MASSACHUSETTS INSTITUTE OF TECHNOLOGY
ARTIFICIAL INTELLIGENCE LABORATORY

A. I. Memo 841 **June, 1985**

RECOGNITION AND LOCALIZATION OF OVERLAPPING PARTS FROM SPARSE DATA

W. Eric L. Grimson
Tomás Lozano-Pérez

Abstract. This paper discusses how sparse 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 non-overlapping parts previously described in [Grimson and Lozano-Pérez 84] and [Gaston and Lozano-Pérez 84].

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 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.

1. Background

In order to interact intelligently with its environment, a robot must know *what* objects are *where*; that is, it must be able to identify and locate objects in its workspace. In this paper, we treat these two tasks under the title of the *recognition problem*. We will stress localization over identification since in most industrial robotics tasks the identity of the objects is known.

A solution to the recognition problem should satisfy the following criteria:

1. A recognition algorithm must degrade gracefully with increasing noise in the sensory measurements.
2. A recognition algorithm should be able to deal with partially occluded objects.
3. A recognition algorithm should be able to identify and locate objects from relatively sparse information. The sparseness of the data may be due to inherent sensor sparseness, occlusion, or noise.
4. A recognition algorithm should be applicable to different sensor types. While particular optimizations will be possible for specific sensors, one would like a recognition technique that serves as a common core for recognition from tactile, ranging, sonar and visual sensors.

This paper presents an approach to the recognition problem that satisfies these criteria. 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 non-overlapping parts previously described in [Grimson and Lozano-Pérez 84] and reviewed in Section 3. The new method is described in Section 4; it handles highly overlapped parts using either two-dimensional visual data or three-dimensional range data (see Figures 1 - 4). We also report in Section 5 our experience with a new formulation of the geometric constraints that do not decouple position and orientation. A number of other extensions are described in Section 6 including a technique for dealing with objects of unknown size. Section 7 is a discussion of our approach to recognition and a review of related work.

2. Problem Definition

The specific problem considered in this paper is how to identify a known object and locate it, relative to a sensor, using relatively few measurements. Because we seek a technique that is applicable to a wide range of sensors, we make few assumptions about the sensory data. We assume only that the sensory data can be processed to obtain sparse measurements of the position and surface orientation of small planar patches of object surfaces in some coordinate frame defined relative to the sensor. The measured positions are assumed to be within a known error volume and measured surface orientations to

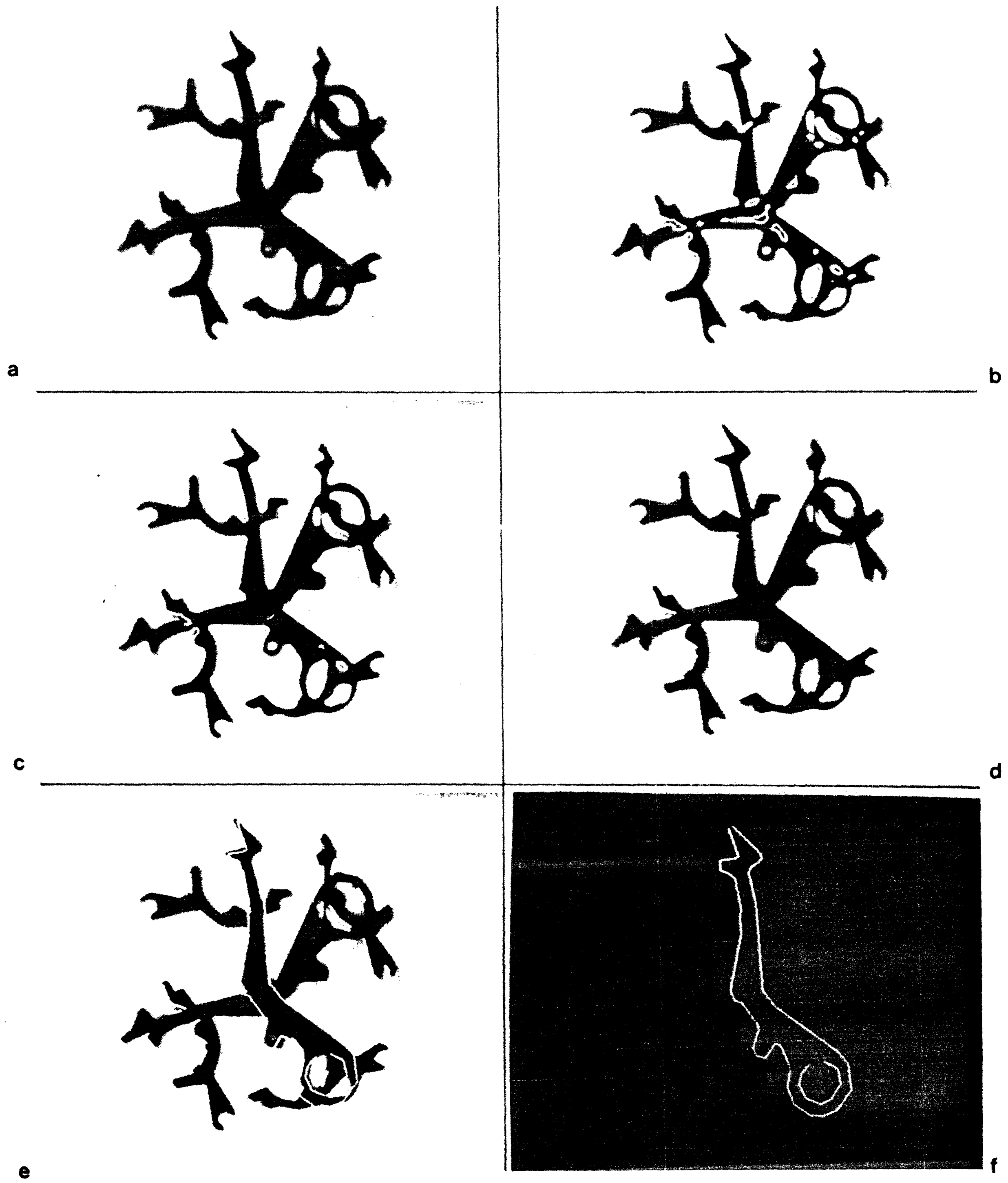

Figure 1. Two dimensional edge data. (a) Grey level images, (b) zero crossings, (c) thresholded zero crossings (d) edge fragments, (e) located object in image, and (f) located object.

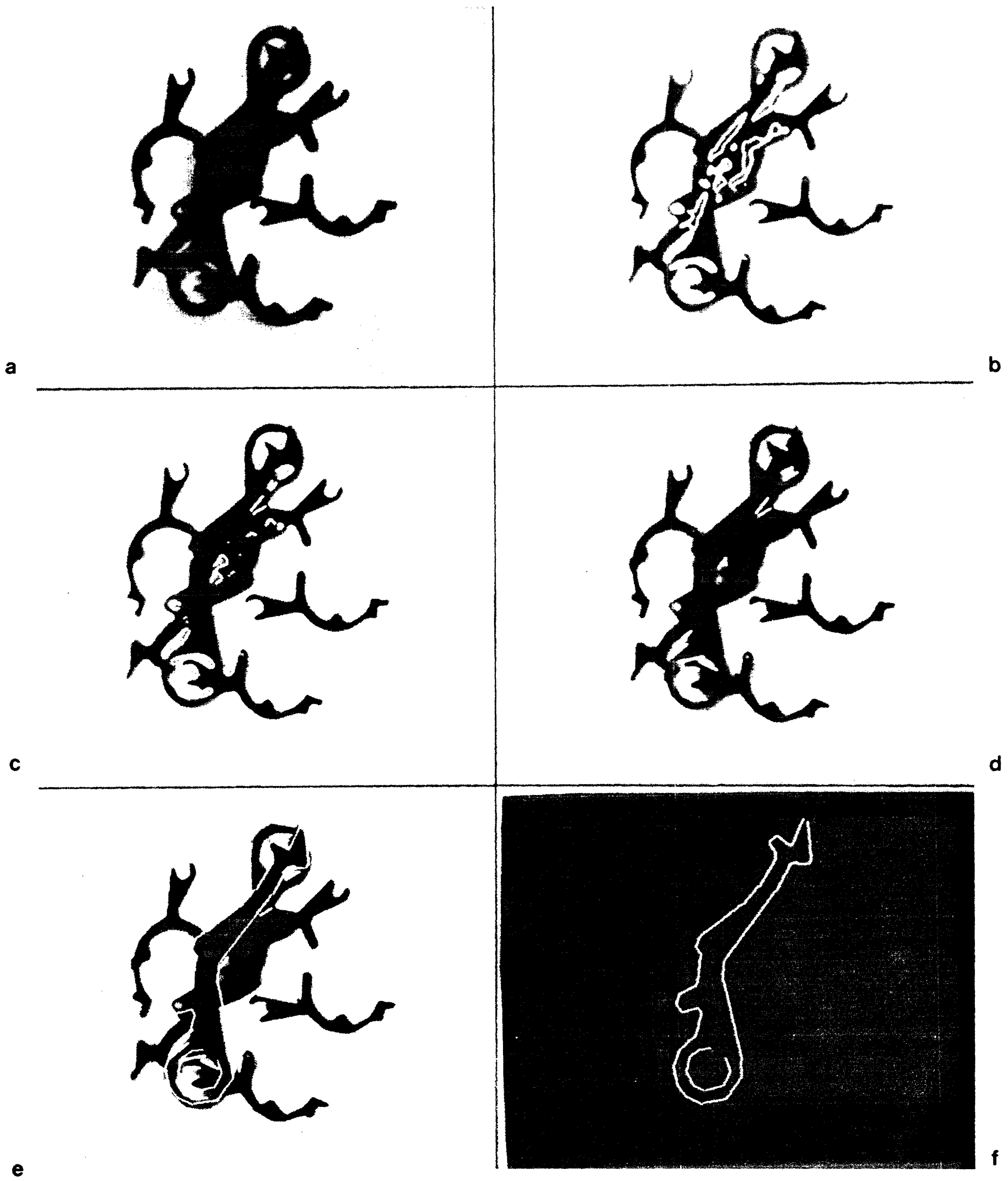

Figure 2. Two dimensional edge data. (a) Grey level images, (b) zero crossings, (c) thresholded zero crossings (d) edge fragments, (e) located object in image, and (f) located object.

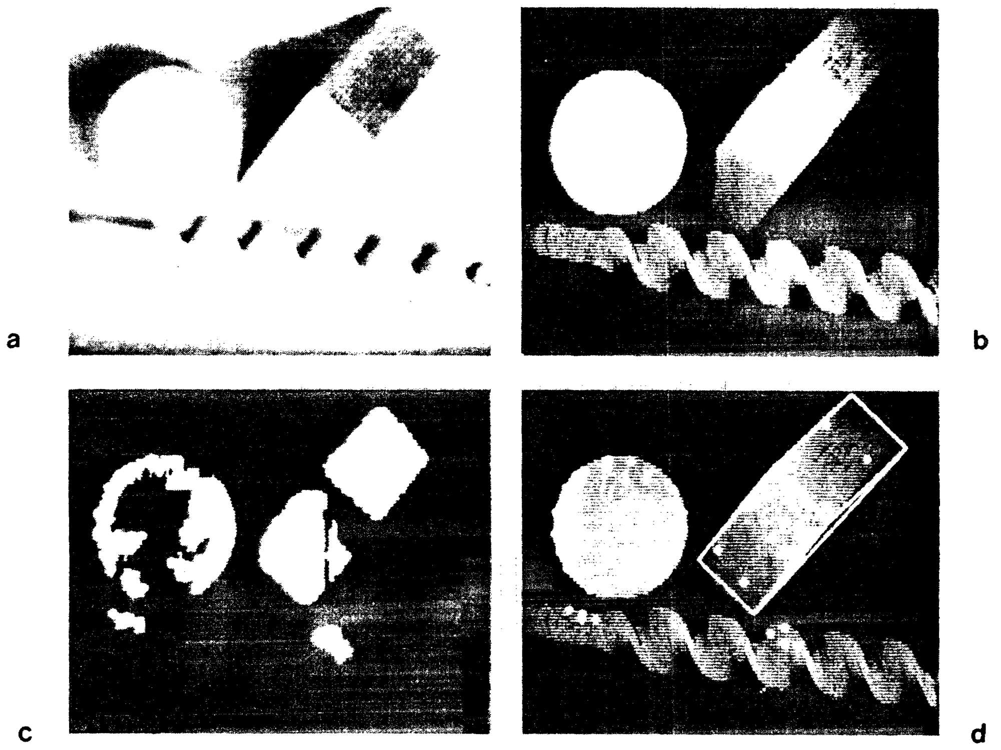

Figure 3. Three dimensional range data. (a) Original scene (b) range data where brightness encodes height, (c) planar patches with representative points, and (d) located object superimposed on range data (filled in circles are data points accounted for).

be within a known error cone. Furthermore, the object may be overlapped by other unknown objects, so that much of the data does not arise from the object of interest.

