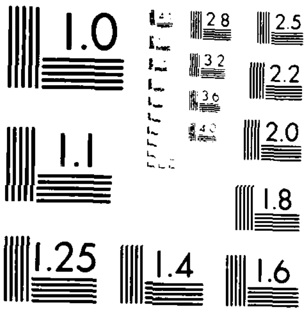
AD-A135 791    MODEL-BASED RECOGNITION AND LOCALIZATION FROM SPARSE    1/1
RANGE OR TACTILE DAT..(U) MASSACHUSETTS INST OF TECH
CAMBRIDGE ARTIFICIAL INTELLIGENCE L..
UNCLASSIFIED    E E GRIMSON ET AL. AUG 83 AI-M-738    F/G 6/4    NL

END
DATE
FILMED
1-84
DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

2

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

AD-A135791

## REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER
AI M[illegible]

2. GOVT ACCESSION NO.
A135791

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)
Model-Based Recognition and Localization from Sparse Range or Tactile Data

5. TYPE OF REPORT & PERIOD COVERED
Memorandum

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)
W. Eric Grimson and Tomas Lozano-Perez

8. CONTRACT OR GRANT NUMBER(s)
N00014-80-C-0505
N00014-82-K-0334

9. PERFORMING ORGANIZATION NAME AND ADDRESS
Artificial Intelligence Laboratory
545 Technology Square
Cambridge, Massachusetts 02139

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS

11. CONTROLLING OFFICE NAME AND ADDRESS
Advanced Research Projects Agency
1400 Wilson Blvd
Arlington, Virginia 22209

12. REPORT DATE
August 1983

13. NUMBER OF PAGES
46

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)
Office of Naval Research
Information Systems
Arlington, Virginia 22217

15. SECURITY CLASS. (of this report)
UNCLASSIFIED

15a. DECLASSIFICATION DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Distribution of this document is unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

None

DEC 1 4 1983
A

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Tactile Recognition
Robotics
Range Data

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This paper discusses how local measurements of three-dimensional positions and surface normals (recorded by a set of tactile sensors, or by three-dimensional range sensors), may be used to identify and locate objects, from among a set of known objects. The objects are modeled as polyhedra having up to six degrees of freedom relative to the sensors. We show that inconsistent hypotheses about pairings between sensed points and object surfaces can be discarded efficiently by using local constraints on: distances between faces, angles between face normals, and angles (relative to the surface normals) of vectors between sensed points. We show by simulation and by mathematical bounds that the number of

DTIC FILE COPY

83 12 12 027

Block 20 Continued:

hypotheses consistent with these constraints is small. We also show how to recover the position and orientation of the object from the sense data. The algorithm's performance on data obtained from a triangulation range sensor is illustrated.

A-1

DTIC COPY INSPECTED 3

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
ARTIFICIAL INTELLIGENCE LABORATORY

A. I. Memo 738 August, 1983

# Model-Based Recognition and Localization From Sparse Range or Tactile Data

W. Eric L. Grimson
Tomás Lozano-Pérez

**Abstract.** This paper discusses how local measurements of three-dimensional positions and surface normals (recorded by a set of tactile sensors, or by three-dimensional range sensors), may be used to identify and locate objects, from among a set of known objects. The objects are modeled as polyhedra having up to six degrees of freedom relative to the sensors. We show that inconsistent hypotheses about pairings between sensed points and object surfaces can be discarded efficiently by using local constraints on: distances between faces, angles between face normals, and angles (relative to the surface normals) of vectors between sensed points. We show by simulation and by mathematical bounds that the number of hypotheses consistent with these constraints is small. We also show how to recover the position and orientation of the object from the sense data. The algorithm's performance on data obtained from a triangulation range sensor is illustrated.

**Acknowledgements.** 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.

© Massachusetts Institute of Technology 1983

# 1. The Problem and the Approach

A central characteristic of advanced applications in robotics is the presence of significant uncertainty about the identities and positions of objects in the workspace of the robot. It is this characteristic that makes sensing of the external environment an essential component of robot systems. The process of sensing can be loosely divided into two stages: first, the measurements of properties of the objects in the environment, and second, the interpretation of those measurements. In the present paper, we will concentrate on the interpretation of sensory data. In investigating this problem, we make only a few, simple assumptions about available sensory measurements, rather than considering specific details of a particular sensor. As a consequence, the interpretation technique that is developed here should be applicable to a wide range of sensing modalities. As well, the interpretation technique may have implications for the design of three-dimensional sensors.

## 1.1. Problem Definition

The specific problem we consider in this paper is to identify an object from among a set of known objects and to locate it relative to the sensor. The object sensed is assumed to be a single, possibly non-convex, polyhedral object (for which we have an accurate geometric model). The object may have up to six degrees of freedom relative to the sensor (three translational and three rotational). The sensor, which could be tactile or range, is assumed to be capable of providing three-dimensional information about the position and local surface orientation of a small set of points on the object. Each sensor point is processed to obtain:

1. Surface points — On the basis of sensor readings, the positions of some points on the sensed object can be determined to lie within some small volume relative to the sensor.
2. Surface normals — At the sensed points, the surface normal of the object's surface can be recovered to within some cone of uncertainty.

Our goal is to use local information about sensed points to determine the set of positions and orientations of an object that are consistent with the sensed data. If there are no consistent positions and orientations, the object is excluded from the set of possible objects.

In this paper we do not discuss how surface points and normals may be obtained from actual sensor data, since this process is highly sensor-dependent (for references to existing measurement methods see Section 1.3). Our aim is to show, instead, how such data may be used in conjunction with object models to recognize and localize objects. The method, in turn, suggests criteria for the design of sensors and sensor-processing strategies.

Our only assumption about the input data is that fairly accurate positions of surface points are obtainable from the sensor, but that significant errors exist in determining normal information. This assumption reflects the type of data obtainable from tactile sensors. Range sensors based on triangulation can be used to obtain high quality measurements of normals from patches of depth data. The availability of good normal data merely increases the efficiency of the method.

## 1.2. Approach

A recent paper [Gaston and Lozano-Pérez 83] introduced a new approach to tactile recognition and localization for polyhedra with three degrees of positional freedom (two translational and one rotational). The present paper generalizes that approach to polyhedra with six degrees of positional freedom. The inputs to the recognition process are: a set of sensed points and normals, and a set of geometric object models for the known objects. The recognition process, as outlined in the earlier paper, 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 point with some object surface of one of the known objects. Interpretations inconsistent with local constraints (derived from the model) on the sense data 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 sense point on an object surface. The sensed point must lie *inside* the object face, not just on the surface.

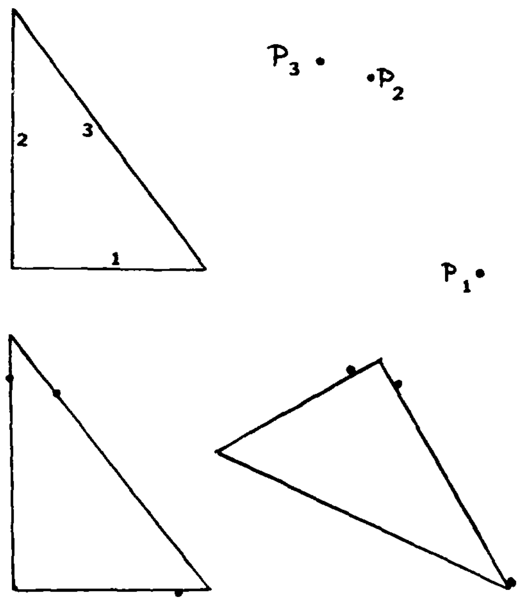
The first step is the key to this process. The number of possible interpretations given $s$ sensed points and $n$ surfaces is $n^s$. Therefore, it is not feasible to carry out a model test on all possible interpretations. The goal of the recognition algorithm is to exploit the local constraints on the sensed data so as to minimize the number of interpretations that need testing. This approach is an instance of a classic paradigm of artificial intelligence: generate and test; see for example [Buchanan, et al. 69].

Consider a simple example of the approach, illustrated in Figure 1. The model is a right triangle, with edge sizes of 3, 4, and 5 respectively. From this model, we can construct a table of ranges of distances between pairs of points on the edges. The table is as follows:

| | Distance Ranges Between Edges | | |
|---|---|---|---|
| | 1 | 2 | 3 |
| 1 | [0,3] | [0,5] | [0,4] |
| 2 | [0,5] | [0,4] | [0,3] |
| 3 | [0,4] | [0,3] | [0,5] |

Now, suppose we know the positions of the three sensed points, $P_1$ through $P_3$, shown in Figure 1. The measured distances between those points are $dist(P_1, P_2) = 3.5$, $dist(P_1, P_3) = 4.4$, $dist(P_2, P_3) = 0.8$. From this we see that any interpretation of the sensed points that assigns $P_1$ and $P_2$ both to edge 1 is inconsistent with the model. Similarly, assigning $P_1$ and $P_2$ to edges 2 and 3 is not consistent. Many other pairwise assignments of points to edges can be discarded simply by comparing the measured distances to the ranges in the table. Note that the sensed positions are

Figure 1. An example of the approach



subject to error, so that a range of actual distances is consistent with the measured positions. It is these distance ranges that must be compared against the ranges in the table. For this example, only 6 of the 27 possible assignments of the three points to the three model edges are legal.

Of the six interpretations consistent with the distance ranges, the two shown in Figure 1, are completely consistent once the line equations of the edges are taken into account. Each of these interpretations leads to a solution for the position and orientation of the triangle relative to the sensor. Furthermore, these positions and orientations of the triangle place the measured points inside the finite edges, not just on the infinite line.

This paper discusses both steps of the recognition process, focusing first on the generate step and then considering the model testing stage. We show, by mathematical analysis and by simulation, that the number of feasible interpretations can be reduced to manageable numbers by the use of local geometric constraints. In particular, we investigate the effectiveness of the different local constraints and the impact of measurement errors on their effectiveness. We further show that the few remaining feasible interpretations can efficiently be subjected to an explicit model test, generally resulting in a single interpretation of the sense data (up to symmetries). We also illustrate the performance of the algorithm on range data obtained by triangulation.

## 1.3. Three Dimensional Sensing

Sensors can be roughly divided into two categories: *non contact* and *contact*. Non contact sensing, especially visual sensing, has received extensive attention in the robotics and artificial intelligence literature. Contact sensing, such as tactile or haptic sensing, plays an equally important role in robotics, but has received much less attention. In this paper, our aim is to develop a sensory interpretation method that is applicable to data from both contact and non contact sensors.

While two-dimensional sensing, for example silhouette or binary vision, may be adequate for restricted situations such as problems with three degrees of freedom in positioning, the general localization and recognition problem requires three-dimensional sensing. Throughout this paper, we will concentrate on the six-degree of freedom recognition and localization problem and the use of three-dimensional sensing. Restrictions of the method to the simpler case of three degrees of freedom are straightforward.

### 1.3.1. Previous Work in Visual Range Sensing

The measurement stage of visual sensing has received extensive attention in the literature. Of particular interest here are methods for obtaining three-dimensional position and surface normal information; see [Jarvis 83] for a detailed survey. Possible methods include edge-based stereo systems [e.g. Grimson 81, Baker and Binford 81, Mayhew and Frisby, 81], which provide three-dimensional positions of sparse sets of points in the image. This sparse data can be used to reconstruct a dense surface representation, from which surface normals can be estimated [Grimson 82, 83; Terzopoulos 83]. Other methods for obtaining three dimensional positions are laser range-finding [e.g. Nitzan, Brain, and Duda 77, Lewis and Johnston 77] and structured light systems [e.g. Shirai and Suwa 71, Popplestone, et al. 75]. Many other visual processes can be used to obtain surface normal information directly, e.g., photometric stereo [e.g. Woodham 78, 80, 81, Ikeuchi and Horn, 79] and texture gradients [Bajcsy 73, Bajscy and Liebermann 76, Kender 80, Stevens 80]. In fact, there is no constraint that the sensory data for one problem must come from one sensory modality. Data from visual sensors and tactile sensors may be combined in one run of the algorithm.

The interpretation stage of visual recognition has received less attention, especially when dealing with three-dimensional objects with six degrees of freedom. Much of the previous work in the area of interpretation of three-dimensional data has focused on the recognition of simple generic objects such as planar patches, regular polyhedra, generalized cylinders, and spheres [e.g., Shirai and Suwa 71, Popplestone, et al. 75, Nitzan, Brain, and Duda 77, Oshima and Shirai 78, Faugeras, et al. 83, Agin and Binford 73, Nevatia and Binford 77]. Some authors have examined the problem we deal with here of recognizing specific objects from three-dimensional data [e.g., Shneier 79, Sugihara 79, Oshima and Shirai 83, Bolles, Horaud, and Hannah 83, Brou 83, Ikeuchi, et al. 83]. The principal difference between previous work on recognition and the approach described here is our reliance on *sparse* data acquired at points. This makes our approach adaptable to contact sensing as well as visual sensing. The sparseness of the data does make the problem of *segmentation*, determining which data is drawn from which objects in a scene, more difficult. Further research on this topic is currently underway.

In the final stages of preparing this paper, we became aware of the work of Faugeras and Hebert [83] which adopts an approach that is similar in many respects to the one described here. Their work, however, focuses on deriving an accurate model test. Their method does not emphasize the problem of enumerating all the legal interpretations of the data. Instead, a measure of the accuracy of the model test (and a simple angle pruning heuristic) is used to drive a best-first search for a good interpretation. This method does not ensure that the interpretation found is the only one consistent with the data, however. Their method and ours are complementary in this respect. Their approach also does not assume sparse data, but it is in fact applicable to that problem.

### 1.3.2. Previous Work in Tactile Sensing

Contact sensors measure the locus of contact and the forces generated when in contact with an object. We make the distinction between *tactile sensors*, which measure forces over small areas, such as a fingertip, and *force sensors*, which measure the resultant forces and torques on some larger structure, such as a complete gripper. A micro-switch, for example, can serve as a simple tactile sensor capable of detecting when the force over a small area, e.g. an elevator button, exceeds some threshold. The most important type of tactile sensors are the *matrix tactile sersors*, composed of an array of sensitive points. The simplest example of a matrix tactile sensor is an array of micro-switches. Much more sophisticated tactile sensors, with much higher spatial and force resolution, have been designed; see [Harmon 82] for a review and [Hillis 82, Overton and Williams 81, Purbrick 81, Raibert and Tanner 82, Schneiter 82] for some recent designs.

For descriptions of previous work in tactile sensing, we refer the reader to two very thorough surveys by Harmon [80, 82]. A more detailed discussion of previous work on tactile recognition can be found in [Gaston and Lozano-Pérez 83]. In this section, we briefly survey the two major alternative approaches to tactile recognition: statistical pattern recognition, and description-building and matching.

Much of the existing work on tactile recognition has been based on statistical pattern recognition or classification. Some researchers have used pressure patterns on matrix sensors primarily [Briot 79, Okada and Tsuchiya 77]. Others have used the joint angles of fingers grasping the object as their data [Briot, Renaud, and Stojilkovic 78, Marik 81, Okada and Tsuchiya 77, Stojilkovic and Saletic 75]. A related approach uses the pattern of activation of on-off contacts placed on the finger links [Kinoshita, Aida, and Mori 75].

The range of possible contact patterns between multiple sensors and complex objects is highly variable and seems to require detailed geometric analysis. Tactile recognition methods based on statistical pattern recognition are limited to dealing with simple objects because they do not exploit the rich geometric data available from object models.

Several proposed recognition methods build a partial description of the object from the sense data and match this description to the model. One approach emulates the feature-based descriptions in vision systems, for example, identification of holes, edges, vertices, pits, and burrs [Binford 72, Hillis 82, Snyder and St. Clair 78]. Another approach is to build surface models, either from pressure distributions on matrix sensors [Overton and Williams 81], or from the displacements of an array of needle-like sensors [Page, Pugh, and Heginbotham 76, Takeda 74]. A related approach builds a representation of an object's cross section [Ozaki et al 82, Kinoshita, Aida, Mori 75].

