Point Probe Decision Trees for Geometric Concept Classes *

Esther M. Arkin(1) † Michael T. Goodrich(2) ‡ Joseph S. B. Mitchell(1) §
David Mount(3) ¶ Christine Piatko(4) Steven S. Skiena(5) ‖

(1)Department of Applied Mathematics and Statistics
State University of New York, Stony Brook, NY 11794-3600
(2)Department of Computer Science
Johns Hopkins University, Baltimore, MD 21218
(3)Department of Computer Science and Institute for Advanced Computer Studies
University of Maryland, College Park, MD 20742
(4)NIST, Gaithersburg, Maryland 20899
University of Maryland, College Park, MD 20742
(5)Department of Computer Science
State University of New York, Stony Brook, NY 11794-4400

Abstract

A fundamental problem in model-based computer vision is that of identifying to which of a given set of *concept classes* of geometric models an observed model belongs. Considering a "probe" to be an oracle that tells whether or not the observed model is present at a given point in an image, we study the problem of computing efficient strategies ("decision trees") for probing an image, with the goal to minimize the number of probes necessary (in the worst case) to determine in which class the observed model belongs. We prove a hardness result and give strategies that obtain decision trees whose height is within a log factor of optimal. Our results generalize recent work on probing strategies which identify which single model is present, rather than which *class* of model is present. In most real pattern matching applications (e.g., character recognition, with multiple font types), it is more important to identify the concept class of an observed model, rather than its individual identity, which may require considerably more effort to determine.

*These results grew out of discussions that began in a series of workshops on Geometric Probing in Computer Vision, sponsored by the Center for Night Vision and Electro-Optics, Fort Belvoir, Virginia, and monitored by the U.S. Army Research Office. The views, opinions, and/or findings contained in this report are those of the authors and should not be construed as an official Department of the Army position, policy, or decision, unless so designated by other documentation.

†Partially supported by NSF Grants ECSE-8857642 and CCR-9204585.

‡Partially supported by the NSF and DARPA under Grant CCR-8908092, by the NSF under Grants CCR-9003299, and IRI-9116843.

§Partially supported by grants from Hughes Research Laboratories, Boeing Computer Services, Air Force Office of Scientific Research contract AFOSR-91-0328, and by NSF Grants ECSE-8857642 and CCR-9204585.

¶Partially supported by NSF Grant CCR-89-08901 and by grant JSA 91-5 from the Bureau of the Census.

‖Partially supported by NSF grant CCR-9109289 and New York Science and Technology Foundation grants RDG-90171 and RDG-90172.

1 Introduction

In computer vision, one is interested in devising algorithms that automatically interprets the contents of a digital image (a "scene"). In *model-based* computer vision, one is also given some information a priori about the objects to be searched for in the scene, specifically, a *library* S of k *models* (or "shapes"), partitioned into c *concept classes* (or "color classes"). The problem is to determine the concept class of each model that appears in the image.

For example, suppose we are given a library S consisting of images of different types of vehicles (say bicycles, buses, cars, and trucks), and we must determining which type of vehicle is present in a given query image q. The most direct approach would be to compare each model in S against q and determine exactly which one is present in q. Although there are only $c = 4$ concept classes in this example, the total number of models k may be much larger than c – e.g., the "car" class may contain an image of every make and model of car on the road. Determining the specific car model is irrelevant for this application, and so a more efficient search strategy would exploit the properties shared by members of a particular class. For example, bicycles are smaller than cars, while buses/trucks are larger.

In another example, the library S may consist a set of alphabetic characters in many different fonts. The concept classes then may be defined to be sets of all characters that represent the same letter, regardless of font or style of print. We are more interested in determining whether the letter present is an "E" or an "F" than in what its precise font is. See Figure 1.

In this paper, we examine a fundamental instance of this model-based computer vision problem. Each model in the library S is given in a fixed position, orientation, and scale, and the given input image contains exactly one instance of one model. This situation arises, for example, after an image segmentation is applied to a scanned-in page of text. Our problem is to determine the class of the object that is present in the scene by asking a sequence of "probe queries" of the following form: "Is there an object at location p in the scene?" We assume that there is an oracle that answers these probe queries, and we measure complexity in terms of the worst-case number of queries to the oracle before identifying the concept class of the model that is actually present in the image. In practice, such an oracle may be implemented as a local operator on a digitized image — e.g., as a measure of local texture or of gradient field.

