MASSACHUSETTS INSTITUTE OF TECHNOLOGY
ARTIFICIAL INTELLIGENCE LABORATORY

A. I. Memo 705 March, 1983

Tactile Recognition and Localization Using Object Models: The Case of Polyhedra on a Plane

Peter C. Gaston
Tomás Lozano-Pérez

Abstract. This paper discusses how data from multiple tactile sensors may be used to identify and locate one object, from among a set of known objects. We use only local information from sensors: (1) the position of contact points, and (2) ranges of surface normals at the contact points. The recognition and localization process is structured as the development and pruning of a tree of consistent hypotheses about pairings between contact points and object surfaces. In this paper, we deal with polyhedral objects constrained to lie on a known plane, i.e., having three degrees of positioning freedom relative to the sensors.

Acknowledgements. This report describes research done in the Artificial Intelligence Laboratory of the Massachusetts Institute of Technology. Support for the Laboratory's Artificial Intelligence research is provided in part by the Office of Naval Research under Office of Naval Research contract N00014–81–K–0494 and in part by the Advanced Research Projects Agency under Office of Naval Research contracts N00014–80–C–0505 and N00014–82–K–0334. Part of the research reported here was carried out while one of the authors (PCG) was a co-op student at Digital Equipment Corporation.

1. Tactile Sensing

Tactile information is useful for locating and identifying objects, determining the texture, hardness, and temperature of objects, and detecting slippage of a grasped object. These capabilities are particularly important when visual information is not readily available as is the case, for example, in underwater manipulation and during the process of grasping an object from a bin of parts. A large number of tactile sensing applications are discussed in a recent survey of the state of the art in tactile sensing research [Harmon 1982].

In this paper we will consider a limited subset of robotic tactile recognition. In particular, we consider how information from several tactile sensors may be used to identify which object, from among a set of known objects, has been grasped and to determine the object's position and orientation relative to the hand. In the recognition process we limit ourselves to using very local information from sensors: (1) the position of a few contact points, and (2) ranges of surface normals at the contact points.

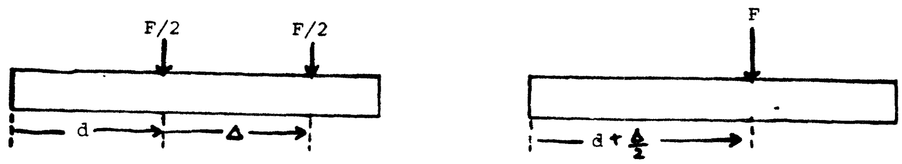
We propose a scheme for concurrent recognition and localization that is simple to implement and has low computational cost. Our primary motivation in this paper is to illustrate that tactile recognition and localization can be done without resorting to statistical pattern recognition or global feature-finding. Statistical pattern recognition, on the one hand, ignores much of the geometric constraint available from object models and cannot be used to locate objects. Global feature-finding, on the other hand, may require the sensor to explore large segments of an object's surface, which is a slow process. A parallel goal is to show that recognition and localization are feasible using data from small, stiff sensors with poor force resolution, but high spatial resolution. We feel that the viability of this recognition approach has important implications on the design of tactile sensors. In particular, it shows the importance of obtaining some constraint on the surface normal at the point of contact.

1.1. Tactile Sensors and Tactile Data

A tactile sensor is a device that can detect the location and, possibly, the forces of contact with an object. 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. We make the distinction between *tactile sensors*, which measure forces at specific points, and *force sensors*, which measure the total forces and torques on some structure. The simple example in Figure 1 illustrates this distinction; the two force systems illustrated there would be equivalent to a force sensor, but distinguishable by an array of tactile sensors.

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 [Harmon 82] for

Figure 1. Tactile sensing versus force sensing.



a review and [Hillis 82, Overton and Williams 81, Raibert and Tanner 82] for some recent designs.

A matrix tactile sensor produces an array of measurements that are a function of the pressure distribution over the sensor. The exact relationship of these measurements to properties of the object is very complex and depends on the particular sensor design [Binford 72, Snyder and St. Clair 78, Stojilkovic and Clot 77]. In practice, the presence of electrical noise, vibrations, limited resolution, and unmodeled compliance make it difficult to determine, much less invert, this relationship in detail. Because of this difficulty in directly interpreting individual tactile data elements, especially from today's sensors, existing approaches to tactile recognition have relied on alternative sources of information (except see [Kinoshita, Aida, and Mori 75]). The two principal styles are those based on statistical pattern recognition and those that build explicit models from the data and match them to object descriptions.

Much of the existing work on tactile recognition has been based on statistical pattern recognition or classification. Some researchers have relied on the contact patterns on matrix sensors [Briot 79, Okada and Tsuchiya 77]. The assumption motivating this line of research has been that the individual (local) data elements are not repeatable and only their statistical parameters can be counted on. The measured statistics are then compared to reference statistics for the known object types. The resulting methods are limited to discriminations among a few simple types of objects.

A second approach to statistical tactile recognition uses patterns of the positions in which the fingers of articulated hands come to rest against the object. A number of researchers have used the joint angles of the fingers as their primary data [Briot, Renaud, and Stojilkovic 78, Marik 81, Okada and Tsuchiya 77, Stojilkovic and Saletic 75] grasping the object. A related approach classifies the pattern of activation of on-off contacts placed on the finger links [Kinoshita, Aida, and Mori 75].

Several tactile recognition methods have been proposed that attempt to build a partial description of the object from the sense data and to match this description to the model. Individual approaches differ on the type of description used.

One group emulates the feature-based approach that has been successful in vision systems. The idea is that the pattern of measurements on a matrix sensor can be used to identify global object features, such as holes, edges, vertices, pits, and burrs [Binford 72, Hillis 82, Snyder and St. Clair 78]. These features may be difficult to locate and identify for objects that are significantly larger than the sensor, however. In particular, it may be difficult to integrate successive sensor readings to obtain reliable features.

Another group attempts 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]. These methods must face the rather complex problem of matching the surface descriptions obtained from the data to those of a model. A related approach that simplifies matching has been to build a representation of subsets of an object's cross-section and match them to object models [Ozaki et al 82, Kinoshita, Aida, Mori 75]. The method described in [Ozaki et al 82] is particularly interesting in this respect as it represents both objects and data as a sequence of unit surface tangents indexed by angle. This representation is invariant with translations and simply shifts with rotation, thus simplifying the matching process.

Note that, in many cases, the tactile sensors are used only to detect contact; it is the relative position of sensors to objects that is the actual source of data. The method described in this paper also uses relative positions, rather than two-dimensional patterns of contacts, as its primary data. The key differences from the methods outlined above are:

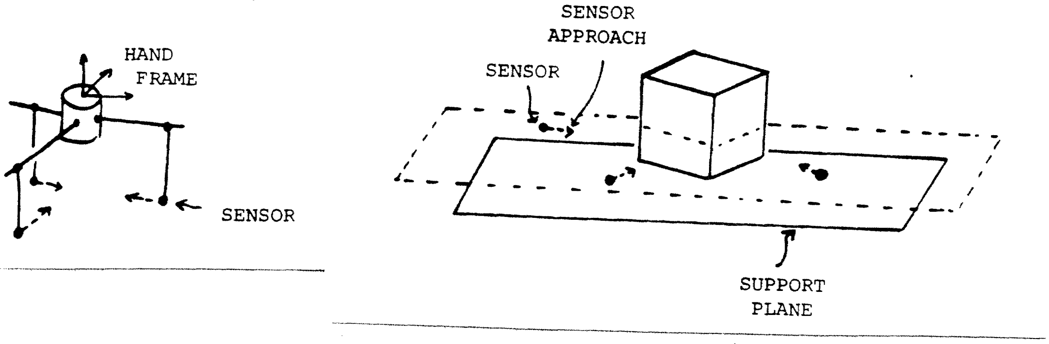
1. Our method uses very sparse data: one point from each sensor.
2. Our method exploits the geometric constraints obtained from complete object models.

