Constructing Piecewise Linear Homeomorphisms of Simple Polygons

Himanshu Gupta* Rephael Wenger†

December 5, 1995

*Stanford University, Stanford CA 94305 (hgupta@cs.stanford.edu) Work done while at The Ohio State University, Columbus, Ohio 43210.

†Ohio State University, Columbus OH 43210 (wenger@cis.ohio-state.edu). Supported in part by NSA grant MDA904-93-H-3026.

Suggested running header:

Constructing PL Homeomorphisms

Send proofs to:

Rephael Wenger
Dept. of Computer and Information Science
2015 Neil Avenue, Ohio State University
Columbus OH 43210

Abstract

Let P and Q be simple polygons with vertex sets $\{p_1, \ldots, p_n\}$ and $\{q_1, \ldots, q_n\}$, respectively. We present an algorithm to construct a piecewise linear homeomorphism between P and Q mapping each vertex $p_i \in P$ to $q_i \in Q$ by constructing isomorphic triangulations of P and Q. These isomorphic triangulations consist of $O(M \log n + n \log^2 n)$ triangles where M is the size of the optimal (minimum size) solution. The algorithm runs in $O(M \log n + n \log^2 n)$ time. We also give an $O(n + L + k \log k)$ algorithm for constructing k pairwise disjoint interior paths between k pairs of vertices in a simple polygon on n vertices using $O(L + k \log k)$ links. The number L is the sum of the interior link distances between the k pairs of vertices.

List of Symbols:

$O(n^2)$
$\Omega(n^2)$ - Omega
$\lfloor \log k \rfloor$
$\lfloor m/2 \rfloor$
$\mathrm{Vis}(p)$
$\hat{\mathrm{V}}\mathrm{is}(p)$
$\mathcal{C}$ - calligraphic C
$\{p_1, \ldots, p_n\}$
$\{q_1, \ldots, q_n\}$
δ - delta
γ - gamma
Γ - Gamma
T_Γ - T sub Gamma
π - pi
Π - Pi
$\hat{\Pi}$ - Pi hat
Ψ - Psi
σ - sigma
Σ - Sigma
τ - tau
ζ - zeta
$\subseteq$
$\in$
$\leq$, $>$, $\neq$, $\cap$, $\setminus$

1 Introduction

Let P and Q be simple polygons with vertex sets $\{p_1, \ldots, p_n\}$ and $\{q_1, \ldots, q_n\}$, respectively, listed in clockwise order around the polygons. A *homeomorphism* is a continuous 1-1 onto map with continuous inverse. A homeomorphism from P to Q is *piecewise linear* if there is a triangulation T_P of P such that the homeomorphism is linear on each triangle in T_P. The triangulation T_P may have vertices in the interior of P. We are interested in the problem of constructing a piecewise linear homeomorphism between P and Q which maps each vertex $p_i \in P$ to the corresponding vertex $q_i \in Q$. Moreover, we want to use as few "pieces" as possible in constructing our homeomorphism.

Problems of constructing homeomorphisms and its variants arise in the process of combining cartographic maps [7], mesh generation in computational fluid dynamics [11], and morphing in computer graphics and animation [4, 8]. Each of these areas has devised its own set of algorithms and heuristics for constructing homeomorphisms. These algorithms work well for their particular applications but they are not guaranteed to succeed on all inputs and their running times are not bounded as a function of input size. This and related papers [1, 5, 9] apply computational geometry to construct and analyze correct, complete algorithms for the problem of constructing a particular type of homeomorphism, a piecewise linear homeomorphism.

A piecewise linear homeomorphism induces a triangulation T_Q of Q which is *isomorphic* to the triangulation T_P of P. A triangulation T_P of P (possibly with interior vertices) is *isomorphic* to a triangulation T_Q of Q if there is a one-to-one, onto mapping f between the vertices of T_P and the vertices of T_Q such that p, p', p'' are vertices of a triangle in T_P if and only if $f(p), f(p'), f(p'')$ are vertices of a triangle in T_Q. Conversely, an isomorphic triangulation of P and Q defines a piecewise linear homeomorphism between P and Q. Each triangle in P with vertices p, p', p'' maps linearly to the triangle in Q with vertices $f(p), f(p'), f(p'')$. The size of a triangulation is the total number of vertices, edges and triangles in the triangulation. Thus our original problem of constructing piecewise linear homeomorphisms reduces to the problem of constructing minimum size isomorphic triangulations of P and Q where $p_i \in P$ is identified with $q_i \in Q$.

Aronov, Seidel and Souvaine in [1] show that any two simple polygons P and Q on n vertices have isomorphic triangulations mapping p_i to q_i of size $O(n^2)$. Their construction translates into an $O(n^2)$ algorithm for finding such a triangulation. This result is asymptotically worst-case optimal in the sense that there exist pairs of polygons which require at least quadratic number of additional vertices to produce isomorphic triangulations. Kranakis and Urrutia in [5] improved on the result by Aronov et al. with an algorithm which constructs isomorphic triangulations of size $O(n + r^2)$ in time $O(n + r^2)$ where r is the number of reflex vertices in P and Q.

The algorithms in [1] and [5] may add $\Omega(n^2)$ interior points, when only

$O(n)$ are required. Aronov et al. ask whether there exists a polynomial time algorithm for finding the minimum size isomorphic triangulation between two simple polygons. We were unable to construct such an algorithm or to show that the problem is NP-hard. Instead we present here an approximation algorithm which constructs isomorphic triangulations mapping p_i to q_i of size $O(M \log n + n(\log n)^2)$ where M is the size of the optimal solution.

In designing our algorithm, we confronted the subproblem of constructing k pairwise disjoint interior paths between k pairs of vertices in a simple polygon using as few line segments as possible. Again we were unable to give a polynomial time algorithm to find the minimum number of line segments required or show that the problem is NP-hard. Instead we give an approximation algorithm which produces k pairwise disjoint interior paths consisting of $O(n+L+k\log k)$ line segments where L is the sum of the interior link distances between each pair of vertices. L is clearly a lower bound on the size of the optimal solution. The algorithm runs in $O(n + L + k \log k)$ time.

