Visible Zone Maintenance for Real-Time Occlusion Culling

Olaf Hall-Holt
SUNY Stony Brook* and Szymon Rusinkiewicz
Princeton University†

1 Introduction

An important challenge in interactive rendering of large scenes is ensuring that computational effort is concentrated on objects that are visible. Many approaches to this problem have been proposed with various trade-offs in preprocessing, level of approximation, choice of occluders, and use of temporal coherence. We describe a point visibility algorithm with near-linear preprocessing, the flexibility to be used for exact or approximate calculations, no distinction between occluders and occludees, and such that temporal coherence is leveraged beyond simply storing information from previous frames. The underlying algorithm is conceptually similar to that of Riviere[2], as it involves keeping track of a substructure of the visibility complex. Our basic 2-D algorithm may be extended to incorporate the notion of partial occlusion and thus be used as an acceleration technique in a 3-D visibility algorithm.

2 Visible Zone Maintenance in 2-D

We begin by considering the problem of maintaining visibility from a moving observer in two dimensions. Let S be a set of N non-overlapping convex polygons in the plane in general position, so that, for example, no three objects share a common tangent line. Let F be the free space surrounding the polygons; namely, the complement of the union of the interiors of elements of S. Let p be a point (the observer) that lies in F. The visibility polygon V is the set of points in F such that a straight line segment from p to a point of V lies entirely within F. At any given moment, the motion of the observer p is assumed to be described by a pseudo-algebraic function of constant degree, such that p remains at all times in F. The function describing the movement of p may be changed (say, in response to the movement of a mouse) as p moves. The problem is to design an efficient on-line algorithm that maintains a list of the objects along the boundary of V at all times as p moves. This list will be called the **visible set**, and may include an object more than once if the object is visible in two distinct places along the visibility polygon.

A radial sweep provides a direct method for computing the visible set of an observer in a static scene. This sweep can be used to build a (radial) trapezoidal subdivision of the plane that is exactly analogous to the classic (vertical) trapezoidal subdivision (see Figure 1). A radial subdivision consists of the polygons of S together with a set of line segments, defined as follows.

Let l be a line through the point observer p that is also tangent to a polygon P of S. A tangent segment of P is the connected portion of l incident to P that lies in free space. Each of the frames of Figure 1 shows a slightly different radial subdivision. Some of the tangent segments extend all the way to the observer p, while others are not incident to p because there are one or more objects between the tangent point and p.

An incremental change in the combinatorial structure of the radial subdivision that comes about as a result of observer motion is called an **elementary step**. In an elementary step, two neighboring tangent segments become momentarily coincident, and then separate again with different incident polygons. Among kinetic algorithms, maintaining a radial subdivision may be considered straightforward, since the updates are simple and the equations to solve for the event times are linear in the motion of p. If the radial subdivision is maintained, ray queries from the observer can be handled efficiently, even if the ray must report the first k objects that it intersects.

However, maintaining a radial subdivision may require many updates far from the observer, even if the visible set does not change. A generalization of the radial subdivision can be defined as follows. Consider a tangent segment to represent not just one segment, but a maximal connected set of segments tangent to a given object and incident to two other given objects. That is, a tangent segment t represents an equivalence class of all segments that can be obtained by sliding t along the boundary of its tangent object without becoming tangent to a second object. The elementary step operation is then a combinatorial update for these equivalence classes that can be applied even when there is no motion, so long as the boundary of two

*Math Tower, Room 1-101 Stony Brook, NY 11794-3600, U. S. A.
olaf@ams.sunysb.edu
†35 Olden St., Princeton, NJ 08544, U. S. A.
smr@cs.princeton.edu

such equivalence classes includes a common element. A visible zone is defined to be any set of tangent segments that can be obtained by applying elementary steps to a radial subdivision.

Although the data structure for a visible zone appears identical to that of a radial subdivision, the maintenance of a visible zone may require far fewer updates (see Figure 2). As the observer moves, no elementary steps are required until the visible set changes, since the incidence relationships between tangent segments and objects need not be affected by the observer motion. By delaying changes to the visible zone as long as possible, the kinetic maintenance algorithm avoids many updates in regions far from the observer. However, this 'lazy' evaluation results in a more complex update procedure when the visibility does change.

The update procedure for maintaining a visible zone consists of recursively identifying elementary steps that must be applied, and then applying them, as illustrated in Figure 3. As the observer moves to the right, the tangent segment C_{right} turns counter-clockwise about its tangent object C, until C_{right} becomes tangent to the left side of A. At that moment, the combinatorial description of C_{right} needs to change. In order to make this change, C_{right} needs to undergo an elementary step with A_{left}. In order to bring A_{left} into alignment with C_{right}, however, A_{left} must be turned counter-clockwise by means of elementary steps with B_{left} and D_{right}, respectively. The precise geometric locations of B_{left} and D_{right} after the elementary step are not recorded in the combinatorial description of each tangent segment. For illustration purposes, however, we choose some angle that is consistent with the combinatorial information, so that for example B_{left} is shown connected to A, rather than C in the second frame. After these two elementary steps, A_{left} is in a position where it can undergo an elementary step with C_{right}. Once the elementary step between A_{left} and C_{right} is applied, the observer can continue moving to the right, and the tangent segments connected to the observer correctly reflect the list of visible objects. The various elementary steps described in this update would have occurred earlier if a radial subdivision were being maintained; the 'lazy' approach of the visible zone avoided these elementary steps until they were necessary.

