Discrete Comput Geom 6:369-381 (1991)

Discrete & Computational
Geometry

A Sparse Graph Almost as Good as the Complete Graph on Points in K Dimensions

Pravin M. Vaidya*

AT&T Bell Laboratories, Murray Hill, NJ 07974, USA

Abstract. A set V of n points in k-dimensional space induces a complete weighted undirected graph as follows. The points are the vertices of this graph and the weight of an edge between any two points is the distance between the points under some L_p metric. Let $\varepsilon \leq 1$ be an error parameter and let k be fixed. We show how to extract in $O(n \log n + \varepsilon^{-k} \log(1/\varepsilon)n)$ time a sparse subgraph $G = (V, E)$ of the complete graph on V such that: (a) for any two points x, y in V, the length of the shortest path in G between x and y is at most $(1 + \varepsilon)$ times the distance between x and y, and (b) $|E| = O(\varepsilon^{-k}n)$.

1. Introduction

A set V of n points in k-dimensional space induces a complete weighted undirected graph as follows. The points are the vertices of this graph and the weight of an edge between any two points is the distance between the points under some L_p metric. Note that the L_p distance between $x = (x_1, x_2, \ldots, x_k)$ and $y = (y_1, y_2, \ldots, y_k)$ is given by $(\sum_{i=1}^{k} |x_i - y_i|^p)^{1/p}$. We study the problem of compactly representing some of the information represented by this graph. Specifically, given a set V of n points in k-dimensional space, we show how to extract a sparse subgraph $G = (V, E)$ of the complete graph on V with the following properties:

(a) Let $d(x, y)$ denote the distance between (x, y), and $D_G(x, y)$ denote the length of the shortest path in G between x and y. (By convention, $D_G(x, x) = 0$.) Then, for any pair of points x, y in V, $D_G(x, y) \leq (1 + \varepsilon)d(x, y)$.

(b) $|E| = O(2^k(3 + 12c_p/\varepsilon)^k n)$ where $c_p = k^{1/p}$ for the L_p metric. Note that c_p is the length of the diagonal of a unit box in the L_p metric.

* Currently at the Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA.

Furthermore, we show how to construct G in $O(n \log n + \varepsilon^{-k} \log(1/\varepsilon)n)$ time for fixed k. Specifically, it is shown that G can be obtained from V in $O(4^k n \log n + k \log(k/\varepsilon)4^k(3 + 12c_p/\varepsilon)^k n)$ time. (Note that G can be obtained in $O(n \log n)$ time for fixed k and ε.)

To motivate the problem we briefly discuss two applications. The first application is mentioned in [1] and concerns the design of communication networks. One way to design a communication network on a set of n points V (in say two or three dimensions) is to use the complete graph on V; this minimizes the transmission distance between two points but there are $\Omega(n^2)$ links which could be too many. Using G instead of the complete graph reduces the number of links to $O(\varepsilon^{-k}n)$ at the expense of increasing the transmission distance by a factor of $(1 + \varepsilon)$.

The second applications is in finding approximate minimum spanning trees [11] in the complete graph on V. Note that the length of a spanning tree is the sum of the lengths of the edges in the spanning tree, and a minimum spanning tree is a spanning tree of minimum length. Since $D_G(x, y) \leq (1 + \varepsilon)d(x, y)$ for a pair of points x, y in V, it follows that there exists a spanning tree in G whose length is at most $(1 + \varepsilon)$ times the length of a minimum spanning tree in the complete graph on V. Thus a minimum spanning tree in G is a good approximate minimum spanning tree in the complete graph on V. The graph G could also be used to find quickly a perfect matching on V whose length is close to the minimum (for details, see [10]).

The problem of approximating the complete graph on a set V of n points in the plane has been studied in [1]. In [1] it is shown that there is a planar subgraph of this complete graph such that the length of the shortest path in the subgraph between any two points x and y in V is at most $\sqrt{10}$ times the euclidean distance between x and y. Specifically, the L_1 Delaunay triangulation [1], [7] of V is such a subgraph. Note that since the subgraph is planar it has $O(n)$ edges. In [4] a similar result is proved about the L_2 Delaunay triangulation [7] of V; specifically, that the length of the shortest path between x and y in the L_2 Delaunay triangulation is at most 5.08 times the euclidean distance between x and y. For a set of points V in the plane the graph G described in this paper is asymptotically a better approximation to the complete graph on V than the Delaunay triangulation, since $D_G(x, y) \leq (1 + \varepsilon)d(x, y)$ for any pair of points x and y in V, and any given L_p metric. Also note that the graph G provides a good approximation to the complete graph on points in any fixed dimensional space rather than just for the complete graph on points in the plane, and that the Delaunay triangulation is not useful for $k > 2$ because it may not be sparse. Furthermore, if ε is fixed, then the time for constructing G is asymptotically the same as the time for constructing the Delaunay triangulation of a set of n points in the plane; the construction of each of them requires $\Theta(n \log n)$ time.

