

MODEL-BASED RECOGNITION AND LOCALIZATION FROM TACTILE DATA

W. Eric L. Grimson
Tomás Lozano-Pérez

Massachusetts Institute of Technology
Artificial Intelligence Laboratory
545 Technology Square
Cambridge, Massachusetts

Abstract. This paper discusses how local measurements of three-dimensional positions and surface normals recorded by a set of tactile 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 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.

1. The Problem and the Approach

The presence of significant uncertainty about the identities and positions of objects in the workspace of the robot is a central characteristic of advanced applications in robotics, and makes sensing of the external environment an essential component of robot systems. The process of sensing can be loosely divided into two stages: the measurements of properties of the objects in the environment, and the interpretation of those measurements. In the present paper, we concentrate on the interpretation of sensory data, from tactile sensors. 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, and 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 tactile 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 tactile sensor 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

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.

sensor 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.

1.2. Approach

A recent paper [8] 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. A more complete and detailed exposition of this generalization may be found in [9].

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 [5].

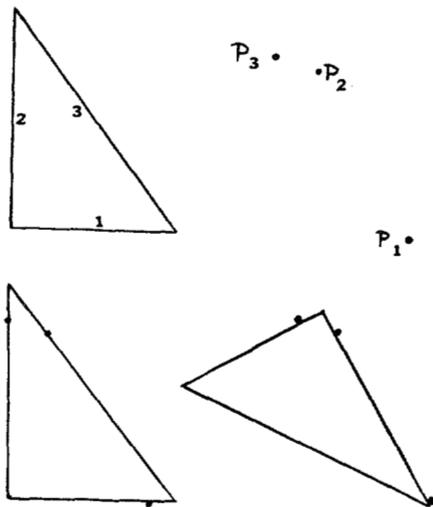


Figure 1. An example of the approach

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 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 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).

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, although we concentrate on the case of 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 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 sensors*, 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 [10] for a review and [12, 18, 21, 22, 23] for some recent designs.

For descriptions of previous work in tactile sensing, we refer the reader to two very thorough surveys by Harmon [10,11]. A more detailed discussion of previous work on tactile recognition can be found in [8]. 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 [3, 17]. Others have used the joint angles of fingers grasping the object as their data [4, 16, 17, 25]. A related approach uses the pattern of activation of on-off contacts placed on the finger links [14].

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 descriptor 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 [1, 12, 24]. Another approach is to build surface

models, either from pressure distributions on matrix sensors [18], or from the displacements of an array of needle-like sensors [20, 26]. A related approach builds a representation of an object's cross section [19, 14].

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 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 [24]. This argues that a stiff VLSI sensor (e.g. [22]) 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 (very different approaches to tactile recognition based on this type of data are outlined in [6, 13]). 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.

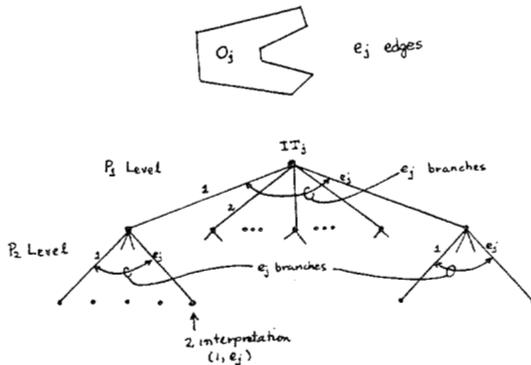


Figure 2. Interpretation Tree

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) [8]. The root node of the 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 [8]. 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 s -interpretations of the data. Other constraints are possible (see, for example, [9]). We will focus on the three constraints above, primarily because they are simple to implement while being quite effective. In particular, they can be used to prune k -interpretations, for $k \geq 2$, thereby collapsing whole subtrees of the IT, without explicitly exploring all the nodes of that subtree, yielding a large computational savings.

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 [2]). 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.

Since the distance ranges between faces can be computed off-line, it is straightforward to implement the distance constraint as a table lookup. In this manner, a pair of faces are consistent with a pair of measurements if the distance between the measured points lies within the range recorded in the appropriate entry of the precomputed table.

We note that it may frequently be the case, e.g. for a flat 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.