In related research, Saalfeld in [7] considers the problem of constructing a piecewise linear homeomorphism between the convex hulls of two point sets $P = \{p_1, p_2, \ldots, p_n\}$ and $Q = \{q_1, q_2, \ldots, q_n\}$ where $p_i \in P$ maps to $q_i \in Q$. He gives an algorithm for constructing such a homeomorphism but possibly using an exponential number of triangles. Souvaine and Wenger in [9] give an $O(n^2)$ algorithm for constructing a homeomorphism between two rectangles with interior point sets $P = \{p_1, p_2, \ldots, p_n\}$ and $Q = \{q_1, q_2, \ldots, q_n\}$ where $p_i \in P$ maps to $q_i \in Q$. Their homeomorphism uses only $O(n^2)$ triangles. Pach, Shahrokhi and Szegedy prove that $\Omega(n^2)$ triangles are sometimes required [6].

2 Constructing Pairwise Disjoint Link Paths

Let P be a simple polygon on a set of n vertices and let s and s' be two points in P. The *link distance* between s and s' is the minimum number of line segments (*links*) in any polygonal path in P connecting s to s'. However, such a polygonal path may intersect the boundary of P in many points other than s and s'. The *interior link distance* between s and s', denoted $l(s, s')$, is the minimum number of line segments (*links*) in any polygonal path in the interior of P connecting s to s'. The interior link distance of s and s' may differ greatly from the link distance of s and s'. (See Figure 1.)

A polygonal path in the interior of P between s and s' which uses the minimum number of links is called a *minimum link interior path.* There are an (uncountably) infinite number of different minimum link interior paths between s and s'. Suri in [10] gives a linear time algorithm for constructing a minimum link path between two points in a simple polygon which can be easily modified to construct a minimum link interior path.

Let s_1, s'_1, s_2, s'_2 be distinct vertices occuring in the given order around P. There exist interior paths, γ_1, γ_2, connecting s_1 to s'_1 and s_2 to s'_2, respectively,

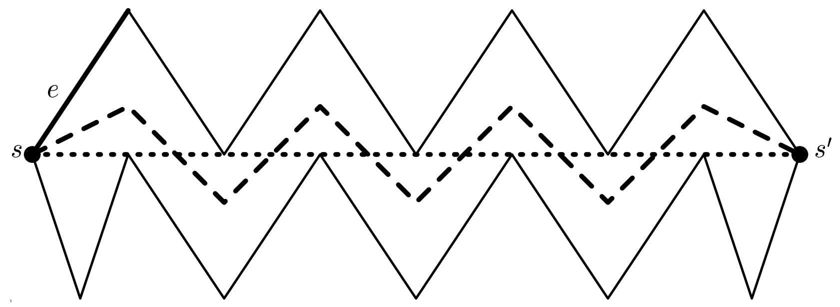

Figure 1: Minimum link and minimum interior link paths.

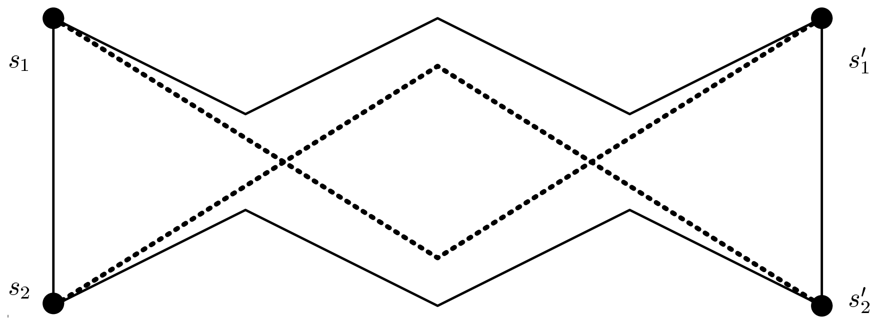

Figure 2: Intersecting Link Paths

such that γ_1 does not intersect γ_2. However, every minimum link interior path connecting s_1 to s'_1 may intersect every minimum link interior path connecting s_2 to s'_2. (See Figure 2.) We can construct pairwise disjoint paths by breaking γ_1 just before and after its first and last intersection point with γ_2 and then connecting the break points with a path which follows γ_2. Pairwise disjoint paths can also be constructed by breaking γ_2 and connecting the break points with a path following γ_1. One of these two constructions will require at most two additional links. Thus two additional links may be needed and will always suffice to connect s_1 to s'_1 and s_2 to s'_2 by interior paths which do not intersect.

A set $\Pi = \{(s, s')\}$ of pairs of distinct vertices of P is *untangled* if for every $(s_1, s'_1), (s_2, s'_2) \in \Pi$ there exist interior paths connecting s_1 to s'_1 and s_2 to s'_2 which do not intersect. Alternately, set Π is untangled if for every $(s_1, s'_1), (s_2, s'_2) \in \Pi$ vertices s_1, s'_1, s_2, s'_2 occur around the boundary of P in the given (clockwise/counter-clockwise) order. If set Π is untangled, then all the pairs $(s, s') \in \Pi$ can simultaneously be connected by pairwise disjoint paths.

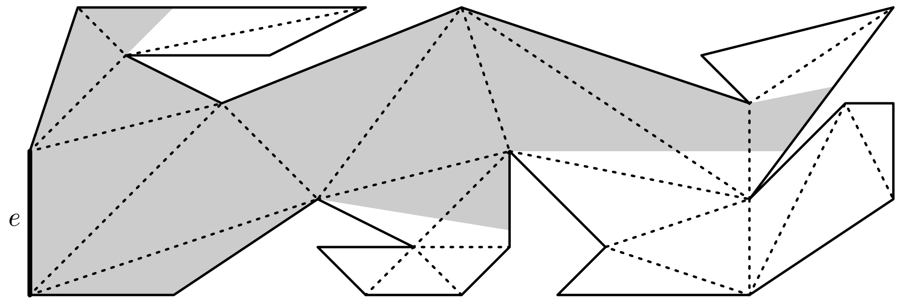

Figure 3: $\hat{\mathrm{V}}\mathrm{is}(e)$.

One can easily check whether Π is untangled in linear time.

