Finding Optimal Geodesic Bridges Between Two Simple Polygons

Amit M. Bhosle* Teofilo F. Gonzalez†

Abstract

Given two simple polygons P and Q we study the problem of finding an optimal geodesic bridge. The objective is to find a bridge that minimizes the largest distance from any point in P to any point in Q. We present an algorithm that finds an optimal geodesic bridge (of minimum weight) in $O(n^2 \log n)$ time. Our algorithm uses as a subalgorithm a simpler $O(n^2 \log n)$ time algorithm that constructs an optimal geodesic bridge from a point to a polygon.

1 Introduction

We study the problem of finding optimal connections between two disjoint simple polygons. The two polygons may represent islands to be connected by a bridge, the goal is to identify a point on each of the two polygons as the end points of the bridge, such that the longest distance from any point on one island to any point on the other is minimized. In cases where it is possible to have flyover-like bridges, the bridge is a straight line between its two end points (immaterial of whether the two points are mutually visible or not). In other cases, the bridge may need to stay outside the interiors of the two polygons. E.g. if instead of a physical bridge, we intend to find a route for a ferry that connects the two islands, the route would need to stay within the water region that separates the two islands.

In this paper we present an algorithm for the geodesic-bridge problem that takes $O(n^2 \log n)$ time. We also present a simpler algorithm for the case when one of the polygons is a single point.

Let P and Q be two disjoint polygons. We use $\rho(X)$ to denote the compact region defined by polygon X, and use $\delta(X)$ to denote the boundary of X. Note that $\rho(X) \cap \delta(X) = \delta(X)$. Formally, for points $p \in \rho(P)$ and $q \in \rho(Q)$ we define the *weight* of the bridge (p, q) as $gd(p, P) + gd_e(p, q) + gd(q, Q)$ which is equal to

$$max_{p' \in \rho(P)}\{gd(p', p)\} + gd_e(p, q) + max_{q' \in \rho(Q)}\{gd(q, q')\}, \quad (1)$$

where $gd_e(p, q)$ denotes the length of the shortest geodesic path from $p \in \rho(P)$ to $q \in \rho(Q)$ that lies completely on the boundary and outside the polygons P and Q, and $gd(x, X)$ is the shortest geodesic distance between x and the geodesic furthest neighbor of x in the polygon X (i.e., $gd(x, X) = max_{x' \in \rho(X)}\{gd(x, x')\}$, where $gd(x, x')$ is the shortest geodesic distance between x and x' without leaving polygon X). A pair (p, q) that minimizes the above expression is called an *optimal geodesic bridge*. An *optimal Euclidean bridge* is defined similarly, but replacing in Eq. 1 $gd_e(p, q)$ by the Euclidean distance between points p and q, and was studied in Ref. [4]. We take the liberty of sometimes using $gd(p, q)$ to denote the actual path (instead of just its length) from p to q that defines the bridge.

2 Related Work

When the two polygons are convex, an optimal Euclidean bridge is also an optimal geodesic bridge. The problem was first studied by Cai, Xu and Zhu [5] who developed an $O(n^2 \log n)$ time algorithm. They proved that for this case the optimal bridge is between points on the boundary of the (convex) polygons which are visible from each other. Different linear time algorithms have been presented in Refs. [3, 9, 8]. The high-dimensional version of the problem has been studied [9, 11].

A 2-approximation algorithm, which finds a bridge with objective function value at most twice that of the optimal one, for convex polygons is given in Ref. [5]. Note that this approximation algorithm always generates a bridge whose endpoints are *mutually visible*. Ahn, Cheong, and Shin [1] present a $\sqrt{2}$-approximation algorithm for convex polygons and show that their technique generalizes to multidimensional space as long as P and Q are are convex regions.

Bhosle and Gonzalez [4] showed that the end points of an optimal Euclidean bridge might not be mutually visible when the polygons are not convex. They establish that an optimal Euclidean bridge always exists such that its endpoints lie on the boundaries of the two polygons. Using this critical property, they developed an algorithm to find an optimal Euclidean bridge in $O(n^2 \log n)$ time. For the case when one of the polygons degenerates to a point an optimal Euclidean bridge can be constructed in $O(n \log n)$ time [4].