A probing strategy is an interactive algorithm that can most naturally be thought of as a binary *decision tree*, in which each node, v, corresponds to a set of candidate models, $S(v)$, which in turn belong to a set of candidate concept classes. The root corresponds to the full set S, and each leaf corresponds to a set of models that all belong to the same concept class. Each internal node has an associated probe point that specifies the query that we ask the oracle at that particular stage of the identification. A path from the root to a leaf in the decision tree represents a possible outcome for a particular scene. An example is illustrated in Figure 2.

In this paper, we study the complexity of constructing minimum height decision trees for geometric concept classes. In other words, given a set S of geometric models, we want to construct (off line) a decision tree so that the worst-case number of probe queries needed to identify the concept class of the model present in the image is as small as possible. Any such decision tree has height at least $\lceil \lg c \rceil$. However, $\Omega(k)$ height decision trees are necessary for certain arrangements and colorings of models, even for two color classes.

Main Results

We formulate and study the geometric decision tree problem from the point of view of concept classes, proving several related results:

- Let $S = S_1 \cup S_2$ be a set of k non-degenerate aligned unit squares of two color classes. Then the problem of constructing a minimum height decision tree to determine the color class is NP-complete.
- Let $S = \cup_{i=1}^{c} S_i$ be a set of k simple polygons in the plane (having a total of n vertices), of classes $1, 2, \ldots, c$, whose arrangement is non-degenerate. Then, we can find, in polynomial time, a decision tree whose height is at most $2 \lg k$ times the height of an optimal decision tree that identifies the class to which the model in the scene belongs. This construction can be done in $O(n \log n + hc|A(S)|)$ time.
- Let $S = \cup_{i=1}^{c} S_i$ be a set of k simple polygons in the plane, of classes $1, 2, \ldots, c$, whose arrangement is possibly degenerate. Then, we can find, in polynomial time, a decision tree whose height is at most $4 \lg k$ times the height of an optimal decision tree that identifies the class to which the model in the scene belongs. The method uses a "double greedy" strategy for selecting pairs of probes to use in succession.

Motivation

We are motivated by real instances of the model-based computer vision problem. The specific instance of our problem (with the assumption that the models be given in a fixed position and orientation) arises in recent approaches to model-based vision suggested by Arkin and Mitchell [1], Bienenstock et al. [4, 5] (for character recognition), Mirelli [19], and Papadimitriou [21]. For example, Papadimitriou suggests a "probing scheme" in which one searches for instances of geometric models anywhere within an image, using our same model of probing, and he reduces the problem to exactly the problem studied here. In effect, the probing schemes of [1, 4, 5, 21] serve to "factor out" the effect of translation and rotation, reducing the final decision problem to that of this paper. The effect of translation and rotation can also be accommodated within our framework by replicating the model instances according to all possible positions within the image, assuming, of course, that there are only a finite number of possible placements.

Relation with previous work

Most previous work on building decision trees [20] has focused on non-geometric instances of the problem. The *abstract* decision tree problem takes as input a finite *universal* set $X = \{1, \ldots, k\}$ and a family of subsets of X, $\mathcal{T} = \{T_1, T_2, \ldots, T_m\}$, representing the set of possible probes. Hyafil and Rivest [14] prove that it is NP-complete to construct a minimum height or a minimum external path length decision tree. Garey [9] presents dynamic programming algorithms for determining an optimal weighted decision tree. Our problem considered here can be viewed as the unweighted decision tree problem in which the set of possible "tests" is defined by the faces in the *arrangement* $A(S)$ determined by a set S of k geometric objects.

Arkin, et. al [2] recently studied the problem of building geometric decision trees, with the goal of identifying the specific model present in the scene (i.e., without considering the notion of color classes). They showed that, although optimal decision trees can be efficiently constructed for a *non-degenerate* set of k polygons that each contain the origin, the problem of constructing a minimum height decision tree is NP-complete if the models are possibly degenerate or if they do not share a common point. (Here, we say that a set of polygons is non-degenerate if no two polygons share a subsegment of their boundaries.)