If the objects have only three degrees of positional freedom relative to the sensor (two translational and one rotational), then the positions and surface normals need only be two-dimensional. If 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.

Since the measured data approximate small planar patches of the object's surface, we assume that the objects can be modeled as sets of planar faces. Only the individual plane equations and dimensions of the model faces are used for recognition and localization. No face, edge, or vertex connectivity information is used or assumed; the model faces do not even have to be connected and their detailed shape is not used. This is important. It is easy to build polyhedral approximations of moderately curved objects, but we cannot expect these approximations to be perfectly stable under sensor variations. The positions of vertices, orientations and lengths of edges, and area of individual faces will all vary depending on the position of the object relative to the sensor. Our recognition method does not rely on the detailed polyhedral model; it relies instead on aggregate geometric information about the object's faces. As a result, the method can be readily applied to

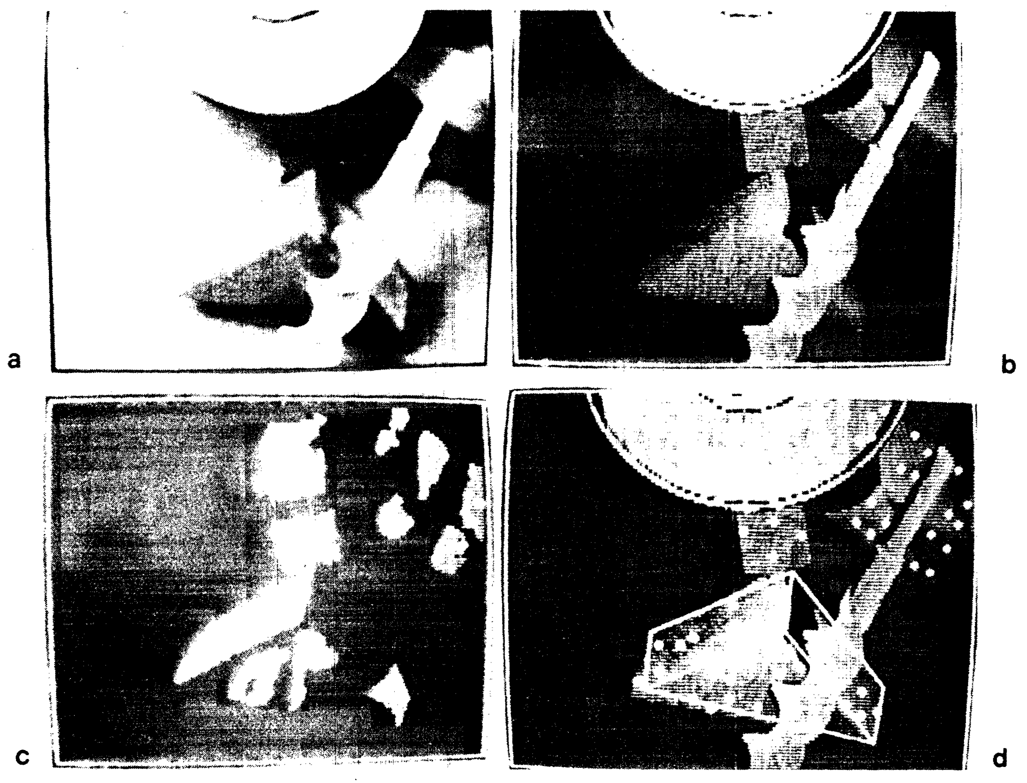

Figure 4. Three dimensional range data. (a) Original scene (b) range data where brightness encodes height, (c) planar patches with representative points, and (d) located object superimposed on range data (filled in circles are data points accounted for).

curved objects approximated by planar patches.

2.1 Basic Approach

Our approach to model-based recognition is to cast it as a search for a consistent matching between the measured surface patches on the one hand, and the surfaces of known object models on the other hand. The search proceeds in two steps:

1. *Generate Feasible Interpretations:* A set of feasible interpretations of the sense data is constructed. Interpretations consist of pairings of each sensed patch with some object surface on one of the known objects. Interpretations in which the sensed data is inconsistent with local 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.

There are several possible methods of actually searching for consistent matches. For example, in Grimson and Lozano-Pérez [84] we chose to structure the search as the

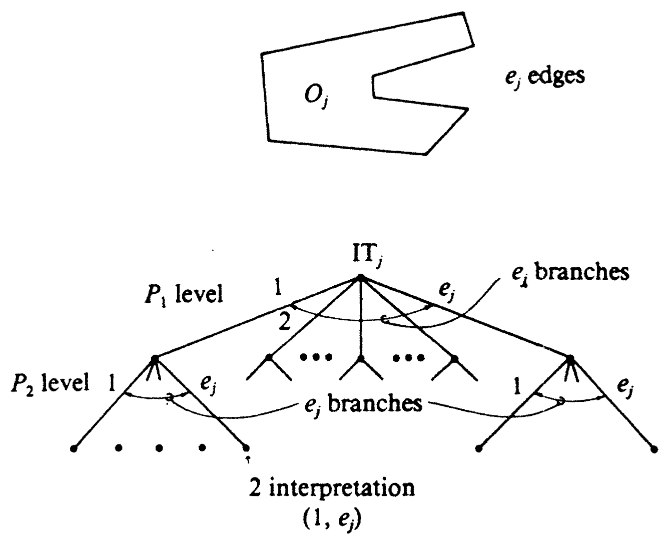

Figure 5. The *interpretation tree.* **A path through this tree represents a set of pairings of measured patches to model faces.**

generation and exploration of an *interpretation tree* (see Figure 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 possible interpretations. Moreover, since the computation of coordinate frame transformations tends to be expensive, we want to apply this part of the technique only as needed.

The goal of the recognition algorithm is thus to exploit local geometric constraints to minimize the number of interpretations that need testing, while keeping the computational cost of each constraint small. In the case of the interpretation tree, we need constraints between the data elements and the model elements that will allow us to remove entire subtrees from consideration without explicitly having to search those subtrees. 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 at the generation stage should satisfy the following criteria:

1. The constraints should be coordinate-frame independent. That is, they should prune large portions of the search space, independent of the particular orientation of the object. That is, the constraints should embody restrictions due to object shape and not to sensing geometry.
2. The constraints should be simple to compute and, at the same time, able to prune large portions of the search space.
3. The constraints should degrade gracefully in the presence of error in the sensory measurements.

4. The constraints should be independent of the specifics of the sensor from which the data came, so that they will apply equally to different types of sensors.

In this paper, we deal with two different, but related, sets of geometric constraints. In the first set position and orientation are decoupled. The decoupling leads to very efficient implementations, but reduces their pruning power. The second set retains the natural coupling between positions and orientations. This set is more powerful, but computationally more complex. Both sets are developed first for the case of a single, isolated object, and then for the case of overlapping objects.

3. Decoupled Constraints

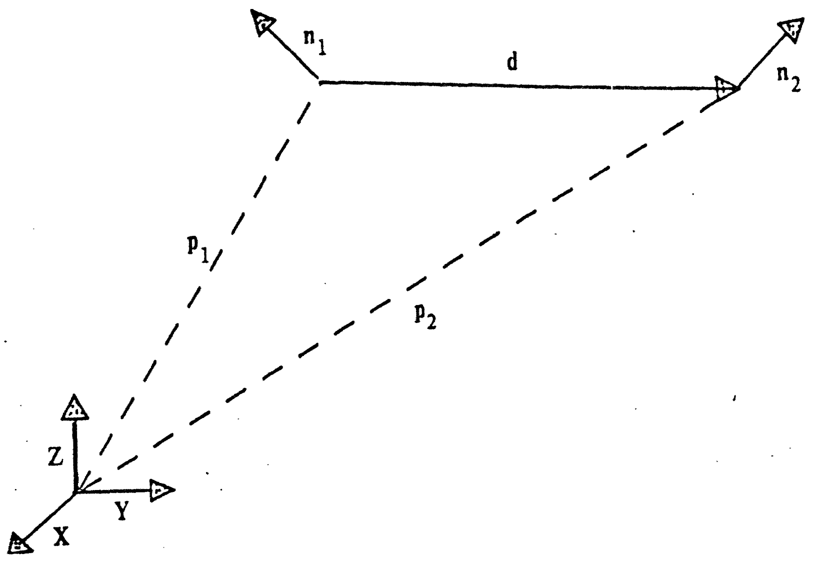

Figure 6. The constraints between pairs of measured surface patches. A given pair of sensory points P_1, P_2 can be characterized by the components of the vector $\mathbf{d}$ between them, in the direction of each of the surface normals $\mathbf{n}_1, \mathbf{n}_2$ and in the direction of their cross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

In this section we review the method for recognizing non-overlapping parts presented in [Grimson and Lozano-Pérez 84]. We begin by deriving a set of coordinate-frame-independent constraints that were presented there. The first question to ask is what types of coordinate-frame-independent constraints are possible, given that the sensory data are sparse planar patches, each consisting of a position measurement and a unit surface normal (see Figure 6). Clearly, a single patch provides no constraint on the model faces that could consistently be matched to it. Therefore, we consider pairs of patches. Each such pair can be characterized by the pair of unit normals, $\mathbf{n}_1$ and $\mathbf{n}_2$, and the separation vector between the patch centers* $\mathbf{d}$, as shown in Figure 6.

3.1. The 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 $\mathbf{d}$ along each of the basis directions and the angle between the two measured normals. More formally,

$$\begin{gathered}\mathbf{n}_1 \cdot \mathbf{n}_2 \\ \mathbf{d} \cdot \mathbf{n}_1 \\ \mathbf{d} \cdot \mathbf{n}_2 \\ \mathbf{d} \cdot \mathbf{u}\end{gathered}$$

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

These measurements are equivalent, but not identical to the set used in [Grimson and Lozano-Pérez 84]. In the earlier paper, we used the magnitude of $\mathbf{d}$ 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 was resolved using a triple product constraint.

To turn these measurements into constraints on the search process, we must relate them to measurements on the model elements. Since objects are modeled as sets of planar faces, the relationship is straightforward. Consider the first measurement, $\mathbf{n}_1 \cdot \mathbf{n}_2$. 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 amount of error in a sensory measurement, we can compute a range of possible values associated with the dot product of two sensed normals (see [Grimson and Lozano-Pérez 84] for details). In this case, if the dot product of the associated model normals is contained in the range of possible values associated with the dot product of the sensed normals, then the corresponding assignment of model faces to sensed points is consistent.

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

*The extent of the patches is assumed small.

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. As in the case of surface normals above, we can incorporate the effects of error in the measurements by using bounds on the magnitude of the error to compute a range of possible values for the components of the sensed vectors, and this range must be consistent with the associated model range, in order for an interpretation to be consistent.

It is easy to see that these constraints satisfy most of our criteria: they are independent of coordinate frames, simple, and general. It remains to establish that they are powerful and degrade gracefully with noise. Grimson [84] argues from a combinatorial analysis that these constraints are very powerful, and in the case of data all obtained from a single object, will converge quickly to a small set of interpretations. The analysis also shows that the constraints should exhibit a graceful degradation with increasing sensor noise. These predictions have been verified empirically, both by simulation and by processing real sensory data. Grimson and Lozano-Pérez [84] report on a large set of simulations run on a series of test objects, for varying types of error conditions.

