An Optimal Algorithm for Constructing an Optimal Bridge between two Simple Rectilinear Polygons

Der-Perng Wang
Department of International Trade
National Pingtung Institute of Commerce, Pingtung, Taiwan, R.O.C.
Email:dpwang@sun1.npic.edu.tw

Abstract

Let P and Q be two disjoint rectilinear polygons in the plane. We say P and Q are in Case 1 if there exists a rectilinear line segment to connect them; otherwise, we say they are in Case 2. In this paper, we present optimal algorithms for solving the following problem. Given two disjoint rectilinear polygons P and Q in the plane, we want to add a rectilinear line segment to connect them when they are in Case 1, or add two rectilinear line segments, one is vertical and the other is horizontal, to connect P and Q when they are in Case 2. Our objective is to minimize the maximum of the L_1-distances between points in one polygon and points in the other polygon through one or two line segments. Let V(P) and V(Q) be the vertex sets of P and Q, respectively, and let |V(P)|=m and |V(Q)|=n. In this paper, we present O(m+n) time algorithms for the above two cases

1. Introduction

Let P be a simple polygon in the plane. For points p, q $\in$P, the geodesic path *gp*(p,q) is the shortest path in P connecting p and q. The geodesic distance *gd*(p,q) is the length of *gp*(p,q). Let P and Q be two disjoint polygons in the plane with boundaries ∂P and ∂Q, respectively. Points p$\in\partial$P and q$\in\partial$Q are mutually visible if the line segment $\overline{pq}$ does not enter the interior areas of P and Q. The bridge problem is to find a pair of mutually visible points p$\in\partial$P and q$\in\partial$Q that minimize

$$\max_{p'\in P} gd(p',p)+\overline{pq}+\max_{q'\in Q} gd(q,q')$$

where $\overline{pq}$ denotes the Euclidean distance between points p and q. Let V(P) and V(Q) be the vertex sets of P and Q, respectively, and let n denote $\max\{|V(P)|,|V(Q)|\}$. In [5], the authors considered three bridge problems according to whether polygons are convex or not. The results are summarized in Table 1.

Problem	Time complexity
convex-convex	O(n)
simple-convex	O(n*log*n)
simple-simple	O(n^2)

Table1

In this paper, we will consider the same problem for rectilinear case, and present an O(n) time algorithm to solve it.

A simple polygon P is rectilinear if the inner angles of all the vertices are either $\pi/2$ or $3\pi/2$. From now on, whenever we talk of a polygon, we mean a simple rectilinear polygon and whenever we talk of a line segment, we mean the line segment is vertical or horizontal.

Let P and Q be two disjoint polygons in the plane. We say P and Q are in Case 1 if there exists a rectilinear line segment to connect them; otherwise, we say they are in Case 2.

Let t_1 and t_2 be two points in a polygon P. The L_1-distance between t_1 and t_2, denoted as $L_1(t_1,t_2)$, is defined to be the length of a shortest rectilinear path connecting them inside of P.

The goal of our paper is to solve the following rectilinear bridge problem. There are two cases.

Case 1: Suppose that polygons P and Q are in Case 1. We want to find a line segment $\overline{pq}$ to connect P and Q, where $p \in \partial P$ and $q \in \partial Q$ and the line segment $\overline{pq}$ does not enter the interior areas of polygons P and Q. Our objective is to minimize

$F_1(p,q)=$

$$\max_{p' \in P} L_1(p',p) + \overline{pq} + \max_{q' \in Q} L_1(q,q'),$$

where $L_1(p',p)$ and $L_1(q,q')$ denote the L_1-distances from p to a point p' inside of a polygon P and from q to a point q' inside of a polygon Q, respectively. We call $\overline{pq}$ as a Type 1 bridge of P and Q.

Case 2: Suppose that polygons P and Q are in Case 2. We want to find two line segments $\overline{pr}$ and $\overline{rq}$, one is vertical and the other is horizontal, to connect P and Q, where $p \in \partial P$ and $q \in \partial Q$ and the line segments $\overline{pr}$ and $\overline{rq}$ do not enter the interior areas of polygons P and Q. Our objective is to minimize

$F_2(p,q)=$

$$\max_{p' \in P} L_1(p',p) + \overline{pr} + \overline{rq} + \max_{q' \in Q} L_1(q,q')$$

We call $\overline{pr} + \overline{rq}$ as a Type 2 bridge of P and Q.

