Title:	Geometric Spanners
Name:	Joachim Gudmundsson[1], Giri Narasimhan[2], Michiel Smid[3]
Affil./Addr. 1:	School of Information Technologies, The University of Sydney, NSW, Australia
Affil./Addr. 2:	School of Computing & Information Sciences Florida International University, FL, USA
Affil./Addr. 3:	School of Computer Science Carleton University, ON, Canada
Keywords:	Computational geometry; geometric networks; spanners; dilation
SumOriWork:	2002; Gudmundsson, Levcopoulos, Narasimhan

Geometric Spanners

Joachim Gudmundsson[1], Giri Narasimhan[2], Michiel Smid[3]

[1] School of Information Technologies, The University of Sydney, NSW, Australia
[2] School of Computing & Information Sciences Florida International University, FL, USA
[3] School of Computer Science Carleton University, ON, Canada

Years aud Authors of Summarized Original Work

2002; Gudmundsson, Levcopoulos, Narasimhan

Keywords

Computational geometry; geometric networks; spanners; dilation

Problem Definition

Consider a set S of n points in d-dimensional Euclidean space. A network on S can be modeled as an undirected graph G with vertex set S of size n and an edge set E where every edge (u, v) has a weight. A geometric (Euclidean) network is a network where the weight of the edge (u, v) is the Euclidean distance $|uv|$ between its endpoints. Given a real number $t > 1$ we say that G is a *t-spanner* for S, if for each pair of points $u, v \in S$, there exists a path in G of weight at most t times the Euclidean distance between u and v. The minimum t such that G is a t-spanner for S is called the stretch factor, or dilation, of G. For a detailed description of many constructions of t-spanners, see the book by Narasimhan and Smid [30]. The problem considered is the construction of t-spanners given a set S of n points in $\mathcal{R}^d$ and a positive real value $t > 1$, where d is a constant. The aim is to compute a good t-spanner for S with respect to the following quality measures:

size: the number of edges in the graph.

degree: the maximum number of edges incident on a vertex.
weight: the sum of the edge weights.
spanner diameter: the smallest integer k such that for any pair of vertices u and v in S, there is a path in the graph of length at most $t \cdot |uv|$ between u and v containing at most k edges.
fault-tolerance: the resilience of the graph to edge, vertex or region failures.

Thus, good t-spanners require large fault-tolerance and small size, degree, weight and spanner diameter. Additionally, the time required to compute such spanners must be as small as possible.

Key Results

This section contains descriptions of several known approaches for constructing a t-spanner of a set of points in Euclidean space. We also present descriptions of the construction of fault-tolerant spanners, spanners among polygonal obstacles and, finally, a short note on dynamic and kinetic spanners.

Spanners of points in Euclidean space

The most well-known classes of t-spanner networks for points in Euclidean space include: Θ-graphs, WSPD-graphs and Greedy-spanners. In the following sections the main idea of each of these classes is given, together with the known bounds on the quality measures.

The Θ-graph The Θ-graph was discovered independently by Clarkson and Keil in the late 80's. The general idea is to process each point $p \in S$ independently as follows: partition $\mathcal{R}^d$ into k simplicial cones of angular diameter at most θ and apex at p, where $k = O(1/\theta^{d-1})$. For each non-empty cone C, an edge is added between p and the point in C whose orthogonal projection onto some fixed ray in C emanating from p is closest to p, see Fig. 1a. The resulting graph is called the Θ-graph on S. The following result is due to Arya et al. [9].

Theorem 1. *The Θ-graph is a t-spanner of S for $t = \frac{1}{\cos\theta - \sin\theta}$ with $O(\frac{n}{\theta^{d-1}})$ edges and can be computed in $O(\frac{n}{\theta^{d-1}} \log^{d-1} n)$ time using $O(\frac{n}{\theta^{d-1}} + n \log^{d-2} n)$ space.*

The following variants of the Θ-graph also give bounds on the degree, spanner diameter, and weight.

