A Deterministic Linear Time Algorithm for Geometric Separators and its Applications

David Eppstein[*] Gary L. Miller[†] Shang-Hua Teng[‡]

Abstract

We give a deterministic linear time algorithm for finding a small cost sphere separator of a k-ply neighborhood system Φ in any fixed dimension, where a k-ply neighborhood system in $\mathbb{R}^d$ is a collection of n balls such that no points in the space is covered by more than k balls. The sphere separator intersects at most $O\left(k^{\frac{1}{d}} n^{\frac{d-1}{d}}\right)$ balls of Φ and it divides the remaining of Φ into two parts: those in the interior and those in the exterior of the sphere, respectively, so that the larger part contains at most δn balls ($\frac{d+1}{d+2} < \delta < 1$). This result improves the $O(n^2)$ time deterministic algorithm of Miller and Teng [29] and answers a major algorithmic open question posed by Miller, Teng, Thurston and Vavasis [23, 25].

The deterministic algorithm hinges on the use of a new method for deriving the separator property of neighborhood systems. Using this algorithm, we devise an $O(kn + n \log n)$ time deterministic algorithm for computing the intersection graph of a k-ply neighborhood system. We give an $O(n \log n)$ time algorithm for constructing a linear space, $O(\log n)$ query time search structure for a geometric query problem associated with k-ply neighborhood systems, and we use this data structure in an algorithm for approximating the value of k within a constant factor in time $O(n \log n)$. We also develop a deterministic linear time algorithm for finding an $O\left(k^{\frac{1}{d}} n^{\frac{d-1}{d}}\right)$-separator for a k-nearest neighborhood graph in d dimensions.

[*]Dept. of Information and Computer Science, University of California, Irvine, CA 92717. Work done in part while the author was visiting Xerox PARC.

[†]School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213. Supported in part by National Science Foundation grant CCR-91-96113.

[‡]Department of Mathematics, MIT, Cambridge, MA 02139. Supported in part by AFOSR F49620-92-J-0125 and Darpa N00014-92-J-1799. Part of the work was done at Xerox PARC.

1 Introduction

Many problems in scientific computing (e.g., sparse Cholesky factorization) and computational geometry (such as space partition) require efficiently computing a small separator of the underlying graph [6, 7, 8, 9, 11, 18, 20, 27, 28, 29]. By small separator we mean a relatively small subset of vertices the removal of which divides the rest of the graph into two disconnected pieces of approximately equal size [19].

In a series of papers [32, 24, 26, 22, 25, 23, 29], Miller, Teng, Thurston and Vavasis have developed a geometric characterization of graphs (embedded in some fixed dimension) that have a small separator. Their characterization is based on a notion of a ***k*-ply neighborhood system**, which is a collection of n balls such that no points in the space is covered by more than k balls. Using this characterization, they gave randomized linear time and processor-efficient randomized NC algorithms for finding a *good* sphere separator – a sphere that has intersection number at most $O\left(k^{\frac{1}{d}} n^{\frac{d-1}{d}}\right)$ and has a constant splitting ratio.

Unfortunately, these previous results require the use of randomization—they are not deterministic. As part of his doctoral dissertation, Teng [29] devised an $O(n^2)$ time deterministic algorithm for finding a good sphere separator. But for many applications in scientific computing and computational geometry, an $O(n^2)$ time algorithm is too slow to be used. The result has been that the randomized linear time algorithm is used, and in general this is the only part in the application program that uses randomness [29]. For various applications such as sparse matrix solving in physical simulation and many of other problems in scientific computing, it is desirable to use a *repeatable* program, i.e., each run of the program on the same input produces the same output. To this end, it is important to have an efficient deterministic sphere separator algorithm.

In this paper, we give the first known deterministic linear time algorithm for finding a small cost sphere separator of a k-ply neighborhood system. Our algorithm works in any fixed dimension, and solves a major algorithmic open question posed by Miller, Teng, Thurston and Vavasis [25, 23]. We first provide a deterministic linear time reduction from the problem of finding sphere separators to an interesting geometric problem, of finding a point covered by a small subset of a system of annuli. We then develop a deterministic linear time algorithm for the second problem using ϵ-cuttings [21] as the basis for a novel geometric prune-and-search technique.

Our algorithm can be applied to several problems in computational geometry. The new algorithm removes the necessity of randomness from the (sphere separator based) divide and conquer paradigm developed in [29, 7]. More specifically, we devise an $O(kn + n\log n)$ time deterministic algorithm for constructing the intersection graph of a k-ply neighborhood system, improving and greatly simplifying the (quad-tree based) deterministic $O(kn\log n)$ time construction of [29]. We give a deterministic $O(n\log n)$ time algorithm for constructing a linear space, $O(\log n)$ query time search structure for a geometric query problem associated with k-ply neighborhood systems. We use this data structure in an algorithm for approximating the value of k within a constant factor in time $O(n\log n)$; the best known algorithm for computing the exact value of k involves constructing a sphere arrangement, and takes time $O(n^d \log n)$. We also develop a deterministic linear time algorithm for finding an $O\left(k^{\frac{1}{d}} n^{\frac{d-1}{d}}\right)$-separator for the k-nearest neighborhood graph in d dimensions.

The outline of this paper is as follows. In Section 2, we introduce the basic concepts and notation. We then review the techniques and results of [25, 23] that will be used later in the paper. In Section 3, we present a new approach for deriving the separator property of neighborhood systems and reduce the problem of finding 'good' sphere separator to a problem of finding a point in few annuli. In Section 4, we give a deterministic linear time algorithm for this problem. Some applications of the new result in computational geometry are presented in Section 5 and an open question is given in Section 6.

