Geodesic Spanners on Polyhedral Surfaces

Sanjiv Kapoor[1] and Xiang-Yang Li[1]*

Department of Computer Science, Illinois Institute of Technology, Chicago, IL, USA,
kapoor@iit.edu, xli@cs.iit.edu

Abstract. In this paper we consider the problem of efficiently constructing geodesic t-spanners. We consider finding spanners on the surface of a 3 dimensional polyhedron. If Steiner vertices are allowed on the surface of the polyhedron, then we are able to construct sparse t-spanners. If no Steiner vertices are allowed, then we establish lower bounds on the maximum node degree, depending on the spanning ratio t and also the total number of vertices of the polyhedron surface. We also consider the case of the surface of a convex polytope $\mathcal{P}$ with V vertices. Using its vertex set P and Steiner points, we can construct a t-spanner with a constant degree and weight $O(MST(U))$, where $MST(U)$ is the minimum spanning tree on the set U of vertices on convex polytope.

1 Introduction

Given an edge weighted graph $G = (V, E, w)$, where V is the set of vertices, E is the set of edges, and w is a weight function with $w(e)$ be the weight of an edge e, let $d_G(u, v)$ denote the shortest distance from node u to node v in graph G; let $\omega(G)$ be the sum of the edge weights of edges in G. A subgraph $H = (V, E', w)$ of G, where $E' \subseteq E$, is called a t-spanner of graph G, if for *any* pair of nodes u and v, $d_H(u, v) \leq t \cdot d_G(u, v)$. The minimum t such that H is a t-spanner of G is called the *stretch factor* of H with respect to G. An Euclidean graph is a graph where the weight of every edge (u, v) is the Euclidean distance $\|uv\|$ between its end-nodes. Given a geometric region Ω and a set V of points in Ω, a geodesic graph is a graph where the weight of each edge (u, v) with $u, v \in V$, is the *geodesic distance* from u to v in the region Ω.

For the case of geodesic spanners, our domain will be a 3-dimensional simplicial polygonal surface $\mathcal{P}$ that is formed of $m = O(n)$ triangles, and a set V of n nodes on the surface $\mathcal{P}$. While spanner construction has been well studied in general graphs and in Euclidean spaces, this is the first study of constructing geodesic spanners on a simplicial polygonal surface with some additional properties such as minimizing the node degree and/or total edge length. Notice that constructing a t-spanner for a polyhedral surface using Steiner points was implicitly studied in [28], in which Lanthier *et al.* proposed a method with time complexity $O(n^3 \log n)$. No additional properties like degree or weight bounds are considered in that paper. Here n is the number of vertices of the polyhedral surface. Another previous study of spanners involving geodesic distances in a 2-dimensional planar domain with obstacles can be found in [23]. Geodesic

* The research of authors is partially supported by NSF CNS-0832120, National Natural Science Foundation of China under Grant No. 60828003, Hong Kong CERG under Grant PolyU-5232/07E, and Hong Kong RGC HKUST 6169/07.

spanner graphs on the surface of the convex polyhedron $\mathcal{P}$ approximate the complete geodesic graph on a set of nodes, U on $\mathcal{P}$. The edges (u, v) have weight corresponding to the geodesic shortest distance between u and v. Note that a number of methods were proposed in the literature [2–4, 31, 40, 41, 43] to compute the exact, or to approximate the geodesic distance between two nodes. In order to construct a spanner, we note that distances on the surface of a polyhedron could be stretched due to the folds of the surface. To capture this effect we define *geodesic dilation factor* to measure the difference between the distance on the geodesic surface and the Euclidean distance between two points. We consider both non-convex and convex three dimensional polytopes (simplicial 2-complex) in this paper.

A greedy algorithm has been used to construct spanners for various graphs [11, 20–22, 36]. Peleg and Schaffer [34] showed that, for any $t > 1$, there exists a graph $G = (V, E)$ with $|V| = n$ and an edge weight, such that *any* t-spanner of this graph needs at least $n^{1+\frac{1}{t+2}}$ edges. Thus there are weighted graphs such that any t-spanner has weight at least $\Omega(n^{\frac{1}{t+2}}\omega(\mathrm{MST}))$ by letting the weight of each edge be 1. Althofer [5] proved that the greedy method produces a sparse t-spanner with at most $n^{1+\frac{2}{t-1}}$ edges; and Regev [37] showed that more precisely it has at most $n^{1+\frac{2}{\lfloor k \rfloor}} + n$ edges if $\lfloor k \rfloor$ is even, and at most $n^{1+\frac{2}{\lceil k \rceil}} + n$ edges if $\lfloor k \rfloor$ is odd. For arbitrary weighted graphs, Chandra *et al.* [12] showed that the greedy algorithm constructs a t-spanner of weight at most $(3 + \frac{16t}{\epsilon^2})n^{\frac{2+\epsilon}{t-1-\epsilon}} \cdot \omega(\mathrm{MST})$ for every $t > 1$ and any $\epsilon > 0$. Regev [37] proved that the t-spanner constructed by the greedy algorithm has weight at most $2e^2 \ln n \cdot n^{\frac{2}{t-1}} \cdot \omega(\mathrm{MST})$ when $t \in [3, 2\log n + 1]$, and has weight at most $(1 + \frac{4\log^2 n + 2\log n}{t+1-\log n}) \cdot \omega(\mathrm{MST})$ when $t > 2\log n + 1$, by studying the girth of the constructed t-spanner.

For the problem of minimizing the weight of a t-spanner (MWSP) when given an arbitrary weighted graph G, it was shown in [17] that MWSP is hard to approximate within a factor $o(\log n)$ for any constant integer $t > 1$. Given an arbitrary weighted graph G, it is also NP-hard to find a t-spanner with the minimum size [19, 26], or even approximate the minimum size t-spanner within factor $O(2^{(1-\epsilon)\ln n})$ for any $\epsilon > 0$, and $t > 2$ [19]. Finding a t-spanner with minimum size is also NP-hard [44] for various special graphs under some conditions of t. The problem of finding a t-spanner with a minimum maximum node degree is also NP-hard [27]: the problem of finding a 2-spanner with minimum maximum node degree is at least as hard to approximate as set cover. Thus, there is no $O(\ln n)$-approximation algorithm for minimum degree t-spanner problem unless $NP \subset \mathrm{DTIME}(n^{\mathrm{polylog}(n)})$. A probabilistic method was presented in [27] to find a 2-spanner whose degree is at most $\tilde{O}(\Delta^{1/4})$ times of the optimum, where Δ is the maximum degree of original graph G.

Constructing t-spanners [1, 6–10, 12, 14–16, 21–24, 32, 33, 35, 38, 39, 42] for Euclidean graphs has been extensively studied in the literature. Arya and Smid [8] first proposed a $o(n^2)$ time method (with time complexity $O(n \log^d n)$ for nodes in d-dimension) to construct a bounded-degree t-spanner with total edge weight $O(\omega(\mathrm{EMST}))$. An $O(n \log n)$ algorithm which uses an algebraic model together with indirect addressing has been obtained in [22].

For graphs that are neither general weighted graphs nor Euclidean graphs, several results were also proposed for constructing t-spanners. Das [13] presented a method to construct a t-spanner with a bounded node degree for visibility graph, defined by 2-dimensional obstacles, in time $O(n \log n)$ without constructing the visibility graph first. A number of results (centralized methods or distributed methods) were proposed in the literature [29, 30] to construct various t-spanners for unit disk graph.

Our Results: For geodesic spanners, we introduce geodesic cones and partition the space into geodesic cones. Using this space partition we get the following results:

We develop an algorithm for computing a spanner graph for a set of nodes $U \subseteq P$ on a 3-D polyhedral surface $\mathcal{P}$. Here P is the set of vertices of $\mathcal{P}$. We construct a sparse t-spanner with Steiner vertices from P with $O(\gamma(\mathcal{P})n/\epsilon)$ edges in time $O(n^2/\epsilon + n^3)$, where $n = |P|$ and $\epsilon > 0$ is any small constant. Since this is the first result for constructing t-spanners for geodesic graphs, we have not attempted to optimize the time complexity. Here $\gamma(\mathcal{P})$ is the dilation factor (defined later) of the polyhedron.

We also prove that, there is a polyhedral surface $\mathcal{P}$ and nodes' placement of U, such that the maximum node degree of any t-spanner, when no Steiner vertices are used in the spanner, is at least $\Omega(n^{1/t})$, for $t > 1$. When $t < 3$, we show by example that the maximum node degree of any t-spanner without using Steiner nodes for this example is at least $\gamma(\mathcal{P})$. Notice that in the worst case, $\gamma(\mathcal{P}) = \Theta(n)$. We also show that traditional greedy methods cannot build a t-spanner with degree bound $o(n)$, for any $t > 1$.

For a surface $\mathcal{P}$ of a convex polyhedron, by using Steiner vertices, we develop an algorithm to compute a t-spanner for a set of nodes U on the surface $\mathcal{P}$ with a constant maximum degree and weight $O(MST(U))$, where $MST(U)$ is the geodesic minimum spanning tree on the set U of nodes on convex polytopes, where the distance between two nodes is the geodesic distance between them.

The rest of the paper is organized as follows. We define some necessary terms and concepts, and then present our methods for constructing geodesic spanners for general polyhedron surfaces in section 2 and for convex polytopes in section 3. We conclude our paper in section 4.