More importantly, [2] define a "greedy" heuristic for constructing decision trees, and prove that it yields a decision tree of height at most $\lceil \lg k \rceil$ times that of an optimal decision tree. They also show that there are geometric instances of the problem for which the greedy heuristic attains the worst-case factor $\Theta(\lg k)$. The greedy decision trees of [2] can be arbitrarily bad, however, when the models are assigned color classes, and one is only interested in identifying the class to which a particular scene belongs.

Recent related work on geometric object-identification has been motivated by applications in model-based computer vision [7, 13] and tactile sensing in robotics [10, 12]. Some results in this area include characterizations by Joseph and Skiena [15] that $n+3$ finger probes are sufficient and $n-1$ necessary to determine a convex n-gon P selected from a finite set ?, improving an earlier result by Bernstein [3], and a result of Lyons and Rappaport [17] that takes S to be a collection of k convex polygons with fixed orientation on a plane, and shows that $k-1$ finger probes are necessary and sufficient to determine the model. An interesting feature of this latter result is that it is independent of the number of sides in the models; this property is lost when the models may assume arbitrary orientations in the scene. See [22] for a survey of related results in geometric probing.

Definitions and Notation

Let S be a set of k simple polygons ("*models*") in the plane, having a total of n vertices. We assume that S is partitioned into c color classes, S_i, and we write $S = \langle S_1, S_2, \ldots, S_c \rangle$. Thus $S_i \cap S_j = \emptyset$ for all $i \neq j$, and $S = \cup_{i=1}^{c} S_i$.

A point in the plane designates a *probe* (or "test"). Each probe P can be identified with a subset of the set of objects that are "Hit" by the probe, $h_P(S) \subseteq S$, and the remaining objects $m_P(S) = S \backslash h_P(S)$ are said to be "Missed" by the probe. If T is any subset of S then we extend the hit and missed subsets in the natural way so that $h_P(T) = h_P(S) \cap T$, and $m_P(T) = m_P(S) \cap T$. We say that a set of probes is *complete* for S, if for any two objects in different color classes there exists a probe that hits one of these objects and misses the other.

For a set $S = \langle S_1, S_2, \ldots, S_c \rangle$ of models and a complete set of probes, we desire a binary decision tree that discriminates between models of different classes without necessarily distinguishing models within the same class. Each node of the tree is associated with a subset of S and each nonleaf node is associated with a probe. A decision tree satisfies the following conditions:

- The subset associated with the root of the tree is S.
- Each leaf of the tree is associated with a (not necessarily proper) subset of a single color class.
- If T is the subset associated with some nonleaf node and P is the probe associated with this node, then the left child of this node is associated with $h_P(T)$, and the right child is associated with the missed set $m_P(T)$.

We let $A(S)$ denote the *arrangement* induced by S; $A(S)$ is a collection of 0-faces (vertices), 1-faces (edges), and 2-faces (cells). All points within an edge or a cell of $A(S)$ intersect the same set of polygons of S, and therefore each point in an edge or a cell has the same discriminating power when it is used as a probe point. Thus, the set of possible probes can be identified with the faces (vertices, edges, and cells) of $A(S)$.

We say that $A(S)$ (or S) is *degenerate* if two distinct edges of polygons in S intersect in more than a single point (i.e., they intersect in a line segment).

2 The Non-Degenerate Case

One obvious lower bound on the height of a decision tree is $\lg c$, since there are at least c leaves. We now show another lower bound on the height of a decision tree, which applies whether the arrangement is degenerate or not. Let $s(\mathcal{C})$ be the minimum number of points needed to "stab" all models in the set $\mathcal{C}$, where a point *stabs* a model if it is in that model. Define $\bar{S}_i = \cup_{j \neq i} S_j$.

Lemma 2.1 *Let $S = \langle S_1, S_2, \ldots, S_c \rangle$ be a partitioning of a set of k simple polygons in the plane into c color classes. Assume that $A(S)$ is non-degenerate. Then, the height of any decision tree identifying the class to which the model in the scene belongs is at least* $\min_i\{s(\bar{S}_i)\}$.