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 implement this, we use the following technique. If u_1 denotes the unit sensed surface normal at a sensed point P_1 , the range of possible values for the actual surface normal will be denoted by the right circular cone

$$\{n_1 \mid n_1 \cdot u_1 \geq \epsilon_1\}.$$

A similar cone describes the set of possible surface normals, in hand coordinates, for a second sensed point P_2 . Then, in order for faces i and k , with associated surface normals v_i and v_k to be consistent with these sensed points, it must be the case that

$$v_i \cdot v_k \in \{n_1 \cdot n_2 \mid n_1 \cdot u_1 \geq \epsilon_1, n_2 \cdot u_2 \geq \epsilon_2\}. \quad (1)$$

If $\cos \alpha_1 = \epsilon_1$, $\cos \alpha_2 = \epsilon_2$, $\alpha_{12} = \alpha_1 + \alpha_2$ and $\cos \gamma_{12} = u_1 \cdot u_2$, then the set of equation (1) is contained in the set

$$\{n_1 \cdot n_2 \mid \cos[\min(\pi, \gamma_{12} + \alpha_{12})] \leq n_1 \cdot n_2 \leq \cos[\max(0, \gamma_{12} - \alpha_{12})]\}. \quad (2)$$

Figure 3 illustrates this result in two dimensions.

Since we can now place bounds on the range of possible dot products between surface normals, a table-lookup implementation of angle pruning similar to that used for distance pruning is now also possible.

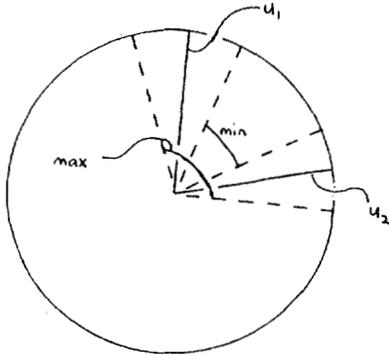


Figure 3. Angle Ranges

2.1.3. Direction Pruning

Consider a pair of sensed points P_1 and P_2 and let 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 w_1 , and the range of possible values for the surface normal is

$$\{v_1 \mid v_1 \cdot w_1 \geq \epsilon_1\}.$$

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

$$\{v_1 \cdot u_{12} \mid v_1 \cdot w_1 \geq \epsilon_1\}. \quad (3)$$

In an interpretation, suppose that point P_1 has been assigned to face i , with normal 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

$$\{s_{ik} \mid s_{ik} \cdot t_{ik} \geq \delta_{ik}\}$$

for some pair t_{ik} and δ_{ik} . Figure 4 illustrates this cone in a two-dimensional example. 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

$$\{n_i \cdot s_{ik} \mid s_{ik} \cdot t_{ik} \geq \delta_{ik}\}. \quad (4)$$

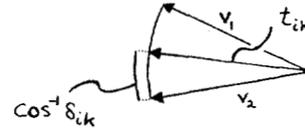
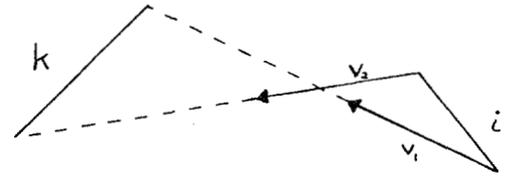


Figure 4. Range of Directions between Sensed Points

Thus, assume that point P_1 is on face i , with normal n_i , that we have measured w_1 , that we know ϵ_1 , and that we have also measured P_2 . A face k , whose direction range from face i is given by the pair (t_{ik}, δ_{ik}) , is a feasible face for point P_2 if the set in equation (4) intersects the cone of equation (3). If $\cos \gamma_{ik} = \delta_{ik}$, and $\cos \phi_{ik} = n_i \cdot t_{ik}$, then the set of equation (4) is contained in the set

$$\{n_i \cdot s_{ik} \mid \cos(\gamma_{ik} + \phi_{ik}) \leq n_i \cdot s_{ik} \leq \cos(\gamma_{ik} - \phi_{ik})\}.$$

Similarly, if $\cos \alpha_1 = \epsilon_1$ and $\cos \omega_{12} = v_1 \cdot u_{12}$, then the set of equation (3) is contained in the set

$$\{v_1 \cdot u_{12} \mid \cos(\alpha_1 + \omega_{12}) \leq v_1 \cdot u_{12} \leq \cos(\alpha_1 - \omega_{12})\}.$$

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.,

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

The direction constraint can also be implemented in a form similar to that used for distance and angle pruning.

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 limits the angle between a surface normal and unit vectors from one face to another. We may also 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 $u_{i,j}$ and $t_{i,k}$ 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 give rise to a cone of directions, but a single value for the normal component of vectors connecting the faces. Note that an interpretation 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.