Let P be a simple polygon on n vertices. Let $\Pi = \{(s, s')\}$ be an untangled set of k pairs of distinct vertices of P. Let $l(s, s')$ be the interior link distance from s to s' and let $L = \sum_{(s,s')\in\Pi} l(s, s')$. Clearly L total links are necessary to connect all the pairs by interior polygonal paths. However, an additional $\Omega(k \log k)$ links may be needed in some cases to ensure that these polygonal paths are pairwise disjoint. We omit the rather intricate construction since it is only tangentially related to the results in this paper. We will show that $O(k \log k)$ additional links suffice and that a set of such non-intersecting paths can be found in $O(n + L + k \log k)$ time.

The *interior link distance* between a point r and an edge e of P, denoted $l(e, r)$, is the minimum number of line segments (*links*) in any polygonal path in the interior of P connecting r to a point in the interior of e. Let R be a subset of the vertices of P of size m, let e be an edge of P and let $L = \sum_{r\in R} l(e, r)$. We first present an $O(n + L + m \log m)$ algorithm to construct m pairwise disjoint interior paths connecting the points in R with e using at most $2L + 2m\lfloor \log m \rfloor$ links. A modification of this algorithm solves the original problem of connecting pairs of vertices.

The *visibility polygon* from a point $p \in P$, denoted by $\mathrm{Vis}(p)$, is the set of points in P visible from p, i.e., $q \in \mathrm{Vis}(e)$ if and only if P contains line segment (p, q). The *visibility polygon* from a line segment $e \subseteq P$, denoted by $\mathrm{Vis}(e)$, is the set of points in P visible from e, i.e., $q \in \mathrm{Vis}(e)$ if and only if there exists a point $p \in e$ such that P contains line segment (p, q). Guibas et al. in [3] give a linear time algorithm for constructing $\mathrm{Vis}(e)$.

A point q may be visible from a point p or edge e, yet every open line segment connecting p and q or e and q may intersect the boundary of P. The

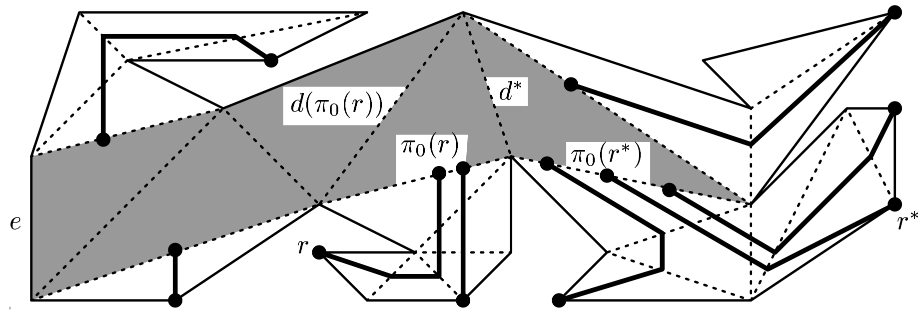

Figure 4: Region Γ, triangulation T_Γ and paths to the boundary of Γ.

clear visibility polygon from a point $p \in P$, denoted by $\hat{\mathrm{V}}\mathrm{is}(p)$, is the set of points in P "clearly visible" from p, i.e., $q \in \hat{\mathrm{V}}\mathrm{is}(p)$ if and only if the open line segment (p, q) lies in the interior of P. (See Figure 1 where s' is in $\mathrm{Vis}(s)$ but is not even on the boundary of $\hat{\mathrm{V}}\mathrm{is}(s)$.) Note that $\hat{\mathrm{V}}\mathrm{is}(p)$ is usually not closed. The *clear visibility polygon* from a line segment $e \subseteq P$, denoted by $\hat{\mathrm{V}}\mathrm{is}(e)$, is the set of points "clearly visible" from e, i.e., $q \in \hat{\mathrm{V}}\mathrm{is}(e)$ if and only if there exists a point $p \in e$ such that the open line segment (p, q) lies in the interior of P. (See Figure 3. Also see Figure 1 where s' is in $\mathrm{Vis}(e)$ but is not even on the boundary of $\hat{\mathrm{V}}\mathrm{is}(e)$.) The algorithm in [3] can be easily modified to construct $\hat{\mathrm{V}}\mathrm{is}(e)$ in linear time.

Let T be a triangulation of polygon P using no vertices other than those of P. Let v and e be a vertex and an edge of P, respectively. Edge d of triangulation T *separates* v from e if every interior path from v to e must cross d. Triangle t of triangulation T *separates* v from e if every interior path from v to e must cross t.

Lemma 2.1 *Let P be a simple polygon on n vertices with distinguished edge $e = \{w_e, w'_e\}$ and let R be a subset of* $\mathrm{Vert}(P) \setminus \{w_e, w'_e\}$ *of size m. A set of m pairwise disjoint interior paths connecting the vertices in R to the interior of e can be constructed in $O(n + L + m \log m)$ time using $2L + 2m\lfloor \log_2 m \rfloor$ total links where L is the sum of the interior link distances from $r \in R$ to e.*

Proof: Construct a triangulation T_P of P using no vertices other than those of P. For each vertex $v \in P$, let m_v be the number of points of R which lie clockwise between e and v, including v. Let r^* be the vertex in R where $m_{r^*} = \lceil m/2 \rceil$. A triangle t separates e from r^* if and only if t has vertices $v \neq r^*$ and $v' \neq r^*$ such that $m_v \leq m/2$ and $m_{v'} > m/2$. Let Γ be the union of

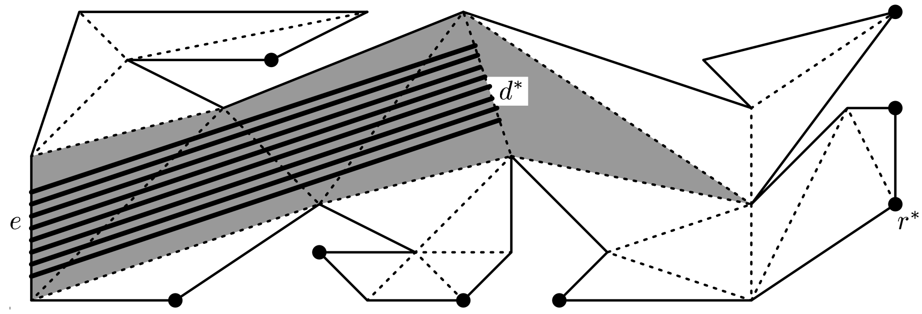

Figure 5: Line segments $\sigma(r)$.

the triangles of T_P which intersect $\hat{\mathrm{V}}\mathrm{is}(e)$ and separate e from r^*. (See Figure 4.) Γ is a simple polygon in P. Triangulation T_P of P induces a triangulation T_Γ of Γ. The diagonals of T_Γ are the edges of T_P which intersect $\hat{\mathrm{V}}\mathrm{is}(e)$ and separate e from r^*.

