# Approximation Algorithms for Geometric Separation Problems

Joseph S. B. Mitchell*

July 13, 1993

**Abstract**

In computer graphics and solid modeling, one is interested in representing complex geometric objects with combinatorially simpler ones. It turns out that via a "fattening" transformation, one obtains a formulation of the approximation problem in terms of *separation*: Find a minimum-complexity surface that separates two sets. In this paper, we provide approximation algorithms for several geometric separation problems, including:

- Given a set of triangles $\mathcal{T}$ and a set $S$ of points that lie within the union of the triangles, find a minimum-cardinality set, $\mathcal{T}'$, of *pairwise-disjoint* triangles, each contained within some triangle of $\mathcal{T}$, that cover the point set $S$.
- Given finite sets of "red" and "blue" points in the plane, determine a simple polygon of fewest edges that separates the red points from the blue points. More generally, given finite sets of points of many color classes, determine a planar "separating" subdivision of minimum combinatorial complexity, which has the property that each face of the subdivision contains points of at most one color class;
- Given two polyhedral terrains, $P$ and $Q$, over a common support set (e.g., the unit square), with $P$ lying above $Q$, compute a nested polyhedral terrain $R$ that lies between $P$ and $Q$ such that $R$ has a minimum number of facets.

Exact solution of the above problems in polynomial time is highly unlikely: The decision versions of all three problems are known to be NP-hard. We provide polynomial-time algorithms that are guaranteed to produce an answer within a logarithmic factor ($O(\log n)$, where $n$ is the complexity of the input problem instance) of optimal. (The error factor is constant in the orthogonal case — coverage by disjoint aligned rectangles, or separation of orthohedral terrains.) We also discuss extensions to higher dimensions.

## 1 Introduction

A fundamental problem in computer graphics and solid modeling is to represent highly complex geometric objects with much lower complexity approximations. A typical polyhedral solid model produced on a modern CAD system may have many thousands of facets. When many such models must be viewed or manipulated in a graphics environment or in a "virtual reality" setting, the complexity of the objects is overwhelming, and even supercomputers are unable to handle real-time computations. Hence, a standard approach to the problem is to replace complex models by simpler ones. The crudest approximation for a solid body may be to use its bounding box. These approximations can then be handled in real-time for purposes of display and/or interference detection. But a box is a very poor approximation to most objects. So, what one really desires is

*Applied Math, SUNY Stony Brook, NY 11794-3600. jsbm@ams.sunysb.edu

a sparsest possible approximate object, subject to the cocnstraint that the approximate object be "close" to the original in some sense — e.g., in Hausdorf distance.

Thus, we are motivated to study the *polyhedral approximation* problem: Given a complex polyhedron, and a tolerance value $\varepsilon$, compute a simpler polyhedron (having smaller "combinatorial complexity") that lies within "distance" $\varepsilon$ of the original. This problem is closely related to that of *polyhedral separation* in which one must compute a polyhedron of small complexity that separates two given disjoint polyhedra, $P$ and $Q$. In particular, by "fattening" a surface $\Sigma$ by an amount $\varepsilon$, one obtains a pair of new surfaces $P$ and $Q$ that "sandwich" $\Sigma$ between them. A minimum polyhedral separator of $P$ and $Q$ is a surface of least combinatorial complexity that approximates $\Sigma$ within a tolerance of $\varepsilon$. By computing a family of approximate surfaces, corresponding to various values of $\varepsilon$, one can construct a hierarchical representation of $\Sigma$, allowing the user the option to use a sparse representation when the exact shape of $\Sigma$ is irrelevant (e.g., when flying an airplane at 36,000 feet over a terrain), or a more detailed representation when the application calls for it (e.g., when flying at 1000 feet over the terrain).

In two dimensions, if the fattened region is an annulus, the methods of Aggarwal et al. [1] or Wang and Chan [21] solve this problem in $O(n \log n)$ time using methods based on link distance computation. For large values of $\varepsilon$, the fattening may create holes, in which case, one wants a minimum-vertex *simple* polygon surrounding all the holes of the fattened region. Guibas et al. [12] give an approximation algorithm for this problem; they also study other planar approximation methods based on finding minimum-link ordered stabbers of disks.

