Rectilinear shortest paths through polygonal obstacles in $O(n\,(\log\,n)^{3/2})$ time

Kenneth L. Clarkson, Sanjiv Kapoor, and Pravin M. Vaidya

AT&T Bell Laboratories
Murray Hill, New Jersey 07974

Rectilinear shortest paths through polygonal obstacles in $O(n\ (\log\ n)^{3/2})$ time

Kenneth L. Clarkson, Sanjiv Kapoor, and Pravin M. Vaidya

AT&T Bell Laboratories
Murray Hill, New Jersey 07974

Abstract

The problem of finding a rectilinear shortest path amongst obstacles may be stated as follows: Given a set of obstacles in the plane find a shortest rectilinear (L_1) path from a point s to a point t which avoids all obstacles. The path may touch an obstacle but may not cross an obstacle. We study the rectilinear shortest path problem for the case where the obstacles are non-intersecting simple polygons, and present an $O(n\ (\log n)^2)$ algorithm for finding such a path, where n is the number of vertices of the obstacles. This algorithm requires $O(n \log n)$ space. Another algorithm is given that requires $O(n(\log n)^{3/2})$ time and space. We also study the case of rectilinear obstacles in three dimensions, and show that L_1 shortest paths can be found in $O(n^2\ (\log\ n)^3)$ time.

1. Introduction

In this paper we consider the problem of finding rectilinear (L_1) shortest paths between points when there may be obstacles present. The problem may be formulated as follows: Given a set of obstacles in the plane find a shortest rectilinear (L_1) path from a point s to a point t which avoids all obstacles. The path may touch an obstacle but does not cross an obstacle. In [RLW] Rezende, Lee and Wu study a version of this problem where the obstacles are n rectangles with sides parallel to the coordinate axis. They present an $O(n \log n)$ algorithm for constructing the optimal path. Other shortest path problems have been studied in [LPW], [LP] where the metric used is Euclidean. Larson and Li [LL] studied a similar problem of finding all minimal distance rectilinear paths among a set of m source-destination pairs in the plane with polygonal obstacles. The algorithm in [LL] runs in $O(m(m^2+n^2))$ time where n is the number of vertices of the obstacles, and m is the number of source-destination pairs.

We shall describe an algorithm for finding a shortest rectilinear (L_1) path between s and t for the case where the obstacles are non-intersecting simple polygons (the polygons may touch at vertices). The algorithm runs in $O(n(\log n)^2)$ time where n is the number of line segments defining the boundary of the simple polygonal obstacles. The input is given as a collection of n line segments which may intersect only at endpoints, and for each line segment it is known whether the interior of an obstacle (if any) lies to the left or the right of the segment. We shall refer to the given line segments as obstacle line segments. To simplify the presentation we shall assume that each obstacle has non-zero area. The algorithm described can be easily modified to handle obstacles which have zero area.

The rectilinear (L_1) shortest path algorithm proceeds by constructing a weighted visibility graph $VIS(V,E)$ whose vertices are points on the plane which include the points s and t, the $2n$ endpoints of the given line segments, and some additional points called Steiner points. Each vertex v is connected to some of the vertices visible from v. The weight of an edge (u,v) is the L_1 distance between u and v. The purpose of introducing Steiner points is to reduce the number of edges in the graph $VIS(V,E)$ to $O(n \log n)$. Finding a shortest path between s and t in the graph $VIS(V,E)$ gives a rectilinear shortest path between s and t that avoids all obstacles, and such a shortest path may be found in $O(n\ (\log n)^2)$ time using Dijkstra's shortest path algorithm. [D]

The shortest path algorithm of Fredman and Tarjan [FT] requires $O(e + v \log v)$ time, for a graph with e edges and v vertices. By modifying the construction of $VIS(V,E)$ to obtain a graph with $O(n(\log n)^{3/2})$ edges and $O(n(\log n)^{1/2})$ vertices, their algorithm can be applied to obtain shortest paths in $O(n(\log n)^{3/2})$ time.

How can adding more vertices to a graph allow it to have fewer edges? The key idea is this: suppose that three points p, q, and r have x-coordinates satisfying $p_x \le q_x \le r_x$ and y-coordinates satisfying $p_y \le q_y \le r_y$. (See Figure 1.1) Then the L_1 distance from p to r is the sum of the L_1 distance from p to q and from q to r. This implies that L_1 distance information between a set of points in the third quadrant around q and a set of points in the first quadrant around q can be succinctly represented via paths through q. Thus the addition of Steiner points

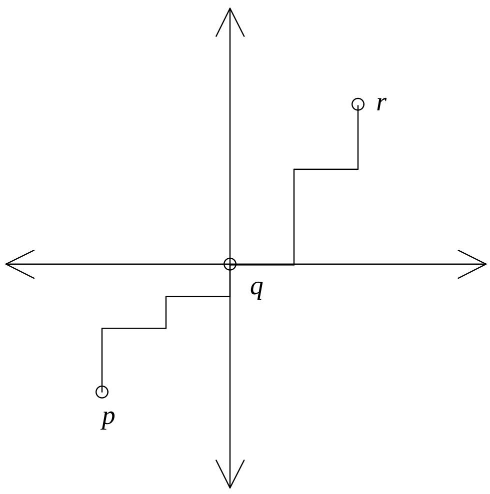

Figure 1.1. A path from p to r via q.