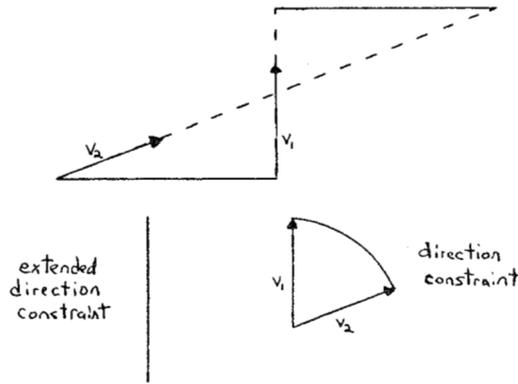


Figure 5. Extended Direction Constraint

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 (p_i, n_i, f_i) , where p_i is the vector representing the sensed position, 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:

$$v_s = Rv_m + v_0$$

where R is a rotation matrix, and v_0 is some translation vector. We need to solve for R and v_0 . We note that a solution could be obtained using a least-squares method, such as is used by [7]. 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, (p_i, n_i, f_i) . The sensed normal is given by v_i and corresponding to face f_i is a face normal m_i . For R to be a legitimate rotation, it should take the normal m_i into 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 r consistent with n_i and m_i ? Any unit rotation vector r taking m_i into n_i must lie on the perpendicular bisector of the line connecting n_i to m_i . Similarly, it must also lie on the perpendicular bisector of the line connecting n_j to 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 unit vector

$$(m_i - n_i) \times (m_j - n_j)$$

to within an ambiguity of 180° .

If there were no error in the sensed normals, we would be done. With error included in the measurements, however, the computed rotation direction r could be slightly wrong. One way to reduce the effect of this error is to compute all possible $r_{i,j}$ 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 r .

Once we have computed a direction of rotation r , we need to determine the angle θ of rotation about it. It is straightforward to show that (see, for example, [15] p. 473)

$$m_i = \cos \theta n_i + (1 - \cos \theta)(r \cdot n_i)r + \sin \theta(r \times n_i).$$

Simple algebraic manipulation, using the fact that $r \cdot m_i = r \cdot n_i$, yields

$$\cos \theta = 1 - \frac{1 - (n_i \cdot m_i)}{1 - (r \cdot n_i)(r \cdot m_i)}$$

$$\sin \theta = \frac{(r \times n_i) \cdot m_i}{1 - (r \cdot n_i)(r \cdot m_i)}$$

Hence, given r , we can solve for θ . Note that if $\sin \theta$ is zero, there is a singularity in determining θ , which could be either 0 or π . In this case, however, r lies in the plane spanned by n_i and m_i and hence, only the $\theta = \pi$ solution is valid. As before, in the presence of error, we may want to cluster the r vectors, and then take the average of the computed values of θ over this cluster.

Finally, given values for both r and θ , if r_x, r_y, r_z denote the components of r , then the rotation matrix R is given by

$$\cos \theta I + (1 - \cos \theta) \begin{bmatrix} r_x^2 & r_x r_y & r_x r_z \\ r_y r_x & r_y^2 & r_y r_z \\ r_z r_x & r_z r_y & r_z^2 \end{bmatrix} + \sin \theta \begin{bmatrix} 0 & -r_z & r_y \\ r_z & 0 & -r_x \\ -r_y & r_x & 0 \end{bmatrix}$$

where I is the 3×3 identity matrix. 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, (r, θ) and $(-r, -\theta)$. 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 $v_s = Rv_m + v_0$, where v_m is a vector in model coordinates, v_s is the corresponding vector in sensor coordinates, and R has been computed as above. Given a triple (p_i, n_i, f_i) from the interpretation, let m_i be the normal of face f_i , with offset d_i , that is, the face is defined by the set of vectors

$$\{v \mid v \cdot m_i = d_i\}.$$

Then by transforming p_i to its corresponding point in model coordinates, the following equation holds

$$(Rm_i) \cdot (p_i - v_0) = d_i.$$

This equation essentially constrains the component of the translation vector in the direction of Rm_i .

Suppose we consider three triplets from the interpretation, (p_i, n_i, f_i) , (p_j, n_j, f_j) , and (p_k, n_k, f_k) such that the triple product $m_i \cdot (m_j \times m_k)$ is non-zero, (i.e. the three face normals are independent). Then, we can construct three independent equations for the components of v_0 in the direction of (Rm_i) , (Rm_j) and (Rm_k) . Hence, the three equations together determine the actual vector v_0 , as given by:

$$\begin{aligned} [m_i \cdot (m_j \times m_k)]v_0 = & ((Rm_i) \cdot p_i - d_i)((Rm_j) \times (Rm_k)) \\ & + ((Rm_j) \cdot p_j - d_j)((Rm_k) \times (Rm_i)) \\ & + ((Rm_k) \cdot p_k - d_k)((Rm_i) \times (Rm_j)) \end{aligned}$$

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 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 $m_i \cdot (m_j \times m_k)$ is greater than some threshold.

