Journal of Discrete Algorithms 35 (2015) 51–61

Contents lists available at ScienceDirect

Journal of Discrete Algorithms

www.elsevier.com/locate/jda

Near optimal line segment queries in simple polygons

Mojtaba Nouri Bygi [a,*], Mohammad Ghodsi [a,b]

[a] Computer Engineering Department, Sharif University of Technology, Iran
[b] School of Computer Science, Institute for Research in Fundamental Sciences (IPM), Iran

ARTICLE INFO

Article history:
Received 1 August 2014
Accepted 3 October 2015
Available online 8 October 2015

Keywords:
Computational geometry
Visibility
Line segment visibility

ABSTRACT

This paper considers the problem of computing the weak visibility polygon (*WVP*) of any query line segment pq (or $WVP(pq)$) inside a given simple polygon P. We present an algorithm that preprocesses P and creates a data structure from which $WVP(pq)$ is efficiently reported in an output sensitive manner.
Our algorithm needs $O(n^2 \log n)$ time and $O(n^2)$ space in the preprocessing phase to report $WVP(pq)$ of any query line segment pq in time $O(|WVP(pq)| + \log^2 n + \kappa \log^2(\frac{n}{\kappa}))$, where κ is an input and output sensitive parameter of at most $|WVP(pq)|$. We improve the preprocessing time and space of current results for this problem [11,6] at the expense of more query time.

1. Introduction

Two points inside a polygon P are *visible* to each other if their connecting segment remains completely inside P. The *visibility polygon* (*VP*) of a point q inside P (or $VP(q)$) is the set of vertices of P that are visible from q. There have been many studies on computing *VP*'s in simple polygons. In a simple polygon P with n vertices, $VP(q)$ can be reported in time $O(\log n + |VP(q)|)$ by spending $O(n^3 \log n)$ time and $O(n^3)$ of preprocessing space [2]. This result was later improved by [1] where the preprocessing time and space were reduced to $O(n^2 \log n)$ and $O(n^2)$ respectively, at the expense of more query time of $O(\log^2 n + |VP(q)|)$.

The visibility problem has also been considered for line segments. A point v is said to be *weakly visible* from a line segment pq if there exists a point $w \in pq$ such that w and v are visible to each other. The problem of computing the *weak visibility polygon* of pq (or $WVP(pq)$) inside P is to compute all points of P that are weakly visible from pq. If P is simple (with no holes), Chazelle and Guibas [5] gave an $O(n \log n)$ time algorithm for this problem. Guibas et al. [9] showed that this problem can be solved in $O(n)$ time if a triangulation of P is given along with P. Since any P can be triangulated in $O(n)$ [4], the algorithm of Guibas et al. always runs in $O(n)$ time [9]. Another linear time solution was obtained independently by [13].

The *WV* problem in the query version has been considered by few. It was shown in [2] that a simple polygon P can be preprocessed in $O(n^3 \log n)$ time and $O(n^3)$ space such that, given an arbitrary query line segment inside P, $O(k \log n)$ time is required to recover k weakly visible vertices. This result was later improved by [1] in which the preprocessing time and space were reduced to $O(n^2 \log n)$ and $O(n^2)$ respectively, at expense of more query time of $O(k \log^2 n)$. In a recent work, we presented an algorithm to report $WVP(pq)$ of any pq in $O(\log n + |WVP(pq)|)$ time by spending $O(n^3 \log n)$ time and $O(n^3)$ space for preprocessing [11]. Later, Chen and Wang considered the same problem and, by improving the preprocessing

* Corresponding author.
E-mail addresses: nouribygi@ce.sharif.edu (M. Nouri Bygi), ghodsi@sharif.edu (M. Ghodsi).