2 Geodesic Spanners For Arbitrary Polyhedral Surfaces

2.1 Terminology

Assume that we are given a polyhedral surface $\mathcal{P}$ in three-dimension. Let P be the set of all vertices of $\mathcal{P}$ and $\mathcal{F}$ be the set of all faces of $\mathcal{P}$. For simplicity, we assume that all faces of $\mathcal{P}$ are triangular faces. Let U be a set of nodes on $\mathcal{P}$. In this paper, we always assume that $U \subseteq P$. For any two nodes $u, v \in U$, we let $\Pi_{\mathcal{P}}(u, v)$ be the shortest geodesic path on $\mathcal{P}$ between u and v; and let $\mathbf{d}_{\mathcal{P}}(u, v)$ denote the geodesic distance between u and v on $\mathcal{P}$. In general, given an edge weighted graph G and $u, v \in G$, let $\mathbf{d}_G(u, v)$ be the shortest distance between nodes u and v in G.

In this paper, we will focus on constructing spanners for the complete weighted graph $\mathcal{K}_{\mathcal{P}}(U, E)$ on the nodes in U with weight function, $\mathbf{d}_{\mathcal{P}} : E \to R^+$. A graph $H = (V, E)$, with $U \subseteq V$, is a geodesic t-spanner for U if $\mathbf{d}_H(u, v) \leq t \cdot \mathbf{d}_{\mathcal{P}}(u, v)$ for every pair of nodes $u, v \in U$. Here the weight of each edge xy in H is the geodesic distance $\mathbf{d}_{\mathcal{P}}(x, y)$ between x and y on the surface $\mathcal{P}$. The geodesic t-spanner H is said to use *Steiner vertices* if $U \subset V$. The geodesic t-spanner H is said to be on U (or without using Steiner vertices) if $V = U$. The geodesic t-spanner H is said to have size η if it has at most η nodes and edges. A *geodesic minimum spanning tree* of a set of vertices U on a surface $\mathcal{P}$ is the minimum spanning tree of graph $\mathcal{K}_{\mathcal{P}}(U, E)$.

Our algorithm to construct a geodesic t-spanner for $\mathcal{K}_{\mathcal{P}}(U, E)$ utilizes a new concept called *geodesic cones*. Given a point u and a region R, we let $P_u(R, d)$ be the set of all points p in region R that are at geodesic distance $\mathbf{d}_{\mathcal{P}}(u, p) = d$ from node u. For example, when R is a 2-D plane $\mathbb{R}^2$, then $P_u(R, d)$ is a circle centered at u with radius d. Geometric cones have been used widely to produce t-spanners in Euclidean space. An important property of the geometric cone $\mathcal{C}$, which was used in obtaining t-spanners, is that for any two points p_1, p_2 of $P_u(\mathcal{C}, d)$, $\|p_1 - p_2\| \leq \epsilon d$ for a small constant $0 < \epsilon < 1$. For example, for 2D geometric cones with angle θ, for any two points $p_1, p_2 \in P_u(\mathcal{C}, d)$, $\|p_1 - p_2\| \leq 2\sin(\frac{\theta}{2}) \cdot d$.

Definition 1 (pairwise-ϵ-neighbor property). *Given a surface $\mathcal{P}$ and the distance metric $\mathbf{d}_{\mathcal{P}}$, a set of points X is said to satisfy the pairwise-ϵ-neighbor property with respect to a node u if, for any two points $p_1, p_2 \in P_u(X, d)$, we have $\mathbf{d}_{\mathcal{P}}(p_1, p_2) \leq \epsilon \cdot d$.*

Given a surface $\mathcal{P}$, a set X of points on $\mathcal{P}$ is called an ϵ-geodesic cone *with respect to (w.r.t.) a node u, termed $\mathcal{C}(u)$, if the following two conditions are satisfied:*

1. *$P_u(X, d)$ satisfies the pairwise-ϵ-neighbor property with respect to a node u, and*
2. *if $x \in X$, then all points on a geodesic shortest path $\mathbf{d}_{\mathcal{P}}(u, x)$ are in X.*

Most of our results rely on constructing some special cone partition (or more precisely, cone covering) of space around every node in P. Notice that here a geodesic cone could be in a finite region. For any node u in the polyhedral surface $\mathcal{P}$, let $\mathcal{F}(u) = \{F_1(u), F_2(u), \cdots, F_p(u)\}$ be the p triangular faces incident onto node u. For each face $F_i(u) = \triangle xuy$ of $\mathcal{F}(u)$, let $\delta_i(u)$ be the radian value of the internal angle $\angle xuy$. Let $\delta(u) = \sum_{F_i(u) \in \mathcal{F}(u)} \delta_i(u)$.

Definition 2 (Dilation Factor). *The* dilation factor $\gamma(u)$ *of a node u in the polyhedral surface $\mathcal{P}$ is defined as $\gamma_{\mathcal{P}}(u) = \frac{\sum_i \delta_i(u)}{2\pi}$. We omit the subscript $\mathcal{P}$ if it is clear from the context. The* geodesic dilation factor, $\gamma(\mathcal{P})$, *of the polyhedral surface $\mathcal{P}$, is defined as $\gamma(\mathcal{P}) = \max_{u \in \mathcal{P}} \gamma(u)$.*

The dilation factor is a measure of the change in length of a “circle” on the surface of the polytope as compared to its length on a planar surface. When the surface $\mathcal{P}$ is planar, e.g. a single planar face, its geodesic dilation factor is $\gamma_{\mathcal{P}} = 1$. In general the dilation factor of a surface could be large and depends on the structure of the surface.

To determine geodesic shortest paths we need following definitions and properties:

1. *Edge-adjacent faces:* Two faces f_1 and f_2 are said to be *edge-adjacent* if they share a common edge, say e. $\mathcal{F} = \{f_1, f_2 \ldots f_k)$ is called a *face-sequence*, if for every $1 \leq i < k$, face f_i is edge-adjacent to face f_{i+1} with e_i as the edge common to f_i and f_{i+1}. The root of the face-sequence is the vertex in f_1 that is not an endpoint of e_1. It is referred to as $root(\mathcal{F})$.
2. *Planar Unfolding:* Suppose f and f' share an edge e. We define the planar unfolding of face f' onto face f as the image of f', denoted as $\mathrm{Img}_f(f')$, when f' is rotated about the line containing e onto the plane containing f such that $f \cap \mathrm{Img}_f(f') = e$. The planar unfolding of the sequence $\mathcal{F}$ is the planar unfolding

of faces $f_2, f_3 \ldots f_k$ sequentially onto the face f_1. We refer the unfolded image of an edge e, with respect to the planar unfolding of the sequence $\mathcal{F}$, as $U(\mathcal{F}, e)$.

3. *Geodesic paths:* A *path* on $\mathcal{P}$ is a simple geodesic path if it is simple and cannot be shortened by slight perturbations. Furthermore, a path Π connects an edge sequence E if Π comprises segments that join interior points of E.

We can state the following lemmas about geodesic paths:

Lemma 1. *[41] If Π is a geodesic path which connects the edge-sequence E then the unfolded image of Π along the edge sequence E is a straight line segment.*

Lemma 2. *[31] The general form of a geodesic path is a path which goes through an alternating sequence of vertices and edge sequences such that the unfolded image of the path along any edge sequence is a straight line segment and the angle of the path passing through a vertex is greater than or equal to π. The general form of an optimal path is the same as that of a geodesic path except that no edge can appear in more than one edge sequence and each edge sequence must be simple.*

Here *the angle of the path passing through a vertex* is defined as the smaller of the angles when measured through the surfaces on the left and right sides of the path.

2.2 Constructing t-Spanner Using Steiner Vertices

In this section, given the polyhedral surface $\mathcal{P}$ with a set of vertices P, a set of nodes $U \subseteq P$ and a parameter $t > 1$, we first present a method to construct a t-spanner $H = (V, E)$ using some Steiner vertices, *i.e.*, $U \subseteq V$. Notice that here we will focus on the case $V \subseteq P$, *i.e.*, Steiner vertices must be a subset of P, where P is the set of vertices used to define the polyhedral surface.

If we can use arbitrary Steiner nodes, we can easily get a t-spanner with the maximum degree bound 3 as follows. Let H' be the union of $|U|$ shortest path trees rooted at every node of U; for any node v, let v_1, v_2, $\cdots$, v_k be the k incident nodes of v in the union of shortest path trees. We then define a complete binary tree $T(v)$ with at least k leaf nodes and at most $2k$ leaf nodes, rooted at v, whose edge lengths are arbitrarily small. This tree $T(v)$ can be formed by at least $2k$ (and at most $4k$) Steiner vertices. We then replace the star of each node v in H', *i.e.*, $\{(v, v_i) \mid i \in [1, k]\}$ by $T(v) \cup \{(s_i, v_i) \mid i \in [1, k]\}$, where s_i is the ith leaf node of tree $T(v)$. Clearly, such modification will result a structure with degree at most 3, and it only increase the distance between any pair of nodes by an arbitrarily small value $\delta > 0$. Obviously, the total weight of the constructed spanner is almost optimum since the extra edges added in $T(v)$ have weight almost 0. In addition, the number of edges added is at most $4k$ for a node with degree k. Thus, the new spanner has edges $O(1)$ times the number of edges of H'. Obviously, the number of Steiner nodes used in this approach is at most $O(m)$, where m is the number of edges in the polytopes. Thus, in the rest of the paper, we always assume that the Steiner vertices in a t-spanner are restricted to the set P of vertices of the polytope.

