# Localizing Overlapping Parts by Searching the Interpretation Tree

W. ERIC L. GRIMSON, MEMBER, IEEE, AND TOMÁS LOZANO-PÉREZ, MEMBER, IEEE

*Abstract*—**This paper discusses how local measurements of positions and surface normals may be used to identify and locate overlapping objects. The objects are modeled as polyhedra (or polygons) having up to six degrees of positional freedom relative to the sensors. The approach operates by examining all hypotheses about pairings between sensed data and object surfaces and efficiently discarding inconsistent ones by using local constraints on: distances between faces, angles between face normals, and angles (relative to the surface normals) of vectors between sensed points. The method described here is an extension of a method for recognition and localization of nonoverlapping parts previously described in [18] and [15].**

*Index Terms*—**Bin-of-parts, computer vision, consistent labeling, constraint satisfaction, object recognition.**

## I. Introduction

THE specific problem considered in this paper is how to locate a known object that may be occluded by other unknown objects, so that much of the sensory data does not arise from the object of interest (see Figs. 1–4). This is a *localization* task. Our goal is to determine the power of simple geometric constraints in reducing the amount of search required to perform this task. While many other kinds of information can be used in recognition, we focus exclusively on the geometric information available from a model. The approach described in this paper is an extension of a method for localization of nonoverlapping parts previously described in [18] and [15].

### A. The Data and the Model

We seek conclusions that are applicable to a wide range of sensor types; therefore, we make very few assumptions about the character of the sense data. We assume only that the sensory data can be processed to obtain the position and surface orientation of planar patches on the object. The measured positions are assumed to be within a known error volume and the measured surface orientations to be within a known error cone.

When the objects have only three degrees of positional freedom relative to the sensor (two translational and one rotational), the positions and surface normals need only be two-dimensional. When the objects have more than three degrees of positional freedom (up to three translational and three rotational), the position and orientation data must be three-dimensional.

We assume that the objects can be modeled as sets of planar faces. Only the individual plane equations and a polygon embedded in each face is required. The model faces do not have to be connected and the model does not have to be complete.

### B. Our Approach to Localization

We approach the localization problem as a search for a consistent matching between the measured surface patches and the surfaces of the known object model. The search proceeds in two steps:

*1) Generate Feasible Interpretations:* Interpretations consist of pairings of sensed patches with some surface on the object model. Interpretations in which the sensed data are inconsistent with local geometric constraints derived from the model are discarded.

*2) Model Test:* The feasible interpretations are tested for consistency with surface equations obtained from the object models. An interpretation is legal if it is possible to solve for a rotation and translation that would place each sensed patch on an object surface. The sensed patch must lie *inside* the object face, not just on the surface defined by the face equation.

We structure the search for consistent matches as the generation and exploration of an *interpretation tree* (IT) (see Fig. 5). That is, starting at a root node, we construct a tree in a depth first fashion, assigning measured patches to model faces. At the first level of the tree, we consider assigning the first measured patch to all possible faces; at the next level, we assign the second measured patch to all possible faces, and so on. The number of possible interpretations in this tree, given $s$ sensed patches and $n$ surfaces, is $n^s$. Therefore, it is not feasible to explore the entire search space in order to apply a model test to all interpretations.

Our algorithm exploits local geometric constraints to remove entire subtrees from consideration. In our case, we require that the distances and angles between all pairs of data elements be consistent with the distances and angles possible between their assigned model elements. In general, the constraints must be coordinate-frame inde-

Manuscript received May 20, 1985; revised February 13, 1987. Recommended for acceptance by S. W. Zucker. This paper describes research done at the Artificial Intelligence Laboratory of the Massachusetts Institute of Technology. The Laboratory's artificial intelligence research is supported in part by a Grant from the System Development Foundation and in part by the Advanced Research Projects Agency under Office of Naval Research Contracts N00014-80-C-0505 and N00014-82-K-0334.

The authors are with the M.I.T. Artificial Intelligence Laboratory, 545 Technology Square, Cambridge, MA 02139.

IEEE Log Number 8715070.

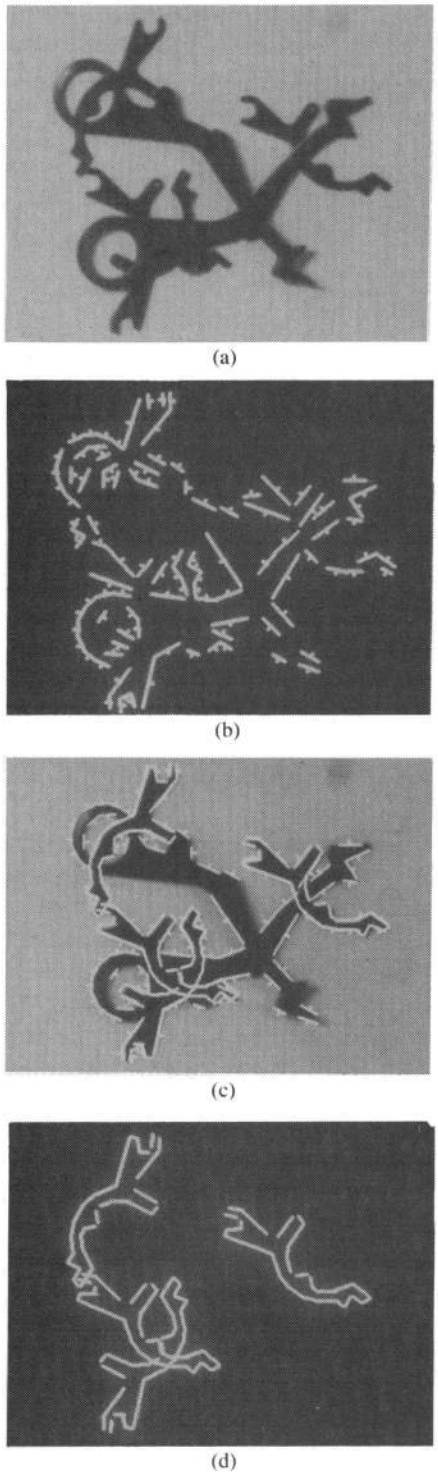



Fig. 1. Two-dimensional edge data. (a) Gray level images, (b) edge fragments, (c) located objects in image, and (d) located objects.

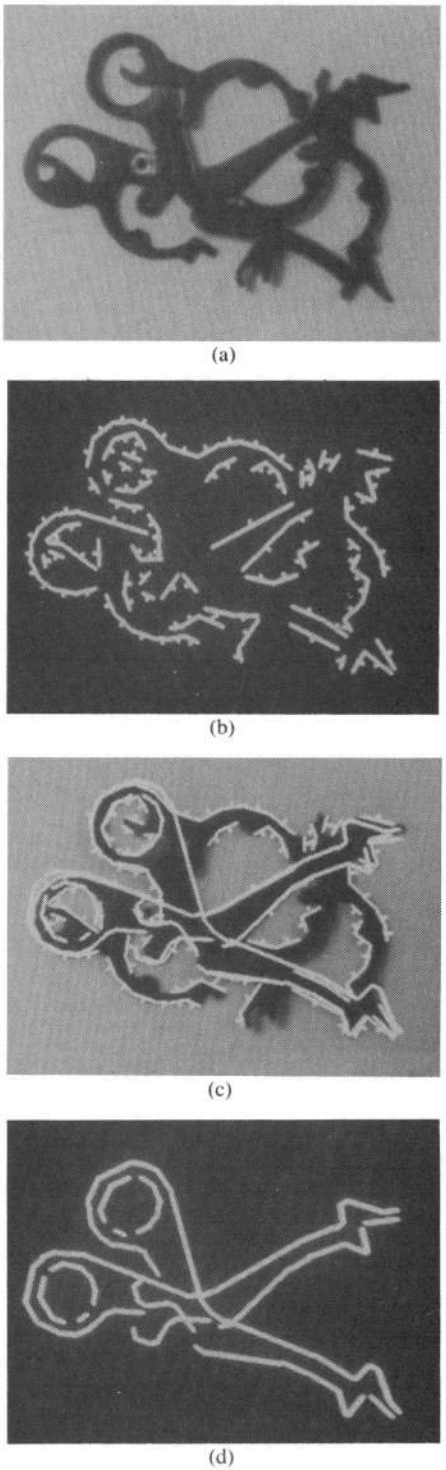


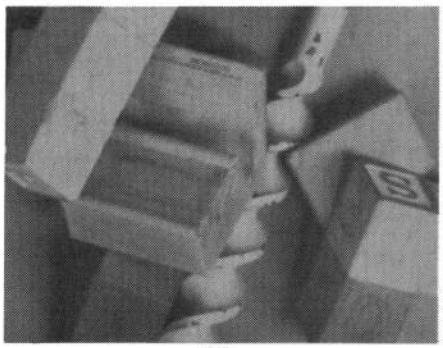

Fig. 2. Two-dimensional edge data. (a) Gray level images, (b) edge fragments, (c) located objects in image, and (d) located objects.

pendent, that is, the constraints should embody restrictions due to object shape and not to sensing geometry.