positioned like q can reduce the number of edges necessary in a visibility graph. This idea has been used previously by Guibas and Stolfi for minimum spanning trees under the L_1 metric. [GS]

In section 2 we describe the visibility graph $VIS(V,E)$, and show that finding a shortest path in $VIS(V,E)$ does give a shortest L_1 path that avoids all obstacles. In the third section we describe how to add Steiner points and construct the graph $VIS(V,E)$. In section 4, a modified version of $VIS(V,E)$ is given that uses more space but requires less time for shortest paths. Finally, in section 5 we describe how the algorithm can be extended to find a shortest rectilinear path in three dimensions when the obstacles are rectilinear objects with sides parallel to the coordinate axis.

2. The Visibility Graph $VIS(V,E)$

In this section we describe a weighted visibility graph $\overline{VIS}(\overline{V},\overline{E})$, where $\overline{V}$ is the set of points comprising the points s and t, and the (at most) $2n$ endpoints of the obstacle line segments. The weight of an edge in $\overline{E}$ is the L_1 distance between its two endpoints. If an edge $(p,\ q)$ is in $\overline{E}$ then p and q are visible from each other, but the converse does not always hold. A shortest path between s and t in the graph $\overline{VIS}(\overline{V},\overline{E})$ gives a shortest L_1 path between s and t which avoids all obstacles. However $\overline{VIS}(\overline{V},\overline{E})$ may have $\Omega(n^2)$ edges, so we actually construct a sparse visibility graph $VIS(V,E)$ whose vertices consist of all the points in $\overline{V}$ together with some extra Steiner points. For each edge (p,q) in $\overline{VIS}(\overline{V},\overline{E})$, there is a path of the same L_1 length between p and q in $VIS(V,E)$.

We shall define some notation. A point q is said to be visible from point p iff the straight line joining q to p does not intersect the interior of an obstacle. We let p_x and p_y denote the x and y co-ordinates of a point p. Let $C(p)$ denote the translated coordinate system with p as the origin. The first quadrant of $C(p)$ is the set of all points q such that $p_x \le q_x$ and $p_y \le q_y$. Number the remaining quadrants counterclockwise about p. The quadrants are closed regions, so a point on the vertical or the horizontal axis of $C(p)$ belongs to more than one quadrant. We define four different kinds of dominance relations among points, and say that p i-dominates q if q is in quadrant i of $C(p)$, for $i = 1,2,3,4$. A point q is said to be i-dominated by p iff p i-dominates q.

The graph $\overline{VIS}(\overline{V},\overline{E})$ is defined as follows: the vertex set $\overline{V}$ consists of the points s and t and the (at most) $2n$ endpoints of the given n obstacle line segments. We will indicate the edges (p,q) in $\overline{VIS}(\overline{V},\overline{E})$ when q is in the first quadrant of $C(p)$. (The other edges are defined symmetrically.) To do this, the following definitions will be useful.

Let L_V and L_H denote the line segments or rays defined as follows. As we move up from p along the y-axis of $C(p)$, L_V is the first obstacle line segment which intersects the positive y-axis of $C(p)$, and also intersects the interior of the first quadrant of $C(p)$. If there is no such line segment, then L_V is the vertical axis of $C(p)$. As we move right from p along the x-axis of $C(p)$, L_H is the first obstacle line segment which intersects the positive x-axis of $C(p)$, and also intersects the interior of the first quadrant of $C(p)$. If there is no such line segment, then L_H is the horizontal axis of $C(p)$.

Let $\pi(p)$ denote the set of points in the first quadrant of $C(p)$ that are not blocked from p by L_H or L_V. That is, a point r in the first quadrant of $C(p)$ is in $\pi(p)$ if and only if the line segment $\overline{pr}$ does not intersect L_H or L_V. We include in $\pi(p)$ all the points on the boundary of $\pi(p)$ except the point p itself. The region $\pi(p)$ is illustrated in Figure 2.1.

For $p,q \in \overline{V}$ and q in the first quadrant of $\overline{VIS}(\overline{V},\overline{E})$, an edge $\{p,q\}$ is included in $\overline{VIS}(\overline{V},\overline{E})$ if and only if q is in a set $S(p)$, which is defined as follows. A point q is in $S(p)$ if and only if both of the following hold.

1. There is no $q' \in \overline{V}$ such that q' is in $\pi(p)$ and q' 1-dominates q.
2. q is visible from p.

We will also need a set $S'(p)$, whose definition follows that of $S(p)$, with the removal of condition 2.

