COMPUTING THE FULL VISIBILITY GRAPH OF A SET OF LINE SEGMENTS

S. K. WISMATH

DEPARTMENT OF MATHEMATICS AND COMPUTER SCIENCE, UNIVERSITY OF LETHBRIDGE, LETHBRIDGE, ALTA., CANADA, T1K-3M4. *

Abstract. Let S be a collection of n non-intersecting line segments in the plane in general position. Two segments u and v are defined as *visible* if a line segment could be drawn from some point of u to some point of v that intersects no other segment in S. The full visibility graph associated with S is denoted as $G(S)$, and defined to be the graph whose n vertices correspond to the line segments of S and whose edge set represents the visibility relation between pairs of segments. A worst case optimal $O(n^2)$ time and space algorithm for computing $G(S)$ is presented.

Keywords. computational geometry, visibility graph

1. Introduction. Problems involving the *visibility* of objects have arisen in several areas of computer science, for example, graphics, VLSI layout, motion planning, and computational geometry. Frequently, the underlying structure of the visibilities is critical and a graph can be created that condenses this structural information into a more usable form. In general, the nodes of such a *visibility graph* correspond to objects and the edges represent the visibilities between pairs of objects, two objects being visible if a line segment could be drawn between them, intersecting no other object. Alternate notions of visibility have been considered [7], and various types of objects have been examined, for example, curves, line segments, endpoints of line segments [9], and the sides of polygons [6], [8]. One of the most fundamental visibility problems, and the one investigated here, is to compute the visibility graph associated with a set of line segments.

Given a set S of n non-intersecting line segments in the plane in general position, the *full visibility graph*, $G(S)$, has n vertices, associated with the line segments, and an edge set which represents the following visibility relation. The definition of visibility used throughout is that two line segments, u and v are *visible* if there exists a point on u and a point on v such that the line segment defined by these two points intersects no other segment of S – see Figure 1. (The term "weak visibility" has sometimes been used to describe this relation [2]). In section 2, a worst case optimal algorithm for the computation of $G(S)$, from S, is presented.

A related problem, motivated by shortest path construction, has been previously studied, in which the vertices of the visibility graph correspond to the $2n$ *endpoints* of the line segments and the edges correspond to visibilities between pairs of these endpoints. Optimal $O(n^2)$ time algorithms for the construction of these endpoint visibility graphs have been reported by Asano *et al* [1], and Welzl,[9]; and an output sensitive algorithm which performs in $O(E + n \log n)$ was presented by Ghosh and Mount [4]. As these endpoint visibility graphs have previously been labelled in the literature as visibility

* This research was supported by N.S.E.R.C. grant OGPIN 007

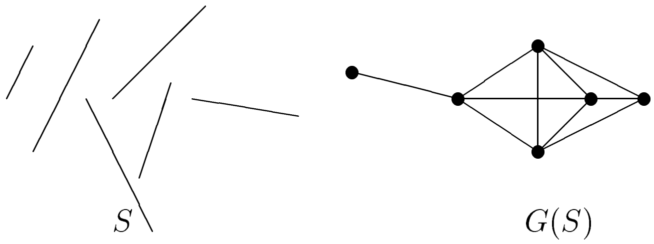

FIG. 1. *Visibility graph of a set of line segments.*

graphs, the term *full visibility graph* will be used in this paper to distinguish between the two cases. Recently, Ghosh and Mount [5] have outlined how their algorithm to compute the endpoint visibility graph can be modified to compute the full visibility graph in $O(E + n \log n)$ time. However, their algorithm relies on sophisticated data structures (finger search trees) and a complicated *Split* procedure and hence is somewhat impractical. The approach in this paper, although inferior in terms of running time, is significantly simpler and easily implemented.

Note that an $O(n^3)$ time brute force algorithm is available for the endpoint visibility case; however, when the visibilities are not restricted to the endpoints, but may appear anywhere along the segments, then no algorithm is obvious – some method of discretizing the problem must be found.

In [2], Avis and Toussaint developed an $O(n)$ algorithm to determine the weak visibility of an n-sided polygon from a given *single* edge of the polygon.

If the visibilities are restricted to lie along a single direction, then a simple $O(n \log n)$ line sweep algorithm can construct the associated visibility graph. Parallel algorithms have also been reported; in [3], the line segments are constrained to be parallel with only the perpendicular visibilities considered, and an $O(\log n)$ algorithm using $O(n)$ processors is given.

2. Computation of the full visibility graph. In this section, the main contribution of the paper is presented, namely, the efficient computation of the full visibility graph of a set of line segments. This algorithm relies on a result of Asano, Asano, Guibas, Hershberger and Imai, [1], and on an observation about visibilities. Consider the following query problem:

> Given a set S of n non-intersecting line segments in the plane and an arbitrary query point q, determine the parts of the segments of S visible from q (called the *visibility polygon* of q).

