Computational Geometry 42 (2009) 852–864

Contents lists available at ScienceDirect

Computational Geometry: Theory and Applications

www.elsevier.com/locate/comgeo

Visibility queries in a polygonal region

Rajasekhar Inkulu [a,*], Sanjiv Kapoor [b]

[a] Institute for Computational Engineering and Sciences, The University of Texas at Austin, USA
[b] Department of Computer Science, Illinois Institute of Technology, Chicago, USA

ARTICLE INFO

Article history:
Received 3 August 2007
Received in revised form 27 January 2009
Accepted 25 February 2009
Available online 4 March 2009
Communicated by K. Mehlhorn

ABSTRACT

In this paper, we consider the problem of computing the region visible to a query point located in a given polygonal domain. The polygonal domain is specified by a simple polygon with m holes and a total of n vertices. We provide two bounds on the complexity of this problem. One approach constructs a data structure with space complexity $O(n^2)$ in time $O(n^2 \lg n)$ and yields a query time of $O((1+\min(m, |V(q)|)) \lg^2 n + m + |V(q)|)$. Here, $V(q)$ represents the set of vertices of the visibility polygon of a query point q, and $|E|$ denotes the number of edges in the visibility graph. The other approach provides a data structure with space complexity $O(\min(|E|, mn) + n)$ in time $O(T + |E| + n \lg n)$ with the query time of $O(|V(q)| \lg n + m)$. Here, T is the time to triangulate the given polygonal region (which is $O(n + m \lg^{1+\epsilon} m)$ for a small positive constant $\epsilon > 0$). In both of these approaches, $O(m)$ additive factor in the query time is eliminated with an additional $O((\min(|E|, mn))^2)$ space and an additional $O(m(\min(|E|, mn))^2)$ preprocessing time.

1. Introduction

Consider a polygonal region, P, defined by a simple polygon with m holes, where each hole is defined by a simple polygon. Two points inside a polygonal region are visible from each other if their connecting line segment remains completely inside the polygon and does not intersect the holes. The *Visibility Polygon* $V(q)$ of a point q in P is defined as the polygonal boundary of the set of points in P that are visible from q. The *Visibility Polygon Query* problem is to design a data structure for P that, given q, reports $V(q)$.

For a simple polygon with no holes, Bose, Lubiw, and Munro [5] compute the visibility polygon $V(q)$ of a given query point q in time $O(\lg n + |V(q)|)$ with $O(n^3 \lg n)$ preprocessing time and $O(n^3)$ space. Also, the same complexities were achieved in Guibas and Raghavan [4]. Later Aronov, Guibas, Teichmann, and Zhang [1] proposed an algorithm which accomplishes the same with the preprocessing time $O(n^2 \lg n)$, space $O(n^2)$ and query time complexity as $O(\lg^2 n + |V(q)|)$.

For a polygon with holes where there is no query involved, worst-case optimal algorithms for constructing the visibility polygon with total time of $O(n \lg n)$ were presented by Asano [2] and, later by Suri and O'Rourke [12]. This was later improved to $O(n + m \lg m)$ by Heffernan and Mitchell [6]. This problem in the query version was first presented by Pocchiola and Vegter [11]. They have considered the case of a set of convex polygons in the plane and given an algorithm which determines the query polygon $V(q)$ of any query point q in time $O(|V(q)| \lg n)$ by $O(n \lg n)$ preprocessing time and $O(n)$ space. Zarei and Ghodsi [13] considered the case of a polygon (not necessarily convex) with holes and gave an algorithm that finds $V(q)$ with $O(n^3 \lg n)$ preprocessing time and $O(n^3)$ space having query complexity $O((1+m') \lg n + |V(q)|)$, where $m' = \min(m, |V(q)|)$.

* Corresponding author.
E-mail addresses: rinkulu@ices.utexas.edu (R. Inkulu), kapoor@iit.edu (S. Kapoor).

0925-7721/$ – see front matter
doi:10.1016/j.comgeo.2009.02.004

Table 1

	Space	Preprocessing time	Query time
This paper with the approach from [1] as a sub-procedure:			
No extra preprocessing	$O(n^2)$	$O(n^2 \lg n)$	$O((1+\min(m, \lvert V(q)\rvert)) \lg^2 n + m + \lvert V(q)\rvert)$
With extra preprocessing	$O(\min(\lvert E\rvert, mn)^2 + n^2)$	$O(m(\min(\lvert E\rvert, mn))^2 + n^2 \lg n)$	$O((1+\min(m, \lvert V(q)\rvert)) \lg^2 n + \lvert V(q)\rvert)$
This paper with the ray-shooting based approach as a sub-procedure:			
No extra preprocessing	$O(\min(\lvert E\rvert, mn) + n)$	$O(T + \lvert E\rvert + n \lg n)$	$O(\lvert V(q)\rvert \lg n + m)$
With extra preprocessing	$O(n + \min(\lvert E\rvert, mn)^2)$	$O(T + \lvert E\rvert + n \lg n + m(\min(\lvert E\rvert, mn))^2)$	$O(\lvert V(q)\rvert \lg n)$
Zarei and Ghodsi [13]	$O(n^3)$	$O(n^3 \lg n)$	$O((1+\min(m, \lvert V(q)\rvert)) \lg n + \lvert V(q)\rvert)$

