MASSACHUSETTS INSTITUTE OF TECHNOLOGY
ARTIFICIAL INTELLIGENCE LABORATORY

A.I. Memo No. 1226 April, 1990

The Effect of Indexing on the Complexity of Object Recognition

W. Eric L. Grimson

Abstract. Many current recognition systems use constrained search to locate objects in cluttered environments. Previous formal analysis 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. If one can group the data into subsets likely to have come from a single object, then terminating the search once a "good enough" interpretation is found reduces the expected search to cubic. Without successful grouping, terminated search is still exponential. These results apply to finding instances of a known object in the data. In this paper, we turn to the problem of selecting models from a library, and examine the combinatorics of determining that a candidate object is not present in the data. We show that the expected search is again exponential, implying that naive approaches to indexing are likely to carry an expensive overhead, since an exponential amount of work is needed to weed out each of the incorrect models. The analytic results are shown to be in agreement with empirical data for cluttered object recognition.

Acknowledgments: This report describes research done at the Artificial Intelligence Laboratory of the Massachusetts Institute of Technology. Support for the laboratory's artificial intelligence research is provided in part by an Office of Naval Research University Research Initiative grant under contract N00014-86-K-0685, and in part by the Advanced Research Projects Agency of the Department of Defense under Army contract number DACA76-85-C-0010 and under Office of Naval Research contract N00014-85-K-0124. The author is supported in part by the Matsushita Chair of Computer Science and Engineering, and by NSF contract number IRI-8900267.

©Massachusetts Institute of Technology 1989.

1 Preview of results and their implications.

This paper considers the problem of identifying and localizing an instance of a known object in noisy sensor data taken from a cluttered environment. Most current approaches to this problem utilize some type of search process, finding interpretations of the data by identifying pairings of data features to model features 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. Lowe 1985, 1987, Ayache & Faugeras 1986, Huttenlocher & Ullman 1987, Huttenlocher 1989], maximal clique methods [e.g. Bolles & Cain 1982] and constrained tree search methods [e.g. Grimson & Lozano-Pérez 1984, 1987, Gaston & Lozano-Pérez 1984, Murray 1987a, 1987b, Murray & Cook 1988, Drumheller 1987, Knapman 1987].

For all of these approaches, it is convenient conceptually to break the problem into three parts:

1. **Selection:** Given a set of data features, extract (possibly overlapping) subsets that are likely to have come from a single object.

2. **Indexing:** Given a library of possible objects, select a subset that are likely to be in the scene, perhaps as a function of the selected data subsets.

3. **Correspondence:** For each subset from the selection step, and for each corresponding object from the indexing step, determine if a match can be found between a subset of the data features and a subset of the model features, consistent with a rigid transformation of the object.

For the case of constrained tree search methods, previous work [Grimson 1989a, 1989b] has analyzed the complexity of different aspects of these problems. In particular, the following results have been established:

1. 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 quadratic in the parameters of the problem. This corresponds to the case in which both selection and indexing work perfectly.

2. If spurious data are allowed, the expected amount of search is bounded above and below by expressions exponential in the problem size. This corresponds to the case in which indexing works perfectly, but selection does not or is not used.

3. If the search is terminated once an interpretation that is "good enough" is found, then the expected amount of search is bounded below by an expression cubic in the problem parameters, and and above by an expression that is exponential, if the scene clutter is too large, but is quartic if the scene clutter is small enough. Note that a definition of what consistitutes "good enough" can be derived from first principles [Grimson & Huttenlocher 1989]. This corresponds to the case of perfect indexing and adequate, but not perfect, selection.

These results basically imply that in the case of constrained search if a selection process produces adequate (but not necessarily perfect) groupings of the data, then the complexity of the recognition process drops from exponential to low order polynomial.

All of these results have assumed that the indexing part of the problem has been solved, so that we are only seeking instances of objects that are known to be in the data. What happens when the indexing stage provides candidate objects that are not, in fact, present in the scene? For example, suppose we have L objects in our library. Naive approaches to indexing simply assume that we can sequentially test each library object for possible interpretations, keeping those model-data matches that are consistent, and discarding the others. Such approaches assume that the cost of deducing that a candidate object is not in the scene is no worse than the cost of identifying an instance of an object, and that both costs are low. While our earlier results show that finding correct interpretations can be done efficiently, it is not clear that the same cost applies to deducing that an object is not present, especially since the use of terminating search was essential to the reduction in complexity. In this article, we show that the expected amount of search needed to deduce that the object is not in fact present is exponential, even when termination of the search is allowed.

Although the actual amount of search is reduced when coupled with good selection (or grouping) mechanisms, the search remains exponential even in this case. This suggests that straightforward approaches to indexing (e.g. linear scanning of the library, or simple voting schemes) will not scale well with increases in library size, as the cost of searching large portions of the library will increase drastically with increase in library size. Hence, some care must be given to the indexing problem in scenarios involving large libraries.

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 perform several tests. First, we have compared the actual number of points that are theoretically searched against the order of growth bounds 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 incorrect indexing into a library of models carries an exponential cost, in the case of constrained serach problems.

2 The constrained search model.

To determine the expected cost of recognizing objects, we first establish the search framework to be used in solving the recognition problem. We then review results from earlier analysis of the constrained search method, before deriving new results on the role of indexing.

We begin by reviewing the constrained search method, used previously in [Grimson & Lozano-Pérez 1984, 1987, Gaston & Lozano-Pérez 1984, Murray 1987a, 1987b, Murray & Cook 1988, Drumheller 1987, Knapman 1987] 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 Figure 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.

