Expert Systems With Applications 45 (2016) 286–293

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

A linear time algorithm for minimum conic link path in a simple polygon

Shuo-Yan Chou [a], Chang-Chien Chou [b,*]

[a] Department of Industrial Management, National Taiwan University of Science and Technology, Taiwan ROC
[b] Department of Information Management, Lunghwa University of Science and Technology, Taiwan ROC

ARTICLE INFO

Keywords:
Path planning
Conic link path
Conic visibility
Circular visibility
Splinegon
Computational geometry

ABSTRACT

We consider the problem of finding, in a simple polygon, from a starting point to a destination point, a piecewise path consisting of conic sections. By considering only one type of conic section, i.e., circular, elliptic, hyperbolic, or parabolic curves, we present an O(n) time algorithm for computing the path with the minimum number of conic sections. The studied problem is the generalization of the straight line link path version. The results can be conducted in versatile applications: the hidden surface removal problem in CAD/CAM, the contour tracing, the red-blue intersection problem, the robot motion planning, and related computational geometry applications. The linear time property is most vital for those applications need to take instant reaction.

1. Introduction

The lion and man problem is one famous game concerning the visibility search problem. A lion tries to capture a man who is escaping in a circular arena. The players have equal speeds. They can observe each other all the time. Noori and Isler (2014) studied a variant of the game in which the lion has only line-of-sight visibility in a polygonal area. That is, it can observe the man's position only if the line segment connecting them does not intersect the boundary of the polygon. They showed that a single deterministic pursuer with line-of-sight visibility constraint can capture an evader whose speed is equal to the pursuer's in any monotone polygon.

The visibility problem has been widely studied in the fields of computational geometry (Alsuwaiyel & Lee, 1993, 1995; Chandru, Ghosh, Maheshwari, Rajan, & Saluja, 1995; Li & Klette, 2008). Until Suri proposed a linear time algorithm in 1986, the visibility problem of the straight line segment in a simple polygon had been treated thoroughly. Suri's method begins with the triangulation of the simple polygon. A tree structure is constructed after the triangulation which can be regarded as the dual graph of the triangulated simple polygon. A dominant property in the tree structure is that: among every paired vertices (from top to leaf), there exists only one unique path. This property reigned the dominance in Suri's linear time algorithm. Maheshwari and Sack (1999) proposed an optimal algorithm for computing the minimum rectilinear link path in a rectilinear polygon. Despite of the simple polygon, when there were holes in the polygon, Mitchell, Rote, and Woeginger (1992) proposed an improved method. Fig. 1 is the example where obstacles appear between the starting and the destination points in a polygon.

Mitchell's method is basically incrementally computing the visible region of the line segment. The algorithm can construct the minimum link path with its length in O($E\alpha(n)\log^2 n$) by using O(E) data structure where n is the number of edges of the obstacles, E is the size of the visibility graph, $\alpha(n)$ is a slow increasing function which is the inversion of the Ackermann function. Furthermore, Fekete, Mitchell, and Schmidt (2012) adopt the visibility problem of line segments into the myopic watchman problem with discrete vision (MWPDV) as shown in Fig. 2.

The myopic watchman needs to walk along a rectilinear area and returns back to the starting point. On the routing, (i) all the points in the polygon must been seen and (ii) the sum of the tour length and the weighted number of monitors must be the minimum. They points out that MWPDV is NP-hard. Mitchell, Polishchuk, and Sysikaski (2014) studied a lot about the link paths problems.

Recently, Klein and Suri (2015) investigate the lion-and-man problem in the visibility-based discrete-time model of pursuit evasion in the plane: how many pursuers are needed to capture an evader in a polygonal environment with obstacles under the minimalist assumption that pursuers and the evader have the same maximum speed? They proved a tight bound of $\Theta\sqrt{n}$ when the environment is a simple polygon with n vertices.

Yang et al. (2014) present an algorithm to compute the weak visibility polygons (WVP) of NonUniform Rational B-spline (NURBS) curves inside simple polygons. In that paper (Yang et al., 2014), the NURBS curve is first subdivided into triangular curves. They then compute the WVP of each triangular curve by shearing that of its

* Corresponding author. Tel.: +886 2 8209 3211.
E-mail addresses: sychou@mail.ntust.edu.tw (S.-Y. Chou), samuelchou7@yahoo.com.tw (C.-C. Chou).

http://dx.doi.org/10.1016/j.eswa.2015.09.034
0957-4174/

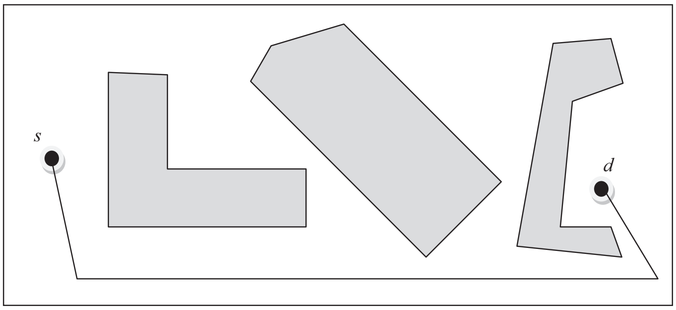

Fig. 1. A polygon with obstacles, there are at least 3 links from s to d.