Observe that when we compute the shortest geodesic path between a pair of nodes u and v, the path found, often uses multiple line segments from different faces of the surface $\mathcal{P}$. We would like to clarify that the end-points of these segments are *not* considered as Steiner nodes in this paper, although they are not from P.

Our method is to partition the space near every vertex $u \in P$ by some ϵ-geodesic cones. Consider a vertex $u \in \mathcal{P}$ and all the triangular faces, $\mathcal{F}_{\mathcal{P}}(u) = \{F_1(u), F_2(u), \cdots, F_p(u)\}$, where $F_i(u)$ is a triangle $v_i u v_{i+1}$, where v_{p+1} is v_1. We define $\lceil \delta(u)/\epsilon \rceil$ cones $\mathcal{C}_{\mathcal{P}}(u) = \{C_1(u), C_2(u), \cdots, C_{\lceil \delta(u)/\epsilon \rceil}(u)\}$, where each of the cones has an angle at most $\epsilon < \pi/3$, where $t(\epsilon) = \frac{1}{1-2\sin\frac{\epsilon}{2}}$ is the spanning ratio that can be achieved by the first phase of our method. For the set of faces $\mathcal{F}(u)$, imagine that we cut the faces $F_1(u)$ and $F_p(u)$ along the segment uv_1 and "unfold" all faces in $\mathcal{F}(u)$ sequentially on $F_1(u)$, using successively the edges $uv_i, i = 1 \ldots p-1$. To construct the required cones we desire to construct rays with apex u on the faces $\mathcal{F}(u)$ such that when we unfold these faces, the angle between any two consecutive rays on the unfolded plane is at most ϵ. Using the unfolding, the cones in $\mathcal{C}(u)$ are produced in the unfolded 2-d space by dividing the surrounding unfolded region $\delta(u)$ (which could have angle arbitrarily larger than 2π for non-convex polyhedron, and smaller than 2π for convex polyhedron) using planar cones (sectors) with an angle at most ϵ, *i.e.*, a cone, when unfolded, has a shape of a sector. Observe that here a cone $C_i(u)$ may contain several triangular faces of $\mathcal{F}(u)$ inside. We can then fold the faces back and this will give us $\lceil \delta(u)/\epsilon \rceil$ rays: two consecutive rays define a cone. It is easy to show that (1) for any point x from $\mathcal{F}(u)$, $\mathbf{d}_{\mathcal{P}}(x,u) = \mathbf{d}(x,u)$; (2) for any two points x and y from $\mathcal{F}(u)$ that fall inside the same cone and $\mathbf{d}_{\mathcal{P}}(x,u) < \mathbf{d}_{\mathcal{P}}(y,u)$, we have $\mathbf{d}_{\mathcal{P}}(x,u) + t(\epsilon) \cdot \mathbf{d}_{\mathcal{P}}(x,y) \leq t(\epsilon) \cdot \mathbf{d}_{\mathcal{P}}(y,u)$ for $t(\epsilon) = \frac{1}{1-2\sin\frac{\epsilon}{2}}$ when $\epsilon < \pi/3$.

Note that here a cone $C_i(u)$ only contains points from $\mathcal{F}(u)$ now. We later will show how to extend the cones to other triangular faces on $\mathcal{P}$ by propagating each cone.

We next present our method to construct a geodesic t-spanner, without using Steiner vertices, in phases:

1. **Phase 1:** $t(\epsilon_1)$**-Spanner Construction** First, for every node $u \in U$, we construct a geodesic cone partition, $\Lambda_{\mathcal{P}}(u)$. The cone partition is achieved by a propagation method which develops the cones starting with $\lceil \frac{\delta(u)}{\epsilon_1} \rceil$ cones $\mathcal{C}_{\mathcal{P}}(u)$ on the faces $\mathcal{F}(u)$ containing u. The process is detailed in procedure $\mathbf{PropagateCones}(u)$ for each node $u \in U$. The process is also repeated for Steiner vertices S of the polyhedron encountered during the cone expansion from nodes in U, *i.e.*, we run procedure $\mathbf{PropagateCones}(u)$ for every node u of S and update S accordingly: adding the encountered node $v \in P$ to S if $v \notin U$. These new vertices are termed Steiner vertices, denoted as S. Let U' be the union of U and S.
   For each ϵ-geodesic cone with apex node u, we add a (directed) edge uv if v is the closest node from P, *i.e.*, first encountered node in P. This phase results in a graph $H'(U \cup S, A')$ where A' is the set of edges added. We will prove that H' is a $t(\epsilon_1)$-spanner for U.
2. **Optional Phase 2: Further Degree Reduction** The process described in this paragraph is used to *possibly* further reduce the node degree. Again, we partition the space around each node v, by cones of angle at most ϵ_2.
   Repeat the following step for each node $v \in U \cup S$. In $H'(U \cup S, A')$, for each node v, let $I(v)$ be the set of incoming neighbors of node v. For all nodes in $I(v)$, build a tree rooted at node v. Let $I_0(v)$ be the set of nodes that already has been processed. Initially, $I_0(v) = \{v\}$. A directed edge (u,v), where $u \in I(v)$, is retained if u is the closest node in some cone of node v, and we add u to the set $I_0(v)$. For each

Algorithm 1 PropagateCones(u)

```
Let $\mathcal{C}_{\mathcal{P}}(u)$ be $\lceil \delta(u)/\epsilon_1 \rceil$ cones around $u$ on the faces $f \in \mathcal{F}(u)$ containing $u$ (each cone has an angle at most $\epsilon_1$).
**for** each cone $C$ in $\mathcal{C}_{\mathcal{P}}(u)$ **do**
    For each face $f = uvw$ inside the cone $C$, let $uv'w'$ be the portion of the face that is completely contained inside $C$. Notice that here $v'w'$ could be a segment of $vw$. Let $x(f)$ be the point on the segment $v'w'$ that is *closest* from $u$, *i.e.*, $\mathbf{d}_{\mathcal{P}}(x(f), u) = \mathbf{d}(x(f), u) \leq \mathbf{d}(y, u)$ for any point $y$ on the segment $v'w'$.
    Let $d = \min_{f \text{intersected by cone } C, f \in \mathcal{F}(u)} \mathbf{d}_{\mathcal{P}}(u, x(f))$. Let $f(C, u)$ (or simply $f$ if no confusion) be the face that has the point $x(f)$ such that $\mathbf{d}_{\mathcal{P}}(u, x(f))$ is minimized among all faces intersected by $C$, *i.e.*, $\mathbf{d}_{\mathcal{P}}(u, x(f(C, u))) = d$.
    **if** $x(f(C, u))$ is a vertex $v$ from $P$ **then**
        Add $v$ to $S$ and add a *directed* edge $(u, v)$ to the structure $H'(U \cup S, A')$.
        If $v$ is not marked *processed*, run procedure $\mathbf{PropagateCones(v)}$.
    **else**
        Let $e$ be the edge that contains the point $x(f(C, u))$ and $f$ be the face $f(C, u)$. Extend cone $C$ across face $f$, adjacent along $e$ and unfolded onto the sequence $F_U$ of unfolded faces by **procedure** *Unfold*. Note: In this procedure we keep unfolding faces until the closest point to $u$ among all points on segments defining polyhedral surface $\mathcal{P}$, say $x(C, d)$, is a vertex $z$ from $P$. Then add an edge $uz$ and update geodesic distance to $z$.
    **end if**
**end for**
Mark node $u$ processed.
```

newly added node u in $I_0(v)$, recursively add directed edges xu where $x \notin I_0(v)$ is the closest node to u in some cone of u.
Let the final structure be $H'(U \cup S, E')$.

3. **Optional Phase 3: Further Edge Reduction** In this phase, edges in $H'(U \cup S, A')$ (or $H'(U \cup S, E')$) are pruned to create the graph $H(U \cup S, E)$ as follows: (1) Sort the edges A' in $H'(U \cup S, A')$ in decreasing order of the geodesic length, $e_1, e_2 \ldots e_{m'}$. Let $E = A'$. (2) Eliminate edge e_i from E if the edges in $E \setminus e_i$ provide a path of length at most $t \cdot \mathbf{d}_{\mathcal{P}}(u, v)$ for every pair of nodes u and v from U, where $t > 1$ is the spanning ratio. (3) Eliminate unnecessary Steiner vertices, where a Steiner vertex is unnecessary if it is not on the shortest path in H between any pair of nodes in U.

The construction of the Geodesic Cone partition is given in Algorithm 1.

Observe that, in Algorithm 1, clearly point $x(f(C, u))$ has only two choices: either it is some node v from P defining the original polyhedral surface, or it is inside some segment vw where uvw is a face incident on u. Note that even if cone C contains multiple triangular faces of $\mathcal{F}(u)$ inside, it is still possible that $x(f(C, u))$ is not one of the vertices in faces $\mathcal{F}(u)$. See Figure 3 for illustration of these two cases.