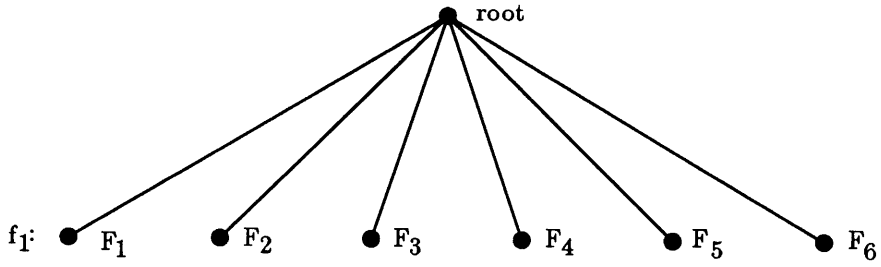

Figure 1: We can build a tree of possible interpretations, by first considering all the ways of matching the first data feature, f_1, to each of the model features, $F_j, j = 1, \ldots, m$.

This is shown in Figure 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.

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, as it simply reverts to an exploration of the entire, exponential-size, search space. A better solution is to explore the interpretation tree, 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, as 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

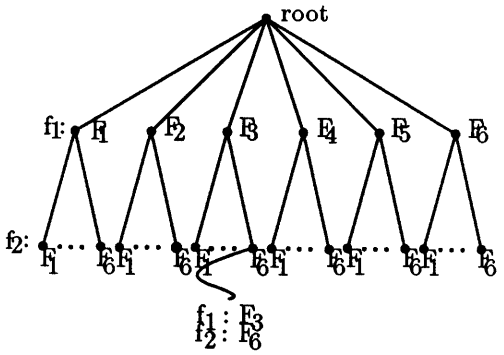

Figure 2: For each pairing of the first data feature with a model feature, we can consider matchings for the second data feature with each of the model features. Each node in the second level of the tree defines a pairing for the first two data points, found by tracing up the tree to the nodes. An example is shown.

such a transformation is generally computationally expensive (however, see [Faugeras & Hebert 1986, Ayache & Faugeras 1986] 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 together, 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 [Grimson & Lozano-Pérez 1984, 1987]. 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

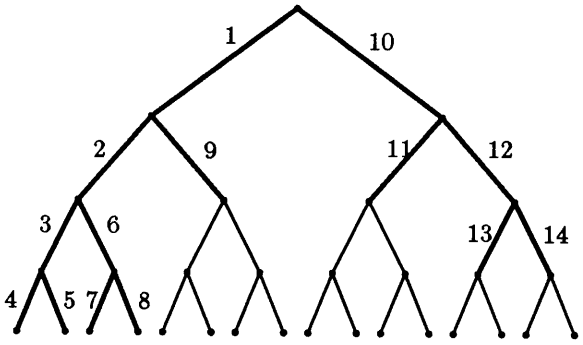

Figure 3: The tree is searched in a depth-first, backtracking manner, starting at the root. If a node is found to be inconsistent, the downward search is terminated, and we backtrack. Any leaf of the tree that is reached by the search constitutes a hypothesized interpretation. The darker edges in the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

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 as a problem of constraint satisfaction, or consistent labelling, a problem that has received considerable attention in the Artificial Intelligence literature [e.g. Freuder 1978, 1982, Gaschnig 1979, Haralick & Elliot 1980, Haralick & Shapiro 1979, Mackworth 1977, Mackworth & Freuder 1985, Montanari 1974, Nudel 1983, Waltz 1975]. 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 back-tracking depth-first search. (See Figure 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, 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 Figure 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.

3 Previous results

This method has been used for recognition in a variety of domains [Grimson & Lozano-Pérez 1984, 1987, Gaston & Lozano-Pérez 1984, Murray 1987a,

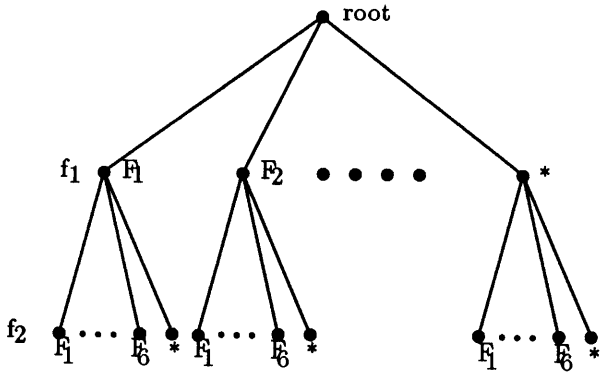

Figure 4: The interpretation tree can be extended by adding the null character $*$ as a final branch for each node of the tree. A match of a data feature and this character indicates that the data feature is not part of the current interpretation. In the example shown, the simple tree of Figure 2 has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

1987b, Murray & Cook 1988, Drumheller 1987]. Our empirical experience is that the method is 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 [Ballard 1981], 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 earlier combinatorial analyses [Grimson 1989a, 1989b], we showed that these empirical observations were supported by formal analysis. The main points of this analysis are summarized below.

1. When all of the data features are known to have come from a single object, the number of interpretations is generally asymptotic to 1.

2. 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

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

3. 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.

4. 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.

5. 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

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

and is bounded below by an expression of order

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

where γ, b, ϵ are constants that depend on the object characteristics and the amount of sensor noise, $\gamma, \epsilon < 1$.

6. If the search is terminated once an interpretation that is "good enough" (see [Grimson & Huttenlocher 1989] for a method for defining "good enough"), then the expected amount of search is bounded below by an expression of order

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

and above by an expression of order

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

where a, κ are small constants, and t is the threshold on the number of matched data features needed to terminate the search. This implies that if the scene clutter is small enough, i.e. selection has worked reasonably well:

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

then the search is basically cubic, while if the selection process is not sufficient, the expected search is still exponential.

As we suggested in the introduction, these results show that 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 fact, reducing the cost to cubic. All of this analysis, however, assumes that an instance of the model is, in fact, present in the data. Our concern in this paper is considering the cost of deducing that an hypothesized model is not present in the data. Empirical experience [Grimson, 1989c] has shown that this cost is considerably higher than that of identifying instances of objects in the data.

4 The formal model

We will derive results on the complexity of indexing 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 Propositions 1–3,

in which we show that the expected search is exponential 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 [Grimson 1989a], we let $q_{i,I}$ denote the probability that assigning the i^{th} data element to the I^{th} 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.

