Computational Geometry 42 (2009) 873–884

Contents lists available at ScienceDirect

Computational Geometry: Theory and Applications

www.elsevier.com/locate/comgeo

Planar rectilinear shortest path computation using corridors

Rajasekhar Inkulu [a,1], 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 July 2007
Received in revised form 27 February 2009
Accepted 27 February 2009
Available online 10 March 2009
Communicated by T.M. Chan

ABSTRACT

The rectilinear shortest path problem can be stated as follows: given a set of m non-intersecting simple polygonal obstacles in the plane, find a shortest L_1-metric (rectilinear) path from a point s to a point t that avoids all the obstacles. The path can touch an obstacle but does not cross it. This paper presents an algorithm with time complexity $O(n + m(\lg n)^{3/2})$, which is close to the known lower bound of $\Omega(n + m \lg m)$ for finding such a path. Here, n is the number of vertices of all the obstacles together.

1. Introduction

In this paper,[2] we are interested in finding a 2-dimensional rectilinear (L_1) shortest path from a point s to another point t in a polygonal region $\mathcal{P}$ comprising m non-intersecting polygonal obstacles with n vertices in total. This problem has numerous applications, especially in automated circuit design. Note that the path itself is not required to be rectilinear; the metric is rectilinear or L_1. De Rezende, Lee and Wu [7] present an $O(n \lg n)$ time complexity solution to the rectilinear shortest path problem when the obstacles are disjoint isothetic rectangles. Clarkson, Kapoor, and Vaidya [6], and Mitchell [15,16] studied the problem where the obstacles are non-intersecting simple polygons. Two algorithms are presented in [6]: one requires $O(n(\lg n)^2)$ time and $O(n \lg n)$ space, and the other takes $O(n(\lg n)^{3/2})$ time and $O(n(\lg n)^{3/2})$ space. Using a continuous Dijkstra's approach, Mitchell [16] obtained an $O(|R| \lg n)$ time algorithm which uses $O(|R|)$ space, where $|R| = O(n \lg n)$ is the number of certain events. The rectilinear shortest path problem has a rich history which also extends to query problems. Chen, Klenk, and Tu [5] have shown how a polygonal domain can be preprocessed, using $O(n^2(\lg n)^2)$ time and $O(n^2 \lg n)$ space, so that queries in a polygonal domain under the L_1 metric can be answered in time $O((\lg n)^2)$. The special case in which obstacles are disjoint axis-aligned rectangles has been studied by Atallah and Chen [2] and by ElGindy and Mitra [8]; $O(\lg n)$ query time is achievable, using $O(n^2)$ preprocessing time and space, or $O(\sqrt{n})$ query time is achievable, using $O(n^{3/2})$ preprocessing time and space. Mitra and Bhattacharya [17], Chen and Klenk [5], and Arikati et al. [1] have obtained approximation algorithms in the special case of disjoint rectangular obstacles.

Typically, the number of obstacles m is much smaller than the number of vertices of all the obstacles together, n. This has been used to provide efficient algorithms for finding Euclidean shortest paths among obstacles in the plane to yield a $O(n + m^2 \lg n)$ time and $O(n)$ space algorithm by Kapoor, Maheshwari and Mitchell [14]. In this paper, we design an algorithm for computing a rectilinear shortest path in $O(n + m(\lg n)^{3/2})$ time and $O(n + m(\lg m)^{3/2})$ space. Hershberger and Suri [10] gave a $O(n \lg n)$ time and $O(n \lg n)$ space algorithm to find an Euclidean shortest path using the continuous Dijkstra approach.

* Corresponding author.
E-mail addresses: rinkulu@ices.utexas.edu (R. Inkulu), kapoor@iit.edu (S. Kapoor).
[1] Postdoc in the Institute for Computational Engineering and Sciences, The University of Texas at Austin, USA.
[2] The shorter version of this paper appeared in [11].

0925-7721/$ – see front matter
doi:10.1016/j.comgeo.2009.02.005

Since the continuous Dijkstra approach [10,15] is complicated, we use a visibility graph based approach. The visibility graph method is based on constructing a graph whose nodes are the vertices of the obstacles and whose edges are the pairs of mutually visible vertices. Welzl [19] provides an algorithm for constructing the visibility graph of n line segments in $O(n^2)$ time. Ghosh and Mount [9], and Kapoor and Maheshwari [13] found an algorithm to construct the visibility graph in $O(n \lg n + |E|)$ time, where $|E|$ is the number of edges in the graph. Applying Dijkstra's algorithm on this graph, one can determine a shortest path in $O(n \lg n + |E|)$ time. Unfortunately the visibility graph can have $\Omega(n^2)$ edges in the worst case, so any shortest path algorithm that depends on an explicit construction of the visibility graph will have a similar worst-case running time.

We propose an algorithm that builds a restricted visibility graph and then applies Dijkstra's shortest path algorithm on this visibility graph. To construct the restricted visibility graph, our algorithm uses a partition of the polygonal region into corridors as in [12,14]. The construction of corridors relies on triangulating the polygonal region using the algorithm by Bar-Yehuda and Chazelle [3]. Each corridor contributes $O(1)$ vertices to the visibility graph and since there are $O(m)$ corridors, this results in a reduced set of vertices in the visibility graph. However, if we construct the complete visibility graph on this reduced set of vertices the number of edges would be $O(m^2)$. To reduce the number of edges further, we consider a generalized version of the staircase structure proposed in [6], as applied to corridor structures and consequently obtain a reduced vertex set. While Steiner vertices have been utilized in [6], we create a set of extra Steiner vertices and along with a reduced set of edges construct a restricted visibility graph G of even smaller size than the one in [6]. The construction of the visibility graph is similar to that used in [6]. These Steiner vertices are chosen such that for every staircase structure S defined w.r.t. a point p, there exists a rectilinear path from p to any chosen vertex on S. This property ensures that the visibility graph G contains a rectilinear shortest path from s to t.

This paper is organized as follows. Section 2 describes corridor-based staircase structures and the construction of a weighted restricted visibility graph that precisely represents the staircases surrounding each point. Section 3 describes another weighted visibility graph that can be constructed efficiently and allows us to find a rectilinear shortest path. The analysis is contained in Section 4.

2. Corridor-based staircase structures and visibility graph

The rectilinear shortest path problem can be stated as: Given a set $\mathcal{P}$ of non-intersecting simple polygonal obstacles in the plane, find a rectilinear (L_1) shortest path from a point s to a point t which avoids all the obstacles. Here, s and t are considered as degenerate obstacles.

This problem can be solved by using a visibility graph $G = (V, E)$ where V is the set of vertices of the polygonal region and E is the set of visibility edges. Each edge in E is weighted by the rectilinear (L_1) distance between its endpoints. However, as noted above, $|E| = \Theta(n^2)$. In this section, we show how this problem can be solved by partitioning the polygonal region into geometric structures called corridors and defining a restricted visibility graph $G_{\mathrm{vistmp}}(V_{\mathrm{vistmp}}, E_{\mathrm{vistmp}})$. The set of vertices, V_{vistmp}, is a union of two kinds of vertices: V_{ortho} and V_1. The vertices in V_{ortho} are obtained from the corridors used to partition the polygonal region and the vertices in V_1 are obtained by horizontal and vertical projections of vertices in V_{ortho}. By adapting the staircase structure from [6] to apply to the set of corridors, we show that it suffices to restrict attention to these sets of vertices and an associated set of restricted visibility edges.

