Applying Uncertainty Reasoning to Planning Sensing Strategies in a Robot Work Cell with Multi-Sensor Capabilities*

S. A. Hutchinson and A. C. Kak

Robot Vision Laboratory
School of Electrical Engineering
Purdue University
West Lafayette, IN 47907

ABSTRACT

In this paper, we describe an approach to planning sensing strategies dynamically, based on the system's current best information about the world. Our approach is for the system to automatically propose a sensing operation, and then to determine the maximum ambiguity which might remain in the world description if that sensing operation were applied. When this maximum ambiguity is sufficiently small, the corresponding sensing operation is applied. To do this, the system formulates object hypotheses and assesses its relative belief in those hypotheses to predict what features might be observed by a proposed sensing operation. Furthermore, since the number of sensing operations available to the system can be arbitrarily large, we group together equivalent sensing operations using a data structure that is based on the aspect graph. Finally, in order to measure the ambiguity in a set of hypotheses, we apply the concept of entropy from information theory. This allows us to determine the ambiguity in a hypothesis set in terms of the number of hypotheses and the system's distribution of belief amongst those hypotheses.

1. INTRODUCTION

With current techniques in geometric modeling, it is possible to generate object models with a large number of features and relationships between those features. Likewise, given the current state of computer vision (both 2-D and 3-D) and tactile sensing, it is possible to derive large feature sets from sensory data. Unfortunately, large feature sets can also require exponential computational resources unless one takes advantage of the fact that most objects can be recognized by a few landmarks. The problem then becomes one of developing computer procedures capable of analyzing geometric models to yield the most discriminating feature sets. In solving this problem, one has to bear in mind that in the robotic cells of today we have available to us a variety of sensors, each capable of measuring a different attribute of the object.

For it to be useful to robotic assembly, we need to add another dimension to the problem as stated above. Say, we have a robot trying to determine the identities of the objects in its work area. The robot should only invoke those sensory operations that are most relevant to the disambiguation of whatever hypotheses the robot might entertain about the identities of those objects. Therefore, the most discriminating features invoked by the robot must be determined at run time and, of course, must make maximum advantage of all the sensors that are available.

Previous work on sensor planning has been divided into two distinct areas. One of these areas is concerned with sensor placement. This problem is to place the sensor so that it can best observe some feature (which is predetermined) or region of 3-space. The other problem is to choose a sensing operation which will prove the most useful in object identification and localization.

In the area of sensor placement, Connolly [3] has implemented a system chooses sensor locations to minimize the number of view required to build up a complete octree model of a scene. Kim, et. al. [14] have developed a system which determines successive camera viewpoints so that the most distinguishing features of the object can be observed. Cowan and Kovesi [4] automatically select sensing strategies based on object and camera models such that a number of constraints are simultaneously satisfied (e.g. that the spatial resolution be better than some minimum value, that the surfaces to be viewed lie within the camera's field of view).

Work on automatically determining optimal sensing strategies has been done by Ikeuchi [13], Hanson and Henderson [8], and Hager and Mintz [7]. In the first two of these, sensing strategies are precompiled into search trees. The run time selection of sensing strategies amounts to choosing a branch in the precompiled search tree based on the information which has already been obtained by the sensing system. In the third of these, decision theoretic techniques are applied to the problem of selecting optimum sensing strategies. This is accomplished by treating sensors as noisy information sources, associating a risk function with each sensing operation, and selecting the sensing operation which minimizes the risk function.

The work that we present in this paper extends the work cited above in a number of directions. First, we give the system the ability to choose sensing strategies based on current hypotheses about the identity and pose of an object which is being examined. It is possible that each such hypothesis will correspond to a different object. Furthermore, the choices of sensing strategies are not limited by the use of a single type of sensor. The sensory types currently incorporated in the system include a 3-D range scanner, 2-D overhead cameras, a manipulator held 2-D camera, a Force/Torque wrist-mounted sensor, and also the manipulator fingers for estimating the grasp width. The vision sensors can be used to examine objects from arbitrary viewpoints, while the manipulator and F/T sensor can be used to measure other features such as weight, depth of occluded holes in the object, etc.

It is important to realize that with these additional sensory inputs, we can discriminate between object identities, aspects and poses that would otherwise appear indistinguishable to just a fixed viewpoint vision-based system. Our system is capable of dynamic viewpoint selection if that's what is needed for optimum disambiguation between the currently held hypotheses.

We attack the problem of viewpoint and sensor-type selection as follows. Once the system has a working set of hypotheses (which is initially developed after application of an arbitrary sensing operation, say the 3-D range scanner), candidate sensing operations are automatically proposed and evaluated with regard to their potential effectiveness, given the current hypothesis set. This evaluation is performed as follows. For each hypothesis in the current hypothesis set, the system determines the set of features that would be observed by the candidate sensing operation if that hypothesis were correct. Using these predicted features, the system determines the hypothesis set that would be formed if these features were actually found by some sensing operation. The ambiguity of this predicted hypothesis set is calculated and noted. This is repeated for each hypothesis in the hypothesis set, and the maximum value of the ambiguities is associated with the proposed sensing operation. When a proposed sensing operation's maximum ambiguity is sufficiently low, that sensing operation is selected for application.

In the remainder of the paper, we will describe each of the above steps in some detail. In Section 2, we will introduce our object representation. This representation is used both to quantize the space of sensing operations and to predict the features which would be observed by a candidate sensing operation. Section 3 describes how our system generates and refines hypothesis sets, as well as how uncertain reasoning is implemented in the system. In Section 4, we define the measure of ambiguity which our system uses. The measure that we describe is based on entropy from information theory. In Section 5, we describe the types of sensors our system uses, and the types of features that they can be used to detect. Section 6 brings together the results of Sections 2 through 5 and presents the formal algorithm for selecting the next best sensing operation. Finally, in Section 7, we describe some of our preliminary experimental results.

2. OBJECT REPRESENTATION

The object representation used in our system plays two key roles. First, it allows us to quantize the space of sensing operations. This is a result of the fact that the representation groups together sets of object features which can be viewed from a single viewpoint (such a set of features is referred to as an *aspect*). This allows us to group together all viewpoints which can observe the same aspect. Second, the representation allows us to easily determine the features of an object which will be observed by a particular sensor from a particular viewpoint relative to the object. This is done by determining which aspect of the object will be observed from the particular viewpoint, and then looking up the object features which are associated with that aspect. In the remainder of this section we will describe aspect graphs and how they are derived by our system.