First, if there are s data features, m model features, of which $c_0 < t$ are consistent with a rigid translation of the model, and the threshold on termination of search is t, then the number of nodes searched is bounded below by:

$$\sum_{\ell=1}^{t}\sum_{r=0}^{\ell}\binom{\ell}{r}(m-1)^r\, p_1^{r-c(r,\ell)} p_2^{\binom{r}{2}-\binom{c(r,\ell)}{2}}$$
$$+\sum_{\ell=t+1}^{s-t}\sum_{r=0}^{t}\binom{\ell}{r}(m-1)^r\, p_1^{r-c(r,\ell)} p_2^{\binom{r}{2}-\binom{c(r,\ell)}{2}}$$
$$+\sum_{\ell=s-t+1}^{s}\sum_{r=\ell-s+t}^{t}\binom{\ell}{r}(m-1)^r\, p_1^{r-c(r,\ell)} p_2^{\binom{r}{2}-\binom{c(r,\ell)}{2}}. \qquad (1)$$

To see this, we note that for the first t levels of the tree, we must consider all possible interpretations. Hence, we can sum over the number of real matches (r) in the interpretation. For each different length of interpretation, we can choose up to $m-1$ different labels for the r matched data features, without including a match that is consistent with a rigid transformation. The probability of consistency of each such interpretation is given by the probability of unary consistency for the random feature assignments

$$p_1^{r-c(r,\ell)}$$

times the probability of binary consistency

$$p_2^{\binom{r}{2}-\binom{c(r,\ell)}{2}}.$$

Here, $c(r,\ell) \leq c_0$ counts the number of data-feature pairings that are actually consistent, as a function of the level of the tree and the number of features not matched to the wild card. For levels of the tree between t and $s-t$, we need only consider interpretations of length at most t, since any longer interpretation would previously have resulted in an interpretation of sufficient length to terminate the search. Finally, for levels of the tree between $s-t$ and s, we need only consider interpretations of sufficient length such that continuing downward in the search might possibly lead to an interpretation of length t.

In the appendix we show that the following lower bound on equation (1):

Proposition 1: If the c_0 data features (out of a total of s data features) consistent with a model with m features are uniformly distributed with

density $\delta = \frac{c_0}{s}$, then the expected amount of search for the case of an incorrect object model is bounded below by

$$p_1 p_2^{-1} \left[\left(\frac{s-t+1}{t+1} \right) [1+\nu]^t - 1 \right]$$

where

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

and where p_1 is the probability of unary consistency and p_2 is the probability of binary consistency, and where t is the threshold on the number of model features in a match sufficient to terminate the search.∎

A simpler version, under the assumption of uniformly distributed data is given by the following corollary.

Corollary 1.1: If $s \gg t$ and the data are uniformly distributed in transform space, then the lower bound on the expected search is roughly

$$\frac{m^2}{\kappa^2} \left[\left(\frac{s}{t} \right) \left[1 + \kappa \left(\frac{\kappa}{m} \right)^{t-2} \right]^t - 1 \right] \tag{2}$$

where κ is a small constant.∎

The main implication of this result is that using these search methods to deduce that a candidate object from the library is not in the data is expected to be an exponential search. Note that typically t is some fraction of m, the number of model features, so that the power of the exponent is considerably reduced from the straightforward British Museum algorithm's search. In fact, previous analysis has shown that one can define the threshold t as a function of the model characteristics, the noise in the system and the number of data and model features [Grimson and Huttenlocher, 1989]. In the limiting case of large numbers of features, t is a linear function of both s and m.

Note that the role of selection is intertwined with the role of indexing in this analysis. Good selection methods will reduce the size of s, and hence both the size of the largest exponential term, and the power t. On the other hand, using indexing with no selection will result in a larger cost for deducing that a candidate object is not present.

Since the expected search is bounded below by an exponential, we expect it to also be bounded above by one, a result we establish below.

To get an upper bound we use

$$\sum_{\ell=1}^{t}\sum_{r=0}^{\ell}\binom{\ell}{r}m^r p_1^{r-c(r,\ell)}p_2^{\binom{r}{2}-\binom{c(r,\ell)}{2}}$$
$$+\sum_{\ell=t+1}^{s-t}\sum_{r=0}^{t}\binom{\ell}{r}m^r p_1^{r-c(r,\ell)}p_2^{\binom{r}{2}-\binom{c(r,\ell)}{2}}$$
$$+\sum_{\ell=s-t+1}^{s}\sum_{r=\ell-s+t}^{t}\binom{\ell}{r}m^r p_1^{r-c(r,\ell)}p_2^{\binom{r}{2}-\binom{c(r,\ell)}{2}}. \qquad (3)$$

Using this, we derive the following result.

Proposition 2: If the c_0 data features (out of a total of s data features) consistent with a model with m features are uniformly distributed with density $\delta = \frac{c_0}{s}$, then the expected amount of search for the case of an incorrect object model is bounded above by

$$\frac{[1+\beta]^{t+1}-[1+\beta]}{\beta}+\binom{s}{t}\frac{s+1}{t}[1+\beta]^t\left(1-\left(1+\frac{t}{s-t+1}\right)^{t-s}\right)$$

where

$$\beta = mp_1^{1-\delta}p_2^{\frac{\delta(1-\delta)}{2}}$$

and where p_1 is the probability of unary consistency and p_2 is the probability of binary consistency, and where t is the threshold on the number of model features in a match sufficient to terminate the search.∎

Corollary 2.1: If $s \gg t$ and the data are uniformly distributed in transform space, then the upper bound on the expected search is roughly

$$\binom{s}{t}\frac{s}{t}m^t.\blacksquare \qquad (4)$$

Proofs of these results are found in the appendix.

The previous two propositions dealt with bounds on the expected search, where the data actually consistent with the model are uniformly distributed among the spurious data. More absolute bounds, without this assumption, can also be derived. In the case of lower bounds, we simply set $\delta = 0$ to handle the worst case distribution. For the upper bound, we need to use $\delta = \min\{1, \frac{c_0}{r}\}$ in a similar derivation to get the worst case distribution.