The organization of our paper is as follows. In Section 2, we will present some useful lemmas about the problem. In Section 3, the detail of the algorithm will be presented. Some further works will be addressed at the final section.

2. Some Useful Lemmas

Let $p \in \partial P$. The L_1-farthest neighbour of p, denoted as f(p), is a point that has the maximal L_1-distance in P to p. In [6] it is shown that f(p) must be a vertex of P. If f(p) is not unique, then we choose the one which is the first one when we traverse the boundary from p clockwise.

Let p_1 and $p_2 \in \partial P$. We use the notation $[p_1;p_2]$ to represent the boundary edges from p_1 to p_2 clockwise.

From [1], we can have the following two lemmas.

Lemma 2.1: Let p_1 and $p_2 \in \partial P$. If $p_2 \in [p_1;f(p_1)]$, then $f(p_2) \in [f(p_1);p_1]$.

Lemma 2.2: Let p_1 and $p_2 \in \partial P$. If $f(p_1)=f(p_2)=v_i$ and v_i is not in $[p_1;p_2]$, then $f(x)=v_i$ for all points $x \in [p_1;p_2]$.

Assume $v_i \in V(P)$. We use $B(v_i)$ to denote the set of points on the boundary of P such that their L_1-farthest neighbours are v_i. That is $B(v_i)=\{x|x \in \partial P$ and $f(x)=v_i\}$. Based upon Lemma 2.2, we have the following lemma.

Lemma 2.3: Let $v_i \in V(P)$. If $B(v_i)$ is not empty, then $B(v_i)=[t_1;t_2]$ where t_1 and t_2 are two points on the boundary of P.

Lemma 2.3 can be proved directly from Lemma 2.2. For the sake of explanation, t_1 and t_2 are said to be the transition points of $B(v_i)$.

We use an example to explain $B(v_i)$. In Figure 1, polygon P has ten vertices which numbered $v_0,\ldots,v_9$ clockwise. For these vertices, we have $B(v_0)=[t_2;t_3]$, $B(v_1)=\phi$, $B(v_2)= \phi$, $B(v_3)= \phi$, $B(v_4)= \phi$, $B(v_5)=[t_3;t_4]$, $B(v_6)=[t_4;t_1]$, $B(v_7)= \phi$, $B(v_8)= \phi$ and $B(v_9)=[t_1;t_2]$. The boundary of P is partitioned into $B(v_0)\cup B(v_5)\cup B(v_6) \cup B(v_9)$. We use $T(P)$ to record all of the transition points. That is, $T(P)=\{t|$ t is a transition point of $B(v_i)$ if $B(v_i)\neq\phi$, for $0\leq i\leq|V(P)|-1\}$. For our example, $T(P)=\{t_1,t_2,t_3,t_4\}$. In the next section, we will prove that the transition points are the endpoints of our candidate Type 1 or Type 2 bridges.

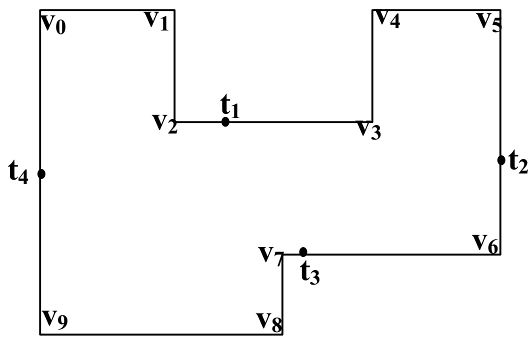


Figure 1

From [6], we have the following lemma.

Lemma 2.4: Suppose $v_i \in V(P)$ and e is an edge of P, then there is one point t on e such that $L_1(v_i,t)= L_1(v_i,e)$ and $L_1(v_i,x)= L_1(v_i,t) + \overline{tx}$, for all points $x \in e$.

We denote the unique point t on e with $L_1(v_i,t)=L_1(v_i,e)$ as the L_1-projection point of v_i onto e. Let $L_1(v_i,t)= t^i_e$. The L_1-distance function from v_i to x, for all points $x \in e$, can be expressed as $y(x)=L_1(v_i,x)= \overline{tx} +t^i_e =|x-t|+t^i_e$. If we consider edge e as x-axis, then the L_1-distance function from v_i to a point x in e can be expressed as a spearhead, as shown in Figure 2.