Kim and Shin [8] developed an algorithm for the *v-bridge* problem where the bridges are restricted to have

*Currently at Amazon.com, 705 5^{th} Ave. S., Seattle, WA - 98144, bhosle@alumni.cs.ucsb.edu

†Department of Computer Science University of California Santa Barbara, CA 93106, teo@cs.ucsb.edu

their endpoints visible from each other. Note that the optimal v-bridge problem and the optimal bridge problem are identical when the polygons are convex; however, Ref. [4] shows that these problems defined over simple (non-convex) polygons have different solutions. This inequivalence holds even for rectilinear polygons [4]. Currently the fastest algorithm is by Tan [10], which runs in $O(n \log^3 n)$ time. This algorithm is quite complex and it makes substantial use of a hierarchical structure that consists of segment trees, range trees and persistent search trees, and a structure that supports dynamic ray shooting and shortest path queries. A restricted version, where the input polygons are simple, but *rectilinear* and the distance between points is measured by the Manhattan distance or L_1 distance, can be solved in linear time [12].

Kim and Shin [8] show that the approximation strategy given in Ref. [5] also applies to the v-bridge problem when the polygons are not convex. Kim and Shin [8] raise the question as to whether or not a better approximation algorithm exists for the Euclidean bridge and v-bridge problems. An optimal v-bridge has total weight within a factor of two times the weight of an optimal bridge between the two polygons. Furthermore, the bound of two is asymptotically best possible [4].

Bhosle and Gonzalez [4] developed approximation schemes that given any positive integer k construct a bridge with objective function value within a factor of $1 + \frac{2}{k+1}$ times that of the optimal one. The approximation algorithms apply to both the versions of the problem (Euclidean/geodesic bridges). It takes $O(kn \log kn)$ time for the Euclidean bridge problem and $O(k^2 n^2)$ for the geodesic bride problem. These approximation algorithms introduce k artificial vertices on each line segment and then find an optimal vertex bridge (both end points must be vertices of the polygons).

3 Point To Polygon Geodesic Bridges

First we identify a set of points on the boundary of the polygon which we call *anchors* and *pseudo anchors*. Then we show that an optimal geodesic bridge must have an endpoint that is a vertex, anchor or pseudo anchor of the polygon. Our algorithm uses this fact to narrow down the search for finding an optimal bridge. A similar approach has been used for the Euclidean bridge problem in [4], but the pseudo anchors for the geodesic bridge problem are different from the ones for the Euclidean bridge algorithm [4].

In Figure 1 the furthest neighbors of point q_1 inside Q_1 are points r and r'. The thick dashed line segments indicate the furthest geodesic paths from q_1 to r and the one from q_1 to r'. These paths are said to consist of a sequence of maximal line segments. As we traverse each of these paths starting at point q_1 the first vertex of the polygon that we visit (after q_1) is called the *first-vertex* of the corresponding furthest point geodesic path. The line segment from q_1 to the first-vertex is called the *first link*. In the polygon Q_1 in Figure 1, the vertices a and a' are the first-vertices, and the line segments (q_1, a) and (q_1, a') are called the corresponding *first-links*.

A point q on the boundary of Q is called an *anchor* if it is not a vertex of polygon Q, and there are at least two different vertices that are the first-vertex of geodesic furthest paths for q. In Figure 1 both $q_1 \in Q_1$ and $q_2 \in Q_2$ are anchors. A point p located immediately to the left of point q_1 (q_2) has line segment (p, q_1) ((p, q_2)) as its geodesic bridge. Note that point $q_3 \in Q_3$ is not an anchor point because all the geodesic furthest paths for point q have the same first-vertex, which is vertex a. A point p located immediately to the left of point q_3 will not have its geodesic bridge being the line segment (p, q_3). Figure 2 illustrates an instance of the problem of finding an optimal geodesic bridge from a point p to a polygon Q (defined by the solid lines). The geodesic furthest point of both q_5 and q_6 is q_4. The first-vertex of the point q_5 is q_2, and that of q_6 is q_7.

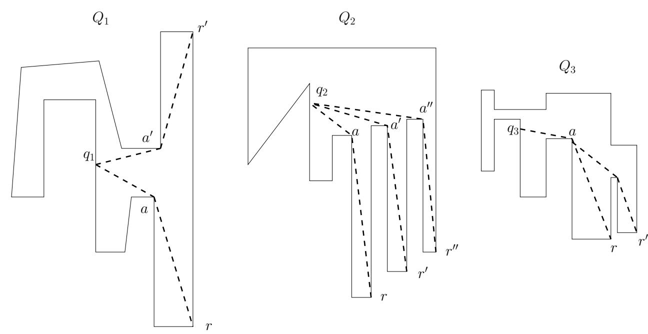

Figure 1: Points q_1 in Q_1 and q_2 in Q_2 are anchors, but point q_3 in Q_3 is not an anchor.