5 Implications of the results

The main conclusion from the above analysis, of course, is that incorrectly extracting candidate models from a library to match a set of sensor data is costly. While we have established this for the case of constrained search approaches to recognition, it is likely to hold for other approaches as well. While in some sense this is an obvious conclusion, it is important to establish formal bounds on the complexity of discarding incorrect models in a recognition task. Our results demonstrate that this cost is exponential, while our earlier results have shown that correct models can be identified in data in low order (cubic) polynomial time, if one has adequate selection methods available, and one terminates search once a "good" interpretation is found.

Corollary 1.1., which establishes a rough lower bound on the expected search, has an exponential whose power is the threshold t and whose base is $1 + \epsilon$ where ϵ is generally a small number. Since the threshold generally depends linearly on s [Grimson and Huttenlocher 1989], this bound will be reduced is indexing is coupled to selection, that is, if we can reduce the effective number of data points that are considered, we can reduce the necessary threshold, and hence the lower bound on the expected search. At the same time, Corollary 2.1 will also be reduced with a reduction in s, and hence, also improves when indexing is coupled with selection. Although the expected cost in rejecting an incorrect model is still exponential in this case, the reduction in the size of that cost may still be important for practical recognition systems.

5.1 Consistency of the formal results

Since we have made a number of assumptions in deriving our bounds on indexing complexity, it is important to obtain independent verifcation of the consistency of the derived results. We have done this in two ways. First, we have computed the actual combinatorial sums of equations (1) and (3), which

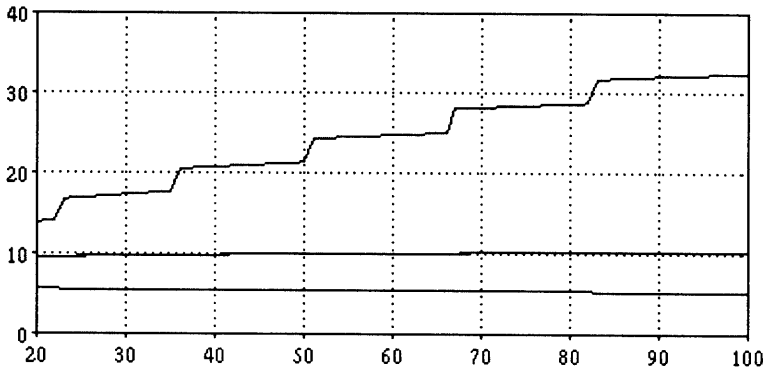

Figure 5: A graph of the log of the number of nodes searched, for fixed error and number of sensory points, as the number of model points increases. The bottom graph shows the lower bound of Proposition 1, the upper graph shows the upper bound of Proposition 2, and the middle graph is actually two graphs of the sums of equations (1) and (3), which on this scale are indistinguishable.

count the number of nodes searched, and compared them with the bounds of the two propositions, for a variety of values for the problem parameters. We find that in all cases, the lower and upper bounds on the expected search do, in fact, bound the actual sums. In general, the actual sums are closer to the lower bound of Proposition 1 than to the upper bound of Proposition 2. We graph some representative examples in Figures 5 and 6. In Figure 5, we keep the error and the number of sensory data features fixed, and vary the number of model features. In Figure 6, we keep the error and the number of model features fixed, and vary the number of data features.

To further demonstrate the relevance of the results derived here, we also compare the predictions of the analysis with data obtained from real examples. In particular, we selected a set of representative cluttered images, all of which excluded an instance of a known object, and extracted a set of features from the image. We then applied the RAF [Grimson & Lozano-Pérez 1984, 1987] recognition system to the resulting data. The threshold on terminating the search was set automatically using the analysis of [Grimson & Huttenlocher, 1989]. We counted the actual number of nodes searched in each case, and compared them to the predictions of the analysis presented here. In Figure 7, we plot the predicted number of nodes searched, the

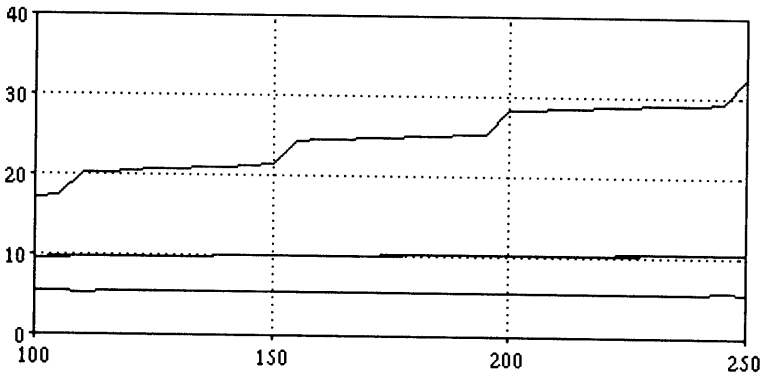

Figure 6: A graph of the log of the number of nodes searched, for fixed error and number of model points, as the number of sensory data points increases. The bottom graph shows the lower bound of Proposition 1, the upper graph shows the upper bound of Proposition 2, and the middle graph is actually two graphs of the sums of equations (1) and (3), which on this scale are indistinguishable.

derived bounds on that number, and the observed number of nodes searched, 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. We find that the actual search is smaller than the numbers predicted by equations (1) and (3), and lies close to the lower bounds of Proposition 1. This in part reflects the fact that while the analysis is based on models with equal length edges, and on a uniform distribution of edges in the images, the actual model had edges of varying lengths, and the image bedges were not necessarily uniformly distributed. Nonetheless, as indicated in Figure 7, the recorded search on real data is in reasonable agreement with the predictions of the formal analysis.

