Approximation Algorithms for Geometric Separation Problems

Joseph S. B. Mitchell*

July 13, 1993
(revised)

Abstract

We give polynomial-time approximation algorithms for some geometric separation 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.
- 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's vertices are among those of P and Q and R has a minimum number of facets.

Exact solution of the above problems in polynomial time is unlikely, since the decision versions are known to be NP-hard. We provide polynomial-time algorithms that are guaranteed to produce an answer within a logarithmic factor of optimal. The approximation factor is constant in the rectilinear/orthohedral cases (e.g., coverage by disjoint aligned rectangles, red-blue separation with a rectilinear polygon, terrain separation with an orthohedral terrain).

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

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

Thus, we are motivated to study the *polyhedral approximation* problem: Given a complex polyhedron, and a tolerance value ϵ, compute a simpler polyhedron (having smaller "combinatorial complexity") that lies within "distance" ϵ 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" ("offsetting") a surface Σ by an amount ϵ, one obtains a pair of new surfaces P and Q that "sandwich" Σ between them. A minimum-facet polyhedral separator of P and Q is a surface of least combinatorial complexity that approximates Σ within a tolerance of ϵ. By computing a family of approximate surfaces, corresponding to various values of ϵ, one can construct a hierarchical representation of Σ, allowing the user the option to use a sparse representation when the exact shape of Σ 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. [2] 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 ϵ, 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. [13] 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 [5, 6]. Thus, Mitchell and Suri [17] 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 [4] has provided an algorithm that gets within a factor $O(\log k^*)$ of optimal, where k^* is the complexity of an optimal separator. Most recently, Brönnimann and Goodrich [3] have improved the results to obtain a *constant* factor approximation algorithm for the separation problem for nested convex polytopes in three dimensions.

There have been several papers on the surface approximation problem in the graphics community (e.g., see [20]); however, these algorithms do not provide any guarantee of how close to minimum the resulting surface is, and some of them also do not guarantee how close the approximating surface is to the original surface.

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 (under the constraint that we use vertices of the given input), but within a constant factor for the orthohedral case.

Another question posed by [17] is that of separating "red" points from "blue" points in the plane; this problem is intimately related to the terrain approximation problem. Recently, Fekete[12] 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 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's vertices are among those of P and Q and R has a minimum number of facets. As a special case, we obtain an approximation algorithm for finding a minimum-facet surface that approximates a given terrain surface to within (vertical) error distance ϵ, subject to the approximating surface using vertices of the given surface. Thus, we can approximate a minimum complexity piecewise-linear function that satisfies a given set of data points, each with a given (vertical) error tolerance.

Our approximation factors for the above problems are logarithmic. In the case of *rectilinear* instances of our problems (e.g., disjoint set cover by rectangles or orthohedral terrain separation), we obtain 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. In the time since the original writing of this report, [1] have independently reported results similar to those reported here, using similar methods.

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 collinear 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 Δ that cover the point set S, with each Δ contained fully within some triangle of $\mathcal{T}$. We call a triangle Δ 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 [12] 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 [14] and Lovász [15] 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).

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 $O(k \log k)$ "canonical subtrapezoids" τ, where each τ 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 τ 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 $2k$ 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 subtriangles into a set of $O(k)$ *canonical subtrapezoids* τ having the following properties:

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

(2) the top/bottom sides of τ are segments lying on a line through a pair of points of S (assuming that τ is not a degenerate trapezoid consisting of a singleton point of S).

Note that there are $O(m^6)$ possible canonical subtrapezoids determined by S. We can trivially check each one in time $O(n)$ to determine which of the possible candidates are actually a subtrapezoid of some one of the n members of $\mathcal{T}$; this gives an overall time of $O(nm^6)$ to identify all canonical subtrapezoids.

