

Region-Fault Tolerant Geometric Spanners

M.A. Abam · M. de Berg · M. Farshi ·
J. Gudmundsson

Received: 15 May 2007 / Revised: 5 September 2008 / Accepted: 1 January 2009 /
Published online: 30 January 2009
© Springer Science+Business Media, LLC 2009

Abstract We introduce the concept of region-fault tolerant spanners for planar point sets and prove the existence of region-fault tolerant spanners of small size. For a geometric graph \mathcal{G} on a point set P and a region F , we define $\mathcal{G} \ominus F$ to be what

Communicated by Joseph S.B. Mitchell.

M.A. Abam was supported by the Netherlands' Organisation for Scientific Research (NWO) under project no. 612.065.307 and by the MADALGO Center for Massive Data Algorithmics, a Center of the Danish National Research Foundation.

M. de Berg was supported by the Netherlands' Organisation for Scientific Research (NWO) under project no. 639.023.301.

M. Farshi was supported by Ministry of Science, Research and Technology of I.R. Iran.

NICTA is funded by the Australian Government as represented by the Department of Broadband, Communications and the Digital Economy and the Australian Research Council through the ICT Centre of Excellence program.

M.A. Abam

MADALGO Center, Department of Computer Science, Aarhus University, Aarhus, Denmark
e-mail: abam@madalgo.au.dk

M. de Berg

Department of Computer Science, TU Eindhoven, P.O. Box 513, 5600 MB Eindhoven,
The Netherlands
e-mail: m.t.d.berg@tue.nl

M. Farshi

School of Computer Science, Carleton University, Ottawa, ON, K1S 5B6, Canada
e-mail: mfarshi@cg.scs.carleton.ca

M. Farshi

Department of Computer Science, Yazd University, P.O. Box 89195-741, Yazd, Iran

J. Gudmundsson (✉)

NICTA, Sydney, Australia
e-mail: joachim.gudmundsson@gmail.com

remains of \mathcal{G} after the vertices and edges of \mathcal{G} intersecting F have been removed. A \mathcal{C} -fault tolerant t -spanner is a geometric graph \mathcal{G} on P such that for any convex region F , the graph $\mathcal{G} \ominus F$ is a t -spanner for $\mathcal{G}_c(P) \ominus F$, where $\mathcal{G}_c(P)$ is the complete geometric graph on P . We prove that any set P of n points admits a \mathcal{C} -fault tolerant $(1 + \varepsilon)$ -spanner of size $\mathcal{O}(n \log n)$ for any constant $\varepsilon > 0$; if adding Steiner points is allowed, then the size of the spanner reduces to $\mathcal{O}(n)$, and for several special cases, we show how to obtain region-fault tolerant spanners of $\mathcal{O}(n)$ size without using Steiner points. We also consider *fault-tolerant geodesic t -spanners*: this is a variant where, for any disk D , the distance in $\mathcal{G} \ominus D$ between any two points $u, v \in P \setminus D$ is at most t times the geodesic distance between u and v in $\mathbb{R}^2 \setminus D$. We prove that for any P , we can add $\mathcal{O}(n)$ Steiner points to obtain a fault-tolerant geodesic $(1 + \varepsilon)$ -spanner of size $\mathcal{O}(n)$.

Keywords Geometric networks · Geometric spanners · Fault-tolerance

1 Introduction

A *geometric network* on a set P of points in d -dimensional space is an undirected graph $\mathcal{G}(P, E)$ with vertex set P whose edges are straight-line segments connecting pairs of points in P . Often the space considered is the Euclidean plane—but this is also the setting we shall consider—but other metrics and/or higher dimensions can be considered as well. Geometric networks naturally model many real-life networks, such as road networks, telecommunication networks, and so on.

When designing a network for a given set P of points, several criteria can be taken into account. In particular, in many applications it is important to ensure a fast connection between every pair of points in P . For this, it would be ideal to have a direct connection between every pair of points—the network would then be a complete graph—but in most applications this is unacceptable due to the high costs. This leads to the concepts of spanners, as defined below. Spanners were introduced by Peleg and Schäffer [19] in the context of distributed computing and by Chew [4] in a geometric context.

For two vertices u, v in a weighted graph \mathcal{G} , we use $d_{\mathcal{G}}(u, v)$ to denote their distance in the graph, that is, the length of the (weighted) shortest path between them. Now consider a weighted graph $\mathcal{G}(V, E)$ and a graph $\mathcal{G}'(V, E')$ on the same vertex set but with edge set $E' \subseteq E$. We say that \mathcal{G}' is a t -spanner of \mathcal{G} if for each pair of vertices $u, v \in V$, we have that $d_{\mathcal{G}'}(u, v) \leq t \cdot d_{\mathcal{G}}(u, v)$. The *dilation* or *stretch factor* of \mathcal{G}' is the minimum t for which \mathcal{G}' is a t -spanner of \mathcal{G} . The *size* of \mathcal{G}' is defined as the number of edges in E' .

For geometric networks, the weight of an edge (u, v) is defined to be the Euclidean distance $d(u, v)$ between u and v . Now we say that a geometric network $\mathcal{G}(P, E)$ is a (*geometric*) t -spanner if $\mathcal{G}(P, E)$ is a t -spanner of $\mathcal{G}_c(G)$, where $\mathcal{G}_c(G)$ is the complete geometric network on P . In other words, for any two points $p, q \in P$, the graph distance in \mathcal{G} is at most t times the Euclidean distance between the two points. Geometric spanners have received a lot of attention over the past few years—see the survey papers [10, 12, 25] and the book by Narasimhan and Smid [18]. From now on, we shall limit our discussion to geometric spanners.

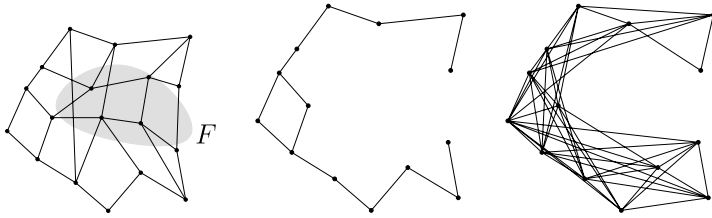


Fig. 1 The input graph \mathcal{G} and a fault region F , the graph $\mathcal{G} \ominus F$, and the graph $\mathcal{G}_c(\mathcal{G}) \ominus F$

The spanner concept captures the notion of “good” networks when short connections between the points are important. The main question is whether spanners exist that have a small stretch factor and a small, ideally near-linear, size. Other desirable properties of a spanner are, for example, that the total weight of the edges is small or that the maximum degree is low. As it turns out, such spanners do indeed exist: it has been shown that for any set P of n points and for any fixed $\varepsilon > 0$, there exists a $(1 + \varepsilon)$ -spanner with $\mathcal{O}(n)$ edges, bounded degree, and whose total weight is $\mathcal{O}(wt(\text{MST}(P)))$, where $wt(\text{MST}(P))$ is the weight of a minimum spanning tree of P [7, 18].

Another useful property of a network is fault tolerance: after one or more vertices or edges fail, the spanner should retain its good properties. In particular, there should still be a short path between any two vertices in what remains of the spanner after the fault. Levcopolous et al. [16] showed the existence of k -vertex (or k -edge) fault-tolerant geometric spanners with $\mathcal{O}(nk \log n)$ edges. This was improved by Lukovszki [17], who presented a fault-tolerant spanner with $\mathcal{O}(nk)$ edges, which is optimal. Later Czumaj and Zhao [6] showed that a greedy approach produces a k -vertex (or k -edge) fault-tolerant geometric $(1 + \varepsilon)$ -spanner with degree $\mathcal{O}(k)$ and total weight $\mathcal{O}(k^2 \cdot wt(\text{MST}(P)))$; these bounds are asymptotically optimal.

The papers on fault-tolerant spanners mentioned above all consider faults that can destroy an arbitrary collection of k vertices or edges. For geometric spanners, however, it is natural to consider *region faults*: faults that do not destroy an arbitrary collection of vertices and edges, but faults that destroy all vertices and edges intersecting some geometric fault region. This is relevant, for instance, when the spanner models a road network and a natural (or other) disaster makes all the roads in some region inaccessible. This is the topic of our paper: we study the existence of sparse spanners in the plane¹ that are tolerant against region faults. Before we present our results, let us define region-fault tolerance more precisely.

Let \mathcal{F} be a family of regions in the plane, which we call the *fault regions*. For a fault region $F \in \mathcal{F}$ and a geometric graph \mathcal{G} on a point set P , we define $\mathcal{G} \ominus F$ to be the part of \mathcal{G} that remains after the points from P inside F and all edges that intersect F have been removed from the graph, see Fig. 1. (For simplicity, we assume that a region fault F does not contain its boundary, i.e., only vertices and edges intersecting the interior of F will be affected.)

¹The concepts and many of the results carry over to d -dimensional Euclidean space. However, we feel that the concept is mainly interesting in the plane, so we confine ourselves to the planar case in this paper.

An \mathcal{F} -fault tolerant t -spanner is a geometric graph \mathcal{G} on P such that for any region $F \in \mathcal{F}$, the graph $\mathcal{G} \ominus F$ is a t -spanner for $\mathcal{G}_c(G) \ominus F$. (Recall that $\mathcal{G}_c(G)$ is the complete geometric graph on P .) We are mainly interested in the case where \mathcal{F} is the family \mathcal{C} of convex sets. It is easy to see that there are no sparse region-fault tolerant t -spanners with respect to nonconvex faults: if \mathcal{HH} denotes the family of regions that are the union of two half-planes, then $\mathcal{G}_c(G)$ is the only \mathcal{HH} -fault tolerant t -spanner for P for any finite t . For example, for each pair p and q in P , one can draw two half-planes region faults which contain all points of P except p and q , which implies that the spanner must have an edge between p and q in order to be an \mathcal{HH} -fault tolerant t -spanner.

We shall also consider the case where we are allowed to add Steiner points to the graph. In other words, instead of constructing a geometric network for P , we are allowed to construct a network for $P \cup Q$ for some set Q of Steiner points. Then we say that a graph \mathcal{G} on $P \cup Q$ is an \mathcal{F} -fault tolerant Steiner t -spanner for P if, for any $F \in \mathcal{F}$ and any two points $u, v \in P \setminus F$, the distance between u and v in $\mathcal{G} \ominus F$ is at most t times their distance in $\mathcal{G}_c(G) \ominus F$.

We also study another variant of region-fault tolerance. In this variant we require that the distance between any two points u, v in $\mathcal{G} \ominus F$ is at most t times the geodesic distance between u and v in $\mathbb{R}^2 \setminus F$. Note that the geodesic distance in $\mathbb{R}^2 \setminus F$ —that is, the length of a shortest path in $\mathbb{R}^2 \setminus F$ —is never more than the distance between u and v in $\mathcal{G}_c(G) \ominus F$. We call a spanner with this property an \mathcal{F} -fault tolerant geodesic $(1 + \varepsilon)$ -spanner. It is not difficult to show that \mathcal{F} -fault tolerant geodesic spanners do not exist unless we are allowed to use Steiner points. To see this, consider a graph with two vertices and one edge connecting them. After any fault which intersects the edge, the distance between the two vertices in the graph is infinite, while the geodesic distance between them is finite. Even in the case of Steiner points, finite size \mathcal{F} -fault tolerant geodesic spanners do not exist when \mathcal{F} is the family \mathcal{C} of all convex sets. Again, consider a set of two points. Then for any finite set of Steiner points, we can destroy all edges crossing the perpendicular bisector of the two points using a (possibly very long and skinny) convex area fault, so after the fault the two points are no longer connected in the spanner. Hence, we restrict our attention to \mathcal{D} -fault tolerant geodesic spanners, where \mathcal{D} is the family of disks in the plane.