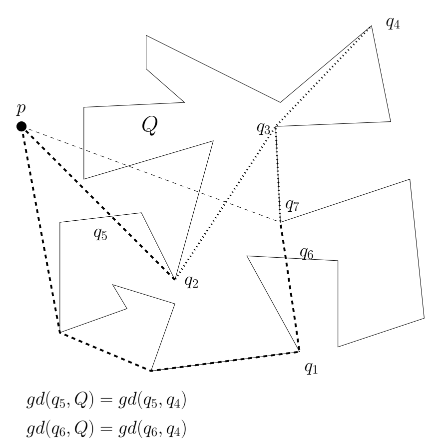

Figure 2: Geodesic Bridge from a Point to a Simple Polygon

In addition to any of the vertices and anchors of Q, points like q_5 and q_6 (see Figure 2) can also be the end point at Q of an optimal geodesic bridge. Points of the type q_5 are similar to the *pseudo-anchors* defined for the Euclidean bridge problem in Ref. [4], and are defined as: the first point of intersection of the line connecting point p to a vertex or anchor y of Q with the boundary of Q. Note that though *similar* to the definition of the pseudo-anchors for the Euclidean bridge problem, such pseudo-anchors for the geodesic bridge problem differ in that when traversing the line (p, y) from p to y, in the geodesic bridge cases, the *first* point of intersection with Q is a pseudo-anchor of Q where as in the Euclidean bridge version, the last intersection point with Q was selected as a pseudo-anchor. Actually, if the line (p, y) intersects the polygon Q multiple times, then we can show that none of the intersection points can support an optimal geodesic bridge. This is because a pseudo-anchor of Q can support an optimal bridge *only if* the vertex (y) of Q that *induces* the pseudo-anchor is its *first vertex* (the first vertex of Q on the path to its geodesic furthest vertex of Q). Otherwise, a better bridge is possible by moving the pseudo-anchor by a small distance along its edge. E.g. In Figure 2, the line (p, q_7) would intersect Q thrice. But q_7 cannot be the *first vertex* of the first intersection point (because they are not *mutually visible*). Among the other intersection points, q_7 can be the first vertex of only the last intersection point. However, by arguments similar to those in the proof of Theorem 2 in Ref. [4], one can show that a geodesic bridge ending at such points can be improved by moving the point by a slight distance in an appropriate direction along the edge. Although we do not need to include the pseudo-anchors generated by these cases (multiple points of intersection with the polygon), we keep them. This avoids the overhead of deleting them, while keeping the asymptotic size of the set of pseudo-anchors the same ($O(n^2)$).

Points of the type q_6 had no significance in the Euclidean bride problem, but they are important for the geodesic bridge problem. Also, these points are independent of the point p, and are defined solely by the polygon Q. Below we formally define these two types of pseudo-anchors.

Definition 1. External Pseudo Anchors: A point q on the boundary of Q that is not a vertex of the polygon nor an anchor point is called an *external pseudo-anchor* (induced by the point p) of Q if there is a vertex or anchor y in Q such that q lies on the line (p, y), and it is the first point on the boundary of Q hit by a ray originating at p in the direction (p, y). In other words, q is the point closest to p among all intersection points between (p, y) and Q.

Definition 2. Internal Pseudo Anchors: A point q on the boundary of Q that is not a vertex of the polygon nor an anchor point is called an *internal pseudo-anchor* of Q if there is a vertex x in Q and a vertex or anchor y, also in Q, such that q lies on the line (x, y), and it is the first point on the boundary of Q hit by a ray originating at x in the direction (x, y). In other words, q is the point closest to x among all intersection points between (x, y) and Q.

For the geodesic bridge problem, the set of pseudo-anchors includes external and internal pseudo anchors. In Figure 2, point q_5 is an external pseudo anchor, while point q_6 is an internal pseudo anchor. It is easy to see that there are $O(n^2)$ internal pseudo-anchors that can be computed in $O(n^2 \log n)$ using ray-shooting techniques as discussed in Ref. [4]. Similarly, the $O(n)$ external pseudo-anchors can be computed in $O(n \log n)$ time.