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

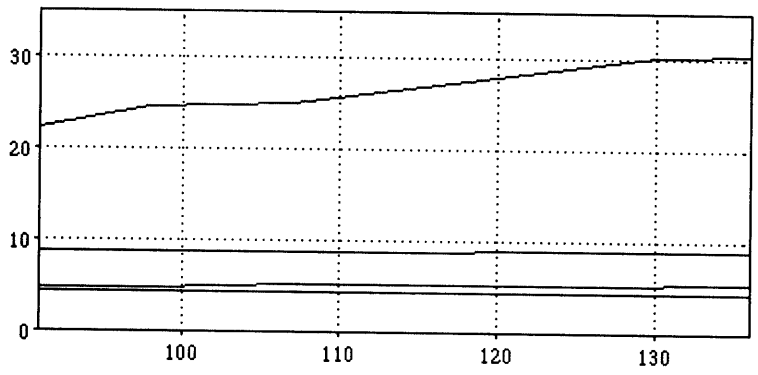

Figure 7: A graph of the log of the number of nodes searched, based on data from real images, as a function of the number of sensor features. The bottom graph shows the lower bound of Proposition 1, the upper graph shows the upper bound of Proposition 2. The graph second from the bottom is the actual number of nodes searched, while the graph second from the top is actually two graphs of the sums of equations (1) and (3), which on this scale are indistinguishable.

6 Conclusion

As a consequence, the main conclusion we can draw is that the cost of rejecting a candidate model from a library is exponential, at least for the class of recognition algorithms based on constrained search. That cost is reduced when indexing is coupled with selection methods, but remains exponential even in this case. In contrast, correctly identifying an instance of a model, when coupled with selection methods, is cubic in the size of the problem parameters. This implies that simple indexing methods will not scale well with increases in the size of the library, and that some effort must be given to finding efficient ways of selecting candidate library models that are highly likely to be consistent with selected subsets of the sensory data.

7 Acknowledgements

The presentation of these results was considerably improved by comments from Tomás Lozano-Pérez and Daniel Huttenlocher.

8 Appendix

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

We begin with a result from earlier analysis [Grimson, 1989a] that is of use in deriving the new results. (Note that the number of the proposition refers to the number used in that article.)

In particular, to obtain order of magnitude expressions on the amount of search required to find the 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 [Grimson, 1989a]: 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

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

in the worst case, and

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

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 1, 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.

$P/D =$	.125	.25	.5	1	2	4	8
$\epsilon_p^* = .01$	.002	.004	.008	.016	.033	.065	.131
$\epsilon_p^* = .1$	.021	.042	.085	.169	.338	.677	1.354
$\epsilon_p^* = .5$	.111	.222	.443	.886	1.772	3.545	7.090

Table 1: Values for the constant 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 $\epsilon_a = \tan^{-1} 2\epsilon_p^*$.

We now present the proofs of the propositions from the text.

First, if there are s data features, m model features, of which $c_0 < t$ are consistent with a rigid translation of the model, and the threshold on termination of search is t, then the number of nodes searched is bounded below by:

$$\begin{aligned} & \sum_{\ell=1}^{t} \sum_{r=0}^{\ell} \binom{\ell}{r} (m-1)^r \, p_1^{r-c(r,\ell)} p_2^{\binom{r}{2}-\binom{c(r,\ell)}{2}} \\ + & \sum_{\ell=t+1}^{s-t} \sum_{r=0}^{t} \binom{\ell}{r} (m-1)^r \, p_1^{r-c(r,\ell)} p_2^{\binom{r}{2}-\binom{c(r,\ell)}{2}} \\ + & \sum_{\ell=s-t+1}^{s} \sum_{r=\ell-s+t}^{t} \binom{\ell}{r} (m-1)^r \, p_1^{r-c(r,\ell)} p_2^{\binom{r}{2}-\binom{c(r,\ell)}{2}}. \end{aligned} \tag{5}$$

To see this, we note that for the first t levels of the tree, we must consider all possible interpretations. Hence, we can sum over the number of real matches (r) in the interpretation. For each different length of interpretation, we can choose up to $m-1$ different labels for the r matched data features, without including a match that is consistent with a rigid transformation. The probability of consistency of each such interpretation is given by the probability of unary consistency for the random feature assignments

$$p_1^{r-c(r,\ell)}$$

times the probability of binary consistency

$$p_2^{\binom{r}{2}-\binom{c(r,\ell)}{2}}.$$

Here, $c(r,\ell) \leq c_0$ counts the number of data-feature pairings that are actually consistent, as a function of the level of the tree and the number of

features not matched to the wild card. For levels of the tree between t and $s - t$, we need only consider interpretations of length at most t, since any longer interpretation would previously have resulted in an interpretation of sufficient length to terminate the search. Finally, for levels of the tree between $s - t$ and s, we need only consider interpretations of sufficient length such that continuing downward in the search might possibly lead to an interpretation of length t.

Since we are mostly concerned with expected complexity, we will focus on the case in which the consistent data is uniformly distributed among the spurious. In this case, we will assume that

$$c(r, \ell) = \lfloor \delta r \rfloor$$

where

$$\delta = \frac{c_0}{s}$$

is the density of consistent data features. Note that we can assume $c_0 < t$ since otherwise we would have a false positive response from our recognition system, and we have assumed that the threshold t has been set sufficiently high to prevent this.