We adopt the partition of the polygonal region into corridors from [12,14]. For the paper to be self-contained, we provide definitions from the same: Consider a triangulation of the complement of the union of the polygons in $\mathcal{P}$. Note that we consider s and t as degenerate obstacles. Let T denote the resulting triangulation, and let G_{T} denote the dual of the triangulation T formed by adding a vertex for each triangle and an edge between vertices iff the corresponding triangle share an edge. G_{T} is a planar graph with $O(n)$ nodes, $O(n)$ edges, and $m+1$ faces. Consider the recursive removal of all nodes of degree one along with its incident edges until no more degree-1 nodes are left in G_{T}. Now, G_{T} has $m+1$ faces and all nodes are of degree 2 and 3. See Fig. 1. Each node of degree 3 corresponds to a triangle in T, called a *junction* of $\mathcal{P}$. Removal of junction triangles from $\mathcal{P}$ results in a set of simple polygons, which we refer to as the *corridors* of $\mathcal{P}$. The boundary of any such corridor, say C, consists of four components: (1) a polygonal chain along the boundary of an obstacle O_1, from a vertex a to a vertex b; (2) A junction triangle edge (diagonal) from b to c, where c is a vertex located on an obstacle O_2 (possibly $O_2 = O_1$); (3) a polygonal chain along the boundary of O_2, from c to a vertex d; and (4) a diagonal (junction triangle edge) from d back to a. The segments ad and bc are the *bounding edges of corridor C* (known as *doors of C* in [12]). The corridors are classified by their structure into two types, *open and closed corridors*. Consider the corridor C with the bounding edges bc and ad. Suppose that there does not exist any pair of mutually visible points p_1 and p_2 such that p_1 is located on bc and p_2 is located on ad: then the corridor C is termed as a closed corridor. Otherwise, C is called an open corridor. See Fig. 2.

We partition the boundary of each closed corridor into four convex chains and an edge (similar to the approach in [14]). There are two apex points and the convex chains correspond to chains incident to each of the apex points. For each apex point a vertex is introduced in G_{vistmp} and an edge e between the two vertices corresponding to the apex vertices of the same closed corridor. The edge e is assigned a weight equal to the length of the L_1-shortest path, between the two apex points, that lies within the corridor. In open corridors there are two convex chains, one from a to b and the other from c to d. Note that one of these convex chains can degenerate to a point. There are $O(m)$ convex chains in total. The rest of the paper uses only these convex chains.

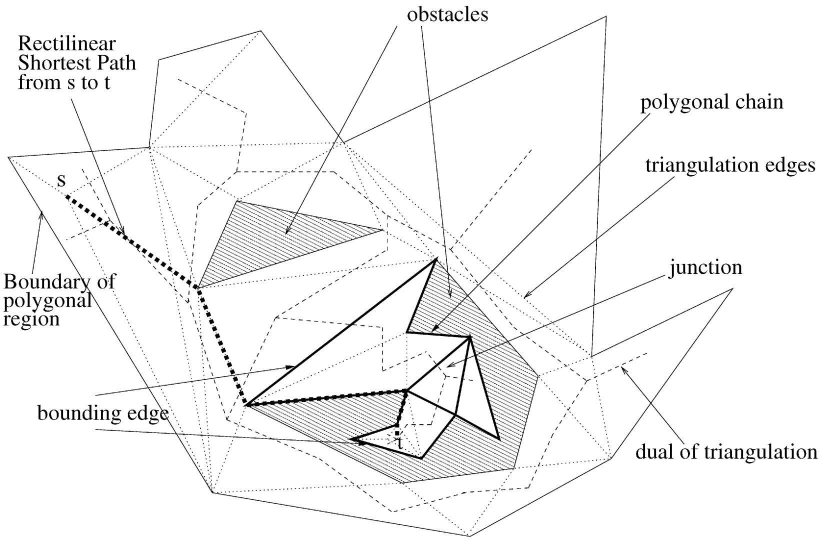


Fig. 1. Corridors in a polygonal domain.

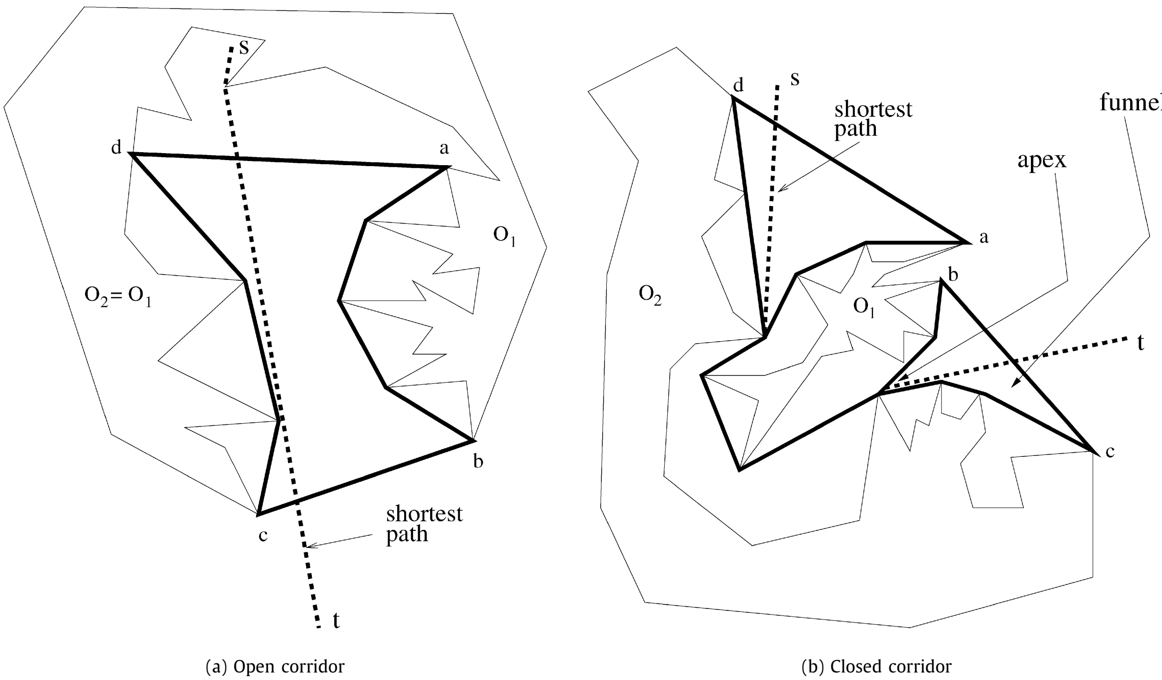


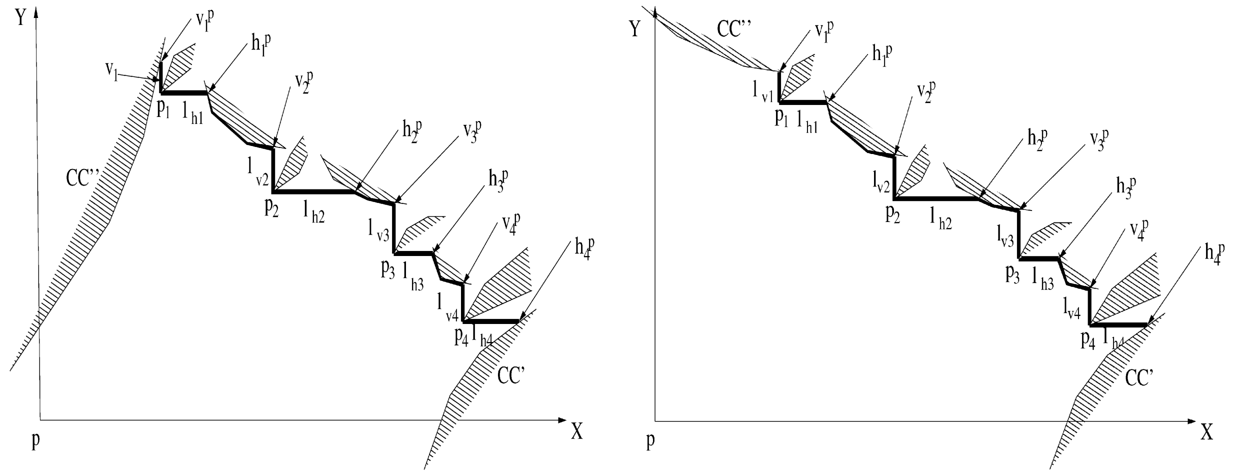
Fig. 2. Types of corridors (courtesy: [12]).

We need the following definitions: Let p and q be points on a convex chain CC. Then the contiguous boundary, along CC, between p and q is known as a *section of* CC. Let p and q be points on a corridor bounding edge e. The line segment joining p and q is known as a *section of* e.

The set of vertices V_{ortho} is defined such that $v \in V_{\text{ortho}}$ iff either of the following is true:

(i) v is an endpoint of a corridor convex chain;
(ii) v is a vertex of some corridor convex chain CC, with the property that there exists either a horizontal or a vertical tangent to CC at v.