It is worth noting that the simple algorithm given below, suggested by T. Feder and N. Nisan [5], will find a subgraph H of the complete graph with distance and sparseness properties similar to those of G.

Choose a tolerance angle θ (related to ε) and process edges in increasing order of length as follows. For each edge (x, y), if H does not contain an edge (x, z) (or (y, z)) such that the angle zxy (or zyx) is less than θ, then add (x, y) to H.

H has bounded degree (for fixed k) but computing it takes $O(n^2 \log n)$ time. The subgraph G on the other hand can be obtained in $O(n \log n)$ time. We also note that in [3] Clarkson describes how to extract a subgraph of the complete graph on points in three-dimensions, with properties similar to those of G, in $O(n^2)$ time; his technique utilizes narrow cones and is quite different from ours.

In Section 2 we give some notation and definitions. In Section 3 we define the graph $G = (V, E)$ and in Section 4 we show that G has the properties (a) and (b) mentioned at the beginning of this section. In Section 5 we describe how to extract the graph G from the set of points V in $O(n \log n + \varepsilon^{-k} \log(1/\varepsilon)n)$ time for fixed k.

2. Notation and Definitions

We give some notation useful in defining the graph G. We define a *box* b to be the product $J_1 \times J_2 \times \cdots \times J_k$ of k intervals (either closed, semiclosed, or open), or, equivalently, the set of those points $x = (x_1, x_2, \ldots, x_k)$ such that x_i lies in the interval J_i, for $i = 1, \ldots, k$. A box is a *cubical* box iff all the k intervals defining the box are of identical length. The size of cubical box b equals the length of each of the intervals defining b and is denoted by *size*(b). Note that the maximum L_p distance between a pair of points in a cubical box b is c_p *size*(b). We only deal with cubical boxes in this paper and box means cubical box. The centre $\psi(b)$ of a box b is defined to be the point $(\psi_1(b), \ldots, \psi_k(b))$ where $\psi_i(b)$ is the centre of the ith interval defining the box b, for $i = 1, \ldots, k$. Let $h_i(b)$ be the hyperplane defined by $h_i(b) = \{x: x_i = \psi_i(b)\}$, let $L(h_i(b))$ be the left open half-space $\{x: x_i < \psi_i(b)\}$, and let $R(h_i(b))$ be the right closed half-space $\{x: x_i \geq \psi_i(b)\}$.

Let *Immediate-successors*(b) be the set of boxes defined by

$$\textit{Immediate-Successors}(b) = \{b': b' = b \cap f_1 \cap \cdots \cap f_k, \text{ where } f_i = L(h_i(b)) \text{ or } f_i = R(h_i(b)), 1 \leq i \leq k\}.$$

Immediate-Successors(b) is the set of 2^k boxes obtained upon cutting up b by k mutually orthogonal hyperplanes passing through the center of b, each plane being perpendicular to one of the k coordinate axes.

Corresponding to a cubical box b, let *shrunk*(b) be a cubical box such that:

1. If $|b \cap V| \leq 1$, then *shrunk*(b) $= b \cap V$.
2. If $|b \cap V| \geq 2$, then (i) *shrunk*(b) $\subseteq b$, (ii) *shrunk*(b) $\cap V = b \cap V$, and (iii) the maximum L_∞ distance between a pair of points in *shrunk*(b) $\cap V$ equals the size of *shrunk*(b).

Intuitively, obtaining *shrunk*(b) corresponds to shrinking b as much as possible without destroying its cubical shape or forcing outside any point in V that is within it.

For a cubical box b such that $|b \cap V| \geq 2$, let *Successors*(b) be the set of boxes defined as

$$\textit{Successors}(b) = \{b': b' = \textit{shrunk}(b''), b'' \in \textit{Immediate-Successors}(b), |b'' \cap V| \geq 1\}.$$

Whereas if $|b \cap V| \leq 1$, then $Successors(b) = \varnothing$. Each box in $Successors(b)$ has been shrunk as much as possible so that further shrinkage would either destroy the cubical shape of the box or push out of the box a point in V that was originally in the box.