We will first establish the following result:

Proposition 1: If the c_0 data features (out of a total of s data features) consistent with a model with m features are uniformly distributed with density $\delta = \frac{c_0}{s}$, then the expected amount of search for the case of an incorrect object model is bounded below by

$$p_1 p_2^{-1} \left[\left(\frac{s - t + 1}{t + 1} \right) [1 + \nu]^t - 1 \right]$$

where

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

and where p_1 is the probability of unary consistency and p_2 is the probability of binary consistency, and where t is the threshold on the number of model features in a match sufficient to terminate the search.∎

Proof: We begin by simplifying the summations in equation (5), using our assumption about $c(r, \ell)$:

$$\sum_{\ell=1}^{t}\sum_{r=0}^{\ell}\binom{\ell}{r}(m-1)^r\, p_1^{r-\lfloor \delta r\rfloor} p_2^{\binom{r}{2}-\binom{\lfloor \delta r\rfloor}{2}}$$
$$+\ \sum_{\ell=t+1}^{s-t}\sum_{r=0}^{t}\binom{\ell}{r}(m-1)^r\, p_1^{r-\lfloor \delta r\rfloor} p_2^{\binom{r}{2}-\binom{\lfloor \delta r\rfloor}{2}}$$
$$+\ \sum_{\ell=s-t+1}^{s}\sum_{r=\ell-s+t}^{t}\binom{\ell}{r}(m-1)^r\, p_1^{r-\lfloor \delta r\rfloor} p_2^{\binom{r}{2}-\binom{\lfloor \delta r\rfloor}{2}}. \qquad (6)$$

Consider the first summation in equation (6). We can simplify it by observing that

$$x-1 \le \lfloor x\rfloor \le x$$

so that this sum is bounded below by

$$\sum_{\ell=1}^{t}\sum_{r=0}^{\ell}\binom{\ell}{r}(m-1)^r\, p_1^{r-\delta r+1} p_2^{\binom{r}{2}-\binom{\delta r-1}{2}}$$

We can expand out the exponent for the p_2 term:

$$\binom{r}{2}-\binom{\delta r-1}{2} = \frac{r^2-r-(\delta r)^2+3\delta r-2}{2}$$

and since $p_2 < 1$, we can replace its exponent with a larger exponent so that the first summation in equation (6) is bounded below by:

$$p_1 p_2^{-1}\sum_{\ell=1}^{t}\sum_{r=0}^{\ell}\binom{\ell}{r}(m-1)^r\, p_1^{r(1-\delta)} p_2^{\left\lceil \frac{(1-\delta^2)\ell+3\delta-1}{2}\right\rceil r}$$

A similar reduction can be performed on the other two summations in equation (6).

We let

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

Since $\ell \le t$ in the first sum, we have as a lower bound for the summation parts of equation (6):

$$\sum_{\ell=1}^{t}\sum_{r=0}^{\ell}\binom{\ell}{r}\nu^r + \sum_{\ell=t+1}^{s-t}\sum_{r=0}^{t}\binom{\ell}{r}\nu^r + \sum_{\ell=s-t+1}^{s}\sum_{r=\ell-s+1}^{s}\binom{\ell}{r}\nu^r. \qquad (7)$$

Since we are seeking a lower bound, we can drop the third summation in equation (7). We can use the following derivation on the second summation:

$$\begin{aligned}\sum_{\ell=t+1}^{s-t}\binom{\ell}{r} &= \sum_{\ell=0}^{s-t}\binom{\ell}{r} - \sum_{\ell=0}^{t}\binom{\ell}{r} \\ &= \binom{s-t+1}{r+1} - \sum_{\ell=0}^{t}\binom{\ell}{r}\end{aligned}$$

where we have used a standard combinatorial identity on the first term in the expansion [Graham, et al. 1989].

This reduces our lower bound to

$$\sum_{\ell=1}^{t}\sum_{r=0}^{\ell}\binom{\ell}{r}\nu^r + \sum_{r=0}^{t}\binom{s-t+1}{r+1}\nu^r - \sum_{\ell=0}^{t}\sum_{r=0}^{t}\binom{\ell}{r}\nu^r. \tag{8}$$

We consider the second term first. Expansion leads to:

$$\sum_{r=0}^{t}\binom{s-t+1}{r+1}\nu^r = \sum_{r=0}^{t}\frac{(s-t+1)(s-t)\ldots(t+1)}{(s-t-r)(s-t-r-1)\ldots(t-r+1)}\frac{t!}{(r+1)r!(t-r)!}\nu^r.$$

Since

$$\frac{a}{b} \le \frac{a-i}{b-i}$$

for positive i provided $b \le a$, we can bound the above sum with the following smaller expression:

$$\sum_{r=0}^{t}\frac{t+1}{r+1}\frac{(s-t+1)(s-t)\ldots(t+2)}{(s-t)(s-t-1)\ldots(t+1)}\binom{t}{r}\nu^r.$$

By cancelling out terms, and noting that the worst case for $(t+1)/(r+1) = 1$, this reduces to

$$\left(\frac{s-t+1}{t+1}\right)\sum_{r=0}^{t}\binom{t}{r}\nu^r$$

and Vandermonde's relation then reduces this to

$$\left(\frac{s-t+1}{t+1}\right)[1+\nu]^t. \tag{9}$$

Now we consider the remaining two terms in equation (8):

$$\sum_{\ell=1}^{t}\sum_{r=0}^{\ell}\binom{\ell}{r}\nu^r - \sum_{\ell=0}^{t}\sum_{r=0}^{t}\binom{\ell}{r}\nu^r.$$

Cancelling common terms leads to

$$-\sum_{\ell=1}^{t}\sum_{r=\ell+1}^{t}\binom{\ell}{r}\nu^r - \sum_{r=0}^{t}\binom{0}{r}\nu^r$$

and only the $r = 0$ term in the second summation survives, yielding -1.

Combining this with equation (9), and the constants that we have dropped while concentrating on the summation part of equation (6), yields the desired result.∎

Corollary 1.1: If $s \gg t$ and the data are uniformly distributed in transform space, then the lower bound on the expected search is roughly

$$\frac{m^2}{\kappa^2}\left[\binom{s}{t}\left[1+\kappa\left(\frac{\kappa}{m}\right)^{t-2}\right]^t - 1\right] \tag{10}$$

where κ is a small constant.∎

Proof: We can simply use Proposition 3 of [Grimson, 1989a] to replace

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

The remaining simplifications follow.∎

To get an upper bound we use

$$\begin{aligned}
&\sum_{\ell=1}^{t}\sum_{r=0}^{\ell}\binom{\ell}{r} m^r p_1^{r-c(r,\ell)} p_2^{\binom{r}{2}-\binom{c(r,\ell)}{2}} \\
+\ &\sum_{\ell=t+1}^{s-t}\sum_{r=0}^{t}\binom{\ell}{r} m^r p_1^{r-c(r,\ell)} p_2^{\binom{r}{2}-\binom{c(r,\ell)}{2}} \\
+\ &\sum_{\ell=s-t+1}^{s}\sum_{r=\ell-s+t}^{t}\binom{\ell}{r} m^r p_1^{r-c(r,\ell)} p_2^{\binom{r}{2}-\binom{c(r,\ell)}{2}}.
\end{aligned} \tag{11}$$