Additionally, in Algorithm 1, the sequence F_U of unfolding is not necessarily unique, and we have to test all necessary sequences of unfolding. For the example illustrated by Figure 3 (a), after we unfold the triangle vwz, we should unfold both triangle vyz along the edge vz, and the triangle zwq along the edge zw to find the closest vertex from P that is inside the (extended) cone C. Here in the example illustrated by Figure 3 (a), either the vertex y or the vertex q, or both could be closer to u than the vertex z. The actual unfolding of faces will use the continuous Dijkstra method [31] as follows to define the **procedure** *Unfold*. For each cone C and each vertex u, we maintain an event heap: the event is the edge e of face f that has a point, denoted as $x(f, c)$, that is closest to the node u. When we unfold a face f represented by three vertices u, v, w along some edge, $e = (u, v)$, it will possibly introduce two new edges e_1 and e_2. We then add these two new edges to the heap based on the distance $\mathbf{d}_{\mathcal{P}}(e_i, u)$ to node u. We also add the distance to the vertex w. The top element of the heap is always the edge or vertex that is closest to the vertex u. Let e be the element in the top of the heap. We process the top event represented by e by unfolding the face adjacent to e as defined above. If the top element of the heap is a vertex v of P such that the distance $\mathbf{d}_{\mathcal{P}}(u, v) \leq \mathbf{d}_{\mathcal{P}}(u, e')$ for every element e' in the heap, the procedure returns v.

2.3 Performance Bounds and Time Complexity

See appendix for proof of the following lemmas.

Lemma 3. *The graph $H(U \cup S, E)$ is a t-spanner for nodes in U.*

Lemma 4. *The graph $H'(U \cup S, A')$ is a t-spanner with a maximum out-degree $O(\gamma(\mathcal{P})/\epsilon_1)$ where $t = \frac{1}{1-2\sin\frac{\epsilon_1}{2}}$.*

By choosing $\epsilon_1 = \epsilon_2 = \epsilon$ for some small value ϵ, we have the following lemma.

Lemma 5. *The graph $H'(U \cup S, E')$ after degree-reduction procedure is a t-spanner with a maximum node degree $O(\min(n, (\gamma(\mathcal{P})/\epsilon)^2))$ where $t = (\frac{1}{1-2\sin(\epsilon/2)})^2$.*

Lemma 6. *Our algorithm constructs $H(U \cup S, E)$ in $O(n^2/\epsilon + n^3)$ time.*

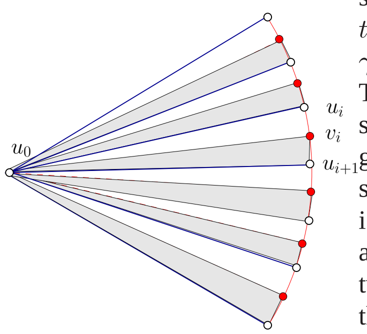


Fig. 1. An example of a surface and the set of nodes $U = \{u_0, u_1, \cdots, u_n\}$. Here u_0v_i defines a valley between u_0u_i and u_0u_{i+1}, $d = \mathbf{d}_{\mathcal{P}}(u_i, u_0)$.

We can construct an example surface and nodes placement (see Figure 1 for illustration) such that for some small enough t, a t-spanner (for some constant t) will have the maximum degree $\Omega(\gamma(\mathcal{P}))$ where $\gamma(\mathcal{P}) = \Theta(n)$ is the *dilation factor* of the surface. The basic idea of the example is as follows: There is a set U of n nodes $u_1, u_2, \cdots, u_n$. There are two triangular faces $u_iu_0v_i$ and $v_iu_0u_{i+1}$ between u_i and u_{i+1} such that the geodesic distance between u_i and u_{i+1} is larger than $(t-1) \cdot d + \eta$, for $t < 3$, $i \in [1, n-1]$, for a small constant $\eta > 0$. Actually, we can place these two triangular faces $u_iu_0v_i$ and $v_iu_0u_{i+1}$ such that the geodesic distance between u_i and u_{i+1} is $2d - \delta$ for any $0 \leq \delta < 2d$. A node u_0 at distance d from these nodes will then have to be connected directly to all these nodes to ensure that it is a t-spanner. Observe that, when $t \geq 3$, the preceding example does not imply that we have to connect u_0 with every node u_i, $i \geq 1$. This is because the geodesic distance between u_i and u_j is at most $\mathbf{d}_{\mathcal{P}}(u_i, u_0) + \mathbf{d}_{\mathcal{P}}(u_0, u_j) \leq 2d$. In this case, we can omit some edges u_0u_i without violating the t-spanner property for $t \geq 3$.

We further study reducing the weight of the structure. Chandra *et al.* [11] proved that for any metric space M, and every n-vertex complete graph G on this metric, if (1) there is an $O(g(n))$ time algorithm that builds a t-spanner for G with $O(f(n))$ edges, where $f(m)/2 \geq f(m/2)$ and $g(m)/2 \geq g(m/2)$ for any $m > 0$, and (2) there exists an $O(h(n))$ time algorithm that can build a spanning tree T for G with weight $O(1)\omega(\mathrm{MST})$, then there exists an $O(\max(g(n), h(n), n \log n))$ time method which builds a $(t+\epsilon)$-spanner with $O(f(n))$ edges and weight $O(\frac{f(n)}{n} \log n)\omega(\mathrm{MST})$. Notice that for geodesic metric, we have methods with $f(n) = O(\gamma(\mathcal{P})n)$ and $g(n) = O(n^3 + n^2/\epsilon)$. Our method for constructing a structure H' implies the following lemma.

Lemma 7. *We can construct a geodesic t-spanner for any polyhedral surface $\mathcal{P}$ such that the total weight of the structure is $O(\gamma(\mathcal{P}) \log n)\omega(\mathrm{MST})$ in time $g(n) = O(n^2/\epsilon)$.*

Observe that with the optional degree-reduction phase 2, the running time of the method becomes $g(n) = O(n^3 + n^2/\epsilon)$. Notice that the method by Chandra [11] cannot preserve the degree bound of the final structure. We leave it as a future work to design a t-spanner structure with bounded degree $O(\gamma(\mathcal{P}))$, and total weight $O(\gamma(\mathcal{P}) \log n)\omega(\mathrm{MST})$, or study whether it is possible to construct a t-spanner with weight $O(\gamma(\mathcal{P}) + \log n)\omega(\mathrm{MST})$.

2.4 Geodesic Spanners Without Using Steiner Vertices

We now study, given the polyhedral surface $\mathcal{P}$ (and its set of vertices P), a set of nodes $U \subseteq P$, and a number $t > 1$, how to construct a t-spanner $H = (U, E)$ without using Steiner vertices. Our objective is to construct a t-spanner with small node degree and small total edge weight. A more general question is following: given a complete weighted graph G with positive edge weights satisfying the triangular inequality, construct a t-spanner $H \subseteq G$ with small maximum degree and small total edge weight $\omega(H)$. Surprisingly, we could not find any results, except [27], in the literature that provide any degree bound on a t-spanner for an arbitrarily weighted graph.

We first show by example that, for any algorithm that constructs a t-spanner, there are inputs such that the constructed t-spanner will have a maximum degree at least $\Omega(n^{\frac{1}{t}})$ for nodes placed on a surface, where n is the size of U.

Lemma 8. *For any $t > 1$, there is a surface $\mathcal{P}$ on a set of nodes P, and a set of nodes $U \subseteq P$, such that the maximum node degree in any t-spanner $H = (U, E)$ without using Steiner vertices is at least $(\frac{n}{2})^{\frac{1}{t}}$, where $n = |U|$. For any $t > 1$, there is a surface $\mathcal{P}$ on a set of nodes P, and a set of nodes $U \subseteq P$, such that the weight of any t-spanner is at least $n^{\frac{1}{t+2}}/2$ times of MST.*

When $t < 3$, the placement the triangles $u_i u_0 v_i$, $1 \leq i \leq n-1$, and triangles $v_i u_0 u_{i+1}$, $1 \leq i \leq n-1$, ensures that the geodesic distance $\mathbf{d}_{\mathcal{P}}(u_i, u_{i+1})$ is $2\mathbf{d}_{\mathcal{P}}(u_0, u_i) - \delta$ (for small $0 < \delta < (3-t)\mathbf{d}_{\mathcal{P}}(u_0, u_i)$) and $\mathbf{d}_{\mathcal{P}}(u_0, u_i) = \mathbf{d}_{\mathcal{P}}(u_0, u_j)$ for $i \neq j$. Thus, we have to connect u_0 to every node u_i since $\mathbf{d}_{\mathcal{P}}(u_0, u_j) + \mathbf{d}_{\mathcal{P}}(u_j, u_i) \geq 3\mathbf{d}_{\mathcal{P}}(u_0, u_i) - \delta > t\mathbf{d}_{\mathcal{P}}(u_0, u_i)$ for every node u_j and $t < 3$.

Lemma 9. *For any t with $1 < t < 3$, there is a surface $\mathcal{P}$ on a set of nodes P, and a set of nodes $U \subseteq P$, such that the maximum node degree in any t-spanner $H = (U, E)$, without using Steiner vertices or all Steiner vertices are restricted to P, is at least $\gamma(\mathcal{P}) = \Theta(n)$, where $n = |U|$.*

Thus, generally, to get a t-spanner, which does not use any Steiner vertices or can only use Steiner vertices from P, with a maximum node degree $o(n)$, we must focus on $t \geq 3$. In this case, Lemma 8 shows that the maximum degree is at least $\Omega(n^{\frac{1}{t}})$.