In three dimensions, the problem of finding a minimum-facet separator for two polyhedral solids is *NP*-complete. The problem remains *NP*-complete even if only one of the solids is nonconvex, or if the two polyhedra are convex nested polytopes [4, 5]. Thus, [16] were motivated to study the problem of *approximating* the optimal solution for polyhedral separation problems. They obtain a polynomial-time algorithm that computes a separator guaranteed to be within a factor $O(\log n)$ of optimal for the case of separating a convex polyhedron from a (possibly nonconvex) polyhedron. Clarkson [2] has obtained very recent results also on this problem. There has been work on the surface approximation problem in the graphics community [19, **?**]; however, these heuristic results provide no guarantee on the quality of the approximation they produce.

However, [16] left open many questions; we begin to answer some of these questions here. In particular, the results presented in this paper can be used to approximate *nonconvex* polyhedral terrain surfaces in three dimensions, getting within a log factor of optimal in general, but within a constant factor for the orthohedral case.

Another question posed by [16] is that of separating "red" points from "blue" points in the plane; this problem is intimately related to the terrain approximation problem. Recently, Fekete[11] has shown that it is NP-complete to determine if there exists a simple polygon having $k$ sides such that all "red" points lie within the polygon and all "blue" points lie outside. We give an approximation algorithm to get within a log factor of the minimum complexity red-blue separator. Our approximation gets within a constant factor for the rectilinear case. Further, our method extends to multiple color classes: If we are given a set of $n$ points, each labelled with one of $k$ distinct colors, we can produce a nearly-optimal subdivision having points of at most one color class in each face.

Fundamental to these problems is the question of *disjoint set cover*, in which one is required to cover a set of points using some specified class of sets, *but with the constraint that the sets used in the cover are pairwise-disjoint.* We use disjoint set cover as our starting point in this paper. We provide approximation algorithms for both the general and the rectilinear versions of the problem, obtaining log-factor and constant-factor bounds, respectively.

In summary, we provide the first polynomial-time approximation algorithms for the following problems:

- Given a set of triangles $\mathcal{T}$ and a set $S$ of points that lie within the union of the triangles, find a minimum-cardinality set, $\mathcal{T}'$, of *pairwise-disjoint* triangles, each contained within some triangle of $\mathcal{T}$, that cover the point set $S$. We call this problem the *Disjoint Set-Cover Problem.*

- Given finite sets of "red" and "blue" points in the plane, determine a simple polygon of fewest edges that separates the red points from the blue points. We call this problem the *Red-Blue Separation Problem.* More generally, given finite sets of points of many color classes, determine a planar "separating" subdivision of minimum combinatorial complexity, which has the property that each face of the subdivision contains points of at most one color class.

- Given two polyhedral terrains, $P$ and $Q$, over a common support set (e.g., the unit square), with $P$ lying above $Q$, compute a nested polyhedral terrain $R$ that lies between $P$ and $Q$ such that $R$ has a minimum number of facets.

Our approximation factors for the above problems are $O(\log n)$, where $n$ is the size of the input. In the case of *rectilinear* instances of our problems (e.g., disjoint set cover by rectangles or orthohedral terrain separation), our method yields constant-factor approximation algorithms.

Our approach to these problems is to transform the original problem into an easier, restricted separation/cover problem, which can be solved by dynamic programming. We show that an optimal solution to the restricted problem is within the claimed factor of an optimal solution to the original problem.

## 2 Disjoint Set Cover

Let $S$ be a set of $m$ points in the plane. Let $\mathcal{T}$ be a set of $n$ triangles whose union covers $S$. We make a nondegeneracy assumption that no three points among $S$ and the vertices of $\mathcal{T}$ are colinear and that no two such points have the same $x$-coordinate. Our problem is then to find a minimum-cardinality set, $\mathcal{T}'$, of *pairwise-disjoint* triangles $\Delta$ that cover the point set $S$, with each $\Delta$ contained fully within some triangle of $\mathcal{T}$. We call a triangle $\Delta$ that is fully contained within some triangle of $\mathcal{T}$ a *subtriangle.* Note that there is always a feasible solution to this problem, since we can always surround each point of $S$ with a very tiny triangle.

S. Fekete [11] has shown that it is NP-complete to decide if there exists a disjoint set cover of size $k$. Thus, we are motivated to devise approximation algorithms for the optimization problem.

First, let us note that the naive approach of taking a set cover of $S$, *ignoring the disjointness constraint*, can lead to very bad approximations. For example, one can use the greedy set cover heuristic of Johnson [13] and Lovász [14] to obtain a subset of $\mathcal{T}$ that covers $S$, such that the cardinality of this subset is within a factor $O(\log n)$ of optimal. However, such a subset may have much overlapping — indeed, the triangles could form a "grid" pattern. We could break this subset of triangles into subtriangles that are pairwise-disjoint, while still covering $S$, but the grid example shows that if we do this naively, we may end up *squaring* the cardinality of the approximate set cover.