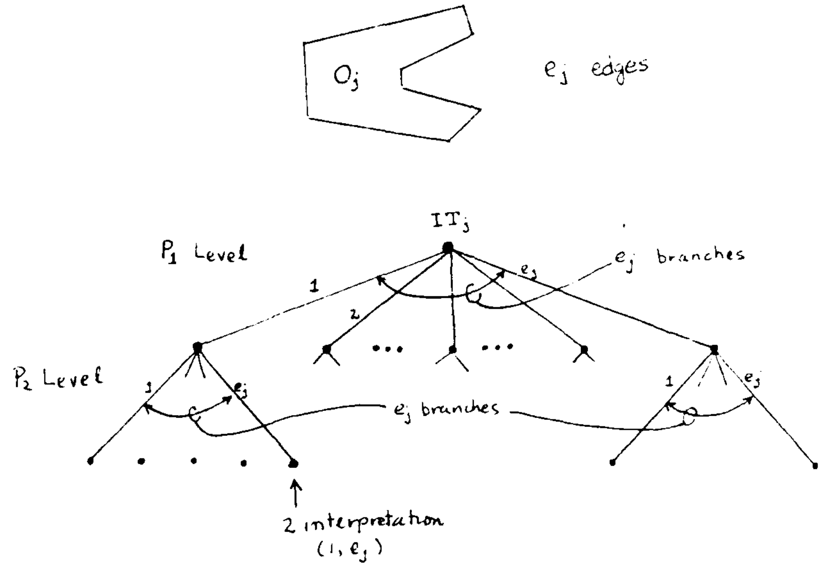
Description based methods are more general than statistical methods but must solve two formidable problems: building accurate object descriptions from tactile data, and matching the descriptions to the models. One major difficulty is that existing sensors do not have the spatial or force resolution needed to build nearly complete object descriptions. Furthermore, there are few methods for matching the partial descriptions obtainable from tactile sensors to object models. In our opinion, part of the problem in tactile data interpretation has been the tendency to adapt the techniques developed for two-dimensional vision, where dense data is readily obtainable, to tactile data, which is naturally sparse.

One lesson from the simulations described later is that some estimate of surface normal is an extremely powerful constraint on recognition and localization. The estimate need not be very tight for performance to improve drastically. There has been little previous emphasis on measuring surface normals with tactile sensors. Accuracy in measuring normals requires some attention to engineering tradeoffs in sensor design, especially the sensor stiffness. In a stiff sensor (one that deforms very little under contact), the normal to the sensor surface at the point of contact directly gives an estimate of the object's surface normal. So, a stiff sensor with high spatial resolution can be used to measure normals. In a soft sensor, the pattern of forces can be analyzed to determine the shape of the object surface. So, a soft sensor with good force measurement accuracy can also be used. Today, it is probably easier to build stiff sensors with poor force resolution than soft sensors with good force resolution [Snyder and St. Clair 78]. This argues that a stiff VLSI sensor [e.g. Raibert and Tanner 82] may be acceptable. Another factor is that the method used here, since it is based on local information, does not require large sensor areas; it can function better with many small sensors.

The approach used in this paper is an instance of a description-based recognition method. The basic departure from previous methods is the reliance on sparse three dimensional positions and surface normals obtained at *points*[1]. This contrasts with the dense *area* data needed in global feature based or surface based description methods. The point based data we use is more readily obtainable from simple tactile sensors and the process of matching it to models is relatively straightforward. Therefore, the method described here could be a powerful addition to approaches based on more complete descriptions.

[1] Very different approaches to tactile recognition based on this type of data are outlined in [Dixon, Salazar, and Slagle 79, Ivancevic 74].

Figure 2. Interpretation Tree



## 2. Generating Feasible Interpretations

After sensing an object, we have the positions of up to $s$ points, $P_i$, known to be on the surface of one of the $m$ known objects, $O_j$, having $n_j$ faces. The range of possible pairings of sensed points and model faces for one object can be cast in the form of an *interpretation tree* (IT) [Gaston and Lozano-Pérez 83]. The root node of the $\mathrm{IT}_j$, for object $O_j$, has $n_j$ descendants, each representing an interpretation in which $P_1$ is on a different face of $O_j$. There are a total of $s$ levels in the tree, level $i$ indicating the possible pairings of $P_i$ with the faces of object $O_j$ (see Figure 2). Note that there may be multiple points on a single face, so that the number of branches remains constant at all levels.

A $k$-interpretation is any path from the root node to a node at level $k$ in the IT; it is a list of $k$ pairings of points and faces. The set of IT's contains a very large number of possible $s$-interpretations

$$\sum_{j=1}^{m}(n_j)^s.$$

In an object with symmetries, of course, the IT is highly redundant [Gaston and Lozano-Pérez 83]. The $m$ IT's, one for each known object, represent the search space for the recognition problem discussed here.

### 2.1. Pruning the IT by Local Constraints

Only a very few interpretations in an IT are consistent with the input data. We can exploit the following local constraints to prune inconsistent interpretations:

1. Distance Constraint The distance between each pair of $P_i$'s must be a possible distance between the faces paired with them in an interpretation.
2. Angle Constraint The range of possible angles between measured normals at each pair of $P_i$'s must include the known angle between surface normals of the faces paired with them in an interpretation.
3. Direction Constraint The range of values for the component of a vector between sensed points $(P_i \mapsto P_j)$ in the direction of the sensed normal at $P_i$ and at $P_j$ must intersect the range of components of possible vectors between points on the faces assigned to $P_i$ and $P_j$ by the interpretation.

These constraints typically serve to prune most of the non-symmetric interpretations of the data. Other constraints are possible, for example, the area of the triangle defined by three sensed points must be contained within the range of areas defined by the faces paired with them, and the pairing of sensed points with faces must not be such as to require that the path of the sensor (beam) pass through some portion of the object before sensing that face [Gaston and Lozano-Pérez 83]. We will focus on the three constraints above, primarily because they are simple to implement while being quite effective. Moreover, they capture all the constraints between pairs of points.

Note that the distance, angle, and direction constraints can be used to prune $k$-interpretations, for $k \geq 2$, thereby collapsing whole subtrees of the IT. This is a crucial point, worth dwelling on for a moment.

Recall that the overall problem we are considering is to determine the position and orientation of an object, using sparse sensory data. In principle, one could consider all possible interpretations of the data, and for each one, determine whether there is a transformation from model coordinates to sensor coordinates that would account for the sensory data. Unfortunately, this is computationally extremely expensive. In order to compute such a model test, we need three points, whose corresponding face normals are linearly independent, as well as the measured normals at those points. Clearly, we would in general need $k$ sensory points to ensure this, where $k \geq 3$. Thus, if $n$ is the number of faces in the object, we would need to consider on the order of $n^k$ model tests, each of which requires considerable computational effort.

On the other hand, using the simple geometric constraints outlined above requires only a straightforward table lookup, and, as we shall see, can drastically reduce the number of interpretations to which a model test must be applied. Since the constraints can be applied near the root of the tree, it is possible to prune whole subtrees from the IT, at virtually no computational expense.

We consider each of the constraints in more detail below.

### 2.1.1. Distance Pruning

If an interpretation calls for pairing two of the sensed points with two object faces, the distance between the sensed points must be within the range of distances between the faces (see also [Bolles and Cain 82]). Note that the distances between *all* pairs of sensed points must be consistent, i.e., there are three distances between three sensed points, and in general $\binom{k}{2}$ distances between $k$ sensed points. Because of this, the distance constraint typically becomes more effective as more sensed points are considered.

Given two faces on a three-dimensional object, we can compute the *range of* distances between points on the faces. The minimum distance may be determined as the minimum of the shortest distance between all pairs of edges and the perpendicular distances between vertices of one face and the plane of the other face (when the vertex projects inside the face polygon). The maximum requires examining distances between pairs of vertices. Note that we can also compute the range of distances between points on *one* face (zero up to the diameter of the face). Sophisticated algorithms may be used to reduce the complexity of these computations, but since they are to be performed off-line, once for each model, their efficiency is not critical to the approach.

The distance constraint can be implemented in the following manner. For object $O_j$, with $f_j$ faces, we construct an $f_j$ by $f_j$ table, whose entries determine the range of possible distances between pairs of faces. In particular, for a pair of faces $(i, k)$, $i \neq k$, the maximum distance between the faces is stored in table location $\mathbf{dtable}_j[\max(i,k), \min(i,k)]$ and the minimum distance between the faces is stored in table location $\mathbf{dtable}_j[\min(i,k), \max(i,k)]$. If $i = k$, we simply store the maximum distance in the diagonal entry $\mathbf{dtable}_j[i,i]$, since the minimum distance defaults to 0. This representation makes checking a distance constraint straightforward, since the set of all pairs of faces $(i, k)$ on object $O_j$ consistent with some measured distance $d$ is given by

$$\left\{(i,k) \mid \mathbf{dtable}_j[\min(i,k), \max(i,k)] \leq d \leq \mathbf{dtable}_j[\max(i,k), \min(i,k)]\right\}$$

plus the pair $(i,i)$ if $d \leq \mathbf{dtable}_j[i,i]$.

Given any $k-1$ interpretation, represented by the set of faces $(i_1, \ldots, i_{k-1})$, and a new $k^{th}$ sensed point, the generation of the next level of the IT below this interpretation can be easily computed by checking the appropriate portions of the distance tables. In particular, if the measured distance between one of the previous sensed points, $i_\ell$, and the new one is given by $d_{i_\ell}$, the set of possible faces that can be assigned to sensed point $P_k$ is given by

$$\bigcap_{\ell=1}^{k-1} \left\{i \mid \mathbf{dtable}_j[\min(i,i_\ell), \max(i,i_\ell)] \leq d_{i_\ell} \leq \mathbf{dtable}_j[\max(i,i_\ell), \min(i,i_\ell)]\right\}$$

unioned with the set

$$\bigcap_{\ell=1}^{k-1} \left\{i_\ell \mid 0 \leq d_{i_\ell} \leq \mathbf{dtable}_j[i_\ell, i_\ell]\right\}.$$

For very complex objects, much more time efficient ways of representing and searching for faces that satisfy a distance constraint are possible. A full discussion of these methods is beyond the scope of this paper, however.

We note that it may frequently be the case, especially for a tactile sensor, that the sensor makes contact along an edge or at a vertex, rather than in the interior of a face. The method described above would still work unchanged under these circumstances. But if the sensor is capable of detecting that contact is at a vertex or edge, then tighter constraints can be applied. This is accomplished by constructing tables of distance ranges between vertices and between edges and applying the pruning algorithm based on those tables when appropriate.

Similarly, in the case of visual sensing, if the edges and vertices of an object can be reliably determined from the sense data, the recognition process is greatly simplified. (Note the relationship to the recognition method used in [Bolles, et al. 82].)

### 2.1.2. Angle Pruning

Sensed points are associated with a range of legal surface normals consistent with the sensory data. If an interpretation calls for pairing two of the sensed points (and normals) with two object faces, the range of angles between the sensed normals must include the angle between the normals of the corresponding object faces.

To see how this information can be used to prune the IT, we first consider the case in which the object has three degrees of freedom (two translation and one rotational). Under this restriction on degrees of freedom, the range of surface normals can be represented as a range of angles relative to the hand frame.

At a sensed point $P$, we can measure the local surface normal as lying in the range of angles

$$\phi \in [\omega - \epsilon, \omega + \epsilon]$$

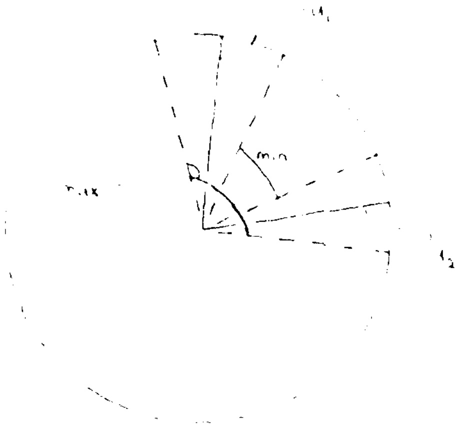
where $\omega$ is the actual measurement, and $\epsilon$ defines the range of possible angles about this measurement. We are given a sensor point $P_1$, with measured normal $\omega_1$, which has been assigned to face $i$, with associated model coordinate surface normal given by $\nu_i$. Next, we record a second point $P_2$, with measured normal $\omega_2$, which has been assigned to face $k$, with associated model coordinate surface normal given by $\nu_k$. For these assignments to be consistent, it must be the case that the angle between the model faces must be included in the range of angles between the ranges of normals determined from the measured normals and the error bounds

$$(\omega_2 - \omega_1) - (\epsilon_1 + \epsilon_2) \leq \nu_k - \nu_i \leq (\omega_2 - \omega_1) + (\epsilon_1 + \epsilon_2).$$

It is clear that an implementation similar to that used for distance pruning will also suffice here. For object $O_j$, with $e_j$ edges, we can set up an $e_j$ by $e_j$, lower diagonal table **atable**$_j$ such that **atable**$_j[\max(i,k), \min(i,k)] = \nu_k - \nu_i$. This representation makes checking a surface normal constraint straightforward, since the set of all pairs of faces $(i,k)$ on object $O_j$ consistent with some measured ranges of surface normals is given by

$$\left\{(i,k) \mid (\omega_2 - \omega_1) - (\epsilon_1 + \epsilon_2) \leq \mathbf{atable}_j[\max(i,k), \min(i,k)] \leq (\omega_2 - \omega_1) + (\epsilon_1 + \epsilon_2)\right\}.$$

Figure 3. Angle Ranges



Given any $k-1$-interpretation, *and a new* $k^{th}$ sensed point, the generation of the next level of the IT below this interpretation can easily be computed by checking the appropriate portions of the angle tables. Note that the $k^{th}$ edge must be *consistent* with the angles between all previous faces.

In the two-dimensional (three degree of freedom) case, the range of possible surface normals at a sensed point was represented by the pair $(\alpha_1, \epsilon_1)$ where $\alpha_1$ denoted the sensed normal, and $\epsilon_1$ denoted the range of error about that sensed point. In three dimensions, the obvious generalization is to use angle *cones*, so that if $\mathbf{u}_1$ denotes the unit sensed surface normal, the range of possible values for the actual surface normal will be denoted by the right circular cone

$$\{\mathbf{n}_1 \mid \mathbf{n}_1 \cdot \mathbf{u}_1 \geq \epsilon_1\}.$$

We could proceed identically to the two-dimensional case by noting that the cone of sensed normals constrains the set of possible three-dimensional rotations between the hand and model coordinate systems. Then, given a second sensed point $P_2$ with some sensed normal, the set of feasible faces would be restricted by the range of possible rotations. This method is quite difficult to implement, however. There is a much simpler alternative method.