3.2. Adding A Model Test

Once the interpretation tree has been pruned, there are typically only a few non-symmetric interpretations of the data remaining. It is important to realize, however, that these constraints are not guaranteed to reject all impossible interpretations. Let d_{ij} be the distance between two sensed patches, P_i and P_j. It could be the case that this measured distance is consistent with the range of distances between faces f_u and f_v, but only if the patches are inside of small regions on the candidate surfaces. Now consider what happens when adding another patch-surface pairing, (P_k, f_w), to an interpretation that already includes (P_i, f_u) and (P_j, f_v). Our constraints permit adding this pairing only if the distances d_{ik} and d_{jk} are consistent with the range of distances between f_u, f_w and f_v, f_w, respectively. In doing this, however, it uses the ranges of distances possible between any pair of points on these faces. It does not take into account the fact that only small regions of f_u and f_v are actually eligible by previous patch-surface pairings.

Because of this decoupling of the constraints, the fact that all pairs of patch-surface assignments are consistent 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 for actually solving for the transformation, one is described in [Grimson and Lozano-Pérez 84]. 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.

When a model test is applied to the feasible interpretations, two effects are noticed. First, some locally consistent interpretations are discarded as being globally inconsistent. Second, of the remaining interpretations, many are observed to be almost identical, differing, for example, only in the assignment of one or two data points to nearby faces. To get a more accurate picture of the effectiveness of the recognition technique, we can cluster the interpretations that satisfy the model test on the basis of their associated transformations. In particular, we cluster solutions whose direction of rotation is within 1.5^o of each other. The number of distinct interpretations is greatly reduced.

4. Data from Overlapping Objects

The method described in the previous section assumes that all the data comes from a single object. In this section we show how the method can be extended to handle data from multiple overlapping objects.

Assume that all of the sensed patches, except one, originate from the same object. Let P_i be the extraneous measurement. Usually, it will be impossible to find an interpretation that includes this measurement. But, not all interpretations will fail at level i in the tree; it may require adding a few more data points to the interpretation before the inconsistency is noted. It is only when all possible single-object interpretations fail that we are certain to have at least one extraneous data point.

It may still be possible to find an interpretation of all the data, including extraneous measurements, that is consistent with the pairwise constraints. It is even possible, by a fortuitous alignment of the data, for interpretations involving extraneous data to pass the model test. There is nothing within the approach described here to exclude this possibility. Of course, the larger the number of patch-surface pairings in the interpretation, the less likely this is to happen. In many cases, it may be necessary to verify the interpretation by acquiring more data. We will not pursue this point here; we will assume, instead, that the presence of extraneous points will cause all interpretations to fail either the local constraints or the model test.

4.1 Generating Interpretations for Subsets of the Data

One straightforward approach to handling extraneous data points is to apply the recognition process to all subsets of the data, possibly ordered by some heuristic. But, of course, this approach wastes much work determining the feasibility of the same partial interpretations. Can we consider all subsets of the data without wasting the work of testing partial interpretations? The simple way we have done this is by adding 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 remainder of the process operates as before except that, when applying the local constraints, the null face behaves as a "wild card"; assigning a patch to the null face will never cause the failure of an interpretation.

It is easy to see that if an interpretation is legal, the process described above will generate all subsets of this interpretation as leaves of the tree. This is true of partial interpretations as well as full interpretations since every combination of assignments of the null face to the sensed patches will still produce a valid interpretation.

The same condition that ensures the validity of this process guarantees its inefficiency. We do not want to generate all subsets of a valid interpretation. In general, we want to generate the interpretations that are consistent with as much as possible of the sensed data. The following simple 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. In addition, the model test is applied to any complete interpretations, that is, any that reach a leaf of the IT. This choice of tree traversal has the effect of considering the legal interpretations essentially by length order (where length is taken to be the number of non-null faces paired with sensed data by the interpretation).

Now, assume an external variable, call it MAX, that keeps track of the longest valid interpretation found so far. At any node in the tree, let M denote the number of non-null faces in the partial match associated with that node. It is only worth assigning a null face to point P_i, if $s - i + M \geq MAX$; s is the total number of sensed patches. Otherwise, the length of the interpretations at all the leaves below this node will be less than that of the longest interpretation already found. If we initialize MAX to some non-zero value, then only interpretations of this length or longer will be found. As longer 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 length s is found, then no null-face assignments will be considered after that point.

Looking for the longest consistent interpretation allows the matching algorithm to overcome the most severe combinatorial problems of the null-face scheme, but it makes the algorithm susceptible to a potentially serious problem. One of the bases of our approach to recognition has been to avoid any global notion of "quality" of match. We have simply defined generous error bounds and found all interpretations that did not violate these bounds. Once all the valid interpretations have been found, a choice between them can be made on a comparative basis rather than on some arbitrary quality measure. The modified algorithm, however, discards valid interpretations that are shorter than the longest valid interpretation. Therefore, a long interpretation on the margin of the error bounds can force us to ignore a shorter interpretation that may be the correct match.

We know of no general solutions to this problem. Quality measures such as how well the transformation maps the measured patches onto the faces [Faugeras and Hebert 83] are useful but also susceptible to error. Our choice would be to consider all the valid interpretations whose length is within one or two of the longest interpretation and which are not subsets of a longer interpretation. This is also heuristic. We have avoided this issue in the rest of the paper and simply coped with the occasional recognition error.

4.2. Heuristics for Limiting Search

The presence of overlapping objects dramatically increases the search space for recognition. One effect is simply the need for more measurements to ensure that there are enough measurements on the object of interest. The other effect is the need to consider many alternative subsets of the measurements. Even when using local constraints and focusing on longest interpretations, complete exploration of the interpretation tree is extremely time consuming. We have achieved a significant improvement in recognition time by using two heuristic techniques, both of which reduce the size of the interpretation tree that is explored. Both of these techniques may cause the recognition process to fail but, in most applications, one prefers to reduce the recognition time even at the expense of occasional failures.

The first heuristic technique avoids considering some subsets of the measurements that are unlikely to lead to the longest interpretation. In particular, once an interpretation I is longer than some threshold, it ignores interpretations in which data points matched to faces by I are matched to the null face. This heuristic can be easily incorporated in the depth-first tree exploration we use. Our experience is that this technique when used with a conservative threshold is quite effective in pruning the search without noticeably increasing the failure rate. We have used this techniques in all our experiments with real data reported later in this section.

The other heuristic technique is based on the Hough transform [Ballard 81]. It does not affect the search algorithm, instead it drastically reduces the size of the initial interpretation tree. 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, θ, 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, θ 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, θ combination (modulo measurement error). We keep track of the range of x, y, θ 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 θ. Clusters of pairings with nearly the same values define candidate interpretations of the data (see Figure 7).

This technique can serve as a recognition method all by itself [Ballard 81], 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

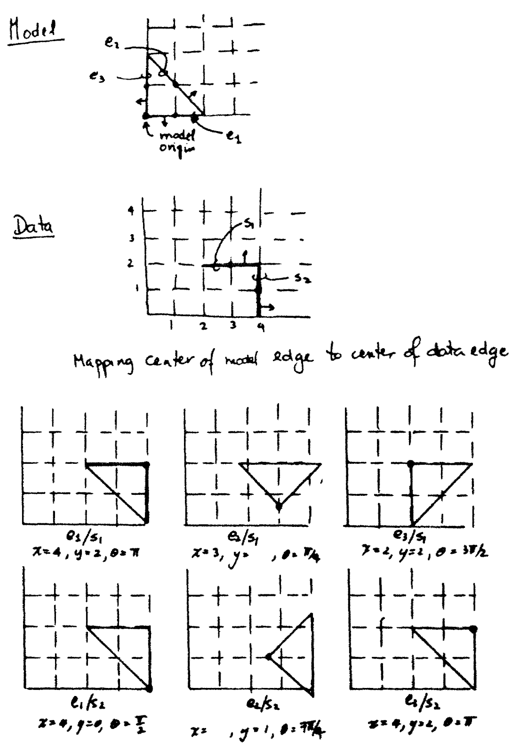

Figure 7. Hough transform preprocessing establishes initial pairings between model edges and data edges.

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 two-stage matching method, interpretation generation followed by model test, is then applied to the largest clusters until a valid interpretation is found. The effect of the initial clustering is to drastically reduce the size of the search space at the expense of initial preprocessing. Typically, data edges in a cluster are associated with only one or two model edges out of the possible thirty or forty edges in the model. Therefore, the branching factor in the interpretation tree is cut down from thirty or forty to one or two. Predictably, this has a noticeable impact on performance. Many of the pairings, however, are still spurious, due to noise and the fact that the parameters do not completely characterize position and orientation. Therefore, it is necessary to use the null-face technique described earlier.

We have tested the resulting method on simulated data as well as on actual data from three types of sensors. The results are described in the following sections.

4.3. Simulations with Two-Dimensional Data

We have done extensive testing of the algorithm with simulated two-dimensional data of the type illustrated in Figure 8. A number of polygons, representing the outlines of parts, are overlapped at random. The position and orientation of a number of data points are determined by computing the outermost intersection of randomly chosen rays with the polygon boundaries. The position and normal information is then corrupted by random errors designed to simulate the effect of imperfect sensors (see Figure 8, the small circles indicate the sensed points).

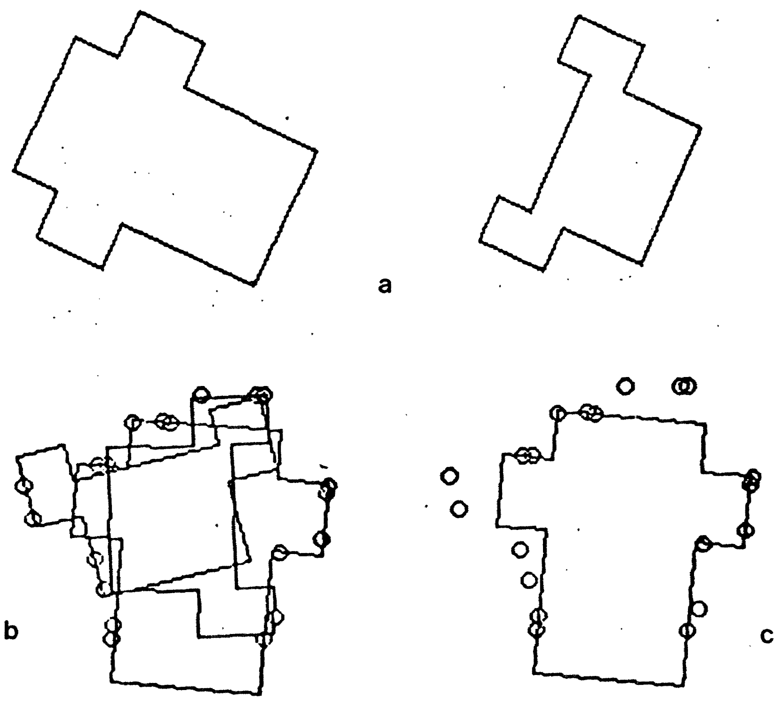

Figure 8. Simulations of overlapping two dimensional parts. A collection of copies of objects selected from the set illustrated in part (a) was overlapped at random, as illustrated in part (b). Points of contact were selected at random along the perimeter of the overlapping group, and corrupted with random error. The circles in part (b) indicate an example of sensory data. The recognition and localization algorithm then searched for interpretations of the data consistent with a specific model, as shown in part (c).

For example, the data reported in Table I were obtained by the following simulation technique. Three objects, two of which were not the object of interest, were overlapped at random. Given the overlapping parts, twenty points of contact were chosen at random and then corrupted by random error whose magnitude was bounded by the limits indicated in the table. Note that the diameter of the object was 2.506 units, so that an error in positional measurement of .1 is roughly 4 percent of the total diameter of the object. Once the interpretations, with their associated coordinate transformations, were found, they were clustered together into distinct interpretations. To do this, transforms whose difference in rotational angle was less than $\frac{\pi}{18}$ and whose translational components differed by less than 0.025 were considered to be the same. The data in Table I lists the results of running this simulation one hundred times for each of the error ranges illustrated. Similar results using arbitrary numbers of overlapping parts, and using different numbers of contact points have also been obtained.