The aspect graph was originally developed by Koenderink and van Doorn [15] (who referred to it as the *visual potential*) to characterize the visual stimulus produced by an object when viewed from different relative positions. They developed a function for the "sensory inflow" produced by an object, in terms of the invariant properties of the object and the relative positions of the viewer and the object. The local behavior of this function is defined in terms of the deformation of the retinal images through changing perspective (and is of no particular relevance to our work). The global behavior of the function is defined in terms of its singularities. Two types of singularities have been considered: point singularities, which determine a system of protrusions facing the observer, and line singularities, which correspond to the curve on an object that divides the it's surface into visible and nonvisible regions. An aspect is characterized by the structure of these singularities for a single view. From most all vantage points, an observer may execute small movements without affecting the aspect. However, when an observer's movement does cause the structure of the singularities to be changed, an *event* is said to have occurred, and a new aspect is brought into view. An *aspect graph* is created by mapping aspects to nodes and mapping the events that take the viewer from one aspect to another to arcs between the corresponding nodes.

In our work, we are not so much interested in retinal images as we are in features which can be observed by the various sensors. Thus, we characterize

*This work was supported by the Engineering Research Center for Intelligent Manufacturing at Purdue University, and the industry supported Purdue CIDMAC program.

aspects, not in terms of the singularities in the function which defines the visual inflow, but in terms of observable features. In particular, we define an aspect to be a set of features which can be observed simultaneously from a particular viewpoint. When a change of viewpoint causes a previously visible feature to no longer be visible, or a new feature to come into view, an event occurs. We use the aspect graph to group viewpoints that see the same aspect into equivalence classes. Associated with a node in the aspect graph is the set of viewpoints from which that aspect can be observed. Arcs in the graph connect nodes with adjacent viewpoints. Also, with each node in the aspect graph, we associate a principal viewpoint, which is essentially a "representative" viewpoint for the aspect.

Aspect graphs for objects can be generated analytically or by an exhaustive examination of the object. Analytic techniques have been reported by Castore and Crawford [1], Stewman and Bowyer [20], and Gigus and Malik [6].

Our system generates aspect graphs exhaustively. This is done by creating a CAD model of the object, centered within a tessellated viewing sphere (we currently use 60 tessellations, which are derived as in [22]). The geometric modeler is then used to view the object from the center point of each tessellation, and the set of visible features is recorded. Using this information, it is a simple matter to generate the aspect graph. Tessels that see the same feature set are grouped together into nodes. The arcs between nodes are generated using tessel adjacency. Finally, each aspect is assigned a principal viewpoint, which is defined as the average location of the centers of the viewing tessels associated with the aspect, with the constraint that it lies within a tessel that observes the aspect.

We quantize the space of sensing operations by only considering viewpoints which correspond to the principal viewpoint of some aspect of a hypothesized object. Furthermore, once a sensing operation is proposed, the set of features which we expect that operation to find is simply the set of features in the aspect which that sensing operation will observe.

3. GENERATING HYPOTHESIS SETS

Generating, and subsequently refining, hypothesis sets begins by matching sensed features to model features, and then assessing the quality of those matches. In our system, a sensed feature can be matched to any of the model features which have attributes that are similar to those of the sensed feature. The degree of similarity will determine the quality of the match. In order to reason about the hypotheses derived from these matches, the system must be able to represent its relative belief in the various feature matches. Furthermore, since an object hypothesis will correspond to a number of feature matches, the system must be able to combine the beliefs in the individual feature matches to assess its belief in an object hypothesis.

When evaluating belief in an object hypothesis, feature matches are not the only source of information. We also determine the relationships between sensed features and compare these to the relationships between the corresponding model features. As their similarity increases, so does the confidence in the corresponding hypothesis. This allows us to accumulate evidence which supports a hypothesis based on its relational consistency. It also allows us to discount hypotheses in which the relationships between sensed features are inconsistent with the corresponding model relationships, thus pruning the number of hypotheses which the system must maintain.

The final source of evidence we consider evaluates the difference between the expected and actual sensed data. Once an object hypothesis has been established, we can derive a pose transformation that expresses the position/orientation of the object if that hypothesis is correct. By using the pose transformation in conjunction with information about the sensing operation that was performed, we can determine what "should have been observed" by the sensor if that hypothesis were correct. Of course we cannot expect that sensing will always find every feature which might be present, so we assign values to each object feature which reflect the prominence of that feature. We then evaluate the quality of the object hypothesis by noting the prominence of the expected features which were (and were not) matched.

In our system, we use the Dempster-Shafer (DS) theory of evidence to implement our reasoning system. For the sake of those not well acquainted with the DS theory, we will now digress to give a very brief introduction. Those familiar with bpa's, belief functions, Dempster's rule, and coarsening and refining might want to skip this section. Those completely unfamiliar with these ideas might want to investigate [17].

3.1. The Dempster-Shafer Theory

In the DS theory, all possible propositions are grouped together in the set Θ, which is referred to as the frame of discernment. When a proposition corresponds to some subset of Θ, it is said that Θ discerns that proposition.

Associated with each subset of Θ is a *basic probability assignment (bpa)* which is the measure of belief in exactly the proposition represented by that subset. A bpa is a function m: $2^{\Theta} \rightarrow [0,1]$, such that:

$$m(\varnothing)=0$$

$$\sum_{A \subset \Theta} m(A) = 1$$

In order to find the total belief in a certain proposition, we must examine the belief in that proposition as well as every proposition that implies it. This is expressed by the function Bel: $2^{\Theta} \rightarrow [0,1]$. The total belief in a proposition, A, is:

$$Bel(A) = \sum_{B \subset A} m(B)$$

If we have two belief functions with bpa's $m_1(\cdot)$ and $m_2(\cdot)$, then we can combine these using Dempster's rule of combination:

$$m(\theta) = \frac{\sum_{A \cap B = \theta} m_1(A)\, m_2(B)}{1-K}$$

where

$$K = \sum_{A \cap B = \varnothing} m_1(A)\, m_2(B)$$

Note that when K = 1, the two belief functions flatly contradict one another, and thus their combination does not exist.*

The sets of propositions involved in different sensor measurements -- i.e. propositions would be elements of the respective frames of discernment -- will in most cases not be identical. When combining evidences from different measurements, we must therefore first establish a common frame of discernment in which the disparate sets of propositions corresponding to different sensory measurements would all become discernible. An important point to note is that the elements of a frame of discernment do not have to represent all the possible outcomes in an experiment at their finest level of detail, but only the propositions relevant to a particular measurement.

