The Combinatorics of Heuristic Search Termination for Object Recognition in Cluttered Environments

W. Eric L. Grimson, *Member, IEEE*

***Abstract*— Many current recognition systems use constrained search to locate objects in cluttered environments. Earlier analysis of one class of methods has shown that the expected amount of search is quadratic in the number of model and data features, if all the data is known to come from a single object, but is exponential when spurious data is included. To overcome this, many methods terminate a search once an interpretation that is "good enough" is found. In this paper, we formally examine the combinatorics of this approach, showing that choosing correct termination procedures can dramatically reduce the search. In particular, we provide conditions on the object model and the scene clutter such that the expected search is at most quartic. The analytic results are shown to be in agreement with empirical data for cluttered object recognition. These results imply that it is critical to use techniques that select subsets of the data likely to have come from a single object before establishing a correspondence between data and model features.**

***Index Terms*— Complexity bounds, constrained search, object recognition.**

I. Introduction

MOST CURRENT object recognition and localization methods use a search process to find solutions from noisy sensor data in cluttered environments. Typically, this search finds interpretations of the data by identifying sets of data feature/model feature pairings that are consistent with a rigid transformation of the object model into sensor coordinates. There are many variations on this approach, including hypothesize and test methods, e.g., [18]–[22], [1], maximal clique methods, e.g., [3], and constrained tree search methods, e.g., [14], [15], [9], [26]–[28], [4]. Formal analysis of the last class of methods [11] has shown that performance is very different when all of the sensory data are known to have come from a single object as opposed to sensory data that includes spurious features. If all of the data are known to have come from a single object, the expected amount of search required to find a correct interpretation is on the order of

$$O(m^2 + ams)$$

where m is the number of model features, s is the number of sensor features, and a is a small constant. In most problems of interest, $s \gg m$ so that the expected amount of search is quadratic in the parameters of the problem and is linear in the number of data-model pairings. On the other hand, if spurious data is allowed, the expected amount of search is bounded above by an expression on the order of

$$O(ms2^c + m^2s^2[1+\epsilon]^c + bm^6 + m[1+\gamma]^s)$$

where again, m is the number of model features, s is the number of sensor features of which c correctly arise from the object, and $\epsilon, \gamma \leq 1$ are small constants. The expected amount of search is bounded below by an expression on the order of

$$o(m2^c + ms).$$

Depending on the specific parameters of the problem, different terms of these expressions will dominate, but in general, the expected search is now exponential in the problem size.

This suggests that one of the hard parts of the recognition problem is in separating out subsets of correct data from the spurious data, where, by correct, we mean a subset of the data that arises from a single object. One means of attacking this problem is to use grouping mechanisms to preselect likely subspaces of the search space on which to focus. This can be done in a data-driven fashion [20]–[22] using first principle methods for clustering data independent of the specific object. It can also be done in a model driven manner, for example, by using the generalized Hough transform [2]. In [12], we investigated the combinatorics of using such model-driven schemes. In particular, we showed that although such methods could reduce the size of the search space, they could not, in general, be used to select subspaces for which all of the sensory features came from a single object without at the same time encurring a nontrivial false positive rate.

A second approach is to use heuristic criteria to terminate the search process once an interpretation that is "good enough" has been found. In this paper, we examine this alternative. The usual method used for terminating search, e.g., [1], [15], [16], [21], [22], is to measure the "goodness" of the interpretation (by determining what fraction of the object is accounted for by the interpretation) and to terminate the search when that measure exceeds some threshold. Typical measures include the number of model features included in the interpretation or the amount of perimeter or surface area of the model included in the interpretation. In using such methods, there are two questions of interest. The first is establishing first principles methods for setting the threshold for termination. In [12], we address this question on a formal basis, showing that estimates for the threshold may be found as a function of the clutter

Manuscript received August 2, 1989; revised March 18, 1991. This work was supported by an Office of Naval Research University Research Initiative grant under contract N00014–86–K–0685 and by the Advanced Research Projects Agency of the Department of Defense under Army contract DACA76–85–K–0124. The author was also supported by the Matsushita Chair of Computer Science and Engineering and by NSF contract IRI–8900267.

The author is with the Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA 02139.

IEEE Log Number 9101439.

of the scene and the size of the model such that no false positive interpretations will be expected. In this paper, we turn to the second question, namely, to what extent does the use of premature termination of the search reduce the expected cost of the search process itself.

II. Preview Of Results And Their Implications

Because the main results of this paper consist of a set of messy combinatorial proofs, we first preview those results and their implications on object recognition. To set the stage for these results, it is convenient to break the recognition problem into three components:

1) *Selection*: Extract subsets of the data features that are likely to correspond to a single object.
2) *Indexing*: Extract a set of object models from the library likely to correspond to a selected data subset.
3) *Correspondence*: Determine if a matching exists between data features of the selected subset and model features from one of the object models such that there exists a legal transformation mapping the model features to roughly align with their matched data features.

In principle, the correspondence method involves an exponential amount of search, and part of our goal is to understand how different methods reduce that search. Our earlier formal analyses implied that if a perfect selection method could be derived and the indexing problem was correctly solved, then constrained search approaches to object recognition would be extremely efficient (i.e., quadratic in expected search). The role of a selection method is to extract subsets of the data on which to concentrate, and by perfect, we mean that such methods can reliably find subsets containing data points only from the given object. This is a strong requirement on a selection method, and one of our earlier analyses examined the effects of relaxing this so that subsets of data were considered, most of which consisted of features from a single object. In this case, we found that although the actual search was reduced, such search methods were still formally exponential. In a separate analysis, we have considered the effects of incorrect indexing [11], showing that the cost of using the wrong model is again exponential. Here, we consider the effects of terminating the search as soon as an interpretation that is sufficiently good is found with the hope that such a method can reduce the expected cost of the search from the exponential domain.

The main results of the paper (contained in the Corollaries to Proposition 3) include the following: 1) Premature termination of the search reduces the expected cost to at most a quartic function of the problem parameters if the scene clutter is small enough; 2) premature termination still has an exponential expected cost if the scene clutter is too large. This implies that selection is a crucial step in efficient object recognition but that it need not be perfect. Provided the selection method keeps the ratio of data to model features small enough (and a formal characterization of "small enough" is given), then the actual recognition stage is at most quartic. At the same time, premature termination is also a critical stage since, without it, the search is exponential.

As with any formal analysis, we make several simplifying assumptions in order to derive tractable results. To verify that these assumptions have not significantly altered the problem, we performed several tests. First, we have compared the actual number of points that are theoretically searched against the order of growth bounds that we have derived. We find that the bounds do correctly bound the actual number and that the true number is much closer to the lower bound. Second, we have applied a real recognition system to a series of real images and recorded the amount of search expended. We find that the median number of nodes searched is in close agreement with the predicted number and with the derived lower bound. We use this to conclude that our formal analysis is of relevance to the original problem and, hence, that premature termination of search, together with selection of subspaces of the search space, convert the object recognition problem from an exponential one to an efficient polynomial one.

III. The Constrained Search Model

To determine the expected cost of using premature termination, we first establish the search framework to be used in solving the recognition problem. We then review results from earlier analysis of the full constrained search method before deriving new results on the use of premature termination.

We begin by reviewing the constrained search method that was used previously in [4], [9], [14], [15], [26]–[28] as a basis for recognizing and locating objects. This approach seeks to match data features to model features in a manner that is consistent with some rigid transformation of the model into the sensory data. We assume that our models are represented by sets of geometric features, such as edges, distinctive points, surface patches, axes of cylinders, etc. and that the sensory data has been processed to obtain similar features. There are many methods for finding matches between such features; the approach taken here is to explore the space of possible correspondences by searching a tree of interpretations.

This tree search can be defined as follows. Suppose we order the data features in some arbitrary fashion. We select the first data feature, and hypothesize in turn that it is in correspondence with each of the model features. We represent this set of alternatives as a set of nodes at the same level of a tree (see Fig. 1).

Given each one of these hypothesized assignments of data feature f_1 to a model feature $F_j, j = 1, \ldots, m$, we turn to the second data feature. Again, we can consider all possible assignments of the second data feature f_2 to model features relative to each of the assignments of the first data feature. This is shown in Fig. 2. Note that the entire set of nodes in the second level of the tree corresponds to all possible matches for the first two data features.

We can continue in this manner, adding new levels to the tree, one for each data feature. A node of the interpretation tree at level n describes a partial n interpretation in that the nodes lying directly between the current node and the root of the tree identify an assignment of model features to the first n data features. Any leaf of the tree defines a complete s interpretation, where s is the total number of data features.

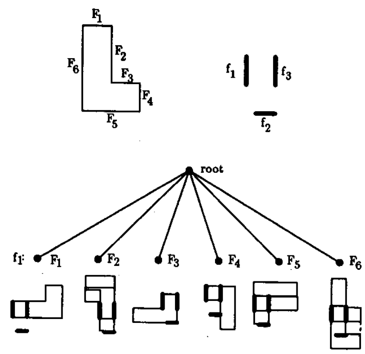


Fig. 1. We can build a tree of possible interpretations by first considering all the ways of matching the first data feature f_1 to each of the model features $F_j, j = 1, \ldots, m$. In the bottom part of the figure, we show an example of these pairings for the model and data shown at the top. In some cases, due to occlusion of the data features, a range of possible poses is given.

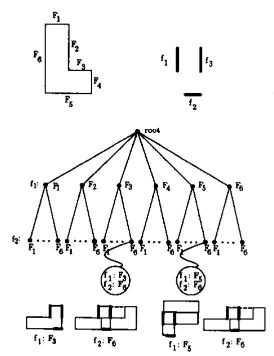


Fig. 2. For each pairing of the first data feature with a model feature, we can consider matchings for the second data feature with each of the model features. Thus, each node in the second level of the tree defines a pairing for the first two data features, which is found by tracing up the tree to the root. Two examples are shown. The example on the left is consistent with a single rigid transformation, as is shown by the fact that the two ranges of poses specified by each of the data-model feature pairings have a common pose. The example on the right is not consistent with a single transformation because its two ranges of poses do not have a common pose.