Suppose that at the second sensed point, the set of possible surface normals in hand coordinates is given by

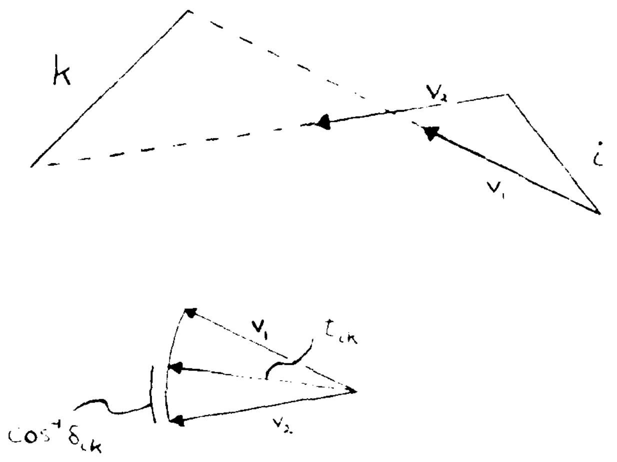
$$\{\mathbf{n}_2 \mid \mathbf{n}_2 \cdot \mathbf{u}_2 \geq \epsilon_2\}.$$

Then, in order for faces $i$ and $k$, with associated surface normals $\mathbf{v}_i$ and $\mathbf{v}_k$ to be consistent it must be the case that

$$\mathbf{v}_i \cdot \mathbf{v}_k \in \{\mathbf{n}_1 \cdot \mathbf{n}_2 \mid \mathbf{n}_1 \cdot \mathbf{u}_1 \geq \epsilon_1, \quad \mathbf{n}_2 \cdot \mathbf{u}_2 \geq \epsilon_2\}.$$

We can rephrase this in the following manner. Let $\cos\alpha_1 = \epsilon_1$, $\cos\alpha_2 = \epsilon_2$, $\alpha_{12} = \alpha_1 + \alpha_2$ and $\cos\gamma_{12} = \mathbf{u}_1 \cdot \mathbf{u}_2$. Then, we claim that the set

Figure 4. Range of Directions between Sensed Points



$$\{\mathbf{n}_1 \cdot \mathbf{n}_2 \mid \mathbf{n}_1 \cdot \mathbf{u}_1 \geq \epsilon_1, \quad \mathbf{n}_2 \cdot \mathbf{u}_2 \geq \epsilon_2\}$$

is contained in the set

$$\{\mathbf{n}_1 \cdot \mathbf{n}_2 \mid \cos[\min(\pi, \gamma_{12} + \alpha_{12})] \leq \mathbf{n}_1 \cdot \mathbf{n}_2 \leq \cos[\max(0, \gamma_{12} - \alpha_{12})]\}. \qquad (1)$$

A proof of this is found in Appendix I. Figure 3 illustrates this result in two dimensions.

An implementation of angle pruning similar to that used for distance pruning is now also possible. For object $O_j$, with $f_j$ faces, we can set up an $f_j$ by $f_j$, lower diagonal table **atable**$_j$ such that **atable**$_j[\max(i,k), \min(i,k)] = \mathbf{v}_i \cdot \mathbf{v}_k$, where recall that $\mathbf{v}_i$ denotes the unit normal to face $i$ in the model.

### 2.1.3. Direction Pruning

Consider a pair of sensed points $P_1$ and $P_2$ and let $\mathbf{u}_{12}$ be the unit direction vector between them. Suppose that we know the measured surface normal at point $P_1$ to within some cone of error, for example, the measured value is $\mathbf{w}_1$, and the range of possible values for the surface normal is

$$\{\mathbf{v}_1 \mid \mathbf{v}_1 \cdot \mathbf{w}_1 \geq \epsilon_1\}.$$

Then the set of possible "angles" between the direction vector and the surface normal of the face is given by

$$\{\mathbf{v}_1 \cdot \mathbf{u}_{12} \mid \mathbf{v}_1 \cdot \mathbf{w}_1 \geq \epsilon_1\}.$$

In an interpretation, suppose that point $P_1$ has been assigned to face $i$, with normal $\mathbf{n}_i$ in the model, and we now consider possible faces $k$ to assign to point $P_2$. Let the range of possible unit vectors (directions) from face $i$ to face $k$ be denoted by the cone

$$\{\mathbf{s}_{ik} \mid \mathbf{s}_{ik} \cdot \mathbf{t}_{ik} \geq \delta_{ik}\}$$

for some pair $\mathbf{t}_{ik}$ and $\delta_{ik}$. Figure 4 illustrates this cone in a two dimensional example. Appendix II shows how this cone may be computed from models of the object faces. In the model, the set of possible angles between legal directions and the surface normal is

$$\{\mathbf{n}_i \cdot \mathbf{s}_{ik} \mid \mathbf{s}_{ik} \cdot \mathbf{t}_{ik} \geq \delta_{ik}\}. \tag{2}$$

Thus, assume that point $P_1$ is on face $i$, with normal $\mathbf{n}_i$, that we have measured $\mathbf{w}_1$, that we know $\epsilon_1$, and that we have also measured $P_2$. A face $k$, whose direction range from face $i$ is given by the pair $(\mathbf{t}_{ik}, \delta_{ik})$, is a feasible face for point $P_2$ if the set in equation (2) intersects the cone

$$\{\mathbf{v}_1 \cdot \mathbf{u}_{12} \mid \mathbf{v}_1 \cdot \mathbf{w}_1 \geq \epsilon_1\}. \tag{3}$$

If $\cos \gamma_{ik} = \delta_{ik}$, and $\cos \phi_{ik} = \mathbf{n}_i \cdot \mathbf{t}_{ik}$, then we know from the derivation in Appendix I that the set of equation (2) is contained in the set

$$\{\mathbf{n}_i \cdot \mathbf{s}_{ik} \mid \cos(\gamma_{ik} + \phi_{ik}) \leq \mathbf{n}_i \cdot \mathbf{s}_{ik} \leq \cos(\gamma_{ik} - \phi_{ik})\}.$$

Similarly, if $\cos \alpha_1 = \epsilon_1$ and $\cos \omega_{12} = \mathbf{v}_1 \cdot \mathbf{u}_{12}$, then the set of equation (3) is contained in the set

$$\{\mathbf{v}_1 \cdot \mathbf{u}_{12} \mid \cos(\alpha_1 + \omega_{12}) \leq \mathbf{v}_1 \cdot \mathbf{u}_{12} \leq \cos(\alpha_1 - \omega_{12})\}.$$

Therefore, for the pairings of $P_1$ with face $i$ and $P_2$ with face $k$ to be consistent with the direction constraint, it must be the case that the intersection of the numerical ranges of dot products is not null, i.e.,

$$[cos(\alpha_1 - \omega_{12}), cos(\alpha_1 + \omega_{12})] \bigcap [cos(\gamma_{ik} - \varphi_{ik}), cos(\gamma_{ik} + \phi_{ik})] \neq \emptyset$$

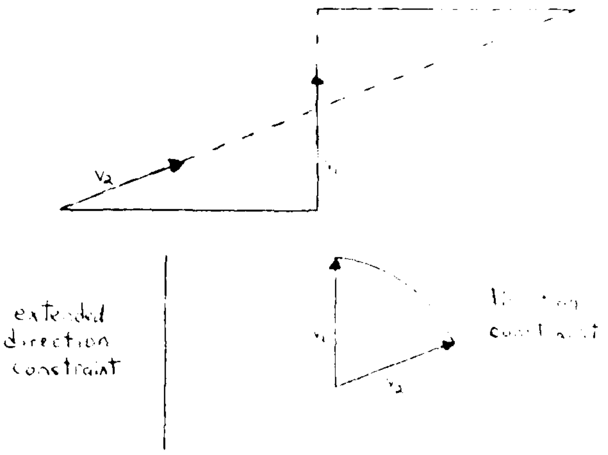
The direction constraint can also be implemented in a form similar to that used for distance and angle pruning. For object $O_j$, with $f_j$ faces, we can set up an $f_j$ by $f_j$ table **ctable**, such that $\mathbf{ctable}_j[i,k] = [\cos(\gamma_{ik} - \phi_{ik}), \cos(\gamma_{ik} + \phi_{ik})]$. Again, the set of all pairs of faces $(i,k)$ on object $O_j$ consistent with some measured ranges of surface normals is given by

$$\left\{(i,k) \mid [\cos(\alpha_1 - \omega_{12}), \cos(\alpha_1 + \omega_{12})] \bigcap \mathbf{ctable}_j[i,k] \neq \emptyset\right\}.$$

Note that the direction constraint is not symmetric, as are the distance and angle constraints, so before pairing $P_2$ to face $k$, we must repeat the test above interchanging the roles of $i$ and $k$. Similarly, the test must be applied to each pairing of sensed points and faces in an interpretation.

The constraint described above places constraints on the angle between a surface normal and unit vectors from one face to another. In addition to constraining the angles of unit vectors, we may constrain the magnitude of the component along the surface normal of the vector between the sensed points. The statement and implementation of the constraint is essentially unchanged, except that $\mathbf{u}_{12}$ and $\mathbf{t}_{ik}$ are no longer unit vectors but the actual vector between the sensed points. The effectiveness of the constraint is in general improved, however, since it now captures some distance and some angular constraint. The difference between this extended direction constraint and the simple direction constraint is illustrated in Figure 5. Two parallel faces (faces 1 and 2 in the figure) displaced relative to each other

Figure 5. Extended Direction Constraint



give rise to a cone of directions, but a single value for the normal component of vectors connecting the faces. Note that an interpretatation that assigns $P_1$ to face 1 and $P_2$ to face 3 is consistent with all the previously mentioned constraints except for the extended direction constraint. The figure also illustrates that the extended direction constraint does not subsume the distance constraint, since direction only constrains the normal component of distance.

There is an alternate form of the direction constraint, useful when no bound on the surface normal is available. It can briefly be described as follows. Given two faces $h$ and $i$ on an object, we can compute the range of directions between points on the faces, forming a cone of possible directions. Similarly for faces $i$ and $j$, we can compute the cone of possible directions. The combination of these two cones defines a range of possible angles for the triplet of faces $h, i, j$.

If an interpretation calls for pairing three of the sensed points with three object faces, the angle formed by this triplet of sensed points must be within the range of possible angles between the triplets of faces. Note that the angles formed by *all* triplets of sensed points must be consistent, i.e. for three sensed points, there are three angles, for $k$ sensed points, there are $3\binom{k}{3}$ angles. Hence, this constraint also becomes more effective as more sensed points are considered.

This form of the direction constraint can be used when only vertices and edges are touched, as it does not require sensing surface normals. Note that this form of the constraint can also be extended to use the magnitude of the vectors between sensed points as well as their direction. This form of the direction constraint allows pruning of the IT for $k \geq 3$. The previous formulation of the constraint allows pruning of the IT for $k \geq 2$. As well, this form of the constraint would require an $n^3$ table, as opposed to an $n^2$ one for the previous formulation. Given the size of $n$ to be expected for typical objects, this is a critical difference.

# 3. Model Testing

Once the interpretation tree has been pruned by the local constraints, there will be some set of possible interpretations of the sensed data, each one consisting of a set of triples $(\mathbf{p}_i, \mathbf{n}_i, f_i)$, where $\mathbf{p}_i$ is the vector representing the sensed position, $\mathbf{n}_i$ is the vector representing the sensed normal, and $f_i$ is the face assigned to this sensed data for that particular interpretation. In the model test stage of the processing, we want to

1. determine the actual transformation from model coordinates to sensor coordinates, corresponding to the interpretation,
2. check that under this transformation, not only are the sensed points transformed to lie on the appropriate planes, but moreover, that the sensed points actually lie within the bounds of the assigned faces.

We will assume that a vector in the model coordinate system is transformed into a vector in the sensor coordinate system by the following transformation:

$$\mathbf{v}_s = R\mathbf{v}_m + \mathbf{v}_0$$

where $R$ is a rotation matrix, and $\mathbf{v}_0$ is some translation vector. We need to solve for $R$ and $\mathbf{v}_0$. We note that a solution could be obtained using a least-squares method, such as is used by [Faugeras and Hebert 83]. This type of solution can be computationally expensive, however, and in the following sections, we develop an alternative method.

## 3.1. Rotation Component

We consider first the rotation component of the transformation. Consider the first triple of a particular interpretation, $(\mathbf{p}_i, \mathbf{n}_i, f_i)$. The sensed normal is given by $\mathbf{v}_i$ and corresponding to face $f_i$ is a face normal $\mathbf{m}_i$. For $R$ to be a legitimate rotation, it should take the normal $\mathbf{m}_i$ into $\mathbf{n}_i$ (ignoring issues of error in the measurements for now).

Now, any rotation can be represented by a direction about which the rotation takes place, and an angle of rotation about that direction. What is the set of possible directions of rotation $\mathbf{r}$ consistent with $\mathbf{n}_i$ and $\mathbf{m}_i$? Any rotation will preserve the angle between the transformed vector and the direction of rotation. Hence, any legitimate rotation direction must be equiangular with $\mathbf{n}_i$ and $\mathbf{m}_i$. Thus, the set of potential directions is given by

$$\left\{\mathbf{r}_{ij} \mid \mathbf{r}_{ij} \cdot \mathbf{m}_i = \mathbf{r}_{ij} \cdot \mathbf{n}_i\right\}.$$

or equivalently

$$\left\{\mathbf{r}_{ij} \mid \mathbf{r}_{ij} \cdot (\mathbf{m}_i - \mathbf{n}_i) = 0\right\}.$$

That is, $\mathbf{r}_{ij}$ is perpendicular to $(\mathbf{m}_i - \mathbf{n}_i)$.

Now, consider a second triple in the interpretation $(\mathbf{p}_j, \mathbf{n}_j, f_j)$ and let $\mathbf{m}_j$ be the normal to face $f_j$. Provided $\mathbf{m}_j \neq \pm\mathbf{m}_i$ and $\mathbf{n}_i - \mathbf{m}_i$ is not (anti-)parallel to $\mathbf{n}_j - \mathbf{m}_j$, we can constrain $\mathbf{r}_{ij}$ to a second set

$$\left\{\mathbf{r}_{ij} \mid \mathbf{r}_{ij} \cdot (\mathbf{m}_j - \mathbf{n}_j) = 0\right\}.$$

Since the rotation is the same, $\mathbf{r}_{ij}$ must lie in both sets, i.e., it must be perpendicular to both vectors. Hence, $\mathbf{r}_{ij}$ is given by the unit vector in the direction

$$(\mathbf{m}_i - \mathbf{n}_i) \times (\mathbf{m}_j - \mathbf{n}_j)$$

to within an ambiguity of 180°.

This derivation can be recast in geometric terms in the following manner. Any unit rotation vector $\mathbf{r}$ taking $\mathbf{m}_i$ into $\mathbf{n}_i$ must lie on the perpendicular bisector of the line connecting $\mathbf{n}_i$ to $\mathbf{m}_i$. Similarly, it must also lie on the perpendicular bisector of the line connecting $\mathbf{n}_j$ to $\mathbf{m}_j$. Since the rotation is the same, it must lie in the intersection of the two perpendicular bisector planes, as above, and hence is given by the specified unit vector

$$(\mathbf{m}_i - \mathbf{n}_i) \times (\mathbf{m}_j - \mathbf{n}_j).$$

If there were no error in the sensed normals, we would be done. With error included in the measurements, however, the computed rotation direction $\mathbf{r}$ could be slightly wrong. One way to reduce the effect of this error is to compute all possible $\mathbf{r}_{ij}$ as $i$ and $j$ vary over the faces of the interpretation, and then cluster these computed directions to determine a value for the direction of rotation $\mathbf{r}$.

Once we have computed a direction of rotation $\mathbf{r}$, we need to determine the angle $\theta$ of rotation about it. It is straightforward to show that (see, for example, [Korn and Korn, 68] p. 473)

$$\mathbf{m}_i = \cos\theta\mathbf{n}_i + (1 - \cos\theta)(\mathbf{r}\cdot\mathbf{n}_i)\mathbf{r} + \sin\theta(\mathbf{r}\times\mathbf{n}_i).$$

Simple algebraic manipulation, using the fact that $\mathbf{r}\cdot\mathbf{m}_i = \mathbf{r}\cdot\mathbf{n}_i$, yields

$$\cos\theta = 1 - \frac{1 - (\mathbf{n}_i \cdot \mathbf{m}_i)}{1 - (\mathbf{r}\cdot\mathbf{n}_i)(\mathbf{r}\cdot\mathbf{m}_i)}$$

$$\sin\theta = \frac{(\mathbf{r}\times\mathbf{n}_i)\cdot\mathbf{m}_i}{1 - (\mathbf{r}\cdot\mathbf{n}_i)(\mathbf{r}\cdot\mathbf{m}_i)}.$$

Hence, given $\mathbf{r}$, we can solve for $\theta$. Note that if $\sin\theta$ is zero, there is a singularity in determining $\theta$, which could be either 0 or $\pi$. In this case, however, $\mathbf{r}$ lies in the plane spanned by $\mathbf{n}_i$ and $\mathbf{m}_i$ and hence, only the $\theta = \pi$ solution is valid.