The process of refining frames of discernment, Θ_1, Θ_2, ..., to a common frame Ω is accomplished by specifying the mapping functions

$$\omega_i: 2^{\Theta_i} \rightarrow 2^{\Omega}$$

which must obey the following properties

$$\omega_i(\{\theta\}) \neq \varnothing, \text{ for all } \theta \in \Theta_i$$

$$\omega_i(\{\theta\}) \cap \omega_i(\{\theta'\}) = \varnothing \text{ for } \theta \neq \theta'$$

$$\bigcup_{\theta \in \Theta_i} \omega_i(\{\theta\}) = \Omega$$

The first property says that any proposition that is discerned in, say, Θ_i must also be discernible in Ω. The second property requires that the mapped propositions in Ω for different propositions in Θ_i be disjoint. Finally, the third property specifies that if Ω is a refinement of Θ_i, then no proposition in Ω be outside the range of mappings corresponding to the different propositions in Θ_i.

To illustrate these notions, consider the two cubes shown in Fig. 1a; the cube on the left has two round holes of unequal diameters, and the cube on the right has one face with a round hole and two faces with rectangular holes of different sizes. For a sensory measurement from the direction shown in Fig. 1b, the frame of discernment might be

$$\Theta_1 = \{a, c, f\}$$

since the sensed face resembles, to some extent, the three faces a,c, and f. Now consider the measurement from the direction shown in Fig. 1c. The frame of discernment for this measurement might be

$$\Theta_2 = \{d, e\}$$

In order to combine the evidences generated by these two measurements, we first construct a refinement of the two frames. The following is a valid refinement which obeys the above three properties:

$$\Omega = \{\{a,d\},\{a,e\},\{c,d\},\{c,e\},\{f,d\},\{f,e\}\}$$

3.2. Generating and Refining Hypothesis Sets

In our previous work [12], hypothesis generation and refinement was a fairly simple process. Each sensed feature was matched to all feasible model features (where feasibility was determined by the similarities of the attributes of the sensed and model features). These matches were then pruned by enforcing relational constraints and aspect consistency. Relational consistency was determined by examining the similarity of the relationships between the sensed features and the corresponding relationships between the matched model features. If the similarity was below some quantitative threshold, the hypothesis was discarded. Aspect consistency (which will be discussed in more detail later) ensured that prominent object features were matched if they could be observed by the performed sensing operation. If they were not matched, the corresponding hypothesis was discarded.

Our current system retains feature matches, relational consistency and aspect consistency as the three measures of a hypothesis' credibility, but, thresholding has been replaced by reasoning with partial evidence. Now, a hypothesis

* In a recent paper, Hummel and Landy [11] have demonstrated that this normalization is not strictly necessary. This was shown by constructing a homomorphic mapping from the space of unnormalized belief states to the space of standard DS belief states.

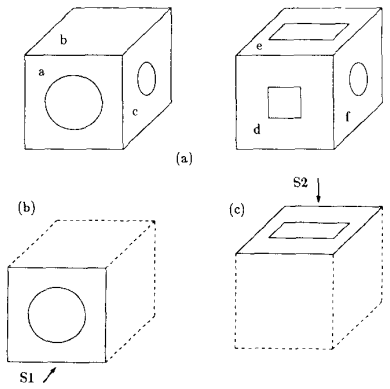

Fig. 1: (a) shows two simple objects. (b) shows the sensed data as seen from S1 (c) shows the sensed data as seen from S2

is given credibility which reflects how well the three criteria above are satisfied. We use three bpa's: $m_f(\cdot)$, $m_r(\cdot)$ and $m_a(\cdot)$ to assign belief to object hypotheses based on the quality of the feature matches, relational consistency and aspect consistency. We combine these using Dempster's rule of combination to determine the total belief in a hypothesis. If a current hypothesis set already exists, the new belief function must be combined with the belief function for the existing hypothesis set to produce the revised hypothesis set and the associated beliefs. We will now describe how the bpa's $m_f(\cdot)$, $m_r(\cdot)$ and $m_a(\cdot)$ are generated, and how they are combined to define belief in sets of object hypotheses.

3.2.1. Consistency of Feature Matches

When a sensing operation finds a set of sensed features, the first step in generating a hypothesis set (or refining the current hypothesis set, if it exists) is to match those sensed features to model features and derive a belief function which expresses the belief in each object hypothesis that can be derived from those matches. We do this in two steps. First, individual belief functions are derived for each sensed feature. These belief functions define the possible matches between sensed and model features, and the corresponding evidence which supports those matches. These individual belief functions are then combined to form object hypotheses which represent possible combinations of the feature matches.

The combination of the individual belief functions cannot be done by simply invoking Dempster's combination rule. The reason for this is as follows. For each sensed feature, S_i, we derive a bpa, $m_i(\cdot)$, which defines our belief in proposition of the form "sensed feature S_i matches model feature f." In other words, Θ_i, the frame of discernment for a particular $m_i(\cdot)$ only includes propositions about the i^{th} sensed feature. In order to combine these individual bpa's, we must first combine the feature matches represented by each Θ_i to obtain object hypotheses. This is done by constructing a common refinement, Ω, of all Θ_i's, then transforming the $m_i(\cdot)$'s to reflect belief over Ω. We can then combine the transformed $m_i(\cdot)$'s. Since the hypotheses discerned by Ω are obtained by combining feature matches, each element of Ω will be an object hypothesis which consists of feature matches (i.e. possible matches between sensed and model features).

There is one further difficulty in the construction of $m_f(\cdot)$. Matching sensed features to model features is a local operation, so no relational constraints can be used at this stage. In other words, a sensed feature will be matched to a model feature based solely on its similarity to that model feature. Because of this, it is possible that distinct model features will be indistinguishable to certain sensors. For example, a cube with holes of different radii in two adjacent faces has a unique labeling of faces. However, an observer viewing a single face which does not contain a hole has no way of knowing which face it is. Because of this, our system groups together features which appear equivalent without the aid of relational information. Each such grouping corresponds to one unique model feature. The set of unique model features is denoted by M_u. The set of all model features is denoted by M. A function, $u: M_u \rightarrow M$ maps unique model features onto the appropriate subsets of M. In the cube example, if the faces of the cube without a hole are labeled a,b,c, and d, and the unique feature X corresponds to a face without a hole, then u(X) = {a,b,c,d}.