Our goal is to find consistent k interpretations, where k is as large as possible, $k \leq s$, and to find these interpretations with as little effort as possible. A simple-minded method would examine each leaf of the tree, testing to see if there exists a rigid transformation mapping each model feature into its associated data feature. This is clearly too expensive because it simply reverts to an exploration of the entire, exponential-size, search space. A better solution is to explore the interpretation tree by starting at its root and testing interpretations as we move downward in the tree. As soon as we find a node that is not consistent, i.e., for which no rigid transform will correctly align model and data feature, we terminate any further downward search below that node because adding new data-model pairings to the interpretation defined at that node will not turn an inconsistent interpretation into a consistent one.

In testing for consistency at a node, we have two different choices. We could explicitly solve for the best rigid transformation and test that all of the model features do, in fact, get mapped into agreement with their corresponding data features. This approach has two drawbacks. First, computing such a transformation is generally computationally expensive (however, see [5] and [1] for an efficient method for updating transformations), and we would like to avoid any unnecessary use of such a computation. Second, in order to compute such a transformation, we will need an interpretation of at least k data-model pairs, where k depends on the characteristics of the features. This means we must wait until we are at least k levels deep in the tree before we can apply our consistency test, and this increases the amount of work that must be done.

Our second choice is to look for less complete methods for testing consistency. We instead seek constraints that can be applied at any node of the interpretation tree, with the property that while no single constraint can uniquely guarantee the consistency of an interpretation, each constraint can rule out some interpretations. The hope is that if enough independent constraints can be combined, their aggregation will prove powerful in determining consistency but at a lower cost than fully solving for a transformation.

In previous work, we developed a set of unary and binary constraints that can be applied to this problem [14], [15]. For example, if we are matching edge segments from a grey-level image, one unary constraint is that the length of the data edge must not be longer than the corresponding model edge plus some bounded amount of error. Binary constraints apply to pairs of data-model pairings; for example, the angle between two data edges must be roughly the same as the angle between the corresponding model edges, and the range of distances between a pair of data edges must be contained within the corresponding range of distances for a pair of model edges, adjusted for error, and so on. Hence, if a unary constraint, applied to such a pairing, is true, then this implies that the data-model pairing may be part of a consistent interpretation. If it is false, however, then that pairing cannot possibly be part of such an interpretation. Binary constraints apply to pairs of data-model pairings with the same logic. These kinds of constraints have the advantages of computational simplicity, while retaining considerable power to separate consistent from inconsistent interpretations, and of applicability at virtually any node in the interpretation tree.

Formulated in this way, our approach to recognition can be considered to be a problem of constraint satisfaction or consistent labeling, which is a problem that has received considerable attention in the artificial intelligence literature, e.g., [6]–[8], [16], [17], [23]–[25], [29], [31]. When we analyze the performance of our method, we will use results from this literature to guide our development.

To use these constraints, we must now specify a means of exploring the interpretation tree. We do this using backtracking depth-first search (see Fig. 3), that is, we begin at the root of the tree and explore downwards along the first branch. At each node, we check the unary constraints applicable to the new data-model pairing, and we check the $n - 1$ sets of binary constraints obtained by considering the new data-model pairing relative to each data-model pairing defined by an ancestor node. If all these constraints are consistent,

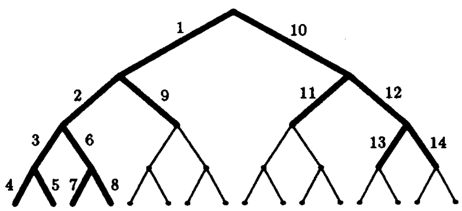


Fig. 3. The tree is searched in a depth-first, backtracking manner, starting at the root. If a node is found to be inconsistent, the downward search is terminated, and we backtrack. Any leaf of the tree that is reached by the search constitutes a hypothesized interpretation. The darker edges in the diagram indicate one example of a backtracking search.

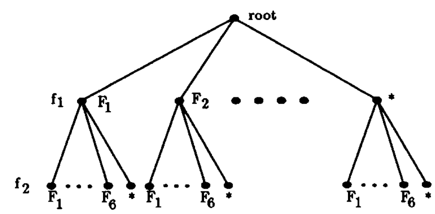


Fig. 4. The interpretation tree can be extended by adding the null character $*$ as a final branch for each node of the tree. A match of a data feature and this character indicates that the data feature is not part of the current interpretation. In the example shown, the simple tree of Fig. 2 has been extended to include the null character.

then we continue downwards in the search. If one of them is inconsistent, we backtrack to the previous node. We then explore the next branch of that node. If there are no more branches, we backtrack another level, and so on. Note that the number of constraints increases as we go lower in the tree, and hence, the likelihood that a consistent interpretation is in fact globally consistent increases.

If we reach a leaf of the tree, we have a possible interpretation of the data relative to the model, which we can verify by solving for a rigid transformation and testing that it does take all of the model features into rough agreement with their associated data features. Even if we do reach a leaf of the tree, we do not abandon the search. Rather, we accumulate that possible interpretation, back track, and continue until the entire tree has been explored and all possible interpretations have been found.

As described, our search method will succeed only when all of the data features come from the object of interest. In general, object recognition must also work in the presence of clutter in the scene in which much of the object may be hidden from view and in which much of the data is spurious, coming from other objects. The tree search method can be straightforwardly extended to handle this by introducing into our matching vocabulary a new model feature, called a *null character* feature. At each node of the interpretation tree, we add as a last resort an extra branch corresponding to this feature (see Fig. 4). This feature (denoted by a $*$ to distinguish it from actual model features F_j) indicates that the data point to which it is matched is to be excluded from the interpretation and treated as spurious data. To complete this addition to our matching scheme, we must define the consistency relationships between data-model pairings involving a null character match. Since the data point is to be excluded, it cannot affect the current interpretation, and hence, any constraint involving a data point matched to the null character is deemed to be consistent.

IV. Previous Results

This method has been used for recognition in a variety of domains [4], [9], [14], [15], [26]–[28]. Our empirical experience was that the method was very efficient when all of the data features are known to have come from a single object. When spurious data is included, however, the method slows down by several orders of magnitude. If methods for preselecting subspaces of the search space, such as the generalized Hough transform [2], are added, the method improves in efficiency. By preselection, we mean that only some subset of the possible data-model pairings are used in the search process, and typically, such subsets are chosen based on an expectation that they give rise to similar transformations of the model. If premature termination is added (i.e., halting the search process as soon as an interpretation that is "good enough" is found), the method improves even further.

In an earlier combinatorial analysis [10], we showed that some of these empirical observations were supported by formal analysis. The main points of this analysis are summarized below; formal statements of the main propositions are included in the appendix for completeness.

- When all of the data features are known to have come from a single object, the number of interpretations is asymptotic to 1.
- When only c of the s data features come from an object with m model features, the number of interpretations n_s^* is bounded above by an expression of order

$$O(n_s^*) = 2^c + [1+\alpha]^s + 2ms[1+p_2]^c$$

where p_2 is the probability of a pair of random data-model pairings satisfying binary consistency, and α is a small (< 1) constant that depends on the object characteristics and the amount of noise in the measurements. The number of interpretations is bounded below by an expression of order

$$o(n_s^*) = 2^c + [1+\beta]^s + 2ms[1+p_2]^c.$$

- The expected probability of two random data-model pairings being consistent p_2 is given by

$$p_2 = \left[\frac{\kappa}{m}\right]^2$$

where κ is a constant (usually less than 1) that can be derived from properties of the object and noise characteristics. The appendix provides details.

- If all s sensory measurements are known to lie on a single object with m equal-sized features, the sensory data is distributed uniformly, and if the noise is small enough, then the expected amount of search needed to find the interpretation is bounded by

$$m^2 \leq N_s \leq m^2 + ams$$

where a is a constant that depends on the object characteristics and the amount of noise in the sensory measurements.

- If c_0 of the s sensory measurements lie on an object with m equal-sized features, the sensory data is distributed uniformly, and if the noise is small enough, then the expected amount of search needed to find the interpretations is bounded above by an expression of order

$$O(N_s^*) = m[1+\gamma]^s + ms2^{c_0} + \delta m^6 + m^2 s^2 [1+\mu]^{c_0}$$

and is bounded below by an expression of order

$$o(N_s^*) = m2^{c_0} + ms$$

where γ, δ, μ are constants that depend on the object characteristics and the amount of sensor noise $\gamma, \mu < 1$.

As we suggested in the Introduction, these results show that the constrained search is polynomial, in fact quadratic, when all of the data is known to come from a single object but is exponential when spurious data is included. One way of reducing this exponential cost is to terminate the search as soon as an interpretation is found that is "good enough." In this paper, we consider the effects of this heuristic on the search process.

V. Setting Up The Termination Model

We define premature termination to be the process of stopping the search when an interpretation is found that is "good enough." We define our measure of goodness to be the number of data features included in an interpretation that are matched to a real model feature and not the null character. Other definitions are possible, such as the fraction of an object's perimeter that is accounted for by the data, but for our purposes, the simple counting of features suffices. Thus, we set a threshold on the size of an interpretation, and we will terminate the search as soon as we find a valid interpretation of that size. In [12], we consider the problem of how properly to select such a threshold so that there are no expected false positives. Here, we simply assume that any interpretation exceeding the threshold is a correct one.

To see how termination can reduce the search process, consider a simple example. Suppose we have a scene with $s = 6$ features, a model with $m = 2$ features, and a threshold of $t = 3$. In principle, the constrained search method would examine a tree of depth 6, the k^{th} level of which would have $(m+1)^k$ nodes to be examined, for a total of

$$\frac{(m+1)^{s+1} - 1}{m}$$

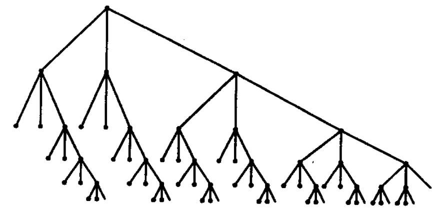


Fig. 5. Portions of the interpretation tree under the first node that need to be explored using search termination. In this case, we have set $m = 2$, $s = 6$, and $t = 3$. The circles indicate nodes actually explored.

different nodes. Of course, many of these nodes would not be examined because ancestor nodes in the tree would be inconsistent with the constraints, and the subsequent subtree could be pruned. Nonetheless, consider what happens when a threshold on search is included.