Asano *et al* [1], solve this query problem in $O(n)$ time, given $O(n^2)$ preprocessing and $O(n^2)$ space. For a given query point q, the (star-shaped) visibility polygon produced by the algorithm of Asano *et al* will be denoted as $VPoly(q)$. The *endpoint* visibility graph can be readily obtained in $O(n^2)$ time using this query result by substituting each endpoint as the query point, as argued in [1]. However, the computation of the *full* visibility graph requires a further insight into the nature of visibility.

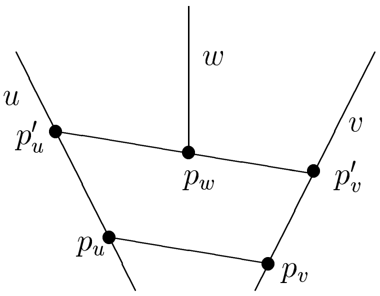

FIG. 2. *Visibility around* p_w

Observation: Let p_u and p_v be a pair of visible points on line segments u and v respectively. Consider the effect of translating this visibility between u and v, continuously and parallel to (p_u, p_v). There are three events that may occur:

- An endpoint of segment u is reached, in which case segment v is visible from this endpoint of u.
- An endpoint of segment v is reached, in which case segment u is visible from this endpoint of v.
- An endpoint p_w of another segment w blocks the visibility of u and v at p'_u and p'_v, where (p'_u, p'_v) is parallel to (p_u, p_v). In this case, we say u and v are visible *around* p_w since there exist visibility line segments between u and v arbitrarily close to p_w. See Figure 2.

It is the exploitation of this observation that permits the problem to be discretized. All visibilities may be determined by examining only the visibility polygons of all the endpoints of the line segments. Consider an endpoint q of some line segment with visibility polygon $VPoly(q)$, and let the vertices of $VPoly(q)$ be in polar order $v_1^q, v_2^q, \ldots v_m^q$, as produced by the algorithm of Asano *et al.*[1] Let $segment(v_i^q)$ denote the line segment of S containing vertex v_i^q.

Note the structure of $VPoly(q)$: If v_i^q is the endpoint of some segment of S, then either v_{i-1}^q or v_{i+1}^q is *not* an endpoint of any segment and furthermore, q, v_i^q and this vertex are collinear. It is clear that the line segment with endpoint q must be reported as visible to $segment(v_1^q)$, $segment(v_2^q)$, $\ldots$ $segment(v_m^q)$ and furthermore this can be performed in time linear in the size of $VPoly(q)$. It is also necessary to report the visibilities *around* q. These visibilities can be computed in linear time by performing a "rotational sweep" around $VPoly(q)$ as follows – refer to Figure 3. Angles are measured in a counter-clockwise fashion.

[1] The visibility polygon $VPoly(q)$ may be unbounded, in which case we assume there are "points at infinity" which serve to close the polygon with artificial sides. When the algorithm reports a visibility with one of these spurious sides, the report may be safely ignored.

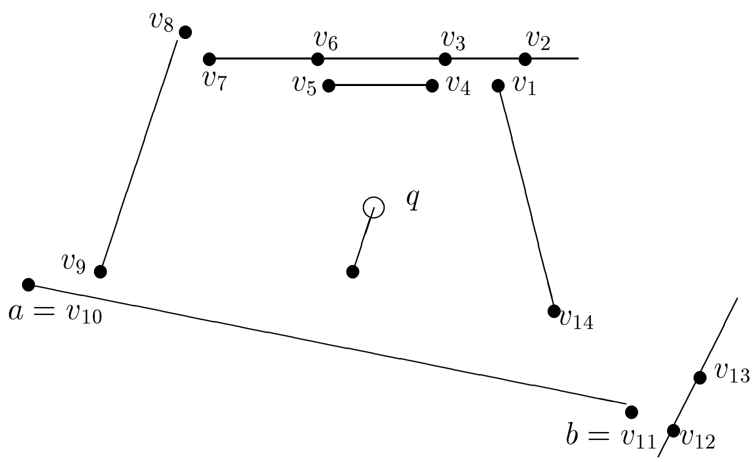

FIG. 3. *Rotational Sweep about q*

Rotational Sweep Algorithm

$i := 1$
$a := \max_j\{\text{angle}(v_1 q v_j) < \pi\}$
$b := \min_j\{\text{angle}(v_1 q v_j) > \pi\}$
WHILE $(\text{angle}(v_1 q v_i) < \pi)$ DO

- report *segment*(v_i) visible to *segment*(v_b)
- $i := i + 1$
- WHILE $(\text{angle}(v_i q v_b < \pi))$ DO
  - ∘ $a := b + 1$ (mod m)
  - ∘ $b := b + 2$ (mod m)
  - ∘ report *segment*(v_i) visible to *segment*(v_b).

The algorithm is linear in the size of $VPoly(q)$ since i, a and b each travel half way around $VPoly(q)$ once without backtracking. The correctness of the algorithm is established by observing that any visibility around q could be rotated about q until some vertex of $VPoly(q)$ is encountered.

Note that the algorithm may report some visibilities more than once – in particular when a and b are updated. This redundancy may easily be removed by including a boolean variable, however at some loss of clarity.