Let $S' \subseteq S$ denote those $O(k)$ points of S involved in defining the boundaries of the subtrapezoids τ. Consider the segment tree based on the x-coordinates of the points S' (see [19]). This segment tree induces a partitioning of each interval $[x_i, x_j)$ $(i < j)$ into $O(\log k)$ subintervals, *canonical x-intervals*, according to the "allocation nodes" that result from insert $[x_i, x_j)$ into the tree. (Note that there are at most $2k$ canonical x-intervals.) The segment tree therefore induces a partitioning of a canonical subtrapezoid τ into $O(\log k)$ (vertical-walled) canonical subtrapezoids whose x-projections are canonical x-intervals. Let $\mathcal{T}''$ denote the resulting set of $O(k \log k)$ canonical subtrapezoids.

The set $\mathcal{T}''$ has the *guillotine property* (or, "binary space partition" (BSP) property), which is defined recursively: if $\mathcal{T}''$ has more than one trapezoid, then there exists a partitioning line ℓ (either vertical or determined by a pair of points of S), not passing through the interior of any

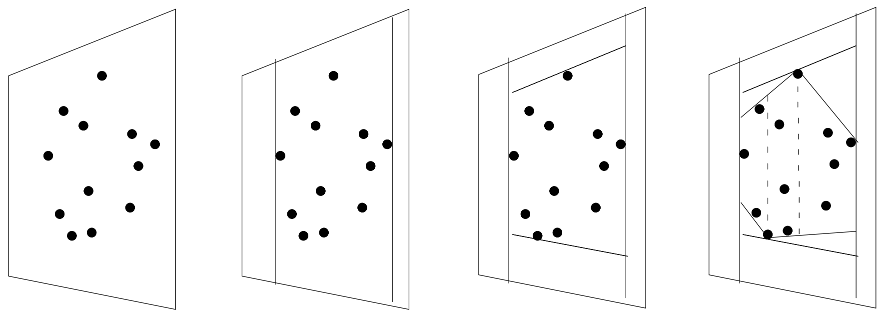

Figure 1: Obtaining canonical subtrapezoids from arbitrary trapezoids.

trapezoid, such that the sets of trapezoids on either side of ℓ also have the guillotine property. 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 $O(k \log k)$ pairwise-disjoint canonical subtrapezoids that cover S and have the guillotine property.*

This lemma tells us that if we can solve exactly the problem of finding a smallest disjoint cover of S by canonical subtrapezoids having the guillotine property, then we will have a method to approximate an optimal disjoint cover to within a factor $O(\log k^*)$, where k^* is the cardinality of an optimal disjoint cover.

Let $V([i,j), e_t, e_b)$ denote the minimum number of pairwise-disjoint canonical subtrapezoids in a guillotine cover of the points of S that lie within the trapezoidal region, $T(i,j,e_t,e_b)$, whose x-projection is the interval $[x_i, x_j)$, whose top boundary is the segment e_b (determined by some pair of points of S) and whose bottom boundary is the segment e_t (determined 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 $V([i,j), e_t, e_b)$:

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

$V([i,j), e_t, e_b) = 1$, 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,

$$V([i,j), e_t, e_b) = \min\{ \quad \min_{e \subset T(i,j,e_t,e_b)} \{V([i,j), e_t, e) + V([i,j), e, e_b)\}, \\ \min_{i<k<j} V([i,k), e_t, e_b) + V([k,j), e_t, e_b)\}$$

where the minimization on e 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 guillotine solution over the region $T(i,j,e_t,e_b)$, either there exists a canonical subtrapezoid τ of full width (spanning $[x_i, x_j)$), or there does not. If there does exist a spanning τ, then a top/bottom edge e of τ 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 τ, then the second term gives us the optimal covering number corresponding to splitting the interval $[x_i, x_j)$ into two subintervals at the coordinate x_k, optimized over choice of k.