Further, given $n$ triangles whose union, $U$, *has no holes*, there are examples to show that we may be required to use at least $\Omega(n^{1.5})$ pairwise-disjoint subtriangles to cover $U$ (or any set $S$ of points that has at least one point in each face of the arrangement). Figure 1 shows such an example for the case of coverage by rectangles (it is straightforward to adjust it for the case of triangles).

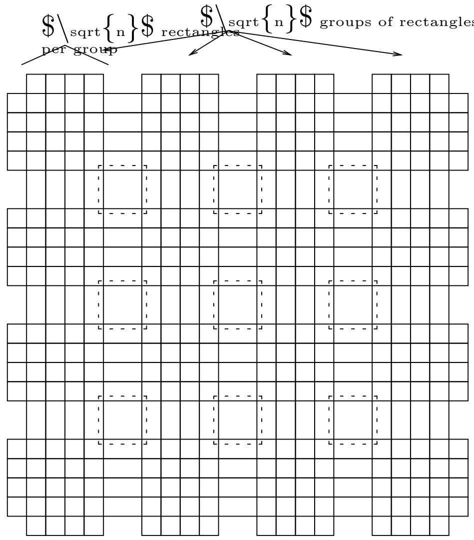


Figure 1: A case in which we require $\Omega(n^{1.5})$ subrectangles in order to cover $S$.

In the figure, we assume that $S$ has a point in each of the $O(n^2)$ faces of the arrangement. There are $\sqrt{n}$ groups of vertical (resp., horizontal) thin rectangles, each group of size $\sqrt{n}$. We also cover the $O(n)$ square holes with a set of rectangles (squares), shown dashed, so that the union of all rectangles has no holes.

It is not hard to see that the smallest disjoint cover by subrectangles is of size $\Omega(n^{1.5})$: Let $H$ denote one of the $O(n)$ square regions that lies at the intersection of one group of horizontal thin rectangles with some other group of vertical thin rectangles. There are $\sqrt{n}$ rectangular faces, $F = \{f_1, \ldots, f_{\sqrt{n}}\}$, ordered bottom to top, that belong to the $\sqrt{n}$ thin horizontal rectangles just right of $H$. Similarly, there are $\sqrt{n}$ faces, $G = \{g_1, \ldots, g_{\sqrt{n}}\}$, ordered right to left, that belong to the $\sqrt{n}$ thin vertical rectangles just below $H$. Consider a pair $(f_i, g_i)$. The face $f_i$ (resp., $g_i$) must be covered by some subrectangle, $\sigma(f_i)$ (resp., $\sigma(g_i)$). It is not possible for the left boundary of $\sigma(f_i)$ to lie left of $H$ *and* the top boundary of $\sigma(g_i)$ to lie above $H$; i.e., either $\sigma(f_i)$ has its left boundary lie in $H$, or $\sigma(g_i)$ has its top boundary lie in $H$, or both. Thus, each pair $(f_i, g_i)$ contributes at least one boundary segment (of some subrectangle) that lies within region $H$. Overall, then, $H$ contains at least $\sqrt{n}$ boundary segments of subrectangles, for any coverage by a set of subrectangles. Since there are a linear number of such regions $H$, we see that any coverage by subrectangles must have at least $\Omega(n^{1.5})$ boundary segments, and hence at least this many subrectangles.

Consider a set $\mathcal{T}'$ of $k$ pairwise-disjoint subtriangles that cover $S$. We now describe how to transform this set of triangles into a set of $O(k)$ "canonical subtrapezoids" $\tau$, where each $\tau$ is a subset of some member of $\mathcal{T}'$ (and hence of some member of $\mathcal{T}$) and the left and right boundary edges of $\tau$ are vertical. First, we split each triangle into two by a vertical cut through its middle vertex (middle with respect to $x$ ordering). We discard any triangle that does not contain at least one point of $S$. Next, for each of the (at most) $2k$ remaining triangles, we sweep inward with two vertical lines, one (sweeping rightward) starting at the leftmost point of the triangle, one (sweeping leftward) starting at the rightmost point of the triangle. Each vertical line stops when it first encounters a point of $S$. (They may encounter the same point of $S$, in which case the containing subtrapezoid will be the trivial "point trapezoid" at that point of $S$.) We are left with at most $2k$