Let $\mathcal{O}(p)$ be the orthogonal coordinate system defined with $p \in V_{\text{ortho}}$ as the origin, and horizontal x-axis and vertical y-axis. We next adopt and redefine the staircase structure in [6] to apply to the subset of vertices, V_{ortho} and to the convex chains. For $i \in \{1, 2, 3, 4\}$, we define a set of points $\pi_i(p)$ as: a point $r \in \pi_i(p)$ iff $r \in V_{\text{ortho}}$ and r is located in the ith quadrant of $\mathcal{O}(p)$. Furthermore, we define a set of points $S_i(p)$ as follows: a point q is in the set $S_i(p)$ (see Fig. 3) iff:



(a) Non-negatively sloped edges in both the sections of CC' and CC''

(b) Non-negatively sloped edges in the section of CC' and negatively sloped edges in the section of CC''

Fig. 3. Staircase structure (in bold) with $S_1(p) = \{p_1, p_2, p_3, p_4\}$.

(i) $q \in \pi_i(p)$,
(ii) there is no p' (distinct from p) such that p' is in $\pi_i(p)$ and q is in $\pi_i(p')$, and
(iii) q is visible from p.

We shall assume that the points in $S_i(p)$ are sorted in increasing x-order.

Lemma 2.1. *Sorting the set of points in $S_1(p)$ in increasing x-order results in the same set of points being sorted in decreasing y-order (or, vice versa).*

Proof. Let $p_1, p_2, \ldots, p_l$ be the points in $S_1(p)$ in increasing x-order. The y-coordinate of p_{j+1} cannot be greater than the y-coordinate of p_j (for $j \in \{1, \ldots, l\}$) without violating condition (ii) in the definition of $S_1(p)$. $\square$

Note that similar arguments to Lemma 2.1 can be given for $S_i(p)$ where $i \in \{2, 3, 4\}$.

We term two points $\{p_u, p_v\} \subseteq S_i(p)$ as *adjacent* in $S_i(p)$ if no point $p_l \in S_i(p)$ occurs in between p_u and p_v when the points in $S_i(p)$ are ordered by either the x- or y-coordinates.

Let $p_1, p_2, \ldots, p_k$ be the points in $S_1(p)$ in increasing x-order. Let h_j be the rightward horizontal ray from p_j. And, let v_j be the upward vertical ray from p_j. The first line/line segment that the ray h_j intersects is either a corridor convex chain or v_{j+1}. Note that since a corridor bounding edge does not obstruct visibility, the intersection of h_j or v_j with a corridor bounding edge is not required to be considered further. Let this point of intersection be h_j^p. The first line/line segment that the ray v_j intersects is either a corridor convex chain or h_{j-1}. Let this point of intersection be v_j^p. If the ray does not intersect any other line or line segment then the point h_j^p or v_j^p could be at infinity. Let R_j ($j \in \{1, \ldots, k\}$) denote the unique sequence of sections of corridor convex chains/bounding edges that join h_j^p and v_{j+1}^p; as will be proved shortly R_j is continuous. Note that for the case in which $h_j^p = v_{j+1}^p$, R_j is empty. The elements in the set $\bigcup_{\forall j \in \{1,2,\ldots,k\}}(v_j \cup h_j \cup R_j)$ form a contiguous sequence, termed as the $S_1(p)$-*staircase* (see Fig. 3). Defined, similarly are $S_i(p)$ for $i \in \{2, 3, 4\}$. Note that the convex chains which may possibly intersect the coordinate axes and do not contain a point in $S_i(p)$ are not defined to be part of the staircases in the ith quadrant of $\mathcal{O}(p)$. We next characterize the structure of the staircase in our domain, which includes corridor chains. This is detailed by the following theorem which is similar in nature to the one in [6]. However, the proof requires a more complicated case analysis and differs substantially.

Theorem 2.1. *Along the $S_1(p)$-staircase, any two adjacent points in $S_1(p)$ are joined by at most three geometric entities. These entities ordered by increasing x-coordinates are*: (i) *a horizontal line segment,* (ii) *a section of a convex chain where each edge in that section has a negative slope, and* (iii) *a vertical line segment.*

Proof. Consider two adjacent points in $S_1(p)$, say p_j and p_{j+1}. Let l_h be the line segment $p_j h^p$, where $h^p = h_j^p$, and l_v be the line segment $p_{j+1} v^p$, where $v^p = v_{j+1}^p$. Either h^p is a point on a convex chain belonging to the staircase structure in the first quadrant of $\mathcal{O}(p)$, or h^p is on l_v. Similarly v^p is a point of a convex chain belonging to the staircase structure in the first quadrant of $\mathcal{O}(p)$ or on l_h. Let $\mathcal{R}$ be the region bounded by pp_j, l_h, and sections of convex chains/corridor bounding edges between h^p and v^p along the staircase, l_v, and $p_{j+1}p$. No convex chain can cross either of pp_j, pp_{j+1} (as both p_j and p_{j+1} are visible to p), or l_h, l_v (because h^p is the chosen projection from p_j; similarly v^p from p_{j+1}); also, no convex chain can have an endpoint strictly in the interior of the region $\mathcal{R}$ (because of the adjacency of p_j and p_{j+1} (Lemma 2.1) along the staircase and the definition of $S_1(p)$). In summary, there does not exist a section of a convex chain which intersects the interior of $\mathcal{R}$.

First, we prove that if h^p is not the same point as v^p then the two are incident to the same convex chain. Note that if $h^p = v^p$ then there is nothing further to prove. Suppose h^p is located on a convex chain CC_k, and v^p is located on a different convex chain CC_l for $CC_k \neq CC_l$. Let $CC_k, CC_{k+1}, CC_{k+2}, \ldots, CC_{l-1}, CC_l$ be the consecutive sequence of sections of convex chains or corridor bounding edges encountered while traversing along the staircase from h^p to v^p. Also, let P be the set consisting of points of intersection of any two adjacent entities (where each entity can be a convex chain or a corridor bounding edge) in this sequence including h^p and v^p. Note that every point of P belongs to the vertex set V_{ortho}. Since we have chosen p_j and p_{j+1} as adjacent points in $S_1(p)$, we obtain a contradiction if there exists at least one point in $P \cap S_1(p)$ whenever $|P| > 2$. We show below that there always exists a point in $P \cap S_1(p)$ whenever $|P| > 2$. From this we can conclude that no point joining two geometric entities (where each entity can be a convex chain or corridor bounding edge) can exist in between p_j and p_{j+1} along the staircase. In other words, at most a section of the convex chain or a section of corridor bounding edge joins h^p and v^p. However due to the staircase definition, no line segment of a staircase can have a point in common with a corridor bounding edge. Hence, a corridor bounding edge cannot join h^p and v^p.

Suppose $P \cap S_1(p) = \emptyset$ and $|P| > 2$. Let CC_j be the first convex chain along the staircase while traversing the staircase in increasing x-order, starting at h^p, such that there exists a tangent pp_t to CC_j where the point p_t is located on CC_j, and p_t is visible to p. If no such convex chain and corresponding point p_t exists on the staircase, then the endpoint of the first convex chain along the staircase (while traversing the staircase from h^p) is a point that is visible to p and belongs to $P \cap S_1(p)$. Thus at least one such CC_j always exists. Let q_b and q_e be the first and last points on CC_j (not necessarily distinct from h^p and v^p) as the staircase is traversed from h^p in increasing x-coordinates order. Let p_t be the first such possible point of tangency (satisfying the above mentioned constraints) when traversing CC_j starting from q_e towards q_b. We prove that there exists a point r located on a section of the convex chain CC_j between (and including) q_b and p_t such that $r \in S_1(p)$; hence, this would lead to a contradiction. Let the sequence of edges along CC_j from q_b to p_t be $e_x, e_{x+1}, \ldots, e_y$ and the vertices be $p_x, p_{x+1}, \ldots, p_{y-1}$. Also, let e_{y+1} be the other edge of CC_j that is incident to p_t. The following exhaustive case analysis is based on the slopes of edges e_y, e_{y+1} and if necessary, the orientation of e_x:

- Case 1: Both e_y and e_{y+1} are either above a horizontal line or to the left/right of a vertical line passing through p_t: then there exists a tangent to CC_j at p_t that is horizontal/vertical and $p_t \in V_{\mathrm{ortho}}$ (see Fig. 4(a)).
- Case 2: Alternately, e_y and e_{y+1} satisfy the two following properties:

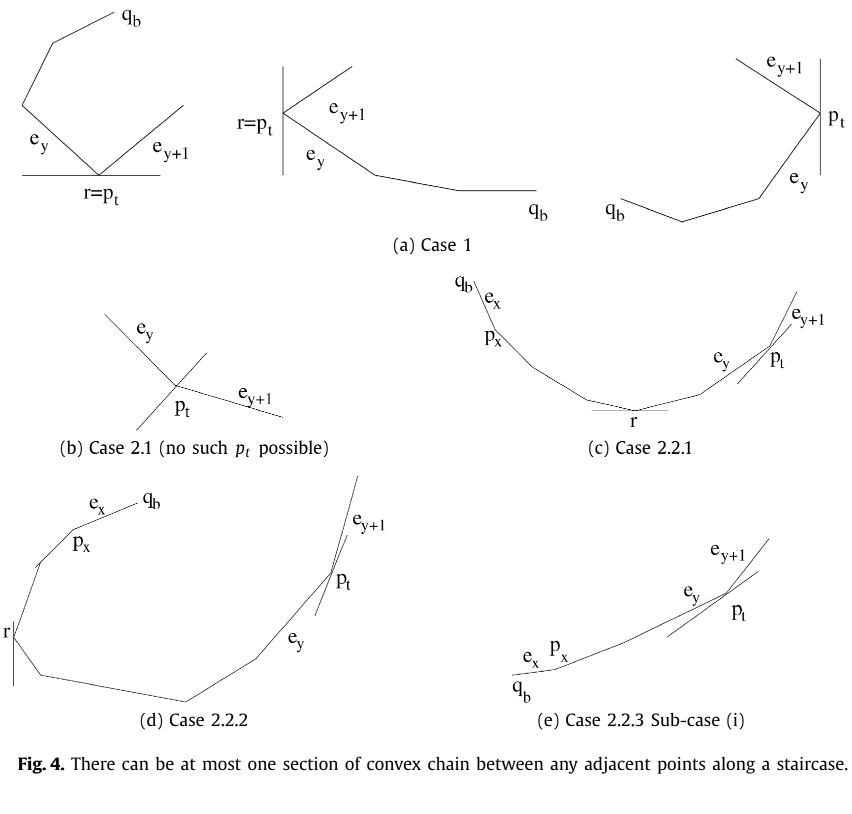


Fig. 4. There can be at most one section of convex chain between any adjacent points along a staircase.

(i) One of the two edges is below and the other above a horizontal line passing through p_t.
(ii) One of the two edges is to the left and the other to the right of a vertical line passing through p_t.

- Case 2.1: Both e_y and e_{y+1} are of negative slope: then pp_t cannot be a tangent to CC_j as it intersects CC_j, contradicting the choice of CC_j (see Fig. 4(b)).
- Case 2.2: Both e_y and e_{y+1} are of non-negative slope
  - Case 2.2.1: e_x is of negative slope with the y-coordinate of p_x being less than q_b: then there must exist a vertex r on CC_j located between q_b and p_t that has a tangent to CC_j parallel to the x-axis; since e_x is of negative slope and CC_j is chosen as the first convex chain along the staircase having a visible point of tangency from p, all points (including r) located on the section of convex chain along CC_j and in between q_b and p_t, are visible to p (see Fig. 4(c)).
  - Case 2.2.2: e_x is of non-negative slope with the y-coordinate of p_x being less than q_b: then there must exist a vertex r on CC_j located between q_b and p_t such that r has a tangent to CC_j parallel to the y-axis. Further, because of the choice of CC_j and since r is the leftmost vertex on CC_j, it must be the case that the r is visible to p and hence r is in $S_1(p)$ (see Fig. 4(d)).
  - Case 2.2.3: e_x is of (i) non-negative or (ii) negative slope with the y-coordinate of p_x being greater than q_b:
    - As CC_j is a convex chain, sub-case (i) is possible iff all the edges between q_b and p_t are of non-negative slope. When this is true, it is trivial to observe that $q_b \in V_{\text{ortho}}$, in turn, $q_b \in S_1(p)$ (see Fig. 4(e)).
    - Since CC_j is a convex chain, sub-case (ii) never arises. Consider the poly-line on the convex chain that connects p_x with p_t. Since the region that includes the obstacle and is bounded by the poly-line must lie outside $\mathcal{R}$, the poly-line shape forces that region to be non-convex, contradicting the convex chain properties.

Now, denote the only possible section of convex chain between p_j and p_{j+1} along the staircase as CC. Next we prove that each edge of this section has a negative slope.

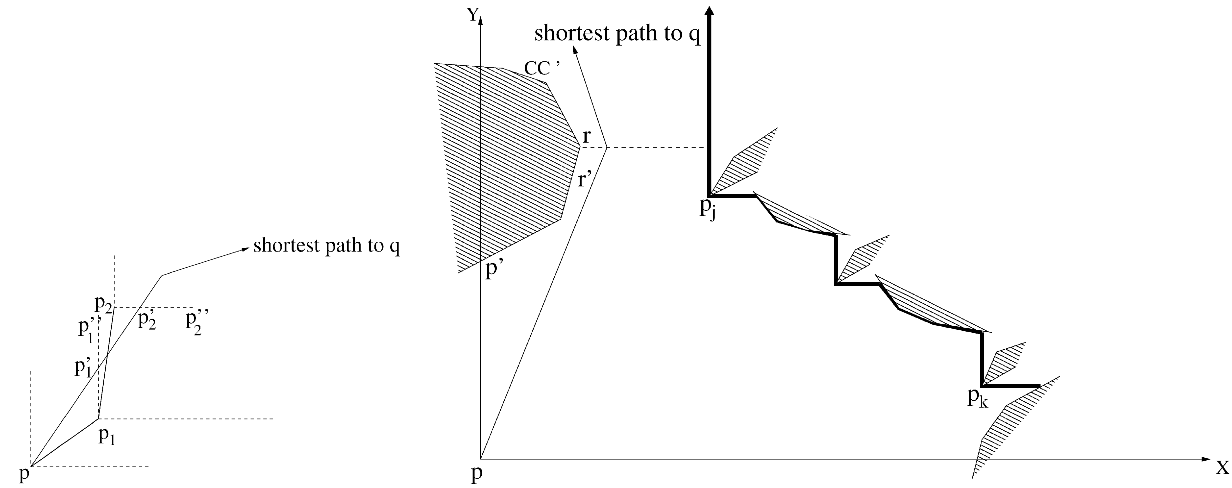
- Consider the case in which the first edge e of CC (while traversing along the staircase from h^p in increasing x-order) has a non-negative slope. Let v_e be the endpoint of edge e while traversing along the staircase from h^p to v^p. We consider the possible sub-cases: either (i) the y-coordinate of v_e is greater than the y-coordinate of h^p, or (ii) the y-coordinate of v_e is less than the y-coordinate of h^p. Since the y-coordinate of h^p is greater than or equal to the y-coordinate of v^p, there exists a vertex, say r, in CC at which a positive and a negatively sloped edge are incident. In sub-case (i), the convex chain has an obstacle polygon below r and hence a convex chain intersects $\mathcal{R}$. In sub-case (ii), if r is not visible from p, then it must be the case that there is another convex chain obstructing the visibility. Thus the open region of $\mathcal{R}$ is intersected by a section of a convex chain. However, this is not possible as proven in the first paragraph of this proof (there does not exist a section of any convex chain that intersects with the interior of $\mathcal{R}$). Hence, r must occur on the staircase between p_j and p_{j+1}, which contradicts the initial assumption of p_j and p_{j+1} being adjacent points along the staircase.
- Consider the case in which the first edge e of CC (while traversing along the staircase from h^p in increasing x-order) has negative slope, and there exist two consecutive edges in CC with negatively sloped one followed by a non-negatively sloped one. Let the intersection of such two edges be p_r, where p_r is chosen as the first such point while traversing the staircase from p_j in increasing x-order. We know that a tangent at p_r to CC is parallel to a coordinate axis, consequently $p_r \in V_{\text{ortho}}$. Since CC is the only section of convex chain possible between h^p and v^p and no convex chain intersects with the region $\mathcal{R}$, the point p_r is visible to p. Therefore, $p_r \in S_1(p)$ (from the definition of $S_1(p)$). But this is a contradiction to the choice of points p_j and p_{j+1} being adjacent in $S_1(p)$. □

Note that similar arguments to Theorem 2.1 can be given for $S_i(p)$ where $i \in \{2, 3, 4\}$. For simplicity we will term the horizontal and vertical line segments on the staircase as *orthogonal line segments*.