Let *Box-Tree* be a rooted tree of boxes defined as follows:

Each node in the *Box-Tree* is a cubical box, the root is a smallest cubical box containing all the points in V, and the children of each nonleaf box (node) b in the *Box-Tree* are the boxes in $Successors(b)$.

Note that each nonleaf node in the *Box-Tree* has at least two children. Furthermore, each leaf is a singleton set containing exactly one point in V, and the leaf boxes partition the set V. The *Box-tree* is similar to the *cell-tree* [2] and the *quad-tree* [6].

3. The Graph $G = (V, E)$

We now describe the graph $G = (V, E)$. Let B be the set of all the boxes (nodes) in the *Box-Tree*. Let $father(b)$ denote the father or box b in the *Box-Tree*. For a box $b \in B$, let $rep(b)$ be a representative point in $b \cap V$. (Note that for a leaf box b, $rep(b)$ is the unique point in $b \cap V$.) Some of the edges in E incident to $rep(b)$ will be defined in terms of a list of boxes in the "neighborhood" of b which will be denoted by $Near(b)$. Let $d_{\min}(b, b')$ denote the minimum distance between a point in b and a point in b'. For a box $b \in B$, $Near(b)$ is defined as

$$Near(b) = \left\{ b' : b' \in B,\ size(b') < size(b),\ size(father(b')) \geq size(b),\ d_{\min}(b, b') \leq \frac{6c_p}{\varepsilon} size(b) \right\}.$$

Note that $Successors(b) \subseteq Near(b)$ and that the boxes in $Near(b)$ are disjoint. Also, observe that the boxes in B are (partially) ordered by containment. $Near(b)$ is obtained by taking the set of all boxes smaller than b whose minimum distance from b is less than (or equal to) a certain threshold $((6c_p/\varepsilon)size(b))$, and then picking the maximal boxes in this set.

The set of edges E (and thus $G = (V, E)$) is defined as follows:

$$E = E_1 \cup E_2,$$

where

$$E_1 = \{(rep(b), rep(b')) : b \in B,\ b' \in successors(b),\ rep(b) \neq rep(b')\}$$

and

$$E_2 = \{(rep(b), rep(b')): b \in B, b' \in B, b' \in Near(father(b))\}.$$

4. $G = (V, E)$ is a Good Approximation to the Complete Graph on V

We show that $G = (V, E)$ has the following properties.

(a) For any two points x, y in V, $D_G(x, y) \leq (1 + \varepsilon)d(x, y)$ where the $D_G(x, y)$ denotes the length of the shortest path in G between x and y.
(b) $|E| = O(2^k(3 + 12c_p/\varepsilon)^k n)$ where $c_p = k^{1/p}$ for the L_p metric.

To prove property (a) we require the following lemma.

Lemma 1. *Let x be a point in V and let b be a box in B such that $x \in b$. Then $D_G(x, rep(b)) \leq 2c_p\, size(b)$.*

Proof. Let $(\{x\} = b_0, b_1, \ldots, b_r = b)$ be the path from $\{x\}$ to b in the *Box-Tree*. Then $b_i \in Successors(b_{i+1})$ and $size(b_i) \leq \frac{1}{2}\, size(b_{i+1})$, $1 \leq i < r$. Moreover, for any of the L_p metrics there is an edge in E_1 (and thereby in E) of length at most $c_p\, size(b_{i+1})$ between $rep(b_i)$ and $rep(b_{i+1})$. Thus

$$D_G(x, rep(b)) \leq \sum_{i=0}^{r-1} D_G(rep(b_i), rep(b_{i+1})) \leq \sum_{i=0}^{r-1} c_p\, size(b_{i+1}) \leq 2c_p\, size(b). \quad \square$$

We now show that G has property (a) described above.

Lemma 2. *For a pair of points x,y in V, $D_G(x, y) \leq (1 + \varepsilon)d(x, y)$.*

Proof. For a point $z \in V$, let $Path(z, r)$ be the set of all boxes $b \in B$ such that b is on the path from $\{z\}$ to the root in the *Box-Tree* and $size(b) > r$. Note that, for each box $b \in Path(z, r)$, $z \in b$ if $r \geq 0$. Let b_x be the smallest box in $Path(x, \varepsilon d(x, y)/6c_p)$, and let b_y be the smallest box in $Path(y, \varepsilon d(x, y)/6c_p)$. Let b'_x be the box in $Successors(b_x)$ such that $x \in b'_x$. (It could be that $b'_x = \{x\}$.) Similarly, let b'_y be the box in $Successors(b_y)$ such that $y \in b'_y$.