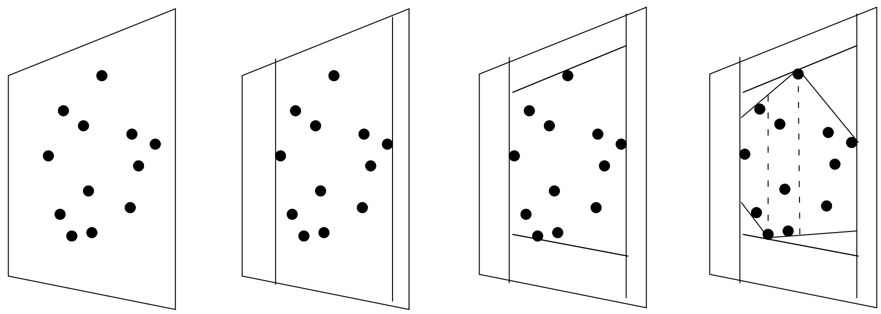

Figure 2: Obtaining canonical trapezoids from arbitrary trapezoids.

vertical-walled trapezoids that cover $S$.

Next, for each trapezoid we sweep its top boundary segment downward until it hits a point $p \in S$. If in fact it hits two points of $S$, we stop. Otherwise, we allow the segment to "fold" at $p$, and we allow the two subsegments to continue downward, pivoting at $p$, until each hits a second point of $S$. (Note that any of these points of $S$ that are hit may in fact be the same points that define the left/right vertical walls of the trapezoid.) We also split the trapezoid with a vertical cut through $p$. We are left with at most 2 new trapezoids, each a subtrapezoid of an original subtriangle of $\mathcal{T}'$, each with the property that the top boundary segment is defined by two points of $S$. Similarly, we sweep the bottom boundary of each of these trapezoids upwards, folding and splitting at some point $q \in S$.

The net result of the above procedure is that we have transformed the set $\mathcal{T}'$ of $k$ triangles into a set of $O(k)$ trapezoids $\tau$ having the following properties:

(1) the left/right sides of $\tau$ are segments through points of $S$ that are contained within $\tau$; and,

(2) the top/bottom sides of $\tau$ are segments through a pair of points of $S$ (assuming that $\tau$ is not a degenerate trapezoid consisting of a singleton point of $S$).

Consider a segment tree based on the $x$-coordinates, $x_1 < x_2 < \cdots < x_m$, of the points $S$ (see [18]). This segment tree induces a partitioning of each interval $[x_i, x_j)$ $(i < j)$ into $O(\log m)$ canonical subintervals (according to the "allocation nodes" that result from insert $[x_i, x_j)$ into the tree). (Note that there are at most $2m$ canonical intervals.) The segment tree therefore induces a partitioning of a subtrapezoid into $O(\log m)$ vertical-walled subtrapezoids, each of which can be horizontally "shrunk" until it obeys (1) and (2) above. These new subtrapezoids $\tau$ now also obey

(3) the $x$-projection of $\tau$ is a canonical interval according to the segment tree of the $x$-coordinates of $S$.

Thus, we are motivated to define a *canonical subtrapezoid* with respect to $S$ and $\mathcal{T}'$ to be a subtrapezoid $\tau$ (of some triangle of $\mathcal{T}'$) that obeys (1)–(3) above. Note that there are only a polynomial number of canonical subtrapezoids in total. We have proved:

**Lemma 2.1** *If there exists a set $\mathcal{T}'$ of $k$ pairwise-disjoint subtriangles (with respect to a set $\mathcal{T}$ of covering triangles) that cover $S$, then there exists a set of $O(k \log m)$ pairwise-disjoint canonical subtrapezoids that cover $S$.*

This lemma tells us that if we can solve exactly the problem of finding a disjoint cover of $S$ by canonical trapezoids, then we will have a method to approximate an optimal disjoint cover to within a factor $O(\log m)$. We now procede to solve the covering problem for canonical trapezoids by a recursion (dynamic programming).