The boundary of Γ is composed of edges and chords of the original polygon P. Let $\mathcal{C}$ be the set of all such chords of P bounding Γ. Each chord $c \in \mathcal{C}$ divides P into two polygons. Let $P(c)$ be the one not containing Γ. Let w_c and w'_c be the endpoints of c. Let $R(c)$ be the points of $R \setminus \{w_c, w'_c\}$ in $P(c)$. For each $c \in \mathcal{C}$, recursively solve the problem of constructing pairwise disjoint paths connecting each $r \in R(c)$ to the interior of c.

Each point $r \in R$ either lies in Γ or lies in $R(c)$ for some $c \in \mathcal{C}$. If $r \in R(c)$, let $\pi_0(r)$ be the endpoint on c of the path from r to c. Otherwise, let $\pi_0(r)$ be r.

If T_Γ is a single triangle adjacent to e, then simply connect the points $\pi_0(r)$, $r \in R$, by pairwise disjoint line segments to e. Assume T_Γ contains more than one triangle. For each point $p \in \Gamma$, let $d(p)$ be the edge d of triangulation T_Γ which lies on the same triangle of T_Γ as p and separates p from e. Let $d^* = d(\pi_0(r^*))$. d^* is a diagonal of T_Γ, so d^* intersects $\hat{\mathrm{V}}\mathrm{is}(e)$. Let (p, p') be an open line segment in the interior of P where $p \in e$ and $p' \in d^*$. By translating (p, p') in both directions, we can find points $p_0, p_1 \in e$ and $p'_0, p'_1 \in d^*$ such that p_0, p_1, p'_0, p'_1 are vertices of a convex quadrilateral lying in P. Choose m pairwise disjoint line segments connecting the open line segment (p_0, p_1) to the open line segment (p'_0, p'_1). (See Figure 5.)

For each $r \in R$, let $\sigma(r)$ be the m_r'th line segment from (p_0, p_1) to (p'_0, p'_1) in counter-clockwise order along e. Note that $\sigma(r)$ intersects every diagonal of T_Γ. Let $\pi_1(r) = \sigma(r) \cap d(\pi_0(r))$ and $\pi_2(r) = \sigma(r) \cap e$. Connect $\pi_0(r)$ to e by

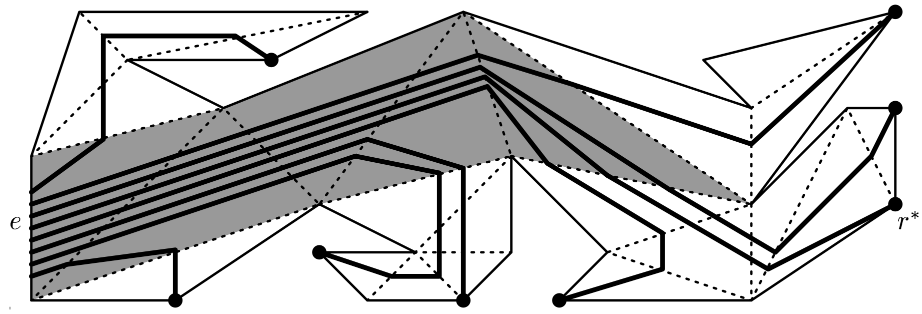

Figure 6: Paths from R to e.

line segments $(\pi_0(r), \pi_1(r))$ and $(\pi_1(r), \pi_2(r))$. This completes the construction of the paths to e. (See Figure 6.)

Let ζ_r be the path connecting $r \in R$ to e. We prove, by induction, that ζ_r has at most $2l(e,r) + 2\lfloor \log_2 m \rfloor$ links.

If r lies on the boundary of Γ, then ζ_r has at most two links and $2 \leq 2l(e,r)$. Assume r does not lie on the boundary of Γ. Path ζ_r intersects the boundary of Γ in a chord c of P. If $r^* \in P(c)$, then no line segment in the interior of P can intersect both c and e. Thus $l(c,r) \leq l(e,r) - 1$. By induction, the algorithm constructs a path from r to c of length at most $2l(c,r) + 2\lfloor \log_2 m \rfloor \leq 2l(e,r) - 2 + 2\lfloor \log_2 m \rfloor$. Since ζ_r contains at most two links in Γ, the length of ζ_r is at most $2l(e,r) + 2\lfloor \log_2 m \rfloor$.

Finally, if $r^* \notin P(c)$, then the size of $R(c)$ is at most $\lfloor m/2 \rfloor$. By induction, the algorithm constructs a path from r to c of length at most $2l(c,r) + 2\lfloor \log_2 m/2 \rfloor = 2l(e,r) + 2\lfloor \log_2 m \rfloor - 2$. Since ζ_r contains at most two links in Γ, the length of ζ_r is at most $2l(e,r) + 2\lfloor \log_2 m \rfloor$. Summing this bound over all paths ζ_r, $r \in R$, bounds the total number of links used by $2L + 2m\lfloor \log_2 m \rfloor$.

Γ and T_Γ can be constructed in linear time by first constructing $\hat{\mathrm{V}}\mathrm{is}(e)$ and then determining which triangles of T_P intersect $\hat{\mathrm{V}}\mathrm{is}(e)$ and separate e from r^*. This results in an $O(n^2)$ bound on the running time of the algorithm. However, using a technique introduced by Suri in [10], we can construct Γ in time proportional to the size of T_Γ after some initial preprocessing. Let τ be the set of triangles separating e from r^*. These triangles can be ordered by adjacency starting at the triangle containing e and ending at the one containing r^*. Let τ_j be the first j triangles in τ. Set τ_j can be constructed in $O(j)$ time by starting at the triangle containing e and processing adjacent triangles to determine which ones separate e from r^*.