As before, in the presence of error, we may want to cluster the $\mathbf{r}$ vectors, and then take the average of the computed values of $\theta$ over this cluster.

Finally, given values for both $\mathbf{r}$ and $\theta$, we can determine the rotation matrix $R$. Let $r_x, r_y, r_z$ denote the components $\mathbf{r}$. Then

$$R = \cos\theta\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{bmatrix} + (1-\cos\theta)\begin{bmatrix}r_x^2 & r_xr_y & r_xr_z\\ r_yr_x & r_y^2 & r_yr_z\\ r_zr_x & r_zr_y & r_z^2\end{bmatrix} + \sin\theta\begin{bmatrix}0 & -r_z & r_y\\ r_z & 0 & -r_x\\ -r_y & r_x & 0\end{bmatrix}$$

Note that in computing the rotation component of the transformation, we have ignored the ambiguity inherent in the computation. That is, there are two solutions to the problem, $(\mathbf{r}, \theta)$ and $(-\mathbf{r}, -\theta)$. We assume that a simple convention concerning the sign of the rotation is used to choose one of the two solutions.

## 3.2. Translation Component

Next, we need to solve for the translation component of the transformation. We know that $\mathbf{v}_s = R\mathbf{v}_m + \mathbf{v}_0$, where $\mathbf{v}_m$ is a vector in model coordinates, $\mathbf{v}_s$ is the corresponding vector in sensor coordinates, and $R$ has been computed as above. Given a triple $(\mathbf{p}_i, \mathbf{n}_i, f_i)$ from the interpretation, let $\mathbf{m}_i$ be the normal of face $f_i$, with offset $d_i$, that is, the face is defined by the set of vectors

$$\{\mathbf{v} \mid \mathbf{v} \cdot \mathbf{m}_i = d_i\}.$$

Then the point in model coordinates corresponding to $\mathbf{p}_i$ is

$$R^{-1}(\mathbf{p}_i - \mathbf{v}_0)$$

and the following equation holds

$$\mathbf{m}_i \cdot \left(R^{-1}(\mathbf{p}_i - \mathbf{v}_0)\right) = d_i$$

or equivalently

$$(R\mathbf{m}_i) \cdot (\mathbf{p}_i - \mathbf{v}_0) = d_i.$$

This equation essentially constrains the component of the translation vector in the direction of $R\mathbf{m}_i$.

Suppose we consider three triplets from the interpretation, $(\mathbf{p}_i, \mathbf{n}_i, f_i)$, $(\mathbf{p}_j, \mathbf{n}_j, f_j)$, and $(\mathbf{p}_k, \mathbf{n}_k, f_k)$ such that the triple product $\mathbf{m}_i \cdot (\mathbf{m}_j \times \mathbf{m}_k)$ is non-zero, (i.e. the three face normals are independent). Then, we can construct three independent equations

$$\begin{aligned}
(R\mathbf{m}_i) \cdot \mathbf{v}_0 &= (R\mathbf{m}_i) \cdot \mathbf{p}_i - d_i \\
\left(R\mathbf{m}_j\right) \cdot \mathbf{v}_0 &= \left(R\mathbf{m}_j\right) \cdot \mathbf{p}_j - d_j \\
(R\mathbf{m}_k) \cdot \mathbf{v}_0 &= (R\mathbf{m}_k) \cdot \mathbf{p}_k - d_k.
\end{aligned}$$

Each of these equations constrains a different, independent component of the translation vector $\mathbf{v}_0$, and hence the three equations together determine the actual vector. Straightforward algebraic manipulation then yields the following solution for the translation component $\mathbf{v}_0$:

$$\begin{aligned}
\left[\mathbf{m}_i \cdot (\mathbf{m}_j \times \mathbf{m}_k)\right]\mathbf{v}_0 =& ((R\mathbf{m}_i) \cdot \mathbf{p}_i - d_i)\left(\left(R\mathbf{m}_j\right) \times (R\mathbf{m}_k)\right) \\
&+ \left(\left(R\mathbf{m}_j\right) \cdot \mathbf{p}_j - d_j\right)((R\mathbf{m}_k) \times (R\mathbf{m}_i)) \\
&+ ((R\mathbf{m}_k) \cdot \mathbf{p}_k - d_k)\left((R\mathbf{m}_i) \times \left(R\mathbf{m}_j\right)\right)
\end{aligned}$$

As in the case of rotation, if there is no error in the measurements, then we are done. The simplest means of attempting to reduce the effects of error on the computation is to average $\mathbf{v}_0$ over all possible trios of triplets from the interpretation. Note that for numerical stability, one may want to restrict the computation to triplets such that $\mathbf{m}_i \cdot (\mathbf{m}_j \times \mathbf{m}_k)$ is greater than some threshold.

Finally, we have computed the transform $(R, \mathbf{v}_0)$ from model coordinates to sensor coordinates. To check a possible interpretation, we consider all triples $(\mathbf{p}_i, \mathbf{n}_i, f_i)$ in the interpretation and compute

$$R^{-1}(\mathbf{p}_i - \mathbf{v}_0).$$

We then check that this point lies within the bounds of face $f_i$ (to within some error range). If it does not, then the interpretation is invalid, and may be pruned. If all such triples satisfy this check, the interpretation is still valid.

We have assumed above that three independent face normals have been measured. When only one normal is available, neither the rotation or translation can be determined. When only two independent normals are available, the rotation can be determined as before, but only a direction of translation can be determined, not the actual magnitude of the translation. A range of possible translations can be determined, however, by interesecting the line, determined by the position of a sensed point and the *translation direction*, with the face assigned to the point by the interpretation. Of course, further sensing along this line to discover the position of the edge would determine the actual translation.

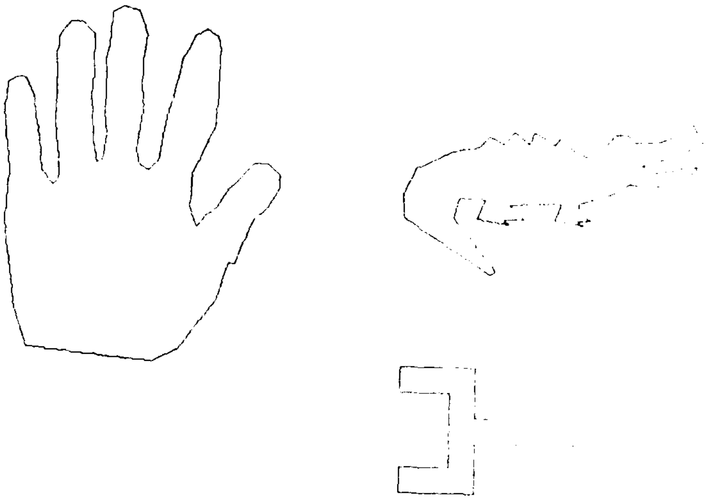
After the model test has been applied to all leaves of the interpretation tree, there may still be several interpretations remaining. Upon examination, one usually finds that these interpretations differ only in the assignment of one or two faces, all other faces being identical. This inability to distinguish between such nearly identical interpretations is a result of the error bounds on the sensing. Thus, as a final stage, we cluster the remaining interpretations in terms of their computed transformations, that is, we cluster the interpretations in terms of the computed orientation of the object in space. Here, we generally find very few such clusters. Indeed, in general there is only one computed orientation for the object, (the correct one), although *occasionally* two or more clusters survive, usually corresponding to symmetric interpretations of the sensed data.

## 4. Simulation Data

In order to test *the efficacy of the* algorithm in pruning the interpretation tree, we ran a large number of simulations. Some simulations for objects with three degrees of freedom (two translational and one rotational) have been described in [Gaston and Lozano-Pérez 83]. We include additional simulation data for objects with three positional freedoms, including the direction constraint. We also provide data for the more general case of three dimensional objects with six degrees of freedom.

Our goals are first to demonstrate that effective pruning of the interpretation tree is possible, at low computational expense, and second to *explore the sensitivity of the* algorithm to error in measuring the surface normal and the position of the sensed points.

Figure 6. 2D Test Models



### 4.1. Three Positional Freedoms

We begin by considering objects with two degrees of translational freedom and one degree of rotational freedom, using sample objects first considered in [Gaston and Lozano-Pérez 83], illustrated in Figure 6. The addition of the direction constraint greatly reduces the extent of the set of possible interpretations. To demonstrate this, a series of 250 runs of the algorithm was executed for each of the objects. Each run determined the number of interpretations consistent with a set of 5 sensed points. The points were determined by first randomly rotating the object about its centroid and then intersecting the object with five lines from its centroid along five evenly spaced directions. The points of intersection farthest from the centroid along each line were used as the sensed point. The (simulated) error in measuring the sensed position was bounded by 0.1 (i.e. a randomly oriented offset vector of random magnitude bounded by 0.1 was added to the point on the object), and the (simulated) error in measuring the angle of the surface normal was $\frac{\pi}{8}$ (i.e. a random vector was chosen whose dot product with the actual normal was bounded by $\cos^{-1}\frac{\pi}{8}$). To place these error ranges in perspective, the diameters of the models in Figure 6 were 9, 14 and 12 units for the wrench, gator and hand respectively.

The following table describes the results of this set of simulations, by histogramming the number of interpretations found. Thus, for $i \leq 10$, the number in the $i^{th}$ column is the number of trial runs which resulted in $i$ possible interpretations. Beyond this point, the histogram is compressed into units of tens. For example, the column labelled 20 lists the number of trial runs resulting in $k$ interpretations, where $10 < k \leq 20$. In order to examine the effectiveness of adding the direction constraint to the algorithm described in [Gaston and Lozano-Pérez 83], the simulations were run both with and without this constraint. For each object in the table, the first histogram corresponds to the case of using the direction constraint, and the second histogram to the case of not using it. Note that the number of edges for the wrench (W), gator (G) and hand (H) is 12, 50 and 67 respectively.

| Table I Two dimensional objects Histograms Errors π/8, 0.1 | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 20 | 30 | 40 | 50 | 60 | 70 | 80 | 90 | 100 | >100 |
| W | | 242 | | 7 | | | | 1 | | | | | | | | | | | | |
| W | | 31 | | 25 | | 2 | | 2 | | | 3 | 19 | 29 | 20 | 34 | 27 | 13 | 20 | 23 | 2 |
| G | 22 | 62 | 31 | 38 | 20 | 23 | 10 | 6 | 8 | 10 | 20 | | | | | | | | | |
| G | 1 | 12 | 9 | 11 | 8 | 23 | 10 | 17 | 15 | 14 | 63 | 34 | 7 | 7 | 7 | 5 | 5 | | 1 | 1 |
| H | 15 | 61 | 36 | 29 | 21 | 20 | 22 | 14 | 11 | 9 | 12 | | | | | | | | | |
| H | 4 | 13 | 17 | 21 | 17 | 17 | 12 | 16 | 18 | 11 | 86 | 18 | | | | | | | | |

The results are striking in a number of different ways. First, note that the maximum number of possible interpretations observed for any of the objects was 20 (in the case of using the direction constraint), which is exceptionally low when considering that the total number of possible interpretations for the gator was $50^5$ or 312,500,000. Second, the median number of possible interpretations was only 2 for the wrench, and 4 for the gator and hand, when using the direction constraint. Without this constraint, the median number of interpretations rose to 48, 12 and 9 for the wrench, gator and hand, respectively. Of course, the results of the simulations will depend to a certain extent on the error ranges, a point that will be explored in some detail in the next section. We note that a tenth of an inch sensitivity in distance over a 10 to 20 inch range is within the range of current tactile sensors. The positioning accuracy of many current manipulators is within 0.01 inches, and the Purbrick tactile sensor has a matrix element separation of 0.06 inches, and the Hillis sensor has an element separation of 0.025 inches.

## 4.2. Six Positional Freedoms

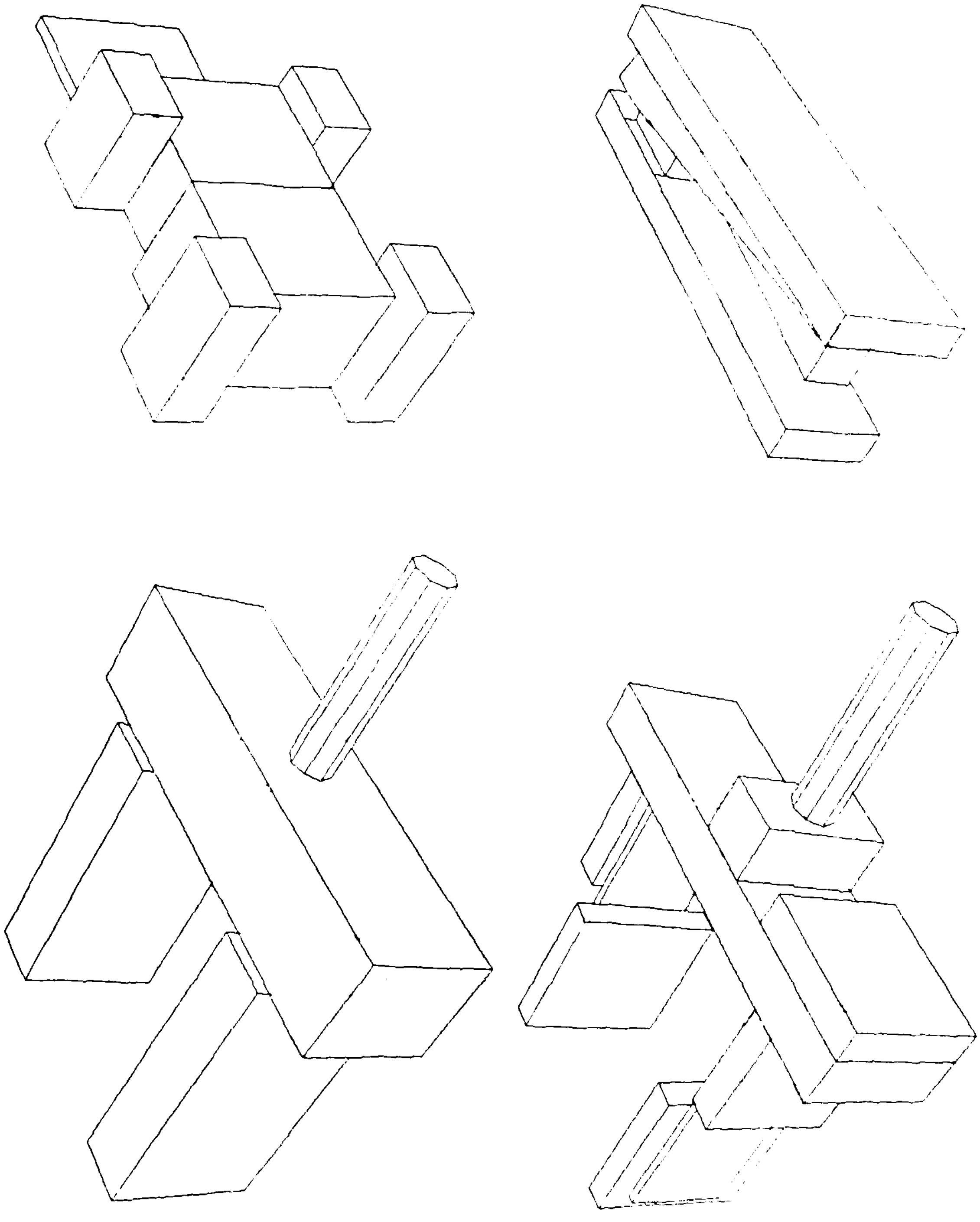
When considering the full three-dimensional problem of objects with six degrees of freedom, we have run extensive simulations on the models illustrated in figure 7. The diameters of these objects (that is the maximum separation of two points on the object) were roughly 4, 7, 8 and 8 inches for the housing, stapler, simple hand and complex hand respectively. In running simulations of the recognition algorithm on these objects, we have used two different sensing strategies, reflecting in part the difference between range and tactile sensing capabilities.