Let $V([i,j),e_t,e_b)$ denote the minimum number of pairwise-disjoint canonical trapezoids required to cover the points of $S$ that lie within the trapezoidal region, $T(i,j,e_t,e_b)$, whose $x$-projection is the canonical interval $[x_i,x_j)$, whose top boundary is the segment $e_b$ (defined by some pair of points of $S$) and whose bottom boundary is the segment $e_t$ (defined by some pair of points of $S$). Our goal is to compute $V([1,m),E_t,E_b)$, where $E_t$ (resp., $E_b$) denotes a long horizontal segment lying above (resp., below) all of the points $S$; e.g., $E_t = [(x_1,\infty),(x_m,\infty)]$, and $E_b = [(x_1,-\infty),(x_m,-\infty)]$. Now, we get a recursion for $V([i,j),e_t,e_b) =$

0, if there are no points of $S$ within $T(i,j,e_t,e_b)$;

1, if there exists a subtrapezoid within $T(i,j,e_t,e_b)$ that contains $S \cap T(i,j,e_t,e_b)$;

otherwise,

$$\min\{ \quad \min_{e\subset T(i,j,e_t,e_b)}\{1+V([i,j),e_t,e)+V([i,j),e,e_b)\}, \\ V([i,\lfloor\tfrac{i+j}{2}\rfloor),e_t,e_b)+V([\lfloor\tfrac{i+j}{2}\rfloor,j),e_t,e_b)\}$$

where the minimization is over all segments $e \subset T(i,j,e_t,e_b)$ that are determined by a pair of points in $S \cap T(i,j,e_t,e_b)$, such that $e$ spans the interval $[x_i,x_j)$ and lies within region $T(i,j,e_t,e_b)$.

The rationale behind the above recursion is simple: In an optimal solution over the region $T(i,j,e_t,e_b)$, either there exists a canonical subtrapezoid $\tau$ of full width (spanning $[x_i,x_j)$), or there does not. If there does exist a spanning $\tau$, then the top edge $e$ of $\tau$ splits the problem into two new problems (with correspondingly new upper/lower bounding segments), and then the minimization over $e$ will optimize over all such partitions. If the optimal solution is not to use a spanning $\tau$, then the second term $(V([i,\lfloor\frac{i+j}{2}\rfloor),e_t,e_b)+V([\lfloor\frac{i+j}{2}\rfloor,j),e_t,e_b))$ gives us the optimal covering number corresponding to splitting the interval $[x_i,x_j)$ into two sub-canonical intervals.

The tabulation of the values of $V([i,j),e_t,e_b)$ is straightforward, starting with the smallest canonical intervals and working upwards. The evaluation clearly takes polynomial time.

**Theorem 2.2** *In polynomial time, one can compute a set of pairwise-disjoint subtriangles that cover $S$, with the number of subtriangles being within a factor $O(\log m)$ of optimal.*

Remarks:

- There is nothing that prevents $S$ from being a set of line segments or a set of polygons instead of a (finite) set of points.
- The above result extends to higher dimensions, with the approximation factor of $O(\log^{d-1} m)$: One partitions the space $\Re^{d-1}$ into canonical hyperrectangles, using multiple segment trees based on the first $d-1$ coordinates of the points $S$. The shrinking process is modified too: Instead of "folding" into two pieces at the point $p \in S$ that is hit when translating a top boundary downward, we must fold into $d$ pieces (giftwrapping, and using Caratheodory's Theorem).

### Rectilinear Case

In the special case that we are dealing with a set of aligned rectangles, $\mathcal{R}$, rather than a set of triangles $\mathcal{T}$, we can get an improvement in our approximation bound from logarithmic to constant.

Consider a set $\mathcal{R}'$ of $k$ pairwise-disjoint subrectangles that cover $S$. By a simple shrinking argument, each of these subrectangles can be assumed to have all four of its edges passing through points of $S$. Now, construct a binary space partition for the resulting set of subrectangles. By [17], this results in $O(k)$ new subrectangles, which can again be shrunk (if necessary) to have all four edges passing through points of $S$. Further, by the decomposition scheme of [17] (using "T-decompositions" and "free cuts"), the new set of subrectangles has the following property (which we call *Property-X*): there exists either a horizontal or a vertical non-trivial separating line, which separates the set of subrectangles into two nonempty sets.

Let $V(i,j,i',j')$ denote the cardinality of the smallest set of pairwise-disjoint subrectangles that cover all points of $S$ in the rectangular region $R(i,j,i',j') = \{(x,y) : x_i \leq x \leq x_j, y_{i'} \leq y \leq y_{j'}\}$ such that the set has Property-X. Then, $V(i,j,i',j')$ equals

0, if there are no points of $S$ within $R(i,j,i',j')$;

1, if there exists a subrectangle within $R(i,j,i',j')$ that contains $S \cap R(i,j,i',j')$;

otherwise, split the region $R(i,j,i',j')$ either with a horizontal or vertical line, and recurse:

$$\min\left\{\min_{\ell\in(i,j)}\left[V(i,\ell,i',j') + V(\ell+1,j,i',j')\right], \min_{\ell\in(i',j')}\left[V(i,j,i',\ell) + V(i,j,\ell+1,j')\right]\right\}$$

We can clearly tabulate all values of $V(i,j,i',j')$ in polynomial time; we do so in increasing size of $j-i$ and $j'-i'$.