We have

$$D_G(x, y) \leq D_G(x, rep(b'_x)) + D_G(rep(b'_x), rep(b'_y)) + D_G(y, rep(b'_y)). \qquad \text{(i)}$$

We bound $D_G(x, rep(b'_x))$, $D_G(y, rep(b'_y))$, and $D_G(rep(b'_x), rep(b'_y))$ in terms of ε and $d(x, y)$. From Lemma 1 it follows that $D_G(x, rep(b'_x)) \leq 2c_p\ size(b'_x)$ and $D_G(y, rep(b'_y)) \leq 2c_p\ size(b'_y))$. Since b_x is the smallest box in $Path(x, \varepsilon d(x, y)/6c_p)$, and $b'_x \in Successors(b_x)$, $size(b'_x) \leq \varepsilon d(x, y)/6c_p$, and hence $D_G(x, rep(b'_x)) \leq (\varepsilon/3)d(x, y)$. Similarly, we can show that $D_G(y, rep(b'_y)) \leq (\varepsilon/3)d(x, y)$. We show below that $D_G(rep(b'_x), rep(b'_y)) \leq (1 + \varepsilon/3)d(x, y)$. Then from (i) above it would follow that $D_G(x, y) \leq (1 + \varepsilon)d(x, y)$.

We now show that $D_G(rep(b'_x), rep(b'_y)) \leq (1 + \varepsilon/3)d(x, y)$. We assume that the boxes b'_x and b'_y are distinct. (Otherwise, $D_G(rep(b'_x), rep(b'_y)) = 0$.) Suppose that $size(b_x) \leq size(b_y)$. (The case where $size(b_y) \leq size(b_x)$ is similar.) Then $b'_y \in Near(b_x)$. This follows from the following three observations and the definition of $Near(b_x)$. First, since $size(b_x) > \varepsilon d(x, y)/6c_p$, we have

$$d_{\min}(b_x, b'_y) \leq d(x, y) \leq \frac{6c_p}{\varepsilon} size(b_x).$$

Second,

$$size(b'_y) \leq \frac{\varepsilon d(x, y)}{6c_p} < size(b_x).$$

Third, $size(father(b'_y)) = size(b_y) \geq size(b_x)$.

Since $father(b'_x) = b_x$ and $b'_y \in Near(b_x)$, $(rep(b'_x), rep(b'_y)) \in E_2$ and thus $(rep(b'_x), rep(b'_y)) \in E$. Then, for any of the L_p metrics,

$$\begin{aligned} D_G(rep(b'_x), rep(b'_y)) &= d(rep(b'_x), rep(b'_y)) \\ &\leq d(x, y) + d(x, rep(b'_x)) + d(y, rep(b'_y)) \\ &\leq d(x, y) + c_p\, size(b'_x) + c_p\, size(b'_y) \\ &\leq d(x, y) + c_p \frac{\varepsilon d(x, y)}{6c_p} + c_p \frac{\varepsilon d(x, y)}{6c_p} \\ &\leq \left(1 + \frac{\varepsilon}{3}\right) d(x, y). \end{aligned}$$

□

To bound the number of edges in G we require a bound on the size of $Near(b)$ for each $b \in B$. Such a bound is provided by the following lemma.

Lemma 3. *For $b \in B$, let $A(b)$ be a subset of B satisfying the following two conditions:*

1. *For each $b' \in A(b)$, $d_{\min}(b, b') \leq r\delta$, and $size(father(b')) \geq \delta \geq size(b)$.*
2. *For any pair of boxes b', b'' in $A(b)$, $b' \cap b'' = \varnothing$.*

Then $|A(b)| \leq 2^k(3 + 2r)^k$.

Before giving a proof of Lemma 3 we give an intuitive sketch. First, observe that a box b' in $A(b)$ is obtained by shrinking a box of size at least $\delta/2$ in *Immediate-successors(father(b'))*; thus the parameter δ is a measure of the size of the empty space around a box in $A(b)$. Based on this observation we can construct a set of disjoint boxes of the same cardinality as $A(b)$, with each box in the set of size $\delta/2$ and contained in a box of size approximately $2r\delta$ around b. A bound on $|A(b)|$ may then be obtained by noting that at most $O(vr)^k$, v fixed, disjoint boxes of size $\delta/2$ may be packed in a box of size $2r\delta$.

