

Distribution-sensitive construction of the greedy spanner (extended abstract)

Citation for published version (APA):

Alewijnse, S. P. A., Bouts, Q. W., Brink, ten, A. P., & Buchin, K. (2014). Distribution-sensitive construction of the greedy spanner (extended abstract). In *30th European Workshop on Computational Geometry (EuroCG 2014, Ein-Gedi, Israel, March 3-5, 2014)* (pp. 1-4)

Document status and date:

Published: 01/01/2014

Document Version:

Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy

If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.

Distribution-Sensitive Construction of the Greedy Spanner

Sander P. A. Alewijnse*

Quirijn W. Bouts*

Alex P. ten Brink*

Kevin Buchin*

Abstract

The greedy spanner is the highest quality geometric spanner (in e.g. edge count and weight, both in theory and practice) known to be computable in polynomial time. Unfortunately, all known algorithms for computing it take $\Omega(n^2)$ time, limiting its applicability on large data sets.

We observe that for many point sets, the greedy spanner has many ‘short’ edges that can be determined locally and usually quickly, and few or no ‘long’ edges that can usually be determined quickly using local information and the well-separated pair decomposition. We give experimental results showing large to massive performance increases over the state-of-the-art on nearly all tests and real-life data sets. On the theoretical side we prove a near-linear expected time bound on uniform point sets and a near-quadratic worst-case bound.

Our bound for point sets drawn uniformly and independently at random in a square follows from a local characterization of t -spanners we give on such point sets.

This characterization gives a $O(n \log^2 n \log^2 \log n)$ expected time bound on our greedy spanner algorithm, making it the first subquadratic time algorithm for this problem on any interesting class of points.

1 Introduction

A *Euclidean graph* on a set of points in the Euclidean plane is a weighted graph with geometric distances as edge weights. If a shortest route in the graph is at most t times longer than the direct geometric distance between its endpoints, we say these endpoints *have a t -path*: a Euclidean graph is a t -spanner - a spanner with *dilation* t - if all pairs of points have t -paths. For any $t > 1$, we can efficiently find a t -spanner with $O\left(\frac{n}{t-1}\right)$ edges in the Euclidean plane [8]. These ‘approximations’ thus have very few edges compared to the complete Euclidean graph, while approximately maintaining distances. This makes them a useful tool in many areas.

A considerable amount of research has been done on the topic of spanners [8] since they were introduced in network design [9] and in geometry [6]. Spanners have been used as components in various geometric and distributed algorithms.

We focus on the greedy spanner, which is a very sparse graph with degree and weight bounds. On uniform point

sets and a dilation of 2, one of its closest well-known competitors, the Θ -graph, has about ten times as many edges, twenty times higher total weight and six times higher maximum degree. Unfortunately, all known algorithms computing the greedy spanner use $\Omega(n^2)$ time [1, 4], making the spanner impractical to compute on large point sets. Algorithms with similar asymptotic behavior and lower asymptotic construction time exist (see [8]), but the greedy spanner stands out because of its favorable dependency on t , which results in its unique strong performance in practice.

We observed that on real-world examples, the greedy spanner contains mostly short edges with at most a few longer edges. Whether an edge is placed depends only on the points and edges in an ellipse with its endpoints as foci and with eccentricity t , which is a small area for short potential edges, hopefully containing few points. We can therefore find these short edges using a bucketing scheme, giving a speedup on such point sets.

For the ‘long’ edges, we consider the ‘long’ well-separated pairs from a WSPD [5]. We first compute information from the ‘short’ edges, attempting to find witnesses that show that certain ‘long’ well-separated pairs will not contain greedy spanner edges. We then perform a standard algorithm [1] on the (hopefully only few) well-separated pairs for which we cannot find such a witness.

We present experimental results showing that the above algorithm works very well on many data sets.

We show that our algorithm has a near-quadratic worst-case time bound. In order to explain its behavior observed in experiments on realistic point sets, we analyze its performance on point sets distributed uniformly and independently at random in a square (or ‘uniformly distributed points’ for short).

In this paper, we look at the ‘global’ property t -spannerness and the greedy spanner, a graph for which the existence of an edge may depend on all other points. Previous analysis techniques do not directly apply on such properties. However, one of our main contributions is to show that with high probability, greedy spanners do admit a local characterization on uniform point sets.

We consider n points distributed uniformly and independently at random in a $\sqrt{n} \times \sqrt{n}$ square. We use a $\sqrt{n} \times \sqrt{n}$ scale for our square so that if a part of this square has area A , then $O(A)$ points lie in it in expectation. We only consider the case of the Euclidean plane – our results may generalize to higher dimensions, but we did not explore this. In this introduction, when stating bounds, we assume t is a constant.