The tabulation of the values of $V([i,j), e_t, e_b)$ is straightforward, starting with the smallest intervals $[i,j)$ and working upwards. The evaluation clearly takes polynomial time; a straightforward counting yields the bound $O(m^8)$. Since an optimal solution of size k^* can be converted to a canonical guillotine solution of size $O(k^* \log k^*)$, and our recursion yields an optimal over all canonical guillotine solutions, we obtain:

Theorem 2.2 *In polynomial time ($O(m^8 + m^6 n)$), one can compute a set of $O(k^* \log k^*)$ pairwise-disjoint subtriangles that cover S, where k^* is the number of subtriangles in an optimal disjoint cover.*

Remarks:

- The result generalizes immediately to the case in which S is a set of line segments or a set of polygons instead of a (finite) set of points.
- We can improve the time bound slightly, to $O(m^7)$, but with a slightly worse approximation factor of $O(\log m)$ (versus $O(\log k^*)$), as follows: In the transformation of $\mathcal{T}''$ to a guillotine cover, we partition each subtrapezoid according to the segment tree canonical x-intervals defined on the x-coordinates of S (versus S'). Since there are only $O(m)$ possible such canonical intervals, we will need to consider only $O(m^5)$ (versus $O(m^6)$) possible canonical subtrapezoids.
- 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 n 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 (BSP) for the resulting set of subrectangles. By [18], this results in $O(k)$ new subrectangles, which can again be shrunk (if necessary) to have all four edges passing through points of S. The new set of subrectangles has the BSP property, that 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 the BSP property. Then, we can obtain a recursion for $V(i,j,i',j')$:

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

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

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

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

We can clearly tabulate all values of $V(i,j,i',j')$ in polynomial time ($O(m^5)$); we tabulate in the order of increasing values of $j-i$ and $j'-i'$.

Theorem 2.3 *In polynomial time ($O(m^5 + m^4 n)$), one can compute a set of pairwise-disjoint subrectangles that cover S, with the number of subrectangles 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. In $O(m^2 n)$ preprocessing time, we can easily tabulate for each of the $O(m^2)$ choices of (I_x, I_y) whether or not there is a single rectangle (from among the n given rectangles, $\mathcal{R}$) covering the points of S within $I_x \times I_y$. Then, the recursion can be solved in total time $O(m^2)$.

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

3 Red-Blue Separation

Let R and B denote finite sets of "red" and "blue" points that lie within the unit square, $U \subset \Re^2$. Let n denote the cardinality of $R \cup B$. In the *Red-Blue Separation* problem, our goal is to compute a simple polygon P that separates R from B, such that P has the fewest possible vertices. This red-blue separation problem is known to be NP-complete (Fekete [12]). We give a polynomial-time approximation algorithm:

Theorem 3.1 *The Red-Blue Separation problem has an $O(\log k^*)$-approximation algorithm that runs in polynomial ($O(n^8)$) time, where k^* is the size of an optimal separator. If we restrict ourselves to rectilinear separating polygons, then there exists a polynomial-time ($O(n^5)$) $O(1)$-approximation algorithm for the Red-Blue Separation problem.*

Proof. Consider an 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, as in the previous section, P^* can be decomposed into $O(k^*)$ trapezoids, each of which can be shrunk to $O(1)$ trapezoids whose edges are defined by $O(k^*)$ points, $R' \subset R$; finally, each of these trapezoids can be partitioned into $O(\log k^*)$ canonical trapezoids, according to the canonical intervals induced by the x-coordinates of R'. We are left with a set $\mathcal{T}^*$ of $O(k^* \log k^*)$ canonical "red-only" trapezoids (i.e., each containing only red points on the interior) that satisfy the guillotine property. Let $\mathcal{T}$ denote the set of *all* $O(n^6)$ canonical red-only trapezoids.

Now, we can simply apply the method we described for disjoint set cover, for the set of regions $\mathcal{T}$ and the set of points $S = R$. The time bound is simply $O(n^8)$, since it is trivial to check in $O(n)$ time whether or not a canonical trapezoid is red-only (and hence a subtrapezoid for the class $\mathcal{T}$). We obtain a cover having at most $O(k^* \log k^*)$ disjoint canonical trapezoids, each having only red points interior. (In the rectilinear case, we obtain $O(k^*)$ disjoint aligned rectangles, each having only red points interior.) These trapezoids (resp., rectangles) can be linked together using $O(k^* \log k^*)$ (resp., $O(k^*)$) line segments (e.g., by constructing a spanning tree), yielding a connected polygonal set containing all red points, whose boundary complexity is $O(k^* \log k^*)$ (resp., $O(k^*)$). ∎

Remarks

- As in the disjoint set cover result of the previous section, we can improve the time bound slightly, to $O(n^7)$, but with a slightly worse approximation factor of $O(\log n)$ (versus $O(\log k^*)$), by decomposing canonical trapezoids according to the segment tree canonical x-intervals defined on the x-coordinates of R (versus R').