For simplicity, we consider the subtree below a node at the first level of the tree. In this case, there are, in principle,

$$\frac{(2+1)^{5+1} - 1}{2} = 364$$

nodes to be explored in this subtree. In Fig. 5, we show the subtree under a node on the first level of the tree that would be searched when a threshold on interpretation length is used to prune the tree. Notice that once we reach a node with t assignments of data features to actual model features (i.e., not to the null character), we can terminate further downward search. Similarly, once we reach a node for which it is not possible to obtain a t interpretation, no matter what happens to the remaining data features, we can again terminate further downward search. In the case shown in the figure, only 64 nodes need to be explored, which is almost an order of magnitude decrease in effort. As in the normal case, some of the nodes will not be reached due to inconsistencies in the constraints, but we can clearly see that in principle, the number of nodes to be explored is reduced from the straightforward case.

VI. The Formal Model

We will derive results on the effects of prematurely terminating the search process in several steps. We begin by defining a formal model for the probability of consistency of a node in the tree. Given that model, we derive an explicit expression for the expected number of nodes searched in a tree. We then bound this expression and use these bounds to derive simpler order-of-growth bounds on the expected search. These are summarized in the corollaries to Proposition 3, in which we show that the expected search is, at most, quartic in the parameters of the problem.

We begin with the formal model for consistency. Since our method uses both unary and binary constraints, we need to model the probability that a data-model assignment is consistent and the probability that a pair of data-model assignments are consistent.

Similar to our earlier analysis [10], we let $q_{i,I}$ denote the probability that assigning the ith data element to the Ith model element is consistent, and we let $q_{i,j;I,J}$ denote the probability that the pair of assignments $i \mapsto I, j \mapsto J$ is consistent. Our model of the recognition problem is defined as follows.

For a single data-model pairing, if the pairing is part of the correct interpretation, the probability of consistency is simply 1. Similarly, any pairing involving the null character is consistent with probability 1. If the pairing is not correct, we let the probability of consistency be p_1. Thus, we have

$$q_{i,I} = \begin{cases} 1 & \text{if } i \mapsto I \text{ is correct} \\ 1 & \text{if } I \text{ is the null character,} \\ p_1 & \text{otherwise.} \end{cases}$$

For a pair of assignments, suppose we are considering a match in which data fragments i, j are paired with model fragments I, J, respectively. We will model the situation by saying that the consistency of this pair of pairs has probability 1 if these pairings are part of the correct interpretation or if either of then is assigned to the null character. Otherwise, we will assume that the probability of consistency is p_2. Note that this is essentially assuming a random distribution of edges. It is also assuming that pairs of model edges are distinctive so that objects with partial symmetries are excluded. Thus, we have

$$q_{i,j;I,J} = \begin{cases} 1 & \text{if } i \mapsto I, j \mapsto J \text{ is correct} \\ 1 & \text{if either } I \text{ or } J \text{ are the null character,} \\ p_2 & \text{otherwise.} \end{cases}$$

Given a partial interpretation at a node, the probability of consistency is given by

$$\prod_i q_{i,I} \prod_{i \neq j} q_{i,j;I,J}.$$

We can use the above definitions for q to derive an explicit expression for the expected number of nodes in the tree. In particular, we need to count the number of nodes in the tree at each level, where by count, we mean measure the probability that the node is consistent as given by the above expression.

Proposition 1: Assume that the data features that actually arise from the object of interest are uniformly interspersed among the spurious features, occuring with frequency

$$\delta = \frac{c}{s}.$$

Assume we are given a partial interpretation based on $\ell - 1$ data features, of which u are correctly assigned, where the remaining $\ell - u - 1$ are matched to the null character. If we assign the next data feature to a real but incorrect model feature, then the number of nodes below this point in the tree that will on average be explored, denoted by $W(s, u, \ell)$, is given by

$$p_1 \left[\sum_{k=0}^{t-u-1} \sum_{i=0}^{k} \binom{k}{i} m^i p_1^{i - \lfloor \delta_i \rfloor} p_2^{\binom{i+u+1}{2} - \binom{u+\lfloor \delta_i \rfloor}{2}} \right.$$

$$+ \sum_{k=t-u}^{s-l-1} \left(\sum_{i=\max\{0, t-s+l+k-u-1\}}^{t-u-2} \binom{k-1}{i} m^{i+1} \right.$$

$$p_1^{(i+1) - \lfloor \delta(i+1) \rfloor} p_2^{\binom{i+u+2}{2} - \binom{u+\lfloor \delta(i+1) \rfloor}{2}}$$

$$+ \sum_{i=\max\{0, t-s+l+k-u\}}^{t-u-2}$$

$$\left. \binom{k-1}{i} m^i p_1^{i - \lfloor \delta_i \rfloor} p_2^{\binom{i+u+1}{2} - \binom{u+\lfloor \delta_i \rfloor}{2}} \right). \quad (1)$$

∎

A proof of this proposition is deferred to the Appendix.

We can check the correctness of this expression by setting p_1 and p_2 to 1. Applying the resulting expression to the case of $u = 0, \ell = 1, m = 2, s = 6, t = 3$, as shown in Fig. 5, yields the correct result of 64 nodes.

We can use **Proposition 1** to establish bounds on the search of such a subtree. This is done in the following proposition:

Proposition 2: Expression (1) of **Proposition 1** can be bounded by

$$\mathrm{W}(s, u, l) \leq p_1 p_2^u \Big[(t-u)^{j_0(u)+1} \mu^{j_0(u)-1}$$

$$+ (s - l - t + u)(t - u - 1)[(s - l - 2)\mu]^{i_0(u, s-l-1)} \times$$

$$\times \left(1 + m p_1^{1-\delta} p_2^{2u+1-\frac{\delta^2 + \delta(2u-1)}{2}} \right) \Big]$$

and by

$$W(s, u, \ell) \geq$$

$$p_1 p_2^{2u} \Big[s - t + u - \ell + 2 \Big] \left[1 - (t-u) m p_1^{1-\delta} p_2^{\frac{2u - \delta(2u-3) - \delta^2 + 2}{2}} \right]$$

where

$$\mu(u) = m p_1^{1-\delta} p_2^{f(u)}$$

$$f(u) = \frac{2u(1-\delta) + 2 + \delta - \delta^2}{2}$$

$$j_0 = \left\lfloor \frac{(t-u)\mu(u) - 1}{1 + \mu(u)} \right\rfloor$$

$$i_0(u, k) = \left\lfloor \frac{(k-1)\nu(u) - 1}{1 + \nu(u)} \right\rfloor$$

$$\nu(u) = m p_1^{1-\delta} p_2^{\frac{2u(1-\delta) + 4 + \delta - 3\delta^2}{2}}.$$

∎

This proposition gives us bounds on the expected search of a particular subtree. How do we use it to bound the search of the whole tree? Under the assumption that the c correct data features are uniformly interspersed throughout the full set of s data features, we can see that at the top level of the tree, we must search m subtrees, with $u = 0, \ell = 1$, that is, for each possible assignment of the first data feature to a real model feature, we must explore the appropriate subtree. Since the first data feature is not part of the true object, once we have exhausted these subtrees, we must move on to interpretations that exclude the first data point by considering the portion of the tree below the node that pairs the first data point to the null character. Under this node, we consider pairings of the second data feature. Again, we must consider m subtrees, with $u = 0, \ell = 2$. We continue this process until we reach level $\ell = \frac{1}{\delta}$. In this case, we have a data feature that does have a correct match, and on average, this will be found after we have searched $\frac{m}{2}$ subtrees at this level. We then repeat this process below this node in the tree, with $u = 1$, and so on. Hence, the expected total amount of search is given by

$$W(s) = \sum_{j=0}^{t-1}\left(\left[\sum_{i=1}^{\frac{1}{\delta}-1} mW\left(s, j, \frac{1}{\delta}j + i\right) + 1\right] + \frac{m}{2}W\left(s, j, \frac{1}{\delta}(j+1)\right) + 1\right). \quad (2)$$

To obtain bounds on this expression, we simply need to substitute from **Proposition 2**, and simplify.

Proposition 3: Given a uniform distribution of correct data features among the spurious and given the previously derived [10] expression for the binary probability of consistency

$$p_2 = \left(\frac{\kappa}{m}\right)^2$$

the expected amount of search is bounded by

$$\begin{aligned} W(s) \le t\frac{1}{\delta} + \frac{mp_1}{\delta}\Bigg[& t^{j_0+1}\mu^{j_0-1}\left(1 + (t-1)\frac{\kappa^2}{m^2}\right) \\ & + \gamma^{i_0} f(t-1)\left(\frac{1-p_2^t}{1-p_2} + \beta\frac{1-p_2^{(3-\delta)t}}{1-p_2^{(3-\delta)}}\right) \\ & - \gamma^{i_0}[(d-1)(t-1)+f]\left(\left[\frac{p_2(1-p_2^t)}{(1-p_2)^2} - \frac{tp_2^t}{1-p_2}\right]\right. \\ & \left. + \beta\left[\frac{p_2^{(3-\delta)}\left(1-p_2^{(3-\delta)t}\right)}{\left(1-p_2^{(3-\delta)}\right)^2} - \frac{tp_2^{(3-\delta)t}}{1-p_2^{3-\delta}}\right]\right)\Bigg] \end{aligned}$$

where

$$\begin{aligned} \mu &= \frac{\kappa^2}{m} \\ \gamma &= (s-3)\mu \\ i_0 &= \lfloor (s-3)\mu - 1 \rfloor \\ j_0 &= \lfloor \kappa^2 - 1 \rfloor \\ \beta &= mp_1^{1-\delta}p_2^{\frac{2-\delta^2+\delta}{2}} \\ f &= s - t - \frac{1}{2}\left(\frac{1}{\delta}+1\right) \\ d &= \frac{1}{\delta} - 1 \end{aligned}$$

and by

$$\begin{aligned} W(s) \le \frac{t}{\delta} + mp_1 & \left[a\frac{1-p_2^{2t}}{1-p_2^2} - b\frac{p_2^2(1-p_2^{2t})}{(1-p_2^2)^2} + \frac{btp_2^{2t}}{1-p_2^2}\right. \\ & \left. + \alpha\left(a\frac{1-p_2^{(3-\delta)t}}{1-p_2^{3-\delta}} - b\frac{p_2^{(3-\delta)}\left(1-p_2^{(3-\delta)t}\right)}{\left(1-p_2^{(3-\delta)}\right)^2} + \frac{btp_2^{(3-\delta)t}}{1-p_2^{(3-\delta)}}\right)\right] \end{aligned}$$

where

$$\begin{aligned} a &= (s-t+2)\left(\frac{1}{\delta} - \frac{1}{2}\right) - \frac{1}{\delta}\frac{1}{2}\left(\frac{1}{\delta} - 1\right) \\ b &= \left(\frac{1}{\delta} - 1\right)\left(\frac{1}{\delta} - \frac{1}{2}\right) \\ \alpha &= mp_1^{1-\delta}p_2^{\frac{3\delta-\delta^2}{2}}. \end{aligned}$$

∎

The key results follow from this proposition.