In addition to $u(\cdot)$, we need a function to map sensed features onto unique features, along with a confidence in that mapping. For this purpose, we provide the function, $f_m: S \rightarrow 2^{M_u \times [0,1]}$. In other words, $f_m(S)$ is a set of 2-tuples, each of which consists of an element of M_u and a confidence value that lies in the closed interval [0,1]. How $f_m(\cdot)$ is obtained depends on the sensing operation which is being used. In general, $f_m(\cdot)$ depends on the similarity between such sensed and model feature attributes as area, surface type, and surface curvature.

To illustrate $u(\cdot)$ and $f_m(\cdot)$, consider again the example shown in Fig. 1. Since faces c and f appear identical, we group them together and give them the unique feature label f_{u1}. Now, if we obtain S_1 as shown in Fig. 1b, since the sensed feature closely resembles face a and slightly resembles the unique feature f_{u1}, we might have:

$$f_m(S_1) = \{ <a, 0.9>, <f_{u1}, 0.3> \}$$

Note that face a is a unique feature, since no other model feature appears identical to it. Further note that the sum of the confidence values assigned by $f_m(\cdot)$ need not be equal to unity.

In order to transform $f_m(\cdot)$ into a hypothesis set with a valid belief function, we must transform the confidence values into bpa's. This simply requires normalization of the confidence values, since the range is already [0,1]. We also map the unique features to the set of model features that they represent, so that the domain of $m_i(\cdot)$ will be the set of model features rather than the set of unique features.

$$m_i(u(F)) = \frac{C}{\sum_{<f,c> \in f_m(S_i)} c}$$

for each $<F,C> \in f_m(S_i)$. Note that since the range of u(F) is 2^M, it is possible that $m_i(\cdot)$ will assign nonzero belief to non-singleton subsets of M. When this occurs, it reflects the system's ignorance about which match for a particular sensed feature is best. In the example just given, we would have:

$$m_1(\{a\}) = \frac{0.9}{1.2} = 0.75$$

$$m_1(\{c, f\}) = \frac{0.3}{1.2} = 0.25$$

Once we have derived $m_i(\cdot)$ for each of the i sensed features, all that remains is to combine these using Dempster's rule to obtain $m_f(\cdot)$. Unfortunately, as stated earlier, this is not trivially done, since each $m_i(\cdot)$ is associated with a unique frame of discernment. We must first find a common refinement Ω of all Θ_i's and perform the combination in the frame Ω. We construct Ω as follows:

$$\Omega = \{\{\theta_1, \theta_2, \cdots \theta_n\} \mid \theta_i \in \Theta_i\}$$

That is, each element of Ω is a collection of feature matches, and each possible combination of feature matches (for the n sensed features) is represented in Ω. In other words, Ω discerns propositions of the form: "sensed feature S_1 matches model feature f_{i1} $\cdots$ sensed feature S_n matches model feature f_{in}."

We can also define ω_i, the refining from Θ_i to Ω as:

$$\omega_i(a) = \{\theta \mid \theta \in 2^{\Omega} \text{ and } a \in \theta\}$$

for singleton subsets of Θ_i, and

$$\omega(A) = \bigcup_{\theta \in A} \omega(\{\theta\})$$

for $A \subset \Theta$. In other words, $\omega_i(a)$ is the subset of Ω that contains all hypotheses which match sensed feature S_i to model feature a.

We now combine the $m_i(\cdot)$'s to obtain $m_f(\cdot)$. In order to do this, we need to transform the $m_i(\cdot)$'s so that they reflect belief in propositions in the frame Ω. To do this, for each $m_i(\cdot)$, we construct $m_i'(\cdot)$ as follows:

$$m_i'(\omega(\theta)) = m_i(\theta)$$

Thus, the belief that $m_i(\cdot)$ reflects in a proposition is passed on to the subset of Ω which corresponds to that proposition.

Now, we can apply Dempster's rule:

$$m_f(A) = \frac{\sum_{\theta_1 \cap \theta_2 \cdots \cap \theta_n = A} \prod_1^n m_i'(\theta_i)}{1 - \sum_{\theta_1 \cap \theta_2 \cdots \cap \theta_n = \emptyset} \prod_1^n m_i'(\theta_i)}$$

As an example, consider that we have two sensed features, S_1 and S_2. Assume that S_1 can be matched to the model features a,b and that S_2 can be matched to the model features c,d and that the nonzero $m_1(\cdot)$ and $m_2(\cdot)$ are:

$$m_1(\{a\}) = m_1(\{b\}) = 0.5$$

$$m_2(\{c,d\}) = 1.0$$

Then we construct $m_1'(\cdot)$ and $m_2'(\cdot)$ and obtain the nonzero values:

$$m_1'(\{\{a,c\}, \{a,d\}\}) = 0.5$$

$$m_1'(\{\{b,c\}, \{b,d\}\}) = 0.5$$

$$m_2'(\{\{a,c\}, \{b,c\}, \{a,d\}, \{b,d\}\}) = 1.0$$

3.2.2. Relational Consistency

Once we have derived $m_f(\cdot)$, we have established a set of object hypotheses based on the possible identities of the n features which were just sensed, say by the k^{th} sensing operation. We are now in a position to combine these hypotheses with the hypothesis set that the system might have already developed. This will produce a refined set of hypotheses to which we will apply our relational consistency measures. We will denote the system's active hypothesis set by Ω_{k-1}. We will use Ω_k to denote the hypothesis set which results from combining Ω_{k-1} with Ω (Ω being the hypothesis set obtained using local feature matches).

The construction of Ω_k is similar to the construction of Ω in the previous section. In particular, we define Ω_k as follows:

$$\Omega_k = \{\phi \cup \psi \mid \phi \in \Omega_{k-1}, \psi \in \Omega\}$$

That is, Ω_k is the set of all hypotheses which can be obtained by combining the feature matches of an existing hypothesis (i.e. some hypothesis in Ω_{k-1}) with the feature matches represented by a hypothesis in Ω.