2 The Background

A d-dimensional *neighborhood system* $\Phi = \{B_1, \ldots, B_n\}$ is a finite collection of balls in $\mathbb{R}^d$. Let p_i be the center of B_i $(1 \le i \le n)$ and call $P = \{p_1, \ldots, p_n\}$ *centers* of Φ. For each point $p \in \mathbb{R}^d$, let the *ply* of p, denoted by $\text{ply}_\Phi(p)$, be the number of balls from Φ that contains p. Φ is a k-*ply* neighborhood system if for all p, $\text{ply}_\Phi(p) \le k$.

Each $(d-1)$-dimensional sphere S in $\mathbb{R}^d$ partitions Φ into three subsets: $\Phi_I(S)$, $\Phi_E(S)$, and $\Phi_O(S)$, those balls that are in the interior of S, in the exterior of S, and that intersect S, respectively. The cardinality of $\Phi_O(S)$ is called the *intersection number* of S, denoted by $\iota_\Phi(S)$.

Notice that the removal of $\Phi_O(S)$ splits Φ into two subsets: $\Phi_I(S)$ and $\Phi_E(S)$, such that no ball in $\Phi_I(S)$ intersects any ball in $\Phi_E(S)$ and vice versa. In analogy to separators in graph theory, $\Phi_O(S)$ can be viewed as a separator of Φ.

Definition 2.1 (Sphere Separators) *A $(d-1)$-sphere S is an $f(n)$-***separator** *that δ-***splits** *a neighborhood system Φ if $\iota_\Phi(S) \le f(n)$ and $|\Phi_I(S)|$, $|\Phi_E(S)| \le \delta n$, where f is a positive function and $0 < \delta < 1$.*

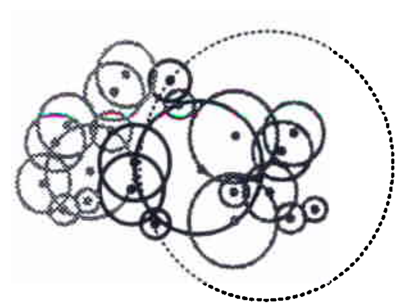

Figure 1: A sphere separator

With these definitions, the following theorem is proved in [23, 25]:

Theorem 2.2 (Sphere Separator Theorem) *Suppose $\Phi = \{B_1, \ldots, B_n\}$ is a k-ply neighborhood system in $\mathbb{R}^d$. Then there is a $(d-1)$-sphere S intersecting at most $O(k^{\frac{1}{d}} n^{\frac{d-1}{d}})$ balls from Φ such that both $|\Phi_I(S)|$*

and $|\Phi_E(S)|$ are at most $\leq \frac{d+1}{d+2}n$, where $\Phi_I(S)$ and $\Phi_E(S)$ are those balls that are in the interior and in the exterior of S, respectively.

We now reviews some of the basic concepts and lemmas used in [25, 24, 29, 23, 26] for proving Theorem 2.2.

A *density function* in $\mathbb{R}^d$ is a real valued nonnegative function $f(x)$ defined on $\mathbb{R}^d$ such that f^k is integrable for all $k = 1, 2, 3, \ldots$. The surface area of a $(d-1)$-dimensional sphere S is given by

$$\text{Area}_f(S) = \int_{v \in S} (f(v))^{d-1} (dv)^{d-1}$$

The total volume of the function f, denoted by Total-Volume(f), is given by

$$\text{Total-Volume}(f) = \int_{v \in \mathbb{R}^d} (f(v))^d (dv)^d$$

Density functions can also be defined on the surface of a sphere. To be consistent with the discussion of subsequent sections, we focus on the unit d-sphere. Suppose U_d is a unit d-sphere in $\mathbb{R}^{d+1}$ and f is a real valued nonnegative function defined on the surface of U_d such that f^k is integrable for all $k = 1, 2, 3, \ldots$. We call f a *density function* of U_d. The *total volume* of f is defined to be

$$\text{Total-Volume}(f) = \int_{v \in U_d} (f(v))^d (dv)^d$$

A *great sphere* of U_d is the intersection of U_d with a hyperplane passing through the center of U_d. The *weighted area* of a great sphere $\mathcal{G}$ of U_d is given by

$$\text{Area}_f(\mathcal{G}) \leq \int_{v \in \mathcal{G}} (f(v))^{d-1} (dv)^{d-1}$$

Let avg(f) be the average area over all great spheres of U_d. The following lemma follows from Hölder's inequality [13] [24, 26, 29, 25]. By A_d we denote the surface area of U_d.

Lemma 2.3 *Let f be a density function on U_d. Then*

$$\text{avg}(f) \leq A_{d-1} \left((\text{Total-Volume}(f))^{\frac{d-1}{d}} \right).$$

To use lemma 2.3 to find a small cost sphere separator, points of $\mathbb{R}^d$ are mapped onto the unit d-sphere U_d in $\mathbb{R}^{d+1}$ centered at the origin o. Notice that the density function f is mapped to a new density function f' on U_d in order to ensure Total-Volume(f) = Total-Volume(f'), after changing variables[1]. To preserve the density function in every dimension simultaneously, the map is required to be *conformal* in the sense that it preserves angles [15, 5, 30]. Rigid motions and dilations are examples of conformal maps. Another conformal map is the stereographic projection which maps $\mathbb{R}^d$ together with a point at infinity onto U_d [15, 24]. The composition of any two conformal maps is also conformal, so we can apply a sequence of conformal maps to obtain a desirable conformal map.