Skip-list spanners: The idea is to generalize skip-lists and apply them to the construction of spanners. Construct a sequence of h subsets, $S_1, \ldots, S_h$, where $S_1 = S$ and S_i is constructed from S_{i-1} as follows (reminiscent of the levels in a skip list). For each point in S_{i-1}, flip a fair coin. The set S_i is the set of all points of S_{i-1} whose coin flip produced heads. The construction stops if $S_i = \emptyset$. For each subset a Θ-graph is constructed. The union of the graphs is the skip-list spanner of S with dilation t, having $O(\frac{n}{\theta^{d-1}})$ edges and $O(\log n)$ spanner diameter with high probability [9].

Gap-greedy: A set of directed edges is said to satisfy the *gap* property if the sources of any two distinct edges in the set are separated by a distance that is at least proportional to the length of the shorter of the two edges. Arya and Smid [6] proposed an algorithm that uses the gap property to decide whether or not an edge should be added to the t-spanner graph. Using the gap property the constructed spanner can be shown to have degree $O(1/\theta^{d-1})$ and weight $O(\log n \cdot wt(MST(S)))$, where $wt(MST(S))$ is the weight of the minimum spanning tree of S.

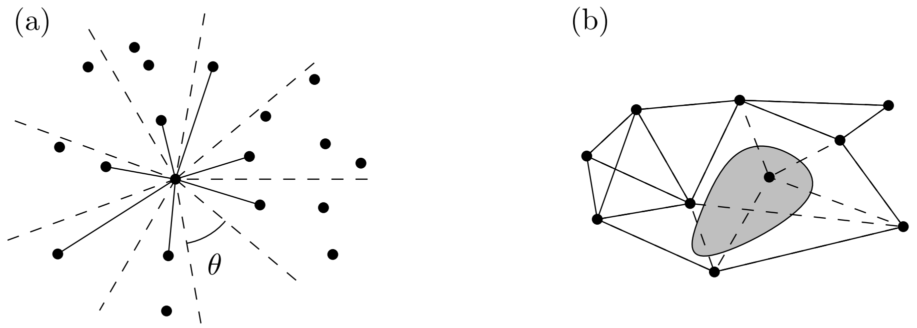

Fig. 1. (a) Illustrating the Θ-graph, and (b) a graph with a region-fault.

The WSPD-graph The well-separated pair decomposition (WSPD) was developed by Callahan and Kosaraju [12]. The construction of a t-spanner using the well-separated pair decomposition is done by first constructing a WSPD of S with respect to a separation constant $s = \frac{4(t+1)}{(t-1)}$. Initially set the spanner graph $G = (S, \emptyset)$ and add edges iteratively as follows. For each well-separated pair $\{A, B\}$ in the decomposition, an edge (a, b) is added to the graph, where a and b are arbitrary points in A and B, respectively. The resulting graph is called the WSPD-graph on S.

Theorem 2. *The WSPD-graph is a t-spanner for S with $O(s^d \cdot n)$ edges and can be constructed in time $O(s^d n + n \log n)$, where $s = 4(t+1)/(t-1)$.*

There are modifications that can be made to obtain bounded spanner diameter or bounded degree.

Bounded spanner diameter: Arya, Mount and Smid [7] showed how to modify the construction algorithm such that the spanner diameter of the graph is bounded by $2 \log n$. Instead of selecting an arbitrary point in each well-separated set, their algorithm carefully chooses a representative point for each set.

Bounded degree: A single point v can be part of many well-separated pairs and each of these pairs may generate an edge with an endpoint at v. Arya et al. [8] suggested an algorithm that retains only the shortest edge for each cone direction, thus combining the Θ-graph approach with the WSPD-graph. By adding a post-processing step that handles all high-degree vertices, a t-spanner of degree $O(\frac{1}{(t-1)^{2d-1}})$ is obtained.