The full visibility graph can be constructed from S as follows:

Full Visibility Algorithm

- Preprocess S to construct an arrangement of lines using $O(n^2)$ time and $O(n^2)$ space, as outlined in [1].
- For each endpoint q of each segment u do
  - compute $VPoly(q)$ as described in [1].
  - report as visible u and each segment in $VPoly(q)$.
  - report the visibilities *around* q with the rotational sweep technique previously described.

To prove that all visible pairs of line segments are reported, we appeal to the Observation. Suppose there is some unreported visibility pair. Displace the associated visibility segment as in the Observation to obtain one of the three cases – each of which is covered by the algorithm.

It is clear that only valid visibility pairs are reported by the algorithm – since no three endpoints are collinear by assumption, the "visibilities" reported *around* each endpoint during the rotational sweep step represent true visibilities by a slight displacement of the rotating visibility line. Furthermore, note that this restriction to points in "general position" is not critical to the operation of the algorithm and can be removed with only minor modifications to the rotating sweep step.

2.1. Analysis. The preprocessing step as shown in [1] requires $O(n^2)$ time and space. Furthermore, the $2n$ visibility polygons of the endpoints can then be computed each in $O(n)$ time.

The two final steps of the algorithm can be performed by a single rotational sweep which as previously argued is linear in the size of $VPoly(q)$. The number of visibilities reported by this (combined) step for each endpoint q is exactly twice the number of vertices in $VPoly(q)$, *i.e.* one report of q visible to each segment in $VPoly(q)$ and one report of the pair of segments around q, 180 degrees apart. For each endpoint q, the number of vertices in $VPoly(q)$ is exactly twice the number of *endpoints* visible from q. Summing over all endpoints and letting E' represent the number of edges in the *endpoint* visibility graph, the number of reports generated is thus $4E'$ in total.

As argued by Ghosh and Mount [5], the number of edges in the endpoint visibility graph is asymptotically equal to the number of edges in the full visibility graph and thus the final step of the algorithm performs in (output sensitive) $O(E)$ time, where E measures the number of edges in the *full* visibility graph.

3. Conclusion and open problems. An $O(n^2 + E)$ time and space algorithm was presented that computes the full visibility graph of a set of n non-intersecting line segments in the plane. The algorithm relies on the procedure of Asano *et al* as a preprocessing step to compute the $2n$ visibility polygons of the endpoints of S in $O(n^2)$ time and space. After this preprocessing step, the algorithm runs in $O(E)$ time. This result is optimal in the worst case, however Ghosh and Mount [5] have recently outlined an output sensitive algorithm (based on funnel sequences) to compute $G(S)$ in $O(E + n \log n)$ time. It is possible that the time complexity of the algorithm presented here can be reduced, since the full power of the Asano *et al* preprocessing step is not required - *general* visibility queries are not needed, but rather exactly $2n$ *fixed* queries. The problem of determining the $2n$ visibility polygons of the endpoints of S in $o(n^2)$ time is currently under investigation.

The alternate problems of determining whether a given graph represents the visibility graph of some set of line segments (the reconstruction problem) and the related "characterization of visibility graphs" problem, are also important and only a few restricted cases have been considered – see for example [7] and [10].

4. Acknowledgments. The author would like to thank Mark Keil and also an anonymous referee for pointing out the output sensitivity of the rotating sweep step.

REFERENCES

[1] T. Asano, T. Asano, L. Guibas, J. Hershberger, and H. Imai, Visibility of disjoint polygons, *Algorithmica*, **1**, No. 1 (1986) 49-64.

[2] D. Avis, G. Toussaint, An optimal algorithm for determining the visibility of a polygon from an edge, *IEEE Transactions on Computers*, Vol. c-30, No. 12 (1981) 910-914.

[3] I. Chan, D. Friesen, An optimal parallel algorithm for the vertical segment visibility reporting problem, International Conference on Computing and Information, (1991), Ottawa, *Springer-Verlag Lecture Notes in Computer Science* # 497, 323-334.

[4] S. Ghosh, D. Mount, An output sensitive algorithm for computing visibility graphs, *Proceedings of the IEEE Foundations of Computer Science* (1987) 11-19.

[5] S. Ghosh, D. Mount, An output sensitive algorithm for computing visibility graphs, *SIAM J. of Computing*, Vol. 20, No. 5 (Oct. 1991) 888-910.

[6] J. O'Rourke, *Art Gallery Theorems and Algorithms*, Oxford University Press (1987).

[7] R. Tamassia, I. Tollis, A unified approach to visibility representations of planar graphs, *Discrete and Computational Geometry*, 1 (1986) 321-341.

[8] G. Toussaint, Shortest path solves edge-to-edge visibility in a polygon, *Pattern Recognition Letters*, 4 (1986) 165-170.

[9] E. Welzl, Constructing the visibility graph for n-line segments in O(n^2) time, *Information Processing Letters* 20 (1985), 167-171.

[10] S. K. Wismath, Characterizing bar line-of-sight graphs, *Proceedings of the 1st ACM Symposium on Computational Geometry*, Baltimore (1985) 147-152.