Notice that the traditional greedy method (sorting edges in increasing order as $e_1, e_2, \cdots, e_m$ and adding an edge $e_i = (u, v)$ only if the added edges from $e_1, e_2, \cdots, e_{i-1}$ do not have a t-spanner path connecting u and v) will still produce a structure with maximum node degree n for the example illustrated in the proof of Lemma 8.

We also show that the following greedy method cannot produce a t-spanner with a bounded degree Δ at all. The method is as follows: Let A be the final added edges and $A = \emptyset$ initially; We sort edges in the complete graph on U in increasing order, and we add an edge $e_i = (u, v)$ to A only if (1) there is a t-spanner path connecting u and v using previously added edges A, and (2) the degrees of u and v in the partial graph formed by edges in A are at most $\Delta - 1$. We construct a network example as follows: U is formed of two sets U_1 and U_2, both have $\Delta + 1$ nodes and is a copy of the node placement illustrated by Figure 1. The distance between U_1 and U_2 is very large compared with the radius of U_1 and U_2. Then this simple greedy method will not build any edges between U_1 and U_2. Thus the final structure is disconnected.

Bounds on total edge weight: The preceding lemmas build lower bounds on the maximum node degree and total edge weight that we can achieve for an arbitrary input of polyhedral surface $\mathcal{P}$ and a set of nodes U. It is hard to approximate the minimum weight t-spanner (*i.e.*, finding a t-spanner with the minimum total edge weight) within a factor $o(\log n)$ for a general weighted graph [17] for any integer $t > 1$. Thus, it is hard to construct a t-spanner with total weight $o(\log n)\omega(\mathrm{MST})$.

A simple greedy algorithm can construct a t-spanner whose total edge weight is at most $O(\ln n \cdot n^{\frac{2}{t-1}}) \cdot \omega(\mathrm{MST})$ when $t \in [3, 2\log n + 1]$ [37]. This currently best known result is still far from the previous lower bound we knew: the ratio of the best known achievable upper bound on weight over the best known lower bound on weight is $\ln n \cdot n^{\frac{t+3}{(t-1)(t+2)}}$ for $t \in [3, 2\log n + 1]$. For arbitrary $t > 1$, results by Khuller [25] show that we can construct a t-spanner in time $O(n^2)$, when we already have the geodesic distances between every pair of nodes in U, with weight at most $n(1 + \frac{2}{t-1})\omega(MST)$. Observe that for arbitrary weighted graph, when $t < 2$, in the case of a complete graph where each edge has a weight 1, any t-spanner must be the complete graph itself. The spanner thus has total edge weight $n \cdot \omega(MST)$. Unfortunately, it is not clear how to design a geodesic graph such that the weight of each edge is 1.

3 Geodesic Spanners for Convex Polytopes

In this section, we study constructing geodesic spanners for a set of nodes on a convex polytope, and the distance is measured by geodesic distance. Our approach is to approximate a convex polytope by a constant number of 2D planar patches, similar to [18].

Let $\mathcal{P}$ be a convex polytope, with a set of polygonal faces $\mathcal{F}$. For any subset $F \subseteq \mathcal{F}$ of faces, let $\mathcal{N}(F) = \{N_f \mid f \in F\}$ be the set of normals to the faces where N_f is the normal to face f. Consider the angular representation of the normals: each normal N is represented by a pair (θ_N, ϕ_N) in the Spherical coordinates system, where $\theta_N, \phi_N \in [0, 2\pi]$ are the angle of the normal vector from the z-axis (called the colatitude or zenith) and the angle from the x-axis. The basic idea of our method for building the spanner is to partition the convex polygonal surface $\mathcal{P}$ into a constant number of convex patches such that each patch is almost flat (*i.e.*, the difference between the normals of any two faces in the patch is a small constant). Note that the patches we construct may overlap.

Definition 3. *A δ-patch of $\mathcal{P}$ is a set of faces, $F \subseteq \mathcal{F}$ such that (1) F forms a continuous region; (2) the patch is* flat, *i.e.,* $\varPsi(F) = \sup_{f,g \in F} \max\left(|\theta_{N_f} - \theta_{N_g}|, |\phi_{N_f} - \phi_{N_g}|\right) \leq \delta$*, i.e., the difference between any two normals is bounded by a constant.*

A δ-partition of $\mathcal{P}$, denoted as $\varDelta_{\mathcal{P}}$, is a partition of the set of faces $\mathcal{F}$ such that each partition is a δ-patch. Here a δ-patch is not necessarily convex.

Definition 4. *A δ-planar projection, $\varXi(F)$ of a δ-patch F is the projection of points in F onto a plane, P, with normal N_P such that $N_P \in \mathcal{N}(F)$.*

A *linear convex patch* G is a connected closed subset of points with a piecewise linear boundary such that its δ-planar projection $\varXi(G)$ is convex.

Definition 5. *A convex extension $\mathcal{E}(F)$ of a δ-patch F is a minimal piece-wise linear convex patch, a collection of polygonal faces, that contains F with the property that $\varPsi(\mathcal{E}(F)) - \varPsi(F) \leq \epsilon$.*

It is not difficult to show that the following property holds for a convex surface $\mathcal{P}$.

Property 1. Low-distortion projection property: Let u and v be two points on a δ-patch, F. Then $\mathbf{d}_{\mathcal{P}}(u, v) \geq d(u, v) \geq \mathbf{d}_{\mathcal{P}}(u, v)/(1 + 2 \cdot \delta)$ where $d(u, v)$ is the Euclidean distance between u and v on $\varXi(F)$ and $\mathbf{d}_{\mathcal{P}}(u, v)$ is the geodesic distance on F.

3.1 Algorithm

Our method for constructing a spanner for convex polytope is as follows:

1. Find a δ-partition, denoted as $\varDelta_{\mathcal{P}} = \{F_1, F_2, \cdots F_p\}$, of $\mathcal{P}$. Here $F_i \subseteq \mathcal{F}$ is a subset of faces that form a δ-patch. Then we construct a convex-extension $\mathcal{E}(\varDelta_{\mathcal{P}})$ as follows. For each δ-patch F_i in the δ-partition, we perform the following steps:
   (a) Find a δ-planar projection, $\varXi(F_i)$ of F_i to some plane with a normal $N \in N(F_i)$.
   (b) Find the convex hull $CH(\varXi(F_i))$ of $\varXi(F_i)$.

    (c) Find the inverse of the projection, i.e., find $\mathcal{E}(\Xi(F_i))$ such that its δ-planar projection is $CH(\Xi(F_i))$.

2. For every δ-patch in $\varDelta_{\mathcal{P}}$ do the following
    (a) For each $u \in U_F$, construct a ϵ-cone partition of the surface as in the previous section 2. Let $\mathcal{C}(u)$ be the cone partition produced. Note that since the difference between normals is small, a simple method of projecting cone partitions of a plane suffices in this case.
    (b) Let U_F be the set of all nodes in F For node $u \in F$, let $\mathcal{I}(u)$ be the set of the intersection segments of all cones with the boundary $\partial\mathcal{E}(F)$. In each intersection region $\mathbf{C} \cap \partial\mathcal{E}(F)$ where $\mathbf{C} \in \mathcal{C}(u)$, we add a Steiner point if the cone **C** that created the region contains a shortest path from the apex, say u, of the cone to some other node $v \notin U_F$, *i.e.*, outside of the δ-patch. Note that this Steiner point is also added to the neighboring δ-patch, which the cone **C** intersects. Let the set of added Steiner points be $\mathcal{S}_F(u)$. Let the set of all Steiner points on F be $\mathcal{S}_F = \cup_{u \in U_F} \mathcal{S}_F(u)$.
    (c) Find a projection $\Xi(F)$ of F to some hyperplane perpendicular to the normal of a face in F.
    (d) Find an Euclidean t-spanner graph $H_F(\Xi(V_F), \Xi(A_F))$ of a constant maximum degree and of weight $O(MST(V_F))$ for the set of vertices, $\Xi(V_F)$, in $\Xi(F)$ where $V_F = U_F \cup \mathcal{S}_F$. $\Xi(A_F)$ is the set of edges created in the spanner graph. This can be done by several methods in the literature [9].
3. Let the spanner be $H(U, A) = \cup_{F \in \varDelta_{\mathcal{P}}} H_F(V_F, A_F)$, where each edge (u, v) in A_F corresponds to an edge $(\Xi(u), \Xi(v))$ and is weighted by shortest geodesic distance between u and v.

It remains to determine a δ-partition and the convex extensions. Given the range of angles θ and ϕ we do the following

1. Partition the domain $[0, 2\pi]$ of θ and ϕ equally into $2\pi/\epsilon$ ranges of size ϵ, indexed by $(i, j), 1 \leq i, j \leq 2\pi/\epsilon$ which indicates that $\theta \in [(i-1) \cdot \epsilon, i \cdot \epsilon]$ and $\phi \in [(j-1) \cdot \epsilon, j \cdot \epsilon]$.
2. For every tuple of ranges (i, j) let $F = \cup f$ such that $\theta_{N_f} \in [(i-1) \cdot \epsilon, i \cdot \epsilon]$ and $\phi_{N_f} \in [(j-1) \cdot \epsilon, j \cdot \epsilon]$.