- Although the disjoint cover results carry over to higher dimensions, the red-blue separation results do not generalize. The problem is that in three dimensions one needs, in general, $O(k^2)$ tetrahedra in order to decompose a k-faceted polyhedron.

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. This problem arises in the work of Erwig [11], who applies such subdivisions to efficient data structures for storing shortest path information in networks embedded in the plane. Since even the 2-color version is known to be *NP*-complete [12], so is this multi-colored version of the problem. We give an $O(\log k^*)$-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 $O(k^* \log k^*)$ vertices, where k^* is the size of a minimum-vertex separating subdivision. If we restrict ourselves to rectilinear separating subdivisions, then there exists a polynomial-time $O(1)$-approximation algorithm for the problem.*

Proof. 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 and then partitioned into $O(\log k^*)$ canonical trapezoids such that each has only points of one color class inside it, and the set has the guillotine property.

As before, we can apply the disjoint set cover approximation algorithm with respect to the set $\mathcal{T}$ of all possible canonical trapezoids and the set S of all points. We obtain a cover having at most $O(k^* \log k^*)$ disjoint single-color trapezoids (or, in the rectilinear case, $O(k^*)$ disjoint single-color rectangles). These trapezoids (rectangles) can be linked together using disjoint line segments, resulting in a color-separating subdivision of the claimed complexity. ∎

As in the case of disjoint set cover, we can apply a simple-minded approach to yield a faster approximation algorithm for the rectilinear case, at the expense of a higher approximation factor:

Theorem 3.3 *Given n points in the plane, each colored with one of k colors, one can find in $O(n^2)$ time a rectilinear subdivision $\mathcal{S}$ that separates the points by color classes, such that $\mathcal{S}$ has at most $O(\log^2 n)$ times as many vertices as does a minimum-vertex separating rectilinear subdivision.*

4 Terrain Approximation

We consider now the terrain surface approximation problem, which can be modelled as a terrain separation problem.

Theorem 4.1 *Let P and Q be polyhedral terrains over a common support set $U \subset \Re^2$ (e.g., the unit square), with P lying above Q. Let V denote the set of all vertices of P and Q, and let $n = |V|$. Then, there exists a polynomial-time ($O(n^8)$) approximation algorithm to compute a polyhedral terrain R, with vertex set $V_R \subseteq V$, that lies between P and Q such that $|V_R| = O(k^* \log k^*)$, where k^* is the minimum number of vertices in a separating terrain surface on vertex set V. If we restrict our attention to orthohedral separating terrains, then there exists a polynomial-time $O(1)$-approximation algorithm for the problem.*

Proof. We say that three vertices, $u, v, w \in V$, determine a *feasible triangle* if the triangle with vertices u, v, and w lies between surface P and Q (i.e., on or below P and on or above Q).

Let R^* be an optimal (minimum-vertex) separating terrain surface, having k^* vertices. The projection of R^* onto U gives a polygonal subdivision of U. If we triangulate the faces of this subdivision, we are left with a partitioning of U into $O(k^*)$ triangles, each determined by a triple of vertices from V that corresponds to a feasible triangle. Let $V' \subseteq V$ be those $O(k^*)$ vertices that determine triangles. By splitting each triangle in two with a vertical segment through its middle vertex, we obtain $O(k^*)$ vertical-walled triangles (degenerate trapezoids), each of which can be partitioned into $O(\log k^*)$ canonical trapezoids, according to the canonical x-intervals induced by the x-coordinates of V'. We are left with a guillotine partitioning of U into $O(k^* \log k^*)$ canonical trapezoids.