Proof of Lemma 3. Recall that *Immediate-Successors*(b') is the set of 2^k boxes obtained by cutting up b' by k mutually orthogonal hyperplanes passing through the center of b', each plane being perpendicular to one of the k coordinate axes. Let

$$A_f(b) = \{b' : b' \in \textit{Immediate-Successors}(\textit{father}(b'')), b' \supseteq b'', b'' \in A(b)\}.$$

We have

$$\forall b' \in A_f(b), \qquad \textit{size}(b') \geq \delta/2 \geq \textit{size}(b)/2.$$

Since the boxes in $A(b)$ are disjoint, the boxes in $A_f(b)$ are also disjoint and thus

$$|A_f(b)| = |A(b)|.$$

We bound $|A_f(b)|$. Note that, for each $b' \in A_f(b)$, $d_{\min}(b, b') \leq r\delta$ and $\textit{size}(b') \geq \delta/2 \geq \textit{size}(b)/2$. Shrink each box $b' \in A_f(b)$ to obtain a box b'' such that $\textit{size}(b'') = \delta/2$ and $d_{\min}(b, b'') = d_{\min}(b, b')$; let $A_f^s(b)$ be the set obtained by shrinking the boxes in $A_f(b)$ in this manner. It is clear that

$$|A_f^s(b)| = |A_f(b)|.$$

Let $\hat{b}$ be a box such that $\psi(\hat{b}) = \psi(b)$ (i.e., b and $\hat{b}$ have the same center) and $\textit{size}(\hat{b}) = (3 + 2r)\delta$. This choice for $\textit{size}(\hat{b})$ together with the condition that $d_{\min}(b, b'') \leq r\delta$ for each $b'' \in A_f^s(b)$, guarantees that each box in $A_f^s(b)$ is a subset of $\hat{b}$.

Since, for each box $b'' \in A_f^s(b)$, $b'' \subseteq \hat{b}$, and as boxes in $A_f^s(b)$ are disjoint we have

$$|A_f^s(b)| \leq \frac{\text{volume of } \hat{b}}{\text{volume of a box in } A_f^s(b)} \leq 2^k(3 + 2r)^k.$$

Thus

$$|A(b)| = |A_f(b)| = |A_f^s(b)| \leq 2^k(3 + 2r)^k. \qquad \square$$

The number of edges in E is bounded as follows. The number of edges in E_1 is upper bounded by the number of nodes (boxes) in the *Box-tree*, and so $|E_1| \leq 2n$. There is an edge in E_2 between $\textit{rep}(b)$ and $\textit{rep}(b')$ only if $b \in \textit{Near}(\textit{father}(b'))$ or $b' \in \textit{Near}(\textit{father}(b))$. Thus

$$|E_2| \leq \sum_{b \in B} |\textit{Near}(b)|.$$

Since the boxes in $\textit{Near}(b)$ are disjoint, we can apply Lemma 3 with $A(b) = \textit{Near}(b)$, $\delta = \textit{size}(b)$, and $r = 6c_p/\varepsilon$, and conclude that

$$|\textit{Near}(b)| \leq 2^k\left(3 + \frac{12c_p}{\varepsilon}\right)^k.$$

Hence

$$|E_2| \leq \sum_{b \in B} |Near(b)| = O\left(2^k\left(3 + \frac{12c_p}{\varepsilon}\right)^k n\right).$$

Finally,

$$|E| \leq |E_1| + |E_2| = O\left(2^k\left(3 + \frac{12c_p}{\varepsilon}\right)^k n\right).$$

We have shown that G has the property (b) mentioned at beginning of the section.

5. Speedily Constructing $G = (V, E)$

We see that the *Box-Tree* can be constructed in $O(4^k n \log n)$ time. We also show that once the *Box-Tree* is available, the sets *Near*(b) can be obtained for all the boxes $b \in B$ in $O(n \log n + k \log(k/\varepsilon)4^k(3 + 12c_p/\varepsilon)^k n)$ time. The representatives in all the boxes in B can be chosen in $O(n)$ time by starting with the leaf boxes in the *Box-Tree*, and, for a nonleaf box b, letting *rep*(b) equal *rep*(b') for some $b' \in$ *Successors*(b). It then follows that G can be obtained in $O(4^k n \log n + k \log(k/\varepsilon)4^k(3 + 12c_p/\varepsilon)^k n)$ time.

