We call the shadow volume the penumbra of P from S, and denote it by $PE(S,P)$.

Without loss of generality, we assume that P and S are not coplanar. Otherwise, the problem can be solved trivially.

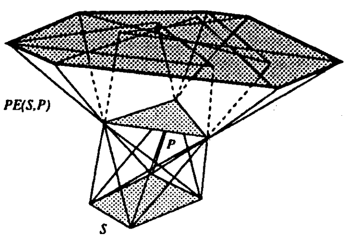

Figure 1: Penumbra volume derivations of Nishita and Nakamae's algorithm

Nishita and Nakamae [NN85] treated each vertex of S as a point light source that has a shadow volume generated from P. They showed that $PE(S,P)$ is the smallest convex polyhedron that contains these shadow volumes (see Figure 1). Based on this observation, they were able to compute $PE(S,P)$ in $O(mn \log mn)$ time.

Campbell [Cam91] characterized the bounding planes of $PE(S,P)$. The bounding planes consist of three types of planes : a plane containing P (type I plane), the planes which support an edge of S and a vertex of P (type II planes), and the planes which support a vertex of S and an edge of P (type III planes).

He also showed that the type II plane supporting an edge in S and a vertex in P makes a smaller angle with the plane containing S than any other plane supporting the same edge in S and another vertex in P does, as illustrated in Figure 2 (a). This is equivalent to the fact that S and P lie on the opposite sides of a type II plane. By a similar argument, the same is true for a type III plane (see Figure 2 (b)).

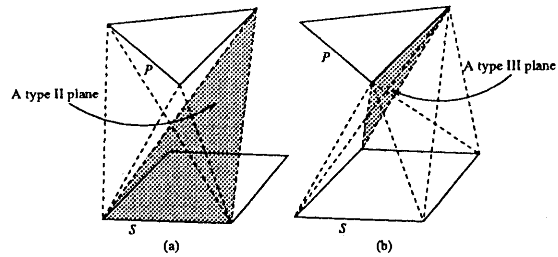

Figure 2: Illustration of the bounding planes of $PE(S,P)$

To compute a type II plane, all planes supporting an edge and the vertices of P are enumerated to find the one that makes the smallest angle with the plane containing S. A type III plane can similarly be found by enumerating the plane supporting an edge in P and the vertices of S. Therefore, it takes $O(mn)$ to find all planes bounding $PE(S,P)$, where m and n are the numbers of vertices in S and P, respectively. For each bounding plane H, let H^+ be the half-space not containing S. Campbell proved that $PE(S,P)$ is the intersection of all such half-spaces as H^+.

This argument is valid if neither the plane containing S intersects P nor that containing P does S(see Figure 3 (a) and (b)). Otherwise, either S or P is split by the intersecting plane, which further increases the cost in time to find the bounding faces.

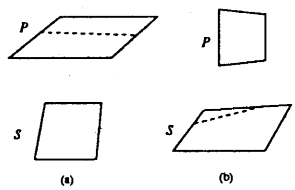

Figure 3: The special cases of Campbell's algorithm

In order to construct a penumbra $PE(S,P)$ in $O(m+n)$ time, we need more sophisticated characterizations of faces in $PE(S,P)$. A face of $PE(S,P)$ is said to be of type I, II, or III if it is contained in a type I, II, or III plane, respectively.

Denoting $CH(q,S)$ be the convex hull of a point p and S, we begin with characterizing a type I face.

Lemma 1 P is the type I face of $PE(S,P)$ if and only if the plane H containing P does not intersect the interior of S.