Conformal maps have another nice property: the pre-image of each great sphere $\mathcal{G}$ of U_d itself is a $(d-1)$-sphere in $\mathbb{R}^d$. Moreover, the interior and the exterior of the pre-image sphere are mapped to the two hemispheres of U_d defined by $\mathcal{G}$, respectively. It simply follows from Lemma 2.3 [24, 25, 29] that,

Lemma 2.4 *Suppose f is a density function of $\mathbb{R}^d$ and π is a conformal map from $\mathbb{R}^d$ to U_d. Let*

$$\mathcal{S}(\pi) = \{S : S \text{ is a pre-image of some great sphere } \mathcal{G}, \; i.e., \; \mathcal{G} = \pi(S)\},$$

and $\text{avg}_\pi(f)$ be the average area with respect to all projected great spheres in $\mathcal{S}(\pi)$. Then

$$\text{avg}_\pi(f) \leq A_{d-1} \left((\text{Total-Volume}(f))^{\frac{d-1}{d}} \right).$$

Suppose $P = \{p_1, \ldots, p_n\}$ is a set of points in $\mathbb{R}^d$ and π is a conformal map from $\mathbb{R}^d$ to U_d. Let $\pi(P) = \{\pi(p_1), \ldots, \pi(p_n)\}$. It follows from the above discussion of conformal mapping that for each great sphere $\mathcal{G}$ of U_d, if S is the pre-image of $\mathcal{G}$ then S δ-splits P iff $\mathcal{G}$ δ-splits $\pi(P)$.

Therefore, to ensure that each sphere of $\mathcal{S}(\pi)$ δ-splits P for some constant $0 < \delta < 1$, the conformal map π needs to satisfy the condition that all hyperplanes containing o δ-split Q. We use a conformal map such that o is a δ-center point of $\pi(P)$, where a point $c \in \mathbb{R}^d$ is a *δ-center point* of a set of points Q if every hyperplane containing c δ-splits Q. It follows from [29, 21, 31] that a δ-center point ($0 < \delta < \frac{d}{d+1}$) can be computed in deterministic linear time. Incorporating with a result of [23], we have:

[1] Technically speaking, there is a Jacobian factor in the integration for changing of variables from $\mathbb{R}^d$ to U_d. Fortunately, because the map is conformal, and the cost function has proper power, in the end, the "effect" of the Jacobian factor is cancelled out. For the simplicity of presentation, we will use the equations Total-Volume(f) = Total-Volume(f') and $\text{Area}_{f'}(S) = \text{Area}_f(\mathcal{G})$, for each great sphere $\mathcal{G}$ of U_d, where S is the pre-image of $\mathcal{G}$ in $\mathbb{R}^d$ of a comformal map.

Lemma 2.5 *A conformal map π such that o is a $(d+1)$-dimensional δ-center point of $\pi(P)$ can be computed in deterministic linear time.*

Suppose $\Phi = \{B_1, \ldots, B_n\}$ is a k-ply neighborhood system in $\mathbb{R}^d$. let r_i be the radius of B_i and let $\gamma_i = 2r_i$, define

$$f_i(x) = \begin{cases} 1/\gamma_i & \text{if } \|x - p_i\| \leq \gamma_i \\ 0 & \text{otherwise} \end{cases}$$

Miller, Teng, and Vavasis [25] introduced the density function below and proved the following two important lemmas:

$$\begin{aligned} f(x) &= L_{d-1}(f_1(x), \ldots, f_n(x)) \\ &= \left(\sum_{i=1}^{n} |f(x)|^{d-1}\right)^{1/(d-1)} . \end{aligned}$$

Lemma 2.6 ([25]) *Suppose $\Phi = \{B_1, \ldots, B_n\}$ is a k-ply neighborhood system in $\mathbb{R}^d$. There are constant c_1 and c_2 depending only on d such that for each $(d-1)$-sphere S, $\iota_\Phi(S) \leq c_1 k + c_2 \mathrm{Area}_f(S)$.*

Lemma 2.7 *Suppose $\Phi = \{B_1, \ldots, B_n\}$ is a k-ply neighborhood system in $\mathbb{R}^d$. There is a constant c_3 depending only on d such that*

$$\text{Total-Volume}(f) \leq c_3 k^{\frac{1}{d-1}} n.$$

3 Great Belts and Sphere Separators

As shown in Section 2, each d-dimensional neighborhood system $\Phi = \{B_1, \ldots, B_n\}$ can be conformally mapped to a unit d-sphere U_d so that each d-dimensional hyperplane containing the center of U_d δ-splits Φ ($\frac{d+1}{d+2} < \delta < 0$). Notice that each B_i is mapped to a patch D_i on U_d, whose boundary C_i has the shape of a (d-1)-sphere (see Figure 3). The radius of D_i is defined to be the radius of C_i. Clearly, the ply of $\{D_i : 1 \leq i \leq n\}$ is k iff the ply of Φ is k. Since the number of patches with surface area greater than a half of the surface area of U_d is bounded by $O(k)$, without loss of generality, we assume that each patch has a surface area no more than a half of the surface area of U_d. Let r_i be the radius of C_i. The volume of the d-dimensional ball with boundary C_i is then equal to $V_d(r_i)^d$, which is a lower bound on the surface area of the patch D_i. In the remaining of this section, we will identify B_i with D_i, and assume that $\Phi = \{B_1, \ldots, B_n\}$ is given on the unit d-sphere.