We obtain the following results.

- In Sect. 2 we present a general method to convert a well-separated pair decomposition (WSPD) [2] for P into a \mathcal{C} -fault tolerant spanner for P . We use this method to obtain linear-size \mathcal{C} -fault tolerant $(1 + \varepsilon)$ -spanners for points in convex position and for points distributed uniformly at random inside the unit square, and to obtain linear-size \mathcal{C} -fault tolerant Steiner $(1 + \varepsilon)$ -spanners for arbitrary point sets.
- In Sect. 3 we study small \mathcal{C} -fault tolerant (non-Steiner) spanners for arbitrary point sets. By combining a more relaxed version of the WSPD with ideas from Θ -graphs [15], we show that any point set P admits a \mathcal{C} -fault tolerant $(1 + \varepsilon)$ -spanner of size $\mathcal{O}(n \log n)$.
- In Sect. 4 we study the geodesic case. We show that for any set P of n points, there exists a \mathcal{D} -fault tolerant geodesic Steiner $(1 + \varepsilon)$ -spanner with $\mathcal{O}(n)$ edges and $\mathcal{O}(n)$ Steiner points.

2 Constructing \mathcal{C} -Fault Tolerant Spanners Using the WSPD

In this section we show a general method to obtain a \mathcal{C} -fault tolerant spanner from a well-separated pair decomposition of a point set P . We show that for some special cases, this approach generates a \mathcal{C} -fault tolerant spanner with only a linear number of edges. We also show how to use the approach to obtain small Steiner spanners. Before we start, we prove a general lemma showing that, when constructing \mathcal{C} -fault tolerant spanners, we can in fact restrict our attention to half-plane faults. This lemma will also be used in later sections. Let \mathcal{H} be the family of half-planes in the plane.

Proposition 2.1 *A geometric graph \mathcal{G} on a set P of points in the plane is a \mathcal{C} -fault tolerant t -spanner if and only if it is an \mathcal{H} -fault tolerant t -spanner.*

Proof Obviously a graph is \mathcal{H} -fault tolerant if it is \mathcal{C} -fault tolerant. To prove the other direction assume that \mathcal{G} is an \mathcal{H} -fault tolerant t -spanner and that $F \in \mathcal{C}$ is an arbitrary convex area fault. We need to prove that between every pair of points $u, v \in P \setminus F$, there is a path in $\mathcal{G} \ominus F$ of length at most t times the length of the shortest path in $\mathcal{G}'_c = \mathcal{G}_c(G) \ominus F$.

If u and v are not connected in \mathcal{G}'_c , we are done. Otherwise, let Π be a shortest path between u and v in \mathcal{G}'_c . We claim that for every edge (p, q) in Π , there is a path in $\mathcal{G} \ominus F$ of length at most $t \cdot d(p, q)$. Since the edge (p, q) lies outside F and F is convex, there must be a half-plane h that contains F but does not intersect (p, q) . Since \mathcal{G} is an \mathcal{H} -fault tolerant t -spanner, there is a path $\Pi_{(p,q)}$ between p and q in $\mathcal{G} \ominus h$ of length at most $t \cdot d(p, q)$. Furthermore, since $F \subset h$, the path $\Pi_{(p,q)}$ also exists in $\mathcal{G} \ominus F$. The claim and, hence, the lemma follows. \square

2.1 The Well-Separated Pair Decomposition

The well-separated pair decomposition (WSPD) was developed by Callahan and Kosaraju [3]. We briefly review (the planar version of) this decomposition here.

Definition 2.1 Let $s > 0$ be a real number referred to as the *separation constant*. We say that two point sets A and B in the plane are *well-separated with respect to s* if there are two disjoint disks D_A and D_B of the same radius, r , such that

- (i) D_A contains A , and D_B contains B .
- (ii) The distance between D_A and D_B is at least $s \cdot r$.

Definition 2.2 Let P be a set of n points in the plane, and let $s > 0$ be a real number. A *well-separated pair decomposition (WSPD) for P with respect to s* is a collection $\mathcal{W} := \{(A_1, B_1), \dots, (A_m, B_m)\}$ of pairs of nonempty subsets of P such that

1. A_i and B_i are well separated w.r.t. s for all $i = 1, \dots, m$.
2. For any two distinct points p and q of P , there is exactly one pair (A_i, B_i) in the collection such that (i) $p \in A_i$ and $q \in B_i$ or (ii) $q \in A_i$ and $p \in B_i$.

The number of pairs, m , is called the *size* of the WSPD. Callahan and Kosaraju [3] show that any set P admits a WSPD of size $m = \mathcal{O}(s^2 n)$.

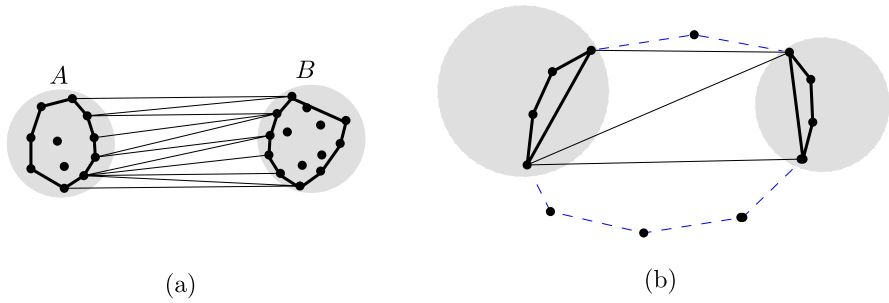


Fig. 2 (a) Illustrating the construction of the WSPD-graph. (b) Points in convex position

2.2 Constructing a \mathcal{C} -Fault Tolerant Spanner

Callahan and Kosaraju [2] showed that the WSPD can be used to obtain a small $(1 + \varepsilon)$ -spanner. Similar ideas were used earlier by Salowe [23, 24] and Vaidya [26–28]. To obtain the $(1 + \varepsilon)$ -spanner one simply computes a WSPD with respect to $s := 4 + 8/\varepsilon$, and then for each well-separated pair (A, B) , one adds an arbitrary edge connecting a point from A to a point in B .

Unfortunately this construction is not \mathcal{C} -fault tolerant, because a fault F can destroy the spanner edge that connects a pair (A, B) , while some other edges between A and B (which are not in the spanner) may survive the fault. Hence, we need to add more than a single edge for (A, B) . Let $\text{CH}(A)$ and $\text{CH}(B)$ denote the convex hulls of A and B , respectively. At first sight it seems that adding the two outer tangents of $\text{CH}(A)$ and $\text{CH}(B)$ to our spanner may lead to a \mathcal{C} -fault tolerant spanner, but this is not the case either. Instead, we will triangulate the region in between the two convex hulls in an arbitrary manner, as illustrated in Fig. 2a.

Let $E(A, B)$ be the set of edges in the triangulation added between $\text{CH}(A)$ and $\text{CH}(B)$, and let \mathcal{G} be the obtained graph. Note that any triangulation between $\text{CH}(A)$ and $\text{CH}(B)$ has the same number of edges. Throughout the paper we will use the notation $|\cdot|$ to denote the number of elements in a set.

Lemma 2.1 *The graph \mathcal{G} is a \mathcal{C} -fault tolerant $(1 + \varepsilon)$ -spanner for P of size $\sum_{(A,B) \in \mathcal{W}} |E(A, B)|$.*

Proof The size of the graph is obviously $\sum_{(A,B) \in \mathcal{W}} |E(A, B)|$, so it remains to show that it is a \mathcal{C} -fault tolerant $(1 + \varepsilon)$ -spanner. Now we observe that for any half-plane h , $\{(A \setminus h, B \setminus h) : (A, B) \in \mathcal{W}\}$ is a WSPD for $P \setminus h$. Hence, by Proposition 2.1 and the properties of the WSPD it is sufficient to show the following: Let h be a half-plane fault, let u, v be points not in h , and let (A, B) be a pair with $u \in A$ and $v \in B$; then there is an edge $e \in E(A, B)$ between $\text{CH}(A)$ and $\text{CH}(B)$ that is outside h .

To see this we first prove that, given a point set P and a triangulation T of P , the graph $T \ominus h$ is connected for any half-plane h . Assume without loss of generality that h is below and bounded by a horizontal line. Since any point of $P \setminus h$ not on the convex hull must have an edge connecting it to a point further away from h , we can walk from p away from h along edges of T until we reach a point on the convex hull

of P . Moreover, any two convex hull points in $P \setminus h$ can be connected by convex hull edges outside h . It follows that $T \ominus h$ is indeed connected.

Now consider any triangulation T on $A \cup B$ that includes $E(A, B)$. Then $T \ominus h$ must be connected. Since $u, v \notin h$, and $u \in A$ and $v \in B$, this means there must be an edge $e \in E(A, B)$ outside h . \square

Unfortunately there exists a point set P such that for any WSPD W of P , we have $\sum_{(A,B) \in \mathcal{W}} (|A| + |B|) = \Omega(n^2)$, see [18, Exercise 9.2]. A simple example of such a point set is n points on a line with exponentially increasing interpoint distance. In other words, the WSPD-approach will never lead to a worst-case subquadratic sized spanner.

2.3 Linear-Size Spanners for Special Cases

The method described above can be used to get small \mathcal{C} -fault tolerant spanners for several special cases. For example, if P is in convex position, then $|E(A, B)| \leq 3$ for any pair (A, B) in the decomposition, see Fig. 2b. The time complexity of the algorithm for constructing such a spanner is straightforward. To compute the WSPD of the point set, we need $\mathcal{O}(n \log n + n/\varepsilon^2)$ time. For adding the edges between each pair in the WSPD, we can compute the convex hull of each set in the pair in constant time, assuming that we compute the convex hull of the input point set beforehand. Therefore in total we need $\mathcal{O}(n \log n + n/\varepsilon^2)$ time to compute a \mathcal{C} -fault tolerant $(1 + \varepsilon)$ -spanner for points on convex position. So we get:

Theorem 2.1 *For any set P of n points in convex position in the plane and any $\varepsilon > 0$, there exists a \mathcal{C} -fault tolerant $(1 + \varepsilon)$ -spanner of size $\mathcal{O}(n/\varepsilon^2)$ which can be computed in $\mathcal{O}(n \log n + n/\varepsilon^2)$ time.*

Next we show that we can also get a \mathcal{C} -fault tolerant spanner whose expected size is linear if the point set P is generated by picking n points uniformly at random in the unit square.

Lemma 2.2 *Let P be a set of n uniformly distributed points in the unit square and A be a sub-square of the unit square. Then the expected number of points on the convex hull of $P \cap A$ is $\mathcal{O}(\log(n \cdot \text{area}(A)))$.*

Proof If n points are uniformly distributed in the unit square, then it is known that the expected number of points on the convex hull of the points is $\mathcal{O}(\log n)$ [14, 21].