The Greedy-spanner The greedy algorithm was first presented in 1989 by Bern and since then the greedy algorithm has been subject to considerable research. The graph constructed using the greedy algorithm is called a Greedy-spanner and the general idea is that the algorithm iteratively builds a graph G. The edges in the complete graph are processed in order of increasing edge length. Testing an edge (u, v) entails a shortest path query in the partial spanner graph G. If the shortest path in G between u and v is at most $t \cdot |uv|$ then the edge (u, v) is discarded, otherwise it is added to the partial spanner graph G.

Das, Narasimhan and Salowe [22] proved that the greedy-spanner fulfills the so-called *leapfrog property*. A set of undirected edges E is said to satisfy the t-leapfrog property, if for every $k \geq 2$, and for every possible sequence $\{(p_1, q_1), \ldots, (p_k, q_k)\}$ of pairwise distinct edges of E,

$$t \cdot |p_1 q_1| < \sum_{i=2}^{k} |p_i q_i| + t \cdot \Big(\sum_{i=1}^{k-1} |q_i p_{i+1}| + |p_k q_1| \Big).$$

Using the leapfrog property, it has been shown that the total edge weight of the graph is within a constant factor of the weight of a minimum spanning tree of S.

Using Dijkstra's shortest-path algorithm, the greedy-spanner can be constructed in $O(n^3 \log n)$ time. Bose et al. [10] improved the time to $O(n^2 \log n)$, while using $O(n^2)$ space. Alewijnse et al. [4] improved the space bound to $O(n)$, while slightly increasing the time bound to $O(n^2 \log^2 n)$.

Das and Narasimhan [21] observed that an approximation of the Greedy-spanner can be constructed while maintaining the leapfrog property. This observation allowed for faster construction algorithms.

Theorem 3. *[27] The greedy-spanner is a t-spanner of S with $O(\frac{n}{(t-1)^d} \log(\frac{1}{t-1}))$ edges, maximum degree $O(\frac{1}{(t-1)^d} \log(\frac{1}{t-1}))$, weight $O(\frac{1}{(t-1)^{2d}} \cdot wt(MST(S))$, and can be computed in time $O(\frac{n}{(t-1)^{2d}} \log n)$.*

The transformation technique Chandra et al. [16; 17] introduced a transformation technique for general metrics that transforms an algorithm for constructing spanners with small stretch factor and size into an algorithm for constructing spanners with the same asymptotic stretch factor and size, but with the additional feature of small weight. Elkin and Solomon [24] refined their approach to develop a transformation technique that achieved the following: It takes an algorithm for constructing spanners with small stretch factor, small size, small degree, and small spanner diameter, and transforms it into an algorithm for constructing spanners with the a small increase in stretch factor, size, degree, and spanner diameter, but that also has small weight and running time.

Using the transformation technique allowed Elkin and Solomon to prove the following theorem.

Theorem 4. *[24] For any set of n points in Euclidean space of any constant dimension d, any $\epsilon > 0$ and any parameter $\rho \geq 2$, there exists a $(1+\epsilon)$-spanner with $O(n)$ edges, degree $O(\rho)$, spanner diameter $O(\log_\rho n + \alpha(\rho))$ and weight $O(\rho \cdot \log_\rho n \cdot wt(MST))$, which can be constructed in time $O(n \log n)$.*

Given the lower bounds proved by Chan and Gupta [13] and Dinitz et al. [23], these results represent optimal tradeoffs in the entire range of the parameter ρ.

Fault-tolerant spanners

The concept of fault-tolerant spanners was first introduced by Levcopoulos et al. [28] in 1998: 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. Czumaj and Zhao [19] showed that a greedy approach produces a k-vertex (or k-edge) fault tolerant geometric t-spanner with degree $O(k)$ and total weight $O(k^2 \cdot wt(MST(S)))$; these bounds are asymptotically optimal. Chan et al. [15] used a "standard net-tree with cross-edge framework" developed by [14; 26], to design an algorithm that produces a k-vertex (or k-edge) fault tolerant geometric $(1+\epsilon)$-spanner with degree $O(k^2)$, diameter $O(\log n)$, and total weight $O(k^2 \log n \cdot wt(MST(S)))$. Such a spanner can be constructed in $O(n \log n + k^2 n)$ time.