We will prove below that a shortest path in $\overline{VIS}(\overline{V},\overline{E})$ between vertices in $\overline{V}$ is a shortest L_1 path that avoids all obstacles. To prove this, a geometrical structure containing $S(p)$, called a staircase, will be useful. Consider that portion of the region $\pi(p)$ which is visible from p, and is not 1-dominated by points in $S(p)$. The boundary of this region consists of points in the positive x and y axes of $C(p)$, points in L_H and L_V, and a section containing the points of $S(p)$, which will be called a staircase. (See Figure 2.2.) On this staircase, the points in $S(p)$

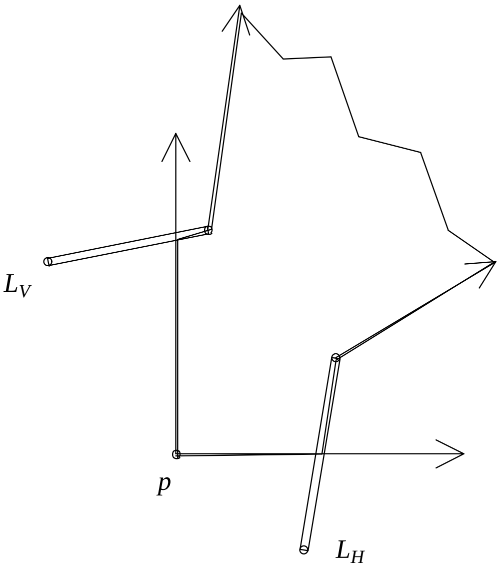

Figure 2.1. The polygonal region $\pi(p)$.

are arranged so that the y-coordinate decreases with increasing x-coordinate. We have the following lemma:

Lemma 2.1. Adjacent vertices (points) on the staircase are connected by at most three line segments: first a horizontal line segment, then a line segment with negative slope, and finally a vertical line segment.

Proof. Consider that subset of the region $\pi(p)$ that is not 1-dominated by points in $S'(p)$. The boundary of this subset of $\pi(p)$ consists of portions of the x and y axes, portions of L_H and L_V, and also of a collection of alternating vertical and horizontal line segments incident to $S'(p)$. Call the latter collection the $S'(p)$-staircase. Suppose that p_1 and p_2 are consecutive vertices in $S(p)$. Consider that portion of the $S'(p)$-staircase between p_1 and p_2. Let h be the horizontal segment incident to p_1, and let v be the vertical segment incident to p_2. Let l be the obstacle line segment whose intersection with h has smallest x coordinate in $C(p)$, and has minimum slope among all such line segments. We will show that l has negative slope, that l must intersect v, and that l does not intersect any of the horizontal or vertical segments in the $S'(p)$-staircase between p_1 and p_2. This is sufficient to prove the lemma.

Suppose that l has negative slope and crosses h. Consider the first vertical segment of the $S'(p)$-staircase that l crosses below its intersection with h. If this is v, then we are done. Otherwise, call this vertical segment v_1, and let p_3 be the point in $S'(p)$ incident to v_1. We will show that p_3 is visible to p.

Suppose p_3 is not visible to p. Then some obstacle line segment l' crosses $\overline{pp_3}$. We know that l' cannot cross $\overline{pp_1}$. It cannot intersect h, by the choice of l. It cannot cross l, since l is an obstacle line segment. Therefore, l' has an endpoint r which is in a region whose boundary points are all not 1-dominated by any point in $\pi(p) \cap \overline{V}$. Since r 1-dominates some of these boundary points, r cannot exist. Therefore, p_3 is visible to p, hence is in $S(p)$. If p_3 has a lower y-coordinate than p_2, then p_2 is not visible to p, which is a contradiction. Otherwise, p_1 and p_2 are not consecutive points in $S(p)$, also a contradiction. Therefore, l intersects the $S'(p)$-staircase for the first time at v. The lemma follows, for the case where l has negative slope and crosses h.

If l has positive slope and crosses h, then either it has an endpoint which is in a region all of whose points are not 1-dominated, or else l crosses either the x or y axis, and either p_1 or p_2 are not visible to p. These are contradictions.

Suppose that l touches h but does not cross h. Then the endpoint r of l on h is in $S'(p)$. If r is p_1, then we are done. If r is visible to p, then it must be p_2, and we are done. If not, then there must be an obstacle line segment that crosses $\overline{pr}$. This line segment cannot cross $\overline{pp_1}$ or h, and hence has an endpoint in a region all of whose points are not 1-dominated. ■

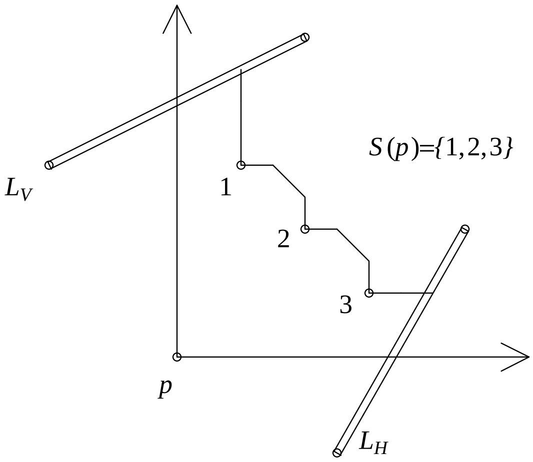

Figure 2.2. The set $S(p)$ and the staircase.