Now let X be the number of points on the convex hull of $P \cap A$, and let $Y := |P \cap A|$. Clearly $\mathbf{E}[Y] = n \cdot \text{area}(A)$. By the law of total expectation [22, Proposition 4.1], if X and Y are two random variables, then $\mathbf{E}[X] = \mathbf{E}[\mathbf{E}[X|Y]]$, therefore

$$\begin{aligned} \mathbf{E}[X] &= \mathbf{E}[\mathbf{E}[X|Y]] \\ &= \mathbf{E}[\mathcal{O}(\log(Y))] \\ &\leq \mathcal{O}(\log(\mathbf{E}[Y])) \quad (\text{Jensen's inequality [22, p. 418]}) \\ &= \mathcal{O}(\log(n \cdot \text{area}(A))). \end{aligned} \quad \square$$

Now we combine the ideas from the previous section with Lemma 2.2 to construct a $(1 + \varepsilon)$ -spanner of the uniformly distributed point set P .

Constructing the quadtree is done in $\mathcal{O}(n \log n)$ time. For a fixed level of the quadtree, the total size of the subsets $P(v)$ is $\mathcal{O}(n)$. Hence, the total expected size of the subsets over all levels is $\mathcal{O}(n \log n)$, since the expected number of levels of the quadtree for a uniformly distributed point set is $\mathcal{O}(\log n)$. It is well known [8] that the convex hull of a point set can be built incrementally in linear time if the points are given in sorted order. Thus, the convex hulls of the set $P(v)$ can be computed in time $\mathcal{O}(|P(v)|)$ after presorting the points from left to right, which in total this takes $\mathcal{O}(n \log n)$ time.

Finally, computing the edges for a fixed pair (A, B) can then be done in linear time in the complexity of their convex hulls; over all pairs this is $\mathcal{O}(n/\varepsilon^2)$ expected time. In total, the expected time to construct the spanner is therefore $\mathcal{O}(n \log n + n/\varepsilon^2)$.

Theorem 2.2 *Let P be a set of n points uniformly distributed in the unit square U . For any $\varepsilon > 0$, there is a \mathcal{C} -fault tolerant $(1 + \varepsilon)$ -spanner for P of expected size $\mathcal{O}(n/\varepsilon^2)$ which can be computed in $\mathcal{O}(n \log n + n/\varepsilon^2)$ expected time.*

Proof Construct a quadtree partitioning of U into smaller and smaller squares, until each square has size (side length) between $1/\sqrt{n}$ and $2/\sqrt{n}$. So the area of any leaf is $\mathcal{O}(1/n)$, which means the expected number of points in a leaf region is $\mathcal{O}(1)$. The quadtree has $\mathcal{O}(n)$ leaves. Level ℓ of the quadtree corresponds to a regular subdivision of U into squares of size $1/2^\ell$. One can show that there exists a WSPD $\mathcal{W} := \{(A_i, B_i)\}_i$ for P of size $\mathcal{O}(n/\varepsilon^2)$ such that for each i , the pair (A_i, B_i) either corresponds to two squares at the same level, or A_i and B_i are both singleton points that lie in adjacent cells (or the same cell) of the final subdivision. Moreover, if we denote by n_ℓ the number of pairs of the WSPD at level ℓ of the quadtree, then $n_\ell = \mathcal{O}(2^{2\ell}/\varepsilon^2)$. The existence of a WSPD with these properties follows rather directly from the results of Fischer and Har-Peled [11]. For completeness, we briefly sketch an argument for our setting.

For a node v of the quadtree, let $P(v)$ denote the subset of points from P inside the square corresponding to v . Consider a level ℓ of the quadtree. For each pair of nodes v, v' at level ℓ such that the point sets $P(v)$ and $P(v')$ are well-separated, while the point sets of the parents of v and v' are not well-separated, we put the pair $(P(v), P(v'))$ into the WSPD. In addition, for pair of leaf nodes μ, μ' such that $P(\mu)$ and $P(\mu')$ are not well-separated, we put a pair $(\{p\}, \{q\})$ into the WSPD for every pair $p \in P(\mu)$ and $q \in P(\mu')$. It is easy to verify that this indeed defines a WSPD. The bound on the number of pairs added for each level follows from a standard packing argument.

Now consider a square σ at level ℓ . By Lemma 2.2, because the area of σ is $1/2^{2\ell}$, the expected size of the convex hull of the points in σ is $\mathcal{O}(\log(n/2^{2\ell}))$.

If (A, B) is an arbitrary pair in \mathcal{W} which appears at level ℓ of the quadtree, then

$$\begin{aligned} \mathbf{E}[|E(A, B)|] &\leq \mathbf{E}[|\text{CH}(A)| + |\text{CH}(B)|] \\ &= \mathbf{E}[|\text{CH}(A)|] + \mathbf{E}[|\text{CH}(B)|] \\ &= \mathcal{O}(\log(n/2^{2\ell})). \end{aligned}$$

Therefore

$$\begin{aligned}
 \mathbf{E} \left[\sum_{(A_i, B_i) \in \mathcal{W}} |E(A_i, B_i)| \right] &= \sum_{(A_i, B_i) \in \mathcal{W}} \mathbf{E}[|E(A_i, B_i)|] \\
 &= \sum_{\ell=1}^{\frac{1}{2} \log n} \mathcal{O}(n_\ell \log(n/2^{2\ell})) \\
 &= \sum_{\ell=1}^{\frac{1}{2} \log n} \mathcal{O}((2^{2\ell}/\varepsilon^2) \log(n/2^{2\ell})).
 \end{aligned}$$

To bound this summation, we set $m := \frac{1}{2} \log n$ and we get

$$\begin{aligned}
 \sum_{\ell=1}^{\frac{1}{2} \log n} 2^{2\ell} \log(n/2^{2\ell}) &= \sum_{\ell=1}^m 2^{2\ell} (2m - 2\ell) \\
 &= 2 \sum_{\ell=1}^m 2^{2\ell} (m - \ell) \\
 &= 2 \sum_{k=0}^{m-1} 2^{2(m-k)} \cdot k \quad (\text{by setting } k = m - \ell) \\
 &= 2^{2m+1} \sum_{k=0}^{m-1} \frac{k}{2^{2k}} \\
 &\leq 2^{2m+1} \sum_{k=0}^{\infty} \frac{k}{2^{2k}} \\
 &= \mathcal{O}(n).
 \end{aligned}$$

Hence the expected size of the generated $(1 + \varepsilon)$ -spanner is $\mathcal{O}(n/\varepsilon^2)$. \square

2.4 \mathcal{C} -Fault Tolerant Steiner Spanners

Above we showed that the WSPD can be used to construct \mathcal{C} -fault tolerant spanners of small size when the points are in convex position or uniformly distributed. For arbitrary point sets, however, the size of the spanner may be $\Omega(n^2)$ [18]. In this section we will show that if we are allowed to add Steiner points, we can always use the above method to get a linear-size spanner:

Theorem 2.3 *For any set P of n points in the plane and any $\varepsilon > 0$, one can construct a \mathcal{C} -fault tolerant Steiner $(1 + \varepsilon)$ -spanner of size $\mathcal{O}(n/\varepsilon^2)$ in $\mathcal{O}(n \log n + n/\varepsilon^2)$ time by adding at most $4(n - 1)$ Steiner points.*

The idea is to add a set Q of Steiner points to P such that $|E(A, B)| = \mathcal{O}(1)$ for any pair (A, B) in the WSPD of $P \cup Q$. Then the theorem immediately follows from Lemma 2.1.

Our method is based on the WSPD construction by Fisher and Har-Peled [11]. Their construction uses a compressed quadtree, which is defined as follows.

Let $\mathcal{T}(P)$ be the quadtree on P . We denote the square corresponding to a node $v \in \mathcal{T}(P)$ by $\sigma(v)$ and the subset of points from P inside $\sigma(v)$ by $P(v)$. When some of the points are very close together, a quadtree can have superlinear size. A *compressed quadtree* $\mathcal{T}^*(P)$ for P therefore removes internal nodes v from $\mathcal{T}(P)$ for which all points from P lie in the same quadrant of $\sigma(v)$. A compressed quadtree has at most $n - 1$ internal nodes. Fisher and Har-Peled [11] show that one can obtain a WSPD of size $\mathcal{O}(s^2n)$ for P that consists of pairs $(P(v_1), P(v_2))$ where v_1 and v_2 are nodes in $\mathcal{T}^*(P)$.

The set Q of Steiner points that we use is defined as follows. Let $\mathcal{T}^*(P)$ be a compressed quadtree for P . Without loss of generality, we may assume that no point from P lies on any of the splitting lines. For each internal node v of $\mathcal{T}^*(P)$, we add the four corner points of $\sigma(v)$ to Q . To avoid degenerate cases, we slightly move each point into the interior of $\sigma(v)$. Note that two (or more) squares $\sigma(v_1)$ and $\sigma(v_2)$ may share, for instance, their top right corner. In this case we add the (slightly shifted) corner point only once. The resulting set Q has size at most $4(n - 1)$. The next lemma finishes the proof of Theorem 2.3.

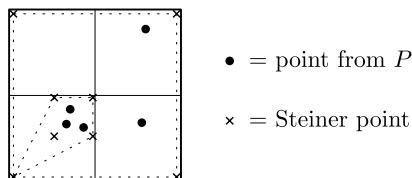
Lemma 2.3 *Let $\mathcal{T}^*(\overline{P})$ be a compressed quadtree for $\overline{P} := P \cup Q$, where the initial bounding square U is the same as for $\mathcal{T}^*(P)$, and let v be an internal node of $\mathcal{T}^*(\overline{P})$. Then $\text{CH}(\overline{P}(v))$ has at most four vertices.*

Proof If the square $\sigma(v)$ contains zero or one point from P , then at most one Steiner point has been added inside $\sigma(v)$, and the lemma is true. If $\sigma(v)$ contains two or more points, then there are two cases, both illustrated in Fig. 3.

Let μ be the node of $\mathcal{T}^*(P)$ such that $P(\mu) = \overline{P}(v) \cap P$. Note that the four shifted corners of $\sigma(\mu)$ were added as Steiner points to Q . If $\sigma(\mu) = \sigma(v)$, then $\text{CH}(\overline{P}(v))$ is a square. Otherwise, $\sigma(\mu) \subset \sigma(v)$. In this case $\text{CH}(\overline{P}(v))$ is formed by three of the four corners of $\sigma(\mu)$ together with the unique corner of $\sigma(v)$ that generated a Steiner point at some ancestor of v in $\mathcal{T}^*(P)$, see Fig. 3. Hence, in this case $\text{CH}(\overline{P}(v))$ has four vertices as well. □

The WSPD, including adding the Steiner points, can be computed in $\mathcal{O}(n \log n + n/\varepsilon^2)$ time. We can compute the convex hull of each set in constant time, because the points on the convex hulls are the Steiner point of the node itself or one of the Steiner

Fig. 3 Illustration for the proof of Lemma 2.3



points of the parent of the node, and we can triangulate the space between convex hulls in constant time since the complexity of convex hulls is constant. So in total the construction takes $\mathcal{O}(n \log n + n/\varepsilon^2)$ time.

3 \mathcal{C} -Fault Tolerant Spanners for Arbitrary Point Sets

In this section we consider the problem of constructing a sparse \mathcal{C} -fault tolerant $(1 + \varepsilon)$ -spanner for an arbitrary set P of n points in the plane without using Steiner points. The method that was described in the previous section does not guarantee a small spanner in general. Here we will describe a method that is guaranteed to result in a spanner of size $\mathcal{O}(n \log n)$.

Throughout this section $d(\cdot, \cdot)$ denotes the shortest distance between two objects (points, disks, etc.), and $\text{radius}(D)$ denotes the radius of a disk D .

