ARTICLE IN PRESS
COMGEO:1475

Computational Geometry ••• (••••) •••–•••

Contents lists available at ScienceDirect

Computational Geometry: Theory and Applications

www.elsevier.com/locate/comgeo

Routing in a polygonal terrain with the shortest beacon watchtower ☆

Bahram Kouhestani *, David Rappaport, Kai Salomaa

School of Computing, Queen's University, Canada

A R T I C L E   I N F O

Article history:
Received 2 March 2015
Accepted 22 March 2017
Available online xxxx

This paper is dedicated to the memory of our good friend and colleague Ferran Hurtado.

Keywords:
Beacon attraction
Greedy routing
Beacon kernel
Inverse beacon kernel
Beacon watchtower

A B S T R A C T

In a paper by Biro et al. [3], a novel twist on guarding in art galleries, motivated by geographical greedy routing in sensor networks, is introduced. A beacon is a fixed point that when activated induces a force of attraction that can move points within the environment. The effect of a beacon is similar to standard visibility with some additional properties. The effects of a beacon are asymmetric leading to separate algorithms to compute the "beacon kernel" and "inverse beacon kernel". In Biro [2] $O(n^2)$ time algorithms are given to compute the beacon kernel and the inverse beacon kernel in simple polygons. In this paper we revisit the problem of computing the shortest watchtower to guard a $2D$ terrain, using the properties of beacons, and we present an $O(n \log n)$ time algorithm that computes the shortest beacon watchtower. In doing this we introduce $O(n \log n)$ time algorithms to compute the beacon kernel and the inverse beacon kernel in a monotone polygon. We also prove that $\Omega(n \log n)$ time is a lower bound for computing the beacon kernel of a monotone polygon.

1. Introduction

Consider a dense network of sensors in a rural area. It is common in practice that routing between two points in the network is performed by greedy geographical routing, where a node sends the message to its neighbor which is closest (by Euclidean distance) to the destination [8,10]. Depending on the geometry of the network, greedy routing may not be successful between all pairs of nodes. Thus, it is essential to determine nodes of the network for which this type of routing works. In particular, it is important to compute points that can send a message to all nodes of the network and points that can receive a message from all nodes.

Motivated by this application, Biro et al. [3] introduced the notion of a *beacon*, as a new variation of visibility. A beacon attracts (sees) all points that can successfully send it a message in a greedy routing model. Biro et al. [3] studied the combinatorics of guarding a polygon with beacons and showed that $\lfloor \frac{n}{2} \rfloor - 1$ beacons are sometimes necessary and always sufficient to route between any pair of points in a simple polygon. They also proved that it is NP-hard to find a minimum cardinality set of beacons to cover a simple polygon. In the greedy geographical routing application, the beacon kernel is the set of points that can receive a message from any point of the environment and the inverse beacon kernel is the set of

☆ A preliminary version of this paper appeared in proceedings of the 26th Canadian Conference on Computational Geometry, 2014 [9]. We thank Daniel Goč for noting an error in that paper.

* Corresponding author.
E-mail addresses: kouhesta@cs.queensu.ca (B. Kouhestani), daver@cs.queensu.ca (D. Rappaport), salomaa@cs.queensu.ca (K. Salomaa).

http://dx.doi.org/10.1016/j.comgeo.2017.05.005
0925-7721/© 2017 Elsevier B.V. All rights reserved.

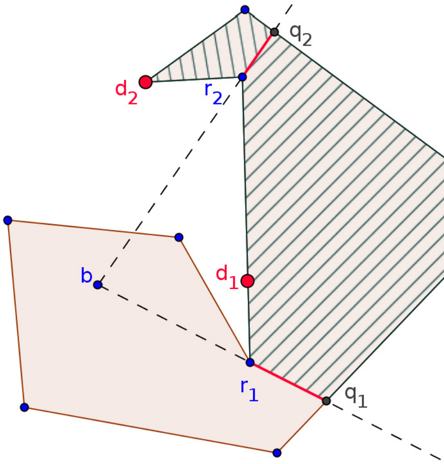

Fig. 1. Attraction and dead regions (shaded regions) of a beacon b. Split edges are $\overline{r_1q_1}$ and $\overline{r_2q_2}$. The line segment $\overline{r_1q_1}$ is a separation edge while $\overline{r_2q_2}$ is not. Points d_1 and d_2 are the dead points.

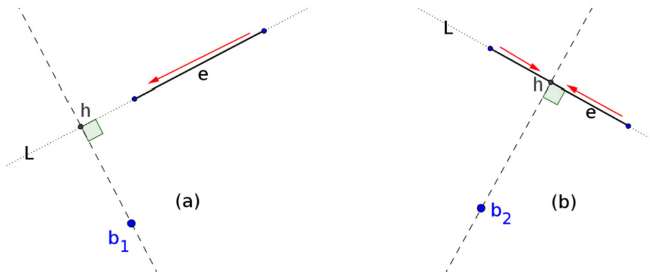

Fig. 2. Sliding on an edge e is towards the orthogonal projection of b on the supporting line of e.

points that can send a message to all of the network. Biro [2] studies beacon and inverse beacon kernels of simple polygons and presents quadratic time algorithms to compute these kernels. For more details on these results see [2–4].

In this paper, we revisit the problem of routing in a polygonal terrain. The shortest beacon watchtower is defined as a beacon on the upper endpoint of a vertical line segment erected on the terrain which can send and receive messages from any other point on the terrain. We present an $O(n \log n)$ algorithm to compute the shortest beacon watchtower. To do so, we present algorithms to compute the beacon kernel and inverse beacon kernel of a monotone polygon in $O(n \log n)$ time.

2. Preliminaries

A *beacon* is a point inside a polygon P that can induce an attraction pull toward itself everywhere in P. When the beacon b is activated, points in P move greedily to minimize their Euclidean distance to b. A point p in the interior of P moves along the ray $\vec{pb}$ until it reaches b or hits the boundary of P. In the latter case it may still reduce its Euclidean distance to b by sliding on the boundary of P. A point in P is *attracted* by b if its Euclidean distance to b is eventually decreased to 0. The *attraction region* of a beacon b is the set of all points in P that b can attract. If a point $d \in P$ (which is not b) remains stationary in the presence of b, then d is called a *dead point*. The set of points in P with fixed resting point d is called the *dead region* of d with respect to b (Fig. 1).

In general, in the attraction pull of a beacon, the path of a point alternates between moving along the ray towards b and sliding along the boundary of P. It eventually either reaches b or gets stuck on a dead point. The boundary between two dead regions or a dead region and the attraction region of b is called a *split edge*. We call a split edge that separates the attraction region of the beacon from a dead region a *separation edge* (Fig. 1). Let s be a split edge for the beacon b. Biro et al. [5] showed that s is a line segment inside P on a line going through b and some reflex vertex r. More precisely, s is a line segment from r along the ray $\overrightarrow{br}$ to the next boundary point of P (see [2] for details). To describe this occurrence we say that r *introduces* s.

Let e be an edge of P and let L be the supporting line of e. Let h be the orthogonal projection of b on L (Fig. 2(a) and (b)). As h is the point with the shortest distance to b among all points on L, sliding on e is always towards h. If h lies on e (Fig. 2(b)), a point sliding on e will reach h and remains on h. Otherwise, it slides all the way to the endpoint of e closer to h. If moving straight towards b from this endpoint is possible (Fig. 3(a)), the point moves straight towards b. Otherwise