Theorem 1 *There is an optimal geodesic bridge from point p to polygon Q whose end point on Q is either a vertex, anchor, or pseudo-anchor from Q.*

Proof. Essentially, one can show that if the end point q of an optimal geodesic bridge on polygon Q is not a vertex, anchor or pseudo-anchor of Q, then the bridge obtained by moving q slightly along the edge (in an appropriate direction) has smaller weight than the assumed optimal bridge (a contradiction). The same proof technique is used in the proof for Theorem 2 in Ref. [4] for the Euclidean bridge problem, but the characterization of the set of candidate points for an optimal bridge differs significantly. □

Corollary 2 *An optimal geodesic bridge from a point p to a simple polygon Q can be computed in $O(n^2 \log n)$ time.*

Proof. The proof follows from the fact that the anchors and pseudo-anchors of Q can be found in $O(n^2 \log n)$ time. Furthermore, using the geodesic furthest site Voronoi diagram reported in Ref. [2], the geodesic furthest point for each vertex, anchor or pseudo-anchor can be found in $O(\log n)$ query time per point. The algorithm first constructs the shortest paths tree of the point p to the set of $O(n^2)$ vertices, anchors and pseudo-anchors. The algorithm reported in Ref. [6] can be used to build this shortest paths tree in $O(n^2 \log n)$ time. For each candidate bridge end point q that is either a vertex, anchor or pseudo-anchor of Q, the algorithm computes the weight of the bridge as $gd_e(p, q, Q) + gd(q, Q)$, where $gd_e(p, q, Q)$ denotes the weight of the geodesic shortest path from point p to point q in the presence of polygon Q as an *obstacle*, and $gd(q, Q)$ denotes the distance from q to its geodesic furthest vertex in Q. A candidate end point which minimizes the bridge weight is selected as the final solution. □

Algorithm *Geodesic-Bridge(p,Q)* outlines in detail our procedure.

```
Procedure Geodesic-Bridge(p,Q): point p and
```

```
simple polygon Q
```

Find all the anchors and pseudo anchors in Q;
Compute $gd(q, Q)$, the length of a geodesic furthest path in Q for each point q that is a vertex, anchor or pseudo-anchor in Q using the algorithm in [2];
Construct the shortest path tree rooted at p to the set of vertices, anchors and pseudo anchors of Q using the algorithm in Ref. [6];
From the tree of shortest paths rooted at p compute the geodesic distance $gd_e(p, u, Q)$ from p to each point u that is a vertex, anchor or pseudo anchor in Q in the presence of obstacle Q;
for every vertex u that is a vertex, anchor or pseudo anchor of Q *do*
Compute the length of the best geodesic bridge with an endpoint at u (endpoint u has the minimum value for $gd_e(p, u, Q) + gd(q, Q)$);
endfor
Return an optimal geodesic bridge;

```
End Procedure Geodesic-Bridge
```

3.1 Point To Polygon: Additional Bridge Properties

We now discuss some additional properties of the point-polygon version of the geodesic bridge problem. Though these properties do not result in any improvement to the point-polygon version, they are important for the two-polygon version of the problem.

Let (p, q) define an optimal geodesic bridge from point p to the polygon Q. On the geodesic path from p to q, $gd_e(p, q)$, let the *second last* vertex of $gd_e(p, q)$ be q^* (we say that the bridge *starts* at p). I.e., (q^*, q) is the *last edge* of the geodesic path $gd_e(p, q)$. Note that in some cases, the point q^* may be the same as the point p. In others, q^* will be a vertex of the polygon P or the polygon Q. By subpath optimality, we know that (q^*, q) defines an optimal geodesic bridge from the point q^* to the polygon Q. Also, the points q^* and q are *mutually visible* (otherwise, since $gd_e(p, q)$ is a *geodesic* shortest path between p and q, q^* could not have been the second last vertex on the bridge). Bridges whose end points are mutually visible are called *visible* bridges. E.g., (q^*, q) is a visible bridge. Finally, note that q^* has to either be a vertex of Q or be the same as the point p. This follows from the fact that a geodesic shortest path *bends* only at vertices of Q, and not at arbitrary points in the plane.

Visible bridges between two simple polygons can be computed in $O(n \log^3 n)$ time using the algorithm reported by Tan in Ref. [10]. The same algorithm can be used for the point to polygon problem when one of the polygons degenerates to a point. However, we use a simpler algorithm for computing optimal visible bridges from a point to a polygon in $O(n \log n)$ time. Let us now outline the algorithm.