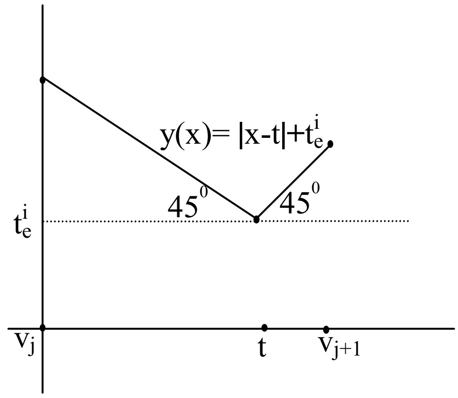


Figure 2

Lemma 2.5: Let e=$\overline{v_r v_{r+1}}$, v_j= f(v_r) and v_k= f(v_{r+1}). Then for all points x∈e, f(x) is either v_j or v_k.

Proof: If j=k, then this lemma is held under Lemma 2.2. Now let us consider j≠k. Assume f(x)=v_i for v_r≤x≤v_{r+1} and j≤i≤k. Let t_i be the L_1-projection point of v_i onto edge e and t_e^i be the L_1-distance from v_i to edge e, for j≤i≤k. Consider the set of functions {$y_i(x)=|x- t_i|+t_e^i$ |for j≤i≤k}. We draw these functions in Figure 3. It is clear that there are two groups of parallel line segments in the plane. Since v_j= f(v_r) and v_k= f(v_{r+1}), the maximum values of {$y_j(v_r)$, $y_{j+1}(v_r)$, … , $y_k(v_r)$} and {$y_j(v_{r+1})$,$y_{j+1}(v_{r+1})$,…, $y_k(v_{r+1})$} are $y_j(v_r)$ and $y_k(v_{r+1})$, respectively. Let y_{max} be the maximum value of $y_j(x)$ and $y_k(x)$ for v_r≤x≤v_{r+1}. That is, y_{max}=max($y_j(x)$,$y_k(x)$) for v_r≤x≤v_{r+1}. From geometric properties, we know that any function of {$y_i(x)= |x- t_i|+t_e^i$ for j<i<k} can not larger than y_{max} for v_r≤x≤v_{r+1}. Thus for any arbitrary point x of e, f(x)=v_j or v_k.

Q.E.D.

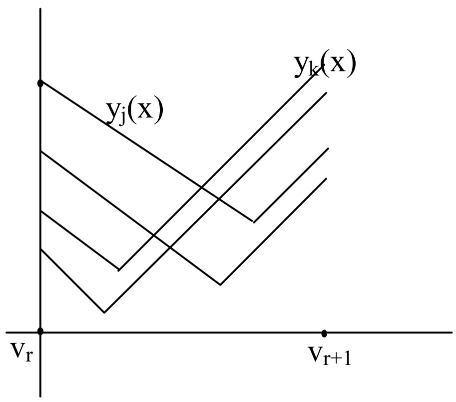
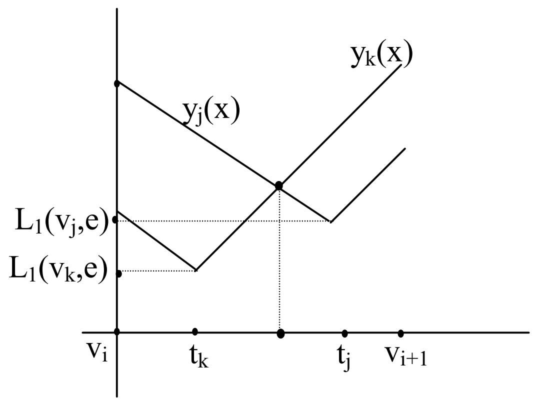


Figure 3

We can find the set *T*(P) of transition points circularly. When e=$\overline{v_i v_{i+1}}$ is considered, we first find out f(v_i) and f(v_{i+1}), w.l.o.g., assume they are v_j and v_k respectively. From Lemma 2.5, we know that edge e has 0 or 1 transition point. If j=k, then e has no transition point. If j≠k, then e has one transition point. In [6], the author had achieved the following results: Suppose v∈ *V*(P) and e is an edge of P. The author can preprocess polygon P in linear time such that both of the L_1-distance from v to e and the L_1-projection point of v onto e can be answered in O(1) time. Thus we can find the L_1-distance from v_j(v_k) to e and the L_1-projection point of v_j(v_k) onto e in O(1) time after linear time preprocessing. We can draw the spearheads of the the L_1-distance function from v_j(v_k) to a point x in e, as shown in Figure 4. Thus we can find the transition point on e in O(1) time.



t_k(t_j) is the L_1-projection point of v_k(v_j) onto e

Figure 4

Lemma 2.6: Given a polygon P with m vertices, the set *T*(P) of transition points can be found in O(m) time.