For geometric spanners it is natural to consider *region faults*, i.e., faults that destroy all vertices and edges intersecting some geometric fault region. For a fault region F, let $G \ominus F$ be the part of G that remains after the points from S inside F and all edges that intersect F have been removed from the graph, see Fig. 1b. Abam et al. [2] showed how to construct region-fault tolerant t-spanners of size $O(n \log n)$ that are fault-tolerant to any convex region-fault. If one is allowed to use Steiner points then a linear size t-spanner can be achieved.

Spanners among obstacles

The visibility graph of a set of pairwise non-intersecting polygons is a graph of intervisible locations. Each polygonal vertex is a vertex in the graph and each edge represents a visible connection between them, that is, if two vertices can see each other, an edge is drawn between them. This graph is useful since it contains the shortest obstacle avoiding path between any pair of vertices.

Das [20] showed that a t-spanner of the visibility graph of a point set in the Euclidean plane can be constructed by using the Θ-graph approach followed by a pruning step. The obtained graph has linear size and constant degree.

Dynamic and kinetic spanners

Arya et al. [9] designed a data structure of size $O(n \log^d n)$ that maintains the skip-list spanner, described in Section , in $O(\log^d n \log\log n)$ expected amortized time per insertion and deletion in the model of random updates.

Gao et al. [26] showed how to maintain a t-spanner of size $O(\frac{n}{(t-1)^d})$ and maximum degree $O(\frac{1}{(t-2)^d} \log \alpha)$, in time $O(\frac{\log \alpha}{(t-1)^d})$ per insertion and deletion, where α denotes the aspect ratio of S, i.e., the ratio of the maximum pairwise distance to the minimum pairwise distance. The idea is to use an hierarchical structure T with $O(\log \alpha)$ levels, where each level contains a set of centers (subset of S). Each vertex v on level i in T is connected by an edge to all other vertices on level i within distance $O(\frac{2^i}{t-1})$ of v. The resulting graph is a t-spanner of S and it can be maintained as stated above. The approach can be generalized to the kinetic case so that the total number of events in maintaining the spanner is $O(n^2 \log n)$ under pseudo-algebraic motion. Each event can be updated in $O(\frac{\log \alpha}{(t-1)^d})$ time.

The problem of maintaining a spanner under insertions and deletions of points was settled by Gottlieb and Roditty [5]: For every set of n points in a metric space of bounded doubling dimension, there exists a $(1+\epsilon)$-spanner whose maximum degree is $O(1)$ and that can be maintained under insertions and deletions of points, in $O(\log n)$ time per operation.

Recently several papers have considered the kinetic version of the spanner construction problem. Abam et al. [1; 3] gave the first data structures for maintaining the Θ-graph, which was later improved by Rahmati et al. [32]. Assuming the trajectories of the points can be described by polynomials whose degrees are at most a constant s, the data structure uses $O(n \log^d n)$ space and handles $O(n^2)$ events with a total cost of $O(n\lambda_{2s+2}(n) \log^{d+1} n)$, where $\lambda_{2s+2}(n)$ is the maximum length of Davenport-Schinzel sequences of order $2s+2$ on n symbols. The kinetic data structure is compact, efficient, responsive (in an amortized sense), and local.

Applications

The construction of sparse spanners has been shown to have numerous application areas such as metric space searching [31], which includes query by content in multimedia objects, text retrieval, pattern recognition and function approximation. Another example is broadcasting in communication networks [29]. Several well-known theoretical results also use the construction of t-spanners as a building block, for example, Rao and Smith [33] made a breakthrough by showing an optimal $O(n \log n)$-time approximation scheme for the well-known Euclidean *traveling salesperson problem*, using t-spanners (or banyans). Similarly, Czumaj and Lingas [18] showed approximation schemes for minimum-cost multi-connectivity problems in geometric networks.