### C. Outline and Summary of Results

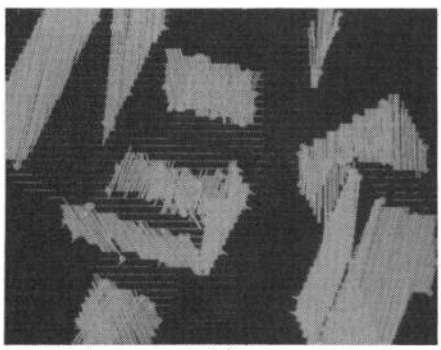
Our earlier papers have established the effectiveness of simple geometric constraints in eliminating large portions of the IT when all the data originates from a single object. We have shown that these constraints are powerful enough that sparse, point-like data can be used as the basis of reliable object localization. The main goal of this paper is to explore whether these conclusions still hold when the data stem from multiple objects.

We model the effect of extraneous data on the search space by adding a *null face* branch below each IT node. Assigning a data patch to this node is equivalent to discarding that patch as inconsistent with the model. The null face acts as a "wild card" in the match.

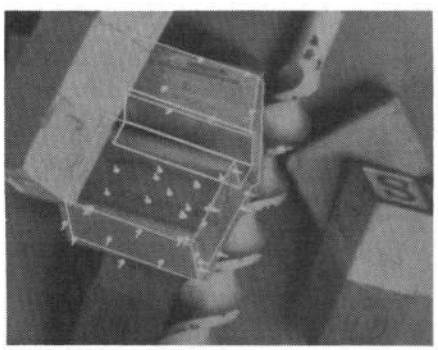


(a)

(b)

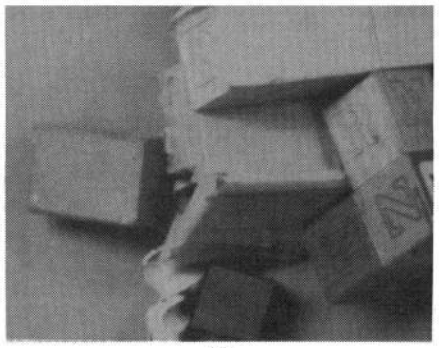


(c)

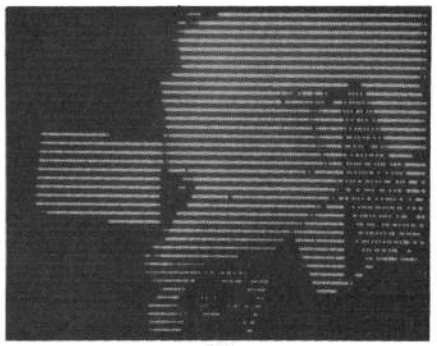


(d)

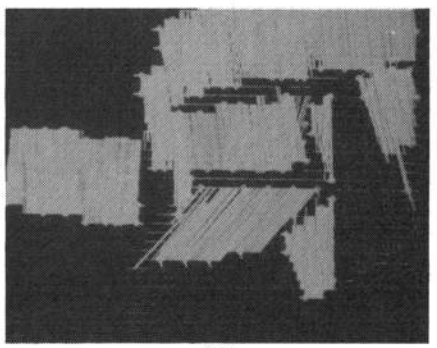
Fig. 3. Three-dimensional range data. (a) Original scene, (b) range where brightness encodes height, (c) planar patches with representative points, and (d) located object superimposed on image (filled in circles are data points accounted for).



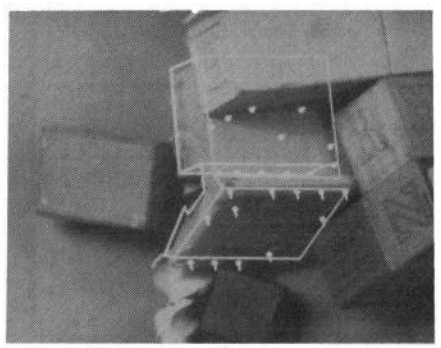
(a)



(b)



(c)



(d)

Fig. 4. Three-dimensional range data. (a) Original scene, (b) range where brightness encodes height, (c) planar patches with representative points, and (d) located object superimposed on image (filled in circles are data points accounted for).

We have found that the straightforward application of this null-face method to sparse, point-like data from overlapping objects has unacceptable performance. The algorithm finds all consistent interpretations, but the execution time is much too large. We have investigated a number of mechanisms of improving the performance to ascertain their relative effectiveness.

The first class of mechanisms we explored involved further constraints on the IT search:

*1) Heuristic Search Ordering:* Rather than attempting

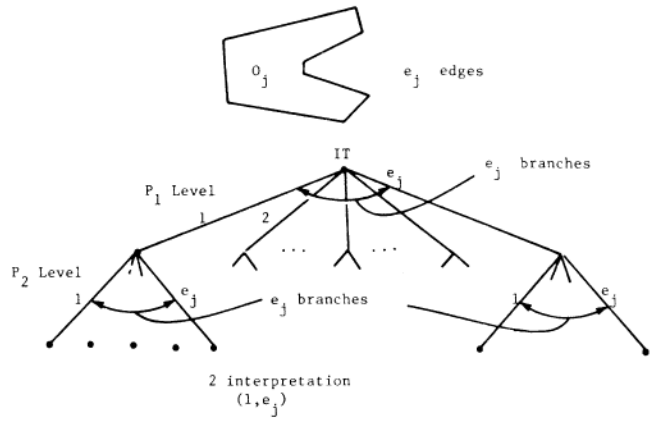



Fig. 5. The *interpretation tree*. A path through this tree represents a set of pairings of measured patches to model faces.

to generate all legal interpretations, the search can be heuristically guided towards a "good" interpretation based on a measure of "quality of match." When the measure reaches an acceptable level, the search is terminated; we call this *search cutoff*.

*2) Coupled Constraints:* The earlier set of geometric constraints considered only matches between pairs of data and model elements and did not propagate the effect of a legal match to matches below it in the IT. The constraints can be strengthened by recording the range of patch positions on the model faces consistent with a match and using this range as the initial range for the next match, and so on.

The second class of mechanisms we explored involved limiting the search space by initial segmentation and grouping of the sense data:

*3) Hough Clustering:* A coarse Hough transform can be used to cluster those sets of data/model matches that lead to a common range of hypothesized poses for the object.

*4) Preprocessing to Find Extended Linear Features:* Preprocessing reduces the number of independent sense data that need to be considered, thereby reducing the search space.

We expected that all of these mechanisms would have a significant impact on performance and that a combination of these mechanisms would be extremely powerful. What we found is that the mechanisms varied significantly in their contribution.

We found that, in contrast to our previous results for the case of isolated objects, preprocessing to find extended features is essential for the overlapping objects case. Having the extended features strengthens the geometric constraints; this helps reduce the effects of an increased search space due to the null face. We expected that the extended features would help; we did not expect that the effect would be so marked.

The use of heuristic search with cutoff proved to be essential in achieving realistic matching times. Without cutoff most of the matching time is spent in refining a legal interpretation. This justifies the use of a cutoff on the search together with a verification step.

Hough clustering has a smaller relative impact than extended features and search cutoff but can reduce matching times still further. The use of Hough clustering, however, can produce a significant number of matching errors.

The use of coupled constraints, surprisingly, proved to be completely ineffective; it did not reduce the search space significantly and it increased the matching time because of the additional overhead. Apparently, the simple decoupled constraints capture most of the necessary geometric information.

In the rest of the paper, we describe the extensions in more detail and document their performance via a number of simulations with two-dimensional data. We then report on a series of experiments with live data. Throughout the rest of the paper, we assume that the input data have been preprocessed to find extended planar features. A description of the preprocessing is included in our discussion of the experiments.

### D. Relation to Previous Work

The literature on object recognition stretches over a period of 20 years. An extensive (70 page) review of much of this literature can be found in [5]. In this section we will simply treat the work most directly relevant to the subject of this paper.

A number of authors have taken a similar view to ours that recognition can be structured as an explicit search for a match between data elements and model elements [2], [3], [7], [8], [12], [16], [21], [26]. Of these, the work of Bolles and his colleagues, Faugeras and his colleagues, and that of Baird are closest to the approach presented here.

