Journal of Discrete Algorithms 44 (2017) 39–47

Contents lists available at ScienceDirect

Journal of Discrete Algorithms

www.elsevier.com/locate/jda

Guarding monotone art galleries with sliding cameras in linear time ☆

Mark de Berg [a,1], Stephane Durocher [b,2], Saeed Mehrabi [c,*]

[a] Department of Mathematics and Computer Science, TU Eindhoven, The Netherlands
[b] Department of Computer Science, University of Manitoba, Canada
[c] Cheriton School of Computer Science, University of Waterloo, Canada

ARTICLE INFO

Article history:
Received 8 November 2016
Received in revised form 26 April 2017
Accepted 27 April 2017
Available online 22 May 2017

Keywords:
Art gallery problem
Orthogonal polygons
Sliding cameras
Dynamic programming

ABSTRACT

A sliding camera in an orthogonal polygon P—that is, a polygon all of whose edges are axis-parallel—is a point guard g that travels back and forth along an axis-parallel line segment s inside P. A point p in P is guarded by g if and only if there exists a point q on s such that line segment pq is normal to s and contained in P. In the minimum sliding cameras (MSC) problem, the objective is to guard P with the minimum number of sliding cameras. We give a dynamic programming algorithm that solves the MSC problem exactly on monotone orthogonal polygons in $O(n)$ time, improving the 2-approximation algorithm of Katz and Morgenstern (2011). More generally, our algorithm can be used to solve the MSC problem in $O(n)$ time on simple orthogonal polygons P for which the dual graph induced by the vertical decomposition of P is a path. Our results provide the first polynomial-time exact algorithms for the MSC problem on a non-trivial subclass of orthogonal polygons.

1. Introduction

Art gallery problems, introduced by Klee in 1973 [23], are some of the most widely studied problems in computational geometry. In the original formulation of the problem, an input polygon P is given, for which a set of point guards must be assigned, using as few guards as possible. Thus, the objective is to find a set of point guards such that every point in P is seen by at least one of the guards, where a guard g sees a point p if and only if the segment gp is contained in P. Chvátal [9] proved that $\lfloor n/3 \rfloor$ point guards are always sufficient and sometimes necessary to guard a simple polygon with n vertices. Lee and Lin [20] showed that finding the minimum number of point guards needed to guard an arbitrary polygon is NP-hard for arbitrary polygons. The art gallery problem is also NP-hard for orthogonal polygons [24] and it even remains NP-hard for monotone polygons [19]. Moreover, Eidenbenz [13] proved that the problem is APX-hard on simple polygons.

Ghosh [14] gave an $O(\log n)$-approximation algorithm that runs in $O(n^4)$ time on simple polygons. King and Kirkpatrick [17] gave an $O(\log \log \mathrm{OPT})$-approximation algorithm for the vertex guards (and in fact when the guards can be anywhere on the boundary of the polygon), where OPT is the size of an optimal solution. Their algorithm is based on the

☆ A preliminary version of this paper appeared in proceedings of the 8th International Conference on Combinatorial Optimization and Applications (COCOA 2014) [11].

* Corresponding author.
E-mail addresses: mdberg@win.tue.nl (M. de Berg), durocher@cs.umanitoba.ca (S. Durocher), smehrabi@uwaterloo.ca (S. Mehrabi).

[1] Work of the author is supported by the Netherlands Organisation for Scientific Research (NWO) under project 024.002.003.
[2] Work of the author is supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC).

http://dx.doi.org/10.1016/j.jda.2017.04.005
1570-8667/

fact that there is an ϵ-net of size $O(\frac{1}{\epsilon}\log\log\frac{1}{\epsilon})$ for the corresponding hitting set problem. Notice that the existence of such an ϵ-net along with the technique of Bronnimann and Goodrich [7] provides the desired approximation factor. King [16] improved the running time of this algorithm to $O(n^3)$ for simple polygons. Krohn and Nilsson [19] gave a constant-factor approximation algorithm on monotone polygons. They also gave a polynomial-time algorithm for the orthogonal art gallery problem that computes a solution of size $O(\mathrm{OPT}^2)$, where OPT is the cardinality of an optimal solution. In terms of parameterized complexity of the art gallery problem, Bonnet and Miltzow [6] recently showed that the problem is W[1]-hard parameterized by the number of guards. See the surveys by O'Rourke [23] or Urrutia [26] for a history of the art gallery problem.

Many variants of the art gallery problem have been studied [1–3,22,27]. The version in which we are interested was introduced recently by Katz and Morgenstern [15], and it concerns *sliding cameras* in orthogonal polygons. A sliding camera in an orthogonal polygon P is a point guard that travels back and forth along an axis-parallel line segment $s \subset P$. The point guard can see a point $p \in P$ if and only if there is a point $q \in s$ such that the line segment pq is normal to s and contained in P. The *minimum sliding cameras (MSC) problem* is to guard P with the minimum number of sliding cameras.