It should be noted that in all the following simulations, the efficiency of the tree pruning mechanism was improved by sorting the sensed points. In particular, rather than using the sensory data in arbitrary order, the points were sorted on the basis of pairwise separation, with the more distant points being ordered first. This sorting on distance tends to place the most effective constraints at the beginning of the process, a point that will be illustrated in Section 4.5.

## 4.3. Grid Sensing

In the first sensing method, the sensory data were generated by projecting a regular grid of points along three orthogonal directions, and noting where contact

Figure 7. 3D Test Models



was made with an arbitrarily oriented model of the object. This arbitrary orientation was obtained by randomly choosing values for the three Euler angles, computing a rotation transformation based on this and applying the rotation to the model. Note that this does not produce a uniform sampling of the space of rotations, but for our purposes it is a sufficiently random sampling. No translation offset was added, since this would not affect the process. The three-dimensional positions of the sensed points and the associated surface normals were then corrupted by noise within some specified bounds. For the simulations discussed below, the number of sensed points on each trial varied between 12 and 20.

The results of the first set of simulations is shown in Tables II, III, and IV. Table II lists statistics of the number of interpretations in the tree following local pruning, for a variety of sensing accuracies. Each simulation consisted of 100 trials, and the minimum and maximum number of interpretations are recorded over this set of trials, as well as the 50th and 90th percentile of the distribution of number of interpretations. Table III lists statistics of the number of interpretations in the IT that survive an explicit model test. It was observed at this stage that while the number of interpretations was not reduced to 1, as might be expected, the surviving interpretations generally tended to differ only in one or two faces. Moreover, the computed transformation parameters were nearly identical, indicating that the multiple interpretations surviving a model test actually corresponded to a single interpretation, to within the error ranges of the algorithm. Thus, Table IV lists statistics of the number of separate transformations computed for each trial. In particular, transformations whose direction of rotation differed by more than 1.5° were judged to be different, yielding a very tight clustering of the computed transformations. This clustering ignores differences in the translation component, a point that is addressed later in Table VI.

Table II No. of Interpretations After Local Pruning

| Object | Normal | Dist | Min | 50th | 90th | Max | Faces |
|---|---|---|---|---|---|---|---|
| Housing | $\pi/15$ | .01 | 1 | 4 | 8 | 36 | 40 |
| | | .05 | 1 | 8 | 28 | 72 | 40 |
| | | .10 | 2 | 40 | 208 | 658 | 40 |
| | $\pi/10$ | .01 | 2 | 4 | 16 | 240 | 40 |
| | | .05 | 2 | 16 | 40 | 256 | 40 |
| | | .10 | 6 | 96 | 440 | 1618 | 40 |
| | $\pi/8$ | .01 | 2 | 8 | 28 | 96 | 40 |
| | | .05 | 2 | 24 | 108 | 576 | 40 |
| | | .10 | 10 | 156 | 1144 | 3576 | 40 |
| Simple Hand | $\pi/12$ | .01 | 4 | 4 | 8 | 16 | 28 |
| | | .05 | 4 | 8 | 16 | 24 | 28 |
| | $\pi/10$ | .01 | 4 | 4 | 12 | 64 | 28 |
| | | .05 | 4 | 8 | 16 | 96 | 28 |
| | $\pi/8$ | .01 | 4 | 8 | 16 | 16 | 28 |
| | | .05 | 4 | 8 | 24 | 96 | 28 |
| Stapler | $\pi/12$ | .01 | 2 | 8 | 32 | 72 | 34 |
| | | .05 | 2 | 52 | 204 | 772 | 34 |
| | $\pi/10$ | .01 | 2 | 14 | 68 | 276 | 34 |
| | | .05 | 8 | 132 | 1104 | 2856 | 34 |
| Complex Hand | $\pi/12$ | .01 | 2 | 24 | 120 | 896 | 64 |
| | | .05 | 8 | 128 | 560 | 2880 | 64 |
| | $\pi/10$ | .01 | 2 | 40 | 240 | 3456 | 64 |
| | | .05 | 12 | 144 | 496 | 4416 | 64 |

In the table above, the *normal* 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* and *max* columns list the minimum and maximum number of interpretations observed; the *50th* column lists the median point of the set of simulations; the *90th* column lists the $90^{th}$ percentile of the set of simulations; and the *faces* column lists the number of faces in the model.

Table III. No. of Interpretations [illegible]

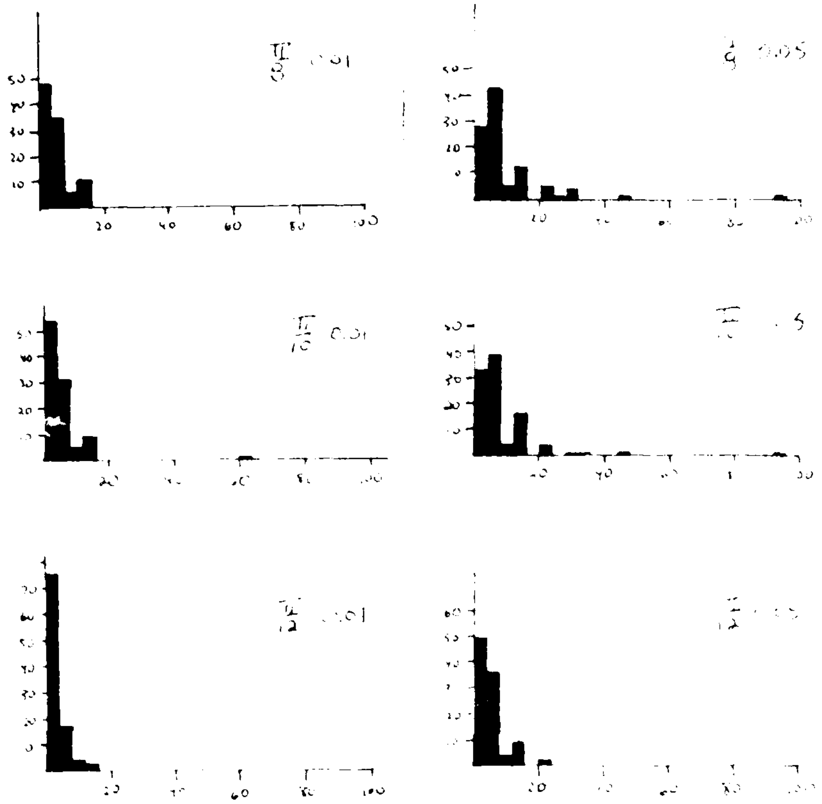
| Object | Normal | Dist | Min | 50th | 90th | Max | Faces |
|---|---|---|---|---|---|---|---|
| Housing | $\pi/15$ | .01 | 1 | [illegible] | [illegible] | [illegible] | [illegible] |
| | | .05 | 1 | [illegible] | [illegible] | [illegible] | [illegible] |
| | | .10 | 1 | [illegible] | [illegible] | [illegible] | [illegible] |
| | $\pi/10$ | .01 | 1 | 2 | [illegible] | [illegible] | [illegible] |
| | | .05 | 1 | 8 | [illegible] | [illegible] | 30 |
| | | .10 | 5 | [illegible] | [illegible] | [illegible] | [illegible] |
| | $\pi/8$ | .01 | 1 | [illegible] | [illegible] | [illegible] | [illegible] |
| | | .05 | 1 | [illegible] | [illegible] | [illegible] | 30 |
| | | .10 | 2 | [illegible] | [illegible] | [illegible] | 30 |
| Simple Hand | $\pi/12$ | .01 | 2 | 2 | [illegible] | 8 | 28 |
| | | .05 | 2 | [illegible] | [illegible] | 12 | 28 |
| | $\pi/10$ | .01 | 2 | [illegible] | 8 | [illegible] | 28 |
| | | .05 | 2 | [illegible] | [illegible] | [illegible] | 28 |
| | $\pi/8$ | .01 | 2 | [illegible] | [illegible] | [illegible] | 28 |
| | | .05 | 2 | [illegible] | [illegible] | [illegible] | 28 |
| Stapler | $\pi/12$ | .01 | 1 | [illegible] | 8 | [illegible] | 34 |
| | | .05 | 1 | [illegible] | [illegible] | [illegible] | 34 |
| | $\pi/10$ | .01 | 1 | [illegible] | [illegible] | [illegible] | 34 |
| | | .05 | [illegible] | [illegible] | [illegible] | [illegible] | 34 |
| Complex Hand | $\pi/12$ | .01 | [illegible] | [illegible] | [illegible] | [illegible] | 64 |
| | | .05 | 4 | [illegible] | [illegible] | [illegible] | 64 |
| | $\pi/10$ | .01 | 1 | 24 | [illegible] | 1728 | 64 |
| | | .05 | 6 | [illegible] | [illegible] | 2208 | 64 |

In the table above, the *normal* column lists the [illegible] about the measured surface normal; the *dist* column lists the error range of the distance sensing; the *min* and *max* columns list the minimum and maximum number of interpretations observed; the *50th* column lists the median point of the set of simulations; the *90th* column lists the 90th percentile of the set of simulations; and the *faces* column lists the number of faces in the model.

| Table IV No. of Transforms After Clustering | | | | | | | |
|---|---|---|---|---|---|---|---|
| Object | Normal | Dist | Min | 50% | 90% | Max | Faces |
| Housing | $\pi/15$ | .01 | 1 | 1 | 1 | 2 | 40 |
| | | .05 | 1 | 1 | 1 | 2 | 40 |
| | | .10 | 1 | 1 | 2 | 12 | 40 |
| | $\pi/10$ | .01 | 1 | 1 | 1 | 2 | 40 |
| | | .05 | 1 | 1 | 1 | 6 | 40 |
| | | .10 | 1 | 1 | 2 | 6 | 40 |
| | $\pi/8$ | .01 | 1 | 1 | 1 | 2 | 40 |
| | | .05 | 1 | 1 | 2 | 4 | 40 |
| | | .10 | 1 | 1 | 2 | 6 | 40 |
| Simple Hand | $\pi/12$ | .01 | 2 | 2 | 2 | 4 | 28 |
| | | .05 | 2 | 2 | 2 | 3 | 28 |
| | $\pi/10$ | .01 | 2 | 2 | 2 | 4 | 28 |
| | | .05 | 2 | 2 | 2 | 4 | 28 |
| | $\pi/8$ | .01 | 2 | 2 | 3 | 4 | 28 |
| | | .05 | 2 | 2 | 3 | 4 | 28 |
| Stapler | $\pi/12$ | .01 | 1 | 1 | 2 | 4 | 34 |
| | | .05 | 1 | 1 | 2 | 4 | 34 |
| | $\pi/10$ | .01 | 1 | 1 | 2 | 4 | 34 |
| | | .05 | 1 | 2 | 3 | 5 | 34 |
| Complex Hand | $\pi/12$ | .01 | 1 | 2 | 3 | 4 | 64 |
| | | .05 | 1 | 2 | 4 | 6 | 64 |
| | $\pi/10$ | .01 | 1 | 2 | 4 | 4 | 64 |
| | | .05 | 1 | 2 | 4 | 7 | 64 |

In the table above, the *normal* 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* and *max* columns list the minimum and maximum number of interpretations observed; the *50th* column lists the median point of the set of simulations; the *90th* column lists the $90^{th}$ percentile of the set of simulations; and the *faces* column lists the number of faces in the model.

Figure 8. Simple Hand Histograms



The first point to stress is that all of these numbers are remarkably low, given that the total number of possible interpretations of 15 sensed points on an object with 40 faces is roughly 1.074 × $10^{24}$. Thus, the local geometric constraints are very effective in reducing the combinatorics of feasible interpretations.

As might be expected, the number of interpretations in all three tables tends to rise with increasing error in the measured parameters. The distributions also tend to be strongly clustered near the low end of the scale, with a very shallow tail on the high end of the distribution. Thus, while the maximum number of interpretations can be high (e.g. 3576 for surface normal error cone of $\pi/8$ and distance error of 0.10), the median point and even the 90th percentile of the distribution are generally much smaller. Sample distributions for the number of interpretations surviving tree pruning are shown in Figure 8. One reason that the maximum number of feasible interpretations can be significantly larger than the median of the distribution is the occasional occurrence of dependent sensor information. For example, if most of the sensed points happen to lie on a single face, the amount of independent information about the object's position is much smaller than when the same number of sensed points lie on different faces. While the sensing strategy used here will reduce the

the error cone about the sensed normal, in radians, and a bound on the magnitude of the position error, in inches.

One reason that the maximum number of feasible interpretations can be significantly larger than the median of the distribution is the occasional occurrence of dependent sensor information. For example, if most of the sensed points happen to lie on a single face, the amount of independent information about the object's position is much smaller than when the same number of sensed points lie on different faces. While the sensing strategy used here will reduce the probability of this occuring, there is still a nonzero chance of such redundant sensing taking place, resulting in an occasional case of a large number of feasible interpretations.

The probability of such redundant sensing is also to a certain extent dependent on the shape of the object. For example, note that the aspect ratio of the stapler is much longer than that of the motor housing. This would tend to suggest that a regular sensing strategy is more likely to yield redundant information for the stapler than the housing. Indeed, a comparison of the appropriate sections of Table II shows that under similar conditions in measurement error, the number of feasible interpretations of the stapler is much higher than the motor housing, even though the stapler has fewer faces. This is partly due to redundant sensing and also partly due to symmetric interpretations of the data.

The number of distinct transformations is almost always 1 in these simulations. It was also observed that the computed transformation was generally very close to the actual one. For example, each row of Table V illustrates the average error in the computed transformations over 100 runs of the algorithm. The *direction* column lists the average angle between the correct and the computed direction of rotation, the *angle* column lists the average angle between the correct and the computed magnitude of rotation about the rotation direction, and the *translation* column lists the average magnitude of the difference between the correct and the computed translation component of the transformation. It can be seen from the table that the average error is remarkably low, generally on the order of 2-3 degrees, even for different objects and different amounts of sensor error. As might be expected, the average error does tend to rise with increases in the sensor error. In no case did the algorithm discard the correct interpretation. Note that the errors illustrated in Table V were recorded from the difference between the correct transformation and the computed transformation for the correct interpretation. There will be other, erroneous interpretations, with much larger differences between the computed and correct transformation.

Table V. Average Errors in Computed [illegible]

| Object | Normal | Dist | Direction (deg) | [illegible] | [illegible] |
|---|---|---|---|---|---|
| Simple Hand | $\pi/12$ | .01 | 2.17 | [illegible] | 0.08 |
| | | .05 | 2.08 | [illegible] | 0.09 |
| | $\pi/10$ | .01 | [illegible] | [illegible] | [illegible] |
| | | .05 | 3.58 | [illegible] | 0.12 |
| | $\pi/8$ | .01 | [illegible] | [illegible] | 0.16 |
| | | .05 | 5.26 | [illegible] | 0.17 |
| Housing | $\pi/15$ | .05 | 2.18 | [illegible] | [illegible] |
| | $\pi/10$ | .01 | 3.42 | [illegible] | 0.12 |
| | | .05 | [illegible] | [illegible] | [illegible] |
| | | .10 | 3.77 | [illegible] | 0.20 |
| | $\pi/8$ | .05 | [illegible] | [illegible] | 0.19 |
| Stapler | $\pi/12$ | .01 | [illegible] | [illegible] | 0.11 |
| | $\pi/10$ | .01 | 2.08 | [illegible] | 0.11 |

In the few cases in which more than one transformation was found, two factors generally are observed. The first is that the noise in the measured data can result in transformations differing by only a few degrees, [illegible] transformations are counted as being distinct. The second, more interesting factor is the possibility of symmetric interpretations of the data, for example due to a rotation of the object relative to the sensor. Consider first the case of a [illegible] symmetric object, such as the simple hand, which has a rotational symmetry [illegible] the algorithm always found at least two distinct transformations [illegible] that were consistent with the sensed data. For objects such as the motor housing, portions of the object are symmetric, for example, the base of the housing, ignoring the projecting lip. If all the sensed points happen to fall only on such a portion of the object, then symmetric interpretations of the data are possible. [illegible] these symmetric interpretations account for most of the cases of multiple transformations, especially when the sensor error is small. The few remaining cases arise when the error in the measurements yields two nearly identical (i.e. differing by only a few degrees of rotation) transformations that account for the data. As the error in the measured data decreases, these multiple interpretations tend to disappear.

