Approximating the d-dimensional complete Euclidean graph*

Jim Ruppert Raimund Seidel
Computer Science Division, University of California at Berkeley
Berkeley, CA 94720, USA

Abstract

Given a set S of n points in R^d, we show a new method for constructing a graph $G(S)$ that approximates the complete Euclidean graph in the following way: for any two points p, q in S, there is a path in $G(S)$ between p and q with length bounded by a constant times the Euclidean distance $dist(p,q)$. Furthermore, the graph will have only $O(n)$ edges. In the plane, for any integer $k > 6$, the path length constant will be $\frac{1}{1-2\sin\frac{\pi}{k}}$, and the number of edges will be at most kn. We give an algorithm for constructing $G(S)$ with running time $O(n(\log n)^{d-1})$ in dimension $d \geq 2$.

1 Introduction

Given a set S of n points in the plane, suppose we wish to design a network in which the distances between all pairs of points are small. The complete Euclidean graph minimizes distances for all pairs, but requires $O(n^2)$ edges. For some applications, it might be acceptable to increase the distances between points by a constant factor in order to reduce the number of edges to $O(n)$.

Let $G = (V, E)$ be an undirected graph with real positive edge weights. For two vertices $v, w \in V$, let $d_G(v, w)$ denote the length of the shortest path between v and w in G. We say that a subgraph $G' = (V, E')$ of G is a *t-approximation for G of sparsity c* iff $|E'| \leq c|V|$ and for each $v, w \in V$ we have $d_{G'}(v, w) \leq t \cdot d_G(v, w)$, where $t \geq 1$ is some real number. Our interest here is the case when G is the complete graph on a set S of n points in R^d, and where the weight of edge (p, q) is equal to the Euclidean distance between p and q. Graphs that approximate the complete Euclidean graph have applications in network design and motion planning problems, especially for algorithms that can utilize approximate shortest paths.

*This work was supported by an NSF Presidential Young Investigator Grant CCR-90-58840.

Chew introduced the notion of graphs that approximate the complete Euclidean graph and discussed some of their applications in [1]. He also showed that the Delaunay triangulation of S in the L_1 metric is a $\sqrt{10}$-approximation of the complete Euclidean graph. Since then, quite a few other types of approximating graphs have been discovered, some with lower constants, and others that have various additional properties. Dobkin, Friedman and Supowit showed that the usual (L_2 metric) Delaunay triangulation also $\frac{1+\sqrt{5}}{2}\pi \approx 5.08$-approximates the complete Euclidean graph [2]. The constant was later improved to $\frac{2\pi}{3\cos\frac{\pi}{6}} \approx 2.42$ by Keil and Gutwin [5]. Elsewhere [4], Keil showed how to construct a graph $G_k(S)$ for any integer $k > 6$, that $\frac{1}{\cos\frac{2\pi}{k}-\sin\frac{2\pi}{k}}$-approximates the complete Euclidean graph. By increasing k the approximation can be made arbitrarily close, but for any fixed k, $G_k(S)$ will have sparsity k, i.e. it will have linearly many edges. Subgraphs that are t-approximations of general graphs (not necessarily the complete Euclidean graph) were studied by Peleg and Schaffer in [7], where they were referred to as t-spanners.

The purpose of this paper is to describe a method of constructing linear-size graphs that approximate the complete Euclidean graph of a point set in R^d. This will be done by modifying the technique of Keil [4]. Along the way, we will also show how to improve the constants in the planar case. For $d \geq 2$ dimensions, we give an algorithm for constructing approximating graphs that runs in time $O(n(\log n)^{d-1})$.

2 The Planar Case

We first describe the construction in the planar case. Our construction is inspired by the *theta-graph* construction of Keil [4].

Given a set S of n points in R^2, and given any integer $k \geq 2$, the graph $G_k(S)$ is defined as follows. For any $p \in S$, let $Cone(p, i)$, $1 \leq i \leq k$, be the cone with apex p bounded by the two rays from p in

the directions $\frac{(i-1)2\pi}{k}$ and $\frac{i2\pi}{k}$. Then the cone axis $Axis(p,i)$ is the ray $\frac{(2i-1)\pi}{k}$, which bisects the cone. Let $Nigh(p,i)$ be the vertex in $Cone(p,i)$ whose orthogonal projection onto $Axis(p,i)$ is nearest p (if there is such a vertex). The vertex set of $G_k(S)$ is S, and the edge set is $\bigcup_{p\in S}\bigcup_{1\le i\le k}(p, Nigh(p,i))$.

Theorem 1 *Given a set S of n points in the plane, and given an integer $k > 6$, the graph $G_k(S)$ defined above has sparsity k and is a $\frac{1}{1-2\sin\frac{\pi}{k}}$-approximation of the complete Euclidean graph on S.*

Proof: $G_k(S)$ will have sparsity k because at each vertex p, we add at most 1 edge for each of the k cones emanating from p.