Let P_j be the polygon formed from the union of τ_j. Instead of constructing $\hat{\mathrm{V}}\mathrm{is}(e)$ in P, find τ_j and P_j for some suitable small j and construct $\hat{\mathrm{V}}\mathrm{is}(e)$ in P_j. If $\hat{\mathrm{V}}\mathrm{is}(e)$ intersects every triangle of τ_j and $\tau_j \neq \tau$, then double j and try again. Otherwise, Γ is a subset of P_j. Determine which triangles in P_j intersect $\hat{\mathrm{V}}\mathrm{is}(e)$ and use them to construct Γ and T_Γ. This construction of Γ takes time proportional to $1 + 2 + 4 + \cdots + j/2 + j < 2j$. Since T_Γ contains at least $j/2$ triangles, the time is proportional to the size of T_Γ.

We are now ready to analyze the running time of the entire algorithm. The initial preprocessing is the triangulation of P and the calculation of m_v for each vertex of P. These two steps are done only once in the entire algorithm, not once for each recursive call. Using Chazelle's algorithm from [2], triangulating P takes linear time. Calculation of m_v for each vertex of P also takes linear time.

Note there is no need to explicitly construct the sets $R(c)$ or $P(c)$ in the recursive calls. Whenever required the number of points of $R(c)$ which lie clockwise between a vertex v and $c = (w_c, w'_c)$ can be calculated in constant time from m_v, m_{w_c} and $m_{w'_c}$.

The rest of the algorithm consists of partitioning P into subpolygons and routing the paths to e through those polygons. The total time spent routing the paths is proportional to the total number of links used which is $O(L + m \log m)$. The time constructing each subpolygon is proportional to the size of its triangulation. Since the subpolygons form a partition of P, the sum of their triangulations is $O(n)$ so the total running time is $O(n + L + m \log m)$. □

The *dual graph* of a triangulation T is obtained by replacing the triangles of T by vertices and connecting vertices corresponding to adjacent triangles. If T is the triangulation of a simple polygon using no vertices other than those of P, then the dual graph is a tree. If v', v'', v''' are three nodes of a tree (not necessarily distinct), then there is a unique node v such that any path from v' to v'' or from v'' to v''' or from v''' to v' must contain v. Correspondingly, if t', t'', t''' are three triangles of T (not necessarily distinct), then there is a unique triangle t such that any path between any two of these triangles must intersect t.

Lemma 2.2 *Let P be a simple polygon on n vertices and let $\Pi = \{(s, s')\}$ be an untangled set of k pairs of distinct vertices of P. A set of k pairwise disjoint interior paths connecting s to s' for each $(s, s') \in \Pi$ can be constructed in $O(n + L + k \log k)$ time using $2L + 4k\lfloor \log_2 k \rfloor + 11k$ total links where $L = \sum_{(s,s') \in \Pi} l(s, s')$.*

Proof: We first claim the following generalization of Lemma 2.1. Let P be a simple polygon on n vertices with distinguished edge e. Let $\hat{\Pi}(r, e_r)$ be a set of pairs of vertices and edges where either $e_r = e$ or e_r separates e from r. The vertices, but not the edges, are all distinct and are not endpoints of e. In

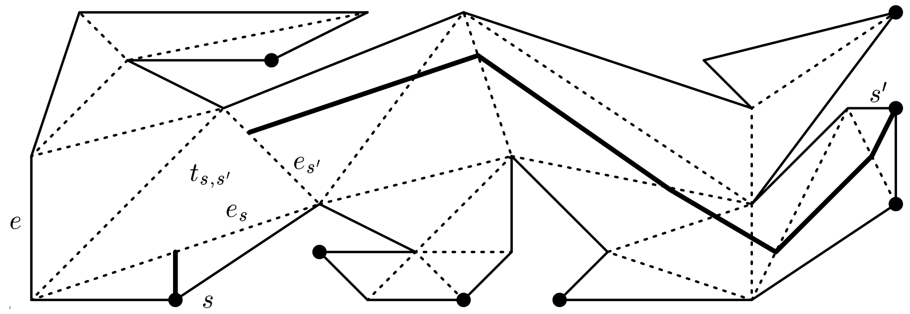

Figure 7: Triangle $t_{s,s'}$.

$O(n + L + m \log m)$ time, we can construct m pairwise disjoint interior paths ζ_r connecting r to the interior of e_r for each $(r, e_r) \in \hat{\Pi}$ where ζ_r has at most $2l(e_r, r) + 2\lfloor \log_2 k \rfloor + 2$ links. The number L is the sum of the interior link distances from r to e_r for each $(r, e_r) \in \hat{\Pi}$.

The proof follows the proof of Lemma 2.1. We triangulate P, define m_v and r^* and construct Γ as in the proof of Lemma 2.1. Note that e_{r^*} may not equal e nor even be in Γ. Removing Γ subdivides P into subpolygons and we recursively solve the problem of connecting pairs of vertices and edges in these subpolygons. We then connect each path with endpoint on the boundary of Γ to the appropriate edge in Γ. The only difference is that the connection may not be to e but to some edge e_r on the path to e.

If $e_r = e$ or r lies in Γ, then it follows directly from the proof of Lemma 2.1, that ζ_r has at most $2l(e_r, r) + 2\lfloor \log_2 k \rfloor$ links. If $e_r \neq e$ and ζ_r crosses the boundary of Γ at a chord c, then ζ_r has at most $2l(c, r) + 2\lfloor \log_2 k \rfloor + 2$ links. Since $l(c, r) \leq l(e_r, r)$, ζ_r has at most $2l(e_r, r) + 2\lfloor \log_2 k \rfloor + 2$ links. If e_r is not in Γ, then ζ_r does not intersect Γ and by induction ζ_r has at most $2l(e_r, r) + 2\lfloor \log_2 k \rfloor + 2$ links. The running time analysis is exactly the same as in Lemma 2.1.