The simulation data listed in Table IV is derived from a clustering of the interpretations based strictly on the rotation component of the transformation, that is, two transformations whose direction of rotation differed by less than 1.5° were considered to be part of the same cluster. This clustering technique, while very tight in the rotation component, ignores possible differences in the translation component of the transformation. To examine such differences, a number of the simulations were run, using a clustering of the interpretations with a rotation sensitivity of 1.5° and a translation sensitivity of either 0.05 or 0.01. The number of distinct transformations under this clustering scheme are indicated in Table VI. Note that while the number of distinct transformations does increase relative to the corresponding entries in Table IV, the change is not significant.

| Table VI No. of Transforms After Clustering | | | | | | | |
|---|---|---|---|---|---|---|---|
| Object | Normal | Dist | Cluster | Min | 50th | 90th | Max |
| Housing | $\pi/10$ | .05 | .01 | 1 | 2 | 4 | 6 |
| | | | .05 | 1 | 1 | 2 | 4 |
| Simple Hand | $\pi/10$ | .05 | .01 | 2 | 4 | 6 | 16 |
| | | | .05 | 2 | 2 | 4 | 12 |
| Stapler | $\pi/10$ | .01 | .01 | 1 | 4 | 14 | 57 |
| | | | .05 | 1 | 2 | 4 | 32 |
| Complex Hand | $\pi/10$ | .01 | .01 | 1 | 5 | 15 | 29 |
| | | | .05 | 1 | 3 | 6 | 16 |

## 4.4. Random Sensing

All of the previous simulations have generated the sensed data by projecting a regular grid of points along three orthogonal directions, generally resulting in between 12 and 20 contact points. Such a sensing strategy would be consistent with visual sensing modalities. A second set of simulations has been run using a sensing strategy more consistent with tactile sensors. Consider a set of three mutually orthogonal, directed rays, which intersect at a point. Suppose this point is taken to be some arbitrary point $(x, y, 0)$, chosen on the $x - y$ plane (note that by the definition of the object models, this plane will interect the object). Each ray is traced along is preferred direction, (with decreasing $z$ component), until either the object or the support plane was contacted. This operation was repeated for several different approaches, using randomly generated values of $x$ and $y$, until between 7 and 9 different contact points were made on the object. Tables VII, VIII and IX summarize the results of running sets of simulations, using sensory data generated in this fashion.

| Table VII - No. of Interpretations After Local Pruning | | | | | | | |
|---|---|---|---|---|---|---|---|
| Object | Normal | Dist | Min | 50th | 90th | Max | Faces |
| Simple Hand | $\pi/10$ | .01 | 2 | 4 | 20 | 90 | 28 |
| | | .05 | 2 | 8 | 44 | 300 | 28 |
| | $\pi/8$ | .01 | 2 | 8 | 48 | 444 | 28 |
| | | .05 | 2 | 12 | 84 | 320 | 28 |
| Housing | $\pi/10$ | .01 | 2 | 10 | 70 | 946 | 34 |
| | | .05 | 2 | 32 | 124 | 1234 | 34 |
| | $\pi/8$ | .01 | 2 | 14 | 74 | 284 | 34 |
| | | .05 | 2 | 62 | 406 | 4053 | 34 |

| Table VIII. No. of Interpretations After Model Test | | | | | | | |
|---|---|---|---|---|---|---|---|
| Object | Normal | Dist | Min | 50% | 90% | Max | Faces |
| Simple Hand | $\pi/10$ | .01 | 2 | 4 | 12 | 66 | 28 |
| | | .05 | 2 | [illegible] | 24 | 116 | 28 |
| | $\pi/8$ | .01 | 2 | 4 | 24 | 98 | 28 |
| | | .05 | 2 | 7 | 29 | 160 | 28 |
| Housing | $\pi/10$ | .01 | 1 | 4 | 32 | 526 | 34 |
| | | .05 | 1 | 16 | 80 | 606 | 34 |
| | $\pi/8$ | .01 | 1 | 8 | 26 | 136 | 34 |
| | | .05 | 1 | 32 | [illegible] | 377 | 34 |

| Table IX. No. of Transforms After Consistency [illegible] Rotation | | | | | | | |
|---|---|---|---|---|---|---|---|
| Object | Normal | Dist | Min | 50% | 90% | Max | Faces |
| Simple Hand | $\pi/10$ | .01 | 2 | 2 | 4 | 10 | 28 |
| | | .05 | 2 | 2 | 6 | 10 | 28 |
| | $\pi/8$ | .01 | 2 | 2 | 6 | 14 | 28 |
| | | .05 | 2 | 3 | 8 | 22 | 28 |
| Housing | $\pi/10$ | .01 | 1 | 1 | 4 | 11 | 34 |
| | | .05 | 1 | 2 | 7 | 13 | 34 |
| | $\pi/8$ | .01 | 1 | [illegible] | 5 | 9 | 34 |
| | | .05 | 1 | 3 | 11 | 14 | 34 |

As in the case of the earlier simulations, the effectiveness of the local constraints in reducing the number of feasible interpretations is clearly demonstrated. Interestingly, the number of distinct transformations tends to be somewhat higher than the earlier cases, especially for the motor housing. This results in part from the following situation. With the exception of one projecting portion, (see Figure 6), the housing is essentially a symmetric object, with respect to two different axes. As a consequence, if the sampled data points do not lie on this distinguishing projection, there could be several consistent, symmetric, interpretations of the data. In the case of sensory sampling on a regular grid of points, it is likely that at least one point will lie on this projection, and the symmetric ambiguity will not arise. In the case of fewer sample points, generated by random approaches to the object, it is much more likely that the feasible transformations will reflect this symmetry, and thus be higher in number.

In cases of ambiguity in interpretation, for example when several orientations of the motor housing are consistent with the sensed data, due to a partial symmetry of the object, it would be useful to have effective means for distinguishing between the possible solutions. A straightforward method would be to add sensory points generated at random until only one interpretation is consistent. This, of course, could be very inefficient, since it could take the addition of several points before a solution is found. In the case of the motor housing, for example, one would need to consider additional sensory points until one lying on the projecting lip of the housing is recorded. A more effective solution is to use the difference in feasible interpretations to find directions along which the points of contact of the different interpretations are widely separated. Such directions then constitute good candidates for generating the next sensed point [Gaston and Lozano-Pérez 83]. Extensions of the method to the six degree of freedom problem are currently under investigation.

### 4.5. Tree Pruning

Tables X and XI contain a final set of statistics that demonstrates the effectiveness of the local contraints in reducing the number of feasible interpretations in the IT. The regular grid approach is used to generate the sensory data. For the data in Table X, the points are sampled in random order as the IT is generated and pruned. For the data in Table XI, the sensed points are sorted on the basis of pairwise separation, with the more distant points being ordered first. This sorting on distance tends to place the most effective constraints at the beginning of the process. Since the point of the local constraints is to prune the IT as efficiently as possible, applying the most effective constraints first should result in pruning out entire subtrees at as early a stage in the tree generation process as possible. Using the sorted sense data, the interpretation tree was generated and pruned. Tables X and XI list statistics of the number of interpretations at each level of the tree, (i.e. the number of $k$-interpretations for different values of $k$), based on trials of 100 simulations each.

Table X – Feasible Interpretations – Unsorted Points

| Points | Min | 50th | 90th | Max |
|---|---|---|---|---|
| 2 | 12 | 96 | 334 | 432 |
| 3 | 4 | 110 | 388 | 678 |
| 4 | 4 | 55 | 373 | 675 |
| 5 | 4 | 40 | 244 | 1000+ |
| 6 | 4 | 26 | 189 | 1000+ |
| 7 | 2 | 24 | 108 | 686 |
| 8 | 2 | 20 | 82 | 636 |
| 9 | 2 | 20 | 76 | 520 |
| 10 | 2 | 20 | 72 | 336 |
| 11 | 2 | 16 | 62 | 280 |
| 12 | 2 | 16 | 64 | 200 |
| 13 | 4 | 20 | 64 | 304 |
| 14 | 2 | 18 | 72 | 304 |
| 15 | 2 | 20 | 80 | 304 |

| Table XI Feasible Interpretations: Sorted Points | | | | |
|---|---|---|---|---|
| Points | Min | 50th | 90th | Max |
| 2 | 4 | 18 | 48 | 94 |
| 3 | 2 | 18 | 38 | 82 |
| 4 | 2 | 12 | 29 | 68 |
| 5 | 2 | 8 | 22 | 50 |
| 6 | 2 | 8 | 24 | 58 |
| 7 | 2 | 8 | 24 | 56 |
| 8 | 2 | 8 | 24 | 72 |
| 9 | 2 | 8 | 22 | 72 |
| 10 | 2 | 8 | 24 | 80 |
| 11 | 2 | 8 | 34 | 192 |
| 12 | 2 | 8 | 36 | 352 |
| 13 | 2 | 12 | 32 | 512 |
| 14 | 2 | 12 | 40 | 480 |
| 15 | 2 | 12 | 48 | 288 |

It can be seen that the median number of feasible interpretations is quite small at all levels of the tree, even as the number of contact points is increased. This data implies that one of the strengths of the approach is the ability to prune out whole subtrees of the IT at a very early stage, thereby ensuring that the total number of tests to be applied is significantly smaller than the size of the entire tree. This leads to very efficient processing of the feasible interpretations.

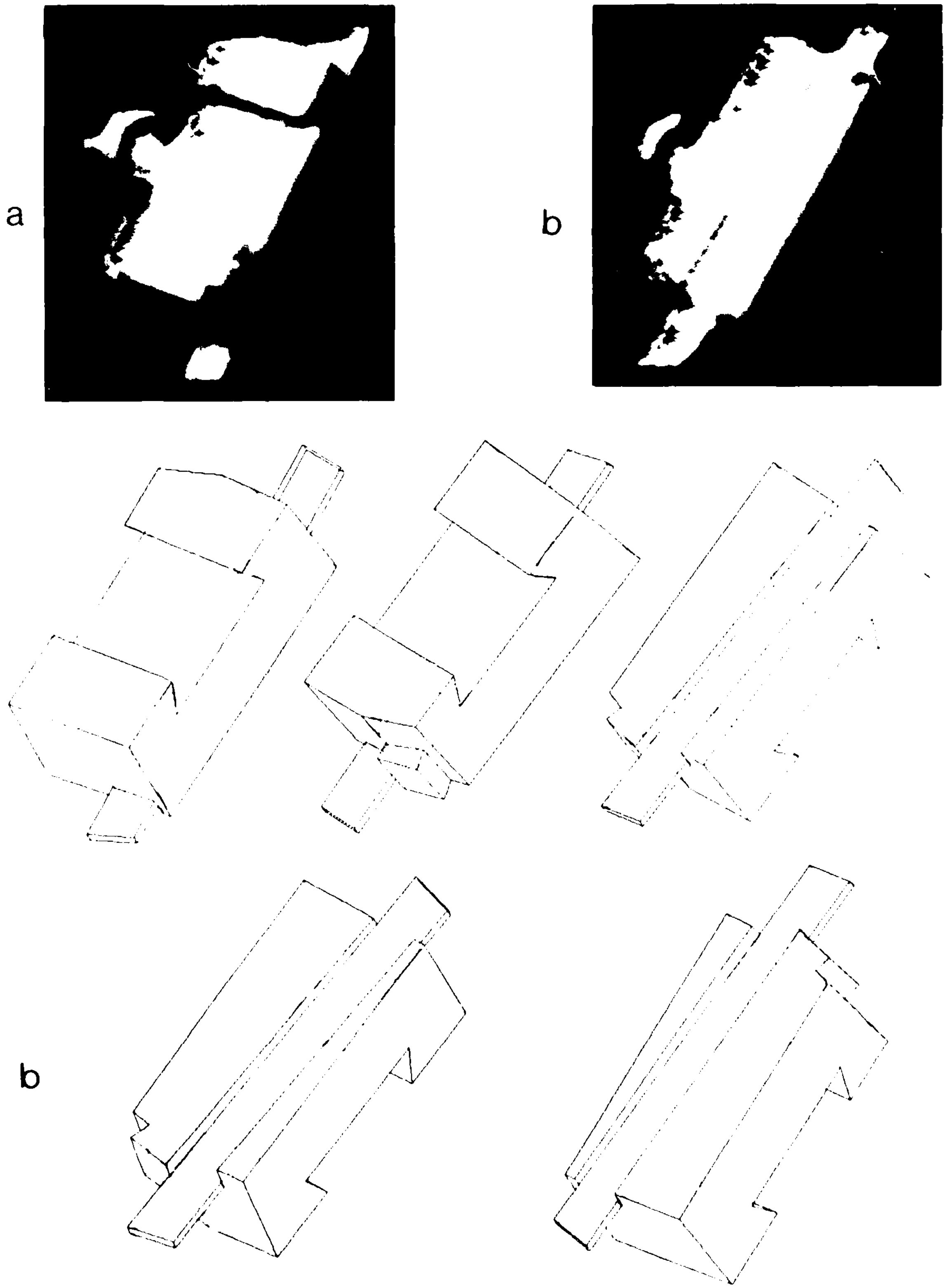
Sorting the points on distance is extremely effective as can be seen from the results reported in Table XI of the same set of runs as those in Table X, but where the points were sorted prior to pruning. The effect on running times of the pruning program is also quite drastic.

## 5. Performance on Range Data

We have performed limited testing of the algorithms described above using high quality range data obtained from a laser based triangulation system developed by Philippe Brou at our laboratory. Two samples of the data we used are shown in Figure 9. The data is obtained at high resolution, approximately 0.04 centimeter grid spacing along $x$ and 0.08 centimeter along $y$. A small number of points were obtained from the dense data by choosing points where a least squares fit to a plane over a 5 × 5 patch produced very low normalized residue errors. Points were chosen that included at least three independent normals. Note that the actual object includes a protrusion that was not present in the model; no data was taken from that region. In the data from figure 9(a), eleven points were used; in the data from figure 9(b) nine points were used. The accuracy bounds we employed were $\pm 0.02$ inch position accuracy and $\pm\frac{\pi}{15}$ accuracy in measuring the normal.

Figure 9 shows the results obtained from running the algorithm on the data described above. There were only 9 and 11 interpretations, respectively, left in the

Figure 9. Sample Range Data and Computed Interpretations



tree after pruning with the local constraints. From these, three valid transformations were found in one case and two in the other; they are shown in the figure. The correct transformation was found each time. The other transformations correspond to rotations that place the sensed points on parallel faces. Note, however, that disambiguations between the valid transformations would be straightforward once the transformations are known.

The quality of the data used in the experiments illustrated in Figure 9 corresponds to nearly the best error conditions used in the simulations. Results with larger error bounds, using data from sections where the data is less accurate, showed results similar to those in the simulations, i.e., more legal interpretations in the tree and more valid transformations but always including the correct one. It tends to reinforce the validity of the conclusions found in the simulations.

## 6. The Combinatorics of Pruning the IT

In the previous sections, we have outlined the basic interpretation algorithm. The crucial issue that determines the viability of this algorithm is the effectiveness of pruning the interpretation tree. Our goal has been to demonstrate that one can use simple local constraints to prune the interpretation tree, so that only a few of the relatively expensive model tests need to be made. The simulation results, under a variety of conditions, and the results on range data provide support for this claim.

It is also possible to provide a combinatorial analysis of the pruning of interpretation trees provided by local constraints. A detailed presentation of such an analysis is contained in a companion paper [Grimson and Lozano-Pérez 83]. Here, we demonstrate the scope of the combinatorical analysis by presenting a detailed discussion of the use of the distance constraint in pruning interpretation trees. Similar results hold for the other constraints.

We stress that the results given below are actually weak bounds on the number of interpretations to be expected after pruning. In practice, numbers close to these bounds are observed only when the sensors are arranged so as to obtain a minimum of information about the object.