*Eindhoven University of Technology, The Netherlands. Q. W. Bouts and K. Buchin are supported by the Netherlands Organisation for Scientific Research (NWO) under project no. 639.023.208 and 612.001.207 respectively. {q.w.bouts,k.a.buchin}@tue.nl

We prove that such point sets are, with high probability, configured in such a way that for any edge E , if there are t -paths between points at most $O(\log n)$ away from each other, then there are t -paths between all points. In particular, we show that we can construct a ‘witness’ of this configuration in $O(n \log^2 n \log \log n)$ expected time if it exists, thus allowing our algorithms to always give the right answer.

This result easily implies that with high probability the greedy spanner has no long edges (longer than $O(\log n)$) and furthermore that the ‘proof’ phase of our algorithm will find the witnesses for this if it exists. As the grid strategy works well on uniformly distributed point sets, we obtain a $O(n \log^2 n \log^2 \log n)$ expected time bound on our algorithm. To the best of our knowledge, this algorithm is the first subquadratic algorithm to compute the greedy spanner on any interesting class of point sets.

The rest of the paper is organized as follows. In Section 2 we introduce bridgedness. In Section 3 we show uniform point sets are locally- $O(\log n)$ -bridged with high probability. In Section 4 we give a fast algorithm that uses this result. Finally, in Section 5 we present experimental results. For the full proofs we refer to [2].

2 Bridging Points

In this section we will introduce the concept of λ -bridgedness for point sets. We will later use this concept in our characterization of t -spanners on uniformly distributed point sets. We will prove two geometric lemmas that will help us with the result of Section 3.

Let P be a finite set of points in \mathbb{R}^2 , let $n = |P|$, and let $t \in \mathbb{R}$ be the intended dilation ($t > 1$). Let $G = (P, E)$ be a graph on P whose edges are weighted with the Euclidean distance between its endpoints. For two points $u, v \in P$, we denote the Euclidean distance between u and v by $|uv|$, and the network distance in G by $\delta_G(u, v)$ (or just $\delta(u, v)$ if G is clear from the context). We say a pair of points (u, v) has a t -path if $\delta(u, v) \leq t \cdot |uv|$. If all pairs of points have a t -path, the graph is called a t -spanner.

Let $a, b, p, q \in P$ be pairwise different points. We say that the pair (p, q) bridges the pair (a, b) if $t \cdot |ap| + |pq| + t \cdot |qb| \leq t \cdot |ab|$. Bridging points guarantee a t -path for (a, b) if (p, q) is an edge and the pairs (a, p) and (q, b) already have t -paths (as well as $|ap|, |qb|, |pq| < |ab|$).

We say that (p, q) is *mandatory* if the ellipse with foci p and q and eccentricity t including its border contains no points in P other than p and q . Any t -path between p and q must fully lie within this ellipse, so if (p, q) is mandatory then $(p, q) \in E$ for any t -spanner E on P .

Let $\lambda \in \mathbb{R}$. We say that a point $a \in P$ is λ -bridged if for all $b \in P$ with $|ab| > \lambda$, there exist some mandatory pair of points (p, q) , $p, q \in P$, bridging (a, b) . We say that the point set P is λ -bridged if all points in P are λ -bridged. We say a point $a \in P$ is *locally- λ -bridged* if it is λ -bridged using only mandatory bridging pairs of points at most λ from a . A point set P is *locally- λ -bridged* if

all points in P are locally- λ -bridged. Lemma 1 shows the usefulness of this concept.

Lemma 1 *Let P be a set of points that is λ -bridged. For any Euclidean graph $G = (P, E)$ it holds that G is a t -spanner if and only if all pairs of points (a, b) , $a, b \in P$, with $|ab| \leq \lambda$ have a t -path in G .*

Lemma 2 *Suppose we are given points $a, b \in P$, rectangles R_1 and R_2 and $t > 1$, such that (as per Fig. 1): R_1 and R_2 lie in between a and b , have a side parallel to ab , have their centers on line segment ab , both have width w and height h , are separated by $s \geq \frac{t+1}{t-1}h$ and R_1 lies closer to a than R_2 .*

Then, for any $p, q \in P$ with p lying in R_1 and q lying in R_2 , (p, q) bridges (a, b) .