The Feature-Focus method developed by Bolles and his colleagues solves the matching problem by solving a maximal clique problem in a matching graph. To reduce the combinatorics, the algorithm uses angle and distance constraints between the features. The method does not exploit the full range of constraints explored here nor does it place as much emphasis on them.

The method developed by Faugeras and Hebert [11] is also structured as a search over possible matches using an angle constraint to prune subsets of the search space. Their search, however, is structured around maximizing the quality of fit between the model and the data.

The interesting method developed by Baird [3] transforms the potential match between a model element and a data element into a constraint in the space of placement parameters of the object. It uses a linear programming algorithm to find the volume of consistent placements. The main advantage of this method is that it leads to provable bounds on the asymptotic performance of the algorithm.

The algorithms developed by Goad [16] and Lowe [21] are the only ones of the methods mentioned above that can locate three-dimensional objects on the basis of two-dimensional data (the location of edges in the image). They both use a combination of search and hypothesis verification.

The interpretation tree approach is an instance of the consistent labeling problem that has been studied extensively in computer vision and artificial intelligence [28], [25], [23], [13], [14], [20], [19], [24]. This paper can be viewed as suggesting a particular consistency relation (the constraints on distances and angles) and exploring its performance in a wide variety of circumstances. An alternative approach to the solution of consistent labeling problems is the use of relaxation. A number of authors have investigated this approach to object recognition [9], [6], [1]. These techniques are more suitable for implementation on parallel machines.

For a review of Hough clustering and its applications see [4]. A representative example of other recognition techniques using the Hough can be found in [27].

## II. Heuristic Search Ordering with Cutoff

As we mentioned in Section I-C, our approach to handling extraneous data from unknown objects is to add one more branch to each node of the interpretation tree, IT. This branch represents the possibility of discarding the sensed patch as extraneous. Call this branch the *null face*. The search proceeds, as before, to explore the IT depth first. As each new assignment of a data patch to a model face is considered, the new interpretation thus formed is tested to see whether it satisfies the geometric constraints. In these tests, the null face behaves as a "wild card"; assigning a patch to the null face will never cause the failure of an interpretation.

Clearly, if an interpretation is legal, all subsets of this interpretation are leaves of the expanded IT. This is true since every combination of legal assignments of the null face to the sensed patches will still produce a valid interpretation. Rather than generating all of these subsets, we want to generate the "best" interpretation. The problem then arises of choosing the quality measure. Reference [11] has explored the use of a measure based on how well the computed model transformation maps measured patches into model faces. We have chosen instead to search for interpretations where the data patches have the largest combined area. The reason for our choice is that this measure is simple and fairly insensitive to measurement error. The following simple search method guarantees that we find only the most complete interpretations.

The IT is explored in a depth-first fashion, with the null face considered last when expanding a node. Now, assume an external variable, call it *MAX*, that keeps track of the best (largest area) valid interpretation found so far. For a node at level $i$ in the tree, let $M$ denote the area of the data patches assigned to non-null faces in the partial match associated with that node. Let $R$ be the area of the data patches below this level of the tree: $P_{i+1}, \cdots, P_n$. It is only worth assigning a null face to patch $P_i$, if $M + R \geq MAX$. Otherwise, the area of the interpretations at all the leaves below this node will be less than that of the best interpretation already found. If we initialize *MAX* to some nonzero value, then only interpretations with area greater than this threshold will be found. As better interpretations are found, the value of *MAX* is incremented, thus ensuring that we find the most complete interpretation of the data. Note that if an interpretation of maximal area (no null-face assignments) is found, then no further null-face assignments will be considered after that point.

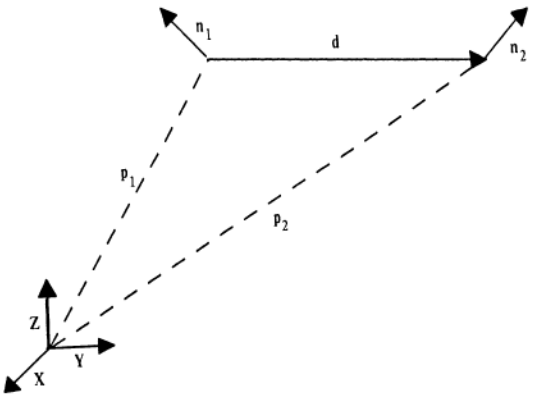



Fig. 6. The constraints between pairs of measured surface patches. A given pair of sensory points $P_1$, $P_2$ can be characterized by the components of the vector $\boldsymbol{d}$ between them, in the direction of each of the surface normals $\boldsymbol{n}_1$, $\boldsymbol{n}_2$ and in the direction of their cross product, $\boldsymbol{n}_1 \times \boldsymbol{n}_2$, and by the angle between the two normals $\boldsymbol{n}_1 \cdot \boldsymbol{n}_2$.

The search process described above can be continued until all the nodes have either been examined or discarded. This can take a very long time for realistic cases. We observed that the search located the correct interpretation fairly early on, but then spent a tremendous amount of time attempting to improve on it. This phenomenon can be avoided by the use of an area threshold (as a percentage of the model's area). The search is discontinued when an interpretation that exceeds that threshold passes the model test. We have found that this search cutoff drastically improves the execution time without adversely affecting the failure rate. Section V describes the simulations supporting this conclusion.

## III. The Constraints

In our earlier work, we did not propagate the cumulative effects of the constraints on the possible positions for the sense data on the model faces. We call these the *decoupled constraints*. The decoupling leads to very efficient implementations, with some loss of pruning power. In this paper we consider a stronger set of constraints that retains the coupling. This set is more powerful, but computationally more complex.

### A. The Decoupled Constraints

First construct a local coordinate frame relative to the sensed data; we use both unit normals as basis vectors. In two dimensions, these define a local system, except in the degenerate case of the unit normals being (anti-)parallel. In three dimensions, the third component of the local coordinate frame can be taken as the unit vector in the direction of the cross product of the normal vectors. In this frame, one set of coordinate-frame-independent measurements is: the components of the vector $\boldsymbol{d}$ along each of the basis directions and the angle between the two measured normals (see Fig. 6). More formally,

$$\boldsymbol{n}_1 \cdot \boldsymbol{n}_2$$
$$\boldsymbol{d} \cdot \boldsymbol{n}_1$$
$$\boldsymbol{d} \cdot \boldsymbol{n}_2$$
$$\boldsymbol{d} \cdot \boldsymbol{u}$$

where $\boldsymbol{u}$ is a unit vector in the direction of $\boldsymbol{n}_1 \times \boldsymbol{n}_2$.

These measurements are equivalent, but not identical to the set used in [18]. In the earlier paper, we used the magnitude of $\boldsymbol{d}$ and two of its components; this is equivalent, up to a possible sign ambiguity, to using the three components of the vector. This possible ambiguity in the earlier set of measurements was resolved using a triple product constraint.

For these measurements to constrain the search process, we must relate them to corresponding model measurements. Consider the first measurement, $\boldsymbol{n}_1 \cdot \boldsymbol{n}_2$. If this is to correspond to a measurement between two faces in the model, then the dot product of the model normals must agree with this measurement. If they do not agree, then no interpretation that assigns those patches to these model faces need be considered. In the interpretation tree, this corresponds to pruning the entire subtree below the node corresponding to that assignment. The test can be implemented efficiently by precomputing the dot product between all pairs of faces in the models. Of course, for the case of exact measurements, the dot product of the measured normals must be identical to that of the associated model normals. In practice, exact measurements are not possible, and we must take possible sensor errors into account. Given bounds on the error in a sensory measurement, we compute a range of possible values associated with the dot product of two sensed normals (see [18] for details).

Similar constraints can be derived for the components of the separation vector in the directions of the unit normals. Each pair of model faces defines an infinite set of possible separation vectors, each one having its head on one face and its tail in the other. We can compute bounds on the components of this set of vectors in the direction of each of the face normals. Again, for an assignment of sensed patches to model faces to be consistent, the measured value must agree with the precomputed model values. Here also we can use the error bounds to compute a range of possible values for the components of the sensed vectors; this range must be consistent with the associated model range.

It is important to realize that these constraints are not guaranteed to reject all impossible interpretations. Consider Fig. 7, for example. Consider matching point $P_i$ to face $f_u$, point $P_j$ to face $f_v$, and point $P_k$ to face $f_w$. These assignments are pairwise consistent, and the sections of the faces that are feasible locations for the sensed points are indicated by the sections labeled $ij$, etc. The assignment is not globally consistent, however, as indicated by the fact that the segments for face $f_u$ and $f_w$ do not overlap.

