The Corner Arc Algorithm

Olaf Hall-Holt
Applied Math. & Statistics
Stony Brook University
Stony Brook, NY 11794-3600
olaf@ams.sunysb.edu

Categories and Subject Descriptors

F.2.2 [**Analysis of Algorithms and Problem Complexity**]: Nonnumerical Algorithms and Problems—*geometrical problems and computations*

General Terms

Algorithms, Theory

Keywords

Visibility, pseudo-triangulation, kinetic data structure

ABSTRACT

In a landmark survey of visibility algorithms in 1974, Sutherland, Sproull, and Schumacker stated that "We believe the principal untapped source of help for hidden surface algorithms lies in frame and object coherence." [11] Making use of such temporal coherence appears to be difficult, however, for scenes with moving objects. Almost three decades later Cohen-Or, Chrysanthou, and Silva stated that "The efficient handing of dynamic scenes is an open area of research at this point." [3] This five minute video describes a kinetic visibility algorithm for moving objects in the plane that is provably efficient in an average case setting.

The corner arc algorithm is a Kinetic Data Structure [2] which makes use of new properties of pseudo-triangulations [9]. The corner arc algorithm and its analysis are given in Chapters 4 and 5 of a thesis on kinetic visibility algorithms [6] where the proofs and more detail for the theorems in this abstract are found.

1. SETTING

Four difficulties arise in designing an appropriate setting for analyzing the computational complexity of maintaining visibility among moving objects. First, the efficiency of a visibility algorithm typically depends on the number of objects visible to the observer, and this number may vary greatly as the objects move. Second, object motion may result in collisions between objects, and changes to object trajectories due to collisions may easily result in complex motions. Third, even in the absence of collisions, the cost of maintaining data structures for objects in regions that happen not to be seen by the observer may nevertheless dwarf the costs of visibility maintenance in those regions that are seen. Fourth, in part as a result of the first three difficulties, worst-case bounds may be overly pessimistic.

Several approaches have been proposed to address these difficulties in creating an appropriate setting. We propose extending the static probabilistic setting of Cohen-Or et. al. [4] as follows. All object positions and initial motion trajectories are randomly drawn from a known distribution. Thus all essential parameters of the setting, such as the number of simultaneously visible objects, can in principal be derived analytically. We focus on asymptotic analysis for the number n of simultaneously visible objects, thus emphasizing the characteristics of algorithms in low density environments where collisions are rare. The issue of avoiding too many non-visibility related costs can be addressed in this setting by choosing an appropriate event horizon, namely choosing to ignore objects that lie so far from the observer that the probability of seeing them is vanishingly small.

Let S be a set of unit circles in the plane whose positions are drawn from a uniform distribution, whose speed is fixed, and whose velocities are chosen uniformly at random from the circle of directions. Let s be the 'spacing' between the objects, namely the side length of a square whose expected occupancy is one. The following lemmata express some basic properties of this setting:

LEMMA 1. *The expected number of objects n visible to a fixed point observer p is πs^2.*

LEMMA 2. *In a static setting where the observer moves a distance of s and the objects are fixed, the expected number of changes in visibility is $2s^3$.*

LEMMA 3. *In the dynamic setting where the observer is fixed and the objects simultaneously each move a distance of s, the expected number of changes in visibility from the point of view of the observer is approximately $4.3493s^3$.*

The following lemma provides one possible distance to the event horizon in the dynamic setting:

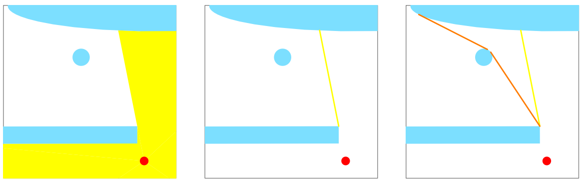
LEMMA 4. *The ratio between the expected number of changes in visibility that occur beyond a distance of $R_0 = s^2 \log s^2$ of the observer and the expected number of visibility changes that occur within distance R_0 is $O((1/s^4) \log^2 s)$.*

SCG'04, June 8–11, 2004, Brooklyn, New York, USA.
ACM 1-58113-885-7/04/0006.

For scenes in which the objects are drawn from a known distribution, one may expect bucketing approaches to perform well. However, even for the uniform distribution described above, the expected average distance between two objects that are adjacent along the boundary of the visibility polygon is $\Theta(s^2)$. As a result, the cost of ray-shooting in a uniform grid with O(1) objects expected per grid cell will be $\Omega(\sqrt{n})$ per change in visibility. Perhaps $\Theta(\sqrt{n})$ is an intuitive bound for the cost of this type of visibility maintenance.