While there will be a large number of these hypotheses, many of them can be eliminated by the application of relational constraints. For example, it is quite likely that some of the hypotheses will match sensed features to model features which are not in the same object. Since we are not currently dealing with occluding objects, we do not allow such matches. This constraint is expressed by the bpa $m_o(\cdot)$. (This restriction will be removed if we later allow for occlusion.) In addition to object consistency, we use a number of other bpa's in the derivation of $m_r(\cdot)$. These relationships vary based on the type of sensing used. For example, when 3-D features are used, they include dot products of surface normals and location of the feature relative to the object's base coordinate frame. These are represented by the bpa's $m_n(\cdot)$ and $m_l(\cdot)$, respectively.

Object consistency is enforced regardless of the type of sensing that was used to derive the features. We want $m_o(\cdot)$ to place all of its belief in the subset of hypotheses which contain only consistent matches, and no belief in any hypothesis which contains an inconsistent match. A hypothesis contains an inconsistent match if any two sensed features are matched to model features from different objects. Thus, for a hypothesis set Ω_k, we define

$$\Omega_c = \{\, \theta \mid \theta \in \Omega_k \text{ and } \theta \text{ contains no inconsistent matches} \,\}$$

and then establish $m_o(\cdot)$ as

$$m_o(\Omega_c) = 1.0$$

As we have mentioned, there are a number of additional belief functions which enter into the derivation of $m_r(\cdot)$, depending on the type of features which are involved. Rather than describe all of these here, we will present some examples in later sections of the paper which deal with the specific sensors.

3.2.3. Aspect Consistency

The final bpa which we consider in evaluating the quality of an object hypothesis is based on the idea that the system can determine which features should be observed if a pose transformation for the hypothesis has been determined, and the type of sensing operation which was applied is known. This bpa, $m_a(\cdot)$, is derived by accumulating positive evidence when expected features are matched. The concept of aspects is important to this process.

In our system, the features associated with an aspect are given weights which reflect the likelihood that they will be found by a sensing operation. These weights are a function of the type of sensor used, and how conspicuous the features are to that sensor. We will use $w_i(f)$ to represent the weight given to model feature f in the i^{th} aspect. By using w_i in conjunction with the quality of the feature matches in an object hypothesis, we derive the aspect consistency. Let q(f) represent the belief associated with matching model feature f to sensed feature S_k, $1 < k \leq n$ (i.e. if f is matched to one of the S_i, then we assign q(f) the value which reflects the quality of that match). The value of q(f) is readily obtained by referring back to the values of the $m_l(\cdot)$'s from Section 3.2.1. Furthermore, let $F_{ai}(\theta)$ represent the set of features which is visible from aspect i if the object hypothesis θ is correct. Then, if the hypothesis θ supposes that the object is being viewed from aspect i:

$$C_a(\theta) = \sum_{f \in F_{ai}(\theta)} w_i(f)\, q(f)$$

Again, we find $m_a(\cdot)$ by normalizing $C_a(\cdot)$.

4. MEASURING AMBIGUITY IN A SET OF HYPOTHESES

Now that we have described how hypothesis sets are generated and subsequently refined to admit new evidence, we need a means of characterizing the ambiguity in a hypothesis set. In our previous work, the ambiguity in a hypothesis set was trivially defined as the number of hypotheses in the set. Of course that approach will not work once an uncertain reasoning scheme is put into place. Consider, for example, a case in which none of the initial hypotheses is ever completely discounted although eventually a single hypothesis accrues enough evidence to emerge as the obvious choice. Clearly, a more sophisticated measure of ambiguity is needed.

Before defining our measure of ambiguity, let us enumerate the qualities that it should possess. If we have a set of hypotheses, with an associated bpa $m(\theta)$, we want to characterize the amount of choice that the system would be required to exercise in order declare a single hypothesis as valid. The more choice required, the higher the amount of ambiguity. Thus, our measure of ambiguity should be highest when all hypotheses are equally likely. Stated another way, given two hypothesis sets, the set whose belief function shows the greater dispersion should have less ambiguity (by dispersion, we mean the degree to which a belief function differs from a uniform distribution). Furthermore, if all hypotheses are equally likely, the ambiguity should increase with the number of hypotheses. Of course, if a hypothesis set has a single element, then its ambiguity should be 0.

Another desirable quality for a measure of ambiguity is that it be consistent across levels of a hierarchical hypothesis space. In particular, if we establish hypothesis sets in a hierarchy, then the ambiguity in a hypothesis set at one level should be equal to a weighted sum of the ambiguity in its descendants. For example, if the top level hypothesis set, H_0, is the set {A,B}, with $m_0(\{A\}) = 0.3$, and $m_0(\{B\}) = 0.7$, and we split A and B to obtain two new hypothesis sets $H_1 = \{a_1, a_2\}$, and $H_2 = \{b_1, b_2\}$, then the ambiguity in H_0 should be equal to 0.3 times the ambiguity in H_1 and 0.7 times the ambiguity in H_2.

The only continuous function satisfying these requirements is of the form:

$$A(\Omega) = -K \sum_{\theta \in \Omega} pr(\theta) \log pr(\theta)$$

where K is some positive constant, and $pr(\theta)$ is a measure of the certainty that θ is the correct hypothesis. A proof of this can be found in [18]. The form of $A(\cdot)$ is not totally unfamiliar. It is also the form of the entropy measure from information theory. This is no mere coincidence, since information theorists use entropy to measure the freedom of choice available in selecting a message, provided that the probabilities associated with the choices are known.

Other work on characterizing the entropy in a hypothesis set has been done by Stephanou and Lu [19], Yager [21], and Higashi and Klir [10]. The measure described in [19] does not suit our purposes because it awards equal entropy to hypothesis sets with different numbers of elements in the case of total ignorance (i.e. the belief function assigns belief of 1.0 to the total frame, and no belief to any subset of the frame). The measure developed in [21] fails to meet our criteria because if any two focal elements have a non-empty intersection, the entropy is 0. Finally, the entropy measure described in [10] fails to satisfy our condition that the entropy be consistent over levels in a hierarchy of hypothesis spaces.

In our equation for ambiguity, note that we did not use $m(\cdot)$ to represent the likelihood that a particular hypothesis was correct. This is because there will be situations in which $m(\cdot)$ assigns belief to non-singleton subsets of Ω_k, and no belief to individual hypotheses. In such cases, we must still be able assess the likelihood of the individual hypotheses. For this purpose, we calculate $pr(\theta)$ as follows:

$$pr(\theta) = \sum_{\theta \in A} \frac{m(A)}{|A|}$$

In this way, when $m(\cdot)$ assigns belief to a non-singleton subset of Ω_k, for the purpose of calculating ambiguity, we treat the individual elements of that subset as being equally likely.