Proof: Omitted $\square$

Now, let f be a type II face lying on the plane H which supports a vertex v_j in P and an edge $e(u_{i-1}, u_i)$ in S. Let $l(p,q)$ be a line containing two points, p and q, and $r(p, \overline{ab})$ denotes a half-line from p in the direction from a to b. Clearly, $e(u_{i-1}, u_i)$ is contained in a half-plane determined by $l(u_{i-1}, v_j)$. This half-plane is denoted by $H^+(u_{i-1}, v_j)$. $H^-(u_{i-1}, v_j)$ denotes the other half-plane in H with respect to $l(u_{i-1}, v_j)$. Similarly, $l(u_i, v_j)$ gives two half-planes, $H^+(u_i, v_j)$ and $H^-(u_i, v_j)$ such that $H^+(u_i, v_j)$ contains $e(u_{i-1}, u_i)$. The following lemma characterizes a type II face f.

Lemma 2 Let f be a type II face lying on the plane H supporting a vertex v_j in P and an edge $e(u_{i-1}, u_i)$ in S. Then, $f = H^-(u_{i-1}, v_j) \cap H^-(u_i, v_j)$. That is, $r(v_j, \overline{u_{i-1}v_j})$ and $r(v_j, \overline{u_i v_j})$ is the boundary of f (see Figure 4 (a)).
Proof: Omitted □

Finally, we characterize a type III face in the following lemma:

Lemma 3 Let f be a type III face lying on the plane H supporting an edge $e(v_{j-1}, v_j)$ in P and u_i in S. Let $H^+(u_i, v_{j-1})$ and $H^+(u_i, v_j)$ be the half-planes containing $e(v_{j-1}, v_j)$. Suppose that u_i is contained in $H^+(v_{j-1}, v_j)$. Then, $f = H^+(u_i, v_{j-1}) \cap H^+(u_i, v_j) \cap H^-(v_{j-1}, v_j)$. That is, the boundary of f is $r(v_{j-1}, \overline{u_i v_{j-1}})$, $e(v_{j-1}, v_j)$, and $r(v_j, \overline{u_i v_j})$ (see Figure 4 (b)).
Proof: Omitted □

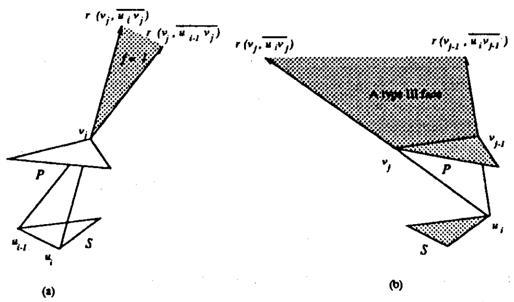

Figure 4: Illustration of the type II and type III faces of $PE(S,P)$

From the proofs of Lemmas 2 and 3, the following results are immediate.

Corollory 1 Every type II or III plane contains only one face.

This corollary is important since we do not need to make an extra effort to construct another face in a plane if a face in it is identified.

Now, we are ready to give a sketch of our algorithm for constructing $PE(S,P)$. Initially, we find a type II face f_0 from an edge $e(u_{i-1}, u_i)$ in S(we can also start with a type III face). This can be done in $O(n)$ time since we need to enumerate every plane supporting $e(u_{i-1}, u_i)$ and a vertex v_j of P. By Lemma 2, f_0 has two edges $r(v_j, \overline{u_{i-1}v_j})$ and $r(v_j, \overline{u_i v_j})$. From $r(v_j, \overline{u_i v_j})$, we find the face on the boundary of $PE(S,P)$ which shares the edge $r(v_j, \overline{u_i v_j})$ with f_0. Now, the new face f_1 plays the role of f_0 to provide a new edge for constructing another face. This process is repeated until f_0 is re-encountered.

In order to make sure that this idea works, we need to elaborate our sketch to provide a correct advancing mechanism. In a general step, we need to find a face f_{k+1} which shares an edge $r(v_j, \overline{u_i v_j})$ with a given face f_k, which is computed in the previous step. There are two cases depending on f_k:

- **case 1:** f_k is a type II face, and

- **case 2:** f_k is a type III face.