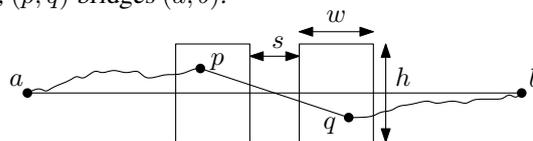


Figure 1: (p, q) bridges (a, b)

We now use Lemma 2 to prove a stronger statement that we will use to prove the full version of Theorem 4. Let $a, p, q \in P$ be pairwise different points and let $A \subseteq \mathbb{R}^2$ with $a, p, q \notin A$. We say that the pair (p, q) bridges (a, A) if for every point $b \in P$ with $b \in A$ we have that (p, q) bridges (a, b) .

Lemma 3 *Assume we are given $a \in P$, a line ℓ through a , an angle $\alpha \leq \pi/4$, a constant c_{max} , rectangles R_1 and R_2 and $t > 1$, such that (as per Fig. 2): R_1 and R_2 have width w and height h , are separated by s , have a side parallel to ℓ , have their centers on ℓ , R_1 lies between a and R_2 , R_2 lies at most c_{max} away from a , R_1 lies at least $h/2$ away from a and $s \geq \sqrt{2} \frac{t+1}{t-1} (2 \sin(\alpha) c_{max} + h) + h$.*

Define A as the area in the cone with apex a , angle 2α and bisector ℓ that is at least $c_{cone} = c_{max} + h/2$ away from a . Then for any $p, q \in P$ with p lying in R_1 and q lying in R_2 , (p, q) bridges (a, A) .

The proof of this Lemma involves defining R'_1 and R'_2 as shown in Fig. 2 and showing that Lemma 2 can be invoked. We now have the tool needed for the main result.

3 Uniform Point Sets

Theorem 4 *There exists c_t dependent only on t such that for every $c > 0$, if P is a set of n points uniformly and independently distributed at random in a $\sqrt{n} \times \sqrt{n}$ square and n is large enough, then with probability at least $1 - n^{-c}$, P is locally- $(c \cdot c_t \log n)$ -bridged.*

To prove this we show that every point in P is locally- $(c \cdot c_t \log n)$ -bridged simultaneously with high probability. We will use Lemma 3 to achieve this.

The rectangles in Lemma 3 can be chosen to have a roughly constant chance of containing a point, and if we

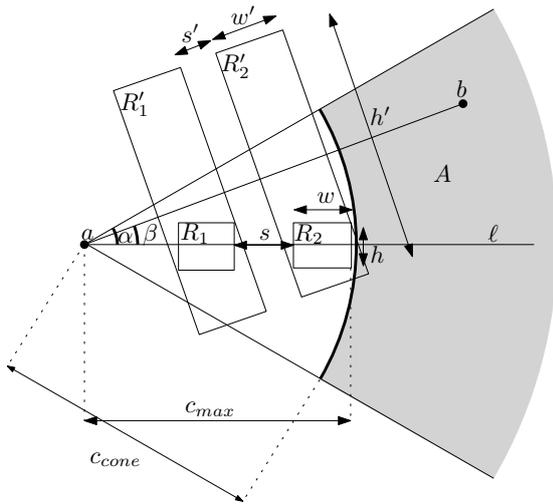


Figure 2: R_1 and R_2 are covered by R'_1 and R'_2 , which satisfy the requirements of Lemma 2

can fulfill the other requirements, the resulting pair of points bridges a relatively large part of \mathbb{R}^2 . In fact, we need only $\lceil \pi/\alpha \rceil$ cones (we will end up picking $\alpha = O(1/\log n)$) to cover the area we wish to cover. We show the likely existence of a pair of mandatory points that bridges a single cone and use a union bound to show such pairs are likely to exist for all cones simultaneously.

4 The Algorithm

We first introduce three tools used in the results below. Let c and c_t be as in Theorem 4 throughout this section. The first is that we can divide the input into a $\frac{\sqrt{n}}{c \cdot c_t \log n} \times \frac{\sqrt{n}}{c \cdot c_t \log n}$ grid in $O(n \log n)$ time, with every cell containing in expectation $O((c \cdot c_t \log n)^2)$ points.

The second tool is the ‘local’ Dijkstra algorithm. It determines for all points at most λ away from a source point s whether it has a t -path to s and if so, their network distance. It differs from the standard Dijkstra algorithm in that it only adds the points to the queue at most λt away from the source s by considering the points lying in cells at most λt away from s , and only considers the edges E_s that lead to such a point. Using the grid this can be done in $O((\lambda^2 + |E_s|) \log \lambda)$ expected time.