**Theorem 2.3** *In polynomial time, one can compute a set of pairwise-disjoint subrectangles that cover $S$, with the number of subrectangles being within a factor $O(1)$ of optimal.*

Note that the polynomial time bound is rather high in the previous algorithm. We now give a very simple-minded approach that gets within a factor of $O(\log^2 m)$ in much less time.

Consider the partitioning of both the $x$- and $y$-axes into canonical intervals, according to the segment trees defined on the $x$- and $y$-coordinates of the points $S$. We refer to the rectangular region of the plane obtained by a product of an $x$-canonical and a $y$-canonical interval as a *canonical rectangle*. Then, any set of $k$ subrectangles that cover $S$ is partitioned into a set of $O(k\log^2 m)$ canonical rectangles in a natural way. For any pair of $x$- and $y$-canonical intervals, $I_x = [x_i, x_j)$ and $I_y = [y_{i'}, y_{j'})$, we get the following recursion for the number, $V(I_x, I_y)$, of disjoint subrectangles required to cover $S$: $V(I_x, I_y) = 1$, if there exists a single subrectangle covering all points of $S$ within the canonical rectangle $I_x \times I_y$; otherwise,

$$V(I_x,I_y) = \min\{V([x_i, x_{\lfloor\frac{i+j}{2}\rfloor}), I_y) + V([x_{\lfloor\frac{i+j}{2}\rfloor+1}, x_j), I_y), V(I_x, [y_{i'}, y_{\lfloor\frac{i'+j'}{2}\rfloor})) + V(I_x, [y_{\lfloor\frac{i'+j'}{2}\rfloor+1}, y_{j'}))\}$$

Now, note that there are only $O(m)$ choices for $I_x$ and only $O(m)$ for $I_y$. We can preprocess in batch to tabulate for each of the $O(m^2)$ choices of $(I_x, I_y)$ whether or not there is a single subrectangle covering the points of $S$ within $I_x \times I_y$. Thus, the recursion can be solved in total time $O(m^2)$.

**Theorem 2.4** *In $O(m^2 + n^2)$, one can compute a set of pairwise-disjoint subrectangles that cover $S$, with the number of subrectangles being within a factor $O(\log^2 m)$ of optimal.*

*Remark:* This same simple-minded approach can be used in the case of disjoint triangle cover as well, yielding a $\log^2$ approximation factor there.

## 3 Red-Blue Separation

Let $R$ and $B$ denote finite sets of "red" and "blue" points in the plane. Let $n$ denote the cardinality of $R \cup B$. Our goal is to compute a simple polygon $P$ that separates $R$ from $B$, such that $P$ has the fewest possible number of vertices. This red-blue separation problem is known to be NP-complete [11]. In this section, we give an $O(\log n)$-approximation algorithm that runs in polynomial time.

**Theorem 3.1** *Given n points in the plane, each colored "red" or "blue", one can find in polynomial time a simple polygon P that separates red points from blue points such that P has at most $O(\log n)$ times as many vertices as does a minimum-vertex separating simple polygon; i.e., there exists a polynomial-time $O(\log n)$-approximation algorithm for the Red-Blue Separation problem. If we restrict ourselves to rectilinear separating polygons, then there exists a polynomial-time $O(1)$-approximation algorithm for the Red-Blue Separation problem.*

*Proof Outline.* Consider any optimal red-blue separator $P^*$; let $k^*$ denote the number of vertices of $P^*$. Without loss of generality, assume that all red points lie inside $P^*$. Now, $P^*$ can be decomposed into $O(k^*)$ trapezoids, and each of these trapezoids can be shrunk to $O(1)$ canonical trapezoids; each of these canonical trapezoids will have only red points inside them. The result is a set $\mathcal{T}^*$ of $O(k^*)$ canonical trapezoids, each containing only red points. Let $\mathcal{T}$ denote the set of *all* canonical trapezoids having only red points within them.