3.1 SSPDs and Fault-Tolerant Spanners

The problem with the WSPD in our application is that, even though the number of pairs in the WSPD is $\mathcal{O}(n)$, the total number of points over all the pairs can be $\Theta(n^2)$. Therefore we will introduce a relaxed version of the WSPD, the SSPD.

Definition 3.1 Let A and B be two sets of points in the plane, and let $s > 0$ be a constant. We say that A and B are *semi-separated with respect to separation constant s* if there are two disjoint disks D_A and D_B such that

- (i) D_A contains A , and D_B contains B .
- (ii) $d(D_A, D_B) \geq s \cdot \min(\text{radius}(D_A), \text{radius}(D_B))$.

Thus we allow the balls D_A and D_B to be of different sizes, and we only require that the distance between the disks is large relative to the smaller disk. Note that using the same notation, we can reformulate the definition of well-separated with respect to s as $d(D_A, D_B) \geq s \cdot \max(\text{radius}(D_A), \text{radius}(D_B))$.

We now define our SSPD.

Definition 3.2 Let P be a set of n points in the plane, and let $s > 0$ be a real number. A *semi-separated pair decomposition* (SSPD) for P w.r.t. s is a collection $\{(A_1, B_1), \dots, (A_m, B_m)\}$ of pairs of nonempty subsets of P such that

1. A_i and B_i are semi-separated w.r.t. s for all $i = 1, \dots, m$.
2. For any two distinct points p and q of P , there is exactly one pair (A_i, B_i) in the collection such that (i) $p \in A_i$ and $q \in B_i$ or (ii) $q \in A_i$ and $p \in B_i$.

The *weight* of a set A , denoted by $|A|$, is defined as the number of points in A , the weight of a semi-separated pair (A, B) is the sum of the weights of A and B , and the weight of an SSPD is the total weight of all the pairs. Later we will prove that it is possible to compute an SSPD of weight $\mathcal{O}(n \log n)$. It is known that the weight of such a decomposition has $\Omega(n \log n)$ lower bound, see [1, 13]. First, however, we

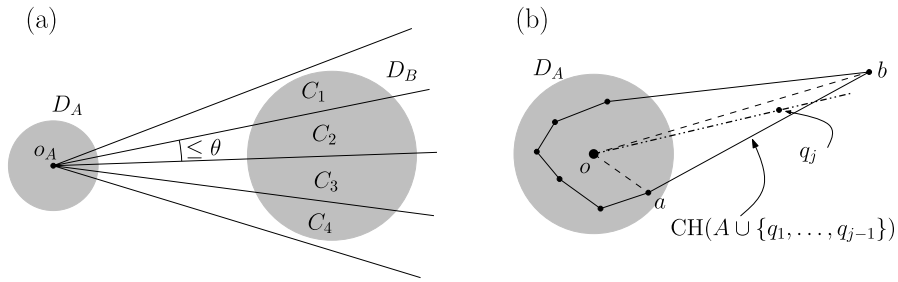


Fig. 4 (a) The cones of angle at most θ defined with respect to o_A and A . (b) Illustration for the proof that q_j is outside $\text{CH}(A \cup \{q_1, \dots, q_{j-1}\})$

will show how to use the SSPD to obtain a \mathcal{C} -fault tolerant spanner. The idea is to add edges to the spanner for each pair in the SSPD. Because the pairs in an SSPD are only semi-separated, however, adding a single edge for every pair does not necessarily lead to a good spanner. Therefore we use an idea that is also used in the construction of Θ -graphs [5, 15].

Consider a pair (A, B) in an SSPD for P . Then there exist two disjoint disks D_A and D_B that contain A and B , respectively, and for which

$$d(D_A, D_B) \geq s \cdot \min(\text{radius}(D_A), \text{radius}(D_B)).$$

Assume without loss of generality that $\text{radius}(D_A) \leq \text{radius}(D_B)$, and let o_A denote the center of D_A , see Fig. 4a. The set $E(A, B)$ of edges added to the spanner for the pair (A, B) is found as follows.

1. Partition the plane into $k := \lceil 2\pi/\theta \rceil$ cones C_1, \dots, C_k , all with apex at o_A and with interior angle at most θ , where θ is a suitable constant to be specified later. Let $B^{(i)} := B \cap C_i$ denote the subset of points from B inside the cone C_i ; here we assume without loss of generality that no point lies on the boundary between two cones.
2. Let $\text{CH}(A)$ be the convex hull of A . For each $B^{(i)}$, we sort the points in $B^{(i)}$ in order of increasing distance to o_A . Let q_1, q_2, \dots denote the sorted list of points. We process each point q_j in order as follows. Let $\text{CH}(A')$ be the convex hull of the set $A' = A \cup \{q_1, \dots, q_{j-1}\}$. We add every possible edge between q_j and the vertices of A on $\text{CH}(A')$ for which the interior of the edge does not intersect $\text{CH}(A')$. Next we update $\text{CH}(A')$ by adding the point q_j . After processing all points $q_i \in B^{(i)}$, we have produced a set $E(A, B^{(i)})$ of edges. The set $E(A, B)$ is simply $\cup_{1 \leq i \leq k} E(A, B^{(i)})$.

Note that in Step 2 for each j , the point q_j is outside $\text{CH}(A \cup \{q_1, \dots, q_{j-1}\})$. To show this assume that q_j is inside $\text{CH}(A')$ for $A' = A \cup \{q_1, \dots, q_{j-1}\}$. Then the ray from $o = o_A$ to q_j intersects an edge of $\text{CH}(A')$, see Fig. 4b. Let a and b be the endpoints of this edge. Note that q_j lies inside $\triangle oab$. On the other hand, $a, b \in A \cup \{q_1, \dots, q_{j-1}\}$, and therefore $d(o, a) \leq d(o, q_j)$ and $d(o, b) \leq d(o, q_j)$. This contradicts the fact that q_j is inside $\triangle oab$.

By construction the edges that we are adding to $E(A, B^{(i)})$ do not cross, hence $E(A, B)$ forms a planar set of edges. Since the number of sets $B^{(i)}$ is $\mathcal{O}(1/\theta)$ and $\sum |B^{(i)}| = |B|$, we have:

Lemma 3.1 $|E(A, B)| = \mathcal{O}(|A|/\theta + |B|)$.

To prove that the approach generates a fault-tolerant spanner, we need the following lemma. Consider the ordered set $B^{(i)}$ of the points in B inside the cone C_i .

Lemma 3.2 *Let h be a half-plane fault such that both A and $B^{(i)}$ have at least one point outside h . Of all the points in $B^{(i)}$ outside h , let q_j be the one with minimum distance to o_A . There is an edge in $E(A, B^{(i)})$ connecting q_j to a point $p \in A$ outside h .*

Proof By assumption, q_j is outside h , and there is at least one point from A outside h . On the other hand, by the choice of q_j , the points q_1, \dots, q_{j-1} are all inside h . As mentioned before, q_j is outside $\text{CH}(A \cup \{q_1, \dots, q_{j-1}\})$, and therefore q_j is a vertex of $\text{CH}(A \cup \{q_1, \dots, q_j\})$. Let a and b be the neighbors of q_j on $\text{CH}(A \cup \{q_1, \dots, q_j\})$. We have two cases:

Case 1: a or b lies outside h . In this case we are done because that neighbor belongs to A .

Case 2: a and b are inside h . If we continue the edges (q_j, a) and (q_j, b) , then the cone with apex at q_j contains $A \cup \{q_1, \dots, q_{j-1}\}$. By assumption there exists at least one point of A outside h , and therefore there is at least one point $p \in A$ which lies on the convex hull $\text{CH}(A \cup \{q_1, \dots, q_{j-1}\})$, see Fig. 5a. The point p is visible by q_j because all the points on $\text{CH}(A \cup \{q_1, \dots, q_{j-1}\})$ and between a and b are visible from q_j . Hence there is an edge connecting q_j to p , and so we are done. \square

Let \mathcal{W} be an SSPD of P , and let $\mathcal{G}(P, E)$ be a graph with vertex set P and edge set $E = \bigcup_{(A,B) \in \mathcal{W}} E(A, B)$. Next we prove that \mathcal{G} is a $(1 + \varepsilon)$ -spanner if we choose the separation constant s and the angle θ suitably and, moreover, that it is \mathcal{C} -fault tolerant. For this we will need the following condition on the structure of the SSPD (which will be satisfied by the SSPD we will construct later).

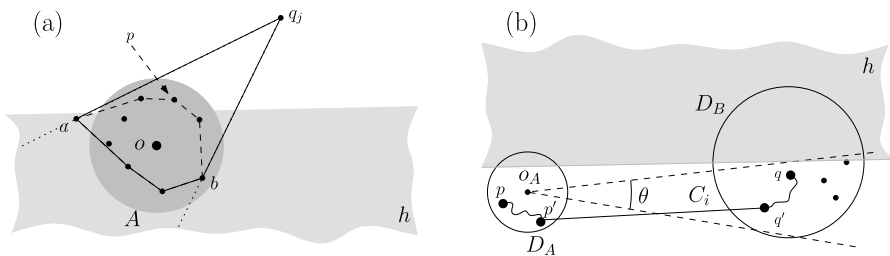


Fig. 5 (a) Illustrating the proof of Lemma 3.2. (b) Illustrating the proof of Lemma 3.3

Monotonicity condition Suppose that p, q are two points that are in the same set X of some pair (A_i, B_i) of the SSPD—thus $X = A_i$ or $X = B_i$ —and let (A_j, B_j) be the unique pair in the SSPD such that $p \in A_j$ and $q \in B_j$, or $p \in B_j$ and $q \in A_j$. Then the weights of A_j and B_j are both less than the weight of X .

Lemma 3.3 *Assume that the SSPD satisfies the monotonicity condition. If the separation constant s of the SSPD is taken as $s := \frac{3t+1}{(\cos\theta - \sin\theta)t-1}$, then the graph \mathcal{G} is an \mathcal{H} -fault tolerant t -spanner.*

Proof Let h be an arbitrary half-plane. To prove the lemma we must show that for each pair of points $p, q \in P$ outside h , there is a t -path, a path of length at most t times $d(p, q)$, connecting them in $\mathcal{G} \ominus h$. According to the definition of the SSPD, there exists a semi-separated pair (A, B) such that $p \in A$ and $q \in B$ (or vice versa).

The proof is done by induction on the maximum weight of A and B .

Base case: If the maximum weight of A and B is 1, then both sets are singletons, and therefore we must have an edge between them.

Induction hypothesis: Assume that the lemma holds for all points in pairs whose maximum weight is less than k for some $k > 1$.

Induction step: Suppose that the maximum weight of A and B is k . Let D_A and D_B be two disks containing A resp. B such that

$$d(D_A, D_B) \geq s \cdot \min(\text{radius}(D_A), \text{radius}(D_B))$$

and assume without loss of generality that $\text{radius}(D_A) \leq \text{radius}(D_B)$. Let $o = o_A$ denote the center of D_A .

Let C_i be the cone with apex o that contains q . Let q' be the point in $B^{(i)} \setminus h$ closest to o . According to Lemma 3.2, there is an edge between q' and some point p' in A outside h , see Fig. 5b. By the induction hypothesis, which we may apply because of the monotonicity condition, there are t -paths from p to p' and from q' to q in $\mathcal{G} \ominus h$. By connecting these paths using the edge (p', q') we obtain a path Π in $\mathcal{G} \ominus h$. Next we prove that Π is a t -path between p and q . Set $r := \text{radius}(D_A)$ and $\lambda := \cos\theta - \sin\theta$. For the rest of the proof, let $|xy|$ denote the (Euclidean) distance between points x and y .