The third tool is called *path-hyperbola*. It is an area given by an origin point $u \in P$, a focus $v \in P$ and an edge set E , and is defined as $PH(u, v, E) = \{a \in \mathbb{R}^2 \mid d_{(P, E)}(u, v) + t \cdot |va| \leq t \cdot |ua|\}$. Obviously, if (p, q) bridges (a, b) , then $b \in PH(a, q, E)$ for every edge set E with t -paths for pairs of points (u, v) with $|uv| \leq |ab|$, making path-hyperbola at least as powerful as bridging points for guaranteeing t -paths.

If we perform a local Dijkstra on s , we either find pairs of points without t -path, or we find a set of network distances that induce a set of path-hyperbola. If s is locally- λ -bridged, the union of path-hyperbola will be a superset of the area more than λ away from s . This union can be computed in $O(\lambda^2 \log \lambda)$ expected time: using polar

coordinates, the union corresponds to a lower envelope. Since the hyperbolas pairwise intersect at most twice, this envelope has linear complexity and can be computed in $O(n_h \log n_h)$ time for n_h hyperbolas [3, 10]. This gives an efficient test of t -paths from s to all other points as least as accurate as local- λ -bridgedness.

Greedy Spanner. Consider the following algorithm that was introduced by Keil [7]:

Algorithm *GreedySpannerOriginal*(V, t)

1. $E \leftarrow \emptyset$
2. **for** every pair of distinct points (u, v) in ascending order of $|uv|$
3. **do if** $\delta_{(V, E)}(u, v) > t \cdot |uv|$
4. **then** add (u, v) to E
5. **return** E

The graph returned by this algorithm is called the *greedy spanner* on V for t and it is obviously a t -spanner, but the algorithm has a $O(n^3 \log n)$ running time. We make the following observation:

Lemma 5 *If P is λ -bridged, then the greedy spanner on P does not have edges longer than λ .*

Our algorithm consists of two phases. First we combine Lemma 5 with Theorem 4 to quickly compute a graph which (on uniform point sets) with high probability is the greedy spanner. In the second phase we use path hyperbola to verify if the resulting graph is indeed the greedy spanner. In this phase we may have to recompute some parts if we could not find a witness using the path hyperbolas. This ensures that we end up with the greedy spanner.

Phase I. We use the algorithm introduced in [1] (we will not explain that algorithm here), except we use our local Dijkstra instead of a normal Dijkstra and only consider well-separated pairs $\{A_i, B_i\}$ with $\min(A_i, B_i) \leq \lambda$, where $\min(A_i, B_i)$ indicates the minimal distance between the bounding boxes of A_i and B_i .

Using the analysis in [1] and using that the greedy spanner has degree $O(1)$, we conclude that if m is the number of considered well-separated pairs, the running time of our modified algorithm is $O(n \log n + \lambda^2 \log \lambda \sum_{i=1}^m \min(|A_i|, |B_i|))$. We therefore need to bound $\sum_{i=1}^m \min(|A_i|, |B_i|) \leq \sum_{i=1}^m |A_i| + |B_i| = \sum_{a \in P} |\{\{A_i, B_i\} \mid a \in A_i \vee a \in B_i\}|$.

For any $r \in \mathbb{R}$, a point p can only be in $O(1)$ well-separated pairs of length at most a constant factor higher or lower than l [5, Lemma 4.6.1]. We can therefore partition the well-separated pairs containing p into $O(1)$ -sized sets of similar length. As the minimal length per set differ by at least a constant factor, we conclude $|\{\{A_i, B_i\} \mid a \in A_i \vee a \in B_i\}| = O\left(\log \frac{\max_i \{\ell(\{A_i, B_i\})\}}{\min_i \{L(\{A_i, B_i\})\}}\right)$, here $\ell(A_i, B_i)$ is the distance between the centers of the bounding boxes. This last expression is $O(\log \lambda)$ in expectation on uniform point sets, giving an expected running time of $O(n \log n + n \lambda^2 \log^2 \lambda)$.

Phase II. The second phase gathers path-hyperbola as described earlier. We then consider the well-separated pairs that did not get considered in the first stage and try to prove for them that they will not produce a greedy spanner edge. For the remaining pairs, we employ the algorithm of [1] to find the remaining greedy spanner edges.

If for a point $u \in A_i$, the bounding of box B_i is covered by the union of path-hyperbola computed for u (testing this takes $O(\log n)$ time), then we say u is *discounted* with respect to $\{A_i, B_i\}$. If all $u \in A_i$ are discounted, then $\{A_i, B_i\}$ will not contain a greedy spanner edge and we say $\{A_i, B_i\}$ is *discounted*. This can be computed in $O(\log n \sum_{i=1}^m |A_i| + |B_i|) = O(n \log^2 \log n)$ expected time by an earlier argument.