For each $p \in \overline{V}$, there are staircase structures in the second, third, and fourth quadrants of $C(p)$ similar to the staircase structure in the first quadrant defined by $S(p)$. These four staircase structures in the four quadrants of $C(p)$ define a region (which may be unbounded) whose interior contains exactly one point in $\overline{V}$, namely the point p. Thus a path from p to q must intersect one of the four staircase structures corresponding to p.

Using the staircase structures we can show that if a shortest L_1 path between s and t crosses the staircase structure and passes through p then it can be modified to pass through the points in $S(p)$. This leads to the following lemma about the

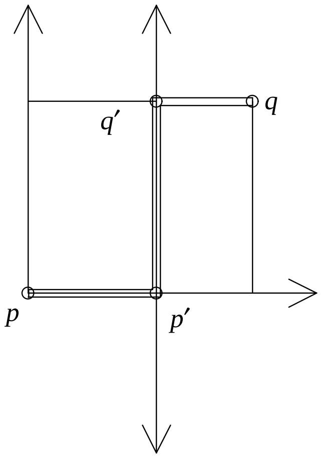

Figure 2.3. A path from p to q for Case 1.

graph $\overline{VIS}(\bar{V},\bar{E})$.

Lemma 2.2. Let p and q be points in $\bar{V}$. Then a shortest path from p to q in $\overline{VIS}(\bar{V},\bar{E})$ defines a shortest L_1 path from p to q which avoids all obstacles.

Proof. Consider a shortest path from p to q which avoids all obstacles. This path must intersect one of the four staircase structures corresponding to p. Suppose the path first crosses the staircase structure defined by $S(p)$ in the first quadrant. (The other three cases are symmetrical). The path crosses the staircase across either a vertical or horizontal line segment originating from a point in $S(p)$, say p_1, and can be altered to pass through the point p_1, without changing. length of the path. The path from p_1 to q can be similarly altered without changing the length of the path. This procedure is finite since no point (vertex) will be repeated in the modified path being constructed. The end result is a path from p to q in which we travel in a straight line between a pair of adjacent vertices in the path. Furthermore, the final path corresponds to a path between p and q in $\overline{VIS}(\bar{V},\bar{E})$. ■

We shall now describe the relationship between $\overline{VIS}(\bar{V},\bar{E})$ and $VIS(V,E)$. Let q be a point in $S(p)$. Consider the rectangle R_{pq} formed by q and p at the diagonal endpoints. Let R_{pq} include all of the points in its interior. We note that none of the obstacle line segments can terminate in the rectangle R_{pq}. If there were an obstacle line segment which terminated in R_{pq}, then there would be a point in $\bar{V} \cap \pi(p) \setminus \{q\}$ which would 1-dominate q, and since q is in the set $S(p)$ this cannot happen. The shortest path between p and q in $VIS(V,E)$ depends on whether the line segments L_H and L_V do or do not intersect the rectangle R_{pq}. There are two cases.

Case 1. Neither L_H nor L_V intersect the rectangle R_{pq}.
In this case the rectangle R_{pq} is contained in the region $\pi(p)$. Also, none of the obstacle line segments can terminate in the region R_{pq}. Therefore, none of the obstacle line segments intersect the rectangle R_{pq}. Consider a vertical line $\hat{L}$ which crosses the rectangle. Let p' and q' be the horizontal projections of p and q onto this vertical line. The construction in section 3 ensures that $VIS(V,E)$ contains the Steiner points p' and q', together with horizontal paths from p to p' and q to q', and a vertical path from p' to q', for some suitable vertical line $\hat{L}$. This gives a path in $VIS(V,E)$ from p to q of the same L_1 length as the straight line path from p to q. This path is illustrated in Figure 2.3.

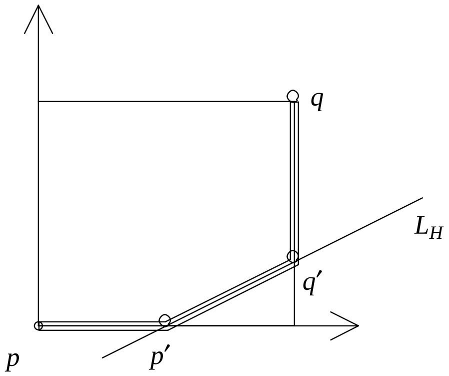

Figure 2.4. A path from p to q for Case 2.

Case 2. At least one of L_H and L_V intersects the rectangle R_{pq}.
Suppose L_H intersects the rectangle R_{pq}. (The case when L_V intersects R_{pq} is similar). Let p' and q' be the projections of p and q onto L_H. Then p' is visible from p, and we shall show that the vertical line through q intersects L_H, and that q' is visible from q. The construction in section 3 ensures that in $VIS(V,E)$ there is is a horizontal path from p to p', a vertical path from q to q', and a path along the line segment L_H from p' to q'. Thus in $VIS(V,E)$ there is a path from p to q of the same L_1 length as the straight line path from p to q. Such a path is illustrated in Figure 2.4. It remains to be shown that q' exists and is visible from q. We note that L_H cannot terminate in the region R_{pq}. If L_H is the x-axis of $C(p)$ then L_H and the vertical line through q must intersect. If L_H is not the x-axis of $C(p)$ then L_H must intersect the interior of the first quadrant of $C(p)$ and thereby must intersect the interior of R_{pq}, and since L_H cannot terminate in R_{pq}, the vertical line through q must intersect L_H. So q' exists. Furthermore, if an obstacle line segment intersects the segment $\overline{qq'}$ then it must terminate in R_{pq}, which also cannot happen as noted earlier. So q' must be visible from q.