### 6.1. Combinatorics of Distance Pruning

We will consider the case in which all faces (or edges in the two-dimensional case) have the same size, and derive bounds on the expected pruning of the IT.

Assume we have some arbitrary labelling of the faces from 1 to $n$ (for example, in the two-dimensional case, based on arc length from some starting point). For each pair of faces, $i$ and $j$, let $d_{ij}$ denote the separation of the midpoints of the faces. Let $\epsilon_{ij}$ be an upper bound on the range of variation in distance, for different sensed points on the two faces, i.e. let

$$\epsilon_{ij} = \text{lim sup}\left\{\epsilon : d_{ij} - \epsilon \leq |\mathbf{x} - \mathbf{y}| \leq d_{ij} + \epsilon, \forall \mathbf{x} \text{ on face } i,\ \forall \mathbf{y} \text{ on face } j\right\}$$

where $|\mathbf{x} - \mathbf{y}|$ is the distance between point $\mathbf{x}$ on face $i$ and point $\mathbf{y}$ on face $j$. Let $\epsilon$ be defined as the maximum over all $i, j$ of $\epsilon_{ij}$ plus some estimate of the maximum error of the sensed distance.

Now assume that we have recorded the position of two sensor points, $P_1$ and $P_2$, and let $s_{12}$ be the measured distance between them. Assume that the first point has been arbitrarily assigned to some face $i$ of the object. We want to determine how many faces $j$ of the object can consistently be assigned to the second point, given the separation $s_{12}$ and the known distribution of distances. Moreover, we want to be able to continue this for $k$ sensor points, determining an upper bound on the number of assignments of faces to sensor points that are consistent with the sensed separation between the faces.

Let the distribution of faces with respect to face $i$ as a function of distance be denoted by $\rho_i(s)$. In other words, $\rho_i(s)$ records the number of faces whose midpoint separation from face $i$ is given by the distance $s$. As a consequence,

$$\int_{s=0}^{d} d\rho_i(s) = n$$

where $n$ is the total number of faces, and $d$ is the diameter, or maximum separation of the object. Note that because $d\rho_i$ is a distribution, this is a Lebesgue-Stieltjes integral. The following bound on the number of nodes at the $k^{th}$ level of the IT holds for both two-dimensional and three-dimensional objects.

**Proposition 1: An upper bound on the expected number of nodes at the $k^{th}$ level of the interpretation tree, $k \geq 2$, is given by**

$$\left(\frac{2\epsilon n}{d}\right)^{k-1} n$$

**where $d$ is the diameter of the object, and $\epsilon$ is a bound on the distance sensitivity of the model.**

**Proof:** The proof proceeds by considering an iterative application of the expected maximum branching factor at each level of the tree. We assume that $b_{k-1}$ denotes a bound on the number of consistent nodes at the $k - 1^{st}$ level of the interpretation tree, and consider the branching factor obtained when adding a $k^{th}$ sensed point. Assume that sensor point $P_{k-1}$ has been assigned to face $i$, and that the measured separation of sensor point $P_{k-1}$ and $P_k$ is $s_k$. This implies that the midpoint separation of the corresponding faces is within $\epsilon$ of $s_k$. Hence, an upper bound on the number of possible faces consistent with $s_k$, given face $i$ assigned to point $P_{k-1}$ is

$$\int_{x=-\epsilon}^{\epsilon} d\rho_i(s_k + x).$$

Since the number of nodes at the $k - 1^{st}$ level of the tree is bounded by $b_{k-1}$, an upper bound on the total number of nodes at the $k^{th}$ level of the tree is

$$b_{k-1} \max_i \int_{x=-\epsilon}^{\epsilon} d\rho_i(s_k + x).$$

We now wish to determine a bound on the expected number of nodes, evaluated over the range of possible values for $s$. If $\Psi(s)$ denotes the distribution of sensed distances, then an upper bound on the expected number of nodes is

$$\frac{\int_{s=0}^{d} b_{k-1}\max_i \int_{r=-\epsilon}^{\epsilon} d\rho_i(s+r)\, d\Psi(s)}{\int_{s=0}^{d} d\Psi(s)}.$$

If we know which object is being sensed, we could derive an explicit form for $d\Psi(s)$. Since we are considering the case of sensing from a set of possible objects, the best we can do is consider the distribution of sensed distances over all possible orientations of all objects, and this is best given by a uniform distribution. Thus,

$$d\Psi(s) = \frac{1}{d}ds$$

and an upper bound on the expected number of nodes becomes

$$b_{k-1}\frac{1}{d}\max_i \int_{s=0}^{d}\int_{r=-\epsilon}^{\epsilon} d\rho_i(s+r)\,ds.$$

Note that this double integration can essentially be considered as a counting problem. That is, we want to count the number of faces whose separation from the sampled face lies in an $\epsilon$-range about some point $s$, with this number being accumulated over all possible $\epsilon$-ranges (i.e. vary the midpoint $s$). Reversing the order of integration basically reverses the order of counting. Thus, rather than counting the number of faces lying within a range and summing over the set of ranges, we count the number of ranges in which each face is included, and sum over the number of faces. Clearly each face can be included in at most $2\epsilon$ ranges (as the midpoint of the range moves from $s-\epsilon$ to $s+\epsilon$), and the total number of faces is $n$. Thus, the branching factor at this level yields the iterative expression

$$b_k = b_{k-1}\frac{2\epsilon n}{d}.$$

The base case of $k=1$ yields the bound of $b_1 = n$, since the initial assignment of the point $P_1$ is arbitrary.

Evaluation of the iterative expression yields

$$b_k = \left(\frac{2\epsilon n}{d}\right)^{k-1} n$$

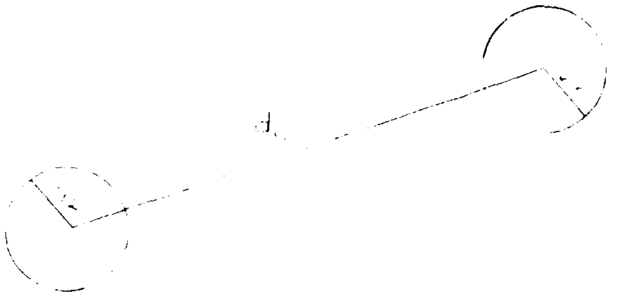
thereby concluding the proof by induction. ∎

While this proposition gives us an upper bound on the expected number of nodes, in order to evaluate it we need some estimate on $\epsilon$. The following two propositions provide this for the two- and three-dimension cases.

**Proposition 2: If all the edges of a two-dimensional object have the same length $e$, then $\forall i,j\ \epsilon_{ij} \leq e$.**

**Proof:** Connect the midpoints of two arbitrary faces, $i$ and $j$, with a line of length $d_{ij}$. Consider first the case of $d_{ij} > e$. The set of all possible orientations

Figure 10. Illustration for Proof of Proposition 2



of each of the edges about its midpoint describes a circle of radius $\frac{\epsilon}{2}$ about that endpoint. We are interested in the extrema in separation of points in these disks (see Figure 10). We claim that the maximum and minimum separation of points in the disks occur for the case of the edges parallel to the midpoint connector, giving a minimum of $d_{ij} - \epsilon$ and a maximum of $d_{ij} + \epsilon$.

While this can be shown algebraically, there is also a simple geometric proof. Construct a coordinate system with origin at the midpoint of edge $i$ and with $x$ axis along the midpoint connector. Now construct a circle of radius $d_{ij} - \epsilon$ about the point $(d_{ij} - \frac{\epsilon}{2}, 0)$. Clearly, this circle grazes the first disk at the point $(\frac{\epsilon}{2}, 0)$. Now, in order for any other point in the second disk to have a shorter distance, we must be able to position a circle of the same radius about that point and still intersect the first disk. This is not possible, by the following argument. The envelope of possible points can be formed by sweeping a circle of radius $d_{ij} - \epsilon$ through a series of positions such that the center of the circle lies at the limit of the second disk. This envelope only intersects the first disk at the above mentioned point, and hence, the minimum possible separation between the two edges is given by $d_{ij} - \epsilon$.

Similarly, the maximum separation can be shown to be $d_{ij} + \epsilon$ by constructing a circle of radius $d_{ij} + \epsilon$ about the point $(d_{ij} + \frac{\epsilon}{2}, 0)$ and using the same argument.

If $d_{ij} \leq \epsilon$, then the minimum distance is clearly bounded below by 0. The construction for the bound on the maximum distance is identical to that above. Hence, we see that $\epsilon_{ij} \leq \epsilon, \forall i, j$. ∎

**Corollary: If all the edges of a two-dimensional object have the same length, and the sensor error in measuring distances is much smaller than the length of an edge, then the expected number of nodes at the $k^{th}$ level of the interpretation tree, $k \geq 2$, that survive distance pruning is bounded above by**

$$\left(\frac{2p}{d}\right)^{k-1} n$$

**where $p$ is the perimeter of the object, and $d$ is its diameter.**

**Proof:** Since the sensor error is much less than the edge length, we see that $\epsilon$ is essentially given by the maximum over all $i, j$ of $\epsilon_{ij}$. From the previous proposition, this is bounded by the [illegible] [illegible]. The corollary follows naturally. ∎

Note that for convex objects, [illegible]

[illegible]

in general the perimeter for non-convex [illegible]. We note, however, that for highly [illegible] faces, then much of the perimeter of the object [illegible] follows from the observation that sensing [illegible] some other portion of the object. Thus [illegible] expression for non-convex objects [illegible] perimeter, which will generally correspond to the perimeter of a [illegible] object.

**Proposition 3:** If all the faces of a three-dimensional object have the same diameter $\epsilon$, and the same area $A_f$ then [illegible] $\leq \epsilon$.

**Proof:** The [illegible] is almost identical to the [illegible]. Here the [illegible] consists of [illegible] about the endpoints of a line of length [illegible] and [illegible] maximum separations of points on the two spheres. As in the [illegible], a geometric construction shows that the extremal cases occur when the diameters of the faces are parallel to the [illegible] connecting [illegible]. ∎

**Corollary:** If all the faces of a three-dimensional object have the same diameter and the same surface area, and the sensor error in measuring distances is much smaller than the diameter of a face, then the expected number of nodes at the $k^{th}$ level of the interpretation tree, $k \geq 2$, that survive distance pruning is bounded above by

$$\left(4\sqrt{\frac{A}{\pi d^2}}\right)^{k-1} n^{\frac{k+1}{2}}$$

where $A$ is the surface area of the object and $d$ is its diameter.

**Proof:** Since the sensor error in measuring distance is much less than the diameter of a face, we see that [illegible] is essentially given by the maximum over all $i, j$ [illegible]. From the previous proposition, this is bounded by the face diameter $\epsilon$. If $A_f$ is the surface area of the face, then $A_f \geq \pi(\frac{\epsilon}{2})^2$. Moreover, $A_f = \frac{A}{n}$, so that [illegible] $\leq 2\sqrt{\frac{A}{\pi n}}$ and the corollary follows. ∎

If the object is convex, then the area $A$ is bounded above by $\pi d^2$, and the upper bound reduces to

$$4^{k-1} n^{\frac{k+1}{2}}.$$

As in the two-dimensional case, non-convex objects can essentially be treated as convex ones, where the surface area of a convoluted object is replaced by the "effective surface area" of a nearly-convex one, and a similar bound will hold.

## 6.2. The Relevance of the Combinatorics

The key point to be stressed here is that the use of distance pruning can be shown to reduce the interpretation problem significantly. In principle, the problem of $k$ sensor points against a model of $n$ faces would result in $n^k$ possible interpretations that must be tested. We have shown that for two-dimensional objects, distance pruning reduces this to a number linear in $n$, and for three dimensional objects, the number is reduced to at most one proportional to $n^{(k+1)/2}$.

We also stress that this is a weak upper bound, in particular because the analysis does not consider the full constraint of distance pruning. The analysis given considers the sequential pruning obtained by iteratively applying the constraint imposed by the sensed distance between the $(k-1)^{st}$ sensed point and the $k^{th}$ one. Clearly, given $k$ sensed points, there are $\binom{k}{2}$ different distance constraints, and taking all of these into account should provide a tighter bound. Moreover, the bounds derived refer to the pruning due to a single type of constraint. Clearly, when all three constraints are used, we would expect the number of possible interpretations to be further reduced.

It was a surprise to the authors that weak upper bounds on the number of interpretations would be less than exponential in the number of sensed points, $k$ (for example in the three degree of freedom case, where the number of interpretations is linear in the number of sensed points). In our experience, however, many people find it surprising that any of the bounds should grow with $k$. Most people expect them to *decrease* with $k$, i.e., as more points are acquired, the constraint should be tighter. Recall, however, that the bounds derived above do not take into account the fact that there are $\binom{k}{2}$ distance constraints at the $k^{th}$ level of the tree; they only apply a single constraint at each level of the tree. There is another important effect that (partially) accounts for the growth in the number of interpretations with $k$. Namely, that for $k \leq 6$ each interpretation corresponds to a continuous range of positions and orientations. For example, for $k = 1$, each interpretation corresponds to the whole space of positions and orientations. As more points are added, the "volume" in the space of positions and orientations consistent with each interpretation decreases, but the number of these interpretations may increase (as they do between $k = 1$ and $k = 2$)[2].

# 7. Discussion

It is important to note that the algorithm described in this paper has quite low computational cost. The pruning algorithm is particularly efficient. The range tables store all the model information needed and pruning is done by simply

[2] We are indebted to John Canny for this observation.

comparing the ranges of values measured (plus or minus error estimates) with those in the tables. Therefore, no arithmetic is done during pruning (except for indexing into tables). It is only the model test that requires any significant computation and, therefore, the desire to minimize the number of times it has to be performed.

To illustrate this point, we have recorded actual run times for a number of simulations. While the times are clearly dependent on a number of factors, such as the type of machine, the specific algorithm, the object sensed, and so on, the order of magnitude of the run times helps illustrate the computational efficiency of the method. For example, using an implementation in Lisp running on a Symbolics 3600 Lisp Machine, simulations on the motor housing with angular error range of $\frac{\pi}{10}$ and positional error range of 0.05 took an average of 1.27 seconds to generate and prune the interpretation tree and an average of [illegible].17 seconds to perform the model check. The time required to generate and prune the tree is clearly dependent on the number of plausible interpretations and grows non-linearly with an increase in this number. The time required to perform model checking grows linearly with the number of interpretations to which such a check must be applied. The average time expended on each model check was 0.21 seconds. In general, the average time to complete the computation was under 5 seconds, for this particular implementation, although this number would occasionally be exceeded in sensing situations in which a large number of interpretations were possible.

The local constraint method developed here requires that all the sensory data be drawn from one object. This is difficult [illegible] or visual domain, when the object is in a bin among other objects. If the hypothesis is made that all the points belong to one object and no feasible interpretations are found, then one can tell that the hypothesis is wrong. Much more research is needed in this area, however.

Throughout the paper we have limited our attention to the number of interpretations, relative to one model, of data obtained from that object. To carry out recognition between several objects, one determines the number of legal interpretations of one set of data relative to multiple object models. This process can simply be performed sequentially on each model. One simple improvement is clearly possible. If one stores with each model the maximum distance between any of the faces, then if one of the measured distances is greater than this upper bound, the model can be discarded at once. This technique quickly separates large objects from small ones. Unfortunately, very small measured distances do not rule out large objects. A second method would be to use direction histograms to rule out certain models. For example, if the angle between two sensed normals was 30°, then a model of a cube would not be consistent with this data, and could quickly be excluded.

After generating and pruning the interpretation tree and performing the model test on each of the known objects, we have a listing of all the positions and orientations of all objects consistent with the measured data. At this point, further discrimination can be carried out by additional unguided sensing as before or by considering the alternatives and choosing a good place to sense next. The recognition problem that remains is now amenable to other techniques as well since it has been reduced to the much more tractable problem of differentiating among a class of objects in known positions and orientations.