Corollary 3.1: The order of magnitude of the expected search is given by:

$$o(W(s)) = ms\frac{s}{c}$$

and by

$$O(W(s)) = amts\frac{s}{c}\left(1 + \frac{\kappa^2}{m}\right)^2\left(\kappa^2\frac{s}{m}\right)^{\lfloor \frac{s}{m}\kappa^2 - 1 \rfloor}$$

where a is a small constant.

Proof: For both bounds, we can simply identify the dominant term and substitute in the bounds on the variables, yielding

$$o(W(s)) = m\left(s - t + 2 - \frac{s}{2c}\right)\frac{s}{c} \quad (3)$$

and

$$O(W(s)) = mt\left(s - t - \frac{1}{2} - \frac{s}{2c}\right)\frac{s}{c}\left(1 + \frac{\kappa^2}{m}\right)^2\left(\kappa^2\frac{s}{m}\right)^{\lfloor \frac{s}{m}\kappa^2 - 1 \rfloor}. \quad (4)$$

The simpler expressions follow. ∎

Corollary 3.2: If the scene clutter and the noise in the data is such that

$$\frac{s}{m} < \frac{2}{\kappa^2}$$

then premature termination has an expected search that is of order

$$O(W(s)) = amts\frac{s}{c}$$

and

$$o(W(s)) = ms\frac{s}{c}.$$

Proof: If the conditions hold, then the exponent in the previous corollary becomes 0, and the upper bound on the search reduces to

$$O(W(s)) = mt\left(s - t - \frac{1}{2} - \frac{s}{2c}\right)\frac{s}{c}\left(1 + \frac{\kappa^2}{m}\right)^2.$$

The simplification follows. ∎

VII. Implications Of The Results

Several interesting conclusions may be drawn from the above analysis. First, we note from **Corollary 3.1**, that we cannot guarantee that terminated search will result in a polynomial time algorithm because the upper bound is still exponential. At the same time, however, the search is clearly reduced because both the base and the exponent are much reduced from the normal constrained search.

Corollary 3.2 provides a fascinating result, however, since it indicates conditions on the problem under which the expected search does become polynomial, basically implying that if the scene clutter is small enough, a polynomial algorithm results. This has two interesting implications. If the number of scene features is small enough relative to the size of the model, it implies that terminated search will perform well. When the scene clutter increases, however, we must provide some form of grouping or selection to reduce the number of scene features actually considered in the search below

$$s < \frac{2m}{\kappa^2}.$$

Here, selection means isolating a subset of the data features where most are believed to have come from a single instance of a known object. Examples of methods for selecting such salient subsets of the data include [20]–[22], [30].

This nicely extends our earlier results on the role of selection in efficient object recognition. The results of [10] imply that for pure constrained search, knowing that all of the data features are from a given object will reduce the expected search to the polynomial domain, but general constrained search remains exponential. This suggests that our selection mechanism must be very accurate in selecting out subsets of the data features for consideration since if even one spurious point is included, we must either use an exponential search method or tolerate having the entire subset of data features being rejected. When premature search termination is added, however, **Corollary 3.2** implies that we can tolerate considerably more uncertainty on the part of the selection process and still have an efficient search method. We simply require that the selection method allows an amount of spurious data that is bounded by the conditions of **Corollary 3.2.**

In addition, note that both **Proposition 3** and its Corollaries involve the constant κ, which is determined by properties of the object model and the sensing system. In particular, κ increases with increasing noise in the sensory data, and this, as expected, implies both that the amount of expected search will increase and that the amount of spurious data that can be tolerated, while maintaining a polynomial algorithm, decreases. Standard values for κ are on the order of

$$\kappa \approx 0.2\frac{P}{D}$$

where P is the total perimeter of the object (for the case of 2D objects), and D is the dimension of the image. Given this, we see that our conditions for a polynomial search are that

$$s \leq \frac{2m}{\kappa^2} \approx 50m\left(\frac{D}{P}\right)^2$$

so that if the object is of a size on the order of the image ($P \approx D$), considerable amounts of spurious data are still tolerable while maintaining a polynomial search algorithm.

A. Comparing Search Results

To more directly compare the results derived here, we can consider some earlier analysis of constrained search in object recognition. In [10], we analyzed the combinatorial behavior of the constrained search approach and show two major results. The first is that if all of the data are known to have come from a single object so that we need not use the null character to exclude spurious data, then the amount of search was bounded by

$$m^2 \leq W_{no-occ} \leq p_1 m^2 + (1 + p_1\kappa)^2 ms.$$

Hence the search process is polynomial in this case.

If, however, spurious data is included, we showed that the search is exponential. In our earlier analysis, we did not use any assumptions on the distribution of the correct sensory data features in the search process. To more directly compare the two methods, below, we derive bounds on the constrained search process under the assumption of uniform distribution of the correct data features.

Proposition 4: If the sensory data arising from a correct interpretation are uniformly distributed among the spurious data, then the amount of search expended by the normal constrained search method is bounded by

$$m\frac{s}{c}2^c \leq$$
$$W_{occ} \leq m\frac{s}{c}2^c + \frac{m}{\epsilon}[1+\epsilon]^s\left[1 + \frac{p_2}{1+\epsilon}\right]^{c-1} + \frac{m^3 s}{\kappa^2}\frac{s}{c}[1+p_2]^c. \tag{5}$$

∎

With these results, it is clear that premature termination of the search process can significantly reduce the work involved in locating an object. From **Corollary 3.2**, we know that if the scene clutter is small enough, the expected search reduces to order

$$ms\frac{s}{c} \leq W_{term} \leq mts\frac{s}{c}.$$

This is clearly significantly smaller than the expressions in **Proposition 4.**

B. Consistency with Real Data

To further demonstrate the relevance of the results derived here, we compare the predictions of the analysis with data obtained from real examples. In particular, we selected a representative cluttered image, including an instance of a known object, and extracted a set of features from the image. We then applied the RAF [14], [15] recognition system to the resulting data. For this image, the data were placed in 100 different random orderings, and the method was applied for each of eight different termination conditions using thresholds of $0.3, 0.4, 0.5$, and $0.6m$, where m is the number of features in the object model, and using thresholds of $0.3, 0.4, 0.5,$

TABLE I
STATISTICS ON RUNNING A REAL SEARCH PROCESS COMPARED WITH PREDICTED BOUNDS. (The first and third lines show the predicted bounds, using equations (3) and (4) respectively. The second line shows the actual theoretical number of nodes explored, obtained by summing equation (2) using equation (1). The median, mean and standard deviation over 100 trials are shown using the number of matched features as a termination procedure, and using the percentage of object perimeter as a termination procedure. Finally the predicted and observed nodes using no termination are shown.)

	th = 0.3 m	0.4 m	0.5 m	0.6 m
Predicted lower bound	1234	1152	1069	987
Actual nodes, average case	1776	1635	1498	1364
Predicted upper bound	7017	8675	9992	10969
Median, using features	2689	2993	2605	2143
Mean, using features	6223	6610	9536	15340
Deviation, using features	9440	9345	30278	47872
Median, using perimeter	6627	8834	8977	9479
Mean, using perimeter	19437	16307	23297	38362
Deviation, using perimeter	50199	34215	50062	104662
Predicted lower bound, Full search	5.4×10^6			
Observed nodes, Full search	10^7			
Predicted upper bound, Full search	3.2×10^8			

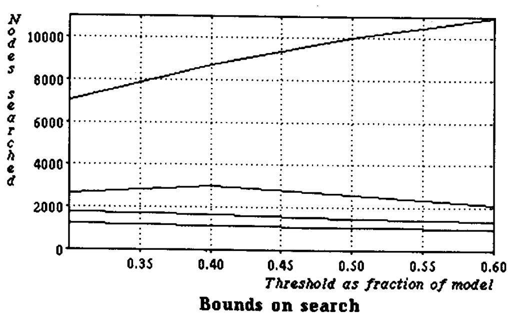


Fig. 6. Graphs on terminated search. The top graph shows the predicted upper bound as a function of the termination condition. The second graph shows the median number of nodes searched for real data. The third graph shows the actual number of nodes predicted. The fourth graph shows the predicted lower bound.

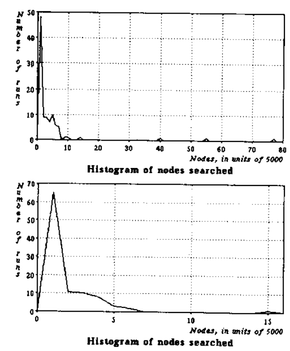


Fig. 7. Histogram of the number of nodes searched for 100 random orderings of the sensor data. The ordinate is the number of trials, and the abscissca is in units of 5000 nodes. The first histogram is for a termination procedure based on the perimeter matched, and the second is for a termination procedure based on the number of features matched.

and $0.6P$, where P is the perimeter of the object model. In the first four cases, the quality of an interpretation was simply the number of matched features, and in the second four cases, the quality of an interpretation was the sum of the lengths of the matched features. In this particular example, $m = 20, s = 35, c_0 = 17$. Table I summarizes the results.

Several observations are possible. First, the derived bounds on the number of explored nodes ((3) and (4)) do correctly contain the actual number obtained by evaluating (1). Second, the median number of nodes searched for real data, using number of features matched as a termination procedure, not only lies within the predicted bounds but also is in close agreement with the actual theoretical number. This is also shown in graphical form in Fig. 6.

The data in Table I shows that the mean number of nodes searched is higher. This is to be expected since the analysis was based on an assumption of uniform distribution of correct data features among the spurious. The increase in search when more spurious data are amoung the first features examined is much larger than the decrease in search when more of the correct features are among the first features examined. This leads to a long tail in the histogram of nodes searched for different trials, as shown in Fig. 7. By comparison, we also list the observed number of nodes explored when no termination is used and the predicted bounds on this number using (5).