As in the lower bound case, since we are mostly concerned with expected complexity, we will focus on the case in which the consistent data is uniformly distributed among the spurious. As before, we will assume that

$$c(r,\ell) = \lfloor \delta r \rfloor$$

where

$$\delta = \frac{c_0}{s}$$

is the density of consistent data features. Note that we can assume $c_0 < t$ since otherwise we would have a false positive response from our recognition system, and we have assumed that the threshold t has been set sufficiently high to prevent this.

With this, we have the following characterization of the expected search:

$$\begin{aligned} & \sum_{\ell=1}^{t} \sum_{r=0}^{\ell} \binom{\ell}{r} m^r p_1^{r-\lfloor \delta r \rfloor} p_2^{\binom{r}{2}-\binom{\lfloor \delta r \rfloor}{2}} \\ + & \sum_{\ell=t+1}^{s-t} \sum_{r=0}^{t} \binom{\ell}{r} m^r p_1^{r-\lfloor \delta r \rfloor} p_2^{\binom{r}{2}-\binom{\lfloor \delta r \rfloor}{2}} \\ + & \sum_{\ell=s-t+1}^{s} \sum_{r=\ell-s+t}^{t} \binom{\ell}{r} m^r p_1^{r-\lfloor \delta r \rfloor} p_2^{\binom{r}{2}-\binom{\lfloor \delta r \rfloor}{2}}. \end{aligned} \tag{12}$$

Using this, we derive the following result.

Proposition 2: If the c_0 data features (out of a total of s data features) consistent with a model with m features are uniformly distributed with density $\delta = \frac{c_0}{s}$, then the expected amount of search for the case of an incorrect object model is bounded above by

$$\frac{[1+\beta]^{t+1} - [1+\beta]}{\beta} + \binom{s}{t} \frac{s+1}{t} [1+\beta]^t \left(1 - \left(1 + \frac{t}{s-t+1}\right)^{t-s}\right)$$

where

$$\beta = m p_1^{1-\delta} p_2^{\frac{\delta(1-\delta)}{2}}$$

and where p_1 is the probability of unary consistency and p_2 is the probability of binary consistency, and where t is the threshold on the number of model features in a match sufficient to terminate the search.∎

Proof: Similar to our proof of Proposition 1, we can replace $\lfloor \delta r \rfloor$ with δr in equation (12). In this case, we replace the exponent for p_2 with a smaller

expression linear in r, specifically $\frac{r(\delta-\delta^2)}{2}$. This leads to the following upper bound on equation (12).

$$\sum_{\ell=1}^{t}\sum_{r=0}^{\ell}\binom{\ell}{r}\beta^r + \sum_{\ell=t+1}^{s-t}\sum_{r=0}^{t}\binom{\ell}{r}\beta^r + \sum_{\ell=s-t+1}^{s}\sum_{r=\ell-s+t}^{t}\binom{\ell}{r}\beta^r. \tag{13}$$

We can bound the second and third term from above by combining them into

$$\sum_{\ell=1}^{t}\sum_{r=0}^{\ell}\binom{\ell}{r}\beta^r + \sum_{\ell=t+1}^{s}\sum_{r=0}^{t}\binom{\ell}{r}\beta^r. \tag{14}$$

The first term of equation (14) reduces to

$$\frac{[1+\beta]^{t+1}-[1+\beta]}{\beta} \tag{15}$$

by applying the binomial theorem and the reduction for geometric series.

We can expand out the second term in equation (14) as

$$\sum_{r=0}^{t}\binom{\ell}{r}\beta^r = \sum_{r=0}^{t}\frac{\ell(\ell-1)\ldots(t+1)t!}{(\ell-r)(\ell-r-1)\ldots(t-r+1)(t-r)!r!}\beta^r.$$

This can be bounded above by:

$$\sum_{r=0}^{t}\left(\frac{t+1}{t+1-r}\right)\left(\frac{t+2}{t+2-r}\right)\cdots\left(\frac{t+i}{t+i-r}\right)\left(\frac{t+i+1}{t+i-r+1}\right)^{\ell-t-i}\binom{t}{r}\beta^r$$

and the worst case for this is when $r=t$, yielding (together with the binomial theorem):

$$(t+1)\left(\frac{t+2}{2}\right)\cdots\left(\frac{t+i}{i}\right)\left(\frac{t+i+1}{i+1}\right)^{\ell-t-i}[1+\beta]^t. \tag{16}$$

Returning this to the second term in equation (14), we have an upper bound on that term of

$$\sum_{\ell=t+1}^{s}\frac{(t+i)!}{t!i!}\left(\frac{t+i+1}{i+1}\right)^{\ell-t-i}[1+\beta]^t.$$

Now the choice of i is arbitrary, that is the choice of the number of terms of the expansion to pull out is open, subject to $1 \le i \le s-t$. In fact, the best bound occurs for $i=s-t$, and substitution leads to

$$[1+\beta]^t\binom{s}{t}\sum_{\ell=1}^{s-t}\left(\frac{s+1}{s-t+1}\right)^{\ell+t-s}$$

and application of the geometric series formula leads to

$$[1+\beta]^t \binom{s}{t} \frac{s-t+1}{t} \left(\left(1 + \frac{t}{s-t+1}\right) - \left(1 + \frac{t}{s-t+1}\right)^{t+1-s} \right). \quad (17)$$

Combining equation (17) and equation (15), plus some simplification, completes the result.∎

Corollary 2.1: If $s \gg t$ and the data are uniformly distributed in transform space, then the upper bound on the expected search is roughly

$$\binom{s}{t} \left(\frac{s}{t}\right) m^t. \quad (18)$$

∎

Proof: We can simply use Proposition 3 of [Grimson, 1989a] to replace

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

The remaining simplifications follow.∎

9 References

Ayache, N. & O.D. Faugeras, 1986, "HYPER: A new approach for the recognition and positioning of two-dimensional objects," *IEEE Trans. Pat. Anal. Mach. Intel.*, **8**(1) , pp. 44–54.