The algorithm proceeds by computing the $O(n)$ anchors of the polygon Q. As discussed in Ref. [4], the anchor points of Q can be identified from the geodesic furthest site Voronoi diagram which can be computed in $O(n \log n)$ time using the algorithm in Ref. [2]. For the visible bridge problem, we need to consider only the vertices and anchors of Q that are *visible* from p and the *external pseudo-anchors* induced by lines connecting the point p to the vertices and anchors of Q, all of which can be found in $O(n \log n)$ total time. Now our set of candidate bridge end points on the polygon Q contains only $O(n)$ points, and using the techniques described in the proof of Corollary 2, the optimal visible bridge can be found in $O(n \log n)$ time. Note that an optimal visible bridge may not be an optimal geodesic bridge.

The algorithm for finding an optimal visible bridge from a point to a polygon can be combined with the fact that (q^*, q) (here, q^* is the second last vertex of the optimal geodesic bridge (p, q)) is an optimal visible bridge to provide a new algorithm for finding an optimal geodesic bridge from a point to a simple polygon. The algorithm begins by precomputing a set of $n+1$ optimal visible bridges. Every visible bridge to polygon Q originate at the point p, and ends at a vertex of Q. Note that these $n + 1$ points form our set of candidate *second last* vertices of the bridges. Next, the shortest paths tree rooted at point p to all the vertices of Q is constructed using the algorithm of Ref. [7] or Ref. [6]. This shortest paths tree provides us with the geodesic distance from p to each candidate second last vertex of the bridge. Using the precomputed value of the optimal visible bridge from a candidate second last vertex q^* to the polygon Q, and the geodesic distance from p to q^*, we can determine the weight of the geodesic bridge which has q^* as its second last vertex. Finally, the point q^* which supports the cheapest geodesic bridge defines the optimal geodesic bridge. Note that the time complexity of this algorithm is $O(n^2 \log n)$, which is the same as the previous one (Corollary 2). The time complexity is dominated by the time required to precompute the optimal visible bridges from all the candidate second last vertices to Q.

4 Geodesic Bridge Between Two Simple Polygons

We now discuss the more general version of the problem which asks to find an optimal geodesic bridge between two simple polygons.

Figure 3 illustrates an instance of the problem of finding a geodesic bridge between two simple polygons. The figure shows two geodesic bridges, (p_1, q_5) and (p_4, q_6), between the polygons P and Q. In this figure, p_4 and q_6 are internal pseudo-anchors of P and Q respectively, while q_5 is an external pseudo-anchor of Q.

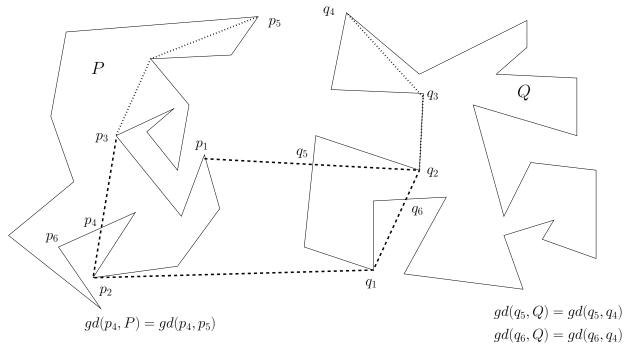

Figure 3: Geodesic Bridge Between Two Simple Polygons

We define the anchors and internal pseudo-anchors in the same way as for the point to polygon version of the problem. As in the Euclidean bridge case, the set of external psuedo anchors now has $O(n^2)$ points as follows.

Definition 3. External Pseudo Anchors: A point p on the boundary of P that is not a vertex of the polygon nor an anchor point is called an *external pseudo-anchor* of P if there is a vertex x in P and a vertex or anchor y in Q such that p lies on the line (x, y), and it is the first point on the boundary of P hit by a ray originating at y in the direction (y, x). In other words, p is the point closest to y among all intersection points between (x, y) and P.

We define the external pseudo anchors for Q in a similar way, and use the term *pseudo-anchors* to refer to the *union* of internal and external pseudo anchors.

Note that as in the case of the point to polygon geodesic bridge problem, if the line (x, y) intersects P multiple times, none of the intersection points can support an optimal geodesic bridge. However, to keep the algorithm simple, we include such points as well (moreover, the asymptotic size of the set of pseudo-anchors remains $O(n^2)$ even after eliminating them).