Because of this decoupling of the constraints, the fact that all pairs of patch-surface assignments are consistent

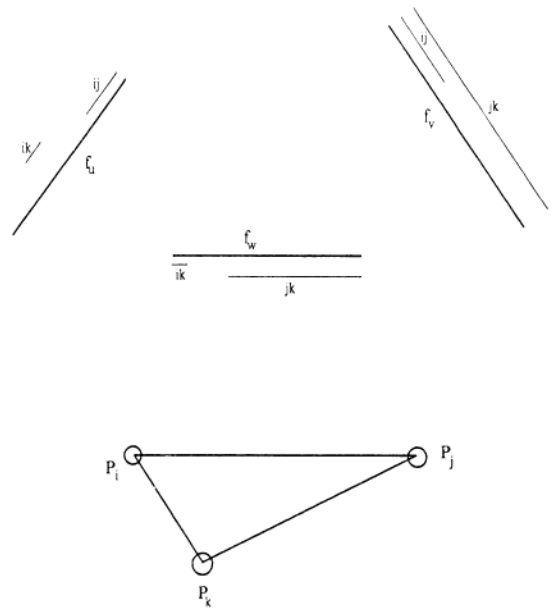



Fig. 7. The constraints are decoupled. Consider matching point $P_i$ to face $f_u$, point $P_j$ to face $f_v$, and point $P_k$ to face $f_w$. These assignments are pairwise consistent, and the sections of the faces that are feasible locations for the sensed points are indicated by the sections labeled $ij$, etc. The assignment is not globally consistent, however, as indicated by the fact that the segments for face $f_u$ and $f_w$ do not overlap.

does not imply that the global assignment is consistent. To determine global consistency, we solve for a transformation from model coordinates to sensor coordinates that maps each of the sensed patches to the interior of the appropriate face. There are many methods to solve for the transformation; one is described in [18], another can be found in [11]. This model test is applied to interpretations surviving pruning so as to guarantee that all the available geometric constraint is satisfied. As a side-effect, the model test also provides a solution to the localization problem.

### B. The Coupled Constraints

It is possible to find constraints that maintain global consistency without requiring an explicit model transformation. One such set of constraints is developed below, first for the two-dimensional case, and then extended to three dimensions.

Consider two edges of an object, oriented arbitrarily in sensor coordinates, as shown in Fig. 8. With each edge we will associate a base point, defined by the vector $\boldsymbol{b}_i$, a unit tangent vector $\boldsymbol{t}_i$, which points along the edge from the base point, and a unit normal vector $\boldsymbol{n}_i$, which points outward from the edge. Thus, the position of a point $P_1$ along edge $f_i$ in this coordinate system is given by

$$\boldsymbol{p}_1 = \boldsymbol{b}_i + \alpha_1 \boldsymbol{t}_i \qquad \alpha_1 \in [0, l_i]$$

where $l_i$ is the length of the edge. Similarly, a point $P_2$ on face $f_j$ can be represented by

$$\boldsymbol{p}_2 = \boldsymbol{b}_j + \alpha_2 \boldsymbol{t}_j \qquad \alpha_2 \in [0, l_j].$$

If the patches $P_1$ and $P_2$ are edge segments of nonnegligible length, this has the effect of cutting down the le-

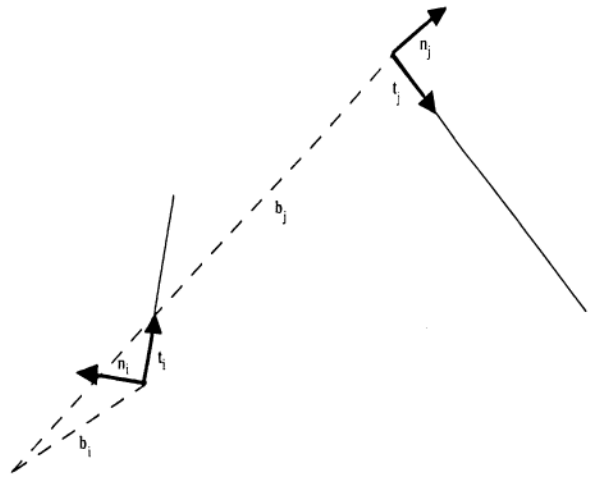



Fig. 8. The constraints are recoupled. With each face, we associate a base vector $\boldsymbol{b}_i$, a tangent vector $\boldsymbol{t}_i$, and a normal vector $\boldsymbol{n}_i$. Then any point on a face can be represented by $\boldsymbol{b}_i + \alpha \boldsymbol{t}_i$ for some $\alpha$ between 0 and the length of the edge.

gal range of $\alpha_1$ and $\alpha_2$ by the length of the edge. In the discussion below we assume that the patches are point-like and that points on either end of the edge segment are chosen to represent the edge. A more exact treatment of the edge segment case is quite analogous to the point case given below.

The vector between two small measured patches is given by

$$\boldsymbol{p}_1 - \boldsymbol{p}_2 = \boldsymbol{d}_{12} = \boldsymbol{b}_i + \alpha_1 \boldsymbol{t}_i - \boldsymbol{b}_j - \alpha_2 \boldsymbol{t}_j. \qquad (1)$$

We know that we can measure $\boldsymbol{d}_{12}$. Because of measurement error, however, the measured points $P_1$ and $P_2$ may not lie exactly on the object edges and as a consequence, what we can measure is

$$\boldsymbol{d}_{12}^* = \boldsymbol{b}_i + \alpha_1 \boldsymbol{t}_i + \boldsymbol{u}_1 - \boldsymbol{b}_j - \alpha_2 \boldsymbol{t}_j - \boldsymbol{u}_2$$

where $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$ are measurement errors whose size can be bounded. We can also measure the surface normal at the point $P_1$, say $\boldsymbol{n}_i^*$, which in the case of perfect data would equal $\boldsymbol{n}_i$. In general, we will only know that $\boldsymbol{n}_i^*$ is within some specified angle of $\boldsymbol{n}_i$.

We can compute

$$\boldsymbol{d}_{12}^* \cdot \boldsymbol{n}_i^* = m_{12}$$

based on our measurements. We know $m_{12}$ is an estimate of

$$\boldsymbol{d}_{12} \cdot \boldsymbol{n}_i.$$

We can compute bounds on the range of errors about the measured value so that we know that the true value of $\boldsymbol{d}_{12} \cdot \boldsymbol{n}_i$ lies in the range

$$\boldsymbol{d}_{12} \cdot \boldsymbol{n}_i \in [m_{12} - \epsilon, m_{12} + \epsilon]$$

where $\epsilon$ can be computed straightforwardly [18].

From (1) we have

$$\boldsymbol{d}_{12} \cdot \boldsymbol{n}_i = (\boldsymbol{b}_i - \boldsymbol{b}_j) \cdot \boldsymbol{n}_i - \alpha_2 (\boldsymbol{t}_j \cdot \boldsymbol{n}_i). \qquad (2)$$

The first term on the right is a constant and is a function of the object only, independent of its orientation. Thus, (2) provides us with a constraint on the value of $\alpha_2$. In particular, if $\boldsymbol{t}_j \cdot \boldsymbol{n}_i = 0$, then this assignment of patches to faces is consistent only if

$$(\boldsymbol{b}_i - \boldsymbol{b}_j) \cdot \boldsymbol{n}_i \in [m_{12} - \epsilon, m_{12} + \epsilon].$$

If this is true, then $\alpha_2$ can take on any value in its current range. If it is false, then the assignment of these patches $P_1$, $P_2$ to these faces $f_i$, $f_j$ is inconsistent and can be discarded.

In the more common case, when $\boldsymbol{t}_j \cdot \boldsymbol{n}_i \neq 0$, we have

$$\alpha_2 (\boldsymbol{t}_j \cdot \boldsymbol{n}_i) \in \big[(\boldsymbol{b}_i - \boldsymbol{b}_j) \cdot \boldsymbol{n}_i - m_{12} - \epsilon, \; (\boldsymbol{b}_i - \boldsymbol{b}_j) \cdot \boldsymbol{n}_i - m_{12} + \epsilon\big].$$

Thus, we have restricted the range of possible values for $\alpha_2$ and hence the set of positions for patch $P_2$ that are consistent with this interpretation.

Similarly, by using the estimates for $\boldsymbol{d}_{12} \cdot \boldsymbol{n}_j$ obtained from the measurements, we can restrict the range of values for $\alpha_1$ and, thereby, the position of $P_1$.