These considerations immediately give the following lemma.

Lemma 2.3. Let p and q be points in V. Then a shortest path from p to q in $VIS(V,E)$ defines a shortest L_1 path from p

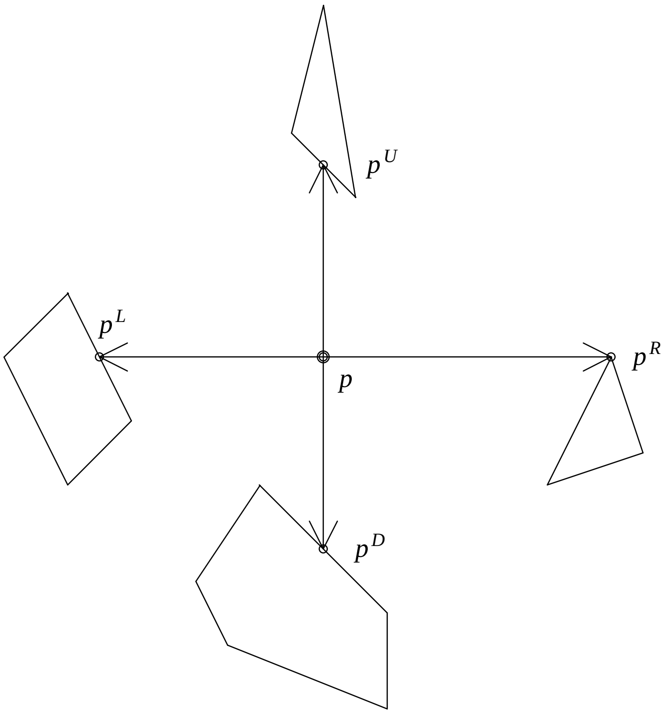

Figure 3.1. The horizontal and vertical projections of p.

to q which avoids all obstacles.

3. Construction of $VIS(V,E)$

In this section, the Steiner points and the edges of $VIS(V,E)$ are specified, and an algorithm is given for the construction of $VIS(V,E)$. The specification of $VIS(V,E)$ will imply that the requirements in Case 1 and Case 2 in section 2 are satisfied, so that Lemma 2.3 holds. Also, the construction of $VIS(V,E)$ will be shown to require $O(n(\log n)^2)$ time. For simplicity we shall assume that each obstacle has non-zero area. The vertex set V will consist of all the vertices in $\overline{V}$ plus some Steiner points.

There are two types of Steiner points in V. One type is suggested by Case 2 in section 2: for any endpoint p of a given obstacle line segment, include in V the horizontal and vertical projections of p onto visible obstacle line segments, as in Figure 3.1. Let p^U, p^D, p^L, and p^R denote these projections up, down, left, and right of p, respectively. Include in E edges between p and each of these (at most) four points. For each obstacle line segment e, there is a linear ordering on the vertices in V which lie on e, and we include in E edges between vertices in $V \cap e$ that are adjacent. These Steiner points, and the associated edges, are exactly those required by Case 2. We shall denote Steiner points which are projections of endpoints onto obstacle line segments as type 2 Steiner points. Note that there are $O(n)$ type 2 Steiner points.

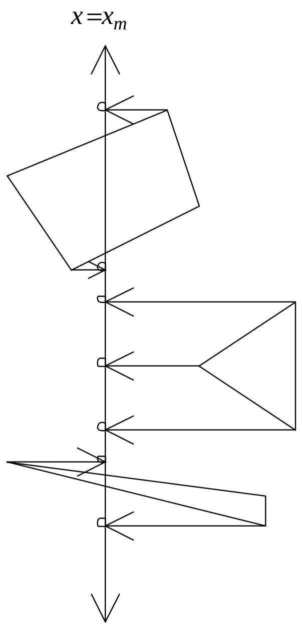

Figure 3.2. Type 1 Steiner points on line $x=x_m$.