The preceding approach clearly creates a constant number of δ-patches because of the monotonicity of the normals for a convex polytope. The δ-patch F is obtained from the convex hull of points on F on the polyhedron $\mathcal{P}$.

Lemma 10. *The convex extension of a δ-patch F is also a δ-patch.*

Note that our spanner uses Steiner points.

Lemma 11. *$H(U, A)$ is a t-spanner of the set of nodes U on $\mathcal{P}$.*

We now compare the weight of the spanner with the minimum spanning tree $MST(U)$ of the set of points U on the surface of $\mathcal{P}$.

Lemma 12. *$H(U, A)$ has a constant maximum degree and weight $O(MST(U))$.*

Lemma 13. *The construction of the t-spanner $H(U, A)$ requires $O((1/\epsilon)^2(n^2 \log n + T_E(n)))$ steps where $T_E(n)$ is the time required by any algorithm to compute a 2-dimensional t-spanner.*

Notice that $T_E(n) = O(n \log n)$ for any d-dimensional nodes. We also observe that our method of constructing t-spanner also works for any d-dimensional convex surface (where the cone must be small enough with an angle $O(\epsilon)$ with $(1 + \epsilon)^d < t$).

4 Conclusion

In this paper, we studied the spanner construction for geodesic graphs for a set of nodes U placed on a 3-dimensional surface $\mathcal{P}$, and for geodesic graphs on convex polytopes. Our main contributions are

1. a polynomial-time algorithm to build a sparse t-spanner $H(U \cup S, E)$ with $O(\gamma(\mathcal{P})n)$ edges when we can use Steiner vertices. We also present some methods to reduce the node degree. For the case when the use of Steiner vertices is restricted, we show lower bounds on the maximum node degree of any t-spanner.
2. a polynomial-time algorithm that builds a bounded degree t-spanner with weight $O(\omega(MST))$ for geodesic graphs on convex polytopes. The spanner uses steiner vertices.

For general polygonal surfaces, it is interesting to design an efficient method to construct a sparse t-spanner without using Steiner vertices such that it can achieve a nontrivial degree bound when $t \geq 3$. Given a weighted graph with edge weights satisfying the triangular inequality and constant $t \geq 3$, we would like to construct a t-spanner whose maximum node degree is asymptotically minimum (in the order of $O(n^{1/t})$ or $O(n^{O(1/t)})$).

References

1. ABAM, M. A., DE BERG, M., FARSHI, M., AND GUDMUNDSSON, J. Region-fault tolerant geometric spanners. In *SODA '07: Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms* (Philadelphia, PA, USA, 2007), Society for Industrial and Applied Mathematics, pp. 1–10.
2. AGARWAL, P. K., HAR-PELED, S., SHARIR, M., AND VARADARAJAN, K. R. Approximating shortest paths on a convex polytope in three dimensions. *J. ACM 44*, 4 (1997), 567–584.
3. ALEKSANDROV, L., MAHESHWARI, A., AND SACK, J.-R. Approximation algorithms for geometric shortest path problems. In *STOC '00: Proceedings of the thirty-second annual ACM symposium on Theory of computing* (New York, NY, USA, 2000), ACM, pp. 286–295.
4. ALEKSANDROV, L., MAHESHWARI, A., AND SACK, J.-R. Determining approximate shortest paths on weighted polyhedral surfaces. *J. ACM 52*, 1 (2005), 25–53.
5. ALTHOFER, I., DAS, G., DOPKIN, D., JOSEPH, D., AND SOARES, J. On sparse spanners of weighted graphs. *Discrete and Computational Geometry 9* (1993), 81–100.
6. ARYA, S., DAS, G., MOUNT, D., SALOWE, J., AND SMID, M. Euclidean spanners: short, thin, and lanky. In *Proc. 27th ACM STOC* (1995), pp. 489–498.
7. ARYA, S., MOUNT, D. M., AND SMID, M. Dynamic algorithms for geometric spanners of small diameter: Randomized algorithms. *Computational Geometry Theory and Applications 13* (1999), 91–107.
8. ARYA, S., AND SMID, M. Efficient construction of a bounded degree spanner with low weight. In *Proc. 2nd Annu. European Sympos. Algorithms (ESA), volume 855 of Lecture Notes in Computer Science* (1994), pp. 48–59.

9. ARYA, S., AND SMID, M. Efficient construction of a bounded degree spanner with low weight. *Algorithmica 17* (1997), 33–54.
10. BOSE, P., GUDMUNDSSON, J., AND SMID, M. Constructing plane spanners of bounded degree and low weight. In *Proceedings of the European Symposium on Algorithms (ESA)* (2002).
11. CHANDRA, B., DAS, G., NARASIMHAN, G., AND SOARES, J. New sparseness results on graph spanners. In *Proc. 8th Annual ACM Symposium on Computational Geometry* (1992), pp. 192–201.
12. CHANDRA, B., DAS, G., NARASIMHAN, G., AND SOARES, J. New sparseness results on graph spanners. *International Journal of Computational Geometry and Applications 5* (1995), 125–144.
13. DAS, G. The visibility graph contains a bounded-degree spanner. In *Proceedings of the 9th Canadian Conference on Computational Geometry* (1997).
14. DAS, G., HEFFERNAN, P., AND NARASIMHAN, G. Optimally sparse spanners in 3-dimensional Euclidean space. *Proceedings of the ninth annual symposium on Computational geometry* (1993), 53–62.
15. DAS, G., AND NARASIMHAN, G. A fast algorithm for constructing sparse Euclidean spanners. *Proceedings of the tenth annual symposium on Computational geometry* (1994), 132–139.
16. DAS, G., NARASIMHAN, G., AND SALOWE, J. A new way to weigh malnourished euclidean graphs. In *SODA '95: Proceedings of the sixth annual ACM-SIAM symposium on Discrete algorithms* (Philadelphia, PA, USA, 1995), Society for Industrial and Applied Mathematics, pp. 215–222.
17. DODIS, Y., AND KHANNA, S. Design networks with bounded pairwise distance. In *Proceedings of the thirty-first annual ACM symposium on Theory of computing* (1999), ACM New York, NY, USA, pp. 750–759.
18. DUDLEY, R. Metric entropy of some classes of sets with differentiable boundaries. *J. Approximation Theory 10*, 3 (1974), 227–236.
19. ELKIN, M. L., AND PELEG, D. Strong inapproximability of the basic k-spanner problem. In *Proc. 27th International Colloquim on Automata, Langauges and Programming* (2000).
20. EPPSTEIN, D. *Handbook of Computational Geometry*. Elsevier Science, 1997, ch. Spanning Trees and Spanners, pp. 425–461.
21. GUDMUNDSSON, J., LEVCOPOULOS, C., AND NARASIMHAN, G. Improved greedy algorithms for constructing sparse geometric spanners. In *Scandinavian Workshop on Algorithm Theory* (2000), pp. 314–327.
22. GUDMUNDSSON, J., LEVCOPOULOS, C., AND NARASIMHAN, G. Fast greedy algorithms for constructing sparse geometric spanners. *SIAM Journal on Computing 31*, 5 (2002).
23. KARAVELAS, M. I., AND GUIBAS, L. J. Static and kinetic geometric spanners with applications. In *Proceeding of the Twelfth Annual Symposium on Discrete algorithms* (2001), pp. 168–176.
24. KEIL, J. Approximating the complete euclidean graph. In *SWAT 88: 1st Scandinavian Workshop on Algorithm Theory* (1988).
25. KHULLER, S., RAGHAVACHARI, B., AND YOUNG, N. Balancing minimum spanning trees and shortest-path trees. *Algorithmica 14*, 4 (1995), 305–321.
26. KORTSARZ, G. On the hardness of approximating spanners. In *APPROX: International Workshop on Approximation Algorithms for Combinatorial Optimization* (1998).
27. KORTSARZ, G., AND PELEG, D. Generating low-degree 2-spanners. In *Proceedings of the fifth annual ACM-SIAM symposium on Discrete algorithms* (2000).
28. LANTHIER, M., MAHESHWARI, A., AND R. SACK, J. Approximating Shortest Paths on Weighted Polyhedral Surfaces. *Algorithmica 30*, 4 (2001), 527–562.