3.1 Great belts

Each great sphere $\mathcal{G}$ can be identified with the pair of points $p_\mathcal{G}$ and $q_\mathcal{G}$ on U that lie on the normal to $\mathcal{G}$ (see Figure 2 for an example in 2 dimensions). There is a dual relation between points on U_d and their great spheres:

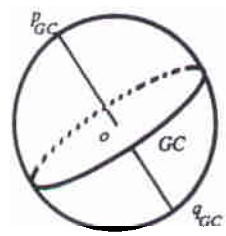

Figure 2: GC and its $p_\mathcal{G}$, $q_\mathcal{G}$

Proposition 3.1 (Duality) *For each pair of great spheres $\mathcal{G}$ and $\mathcal{G}'$ of U_d, $\mathcal{G}$ contains $p_{\mathcal{G}'}$ (and hence $q_{\mathcal{G}'}$ as well) if and only if $\mathcal{G}'$ contains $p_\mathcal{G}$ (and hence $q_\mathcal{G}$).*

Define a *great belt*[2] be the set of points of U_d that lie between a pair of parallel hyperplanes symmetric to the center of U_d. The *width* of a great belt is then the distance between its two hyperplanes. Notice that a great sphere is a great belt with width 0. Clearly, the surface area of a great belt of width r is bounded by $A_{d-1}r$. It simply follows from Proposition 3.1 that

Lemma 3.2 *Suppose $\Phi = \{B_1, \ldots, B_n\}$ is a neighborhood system on U_d. Then for each $1 \leq i \leq n$, there is a great belt R_i such that a great sphere $\mathcal{G}$ intersects B_i iff $p_\mathcal{G}$ and $q_\mathcal{G}$ is contained in R_i. Moreover, the width of R_i is equal to $2r_i$, where r_i is the radius of B_i (see Figure 3).*

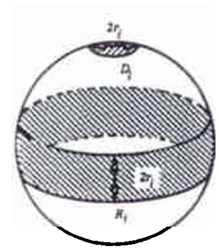

Figure 3: The great belt induced by B_i

[2] In [29], a great belt is called a great ring.

For each point x on U_d, let $\iota_\Phi(x)$ be the number of balls in Φ that intersect the great sphere $\mathcal{G}(x)$ associated with x. Let

$$\psi(\Phi) = \frac{1}{A_d}\left(\int_{x\in U_d} \iota_\Phi(x)(dx)^d\right).$$

By Proposition 3.1, we have

Proposition 3.3 *The above defined $\psi(\Phi)$ is equal to the expected intersection number of a random great sphere of U_d.*

Lemma 3.4 *Suppose $\Phi = \{B_1,\ldots,B_n\}$ is a neighborhood system on U_d. Let R_i be the great belt defined by B_i. Then*

$$\psi(\Phi) = \frac{1}{A_d}\left(\sum_{i=1}^{n} \text{Area}(R_i)\right)$$

Proof: For each $1 \le i \le n$, for each $x \in U_d$, Let $\Phi_i(x)$ be the function which takes value 1 if $\mathcal{G}(x)$ intersects B_i and 0 otherwise.

$$\begin{aligned}
\psi(\Phi) &= \frac{1}{A_d}\left(\int_{x\in U_d} \iota_\Phi(x)(dx)^d\right)\\
&= \frac{1}{A_d}\left(\int_{x\in U_d} [|\{B_i : \mathcal{G}(x)\cap B_i \neq \emptyset\}|](dx)^d\right)\\
&= \frac{1}{A_d}\left(\sum_{i=1}^{n}\left[\int_{x\in U_d} \Phi_i(x)(dx)^d\right]\right)\\
&= \frac{1}{A_d}\left(\sum_{i=1}^{n}[\text{Area}(R_i)]\right)
\end{aligned}$$

The first equality follows from the definition of $\iota_\Phi(x)$; the second equality is obvious; and the third equality follows from Lemma 3.2. □

3.2 The total area of great belts

Lemma 3.5 *Suppose $\Phi = \{B_1,\ldots,B_n\}$ is a k-ply neighborhood system on U_d. Let R_i be the great belt defined by B_i. Then*

$$\sum_{i=1}^{n} \text{Area}(R_i) = O\left(k^{\frac{1}{d}} n^{\frac{d-1}{d}}\right).$$

Proof: Let π be the stereographic projection which maps $\mathbb{R}^d$ onto U_d. Let Φ' be the pre-image of Φ in $\mathbb{R}^d$, i.e., $\Phi' = \{B_1',\ldots,B_n'\}$ such that $\pi(B_i') = B_i$. Clearly, Φ' is a neighborhood system in $\mathbb{R}^d$, whose density is also k. Let f_i' be the local density function defined on B_i' as in Section 2 and let $f' = L_{d-1}(f_1',\ldots,f_n')$. It follows from Lemma 2.7 that

$$\text{Total-Volume}(f') = O\left(k^{\frac{1}{d-1}} n\right).$$

Let f be the density function on U_d obtained from f' via the stereographic projection π, after changing of variables. We have

$$\text{Total-Volume}(f) = \text{Total-Volume}(f') = O\left(k^{\frac{1}{d-1}} n\right).$$

It follows from Lemma 2.3, that $\text{avg}(f) = O\left(k^{\frac{1}{d}} n^{\frac{d-1}{d}}\right)$. Recall that $\text{avg}(f)$ denotes the average area of great spheres of U_d. We now relate $\text{avg}(f)$ with $\psi(\Phi)$.