In order to apply this measure of ambiguity to the problem of selecting a best next sensing operation, we predict the hypothesis sets which might occur if a particular sensing operation is applied. We then find the ambiguity associated with each of these possible hypothesis sets, and use the worst case value as a measure of the effectiveness of that sensing operation. We will use the symbol A^i_{max} to refer to the maximum ambiguity associated with a proposed i^{th} sensing operation. The goal of the system is then to choose a sensing operation which minimizes the value of A^i_{max}.

5. OBSERVABLE FEATURES

In this section, we describe the features which can be observed by each of the sensors that our system uses. These sensors include a structured light scanner to obtain 3-D information about the scene, overhead and a manipulator held cameras to obtain 2-D information about the scene, a force/torque sensor mounted on the robot's wrist, and a manipulator which can be queried to find the distance between its fingers.

5.1. 3-D Features

The richest set of features available to the system comes from range data. Range data is gathered for a set of points in the scene, using a range scanner which the robot manipulates. This initial data is converted to *x,y,z* data. Subsequent processing of this x,y,z data produces a list of surfaces, attributes of those surfaces and relations between the surfaces. The types of attributes provided by range data processing include surface area, orientation, location, surface type, etc. Relations include adjacency, coplanarity, etc. The methods that we use to determine 3-D features are documented in [2,5,23,24].

As we mentioned earlier, each type of sensor has associated methods for determining the quality of feature matches and relational consistency. While this paper is not greatly concerned with how these values are determined, for clarity we now briefly describe the derivation of $m_n(\cdot)$ and $m_l(\cdot)$. These bpa's assess the confidence in an object hypothesis based on dot products of surface normals and location of surfaces relative to the object's base coordinate frame.

In order to derive $m_n(\cdot)$, we need to define two additional functions: $n_S(\phi)$ returns the surface normal of the sensed feature matched in ϕ, and $n_M(\phi)$ returns the surface normal of the model feature matched in ϕ. Note that ϕ corresponds to a feature match in a hypothesis (i.e. each element of Ω_k corresponds to a single object hypothesis which contains a number of matches between sensed and model features). Using these two functions, we can compute the magnitude of the difference in dot products of sensed and model surface normals as follows.

$$E = | n_S(\phi) \cdot n_S(\psi) - n_M(\phi) \cdot n_M(\psi) |$$

for ϕ and ψ in θ, and $\theta \in \Omega_k$. Since E is the magnitude of the difference in two values which are in the interval [0,1], E will lie in the interval [0,2], with E=0 corresponding to an exact match, and E=2 corresponding to the worst possible error. In order to capture the notion of conjunction, we define $C_n(\theta)$ as:

$$C_n(\theta) = \prod_{\phi,\psi \in \theta} (2 - | n_S(\phi) \cdot n_S(\psi) - n_M(\phi) \cdot n_M(\psi) |)$$

Finally, we transform C_n into a bpa,

$$m_n(\theta) = \frac{C_n(\theta)}{\sum_{\psi \in \Omega_k} C_n(\psi)}$$

If we have enough feature matches in a hypothesis, we can derive a pose transformation for that hypothesis, T_{obj}. We can then use this transformation to measure the quality of a match between sensed and model features based on the proximity of the sensed feature to the location at which we expect to find it based on T_{obj}. We use the function $L_S(\phi)$ to refer to the location of the sensed feature matched in ϕ, and $L_M(\phi)$ to refer to the location of the model feature matched in ϕ. Therefore, the difference in $L_S(\phi)$ and $T_{obj}L_M(\phi)$ is a measure of the quality of the match expressed in ϕ. Since this difference is essentially unbounded, we apply a weighting factor.

$$c(\phi) = 1 - | L_S(\phi) - T_{obj}L_M(\phi) | e^{-\tau}$$

The exponent τ controls how quickly the exponential function decays and is based on the accuracy of the sensors used. We combine the $c(\cdot)$'s to obtain a confidence in the proposition θ by taking their product over the feature matches in θ.

$$C_1(\theta) = \prod_{\phi \in \theta} c(\phi)$$

We obtain $m_1(\cdot)$ by normalizing $C_1(\cdot)$.

5.2. 2-D Features

The features which are visible to the 2-D camera are not nearly as robust as those visible to the range scanner. In particular, as mentioned earlier, surface types typically cannot be determined from 2-D data, edge detection is not as good (since only gray level edge detecting can be used), and relationships between surfaces cannot be measured (except for adjacency). The primary advantage of 2-D vision is that it is computationally less expensive than 3-D vision. Also, since our range scanner is held by the robot, and one robot move is required for each projected stripe, using 2-D vision reduces the number of required manipulations from the large number required to scan a scene to the much smaller number required to grasp the hand held camera and position it at the appropriate viewpoint.

The local features (i.e. features that are confined to local areas of the object, such as a single surface or edge) that we can obtain from 2-D image processing include holes in the object, surface texture and intensity edge information. In our current experiments, the object surfaces are all smooth, containing little or no surface texture information. Therefore, the primary 2-D features that we use are holes and grey level edges.

Although gray level edge detection is not as robust as the 3-D edge detection, it is generally much faster. Furthermore, using object hypotheses to guide the application of the edge detector, the problem is reduced from edge detection to edge verification. In particular, once we have an object hypothesis which includes a position hypothesis, we can predict the set of edges visible to the 2-D camera. If we know the camera transformation, we can predict where these edges will be found in the image plane. The image obtained from the camera can then be used to verify the presence of the edge. This edge verification is done using the Dempster-Schafer formalism applied to a binary frame of discernment (i.e. edge-present/edge-not-present) [16].

In addition to using the hand held 2-D camera to derive 2-D features, our system also uses an overhead camera to guide the initial application of the range scanner. In particular, the overhead camera is used to obtain an estimate of the positions and orientations of the objects in the work space. This initial application of the 2-D camera can also measure certain global features about the objects, for example: aspect ratio, moments of inertia, and object size.

5.3. F/T Sensed Features

The last type of sensing that our system can perform is active sensing of the environment using the robot manipulator. In our current system, the manipulator can be used in either of two ways. Its fingers can be closed on an object to measure its width, or, the manipulator fingers can be closed, and used as a probe. When in the latter mode, force/torque sensing is used to execute a guarded move toward an object feature to precisely measure its height. Using these techniques, we can precisely (to within the known error of the manipulator position) measure features on the objects in the world. Like range scanning, using this type of sensing requires the active participation of the robot, thus incurring the additional overhead of planning and executing robot motions.