The *Box-Tree* can be obtained in $O(4^k n \log n)$ time as a byproduct of the All-Nearest-Neighbors Algorithm in [9]. We start with a tree consisting of just the root box, and grow the tree by splitting a leaf box in the current tree that has the largest volume among all the leaf boxes in the current tree. A box b is split by k mutually perpendicular hyperplanes through its center to give the boxes in *Immediate-Successors*(b) and the boxes in *Immediate-Successors*(b) are then suitably shrunk to obtain the boxes in *Successors*(b). The leaf boxes in the current tree partition the points in V. For each leaf box b in the current tree, k sorted lists of the points in $b \cap V$ are maintained, each list containing the points ordered on one of the k coordinates. The ordered lists enable efficient splitting of boxes. For details the reader may refer to [9].

We now describe how to obtain the sets *Near*(b) for the boxes in B. The boxes in B are processed in non-increasing order of size; B_p denotes the set of processed boxes in B. Let B_s be the set of boxes given by

$$B_s = \{b: \exists b' \in B_p \text{ s.t. } b \in Successors(b'), b \notin B_p\}.$$

For each box $b \in B_s$, we maintain two sets of boxes, $\alpha(b)$ and $\beta(b)$, where

$$\alpha(b) = \left\{b': b' \in B_s, d_{\min}(b, b') \leq \frac{6c_p}{\varepsilon} size(b)\right\}$$

and

$$\beta(b) = \{b' : b \in \alpha(b')\}.$$

The set $\alpha(b)$ is eventually used to obtain the set $Near(b)$.

At each step the largest box in B_s, denoted by b_L, is processed. The set $Near(b_L)$ is obtained from $\alpha(b_L)$, b_L is moved from B_s to B_p and the boxes in $Successors(b_L)$ are added to B_s. The α and β sets are created for the boxes in $Successors(b_L)$, and suitably updated for the boxes in $\alpha(b_L) \cup \beta(b_L)$. We note that the α sets of only the boxes in $Successors(b_L) \cup \beta(b_L)$ are affected during a step, and the β sets of only the boxes in $Successors(b_L) \cup \alpha(b_L)$ are affected during a step.

To compute $Near(b_L)$ quickly during a step we rely on the following lemma which is proved later.

Lemma 4. *At the start of each step,*

$$Near(b_L) \subseteq \left(\alpha(b_L) \cup \left(\bigcup_{b \in \alpha(b_L)} Successors(b)\right)\right).$$

To create and update the α, β sets efficiently we utilize the following observations:

1. A box b is added to or deleted from $\beta(b')$ whenever b' is added to or deleted from $\alpha(b)$.
2. For each $b \in Successors(b_L)$, $\alpha(b) \subseteq (\alpha(b_L) \cup Successors(b_L))$ and $\beta(b) \subseteq (\beta(b_L) \cup Successors(b_L))$.
3. For each $b \in \alpha(b_L)$ (*resp.* $b \in \beta(b_L)$), only a box in $Successors(b_L)$ can get added to $\beta(b)$ (*resp.* $\alpha(b)$).

We now give the algorithm for computing the sets $Near(b)$. Initially, B_p is empty and B_s is the singleton set containing the root of the *Box-Tree*.

Procedure Construct-Near-sets
Begin
/* **Initialize** */
$B_p := \varnothing$; $B_s := \{root\}$; $b_L := root$;
$\alpha(root) := \{root\}$; $\beta(root) := \{root\}$;
While $size(b_L) > 0$ do
Begin
/* **Process** b_L */
/* **Compute** $Near(b_L)$ from $\alpha(b_L)$ */
$Near\ (b_L) := \alpha(b_L) \cup (\bigcup_{b \in \alpha(b_L)} Successors(b))$;
Delete from $Near(b_L)$ each box b such that
$size(b) \geq size(b_L)$ or $size(father(b) < size(b_L)$;
Delete from $Near(b_L)$ each box b such that $d_{min}(b_L, b) > (6c_p/\varepsilon)\ size(b_L)$;
/* **Update** α, β sets and B_p, B_s */