## Acknowledgements

Philippe Brou contributed freely of his time and effort to provide us with the range data used in our experiments; we are very grateful. We also appreciated his comments on the presentation of this paper. We are very thankful to Bob Bolles and Berthold Horn for their detailed comments and suggestions on an earlier draft. We have also benefited from discussions with Olivier Faugeras.

## References

**Agin, G. J. and T. O. Binford** "Computer Description of Curved Objects," *Third International Joint Conference on Artificial Intelligence* , 1973, 629-639.

**Bajcsy, R.** "Computer Identification of Visual Surface," *Computer Graphics and Image Processing* 2, 2 (1973), 118-130.

**Bajcsy, R. and L. Liebermann** "Texture Gradient as a Depth Cue," *Computer Graphics and Image Processing* 5, 1 (1976), 52-67.

**Baker, H. H. and T. O. Binford** "Depth from Edge and Intensity Based Stereo," *Seventh International Joint Conference on Artificial Intelligence* , August 1981, 631-636.

**Binford, T. O.** "Sensor Systems for Manipulation," *Remotely Manned Systems Conference* , 1972.

**Bolles, R. C. and R. A. Cain** "Recognizing and Locating Partially Visible Objects: The Local-Feature-Focus Method," *Robotics Research* 1, 3 (1982), 57-82.

**Bolles, R. C., P. Horaud and M. J. Hannah** "3DPO: A Three-Dimensional Part Orientation System," *First International Symposium of Robotics Research* , Bretton Woods, N.H., August, 1983.

**Briot, M.** "The Utilization of an 'Artificial Skin' Sensor for the Identification of Solid Objects," *Ninth ISIR* , Washington, D. C., March 1979, 529-548.

**Briot, M., M. Renaud, and Z. Stojilkovic** "An Approach to Spatial Pattern Recognition of Solid Objects," *IEEE Transactions on Systems, Man, and Cybernetics* SMC-8, 9 (September 1978), 690-694.

**Brou, P.** Finding the Orientation of the Objects in Vector Maps, Ph. D. Thesis, Department of Elec. Engr. and Comp. Sci., Massachusetts Institute of Technology, August 1983.

**Buchanan, B., G. Sutherland, and E. A. Feigenbaum** "Heuristic DENDRAL: A Program for Generating Explanatory Hypotheses in Organic Chemistry," *Machine Intelligence 4, (B. Melzer and D. Michie, eds)* (1969).

**Dixon, J. K., S. Salazar, and J. R. Slagle** "Research on Tactile Sensors for an Intelligent Robot," *Ninth ISIR*, Washington D. C., March 1979, 507–518.

**Faugeras, O. D., et al.** "Toward a Flexible Vision System," *Robot Vision*, A. Pugh (editor), IFS Publications, United Kingdom, 1983.

**Faugeras, O. D. and M. Hebert** "A 3-D Recognition and Positioning Algorithm using Geometrical Matching between Primitive Surfaces," *Eighth International Joint Conference on Artificial Intelligence*, 1983, 996-1002.

**Gaston, P. C. and T. Lozano-Pérez** "Tactile Recognition and Localization Using Object Models," MIT Artificial Intelligence Laboratory, AIM-705, 1983.

**Grimson, W. E. L.** "A Computer Implementation of a Theory of Human Stereo Vision," *Philosophical Transactions of the Royal Society of London* *B* **292** (1981), 217-253.

**Grimson, W. E. L.** *From Images to Surfaces: A computational study of the human early vision system*, MIT Press, Cambridge Mass., 1981.

**Grimson, W. E. L.** "A Computational Theory of Visual Surface Interpolation," *Phil. Trans. Roy. Soc. Lond.* *B* **298** (1982), 395–427

**Grimson, W. E. L.** "An Implementation of a Computational Theory of Visual Surface Interpolation," *Computer Vision, Graphics and Image Processing* **22** (1983), 39–69.

**Grimson, W. E. L. and T. Lozano-Pérez** "A Combinatorial Analysis of Recognition and Localization using Object Models," MIT Artificial Intelligence Laboratory, to appear, 1983.

**Harmon, L. D** "Automated Tactile Sensing," *Robotics Research* **1**, 2 (Summer 1982), 3–32.

**Harmon, L. D.** "Touch-Sensing Technology: A review," Society of Manufacturing Engineers, MSR80-03, 1980.

**Hillis, W. D.** "A High-Resolution Image Touch Sensor," *Robotics Research* **1**, 2 (Summer 1982), 33–44.

**Ikeuchi, K., Horn, B. K. P., Nagata, S., Callahan, T. and O. Feingold** "Picking Up An Object From A Pile Of Objects," *First International Symposium on Robotics Research*, Bretton Woods, N.H., 1983.

**Ikeuchi, K. and B. K. P. Horn** "An Application of Photometric Stereo," *Sixth Intl. Joint Conf. on Artificial Intelligence*, 1979, 413–415.

**Ivancevic, N. S.** "Stereometric Pattern Recognition by Artificial Touch," *Pattern Recognition* **6** (1974), 77–83.

**Jarvis, R. A.** "A Perspective on Range Finding Techniques for Computer Vision," *IEEE Trans. on Pattern Analysis and Machine Intelligence* **PAMI-5**, 2 (March 1983), 122–139.

**Kender, J. R.** "Shape from Texture," Carnegie Mellon University Computer Science Report, CMU-CS-81-102, 1980.

**Kinoshita, G., S. Aida, and M. Mori** "A Pattern Classification by Dynamic Tactile Sense Information Processing," *Pattern Recognition* 7 (1975), 243.

**Korn, G. A. and T. M. Korn** *Mathematical Handbook for Scientists and Engineers* , McGraw-Hill, New York, 1968.

**Lewis, R. A. and A. R. Johnston** "A Scanning Laser Range Finder for a Robotic Vehicle," *Fifth Intl. Joint Conf. on Artificial Intelligence* , 1977, 762-768.

**Lozano-Pérez, T.** "Spatial Planning: A Configuration Space Approach," *IEEE Transactions on Computers* **C-32**, 2 (1983), 108-120.

**Marik, V.** "Algorithms of the Complex Tactile Information Processing," *Seventh Intl. Joint Conf. on Artificial Intelligence* , 1981, 773-774.

**Mayhew, J.E.W. and J.P. Frisby** "Psychophysical and Computational Studies towards a Theory of Human Stereopsis," *Artificial Intelligence* 17 (1981), 349-385.

**Nevatia, R. and T. O. Binford** "Description and Recognition of Curved Objects," *Artificial Intelligence* 8 (1977), 77-98.

**Nitzan, D., A. E. Brain, and R. O. Duda** "The Measurement and Use of Registered Reflectance and Range Data in Scene Analysis," *Proc. of IEEE* **65** (February 1977), 206-220.

**Okada, T. and S. Tsuchiya** "Object Recognition by Grasping," *Pattern Recognition* 9, 3 (1977), 111-119.

**Oshima, M. and Y. Shirai** "A Scene Description Method using Three-Dimensional Information," *Pattern Recognition* **11** (1978), 9-17.

**Oshima, M. and Y. Shirai** "Object Recognition Using Three-dimensional Information," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **PAMI-5**, 4 (July, 1983), 353-361.

**Overton, K.J. and T. Williams** "Tactile Sensation for Robots," *Seventh Intl. Joint Conf. on Artificial Intelligence* , 1981, 791-795.

**Ozaki, H., S. Waku, A. Mohri, and M. Takata** "Pattern Recognition of a Grasped Object by Unit-Vector Distribution," *IEEE Transactions on Systems, Man, and Cybernetics* **SMC-12**, 3 (May/June 1982), 315-324.

**Page, C. J., A. Pugh, and W. B. Heginbotham** "Novel Techniques for Tactile Sensing in a Three-Dimensional Environment," *Sixth ISIR* , University of Nottingham, March 1976.

**Popplestone, R. J., C. M. Brown, A. P. Ambler, and G. F. Crawford** "Forming Models of Plane and Cylinder Faceted Bodies from Light Stripes," *Fourth Intl. Joint Conf. on Artificial Intelligence* , Tbilisi, Georgia, USSR, September 1975, 664-668.

**Purbrick, J. A.** "A Force Transducer Employing Conductive Silicone Rubber," *First International Conference on Robot Vision and Sensory Controls* , Stratford-upon-Avon, United Kingdom, April, 1981.

**Raibert, M. H. and J. E. Tanner** "Design and Implementation of a VLSI Tactile Sensing Computer," *Robotics Research* 1, 3 (1982), 3-18.

**Shirai, Y. and M. Suwa** "Recognition of Pohyhedrons with a Range Finder," *Second Intl. Joint Conf. on Artificial Intelligence* , 1971.

**Schneiter, J. L.** An Optical Tactile Sensor for Robots, S. M. Thesis, Department of Mech. Engr., Massachusetts Institute of Technology, August 1982.

**Shneier, M.** "A Compact Relational Structure Representation," *Sixth International Joint Conference on Artificial Intelligence* , 1979, 818-826.

**Snyder, Wesley E. and J. St. Clair** "Conductive Elastomers as Sensor for Industrial Parts Handling Equipment," *IEEE Transactions on Instrumentation and Measurement* IM-27, 1 (March 1978), 94-99.

**Stevens, K. A.** "The Information Content of Texture Gradients," *Biological Cybernetics* 42 (1981), 95-105.

**Stojilkovic, Z. and D. Saletic** "Learning to Recognize Patterns by Belgrade Hand Prosthesis," *Fifth ISIR* , IIT Research Institute, Chicago, 1975, 407-413.

**Sugihara, K.** "Range-data Analysis Guided by a Junction Dictionary," *Artificial Intelligence* 12 (1979), 41-69.

**Takeda, S.** "Study of Artificial Tactile Sensors for Shape Recognition: Algorithm for Tactile Data Input," *Fourth ISIR* , 1974, 199-208.

**Terzopoulos, D.** "Multilevel Computational Processes for Visual Surface Reconstruction," *Computer Vision, Graphics and Image Processing* (1983), to appear.

**Woodham, R. J.** "Photometric Stereo: A Reflectance Map Technique for Determining Surface Orientation from Image Intensity," *Image Understanding Systems and Industrial Applications, Proc. SPIE* 155 , 1978.

**Woodham, R. J.** "Photometric Method for Determining Surface Orientation from Multiple Images," *Optical Engineering* 19, 1 (1980), 139-144.

**Woodham, R. J.** "Analysing Images of Curved Objects," *Artificial Intelligence* 17 (1981), 117-140.

## Appendix I

Here, we establish the claim of section 2.1.2 that the set

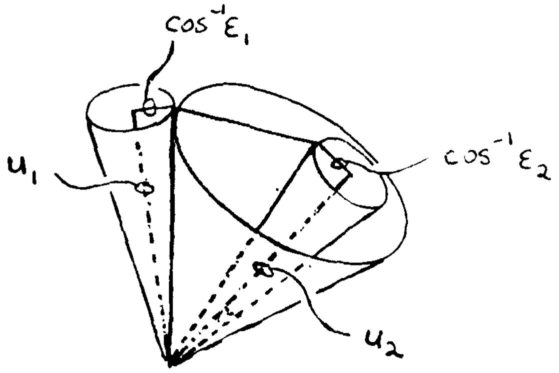
$$\{\mathbf{n}_1 \cdot \mathbf{n}_2 \mid \mathbf{n}_1 \cdot \mathbf{u}_1 \geq \epsilon_1, \quad \mathbf{n}_2 \cdot \mathbf{u}_2 \geq \epsilon_2\}$$

is contained in the set

$$\{\mathbf{n}_1 \cdot \mathbf{n}_2 \mid \cos[\min(\pi, \theta_{12} + \phi_1 + \phi_2)] \leq \mathbf{n}_1 \cdot \mathbf{n}_2 \leq \cos[\max(0, \theta_{12} - \phi_1 + \phi_2)]\}$$

where

Figure 11. Extremal values of dot products between two cones



$$\cos\phi_1 = \epsilon_1$$
$$\cos\phi_2 = \epsilon_2$$
$$\cos\theta_{12} = \mathbf{u}_1 \cdot \mathbf{u}_2 = \gamma.$$

While it is possible to prove this algebraically, it is simpler to see this by the following geometric construction (see Figure 11). We wish to determine the extremal values of the dot product between unit vectors in the two cones, or equivalently, extremal values in the angle between any two such vectors. If the cones about $\mathbf{u}_1$ and $\mathbf{u}_2$ intersect, clearly the maximum value of the dot product is 1. If the cones are antipodal, clearly the minimum value is $-1$.

We now consider cases in which the cones do not overlap. We claim that the extremal values for the dot product occur when the two vectors lie in the plane spanned by $\mathbf{u}_1$ and $\mathbf{u}_2$, with the vectors lying at the limits of the cone within this plane. That is, if we let

$$\rho_i = \sqrt{\frac{1-\epsilon_i^2}{1-\gamma^2}}$$

then the extrema occur at

$$\mathbf{n}_1 = (\epsilon_1 - \gamma\rho_1)\mathbf{u}_1 + \rho_1\mathbf{u}_2$$
$$\mathbf{n}_2 = \rho_2\mathbf{u}_1 + (\epsilon_2 - \gamma\rho_2)\mathbf{u}_2$$

and

$$\mathbf{n}_1 = (\epsilon_1 + \gamma\rho_1)\mathbf{u}_1 - \rho_1\mathbf{u}_2$$
$$\mathbf{n}_2 = -\rho_2\mathbf{u}_1 + (\epsilon_2 + \gamma\rho_2)\mathbf{u}_2$$

The first case can be shown to correspond to the minimal angle between vectors in the two cones, by the following construction. Construct a cone centered about $\mathbf{n}_1$ with radius such that $\mathbf{n}_2$ lies on the boundary of the cone, that is the new cone grazes the $\mathbf{u}_2$ cone at $\mathbf{n}_2$. If there is a smaller angle, it must be possible to reposition this cone so that it is centered at some other point in the $\mathbf{u}_1$ cone and yet still intersects the $\mathbf{u}_2$ cone. This is clearly not possible, and hence the minimum value of the dot product is given by the stated choice of $\mathbf{n}_1$ and $\mathbf{n}_2$. Expanding the dot product for this case, and making the appropriate trigonometric substitutions yields the required expression. A similar construction holds for the maximum angle (or minimum dot product).

## Appendix II

Here, we show how to compute the range of possible direction vectors between $\mathit{face}_i$ and $\mathit{face}_j$ in the object model. Let us erect a coordinate system on $\mathit{face}_i$ at the centroid of the face and whose $z$ axis points along the normal of the face. Then, it is clear that the set of possible direction vectors is the set

$$\{\mathbf{v}_j - \mathbf{v}_i \mid \mathbf{v}_j \in \mathit{face}_j \;\&\; \mathbf{v}_i \in \mathit{face}_i\}$$

where both $\mathbf{v}_i$ and $\mathbf{v}_j$ are expressed relative to the frame on $\mathit{face}_i$. Assume, for now, that both faces are convex. It can be shown [Lozano-Pérez 83] that this set is equivalent to

$$ch(\{\mathbf{v}_j - \mathbf{v}_i \mid \mathbf{v}_j \in \mathit{vert}(\mathit{face}_j) \;\&\; \mathbf{v}_i \in \mathit{vert}(\mathit{face}_i)\})$$

where $ch()$ is the convex hull of a set of points and $\mathit{vert}()$ is the set of vertices of a face. Because of convexity, the extrema of the component of the direction vectors along the normal of $\mathit{face}_i$ occur at the vertices of this convex hull. Clearly, the vertices of the convex hull of a set of points are drawn from the set of points itself. Therefore, we need only find the extrema of the finite set

$$\{\mathbf{n}_i \cdot (\mathbf{v}_j - \mathbf{v}_i) \mid \mathbf{v}_j \in \mathit{vert}(\mathit{face}_j) \;\&\; \mathbf{v}_i \in \mathit{vert}(\mathit{face}_i)\}$$

where $\mathbf{n}_i$ is the normal to $\mathit{face}_i$.

When the faces are non-convex, the procedure above will generate a conservative bound.

LMED

8