For each great sphere $\mathcal{G}$ of U_d, let S be the pre-image of $\mathcal{G}$ in $\mathbb{R}^d$, i.e., S is a $(d-1)$-sphere $\mathbb{R}^d$ such that $\pi(S) = \mathcal{G}$. By the definition of f and the fact that π is conformal, $\text{Area}_{f'}(S) = \text{Area}_f(\mathcal{G})$. Furthermore, the intersection number of $\mathcal{G}$ over Φ is equal to the intersection number of S' over Φ'. It follows from Lemma 2.6 that $\iota_\Phi(p_\mathcal{G}) \le c_1 k + c_2\text{Area}_f(\mathcal{G})$. In other words, for each point $x \in U_d$, $\iota_\Phi(x) \le c_1 k + c_2 \text{Area}_f(\mathcal{G}(x))$. Therefore,

$$\begin{aligned}
\psi(\Phi) &= \frac{1}{A_d}\left(\int_{x\in U_d} \iota_\Phi(x)(dx)^d\right)\\
&\le \frac{1}{A_d}\left(\int_{x\in U_d} [c_1 k + c_2\text{Area}_f(\mathcal{G}(x))]\,(dx)^d\right)\\
&= \frac{1}{A_d}(c_1 A_d k + c_2 A_d \text{avg}(f))\\
&= c_1 k + c_2 \text{avg}(f)\\
&= O\left(k^{\frac{1}{d}} n^{\frac{d-1}{d}}\right)
\end{aligned}$$

The lemma then follows from Lemma 3.4. □

4 The ϵ-cutting and the Linear time Algorithm

From the last section, we have $\sum_{i=1}^{n}\text{Area}(R_i) \le O(k^{\frac{1}{d}} n^{\frac{d-1}{d}})$. Hence, there is a point on U_d which is contained in at most $O(k^{\frac{1}{d}} n^{\frac{d-1}{d}})$ great belts from $\{R_1,\ldots,R_n\}$. Let $\text{ply}(p)$ denote the number of great belts from $\{R_1,\ldots,R_n\}$ containing $p \in U_d$. For a given subset S of U_d, let $\text{ply}(S)$ be the average value of ply(p) over those points in S; that is, $\text{ply}(S) = \sum_{i=1}^{n} \text{Area}(R_i \cap S)/\text{Area}(S)$.

Now the problem of computing a sphere separator of small intersection number is reduced to the following geometry problem:

- Given a set of great belts $\{R_1, \ldots, R_n\}$ on U_d, compute a point p on U_d such that $\text{ply}(p) \leq \text{ply}(U_d)$.

In [29], an $O(n^2)$ time deterministic algorithm is given for this problem. We now show a deterministic linear time algorithm. The basic idea is the following. We reduce the portion of the sphere under consideration, starting from the whole sphere, and proceeding through smaller and smaller "spherical" simplices, maintaining the property that in each simplex S the average covering number ply(S) is small (i.e., $\text{ply}(S) \leq \text{ply}(U_d)$). When we reach a small enough simplex, there will be few enough belts with boundaries crossing the simplex that we can solve the problem by brute force.

We find the sequence of simplices using a construction known as an ϵ-cutting [14, 21, 31]. This is a partition of space normally used for geometric divide-and-conquer techniques in problems such as simplex range searching [14, 21]. In our application, we instead use ϵ-cuttings as the basis of a prune-and-search technique, in which we reduce the problem to smaller and smaller simplices in which fewer and fewer great belts are involved.

A *cutting* is a partition of $\mathbb{R}^d$ into simplices. Given a set of n hyperplanes in $\mathbb{R}^d$, an ϵ-cutting is a cutting for which each simplex is crossed by at most ϵn hyperplanes. Matoušek [21] showed that an ϵ-cutting involving $O(\epsilon^{-d})$ simplices can be computed in time $O(n\epsilon^{-d-1})$. The number of simplices is within a constant factor of optimal, and the time bound is tight if one wants to enumerate the hyperplanes crossing each simplex.

This fits our present application as follows. If we imagine the sphere U_d as embedded in $\mathbb{R}^{d+1}$, the $2n$ $(d-1)$-spheres forming the boundaries of great belts can be extended to d-dimensional hyperplanes. A partition of space into simplices will partition U_d into "spherical polytopes" of complexity $O(1)$; with a further triangulation step these polytopes can be partitioned into spherical simplices, without further increase in complexity. At most $2\epsilon n$ great belt boundaries will cross any given spherical simplex.

For any given spherical simplex S, we can compute the average number of great belts covering each point, ply(S), by finding the spherical polytopes formed by intersecting each great belt with the simplex, adding the volumes of all such polytopes, and dividing by the volume of the whole simplex. Each polytope has at most $d + 3 = O(1)$ facets, and there are $O(\epsilon^{-d})$ simplices for which we can compute this quantity, so this computation takes $O(n\epsilon^{-d})$ time.

If U_d is partitioned into simplices S_i, the average number of belts covering points on the whole sphere, ply(U_d), can be computed as the weighted average $\text{ply}(U_d) = \sum_{i=1}^{k} \text{ply}(S_i)\text{Area}(S_i)/\text{Area}(U_d)$. The minimum is at most the average, so for some simplex S_i, $\text{ply}(S_i) \leq \text{ply}(U_d)$. Thus if ϵ is some fixed fraction, in linear time we can reduce the problem of searching for a point in U_d to that of searching for one in S_i. All but at most $2\epsilon n$ great belts (referred as *active belts*) either miss S_i entirely, or entirely cover S_i; in either case we need not account for them in future computation. Thus we have reduced the problem to one in which there are $O(\epsilon n)$ great belts.

Similarly, if we start with a simplex S with $\text{ply}(S) \leq \text{ply}(U_d)$, for which some number m of the great belts remain neither completely covering S nor completely avoiding S, we can in $O(m)$ time reduce the problem to a smaller simplex S', for which $\text{ply}(S') \leq \text{ply}(S) \leq \text{ply}(U_d)$, and for which $O(\epsilon m)$ belts remain "active". If we iterate this approach, we eventually reach a final simplex F in which no belts are active. Then ply(x) is a constant function for all points $x \in F$, and $\text{ply}(x) = \text{ply}(F)$. Any such point from this simplex gives the desired solution to our geometric problem, and hence (returning from dual points to their primal great spheres) the geometric separator we are seeking. The time for this whole series of reductions will be $\sum_{i=0}^{\log n} O(n\epsilon^i) = O(n)$. We have proved the following results.