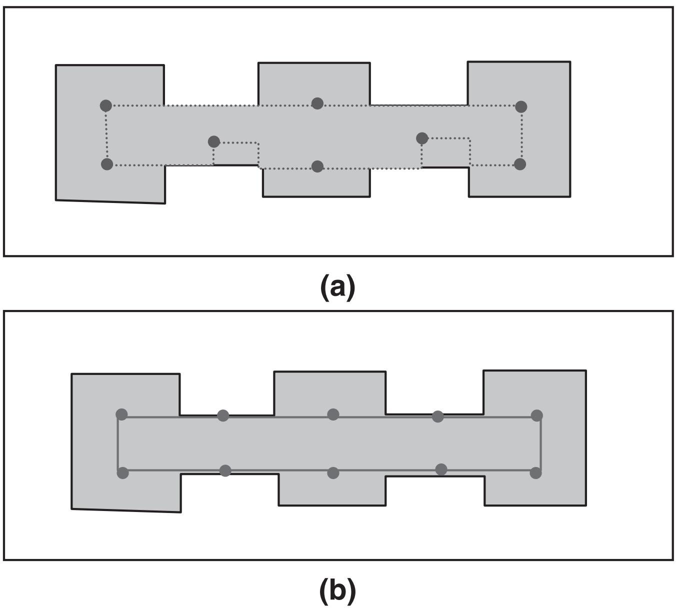

Fig. 2. (a) One solution of MWPDV with minimum monitor; (b) solution of MWPDV with shortest path.

triangle hull. Finally, all triangular curves' WVPs are merged together to obtain the WVP of the NURBS curve. Yang's paper is to compute the WVPs of one given NURBS curve segment. This is not to compute the NURBS curve form a given point to a destination point in a polygon like our work did. It essentially is also straight line visibility emanating from points on an NURBS curve segment. However, only Yang's paper has better correspondence to our work rather than the other papers mentioned above. All literatures are mostly discussing about straight line visibility. There is no literature discussing the studied problem proposed by this paper so far.

Since straight lines are degenerate conic sections, once the solutions are obtained in conical version, the applications are essentially broaden against the linear version. The algorithm proposed in this paper is the generalization of the linear link path problem (Suri, 1986). There are many physical phenomena that can be characterized by conic sections: the parabolic trajectories of objects under gravity and particles traveling in the electric and magnetic fields.

A segment of a conic section is called a link. The problem studied in this paper is to compute a conic link path (*CLP*) with the minimum number of links from a starting point to a destination point in a simple polygon. The conic link paths consist of one of the four types of conic sections: circular, elliptic, hyperbolic, or parabolic curves. The traces of conics can be considered as trajectories of visibility and had been used to establish conic visibility for years. Among the conic visibility problems, the most-studied one is the circular visibility problem which was introduced by Agarwal and Sharir (1993). In their

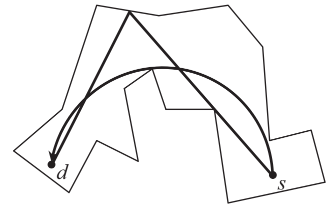

Fig. 3. Illustration of *MLLP* and *MCLP* in a simple polygon.

paper (Agarwal & Sharir, 1993), the time to compute the circular visibility polygon (*CVP*) can be achieved in $O(n\log n)$ time and the first intersection from a point to an edge of the simple polygon can be carried out in $O(\log n)$ time after an $O(n\log n)$ preprocessing. Chou and Woo (1995) improved the above result by developing a linear time algorithm to compute the *CVP* from a point inside a simple polygon.

García-López and Ramos (2000) further proposed a unified approach to compute the *CVP* comprising all four types of conic curves inside a simple polygon. They took advantage of *inversion* to transform one conic visibility problem to one corresponding visibility problem of straight lines emanating from ∞ to a *splinegon* (to be defined later). As the corresponding inversed linear visibility problem can be solved in linear time, the original *CVP* problem can be solved optimally in linear time accordingly.

The linear link path (*LLP*) in a simple polygon refers to the set of piecewise line segments connecting from the starting point to the destination point. The *LLP* with the minimum number of line segments is called the minimum linear link path (*MLLP*). In this work, the link path problem is extended by considering conic segments and thus defining the problem of finding the minimum conic link path (*MCLP*) in a simple polygon. Since straight lines are degenerate conics, *LLP*s are therefore degenerate *CLP*s. Due to the enhanced reach-ability resulting from conics, the number of the links of *MCLP* is in general less than or equal to that of *MLLP*. Fig. 3 depicts the case of a simple polygon for which *MLLP* contains two links whereas *MCLP* contains only one link. We have Lemma 1.

Lemma 1. *The number of links in MCLP presented by* $\#Links(MCLP)$ *is less than or equal to the number of links in MLLP in the identical polygon with the same starting and destination points, i.e.,*

$$\#Links(MCLP) \leq \#Links(MLLP).$$

This paper utilizes the approaches developed by García-López and Ramos (2000) as an internal loop computing engine and then develops a linear time algorithm for computing the *MCLP* in a simple polygon. Since conic sections own the ability to "make a turn", the results from this paper can be conducted in a variety of applications: such as, the hidden surface removal (Halperin & Overmars, 1998; Sando, Barada, & Yatagai, 2013), the contour tracing (Liu, Tang, Yi, & He, 2013), the conic version of red-blue intersection (Agarwal & Sharir, 1990), and the robot motion planning (Boissonnat, Ghosh, Kavitha, & Lazard, 2002; Wang & Cao, 2014).

The rest of this paper is organized as follows. Section 2 describes the fundamentals of this work. Section 3 gives the algorithms and the proofs of their correctness and optimality along with an example. Section 4 concludes this paper and points out some potential applications and the future works.

2. Preliminaries

Let P denote the given polygon and $conic(i, j)$ denotes the conic segments connecting two points i and j. For any point $x \in P$, the *CVP* from x, denoted by $V_c(x)$, is the sets of points in P visible from x, i.e.,

$$V_C(x) = \{z \in P | conic(x, z) \cap P = conic(x, z)\}.$$