In case 1, f_k is contained in the type II plane supporting $e(u_{i-1}, u_i)$ and v_j, and is bounded by $r(v_j, \overline{u_{i-1}v_j})$ and $r(v_j, \overline{u_iv_j})$. Since f_{k+1} shares $r(v_j, \overline{u_iv_j})$ with f_k, the plane H_{k+1} containing f_{k+1} must have both v_j and u_i. We can determine the H_{k+1} if we find a third point, which is either a vertex of P or that of S. Clearly, this point is adjacent to either v_j in P or u_i in S, *i.e.*, one of v_{j-1}, v_{j+1}, u_{i-1}, and u_{i+1}. Since S and P are both convex, any other choice would generate a plane which cannot separate S and P on its opposite sides. u_{i-1} cannot be a choice since f_k contains v_j, u_{i-1}, and u_i. Therefore, there are three candidate vertices for constructing f_{k+1}, given $r(v_j, \overline{u_iv_j})$.

In order to obtain an insight on the configuration of S and P. Let four half-planes H_k, H_1^*, H_2^*, and H_3^* be defined as follows:

- H_k = the plane containing the current plane f_k, *i.e.*, the plane containing

  $r(v_j, \overline{u_iv_j})$ and u_{i-1},

- H_1^* = the plane containing $r(v_j, \overline{u_iv_j})$ and v_{j-1},

- H_2^* = the plane containing $r(v_j, \overline{u_iv_j})$ and v_{j+1}, and

- H_3^* = the plane containing $r(v_j, \overline{u_iv_j})$ and u_{i+1}.

H_s^*, $s = 1, 2, 3$ are the candidates for H_{k+1} containing f_{k+1}. H_k gives two half-spaces H_k^+ and H_k^- such that $P \subset H_k^+$ and $S \subset H_k^-$. The line containing $r(v_j, \overline{u_iv_j})$ divides H_k into two half-planes, $H^+(u_i, v_j)$ and $H^-(u_i, v_j)$ such that $f_k \subset H^+(u_i, v_j)$. Let the angle between H_k and H_s^*, s = 1,2,3 be the convex dihedral angle determined by $H^-(u_i, v_j)$ and the half-plane of H_s^* lying in H_k^+. We show that $H_{k+1} = H_s^*$ for some s if and only if H_s^* makes the smallest angle with H_k among all candidate planes.

Suppose that $H_{k+1} = H_s^*$ for some s. From Corollary 1, H_s^* contains f_{k+1} which shares $r(v_j, \overline{u_iv_j})$ with f_k. If we rotate H_k about the line containing $r(v_j, \overline{u_iv_j})$, then it should hit one of v_{j-1}, v_{j+1}, and u_{i+1} before intersecting the interior of S and P, while separating S and P on the opposite sides of H_k. If H_k hits one of the three points, H_k coincides with H_s^*. Therefore, H_s^* makes the smallest angle with f_k.

Now, suppose that H_s^* and H_k give the smallest angle for some s. If we rotate H_s^* about the line containing $r(v_j, \overline{u_iv_j})$, H_s^* intersects the interior of S or P. Clearly, H_s^* is either a type II or III plane, which has exactly one face such that $r(v_j, \overline{u_iv_j})$ is its one edge by Corollary 1. This face is f_{k+1}, and thus our claim holds true.

Given H_{k+1}, we can identify the new edges in it. There are three cases depending on H_{k+1} to identify these edges:

- **case 1(a):** H_{k+1} contains v_{j-1},

- **case 1(b):** H_{k+1} contains v_{j+1}, and

- **case 1(c):** H_{k+1} contains u_{i+1}.