We now state the following theorem that limits the set of points on the boundaries of the two simple polygons P and Q that can possibly support an optimal geodesic bridge.

Theorem 3 *There is an optimal geodesic bridge whose end points are vertices, anchors, or pseudo-anchors from P and Q.*

Proof. The proof is a generalization of the proof of Theorem 1. □

The above theorem directly implies an $\tilde{O}(n^4)$ time algorithm for the optimal geodesic bridge[1]. The algorithm begins by building the geodesic furthest-site Voronoi diagram for the polygons P and Q. For each vertex, anchor or pseudo-anchor of P and Q, find their geodesic furthest neighbors in their respective polygons. Next, for each vertex, anchor and pseudo-anchor of P, build the shortest paths tree to the $O(n^2)$ vertices, anchors and pseudo-anchors of Q. Finally, for a candidate pair (p, q), find the weight of the bridge with p and q as the end points as $gd(p, P) + gd_e(p, q, P, Q) + gd(q, Q)$, where $gd_e(p, q, P, Q)$ denotes the geodesic distance between p and q in presence of polygonal obstacles P and Q, and select the pair with the minimum weight. It is easy to verify that this algorithm runs in $\tilde{O}(n^4)$ time.

[1] $\tilde{O}(f(n))$, where $f(n)$ is a polynomial function in n, is used to denote $O(f(n) \cdot polylog(n))$

4.1 Algorithm for Finding an Optimal Geodesic Bridge between Two Simple Polygons

Before we discuss our efficient algorithm for finding an optimal geodesic bridge connecting two simple polygons P and Q in $O(n^2 \log n)$ time, we establish an important property of optimal geodesic bridges.

Lets go back to the properties discussed in Section 3.1 for the *second last* vertices of the bridge. Let p^* and q^* be the second and second last vertices on the geodesic bridge defined by points $p \in P$ and $q \in Q$, when traversing the path $gd_e(p, q)$ from the point p to the point q. By previous arguments, (p^*, p) is an optimal visible bridge from the point p^* to the polygon P, and (q^*, q) is an optimal visible bridge from the point q^* to Q. Note that in some cases, the points p^* and q^* may overlap with the points p and/or q. Also, it may be possible for the the point p^* (resp. q^*) to lie on the polygon Q (resp. P). In general, at least one of the two points p^* and q^* is a vertex. In the only case when neither of these is a vertex, the optimal geodesic bridge (p, q) is in fact a *visible* bridge. The algorithm by Tan [10] finds an optimal visible bridge in $O(n \log^3 n)$ time.

The algorithm first precomputes an optimal visible bridge from each vertex in P and Q to both the polygons P and Q. If r is a vertex of P or Q, let $\nu(r, X)$ denote an optimal visible bridge from r to the polygon X, for $X \in \{P, Q\}$. When computing a visible bridge from r to the polygon X, we consider only the pseudo-anchors *induced* by r and the vertices and anchors of X. Furthermore, if a vertex or anchor of X is not visible from r (i.e. the line segment connecting r to the vertex or anchor of the polygon X intersects the other polygon before intersecting X), we ignore the pseudo-anchor for the simple reason that r cannot have a *visible* bridge in conjunction with this pseudo-anchor. E.g. In Figure 3, the line segment (p_6, q_2) intersects P before intersecting Q, and an optimal visible bridge from p_6 to Q cannot have this pseudo-anchor as the other end point of q_6's visible bridge. If none of the vertices and anchors of the polygon X are visible from r, $\nu(r, X)$ is considered to have a weight of ∞.

Every optimal geodesic bridge falls into one of the following three categories.

1. One-link bridges: Such bridges have a single edge in the geodesic shortest path between its end points.
2. Two-link bridges: Such bridges have two edges in the geodesic shortest path between its end points.
3. Multi-link bridges: Such bridges have more than two edges in the geodesic shortest path between its end points.

Clearly, one-link bridges are visible bridges, and an optimal visible bridge can be found in $O(n \log^3 n)$ time using Tan's algorithm [10].

In the case of two-link bridges, let v be the vertex of P or Q where the two edges of $gd_e(p,q)$ meet. Note that in such a case, $v = p^* = q^*$. By previous arguments, (v,p) and (v,q) define the optimal visible bridges from the point v to the polygons P and Q respectively. There can be at most $2n$ such bridges - one for each vertex of P and Q. Once we have precomputed the optimal visible bridges from each vertex of P and Q to both the polygons P and Q, the weights of these $2n$ bridges can be computed in constant time per vertex v. As discussed earlier, an optimal visible bridge from a point to a polygon can be computed in $O(n \log n)$ time. Consequently, the weights of all these $O(n)$ bridges can be computed in $O(n^2 \log n)$ time.