We now define the weighted restricted visibility graph $G_{\text{vistmp}}(V_{\text{vistmp}} = V_{\text{ortho}} \cup V_1, E_{\text{vistmp}} = E_{\text{occ}} \cup E_1 \cup E_{\text{tmp}})$:

- For each $v \in V_{\text{ortho}}$, consider two horizontal rays h_L and h_R which start at v and proceed in the leftward and rightward direction, respectively. Let v_L (resp. v_R) be the projection of v onto the first corridor convex chain encountered on traversing along h_L (resp. h_R). If no such corridor chain is encountered then the projection occurs at infinity. Similarly, let the vertical projections of v in increasing and decreasing direction of y-coordinates be v_U and v_D, respectively. For each point $p \in \{v_L, v_R, v_D, v_U\}$, if the distance of p from v is finite then p is added to V_1 and the edge pv is added to E_1. The weight of the edge $e \in E_1$ is the rectilinear distance between its two endpoints.
- An edge $e = (p, q)$ belongs to E_{occ} iff the following conditions hold (i) $\{p, q\} \subseteq V_{\text{vistmp}}$, (ii) both p and q belong to the same corridor convex chain, and (iii) no point in V_{vistmp} lies between p and q along the chain. The weight of edge e is the L_1 distance along the section of convex chain between p and q.
- An edge $e' = (p', q')$ with $p' \in V_{\text{ortho}}$ belongs to E_{tmp} iff $q' \in S_i(p')$. The weight of e' is the rectilinear distance along e'.

Theorem 2.2. *Let $\{p, q\} \subseteq V_{\text{vistmp}}$. Then a shortest path from p to q in G_{vistmp} defines a shortest L_1 path from p to q that does not intersect any of the obstacles.*



(a) Case (A)

(b) Case (B)

Fig. 5. Replacing a shortest path from p to q with edges in G_{vistmp}.

Proof. Consider a shortest path P from p to q that avoids all the obstacles. We need to consider two cases:

Case (A) – The shortest path P crosses a staircase structure defined w.r.t. point p. Since convex chains have an obstacle on one side, the shortest path P does not intersect any of the convex chains in the staircase. Therefore, the shortest path P intersects the staircase at either a point in $S_i(p)$, or an orthogonal line segment in the staircase. Let p_j and p_k be the points in $S_i(p)$ with the minimum and maximum x-coordinates. For a point $p_1 \in S_i(p)$, suppose the path P crosses an orthogonal segment $p_1 p_1''$ of the staircase at p_1' (see Fig. 5(a)). Consider replacing the path from p to p_1' with two line segments, one joining p to p_1, and the other from p_1 to p_1'. Note that the L_1 distance along the line joining p to p_1' is the same as the L_1 distance along the altered path. This new rectilinear path is always guaranteed to exist because:

(i) no point of V_{ortho} exists in the region bounded by the staircase and the line segments pp_j, pp_k;
(ii) neither of the convex chains intersecting the coordinate axes intersects with the interior of the altered path.

The path from p_1 to q can be altered similarly without changing the length of the path. Since the distance from p_1 to q is shorter than the distance from p to q, the process terminates. Since a shortest path from p to q does not repeat any vertex, the alteration procedure will terminate. Note that the altered path is in G_{vistmp} because for p and every $p_l \in S_i(p)$, the edge $pp_l \in E_{\mathrm{vistmp}}$. Therefore, the rectilinear shortest path P between p and q in the given polygonal region can be found by determining the shortest path from p to q in the graph G_{vistmp}.

Case (B) – The shortest path P does not cross any of the staircase structures defined w.r.t. point p. Consider the case when the first line segment pq' along the shortest path P is in the first quadrant of $\mathcal{O}(p)$ (other cases can be argued symmetrically). Let p_j and p_k be the points in $S_1(p)$ with the minimum and maximum x-coordinates (see Fig. 5(b)). Since the shortest path P does not cross the $S_1(p)$ staircase structure, it must be the case that the x-coordinate of q' is either less than the x-coordinate of p_j or greater than the x-coordinate of p_k. Consider the former case (the other case is symmetric). If no convex chain intersects the y-axis in the first quadrant, then either $q' \in S_1(p)$ or the interior of pq' does not intersect with the first quadrant of $\mathcal{O}(p)$ – leading to a contradiction.

Alternately, let CCY be the first convex chain that intersects the upward vertical ray from p. Also, let CC' be the section of CCY in the first quadrant of $\mathcal{O}(p)$ and let p' be the vertical projection of p onto CC'. Consider the slopes of edges in CC' starting at p'. Either CC' starts with a negatively sloped edge or with a positive sloped edge. Note that CC' cannot intersect the line segment pp_j. Let r be either the endpoint of CC' or the point at which the slope of CC' changes sign. By definition, $r \in V_{\mathrm{ortho}}$ and the x-coordinate of r is less than the x-coordinate of p_j. Consider the case in which CC' starts with a negatively sloped edge at p'. Then r is visible and thus in $S_1(p)$, contradicting the choice of p_j. Next consider the case when CC' starts with positive slope and the vertex r, not being in $S_1(p)$, is not visible from p. Consider the set of points in V_{ortho} in the first quadrant, with x-coordinate less than p_j. None of these points are visible from p. Consider the point with the least y-coordinate from amongst such points. We denote this point as r''. Note that r'' may be r itself. Let r' be the intersection of the horizontal projection of r'' onto CC'. (See Fig. 5(b)). Now replace the path from p to q with an equivalent cost path consisting of (i) a vertical line segment pp', (ii) a path from p' to r' along CC', and (iii) a path from r' to q'. The L_1 distance along P is same as the L_1 distance along the altered path since the slopes of edges on CC' along the path from p' to r' cannot be negative. The new rectilinear path is always guaranteed to exist because $pp' \in E_1$ and the edges comprising the path from p' to r' along CC' are in E_{occ}. The path P from r' onwards can be similarly modified. $\square$

Let q be the number of points in $S_3(p)$, and r be the number of points in $S_1(p)$. Consider the path from a point $p_k \in S_1(p)$ to $p_l \in S_3(p)$. This path can be altered to one which traverses p, i.e. along the line segments p_k to p, and p to p_l. Note that the altered path does not change the L_1 distance from p_k to p_l. By having an edge joining every point in $S_3(p) \cup S_1(p)$ with p, the number of visibility edges around p are reduced from $O(qr)$ to $O(q+r)$. Similar savings can be achieved by eliminating the possible edges between $S_4(p)$ and $S_2(p)$.

However, it is required to construct staircase structures surrounding each point $p \in V_{\text{ortho}}$ in sub-quadratic time. To achieve this, we use the technique of [6] to introduce Type-II Steiner points and devise an approach in the next section.

3. Visibility graph with Steiner points

In this section, we detail the construction procedure of a modified restricted weighted visibility graph $G_{\text{vis}}(V_{\text{vis}} = V_{\text{ortho}} \cup V_1 \cup V_2, E_{\text{vis}} = E_{\text{occ}} \cup E_1 \cup E_2)$. This differs from G_{vistmp} in the addition of the Steiner vertices V_2 and the Steiner edges E_2. Also, the edges in E_{tmp} are removed from G_{vistmp} in order to obtain G_{vis}. The vertices $V_1 \cup V_2$ and the edges $E_1 \cup E_2$ are defined so that for any edge $e = (v_p, v_q) \in E_{\text{tmp}}$ of G_{vistmp}, there is a path of the same L_1 length between v_p and v_q in $G_{\text{vis}}(V_{\text{vis}}, E_{\text{vis}})$. Following [6], the vertices and edges of G_{vis} are divided into two types, Type-I $(V_{\text{ortho}} \cup V_1, E_{\text{occ}} \cup E_1)$ and Type-II (V_2, E_2), whose construction is described below.