The algorithm performs quite well in this application, even without using the heuristic techniques described earlier. The extensive simulations reported in Table I indicate that the technique can reliably recognize and locate overlapping two-dimensional objects, even when only using a few points of sensory data obtained along the silhouette of the overlapping parts. As the amount of the overlapping of the parts becomes more extensive, the algorithm may begin to find more than one feasible interpretation, but the number of solutions degrade gracefully. In general, as long as enough points are sensed on the desired object, the algorithm can locate it. Furthermore, the time to do the recognition and localization is relatively low: on the order of two or three seconds on a Symbolics 3600 Lisp Machine. The time grows when the measurement error grows, in the manner illustrated by the simulations reported in [Grimson and Lozano-Pérez 84].

4.4. Simulations with Three-Dimensional Data

We have performed similar simulations with overlapping three dimensional objects, each with six degrees of freedom, as illustrated in Figure 9. In this case, a number of polyhedra are overlapped at random, and the position and orientation of a number of data points are determined by computing the outermost intersection of randomly-chosen rays with the polyhedral boundaries. The position and orientation information is then corrupted by random errors.

The data reported in Table II were obtained using the following technique. Two objects were overlapped at random, and fifteen points of contact were chosen at random and then corrupted by random error whose magnitude was bounded by the limits indicated in the table. In this case, the diameter of the object was 3.95 units, so that an error in positional measurement of 0.15 is roughly 4 percent of the total diameter of the object.

In the two dimensional case reported in Table I, model tests were directly intermixed with the generation of the tree of interpretations. That is, whenever a leaf of the tree was reached, a model transformation was computed and the sensory points were checked to ensure that the transformation did correctly transform the points onto their associated faces. While this ensures that the use of the external variable MAX will not exclude any correct interpretations, the algorithm is potentially slow because of the expense of computing and applying model transformations. A faster but less reliable alternative is to generate the tree of interpretations, using the variable MAX to cutoff search as before, but in this case to simply collect all pairwise consistent interpretations, ordered by the number of non-null matches. That is, any leaf in the `IT` is added to a list of possible interpretations, ordered by the number of real matches in the interpretation. Once all such interpretations of the `IT` have been collected into sets of equal length (as defined by the number of non-null matches), we apply a model test to each interpretation of a set, starting at the set of interpretations with the largest number of real matches, and continuing until at least one interpretation is judged to be globally consistent. The interpretations from the corresponding set that pass a model test constitute the set of interpretations of the data.

While such a technique is clearly expected to be faster than applying a model test at each leaf of the tree, it is not guaranteed to find the correct interpretation. In particular, suppose that the correct interpretation has m real points of contact, but that the pairwise constraints allow an interpretation with $m + 1$ non-null matches to pass through. This will cause MAX to be at least $m + 1$, and may therefore prevent the search process from

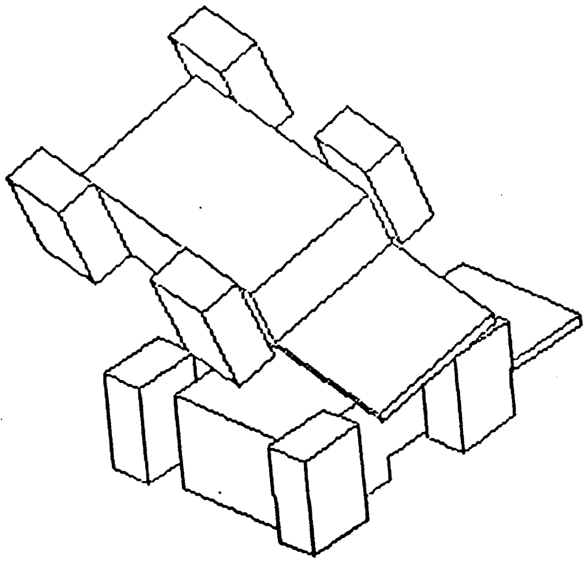

Figure 9. Simulations of overlapping three dimensional parts. Simulations similar to those shown in Figure 8 were performed in three dimensions, with overlapping parts such as those shown above.

finding the correct interpretation. When the model test is applied after the generation of the IT, it will exclude the $m + 1$ match, and this will cause the process to find no consistent interpretation.

Since the technique is much faster, it is of interest to know how often such a method will fail to find the correct answer. Thus, in our simulations of 3D overlapping parts, we have applied this faster, but less reliable technique, and recorded the percentage of cases in which no interpretation was found. The data reported in Table II illustrate the results of running 100 simulations for each of the ranges of error listed. As in the 2D case, once the interpretations, together with their associated transformations, were found, they were clustered on the basis of their transformations, and the statistics of histogramming these clustered interpretations are reported. As well, the percentage of cases in which no interpretation was found are also listed.

We note that the algorithm appears to be less effective at finding unique interpretations of three-dimensional objects than of two-dimensional ones. In part this follows from the slightly weaker form of the constraints in three dimensions, especially when using only points of contact, rather than extended regions. As well, 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 Figure 9, 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.

4.5. Edge Fragments from Gray–Level Images

A modified version of the algorithm described here has been applied to locating a complex object in cluttered scenes, using edge fragments from images obtained by a camera located (almost) 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 [Marr and Hildreth 80]. Edge points are marked only when the gradient at that point exceeds a predefined threshold; this is done to eliminate some shallow edges due to shadows. The algorithm is applied to some predefined number of the longest edge fragments.

This application requires extensions to the general method. One point to notice is that we have large edge fragments rather than small patches; therefore, we can use the length of the fragments as an additional local constraint. In our implementation, we do not assign edge fragments to model edges that are shorter than the measured fragment; we do assign small edge fragments to long model edges. More importantly, we could compute whole ranges of measurements from the edge fragments (as we do from model edges) rather than the single values from point–like patches we assume elsewhere. The constraints would then require that the measured range be contained in the model range. An easy way of approximating these stronger constraints is by treating the edge as two small patches located at endpoints of the edges, but constraining both patches to be assigned to the same model edge. Both of these approaches can be generalized to three–dimensions.

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, for example, a light–colored object on a dark background. 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 possible 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 Figures 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 Figures 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.

The time to perform the recognition on this class of problems using the method described in this paper varies significantly, depending on the complexity of the problem, whether the complete normal is available and whether the heuristic techniques of section 4.2 are employed. In all cases, the preprocessing to obtain edges is common. In our current implementation it takes approximately 30 seconds of elapsed time on a Symbolics 3600 Lisp Machine to process an image to obtain 80 edge fragments; this time, however, can be reduced by an order of magnitude by using existing hardware. We do not include this time in the times quoted below (all times are for our Lisp implementation).

When the sign of the normal is unknown and without using the Hough preprocessing, difficult cases such as in Figures 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.

The effect of the Hough preprocessing is to make the recognition time nearly independent of the complexity of the scene. In our testing, we used the full set of x, y, θ 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.

4.6. Range Data from Structured Light

We have applied our algorithm to locating a simple three-dimensional object in cluttered scenes, using relatively dense range data. The range data was obtained from a laser-striping system developed by Philippe Brou at our laboratory. The sensor is a vidicon camera above the scene. On either side of the camera there is a low-power laser at a known distance and angle from the camera. Each laser beam is converted into plane of light by a cylindrical lens; the resulting planes are scanned across the field of view by computer controlled mirrors. A Lisp machine identifies the xy position of points on the scene illuminated by the laser. The xy positions of these points are then used to compute the height above the plane of points intersecting the plane of light. Two lasers stripes, originating on opposite sides of the camera, are used so as to allow sensing in areas that would be shadowed for a single laser.

The data obtained from the ranging system is dense along each stripe, but the distance between stripes is under program control. The data used in our experiments (see Figures 3 and 4) were taken at a resolution of 0.03 cm in the vertical direction and 0.12 cm in the horizontal direction. The resolution in depth of our data is approximately 0.025 cm.*

Once the depth map is obtained, we preprocess the data to obtain planar patches. A least-square planar fit is done at every data point. Regions are then formed whose normals are within a user specified angle from some "seed" normal. Many techniques have been developed for obtaining planar regions for range data, e.g. [Faugeras, Hebert, Pauchon 83], 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 to more tightly constrain the matching process. We do this by selecting, within each planar region, four representative points that span the xy range of the region (see Figures 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.

In some cases, the algorithm will produce several very different interpretations that account for the same number of data points. In those cases some type of verification is required. Two simple types of verification tests available for range data are: (a) test that the computed position and orientation of the model does not have it penetrating the known support surface, and (b) make sure 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 effective in many cases (Figure 10).

Our testing with the range data has been limited to a few objects, such as the wedge in Figure 10, in rather complex environments. 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. This growth is due to the relative weakness of the three-dimensional constraints compared to the two-dimensional constraints. We expect that the Hough preprocessing will help reduce this problem, but we have yet to test the three-dimensional preprocessing.

4.7. Range Data from an Ultrasonic Sensor

Michael Drumheller [Drumheller 84] has developed a modified version of the algorithm described above 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

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

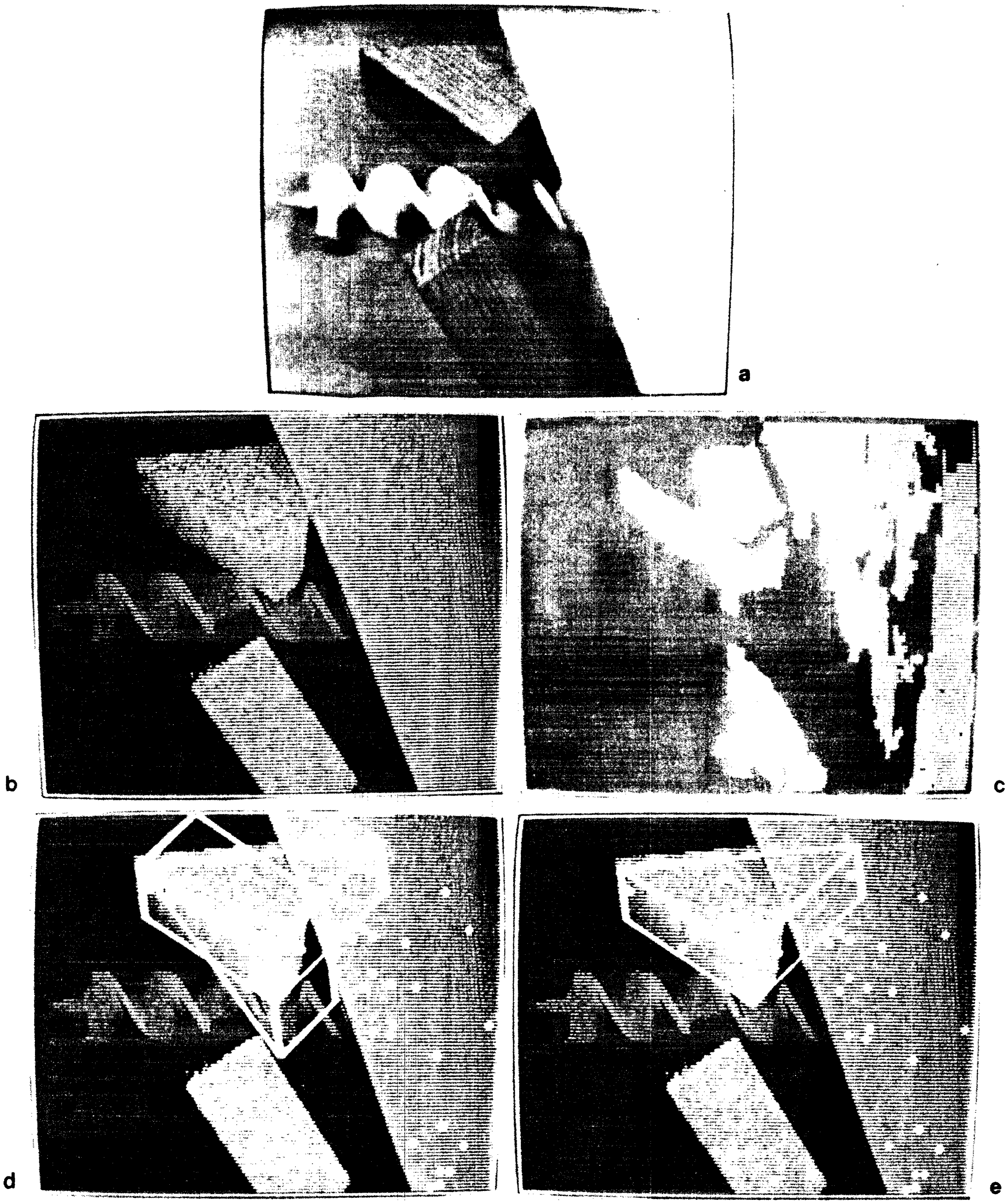

Figure 10. Three dimensional range data with partial verification. (a) Original scene (b) range data where brightness encodes height, (c) planar patches with representative points, (d) legal interpretations superimposed on range data (filled in circles are data points accounted for), and (e) interpretations that pass verification.