Ballard, D.H., 1981, "Generalizing the Hough transform to detect arbitrary patterns," *Pattern Recogn.*, **13**(2) , pp. 111–122.

Bolles, R.C. & R.A. Cain, 1982, "Recognizing and locating partially visible objects: The Local Feature Focus Method," *Int. Journ. Robotics Res.*, **1**(3) , pp. 57–82.

Drumheller, M., 1987, "Mobile robot localization using sonar," *IEEE Trans. Pat. Anal. Mach. Intel.*, **9**(2) , pp. 325–332.

Faugeras, O.D. & M. Hebert, 1986, "The representation, recognition and locating of 3-D objects," *Int. J. Robotics Research*, **5**(3) , pp. 27–52.

Freuder, E.C., 1978, "Synthesizing constraint expressions," *Comm. of the ACM*, **21**(11) , pp. 958–966.

Freuder, E.C., 1982, "A sufficient condition for backtrack-free search," *J. ACM*, **29**(1) , pp. 24–32.

Gaschnig, J., 1979, Performance measurement and analysis of certain search algorithms, Ph. D. Thesis, Carnegie-Mellon University, Computer Science.

Gaston, P.C. & T. Lozano-Pérez, 1984, "Tactile recognition and localization using object models: The case of polyhedra on a plane," *IEEE Trans. Pat. Anal. Mach. Intel.*, **6**(3) , pp. 257–265.

Graham, R.L., D.E. Knuth & O. Patashnik, 1989, *Concrete Mathematics*, Reading, Mass., Addison-Wesley.

Grimson, W.E.L., 1989a, "The combinatorics of object recognition in cluttered environments using constrained search," *Artificial Intelligence*, to appear.

Grimson, W.E.L., 1989b, "The combinatorics of heuristic search termination for object recognition in cluttered environments," MIT Artificial Intelligence Laboratory Memo 1111.

Grimson, W.E.L., 1989c, "On the recognition of curved objects," *IEEE Trans. Pat. Anal. Mach. Intel.*, **11**(6) , pp. 632–643.

Grimson, W.E.L. & D.P. Huttenlocher, 1988, "On the sensitivity of the Hough transform for object recognition," *Second Intl. Conf. on Computer Vision*, Tarpon Springs, FL., pp. 700–706.

Grimson, W.E.L. & D.P. Huttenlocher, 1989, "On Choosing Thresholds for Terminating Search in Object Recognition," Memo 1110, M.I.T. Artificial Intelligence Laboratory, to appear.

Grimson, W.E.L. & T. Lozano-Pérez, 1984, "Model-based recognition and localization from sparse range or tactile data," *Int. Journ. Robotics Res.*, **3**(3) , pp. 3–35.

Grimson, W.E.L. & T. Lozano-Pérez, 1987, "Localizing overlapping parts by searching the interpretation tree," *IEEE Trans. Pat. Anal. Mach. Intel.*, **9**(4) , pp. 469–482.

Haralick, R.M. & G.L. Elliot, 1980, "Increasing tree search efficiency for constraint satisfaction problems," *Artificial Intelligence*, **14**, pp. 263–313.

Haralick, R.M. & L.G. Shapiro, 1979, "The consistent labeling problem: Part 1," *IEEE Trans. Pattern Anal. Machine Intell.*, **1**(4) , pp. 173–184.

Huttenlocher, D.P. & S. Ullman, 1987, "Object recognition using alignment," *Proc. First Intern. Conf. Comp. Vision*, London, pp. 102–111.

Huttenlocher, D.P., 1989, "Three-Dimensional Recognition of Solid Objects from a Two-Dimensional Image," Memo 1045, M.I.T. Artificial Intelligence Laboratory.

Jacobs, D.W., 1988, "The Use of Grouping in Visual Object Recognition," 1023, M.I.T. Artificial Intelligence Laboratory.

Knapman, J., 1987, "3D Model Identification from Stereo Data," *Proc. First Intern. Conf. Comp. Vision*, London, pp. 547–551.

Lowe, D.G., 1985, *Perceptual Organization and Visual Recognition*, Boston, Kluwer Academic Publishers.

Lowe, D.G., 1987, "Three-Dimensional Object Recognition from Single Two-Dimensional Images," *Artificial Intelligence*, **31**, pp. 355–395.

Mackworth, A.K., 1977, "Consistency in networks of constraints," *Artificial Intelligence*, **8**, pp. 99–118.

Mackworth, A.K. & E.C. Freuder, 1985, "The complexity of some polynomial network consistency algorithms for constraint satisfaction problems," *Artificial Intelligence*, **25**, pp. 65–74.

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

Murray, D.W., 1987a, "Model-based recognition using 3D structure from motion," *Image and Vision Computing*, pp. 85–90.

Murray, D.W., 1987b, "Model-based recognition using 3D shape alone," *Computer Vision, Graphics and Image Processing*, **40**, pp. 250–266.

Murray, D.W. & D.B. Cook, 1988, "Using the orientation of fragmentary 3D edge segments for polyhedral object recognition," *Intern. Journ. Computer Vision*, **2**(2) , pp. 153–169.

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

Sha'ashua, A. & S. Ullman, 1988, "Structural Saliency: The detection of globally salient structures using a locally connected network," *Second Intl. Conf. on Computer Vision*, Tarpon Springs, FL., pp. 321–327.

Waltz, D., 1975, "Understanding line drawings of scenes with shadows," *The Psychology of Computer Vision*, edited by P. Winston, New York, McGraw Hill, pp. 19 – 91.