The data we use for recognition and localization are estimates of the position and normal vector of a few points on the surface of the touched object:

1. Surface point — On the basis of sensor readings, some points on the sensor can be identified as being in contact with external objects. In real sensors, there is some uncertainty as to the actual contact point, but its position can be constrained within some small area. If the sensor's shape and location in space are known, one can determine the position of some point on the touched object, to within some uncertainty volume.
2. Surface normal — At the contact points, the known surface normal to the sensor must be the negative of the object's surface normal at that point. This is exactly true only for a rigid sensor and object in the absence of measurement error. In practice, weaker but still useful constraints on the surface normal can be recovered.

We do not dicuss how this data may be obtained from actual sensor data, since this process is completely sensor-dependent. Our aim is to show, instead, how such data may be used in conjunction with object models to recognize and localize objects. Different approaches to tactile recognition based on this type of data are outlined in [Dixon, Salazar, and Slagle 79, Ivancevic 74].

Figure 2. Hand geometry



Position and normal data can be obtained reliably only if the tactile sensors have high spatial resolution; such sensors are currently under development. The sensor described by [Hillis 82], for example, has 256 sensitive points on an area of one square centimeter. Sensors with even higher resolutions are feasible. Fortunately, the information required by our recognition method is very local, so the sensor need not be large. A related requirement on the sensor is that it be fairly stiff; otherwise, the accuracy of the position and normal information will suffer.

Tactile sensors, by their very nature, provide information over a relatively small area of an object. This limitation is overcome either by mechanically scanning the sensor, which is slow, or by using multiple sensors. In this paper, we assume that a small number of sensors, typically three, are used in conjunction. The three sensors may be, for example, at the tip of three fingers used to grasp an object [Salisbury 82].

In addition to the data provided by contact, there is an important additional constraint provided by lack of contact. For example, if the sensors travelled some distance before contact with an object, any valid interpretation of the sensory data must not predict an earlier contact along the path. The principle that a lack of data can provide constraints on interpretation has been exploited in the interpretation of visual data; see [Grimson 81]. We will see later how this constraint can be exploited in the tactile domain.

1.2. Problem Definition

The specific problem we consider in this paper is that of identifying an object from among a set of known objects and of locating it relative to a "hand". We assume that the hand is equipped with three narrow circular fingers[1], equipped with tactile sensors, that can be moved along linear paths. The sensor paths are parallel to, but possibly at different normal distances from, a pre-specified *support plane* (see Figure 2). The hand frame and the positions of the sensors relative to the hand frame are known to high accuracy. Each sensor is processed to obtain (as

[1]The effect of sensor shape can be quite complex, and is outside of the scope of this paper. We have simplified the problem definition by neglecting this effect.

above): (1) one point known to be on the object surface (within some error bound), and (2) a range of feasible surface normals at the point of contact.

The object touched is assumed to be a single polyhedral object that is on the support plane in a stable state. Hence the object has three degrees of positional freedom, x, y, and θ, relative to the frame of the support plane. We call the vector of parameters that uniquely specify the position and orientation of the object its *configuration*. In this case, the vector (x, y, θ) will be the configuration. The different stable states of the object are treated, conceptually, as if they were separate objects. This set of assumptions is similar to those used in many binary vision sytems, e.g., [Gleason and Agin 79].

The key limitation in this problem definition is the one limiting the number of degrees of positional freedom of the object relative to the hand[2]. In bin-picking problems, for example, the objects may have up to six-degrees of positional freedom relative to the hand. Note, however, that if one can locate any planar surface on an object, e.g., by aligning a planar sensor with it or from visual data, then the resulting localization problem is reduced to three degrees of freedom (relative to this surface).

2. Basic Algorithm

In this section we illustrate the basic algorithm for the tactile recognition problem described above. We first illustrate the approach for three sensors moving in a plane, therefore objects can be taken as being polygonal. We will assume that there is no error in determining the position of points on the object's surface. We consider extensions in the next section.

2.1. Interpretation Tree

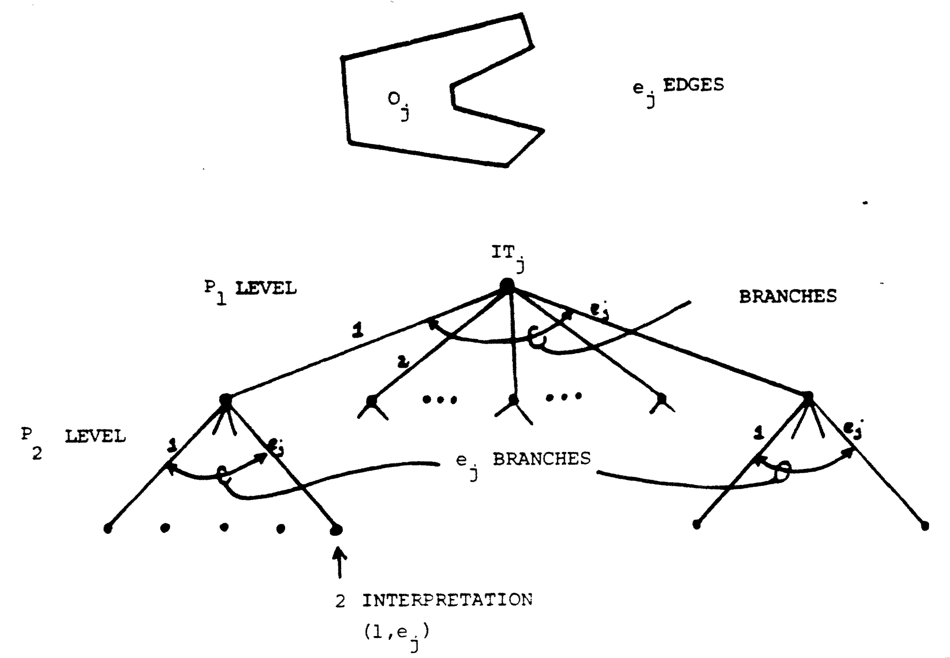
After closing an f-fingered hand on an object, we have the positions of f points, P_i, known to be on the surfaces of one of the n known objects, O_j, having e_j edges. Our first problem is determining on which of the edges of which object each of the P_i is located. From this information, we will be able to compute the location of the object relative to the hand.

The range of possible pairings of contact points and edges for one object can be cast in the form of an *interpretation tree* (IT). The root node of the IT, for object O_j, has e_j descendants, each representing an interpretation in which P_1 is on a different edge of O_j. There are a total of f levels in the tree, level i indicating the possible pairings of P_i with the edges of object O_j (see Figure 3). Note that there may be multiple points on a single edge, so that the number of branches is 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 edges. An f-interpretation is an

[2]The extension of the basic approach described here to the general six freedom case is currently under study [Lozano-Pérez and Grimson 83].