In case 1(a), H_{k+1} contains a type III face f_{k+1}. By Lemma 3, the new edges are $r(v_{j-1}, \overline{u_iv_{j-1}})$ and $e(v_{j-1}, v_j)$, as illustrated in Figure 5 (a). Since two faces sharing the other edge $r(v_j, \overline{u_iv_j})$ have already found, $r(v_{j-1}, \overline{u_iv_{j-1}})$ is used for the next move, which is described in case 2. By the symmetric argument, in case 1(b), f_{k+1} is also a type III face contained in H_2^*, and $r(v_{j+1}, \overline{u_iv_{j+1}})$ is used for the next move to case 2 (see Figure 5 (b)). Finally, in case 1(c), f_{k+1}

is a type II face, as illustrated in Figure 5 (c). From Lemma 2, $r(v_j, \overline{u_{i+1}v_j})$ is the new edge. With this edge, we repeat case 1.

Given a type II face f_k and its edge $r(v_j, \overline{u_i v_j})$, it takes constant time to find f_{k+1} sharing $r(v_j, \overline{u_i v_j})$ and a new edge of f_{k+1}, since we only consider three vertices, v_{j-1}, v_{j+1}, and u_{i+1}, which are either adjacent to v_j in P or to u_i in S.

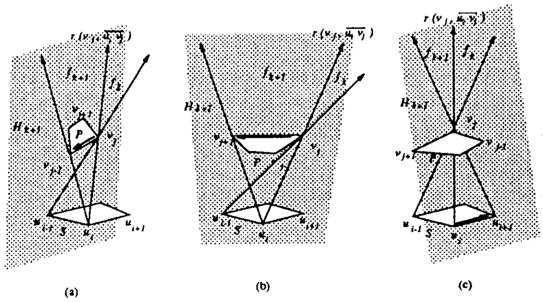

Figure 5: Finding a face f_{k+1} sharing an edge $r(v_j, \overline{u_i v_j})$ with a given type II face f_k

In case 2, *i.e.*, f_k is a type III face, we can also compute the next face f_{k+1} sharing an edge $r(v_j, \overline{u_i v_j})$ with f_k using a similar method in case 1.

With our advancing mechanism as building blocks, we can construct the boundary of $PE(S, P)$ from an initial type II face, stepping forward from a face of type II or III to a face of type II or III, until we revisit the initial face. At each step, we can always determine the unique face with a given edge. This face gives a new edge for the next move.

There are $O(m+n)$ faces of type II or III since every type II or III face supports exactly an edge of S or an edge of P. Our algorithm generates all of these faces and duplicates none of them, Hence, the following result is immediate.

Theorem 1 *In the 3D space, given a convex polygon P with n vertices and a convex area source S with m vertices, $PE(S, P)$ can be computed in $O(n+m)$ time.*

3 Computing the Umbra

An area light source S may generate a shadow volume for a scene polygon P such that the visibility of the interior of the volume and S can be completely blocked by P. We call the shadow volume the umbra of P from S and denote it by $UM(S, P)$. In this section, we propose an $O(m + n \log n)$ time algorithm for computing $UM(S, P)$.

Without loss of generality, we assume the following two conditions : S and P are not coplanar, and the plane containing P does not intersect S. If either of the above conditions is not true, then $UM(S, P)$ is empty.

Nishita and Nakamae [NN85] proved that $UM(S, P)$ is the intersection of the shadow volumes with respect to all vertices of S (see Figure 6). Based on this observation, they were able to compute $UM(S, P)$ in $O(mn \log mn)$ time.

Campbell [Cam91] characterized the bounding planes of $UM(S, P)$. The bounding planes consist of two types of planes : a plane containing P (called type A plane) and planes each of which supports an edge of P and a vertex of S (called type B planes). He also showed that the type B plane supporting an edge of P and a vertex of S makes the largest angle with the plane supporting S among all planes supporting the same edge in P and the vertices in S. Therefore, it is true that P and S lie the same side of a type B plane.