Finally, we have computed the transform (R, v_0) from model coordinates to sensor coordinates. To check a possible interpretation, we consider all triples (p_i, n_i, f_i) in the interpretation and compute

$$R^{-1}(p_i - v_0).$$

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 intersecting 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, which are reported in full in [9]. Here, we describe those simulations relevant to the problem of tactile sensing of 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.

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 6. The diameters of these objects

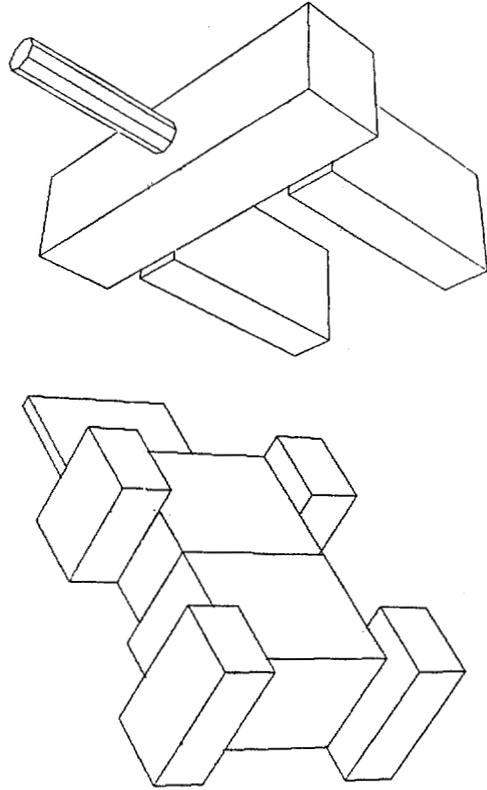


Figure 6. 3D Test Models

(that is the maximum separation of two points on the object) were roughly 4 and 8 inches for the housing and simple hand respectively. We note that the simulated sensitivity in distance (as recorded in the tables below) is well 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.

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.

The set of simulations reported below have been run using a sensing strategy consistent with tactile sensors, mounted on a multi-fingered hand. 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 intersect the object). Each ray is traced along its 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 I, II and III summarize the results of running sets of simulations, using sensory data generated in this fashion. Table I lists statistics of a histogram of the number of interpretations remaining in the tree after local pruning, Table II lists similar statistics after the interpretations have been subjected to a model

test, and Table III lists similar statistics when the results of Table II are clustered on the basis of their computed transformations. In this particular case, two transformations are considered the same, if their directions of rotation differ by less than 1.5°.

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

Object	Normal	Dist	Min	50th	90th	Max	Faces
Simple Hand	$\pi/10$.01	2	4	12	60	28
		.05	2	4	24	116	28
	$\pi/8$.01	2	4	24	98	28
		.05	2	7	39	160	28
Housing	$\pi/10$.01	1	4	32	516	34
		.05	1	16	80	606	34
	$\pi/8$.01	1	8	26	136	34
		.05	1	32	164	377	34

Object	Normal	Dist	Min	50th	90th	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	1	5	9	34
		.05	1	3	10	14	34

In the tables 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 90th percentile of the set of simulations; and the *faces* column lists the number of faces in the model.

The effectiveness of the local constraints in reducing the number of feasible interpretations is clearly demonstrated, since the average number of interpretations tends to be very close to 1. The fact that occasionally additional interpretations are still possible 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. For an object such as the simple hand, the complete rotational symmetry of the object forces at least two distinct interpretations of the sensory data, for any set of sensed points.

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 [8]. Extensions of the method to the six degree of freedom problem are currently under investigation.

5. 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 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 must 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 $\pi/8$ 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 3.17 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 0.24 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 to guarantee, in the tactile or visual domain, when the object is in a bin among other objects. Of course, if a 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

We are very thankful to Bob Bolles, Philippe Brou, John Hollerbach and Berthold Horn for their detailed comments and suggestions on an earlier draft. We have also benefited from discussions with Olivier Faugeras.