For all $b \in Successors(b_L)$ do
$\alpha(b) := \alpha(b_L) \cup Successors(b_L) - \{b_L\}$;
$\beta(b) := \beta(b_L) \cup Successors(b_L) - \{b_L\}$;
For all $b \in (\beta(b_L) - \{b_L\})$ do
$\alpha(b) := \alpha(b) \cup Successors(b_L) - \{b_L\}$;
For all $b \in (\alpha(b_L)) - \{b_L\})$ do
$\beta(b) := \beta(b) \cup Successors(b_L) - \{b_L\}$;
For all $b \in (Successors(b_L) \cup \beta(b_L) - \{b_L\}$ do
Delete from $\alpha(b)$ each box b' such that $d_{min}(b, b') > (6c_p/\varepsilon)\, size(b)$
and whenever b' is deleted from $\alpha(b)$ also delete b from $\beta(b')$;
$B_p := B_p \cup \{b_L\}$; $B_s := B_s \cup Successors(b_L) - \{b_L\}$;
$b_L :=$ largest box in B_s;
endwhile
For each leaf box $b \in B$, $Near(b) := \varnothing$;
end Construct-Near-sets

First, we show the correctness of the above procedure, and then bound the time requirement. An execution of the while loop in the above procedure is referred to as a step. It is easily seen that the α and β sets are correctly updated at each step.

Lemma 5. *Let $\Pi = (b_0, b_1, \ldots, b_m)$, $m \geq 1$, be any path in the Box-Tree such that b_0 is a leaf, $b_{i+1} = father(b_i)$, $0 \leq i < m$, and b_m is the root. At the start of each step Π satisfies the following condition:*

There exists an $r \geq 0$ such that (1) $b_r \in B_s$, (2) $b_i \in (B - (B_p \cup B_s))$, $0 \leq i < r$, and (3) $b_j \in B_p$, $r < j \leq m$.

Proof. By easy induction based on the observation that at the end of a step b_L is moved from B_s to B_p and all the boxes in $Successors(b_L)$ are added to B_s. □

From Lemma 4 it follows that when the above procedure has terminated, $Near(b)$ has been correctly computed for each box in B_p. From Lemma 5 it follows that when the procedure terminates, $B_p \cup B_s = B$, and each box in B_s is a leaf. So the procedure correctly computes $Near(b)$ for all $b \in B$.

To bound the time requirement of the procedure we need to bound the sizes of the α and β sets. The next lemma is useful in proving bounds on the sizes of the α and β sets.

Lemma 6. *At the start of each step the following statements hold:*

(1) *The boxes in B_s are disjoint.*
(2) *For each $b \in B_s$, $size(father(b)) \geq size(b_L)$.*

Proof. (1) follows from Lemma 5. (2) follows by an easy induction based on the observation that during each step the largest box in B_s is processed. □

We next bound the sizes of the α and β sets.

Lemma 7. *At the start of each step, for each box $b \in B_s$ the following bounds hold:*

(I) $|\alpha(b)| \leq 2^k(3 + 12c_p/\varepsilon)^k$.
(II) $|\beta(b)| \leq 2^k(3 + 12c_p/\varepsilon)^k$.

Proof. From the definition of $\alpha(b)$ and Lemma 6 it follows that the conditions of Lemma 3 are satisfied with $A(b) = \alpha(b)$, $\delta = size(b)$, and $r = 6c_p/\varepsilon$. (I) then follows by the application of Lemma 3.

Next note that, for each box $b' \in \beta(b)$,

$$d_{\min}(b, b') \leq \frac{6c_p}{\varepsilon} size(b') \leq \frac{6c_p}{\varepsilon} size(b_L).$$

Then from Lemma 6 it follows that the conditions of Lemma 3 are satisfied with $A(b) = \beta(b)$, $\delta = size(b_L)$, and $r = 6c_p/\varepsilon$. (II) then follows by the application of Lemma 3. □

We now show Lemma 4.

Lemma 4. *At the start of each step,*

$$Near(b_L) \subseteq \left(\alpha(b_L) \cup \left(\bigcup_{b \in \alpha(b_L)} Successors(b)\right)\right).$$

Proof. Note that

$$Near(b_L) = \left\{b: size(b) < size(b_L), size(father(b)) \geq size(b_L), d_{\min}(b_L, b) \leq \frac{6c_p}{\varepsilon} size(b_L)\right\}.$$

We first show that

$$Near(b_L) \subseteq \left(B_s \cup \left(\bigcup_{b \in B_s} Successors(b)\right)\right). \tag{i}$$

From Lemmas 5 and 6 it follows that, for all $b \in B_p$, $size(b) \geq size(b_L)$. Thus

$$Near(b_L) \cap B_p = \varnothing. \tag{ii}$$

From Lemma 5 we get that, for each $b \in B - (B_p \cup B_s)$, $size(b) < size(b_L)$. Hence,

$$\text{for each} \quad b \in Near(b_L), \qquad father(b) \in (B_p \cup B_s). \tag{iii}$$

Equation (i) then follows from (ii), (iii), and Lemma 5.