Theorem 4.1 *Suppose we are given a set R of n great belts on U_d. Then in time $O(n)$ we can compute a point $x \in U_d$, with $\text{ply}(x) \leq \text{ply}(U_d)$.*

Theorem 4.2 *Given a k-ply neighborhood system of n balls in $\mathbb{R}^d$, we can find an $O(k^{\frac{1}{d}} n^{\frac{d-1}{d}})$ sphere separator that δ-splits the system for any $\frac{d+1}{d+2} < \delta < 1$ in time $O(n)$.*

5 Applications

The sphere separator theorem has many applications [29], especially in scientific computing and computational geometry [18, 11, 27, 8, 9, 12]. In this section, we focus on the application of the deterministic linear time sphere separator algorithm in computational geometry.

5.1 Separators for geometric graphs

As shown in [25, 29], the sphere separator results lead to vertex separators for various geometric graphs including the intersection graph and the k-nearest-neighborhood graph.

The *intersection graph* of a neighborhood system $\Phi = \{B_1, \ldots, B_n\}$ has vertices $V = \Phi$ and edges $E = \{(B_i, B_j) : B_i \cap B_i \neq \emptyset\}$. It simply follows from the definition of the intersection graph [29] that if S is a sphere that δ-splits the centers of a neighborhood system Φ, then $\{B_i : B_i \cap S \neq \emptyset\}$ is an $\iota_\Phi(S)$-separator that δ-splits the intersection graph of Φ. Therefore,

Theorem 5.1 *Given a k-ply neighborhood system Φ in $\mathbb{R}^d$, an $O(k^{\frac{1}{d}} n^{\frac{d-1}{d}})$-separator of the intersection graph of Φ can be computed in linear time deterministically.*

Suppose $P = \{p_1, \ldots, p_n\}$ is a set of n points in $\mathbb{R}^d$. For each $p_i \in P$, let $N(p_i)$ be a closest neighbor of p_i in P, where ties are broken arbitrarily. Similarly, for any integer k, let $N_k(p_i)$ be the set of k nearest neighbors of p_i in P; here too ties are broken arbitrarily.

A *k-nearest neighborhood graph* of $P = \{p_1, \ldots, p_n\}$ in $\mathbb{R}^d$, is a graph with vertices $V = P$, and edges $E\{(p_i, p_j) | p_i \in N_k(p_j) \text{ or } p_j \in N_k(p_i)\}$.

For each i, let B_i be the largest ball centered at p_i whose interior contains at most $k-1$ points from P. Then, $\{B_1, \ldots, B_n\}$ is called *the k-nearest neighborhood system* of P. It has been shown in [25, 29] that the k-neighborhood system of P is $\tau_d k$-ply, where τ_d is the kissing number in d dimensions, i.e., the maximum number of nonoverlapping unit balls in $\mathbb{R}^d$ that can be arranged so that they all touch a central unit ball [3].

By the definition of k-nearest neighborhood graph, there is an edge between balls B_i and B_j only if either p_i is in B_j or p_j is in B_i and hence each k-nearest neighborhood graph in $\mathbb{R}^d$ is a subgraph of a k-neighborhood system in $\mathbb{R}^d$.

Corollary 5.2 *Let P be a set of n points in $\mathbb{R}^d$. An $O(k^{\frac{1}{d}} n^{\frac{d-1}{d}})$-separator of the k-nearest neighborhood graph of P can be computed in $O(kn)$ time if the k-nearest neighborhood graph is given. Otherwise, it can be computed in $O(kn \log k + n \log n)$ time.*

The second part of the corollary follows from results of Vaidya [33] and Drysdale [4] that the k-nearest neighborhood graph of a set of points can be computed in $O(kn \log k + n \log n)$ time.

5.2 Separator Based Divide and Conquer

The sphere separator result for neighborhood systems closed under the subset operation immediately leads to divide and conquer recursive algorithms for a variety of applications [29, 7]. Frieze, Miller, and Teng [7] developed a sphere separator based divide and conquer paradigm for solving problems in geometry. At a high level, the separator based divide and conquer paradigm is very simple and intuitive. Given a neighborhood system Φ (explicitly or implicitly), it finds a sphere separator of a low intersection number and partitions the balls into two subsets of roughly equal size, and then recursively solves the problem associated with the two sub-systems; the solutions for the two sub-systems are then combined to a solution to the whole problem. Our new result removes the necessity of randomness from the separator based divide and conquer paradigm. We now illustrate some applications.

5.2.1 Neighborhood query problem

The *neighborhood query problem* is defined as: given a neighborhood system $\Phi = \{B_1, \ldots, B_n\}$ in d-space, preprocess the input to organize it into a search structure so that queries of the form "output all neighborhoods that contain a given point p" can be answered efficiently.

Like other geometry query problems, there are three costs associated with the neighborhood query problem: the *preprocessing* time $T(n)$ required to build the search structure, the *query* time $Q(n)$ required to answer a query, and the *space* $S(n)$ required to represent the search structure in memory.