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.

5. Coupled Constraints

As we noted earlier, the decoupled constraints typically prune most of the non-symmetric interpretations of the data, but they are not guaranteed to reject all impossible interpretations. Consider Figure 11, 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. Thus, since the points are pairwise consistent with the candidate faces, they are accepted as part of a feasible interpretation, even though clearly they are not. Using the decoupled constraints, it is only after the model test is applied to interpretations surviving pruning that all the available geometric constraint is exploited. For the case of a single object, this merely implies some inefficiency. For the case of multiple, overlapping objects, we may actually miss a correct interpretation. For example, a locally consistent (but globally inconsistent) interpretation of length M will cause us to ignore a globally consistent interpretation of length $m < M$. We would not discover our error until the model test is applied after pruning of the interpretation tree.

One solution is to interdigitate the model test with the tree generation stage. That is, whenever we reach a leaf of the interpretation tree, we apply the model test to ensure that the interpretation is globally consistent. If it is, then we update our global counter MAX, and continue. If it is not, then we continue our search with the current value of MAX. The problem with this method is that it may be computationally expensive. As we stated, the purpose of finding effective local constraints is to enable us to avoid applying an expensive model transformation, except when necessary.

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

5.1. The Coupled Constraints in Two Dimensions

Suppose we consider two edges of an object, oriented arbitrarily in sensor coordinates, as shown in Figure 12. With each edge we will associate a base point, defined by the vector $\mathbf{b}_i$, a unit tangent vector $\mathbf{t}_i$, which points along the edge from the base point, and a unit normal vector $\mathbf{n}_i$, 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

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

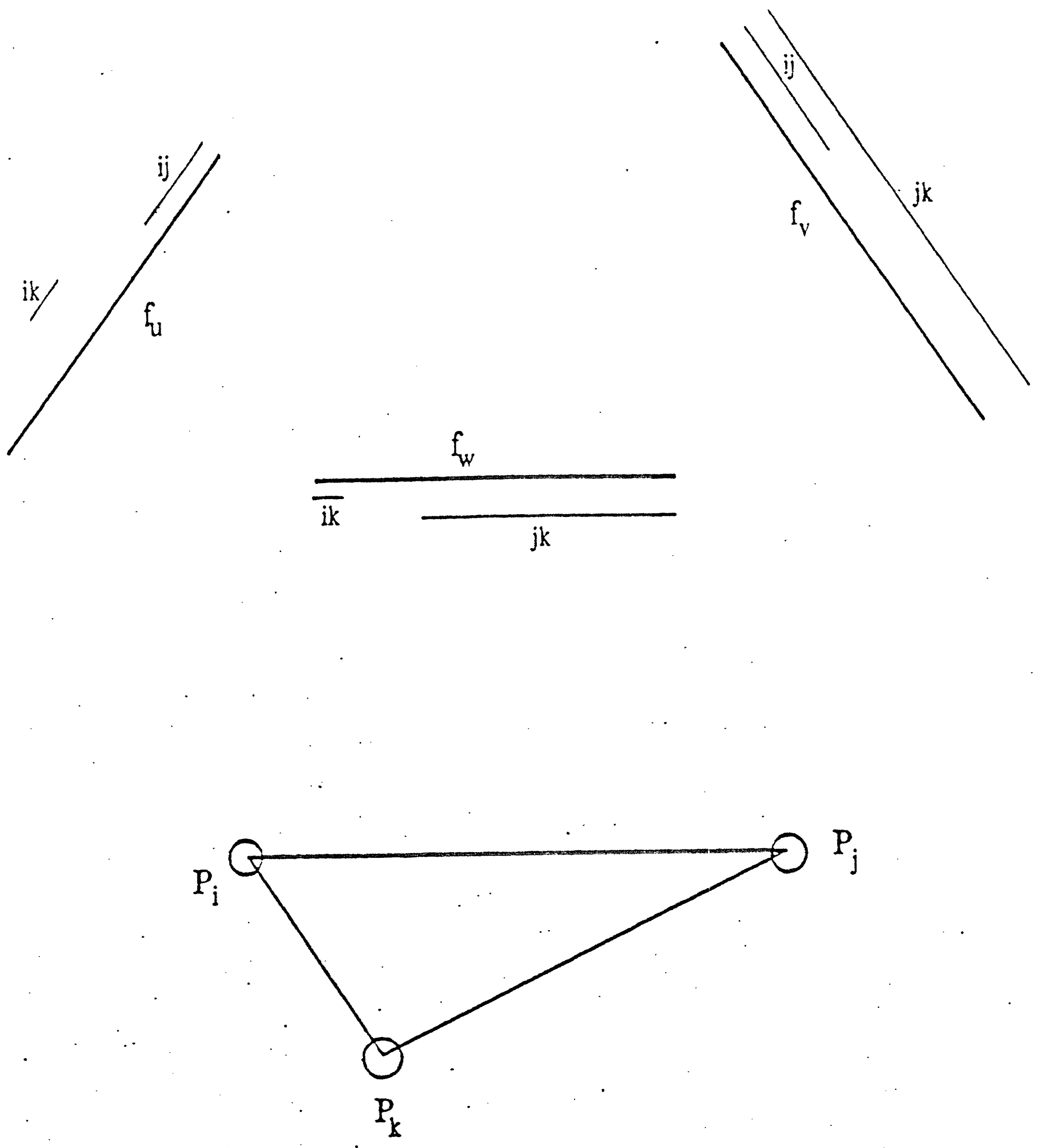

Figure 11. The constraints are decoupled. Consider matching point P_i to face f_u, point P_j to face f_v and point P_k to face f_w. These assignments are pairwise consistent, and the sections of the faces that are feasible locations for the sensed points are indicated by the sections labeled ij, etc. 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.

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

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

The vector between two small measured patches is given by

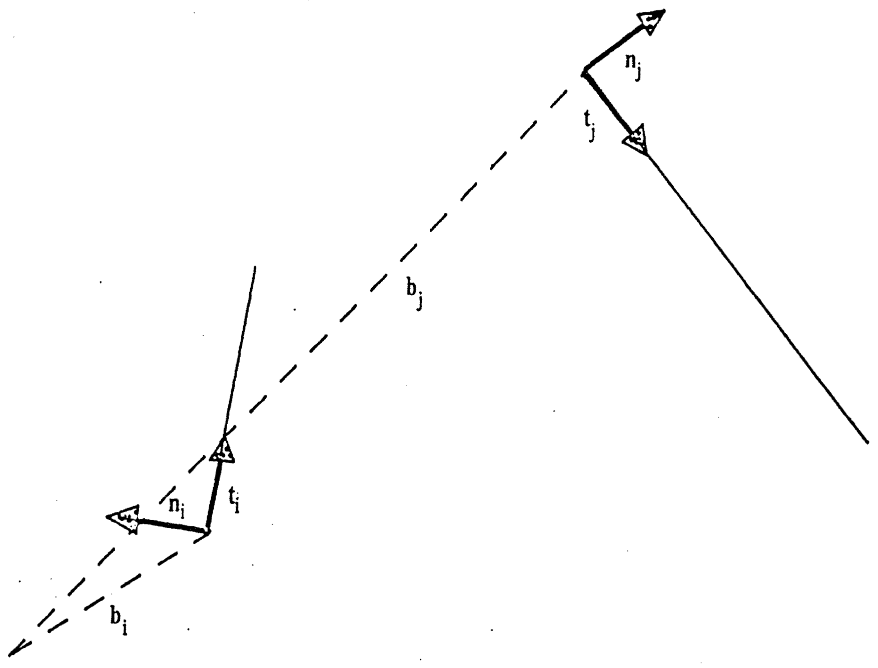

Figure 12. The constraints are recoupled. With each face, we associate a base vector $\mathbf{b}_i$, a tangent vector $\mathbf{t}_i$ and a normal vector $\mathbf{n}_i$. Then any point on a face can be represented by $\mathbf{b}_i + \alpha\mathbf{t}_i$ for some α between 0 and the length of the edge.

$$\mathbf{p}_1 - \mathbf{p}_2 = \mathbf{d}_{12} = \mathbf{b}_i + \alpha_1\mathbf{t}_i - \mathbf{b}_j - \alpha_2\mathbf{t}_j. \tag{1}$$

As in the earlier case, we know that we can measure $\mathbf{d}_{12}$. 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

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

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

We can compute

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

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

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

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

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

where ϵ can be computed straightforwardly [Grimson and Lozano-Pérez 84].

From (1) we have

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

The first term on the right is a constant and is a function of the object only, independent of its orientation. Thus, equation (2) provides us with a constraint on the value of α_2. In particular, if $\mathbf{t}_j \cdot \mathbf{n}_i = 0$, then this assignment of patches to faces is consistent only if

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

If this is true, then α_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 $\mathbf{t}_j \cdot \mathbf{n}_i \neq 0$, we have

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

Thus, we have restricted the range of possible values for α_2 and hence the set of positions for patch P_2 that are consistent with this interpretation.

Similarly, by using the estimates for $\mathbf{d}_{12} \cdot \mathbf{n}_j$ obtained from the measurements, we can restrict the range of values for α_1 and, thereby, the position of P_1.

We can also consider the coordinate-frame-independent term

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

As before, we can place bounds on the measured value for $\mathbf{d}_{12} \cdot \mathbf{t}_i$ when error in the sensory data is incorporated. Then, given a legitimate range for α_1 we can restrict the range of α_2 and vice versa. A similar argument holds for $\mathbf{d}_{12} \cdot \mathbf{t}_2$.

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 α_1 is

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

We now assign patch P_2 to face f_j, with

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

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 α_3, we find that the constraints are generally much tighter, since the legal ranges for α_1 and α_2 have already been reduced. Moreover, both α_1 and α_2 must be consistent with α_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 α_3 may in turn reduce the legal ranges for α_1 and α_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 α_i becomes empty, the interpretation can be discarded as inconsistent without further exploration.

We thus have the basis for a second recognition and localization technique. As before, we generate and prune a tree of interpretations, by assigning sensed patches to faces of an object. Here there are two types of constraints. The first is that the angle between two sensed normals, modulo error in the sensor, must be consistent with the angle between the corresponding face normals, as in the previous case. The second involves the relaxation of mutual constraints on the range of positions on a face consistent with points of contact on those faces, as described above.