http://dx.doi.org/10.1016/j.jda.2015.10.002
1570-8667/

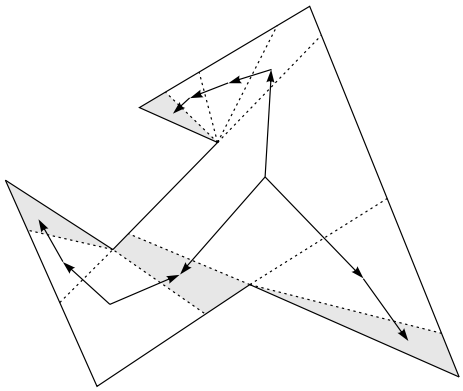

Fig. 1. The visibility decomposition induced by the critical constraint edges and its dual graph. The sink regions are shown in gray.

time of the visibility algorithm of Bose et al. [2], they improved the preprocessing time to $O(n^3)$ [6]. Alternatively, they showed how to build a data structure of size $O(n)$ in time $O(n)$ which can answer weak visibility queries in $O(k \log n)$ time [7].

In this paper, we show that *WVP* of a line segment pq can be reported in near optimal time of $O(|WVP(pq)| + \log^2 n + \kappa \log^2(n/\kappa))$, after preprocessing the input polygon in time and space of $O(n^2 \log n)$ and $O(n^2)$ respectively, where κ is an input and output sensitive parameter of at most $|WVP(pq)|$. Compared to the algorithms of [11] and [6], the storage and preprocessing time has one fewer linear factor, at expense of more query time. Our approach is inspired by Aronov et al.'s algorithm for computing the visibility polygon of a point [1]. In Section 3, we first show how to compute the *partial weak visibility polygon PWVP*(pq). Then, in Section 4, we use a balanced triangulation to compute and report the final weak visibility polygon.

2. Preliminaries

In this section, we introduce some basic terminologies used throughout the paper. For a better introduction to these terms, we refer the readers to Guibas et al. [9], Bose et al. [2], and Aronov et al. [1]. For simplicity, we assume that no three vertices of the polygon are collinear.

2.1. Visibility decomposition

Let P be a simple polygon with n vertices. Also, let p and q be two points inside P. The *visibility* of a point p is the cyclical sequence of vertices and edges of P that are visible from p. A *visibility decomposition* of P is to partition P into a set of *visibility regions*, such that, for each region, any point inside the region has the same visibility, which we call the visibility of the region. This partition is induced by the *critical constraint edges*. A critical constraint edge is a line segment inside and limited by the boundary of P, which passes through two vertices of P, and is *tangent* to P at one (or both) of these two vertices. A line is tangent to P at a vertex v if it passes through v and is on the same side of P before and after the pass.

In a simple polygon, the visibility of two *neighboring* visibility regions which are separated by an edge, differs only in one vertex. This fact is used to reduce the space complexity of maintaining the visibility of the regions [2]. This is done by defining the *sink regions*. A sink is a region with the smallest visibility compared to all of its adjacent regions. Therefore, it is sufficient to maintain the visibility of the sinks, from which the visibility of all other regions can be computed. By constructing a directed dual graph over the visibility regions (see Fig. 1), one can maintain the difference between the visibility of the neighboring regions [2].

In a simple polygon with n vertices, the number of visibility and sink regions are $O(n^3)$ and $O(n^2)$, respectively [2].

2.2. A linear time algorithm for computing WVP

Here, we present the $O(n)$ time algorithm of Guibas et al. for computing $WVP(pq)$ of a line segment pq inside P, as described in [8]. This algorithm is used in computing the partial weak visibility polygons in an output sensitive way, to be explained in Section 3.2. For simplicity, we assume that pq is a convex edge of P, but we will show that this can be extended for any line segment in the polygon.

Let $SPT(p)$ denote the shortest path tree in P rooted at p. The algorithm traverses $SPT(p)$ using a DFS and checks the turn at each vertex v_i in $SPT(p)$. If the path makes a right turn at v_i, then we find the descendant of v_i in the tree with the largest index j (see Fig. 2). As there is no vertex between v_j and v_{j+1}, we can compute the intersection point z of $v_j v_{j+1}$ and $v_k v_i$ in $O(1)$ time, where v_k is the parent of v_i in $SPT(p)$. Finally the counter-clockwise boundary of P is removed from v_i to z by inserting the segment $v_i z$.