Consider the triangle $\Delta oqq'$. Since $\angle qoq' \leq \theta$, we have

$$|qq'| \leq |oq| - (\cos\theta - \sin\theta) \cdot |oq'|.$$

The total length of Π , denoted $\text{length}(\Pi)$, can now be bounded as follows:

$$\begin{aligned} \text{length}(\Pi) &\leq t \cdot |pp'| + |p'q'| + t \cdot |qq'| \\ &\leq 2rt + (r + |oq'|) + t \cdot (|oq| - (\cos\theta - \sin\theta) \cdot |oq'|) \\ &= 2rt + (r + |oq'|) + t(|oq| - r) + tr - t\lambda \cdot |oq'| \\ &\leq 3rt + (r + |oq'|) + t \cdot |pq| - t\lambda \cdot |oq'| \\ &= t \cdot |pq| + r(3t + 1) + (1 - t\lambda) \cdot |oq'|. \end{aligned}$$

Since $d(D_A, D_B) \geq s \cdot r$, we have $|oq'| \geq s \cdot r$. We get

$$\text{length}(\Pi) \leq t \cdot |pq| + r(3t + 1) + sr(1 - t\lambda)$$

$$\begin{aligned}
 &= t \cdot |pq| + r \cdot \frac{3t+1}{\lambda t - 1} \cdot ((\lambda t - 1) + (1 - t\lambda)) \\
 &= t \cdot |pq|.
 \end{aligned}$$

This completes the proof of the lemma. \square

Let $t = 1 + \varepsilon$ for some small positive ε . (More precisely, we require $0 < \varepsilon < 1/2$.) Note that for $0 < \theta < 1$, we have $\cos \theta - \sin \theta > 1 - \theta - \theta^2/2$. Hence, by taking $\theta = \varepsilon/4$ we get

$$\cos \theta - \sin \theta > 1 - \theta - \theta^2/2 > 1 - 2\theta = 1 - \varepsilon/2.$$

Hence, by Lemma 3.3 we have a $(1 + \varepsilon)$ -spanner if we take

$$s = \frac{3t+1}{(\cos \theta - \sin \theta)t - 1} < \frac{6}{(1 - \varepsilon/2)(1 + \varepsilon) - 1} < 24/\varepsilon.$$

This leads to the following theorem.

Theorem 3.1 *For any set P of n points in the plane and any $\varepsilon > 0$, there exists a \mathcal{C} -fault tolerant $(1 + \varepsilon)$ -spanner of P with $\mathcal{O}((n/\varepsilon^3) \log n)$ edges. The spanner can be constructed in $\mathcal{O}((n/\varepsilon^2) \log^2 n)$ time.*

Proof By combining Proposition 2.1 and Lemma 3.3, the graph constructed by the algorithm is \mathcal{C} -fault tolerant. By Lemma 3.1 and with the construction algorithm presented below for constructing an SSPD of weight $\mathcal{O}(s^2 n \log n)$, the size of the constructed graph is

$$\begin{aligned}
 \sum_{(A,B) \in \text{SSPD}} |E(A, B)| &= \sum_{(A,B) \in \text{SSPD}} (|A|/\theta + |B|) \\
 &\leq \frac{1}{\theta} \sum_{(A,B) \in \text{SSPD}} (|A| + |B|) \\
 &= \mathcal{O}\left(\frac{s^2}{\theta} n \log n\right) \\
 &= \mathcal{O}\left(\frac{1}{\varepsilon^3} n \log n\right).
 \end{aligned}$$

This proves the first part of the theorem. To prove the running time, let (A, B) be an arbitrary pair in the SSPD and assume that $\text{radius}(D_A) \leq \text{radius}(D_B)$, where D_A and D_B are two disks containing A and B , respectively, that satisfy the semi-separated condition.

The first step of the algorithm can be done in $\mathcal{O}(|B| \log |B|)$ time. In the second step, we can compute the convex hull of A in $\mathcal{O}(|A| \log |A|)$ time. For each set $B^{(i)}$, sorting can be done in $\mathcal{O}(|B^{(i)}| \log |B^{(i)}|)$. For every j , to add the noncrossing edges between q_j and the points on $\text{CH}(A')$, it is sufficient to connect q_j to all the points

on $CH(A')$ which are between the two tangent lines of $CH(A')$ passing through q_j . Therefore the time we need for adding noncrossing edges is proportional to the number of edges times $\mathcal{O}(\log n)$. Finally we can update $CH(A')$ in $\mathcal{O}(\log m)$ time, where m is the number of points on the convex hull of A' , using an online convex hull algorithm, see [20, Chap. 3.3.6].

So in total the time for processing the pair (A, B) is bounded by

$$\mathcal{O}(|A| \log |A| + |B| \log |B| + |B| \log(|A| + |B|) + |E(A, B)| \log n).$$

Hence the total running time is

$$\begin{aligned} &\mathcal{O}\left(\sum_{(A,B) \in \mathcal{W}} (|A| \log |A| + |B| \log |B| + |B| \log(|A| + |B|) + |E(A, B)| \log n)\right) \\ &\leq \mathcal{O}\left(\sum_{(A,B) \in \mathcal{W}} ((|A| + |B|) \log n + |E(A, B)| \log n)\right) \\ &= \mathcal{O}(s^2 n \log^2 n + n \log^2 n) \\ &= \mathcal{O}(s^2 n \log^2 n). \end{aligned}$$

As we will see in the next section, we can compute the SSPD in $\mathcal{O}(s^2 n + n \log n)$ time, see Lemma 3.10, which proves the time complexity of the algorithm. \square

3.2 Computing an SSPD

To compute an SSPD for a given point set P , we use a BAR-tree, as introduced by Duncan et al. [9]. A BAR-tree for a point set P is a BSP-tree with the following properties:

1. Each leaf region contains at most one point from P .
2. The tree has size $\mathcal{O}(n)$.
3. If we go down two levels in the tree, then the size of the subtree reduces with a factor of β for some constant $1/2 < \beta < 1$, so its depth is $\mathcal{O}(\log n)$.
4. The region $\mathcal{R}(v)$ associated with an (internal or leaf) node v has aspect ratio at most α for some constant $\alpha > 1$, that is, there are concentric disks $D_I \subset \mathcal{R}(v)$ and $D_O \supset \mathcal{R}(v)$ with $\text{radius}(D_O) = \alpha \cdot \text{radius}(D_I)$.

Moreover, BAR-trees only use splitting lines that are horizontal, vertical, or diagonal, therefore the complexity of every node’s region in a BAR-tree is constant.

Let \mathcal{T} be a BAR-tree on the point set P . For a node v , we use $\text{pa}(v)$ to denote the parent of v , and we use $P(v)$ to denote the subset of points from P that are stored in the leaves of the subtree \mathcal{T}_v rooted at v . The *weight* of a node v is the number of points in $P(v)$ and is denoted $|P(v)|$. We say that a node v in \mathcal{T} has *weight class* ℓ for some integer ℓ if and only if $|P(v)| \leq n/2^\ell$ and $|P(\text{pa}(v))| > n/2^\ell$. The weight class of the root is defined to be zero. We denote the collection of nodes of weight class ℓ by $N(\ell)$. Obviously we have $\lfloor \log n \rfloor$ weight classes. Note that some of the nodes in the tree may not be in any weight class; this can happen when the weight

of a node v is almost the same as the weight of its parent. For example, this happens when $|P(\text{pa}(v))| = n/2^\ell$ for some ℓ and $|P(v)| = n/2^\ell - 1$. It can also happen that a node belongs to more than one weight class, namely when the weight of a node is much smaller than the weight of its parent. The following lemma is straightforward.

Lemma 3.4 *Every leaf node is in weight class ℓ_{\max} , where $\ell_{\max} = \lfloor \log n \rfloor$. Furthermore, on any root-to-leaf-path there is exactly one node with weight class ℓ for any $0 \leq \ell \leq \ell_{\max}$.*

For a node $v \in N(\ell)$, we define its ℓ -parent to be the node $v' \in N(\ell - 1)$ that is on the path from the root of \mathcal{T} to v (including v itself). We denote the ℓ -parent of v by $\text{pa}(\ell, v)$. Observe that v can be its own ℓ -parent, namely when $v \in N(\ell)$ and $v \in N(\ell - 1)$. By Lemma 3.4, if $v \in N(\ell)$, then one of its ancestors (possibly itself) must be in weight class $\ell - 1$, so it must have an ℓ -parent. If μ is the ℓ -parent of v , then we call v an ℓ -child of μ .

For a node v in the BAR-tree, the region corresponding to v is denoted by $\mathcal{R}(v)$ and for a region R , we let $\text{diam}(R)$ denote the diameter of the region R . As mentioned before, all nodes in the BAR-tree have bounded aspect ratio, that is, all aspect ratios are bounded by some fixed constant α .

Lemma 3.5 *If $d(\mathcal{R}(v), \mathcal{R}(\mu)) \geq \frac{(s+1)\alpha}{2} \cdot \min\{\text{diam}(\mathcal{R}(v)), \text{diam}(\mathcal{R}(\mu))\}$, then there are two disks $D_v \supset \mathcal{R}(v)$ and $D_\mu \supset \mathcal{R}(\mu)$ such that*

$$d(D_v, D_\mu) \geq s \cdot \min\{\text{radius}(D_v), \text{radius}(D_\mu)\}.$$

Proof Without loss of generality assume that $\text{diam}(\mathcal{R}(v)) \leq \text{diam}(\mathcal{R}(\mu))$. Let h be a half-plane which contains $\mathcal{R}(\mu)$ such that the distance between h and $\mathcal{R}(v)$ is $d(\mathcal{R}(v), \mathcal{R}(\mu))$. Note that the half-plane h can be viewed as a disk with infinite radius that contains $\mathcal{R}(\mu)$. Now let D_v and D_I be two concentric disks such that $D_I \subset \mathcal{R}(v) \subset D_v$ with $\text{radius}(D_v)/\text{radius}(D_I) = \alpha$, see Fig. 6. It is easy to see that

$$2 \text{radius}(D_I) \leq \text{diam}(\mathcal{R}(v)) \leq 2 \text{radius}(D_v).$$

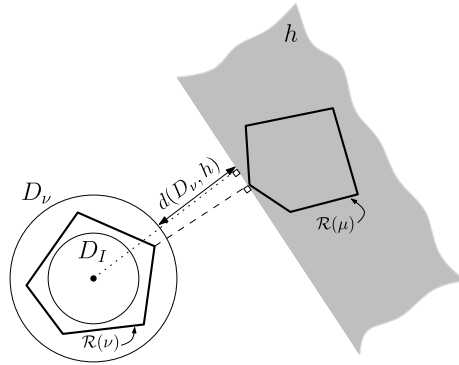
Then

$$\begin{aligned} d(D_v, h) &= d(D_I, h) - (\text{radius}(D_v) - \text{radius}(D_I)) \\ &= d(D_I, h) - (\alpha - 1) \text{radius}(D_I) \\ &\geq d(D_I, h) - \alpha \cdot \text{radius}(D_I) \\ &\geq d(D_I, h) - \frac{\alpha}{2} \cdot \text{diam}(\mathcal{R}(v)) \\ &\geq d(\mathcal{R}(v), \mathcal{R}(\mu)) - \frac{\alpha}{2} \cdot \text{diam}(\mathcal{R}(v)). \end{aligned}$$

Therefore by setting $D_\mu = h$ we have

$$d(D_v, D_\mu) \geq d(\mathcal{R}(v), \mathcal{R}(\mu)) - \frac{\alpha}{2} \cdot \text{diam}(\mathcal{R}(v))$$

Fig. 6 Illustrating the proof of Lemma 3.5



$$\begin{aligned}
 &\geq \frac{(s + 1)\alpha}{2} \cdot \text{diam}(\mathcal{R}(v)) - \frac{\alpha}{2} \cdot \text{diam}(\mathcal{R}(v)) \\
 &\geq \frac{s \cdot \alpha}{2} \cdot \text{diam}(\mathcal{R}(v)) \\
 &\geq s \cdot \alpha \cdot \text{radius}(D_I) \\
 &= s \cdot \text{radius}(D_v) \\
 &\geq s \cdot \min\{\text{radius}(D_v), \text{radius}(D_\mu)\}.
 \end{aligned}$$

So we are done. □

Now we construct an SSPD \mathcal{S} of the point set P using the following algorithm.