The utility of measuring object widths becomes evident when we have competing object hypotheses, and the difference in sizes of visible features of the two objects is less than what can be perceived by the 3-D or 2-D vision systems. Of course 2-D vision is very imprecise, due to the use of an inverse perspective transformation which estimates the world Z coordinate. Using our current range scanner, precision in 3-D data is a function (among other things) of the baseline distance between the camera and the stripe projector [5]. Furthermore, the smallest feature which can be detected using the range scanner is a function of the distance between projected light stripes. To compensate for these inaccuracies, the manipulator can be used to perform the more precise measurements, only when they are required.

Measuring the height of object surfaces becomes particularly useful when those surfaces are obscured from the view of the vision systems. In such cases, the manipulator can be used as a probe to resolve the ambiguities. Probing is achieved by executing guarded motion toward the surface whose height is to be measured. When a threshold force is exceeded (indicating contact with the object), the position of the manipulator is used to determine the actual height of the surface. Manipulator probing can also be used to determine the existence of protrusions from object surfaces, especially when these protrusions are obscured from the view of the vision sensors (e.g. when the work piece is positioned such that it occludes the surface which has the protrusion).

Object features which can be detected by using the manipulator as a sensor are stored in tables. These tables are indexed by object surfaces so that it is a simple matter to determine which of these features might be present once object hypotheses have been made (since each object hypothesis includes a list of matches between sensed and model surfaces). Thus, for any object hypothesis, it is a simple matter to consult a table to obtain, for example, a list of holes and protrusions which are a part of the surfaces of the hypothesized object. When such features are sufficient to distinguish between competing hypotheses, the manipulator is used to measure them.

Determining when to use the manipulator to resolve ambiguities that are too subtle to be observed by the vision systems is more difficult, since the resolution from the vision systems is dependent on the implementation and run time parameters of those systems. In light of this difficulty, we have chosen to fix an upper bound on the resolution which can be obtained by using the vision systems. We implicitly represent this bound by enumerating the pairs of features which can only be differentiated by using the manipulator to perform measurements. For example, if the widths of two objects are so close that the 3-D vision system cannot distinguish between them, then it is noted that a precise manipulator measurement of their widths can be used to discriminate between the two. Currently, we have not fully implemented this part of the system.

6. CHOOSING THE BEST SENSING STRATEGY

In this section, we will describe the algorithm which is used to choose a sensing strategy. In essence, this is simply a search problem. The search space consists of the possible sensing operations from the possible viewpoints. Goal states are recognized using A^i_{max}. As we have mentioned earlier, since the space of sensing operations can be arbitrarily large (consider that the manipulator can be used *anywhere* in the robot's work envelope), we must use some heuristic to guide the search. The method that we use is to only consider sensing operations applied from the principal viewpoints of aspects of the hypothesized objects. Once we have limited the number of sensing operations which will be considered, each is investigated until one is found which produces a sufficiently small A^i_{max}.

There are three basic components to the algorithm. First, the function, *predict-ambiguity* computes the predicted ambiguity for a specified view point, hypothesis set, sensor and set of predicted feature values. (The predicted feature values are computed based on the object hypothesis, sensor and proposed viewpoint by determining which aspect of the object would be viewed by the proposed sensing operation.) The first step in predict-ambiguity is to refine the hypothesis set using the predicted feature values (as described in Section 3). Once this is done, the ambiguity is calculated and returned. This algorithm is shown in Fig. 2. The function *max-ambiguity* calls predict-ambiguity with different predicted feature sets, recording its maximum value for a candidate sensing operation.

```
predict-ambiguity(VP,Ω_{k-1},S,sensor)
    Ω_k ← refine-hyp-set(Ω_{k-1},S)
    A ← 0
    foreach θ ∈ Ω_k
        A ← A - pr(θ) log pr(θ)
    return(A)
```

Fig. 2: Algorithm for predict-ambiguity.

Finally, the top level function used to determine the next sensing operation is *choose-next-view*, shown in Fig. 3. This function merely iterates over each possible node in the aspect graphs for each object hypothesis for each possible sensor. Note the use of the predicate *valid-vp*. This predicate is used to insure that candidate viewpoints can actually be achieved using the robot (e.g. viewpoints which lie below the work table are eliminated from consideration).

7. EXPERIMENTAL RESULTS

So far, we have not yet fully implemented the uncertain reasoning system described in Section 3. We have verified our approach using a standard hypothesis refinement technique using the object shown in Fig. 4. Notice that the orientation of this object can be determined only if the position of the hole in one end is known.

First, a geometric model was created for the part using the PADL2 system [9], a CSG based modeler, which we have modified so that it can be interfaced with a LISP environment. The aspect graph was constructed by using PADL2 to

```
choose-next-view(Ω_{k-1})
    Amax ← 100
    foreach h ∈ Ω_{k-1}
        T ← h.transform
        Node-list ← h.aspect-graph.nodes
        foreach S ∈ sensors
            foreach node ∈ Node-list
                VP ← node.princple-view
                W-VP ← T * VP
                NAmax ← max-ambiguity(Ω_{k-1},W-VP,S)
                if (valid-vp(W-VP) and NAmax < Amax) then
                    Amax ← NAmax
                    Sensor ← S
                    V ← W-VP
                if (Amax < δ ) then
                    return(Amax,V,Sensor)
    return(Amax,V,Sensor)
```

Fig. 3: Algorithm for choose-next-view.

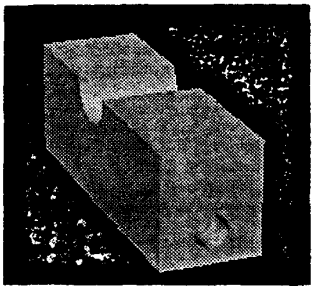

Fig. 4: Rendering of experimental object.

automatically view the object from each of the 60 tessels on the viewing sphere. Tessels which viewed the same set of surfaces were grouped together into aspects, and aspects containing adjacent tessels were linked by arcs.