3.1. Type-I points and edges

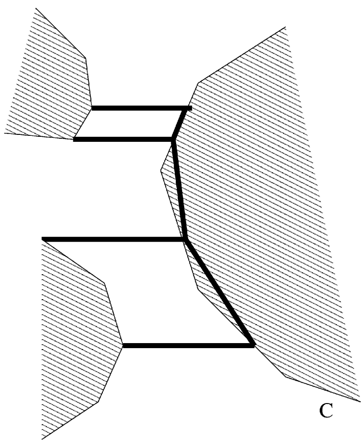
The Type-I points and edges are defined in Section 2 (see Fig. 6(a)). These points are obtained by sweeping the obstacle space by orthogonal sweep lines using a standard sweep-line procedure. Since there are four orthogonal projections possible for a point, the algorithm utilizes a sweep-line algorithm wherein the method sweeps a vertical sweep line from left to right and from right to left, and a horizontal sweep line from top to bottom and from bottom to top.

As usual, we assume that the obstacle vertices and edges are provided in order along each polygon and thus the vertices of any convex chain are available in order along that chain. Any convex chain can be decomposed into monotone parts. For simplicity, we assume that the convex chains are y-monotone i.e., all the vertices are sorted in increasing y-order.

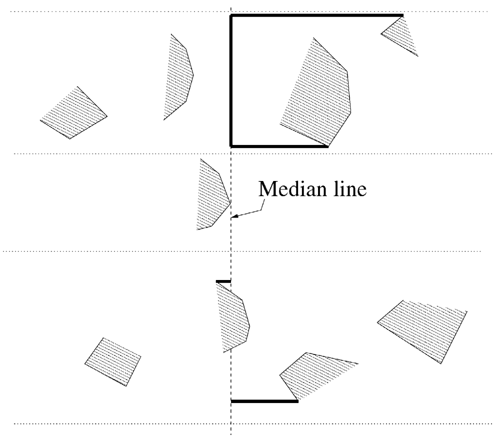
Consider sweeping a vertical line from left to right to generate the horizontal projections from vertices in V_{ortho}. The status of the sweep line is maintained as a sorted set of points, each of which has a possible projection onto a convex chain on the right. As usual, this set is stored in a dynamic balanced binary tree structure. We summarize the events that occur during the sweep-line algorithm: When a point $p \in V_{\text{ortho}}$, located on the convex chain CC', is encountered by the sweep line, we insert p to the sweep-line data structure iff the horizontal ray originating from p and directed towards the right does not intersect with the obstacle bounded by CC'.

Let p be the first point encountered on the chain CC'. Also, let r' be a point in the sweep-line data structure at the time p is encountered by the sweep line. If r' projects onto CC' at r'', then the y-coordinate of r'' lies within the range of y-coordinates of points on CC'. If the condition is met, then the edge corresponding to the line segment $r'r''$ is added to E_1 and r' is deleted from the sweep-line data structure. Further details of the sweep-line algorithm are similar to the well-known solution for finding the intersection points between a set of line segments and a set of piecewise linear curves using the plane sweep-line algorithm [4,18].

Similarly, we define the other three sweeps. At the end of these four sweeps, an ordered list of the Type-I points along a convex chain are obtained by merging the ordered lists. This list readily gives the complete set of vertices V_{ortho} and E_{occ}.



(a) Type-I points and edges due to projections onto corridor C



(b) Type-II points and edges due to projections onto a median line from two points in each of the two distinct strips

Fig. 6. Type-I and Type-II points and edges.

procedure *TypeIISteinerPoints*(V'')

1: Whenever $|V''|$ is greater than $\sqrt{\lg m}$, divide the points of V'' into $O(|V''|/\sqrt{\lg m})$ strips parallel to the x-axis with each strip having $O(\sqrt{\lg m})$ points. Otherwise, a single strip comprises of all the input points, V''.
2: Let x_m be the median of the x-coordinates of points in V''; let M be the vertical line through x_m; let $\mathcal{R}$ be the collection of sets such that each set in $\mathcal{R}$ consists of all the points from a distinct strip and $\bigcup_{R\in\mathcal{R}} R = V''$.
3: **for** each set $R \in \mathcal{R}$ **do**
4: Let p_t (resp. p_b) be the point in R with the largest (resp. smallest) y-coordinate among all the points in R such that the point p'_t (resp. p'_b), obtained by projecting p_t (resp. p_b) parallel to x-axis onto M, is visible from p_t (resp. p_b). Then the point p'_t (resp. p'_b) is added to V_2; note that if no such point p_t (resp. p_b) exists, then no such p'_t (resp. p'_b) is introduced.
5: $R' \leftarrow R \cup \{p'_t, p'_b\}$
6: **end for**
7: For every pair of points $p, q \in R'$, we include an edge in E_2 iff the rectangle formed with p and q at the diagonal endpoints does not contain a point in V_{ortho}, and p is visible from q.
8: Whenever $|V''| > 1$, let V'_{tmp} (resp. V''_{tmp}) be the set of points in V'' with x-coordinates less (resp. greater) than x_m and invoke *TypeIISteinerPoints* twice: once with V'_{tmp} as the argument, and next with V''_{tmp} as the argument.

3.2. Type-II points and edges

The main procedure to obtain the Type-II Steiner points and Steiner edges invokes the *TypeIISteinerPoints* procedure with $V_{\text{ortho}} \cup V_1$ as the argument (see Fig. 6(b)).

To facilitate subdividing points into strips in procedure *TypeIISteinerPoints*, we maintain two lists corresponding to the sorted sequences of points in V_{vis} along the x and y-coordinates. These are available at the root of the recursion tree. The lists required at a node in the recursion tree are obtained from the lists available at its parent. The Type-II points/edges corresponding to a strip at a recursive step are obtained using these lists.

The Type-II points are found as follows: first, find the horizontal and vertical projections of each point onto the nearest convex chain. These projections have been already discovered via the sweep-line technique when determining the set V_1. The remaining procedure is similar to the one in [6]. At each recursion step, given the sorted y-coordinates of the input points, V'', it is easy to partition V'' into strips of size $O(\sqrt{\lg m})$. For a given point p, let p_l, p_r, p_u and p_d be the projections of p onto the closest chain to the left, right, up and down, respectively (these points are at infinity if no such closest point exists). For each strip and a median vertical line, the points p'_t and p'_b can be determined from all projection line segments pp_l and pp_r that intersect the median line by processing the points in sorted y-order.

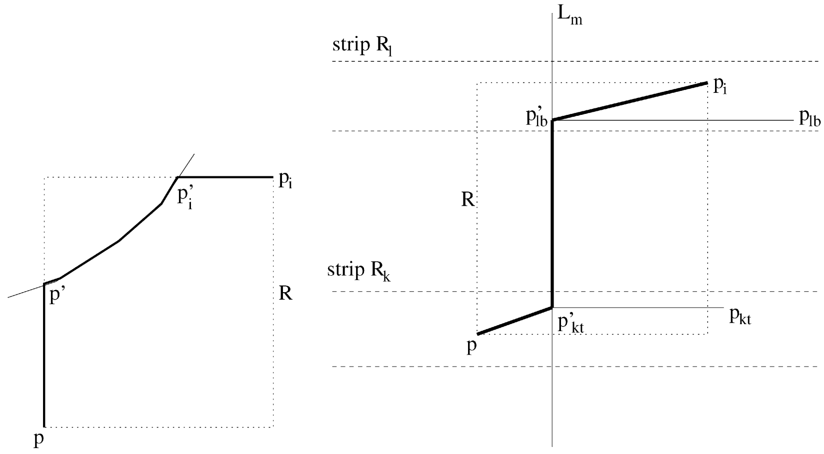
Further, as specified in line 7 of procedure *TypeIISteinerPoints*, for every pair of points p and q in a strip R', we need to determine the following conditions for adding an edge (p, q) to E_2: (i) p is visible from q and (ii) the rectangle R_{pq} determined by p and q does not contain a point in V_{ortho}. As in [6], given a point p, we can determine all points q that satisfy condition (ii) by a line sweep from left to right. W.l.o.g. assume that the x and y-coordinates of p are less than that of q (the other cases are similar). The visibility between p and q can be determined from the points p_r, p_u, q_l, q_d. These can be used to determine whether there is a convex chain that intersects the rectangle R_{pq} defined by p and q. Note that no convex chain can end inside R_{pq}, for otherwise R_{pq} will contain a point in V_{ortho}. Instead of using a different sweep line at every step of the recursion, an alternative procedure using a common sweep-line algorithm can be used to determine the elements of the set E_2 as described below.