1. Construct a BAR tree \mathcal{T} on P . Let α be the maximum aspect ratio of the region $\mathcal{R}(v)$ for any node $v \in \mathcal{T}$. Compute the weight classes of all nodes in \mathcal{T} .
2. For each weight class ℓ with $0 \leq \ell \leq \ell_{\max}$, do the following: add to \mathcal{S} all pairs $(P(v), P(\mu))$ such that
 - (i) $v, \mu \in N(\ell)$
 - (ii) v and μ are leaves or $d(\mathcal{R}(v), \mathcal{R}(\mu)) \geq \frac{(s+1)\alpha}{2} \cdot \min\{\text{diam}(\mathcal{R}(v)), \text{diam}(\mathcal{R}(\mu))\}$, and
 - (iii) $d(\mathcal{R}(\text{pa}(\ell, v)), \mathcal{R}(\text{pa}(\ell, \mu))) < \frac{(s+1)\alpha}{2} \cdot \min\{\text{diam}(\mathcal{R}(\text{pa}(\ell, v))), \text{diam}(\mathcal{R}(\text{pa}(\ell, \mu)))\}$.

Lemma 3.6 \mathcal{S} is an SSPD for P with respect to s .

Proof By Lemma 3.5, all the pairs reported by the algorithm are semi-separated. The only thing that remains to be verified is that for every pair of points p, q , there is a unique pair $(P(v), P(\mu)) \in \mathcal{S}$ such that $p \in P(v)$ and $q \in P(\mu)$, or vice-versa.

For any $0 \leq \ell \leq \ell_{\max}$, define $v(p, \ell)$ and $v(q, \ell)$ to be the nodes of $N(\ell)$ on the search path to p and q , respectively. Observe that these nodes exist and are uniquely defined by Lemma 3.4. We have $v(p, 0) = v(q, 0) = \text{root}(\mathcal{T})$, so the sets $P(v(p, 0))$ and $P(v(q, 0))$ are the same and therefore not semi-separated. On the other hand, $v(p, \ell_{\max})$ and $v(q, \ell_{\max})$ are leaves, and so the sets $P(v(p, \ell_{\max}))$ and $P(v(q, \ell_{\max}))$ are singletons, and therefore it fulfills condition (ii) of the algorithm.

Hence, there must be a value ℓ and two nodes $v(p, \ell)$ and $v(q, \ell)$ such that they fulfill conditions (ii) and (iii) of the algorithm. The region of any node v is contained in the region of its parent, which is easily seen to imply that ℓ is unique. \square

To bound the weight of the SSPD, we first prove two auxiliary lemmas.

Lemma 3.7 *A node v in \mathcal{T} can be an ℓ -parent of at most a constant number of nodes in \mathcal{T} .*

Proof Consider a node $v \in N(\ell - 1)$ and let v' be a node such that $v = \text{pa}(\ell, v')$. Then v' is a node in \mathcal{T}_v (the subtree of \mathcal{T} rooted at v) in weight class ℓ . Note that no other node than v' in \mathcal{T}_v can have v as its ℓ -parent. Recall that the weight of a node reduces with a factor of β when we go down two levels in a BAR-tree. Since $v' \in N(\ell)$, its (normal) parent has weight at least $n/2^\ell$. On the other hand, $v \in N(\ell - 1)$, so the weight of v is at most $n/2^{\ell-1}$. Hence, the path between v and v' consists of at most $2k$ links, where $\beta^k = 1/2$. It follows that the total number of nodes in \mathcal{T} that have v as an ℓ -parent is bounded by 2^{2k} , which is a constant since k is a constant. \square

Lemma 3.8 *Let $\overline{\mathcal{S}}(\ell)$ be the set of all pairs (v, μ) such that $v, \mu \in N(\ell)$ and $d(\mathcal{R}(v), \mathcal{R}(\mu)) < \frac{(s+1)\alpha}{2} \cdot \min\{\text{diam}(\mathcal{R}(v)), \text{diam}(\mathcal{R}(\mu))\}$, where $0 \leq \ell \leq \ell_{\max}$. Then $|\overline{\mathcal{S}}(\ell)| = \mathcal{O}(\alpha^4(s+1)^2 \cdot 2^\ell)$ and*

$$\sum_{(v, \mu) \in \overline{\mathcal{S}}(\ell)} (|P(v)| + |P(\mu)|) = \mathcal{O}(\alpha^4(s+1)^2 \cdot n).$$

Proof We reorder the nodes in the pairs (v, μ) such that

$$\text{diam}(\mathcal{R}(v)) \leq \text{diam}(\mathcal{R}(\mu)).$$

We claim that any node v appears in a constant number of pairs as the first element of the pair. To show this, let (v, μ) be an arbitrary ordered pair. Let $D_{P(v)}$ be the smallest enclosing disk of $P(v)$, and let o be its center. Consider the annulus A between the disks D_1 and D_2 with center o and radii $r_1 := ((s+1)\alpha + 1) \cdot \text{radius}(D_{P(v)})$ and $r_2 := r_1 + \text{radius}(D_{P(v)})$. Note that

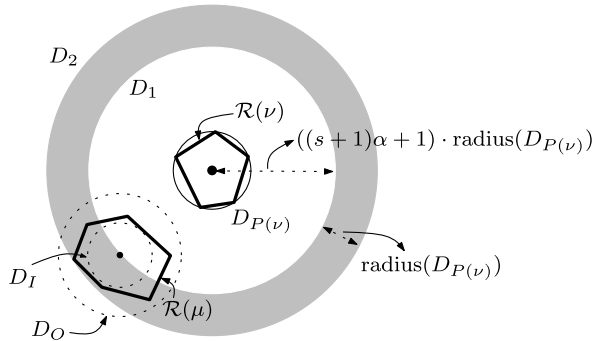
$$\text{diam}(\mathcal{R}(v))/2 \leq \text{radius}(D_{P(v)}) \leq \text{diam}(\mathcal{R}(v)).$$

Since

$$d(\mathcal{R}(v), \mathcal{R}(\mu)) < \frac{(s+1)\alpha}{2} \cdot \text{diam}(\mathcal{R}(v)),$$

the region $\mathcal{R}(\mu)$ intersect D_1 —see Fig. 7. Now we have two cases:

Fig. 7 Illustrating the proof of Lemma 3.8



Case 1: The region $\mathcal{R}(\mu)$ lies partially outside D_2 . By the Packing Lemma, [9, Lemma 3.2] this can happen for $\mathcal{O}(\alpha^2(r_1/(r_2 - r_1))) = \mathcal{O}(\alpha^3(s + 1))$ regions.

Case 2: In this case the region $\mathcal{R}(\mu)$ lies inside D_2 . Because the aspect ratio of the region $\mathcal{R}(\mu)$ is at most α , there are two disks D_I and D_O such that $D_I \subset \mathcal{R}(\mu) \subset D_O$ and $\text{area}(D_O) \leq \alpha^2 \cdot \text{area}(D_I)$, where $\text{area}(A)$ denotes the area of the region A . Therefore

$$\begin{aligned} \text{area}(\mathcal{R}(\mu)) &\geq \text{area}(D_I) \\ &\geq \frac{1}{\alpha^2} \cdot \text{area}(D_O) \\ &= \frac{1}{\alpha^2} \cdot \pi (\text{radius}(D_O))^2 \\ &\geq \frac{1}{4\alpha^2} \cdot \pi (\text{diam}(\mathcal{R}(\mu)))^2 \\ &\geq \frac{1}{4\alpha^2} \cdot \pi (\text{diam}(\mathcal{R}(v)))^2 \\ &\geq \frac{\pi r_2^2}{4\alpha^2 \cdot ((s + 1)\alpha + 2)^2}. \end{aligned}$$

On the other hand, the area of D_2 is πr_2^2 , which means that we can have at most $\mathcal{O}(\alpha^4(s + 1)^2)$ such regions.

Hence in total we can have $\mathcal{O}(\alpha^4(s + 1)^2)$ pairs that have v as the first element. Since $|N(\ell)| = \mathcal{O}(2^\ell)$, we can have $\mathcal{O}(2^\ell)$ nodes as the first element of the pair so $|\overline{\mathcal{S}}(\ell)| = \mathcal{O}(\alpha^4(s + 1)^2 \cdot 2^\ell)$. The lemma follows since $|P(v)| \leq n/2^\ell$ for each $v \in N(\ell)$. \square

Corollary 3.1 *The number of pairs in the SSPD \mathcal{S} generated by the construction algorithm is $\mathcal{O}(\alpha^4(s + 1)^2 \cdot n)$.*

Proof By the construction algorithm, if $(P(v), P(\mu)) \in \mathcal{S}$ and $v, \mu \in N(\ell)$, then $(\text{pa}(\ell, v), \text{pa}(\ell, \mu)) \in \overline{\mathcal{S}}(\ell - 1)$. By combining this with Lemma 3.7, we conclude

that the number of pairs in \mathcal{S} is bounded by $\mathcal{O}(\sum_{\ell=0}^{\log n} |\overline{\mathcal{S}}(\ell)|)$. Using Lemma 3.8, we have

$$\begin{aligned} |\mathcal{S}| &= \mathcal{O}\left(\sum_{\ell=0}^{\log n} |\overline{\mathcal{S}}(\ell)|\right) \\ &= \mathcal{O}\left(\sum_{\ell=0}^{\log n} \alpha^4(s+1)^2 \cdot 2^\ell\right) \\ &= \mathcal{O}(\alpha^4(s+1)^2 \cdot n). \end{aligned} \quad \square$$

Now we are finally ready to bound the weight of \mathcal{S} .