We return to our original problem of connecting k pairs of vertices $(s, s') \in \Pi$. Arbitrarily choose an edge e of P. For each pair $(s, s') \in \Pi$, there is a unique triangle $t_{s,s'}$ such that any path from s to s' or any path from s to e or any path from s' to e must pass through $t_{s,s'}$. If triangle $t_{s,s'}$ does not contain s, let e_s be the edge of triangle $t_{s,s'}$ which separate $t_{s,s'}$ from s. Otherwise, let e_s be the edge of triangle $t_{s,s'}$ containing s. Similarly, let $e_{s'}$ be the edge of triangle $t_{s,s'}$ either separating $t_{s,s'}$ from s' or containing s'. (See Figure 7.) We construct a path from s to s' by constructing paths from s to e_s and from s' to $e_{s'}$ and then

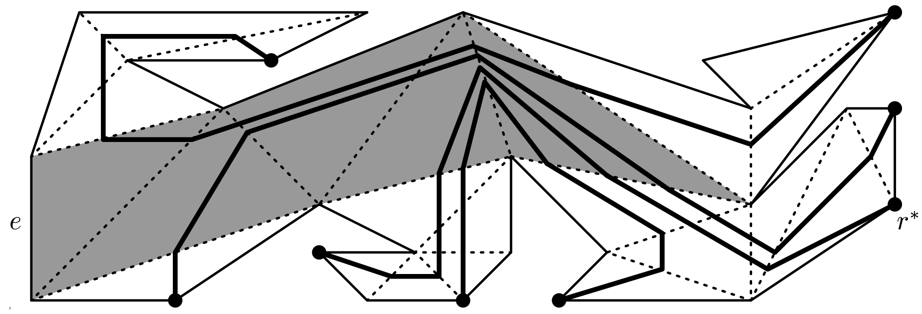

Figure 8: Paths connecting vertices.

connecting these paths with a line segment. If s and s' happen to lie on the same triangle, then this construction connects s to s' by a single line segment.

Let $\hat{\Pi}$ be the set of $2k$ pairs (s, e_s) where $(s, s') \in \Pi$ (or $(s', s) \in \Pi$.) Run the algorithm to connect the pairs of vertices and edges in $\hat{\Pi}$. Finally, connect the paths ζ_s and $\zeta_{s'}$ by a line segment. (See Figure 8.)

Since ζ_s has at most $2l(e_s, s) + 2\lfloor \log_2 2k \rfloor + 2$ links and $\zeta_{s'}$ has at most and $2l(e_{s'}, s') + 2\lfloor \log_2 2k \rfloor + 2$ links, the path from s to s' has at most $2l(e_s, s) + 2l(e_{s'}, s') + 4\lfloor \log_2 2k \rfloor + 5$ links. Since $l(s, s') + 1 \geq l(e_s, s) + l(e_{s'}, s')$, the path from s to s' has at most $2l(s, s') + 4\lfloor \log_2 2k \rfloor + 7$ or $2l(s, s') + 4\lfloor \log_2 k \rfloor + 11$ links. Summing this bound over all k paths give a bound of $2L + 4k\lfloor \log_2 k \rfloor + 11k$ on the total number of links. $\Box$

The *interior link diameter* δ of a polygon P is the maximum interior link distance between any two points in P. The following corollary follows directly from Lemma 2.2.

Corollary 2.1 *Let P be a simple polygon on n vertices of interior link diameter δ and let $\Pi = \{(s, s')\}$ be an untangled set of k pairs of distinct vertices of P. A set of k pairwise disjoint interior paths connecting s to s' for each $(s, s') \in \Pi$ can be constructed in $O(n\delta + k \log k)$ time using $2n\delta + 4k\lfloor \log_2 k \rfloor + 11k$ total links.*

The algorithms described in Lemma 2.2, Lemma 2.1 and Corollary 2.1 use coordinates whose precision is a fixed constant times the precision of the input. However, Suri's paper [10] on constructing minimum link paths assumes the less restrictive real RAM model of computation. The previous results can be improved under the real RAM model by constructing Γ directly from $\hat{\text{Vis}}(e)$

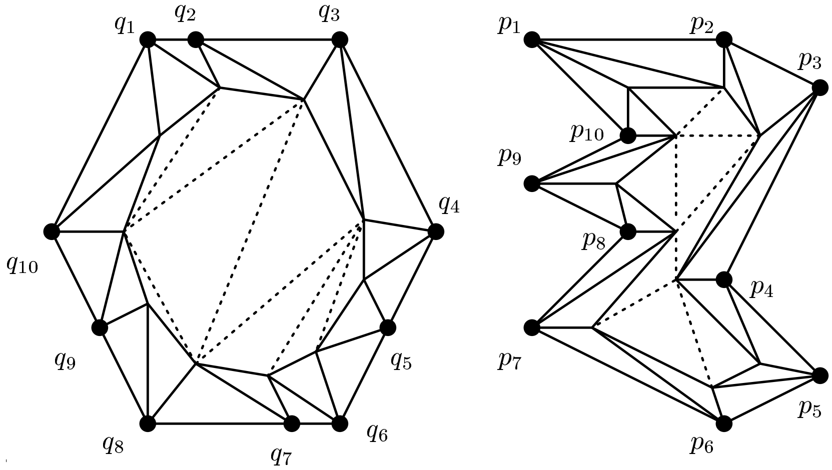

Figure 9: Isomorphic Triangulations

instead of from the triangles of T_P intersected by $\hat{\mathrm{V}}\mathrm{is}(e)$. Doing so enables many paths to intersect Γ in only one line segment, instead of two. The modified algorithm reduces the $2L+4k\lfloor\log_2 k\rfloor+11k$ term in Lemma 2.2 to $L+2k\lfloor\log_2 k\rfloor+11k$. Similar reductions can also be made to Lemma 2.1 and Corollary 2.1.

3 Constructing Isomorphic Triangulations

As before, let P and Q be simple polygons with vertex sets $\{p_1, p_2, \ldots, p_n\}$ and $\{q_1, q_2, \ldots, q_n\}$, respectively, listed in clockwise order around the polygons. Aronov, Seidel and Souvaine in [1] observe that if one of the polygons, say Q, is convex, then any triangulation of P induces a corresponding triangulation of Q. Thus if Q is convex, then P and Q have isomorphic triangulations mapping p_i to q_i with $2n-3$ edges.

We need a slight generalization of the previous result to the case where not all the vertices, $\{q_1, q_2, \ldots, q_n\}$, of Q are extreme, i.e., q_i may be collinear with q_{i-1} and q_{i+1}. (See Figure 9.) In this case, a diagonal from q_{i-1} to q_{i+1} would create an unacceptable degenerate triangle q_{i-1}, q_i, q_{i+1}. To avoid this, triangulate Q and perturb the vertices of Q toward the interior of Q so that they form the extreme points on a convex polygon Q'. Similarly, triangulate P and perturb the vertices of P toward the interior of P so that they form a shrunken version P' of P. Discard the original triangulations of P and Q. Connect the perturbed vertices in P and Q in cyclic order and connect each perturbed point on P' and Q' to its original version on P and Q. Triangulate each of the quadrilaterals between P and P' and Q and Q' in corresponding manner. Finally, triangulate P' and construct the corresponding triangulation of Q'. The total number of new edges added is $4n-3$. Note that this procedure does not add any new vertices to the boundary of P and Q.