Now suppose that $b \in Near(b_L)$. From the definition of $Near(b_L)$ we get that $d_{min}(b_L, b) \leq (6c_p/\varepsilon)size(b_L)$ and that $d_{min}(b_L, father(b)) \leq (6c_p/\varepsilon)\, size(b_L)$. Thus if $b \in B_s$ we can conclude that $b \in \alpha(b_L)$. So suppose $b \notin B_s$. Then from (i) we get that $father(b) \in B_s$ and hence $father(b) \in \alpha(b_L)$. Thus we may conclude that, for each $b \in Near(b_L)$, either $b \in \alpha(b_L)$ or $father(b) \in \alpha(b_L)$. Lemma 4 then follows. □

The running time of *Procedure Construct-Near-sets* is upper bounded as follows. By maintaining a heap [8] for the boxes in B_s, b_L may be selected in $O(\log n)$ time per step; the total time for heap maintainence and selection of b_L is $O(n \log n)$. From Lemma 7 it follows that the sizes of α and β sets are bounded by a constant (dependent on k and ε) and hence there are a constant number of additions to α and β sets during each step. Thus the total number of additions to and deletions from α and β sets is $O(n)$. For a box b, we implement $\alpha(b)$ by a data structure which allows insertions and deletions to be performed in $O(\log(|\alpha(b)|))$ time, and allows access to a box b', with the largest value of the parameter $d_{min}(b, b')$ in $O(\log(|\alpha(b)|))$ time. The set $\beta(b)$ is also implemented by an identical data structure. Note that it suffices to implement $\alpha(b)$ and $\beta(b)$ as a heap or a 2–3 tree [8]. So the α and β sets can be maintained in a total of $O(n)$ time. Furthermore, during a step $Near(b_L)$ is also obtained in constant time; so computing the *Near* sets also requires a total of $O(n)$ time. Explicitly evaluating the constants gives a running time of $O(n \log n + k \log(k/\varepsilon)4^k(3 + 12c_p/\varepsilon)^k n)$ for the *Procedure Construct-Near-sets*.

6. Remarks

The number of edges in G have an exponential dependence on the dimension k which could be as bad as $(ck)^k$ where c is a constant that does not depend on k. An open question is whether this dependence on the dimension can be reduced to $(c')^k$ where c' does not depend on k; furthermore we would like c' to be small, say less than or equal to 4. Better still is it possible to reduce the dependence on k to a polynomial in k at the expense of increasing the dependence on n by a small factor, say $\log n$?

References

1. P. Chew, There is a planar graph almost as good as the complete graph, *Proc. 2nd Annual Symposium on Computational Geometry*, 1986, pp. 169–177.
2. K. L. Clarkson, Fast algorithms for the all-nearest-neighbors problem, *Proc. 24th Annual Symposium on Foundations Computer Science*, 1983, pp. 226–232.
3. K. L. Clarkson, Approximation algorithms for shortest path motion planning, *Proc. 19th Annual ACM Symposium Theory of Computing*, 1987, pp. 56–65.
4. D. P. Dobkin, S. J. Friedman, and K. J. Supowit, Delaunay graphs are almost as good as complete graphs, *Discrete Comput. Geom.* **5** (1990), 399–408.
5. T. Feder, Personal communication, 1989.
6. R. A. Finkel and J. L. Bentley, Quad-trees: a data structure for retrieval on composite keys, *Acta Inform.* **4** (1974), 1–9.

7. F. P. Preparata and M. I. Shamos, *Computational Geometry: An Introduction*, Springer-Verlag, New York, 1985.
8. E. M. Reingold, J. Nievergelt, and N. Deo, *Combinatorial Algorithms: Theory and Practice*, Prentice Hall, Englewood Cliffs, NJ, 1977.
9. P. M. Vaidya, An $O(n \log n)$ algorithm for the all-nearest-neighbors problem, *Discrete Comput. Geom.* **4** (1989), 399–408.
10. P. M. Vaidya, Approximate minimum weight matching on points in k-dimensional space, *Algorithmica*, **4** (1989), 569–584.
11. A. C. Yao, On construction minimum spanning trees in k-dimensional space and related problems, *SIAM J. Comput.*, **11** (1982), 721–736.

Received April 14, 1988, and in revised form November 26, 1990.