Related work Katz and Morgenstern [15] first considered a restricted version of the MSC problem in which only *vertically sliding cameras* are allowed; that is, point guards that travel back and forth along a segment s that is parallel to the y-axis. They solved this restricted version in polynomial time for simple orthogonal polygons. For the unrestricted version, where both vertically and horizontal sliding cameras are allowed, they gave a 2-approximation algorithm for x-monotone orthogonal polygons. An orthogonal polygon P is x-monotone if the intersection of P with any every vertical line is connected. Durocher and Mehrabi [12] showed that the MSC problem is NP-hard when the polygon P is allowed to have holes. They also considered a variant of the problem, called the MLSC problem, in which the objective is to minimize the sum of the lengths of line segments along which cameras travel, and proved that the MLSC problem is polynomial-time solvable even on orthogonal polygons with holes (see also [21]). For orthogonal polygons with holes, Biedl et al. [5] showed that the problem is still NP-hard if only horizontal sliding cameras are allowed. They also gave an $O(1)$-approximation algorithm for the MSC problem based on ϵ-nets, and showed that the problem becomes polynomial-time solvable if the dual graph of a so-called pixelation of the polygon has bounded treewidth.

Biedl et al. [4] studied the MSC problem under k-visibility; that is, the line of sight of a camera can intersect the boundary of the polygon at most k times (note that when $k = 0$, we have the standard MSC problem studied in this paper). The k-visibility has been already studied under the classical art gallery problem [1,2,8]. Biedl et al. [4] showed that the MSC problem under k-visibility is NP-hard on simple orthogonal polygons for any $k > 0$, even if the polygon is monotone. They also gave an $O(1)$-approximation algorithm for any fixed $k > 0$. Seddighin [25] proved that the MLSC problem (i.e., minimizing the total length of cameras) is NP-hard under k-visibility for any fixed $k > 0$.

Our main interest is in the standard MSC problem, where the objective is to minimize the number of cameras. As discussed above, the complexity of the MSC problem on simple orthogonal polygons remains unknown. Indeed, even for x-monotone orthogonal polygons there is only an approximation algorithm for the problem. Recall that the classical art gallery problem is NP-hard on simple orthogonal polygons [24], simple monotone polygons [19] and even on terrains [18].

Our results In this paper, we give a linear-time dynamic programming algorithm for the MSC problem on orthogonal x-monotone polygons P. This not only improves the 2-approximation algorithm of Katz and Morgenstern [15], but also provides, to the best of our knowledge, the first polynomial-time algorithm for the MSC problem on a non-trivial subclass of orthogonal polygons. We also show how to extend this result to so-called *orthogonal path polygons*. These are orthogonal polygons for which the dual graph induced by the vertical decomposition of P is a path. (The vertical decomposition of an orthogonal polygon P is the decomposition of P into rectangles obtained by extending the vertical edge incident to every reflex vertex of P inward until it hits the boundary of P. The dual graph of the vertical decomposition is the graph that has a node for each rectangle in the decomposition and an edge between two nodes if and only if their corresponding rectangles are adjacent.) Observe that the class of orthogonal monotone polygons is a subclass of orthogonal path polygons.

2. Preliminaries

A polygon is orthogonal if all of its edges are axis-parallel. A simple orthogonal polygon P is *x-monotone* if the intersection of P with any vertical line is at most one single line segment. For a simple orthogonal and x-monotone polygon P, the leftmost and rightmost vertical edges of P are unique and we denote them by leftEdge(P) and rightEdge(P), respectively. For a sliding camera s in P, we define the *visibility polygon* of s as the maximal subpolygon $P(s)$ of P such that every point in $P(s)$ is guarded by s.

Let $V_P = \{e_1 = \text{leftEdge}(P), e_2, \ldots, e_m = \text{rightEdge}(P)\}$, for some $m > 0$, be the set of vertical edges of P ordered from left to right. For simplicity we assume that every two vertical edges in V_P have distinct x-coordinates, but it is easy to adapt the algorithm to handle degenerate cases. Let P_i^+ (resp., P_i^-), for some $1 \le i \le m$, denote the subpolygon of P that lies to the right (resp., to the left) of the vertical line through e_i.

For an axis-parallel line segment s in P, we denote the left endpoint and the right endpoint of s by left(s) and right(s), respectively. If s is vertical, we define its left and right endpoints to be its upper and lower endpoints, respectively. We denote the x-coordinate of a point p by $x(p)$. Let s_i and s_j be two horizontal line segments in P. We define the *overlap*