We can also consider the coordinate-frame-independent term

$$\boldsymbol{d}_{12} \cdot \boldsymbol{t}_i = (\boldsymbol{b}_i - \boldsymbol{b}_j) \cdot \boldsymbol{t}_i + \alpha_1 - \alpha_2 (\boldsymbol{t}_j \cdot \boldsymbol{t}_i). \qquad (3)$$

As before, we can place bounds on the measured value for $\boldsymbol{d}_{12} \cdot \boldsymbol{t}_i$ when error in the sensory data is incorporated. Then, given a legitimate range for $\alpha_1$ we can restrict the range of $\alpha_2$ and vice versa. A similar argument holds for $\boldsymbol{d}_{12} \cdot \boldsymbol{t}_j$.

These constraints allow us to compute intrinsic ranges for the possible assignments of patches to faces. The key to them is that we can propagate these ranges as we construct an interpretation. For example, suppose that we assign patch $P_1$ to face $f_i$. Initially, the range for $\alpha_1$ is

$$\alpha_1 \in [0, l_i].$$

We now assign patch $P_2$ to face $f_j$, with

$$\alpha_2 \in [0, l_j]$$

initially. By applying the constraints derived above, we can reduce the legitimate ranges for these first two patches to some smaller set of ranges. We now consider adding patch $P_3$ to face $f_k$. When we construct the range of legal values for $\alpha_3$, we find that the constraints are generally much tighter, since the legal ranges for $\alpha_1$ and $\alpha_2$ have already been reduced. Moreover, both $\alpha_1$ and $\alpha_2$ must be consistent with $\alpha_3$, so the legal range for this patch is given by the intersection of the ranges provided by the constraints. Finally, the refined range of consistent values for $\alpha_3$ may in turn reduce the legal ranges for $\alpha_1$ and $\alpha_2$ and these new ranges may then refine each other by another application of the constraints, and so on. In other words, the legal ranges for the assignment of patches to faces may be relaxed via the constraint equations, and in this manner, a globally consistent assignment is maintained. Of course, if any of the ranges for $\alpha_i$ becomes empty, the interpretation can be discarded as inconsistent without further exploration.

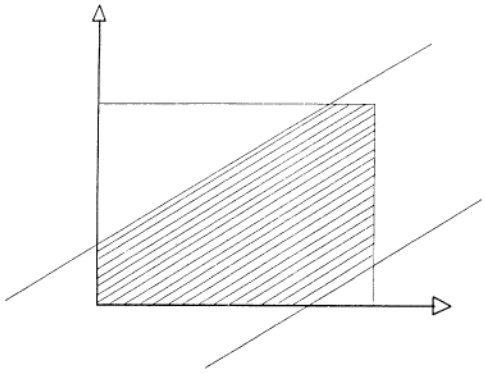


Fig. 9. The intersection of legitimate face ranges for three-dimensional data.

The constraints derived above for the two-dimensional case can be extended to three dimensions as well. In this case, we represent points on a face by

$$\boldsymbol{b}_i + \alpha \boldsymbol{u}_i + \beta \boldsymbol{v}_i$$

where $\boldsymbol{b}_i$ is a vector to a designated *base* vertex of the face, and $\boldsymbol{u}_i$ and $\boldsymbol{v}_i$ are orthonormal vectors lying in the plane of the face. Furthermore, $\alpha$ and $\beta$ are constrained to lie within some polygonal region, defined by the shape of the face. In the simplest case,

$$\alpha \in [0, \mathfrak{A}] \qquad \beta \in [0, \mathfrak{B}].$$

These constraints describe a region in a two-dimensional space spanned by $\alpha$ and $\beta$, as illustrated in Fig. 9. Given a current polygonal region of consistency for $\alpha$ and $\beta$, we can intersect the region with this new range, to obtain a tighter region of consistency, as shown in the figure. Similar to the two dimensional case, as additional sensed patches are considered, the constraints they generate may be propagated among one another. If any polygonal region corresponding to a sensed patch vanishes, the interpretation is inconsistent and the procedure can stop exploring that portion of the interpretation tree.

## IV. Hough Clustering

In addition to modifying the search algorithm, one can attempt to reduce the size of the initial interpretation tree. Using extended features does some of this. Another common technique is the Hough transform [4]. The method works as follows for two-dimensional data. We are given a set of measured edge fragments and a set of model edges. For each pair of model edge and data edge, there is a rotation $\theta$ of the model's coordinate system that aligns the model edge to the data edge. Then, there is a range of translations $x$, $y$ that displace the model so that the chosen model edge overlaps the chosen data edge. If the pairing of model edge and data edge is part of the correct interpretation of the data, then one of the range of combinations of $x$, $y$, $\theta$ obtained in this way will describe the correct transformation from model to sensor coordinates. All the model/data edge-pairings corresponding to that legal interpretation will also produce the correct $x$, $y$, $\theta$ combination (modulo measurement error). We keep track of the range of $x$, $y$, $\theta$ values produced by each model/data edge-pairing; this can be done with a three-dimensional array, with each dimension representing the quantized values of one of $x$, $y$, and $\theta$. Clusters of pairings with nearly the same values define candidate interpretations of the data (see Fig. 10).

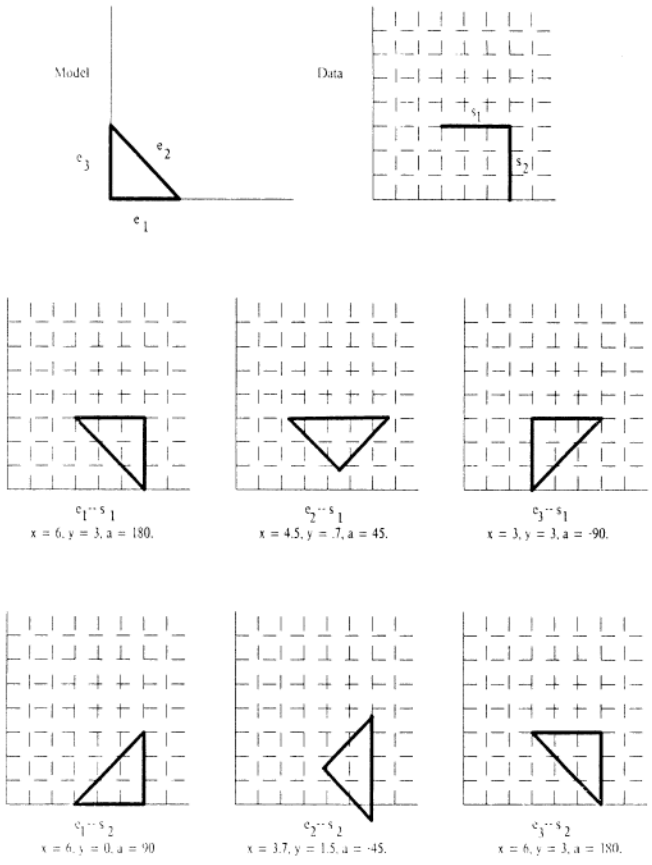



Fig. 10. Hough transform preprocessing establishes initial pairings between model edges and data edges.

This technique can serve as a recognition method all by itself [4], but in that context it has some important drawbacks. One problem is simply the growth in memory requirements as the degrees of freedom increase. A related but more important problem is the difficulty of characterizing the range of transformations that map the model into the data in three dimensions. Consider the case of mapping a planar model face into a measured planar patch. The orientation of the model coordinate system relative to the sensor coordinate system is specified by three independent parameters, but the constraint of making a model plane parallel to a data plane only provides two constraints (rotation around the plane normal is unconstrained). Therefore, each model/data pairing generates a one parameter family of rotations. Associated with each rotation, there is a range of displacements that manage to overlap the model face with the measured patch. Computing these ranges exactly is quite difficult and time consuming.

What we have done is to use the Hough transform as a coarse filter to produce an initial set of possible model/data pairings—not to localize the objects. First, each potential model/data pairing is used to define a range of parameter values related to the position and orientation of

the model relative to the sensor. These parameters, however, need not be the full set of parameters that define the coordinate transform between model and sensor. In two dimensions, for example, the parameter set may contain only the rotation angle or both the angle and the magnitude of the displacement vector, or the full set of angle and displacement parameters. In three dimensions, we can use, for example, the magnitude of the displacement vector and the two angles of the spherical coordinate representation of some fixed vector on the model. The model/data pairings are clustered on the basis of a coarse quantization of these parameters. Each cluster associates with each data edge a set of candidate model edges; this is precisely what defines the interpretation tree for our method.

The interpretation-tree method is then applied to the largest clusters until a valid interpretation is found. The effect of the initial clustering is to reduce the size of the search space at the expense of initial preprocessing. Typically, data edges in a cluster are associated with a subset of the model edges, thus cutting down the branching factor in the interpretation tree. Predictably, this has an impact on performance, but not as great as we had anticipated. Many of the pairings are still spurious, primarily due to noise and data from other objects. Therefore, it is still necessary to use the null-face technique described earlier. As we will see in the next section, there are enough spurious matches in the correct bucket to require extensive searching in the worst case. Therefore, the Hough clustering technique is relatively weak by itself, but very effective when combined with heuristic search with cutoff.