To include in $VIS(V,E)$ Steiner points that are appropriate for Case 1 in section 2, the following recursive construction is employed: let x_m denote the median of the x-coordinates of the obstacle line segment endpoints. For each endpoint p, include in V the Steiner point $p'=(x_m,p_y)$, if that point is visible to p. (For example, if $p_x<x_m$ and $p_x^R>x_m$, then p' will be included in V.) For each endpoint p and each such Steiner point p', include in E an edge between p and p'. Apply this construction recursively for the endpoints with x-coordinates less than x_m, and for those with x-coordinates greater than x_m. A similar construction is also applied using the y-coordinates of the endpoints. After finding all Steiner points, we include edges between vertices with the same x (y) coordinate as follows. For all of the vertices V' in V with the same x (or y) co-ordinate, include in E edges between adjacent (consecutive) vertices in V' that are visible to each other (see Figure 3.2). This construction ensures that for any two endpoints, there is an appropriate line $\hat{L}$ between the two, as required in Case 1, together with the necessary Steiner points and edges. These Steiner points, which are projections of endpoints onto non-obstacle vertical and horizontal lines, will be called type 1 Steiner points. Observe that at a recursive step, $O(n)$ type 1 Steiner points are introduced, so that $O(n\log n)$ type 1 Steiner points are included in V by this construction. Note also that the given construction describes an algorithm for computing the type 1 Steiner points in $O(n\log n)$ time, once the type 2 Steiner points are available.

This completes the specification of $VIS(V,E)$. We note that in the above construction of $VIS(V,E)$ s and t are treated just like endpoints, so s and t are also projected and there are Steiner points corresponding to s and t. Also, note that there are $O(n\log\ n)$ vertices in V. It may be helpful in understanding the construction to note that for each endpoint p, the Steiner points associated with p are on the segments $\overline{p^L p^R}$ and $\overline{p^D p^U}$, and there are $O(\log\ n)$ such Steiner points. The Steiner points on $\overline{p^L p^R}$ thus have y-coordinate p_y, and the x-coordinates of such points are x-coordinates of some endpoints. Similarly, the Steiner points on $\overline{p^D p^U}$ have x-coordinate p_x, and the y-coordinates of such points are the y-coordinates of some endpoints.

The following lemma is implied by the above discussion, together with the observation that there are $O(1)$ edges incident to any Steiner point, and $O(\log\ n)$ edges incident on obstacle line segment endpoints.

Lemma 3.1. The graph $VIS(V,E)$ has $O(n\log\ n)$ vertices and edges.

It remains to describe an algorithm for constructing $VIS(V,E)$. The algorithm begins with a standard sweep line technique. Type 2 Steiner points may be obtained in $O(n\ \log\ n)$ time by first sweeping the obstacles by a horizontal line, and then by a vertical line. During each sweep the projections onto an obstacle line segment are generated in order, and at the end of the two sweeps an ordered list of the Steiner points on an obstacle line segment may obtained by merging two ordered lists. This ordered list readily gives the edges of $VIS(V,E)$ between the Steiner points on each obstacle line segment.

As noted above, the type 1 Steiner points may be generated by the recursive procedure described above. The ordering of the vertices on each dividing line (such as $x = x_m$) is readily determined via sorting in $O(n(\log\ n)^2)$. The visibility relations among the vertices on the same horizontal or vertical line, and hence the edges between such vertices, are also readily determined in $O(n(\log\ n)^2)$ time using a sweeping line: as a line sweeps through the obstacles, the intersection order of the obstacle line segments is maintained using a binary search tree. In addition to performing updates at line segment endpoints, the search tree is used to determine, for each vertex, the interval between obstacles, on a horizontal or vertical line, that contains the vertex. This yields the necessary visibility relations, and requires $O(\log\ n)$ time per vertex.

From this discussion we have:

Lemma 3.2. The graph $VIS(V,E)$ can be found in $O(n(\log\ n)^2)$ time.

4. A different construction of $VIS(V, E)$ for better running time

In this section we shall describe a construction for $VIS(V, E)$ slightly different from the one described in section 3. Type 2 Steiner points and edges between them will be introduced in exactly the same manner as in section 3. However, Type 1 Steiner points appropriate for Case 1 in section 2 will be obtained in a slightly different manner. A total of $O(n\ (\log n)^{1/2})$ Type 1 Steiner points (rather than $O(n\ \log n)$ as in section 3) will be introduced. $VIS(V, E)$ will contain $O(n\ (\log n)^{1/2})$ vertices, and $O(n\ (\log n)^{3/2})$ edges, and it will require $O(n\ (\log n)^{3/2})$ time to extract $VIS(V, E)$. We note that by the construction in this section $VIS(V, E)$ will contain fewer vertices but more edges than by the construction in section 3. This will enable us to find a shortest path between s and t in $O(n\ (\log n)^{3/2})$ time using the shortest path algorithm in [FT]. We shall now describe the modified construction for $VIS(V, E)$.

Type 2 Steiner points and edges between them are obtained in exactly the same manner as in section 3. Type 2 Steiner points and the associated edges are exactly those required by Case 2 in section 2.