Our approximation algorithm is to perform the disjoint set cover approximation algorithm with respect to the set $\mathcal{T}$ and the set $S$ of red points. We obtain a cover having at most $O(\log n) \times k^*$ disjoint trapezoids (or, in the rectilinear case, a constant-factor approximation). These trapezoids can be joined with $O(k^*)$ line segments (e.g., a spanning tree), yielding a connected set whose boundary complexity is $O(k^* \log n)$. ∎

*Remark:* Although the disjoint cover results carry over to higher dimensions, the red-blue separation results do not generalize immediately. The problem is that, for example in three dimensions one needs $O(k^2)$ tetrahedra in order to decompose a $k$-faceted polyhedron, which means that the above transformation to the disjoint set cover problem fails.

More generally, let $S_1, S_2, \ldots, S_k$ denote a partitioning of a set $S$ of $n$ points into $k$ color classes. Our goal is to compute a polygonal subdivision $\mathcal{S}$ that separates $S$ according to color classes, such that $\mathcal{S}$ has the fewest possible number of vertices. Since even the 2-color version is known to be NP-complete [11], so is this multi-colored version of the problem. We give an $O(\log n)$-approximation algorithm that runs in polynomial time; the method is a direct generalization of the red-blue case:

**Theorem 3.2** *Given n points in the plane, each colored with one of k colors, one can find in polynomial time a polygonal subdivision $\mathcal{S}$ that separates the points by color classes, such that $\mathcal{S}$ has at most $O(\log n)$ times as many vertices as does a minimum-vertex separating polygonal subdivision. If we restrict ourselves to rectilinear separating subdivisions, then there exists a polynomial-time $O(1)$-approximation algorithm for the problem.*

*Proof Outline.* Consider any optimal color-separating subdivision $\mathcal{S}^*$; let $k^*$ denote the number of vertices of $\mathcal{S}^*$. Each face of $\mathcal{S}^*$ is a simple polygon, which can be decomposed into a linear number (linear in the size of the polygon) of trapezoids, and each of these trapezoids can be shrunk to $O(1)$ canonical trapezoids; each of these canonical trapezoids will have only points of one color class inside them. The result is a set $\mathcal{T}^*$ of $O(k^*)$ canonical trapezoids, each containing only points of a single color class. Let $\mathcal{T}$ denote the set of *all* canonical trapezoids having only points of a single color class within them.

Our approximation algorithm is to perform the disjoint set cover approximation algorithm with respect to the set $\mathcal{T}$ and the set $S$ of all points. We obtain a cover having at most $O(\log n) \times k^*$ disjoint trapezoids (or, in the rectilinear case, a constant-factor approximation). These trapezoids can be joined with $O(k^*)$ line segments into a polygonal subdivision. ∎

# 4 Terrain Approximation

We now turn our attention to the problem of terrain surface separation:

**Theorem 4.1** *Given two polyhedral terrains, $P$ and $Q$, over a common support set (e.g., the unit square), with $P$ lying above $Q$, there exists a polynomial-time approximation algorithm to compute a nested polyhedral terrain $R$ that lies between $P$ and $Q$ such that $R$ has at most $O(\log n)$ times as many vertices as does a minimum-vertex nested terrain, where $n$ is the number of vertices of $P$ and $Q$.*

*Proof Outline.* Our method is to convert the problem into a disjoint set cover problem.

For a point $p$ on one of the terrains ($P$ or $Q$), let $p'$ denote the (unique) point on the other terrain ($Q$ or $P$) that lies vertically above or below $p$ (i.e., such that $p_x = p'_x$ and $p_y = p'_y$); we refer to $p'$ as the *partner* of $p$. For any set of points on the terrains, we can join each point to its partner and obtain a set of vertical line segments. Clearly, any nested terrain $R$ must stab every such vertical segment, for any (finite or infinite) set of points and their partners. We say that a trapezoid (or triangle) in 3-space is *valid* if it does not intersect $P$ or $Q$.

Our proof makes use of the following claim:

*There exists a* finite, polynomial-size *set $S$ of points on $P$ and $Q$ such that, if we find $k$ valid trapezoids (or triangles), with disjoint projections into the $(x, y)$-plane, that stab all of the vertical segments obtained by pairing each point of $S$ with its partner, then we can extend this set of trapezoids (or triangles) to a nested terrain surface of complexity $O(k)$.*

Once we have such a set $S$, we simply apply our earlier methods to find an approximately optimal set of valid *canonical* trapezoids whose projections onto the $(x, y)$-plane cover the projection of $S$. (Here, canonical means canonical with respect to $S$, in the projection onto the $(x, y)$-plane.) We can easily check each canonical trapezoid in the projected problem to see if it is valid — such a test is simply a test to see if there is a plane separating the two subsurfaces one obtains by restricting $P$ and $Q$ to lie within the vertical cylinder defined by the (projected) canonical trapezoid.