5.2. Extensions to Three Dimensions

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

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

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

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

Given two points of contact, P_1 and P_2, assigned to faces f_i and f_j, the vector between them is given by

$$\mathbf{d}_{12} = \mathbf{b}_i - \mathbf{b}_j + \alpha_1 \mathbf{u}_i + \beta_1 \mathbf{v}_i - \alpha_2 \mathbf{u}_j - \beta_2 \mathbf{v}_j .$$

As before, we can measure the component of this contact vector in the direction of the surface normals recorded at each patch. This leads, for example, to an estimate for

$$\mathbf{d}_{12} \cdot \mathbf{n}_i = (\mathbf{b}_i - \mathbf{b}_j) \cdot \mathbf{n}_i - \alpha_2 (\mathbf{u}_j \cdot \mathbf{n}_i) - \beta_2 (\mathbf{v}_j \cdot \mathbf{n}_i) . \tag{4}$$

As in the previous case, the term on the left hand side of equation (4) is measurable, and given bounds on the errors in the sensor, actually defines a range

$$\mathbf{d}_{12} \cdot \mathbf{n}_i \in [\ell, h] .$$

The first term on the right hand side of equation (4) is a constant, defined independent of the coordinate system by the relationship between the two faces. Thus, equation (4) essentially reduces to a linear constraint of the form

$$\alpha_2 (\mathbf{u}_j \cdot \mathbf{n}_i) + \beta_2 (\mathbf{v}_j \cdot \mathbf{n}_i) \in [(\mathbf{b}_i - \mathbf{b}_j) \cdot \mathbf{n}_i - h, (\mathbf{b}_i - \mathbf{b}_j) \cdot \mathbf{n}_i - \ell] .$$

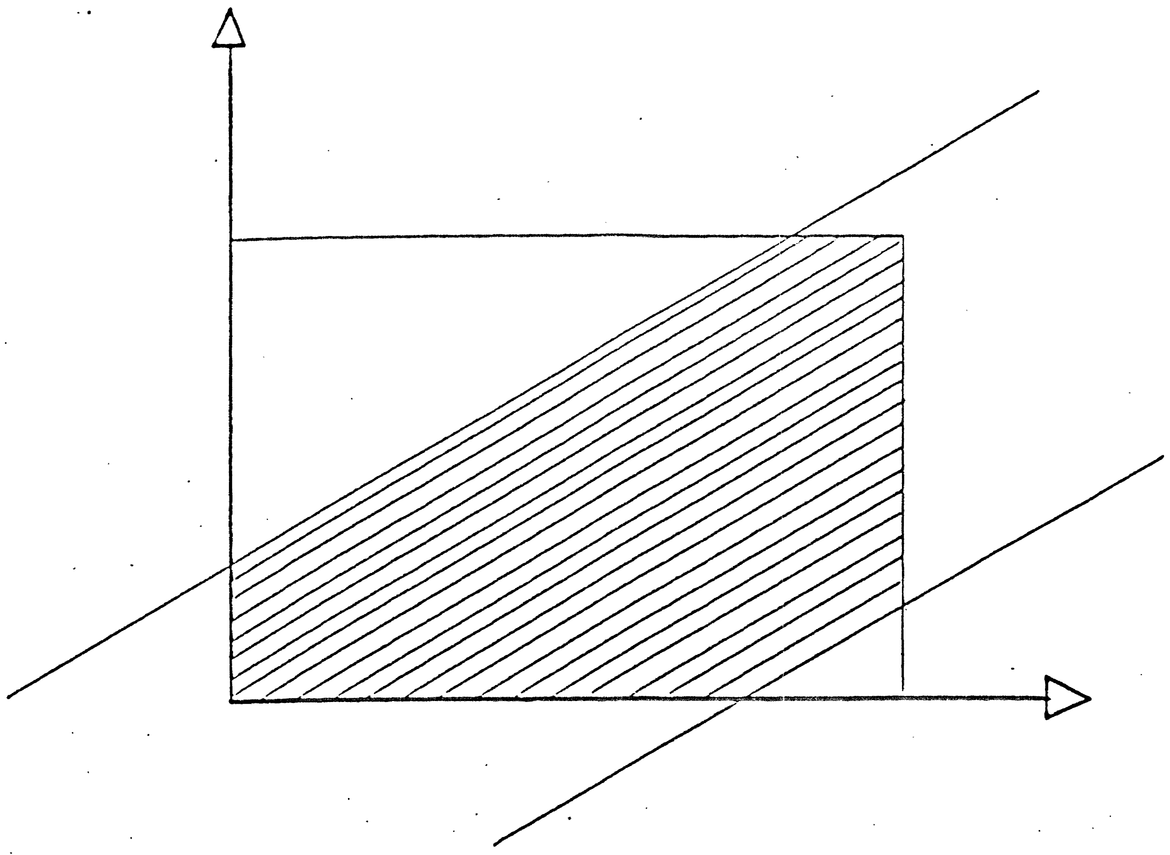

Figure 13. The intersection of legitimate face ranges for three dimensional data.

A similar expression holds for the other normal $\mathbf{n}_{j'}$.

These constraints actually describe a region in a two-dimensional space spanned by α and β, as illustrated in Figure 13. Given a current polygonal region of consistency for α and β, 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.

Clearly both the two dimensional and the three dimensional constraints developed here can be extended to deal with *null faces* in the same manner as the first algorithm.

5.3. Testing

We have tested the range propagation method on simulated data as well as on actual data. We will first describe these tests and then discuss our conclusions from these experiments.

We have done extensive testing of the algorithm with simulated two-dimensional data of the type illustrated in Figure 8. As with the results reported in Section 4.2, a number of polygons, representing the outlines of parts, were overlapped at random, with random orientation, and with random offset relative to the origin of the sensor's coordinate system. The position and orientation of a predetermined number of data points were determined by tracing along equally-spaced rays from the origin of the coordinate system, and computing the outermost intersection of those rays with the polygon boundaries. The position and normal information were then corrupted by random errors designed to simulate the effect of imperfect sensors. The number of interpretations obtained under these sensing conditions, using the propagation of ranges of feasible contact were very similar to those obtained in Section 4.2.

One of the interesting questions to consider for the range propagation technique is to what extent the implicit recoupling of the local constraints reduces the amount of explicit exploration of the interpretation tree. To test this, we ran the following set of simulations. For each of several ranges of sensing error, 100 simulations of the normal recognition and localization process were run. The number of nodes in the interpretation tree that were explicitly explored were recorded for each run. Then, using the same sensory data, a second run of 100 simulations was performed, now using the range propagation technique of section 5. Again, we recorded the number of nodes of the interpretation tree that were explicitly explored. In Table III, we record the median number of nodes explored, over the set of 100 runs, for both the normal and the range propagation techniques. Finally, we consider the ratio of the number of nodes explored using the range propagation technique, to the number of nodes explored for the same data using the normal technique. Here, we record the minimum, mean, median and maximum values for this ratio, computed over the entire 100 runs, as well as the standard deviation of this distribution of ratios. The results are shown in Table III.

We note that since the error ranges associated with each constraint differ between the normal constraints and the coupled constraints, it is possible for the coupled constraints to actually be less effective in removing portions of the interpretation tree. This is especially noticeable for large values of error in the surface normal. Overall, we see from the results shown in Table III, that the average number of nodes explored in the interpretation tree is not significantly reduced from the normal method. Given the additional overhead associated with computing and intersecting the ranges of feasible positions along edges, it may not be worth while to use the range propagation method. We note that these results may differ when considering objects whose faces do not all form right angles with one another. To test this, we also ran a smaller set of simulations with a randomly constructed object, shown in Figure 11. The results, summarized in Table III suggest that this is not a critical factor.

6. Extensions

In this paper we have described a framework for a class of recognition algorithms. We considered two major variations depending on the class of constraints employed; some minor variations were employed in dealing with different types of sensors, notably grey-scale edges and laser range data. These variations, however, share many common assumptions as to the structure of the search for consistent matchings. We have assumed, for example, that we match some subset of the data elements against all the model elements at once; that we obtain all (longest) consistent interpretations; that the objects have comparable number of degrees of freedom as the measurements. Beyond these algorithmic assumptions, we have preserved some assumptions about our domain. We have assumed, for example, that the data is made up of simple local measurements such as surface patches; that the model is made up of planar faces; that the dimensions of the objects are fixed and known *a priori*. All of these assumptions can be relaxed while retaining the characteristic flavor of the approach presented here. In this section we briefly explore these extensions. We have implemented all of these extensions with relatively minor modifications to the program code.

6.1. Partial Models – Features?

The size of the search space given an object with n faces (and the null face) and s sensed patches is $(n+1)^s$. We can reduce the size of the space by reducing the number of sensed patches and/or the number of model faces. If the input data is relatively dense, as it is for grey-scale edges, one interesting approach is to define small subsets of the model faces that are distinctive and match them against all the available data patches. Because of the non-linear nature of the increase in search space size, it is usually more efficient to do two matches with half as many faces than a single match with all the faces. This may not be true, however, in cases where the pruning constraints are very efficient.

Having found one match for a partial model, the rest of the model can be used to predict where other faces may be found. The model test can be used to verify these predictions. One difficult problem encountered here involves deciding what counts as positive and negative evidence for evaluating a hypothesis and how to combine this evidence to accept or reject a hypothesis. These decisions depend strongly on how reliable the sensor is and *a priori* expectations about the data.

The use of a partial model gains some of the combinatorial advantages of feature-based systems without requiring preprocessing of the data. Of course, this approach also inherits some of the disadvantages of feature-based systems. In particular, if the data is very sparse then there is a risk that none of the data corresponds to the selected model subsets.

Figure 14 shows some examples of matching edge data to partial models.

6.2. Choosing the first interpretation

Most approaches to recognition seek a single interpretation of the data, either the first consistent one or the best under some measure. In contrast, the approach to recognition described in this paper has opted for finding all longest consistent interpretations of the data. This choice was made both for reasons of generality and to avoid having to define an absolute measure for "goodness of match". This approach may require considering substantially more pairings of data and model elements than an approach that settled for the first acceptable match.

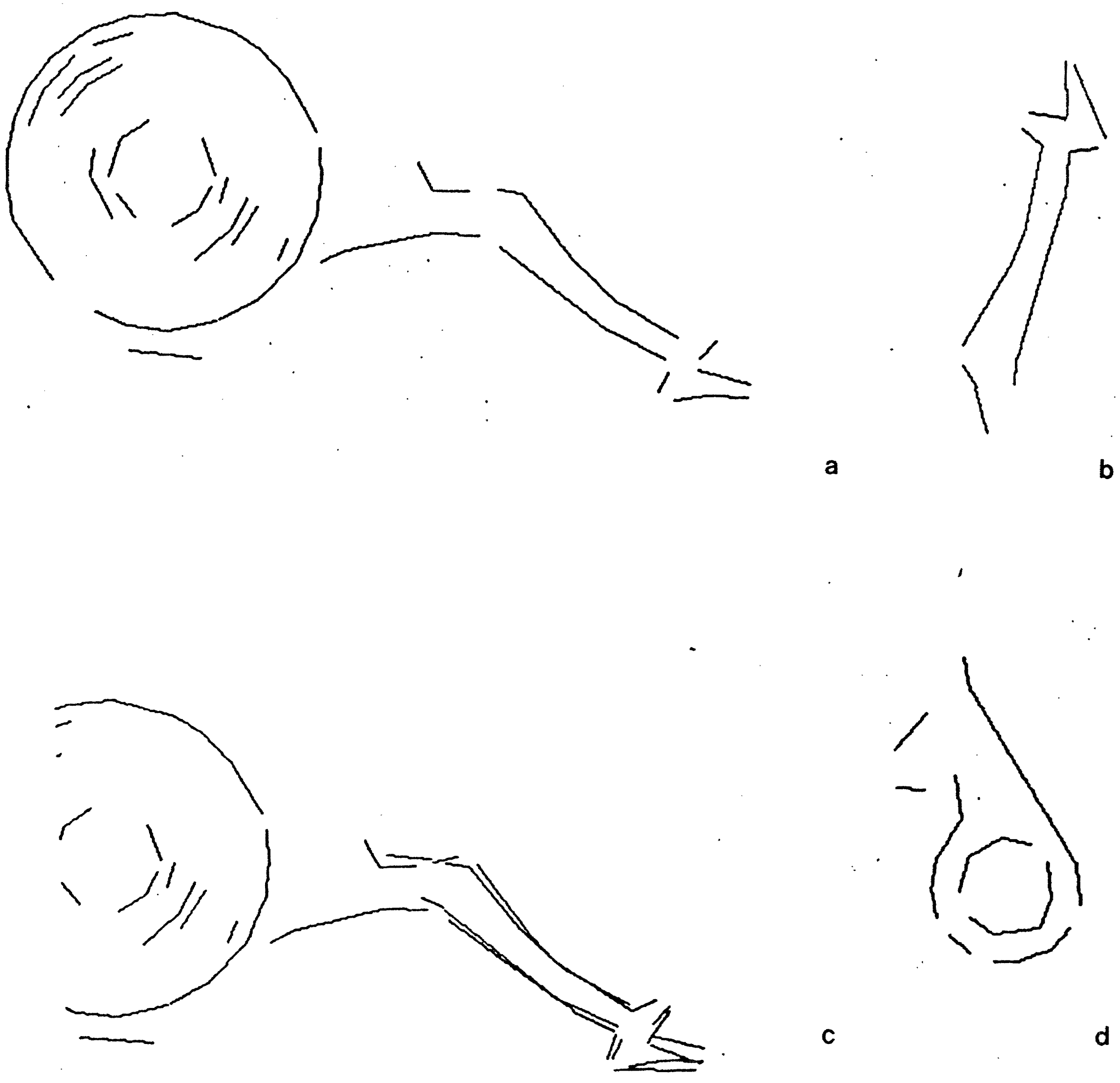

Figure 14. Examples of matches to partial models in grey-scale edges. Part (a) shows a set of grey-level edge fragments. Part (b) shows a partial model and part (c) shows the interpretation of that partial model, both in isolation, and overlapped with the original data. Part (d) shows a second partial model, for which no interpretation was found.