We then perform the algorithm from [1], with small differences. We ignore pairs that have been discounted in the previous phase, and we do not perform a Dijkstra operation on points which have been discounted with respect to that pair as well. By Theorem 4, all pairs are discounted with high probability and hence this phase takes constant time in expectation on uniform point sets.

Using the expected running time of $O(n \log n + n \lambda^2 \log^2 \lambda)$ and $\lambda = c \cdot c_t \log n$ gives the following theorem

Theorem 6 *There is an algorithm that, given t and a point set P whose points are uniformly distributed in a $\sqrt{n} \times \sqrt{n}$ square, computes in $O((n+|E|)(c_t \log n)^2 \log^2(c_t \log n))$ expected time its greedy spanner, where c_t is a constant dependent only on t . The algorithm uses $O(n^2 \log^2 n)$ time on arbitrary P .*

5 Experimental results

We have tested the performance of our algorithm and WSPD-Greedy from [1]. All other published algorithms use quadratic space, and therefore running them on more than 10.000 points quickly becomes infeasible so we decided not to include them in our experiments. For a detailed comparison between the major quadratic space algorithms and WSPD-Greedy we refer to [1]. We generated point sets according to several distributions. We have recorded space usage and running time (wall clock time). The results are averages over several runs where new point sets were generated for each run. We discuss the performance on uniform point sets and on clustered point sets as these represent the best and worst cases respectively for our algorithm (with respect to our set of tests). To generate the clustered point set we used the same method as [1], that is, for n points, it consists of \sqrt{n} uniformly distributed point sets of \sqrt{n} uniformly distributed points.

Dependence on instance size. We have compared running time and space usage of WSPD-Greedy and our algorithm for different values of n . The space usage of our algorithm in both test sets was a small constant factor lower than that of the WSPD-Greedy algorithm when using $t = 2$ as the dilation decreases the space usage on

clustered point sets becomes more similar whereas on uniform point sets our algorithm becomes significantly more efficient. The running time on the clustered point set using $t = 2$ is about a factor 6 faster. On uniform point sets our algorithm appears to run in linear time making it much faster than WSPD-Greedy. On such a point set with 300.000 points, our new algorithm needs 19 minutes to compute greedy spanner for $t = 2$, whereas WSPD greedy needs 17 hours on the same point set.

Aside from generated instances we also experimented on some real point sets from the TSPLIB¹. The performance of our algorithm on these real datasets seems to be close to the uniform point sets and hence our algorithm also gives a massive improvement on these datasets.

References

- [1] S. P. A. Alewijnse, Q. W. Bouts, A. P. ten Brink, and K. Buchin. Computing the greedy spanner in linear space. In *Proc. 21th European Symposium on Algorithms (ESA)*, pages 37–48. Springer, 2013, arXiv:1306.4919.
- [2] S. P. A. Alewijnse, Q. W. Bouts, A. P. ten Brink, and K. Buchin. Distribution-sensitive construction of the greedy spanner. *CoRR*, arXiv:1401.1085, 2014.
- [3] M. Atallah. Some dynamic computational geometry problems. *Computers and Mathematics with Applications*, 11:1171–1181, 1985.
- [4] P. Bose, P. Carmi, M. Farshi, A. Maheshwari, and M. Smid. Computing the greedy spanner in near-quadratic time. *Algorithmica*, 58(3):711–729, 2010.
- [5] P. B. Callahan. *Dealing with Higher Dimensions: The Well-Separated Pair Decomposition and Its Applications*. PhD thesis, Johns Hopkins University, Baltimore, Maryland, 1995.
- [6] L. P. Chew. There are planar graphs almost as good as the complete graph. *J. Comput. System Sci.*, 39(2):205 – 219, 1989.
- [7] J. M. Keil. Approximating the complete Euclidean graph. In *1st Scandinavian Workshop on Algorithm Theory (SWAT)*, volume 318 of *LNCS*, pages 208–213. Springer, 1988.
- [8] G. Narasimhan and M. Smid. *Geometric Spanner Networks*. Cambridge University Press, 2007.
- [9] D. Peleg and A. A. Schäffer. Graph spanners. *Journal of Graph Theory*, 13(1):99–116, 1989.
- [10] M. Sharir and P. Agarwal. *Davenport-Schinzle Sequences and their Geometric Applications*. *camp-up*, 1995.

¹<http://comopt.ifl.uni-heidelberg.de/software/TSPLIB95/>