Type 1 Steiner points and associated edges are obtained by a modification of the recursive construction in section 3. Let x_m be the median of the x-coordinates of the obstacle line segment endpoints. Instead of projecting all the endpoints onto the line $x = x_m$, we shall project only $O(n/\sqrt{\log n})$ of them. The endpoints are divided into $n/\sqrt{\log n}$ horizontal strips, each strip containing $O(\sqrt{\log n})$ endpoints. Let U be the set of points defined as follows. For each end point p, include in U the point (x_m, p_y) if and only if (x_m, p_y) is visible from p. U is the set of potential Steiner points, in that some of the points in U will be added to V. For each horizontal strip, Steiner points and edges are included in $VIS(V, E)$ as follows. Consider a horizontal strip H. Let p^t and p^b be those points in $H \cap U$ which have the largest and the smallest y-coordinate respectively, among all the points in $H \cap U$. Include in V both the points p^t and p^b. Let V_H be the set of all the endpoints located in the strip H together with the Steiner points p^t and p^b. For a pair of points p and q in V_H, we include the edge (p, q) in the edge set E if and only if

1. The rectangle R_{pq} formed by p and q at the diagonal endpoints does not contain an obstacle line segment endpoint other than p or q.
2. p is visible from q.

The above procedure for generating Type 1 Steiner points is repeated for those endpoints with x-coordinate less than x_m, and for those endpoints with x-coordinate greater than x_m. This generates a binary recursion tree of depth at most $\log_2 n$. At a node in this recursion tree, with r points, the endpoints are divided into $\frac{r}{\sqrt{\log n}}$ horizontal strips, each strip containing

$O(\sqrt{\log n})$ points. Thus at each level of recursion $O(\frac{n}{\sqrt{\log n}})$ Type 1 Steiner points are added to V and $O(n\ (\log n)^{1/2})$ edges are added to E. So a total of $O(n\ (\log n)^{1/2})$ Type 1 Steiner points and $O(n\ (\log n)^{3/2})$ edges get added to $VIS(V,\ E)$.

The Type 1 Steiner points and associated edges may be obtained in $O(n\ (\log n)^{3/2})$ time as follows. During the recursion we maintain two lists containing the endpoints sorted along the x and the y coordinates respectively. Let v be a node in the recursion tree, and let r be the number of endpoints associated with v. From the sorted lists at v, the sorted lists at the two children of v may be obtained in $O(r)$ time. Using these lists the endpoints at v may be split into horizontal strips also in $O(r)$ time. Once Type 2 Steiner points are available, a y-sorted list of the set of potential Steiner points U associated with node v is also obtained in $O(r)$ time, and using this list the Type 1 Steiner points at v are generated in $O(r)$ time. It then follows that the total time to split the endpoints into horizontal strips and to obtain Type 1 Steiner points during the entire recursion takes $O(n\ \log n)$ time. To show that the total time for extracting the associated edges is $O(n\ (\log n)^{3/2})$, it suffices to show the the edges corresponding to a strip may be extracted in $O(\log n)$ time as the total number of horizontal strips processed during the recursion is $O(n\ (\log n)^{1/2})$.

Consider a horizontal strip H. We first sort the points in V_H by x and y coordinates in $O(\sqrt{\log n}\ \log\log n)$ time. Let $p \in V_H$. Let $V_H(i,\ p)$ be the set of all those points in $V_H - \{p\}$ located in quadrant i of $C(p)$. The points in $V_H(i,\ p)$ that are not i-dominated by any other point in $V_H(i,\ p)$ may be isolated in $O(\sqrt{\log n})$ time, once x and y sorted lists of points in V_H are available. (For example, points in $V_H(1,\ p)$ would be processed in increasing x-order, and among the points that are not 1-dominated and that have thus far been encountered only the one with the smallest y-coordinate would be maintained). For each point q in $V_H(i,\ p)$, the edge $(p,\ q)$ is included in E if and only if

1. q is not i-dominated by any other point in $V_H(i,\ p)$.
2. Each of the projections p^U, p^D, p^L, p^R, q^U, q^D, q^L, q^R (these are the vertical and horizontal projections of p and q onto visible obstacle line segments) is outside the rectangle R_{pq} formed by p and q at diagonal endpoints.

Once these projections onto obstacle line segments (Type 2 Steiner points) are available, the edges between p and points in V_H are obtained in $O(\sqrt{\log n})$ time, and thus the edges corresponding to strip H are obtained in $O(\log n)$ time.

Once all the Steiner points have be generated, edges between vertices in V with the same x (y) coordinate are included as follows. For all of the vertices V' in V with the same x (y) coordinate, include in E edges between adjacent (consecutive) vertices in V' that are visible to each other. These edges may be obtained in a manner similar to section 3, at a cost of $O(\log n)$ per vertex in V, for a total of $O(n\ (\log n)^{3/2})$.

Finally, to see that the Type 1 Steiner points and the associated edges suffice for Case 1 in section 2, note that if the points p and q (in Case 1, section 2) are in the same horizontal strip the above construction ensures that there is an edge between them, otherwise there is a path between p and q via Steiner points located on a suitable vertical line L that separates them.

From the modified construction in this section and the algorithm of Fredman and Tarjan [FT], we have:

Lemma 4.1. Given polygonal obstacles in the plane with n vertices, a shortest L_1 path between given points s and t can be found in $O(n(\log\ n)^{3/2})$ time.

5. The Three-Dimensional Case

In this section we consider finding a shortest L_1 path in three dimensions between points s and t among three-dimensional non-intersecting rectilinear obstacles. Each obstacle may be a union of boxes (which may intersect) with sides parallel to the coordinate axes.