We have experimented with a variant of the recognition algorithm that returns the first interpretation longer than a specified threshold that passes the model test. The difference in the number of search tree nodes examined for this variant compared to the one that returns all interpretations is shown in Table IV. The table also indicates the number of times that the modified algorithm failed to find the correct interpretation while the original algorithm did find it. A sample of the data used in these tests is shown in Figure 8.

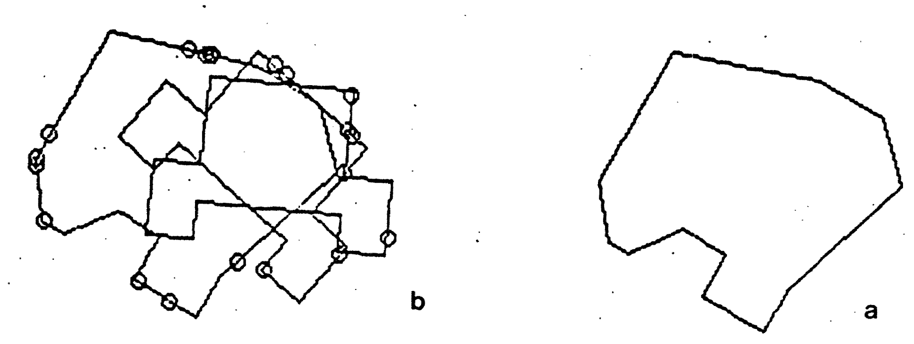

Figure 15. Examples of data used when choosing the first acceptable interpretation. Part (a) shows a random, non-symmetric part. Part (b) shows an example of data taken from a number of overlapped parts, similar to Figure 8.

We note that in the data reported for this case, we used a slightly different sensing technique. Each of the objects was rotated by some arbitrary amount, and then displaced by an arbitrary vector, whose magnitude was less than some bound. Then a predefined number of equally spaced rays were traced from the origin of the coordinate system, and the outermost intersection of each ray with any of the objects was taken as a sensed point. The position and orientation at this point was then corrupted by noise, and these corrupted points formed the input to the recognition algorithm.

It can be seen from the data of Table IV that while the number of nodes actually explored in this first-acceptable mode is significantly reduced, it appears to be at the expense of finding the correct answer. This is with a relatively high cutoff on the number of matches needed for an acceptable interpretation, in this case, 10 out of 20. Clearly, as we reduce this cutoff, the number of nodes explored will decrease, but the number of incorrect interpretations is liable to rise further.

There are several factors that can lead to the high percentage of incorrect interpretations. One possibility is that the symmetric nature of the object used leads to incorrect interpretations. To test this, we ran a similar set of simulations of the sort illustrated in Figure 15, using a non-symmetric object. The results, indicated in Table IV, show that even in this case, a significant number of incorrect interpretations can still occur. This suggests that partial symmetries are not the only cause of incorrect interpretations, when terminating on the first acceptable interpretation. The basic problem is that the length of an interpretation is not a reliable indicator of quality of match. Many of the matches, for example, may be from one part of the object and leave another part unconstrained.

6.3. Constrained degrees of freedom

We have assumed that if the objects are constrained to lie on a plane, then the data on each face is two-dimensional, and if the objects are completely unconstrained in position and orientation then the data is 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.

6.4. 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 [Bolles and Cain 82].

6.5. Curved Objects

We have indicated that our algorithm is applicable to recognizing curved objects, approximated as a set of planar faces. The tree-pruning part of the algorithm can be extended readily to recognizing objects composed of planar and curved faces from sparse point-like measurements. The decoupled constraints described in Section 3 merely compare measured values to the ranges of distances and dot–products of normals between faces. These ranges can be computed for curved faces; the main difference is that one gets a range of dot products between normals, instead of a single dot product as with planar faces.

The difficulty comes when we try to solve for the position and orientation of the object given a candidate interpretation. The problem is that normal measurements are coupled to position on the face. Solving for orientation and position requires, in general, solving a numerical optimization problem. This process is slow and error prone. In fact, planar approximations to the object model are probably the best way to get a good initial guess for the numerical solution. Having found this solution, it is questionable whether the possible improvement is worth the cost of the numerical iteration.

We note that the examples of grey level edges, shown in Figures 1 and 2, illustrate the robustness of the technique in the presence of curved objects. The original object contains a number of slowly and rapidly curving edges. The model was constructed by sensing the object in isolation, and fitting straight line segments to the recorded grey level edges (zero crossings of a Laplacian of a Gaussian operator). As a consequence, the model is really an approximation to the actual object, as can be seen in Figure 1. When sensing the object in different positions, there is no guarantee that the same linear approximation to the edges will be obtained, and in fact, our empirical experience is that the straight line approximations obtained under different sensing conditions are almost always somewhat different. Nonetheless, the recognition technique is robust enough to treat the differences as additional noise, and still recognize the object, as shown in Figure 1.

6.6. Free Parameters

We have assumed, throughout this paper, that models are metrically accurate, so that measured dimensions corresponded to model dimensions. This might not 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 general recognition 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.

The application of this method to global parameters such as scale is straightforward. It might not be so evident that this also applies to model parameters such as the distance or angle between two faces. The distance and angle range information between faces will be expressed as functions of the model parameters. Applying the method above requires us to be able to invert these functions to solve for the range of legal parameters on the basis of tentative matches between data elements and model elements. The more complicated the relationship between model elements, the more difficult this is to do. In particular, angular variation requires solving trigonometric equations.

We have extended the recognition algorithm straightforwardly to allow for a linear scale factor, as illustrated in Figure 16. 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 the case of a known one. Also, as shown in Figure 16, 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.

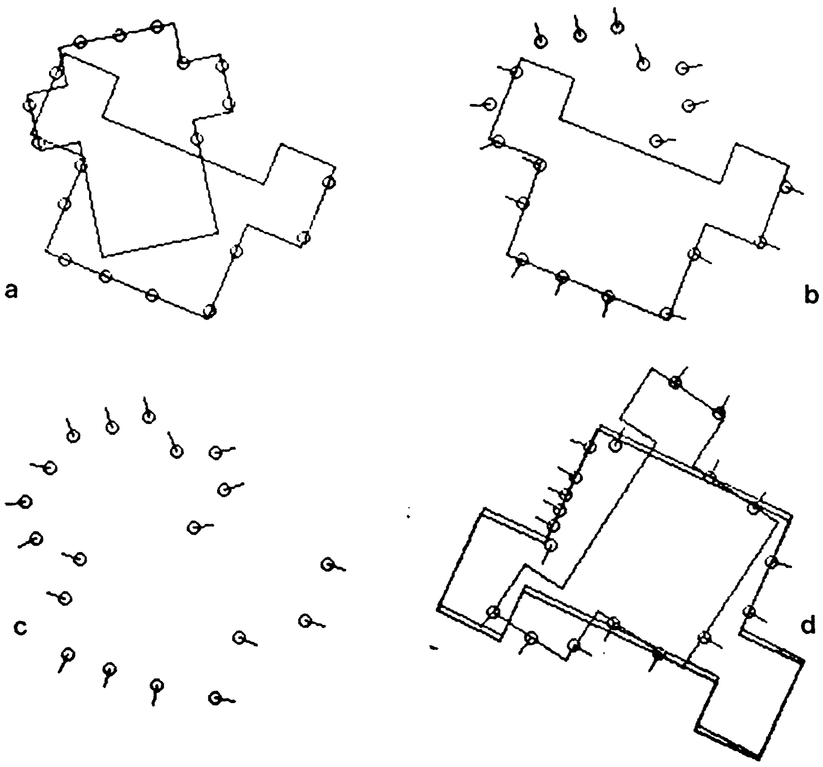

Figure 16. Examples of parameterized two-dimensional models. A scaled version of one of the models is intermixed with another model. The recognition algorithm correctly identifies the object, and determines its scale factor as well as its position and orientation. Part (a) shows a set of sampled data, and part (b) shows the interpretation of that data. Part (c) shows a second set of sampled data. Part (d) indicates that several interpretations of the data may be feasible.

7. Recognition as Search: Discussion

In our view of recognition as a search for a match between data elements and model elements, the crucial questions are the size of the search space and the constraints present between assignments. Much of the variation between existing recognition schemes can be accounted for by the choice of descriptive tokens to match. Some methods rely on computing a few very distinctive descriptors (*features*) that sharply constrain the identity and/or location of the object. Others use less distinctive descriptors and rely more on the relationships between them to effect recognition and localization.

The use of a few distinctive features sharply constrains the size of the search space. The resulting interpretation tree is very narrow and the search process can be very efficient. As an extreme example, to recognize a soft drink can from visual data, we could process the image to obtain the UPC bar code, which would uniquely identify the type of can. Moreover, knowing the position of the UPC code on the can and in the image would allow us to determine the position and attitude of the can in the scene. Of course, not all features will be as distinctive as a UPC code. Simpler examples might include corners, holes, notches and other local features. The idea is that very few such distinctive features should be needed to identify the object, and the search space can be effectively collapsed. Examples of techniques in this vein include the use of a few extended features [Perkins 78, Ballard 81], or the use of one feature as a focus, with the search restricted to a few nearby features [Tsuji and Nakamura 75, Holland 76, Sugihara 79, Bolles and Cain 82, Bolles, Horaud and Hannah 83].

The approach of using a few distinctive descriptors is common to many commercial systems (see, for example, [Bausch and Lomb 76, Gleason and Agin 79, Machine Intelligence Corporation 80, Reinhold and Vanderbrug 80]). These systems characterize both the measurements and models by a vector of global properties of binary images, such as area, perimeter, elongation and Euler number. Because of their global support, these descriptions do not extend well to overlapping or occluded parts.

Another type of recognition method relies on building elaborate high-level descriptions of the measured data before matching to the model. These approaches also rely on reducing the size of the search space by matching on a few distinctive descriptors. Examples of this approach include, for example, [Nevatia 74, Nevatia and Binford 77, Marr and Nishihara 78, Brooks 81, Brady 82].

Approaches that rely on a few distinctive features have some weaknesses. First, the cost of the search has been greatly reduced, but at the expense of global preprocessing of the sensory data. Sensors do not provide distinctive descriptors directly, the descriptors must be computed from the mass of local data provided by the sensor. In some sensing modalities, such as tactile sensing, searching for data to build distinctive descriptors can be very time consuming. Second, heavy reliance on a few features can make recognition susceptible to measurement noise. If the imaging device is out of focus, for example, so that the image of the UPC bar code is blurred significantly, recognition may be altogether impossible. In this case, degradation in the presence of error is not graceful. Third, useful features are by definition sparse or they cease to be distinctive; this sparsity may be a problem when dealing with occlusion. In our UPC code example, if some other object occludes the UPC bar code from the sensor, we will not be able to recognize the can. This may occur even though virtually all of the rest of the can is available to the sensor.

An alternative approach to recognition relies more on the geometric relationships between simpler descriptors, rather than on a few distinctive features. The idea is that these descriptors are densely distributed and not particularly distinctive taken individually, for example, surface normals fit into this category. In these circumstances, the search space is large and constraints to prune it are critical. While the size of the search space explored by these methods will be larger than in the feature-based methods, the expectation is that the individual tests are very efficient. Representative examples of such schemes include [Horn 83, Horn and Ikeuchi 83, Ikeuchi 83, Faugeras and Hebert 83, Gaston and Lozano-Pérez 84, Grimson and Lozano-Pérez 84, Stockman and Esteva 84, Brou 84]

The key difference between matching on these low-level descriptors and on distinctive features lies in the availability of descriptors. The simpler sensor measurements are likely to be dense over the object. As a consequence, recognition schemes based on such simple measurements should be applicable to sparse sensors, and should be less sensitive to problems of occlusion and sensor error, since an input description can always be obtained and matched to the model. In this paper, we explore a recognition scheme that uses very simple sensor primitives that can be computed over the entire object. We rely on the power of geometric constraints to keep the combinatorics of the search process reasonably controlled.