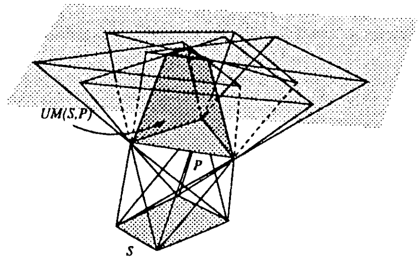

Figure 6: A umbra by Nishita and Nakamae's algorithm

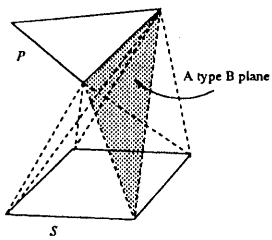

Figure 7: Choosing a plane containing a boundary face of $UM(S,P)$

Campbell proved that $UM(S,P)$ is the intersection of all half-spaces containing P with respect to the bounding planes. To compute a type B plane, all planes supporting an edge of P and the vertices of S are examined to determine the one that makes the largest angle with the plane containing S. Therefore, it takes $O(mn)$ to find all bounding planes of $UM(S,P)$.

In order to obtain an algorithm of better time bound, we investigate the boundary of the convex hull $CH(S,P)$ of S and P. Excluding S and P, every face of $CH(S,P)$ contains either an edge of P and a vertex of S or vice versa. Therefore, the plane containing a face in $CH(S,P)$, which have an edge of P and a vertex of S, is a type B plane. If every type B plane contains a face of $CH(S,P)$, then all type B planes can be obtained by constructing $CH(S,P)$.

Lemma 4 If a face of $CH(S,P)$ has an edge of P and a vertex of S, then the plane containing the face is of type B. Moreover, every type B plane contains a face of $CH(S,P)$.

Now, we are ready to give our algorithm for constructing $UM(S,P)$. First, we find the convex hull of P and S using the merge step of the algorithm by Preparata and Hong [PH77] in $O(m+n)$ time. And then, we select the type B planes among the planes containing the faces of the convex hull. For each plane H, let H^+ be the half-space containing P. Since $UM(S,P)$ is the intersection of all such half-spaces as H^+, we finally compute the intersection of $n+1$ half-spaces in $O(n \log n)$ time using an algorithm of Preparata and Muller [PM79]. Therefore, the following theorem is obtained.

Theorem 2 *In the 3D space, given a convex polygon P with n vertices and a convex area source S with m vertices, $UM(S,P)$ can be obtained in $O(m + n \log n)$ time.*

4 Concluding Remarks

Suppose that a convex polygon P with n vertices and a convex area light source S with m vertices are given. We present two algorithms for computing the penumbra and the umbra of

P from S in $O(m+n)$ and $O(m+n\log n)$ time, respectively.

In general, the intersection of n half spaces in the 3D space is computed in $\Theta(n\log n)$ time. However, we can observe that the boundary faces of the umbra is ordered along the boundary of P. Moreover, they can be computed in $O(m+n)$ time, as illustrated in Section 3. It is an interesting problem to compute the umbra in $O(m+n)$ time.

References

[Cam91] A. T. Campbell. *Modeling Global Diffuse Illumination for Image Synthesis.* PhD Thesis, Dept. of Computer Sciences, Univ. of Texas at Austin, December 1991.

[Hec92] P. Heckbert. Discontinuity meshing for radiosity. In *Third Eurographics Workshop on Rendering*, pages 203–226, May 1992.

[NN85] T. Nishita and E. Nakamae. Continuous-tone representation of three-dimensional objects taking account of shadow and interreflection. *ACM Computer Graphics(Proceedings of SIGGRAPH '85)*, 19(3):23–30, 1985.

[PH77] F. P. Preparata and S. J. Hong. Convex hulls of finite sets of points in two and three dimensions. *Communication of ACM*, 2(20):87–93, 1977.

[PM79] F. P. Preparata and D. E. Muller. Finding the intersection of n half-spaces in time $o(n\log n)$. *Theoretical Computer Science*, 8(1):45–55, 1979.

[PY90] M. S. Paterson and F. F. Yao. Efficient binary partitions for hidden-surface removal and solid modeling. *Discrete Computational Geometry*, 5:485–503, 1990.