Proof. We give an adversary argument. Consider the path in the probe tree from the root to a leaf, in which each probe is a "Miss". If the points along this path in which we probed do not stab all the models in two or more color classes, than we can not tell those classes apart at that leaf. □

With this lower bound, we can easily adapt a proof from Arkin *et al.* [2] that constructing an optimal decision tree is NP-complete, even for two color classes:

Theorem 2.2 *Let $S = \langle S_1, S_2 \rangle$ be a set of k non-degenerate aligned unit squares of two color classes. Then the problem of constructing a minimum height decision tree for S is NP-complete.*

Proof. The problem is clearly in NP, since a complete set of candidate probe points can be concisely expressed by taking midpoints of segments joining vertices of polygons of S.

To prove the NP-hardness of determining the existence of a decision tree of height $\leq h$, we show a reduction from the problem of deciding whether h points suffice to stab a given set of aligned unit squares, where a set of points *stabs* a given set of squares if each square contains at least one of the points. This problem was (essentially) shown to be hard by Fowler, Paterson and Tanimoto [8]. (They show that 3-SAT can be reduced to the problem of deciding whether h aligned unit squares suffice to cover a given set of points. Almost the same reduction can be made from 3-SAT to the problem of stabbing a given set of aligned unit squares by h points.)

Given an instance of the square stabbing problem (SSP), we create two disjoint copies of it, one of each color class. We claim that a decision tree of height less than h exists if and only if the squares of the (SSP) instance can be stabbed by h points. Given h points that stab all the squares of an SSP instance, we use them as probe points for the objects in S_1. If one or more probes is a "Hit" then our model is of class S_1, otherwise it is of class S_2. The "only if" direction follows from our lower bound. □

An easy case in which an optimal height decision tree can be constructed exactly is that in which there are two classes, one of which consists of a single model. (Refer to Figure 3.)

Lemma 2.3 *Let $S = \langle S_1, S_2 \rangle$ be a set of k non-degenerate models, with $|S_1| = 1$. Then a minimum height decision tree for S is of height 1 or 2 and the tree can be constructed in $O(n \log n + |A(S)|)$ time.*

Proof. Clearly, it can be checked in time $O(|A(S)|)$ whether one probe point suffices. We show that at most *two* probes are needed to discriminate between the two classes: In the arrangement of all objects, consider a face f that is contained in the object of class S_1 and that has on its boundary some portion of the boundary of that object. (At least one such face exists, and can be found once the arrangement is built.) Probe in face f. If this is a "Miss", then we know the target object is of class S_2. If it is a "Hit", then probe in a face f' that is adjacent to f along some portion of the boundary of the S_1 object. Because of non-degeneracy, the set of models hit is identical to those hit by the previous probe, except for the S_1 object. If this second probe is a "Hit", then we conclude that an S_2 object is present; otherwise, we conclude S_1. This gives us an optimal decision tree. This tree can easily be constructed by a depth-first traversal of the arrangement, assuming it is given as a polygon arrangement (e.g., see Goodrich [11]), which can be constructed in $O(n \log n + |A(S)|)$ time by a method of Chazelle and Edelsbrunner [6]. $\square$

Remark. A generalization of the strategy given in the proof above shows that one can always achieve a decision tree of height $2|S_1|$ for the case of two color classes.

We use Lemma 2.3 to find a decision tree for the general non-degenerate problem, and show that its height is at most $2 \lg k$ times that of the optimal tree.

Theorem 2.4 *Let $S = \langle S_1, S_2, \ldots, S_c \rangle$ be a partitioning of a set of k simple polygons in the plane into c color classes. Assume that $A(S)$ is non-degenerate. Then, we can find, in $O(n \log n + hc|A(S)|)$ time, a decision tree whose height is at most $2 \lg k$ times the height of an optimal decision tree identifying the class to which the model in the scene belongs.*

Proof. Our strategy is to approximate a minimum height decision tree that identifies which specific model is present (if any) among the models in $\bar{S}_i = \cup_{j \neq i} S_j$, where i is the class achieving the $\min_i\{s(\bar{S}_i)\}$. Probe along this tree, until reaching a leaf which corresponds to identifying a specific model. Since we identified the exact model we know which class it is in. (Clearly, this class is not S_i.) We have thus reduced our problem to that of deciding between one model, which is in a specific (and therefore known) color class other than S_i, and any of the models of color class S_i. But this problem can be done with only 2 more probes, by Lemma 2.3.

Let h be the height of the decision tree obtained by this method, and let h^* be the height of the minimum probe tree. We know by Lemma 2.1 that

$$h^* \geq \min_i\{s(\bar{S}_i)\}.$$

Also, by Theorem 5 in [2] we can build a decision tree that identifies which model is present among a set of k' non-degenerate models stabbed by $s \geq 2$ given points, and the tree is of height at most $s - 1 + \lceil \lg \lfloor k'/(s-1) \rfloor \rceil$. To this tree we add at most two more probes, as in Lemma 2.3. In our case we use $k' = |\cup_{j \neq i} S_j|$. ($k' < k$.) However, we can not find the exact stabbing number of $\bar{S}_i$, since that problem is NP-hard, so instead we approximate it to within a $\lg k'$ factor (e.g., by Lovasz [16]), allowing us to use $s \leq \lg k' \min_i\{s(\bar{S}_i)\}$. We have,

$$h \leq \lg k' \min_i\{s(\bar{S}_i)\} + \lceil \lg \lfloor k'/(s-1) \rfloor \rceil + 1,$$

implying that

$$\frac{h}{h^*} \leq \lg k' + \frac{\lceil \lg \lfloor k'/(s-1) \rfloor \rceil + 1}{\min_i\{s(\bar{S}_i)\}}.$$

Now, if $\min_i\{s(\bar{S}_i\} \in \{1, 2\}$, then we can easily obtain a tree of height at most $\lg k' + 2$ — a $\lg k'$ height tree can identify exactly which model in $\bar{S}_i$ is present (if at all), and at most 2 additional probes complete the classification. If we assume that $\min_i\{s(\bar{S}_i\} \geq 3$, then we get

$$\frac{h}{h^*} \leq \lg k' + \frac{\lceil \lg \lfloor k'/(s-1) \rfloor \rceil + 1}{\min_i\{s(\bar{S}_i)\}} \leq 2 \lg k.$$

The bottleneck in the algorithm is implementing a greedy strategy for finding i. This can easily be done in $O(n \log n + hc|A(S)|)$ time by a series of depth-first traversals of the arrangement $A(S)$. □

In the next section we show that we can still get an $O(\lg k)$ approximation factor in the degenerate case, although the constant factor in this bound is not quite as good as in the non-degenerate case.

3 The Degenerate Case

Our proof for designing a good decision tree for geometric concept classes depends crucially on the fact that no two models share a common edge, for this implies that the number of models covering a particular face in the arrangement of models differs by exactly 1 with the number of models covering an adjacent face. In this section we give a set of strategies for dealing with the "degenerate" case when several models can share common edges. Such a "degenerate" situation can arise, for example, in character recognition, where different letters (such as "R", "E", and "D") can share common edges. (See Figure 4.)

We begin with a simple result, showing that the degenerate case is hard, even if there are only two color classes and the models are convex and all share a common point:

Theorem 3.1 *Let $S = \langle S_1, S_2 \rangle$ be a set of k convex, possibly degenerate models of two color classes, with the property that all models (of both classes) have a point in common. Then the problem of constructing a minimum height decision tree for S is NP-complete.*

Proof. The problem is clearly in NP, since a complete set of candidate probe points can be concisely expressed by taking midpoints of segments joining vertices of polygons of S.

To prove the NP-hardness of determining the existence of a decision tree of height $\leq h$, we show a reduction from the set cover problem: Given a collection of subsets $C = \{C_1, C_2, \ldots C_m\}$ of $U = \{1, 2, \ldots, n\}$ and a number h, is there a subset $C' \subseteq C$ with $|C'| \leq h$, such that every element of U belongs to at least one member of C'?

Given an instance of the set cover problem, we create a regular $2m$−gon, M, with sides of unit length centered at the origin. Consider the sides of M to be indexed $j = 1, \ldots, 2m$. For each edge j, let $v_j \notin M$ be a point "just outside" edge j, and let Δ_j be the triangle determined by edge j and point v_j. Choose v_j such that the polygon $M \cup (\bigcup_j \Delta_j)$ is convex. The instance of our probe tree problem created will have n models of each color class, say red and blue, such that each element in U will have two corresponding models, one of each color. Each red model corresponds to an element of $i \in U$, and is a convex polygon R_i

$$R_i = M \cup \left[\bigcup_{\{j \ : \ i \in C_j\}} \Delta_j \right].$$

Similarly, each blue model corresponds to an element $i \in U$ and is a convex polygon B_i

$$B_i = M \cup \left[\bigcup_{\{j \ : \ i \in C_j\}} \Delta_{m+j} \right].$$

We claim that a decision tree of height less than h exists if and only if there is a set cover of size h. Given h subsets that form a set cover, we use as probe points points in the corresponding (red) triangles Δ_j for j in the set cover. If one or more probes is a "Hit" then our model is of class S_1, (red), otherwise it is of class S_2, (blue).

Note that a probe point in M gives no information, since all models in S are present at such a point, thus we assume that any probetree will contain no such probes. The "only if" direction follows from the observation above and our lower bound of Lemma 2.1. □

An interesting open problem is: Can we design, in polynomial time, an optimal probe tree for the case in which the models are of two classes, are non degenerate, (convex) and all share a point in common.

3.1 The Framework

For a decision tree to exist, the only assumption we must make is that for any pair of models X and Y, there is some probe point p that distinguishes X from Y (either because $p \in X$ and $p \notin Y$, or vice versa). Because of a lack of nice geometric structure, we model the degenerate case in a more abstract fashion than we have used above.

Henceforth, we will refer to nodes of a decision tree by their associated sets. Consider some node in a decision tree associated with the set $T = \langle T_1, T_2, \ldots, T_c \rangle$. We will use lower-case, $\langle t_1, t_2, \ldots, t_c \rangle$ denote the respective cardinalities of the color classes. Define the *weight* of T to be its cardinality, $wgt(T) = |T|$. For each color i, define the *i-weight* of T to be the sum of cardinalities except class i.

$$wgt_i(T) = \sum_{j \neq i} t_j = wgt(T) - t_i.$$

Observe that for any leaf in the decision tree, one of the weight functions is zero, namely the weight function of the only surviving color class.

One measure of the quality of a probe is how evenly it partitions the set of models among its children. Another way to think about this is, for each edge (T, T') in the decision tree, consider how many models in T have been eliminated from consideration in T'. We define the *total elimination* from T to T' to be

$$elim(T, T') = wgt(T) - wgt(T').$$

The *class most heavily eliminated* is the class i that maximizes $t_i - t'_i$. Finally, for any class i, define the *i-elimination* of the edge (T, T') to be the total elimination excluding class i,

$$elim_i(T, T') = wgt_i(T) - wgt_i(T') = elim(T, T') - (t_i - t'_i).$$

Of course, there must be a balance between the number of models eliminated from the left and right children of a node, since every model eliminated from one child will be present in the other child. In the classless case of [2] (or equivalently, where each object is in its own class) the greedy strategy is chosen to maximize the minimum elimination from each of the two children. In the presence of classes, this is not quite the right strategy, because it is quite acceptable to have one large color class.

3.2 The "Double-Greedy" Strategy

We now define the *double-greedy* heuristic for constructing decision trees. We select probes to be applied in ensembles of two consecutive probes, each of which is chosen by a greedy criterion (hence the term, double-greedy). Given a node T (with at least two color classes still active) the first probe of each ensemble is chosen exactly as in the standard greedy algorithm to maximize the minimum number of models eliminated along each of its two outgoing edges. That is, we select the probe P that maximizes

$$\min(elim(T, h_P(T)), elim(T, m_P(T))).$$

For the second probe, consider each of the two children, T' and T'' of T. For T', let i be the most heavily eliminated class by the previous probe. The second probe is chosen to maximize the minimum i-elimination for each of the two resulting grandchildren. That is, we select the probe P' that maximizes

$$\min(elim_i(T', h_{P'}(T')), elim_i(T', m_{P'}(T'))).$$

We do the same for T'' with whatever its most heavily eliminated class is.

The main result of this section is that the height of the double-greedy decision tree is at most a logarithmic factor greater than the height of the optimum decision tree.

Theorem 3.2 *Let $S = \langle S_1, S_2, \ldots, S_c \rangle$ be a set of k simple polygons in the plane, partitioned into c color classes, that are in a possibly degenerate arrangement. Let h^* denote the minimum height among all decision trees, and let h_g denote the height of the double-greedy decision tree. Then*

$$\frac{h_g}{h^*} \leq 4 \lg k.$$

This theorem is proved by a variation of the argument appearing in [2] for standard decision trees. The argument is more complicated in this case because of the extra complexity of the heuristic, and the subtleties of how probes eliminate models between different classes. Consider the longest path in the double-greedy decision tree. The edges on this path can be partitioned into consecutive pairs since the probes have been chosen in ensembles. Let d denote the color class that is nonempty in the leaf of this path. At the root, the d-weight is at most k, and at the leaf it is zero. Observe that the d-weights decrease monotonically (not strictly) along the path. Classify an edge on this path as being *light* if the d-weight of the child is at most one half of the parent, and otherwise it is *heavy*. Along any path there can be at most $\lg k$ light edges, and so for the remainder of the proof it suffices to consider only heavy edges.

For each node T at the head of an ensemble, let $m(T)$ denote the length of the longest subpath (of ensemble edge pairs) that reduces the d-weight by a factor of no more than one half. Let m denote the maximum value of this function over the entire path. Observe that the weight cannot be halved more than $\lg k$ times, and so $(m+2)\lg k$ is an upper bound on the height of the double-greedy decision tree. To complete the proof it suffices to show that for some constant C, $C \cdot m$ is a lower bound on the height of the optimum decision tree.

Consider the subpath that defines m. Consider any ensemble edge pair on this subpath. If d is the most heavily eliminated class from the first probe, then we *mark* the second edge of the pair, and otherwise we *mark* the first edge. The number of marked edges on the path is $m/2$.

Let W denote the d-weight at the start of the subpath. At the end of the path the d-weight is at least $W/2$, and so by the pigeonhole principal, along some marked edge (T, T') on the path the d-weight has decreased by at most

$$\frac{W - (W/2)}{m/2} = \frac{W}{m}.$$

Call the marked probe P that minimizes the decrease in d-weight the *limiting probe*. We claim that after applying this probe, no later probe along an edge can decrease the d-weight by a factor greater than twice this amount, that is, $2W/m$. Before showing this, let us see why this suffices to complete the proof. At the end of the subpath the current d-weight is at least $W/2$. Clearly the height of the optimum tree to complete the discrimination of this subset of models is a lower bound on the height of the optimum tree for the entire problem. However, since no future probe can decrease the d-weight by more than $2W/m$, at least

$$\frac{W/2}{2W/m} = \frac{m}{4}$$

probes are needed even in the optimum tree. Thus the optimum decision tree has height at least $m/4$, and this will complete the proof. (We illustrate the main ideas of this proof in Figure 5.) All that remains is to prove the following claim.

Lemma 3.3 *If the limiting probe P applied along edge (T, T') reduces the d-weight by E, then no later probe can reduce the d-weight by more than $2E$.*

Proof. We may assume that the probe P reduces the d-weight of T by less than half, since otherwise the lemma is trivially true. Thus, there are two cases to consider.

If the limiting probe is applied along the second edge of an ensemble, then this probe was selected to reduce the d-weight by the largest amount possible, that is, to distinguish as many elements as possible from among all classes except d. Since later nodes involve subsets of T, any probe that distinguishes more than E models from these classes could have been applied at T to decrease the d-weight even more. The greedy choice of P implies that no such probe can exist.

If, on the other hand, the limiting probe was applied to the first edge of the ensemble, then by the definition of the double-greedy strategy we know that class d was not the most heavily eliminated class in this probe. Thus, the number of models eliminated from class d by this probe can be no more than half the number of models eliminated from all the other classes, and hence

$$elim(T, T') \leq 2E.$$

Now, suppose towards a contradiction that some later probe reduces the d-weight by more than $2E$. Such a probe must distinguish at least $2E$ total models from a subset of T, and hence could have been applied at T to eliminate more than $2E$ models, contradicting the choice of this probe.
□

4 Conclusions

We have introduced a general framework for constructing provably good point-probe decision trees for geometric concept classes. Our methods all produce trees that are off from the optimal by at most a logarithmic factor. Moreover, all of our algorithms run in polynomial time.

There are a number of interesting questions that this work has uncovered. Consider, for example, the following open questions:

- Is $O(\lg k)$ the best approximation factor achievable in polynomial time?
- Can an $O(\lg k)$ approximation factor be achieved in time $O(n \log n + |A(S)|)$ in the general (even non-degenerate) case?
- How do the results of this paper extend to higher dimensions?

5 Acknowledgements

These results grew out of discussions that began in a series of workshops on Geometric Probing in Computer Vision, sponsored by the Center for Night Vision and Electro-Optics, Fort Belvoir, Virginia, and monitored by the U.S. Army Research Office. In particular, we thank Teresa Kipp and Vince Mirelli of the Center for Night Vision and Electro-Optics, and we thank all of the participants of the series of workshops. Moreover, we especially thank Christine Piatko and Kathleen Romanik for a number of helpful and clarifying discussions.

The views, opinions, and/or findings contained in this report are those of the authors and should not be construed as an official Department of the Army position, policy, or decision, unless so designated by other documentation.

References

[1] E. Arkin and J. Mitchell. Applications of combinatorics and computational geometry to pattern recognition. Technical report, Cornell University, 1990.

[2] E.M. Arkin, H. Meijer, J.S.B. Mitchell, D. Rappaport, and S.S. Skiena. Decision trees for geometric models. *Proc. Ninth ACM Symposium on Computational Geometry*, 1993.

[3] H. Bernstein. Determining the shape of a convex n-sided polygon by using $2n + k$ tactile probes. *Information Processing Letters*, 22:255–260, 1986.

[4] E. Bienenstock, D. Geman, and S. Geman. A relational approach in object recognition. Technical report, Brown University, 1988.

[5] E. Bienenstock, D. Geman, S. Geman, and D.E. McClure. Phase II technical report, development of laser radar ATR algorithms. Technical Report Contract No. DAAL02-89-C-0081, CECOM Center for Night Vision and Electro-Optics, October 1990.

[6] B. Chazelle and H. Edelsbrunner. An optimal algorithm for intersecting line segments in the plane. *J. ACM*, 39:1–54, 1992.

[7] R. Chin and C. Dyer. Model-based recognition in robot vision. *Computing Surveys*, pages 67–108, 1986.

[8] R. J. Fowler, M. S. Paterson, and L. Tanimoto. Optimal packing and covering in the plane are NP-complete. *Information Processing Letters*, 12:133–137, 1981.

[9] M. Garey. Optimal binary identification procedures. *SIAM J. Appl. Math.*, 23:173–186, 1972.

[10] P. Gaston and T. Lozano-Perez. Tactile recognition and localization using object models: The case of polyhedra on a plane. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6:257–266, May 1984.

[11] M. T. Goodrich. A polygonal approach to hidden-line and hidden-surface elimination. *CVGIP: Graphical Models and Image Processing*, 54:1–12, 1992.

[12] W. E. Grimson. The combinatorics of local constraints in model-based recognition and localization from sparse data. *J. ACM*, 33:658–686, 1986.

[13] W. E. Grimson and T. Lozano-Perez. Localizing overlapping parts by searching the interpretation tree. *IEEE Transactions on Pattern Recognition and Machine Intelligence*, PAMI-9:469–482, July 1987.

[14] L. Hyafil and R. Rivest. Constructing optimal binary decision trees is NP-complete. *Information Processing Letters*, 5:15–17, 1976.

[15] E. Joseph and S. Skiena. Model-based probing strategies for convex polygons. *Computational Geometry: Theory and Applications*, 2:209–221, 1992.

[16] L. Lovász. On the ratio of optimal integral and fractional covers. *Discrete Mathematics*, 13:383–390, 1975.

[17] K. Lyons and D. Rappaport. An efficient algorithm for identifying objects using robot probes. *The Visual Computer*, to appear.

[18] N. Megiddo and A. Tamir. On the complexity of locating linear facilities in the plane. *Oper. Res. Lett.*, 1:194–197, 1982.

[19] V. Mirelli. Computer vision is a highly structured optimization problem. Manuscript, Center for Night Vision and Electro-Optics, Fort Belvoir, VA, 1990.

[20] B. Moret. Decision trees and diagrams. *Computing Surveys*, pages 593–623, 1982.

[21] C. H. Papadimitriou. On certain problems in algorithmic vision. Technical report, Computer Science, UCSD, 1991.

[22] S. Skiena. Interactive reconstruction via probing. *IEEE Proceedings*, 80:1364–1383, 1992.