For any two vertices p and q, we now show how to construct a path P in $G_k(S)$ of the required length. (P may not be the shortest p-q path in $G_k(S)$.) Starting at p, we will add edges one at a time to the path. Let the vertices on the path be $p_0, p_1, \ldots$, with $p_0 = p$. From p_i, the next edge on the path will be the edge p_iq, if it is present in the graph. Otherwise, choose j such that $Cone(p_i, j)$ contains q, and take the edge to $p_{i+1} = Nigh(p_i, j)$. Let r be the number of edges in the constructed path. Let e_i be the edge from p_{i-1} to p_i, call its length δ_i, and let l_i be the Euclidean distance from p_i to q. Below, we will show that

$$\delta_i + l_i \le l_{i-1} + 2\delta_i \sin\phi \qquad (**)$$

holds for each edge along the path. Here, $\phi = \frac{\pi}{k}$, and 2ϕ corresponds to θ in [4]. Rearranging and summing over all r edges in the path we have

$$(1 - 2\sin\phi)\sum_{i=1}^{r}\delta_i \le \sum_{i=1}^{r}(l_{i-1} - l_i).$$

The right hand side is a telescoping sum, and $l_r = 0$ and $l_0 = dist(p,q)$, so we get

$$\sum_{i=1}^{r}\delta_i \le \frac{dist(p,q)}{1-2\sin\phi},$$

which is the desired bound on the length of a p-q path.

It remains to show that (**) holds at each edge along the path. Edge e_i connects p_{i-1} and p_i. If $p_i = q$, then $l_i = 0$, $\delta_i = l_{i-1}$, so (**) holds. Otherwise, we distinguish three different configurations for the points p_{i-1}, p_i and q. Suppose the rays $p_{i-1}p_i$ and $p_{i-1}q$ make angles α and β with the cone's axis, where $\alpha \le \phi$ and $\beta \le \phi$. Case 1 occurs when $\beta < \alpha$, case 2 is when $\beta \ge \alpha$ and p_i and q are on the same side of the cone axis, and case 3 when $\beta \ge \alpha$ and p_i and q are on opposite sides of the cone axis.

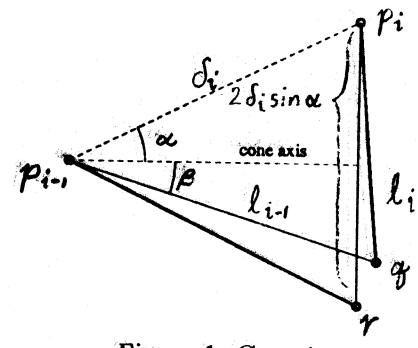

Figure 1: Case 1.

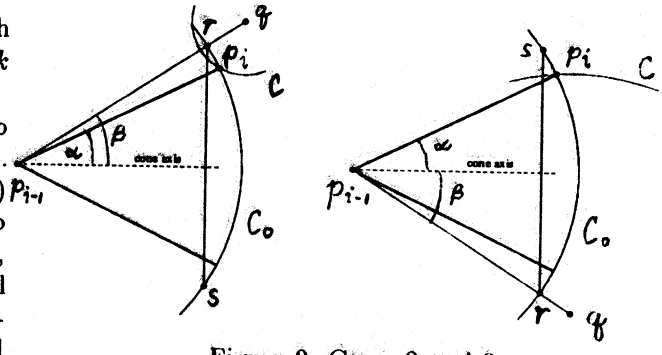

Figure 2: Cases 2 and 3.

In case 1 (see Figure 1), we know that since p_{i-1} wasn't adjacent to q, q must lie to the right of the vertical line through p_i. Let r be the mirror image of p_i across the cone axis. Then the segment $p_{i-1}r$ has length δ_i and the segment p_ir has length $2\delta_i \sin\alpha$, which is $\le 2\delta_i \sin\phi$, since $0 \le \alpha \le \phi$. Thus, showing (**) amounts to showing that in Figure 1, the sum of the lengths of the two thick solid lines is no greater than the sum of the two thin solid lines. This is certainly the case, by two applications of the triangle inequality.

For cases 2 and 3, we will again argue geometrically that (**) holds. Let C_0 be the circle through p_i centered at p_{i-1}. Let r be the intersection of C_0 with the ray $p_{i-1}q$, and let s be the mirror image of r across the cone axis (see Figure 2). To show (**), it suffices to show that

$$dist(p_{i-1}, p_i) + dist(p_i, q) \le dist(p_{i-1}, q) + dist(r, s),$$

since $dist(p_{i-1}, p_i) = \delta_i$, $dist(p_i, q) = l_i$, $dist(p_{i-1}, q) = l_{i-1}$, and $dist(r,s) \le 2\delta_i \sin\phi$. The last fact holds because $0 \le \beta \le \phi$.