## V. Simulations

We have tested several variations of the algorithm with simulated two-dimensional data of the type illustrated in Fig. 11. This testing gives us a quantitative evaluation of the different mechanisms described above.

Each simulation trial took as input a total of five objects, chosen from a set of two different models. At least one instance of each model was present. The objects were placed in random orientation, and randomly translated within a window whose sides were on the order of the extent of the largest object.

The input data was obtained from this set of objects as follows: For each edge of each object, a random number between 0 and 1 was chosen. If the number was greater than 0.25, the edge was kept, otherwise it was ignored. For the chosen edges, a second random number between 0.5 and 1 was chosen. This variable determined the length of the edge segment to be constructed from this edge, with 1 denoting the full edge. The starting point for the edge segment was chosen arbitrarily along the given edge, subject to the condition that the chosen segment was completely contained within the original edge. Next, the position of the two endpoints of the edge segment were corrupted. For each endpoint, a direction was chosen at random, and a distance was chosen at random from the range $[0, d_0]$, where $d_0$ was a prespecified maximum deviation. The two endpoints of the edge segment were then corrupted by displacement of the chosen distance along the chosen direction. In the trials reported here, $d_0$ was ten pixels. This process was repeated for each edge of each of the 5 objects, creating a set of input edge fragments.

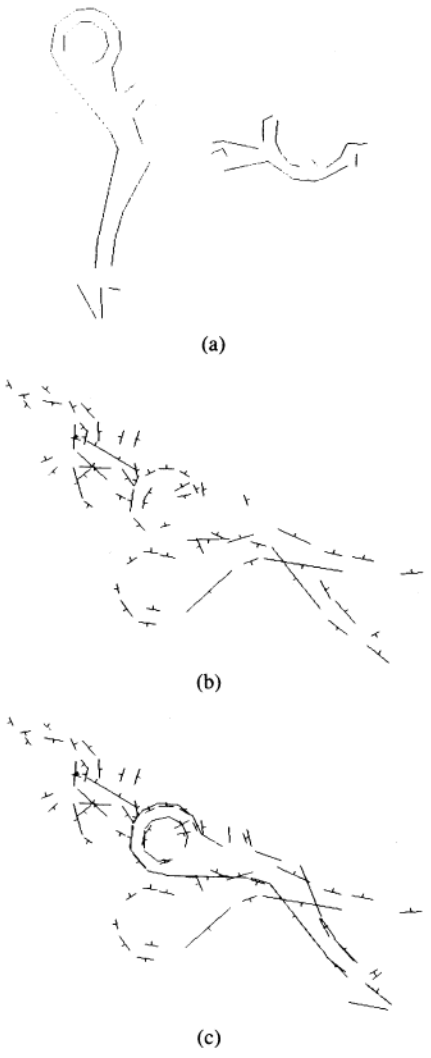



Fig. 11. Simulations of overlapping two dimensional parts. A collection of copies of objects selected from the set illustrated in (a) was overlapped at random, as illustrated in (b). Edge fragments were selected at random along the perimeter of the overlapping group, and corrupted with random error. The recognition and localization algorithm then searched for interpretations of the data consistent with a specific model, as shown in (c).

A set of 100 trials were run. For each set of data, a series of tests were performed using different combinations of mechanisms: search cutoff (based on perimeter matched) alone, search cutoff with coupled constraints, Hough, Hough with coupled constraints, Hough with search cutoff, Hough with search cutoff and coupled constraints. In each case, the recognition process was run until the first acceptable interpretation was found.

The labels on the tables below are:

- Nodes—Number of nodes of the search tree explored.

- Model—Number of model tests applied.
- Correct—Number of trials (out of 100) in which the correct answer was found.
- Dist—Error in computing the translation component.
- Angle—Error in computing the rotation component.
- Scale—Error in computing the scale component.
- Bucket Percent—The percentage of entries in the Hough bucket actually used in the interpretation.

The results for the trials using edge data are reported below, ordered from worst to best.

Hough Clustering without coupled constraints

| | Nodes | Model | Correct | Dist | Angle | Scale | Bucket Percent |
|---|---|---|---|---|---|---|---|
| Mean | 111123 | 1.6 | 97 | 3.15 | .0057 | .0074 | .65 |
| Med | 58920 | 1 | | 2.36 | .0043 | .0054 | .64 |

Hough Clustering with coupled constraints

| | Nodes | Model | Correct | Dist | Angle | Scale | Bucket Percent |
|---|---|---|---|---|---|---|---|
| Mean | 109493 | 1.5 | 97 | 2.99 | .0050 | .0070 | .65 |
| Med | 58683 | 1 | | 2.31 | .0040 | .0055 | .63 |

Search cutoff without coupled constraints

| Stat | Nodes | Model | Correct | Dist | Angle | Scale |
|---|---|---|---|---|---|---|
| Mean | 314.6 | 10.3 | 100 | 12.10 | .0138 | .0287 |
| Med | 23 | 1 | | 10.27 | .0093 | .0211 |

Search cutoff with coupled constraints

| Stat | Nodes | Model | Correct | Dist | Angle | Scale |
|---|---|---|---|---|---|---|
| Mean | 372.9 | 20.5 | 100 | 12.20 | .0144 | .0289 |
| Med | 21.5 | 1 | | 10.27 | .0098 | .0211 |

Hough Clustering with search cutoff and without coupled constraints

| | Nodes | Model | Correct | Dist | Angle | Scale | Bucket Percent |
|---|---|---|---|---|---|---|---|
| Mean | 12.3 | 1.5 | 97 | 10.44 | .0122 | .0261 | .21 |
| Med | 5 | 1 | | 7.80 | .0083 | .0187 | .19 |

Hough Clustering with search cutoff and coupled constraints

| | Nodes | Model | Correct | Dist | Angle | Scale | Bucket Percent |
|---|---|---|---|---|---|---|---|
| Mean | 11.1 | 1.4 | 97 | 10.44 | .0122 | .0261 | .21 |
| Med | 5 | 1 | | 7.80 | .0083 | .0187 | .19 |

For the case of point data, some of the same simulations were run. The difference is that in this case, the length of the edge fragment was fixed at 2 pixels in length. Two sets of test tests were performed using point data, one using heuristic search with cutoff and the other with Hough clustering with search cutoff. Neither of these tests used the coupled constraints since they have a relatively minor effect. The other methods are inadequate for point data; they typically explore millions of nodes.

Search cutoff without coupled constraints

| Stat | Nodes | Model | Correct | Dist | Angle | Scale |
|---|---|---|---|---|---|---|
| Mean | 141860 | 54.8 | 99 | 12.74 | .0549 | .0193 |
| Med | 97376 | 27 | | 5.18 | .0080 | .0090 |

Hough clustering with search cutoff and without coupled constraints

| | Nodes | Model | Correct | Dist | Angle | Scale | Bucket Percent |
|---|---|---|---|---|---|---|---|
| Mean | 11068 | 17.2 | 75 | 22.01 | .0776 | .0226 | .33 |
| Median | 2327 | 4 | | 7.05 | .0094 | .0135 | .33 |

Note that there was only one failure when only the search cutoff was used while there were 25 failures with the Hough clustering. On the other hand, the Hough method examined a factor of 10 fewer nodes.

A number of conclusions can be drawn from these data:

- The coupled constraints add very little to the pruning process. Not only do they not reduce the number of nodes; the coupled constraints significantly increase the computation at each node.
- Hough clustering helps reduce the search space, but the number of edges in a bucket is still significant. Less than two thirds of the pairings in a bucket are used in the actual interpretation. Therefore, there is a significant amount of search required to find the best legal interpretation. Note also that Hough clustering is the only technique that introduces a significant number of recognition failures.
- Heuristic search with cutoff is the single most effective way of reducing the search without increasing the failure rate. Much of the search in the original method is devoted to improving a match rather than finding an initial match. This helps explains the effectiveness of hypothesize-verify techniques.
- The hybrid technique employing extended features, Hough clustering and heuristic search with cutoff is several orders of magnitude more effective than the straightforward search technique even when prefaced by Hough clustering.

## VI. Experience with Live Data

The hybrid approach described above using extended linear features, heuristic search with cutoff, and optionally Hough clustering, has been tested in thousands of experiments with a variety of sensory data. This section summarizes our experience and discusses some of the special considerations we faced in the individual experiments.