2. CORNER ARC ALGORITHM

A **boundary segment** is a line segment on the boundary of the visibility polygon that connects two objects. For a given boundary segment b, let T be the incident object closer to the observer (the **tangent** object), and let F be the incident object farther from the observer (the **far** object). Let σ be the shortest path homotopic to b beginning at the tangent point of b on T, extending away from the part of the visibility polygon next to b, and ending on the far side of F. The **corner arc** of b is exactly that part of σ which extends up to F, but does not curve around it (see Figure 1).



(a) visibility polygon | (b) boundary segment | (c) corner arc

Figure 1: A corner arc can be obtained from a boundary segment by imagining that the segment is made of elastic and pulling the end of the segment along the far object, away from the visibility polygon, until it becomes tangent to the far object.

Consider two boundary segments b_l and b_r that both end on a single far object F, such that the portion of the boundary of F between them is not seen by the observer. The **shared corner arc** of b_l and b_r is the shortest path homotopic to the curve that follows b_l from its tangent point to the far object, that then follows the boundary of the far object to b_r, and then follows b_r to its tangent point.

To compute the time for a visibility event (namely a change in combinatorial structure of the visibility polygon), it suffices to either compute the time that two neighboring boundary segments coalesce, or the time that the first object after the tangent object along a corner arc becomes tangent to the associated boundary segment.

The spatial data structure of the corner arc algorithm consists of three elements: 1) boundary segments, which encode the visible objects; 2) corner arcs, which permit events relevant to boundary segments to be computed; and 3) a pseudo-triangulation, which includes the bitangents along all the corner arcs. The video describes the SegSide procedure, which modifies a pseudo-triangulation to include the bitangents of the corner arc of any given boundary segment. This procedure forms the basis for both initialization and maintenance of corner arcs. The event handling is described in the video. Further details may be found in [6].

3. AVERAGE CASE COMPLEXITY

The complexity of the corner arc algorithm depends on both the number and cost of the various kinetic updates.

THEOREM 1. *The expected average number of flips per visibility event in the corner arc algorithm is no more than 19.046, regardless of the spacing s.*

LEMMA 5. *The number of pseudo-triangle events that occur if the objects move a distance of s that arise due to visibility maintenance near the observer is $O(s^3 \log^2 s)$.*

Combining these and other costs:

THEOREM 2. *The expected average cost per change in visibility in the average case dynamic scene, including the costs of non-visibility related events, is $O(\log^2 n)$.*

Acknowledgements

O. Hall-Holt is partially supported by the National Science Foundation (DMS-9983196).

4. REFERENCES

[1] P. K. Agarwal, J. Basch, L. J. Guibas, J. Hershberger, and L. Zhang. Deformable free space tiling for kinetic collision detection. In *Proc. 4th Workshop Algorithmic Found. Robot.*, 2000.

[2] J. Basch, L. J. Guibas, and J. Hershberger. Data structures for mobile data. *J. Algorithms*, 1:1–28, 1999.

[3] D. Cohen-Or, Y. Chrysanthou, and C. T. Silva. A Survey of Visibility for Walkthrough Applications. In *SIGGRAPH 2000 Course Notes: Visibility, problems, techniques and applications*, 2000.

[4] D. Cohen-Or, G. Fibich, D. Halperin and E. Zadicario. Conservative visibility and strong occlusion for viewspace partitioning of densely occluded scenes *Comput. Graph. Forum*, 17:C243–C253, 1998. Eurographics '98.

[5] F. Durand, R. Orti, S. Rivière, and C. Puech. Radiosity for dynamic scenes in flatland with the visibility complex. *Comput. Graph. Forum*, 16:237–249, 1996. Eurographics '96.

[6] O. A. Hall-Holt. Kinetic Visibility. Ph.D. Dissertation, Computer Science, Stanford University, California, USA, Aug. 2002. `http://graphics.stanford.edu/~olaf/d_orig.pdf`

[7] H. Hubschman and S. W. Zucker. Frame-to-frame coherence and the hidden surface computation: constraints for a convex world. *ACM Trans. Graph.* 1:129–162, 1982.

[8] K. Nechvíle and Petr Tobola. Dynamic Visibility in the Plane. In *Seventh Int. Conf. in Central Europe on Computer Graphics and Visualization, WSCG '99*, Feb. 1999.

[9] M. Pocchiola and G. Vegter. Topologically sweeping visibility complexes via pseudo-triangulations. *Discrete Comput. Geom.* 16:419–453, Dec. 1996.

[10] S. Rivière. Dynamic visibility in polygonal scenes with the visibility complex. In *Proc. 13th ACM Sympos. Comput. Geom.*, pages 421–423, 1997.

[11] I. E. Sutherland, R. F. Sproull, and R. A. Schumacker. A Characterization of Ten Hidden-Surface Algorithms *ACM Comput. Surv.*, 6(1):1–55, Mar. 1974.