References

- [1] T. O. Binford, "Sensor Systems for Manipulation," *Remotely Manned Systems Conference*, 1972.
- [2] R. C. Bolles and R. A. Cain, "Recognizing and Locating Partially Visible Objects: The Local-Feature-Focus Method," *Robotics Research*, vol.1, no.3, pp. 57-82, 1982.
- [3] M. Briot, "The Utilization of an 'Artificial Skin' Sensor for the Identification of Solid Objects," *Ninth ISIR*, pp. 529-548, Washington, D. C., March, 1979.
- [4] M. Briot, M. Renaud, and Z. Stojilkovic, "An Approach to Spatial Pattern Recognition of Solid Objects," *IEEE Transactions on Systems, Man, and Cybernetics*, vol.SMC-8, no.9, pp. 690-694, September, 1978.
- [5] B. Buchanan, 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.
- [6] J. K. Dixon, S. Salazar, and J. R. Slagle, "Research on Tactile Sensors for an Intelligent Robot," *Ninth ISIR*, pp. 507-518, Washington D. C., March, 1979.
- [7] O. D. Faugeras, and M. Hebert, "A 3-D Recognition and Positioning Algorithm using Geometrical Matching between Primitive Surfaces," *Eighth International Joint Conference on Artificial Intelligence*, pp. 996-1002, 1983.
- [8] P. C. Gaston, and T. Lozano-Pérez, "Tactile Recognition and Localization Using Object Models," MIT Artificial Intelligence Laboratory, Report AIM-705, 1983.
- [9] W. E. L. Grimson, and T. Lozano-Pérez, "Model-Based Recognition and Localization From Sparse Range or Tactile Data," MIT Artificial Intelligence Laboratory, Report AIM-738, 1983.
- [10] L. D. Harmon, "Automated Tactile Sensing," *Robotics Research*, vol.1, no.2, pp. 3-32, Summer 1982.
- [11] L. D. Harmon, "Touch-Sensing Technology: A review," Society of Manufacturing Engineers, Report MSR80-03, 1980.
- [12] W. D. Hillis, "A High-Resolution Image Touch Sensor," *Robotics Research*, vol.1, no.2, pp. 33-44, Summer 1982.
- [13] N. S. Ivancevic, "Stereometric Pattern Recognition by Artificial Touch," *Pattern Recognition*, vol.6, pp. 77-83, 1974.
- [14] G. Kinoshita, S. Aida, and M. Mori, "A Pattern Classification by Dynamic Tactile Sense Information Processing," *Pattern Recognition*, vol.7 pp. 243, 1975.
- [15] G. A. Korn, and T. M. Korn, *Mathematical Handbook for Scientists and Engineers*. New York: McGraw-Hill, 1968.
- [16] V. Marik, "Algorithms of the Complex Tactile Information Processing," *Seventh Intl. Joint Conf. on Artificial Intelligence*, pp. 773-774, 1981.
- [17] T. Okada, and S. Tsuchiya, "Object Recognition by Grasping," *Pattern Recognition*, vol.9, no.3, pp. 111-119, 1977.
- [18] K. J. Overton, and T. Williams, "Tactile Sensation for Robots," *Seventh Intl. Joint Conf. on Artificial Intelligence*, pp. 791-795, 1981.
- [19] H. Ozaki, S. Waku, A. Mohri, and M. Takata, "Pattern Recognition of a Grasped Object by Unit-Vector Distribution," *IEEE Transactions on Systems, Man, and Cybernetics*, vol.SMC-12, no.3, pp.315-324, May/June 1982.
- [20] C. J. Page, A. Pugh, and W. B. Heginbotham, "Novel Techniques for Tactile Sensing in a Three-Dimensional Environment", *Sixth ISIR*, University of Nottingham, March 1976.
- [21] J. A. Purbrick, "A Force Transducer Employing Conductive Silicone Rubber," *First International Conference on Robot Vision and Sensory Controls*, Stratford-upon-Avon, United Kingdom, April, 1981.
- [22] M. H. Raibert, and J. E. Tanner, "Design and Implementation of a VLSI Tactile Sensing Computer," *Robotics Research*, vol.1, no.3, pp.3-18, 1982.
- [23] J. L. Schneider, "An Optical Tactile Sensor for Robots," S. M. Thesis, Dept. of Mech. Engr., MIT, August 1982.
- [24] W. E. Snyder, and J. St. Clair, "Conductive Elastomers as Sensor for Industrial Parts Handling Equipment," *IEEE Transactions on Instrumentation and Measurement*, vol.IM-27, no.1, pp.94-99 March 1978.
- [25] Z. Stojilkovic, and D. Saletic, "Learning to Recognize Patterns by Belgrade Hand Prosthesis," *Fifth ISIR* pp.407-413, IIT Research Institute, Chicago, 1975.
- [26] S. Takeda, "Study of Artificial Tactile Sensors for Shape Recognition: Algorithm for Tactile Data Input," *Fourth ISIR*, pp. 199-208, 1974.