Figure 3. Interpretation Tree



interpretation of length f, i.e., a path from the root of the IT to one of its leaves. Clearly, the IT typically contains a very large number of possible f–interpretations

$$\sum_{j=1}^{n} (e_j)^f$$

In an object with symmetries, of course, the IT is highly redundant. The problem of detecting symmetries is beyond the scope of this paper. The interested reader is referred to [Bolles and Cain 82] for a recent treatment of the topic. Once symmetries are identified, a representative subset of the edges is chosen for the first level of the IT. Once final solutions are found in this IT, the other symmetric solutions can be identified directly. Figure 4 illustrates this.

The n IT's, one for each known object, represent the search space for the tactile recognition problem discussed here. The basic control structure of the algorithm is to generate each level of the IT in a breadth first fashion, pruning interpretations that are inconsistent with input data.

2.2. Pruning

Very few interpretations in an IT are consistent with the input data. In this paper, we exploit the following constraints to prune infeasible interpretations:

1. Distance Constraint — The distances between each pair of P_i must be a possible distance between the edges paired with them in an interpretation.
2. Angle Constraint — The range of possible angles between measured normals at each pair of P_i must include the known angle between surface normals of the edges paired with them in an interpretation.

Figure 4. The effect of object symmetry on the IT

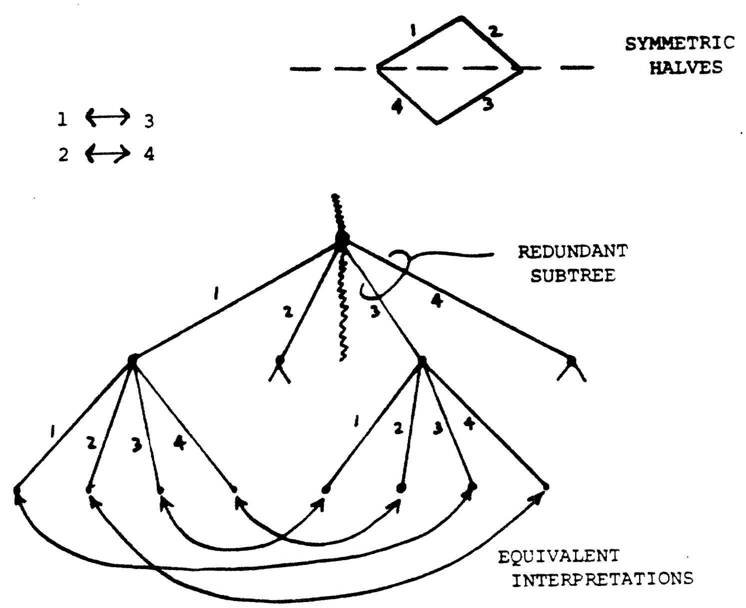
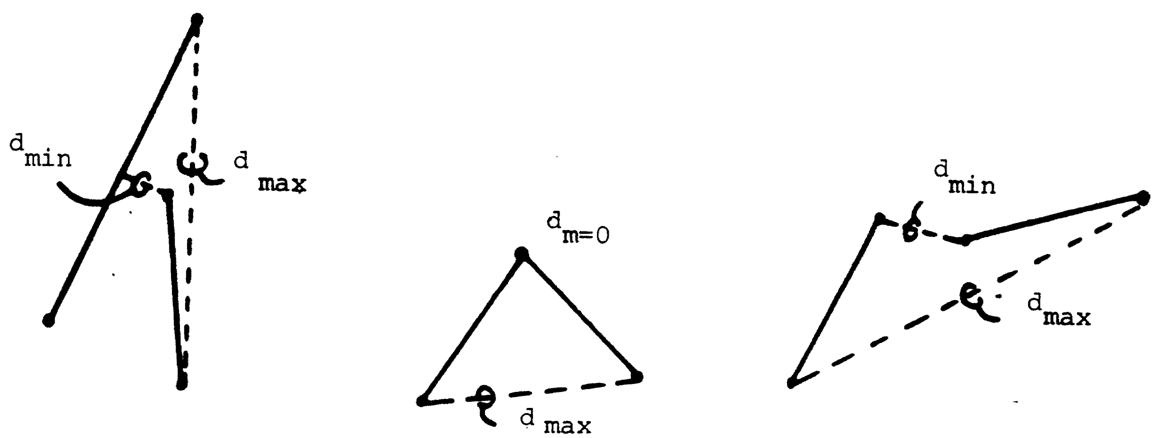


Figure 5. Distance Pruning



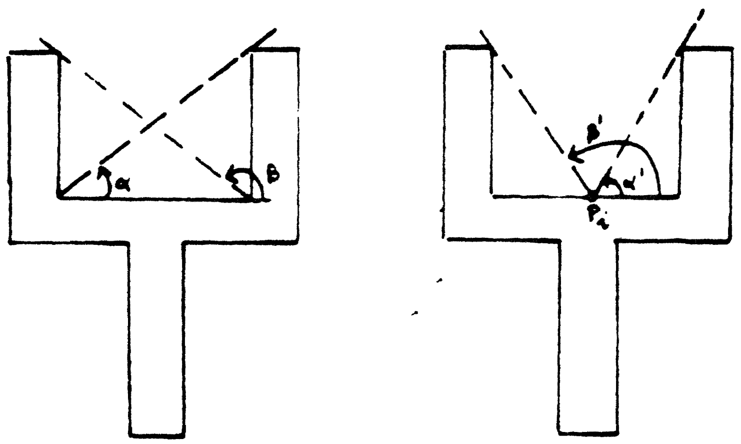
3. Model Constraint — The positions of the P_i must satisfy the equations of the edges paired with them for some position and orientation of the object.

These constraints typically serve to prune away all except a few non-symmetric f–interpretations of the data. Other constraints are possible, e.g., that on the angles in the triangle formed by three contact points.

Note that the distance and angle constraints can be used to prune k–interpretations, for $k > 1$, thereby collapsing the IT. We consider each of the constraints in more detail below.

2.2.1. Distance Pruning

Figure 6. Angle Pockets



CONSERVATIVE ANGLE POCKET (FOR EDGE) $[\alpha, \beta]$

EXACT ANGLE POCKET (FOR POINT) $[\alpha', \beta']$

Given two edges on an object, we can easily compute the range of distances between points on the edges. If the edges touch at a common vertex, the distances will range from zero, at the vertex, to the distance between the other two endpoints of the edges (see Figure 5). Note that we can also compute the range of distances between points on *one* edge (zero to length of the edge).

If an interpretation calls for pairing two of the contact points with two object edges, the distance between the contact points must be within the range of distances between the edges (see also [Bolles and Cain 82]). In fact, the measured distance is subject to measurement error, so the actual constraint is that the range of measured distance plus or minus the estimated error intersects the legal range of distances between the edges. Note that the distances between *all* pairs of contact points must be consistent, i.e., there are three distances between three contact points. Because of this, the distance constraint typically becomes more effective as more contact points are considered.

2.2.2. Angle Pruning

Contact points may be associated with a range of legal surface normals obtained from analyzing the sensory data. Given our restriction on degrees of freedom, the range of normals can be represented as a range of angles relative to the hand frame. The range of normal directions can be directly converted to a range of legal orientations for the touched object. This is not the only source of constraints on the orientation of the object, however.

We also know that if an interpretation associates a contact point with an edge, then the path of the sensor to that contact point must not touch any part of the object before the specified edge. Hence, for each point on an edge, we can identify a range of forbidden approach directions which would violate this constraint[3]. We want to use this constraint to prune impossible interpretations, so we want a conservative estimate of the forbidden directions; hence, we take the intersection of the forbidden ranges for all points on the edge. The complement of this intersection is called the *conservative angle pocket* for the edge. Given an actual or hypothesized contact point on an edge, an *exact angle pocket* can be computed. Angle pockets are represented as ranges of angles relative to a reference frame fixed on the object (see Figure 6).