The separator based divide and conquer construction for this problem is very simple [29, 7]: Given a k-ply neighborhood system Φ, a binary tree of height $O(\log n)$ is built to guide the search in answering a query. Associated with each leaf of the tree is a subset of balls in Φ, and the search structure has the property that for all $p \in \mathbb{R}^d$, the set of balls covering p can be found in one of the leaves. The binary search tree is constructed as follows:

ALGORITHM Neighborhood_Querying

1. Find a 'good' sphere separator S;
2. Let $\Phi_0 = \Phi_I(S) \cup \Phi_O(S)$ and $\Phi_1 = \Phi_E(S) \cup \Phi_O(S)$;
3. Recursively construct the search structure T_0 for Φ_0 and T_1 for Φ_1;
4. Construct a tree whose left subtree is T_1 and whose right subtree is T_2, and whose root stores the information of S, i.e., its center and its radius.

The recursive construction stops when the subsets have cardinality smaller than $\max(\log n, ck)$ for some constant c large enough that our sphere separator algorithm can find a nontrivial separator.

To answer a query when given a point $p \in \mathbb{R}^d$, we first check p against S, the sphere separator associated with the root of the search tree. There are three cases: (1) If p is in the interior of S then recursively search on the left subtree of S; (2) If p is in the exterior of S then recursively search the right subtree of S; and (3) If p is on S then recursively search on the left subtree of S. When reaching a leaf, we then check p against all neighborhoods associated with the leaf and output all those that cover p. It has been shown in [29] that the above search structure and searching procedure is correct, and moreover, the query time is $Q(n) = O(\log n + k)$ and the space requirement is $S(n) = O(n)$.

We now analyze the time complexity of the above algorithm when the deterministic linear time sphere separator algorithm is used.

Clearly, Step (1) takes $O(n)$ time. Step (2) is has two recursive call. Let $\alpha = \frac{d-1}{d}$. We have $|\Phi_0|, |\Phi_1| \leq \delta n + n^{\alpha}$, and $|\Phi_1| + |\Phi_2| \leq n + n^{\alpha}$. Step (3) takes constant time.

Let $T(m)$ be the time to build a search structure for a neighborhood system of m balls. We have $T(m) \leq 1$ if $m \leq \log n$, and $T(\delta_1 m + k^{\frac{1}{d}} m^{\alpha}) + T((1-\delta_1)m) + O(m)$ if $m > \log n$, where $\delta_1 \leq \delta$. We can show that $T(n) = O(n \log n)$.

Theorem 5.3 *A search structure for the neighborhood query problem with $Q(n) = O(k + \log n)$ and $S(n) = O(n)$ can be computed in $O(n \log n)$ time.*

5.2.2 Computing the ply

The construction above assumes that the value of k is known, but we can achieve the same bounds if we are only given a neighborhood system but not told its ply. In that case we cannot stop when $m \leq ck$. since we do not know k, but we can instead find a good lower bound for k by examining the cardinality of the sphere separators produced by our algorithm. Indeed, this technique gives us a method for approximating the ply of the neighborhood system:

Theorem 5.4 *Given a neighborhood system, we can compute a number k' such that $k' \leq k \leq ck'$ for some absolute constant c, in time $O(n \log n)$.*

Proof: Algorithm Neighborhood_Querying constructs a separator tree such that the spheres covering any single point will be found at some leaf of the tree, so in particular some leaf will contain at least k spheres. Further, even using a lower bound estimate for k instead of k itself, the algorithm will subdivide the problem until all leaves contain at most ck spheres. Therefore we can estimate k by examining the leaf containing the largest number of spheres.

In contrast, the best known algorithm for finding the exact ply of a neighborhood system works by constructing the corresponding sphere arrangement, and takes time $O(n^d \log n)$.

The approach above also reduces the problem of computing exact ply of a neighborhood system of n balls to $O(n/k)$ problems, each with $O(k)$ balls. Therefore,

Theorem 5.5 *We can compute the ply of a given neighborhood system in $O(n \log n + k^{d-1} n \log k)$ time.*

It worthwhile to mention that for variable dimension, computing the exact ply of a neighborhood system is NP-complete (the special case of halfspaces through the origin is equivalent to OPEN HEMISPHERE [16]).

5.2.3 Constructing intersection graphs

Again, we use the sphere separator based divide and conquer paradigm. The algorithm can be described as:

ALGORITHM Intersection_Graph

1. Find a 'good' sphere separator S;
2. Let $\Phi_0 = \Phi_I(S) \cup \Phi_O(S)$ and $\Phi_1 = \Phi_E(S) \cup \Phi_O(S)$;
3. Recursively construct the intersection graphs G_0 for Φ_0 and G_1 for Φ_1;
4. Remove all parallel edges in $G_1 \cup G_2$ to obtain the intersection graph G of Φ;

It had been proved in [29] that the intersection graph of a k-ply neighborhood system has at most $O(kn)$ edges; therefore the above algorithm takes time $O(kn \log n)$. We can speed this up by eliminating the generation of parallel edges: we generate all edges in Φ_0, but only those edges in Φ_1 that do not have both endpoints in S. We keep track of which separators each vertex is contained in using an $O(\log n)$-bit bit-vector; at the bottom level of the recursion, an edge is generated only if the bitwise and of the vectors for its two endpoints is zero. Thus we can show

Theorem 5.6 *The intersection graph of a k-ply neighborhood system Φ in $\mathbb{R}^d$ can be computed in $O(kn + n\log n)$ time.*

6 Open Questions

One interesting problem remains open is whether we can find in $O(n)$ time a point with the smallest ply when given a set of great belts on U_d. An affirmative answer will enable us to find the optimal sphere separator of a k-ply neighborhood system with respect to a given map. Is that an efficient appromixation algorithm for this problem?

Acknowledgments We would like to thank Steve Vavasis for helpful discussions.

References

[1] Bentley, J. L. "Multidimensional divide-and-conquer". CACM 23: 214–229, 1980.