Triangulating P, P' and Q takes only linear time [2]. Perturbing the vertices and connecting them also takes linear time. We have shown:

Lemma 3.3 *Let P be a simple polygon with vertices $\{p_1, p_2, \ldots, p_n\}$ and Q be a convex polygon with vertices $\{q_1, q_2, \ldots, q_n\}$, not all of which are necessarily extreme. Isomorphic triangulations mapping p_i to q_i using $5n-3$ edges can be constructed in linear time. Moreover, no new vertices are added to the boundary of P and Q in these triangulations.*

Combining Lemma 3.3 and Corollary 2.1 gives an algorithm for constructing isomorphic triangulations between a simple polygon and a polygon with interior link diameter δ.

Lemma 3.4 *Let P be a simple polygon with vertices $\{p_1, p_2, \ldots, p_n\}$ and Q be a simple polygon with vertices $\{q_1, q_2, \ldots, q_n\}$ and interior link diameter δ. Isomorphic triangulations mapping p_i to q_i using at most $20n\delta + 40n\lfloor \log_2 n \rfloor + 115n$ edges can be constructed in $O(n\delta + n \log n)$ time. Moreover, no new vertices are added to the boundary of P and Q in these triangulations.*

Proof: Triangulate P using $n-3$ diagonals. For each diagonal (p_i, p_j) there is a corresponding pair of vertices (q_i, q_j) in Q. By Corollary 2.1, $n-3 \leq n$ non-intersecting interior paths connecting each pair (q_i, q_j) can be constructed in $O(n\delta + n \log n)$ time using a total of at most $2n\delta + 4n\lceil \log_2 n \rceil + 11n$ links. For each new vertex q' on the path from q_i to q_j create a corresponding new vertex p' on diagonal (p_i, p_j).

The triangulation diagonals split polygon P into triangles $P_1, P_2, \ldots, P_{n-2}$. The interior paths split polygon Q into corresponding subpolygons $Q_1, Q_2, \ldots, Q_{n-2}$. By Lemma 3.3, isomorphic triangulations of P_i and Q_i can be constructed in $O(n_i)$ time using $5n_i - 3 \leq 5n_i$ edges where n_i is the number of vertices of P_i and Q_i. (Note that P_i may contain many new vertices p' which are not extreme.) Since the isomorphic triangulations of Lemma 3.3 do not contain any new boundary vertices, their union is an isomorphic triangulation of P and Q.

The total number of edges in the isomorphic triangulation of P and Q is bounded by $\sum_{i=1}^{n-2} 5n_i$. Since each edge of Q_i is either an edge of Q or an edge on one of the $n-3 \leq n$ non-intersecting interior paths, $\sum_{i=1}^{n-2} n_i < n + 2(2n\delta + 4n\lfloor \log_2 n \rfloor + 11n)$. Thus the total number of edges is bounded by $5n + 20n\delta + 40n\lfloor \log_2 n \rfloor + 110n$. □

Note that a vertex of P or of Q in Lemma 3.4 could be collinear with its neighbors, i.e., p_i could be collinear with p_{i-1} and p_{i+1}.

We are now ready for our main result.

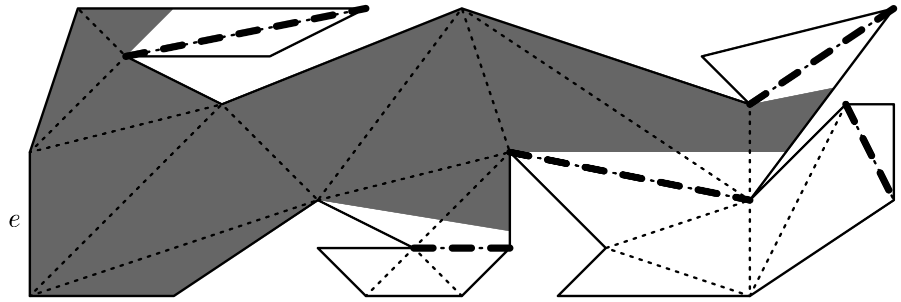

Figure 10: Choosing diagonals.

Theorem 3.1 *Let P be a simple polygon with vertices $\{p_1, p_2, \ldots, p_n\}$ and Q be a simple polygon with vertices $\{q_1, q_2, \ldots, q_n\}$. Let M be the minimum size of any isomorphic triangulation of P and Q mapping p_i to q_i. Isomorphic triangulations mapping p_i to q_i using $O(M \log n + n \log^2 n)$ edges can be constructed in $O(M \log n + n \log^2 n)$ time. Moreover, no new vertices are added to the boundary of P and Q in these triangulations.*

Proof: Construct a triangulation T_P of P using no vertices other than those of P. Choose a set of edges $\mathcal{C}$ from T_P as follows. Arbitrarily, pick an edge e of P. Let Ψ be the union of the set of triangles of T_P which intersect $\hat{\mathrm{V}}\mathrm{is}(e)$. The boundary of Ψ is composed of edges and chords of the original polygon P. Add all the chords bounding Ψ to $\mathcal{C}$. Each such chord c divides P into two polygons. Let $P(c)$ be one not containing Ψ. Recursively apply this procedure to each such polygon $P(c)$ starting at its edge c, to choose a set of edges from $P(c)$ and add them to $\mathcal{C}$. (See Figure 10.)

For each diagonal (p_i, p_j) in P there is a corresponding pair of vertices q_i, q_j in Q. Let k be the size of $\mathcal{C}$. By Lemma 2.2, a set of k non-intersecting interior paths connecting each pair q_i, q_j can be constructed in $O(n + L + k \log k)$ time using a total of at most $2L + 4k\lfloor \log_2 k \rfloor + 11k$ links, where L is the sum of the interior link distances between all pairs q_i, q_j. For each new vertex q' on the path from q_i to q_j create a corresponding new vertex p' on diagonal (p_i, p_j).