Open Problems

A few open problems are mentioned below:

1. Determine if there exists a fault-tolerant t-spanner of linear size for convex region faults.
2. Can the k-vertex fault tolerant spanner be computed in $O(n \log n + kn)$ time?

Experimental Results

The problem of constructing spanners has received considerable attention from a theoretical perspective but not much attention from a practical, or experimental perspective. Navarro and Paredes [31] presented four heuristics for point sets in high-dimensional space ($d = 20$) and showed by empirical methods that the running time was $O(n^{2.24})$ and the number of edges in the produced graphs was $O(n^{1.13})$. Farshi and Gudmundsson [25] performed a thorough comparison of the construction algorithms discussed in Section . The results showed that the spanner produced by the original greedy algorithm is superior compared to the graphs produced by the other approaches discussed in Section when it comes to number of edges, maximum degree and weight. However, the greedy algorithm requires $O(n^2 \log n)$ time [10] and uses quadratic space, which restricted experiments in [25] to instances containing at most 13,000 points. Alewijnse et al. [4] showed how to reduce the space usage to linear only paying an additional $O(\log n)$ factor in the running time. In their experiments they could handle more than a million points. In a follow-up paper Bouts et al. [11] gave further experimental improvements.

Cross-References

Well Separated Pair Decomposition
Applications of Geometric Spanners
Dilation of Geometric Networks
Single-Source Shortest Paths
Sparse Graph Spanners
Algorithms for Spanners in Weighted Graphs
Approximating Metric Spaces by Tree Metrics
Algorithms for Spanners in Weighted Graphs
Degree-Bounded Planar Spanner with Low Weight

Recommended Reading

1. Abam MA, de Berg M (2011) Kinetic spanners in $\mathbb{R}^d$. Discrete & Computational Geometry 45(4):723–736
2. Abam MA, de Berg M, Farshi M, Gudmundsson J (2009) Region-fault tolerant geometric spanners. Discrete & Computational Geometry 41:556–582
3. Abam MA, de Berg M, Gudmundsson J (2010) A simple and efficient kinetic spanner. Comput Geom 43(3):251–256
4. Alewijnse SPA, Bouts QW, ten Brink AP, Buchin K (2013) Computing the greedy spanner in linear space. In: Bodlaender HL, Italiano GF (eds) 21st Annual European Symposium on Algorithms, Springer, Lecture Notes in Computer Science, vol 8125, pp 37–48