An additional source of constraint on legal surface normals arises from the static force balance between the sensor and the surface. For the sensor to come to rest on the surface, the force applied by the sensor must point into the surface's friction cone, i.e., the tangential component of the applied force must be less than the maximum frictional force. This constraint can be incorporated into the computation of an edge's angle pocket, although it is fairly weak. It is only useful when no estimate on normal is available from the sensory data.

Given a pairing of a contact point with an object edge we can compute two ranges of orientations of the object's reference frame relative to the hand frame. One range follows from the requirement that the approach direction is within the angle pocket; the other from the requirement that the actual edge normal direction be within the range of measured normal directions. Let ϕ be the orientation of the approach path relative to the hand frame, $[\eta_1, \eta_2]$ be the angle pocket relative to the object's frame, ψ be the orientation of the edge normal relative to the object's frame, and $[\theta_1, \theta_2]$ be the measured range of surface normal angles relative to the hand. The range obtained from the approach direction constraint is $[\phi - \eta_2, \phi - \eta_1]$. The range obtained from the measured normal constraint is $[\theta_1 - \psi, \theta_2 - \psi]$. The intersection of these two ranges represent the range of legal object orientations relative to the hand (see Figure 6).

Given additional pairings of a contact point and an edge, the resulting range of object orientations must be consistent with the intersection of ranges of orientations from previous pairings in the interpretation. A null intersection indicates that the interpretation may be pruned.

2.2.3. Model Pruning

The two pruning methods described above are approximate in that they rule out certain interpretations, but cannot completely determine the configuration of the object. Model pruning proceeds by determining directly what configurations are consistent with the interpretation. If there are none, the branch can be pruned.

From the sensors, we have the position of the P_i relative to the hand's coordinate frame. In our geometric model for the object we have equations for the

[3] Since we are dealing with three-dimensional objects and fingers, this computation must be three-dimensional although the results are two-dimensional. The required computation is to *grow* [Lozano-Pérez and Wesley 79, Lozano-Pérez 81] the object with the finger shape and to take a cross section of the resulting object. The forbidden directions for points approaching edges of this polygon are the ones needed. The details are beyond the scope of this paper.

lines on which edges lie relative to some reference frame fixed on the object. Our goal is to identify the coordinate transformations from the hand frame to the object frame such that each of the P_i falls within the edge specified by the interpretation.

Let the equation for the j^{th} edge line be $F_j(P) = 0$, where $P = (x, y, 1)$ and let $R(x_0, y_0, \theta_0)$ be a homogeneous transformation relating points in the hand frame to those in the object frame. We must solve for the transformation parameters given the equations $F_j(R(x_0, y_0, \theta_0)P_i) = 0$ for each i, j pairing of contact point and edge in the interpretation. For three edges and three points, these equations can be solved analytically; in more complex situations, e.g. curved surfaces, numerical solutions would be required.

In the two-dimensional case with no error, we need three independent equations to locate an object. When multiple contact points are matched to a single edge or parallel edges, only the orientation of the object and not its position may be determinable. If more than three contact points are available, the remaining equations may be used for disambiguation or double-checking, when necessary.

Any legal solutions to the system of equations must satisfy two additional criteria. The first is that the transformed contact points must fall within the finite edge segments of the model. The existence of a solution for the equations guarantees only that the points are on the infinite line containing the edge segment. If the equation system fails to be solvable or if the solution places the points outside the edges, the interpretation can be pruned. Another constraint that must be satisfied is that the approach paths must lie within the exact angle pockets of each point on each edge. Angle pruning, since it does not know the position of the contact point on the edge can only use the conservative angle pockets, which are a weaker constraint.

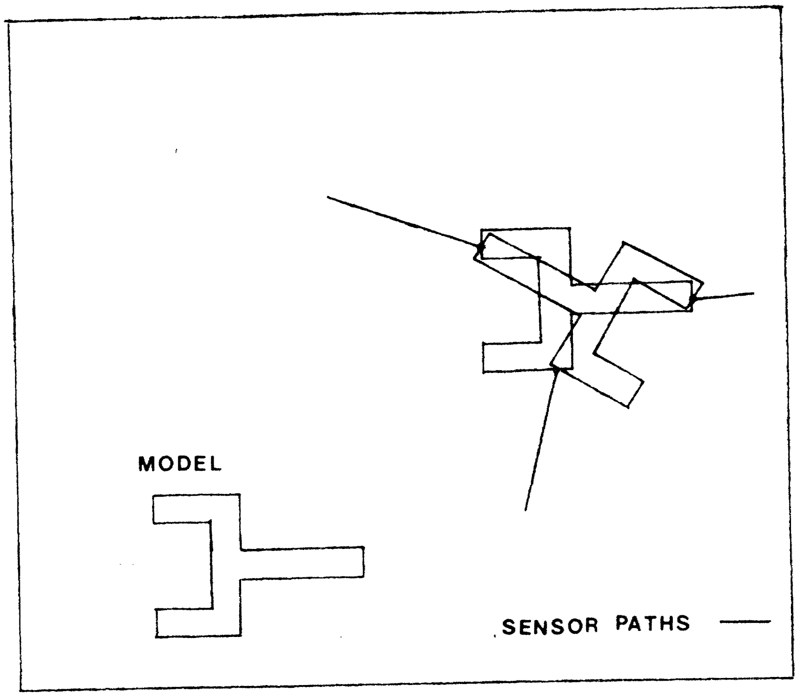
The model pruning test should be a last resort since it requires a 3-interpretation and it is a computationally expensive test. In our implementation, the model test was approximately fifty times slower than the distance or angle test. The principal performance goal of the algorithm is to minimize the number of times that model pruning must be used.

2.3. Examples

Figure 7 shows a model of a twelve-sided polygon, and three approach paths terminating at three contact points on the object. Level 1 of the IT has twelve branches, each representing the possible pairings of P_1 with one of the edges E_j of the object. All 1–interpretations are feasible so the algorithm expands the next level of the tree, which has 144 2-interpretations.

The 2–interpretations are eligible for distance and angle pruning. Only 52 of these interpretations pass the first level of distance pruning and, of these, only 34 survive angle pruning based only on the approach direction (no measured normals are used). At this point, the surviving interpretations can then be expanded in the next level of the tree. Each surviving interpretation has twelve descendants, so a total of 408 interpretations must be considered. Of these, only 23 pass the