29. LI, X.-Y., SONG, W.-Z., AND WANG, W. A unified energy efficient topology for unicast and broadcast. In *ACM MobiCom* (2005).
30. LI, X.-Y., AND WANG, Y. Efficient construction of low weight bounded degree planar spanner. *International Journal of Computational Geometry and Applications 14*, 1-2 (2004), 69–84.
31. MITCHELL, J., MOUNT, D., AND PAPADIMITRIOU, C. The discrete geodesic problem. *SIAM Journal on Computing 16*, 4 (1987), 647–668.
32. NARASIMHAN, G., AND SMID, M. *Geometric Spanner Networks*. Cambridge University Press, 2007.
33. NARASIMHAN, G., AND SMID, M. H. M. Approximating the stretch factor of euclidean graphs. *SIAM J. Comput. 30*, 3 (2000), 978–989.
34. PELEG, D., AND SCHAFFER, A. Graph spanners. *Journal of Graph Theory 13*, 1 (1989), 99–116.
35. RAO, S. B., AND SMITH, W. D. Approximating geometrical graphs via spanners and banyans. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing* (1998).
36. REGEV, H. *The Weight of the Greedy Graph Spanner*. Weizmann Institute of Science, Dept. of Applied Mathematics and Computer Science, 1995.
37. REGEV, H. The weight of the greedy graph spanner. Tech. Rep. CS95-22, 1, 1995.
38. RUPERT, J., AND SEIDEL, R. Approximating the d-dimensional complete euclidean graph. In *Proc. of 3rd Canadian Conf. on Computational Geometry* (1991), pp. 207–210.
39. SALOWE, J. Constructing of multidimensional spanner graphs with applications to minimum spanning trees. In *ACM Symposium on Computational Geometry* (1991), pp. 256–261.
40. SCHREIBER, Y., AND SHARIR, M. An optimal-time algorithm for shortest paths on a convex polytope in three dimensions. In *SCG '06: Proceedings of the twenty-second annual symposium on Computational geometry* (New York, NY, USA, 2006), ACM, pp. 30–39.
41. SHARIR, M., AND SCHORR, A. On shortest paths in polyhedral spaces. *SIAM J. Comput. 15*, 1 (1986), 193–215.
42. VAIDYA, P. A sparse graph almost as good as the complete graph on points in K dimensions. *Discrete and Computational Geometry 6*, 1 (1991), 369–381.
43. VARADARAJAN, K. R., AND AGARWAL, P. K. Approximating shortest paths on a nonconvex polyhedron. *SIAM J. Comput. 30*, 4 (2000), 1321–1340.
44. VENKATESAN, G., ROTICS, U., MADANLAL, M., MAKOWSKY, J., AND RANGAN, C. Restrictions of Minimum Spanner Problems. *Information and Computation 136*, 2 (1997), 143–164.

Appendix

Proof of lemma 3

PROOF. Order all the edges in the complete graph $\mathcal{K}_{\mathcal{P}}(U,E)$ in the increasing order of the geodesic shortest distances between the end-points. The proof is by induction on the distances. Let u and v be two vertices from U with shortest geodesic distance $\mathbf{d}_{\mathcal{P}}(u,v) = d$. Assume that the geodesic path $\Pi_{\mathcal{P}}(u,v) = u \leftrightsquigarrow v_1 \leftrightsquigarrow v_2 \leftrightsquigarrow \cdots \leftrightsquigarrow v_a$ where v_i, $1 \leq i \leq a$ are vertices of P, and v_a is v. Here $v_i \leftrightsquigarrow v_{i+1}$ is the shortest geodesic path connecting nodes v_i and v_{i+1}, which is a piece-wise linear path passing through a number of triangular faces. Suppose that $v_1 \in C$ where $C \in \mathcal{C}(u)$, i.e., the starting segment $u \leftrightsquigarrow v_1$ of the geodesic shortest path $\Pi_{\mathcal{P}}(u,v)$ from u to

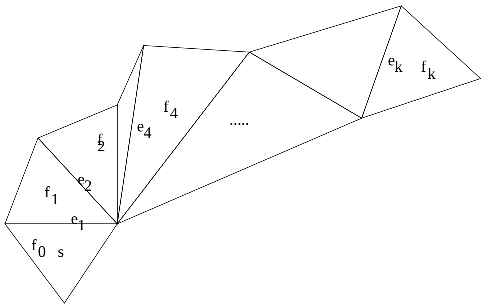
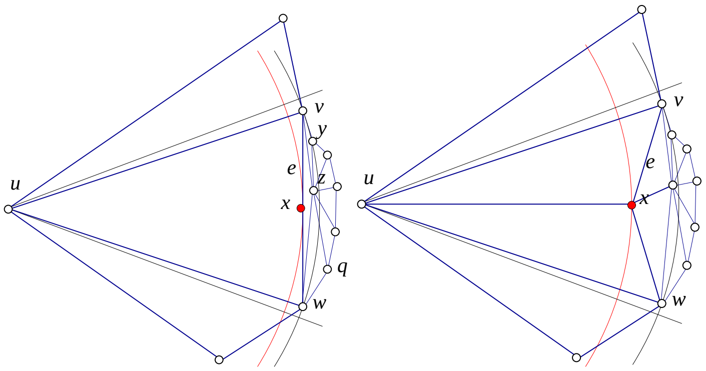


Fig. 2. A Planar unfolding example.



(a) close point is on a segment (b) closest point is a vertex

Fig. 3. An example of propagating cone to other faces.

v lies in the cone C. Let w be the closest, in order of geodesic distance, to u, from amongst all other nodes in the cone. Then our algorithm will add the edge uw. It is easy to show that $\mathbf{d}_{\mathcal{P}}(w, v_1) < \mathbf{d}_{\mathcal{P}}(u, v_1) \leq \mathbf{d}_{\mathcal{P}}(u, v)$. Then by induction there is a path in $H'(U \cup S, A')$ between u and w and between w and v_1 of geodesic length at most $\mathbf{d}_{\mathcal{P}}(u, w)$ and $t(\epsilon_1) \cdot \mathbf{d}_{\mathcal{P}}(w, v_1)$, respectively. It is not difficult to show that this path has length at most $\mathbf{d}_{\mathcal{P}}(u, w) + t(\epsilon_1) \cdot \mathbf{d}_{\mathcal{P}}(w, v_1) \leq t(\epsilon_1) \cdot \mathbf{d}_{\mathcal{P}}(u, v_1)$, where $t(\epsilon_1) = \frac{1}{1-2\sin(\epsilon_1/2)}$. Similarly, we can prove that there is a path in $H'(U \cup S, A')$ between v_i and v_{i+1} with length at most $t(\epsilon_1) \cdot \mathbf{d}_{\mathcal{P}}(v_i, v_{i+1})$.

Consequently, the shortest path $P(u, v)$ in $H'(U \cup S, A')$ from u to v that uses w has length at most $\mathbf{d}_H(u, v) \leq \sum_{i=0}^{a-1} t \cdot \mathbf{d}_{\mathcal{P}}(v_i, v_{i+1}) = t(\epsilon_1)\mathbf{d}_{\mathcal{P}}(u, v)$ where v_0 is u. Observe that if the graph $H'(U \cup S, A')$ has a path of length at most $t(\epsilon_1)\mathbf{d}_{\mathcal{P}}(u, v)$, the graph $H(U \cup S, E)$ will obviously also preserve at least one such path. In other words, graph $H(U \cup S, E)$ is a $t(\epsilon_1)$-spanner. We can choose ϵ_1 such that $t(\epsilon_1) = t$ if Phase 2 is not executed.

Note that, if we run the optional procedure to further reduce the node degree, a directed edge uv in $H'(U \cup S, A')$ may now be replaced by a path with geodesic length at most $t(\epsilon_2) = \frac{1}{1-2\sin\frac{\epsilon_2}{2}}\mathbf{d}_{\mathcal{P}}(u, v)$, depending on the angle ϵ_2 of the cones. In this case, to ensure the final structure as a t-spanner, we have to choose the angle ϵ_1 and ϵ_2 of the cones to satisfy $t = t(\epsilon_1) \cdot t(\epsilon_2)$. This finishes the proof. ❑

Proof of Lemma 4

PROOF. To show the degree bound, consider the edges in $H'(U \cup S, A')$. Our t-spanner construction method (Algorithm 1) will add at most 1 directed edge in each cone $\mathcal{C}(u)$ for every node u. Thus the total number of directed outgoing edges incident on u is bounded by the number of cones in $\mathcal{C}(u)$, which is $O(\gamma(\mathcal{P})/\epsilon_1)$. ❑

Proof of Lemma 5

PROOF. Observe that in the degree-reduction procedure, each node u will participate in the degree-reduction steps of a node v if and only if we have a directed edge (u, v), thus u will participate in at most degree-reduction of $O(\gamma(\mathcal{P})/\epsilon_1)$ nodes. Note that, for each degree reduction procedure, we will add at most $O(\gamma(\mathcal{P})/\epsilon_2)$ additional edges to node u. By choosing $\epsilon_1 = \epsilon_2 = \epsilon$ for some small value ϵ, we have the lemma. ❑

Proof of lemma 6

PROOF. The first phase of the algorithm constructs $O(1/\epsilon)$ cones from each vertex in $\mathcal{P}$. Propagation of these cones requires maintenance of $1/\epsilon$ intervals on each edge of the polyhedral surface $\mathcal{P}$. Extending the cone requires $O(1)$ steps at each edge. Thus, it will take time $O(n^2/\epsilon)$ to construct graph $H'(U \cup S, A')$. The second phase of the algorithm requires computing shortest paths for every pair of vertices and removing redundant nodes. This can be done in time $O(n^3)$ using all pairs shortest path method. ❑