We provide a method which covers the space with simple polygons and utilizes a sub-procedure for computing the visibility polygon from a point inside a simple polygon. The computation of visibility queries in a simple polygon has been well researched, and we use either of the following two algorithms as a sub-procedure in our algorithm: one given by Aronov, Guibas, Teichmann, Zhang [1], and the other that uses ray-shooting by Hershberger and Suri [7]. Using the approach from [1] as a sub-procedure, we construct a data structure with space complexity $O(n^2)$ in time $O(n^2 \lg n)$ so that the query complexity is $O((1+\min(m, |V(q)|)) \lg^2 n + m + |V(q)|)$. Here, $V(q)$ represents the vertices of the visibility polygon of a query point q, and $|E|$ is the number of edges in the visibility graph. Using the ray-shooting based approach as a sub-procedure, we construct a data structure with space complexity $O(\min(|E|, mn) + n)$ in time $O(T + |E| + n \lg n)$ which yields a query time of $O(|V(q)| \lg n + m)$. Here, $O(T)$ is the time to triangulate the given polygonal region ($O(n + m \lg^{1+\epsilon} m)$ for a small positive constant $\epsilon > 0$) using the algorithm given by Bar-Yehuda and Chazelle [3]. The preprocessing time and space of our algorithm using either of these sub-procedures improves upon [13]. When $|V(q)| \geqslant m$, our algorithm with the first approach provides a query complexity of $O(m \lg^2 n + |V(q)|)$ close to the $O(m \lg n + |V(q)|)$ query time achieved by Zarei and Ghodsi [13]. Our algorithm is especially useful when the number of holes is a small constant. Moreover, in both of these approaches, the $O(m)$ additive factor in the query time is eliminated altogether with an additional $O((\min(|E|, mn))^2)$ space and an additional $O(m(\min(|E|, mn))^2)$ preprocessing time. Table 1 summarizes the results.

1.1. Overview

The algorithm uses a partition of the space into simple polygons, known as corridors, similar to that in [8,10]. The visibility polygon of q is determined in two phases. In the first phase, we find the sides of corridors that contain at least one visible point. And in the second phase, we find the visible vertices in each such corridor.

When the query point q is inside the corridor C, we apply the algorithm from either [1] or [7] (both determine the visibility inside a simple polygon) to determine the visible region interior to C. To facilitate in determining the visible region for each corridor C' when q is external to C', corridor C' and the region outside C' are pre-processed to construct a set of simple polygons (refer to Section 3.4). During the query time, algorithm from either [1] or [7] is applied on these simple polygons.

Determining the sides of corridors that contain at least one visible vertex could be done by scanning the corridors, which would require considering each of $O(m)$ corridors. A more effective procedure is adopted which traverses relevant corridors and maintains a potentially visible region, that becomes more and more restricted as the query algorithm proceeds. (See Fig. 1.) Initially the region is constructed using the sides of the corridor containing the query point q. This requires locating the corridor containing the query point and constructing supporting lines to the sides of the corridors. To accomplish this, pre-processing is required to set up a planar point location structure. The traversal of relevant corridors is determined by constructing a tree of corridors, each having at least one vertex visible from q (refer to Section 3.5). Achieving the mentioned query time requires a pre-processed data structure, which contains visible supporting lines from each vertex in the polygonal region to the sides of the corridors (refer to Section 3.2). These supporting lines are constructed during pre-processing using the visibility graph construction from Kapoor and Maheshwari [9].

The rest of the paper is organized as follows: Section 2 lists properties and definitions. The data structures used and the preprocessing phase are described in Section 3. The query algorithm to construct the visibility polygon is detailed in Section 4. Section 5 analyzes algorithm for both the time and space complexities. The proof of correctness is given in Section 6. Section 7 provides the conclusions.

2. Properties and definitions

First, we describe the corridor structures used in this paper. These are a slight variation to the corridors defined in [8,10]. Consider a triangulation of the given polygonal region, P, and the dual graph G_D formed from the triangulation. The dual graph is first pruned by iteratively removing vertices of degree one and the edge incident on each such vertex. The subset of edges removed forms a subgraph H, which is a forest. The graph $G_d \backslash H$ has nodes which are of degree two or three. The decomposition into corridors is obtained as follows: identify triangles in P corresponding to vertices of degree three in