8. Summary

We have presented a recognition technique based on a search for consistent matches between local geometric measurements and model faces. The technique offers a number of advantages: it is very simple yet efficient; it can operate on sparse data; it is applicable to a wide range of sensors and choice of features; it degrades gracefully with error. In addition to the advantages of the particular technique, the framework within which it has been developed has proven useful both to analyze expected performance of this method and to model a number of other methods. In summary, we believe the approach described here represents a useful tool in a wide variety of recognition situations.

Acknowledgments

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 Keith Nishihara and Noble Larson. We thank Philippe Brou for kindly providing the laser ranging system with which we obtained the data reported in Figures 3 and 4. We also thank Dan Huttenlocher for his comments on an earlier draft.

References

Ballard, D. H. 1981. Generalizing the Hough transform to detect arbitrary shapes. *Pattern Recognition* 13(2):111–122.

Bausch and Lomb. 1976. Bausch and Lomb Omnicon Pattern Analysis System. Analytic Systems Division brochure. Rochester, New York: Bausch and Lomb.

Bolles, R. C., and Cain, R. A. 1982. Recognizing and locating partially visible objects: The Local-Feature-Focus method. *Int. J. Robotics Res.* 1(3):57–82.

Bolles, R. C., Horaud, P., and Hannah, M. J. 1983. 3DPO: A three-dimensional part orientation system. Paper delivered at First International Symposium of Robotics Research, Bretton Woods, N.H. (Also in Robotics Research: The First International Symposium, edited by M. Brady and R. Paul, MIT Press, 1984, pp. 413–424.)

Brady, M. 1982. Smoothed local symmetries and frame propagation. *Proc. IEEE Pattern Recog. and Im. Proc.*.

Brooks, R. 1981. Symbolic reasoning among 3-dimensional models and 2-dimensional images. *Artificial Intell.* 17:285–349.

Brou, P. 1984. Using the gaussian image to find orientations of objects. *Int. J. Robotics Res.* 3(4):89–125.

Drumheller, M. 1984. Robot Localization Using Range Data. S. B. Thesis, Dept. of Mechanical Engineering, MIT. (see also MIT AI Lab Memo 826, Mobile Robot Localization Using Sonar.)

Faugeras, O. D., and Hebert, M. 1983 (Aug., Karlsruhe, W. Germany). A 3-D recognition and positioning algorithm using geometrical matching between primitive surfaces. *Proc. Eighth Int. Joint Conf. Artificial Intell.* Los Altos: William Kaufmann, pp. 996–1002.

Faugeras, O. D., Hebert, M., and Pauchon, E. 1983 (June, Washington DC). Segmentation of range data into planar and quadratic patches. *Proc. CVPR'83.*

Gaston, P. C., and Lozano-Pérez, T. 1984. Tactile recognition and localization using object models: The case of polyhedra on a plane. *IEEE Trans. Pattern Anal. Machine Intell.* PAMI-6(3):257–265.

Gleason, G., and Agin, G. J. 1979 (Mar). A modular vision system for sensor-controlled manipulation and inspection. *Proc. Ninth Int. Symp. Industrial Robots.* Dearborn, Mich.: Society of Manufacturing Engineers, pp. 57–70.

Grimson, W. E. L., 1984. The combinatorics of local constraints in model-based recognition and localization from sparse data. MIT AI Lab Memo 763.

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

Holland, S. W. 1976 (Feb.) A programmable computer vision system based on spatial relationships. General Motors Publ. GMR-2078. Detroit: General Motors.

Horn, B. K. P. 1983. Extended Gaussian images. AIM-740. Cambridge, Mass.:M I T Artificial Intelligence Laboratory.

Horn, B. K. P., and Ikeuchi, K. 1983. Picking parts out of a bin. AIM-746. Cambridge, Mass.:M I T Artificial Intelligence Laboratory.

Ikeuchi, K. 1983. Determining attitude of object from needle map using extended gaussian image. AIM-714. Cambridge, Mass.:M I T Artificial Intelligence Laboratory.

Machine Intelligence Corporation. 1980. Model VX-100 machine vision system. Product description. Mountain View, Calif.: Machine Intelligence Corporation.

Marr, D. 1982. *Vision.* San Francisco:W. H. Freeman and Company.

Marr, D. and Hildreth, E. C. 1980. Theory of edge detection. *Proc. R. Soc. Lond. B* 207:187–217.

Marr, D., and Nishihara, H. K. 1978. Representation and recognition of the spatial organization of three-dimensional shapes. *Proc. R. Soc. Lond. B* 200:269–294.

Nevatia, R. 1974. Structured descriptors of complex curved objects for recognition and visual memory. Ph.D. thesis, Stanford University. AIM 250. Stanford, Calif.: Stanford University Artificial Intelligence Laboratory.

Nevatia, R., and Binford, T. O. 1977. Description and recognition of curved objects. *Artificial Intell.* 8:77–98.

Perkins, W. A. 1978. A model-based vision system for industrial parts. *IEEE Trans. Comput.* C-27:126–143.

Reinhold, A. G., and VanderBrug, G. J. 1980. Robot vision for industry; the autovision system. *Robotics Age*, Fall, pp. 22–28.

Stockman, G., and Esteva, J. C. 1984. Use of geometrical constraints and clustering to determine 3D object pose. TR84-002. East Lansing, Mich.:Michigan State University Department of Computer Science.

Sugihara, K. 1979. Range-data analysis guided by a junction dictionary. *Artificial Intell.* 12:41–69.

Tsuji, S., and Nakamura, A. 1975 (Aug., Cambridge, Mass.). Recognition of an object in a stack of industrial parts. *Proc. Fourth Int. Joint Conf. Artificial Intell.* Los Altos: William Kaufmann, pp. 811–818.

Tables

Table I – No. of Interpretations For Multiple Objects – 2D					
Object	Angle	Dist	Min	50th	95th
Housing Top	$\pi/10$	.01	1	1	3
		.05	1	1	2
		.10	1	1	5
	$\pi/5$	.01	1	1	7
		.05	1	1	5
		.10	1	2	6
	$\pi/4$	.01	1	1	4
		.05	1	2	4
		.10	1	2	14

Table I illustrates statistics in the performance of the localization process for three overlapping two-dimensional objects, two of which are not the object of interest. Each row lists parameters of a histogram of the number of interpretations consistent with the local constraints, based on 100 trials with the object randomly oriented with 3 degrees of freedom. In the table, the *angle* column lists the radius of the error cone about the measured surface normal; the *dist* column lists the error range of the distance sensing. The *min* column list the minimum number of interpretations observed without a model test; the *50th* column lists the median point of the set of simulations with a model test and clustering; the *95th* column lists the 95^{th} percentile of the set of simulations with a model test and clustering. As well, the object was overlapped with several other objects at random, as given by the Objects column. In each case, 20 sensory data points were used.

Table II – No. of Interpretations For Multiple Objects – 3D						
Object	Angle	Dist	Min	50th	95th	Missed
Housing	$\pi/10$	.03	1	2	9	.14
		.07	1	2	13	.07
		.15	1	5	44	.09
	$\pi/5$	.03	1	5	57	.02
		.07	1	7	47	.03

Table II illustrates statistics in the performance of the localization process for overlapping three-dimensional objects. Each row lists parameters of a histogram of the number of interpretations consistent with the local constraints, based on 100 trials with the object randomly oriented with 6 degrees of freedom. As well, the object was overlapped with a second object at random. In each case, 15 sensory data points were used. The *missed* column reports the percentage of cases in which no interpretation was found, due to the separation of the model test from the application of the constraints; other columns are as in Table I.

Table III – No. of Nodes – Regular vs Range Propagation									
Obj	Angle	Dist	Med	Med	Min	Mean	Med	Max	Dev
Lip	$\pi/10$	0.10	3630	3382	.334	.949	.908	2.479	.281
		0.05	2802	2528	.382	.915	.938	1.428	.157
		0.01	2351	2194	.199	.925	.924	2.286	.218
	$\pi/20$	0.10	2457	2323	.217	.954	.950	1.593	.196
		0.05	2044	1896	.488	.937	.948	1.318	.101
		0.01	2022	1729	.574	.954	.957	1.303	.094
Rand	$\pi/10$	0.10	5276	3738	.048	.805	.738	3.757	.513
		0.05	3761	2682	.237	.744	.746	2.328	.231
		0.01	2642	1961	.164	.730	.716	2.070	.208
	$\pi/20$	0.10	1646	1360	.154	.863	.878	1.521	.227
		0.05	1251	1050	.155	.841	.868	1.405	.188
		0.01	1257	1081	.187	.872	.863	1.587	.180

Table III compares statistics in the performance of the normal localization process, with the process in which coupled constraints are used to propagate ranges of possible positions. Each row lists parameters of a histogram, based on 100 trials with the object randomly oriented with 3 degrees of freedom. As well, the object was overlapped with another object at random. In each case, 20 sensory data points were used. In the table, the *angle* column lists the angle of the error cone about the measured surface normal; the *dist* column lists the error range of the distance sensing. The next two columns list the median number of nodes actually explored in the interpretation tree for the normal case and for the range propagation case, respectively, taken over the 100 simulations. The following four columns list the minimum, mean, median and maximum obtained by taking the ratio of the number of nodes explored in the propagation case over the number of nodes explored in the normal case. The final column indicates the standard deviation of this histogram of ratios. The process was run using the *lip* and *random* objects illustrated in Figures 15.

Table IV – No. of Nodes – Regular vs First Acceptable												
Obj	Angle	Dist	Min	Med	Max	Min	Med	Max	Min	Med	Max	Miss
Lip	$3\pi/20$	0.10	104	3516	15811	9	16	5358	0.004	0.01	0.34	0.44
		0.05	165	3581	12462	9	274	5113	0.007	0.08	0.46	0.34
		0.01	317	2912	15386	9	414	7795	0.009	0.14	0.56	0.21
	$2\pi/20$	0.10	287	2930	10092	9	172	5110	0.007	0.06	0.51	0.38
		0.05	196	2682	9541	9	137	4582	0.009	0.05	0.60	0.21
		0.01	317	2462	9129	10	466	5754	0.010	0.19	0.72	0.11
	$\pi/20$	0.10	186	2178	8614	9	112	3149	0.009	0.05	0.42	0.23
		0.05	249	1919	10302	10	337	4786	0.011	0.18	0.56	0.12
		0.01	243	1759	6817	9	358	3291	0.011	0.20	0.51	0.04
Rand	$\pi/10$	0.10	91	5285	23943	9	1326	9013	0.005	0.25	0.47	0.66
		0.05	91	3634	17176	10	1790	6965	0.011	0.47	0.57	0.22
		0.01	134	2713	9043	11	1635	6896	0.027	0.56	0.76	0.04
	$\pi/20$	0.10	182	1946	6408	9	787	2664	0.034	0.42	0.60	0.01
		0.05	481	1249	4104	10	586	2182	0.021	0.45	0.56	0.00
		0.01	402	1353	5301	11	416	2218	0.027	0.32	0.42	0.00

Table IV compares statistics in the performance of the normal localization process, and the process in which the technique is terminated once an interpretation with a sufficient number of matched points is obtained. Each row lists parameters of a histogram, based on 100 trials with the object randomly oriented with 3 degrees of freedom. As well, the object was overlapped with one other object at random (similar results hold for more than one occluding object). In each case, 20 sensory data points were used. In the table, the *angle* column lists the angle of the error cone about the measured surface normal; the *dist* column lists the error range of the distance sensing. The next three columns list the minimum, median and maximum number of nodes actually explored in the interpretation tree for the normal case, taken over the 100 simulations. To test the effects of terminating the localization process, a cutoff of 10 matched points was used. The next three columns list the minimum, median and maximum number of nodes actually explored when the process was terminated in this manner. The next three columns list the minimum, median and maximum obtained by taking the ratio of the number of nodes explored in the truncated case over the number of nodes explored in the normal case. Finally the last column lists the percentage of cases in which terminating the process after 10 points were matched, caused an incorrect interpretation to be accepted. The process was run using the *lip* and *random* objects illustrated in Figure 15.