Since $dist(p_{i-1}, p_i) = dist(p_{i-1}, q) - dist(r, q)$, we must show

$$dist(p_i, q) \le dist(r, q) + dist(r, s).$$

We can use the triangle inequality to reduce the right hand side to $dist(s, q)$. To see that $dist(p_i, q) \le$

$dist(q,s)$, we show that s is outside the circle C centered at q passing through p_i. C and C_0 intersect at two points that are mirror images across the line $p_{i-1}q$. Since one of these points is p_i, and s is farther (along C_0) from the line $p_{i-1}q$, s must be outside of C. Hence we have shown (**) in all 3 cases, and completed the proof of the theorem.

■

3 d-Dimensional Case

In $d > 2$ dimensions, we need to generalize the idea of choosing neighbors from each cone surrounding a vertex p. The most straightforward generalization is to choose a set of (overlapping) circular cones that cover space. All cones should have the same angle ϕ from axis to boundary. This can be done with a number of cones that depends only on d and ϕ. (ϕ corresponds to $\frac{\pi}{k}$ in the planar case.) Fejes Tóth surveys the techniques for sphere packing and covering in [3], and Rogers discusses the d-dimensional sphere covering problem in [8].

To construct an approximating graph, we choose neighbors for vertex p from the cones centered at p. In each cone, we again choose the point whose projection to the cone axis is nearest to p. As in Theorem 1, the resulting graph is a $\frac{1}{1-2\sin\frac{\phi}{2}}$-approximation of the d-dimensional complete Euclidean graph.

However, the computation of neighbors is difficult to do efficiently with circular cones. To enable us to get an efficient algorithm in the next section, we will use a *simplicial cone covering* instead. A simplicial cone in R^d is the convex hull of d rays sharing a common endpoint. A set of simplicial cones sharing an apex is a covering if their union covers all of space (the cones may overlap). For each cone, there should be a canonical direction to use as an axis, with the condition that from the apex, every point in the cone makes an angle $\leq \phi$ with the axis. The same set of cones is translated to each vertex for use in choosing its neighbors.

The crucial point is that the number of cones required in such a covering depends only on d and ϕ, and so will be a constant $C_{\phi,d}$ with respect to our graph construction algorithm. There are a number of ways to construct such a *simplicial cone covering* and associated set of directions at each vertex p. For instance, one can cover the surface of the $(d-1)$-sphere with appropriate spherical caps and then compute the Delaunay triangulation of the centers of the caps. Together with p, the resulting triangles define a set of cones. The canonical directions go from p through the Delaunay circumcenters.

The edge set of the approximating graph $G_\phi(S)$ will be $\bigcup_p \bigcup_i (p, Nigh(p,i))$. Here p ranges over all points in S, and $1 \leq i \leq C_{\phi,d}$. $Nigh(p,i)$ is the vertex in the ith simplicial cone whose projection to the ith canonical direction is nearest p.

Theorem 2 *Given a set S of n points in R^d, and given any $\phi > \frac{\pi}{6}$, the graph $G_\phi(S)$ is a $\frac{1}{1-2\sin\frac{\phi}{2}}$-approximation of the d-dimensional complete Euclidean graph on S. $G_\phi(S)$ will have sparsity $C_{\phi,d}$, which is a constant, assuming ϕ and d are fixed.*

Proof: The sparsity of $G_\phi(S)$ will be $C_{\phi,d}$, since we add at most $C_{\phi,d}$ edges around each vertex p. The proof of the path length bound is exactly the same as that in the planar case. We construct a path $p_0, p_1, \ldots$, by connecting p_{i-1} to p_i, the vertex in the cone of q with projection closest to p_{i-1}. At each edge $p_{i-1}p_i$ in the path from p to q, we just analyze what happens in the plane defined by p_{i-1}, p_i and q, where the cone axis is projected perpendicularly to this plane. The intersection of this plane with the simplicial cone will be a (2-dimensional) cone in which both boundary rays make an angle of less than ϕ with the (projected) axis. Hence α and β will both be less than ϕ, and the rest of the proof is the same as before. ■

4 Subquadratic Algorithm

The above algorithm for constructing d-dimensional approximating graphs can easily be implemented to run in $O(n^2)$ time, but since the output size is only $O(n)$, perhaps we can do better. Keil gives an algorithm for $d = 2$ that uses plane sweeps to get a running time of $O(n \log n)$ [4]. This algorithm can be adapted to generate our approximating graphs in the planar case in time $O(n \log n)$, but it does not appear to generalize to higher dimensions.