Proof: In [6], the author can find $f(v_i)$, for $0 \le i \le m-1$, in O(m) time. As discussed above, we can find one transition point in O(1) time. Since there are at most m transition points of P, it needs O(m) time to find all of the transition points of P.

Q.E.D.

Given a polygon P with m vertices, we use L_1P(P) to record the set of L_1-projection points from a vertex v_j onto an edge e= $\overline{v_i v_{i+1}}$, $0 \le i \le m-1$, where $v_j=f(x)$ for $x \in e$. Note that v_j may not be unique, but based upon Lemma 2.5, the number of v_j is at most two. In the next section, we will show that the set L_1P(P) of L_1-projection points may be also the end points of our candidate Type 1 or Type 2 bridges.

We can also find the set L_1P(P) of L_1-projection points circularly. When e= $\overline{v_i v_{i+1}}$ is considered, we first find out $f(v_i)$ and $f(v_{i+1})$, w.l.o.g., assume they are v_j and v_k respectively. Using the results of [6], we can find the L_1-projection points from vertices v_j and v_k onto e in O(1) time. Thus, we have the following Lemma.

Lemma 2.7: Given a polygon P with m vertices, the set L_1P(P) of L_1-projection points can be found in O(m) time.

3. Geometric Analysis and the Algorithm

Lemma 3.1: Suppose that polygons P and Q are in Case 1. Then there exists an optimal Type 1 bridge $\overline{pq}$ of P and Q such that p and q must be falling in one of the following two cases.

(1) $p \in V(P) \cup T(P) \cup L_1P(P)$, or

(2) $q \in V(Q) \cup T(Q) \cup L_1P(Q)$.

Proof: Suppose that $\overline{pq}$ is an optimal Type 1 bridge of P and Q and $p \notin V(P) \cup T(P) \cup L_1P(P)$ and $q \notin V(Q) \cup T(Q) \cup L_1P(Q)$. W.l.o.g., assume $\overline{pq}$ is horizontal and does not overlap any edges of P and Q.

Let us consider two Type 1 bridges $\overline{p_1q_1}$ and $\overline{p_2q_2}$ that satisfy one of the above two conditions and are nearest to $\overline{pq}$ at the upper and lower sides of $\overline{pq}$, respectively. It is clear that the points p,p_1 and p_2 are in the same edge of P and the points q,q_1 and q_2 are also in the same edge of Q. Therefore $\overline{pq} = \overline{p_1q_1} = \overline{p_2q_2}$. Based upon Lemma 2.5, we know that $f(p_1)=f(p)=f(p_2)$ and $f(q_1)=f(q)=f(q_2)$. Let $v_i=f(p_1)=f(p)=f(p_2)$ and v_j'= $f(q_1)=f(q)=f(q_2)$ where $v_i \in V(P)$ and $v_j' \in V(Q)$. The L_1-distance from v_i to the points in $[p_1;p_2]$ is monotone inscreasing or decreasing. Therefore we have $L_1(p,v_i) > \min(L_1(p_1,v_i),L_1(p_2,v_i))$. W.l.o.g., assume $L_1(p_2,v_i) < L_1(p,v_i)$. From Lemma 2.4, we know that $L_1(p,v_i)-L_1(p_2,v_i)= \overline{p_2p}$. Now, let us consider polygon Q. The L_1-distance from v_j' to the points in $[q_2;q_1]$ is also monotone increasing or decreasing. Therefore $L_1(q,v_j') > \min(L_1(q_1,v_j'), L_1(q_2,v_j'))$. If $L_1(q,v_j') > L_1(q_2,v_j')$, then $F_1(p,q)>F_1(p_2,q_2)$ and our assumption is false, i.e., $\overline{pq}$ is not an optimal Type 1 bridge. Otherwise, we know that $L_1(q_2,v_j')-L_1(q,v_j')= \overline{q_2q}$. Since $\overline{p_2p} = \overline{q_2q}$, $F_1(p,q)=F_1(p_2,q_2)$. $\overline{p_2q_2}$ is also an optimal Type 1 bridge.

Q.E.D.

Using the same reasoning of Lemma 3.1, we can have the following lemma.