Figure 7. Example with twelve-sided polygon



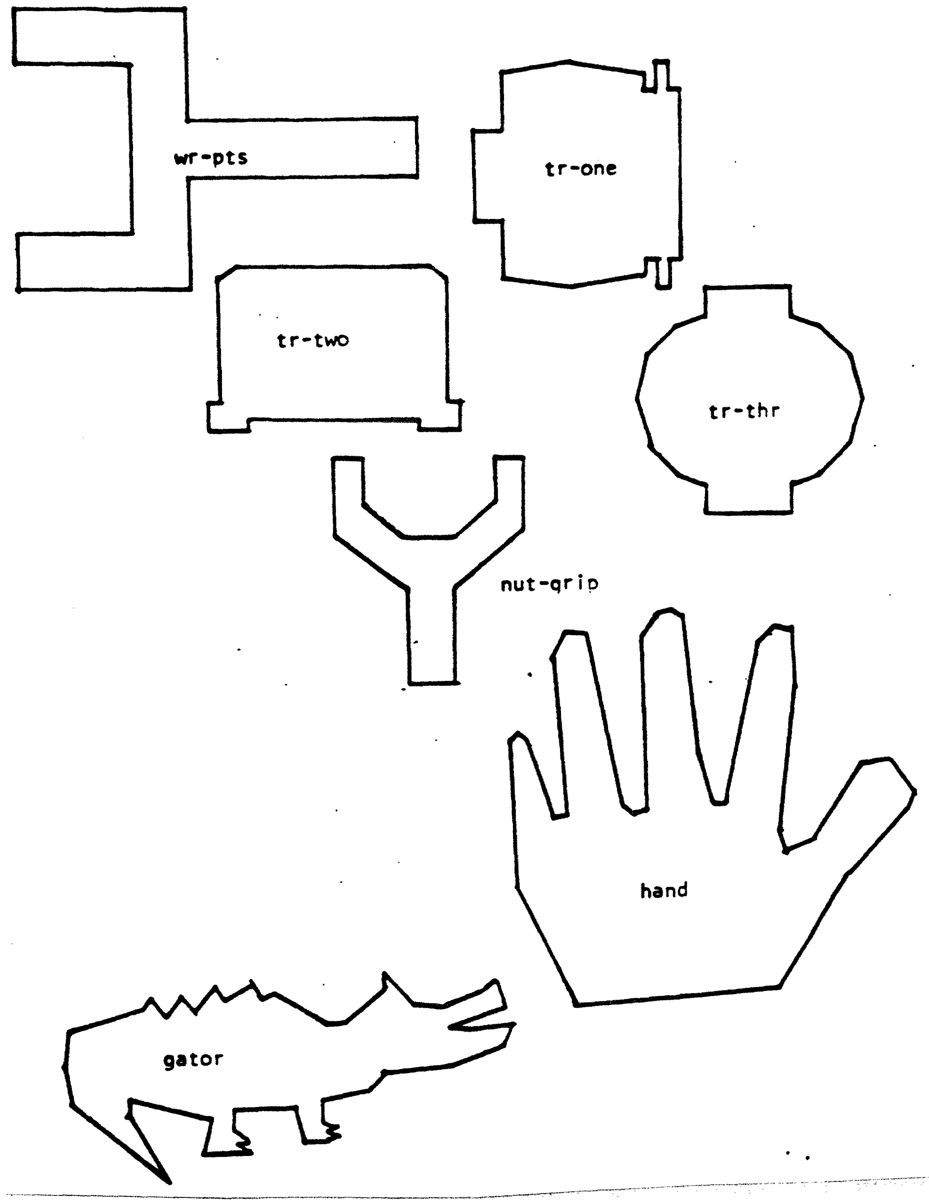
distance test and, of these, only 14 pass the angle test. Of these fourteen remaining interpretations, only two provide solutions for the transformations between hand and object.

To summarize, of the 1728 possible interpretations, only 2 are possible. The distance test was performed on 552 interpretations, the angle test on 65, and the model test only on 14, i.e. less than 1 percent. In fact, had we had tighter angle constraints, fewer total interpretations would have been examined. This example illustrates the surprising effectiveness of the simple pruning mechanisms.

Figure 8 shows several other objects that were handled by an implemented program that embodies the basic algorithm described above. The number of legal configurations depends on symmetries and on the choice of contact points. Table I gives pruning statistics for these objects when distance pruning is used first. Table II gives the statistics when angle pruning is used first. The statistics are given for particular representative choices of approach directions. The results can be better or worse depending on the actual contact points. If the contact points are clustered together, then little pruning can be done. We have found that the best results are obtained when the approach directions are evenly spaced around the object, which is intuitively appealing. Figure 9 shows some results of running the algorithm to differentiate among several objects.

The program used on these examples employed only the constraint imposed by the approach direction, i.e., it does not use measured estimates of the surface normal. For this reason, angle pruning is significantly less effective as a first pruning

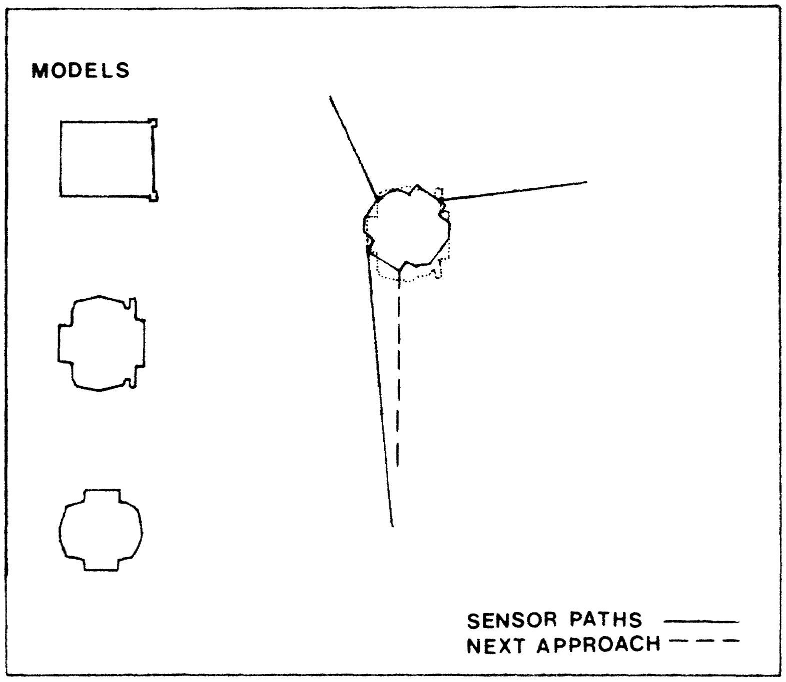
Figure 8. Other objects tested

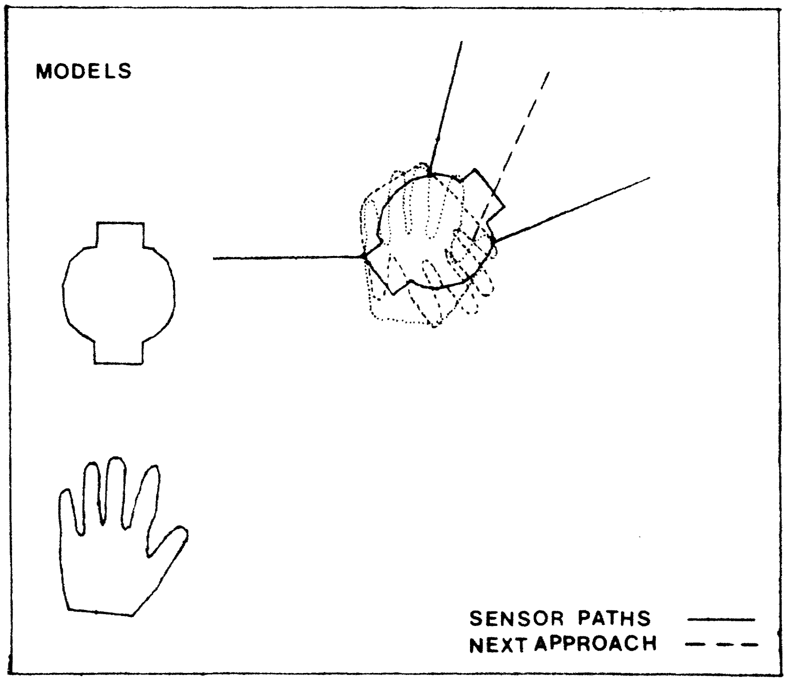


step than distance pruning in these examples. Note that only a small percentage of the interpretations are examined in detail, but that for complex objects the absolute numbers are still large. The use of hierarchic object models as discussed in the next section is intended to address this problem.