Lemma 3.9 *For the SSPD \mathcal{S} generated by the construction algorithm, we have*

$$\sum_{(v,\mu) \in \mathcal{S}} (|P(v)| + |P(\mu)|) = \mathcal{O}(\alpha^4(s+1)^2 \cdot n \log n).$$

Proof Since the number of weight classes in \mathcal{T} is $\mathcal{O}(\log n)$, it suffices to prove that for every fixed ℓ , it holds that

$$\sum_{\substack{(v,\mu) \in \mathcal{S} \\ v,\mu \in N(\ell)}} (|P(v)| + |P(\mu)|) = \mathcal{O}(\alpha^4(s+1)^2 n). \tag{1}$$

Obviously $|P(v)| \leq |P(\text{pa}(\ell, v))|$ for each node v , so we can bound (1) by

$$\sum_{\substack{(v,\mu) \in \mathcal{S} \\ v,\mu \in N(\ell)}} (|P(\text{pa}(\ell, v))| + |P(\text{pa}(\ell, \mu))|). \tag{2}$$

From the algorithm we know that

$$\begin{aligned} &d(\mathcal{R}(\text{pa}(\ell, v)), \mathcal{R}(\text{pa}(\ell, \mu))) \\ &< \frac{(s+1)\alpha}{2} \cdot \min\{\text{diam}(\mathcal{R}(\text{pa}(\ell, v))), \text{diam}(\mathcal{R}(\text{pa}(\ell, \mu)))\}. \end{aligned}$$

Furthermore, by Lemma 3.7 each node can be an ℓ -parent of a constant number of nodes. Hence, (2) can be bounded by

$$\sum_{(v,\mu) \in \overline{\mathcal{S}}(\ell-1)} \mathcal{O}(|P(v)| + |P(\mu)|), \tag{3}$$

where $\overline{\mathcal{S}}(\ell-1)$ is the set of all pairs (v, μ) such that $v, \mu \in N(\ell-1)$ and

$$d(\mathcal{R}(v), \mathcal{R}(\mu)) < \frac{(s+1)\alpha}{2} \cdot \min\{\text{diam}(\mathcal{R}(v)), \text{diam}(\mathcal{R}(\mu))\}.$$

According to Lemma 3.8, summation (3) is $\mathcal{O}(\alpha^4(s+1)^2 \cdot n)$, which completes the proof of the lemma. \square

Lemma 3.10 *The SSPD of a set P of n points w.r.t. a constant s can be computed in $\mathcal{O}(s^2n + n \log n)$ time.*

Proof The BAR tree \mathcal{T} and the weight classes of nodes of \mathcal{T} require $\mathcal{O}(n \log n)$ time to compute [9]. Then we make a tree \mathcal{T}' from \mathcal{T} such that the level of each node in \mathcal{T}' represent its weight class. We do this by making the following changes in \mathcal{T} .

1. Remove all the nodes v with no weight class from the tree and connect the children of v (if exist) to the parent of v .
2. If a node appears in k weight classes ($k > 1$), then repeat it k times.

By Lemma 3.7, each node in the tree \mathcal{T}' has constant degree, and also the depth of the tree is $\mathcal{O}(\log n)$.

Using an algorithm similar to the algorithm for constructing a WSPD, we can construct an SSPD. That is, for each internal node v of the tree \mathcal{T}' , run the algorithm **FindPairs**(v_1, v_2), where v_1 and v_2 are the children of v . This algorithm tests whether the pair satisfies condition (ii) of the construction described just before Lemma 3.6. If they do, it reports the node pair. Otherwise it recurses on the children of v_1 and v_2 , i.e., for each child μ_1 of v_1 and each child μ_2 of v_2 , it calls **FindPairs**(μ_1, μ_2).

Via similar arguments as in the analysis for the WSPD, the running time of the algorithm can be shown to be $\mathcal{O}(m)$, where m is the size of the SSPD computed by the algorithm. \square

The following theorem summarizes the results on the SSPD construction.

Theorem 3.2 *Given a set P of n points in the plane and $s > 0$, we can compute an SSPD w.r.t. s of weight $\mathcal{O}(s^2n \log n)$ in time $\mathcal{O}(s^2n + n \log n)$.*

4 Fault-Tolerant Geodesic Spanners

We now consider the problem of constructing fault-tolerant geodesic $(1 + \varepsilon)$ -spanners for a set P of n points in the plane. Here we require that between any two points $u, v \in P$ outside the region fault F , there is a path in $\mathcal{G}_{\ominus F}$ whose length is at most t times the geodesic distance between u and v in $\mathbb{R}^2 \setminus F$. As remarked in the introduction, fault-tolerant geodesic spanners do not exist unless we are allowed to use Steiner points. As a simple example, consider a set $P = \{p, q\}$ of two points. A spanner \mathcal{G} without Steiner points would have to connect these points by an edge. However, this edge can be destroyed by an area fault F , leading to a situation where $d_{\mathcal{G}_{\ominus F}}(p, q) = \infty$, whereas the geodesic distance between p and q in $\mathbb{R}^2 \setminus F$ is finite. Even if we are allowed to add the Steiner points, it is easy to see that a finite-size fault-tolerant geodesic $(1 + \varepsilon)$ -spanner does not exist when \mathcal{F} is the family \mathcal{C} of all convex sets. To see this, consider two points with a convex but skinny region fault in between. The number of Steiner points depends on the aspect ratio of the region. Hence, we restrict the faults to the family \mathcal{D} of disks in the plane.

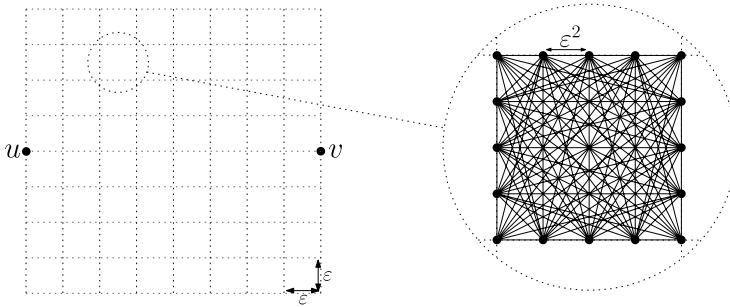


Fig. 8 The Steiner points added for u, v

Our method for constructing a fault-tolerant geodesic spanner works as follows. We first augment P with a set of $4(n - 1)$ Steiner points as described in Sect. 2. This way we can get an $\mathcal{O}(n/\varepsilon^2)$ size WSPD consisting of pairs (A, B) where the convex hull of both A and B have at most four vertices. Now fix a pair (A, B) . For every pair of points u, v , where u is a vertex of $\text{CH}(A)$ and v is a vertex of $\text{CH}(B)$, we will add a collection of $\mathcal{O}(1/\varepsilon^3)$ Steiner points with $\mathcal{O}(1/\varepsilon^4)$ edges between them to ensure the following: whenever both u and v are outside the fault disk D , there is a path connecting u and v through those Steiner points and outside D whose length is at most $(1 + \varepsilon)$ times the geodesic distance between u and v . This is sufficient because whenever there are points $p \in A$ and $q \in B$ outside D , there are convex hull points $u \in \text{CH}(A)$ and $v \in \text{CH}(B)$ outside D . Hence, we can go from p to u with a short path (by induction), then from u to v (by construction), and then from v to q (by induction).

Next we describe how to add Steiner points for the pair u, v . Without loss of generality we assume u and v are on a horizontal line at distance 1. Consider a unit square placed so that uv partitions it into two equal halves. We partition this square into a regular $(1/\varepsilon) \times (1/\varepsilon)$ grid whose cells have size $\varepsilon \times \varepsilon$ —we assume for simplicity that $1/\varepsilon$ is an integer—and we put $1/\varepsilon^2$ equally-spaced Steiner points on each grid line, as shown in Fig. 8. Notice that each grid cell has $1/\varepsilon$ Steiner points on each of its sides and that we have $\mathcal{O}(1/\varepsilon^3)$ Steiner points in total. For each grid cell, we add edges between every pair of Steiner points on its boundary, thus adding $\mathcal{O}(1/\varepsilon^2)$ edges per cell and $\mathcal{O}(1/\varepsilon^4)$ edges in total.

It remains to prove that for every pair of points, say u and v , we can always get a path whose length is close to their geodesic distance when we have a disk fault D . This can be argued as follows. Assume without loss of generality that the center of D lies below (or on) the line through u and v . Then the geodesic between u and v will go around D on the top side. Let D' be the disk with the same center as D but with radius $r + \sqrt{2}\varepsilon$, where r is the radius of D . Note that D' may contain u or v or both.

Let ∂D and $\partial D'$ denote the boundaries of D and D' , respectively. Because the diagonal of the grid cells is $\sqrt{2}\varepsilon$ and we have the same distance between D and D' , we have:

Observation 4.1 *No grid cell can intersect both the interior of D and $\partial D'$.*

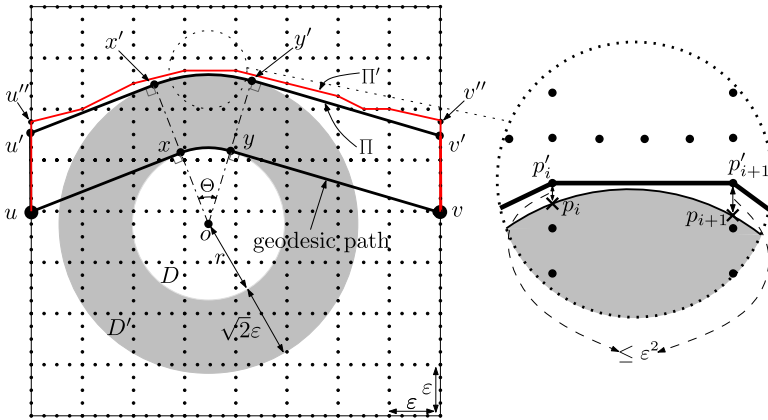


Fig. 9 The paths Π and Π'

The geodesic from u to v consists of a straight line segment connecting u to some point x on ∂D , followed by a circular arc along ∂D from x to some point y , denoted by \widehat{xy} , followed by a straight line segment connecting y to v . Draw two rays from o , the common center of D and D' , through x and y , and let x' and y' denote the points where these rays intersect $\partial D'$. Next, draw the lines tangent to D' at x' and y' —these are parallel to ux and yv , respectively—and let u' and v' be the intersections of these lines with the vertical lines through u and v , see Fig. 9. Finally, define Π to be the path consisting of the segments uu' and $u'x'$, followed by the arc along $\partial D'$ from x' to y' , followed by the segments $y'y'$ and $v'y'$.

From Π we construct a path Π' that uses the edges in our spanner. To this end, let p_1, p_2, \dots, p_k denote the intersection points of Π with the grid lines, ordered from u to v , see Fig. 9. Note that $u' = p_1$ and $v' = p_k$. For each intersection point p_i , let p'_i be the closest Steiner point above or on Π on the same grid line. We define Π' to be the path through p'_1, \dots, p'_k . Since each edge in Π' is inside a grid cell intersected by $\partial D'$, by Observation 4.1, the path Π' does not intersect D .

It remains to show that Π' approximates the geodesic distance. Firstly we show the length of the path Π , denoted by $\text{length}(\Pi)$, is roughly the same as the geodesic distance between u and v . Throughout this section we denote the (Euclidean) distance between points x and y by $|xy|$.