Proof of lemma 8

PROOF. Notice that Peleg and Schaffer [34] showed that any t-spanner of some arbitrary weighted graph needs at least $n^{1+\frac{1}{t+2}}$ edges. Thus, for arbitrary weighted graph, the maximum degree node degree of any t-spanner is at least $n^{\frac{1}{t+2}}/2$. Unfortunately, their example (the weight of each edge is 1) cannot be realized using geodesic distance. Here we construct an example in 3-dimension as follows. Figure 1 illustrates such an example. There is a node u_0 at the origin and n other nodes u_i with coordinate $(\sin((i-1)\alpha), \cos((i-1)\alpha), 0)$. Additional $n-1$ nodes v_i with coordinate $(\sin((i-1+1/2)\alpha), \cos((i-1+1/2)\alpha), 1)$ are placed also. The polyhedral surface $\mathcal{P}$ is composed of nodes u_i, $0 \leq i \leq n$ and v_i, $1 \leq i \leq n-1$, and triangles $u_i u_0 v_i$ $1 \leq i \leq n-1$ and triangles $v_i u_0 u_{i+1}$, $1 \leq i \leq n-1$. We want to construct a t-spanner for the set of nodes $U = \{u_0, u_1, u_2, \cdots, u_n\}$. It is not difficult to show that the geodesic distance $\mathbf{d}_{\mathcal{P}}(u_i, u_j)$ satisfies $1 \leq \mathbf{d}_{\mathcal{P}}(u_i, u_j) \leq 2$ for every pair of nodes u_i and u_j. Assume that we have t-spanner $H = (U, E)$ for U with the maximum degree $D+1$. Then consider the breadth-first-spanning tree rooted at node u_0. Clearly, the depth h of the BFS tree should satisfy that $\sum_{j=0}^{h} D^j \geq n$, *i.e.*, $\frac{D^{h+1}-1}{D-1} \geq n$. Further, consider any node v with depth h in H, then the hop-distance between u_0 and v in H is at least h. Thus, the distance $h \leq \mathbf{d}_H(u_0, v) \leq 2h$. Since H is a t-spanner, we have $\mathbf{d}_H(u_0, v) \leq t\mathbf{d}_{\mathcal{P}}(u_0, v) = t$. Thus, $h \leq t$. Consequently, $D^{t+1} \geq D^{h+1} \geq n(D-1)+1$. Thus, $D > (\frac{n}{2})^{\frac{1}{t}}$, *i.e.*, $D = \Omega(n^{\frac{1}{t}})$.

The result by Peleg and Schaffer [34] also implied that the total edge weight of any t-spanner for the example constructed above is at least $n^{\frac{1}{t+2}}/2$ times of the weight of minimum spanning tree for any $t > 1$ since the weight of MST is at least n here and any t-spanner H has at least $n^{1+\frac{1}{t+2}}$ edges. $\square$

Proof of property *Low-distortion projection property*

Property 2. Low-distortion projection property: Let u and v be two points on a δ-patch, F. Then $\mathbf{d}_{\mathcal{P}}(u, v) \geq d(u, v) \geq \mathbf{d}_{\mathcal{P}}(u, v)/(1 + 2 \cdot \delta)$ where $d(u, v)$ is the Euclidean distance between u and v on $\Xi(F)$ and $\mathbf{d}_{\mathcal{P}}(u, v)$ is the geodesic distance on F.

PROOF. Consider the geodesic path between the two points u and v. We first provide an upper bound on its length. Consider the planes, orthogonal to the normals $N(u)$ and $N(v)$, at the points u and v, termed $T(u)$ and $T(v)$, respectively. The shortest geodesic path from u to v when unfolded onto the plane $T(u)$ is a straight line. Let v' be the unfolded image of v. Consider the length of the straight line uv' in comparison to $|uv|$ Since the faces reside on a δ-patch, the angle between uv and uv' is almost δ. Thus $|uv| \geq |uv'|(1-\delta)$. Comparing with the projection of uv onto the plane gives an error with factor $(1 + 2 \cdot \delta)$. $\square$

Proof of Lemma 10

PROOF. W.l.o.g assume that the faces of $\mathcal{P}$ are triangles. Let u and v be two vertices on the convex hull of the δ-patch. Let the straight line-segment (u, v) lie on a face A on the convex extension but which is not part of the δ-patch. Let B and C be the faces on the δ-patch on which the vertices u and v reside. Since $\mathcal{P}$ is convex the normal to A is a vector with angle α whose value lies in the cone defined by the vectors β and β', the normals to the faces B and C, respectively. □

Proof of lemma 11

PROOF. Because of the *Low-distortion projection property*, the spanner on $\Xi(F)$ defines a spanner on F also: the spanning ratio on F depending on the spanning ratio of the spanner on $\Xi(F)$. It suffices to show that the spanner property is true for $u \in F_1$ and $v \in F_2$, where F_1 and F_2 are two different patches. Consider the shortest path between nodes u and v, residing in a cone in $\mathcal{C}(u)$. For this cone a Steiner vertex, p, is added to the boundary , $\partial\mathcal{E}(F)$. This Steiner point is also present in the neighboring patch. Using standard arguments, a $(1 + \epsilon)$ approximate path can be obtained from the path between u and p and then from p to v. By induction on the rank of geodesic distances between pairs of vertices, both these paths are within a factor $t = (1 + \epsilon)$ of the optimum. The combined path is also a t-factor path between u and v, since p is contained in the $(1+\epsilon)$ cone with apex u. Then the constructed subgraph is a spanner. Observe that our spanner does use Steiner points. □

Proof of lemma 12

PROOF.

The degree bound follows from the degree bound provided by the Euclidean spanner construction.

To bound the weight, consider a δ-patch, F, comprising a set of points $V = U_F \cup \mathcal{S}_F$. Let the spanner edges, restricted to F, be $H_F(V_F, A_F)$. First, construct a TSP tour, T, from $MST(U)$ on the surface of $\mathcal{P}$ using a shortcut of the Eulerian tour of the MST. Shortcut the tour to obtain T_F, a tour restricted to the patch F. The weight of the Euclidean spanner $H_F(V_F, A_F)$ on $\Xi(F)$ is $O(MST(\Xi(V_F))$. Furthermore, $MST(V_F) \leq MST(\Xi(V_F))(1 + \delta)$.

We now need to relate $MST(V_F)$ to T_F where $V_F = U_F \cup \mathcal{S}_F$. Clearly, $MST(V_F) \leq MST(U_F) + MST(\mathcal{S}_F) + w(E)$ where E is some special edge joining some point u in U_F to the corresponding Steiner node $\mathcal{S}_F$ (added due to u) and $MST(U_F) \leq T_F$. The Steiner points all lie on the boundary of F and are constructed by adding a point in each cone $C \in \mathcal{C}(u)$. There are $O(\gamma/\epsilon)$ cones in $\mathcal{C}(u)$.

Consider a cone C. The geodesic distances within F are closely related to the straight line distance between two points and so for ease of understanding we will consider the cones as defined by straight lines and the δ-patch a planar surface. Consider the set of Steiner points, $\mathcal{S}_F(u, \alpha)$ generated by the set of cones

$$\{C(u) \mid u \in U_F,\ \text{cone } C(u) \text{ in one single direction } \alpha,\ \text{each cone of angle } \epsilon\}$$

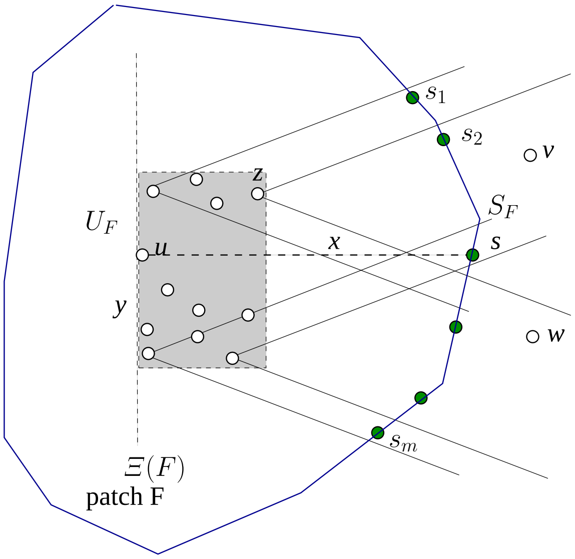


Fig. 4. We bound the length of minimum spanning tree of nodes in $U_F \cup S_F$. Here the nodes s_i on the boundary of $\varXi(F)$ are Steiner nodes. Here x is the longest distance between Steiner nodes on boundary and the original input nodes, denoted as U_F, inside the patch $\varXi(F)$; y is the longest distance between nodes U_F.

Let the set U_F be of diameter y and let the maximum distance of points in U_F from the set of Steiner points be x. Since the Steiner points, $\mathcal{S}_F$, are generated from points in U_F using the cone, the points in $\mathcal{S}_F$ are at most $y+(y+x)\cdot\epsilon \leq (y+x)(1+\epsilon)$ distance apart. Consider the spanning tree of $\mathcal{S}_F$ obtained by joining the points in sequence along the boundary of F. This distance is bounded by $O(y+x)$ from the fact that the boundary of the patch is convex. Moreover let E be an edge obtained from joining a node u in U_F to a corresponding Steiner point in $\mathcal{S}_F$ generated by a shortest path from u to a node $w \notin F$. This implies that $w(E) \leq MST(T_F)$. Moreover, $x \leq y + w(E)$. Thus $MST(V_F) \leq MST(U_F) + MST(\mathcal{S}_F) + w(E) = O(T_F) = O(MST(U))$. ◻

Proof of Lemma 13

PROOF. The algorithm constructs the cone partition in $(n/\epsilon)^2$ steps. Computation of the shortest path between each pair of vertices requires $O(n^2 \log n)$ steps [40]. Further, for each of the $1/\epsilon^2$ δ-patches, the algorithm requires $T_E(n)$ steps where $T_E(n)$ is the time required by any algorithm to compute a planar t-spanner. Note that finding the convex extensions and the projections can be performed within the time bounds stated. ◻