The k diagonals in $\mathcal{C}$ split polygon P into $k+1$ subpolygons $P_1, P_2, \ldots, P_{k+1}$. The interior paths split polygon Q into $k+1$ corresponding subpolygons Q_1, $Q_2, \ldots, Q_{k+1}$. Each subpolygon P_i has an edge e which lies within link distance two of every point in P_i. Thus each P_i has link diameter at most five. By Lemma 3.4, isomorphic triangulations of P_i and Q_i can be constructed in $O(n_i \log n_i)$ time using $40 n_i \lfloor \log_2 n_i \rfloor + O(n_i)$ edges where n_i is the number of vertices of P_i and Q_i. (Note that P_i may contain many new vertices p' which are not extreme.) Since the isomorphic triangulations of Lemma 3.4 do not contain any new boundary vertices, their union is an isomorphic triangulation of P and Q.

The total number of edges in the isomorphic triangulations of P and Q is at most $\sum_{i=1}^{k+1} 40n_i\lfloor\log_2 n_i\rfloor + O(n_i)$. Since each edge of Q_i is either an edge of Q or an edge on one of the k non-intersecting interior paths, $\sum_{i=1}^{k+1} n_i < n + 2(2L + 4k\lfloor\log_2 k\rfloor + 11k)$. Bounding $\log_2 n_i$ by $\log_2 n$ and k by n gives a bound of

$$40(23n + 4L + 8n\lfloor\log_2 n\rfloor)\lfloor\log_2 n\rfloor + O(n + L + n\log n) = O(L\log n + n\log^2 n).$$

Let T_P^* and T_Q^* be minimum size isomorphic triangulations of P and Q mapping p_i to q_i and let M be the sizes of T_P^* and T_Q^*. We now show that $L = O(M)$. By the choice of $\mathcal{C}$, no triangle of T_P^* can intersect more than three diagonals in $\mathcal{C}$. On the other hand, if l is the interior link distance from q_i to q_j, then any path from q_i to q_j must intersect at least l triangles from T_Q^*. Any path between the corresponding vertices $p_i, p_j \in P$, must also intersect at least l triangles from T_P^*. Thus the diagonals of $\mathcal{C}$ intersect at least $L/3$ different triangles in T_P^* where L is the sum of the interior link distances between all pairs q_i, q_j corresponding to diagonals in $\mathcal{C}$. Therefore, $M \geq L/3$ and $L\log n + n\log^2 n = O(M\log n + n\log^2 n)$.

Finally, we show that our algorithm runs in $O(M\log n + n\log^2 n)$ time. To construct $\mathcal{C}$ we need to construct polygons $\hat{\mathrm{V}}\mathrm{is}(c)$ in $P(c)$ for each $c \in \mathcal{C}$. Guibas et al. give a linear time algorithm for constructing the visibility polygon from a line segment [3]. This would result in an $O(n^2)$ algorithm for constructing $\mathcal{C}$. However, Suri shows in [10] that by only processing triangles which intersect the visibility polygon and their neighbors, a polygon can be broken into weakly visible polygons in $O(n)$ time. A similar analysis and technique will construct $\mathcal{C}$ in linear time.

Constructing the k paths takes a total of $O(n + L + k\log k)$ time by Lemma 2.2. Constructing isomorphic triangulations between the subpolygons P_i and Q_i takes $O(n_i\log n_i)$ time by Lemma 3.4. Since $\sum n_i = O(L + n\log n)$ and $L = O(M)$, the running time of the whole algorithm is bounded by $O(M\log n + n\log^2 n)$. $\square$

4 Conclusion

We have presented approximation algorithms for constructing pairwise disjoint paths and isomorphic triangulations in simple polygons. One open question is whether optimal solutions to these problems can be found in polynomial time. A less ambitious question is can solutions be found which are bounded by a constant instead of $\log n$ or $\log^2 n$ times the optimal?

Suri's paper in [10] on finding the minimum link distance between a pair of vertices assumes a real RAM model of computation. Can this problem be solved in linear time under a more realistic model of computation?

We can also consider the problem of constructing isomorphic triangulations between simple polygons where certain interior points are constrained to map to each other. Is there an approximation algorithm for finding such isomorphic triangulations which produces a result which is a constant or $\log n$ times the optimal solution?

References

[1] Aronov, B., Seidel, R., and Souvaine, D. On compatible triangulations of simple polygons. *Comput. Geom. Theory Appl. 3*, 1 (1993), 27–35.

[2] Chazelle, B. Triangulating a simple polygon in linear time. *Discrete Comput. Geom. 6* (1991), 485–524.

[3] Guibas, L. J., Hershberger, J., Leven, D., Sharir, M., and Tarjan, R. E. Linear-time algorithms for visibility and shortest path problems inside triangulated simple polygons. *Algorithmica 2* (1987), 209–233.

[4] Kent, J. R., Carlson, W. E., and Parent, R. E. Shape transformation for polyhedral objects. In *Computer Graphics Proceedings* (1992), pp. 47–64.

[5] Kranakis, E., and Urrutia, J. Isomorphic triangulations with small number of Steiner points. In *Proc. 7th Canad. Conf. Comput. Geom.* (1995), pp. 291–296.

[6] Pach, J., Shahrokhi, F., and Szegedy, M. Applications of the crossing number. In *Proc. 10th Annu. ACM Sympos. Comput. Geom.* (1994), pp. 198–202.

[7] Saalfeld, A. Joint triangulations and triangulation maps. In *Proc. 3rd Annu. ACM Sympos. Comput. Geom.* (1987), pp. 195–204.

[8] Shapira, M., and Rappoport, A. Shape blending using the star-skeleton representation. *IEEE Comput. Graph. Appl.* (1995), 44–50.

[9] Souvaine, D., and Wenger, R. Constructing piecewise linear homeomorphisms. Technical Report 94–52, DIMACS, New Brunswick, New Jersey, 1994.

[10] Suri, S. A linear time algorithm for minimum link paths inside a simple polygon. *Comput. Vision Graph. Image Process. 35* (1986), 99–110.

[11] THOMPSON, J., WARSI, Z., AND MASTIN, C. *Numerical Grid Generation: Foundations and Applications.* Elsevier Science, 1991.