In the tables below, the column labels are as follows. Column 1 indicates the number of nodes in the first level of the IT, which is the number of edges in the object (only half the edges of object tr-1 are listed due to symmetry). Column 2 is the number of nodes in the second level of the IT which is equal to column 1 times the number of edges in the object. Column 2D is the number of 2-interpretations surviving distance pruning. Column 2A is the number of such interpretations

Figure 9. Examples showing recognition from among several models





surviving angle pruning. The order of the columns indicates which type of pruning is done first. Column 3 indicates the number of possible 3-interpretations. Columns 3D and 3A indicate the number of 3-interpretations that survive distance and angle pruning respectively. Column M indicates the number of 3-interpretations that pass the model test.

Table I – Pruning Statistics (Distance First)								
Object	1	2	2D	2A	3	3D	3A	M
tr-1	11	242	71	3	66	11	4	2
tr-2	26	676	190	125	3,250	178	58	2
tr-3	14	196	54	36	504	60	11	2
grip	14	196	42	20	280	80	39	4
gator	49	2,401	363	215	10,535	614	278	1
hand	66	4,356	516	243	16,038	171	118	2

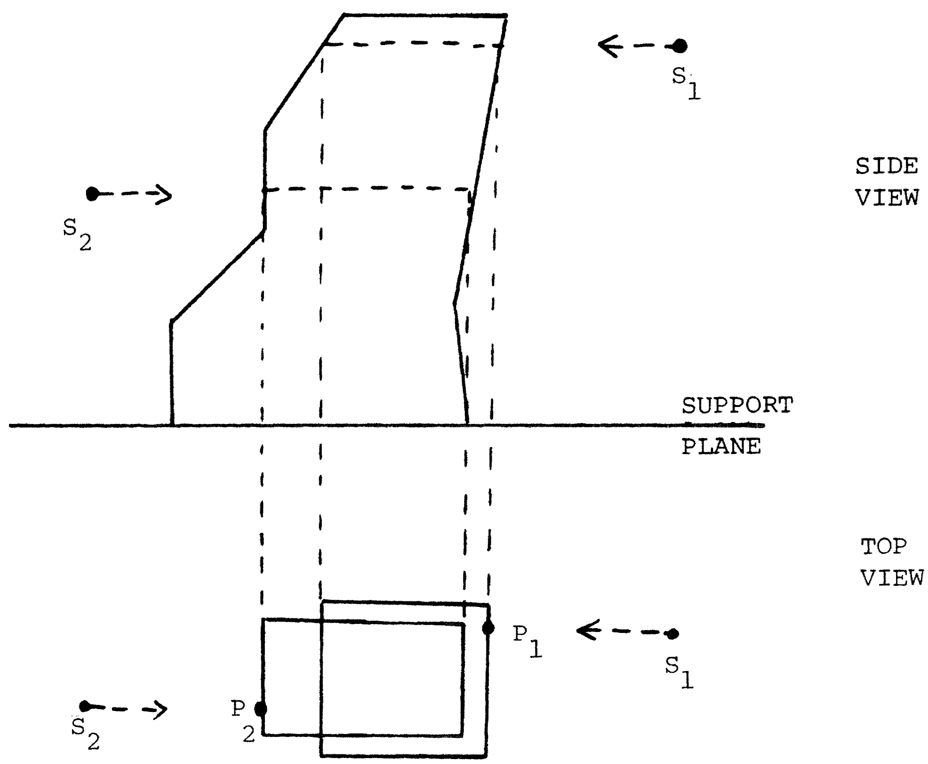
Table II – Pruning Statistics (Angle First)								
Object	1	2	2A	2D	3	3A	3D	M
tr-1	11	242	147	3	66	31	4	2
tr-2	26	676	375	125	3,250	1,317	58	2
tr-3	14	196	133	36	504	247	11	2
grip	14	196	84	20	280	120	39	4
gator	49	2,401	1,481	215	10,535	4,711	278	1
hand	66	4,356	1,994	243	16,038	6,270	118	2

In Table III below, we recast the statistics above into pruning efficiencies, i.e., the ratio of the number of interpretations that are eliminated by one or more pruning tests to the number of initial candidate interpretations. We refer to the columns in Tables I and II by prefixing the table number to the column name, e.g., the fourth column of Table I will be denoted I2D. The columns in Table III are computed as follows. Column D2 is $\frac{I2-I2D}{I2}$. Column A2 is $\frac{II2-II2A}{II2}$. Column DA2 is $\frac{I2-I2A}{I2}$. Column D3 is $\frac{I3-I3D}{I3}$. Column A3 is $\frac{II3-II3A}{II3}$. Column DA3 is $\frac{I3-I3A}{I3}$.

Table III – Pruning Statistics (Efficiencies)						
Object	D2	A2	DA2	D3	A3	DA3
tr-1	.707	.392	.988	.833	.530	.939
tr-2	.719	.445	.815	.945	.595	.982
tr-3	.724	.321	.816	.881	.501	.978
grip	.786	.571	.898	.714	.571	.861
gator	.849	.383	.910	.941	.553	.974
hand	.882	.542	.944	.989	.609	.993

Note the surprisingly high efficiency of the distance test.

Figure 10. Sensors at different heights generate multiple cross sections



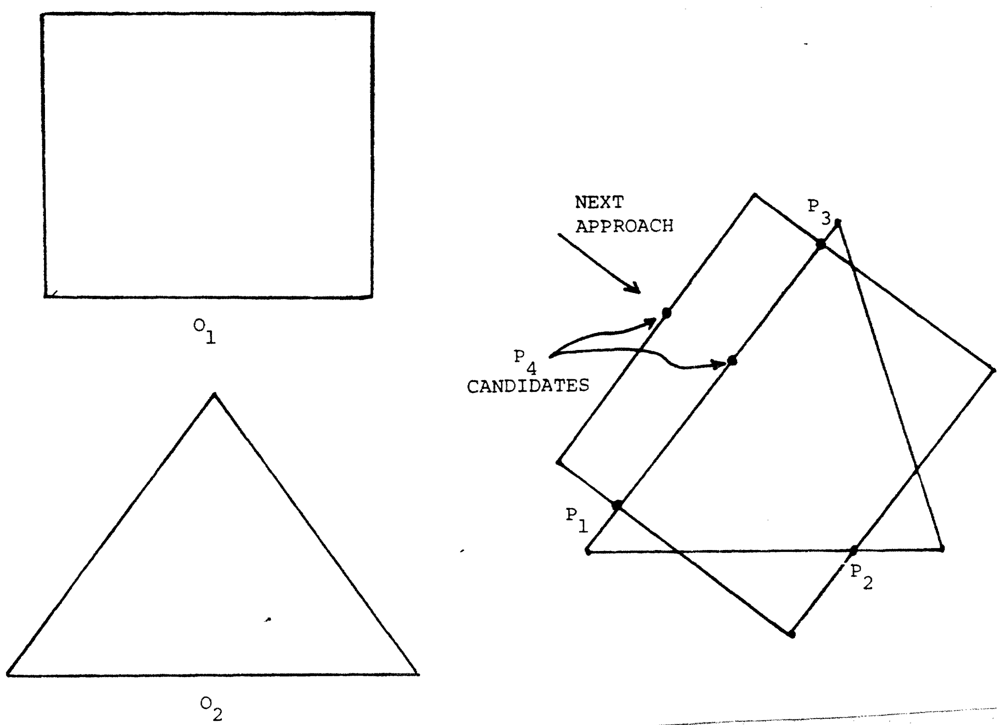
3. Suggestions for Enhancements to the Basic Algorithm

In this section, we consider extensions to the basic algorithm that may improve its performance as well as extend its range of applicability. The ideas discussed here are the subject of ongoing research [Gaston 83, Lozano-Pérez and Grimson 83].