Let $\mathcal{S}$ be the collection of sets such that each set in $\mathcal{S}$ consists of all the points with the same x-coordinate from the set of Steiner points V_2 and $\bigcup_{S_x\in\mathcal{S}} S_x = V_2$. Let each such S_x be in sorted y-order. For every two adjacent vertices s_i, s_j in each S_x that are visible to each other, we include an edge joining s_i and s_j in E_2. (Note that both V_2 and E_2 are defined in the procedure *TypeIISteinerPoints*.) The projection of points in V_2 vertically onto obstacles can be computed as mentioned above. Consider two adjacent points in any S_x, say s_i and s_j, when S_x is in y-order. W.l.o.g. let the y-coordinate of s_j be greater than that of s_i. An edge is added between s_i and s_j iff the projection from s_i vertically above has y-coordinate greater than or equal to that of s_j.

Theorem 3.1. *Let p and q be points in V_{vis}. Then a shortest path from p to q in G_{vis} ($V_{\text{vis}}, E_{\text{vis}}$) defines a shortest L_1 path from p to q that avoids all the obstacles.*

Proof. To prove this, we show that if there is an edge of length l between two points in G_{vistmp}, then there exists a path of length l in the graph $G_{\text{vis}}(V_{\text{vis}}, E_{\text{vis}})$ between the same two points. W.l.o.g. we consider the edges contained in the first quadrant of $\mathcal{O}(p)$. For a point $p \in V$, we know that an edge $pp_i \in E$ $\forall p_i \in S_1(p)$. Suppose $p_i \in S_1(p)$ and let the L_1 length of edge pp_i be l. Let R be the rectangle obtained by having p and p_i as diagonal endpoints. We need to consider the following two cases:

Case (A) – The interior of R intersects with some corridor convex chain CC (see Fig. 7(a)). Because no vertex of the chain CC that is inside R can be in V_{ortho} and the diagonal of R cannot be intersected by CC, it must be the case that the CC enters and exits R via a vertical and an adjacent horizontal side, respectively, of R. W.l.o.g. assume that CC intersects the vertical side of R incident to p and the horizontal side of R incident to p_i. Let CC be the first such chain encountered as we move up the vertical side of R starting from p. Let p' and p'_i be the points of intersection of CC with the vertical and horizontal sides of R that are incident at p and p_i, respectively. No other convex chain intersects the line segments



(a) Case (A) (b) Case (B)

Fig. 7. Proof of Theorem 3.1.

pp' and $p_i p'_i$. To show this, consider the case in which a convex chain intersects one of these line segments. Since this chain must enter and leave R via the same side of R or have an endpoint in R, there is a vertex of the chain p', distinct from p, such that $p' \in V_{\text{ortho}}$, and p' satisfies the condition: $(p' \in \pi_1(p)) \wedge (p_i \in \pi_1(p'))$. However, then the point p_i does not belong to $S_1(p)$ (due to the condition (ii) in the definition of $S_1(p)$), a contradiction. The above is true for both pairs of adjacent sides on either side of the diagonal pp_i. In other words, there are projections from points p and p_i that are always incident to the same convex chain CC. The Type-I points due to the orthogonal projections of p and p_i onto CC are p' and p'_i respectively. Let CC' be the section of CC from p' and p'_i. Note that no vertex of CC' belongs to V_{ortho}. By the above discussion, CC' has only non-negatively sloped edges as CC' is traversed from p' to p'_i. Consider the path comprising the edge $pp' \in E_1$, path from p' to p'_i comprising edges from E_{occ}, and the edge $p'_i p_i \in E_1$. The L_1 distance along this path is l.

Case (B) – The interior of R does not intersect with any corridor convex chain (see Fig. 7(b)). Let p and p_i reside in distinct strips R_k and R_l respectively, wherein both the strips R_k and R_l belong to the same stage of the recursion. Assume that the strip R_k is located below R_l (the other case can be argued symmetrically). Then at some node of the recursion tree, there must exist a median line, say M, located in between p and p_i (including p and p_i). Choose p_{kt} and p_{lb} to be the points with the highest and lowest y-coordinate points in strips R_k and R_l, respectively, such that the corresponding horizontal projection onto M, p'_{kt} and p'_{lt}, have the property that p_{kt} is visible from $p_{\text{kt}'}$ and $p_{\text{lb}'}$ is visible from p_{lb} (considering the case when either p_{kt} or p_{lb} resides on M itself as a degenerate case). Since $p_i \in S_1(p)$ and the rectangle R does not intersect with any convex chain, the interior of rectangle R does not contain any obstacles. Hence for p distinct from p_{kt}, as $p_{\text{kt}'}$ is located interior to R, the algorithm adds a Type-II Steiner edge joining p and $p_{\text{kt}'}$ (line 7 of procedure *TypeIISteinerPoints*). Similarly, for p_i distinct from p_{lb}, as $p_{\text{lb}'}$ is located interior to R, there exists a Type-II Steiner edge joining p_i and $p_{\text{lb}'}$. Suppose that there is no such p_{kt} which is distinct from p. Since no obstacle intersects the interior of rectangle R, the horizontal projection p' of p onto M is visible from p. Hence p' is the same as $p_{\text{kt}'}$. A symmetric argument can be given for the case in which there is no p_{lb} distinct from p_i. Therefore, Type-II edges $pp_{\text{kt}'}$ and $p_i p_{\text{lb}'}$ always exist. Also, no convex chain can intersect M in between $p_{\text{kt}'}$ and $p_{\text{lb}'}$ as there is no obstacle strictly inside the rectangle R. Since p_{kt} and p_{lb} are chosen such that they are the top most and bottoms most points in strips R_k and R_l respectively, the L_1 distance of the path consisting of Type-II edges $pp_{\text{kt}'}, p_{\text{kt}'}p_{\text{lb}'}, p_{\text{lb}'}p_i$ is of length l.

In the special case when both p and p_i reside in the same strip, the algorithm adds the edge pp_i (line 7 of procedure *TypeIISteinerPoints*). The edge pp_i is guaranteed to exist and has the L_1 length l.

This proves that every edge in G_{vistmp} is represented as a path with the same L_1 length in G_{vis}. This, together with Theorem 2.2, leads to the proof. □

4. Analysis

Lemma 4.1. *The set V_{ortho} is of size $O(m)$.*

Proof. Consider the interior of the region $\mathcal{R}$ bounded by a convex chain CC and the line segment joining the endpoints of CC. As each such region $\mathcal{R}$ is convex (from the definition of a corridor convex chain in [14]), there can be at most a constant number of orthogonal tangents incident to a vertex of each corridor convex chain. Hence each corridor convex chain can contribute at most a constant number of points to V_{ortho}. Since there are $O(m)$ corridors, the claim follows. □

Lemma 4.2. *Computing all the possible horizontal/vertical tangents to all the convex chains together takes $O(m \lg n)$ time.*

Proof. Given that the edges of each convex chain are ordered in a linked list, we build a binary search tree for each convex chain in time linear in the number of edges. This facilitates in finding all the points of tangency on a convex chain such that the tangent at such a point is either horizontal or vertical in $O(\lg n)$ time. Using Lemma 4.1, the claim follows. □

Lemma 4.3. *There are $O(m)$ Type-I points and $O(m)$ Type-I edges.*

Proof. There are $O(m)$ points in V_{ortho} (by Lemma 4.1). Since each of them would be projected to at most a constant number of obstacles, there can be at most $O(m)$ points in V_1. Each Type-I point causes at most a constant number of edges in E_1, hence $|E_1|$ is $O(m)$. Since there are $O(m)$ Type-I points and at most two edges of E_{occ} incident to a Type-I point, the size of E_{occ} is $O(m)$. □