5. Arikati SR, Chen DZ, Chew LP, Das G, Smid M, Zaroliagis CD (1996) Planar spanners and approximate shortest path queries among obstacles in the plane. In: Proc 4th European Symposium on Algorithms, Lecture Notes in Computer Science, vol 1136, pp 514–528
6. Arya S, Smid M (1997) Efficient construction of a bounded-degree spanner with low weight. Algorithmica 17:33–54
7. Arya S, Mount DM, Smid M (1994) Randomized and deterministic algorithms for geometric spanners of small diameter. In: Proc 35th IEEE Symposium on Foundations of Computer Science, pp 703–712
8. Arya S, Das G, Mount DM, Salowe JS, Smid M (1995) Euclidean spanners: short, thin, and lanky. In: Proc 27th ACM Symposium on Theory of Computing, pp 489–498
9. Arya S, Mount DM, Smid M (1999) Dynamic algorithms for geometric spanners of small diameter: Randomized solutions. Computational Geometry - Theory and Applications 13(2):91–107
10. Bose P, Carmi P, Farshi M, Maheshwari A, Smid MHM (2010) Computing the greedy spanner in near-quadratic time. Algorithmica 58(3):711–729
11. Bouts QW, ten Brink AP, Buchin K (2014) A framework for computing the greedy spanner. In: Cheng SW, Devillers O (eds) Symposium on Computational Geometry, ACM, pp 11–20
12. Callahan PB, Kosaraju SR (1995) A decomposition of multidimensional point sets with applications to k-nearest-neighbors and n-body potential fields. Journal of the ACM 42:67–90
13. Chan HTH, Gupta A (2009) Small hop-diameter sparse spanners for doubling metrics. Discrete & Computational Geometry 41(1):28–44
14. Chan HTH, Gupta A, Maggs B, Zhou S (2005) On hierarchical routing in doubling metrics. In: Symposium on Discrete Algorithms, ACM, pp 762–771
15. Chan HTH, Li M, Ning L, Solomon S (2013) New doubling spanners: Better and simpler. In: Automata, Languages, and Programming, Springer Berlin Heidelberg, pp 315–327
16. Chandra B, Das G, Narasimhan G, Soares J (1992) New sparseness results on graph spanners. In: Proc 8th Annual Symposium on Computational Geometry, pp 192–201
17. Chandra B, Das G, Narasimhan G, Soares J (1995) New sparseness results on graph spanners. International Journal of Computational Geometry and Applications 5:124–144
18. Czumaj A, Lingas A (2000) Fast approximation schemes for Euclidean multi-connectivity problems. In: Proc 27th International Colloquium on Automata, Languages and Programming, Springer-Verlag, Lecture Notes in Computer Science, vol 1853, pp 856–868
19. Czumaj A, Zhao H (2004) Fault-tolerant geometric spanners. Discrete & Computational Geometry 32(2):207–230
20. Das G (1997) The visibility graph contains a bounded-degree spanner. In: Proc 9th Canadian Conference on Computational Geometry
21. Das G, Narasimhan G (1997) A fast algorithm for constructing sparse Euclidean spanners. International Journal of Computational Geometry and Applications 7:297–315
22. Das G, Narasimhan G, Salowe J (1995) A new way to weigh malnourished Euclidean graphs. In: Proc 6th ACM-SIAM Symposium on Discrete Algorithms, pp 215–222
23. Dinitz Y, Elkin M, Solomon S (2010) Low-light trees, and tight lower bounds for Euclidean spanners. Discrete & Computational Geometry 43(4):736–783
24. Elkin M, Solomon S (2013) Optimal euclidean spanners: really short, thin and lanky. In: Proceedings of the forty-fifth annual ACM symposium on Theory of computing, pp 645–654
25. Farshi M, Gudmundsson J (2009) Experimental study of geometric t-spanners. ACM Journal of Experimental Algorithmics 14(1):3–39
26. Gao J, Guibas LJ, Nguyen A (2004) Deformable spanners and applications. In: Proc 20th ACM Symposium on Computational Geometry, pp 190–199
27. Gudmundsson J, Levcopoulos C, Narasimhan G (2002) Improved greedy algorithms for constructing sparse geometric spanners. SIAM Journal on Computing 31(5):1479–1500
28. Levcopoulos C, Narasimhan G, Smid M (2002) Improved algorithms for constructing fault-tolerant spanners. Algorithmica 32(1):144–156
29. Li XY (2003) Applications of computational geomety in wireless ad hoc networks. In: Cheng XZ, Huang X, Du DZ (eds) Ad Hoc Wireless Networking, Kluwer, pp 197–264
30. Narasimhan G, Smid M (2007) Geometric spanner networks. Cambridge University Press
31. Navarro G, Paredes R (2003) Practical construction of metric t-spanners. In: Proc 5th Workshop on Algorithm Engineering and Experiments, SIAM Press, pp 69–81
32. Rahmati Z, Abam MA, King V, Whitesides S (2014) Kinetic data structures for the semi-Yao graph and all nearest neighbors in $\mathbb{R}^d$. In: He M, Zeh N (eds) Canadian Conference on Computational Geometry

33. Rao S, Smith WD (1998) Approximating geometrical graphs via spanners and banyans. In: Proc 30th ACM Symposium on Theory of Computing, pp 540–550