Let $V([i,j), e_t, e_b)$ denote the minimum number of canonical trapezoids in a guillotine partitioning of the trapezoidal region, $T(i,j,e_t,e_b) \subset \Re^2$ whose x-projection is the interval $[x_i, x_j)$, whose top segment is e_t and whose bottom segment is e_b, with both e_t and e_b determined by pairs of vertices of P and Q. Our goal is to partition U into a minimum number of canonical trapezoids (subject to the guillotine property). We do this by solving the recursion:

$V([i,j), e_t, e_b) = 1$, if the region $T(i,j,e_t,e_b)$ corresponds to a canonical trapezoid;

otherwise,

$$V([i,j),e_t,e_b) = \min\{ \quad \min_{e \subset T(i,j,e_t,e_b)} \{V([i,j),e_t,e) + V([i,j),e,e_b)\}, \\ \min_{i<k<j} V([i,k),e_t,e_b) + V([k,j),e_t,e_b)\}$$

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

Since there are only $O(n^2)$ choices for e, and there are $O(n^6)$ subproblems corresponding to regions $T(i,j,e_t,e_b)$, the claim follows.

Consider now the case of orthohedral separating terrains. By definition, such terrains have facets that are normal to the coordinate axes; thus, their horizontal facets project to axis-aligned rectangles in $\Re^2$. (We allow P and Q to be arbitrary polyhedral terrains.) Let R^* be an optimal orthohedral separating terrain, having k^* vertices. Then, R^*'s horizontal facets project to a set of rectangles in $\Re^2$ that have a BSP partition of size $O(k^*)$. It is easy to perturb the edges of this partition so that all of its edges pass through points that are projections of V into $\Re^2$. Thus, exactly as in Section 2, we can obtain a simple recursion that results in a constant factor approximation algorithm, with running time $O(n^5)$. ∎

Once again, we can apply a simple-minded approach to yield a faster approximation algorithm for the orthohedral case:

Theorem 4.2 *In $O(n^2)$ time, one can compute an orthohedral terrain surface R that separates two polyhedral terrains, P and Q (having n vertices), such that R has $O(k^* \log^2 n)$ vertices, where k^* is the minimum number of vertices in an orthohedral terrain separator.*

5 Conclusion

The main outstanding open theoretical problem is to give a polynomial-time approximation algorithm for the general nonconvex surface approximation problem.

Acknowledgements

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

References

[1] P. K. Agarwal and S. Suri. Surface approximation and geometric partitions. In *Proc. Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 34–43, 1994.

[2] 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.

[3] H. Brönnimann and M. T. Goodrich. Almost optimal set covers in finite VC-dimension. In *Proc. 10th Annu. ACM Sympos. Comput. Geom.*, pages 293–302, 1994.

[4] K. L. Clarkson. Algorithms for polytope covering and approximation. In *Proc. 3rd Workshop Algorithms Data Struct.*, volume 709 of *Lecture Notes in Computer Science*, pages 246–252, 1993.

[5] G. Das and M. T. Goodrich. On the complexity of approximating and illuminating three-dimensional convex polyhedra. In *Proc. 4th Workshop Algorithms Data Struct.*, Lecture Notes in Computer Science, page To appear. Springer-Verlag, 1995.

[6] G. Das and D. Joseph. Minimum vertex hulls for polyhedral domains. *Theoret. Comput. Sci.*, 103:107–135, 1992.

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

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

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

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

[11] M. Erwig. "Encoding shortest paths in spatial networks," Report 116, FernUniversität Hagen, 1991 (revised 1993, 1994).

[12] S. Fekete. On the complexity of min-link red-blue separation. Manuscript, 1992.

[13] L. J. Guibas, J. E. Hershberger, J. S. B. Mitchell, and J. S. Snoeyink. Approximating polygons and subdivisions with minimum link paths. *Internat. J. Comput. Geom. Appl.*, 3(4):383–415, December 1993.

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

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

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

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

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

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

[20] W. J. Schroeder, J. A. Zarge, and W. E. Lorensen. Decimation of triangle meshes. In *SIGGRAPH '92*, volume 26, pages 65–70, July 1992.

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