3 Extensions to the Visible Zone Algorithm

While two dimensional algorithms are not directly applicable to many environments of interest in computer graphics, the visible zone is a flexible structure that can be adapted to various other settings. For example, consider a three dimensional scene where the objects all lie near to a given ground plane. A ray query in the three dimensional scene, viewed from above, appears similar to a ray query in the plane, except that the ray may pass through the interior of (the projection of) an object. To accelerate 3-D ray queries, we use a two dimensional query structure that allows rays to pass through objects. Such a ray query can be handled by the visible zone by updating the tangent segments near the ray to line up with the observer, and thus mimic the operation of such a query in a radial subdivision.

In order to apply this exact visibility algorithm to a set of complex 3-D objects like trees (see Figure 4), we represent each object as a pair of axis-aligned boxes. The outer bounding box contains all the polygons of the tree, and the inner occluder box approximates the areas of dense foliage. Just as with 2-D rays that pass through objects according to some predicate, we allow 3-D rays to pass through the inner occluder boxes subject to an aggregate occlusion function, thus modeling gaps in the middle of dense foliage. Finally, instead of tracing individual rays to establish visibility, we collect the effect of ray bundles in a radial sweep about the observer that occurs simultaneously in the full 3-D space and the projected space, using information in the plane to speed up the top level sweep. We obtain perceptually accurate visibility at interactive frame-rates in a dense forest scene.

Table 1: Statistics about the forest scene flythrough.

Number of objects in scene	835,089
Total polygons in scene	10,000,008,491
Avg objects drawn per frame	46
Avg visibility computation time	5.3 ms. per frame
Time to generate initial zone	30 sec.

All measurements were made on a PC with 1.8 GHz Intel Pentium IV processor and NVidia GeForce 3 graphics.

References

[1] Basch, J., Guibas, L., and Herschberger, J. "Data Structures for Mobile Data," *Journal of Algorithms*, Vol. 31, No. 1, 1999.

[2] Rivière, S. "Walking in the Visibility Complex with applications to Visibility Polygons and Dynamic Visibility," *Proc. Canadian Conf. on Comp. Geom.*, 1997.

[3] Sutherland, I., Sproull, R., and Shumacker, R. "A Characterization of Ten Hidden-Surface Algorithms," *ACM Computing Surveys*, Vol. 6, No. 1, 1974.

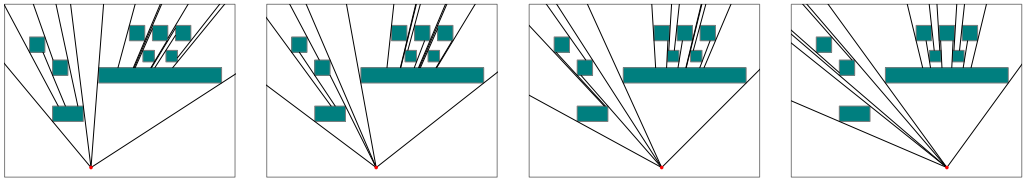


Figure 1: The **radial subdivision** consists of a set of segments tangent to scene objects and aligned with the observer.

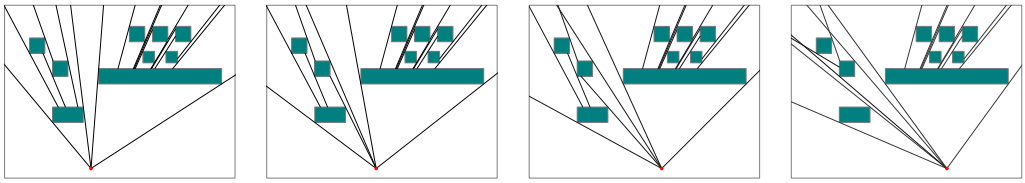
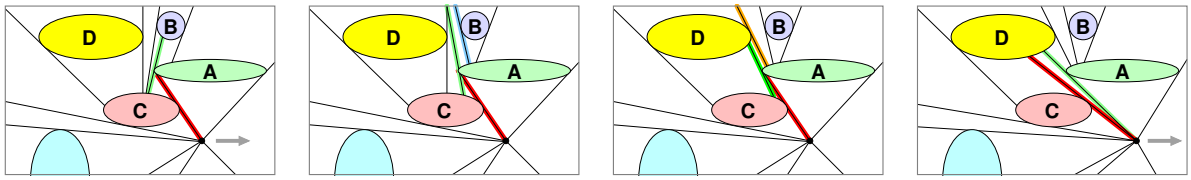


Figure 2: The **visible zone** permits fewer updates yet retains the segments incident to the observer.



(a)

As the observer moves, the tangent segment $t_{C,right}$ becomes tangent to A. In order to perform an elementary step, the segment $t_{A,left}$ must be rotated.

(b)

In order to rotate $t_{A,left}$, we perform an elementary step on $t_{A,left}$ and $t_{B,left}$.

(c)

We continue rotating $t_{A,left}$, and perform an elementary step on $t_{A,left}$ and $t_{D,right}$.

(d)

We are finally able to complete the elementary step on $t_{C,right}$ and $t_{A,left}$. The visible set now includes the object D.

Figure 3: A recursive update to the visible zone.

Figure 4: A frame from a flythrough of a forest scene accelerated using visible zone maintenance. The scene consists of 835,089 tree objects, and 10 billion total polygons. For this frame, 50 trees and 1,887,714 polygons were rendered. The rendering time was 85 ms., while the time devoted to visibility maintenance was 6 ms.