We next show that there is a shortest (L_1) path from s to t which passes through the vertices of the obstacles and an additional set of points added onto the edges on the boundaries of the obstacles. Let n be the number of obstacle vertices, and let $(z_1, z_2, z_3, \cdots, z_n)$, $(y_1, y_2, \cdots, y_n)$ and $(x_1, x_2, \cdots, x_n)$ be the z, y and x co-ordinates of the endpoints of the obstacles and of s and t. The additional set of points is obtained by intersecting the obstacles with the planes $z=z_1$, $z=z_2$, $\cdots$, $z=z_n$, $y=y_1$, $y=y_2$, $\cdots$, $y=y_n$ and $x=x_1$, $x=x_2$, $\cdots$, $x=x_n$. The number of such points is $O(n^2)$. We let V denote the set of the vertices of the obstacles, together with these extra $O(n^2)$ points. As in the planar case, we can define a set $S(p)$ of vertices related to p, such that a shortest path from any point to p can be altered to pass through vertices in $S(p)$. Let C_{pq} denote the rectilinear box with p and q at opposite diagonals. The additional vertices that lie on the edges of the obstacles ensure that for any vertex q in $S(p)$, the box C_{pq} intersects the obstacles only at p and q, as in Case 1 for the planar algorithm. (The analog of case 2 cannot occur for rectilinear obstacles.) Thus one only needs to ensure that there is a rectilinear path (in the constructed graph) from p to vertices in $S(p)$. A Steiner point construction, sketched below, will establish the existence of such paths. This construction will ensure that there will be a plane, which passes through C_{pq}, onto which p and q are projected to obtain Steiner points. On this plane, a shortest path between the projections can be established using the planar Steiner point construction.

We actually construct a graph among the three-dimensional points. First, Steiner points are added to the set of vertices V to form a set V^1. Each point p in V is projected along the z-axis in both the positive and the negative direction until it strikes an obstacle. These projected points, say p_r and p_l, are added to the set of vertices as Steiner points. These projections can be found in $O(n^2(\log n)^2)$ operations by doing a line sweep on a $x-z$ plane along the z axis for each of the $O(n)$ planes corresponding to the $O(n)$ y-coordinate values. The next set of Steiner points is obtained recursively as follows. Let P_{zm} be the plane $z=z_m$ perpendicular to the z axis, where z_m is the median of the z coordinate values of the vertices in V. The points on either side of this plane are projected onto this plane P_{zm}. The projection of a point p is obtained by intersecting the line $\overline{p_l p_r}$ with the P_{zm}. Moreover, the edges and the Steiner points among the points on the plane P_{zm} are added using the planar algorithm. The above procedure is repeated for the vertices on either side of the plane P_{zm}. The number of vertices and edges on the plane P_{zm} is $O(n^2 \log n)$ and can be generated in $O(n^2(\log n)^2)$ time, as described in the previous section. The entire recursive procedure generates a total of $O(n^2(\log n)^2)$ vertices and edges, and requires $O(n^2(\log n)^3)$ time. The edges joining adjacent points along the line $\overline{p_l p_r}$ are added after all the Steiner points have been introduced, by intersecting that line with the median planes obtained by the recursive procedure defined above. There are $O(\log n)$ such edges per vertex of V, giving a total of $O(n^2 \log n)$ such edges. This completes the specification of the addition of the edges. Now a shortest L_1 path from s to t can be found in this graph in $O(n^2(\log n)^3)$ time, using Dijkstra's shortest path algorithm. [D]

We have sketched the proof of the following:

Lemma 5.1. Given rectilinear obstacles in three dimensions with n vertices, a shortest L_1 path between given points s and t can be found in $O(n^2(\log n)^3)$ time.

6. Conclusion

We have described an $O(n(\log n)^2)$ algorithm for finding a shortest rectilinear path from s to t through simple polygonal obstacles, where n is the number of vertices on the obstacles. These ideas also led to an algorithm for shortest paths in three dimensions through rectilinear obstacles.

7. References

[D] E. W. Dijkstra, A note on two problems in connexion with graphs. Numer. Math., 1 (1959) pp. 269-271.

[FT] M. L. Fredman and R. E. Tarjan, Fibibonacci heaps and their uses, Proc. 25th IEEE Symp. on Theory of Computing, pp. 338-346, 1983.

[GS] L. Guibas and J. Stolfi, On computing all north-east nearest neighbors in the L_1 metric. Unpublished manuscript.

[LL] R. C. Larson and V. O. Li, Finding minimum rectilinear paths in presence of barriers, Networks, 11, 1981, pp. 285-304.

[LP] D. T. Lee and F. P. Preparata, Euclidean shortest paths among rectilinear barriers, Networks, 14, 1984, pp. 393-410.

[LPW] T. Lozano-Perez and M. A. Wesley, An algorithm for planning collision-amongst polyhedral obstacles, CACM, 22, 1979, pp. 560-570.

[RPW] P. J. de Rezende, D. T. Lee, and Y. F. Wu, Rectilinear shortest paths with rectangular barriers, Proc. 2nd Annual Conf. Computational Geometry, pp. 205-213, 1986.