Finally, the multi-link bridges have two distinct p^* and q^* points which are respectively the second and second last vertices on $gd_e(p,q)$ when traversing the path $gd_e(p,q)$ from p to q. Also, in such bridges, both p^* and q^* are vertices of P or Q. Given the set of $2n$ vertices, we can compute the shortest geodesic paths between each of the $O(n^2)$ pairs of points in $O(n^2 \log n)$ time using the algorithm reported in Ref. [6]. For each pair of candidate points p^* and q^*, the optimal bridge with these two points as the second and second last vertices is the better of $\nu(p^*,P)+gd(p^*,q^*,P,Q)+\nu(q^*,Q)$ and $\nu(p^*,Q)+gd(p^*,q^*,P,Q)+\nu(q^*,P)$.

We state below the main theorem of this section.

Theorem 4 *Given two simple polygons, P and Q, an optimal geodesic bridge connecting the two polygons can be found in $O(n^2 \log n)$ time.*

Proof. Using the above arguments one can establish that there are $O(n^2)$ candidate bridges of the types *one-link*, *two-link* and *multi-link*. Furthermore, as we show above these candidate brides can be computed in $O(n^2 \log n)$ total time. Therefore, an optimal geodesic bridge connecting the two simple polygons P and Q is one of these candidate bridges which has the minimal total weight. □

Algorithm `Geodesic-Bridge(P,Q)`, is omitted as it follows from the above discussion.

5 Concluding Remarks

We have presented the first polynomial time algorithm for finding an optimal geodesic bridge connecting two simple polygons. The time bound of our algorithm, $O(n^2 \log n)$, matches that for the Euclidean bridge problem [4], though the algorithms are structurally different.

We conjecture that it would be relatively easier to improve the time bound for the Euclidean version of the problem than the geodesic version. We leave open the challenging question of improving the $O(n^2 \log n)$ time bound. Another interesting problem would be to design $o(n^2)$-time approximation algorithms for the geodesic bridge problem. Our algorithms also apply when there are a constant number of obstacles between the two polygons.

References

[1] H.-K. Ahn, O. Cheong, , and C.-S. Shin. Building bridges between convex regions. *Computational Geometry: Theory and Applications*, 25(1/2):161–170, 2003.

[2] B. Aronov, S. Fortune, and G. Wilfong. The furthest-site geodesic Voronoi diagram. *Discrete Comput. Geom.*, 9:217–255, 1993.

[3] B. Bhattacharya and R. Benkoczi. On computing the optimal bridge between two convex polygons. *Inform. Proc. Lett.*, 79(5):215–221, 2001.

[4] A. M. Bhosle and T. F. Gonzalez. Exact and approximation algorithms for finding an optimal bridge connecting two simple polygons. *IJCGA*, 15(6):609–630, 2005.

[5] L. Cai, Y. Xu, and B. Zhu. Computing the optimal bridge between two convex polygons. *Inform. Proc. Lett.*, 69:127–130, 1999.

[6] J. Hershberger and S. Suri. An optimal algorithm for euclidean shortest paths in the plane. *SIAM J. Comput.*, 28(6):2215–2256, 1999.

[7] S. Kapoor, S. N. Maheshwari, and J. S. B. Mitchell. An efficient algorithm for euclidean shortest paths among polygonal obstacles in the plane. *Discrete & Computational Geometry*, 18(4):377–383, 1997.

[8] S. K. Kim and C. S. Shin. Computing the optimal bridge between two polygons. *Theory of Computing Systems*, 34(4):337–352, 2001.

[9] X. Tan. On optimal bridges between two convex regions. *Inform. Proc. Lett.*, 76:163–168, 2000.

[10] X. Tan. Finding an optimal bridge between two polygons. *IJCGA*, 12(3):249–262, 2002.

[11] T. Tokuyama. Efficient algorithm for the minimum diameter bridge problem. *LNCS*, 2098:362–369, 2001.

[12] D. P. Wang. An optimal algorithm for constructing an optimal bridge between two simple rectilinear polygons. *Inform. Proc. Lett.*, 79:229–236, 2001.