Lemma 4.1 *If γ is the geodesic distance between u and v , then*

$$\text{length}(\Pi) \leq (1 + (\pi + 2)\sqrt{2}\varepsilon)\gamma.$$

Proof By Fig. 9, it is clear that $|uu'| + |u'x'| \leq |ux| + \sqrt{2}\varepsilon$ and $|vv'| + |v'y'| \leq |vy| + \sqrt{2}\varepsilon$. Let $\Theta = \angle xoy$ be the smaller angle between ox and oy ; then we have

$$\begin{aligned} \text{length}(\widehat{x'y'}) &= (r + \sqrt{2}\varepsilon)\Theta \\ &= r\Theta + \sqrt{2}\varepsilon\Theta \end{aligned}$$

$$\begin{aligned}
 &= \text{length}(\widehat{xy}) + \sqrt{2}\varepsilon\Theta \\
 &\leq \text{length}(\widehat{xy}) + \sqrt{2}\pi\varepsilon.
 \end{aligned}$$

Therefore

$$\begin{aligned}
 \text{length}(\Pi) &= |uu'| + |u'x'| + \text{length}(x'y') + |y'v'| + |v'v| \\
 &\leq |ux| + \sqrt{2}\varepsilon + \text{length}(\widehat{xy}) + \sqrt{2}\pi\varepsilon + |vy| + \sqrt{2}\varepsilon \\
 &\leq \gamma + (2\sqrt{2} + \sqrt{2}\pi)\varepsilon \\
 &\leq (1 + (\pi + 2)\sqrt{2}\varepsilon)\gamma \quad (\text{since } \gamma \geq 1).
 \end{aligned}$$

Note that in the case where u or v , or both, lie inside D' , the same arguments can be used. □

Now we show that Π' approximates Π .

Lemma 4.2 *The part of the path Π between u' and v' intersect at most $(2 \cdot \text{length}(\Pi) + 1)/\varepsilon$ grid cells.*

Proof If the path Π intersects one cell in each column, we are done. Otherwise Π intersect at least two cells in the same column. It is easy to see that the length of the part of the path Π which lies inside any two consecutive cells in the column is at least ε . Therefore in total it can intersect at most $(2 \cdot \text{length}(\Pi) + 1)/\varepsilon$ cells. □

Lemma 4.3 $\text{length}(\Pi') \leq (1 + 6\varepsilon)\text{length}(\Pi)$.

Proof Let ℓ_i be the length of the part of Π inside the i th cell and (p'_i, p'_{i+1}) be the edge on Π' in the same cell. Then obviously $|p'_i p'_{i+1}| \leq \ell_i + 2\varepsilon^2$. So

$$\begin{aligned}
 \text{length}(\Pi') &= |up'_1| + \sum_{i=1}^{k-1} |p'_i p'_{i+1}| + |p'_k v| \\
 &\leq |uu'| + \varepsilon^2 + \sum_{i=1}^{k-1} (\ell_i + 2\varepsilon^2) + |v'v| + \varepsilon^2 \\
 &= \left(|uu'| + \sum_{i=1}^{k-1} \ell_i + |v'v| \right) + \sum_{i=1}^{k-1} 2\varepsilon^2 + 2\varepsilon^2 \\
 &\leq \text{length}(\Pi) + 2k\varepsilon^2 \\
 &\leq \text{length}(\Pi) + 2(2 \cdot \text{length}(\Pi) + 1)\varepsilon \quad (\text{Lemma 4.2}) \\
 &\leq (1 + 6\varepsilon)\text{length}(\Pi) \quad (\text{since } 1 \leq \text{length}(\Pi) \leq 2). \quad \square
 \end{aligned}$$

We obtain the following theorem.

Theorem 4.1 *For any set P of n points and any $\varepsilon > 0$, there exists a \mathcal{D} -fault tolerant geodesic Steiner $(1 + \varepsilon)$ -spanner of P with $\mathcal{O}(n/\varepsilon^6)$ edges that uses $\mathcal{O}(n/\varepsilon^5)$ Steiner points.*

Proof In the procedure of constructing a \mathcal{D} -fault tolerant spanner, we added $\mathcal{O}(n)$ Steiner points to have a WSPD of size $\mathcal{O}(n/\varepsilon^2)$ such that the convex hull of each set in the WSPD contains at most four points. Then for each pair (A, B) in the WSPD and for each pair of points on $\text{CH}(A)$ and $\text{CH}(B)$, we added $\mathcal{O}(1/\varepsilon^3)$ Steiner points and $\mathcal{O}(1/\varepsilon^4)$ edges between them. Therefore in total we added $\mathcal{O}(n/\varepsilon^5)$ Steiner points, and the graph contains $\mathcal{O}(n/\varepsilon^6)$ edges. \square

5 Concluding Remarks

We introduced the concept of region-fault tolerant spanners for planar point sets and proved the existence of region-fault tolerant spanners of small size. We showed that for any set of n points in the plane, we can construct a \mathcal{C} -fault tolerant spanner of size $\mathcal{O}(n \log n)$ in $\mathcal{O}(n \log^2 n)$ time. The main open problem is to determine whether our $\mathcal{O}(n \log n)$ bound on the spanner size for arbitrary point sets can be improved to $\mathcal{O}(n)$.

Our spanner construction for arbitrary point sets uses the SSPD, a relaxation of the WSPD. A similar variant with weaker properties, the SSD, was introduced by Varadarajan [29], who used it to solve min-cost perfect matching for points in the plane. His algorithm runs in \sqrt{n} phases, where each phase takes time proportional to the weight of the SSD plus the time it takes to compute the SSD. Since our SSPD satisfies the conditions of Varadarajan's SSD, and we can compute it more quickly, we can improve the running time the min-cost perfect matching algorithm from $\mathcal{O}(n^{3/2} \log^5 n)$ to $\mathcal{O}(n^{3/2} \log^2 n)$.

An interesting question is the following: given a $(1 + \varepsilon)$ -spanner \mathcal{G} with n vertices and m edges, how can we check whether it is \mathcal{C} -fault tolerant $(1 + \varepsilon)$ -spanner? It is not hard to see that the graph \mathcal{G} is \mathcal{C} -fault tolerant if it is \mathcal{H}' -fault tolerant, where \mathcal{H}' is a family of $\mathcal{O}(n^2)$ half-planes. Using a naïve approach for checking all the half-planes leads to an algorithm which in $\mathcal{O}(mn^3 + n^4 \log n)$ time using $\mathcal{O}(m)$ space can find the answer. It would be interesting to see if this can be improved.

In our approach, we used straight line segments as edges in our spanners. For geodesic spanners, it is interesting to see what happens if we are allowed to add curved edges.

Acknowledgements The authors would like to thank Michiel Smid for pointing out the lower bound on the weight of an SSPD and the anonymous referees for many insightful comments and suggestions.

References

1. Bollobás, B., Scott, A.: On separating systems. *Eur. J. Comb.* **28**(4), 1068–1071 (2007)
2. Callahan, P.B., Kosaraju, S.R.: Faster algorithms for some geometric graph problems in higher dimensions. In: *SODA'93: Proceedings of the 4th Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 291–300. Society for Industrial and Applied Mathematics, Philadelphia (1993)

3. Callahan, P.B., Kosaraju, S.R.: A decomposition of multidimensional point sets with applications to k -nearest-neighbors and n -body potential fields. *J. ACM* **42**, 67–90 (1995)
4. Chew, L.P.: There is a planar graph almost as good as the complete graph. In SCG'86: Proceedings of the 2nd Annual ACM Symposium on Computational Geometry, pp. 169–177 (1986)
5. Clarkson, K.L.: Approximation algorithms for shortest path motion planning. In STOC'87: Proceedings of the 19th Annual ACM Symposium on Theory of Computing, pp. 56–65 (1987)
6. Czumaj, A., Zhao, H.: Fault-tolerant geometric spanners. In: SCG'03: Proceedings of the 19th Annual ACM Symposium on Computational Geometry, pp. 1–10. ACM, New York (2003)
7. Das, G., Narasimhan, G.: A fast algorithm for constructing sparse Euclidean spanners. *Int. J. Comput. Geom. Appl.* **7**, 297–315 (1997)
8. de Berg, M., Cheong, O., van Kreveld, M., Overmars, M.: *Computational Geometry: Algorithms and Applications*, 3rd edn. Springer, Berlin (2008)
9. Duncan, C.A., Goodrich, M.T., Kobourov, S.: Balanced aspect ratio trees: Combining the advances of k -d trees and octrees. *J. Algorithms* **38**, 303–333 (2001)
10. Eppstein, D.: Spanning trees and spanners. In: Sack, J.-R., Urrutia, J. (eds.) *Handbook of Computational Geometry*, pp. 425–461. Elsevier, Amsterdam (2000)
11. Fischer, J., Har-Peled, S.: Dynamic well-separated pair decomposition made easy. In CCCG'05: Proceedings of the 17th Canadian Conference on Computational Geometry, pp. 235–238 (2005)
12. Gudmundsson, J., Knauer, C.: Dilatation and detour in geometric networks. In: Gonzalez, T. (ed.) *Handbook on Approximation Algorithms and Metaheuristics*. Chapman & Hall, London (2007)
13. Hansel, G.: Nombre minimal de contacts de fermeture nécessaires pour réaliser une fonction booléenne symétrique de n variables. *C. R. Acad. Sci. Paris* **258**, 6037–6040 (1964). Russian transl., *Kibern. Sb. (Nov. Ser.)* **5**, 47–52 (1968)
14. Har-Peled, S.: On the expected complexity of random convex hulls. Technical Report 330/98, School Math. Sci., Tel-Aviv Univ., Tel-Aviv, Israel (1998)
15. Keil, J.M.: Approximating the complete Euclidean graph. In: SWAT'88: Proceedings of the 1st Scandinavian Workshop on Algorithm Theory. Lecture Notes in Computer Science, vol. 318, pp. 208–213. Springer, Berlin (1988)
16. Levcopoulos, C., Narasimhan, G., Smid, M.: Improved algorithms for constructing fault-tolerant spanners. *Algorithmica* **32**, 144–156 (2002)
17. Lukovszki, T.: New results of fault tolerant geometric spanners. In: WADS'99: Proceedings of the 6th Workshop on Algorithms and Data Structures. Lecture Notes in Computer Science, vol. 1663, pp. 193–204. Springer, Berlin (1999)
18. Narasimhan, G., Smid, M.: *Geometric Spanner Networks*. Cambridge University Press, Cambridge (2007)
19. Peleg, D., Schäffer, A.: Graph spanners. *J. Graph Theory* **13**, 99–116 (1989)
20. Preparata, F.P., Shamos, M.I.: *Computational Geometry: An Introduction*. Springer, New York (1985)
21. Rényi, A., Sulanke, R.: Über die konvexe hülle von n zufällig gewählten punkten I. *Z. Wahrscheinlichkeitstheor. Verw. Geb.* **2**, 75–84 (1963)
22. Ross, S.: *A First Course in Probability*, 5th edn. Prentice-Hall, New York (1998)
23. Salowe, J.S.: Constructing multidimensional spanner graphs. *Int. J. Comput. Geom. Appl.* **1**, 99–107 (1991)
24. Salowe, J.S.: Enumerating interdistances in space. *Int. J. Comput. Geom. Appl.* **2**(1), 49–59 (1992)
25. Smid, M.: Closest point problems in computational geometry. In: Sack, J.-R. (ed.) *Handbook of Computational Geometry*, pp. 877–935. Elsevier, Amsterdam (2000)
26. Vaidya, P.M.: Minimum spanning trees in k -dimensional space. *SIAM J. Comput.* **17**(3), 572–582 (1988)
27. Vaidya, P.M.: An $O(n \log n)$ algorithm for the all-nearest-neighbors problem. *Discrete Comput. Geom.* **4**, 101–115 (1989)
28. Vaidya, P.M.: A sparse graph almost as good as the complete graph on points in K dimensions. *Discrete Comput. Geom.* **6**(4), 369–381 (1991)
29. Varadarajan, K.R.: A divide-and-conquer algorithm for min-cost perfect matching in the plane. In FOCS'98: Proceedings of the 39th Annual IEEE Symposium on Foundations of Computer Science, pp. 320–331 (1998)