Lemma 3.2: Suppose that polygons P and Q are in Case 2. Then there

exists an optimal Type 2 bridge $\overline{pr}+\overline{rq}$ of P and Q such that p∈ *V*(P) ∪*T*(P) ∪L_1P(P) and q∈ *V*(Q) ∪*T*(Q) ∪L_1P(Q).

Let us discuss how to solve Case 2 of the rectilinear bridge problem first. Base upon Lemma 3.2, we can transform the objective function F_2(p,q) of Case 2 into $F_2(p,q)=L_1(p,f(p))+\overline{pr}+\overline{rq}+L_1(q,f(q))$, for p∈ *V*(P) ∪*T*(P) ∪L_1P(P) and q∈ *V*(Q) ∪*T*(Q) ∪L_1P(Q).

Since there are $O(n^2)$ pairs in the above objective function, it needs $O(n^2)$ time to find the minimum value from these pairs by a bruteforce method. Before we present our linear time algorithm, some notations must be introduced first. It is clear that polygons P and Q are in Case 2 iff there exist two lines l_1 and l_2, one is vertical and the other is horizontal, such that polygons P and Q are in different sides of them.W.l.o.g., assume l_1 and l_2 are x and y axises, respectively, and polygons P and Q are in the first and third quadrants, respectively. For the sake of explanation, we say r is a turning point of a Type 2 bridge $\overline{pr}+\overline{rq}$. It is clear that r is in the second or fourth quadrants. Let p∈∂P and l be a vertical or horizontal line. We say p and l are mutually visible if there exists a point x∈l such that p and x are mutually visible. We use l_1(P) and l_2(P) [l_1(Q) and l_2(Q)] to denote the subset of *V*(P) ∪*T*(P) ∪L_1P(P) [*V*(Q) ∪*T*(Q) ∪L_1P(Q)] that are mutually visible from l_1 and l_2, respectively. For every point p in the plane, we use x_p and y_p to denote the x and y coordinates of p, respectively. Then the objective function F_2(p,q) for Case 2 can be transformed into

$F_2(p,q)=L_1(p,f(p))+|x_p|+|y_p|+L_1(q,f(q))+|x_q|+|y_q|$,

for p∈l_2(P) and q∈l_1(Q), if the turning point r is in the second quadrant.

for p∈l_1(P) and q∈l_2(Q), if the turning point r is in the fourth quadrant.

Let $\mu(p)=L_1(p,f(p))+|x_p|+|y_p|$ and $\mu(q)=L_1(q,f(q))+|x_q|+|y_q|$. Then the objective function F_2(p,q) for Case 2 can be transformed into

$F_2(p,q)=\mu(p)+\mu(q)$

for p∈l_2(P) and q∈l_1(Q), if the turning point r is in the second quadrant.

for p∈l_1(P) and q∈l_2(Q), if the turning point r is in the fourth quadrant.

The following is the formal algorithm for finding an optimal Type 2 bridge between polygons P and Q.

Algorithm

Optimal_Type_2_bridge(P,Q)

Input: Two polygons P and Q in Case 2

Output: An optimal Type 2 bridge $\overline{pr}+\overline{rq}$

1: Find the sets *T*(P) and *T*(Q) of transition points and the sets L_1P(P) and L_1P(Q) of L_1-projection points.

2: Find l_1(P), l_2(P), l_1(Q) and l_2(Q).

3: Let $\mu(p_1) = \min_{p\in l_1(P)} \mu(p)$, $\mu(p_2) = \min_{p\in l_2(P)} \mu(p)$, $\mu(q_1) = \min_{q\in l_1(Q)} \mu(q)$ and $\mu(q_2) = \min_{q\in l_2(Q)} \mu(q)$.

4. If $F_2(p_1,q_2) \le F_2(p_2,q_1)$ then return $\overline{p_1 r} + \overline{r q_2}$ where $r=(x_{p_1},y_{q_2})$; otherwise, return $\overline{p_2 r} + \overline{r q_1}$ where $r=(x_{q_1},y_{p_2})$.

The correctness of Algorithm Optimal_Type_2_bridge is derived from Lemma 3.2, we will show that it needs linear time to execute it.

Lemma 3.3: Suppose that polygons P and Q are in Case 2, we can find an optimal Type 2 bridge in linear time.

Proof: Based upon Lemmas 2.6 and 2.7, it needs linear time to execute Step 1. Step 2 can be done in linear time by using the result of [4]. It needs linear time to execute Step 3 [6]. Step 4 can be done in O(1) time.

Q.E.D.