3.1. Sensors at Different Heights from the Support Plane

The problem statement in section 2 requires that the sensors be at same height above the support plane, effectively reducing the recognition and localization problem to two dimensions. The generalization to sensors moving at different heights above the support plane is straightforward. Each P_i is constrained to be on a different cross section of the object parallel to the support plane. These cross sections are fixed rigidly relative to each other (see Figure 10). Hence, on each level of the IT the set of edge candidates for pairing with a contact point is drawn from a different cross section (see Figure 10). Distance pruning is unchanged under these circumstances, except that only distance along the support plane is considered. Angle pruning and model pruning are unchanged.

Figure 11. Next approach disambiguates among legal configurations



3.2. Disambiguation

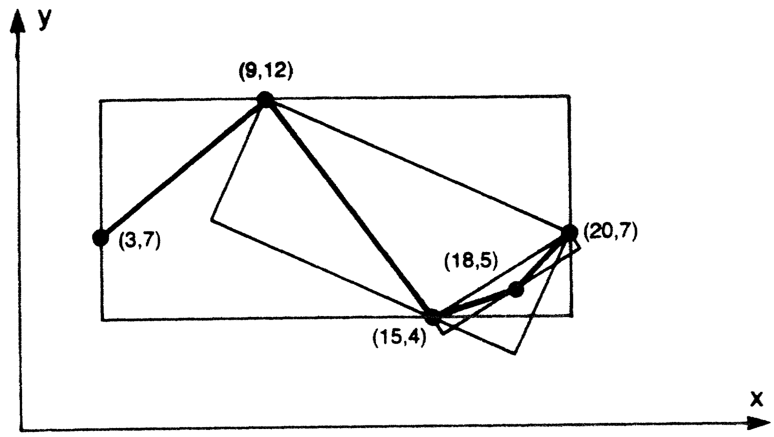
In general, multiple interpretations (several objects and several configurations of those objects) will be consistent with the distance, angle, and model constraints; we saw this in the examples in Section 2.3. There are two main sources of ambiguities: uncertainties in measuring the surface normals and symmetries.

Disambiguating between legal interpretations requires additional data, which may be obtained by moving the sensors on the object. An alternative to moving the sensor is the use of four or more sensors, instead of the minimum of three, so as to reduce the number of ambiguous interpretations. With redundant sensors, the number of interpretations that will require the model test should also be significantly fewer.

One possible strategy for obtaining the additional constraints required for disambiguation is simply to pick a new grip at random and apply the algorithm again. Only the interpretations compatible with the first grip need be examined; a new grip is no different from having double the number of sensors to begin with. This process is repeated until a single configuration of one object is consistent with the data from all grips.

A second strategy is to rotate the hand slightly while maintaining surface contact, thereby obtaining position information from nearby points. This method is most useful when the ambiguity is due to paucity of surface normal information. It is less likely to be useful in the presence of symmetry.

Figure 12. Strip Trees [Ballard 81]



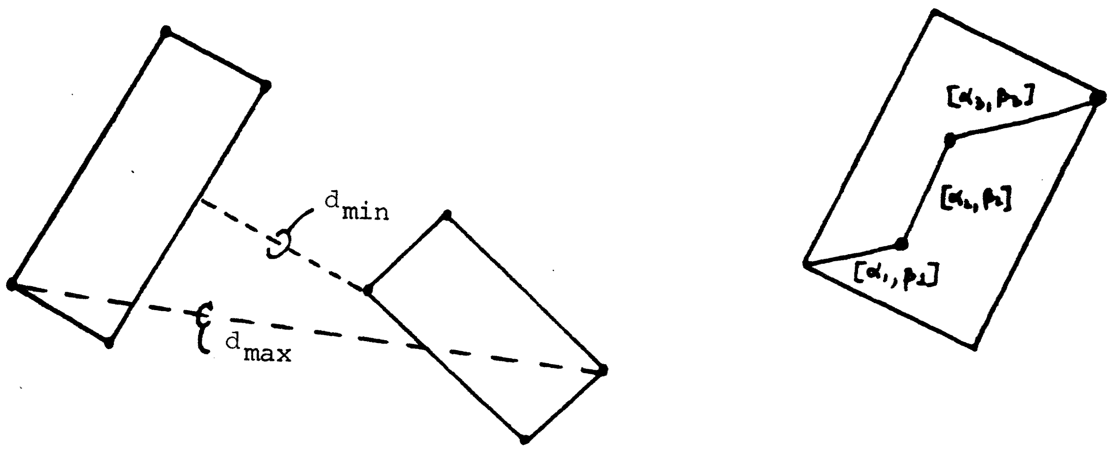
A third strategy is to choose a new grip such that the approach directions of the fingers are guaranteed to disambiguate among the possible objects and configurations (or provide the maximal information). This can be done by choosing approach directions for the fingers such that, between them, the fingers cross one edge for each object or configuration, and furthermore, that the possible crossing points along each approach path be separated from each other by a perceptible amount (see Figure 11). Each of the crossing points of the approach directions and an edge represents the position of the contact point to be expected if that interpretation holds.

Note that the chosen next approach direction must be guaranteed to reach the edge, so the direction should be within the intersection of the exact angle pockets for all the points on all the edges. Because the candidate interpretations are known, these angle pockets are available as angles relative to the hand frame. One possible next approach direction found by an implementation of a simple form of this algorithm is shown in some of the examples in Section 2.3 and labeled "next approach".

3.3. Using Hierarchical Object Models

For objects with large numbers of edges, n, it may be too expensive to even consider the n^2 2-interpretations in the IT for pruning. The "hand" object in Section 2.3, for example, had 66^2 nodes at level 2. In these circumstances, we can use a hierarchical representation of the object's boundary to limit the combinatorial explosion. A good choice of representations for the object boundary is the *strip tree* representation suggested by [Ballard 81] (see Figure 12). So as to accomodate angle pruning, each strip must represent a list of the edge normals within the strip,

Figure 13. Distance and angle pruning generalized to strips



and the angle pocket for the strip, which is the union of the angle pockets for the edges in the strip.

We can now apply the basic algorithm of Section 2 to any level of the strip tree representation of an object's boundary. In particular, distance and angle pruning can be simply generalized to strips. Distance pruning is based on the ranges of distances between strips instead of those between edges. Angle pruning must deal with unions of angle ranges arising from the individual angles in each strip. These generalizations are illustrated in Figure 13. Model pruning is postponed until the most detailed level of the strip tree, corresponding to the original edge list.

Each remaining legal interpretation from one level of the strip tree defines a limited object model to which the basic algorithm can be applied. In the next iteration of the algorithm, a P_i is limited to pairing with the sub–strips of the strip paired with that contact point at the current level of the strip tree (see Figure 14).