As a second test of our analysis, we applied our recognition system to a series of ten real images for the particular termination condition of the number of matched features exceeding $0.3m$, where m is the total number of model features. In Fig. 8, we plot the predicted number of nodes

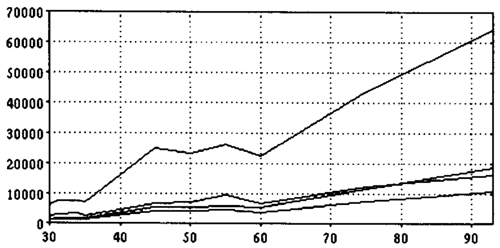


Fig. 8. Graphs on terminated search. The top graph shows the predicted upper bound as a function of the number of sensory data features. The second graph shows the observed median number of nodes searched for real data. The third graph shows the predicted actual number of nodes to be searched. The fourth graph shows the predicted lower bound.

searched, the derived bounds on that number, and the median observed number of nodes searched over 100 trials on the image, all as a function of the number of sensor features. Of course, there are other factors that influence both the actual and predicted search required, including the amount of occlusion and the particular arrangement of data features. These graphs are simply intended to display the statistics of the test in a convenient form. One can seen that not only is the predicted number always contained within the bounds, it is much closer to the lower bound and that the observed number of nodes closely follows the prediction.

From these tests, we can conclude that the assumptions made in deriving our formal analysis are in reasonable agreement with actual practice and, hence, are of relevance in judging the impact of premature termination on constrained search.

VIII. Conclusion

As a consequence, the main conclusion we can draw is that premature termination of a constrained search method can dramatically reduce the expected search required to recognize and locate objects in cluttered noisy data. To obtain polynomial time algorithms for recognition, we must keep the ratio of scene clutter to object size below a well defined bound, and this implies that for significantly cluttered scenes, some type of grouping or selection mechanism is needed to select out subsets of the data features that are likely to include a subset arising from an instance of a known object.

Appendix

In this Appendix, we present formal proofs of the propositions stated in the main text.

We begin by stating the main propositions from earlier analysis in Grimson [10]. (Note that the numbers of the propositions refer to the numbers used in that article.) In that analysis, we first derived bounds on the number of consistent interpretations both in the case of data known to have come from a single object and in the case of spurious data.

Proposition 1 [10]: If all of the k sensory measurements are known to lie on a single object with m features, then the number of interpretations n_k is bounded by

$$n_k \le \left[1 + (m-1)p_1 p_2^{\frac{k-1}{2}}\right]^k$$

and by

$$n_k \ge 1 + \left[p_2^{\frac{1}{2}} + p_1(m-1)\right]^k p_2^{\frac{k(k-1)}{2}} - p_2^{\frac{k^2}{2}}$$

where p_1 is the probability of a random data-model assignment satisfying unary consistency, and p_2 is the probability of a pair of random data-model assignments satisfying binary consistency. ∎

Proposition 2 [10]: Given an object with m faces and given k sensory data points of which c actually lie on the object, the number of interpretations n_k^* is bounded by

$$\begin{aligned} n_k^* \le\, & 2^c - \left[1 + p_2\right]^c + \left[1 + mp_1 p_2^{\frac{1}{2}}\right]^{k-c}\left[p_2 + 1 + mp_1 p_2^{\frac{1}{2}}\right]^c \\ & + mp_1\left[1 - p_2^{\frac{c+1}{2}}\right]\left[1 + p_2\right]^{c-1}\left[k + p_2(k-c)\right] \end{aligned}$$

and by

$$\begin{aligned} n_k^* \ge\, & 2^c - \left[1 + p_2^{\frac{k-c}{2}}\right]^c + \left[1 + (m-1)p_1 p_2^{\frac{k-1}{2}}\right]^{k-c} \\ & \left[1 + (m-1)p_1 p_2^{\frac{k-1}{2}} + p_2^{\frac{k-1}{2}}\right]^c \\ & + p_1(m-1)\left[1 + p_2\right]^{c-1}\left[k + p_2(k-c)\right] \\ & - p_1(m-1)p_2^{\frac{k-1}{2}}\left[1 + p_2^{\frac{k-c}{2}}\right]^{c-1}\left[k + p_2^{\frac{k-c}{2}}(k-c)\right] \end{aligned}$$

where p_1 is the probability of a random data-model assignment satisfying unary consistency, and p_2 is the probability of a pair of random data-model assignments satisfying binary consistency. ∎

To obtain order of magnitude expressions on the amount of search required to find these interpretations, we need to relate the probability of consistency to aspects of the problem. We established that the probability of consistency is inversely proportional to the number of model features for a fixed amount of sensor noise and a fixed size object model:

Proposition 3 [10]: Given a two-dimensional object with m equal sized edges of length L and given sensory data that is distributed uniformly in transform space with a uniform distribution of lengths, the expected probability of two random data-model pairings being consistent p_2 is given by

$$p_2 = \left[\frac{\kappa}{m}\right]^2$$

where

$$\begin{aligned} \kappa &= \kappa_w \\ &= \sqrt{\frac{4\epsilon_a}{\pi}\left[\pi(\epsilon_p^*)^2 + 2\epsilon_p^*(1-h^*)\right] + \frac{\sin\epsilon_a}{\pi}(1-h^*)^2\left[\frac{P}{D}\right]} \end{aligned}$$

in the worst case, and

$$\begin{aligned} \kappa &= \kappa_u \\ &= \sqrt{\frac{4\epsilon_a}{\pi}\left[\pi(\epsilon_p^*)^2 + \epsilon_p^*(1-h^*)\right] + \frac{\sin\epsilon_a}{2\pi^2}(1-h^*)^2\left[\frac{P}{D}\right]} \end{aligned}$$

in the uniform distribution case and where ϵ_a is a bound on the error in measuring orientation, ϵ_p is a bound on the error in measuring position, h is the minimum length data edge, $\epsilon_p^* = \frac{\epsilon_p}{L}$, $h^* = \frac{h}{L}$, P is the perimeter of the object, and D is the dimension (width) of the image. ∎

To illustrate the range of values for this constant, in Table II, we list the values for κ_u for a range of values of ϵ_p^* and a range of values of P/D. We fix $h^* = 2\epsilon_p^*$ and $\epsilon_a = \tan^{-1} 2\epsilon_p^*$. As expected, the constant κ_u increases with increasing noise and as the size of the object increases.

A similar result holds for three-dimensional objects [10]. This result can be used to establish the following two sets of bounds on the amount of search involved.

Proposition 6 [10]: If all of the s sensory measurements are known to lie on a single two-dimensional object with m equal sized edges of length L, $m \geq 3$, the sensory data is distributed uniformly in transform space, with a uniform length distribution, and if the noise is small enough, then the expected amount of search needed to find the interpretation is bounded by

$$m^2 \leq N_s \leq m^2 + ams$$

where a is a constant that depends on the object characteristics and the amount of noise in the sensory measurements. ∎

Proposition 9 [10]: If c_0 of the k sensory measurements lie on a two-dimensional object with m equal sized edges of length L, the sensory data is distributed uniformly in transform space, with a uniform length distribution, and if the noise is small enough, then the expected amount of search needed to find the interpretations, for m large, is bounded by

$$\begin{aligned} N_s^* \leq m\Bigg[&\frac{[1+p_1\kappa]^s}{p_1\kappa} + 2^{c_0}[s - c_0 + 1] \\ &+ p_1 m\left[\frac{1}{\alpha^2} + [1+\alpha]^{c_0}\left[\binom{s}{2} - \binom{c_0}{2} + \frac{c_0}{\alpha(1+\alpha)}\right]\right]\Bigg] \\ N_s^* \geq m&\left[2^{c_0+1} + s - c_0 - 3\right] \end{aligned}$$

where

$$\alpha = \frac{\kappa^2}{m^2}$$

and where κ is a constant that depends on the object characteristics and the amount of sensor noise, and p_1 is the probability of a random data-model assignment satisfying unary consistency. ∎

TABLE II
VALUES FOR THE CONSTANT FOR A κ_u RANGE OF VALUES OF ϵ_p^* AND A RANGE OF VALUES OF P/D. (We fix $h^* = 2\epsilon_p^*$ and $\epsilon_a = \tan^{-1} 2\epsilon_p^*$.)

$P/D =$	0.125	0.25	0.5	1	2	4	8
$\epsilon_p^* = 0.01$	0.002	0.004	0.008	0.016	0.033	0.065	0.131
$\epsilon_p^* = 0.1$	0.021	0.042	0.085	0.169	0.338	0.677	1.354
$\epsilon_p^* = 0.5$	0.111	0.222	0.443	0.886	1.772	3.545	7.090

Given these results as a basis, the text of the paper presents a similar analysis for the case of premature termination. The main results, with proofs, are summarized below.

Proposition 1: Assume that the data features that actually arise from the object of interest are uniformly interspersed among the spurious features occuring with frequency

$$\delta = \frac{c}{s}.$$

Assume we are given a partial interpretation based on $\ell - 1$ data features of which u are correctly assigned, where the remaining $\ell - u - 1$ is matched to the null character. If we assign the next data feature to a real, but incorrect, model feature, then the number of nodes below this point in the tree that will, on average, be explored, denoted by $W(s, u, \ell)$, is given by the equation at the bottom of the page.

Proof: We can see how this sum arises by the following argument. First, the probability that this assignment satisfies the unary constraints is given by p_1, which multiplies the remaining summations. Since we already have a u interpretation in hand, and since we are assigning the next data feature a nonnull character, we must explore the next $t - u - 1$ levels in detail. Hence, the first sum in the expression counts the number of nodes in this case. The summation over k counts the number of nodes at each succeeding level, and the summation over i counts the nodes at a particular level by considering the number of features assigned a nonnull character. For i such features, there are $\binom{k}{i}$ different ways of selecting them, and for each one, there are m possible assignments. To determine the consistency, we multiply by the probability of applicable unary and binary constraints holding true. Note that the exponent for the unary constraint probability counts those data-model

$$\begin{aligned} p_1\Bigg[&\sum_{k=0}^{t-u-1}\sum_{i=0}^{k}\binom{k}{i} m^i p_1^{i-\lfloor\delta i\rfloor} p_2^{\binom{i+u+1}{2}-\binom{u+\lfloor\delta i\rfloor}{2}} \\ &+ \sum_{k=t-u}^{s-\ell-1}\Bigg(\sum_{i=\max\{0,t-s+\ell+k-u-1\}}^{t-u-2}\binom{k-1}{i} m^{i+1} p_1^{(i+1)-\lfloor\delta(i+1)\rfloor} p_2^{\binom{i+u+2}{2}-\binom{u+\lfloor\delta(i+1)\rfloor}{2}} \\ &\qquad + \sum_{i=\max\{0,t-s+\ell+k-u\}}^{t-u-2}\binom{k-1}{i} m^i p_1^{i-\lfloor\delta i\rfloor} p_2^{\binom{i+u+1}{2}-\binom{u+\lfloor\delta i\rfloor}{2}}\Bigg)\Bigg]. \end{aligned}$$

∎

assignments that are not correct. The exponent for the binary constraint probability counts the total number of possible pairs of data-model assignments minus those involving only correct assignments.