Instead, we will preprocess the point set S to build a data structure that allows efficient *orthogonal range queries* and use the results of these queries to decide which edges to use. We illustrate the technique in the planar case first. The algorithm proceeds in k ($=\frac{\pi}{\phi}$) phases. In the ith phase, we will determine $Nigh(p,i)$ for all $p \in S$, by performing a plane sweep in the direction $\frac{(2i-1)\pi}{k}$. Note that if q is in $Cone(p,i)$, then p will be in the symmetric cone (call it $Cone'(q,i)$) with apex q (see Figure 3). As the sweepline moves, we maintain the

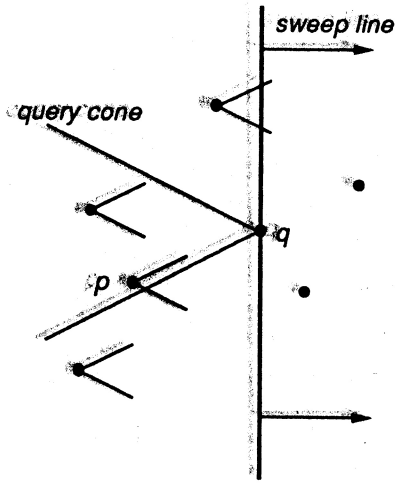

Figure 3: How to find all p such that q is in the cone anchored at p.

invariant that for all points p behind the sweepline, $Nigh(p,i)$ has been computed if $Nigh(p,i)$ is also behind the sweepline. When the sweepline reaches a point q, we query the data structure to find out which points are in $Cone'(q,i)$. For each such point p, let $Nigh(p,i) = q$, and remove p from the data structure. In order to use orthogonal range queries, we must use an affine transformation on the coordinates so that the query cones become quadrants.

In d dimensions, we need one hyperplane sweep for each of the $C_{\phi,d}$ canonical directions. The data structure to be used in the plane sweeps uses *range trees* and *priority search trees*, both described in the book by Mehlhorn [6]. We need $d-2$ levels of range trees, plus a level of priority search trees to handle the last 2 dimensions. Before each sweep, we must apply an appropriate affine transformation to the coordinates so that the (transformed) hyperplanes bounding the simplicial cone queries will be orthogonal. The total preprocessing time to build the data structure will be $O(n(\log n)^{d-1})$, and it will require $O(n(\log n)^{d-1})$ space. The total time for all $C_{\phi,d}$ sweeps, each with n queries, will be $O(n(\log n)^{d-1})$.

5 Remarks and Conclusion

In the planar case, some improvement can be made on the constants. In particular, when k is odd, there is an asymmetry between the cones at points p and q that we can take advantage of by growing paths from both ends. Interestingly, this asymmetry allows us to prove a bound near 10 on the path lengths even for the case $k = 5$. For even k, similar improvements

$\phi =$	$\pi/7$	$\pi/8$	$\pi/9$	$\pi/10$	$\pi/15$	$\pi/20$
Keil's method	–	–	8.11	4.52	1.97	1.56
Modification	7.56	4.26	3.17	2.62	1.71	1.46

Figure 4: Dependence of path length constant on ϕ for the planar case.

can be made by altering the apertures of some cones at each point. The details are omitted here due to lack of space.

We have shown an improvement to Keil's technique for constructing theta-graphs as approximations of the complete graph, and we have shown how to generalize our technique to higher dimensions. Figure 4 shows how the path length constant varies with ϕ, for the two methods. The θ of [4] corresponds to 2ϕ. In higher dimensions, the path length constant has the same dependence on ϕ, but the number of edges will increase.

References

[1] L.P. Chew. There is a planar graph almost as good as the complete graph. In *Proceedings of the Second Annual Symposium on Computational Geometry*, pages 160–177. ACM, 1986.

[2] D.P. Dobkin, S.J. Friedman, and K.J. Supowit. Delaunay graphs are almost as good as complete graphs. In *Proc. 28th Annual IEEE Symposium on Foundations of Computer Science*, pages 20–26, 1987. Revised version in Discrete and Computational Geometry 5:399–407, 1990.

[3] G. Fejes Tóth. *New Results in the Theory of Packing and Covering*, pages 318–359. Birkhäuser, 1983.

[4] M. Keil. Approximating the complete Euclidean graph. In *Proc. First Scandinavian Workshop on Algorithm Theory*, pages 208–213, 1988.

[5] M. Keil and C. Gutwin. The Delaunay triangulation closely approximates the complete graph. In *Proc. First Canadian Workshop on Algorithms and Data Structures*, pages 47–56, 1989. Revised version of this and above reference to appear in Discrete and Computational Geometry.

[6] K. Mehlhorn. *Data Structures and Algorithms 3: Multi-dimensional Searching and Computational Geometry*. Springer-Verlag, 1984.

[7] D. Peleg and A.A. Schaffer. Graph spanners. *Journal of Graph Theory*, 13(1):99–116, 1989.

[8] C.A. Rogers. Covering a sphere with spheres. *Mathematika*, 10:157–164, 1963.