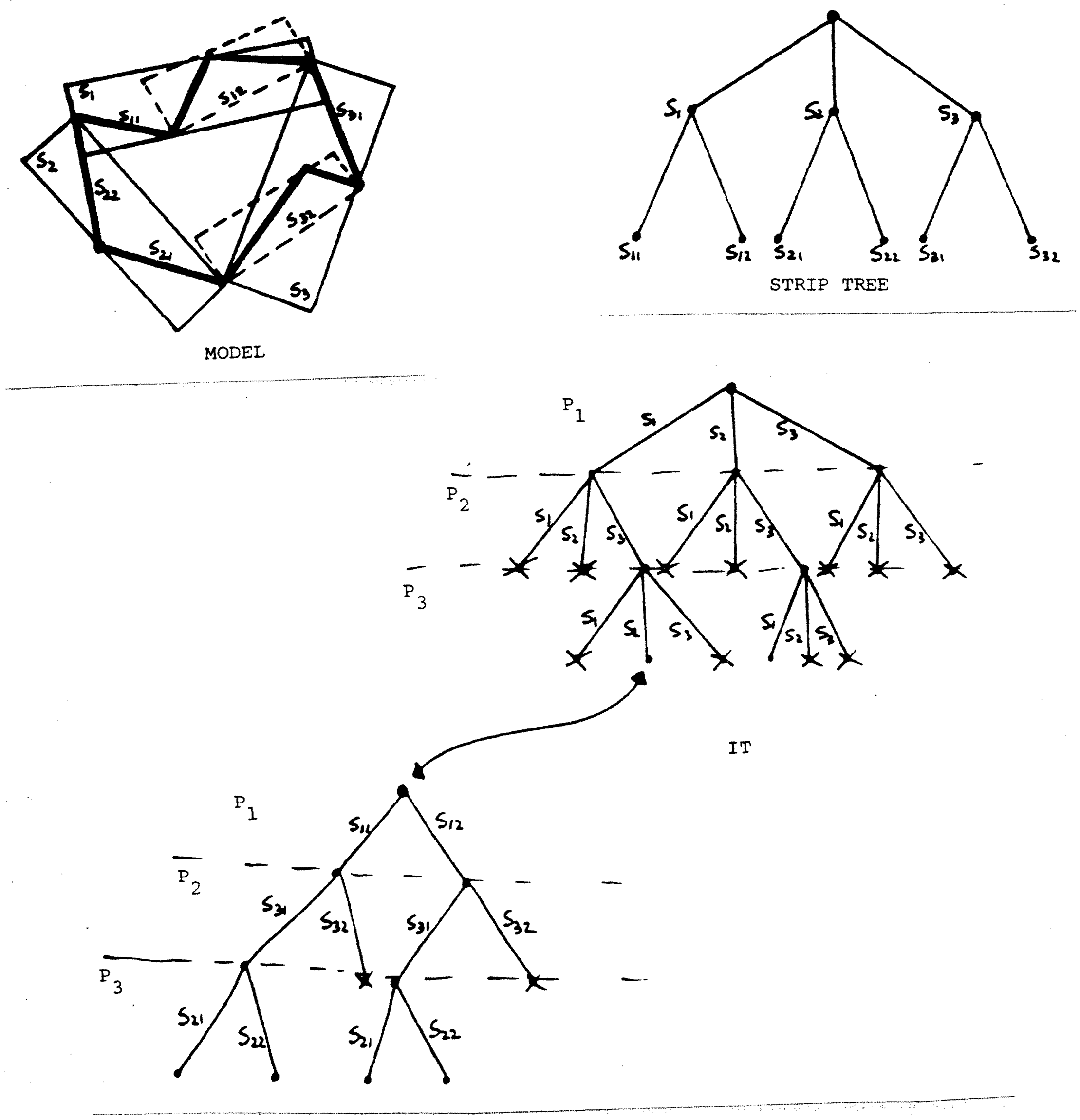
In the worst case, e.g., when all the interpretations are legal, the strip tree approach leads to additional work with no savings. We expect that on average it will produce substantial savings for very large object models.

3.4. Measurement Error

We have assumed, thus far, that the position of the contact points are known exactly. In practice, the measured position is subject to error from a variety of sources, including sensor deflection, the sensor's limited spatial resolution, and errors in the hand's position sensors. The object model also is limited in accuracy.

Distance pruning can be readily extended to deal with errors by using the technique discussed for strip trees. Each edge can be enclosed in a strip that encloses all possible measured positions of a contact point that could be on the edge. When an interpretation involving two such strips is pruned, it means that the

Figure 14. Recursive expansion of the IT with strip trees



interpretation is impossible even taking error into account. One can expect that the efficiency of distance pruning will deteriorate as the expected error increases.

Model pruning, as described earlier, is impossible in the presence of error. In general, the edge equations will be inconsistent with the measured data. The approach we are pursuing is to solve numerically for the object's configuration that minimizes the distances of the contact points from the edges paired with them in the interpretation. If any of the minimal distances exceeds a maximum error bound, the interpretation is invalid. The key problem in implementing this method is choosing initial values for the configuration parameters of the object given a pairing of edges and contact points. Further work is underway in this area.

4. Summary

This paper has introduced a simple and efficient approach to the recognition and localization of objects using object models and very local tactile information: positions of surface points and constraints on surface normals. Using simple pruning mechanisms, we were able to achieve drastic reductions of the combinatorics in the recognition process.

The method described here is limited to polyhedral objects having three degrees of positional freedom relative to the hand. The generalization of the method to objects with curved surfaces and six degrees of positional freedom is the subject of ongoing research; the techniques described in this paper appear to generalize fairly directly.

Acknowledgements

We would like to thank Eric Grimson for his enthusiasm, technical suggestions, and for his general help in the preparation of this report. We would also like to thank Mike Brady, Rod Brooks, John Hollerbach, John Schneiter, and Ken Salisbury for reading an earlier draft of this paper and contributing many suggestions on content and presentation.

5. References

Ballard, D. H. "Strip trees: A hierarchical representation for curves," *Communications of the ACM* 24, 5 (May 1981), 310–321.

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.

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.

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.

Gaston, P. C. Robotic Tactile Recognition, S. M. Thesis, Department of Elec. Engr. and Comp. Sci., Massachusetts Institute of Technology, January 1983.

Gleason, G. and G. J. Agin "A modular vision system for sensor-controlled manipulation and inspection," *Ninth ISIR* , Washington D. C., March 1979, 57–70.

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

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

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

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

Kinoshita, G., S. Aida, and M. Mori "A pattern classification by dynamic tactile sense information processing," *Pattern Recognition* **7** (1975), 243.

Lozano-Pérez, T. "Automatic Planning of Manipulator Transfer Movements," *IEEE Transactions on Systems, Man, and Cybernetics* **SMC-11**, 10 (October 1981), 681–698.

Lozano-Pérez, T. and W. E. L. Grimson "Local Constraints in Tactile Recognition," MIT Artificial Intelligence Laboratory, forthcoming.

Lozano-Pérez, T., and Wesley, M. A. "An algorithm for planning collision-free paths among polyhedral obstacles," *Communications of the ACM* **22**, 10 (October 1979), 560-570.

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

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

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.

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

Salisbury, J. K. Kinematic and Force Analysis of Articulated Hands, Ph. D. Thesis, Department of Mechanical Engineering, Stanford University, 1982.

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.

Stojilkovic, Z. and Clot, J. "Integrated Behavior of Artificial Skin," *IEEE Transactions on Biomedical Engineering* **BME-24**, 4 (July 1977), 396–399.

Stojilkovic, Z. and D. Saletic "Learning to recognize patterns by Belgrade hand prosthesis," *Fifth ISIR* , IIT Research Institute, Chicago, 1975, 407–413.

Takeda, S. "Study of artificial tactile sensors for shape recognition: algorithm for tactile data input," *Fourth ISIR* , 1974, 199–208.