### *A. Edge Fragments from Gray-Level Images*

In situations such as those illustrated in Figs. 1 and 2, we have used edge fragments from images obtained by a camera located directly overhead. The images are obtained under lighting from several overhead fluorescent lights. The camera is a standard vidicon located approximately five feet above the scene. The edge fragments are obtained by linking edge points marked as zero crossings in the Laplacian of Gaussian-smoothed images [22]. Edge points are marked only when the gradient at that point exceeds a predefined threshold; this is done to eliminate some shallow edges. The algorithm is applied to some predefined number of the longest edge fragments.

We can exploit our knowledge of the extent of the edge fragment to more tightly constrain the matching process. We do this by selecting the endpoints of the edge fragment as representative points. The matching algorithm is applied to these representative points and their corresponding normals. We require that both points be assigned to the same model face.

The most difficult problem faced in this application is that we cannot reliably tell which side of the edge contains the object, that is, the edge normals can be determined only up to a sign ambiguity. Although region brightness can sometimes be used to separate figure from

ground, it is not always reliable. The algorithm can be modified to keep track of the two possible assignments of sign and to guarantee that all the pairings in an interpretation have consistent assignments of sign. This approach, however, causes a noticeable degradation in the performance of the algorithm, since it reduces the pruning power of the constraints. Fortunately, we can use another form of the constraints to reduce the effect of this ambiguity.

As long as two edges do not cross or are not collinear, at least one edge must be completely within one of the half planes bounded by the other. This means that the components along one of the edge normals of all posible separation vectors will always have the same sign. Given a tentative pairing of two measured edge fragments and two model edges, we can use this property to pick the sign of one of the normals. The angle constraint between normals can then be used to consistently select the signs for other edges in that interpretation. Of course, the sign assignment is predicated on the initial pairing being correct, which it may not be, so we have lost some pruning power in any case.

We have also tested the algorithm in situations where the sign of the filtered image could be used to determine the edge normal reliably. The algorithm performs substantially better under these circumstances.

With or without the complete normal, the algorithm succeeds in locating the desired object in images where the edge data from any single object is very sparse (see Figs. 1 and 2). To test the reliability of the algorithm on real data, we ran the following set of tests. A carton containing a total of eight parts selected from three different types of parts (two types are shown in Figs. 1 and 2), was placed under a camera. The carton was arbitrarily perturbed to randomly orient and overlap the parts and the recognition process was then applied. This process was repeated 100 times, and in each case an instance of a selected object was correctly identified and located in the image. The number of nodes of the interpretation tree actually explored in solving this problem was found to vary by up to an order of magnitude, depending on the difficulty of the image, but in all cases a correct interpretation was found.

When the sign of the normal is unknown and without using the Hough preprocessing, difficult cases such as in Figs. 1 and 2 require a minute or more of matching time. In situations where the overlapping is slight, the matching time is closer to 30 seconds. This is almost twice as long as the performance of the algorithm on the same images when the sign of the normal is available. In this case, the typical matching time for lightly overlapped parts is around 10 to 15 seconds, with the worst-case times ranging from 30 seconds to minutes.

Using search cutoff and Hough preprocessing makes the recognition time nearly independent of the complexity of the scene. In our testing, we used the full set of $x$, $y$, $\theta$ parameters for clustering the model/data edge-pairings. The Hough preprocessing itself takes on the order of seven or eight seconds for 80 data edges and 30 model edges. The recognition time after that is only from two to four seconds. The total recognition time is usually around 10 seconds. This is slightly longer than the time required by simple cases without the Hough preprocessing, but an order of magnitude better than the time required for the worst cases.

### B. *Range Data from Structured Light*

We have also applied our algorithm to relatively dense range data obtained from a laser-striping system developed by Philippe Brou at our laboratory. The data used in our experiments (see Figs. 3 and 4) were taken at a resolution of 0.3 cm in the vertical and horizontal directions. The resolution in depth of our data is approximately 0.025 cm.[1]

We preprocess the images of the laser stripe data to obtain planar patches. This is done by finding sets of connected stripes that are nearly parallel. These stripes arise due to the intersection of the laser plane with a planar face. The $x$, $y$, $z$ coordinates of the points on these stripes are then used to compute a least-square planar fit. This method is very efficient and quite reliable. Many other techniques have been developed for obtaining planar regions for range data, e.g., [12], any of these would also be applicable here.

As in the edge-fragment case described earlier, we can exploit our knowledge of the extent of the planar patches in the matching process. We do this by selecting, within each planar region, four representative points that span the $xy$ range of the region (see Figs. 3 and 4). The matching algorithm is applied to these representative points and their corresponding normals. As in the case of edge fragments, we require that all four points be assigned to the same model face.

Our testing with the range data has been limited to a few objects, such as those illustrated in Figs. 3 and 4, in rather complex environments. The algorithm has found the correct interpretation in hundreds of tests using live data. The combined preprocessing and recognition time for these examples is approximately two minutes but, typically, only about 30 seconds of that is recognition time. It is the case, however, that the matching time grows fairly rapidly with the complexity of the model. In part this follows from the slightly weaker form of the constraints in three dimensions. Also, objects which exhibit partial symmetries (especially relative to the amount of error inherent in the sensory data) can frequently lead to multiple interpretations, when using sparse sensor information. For example, for the case illustrated in Fig. 3, if the sensory data all happen to lie on the block-like central portion of the object, and do not sample the projecting lip, the algorithm will discover several interpretations of the data, consisting of symmetric rotations of the object. Clearly, additional sampling of the object should reduce this ambiguity.

[1]The sensor has a depth resolution of about 1 part in 500 over a range of 12 cm.

In some cases, the algorithm will produce several very different interpretations that account for the same number of data patches. In those cases some type of verification is required. Two simple types of verification tests available for range data are: 1) test that the computed position and orientation of the model does not have it penetrating the known support surface, and 2) test that there are no known patches whose $xy$ projection lies on the localized object but whose $z$ value is *less* than that indicated by the model. These tests are relatively easy to implement and are quite effective.

### C. Range Data from an Ultrasonic Sensor

Michael Drumheller [10] has developed a modified version of our algorithm and applied it to range data obtained from an unmodified Polaroid ultrasonic range sensor. The intended application is navigation of mobile robots. The system matches the range data obtained by a circular scan from the robot's position towards the walls of the room. The robot has a map of the walls of the room, but much of the data obtained arises from objects on the walls, such as bookshelves, or between the robot and the walls, such as columns. The algorithm first fits line segments to the range data and attempts to match these line segments to wall segments. After matching, the robot can solve for its position in the room.

## VII. Extensions

In this section we briefly present some extensions of the algorithm presented above. These extensions are meant to suggest the range of application of the algorithm.

### A. Constrained Degrees of Freedom

We have assumed that if the objects are constrained to lie on a plane, then the data on each face are two-dimensional, and if the objects are completely unconstrained in position and orientation then the data are three-dimensional. In many applications, however, we can obtain three-dimensional data on objects constrained to be stably supported by a known plane, for example, a worktable. If we know the repertoire of the object's stable states, then we can exploit this knowledge as additional constraint to the matching process. Given a single data patch, the only candidate model faces for matching to it are those with similar values of the dot product between the face normal and the support plane normal. This constraint has the effect of drastically reducing the possible matches. This constraint is applicable even if we know that the object is not flush on the plane, but there is a known bound on its tilt relative to the plane.

### B. More Distinctive Features

If distinctive features, such as the location of holes or corners, are readily available from the data, then the algorithm described here can still be applied to exploit the geometric constraints between the positions and orientations of these features. The resulting algorithm is similar in effect to the Local-Feature-Focus method [7].

### C. Scale

We have assumed, throughout this paper, that models are metrically accurate, so that measured dimensions corresponded to model dimensions. This might fail to be true for two different reasons: we might be ignorant of some parameters in the sensing operation, such as viewing distance, or we might be dealing with variable objects, such as a family of motors. The approach we have described can be extended to deal with some of this variability.

The basic idea is that for a match of a measurement to a model entity to be valid, we must make some assumptions about the values of all unknown parameters, such as object scale. All the matches in a (partial) interpretation must imply consistent values for the parameters, otherwise the interpretation (and its descendants) can be pruned.

We have extended the recognition algorithm straightforwardly to allow for a linear scale factor, as illustrated in Fig. 12. As might be expected, we find that the number of nodes of the interpretation tree actually searched by the algorithm in this case is increased significantly from the comparable case of a known scale factor. This increase in the search space can be as large as an order of magnitude, depending on the amount of error inherent in the sensory data. As well, the mean number of interpretations, given the same number of data points, is slightly higher in the case of an unknown scale factor than in the case of a known one. Also, as shown in Fig. 12, including an additional parameter in the recognition process may lead to multiple interpretations, in which different values of the parameter lead to different feasible interpretations.