∎

### Rectilinear Case

Now we get a constant factor exactly as in the disjoint cover by rectangles. Further, the simple-minded approach gets within a factor of $O(\log^2 n)$, using only roughly quadratic time. (There is an additional log factor: A simple way to check if the region $I_x \times I_y$ has a single separating horizontal plane or not is to do rectangular range queries on the vertices of the upper and lower terrain surfaces, to find the highest and lowest points of each within the region.)

## 5 Conclusion

The main outstanding open problem is to solve the general nonconvex surface approximation problem. The methods used in this paper can probably yield a polynomial-time algorithm to get within a polylog factor of optimal. However, the exponents in the polynomial-time bound are likely to be very high.

An interesting practical problem is to devise very efficient, low-degree polynomial-time approximation methods for the separation problems we have discussed.

### Acknowledgements

I thank Esther Arkin, Sandor Fekete, and Subhash Suri for extensive discussions on this research.

## References

[1] A. Aggarwal, H. Booth, J. O'Rourke, S. Suri and C. K. Yap. "Finding minimal convex nested polygons," *Information and Control*, 83, pp. 98–110, 1989.

[2] K.L. Clarkson, "Algorithms for polytope covering and approximation," *Workshop on Algorithms and Data Structures*, to appear, 1993.

[3] G. Das. "Approximation schemes in computational geometry," Ph.D. Thesis, University of Wisconsin, 1990.

[4] G. Das and D. Joseph. "Minimum vertex hulls for polyhedral domains," *Symposium on Theoretical Aspects of Computer Science*, 1990, LNCS, Springer-Verlag.

[5] G. Das and D. Joseph. "The complexity of minimum convex nested polyhedra," *Canadian Conference on Computational Geometry*, 1990.

[6] D. Dobkin and D. Kirkpatrick. "Determining the separation of preprocessed polyhedra— a unified approach," *Proc. of ICALP '90*, pp. 400–413, 1990.

[7] H. Edelsbrunner. *Algorithms in Combinatorial Geometry*, volume 10 of *EATCS Monographs on Theoretical Computer Science*. Springer-Verlag, 1987.

[8] H. Edelsbrunner and F. P. Preparata. "Minimum polygon separation," *Information and Computation,* 77, pp. 218-232, 1987.

[9] H. Edelsbrunner, A. D. Robinson and X. Shen. "Covering convex sets with non-overlapping polygons," *Discrete Mathematics*, 81, pp. 153–164, 1990.

[10] M. Erwig, "???", Report 116, FernUniversität Hagen, 1991 (revised 1993).

[11] S. Fekete. Personal communication, 1992.

[12] L. J. Guibas, J. E. Hershberger, J. S. B. Mitchell, and J. S. Snoeyink. "Approximating polygons and subdivisions with minimum link paths," *Second Annual International Symposium on Algorithms*, December 16-18, 1991, Taipei, Taiwan, R.O.C. *Springer Lecture Notes in Computer Science*, Vol. 557, pp. 151–162.

[13] D. S. Johnson. "Approximation algorithms for combinatorial problems," *Journal of Computer and System Sciences*, 256-278, 1974.

[14] L. Lovász. "On the ratio of optimal integral and fractional covers," *Discrete Mathematics*, 383–390, 1974.

[15] N. Megiddo. "On the complexity of polyhedral separability," *Discrete and Computational Geometry*, pp. 325-337, 1988.

[16] J. S. B. Mitchell and S. Suri. "Separation and approximation of polyhedral surfaces," *Proc. 3rd ACM-SIAM Sympos. Discrete Algorithms*, pp. 296-306, 1992.

[17] M. S. Paterson and F. F. Yao. "Optimal binary space partitions for orthogonal objects," *J. Algorithms*, 13, pp. 99-113, 1992.

[18] F. P. Preparata and M. I. Shamos. *Computational Geometry.* Springer-Verlag, New York, 1985.

[19] W. J. Schroeder, J. A. Zarge, and W. E. Lorensen, "Decimation of triangle meshes," *SIGGRAPH*, 1992, pp. 65–70.

[20] C. B. Silio. "An efficient simplex coverability algorithm in $E^2$ with application to stochastic sequential machines," *IEEE Transactions on Computers*, C-28, pp. 109-120, 1979.

[21] C. A. Wang and E. Chan. "Finding the minimum visible vertex distance between two nonintersecting simple polygons," *Proc. of 2nd Symposium on Computational Geometry*, pp. 34-42, 1986.