Once we have reached the level in the tree at which the first possible t interpretations may occur, our search narrows. In particular, we need not consider exploring portions of the tree for which interpretations of size larger than t are involved, and we need not consider exploring portions of the tree for which interpretations of size t are impossible. The remaining two summations count these cases, where the first one counts those cases in which the most recently assigned data feature has been given a nonnull character and the second one counts those cases for which the most recently assigned data feature has been matched to the null character. ∎

Proposition 2: The expression of **Proposition 1** can be bounded by

$$\begin{aligned}&W(s,u,\ell)\\&\leq p_1 p_2^u\Big[(t-u)^{j_0(u)+1}\mu^{j_0(u)-1}\\&\quad+(s-\ell-t+u)(t-u-1)[(s-\ell-2)\mu]^{i_0(u,s-\ell-1)}\times\\&\quad\times\left(1+mp_1^{1-\delta}p_2^{2u+1-\frac{\delta^2+\delta(2u-1)}{2}}\right)\Big]\end{aligned}$$

and by

$$\begin{aligned}W(s,u,\ell)\geq&\\&p_1p_2^{2u}[s-t+u-\ell+2]\\&\left[1-(t-u)mp_1^{1-\delta}p_2^{\frac{2u-\delta(2u-3)-\delta^2+2}{2}}\right]\end{aligned}$$

where

$$\begin{aligned}\mu(u)&=mp_1^{1-\delta}p_2^{f(u)}\\f(u)&=\frac{2u(1-\delta)+2+\delta-\delta^2}{2}\\j_0(u)&=\left\lfloor\frac{(t-u)\mu(u)-1}{1+\mu(u)}\right\rfloor\\i_0(u,k)&=\left\lfloor\frac{(k-1)\nu(u)-1}{1+\nu(u)}\right\rfloor\\\nu(u)&=mp_1^{1-\delta}p_2^{\frac{2u(1-\delta)+4+\delta-3\delta^2}{2}}.\end{aligned}$$

Proof: To use (1), we want to obtain closed form bounds on the sums. We begin with an upper bound.

Consider the first sum in (1). First, we can use

$$\delta i-1\leq\lfloor\delta i\rfloor\leq\delta i$$

to remove the dependence on $\lfloor.\rfloor$. Second, since $p_2\leq 1$, we can get an upper bound on the expression by replacing the resulting exponent for p_2 with a linear expression in i that is smaller than the current exponent, in particular, by replacing terms in i^2 by similar terms in i. This leads to the upper bound for the first summation of

$$p_2^u\sum_{k=0}^{t-u-1}[1+\mu]^k$$

where

$$\begin{aligned}\mu(u)&=mp_1^{1-\delta}p_2^{f(u)}\\f(u)&=\frac{2u(1-\delta)+2+\delta-\delta^2}{2}.\end{aligned}$$

We can simplify this by using the geometric progression to yield

$$p_2^u\frac{[1+\mu]^{t-u}-1}{\mu}. \tag{7}$$

This is still an exponential, albeit a small one. We can reduce this further by observing that

$$[1+\mu]^{t-u}=\sum_{j=0}^{t-u}\binom{t-u}{j}\mu^j \tag{8}$$

and asking when the largest term occurs. In general

$$\sum_{j=0}^{n}\binom{m}{j}\epsilon^j$$

will reach a maximum for the smallest j such that the jth term is larger than the $j+1$st term. This implies

$$j+1\geq(m-j)\epsilon$$

or, equivalently, that the index for the largest term is

$$j=\left\lfloor\frac{m\epsilon-1}{1+\epsilon}\right\rfloor. \tag{9}$$

In our particular case, we let

$$j_0=\left\lfloor\frac{(t-u)\mu-1}{1+\mu}\right\rfloor.$$

An examination of μ under the limits on δ and u shows that

$$mp_1p_2^u\leq\mu\leq mp_2.$$

In [10], we showed that if the data features are randomly distributed, then the probability of binary consistency is given by

$$p_2=\left(\frac{\kappa}{m}\right)^2$$

where κ is a constant that depends on the characteristics of the object model and the amount of sensor noise. Substituting, we see that

$$0\leq\mu\leq\frac{\kappa^2}{m}.$$

Hence

$$j_0\leq\lfloor\kappa^2-1\rfloor.$$

Using (7), we have

$$\begin{aligned}[1+\mu]^{t-u}&\leq 1+(t-u)\binom{t-u}{j_0}\mu^{j_0}\\&\leq 1+(t-u)^{j_0+1}\mu^{j_0}\end{aligned}$$

and substitution into (6) implies that an upper bound on the first summation in (1) is given by

$$p_2^u(t-u)^{j_0(u)+1}\mu^{j_0(u)-1}. \tag{10}$$

Now consider the second summation in (1). This is bounded above by

$$\sum_{k=t-u}^{s-\ell-1} \cdot \sum_{i=0}^{t-u-2} \binom{k-1}{i} m^{i+1} p_1^{(i+1)-\lfloor \delta(i+1) \rfloor} p_2^{\binom{i+u+2}{2}-\binom{u+\lfloor \delta(i+1) \rfloor}{2}}. \tag{11}$$

We can use the same method as above, replacing the exponent for p_2 with a smaller exponent linear in i, which yields an upper bound on expression (6) of

$$\sum_{k=t-u}^{s-\ell-1} \cdot \sum_{i=0}^{t-u-2} \binom{k-1}{i} m p_1^{1-\delta} p_2^{2u+1-\frac{\delta^2+\delta(2u-1)}{2}} \left(m p_1^{1-\delta} p_2^{g}(u)\right)^i$$

where

$$g(u) = \frac{2u(1-\delta)+4+\delta-3\delta^2}{2}.$$

If we let

$$\nu(u) = m p_1^{1-\delta} p_2^{g(u)}$$

then a similar analysis to the first case indicates that the largest term occurs for index

$$i_0(u,k) = \left\lfloor \frac{(k-1)\nu(u)-1}{1+\nu(u)} \right\rfloor.$$

This allows us to further reduce the bound for (10) to

$$\sum_{k=t-u}^{s-\ell-1} m p_1^{1-\delta} p_2^{2u+1-\frac{\delta^2+\delta(2u-1)}{2}} (t-u-1)[(k-1)\nu]^{i_0(u,k)}. \tag{12}$$

Now, the maximum value for ν is the same as the maximum for μ, namely

$$\nu \leq \frac{\kappa^2}{m}.$$

Hence, $k\nu$ can be on the order of $\frac{s}{m}\kappa^2$. which in general will be larger than 1. This implies that the largest term in the summation in (11) will occur for i_0 as large as possible, and this leads to the following bound for the second summation in (1):

$$m p_1^{1-\delta} p_2^{2u+1-\frac{\delta^2+\delta(2u-1)}{2}} (s-\ell-t+u)(t-u-1)[(s-\ell-2)\nu]^{i_0(u,s-\ell-1)}. \tag{13}$$

A similar analysis of the third summation yields a bound of

$$(s-\ell-t+u)(t-u-1)p_2^u[(s-\ell-2)\mu]^{i_1(u,s-\ell-1)} \tag{14}$$

where

$$i_1(u,k) = \left\lfloor \frac{(k-1)\mu-1}{1+\mu} \right\rfloor.$$

By piecing together (9), (12), and (13) and by noting that $\nu \leq \mu$, we have as an upper bound

$$W(s,u,\ell) \leq p_1 p_2^u \Big[(t-u)^{j_0(u)+1}\mu^{j_0(u)-1} + (s-\ell-t+u)(t-u-1)[(s-\ell-2)\mu]^{i_0(u,s-\ell-1)} \times \times \left(1 + m p_1^{1-\delta} p_2^{2u+1-\frac{\delta^2+\delta(2u-1)}{2}}\right)\Big]. \tag{15}$$

Now, consider a lower bound on (1). Consider the first sum:

$$\sum_{k=0}^{t-u-1} \sum_{i=0}^{k} \binom{k}{i} m^i p_1^{i-\lfloor \delta i \rfloor} p_2^{\binom{i+u+1}{2}-\binom{u+\lfloor \delta i \rfloor}{2}}. \tag{16}$$

To reduce this expression, we need to replace the exponent for p_2 with a larger expression linear in i. Replacing $\lfloor \delta i \rfloor$ with $\delta i - 1$ and replacing quadratic terms in i with linear ones, we get a lower bound on (15) of

$$\sum_{k=0}^{t-u-1} \sum_{i=0}^{k} \binom{k}{i} m^i p_1 p_1^{(1-\delta)i} p_2^{2u} p_2^{h(u,k)i} \tag{17}$$

where

$$h(u,k) = \frac{k(1-\delta^2)+2u+1+2\delta(1-u)+\delta}{2}.$$

By Vandermondes binomial theorem, this reduces to

$$p_1 p_2^{2u} \sum_{k=0}^{t-u-1} \left[1 + m p_1^{(1-\delta)} p_2^h\right]^k.$$

Since we are seeking a lower bound on this expression, we note that the term in the summation is always greater than 1, and we obtain as a lower bound for (16)

$$p_1 p_2^{2u}(t-u). \tag{18}$$

Now, consider the second summation in (1):

$$\sum_{k=t-u}^{s-\ell-1} \sum_{i=\max\{0,t-s+\ell+k-u-1\}}^{t-u-2} \binom{k-1}{i} m^{i+1} p_1^{(i+1)-\lfloor \delta(i+1) \rfloor} p_2^{\binom{i+u+2}{2}-\binom{u+\lfloor \delta(i+1) \rfloor}{2}}.$$