Once the model was created, the part was placed in the robot's work space. Range scanning was done (using the structured light scanner), finding the three top surfaces. Using these surfaces, and their attributes, the system was able to develop two competing hypotheses, one merely a 180 degree rotation of the other. Given these two hypotheses, our algorithm chose the next sensing operation to be viewing the object with the hand held 2-D camera as shown in Fig. 5. As can be seen in the figure, this sensing operation allows the end surface of the object to be viewed, and thus the presence or absence of the hole in that surface will determine the object's orientation.

8. CONCLUSIONS

In this paper, we have addressed the issue of planning sensing strategies dynamically, based on an active set of hypotheses. Our algorithm uses the aspect graphs of the hypothesized objects to propose candidate sensing operations. Then, using the pose transformation and aspect graph which are associated with each hypothesis we predict the feature sets which would be observed upon application of the candidate sensing operation. Given these predictions, we are also able to predict the resulting set of hypotheses which could remain active. By repeating this process for different viewpoints and sensing operations, we are able to choose the sensing operation which minimizes the maximum ambiguity in these possible hypothesis sets, thereby minimizing the amount of ambiguity which can remain after the next sensing operation is applied.

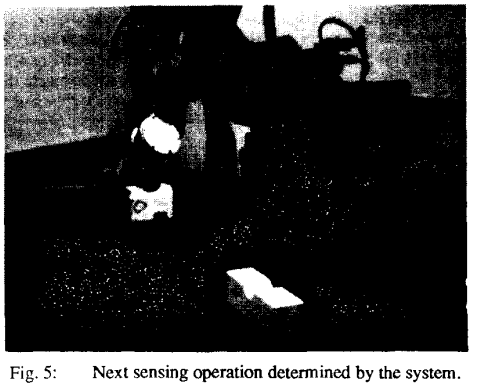

Fig. 5: Next sensing operation determined by the system.

REFERENCES

[1] G. Castore, C. Crawford, "From Solid Model to Robot Vision," *Proc. of the IEEE Int'l Conf. on Robotics and Automation*, 1984, pp. 90-92.

[2] C. H. Chen and A. C. Kak, "Modeling and Calibration of a Structured Light Scanner for 3-D Robot Vision," *Proc. of the IEEE Int'l Conf. on Robotics and Automation*, 1987, pp. 807-815.

[3] C. I. Connolly, "The Determination of Next Best Views," *Proc. of the IEEE Int'l Conf. on Robotics and Automation*, 1985, pp. 432-435.

[4] G. K. Cowan and P. D. Kovesi "Automatic Sensor Placement from Vision Task Requirements," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 10, No. 3, May 1988, pp. 407-416.

[5] R. L. Cromwell, "Low and Intermediate Level Processing of Range Maps," Purdue University Technical Report TR-EE-87-41, 1987.

[6] Z. Gigus and H. Malik "Computing the Aspect Graph for Line Drawings of Polyhedral Objects," *Proc. of the Computer Society Conf. on Computer Vision and Pattern Recognition*, IEEE Computer Society Press, 1988, pp. 654-651.

[7] G. Hager and M. Mintz, "Searching for Information," *Proc. of the AAAI Workshop on Spatial Reasoning and Multi-sensor Fusion*, 1987, Morgan Kaufmann Publishers, Inc., pp. 313-322.

[8] C. Hansen and T. Henderson, "CAGD-Based Computer Vision," *Proc. of the Workshop on Computer Vision*, Dec. 1987, pp. 100-105.

[9] E. E. Hartquist and H. A. Marisa, *PADL-2 User's Manual*, Production Automation Project, University of Rochester, 1985.

[10] M. Higashi, and G. J. Klir, "Measures of Uncertainty and Information Based on Possibility Distributions," *Int'l Journal of General Systems*, Vol. 9, 1982, pp. 43-58.

[11] R. A. Hummel, and M. S. Landy, "A Statistical Viewpoint on the Theory of Evidence," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 10, No. 2, March 1988, pp. 235-247.

[12] S. A. Hutchinson, R. L. Cromwell and A. C. Kak, "Planning Sensing Strategies in a Robot Work Cell with Multi-Sensor Capabilities," *Proc. of the IEEE Int'l Conf. on Robotics and Automation*, 1988, pp. 1068-1075.

[13] K. Ikeuchi, "Generating an Interpretation Tree from a CAD Model for 3D-Object Recognition in Bin Picking Tasks," *Int'l Journal of Computer Vision*, 1987, pp. 145-165.

[14] H. S. Kim, R. C. Jain and R. A. Volz, "Object Recognition Using Multiple Views," *Proc. of the IEEE Int'l Conf. on Robotics and Automation*, 1985 pp. 28-33.

[15] J. J. Koenderink and A. J. Van Doorn, "The Internal Representation of Solid Shape with Respect to Vision," *Biological Cybernetics*, Vol. 32, 1979, pp. 211-216.

[16] R. J. Safranek, S. N. Gottschlich and A. C. Kak, "Evidence Accumulation Using Binary Frames of Discernment for Verification Vision," Submitted for publication.

[17] G. Shafer, *A Mathematical Theory of Evidence*, Princeton University Press, 1976, Princeton.

[18] C. E. Shannon and W. Weaver, *The Mathematical Theory of Communication*, University of Illinois Press, 1963, Urbana.

[19] H. E. Stephanou, and S. Y. Lu, "Measuring Consensus Effectiveness by a Generalized Entropy Criterion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 10, No. 4, July 1988, pp. 544-554.

20] J. H. Stewman, K. W. Bowyer, "Aspect Graphs for Convex Planar-Face Objects," *Proc. of the IEEE Workshop on Computer Vision*, Dec. 1987.

21] R. R. Yager, "Entropy and Specificity in a Mathematical Theory of Evidence," *Int'l Journal of General Systems*, Vol. 9, 1983, pp. 249-260.

22] H. S. Yang and A. C. Kak, "Determination of the Identity, Position, and Orientation of the Topmost Object in a Pile," *Proc. Third IEEE Workshop on Computer Vision: Representation and Control*, Oct. 1985, pp. 38-48.

[23] H. S. Yang and A. C. Kak, "Determination of the Identity, Position, and Orientation of the Topmost Object In a Pile," *Computer Vision, Graphics, and Image Processing*, Vol. 36, 1986, pp. 229-255.

[24] H. S. Yang and A. C. Kak, "Determination of the Identity, Position, and Orientation of the Topmost Object in a Pile: Some Further Experiments," *Proc. of the IEEE Int'l Conf. on Robotics and Automation*, 1986, pp. 293-298.