Now, let us consider Case 1. Base upon Lemma 3.1, we know that one endpoint of the optimal Type 1 bridge must be in $V(P) \cup T(P) \cup L_1P(P) \cup V(Q) \cup T(Q) \cup L_1P(Q)$. In [4], the author proposed the following problem. Given a polygonal curve C with vertices $v_1,v_2,\ldots,v_n$, extend two horizontal segments from each vertex v_i, one in each direction until it meets another point of C. In [4], the author can solve this problem in O(n) time. Using the algorithm of [4], we can find the candidate Type 1 bridges in linear time. The following is the formal algorithm for finding an optimal Type 1 bridge between polygons P and Q.

Algorithm

Optimal_Type_1_bridge(P,Q)

Input: Two polygons P and Q in Case 1

Output: An optimal Type 1 bridge $\overline{pq}$

1: Candidate_bridge_set=ϕ.

2: Find the sets T(P) and T(Q) of transition points and the sets L_1P(P) and L_1P(Q) of L_1-projection points.

3: For every element p in $\{V(P) \cup T(P) \cup L_1P(P)\}$, find a set of points $S_q \subset \partial Q$, such that $\overline{pq}$ is a Type 1 bridge of polygons P and Q for $q\in S_q$. Candidate_bridge_set= Candidate_bridge_set$\cup\{\overline{pq}\}$ for $q\in S_q$.

4: For every element q in $\{V(Q) \cup T(Q) \cup L_1P(Q)\}$, find a set of points $S_p \subset \partial P$, such that $\overline{pq}$ is a Type 1 bridge of polygons P and Q for $p\in S_p$. Candidate_bridge_set= Candidate_bridge_set$\cup\{\overline{pq}\}$ for $p\in S_p$.

5:Evaluate the objective function F_1(p,q) for every element $\overline{pq}$ in Candidate_bridge_set.

6:Output $\overline{pq}$ which is an element in Candidate_bridge_set and has a minimum value of the objective function F_1(p,q).

The correctness of Algorithm

Optimal_Type_1_bridge is derived from Lemma 3.1. We will show that it needs linear time to execute it.

Lemma 3.4: Suppose that P and Q are in Case 1, we can find an optimal Type 1 bridge in linear time.

Proof: Based upon Lemmas 2.6 and 2.7, it needs linear time to execute Step 2 . Using the results of [4], it needs (m+n) time to do Steps 3 and 4. And using the rseults of [6], for every element $\overline{pq}$ of the Candidate_bridge_set, we can evaluate the objective function F_1(p,q) in O(1) time after linear time preproceing. Therefore, it needs O(m+n) time to execute Step 5. It needs O(m+n) time to execute Step 6 by finding the minimum value from the O(m+n) values.

Q.E.D.

Based upon Lemmas 3.3 and 3.4, we can have the following theorem.

Theorem 3.5: Given two non-intersecting polygons P and Q with m and n vertices, respectively. We can solve the rectilinear bridge problem in O(m+n) time.

4. Conclusions

In this paper we present an optimal linear time algorithm to solve the optimal bridge problem under the rectilinear case. It is simple and easy to implement. We think this algorithm may have some applications in the VLSI layout design and the Euclidean Steiner tree problems.

A natural generalization of the considered problem is to modify the objective function, for example, rectilinear link distance metric[2] or combined L_1 and link distance metric[3].

5. References

[1] B. Aronov, S. J. Fortune and G. Wilfong, Furthest-site Geodesic Voronoi diagram, Discrete Computational Geometry, Vol 9, 1993, 217-255.

[2] M. de Berg, On rectilinear link distance, Computational Geometry: Theory and Applications, 1 (1991)13-34.

[3] M. de Berg, M. van Kreveld, B. Nilsson, and M. Overmars, Finding shortest paths in the presence of orthogonal obstacles using a combined L_1 and link metric, SWAT (1990) 213-224.

[4] B. Chazelle, Triangulating a simple polygon in linear time, Discrete Computational Geometry, 6 (1991) 485-524.

[5] Sung Kwon Kim and Chan-Su Shin, Computing the optimal bridge between two polygons, HKUST Research Report TCSC-99-14, HKUST, HongKong, 1999, (available at http: // www.cs.ust.hk /tcsc /RR/index_2.html).

[6] Sven Schuierer, An optimal data structure for shortest rectilinear path queries in a simple rectilinear polygon, International Journal of Computational Geometry & Applications, Vol. 6, No. 2 (1996) 205-225.