Lemma 4.4. *Computing the Type-I points and edges takes $O(n + m\lg n)$ time.*

Proof. As there are $O(m)$ Type-I points (from Lemma 4.3), there can be at most $O(m)$ points that would be inserted into the sweep-line data structure (binary tree) during the sweep-line algorithm. Also, a point resides in this tree as long as it is not projected to a convex chain; once it is projected, it would be deleted from the tree and it never reenters the sweep-line data structure during that particular sweep. An insertion/deletion to/from this data structure takes $O(\lg m)$ time. The algorithm considers a constant number of sweep lines. Hence in adding (resp., deleting) each point in V_{ortho} to (resp., from) the sweep-line data structure, we do at most one binary search over the tree structure representing the sweep-line status. Since there are $O(m)$ points in V_{ortho}, the overall cost is of $O(m\lg m)$. Corresponding to each convex chain we create a binary search tree (as described in Lemma 4.2), finding the intersection of an orthogonal projection line from a point to a convex chain takes $O(\lg n)$ time. Since a point can be orthogonally projected to at most a constant number of convex chains, the computation involved in finding all the projection points takes $O(m\lg n)$. Once all the Type-I points along each convex chain are readily available, finding and computing the weights of the edges in E_{occ} can be done by traversing the edges of the convex chains in $O(n)$ time. □

Lemma 4.5. *The number of Type-II points and edges are $O(m(\lg m)^{1/2})$ and $O(m(\lg m)^{3/2})$ respectively.*

Proof. From Lemma 4.3, the size of the set $V_{\text{ortho}} \cup V_1$ is $O(m)$. At a node v of depth d in the recursion tree, there are $O(m/((2^d)(\lg m)^{1/2}))$ strips, where each strip comprises $O((\lg m)^{1/2})$ points. Hence each level of this tree has $O(m/(\lg m)^{1/2})$ strips. Since the depth of the tree is $O(\lg m)$, and each strip contributes at most $O(1)$ Type-II points at a given level in the recursion tree, the total number of Type-II points are $O(m(\lg m)^{1/2})$. Since each strip has $O((\lg m)^{1/2})$ points, processing a strip can create at most $O(\lg m)$ Type-II edges. This yields the number of Type-II edges as $O(m(\lg m)^{3/2})$. □

Lemma 4.6. *Computing the Type-II points and edges takes $O(m(\lg m)^{3/2})$ time.*

Proof. The $O(m)$ points in $V_{\text{ortho}} \cup V_1$ are initially sorted in $O(m\log m)$ steps so that we can determine the sorted list of points required at any node of the recursion tree in linear time. To determine if an edge (p,q) belongs to E_2, a linear time test can be done at each step of recursion, requiring $O(m(\lg m)^{1/2})$ per level of recursion. Alternately, with the common sweep-line approach, since there are $O(m(\lg m)^{1/2})$ Type-II points, the sweep-line technique takes $O(m(\lg m)^{3/2})$ time in determining all the Type-II points and edges together (similar to Lemma 4.4). Since the total number of strips are $O(m(\lg m)^{1/2})$ (Lemma 4.5), maintaining y-coordinates of all strips together takes $O(m(\lg m)^{3/2})$. Finally, computing edges between adjacent points on the median lines can be done using the sweep-line approach in $O(m(\lg m)^{3/2})$ steps, since the number of points in V_2 is bounded by $O(m(\lg m)^{1/2})$. □

Theorem 4.1. *Computing a rectilinear shortest path from s to t takes $O(n+m(\lg n)^{3/2})$ time and $O(n+m(\lg m)^{3/2})$ space complexity, where n is the number of vertices of all the obstacles together and m is the set of non-intersecting simple polygonal obstacles.*

Proof. From Lemmas 4.3 and 4.5, there are $O(m(\lg m)^{1/2})$ Type-I and Type-II points together. Hence, $|V_{\text{vis}}| = O(m(\lg m)^{1/2})$. There are $O(m(\lg m)^{3/2})$ Type-I and Type-II edges together. Hence, $|E_{\text{vis}}| = O(m(\lg m)^{3/2})$. Applying Dijkstra's algorithm takes $O(|E_{\text{vis}}| + |V_{\text{vis}}|\lg|V_{\text{vis}}|)$ i.e., $O(m(\lg m)^{3/2})$. Using the algorithm by Bar-Yehuda and Chazelle [3] the triangulation of the polygonal region takes $O(n+m(\lg m)^{1+\epsilon})$, where ϵ is a positive constant less than one. Finding corridors and junctions, given the triangulation, takes $O(n + m\lg n)$. The time involved in precomputing the rectilinear shortest distance between the apex points of all the closed corridors together takes $O(n)$ time. From Lemmas 4.4 and 4.6, computing the visibility graph G_{vis} takes $O(n + m(\lg n)^{3/2})$ time. Including the time to find the vertices of V_{ortho} (Lemma 4.2), the overall time complexity is $O(n+m(\lg n)^{3/2})$. Only binary trees and lists are used in the algorithm. And, no data structure uses more space than the total number of Type-I/Type-II points/edges, hence the space complexity (including the input complexity). □

References

[1] S.R. Arikati, D.Z. Chen, L.P. Chew, G. Das, M.H.M. Smid, C.D. Zaroliagis, Planar spanners and approximate shortest path queries among obstacles in the plane, in: Lecture Notes in Computer Science, vol. 1136, 1996, pp. 514–528.
[2] M.J. Atallah, D.Z. Chen, On parallel rectilinear obstacle-avoiding paths, Computational Geometry Theory and Applications 3 (6) (1993) 307–313.
[3] R. Bar-Yehuda, B. Chazelle, Triangulating disjoint Jordan chains, International Journal of Computational Geometry and Applications 4 (4) (1994) 475–481.
[4] J.L. Bentley, T. Ottmann, Algorithms for reporting and counting geometric intersections, IEEE Transactions on Computing 28 (9) (1979) 643–647.
[5] D.Z. Chen, K.S. Klenk, Rectilinear short path queries among rectangular obstacles, Information Processing Letters 57 (6) (1996) 313–319.
[6] K.L. Clarkson, S. Kapoor, P.M. Vaidya, Rectilinear shortest paths through polygonal obstacles in $O(n(\lg n)^2)$ time, in: Proceedings of the ACM Symposium on Computational Geometry, 1987, pp. 251–257.
[7] P.J. de Rezende, D.T. Lee, Y.-F. Wu, Rectilinear shortest paths in the presence of rectangular barriers, Discrete and Computational Geometry 4 (1989) 41–53.
[8] H. ElGindy, P. Mitra, Orthogonal shortest route queries among axis parallel rectangular obstacles, International Journal of Computational Geometry and Applications 4 (1994) 3–24.
[9] S.K. Ghosh, D.M. Mount, An output-sensitive algorithm for computing visibility graphs, SIAM Journal on Computing 20 (5) (1991) 888–910.
[10] J. Hershberger, S. Suri, An optimal algorithm for Euclidean shortest paths in the plane, SIAM Journal on Computing 28 (6) (1999) 2215–2256.
[11] R. Inkulu, S. Kapoor, Finding a rectilinear shortest path in R^2 using corridor based staircase structures, in: Proceedings of the Foundations of Software Technology and Theoretical Computer Science, 2007, pp. 412–423.
[12] S. Kapoor, S.N. Maheshwari, Efficient algorithms for Euclidean shortest path and visibility problems with polygonal obstacles, in: Proceedings of the ACM Symposium on Computational Geometry, 1988, pp. 172–182.
[13] S. Kapoor, S.N. Maheshwari, Efficiently constructing the visibility graph of a simple polygon with obstacles, SIAM Journal on Computing 30 (3) (2000) 847–871.
[14] S. Kapoor, S.N. Maheshwari, J.S.B. Mitchell, An efficient algorithm for Euclidean shortest paths among polygonal obstacles in the plane, Discrete and Computational Geometry 18 (4) (1997) 377–383.
[15] J.S.B. Mitchell, An optimal algorithm for shortest rectilinear paths among obstacles, in: Abstracts 1st Canadian Conference on Computational Geometry, 1989.
[16] J.S.B. Mitchell, L_1 shortest paths among polygonal obstacles in the plane, Algorithmica 8 (1) (1992) 55–88.
[17] P. Mitra, B.K. Bhattacharya, Efficient approximate shortest-path queries among isothetic rectangular obstacles, in: Proceedings of the Third Workshop on Algorithms and Data Structures, Springer-Verlag, 1993, pp. 518–529.
[18] J. Pach, M. Sharir, On vertical visibility in arrangements of segments and the queue size in the Bentley–Ottmann line sweeping algorithm, SIAM Journal on Computing 20 (3) (1991) 460–470.
[19] E. Welzl, Constructing the visibility graph for n-line segments in $O(n^2)$ time, Information Processing Letters 20 (4) (1985) 167–171.