## VIII. Summary

We have shown how an object localization technique based on searching an interpretation tree could be extended to locate obscured parts. We observed that a straightforward application of the method was inadequate. We then explored a number of mechanisms for overcoming the problems of overlapping data. We found that:

- Although point-like data are adequate for localization in the isolated object case, they are not efficient in the overlapping object case.
- Search cutoff based on a quality measure is essential to limit the total search.
- A mixed strategy using Hough clustering as a preprocessing step can be used to reduce the size of the search space.
- The more complex coupled constraints do not provide a substantial benefit in reducing the search.

The recognition method obtained by incorporating these mechanisms into the IT search is quite robust and efficient, in spite of the fact that it does not exploit a great deal of readily available information. We have consciously avoided including additional information so that we could better explore the power of a few simple geometric features and constraints. We believe the approach can be readily extended to incorporate other information

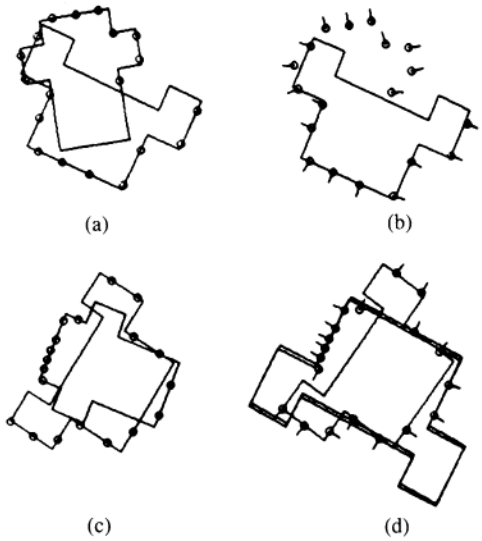



Fig. 12. Examples of parameterized two-dimensional models. A scaled version of one of the models is intermixed with another model. The recognition algorithm correctly identifies the object, and determines its scale factor as well as its position and orientation. (a) shows a set of sampled data, and (b) shows the interpretation of that data. (c) shows a second set of sampled data. (d) indicates that several interpretations of the data may be feasible.

such as adjacency between patches, positions of edges and vertices, and higher-level features. These extensions would make the approach more efficient at a cost in generality.

## Acknowledgment

The idea of using a null face to handle multiple objects was first suggested to one of us (TLP) by V. Melenkovic of CMU; we are very thankful for his remark. The image processing was done on a hardware/software environment developed by K. Nishihara and N. Larson. We thank P. Brou for kindly providing the laser ranging system with which we obtained the data reported in Figs. 3 and 4. We thank D. Huttenlocher for his comments on an earlier draft and we thank the referees for their valuable suggestions.

## References

[1] N. J. Ayache and O. D. Faugeras, "Recognition of partially visible planar shapes," in *Proc. 6th Int. Conf. Pattern Recognition*, Munich, West Germany, 1982.

[2] ——, "HYPER: A new approach for the recognition and positioning of two-dimensional objects," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-8, pp. 44–54, Jan. 1986.

[3] H. Baird, *Model-Based Image Matching Using Location.* Cambridge, MA: M.I.T. Press, 1986.

[4] D. H. Ballard and C. M. Brown, *Computer Vision.* Englewood Cliffs, NJ: Prentice-Hall, 1982.

[5] P. J. Besl and R. C. Jain, "Three-dimensional object recognition," *ACM Comput. Surveys*, vol. 17, no. 1, pp. 75–145, 1985.

[6] B. Bhanu and O. D. Faugeras, "Shape matching of two-dimensional objects," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-6, May 1984.

[7] 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.

[8] R. C. Bolles, P. Horaud, and M. J. Hannah, "3DPO: A three-dimensional part orientation system," presented at the *First Int. Symp. Robotics Res.*, Bretton Woods, NH, 1983; Also in *Robotics Research: The First International Symposium*, M. Brady and R. Paul, Eds. Cambridge, MA: M.I.T. Press, 1984, pp. 413–424.

[9] L. Davis, "Shape matching using relaxation techniques," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-1, pp. 60–72, Jan. 1979.

[10] M. Drumheller, "Robot localization using range data," S. B. thesis, Dep. Mechanical Eng., Massachusetts Inst. Technol., see also M.I.T. AI Lab Memo. 826, "Mobile robot localization using sonar."

[11] O. D. Faugeras and M. Hebert, "A 3-D recognition and positioning algorithm using geometrical matching between primitive surfaces," in *Proc. Eighth Int. Joint Conf. Artificial Intell.* Los Altos, CA: William Kaufmann, 1983, pp. 996–1002.

[12] O. D. Faugeras, M. Hebert, and E. Pauchon, "Segmentation of range data into planar and quadratic patches," in *Proc. CVPR'83*, Washington, DC.

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

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

[15] P. C. Gaston and T. Lozano-Pérez, "Tactile recognition and localization using object models: The case of polyhedra on a plane," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-6, pp. 257–265, May 1984.

[16] C. Goad, "Special purpose automatic programming for 3d model-based vision," in *Proc. DARPA Image Understanding Workshop*, 1983.

[17] W. E. L. Grimson, "The combinatorics of local constraints in model-based recognition and localization from sparse data," M.I.T. AI Lab Memo 763; see also *J. ACM*, vol. 33, no. 4, pp. 658–686, 1986.

[18] 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.

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

[20] R. M. Haralick and L. G. Shapiro, "The consistent labeling problem: Part I," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-1, pp. 173–184, 1979.

[21] D. G. Lowe, "Three-dimensional object recognition from single two-dimensional images," Courant Inst. Robotics Rep. 62, 1986.

[22] D. Marr and E. C. Hildreth, "Theory of edge detection, *Proc. Roy Soc. London*, vol. B207, pp. 187–217, 1980.

[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.

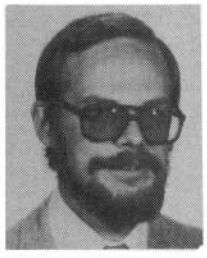
[25] U. Montanari, "Networks of constraints: Fundamental properties and

applications to picture processing," *Inform. Sci.*, vol. 7, pp. 95–132, 1974.

[26] G. Stockman and J. C. Esteva, "Use of geometrical constraints and clustering to determine 3D object pose," Dep. Comput. Sci., Michigan State Univ., East Lansing, Tech. Rep. TR84-002, 1984.

[27] J. L. Turney, T. N. Mudge, and R. A. Volz, "Recognizing partially occluded parts," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-7, pp. 410–421, July 1985.

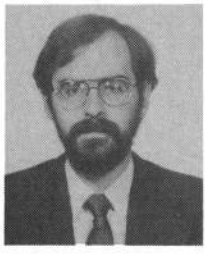
[28] D. Waltz, "Understanding line drawings of scenes with shadows," in *The Psychology of Computer Vision*, P. Winston, Ed. New York: McGraw-Hill, 1975, pp. 19–91.



**W. Eric L. Grimson** (M'84) was born in Estevan, Sask., Canada, in 1953. He received the B.Sc. degree in mathematics and physics from the University of Regina, Sask., in 1975, and the Ph.D. degrees in mathematics from the Massachusetts Institute of Technology, Cambridge, in 1980.

From June 1980 to June 1984 he was a Research Scientist at the Massachusetts Institute of Technology Artificial Intelligence Laboratory. Since July 1984 he has been an Assistant Professor in the Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology. 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* and is the editor of *AI in the 1980's and Beyond: An MIT Survey*, both published by M.I.T. Press.

Dr. Grimson is a member of the Association for Computing Machinery, and the American Mathematical Society. He is also an Associate Editor of IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE.



**Tomás Lozano-Pérez** (M'86) received the B.S., M.S., and Ph.D. degrees in computer science from Massachusetts Institute of Technology, Cambridge, in 1973, 1976, and 1980, respectively.

He is Associate Professor of Computer Science and Engineering at M.I.T. where he is a member of the Artificial Intelligence Laboratory. Before joining the M.I.T. faculty in 1981 he was on the Research Staff at IBM Thomas J. Watson Research Center during 1977, and at the M.I.T. AI Laboratory during 1974 and again during 1980. His research interests are in artificial intelligence and robotics; he teaches courses in these areas at M.I.T. He has written over 30 research papers, mostly in robot manipulation and computer vision. He is a co-editor of the book *Robot Motion* (M.I.T. Press, 1982).

Prof. Lozano-Pérez is co-editor of the *International Journal of Robotics Research*. He was Program Chairman of the 1985 IEEE International Conference on Robotics and Automation. He is a recipient of a 1985 Presidential Young Investigator Award.