[2] Bern, M., D. Eppstein and J. R. Gilbert. "Provable good mesh generation". In *31st Annual Symposium on Foundations of Computer Science, IEEE*, 231–241, 1990.

[3] Conway, J. H., and N. J. A. Sloane. *Sphere Packings, Lattices and Groups.* Springer-Verlag, 1988.

[4] Drysdale, S. "Solving closest-point problems by searching the Delaunay diagram". In *2nd MSI Workshop on Computational Geometry*, SUNY Stony Brook, 1992.

[5] Dubrovin, B. A., A. T. Fomenko, and S. P. Novikov. *Modern Geometry – Methods and Applications.* Springer–Verlag, 1984.

[6] Fredickson, G. N., and R. Janardan. "Separator-based strategies for efficient message routing". In *27st Annual Symposium on Foundation of Computation Science,IEEE*, 428–237, 1986.

[7] Frieze, A. M., G. L. Miller, and S.-H. Teng. "Separator based parallel divide and conquer in computational geometry". In *4th Annual ACM Symposium on Parallel Algorithms and Architectures*, pp 420-430, 1992.

[8] George, J. A., "Nested dissection of a regular finite element mesh". *SIAM J. Numerical Analysis*, 10: 345–363, 1973.

[9] George, A and J. W. H. Liu. *Computer Solution of Large Sparse Positive Definite Systems.* Prentice-Hall, 1981.

[10] Gilbert, J. R., J. P. Hutchinson, and R. E. Tarjan. "A separation theorem for graphs of bounded genus". J. Algorithms, 5, 391–407, 1984.

[11] Gilbert, J. R. and R. E. Tarjan. "The analysis of a nested dissection algorithm". *Numerische Mathematik*, 50(4):377–404, 1987.

[12] Hammond, S. and R. Schreiber. "Solving unstructured grid problems on massively parallel computers". Technical Report TR 90.22, Research Institute for Advanced Computer Science, 1990.

[13] Hardy, G., J. E. Littlewood and G. Pólya. *Inequalities* Second edition, Cambridge University Press, 1952.

[14] Haussler, D. and E. Welzl. "ϵ-net and simplex range queries". *Discrete & Computational Geometry*, 2: 127–151, 1987.

[15] Hilbert, D. and S. Cohn-Vossen. *Geometry and the Imagination.* (Translated by P. Nemenyi), Chelsea Publishing Company, New York, 1952.

[16] Johnson, D. S., and F. P. Preparata. "The densest hemisphere problem". *Theoretical Computer Science* 6: 93–107, 1978.

[17] Knuth, D. E. *The Art of Computer Programming.* vol 1, vol 2, vol 3. Addison-Wesley, 1973.

[18] Lipton, R. J., D. J. Rose, and R. E. Tarjan. "Generalized nested dissection". *SIAM J. on Numerical Analysis*, 16:346–358, 1979.

[19] Lipton, R. J. and R. E. Tarjan. "A separator theorem for planar graphs". *SIAM J. of Appl. Math.*, 36:177–189, April 1979.

[20] Lipton, R. J. and R. E. Tarjan. "Applications of planar separator theorem". *SIAM J. Comput*, 9(3): 615–627, August 1981.

[21] Matoušek, J. "Approximations and optimal geometric divide-and-conquer". In *Proceedings of the 23rd Annual ACM Symposium on Theory of Computing.* 505–511, 1991.

[22] Miller, G. L. and S.-H. Teng. *Center Points and Point Divisions.* manuscript, School of Computer Science, Carnegie Mellon University, 1990.

[23] Miller, G. L., S.-H. Teng, W. Thurston, and S.A. Vavasis. "Automatic Mesh Partitioning." To appear in *Proceedings of the 1992 Workshop on Sparse Matrix Computations: Graph Theory Issues and Algorithms*, Institute for Mathematics and its Applications.

[24] Miller, G. L. and W. Thurston. "Separators in two and three dimensions". In *ACM STOC*, pages 300–309, Baltimore, May 1990.

[25] Miller, G. L., S.-H. Teng, and S. A. Vavasis. "An unified geometric approach to graph separators". In *32nd Annual Symposium on Foundations of Computer Science, IEEE*, to appear, 1991.

[26] Miller, G. L. and S. A. Vavasis. "Density graphs and separators". In *Second Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 331–336, San Fransciso, January 1991. ACM-SIAM.

[27] Pan, V. and J. Reif. "Efficient parallel solution of linear systems". In *ACM STOC*, pages 143–152, Providence,RI, May 1985.

[28] Strang, G., and G. J. Fix. *An Analysis of the Finite Element Method*, Prentice-Hall, 1973.

[29] Teng, S.-H. *Points, Spheres, and Separators: a unified geometric approach to graph partitioning.* Ph.D. Thesis, Carnegie Mellon University, CMU-CS-91-184, 1991.

[30] Thurston, W. P. *The Geometry and Topology of 3-manifolds.* Princeton University Notes, 1988.

[31] Vapnik, V. N. and A. Ya. Chervonenkis. "On the uniform convergence of relative frequencies of events to their probabilities. *Theory Probab. Appl.*, 16: 264-280, 1971.

[32] Vavasis, S. A. "Automatic domain partitioning in three dimensions". *SIAM J. Sci. Stat. Comp.*, July 1991. to appear.

[33] Vaidya, P. M. "An $O(n \log n)$ algorithm for the All-nearest-neighbors problem". *Discrete & Computational Geometry*,4:101–115, 1989.

[34] Yao, F. F. "A 3-space partition and its application." In *Proceedings of the 15th Annual ACM Symposium on Theory of Computing, ACM*, 258–263, 1983.