We can bound this below by only considering terms for which i runs from 0 to $t-u-2$:

$$\sum_{k=t-u}^{s+u+1-t-\ell} \cdot \left[\sum_{i=0}^{t-u-2} \binom{k-1}{i} m^{i+1} p_1^{(i+1)-\lfloor \delta(i+1) \rfloor} p_2^{\binom{i+u+2}{2}-\binom{u+\lfloor \delta(i+1) \rfloor}{2}}\right].$$

As in the previous case, we can obtain a new lower bound by replacing the exponent for p_2 with a larger expression that is linear in i. Using the same method as above, this leads to a lower bound on the second summation of

$$m p_1^{2-\delta} p_2^{3u-\frac{\delta(2u-3)+\delta^2}{2}} (s-2t+2u-\ell+2). \tag{19}$$

Similarly, the third summation can be bounded below by the same methods by

$$p_1 p_2^{2u}(s - 2t + 2u - \ell + 2). \tag{20}$$

By piecing together (17)–(19), we obtain

$$W(s,u,\ell) \geq p_1 p_2^{2u-1}[s - t + u - \ell + 2] \left[1 - (t-u) m p_1^{1-\delta} p_2^{\frac{2u - \delta(2u-3) - \delta^2 + 2}{2}}\right]. \tag{21}$$

∎

Proposition 3: Given a uniform distribution of correct data features among the spurious and given the previously derived [10] expression for the binary probability of consistency

$$p_2 = \left(\frac{\kappa}{m}\right)^2$$

the expected amount of search is bounded by

$$\begin{aligned} W(s) \leq & t\frac{1}{\delta} + \frac{mp_1}{\delta}\left[t^{j_0+1}\mu^{j_0-1}\left(1 + (t-1)\frac{\kappa^2}{m^2}\right)\right. \\ & + \gamma^{i_0} f(t-1)\left(\frac{1-p_2^t}{1-p_2} + \beta\frac{1-p_2^{(3-\delta)t}}{1-p_2^{(3-\delta)}}\right) \\ & - \gamma^{i_0}[(d-1)(t-1) + f]\left(\left[\frac{p_2(1-p_2^t)}{(1-p_2)^2} - \frac{tp_2^t}{1-p_2}\right]\right. \\ & \left.\left. + \beta\left[\frac{p_2^{(3-\delta)}(1-p_2^{(3-\delta)t})}{(1-p_2^{3-\delta})^2} - \frac{tp_2^{(3-\delta)t}}{1-p_2^{3-\delta}}\right]\right)\right] \end{aligned}$$

where

$$\begin{aligned} \mu &= \frac{\kappa^2}{m} \\ \gamma &= (s-3)\mu \\ i_0 &= \lfloor (s-3)\mu - 1 \rfloor \\ j_0 &= \lfloor \kappa^2 - 1 \rfloor \\ \beta &= m p_1^{1-\delta} p_2^{\frac{2-\delta^2+\delta}{2}} \\ f &= s - t - \frac{1}{2}\left(\frac{1}{\delta} + 1\right) \\ d &= \frac{1}{\delta} - 1 \end{aligned}$$

and by

$$\begin{aligned} W(s) \geq & \frac{t}{\delta} + mp_1\left[a\frac{1-p_2^{2t}}{1-p_2^2} - b\frac{p_2^2(1-p_2^{2t})}{(1-p_2^2)^2} + \frac{btp_2^{2t}}{1-p_2^2}\right. \\ & \left. + \alpha\left(a\frac{1-p_2^{(3-\delta)t}}{1-p_2^{(3-\delta)}} - b\frac{p_2^{(3-\delta)}(1-p_2^{(3-\delta)t})}{(1-p_2^{(3-\delta)})^2} + \frac{btp_2^{(3-\delta)t}}{1-p_2^{(3-\delta)}}\right)\right] \end{aligned}$$

where

$$\begin{aligned} a &= (s-t+2)\left(\frac{1}{\delta} - \frac{1}{2}\right) - \frac{1}{2}\frac{1}{\delta}\left(\frac{1}{\delta} - 1\right) \\ b &= \left(\frac{1}{\delta} - 1\right)\left(\frac{1}{\delta} - \frac{1}{2}\right) \\ \alpha &= m p_1^{1-\delta} p_2^{\frac{3\delta - \delta^2 + 2}{2}}. \end{aligned}$$

Proof: The previous claim gives us a lower bound on the expected search of a particular subtree. How do we use it to bound the search of the whole tree? Under the assumption that the c correct data features are uniformly interspersed throughout the full set of s data features, we can see that at the top level of the tree, we must search m subtrees, with $u = 0, \ell = 1$, that is, for each possible assignment of the first data feature to a real model feature, we must explore the appropriate subtree. Since the first data feature is not part of the true object, once we have exhausted these subtrees, we must move on to interpretations that exclude the first data point by considering the portion of the tree below the node that pairs the first data point to the null character. Under this node, we consider pairings of the second data feature. Again, we must consider m subtrees, with $u = 0, \ell = 2$. We continue this process until we reach level $\ell = \frac{1}{\delta}$. In this case, we have a data feature that does have a correct match, and on average, this will be found after we have searched $\frac{m}{2}$ subtrees at this level. We then repeat this process below this node in the tree, with $u = 1$, and so on. Hence, the expected total amount of search is given by

$$W(s) = \sum_{j=0}^{t-1}\left(\left[\sum_{i=1}^{\frac{1}{\delta}-1} mW(s,j,\frac{1}{\delta}j + i) + 1\right] + \frac{m}{2}W(s,j,\frac{1}{\delta}(j+1)) + 1\right).$$

To derive a lower bound on this expression, we can substitute from (20) and execute the summation over i to obtain

$$W(s) \geq t\left(\frac{1}{\delta}\right) + mp_1 \sum_{j=0}^{t-1}\sum_{i=1}^{\frac{1}{\delta}-1} p_2^{2j}[a - bj]\left[1 + \alpha(t-j)p_2^{(1-\delta)j}\right]$$

where

$$\begin{aligned} a &= (s-t+2)\left(\frac{1}{\delta} - \frac{1}{2}\right) - \frac{1}{2\delta}\left(\frac{1}{\delta} - 1\right) \\ b &= \left(\frac{1}{\delta} - 1\right)\left(\frac{1}{\delta} - \frac{1}{2}\right) \\ \alpha &= m p_1^{1-\delta} p_2^{\frac{3\delta - \delta^2 + 2}{2}}. \end{aligned}$$

To further simplify this expression, we can use the identity for the geometric progression

$$\sum_{i=1}^{t-1} q^i = \frac{q^t - q}{q - 1}$$

and a derivative of this to yield

$$\sum_{i=1}^{t-1} iq^i = \frac{q(1-q^t)}{(1-q)^2} - \frac{tq^t}{1-q}$$

to get

$$W(s) \geq \frac{t}{\delta} + mp_1\left[a\frac{1-p_2^{2t}}{1-p_2^2} - b\frac{p_2^2(1-p_2^{2t})}{(1-p_2^2)^2} + \frac{btp_2^{2t}}{1-p_2^2}\right.$$

$$+\alpha\left(a\frac{1-p_2^{(3-\delta)t}}{1-p_2^{(3-\delta)}}-b\frac{p_2^{(3-\delta)}(1-p_2^{(3-\delta)t})}{(1-p_2^{(3-\delta)})^2}+\frac{btp_2^{(3-\delta)t}}{1-p_2^{(3-\delta)}}\right)\Bigg]. \quad (22)$$

We can also derive an upper bound on the total search involved by considering

$$W(s)\le\sum_{j=0}^{t-1}\sum_{i=1}^{\frac{1}{\delta}} mW(s,j,\frac{1}{\delta}j+i)+1$$
$$=t\frac{1}{\delta}+m\sum_{j=0}^{t-1}\sum_{i=1}^{\frac{1}{\delta}}W(s,j,\frac{1}{\delta}j+i).$$

We can substitute from (14) and reduce the summation over i by bounding terms from above to yield

$$W(s)\le t\frac{1}{\delta}+\frac{mp_1}{\delta}\sum_{j=0}^{t-1}p_2^j\Big[(t-j)^{j_0(j)+1}\mu^{j_0(j)-1}$$
$$+\left[s-t-\frac{1}{2}-\frac{1}{2\delta}-j\left(\frac{1}{\delta}-1\right)\right][t-j-1]\times$$
$$\times\left(1+\beta p_2^{(2-\delta)j}\right)\left[\left(s-\frac{1}{\delta}j-3\right)\mu\right]^{i_0(j,s-\frac{1}{\delta}j-2)}\Big]$$

where

$$\beta=mp_1^{1-\delta}p_2^{\frac{2-\delta^2+\delta}{2}}.$$

To reduce this, we note from our previous analysis that

$$j_0(j)\le\lfloor\kappa^2-1\rfloor$$

and similarly

$$i_o(j,s-\frac{1}{\delta}j-2)\le\lfloor(s-3)\mu-1\rfloor.$$

This yields

$$W(s)\le t\frac{1}{\delta}+\frac{mp_1}{\delta}\Bigg[\sum_{j=0}^{t-1}(t-j)^{j_0+1}\mu^{j_0-1}p_2^j$$
$$+\sum_{j=0}^{t-1}p_2^j\left(1+\beta p_2^{(2-\delta)j}\right)(f-dj)(t-j-1)\gamma^{i_0}\Bigg]$$

where

$$\gamma=(s-3)\mu$$
$$f=\left(s-t-\frac{1}{2}\left(\frac{1}{\delta}+1\right)\right)$$
$$d=\left(\frac{1}{\delta}-1\right)$$
$$i_0=\lfloor(s-3)\mu-1\rfloor$$
$$j_0=\lfloor\kappa^2-1\rfloor.$$

We can reduce the remaining summations by expanding out the first term and then bounding each remaining term in the summation by the largest term, which in this case is the second term. Using the results from above on the geometric progression and its derivatives, this leads to

$$W(s)\le t\frac{1}{\delta}+\frac{mp_1}{\delta}\Bigg[t^{j_0+1}\mu^{j_0-1}\left(1+(t-1)\frac{\kappa^2}{m^2}\right)$$
$$+\gamma^{i_0}f(t-1)\left(\frac{1-p_2^t}{1-p_2}+\beta\frac{1-p_2^{(3-\delta)t}}{1-p_2^{(3-\delta)}}\right)$$
$$-\gamma^{i_0}[(d-1)(t-1)+f]\left(\left[\frac{p_2(1-p_2^t)}{(1-p_2)^2}-\frac{tp_2^t}{1-p_2}\right]\right.$$
$$\left.+\beta\left[\frac{p_2^{(3-\delta)}(1-p_2^{(3-\delta)t})}{(1-p_2^{3-\delta})^2}-\frac{tp_2^{(3-\delta)t}}{1-p_2^{3-\delta}}\right]\right)\Bigg]. \quad (23)$$

∎

Proposition 4: If the sensory data arising from a correct interpretation are uniformly distributed among the spurious data, then the amount of search expended by the normal constrained search method is bounded by

$$m\frac{s}{c}2^c\le$$
$$W_{occ}\le m\frac{s}{c}2^c+\frac{m}{\epsilon}[1+\epsilon]^s\left[1+\frac{p_2}{1+\epsilon}\right]^{c-1}+\frac{m^3s}{\kappa^2}\frac{s}{c}[1+p_2]^c.$$

Proof: In [10], we showed that the number of nodes at the kth level of the tree is bounded by

$$2^{c(k)}\le n_k\le 2^{c(k)}$$
$$+\left[1+mp_1p_2^{\frac{1}{2}}\right]^{k-c(k)}\left[1+p_2+mp_1p_2^{\frac{1}{2}}\right]^{c(k)}$$
$$+mp_1\left[1-p_2^{\frac{c(k)+1}{2}}\right][1+p_2]^{c(k)-1}[k+p_2(k-c(k))]$$

where $c(k)$ is the number of data features actually part of the correct interpretation. Using our earlier assumption that

$$c(k)=\delta k$$

we can estimate bounds on the amount of search in the normal case by considering

$$m\sum_{k=1}^{s-1}n_k\le m\sum_{k=1}^{s-1}2^{\delta k}$$
$$+\left[1+mp_1p_2^{\frac{1}{2}}\right]^{k-\delta k}\left[1+p_2+mp_1p_2^{\frac{1}{2}}\right]^{\delta k}$$
$$+mp_1\left[1-p_2^{\frac{\delta k+1}{2}}\right][1+p_2]^{\delta k-1}[k+p_2(k-\delta k)].$$

Consider the first term

$$m\sum_{k=1}^{s-1}s^{\delta k}.$$

Actually, if we are careful in our considerations, this sum is really

$$m\sum_{k=1}^{s-1}s^{\lfloor\delta k\rfloor}$$

and this reduces to

$$m\frac{1}{\delta}\sum_{k=1}^{c-1}2^k\le m\frac{s}{c}2^c.$$

Similarly, the second term is

$$m\sum_{k=1}^{s-1}[1+\epsilon]^{k-\lfloor\delta k\rfloor}[1+p_2+\epsilon]^{\lfloor\delta k\rfloor}$$

where

$$\epsilon = mp_1p_2^{\frac{1}{2}} = \kappa p_1.$$

This reduces to

$$m\sum_{k=1}^{c-1}\left(\sum_{j=1}^{\frac{1}{\delta}-1}[1+\epsilon]^{\frac{1}{\delta}k+j}\right)\left[1+\frac{p_2}{1+\epsilon}\right]^k$$

and by using the same trick of finding the maximal term in a sum, this reduces to

$$\frac{m}{\epsilon}[1+\epsilon]^s\left[1+\frac{p_2}{1+\epsilon}\right]^{c-1}.$$

A similar argument can be applied to the remaining term in the summation, yielding

$$\frac{mp_1 s}{p_2}\frac{s}{c}[1+p_2]^c.$$

Combining all three of these bounds together, we have

$$W_{occ} \le m\frac{s}{c}2^c + \frac{m}{\epsilon}[1+\epsilon]^s\left[1+\frac{p_2}{1+\epsilon}\right]^{c-1} + \frac{m^3 s}{\kappa^2}\frac{s}{c}[1+p_2]^c.$$

∎

ACKNOWLEDGMENT

The presentation of these results was considerably improved by comments from B. Horn, D. Huttenlocher, and T. Lozano-Pérez.

REFERENCES

[1] N. Ayache and O. D. Faugeras, "HYPER: A new approach for the recognition and positioning of two-dimensional objects," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. 8, no. 1, pp. 44–54, 1986.

[2] D. H. Ballard, "Generalizing the Hough transform to detect arbitrary patterns," *Patt. Recogn.*, vol. 13, no. 2, pp. 111–122, 1981.

[3] R. C. Bolles and R.A. Cain, "Recognizing and locating partially visible objects: The local feature focus method," *Int. J. Robotics Res.*, vol. 1, no. 3, pp. 57–82, 1982.

[4] M. Drumheller, "Mobile robot localization using sonar," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. 9, no. 2, pp. 325–332, 1987.

[5] O. D. Faugeras and M. Hebert, "The representation, recognition, and locating of 3-D objects," *Int. J. Robotics Res.*, vol. 5, no. 3, pp. 27–52, 1986.

[6] E. C. Freuder, "Synthesizing constraint expressions," *Comm. ACM*, vol. 21, no. 11, pp. 958–966, 1978.

[7] ——, "A sufficient condition for backtrack-free search," *J. ACM*, vol. 29, no. 1, pp. 24–32, 1982.

[8] J. Gaschnig, "Performance measurement and analysis of certain search algorithms," Ph. D. thesis, Carnegie-Mellon Univ., Comput. Sci. Dept., Pittsburgh, PA, 1979.

[9] P. C. Gaston and T. Lozano-Pérez, "Tactile recognition and localization using object models: The case of polyhedra on a plane," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. 6, no. 3, pp. 257–265, 1984.

[10] W. E. L. Grimson, "The combinatorics of object recognition in cluttered environments using constrained search," *Artificial Intell.*, vol. 44, nos. 1–2, pp. 121–165, 1990.

[11] ——, "The effect of indexing on the complexity of object recognition," in *Proc. Third Int. Conf. Comput. Vision* (Osaka, Japan), 1990, pp. 644–651.

[12] W. E. L. Grimson and D.P. Huttenlocher, "On the sensitivity of the Hough transform for object recognition," in *Proc. Second Int. Conf. Comput. Vision* (Tarpon Springs, FL), 1988, pp. 700–706.

[13] ——, ,"On the verification of hypothesized matches in model-based recognition," *IEEE Trans. Patt. Anal. Machine Intell.*, to be published.

[14] W. E. L. Grimson and T. Lozano-Pérez, "Model-based recognition and localization from sparse range or tactile data," *Int. J. Robotics Res.*, vol. 3, no. 3, pp. 3–35, 1984.

[15] ——, "Localizing overlapping parts by searching the interpretation tree," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. 9, no. 4, pp. 469–482, 1987.

[16] R. M. Haralick and G.L. Elliot, "Increasing tree search efficiency for constraint satisfaction problems," *Artificial Intell.*, vol. 14, pp. 263–313, 1980.

[17] R. M. Haralick and L.G. Shapiro, "The consistent labeling problem: Part 1," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. 1, no. 4, pp. 173–184, 1979.

[18] D. P. Huttenlocher and S. Ullman, "Object recognition using alignment," in *Proc. First Int. Conf. Comput. Vision* (London), 1987, pp. 102–111.

[19] D. P. Huttenlocher, "Three-dimensional recognition of solid objects from a two-dimensional image," M.I.T. Artificial Intell. Lab., TR 1045, 1989.

[20] D. W. Jacobs, "The use of grouping in visual object recognition," M.I.T. Artificial Intell. Lab., Memo 1023, 1988.

[21] D. G. Lowe, *Perceptual Organization and Visual Recognition*. Boston: Kluwer Academic, 1985.

[22] ——, "Three-dimensional object recognition from single two-dimensional images," *Artificial Intell.*, vol. 31, pp. 355–395, 1987.

[23] A. K. Mackworth, "Consistency in networks of constraints," *Artificial Intell.*, vol. 8, pp. 99–118, 1977.

[24] A. K. Mackworth and E.C. Freuder, "The complexity of some polynomial network consistency algorithms for constraint satisfaction problems," *Artificial Intell.*, vol. 25, pp. 65–74, 1985.

[25] U. Montanari, "Networks of constraints: Fundamental properties and applications to picture processing," *Inform. Sci.*, vol. 7, pp. 95–132, 1974.

[26] D. W. Murray, "Model-based recognition using 3D structure from motion," *Image Vision Comput.*, pp. 85–90, 1987.

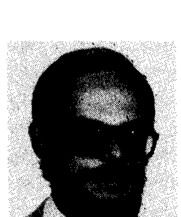
[27] ——, "Model-based recognition using 3D shape alone," *Comput. Vision Graphics Image Processing*, vol. 40, pp. 250–266, 1987.

[28] D. W. Murray and D.B. Cook, "Using the orientation of fragmentary 3D edge segments for polyhedral object recognition," *Int. J. Comput. Vision*, vol 2, no. 2, pp. 153–169, 1988.

[29] B. Nudel, "Consistent-labeling problems and their algorithms: Expected-complexities and theory-based heuristics," *Artificial Intell.*, vol. 21, pp. 135–178, 1983.

[30] A. Sha'ashua and S. Ullman, "Structural saliency: The detection of globally salient structures using a locally connected network," in *Proc. Second Int. Conf. Comput. Vision* (Tarpon Springs, FL), 1988, pp. 321–327.

[31] D. Waltz, "Understanding line drawings of scenes with shadows," in *The Psychology of Computer Vision* (P. Winston, Ed.). New York: McGraw Hill, 1975, pp. 19 – 91.



W. Eric L. Grimson (M'84) received the B.Sc. degree in mathematics and physics from the University of Regina, Canada, in 1975 and the Ph.D. degree in mathematics from the Massachusetts Institute of Technology (M. I. T.), Cambridge, in 1980.

He is currently an Associate Professor of Computer Science and Engineering at M. I. T. He has published research articles in machine vision, human vision, robotics, artificial intelligence, neural nets, and finite mathematics. He is the author of *From Images to Surfaces: A Computational Study of the Human Early Visual System* (MIT Press, 1981), *AI in the 1980's and Beyond: An MIT Survey* (MIT Press, 1986), and *Object Recogntion by Computer: The Role of Geometric Constraints* (MIT Press, 1990).

Dr. Grimson has served as an Associate Editor of IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE.