convenient from the practical viewpoint that rough models are used because of the ease of model input. Thus, the availability of rough models can be considered to be an advantage of this system.

Object models can be approximated either by omitting details or replacing real complicated shapes by simple primitives. In the experiments, bolts and nuts were ignored and not used in recognition. The middle section with a complicated shape like a sphere in the object in Fig. 10 was also omitted from recognition due to the difficulty of description. Although the part is represented by a cylinder connecting the two flanges in Fig. 10, this is only for display purposes and is not used in the strategy generation process. A human operator must indicate this point to the system. In the object in Fig. 13, the handle section with a torus and some other primitives are approximated by a flat cylinder. Thus, only one 3-D disc FGE, which generates an ellipse, is considered in the strategy generation process. Although this approximation is valid for recognition, it cannot be used for localization. The operator must also point out to the system when such an approximation is adopted.

We believe that such approximations are natural for a human operator. However, further study will be required on the suitability of approximated models. In particular, when similar objects may exist in a target scene, a modeling system that can interactively or automatically construct models from which efficient strategies to discriminate such similar objects can be generated is desirable.

VIII. Conclusion

A new model-based vision system has been proposed in this paper. The system generates a 3-D object recognition strategy, or order, for searching for features from a 3-D model of an object. A feature detector based on a hypothesize-and-test process finds features according to the strategy and recognizes the object. Then, the precise position and attitude of the object are obtained by comparing a line representation generated from the model with the image features. Experiments have yielded promising results. Active use of the system for robot vision will be studied to realize a practical vision system that operates under various conditions.

References

[1] R. A. Brooks, "Symbolic reasoning among 3-D models and 2-D images," *Artificial Intell.*, vol. 17, pp. 285–348, Aug. 1981.
[2] C. Goad, "Special purpose automatic programming for 3D model-based vision," in *Proc. Image Understanding Workshop* (Arlington, VA), June 1983, pp. 94–104.
[3] C.K. Cowan, D.M. Chelberg, and H.S. Lim, "ACRONYM model-based vision in the intelligent task automation project," in *Proc. AI Applications* (Denver, CO), Dec. 1984, pp. 176–183.
[4] K. Ikeuchi and T. Kanade, "Automatic generation of object recognition programs," *Proc. IEEE*, vol. 76, pp. 1016–1035, Aug. 1988.
[5] K.S. Hong and K. Ikeuchi, "Generating a strategy for configuration determination: A module of vision algorithm compiler for object localization programs," in *Preprints 5th Int. Symp. Robotics Res.* (Tokyo), Aug. 1989, pp. 164–171.
[6] Y. Lamdan and H.J. Wolfson, "Geometric hashing: A general and efficient model-based recognition scheme," in *Proc. 2nd ICCV* (Tampa, FL), Dec. 1988, pp. 238–249.
[7] C. Hansen and T. Henderson, "Towards the automatic generation of recognition strategies," in *Proc. 2nd ICCV* (Tampa, FL), Dec. 1988, pp. 275–279.
[8] R.C. Bolles and R.A. Cain, "Recognizing and locating partially visible objects: The local-feature-focus method," *Int. J. Robotics Res.*, vol. 1, no. 3, pp. 57–82, 1982.
[9] R.C. Bolles and P. Horaud, "3DPO: A three-dimensional part orientation system," *Int. J. Robotics Research*, vol. 5, no. 3, pp. 3–26, 1986.
[10] W.E.L. Grimson and T. Lozano-Pérez, "Model-based recognition and localization from sparse range or tactile data," *Int. J. Robotics Res.*, vol. 3, no. 3, pp. 3–35, 1984.
[11] D. Shoham and S. Ullman, "Aligning a model to an image using minimal information," in *Proc. 2nd ICCV* (Tampa, FL), Dec. 1988, pp. 259–263.
[12] D.G. Lowe, "Three-dimensional object recognition from single two-dimensional images," *Artificial Intell.*, vol. 31, pp. 355–395, Mar. 1987.
[13] ——, "The viewpoint consistency constraint," *Int. J. Comput. Vision*, vol. 1, no. 1, pp. 57–72, 1987.
[14] D. Marr and H.K. Nishihara, "Visual information processing: Artificial intelligence and the sensorium of sight," *Technol. Rev.*, vol. 81, pp. 2–23, Oct. 1978.
[15] Y. Okamoto, Y. Kuno, K. Onoguchi, M. Watanabe, and H. Asada, "Object recognition using a tree-like procedure generated from 3-D model," in *Proc. IAPR Workshop Comput. Vision* (Tokyo), Oct. 1988, pp. 441–446.
[16] R. Horaud, "New methods for matching 3-D objects with single perspective views," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. PAMI-9, pp. 401–412, May 1987.
[17] M. Dhome, M. Richetin, J.T. Lapresté, and G. Rives, "Determination of the attitude of 3-D objects from a single perspective view," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. 11, pp. 1265–1278, Dec. 1989.
[18] J. Ponce and D.J. Kriegman, "On recognizing and positioning curved 3D objects from image contours," in *Proc. IEEE Workshop Interpretation 3D Scenes* (Austin, TX), Nov. 1989, pp. 61–67.
[19] Y. Okamoto, Y. Kuno, and S. Okada, "Robot vision using object models for recognition strategy generation and precise localization," in *Proc. IECON '90 16th Ann. Conf. IEEE Ind. Electron. Soc.* (Pacific Grove, CA), Nov. 1990, pp. 558–563.
[20] M. Ishikawa and H. Hoshino, "High performance image processor for robot vision," in *Spring Nat. Conv. Rec., Inst. Electron, Inform. Commun. Eng. Japan* (Tokyo), Mar. 1988, p. 2–187 (in Japanese).
[21] R. Bajcsy, "Active perception," *Proc. IEEE*, vol. 76, pp. 996–1005, Aug. 1988.

Achieving Generalized Object Recognition through Reasoning about Association of Function to Structure

Louise Stark and Kevin Bowyer

***Abstract*— Much current work in computer vision assumes that the recognition system will have a knowledge base consisting of, or derived from, an exact geometric model of each object that may be encountered. The purpose of the work described here is to demonstrate the feasibility of defining an object category in terms of the functional properties shared by all objects in the category. This form of representation should allow much greater generality. A complete system has been implemented that takes the boundary surface description of a 3-D object as its input and attempts to recognize whether the object belongs to the category chair and, if so, into which subcategory it falls. This is, to our knowledge, the first (only) implemented system to explore the use of a purely function-based definition of an object category (that is, no explicit geometric or structural model) to recognize 3-D objects. System competence has been evaluated on a database of over 100 objects, and the results largely agree with human interpretation of the objects.**

***Index Terms*— Computer vision, function-based modeling, function-based object recognition, shape analysis, 3-D object representation.**

Manuscript received October 1, 1990; revised December 30, 1990.
This work was supported by Air Force Office of Scientific Research grant AFOSR-89-0036, National Science Foundation grant IRI-8817776, and a Patricia Harris Fellowship.
The auhors are with the Department of Computer Science and Engineering, University of South Florida, Tampa, FL 33620.
IEEE Log Number 9102649.

0162–8828/91$01.00 © 1991 IEEE

I. Introduction

Model-based vision has been popular for some time yet still appears far from being able to demonstrate any general-purpose 3-D object recognition system. One current "hot" paradigm is "CAD-based vision" — the use of exact geometric descriptions as might be available from a CAD system. With a CAD-based vision system, a unique 3-D model is stored for each object that the system is able to recognize. Recognition may require, in the worst case, that the input stimuli be compared to each model. Another problem encountered with such systems is that the size of the database grows in direct proportion to the number of objects the system is made capable of recognizing. One way of alleviating this problem to some degree is to allow parameterized representations so that objects that have the same essential geometry or structure can be recognized [2], [6], [7]. Still, it seems impossible to anticipate and parameterize all possible geometric and/or structural variations that may occur within an object category.

Consider the domain of human artifacts, that is, man-made objects that serve some specific purpose that is reflected in their external physical structure (e.g., furniture, hand tools, utensils). For any particular object category, there is some set of functional properties shared by all objects in that category. It is part of the thesis of our work that the existence or nonexistence of these properties can be deduced by analyzing the shape of an object and that this information can be used for recognition (or, if you like, categorization). Rather than concentrating our initial efforts on a purely theoretical elaboration of this concept, we have chosen to develop a complete system for a particular case study category. Our system represents the definition of object categories and subcategories in terms of required functional properties and represents the functional properties using procedural knowledge. A major advantage of this representation scheme is that the system can recognize truly novel objects, at least at the category level, even though *the system knows no specific geometric or structural model for any object.*

Section II reviews related research dealing with function-based representation. Section III describes the recognition system, followed by a detailed example and experimental results of the analysis of over 100 objects in Section IV. The paper concludes in Section V with suggestions for future directions of research.

Before proceeding, it is best to explicitly define some of the terminology we have adopted:

- *Category*: Using Rosch's terminology, we are considering the basic level category [10]. Rosch states that "basic categories are those which carry the most information, possess the highest category of validity, and are, thus, the most differentiated from one another" (see p. 382 of [10]).
- *Subcategory*: the term given subordinate categories (categories below the basic level). Each subcategory has its own set of functional attributes that may overlap with other subcategories.
- *Input Object*: an input to the system in the form of an uninterpreted 3-D boundary description.
- *Exemplar*: an object categorized by the system as belonging to a specific subcategory.
- *Functional Plan*: the function-based definition of a specific category or subcategory.
- *Function Label*: simply a name for the functional property being evaluated, for example, *provides sittable surface.*
- *Functional Element*: a portion of the input object that fulfills the functional requirements associated with a specific *function label.* There are three types of functional elements that can be identified: 1) a single surface of the object, such as the seat of a chair that provides a sittable surface; 2) a group of surfaces acting together to fulfill the required function, such as slats on the back of a chair act together to provide back support; 3) a three-dimensional portion (module) of the structure.
- *Association Measure*: a measure that reflects the strength of the association of the function label to the functional element or, cumulatively, the strength of the (sub)category membership of an object.
- *Procedural Knowledge Primitive (PKP)*: primitive procedures used to qualitatively evaluate the shape of an input object.

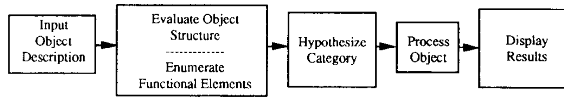

Fig. 1. Flow of execution.

II. Background

Winston *et al.* have discussed the use of function-based definitions of object categories [13]. They point out that there can be an infinity of individual physical descriptions for objects in a category as simple as "cup" but that a single functional description can be used to represent all possible cups in a concise manner. This work is, of course, related to Winston's classic "arch-learning" program [14]. This earlier program was able to learn structural descriptions (*not* function-based descriptions) of object families, such as "arch," from line drawings of examples.

Brady *et al.* also discussed the relation between geometric structure and functional significance in their design of the "Mechanic's Mate" system [1], [3]. In part of this work, semantic net descriptions are computed from 2-D shapes, and a generalized structural description is learned from a sequence of positive examples.

Part of the inspiration for our work came from ideas expressed by Minsky in his recent book [9] and in network news articles. In fact, the category chair is used as an example by Minsky in his suggestion that knowledge about function must be combined with knowledge about structure.

Efforts that are more recent and closely related to ours are those of Ho [8] and of DiManzo *et al.* [4]. Ho considers two specific functional concepts (chair and support) in the context of what is needed to represent function for recognition. The analysis is done in the ideal 2-D cross section of the object and assumes that the object appears in its upright orientation. DiManzo proposes a system design that utilizes functional knowledge within an expert system framework. Primitives are defined in the form of individual expert systems that evaluate the 3-D information. A prototype system is being implemented that receives a description of a scene generated by an octree solid modeler.

III. System Description

A high-level diagram of the system is depicted in Fig. 1.

This system reads the boundary description of an unknown 3-D polyhedral object in terms of face lists and vertex coordinates and, without user intervention, attempts to recognize whether the object belongs to the category chair and, if so, into which subcategory it falls. The size of the input object is treated as actual metric units so that objects may be "too big" or "too small" to function properly. (The system has the option of scaling the input object prior to analysis. The scale factor is calculated as the ratio of the volume of the convex hull of the input object to the volume of the convex hull of a "typical" straight back chair.)

In the first stage of the evaluation process, the input object is analyzed to identify all potential *functional elements.* This includes a list of individual surfaces (related to the faces of the object) and a list of combined surfaces. A *function label* can be associated to any of the three types of functional elements described above. The categorization performed by the system identifies functional elements of an input object by associating them with their proper function label.

At this time, the hypothesis of category chair is always made by the system without using any information derived from the structure of the input object. When the number of categories represented is expanded, heuristics will be invoked to hypothesize and prioritize a subset of categories. For example, one possible heuristic could evaluate the size of the object and select possible categories according to expected size ranges. For example, the 3-D volume of a couch would typically be much greater than a chair.

Processing of the object is guided by the function-based definition of the hypothesized category. This control structure holds the definition of the individual functional plans. Each functional plan has associated requirements. In turn, each requirement is processed as an ordered execution of primitives that qualitatively evaluate the input shape. We have identified a set of five PKP's that can be used to define functional requirements for the category *chair*.

The output of the system consists of whether the input object belongs to the category chair and, if so, into which subcategory it falls, as well as a cumulative association measure.

A. Procedural Knowledge Primitives

Each function label is defined using a combination of PKP's. The PKP's currently used are *relative orientation*, *dimensions*, *stability*, *proximity*, and *clearance*. (This list is not assumed to be complete for all possible categories, but we expect it to be sufficient for the superordinate category furniture.) These primitives are procedures that make qualitative decisions about whether an object possesses a certain primitive property. During the initial system design, we began with a somewhat lengthier list of what we felt intuitively were the primitive functional concepts. As our system progressed, we often found that several of our intuitive primitives (for example, essentially parallel and essentially orthogonal) could be subsumed into one general routine (relative orientation), which was actually more useful (when we added the functional plan of the subcategory lounge chair).

The PKP *relative orientation* analyzes the orientation between two surfaces by evaluating the angle between the surface normals. For example, the sittable surface of the chair is expected to be essentially parallel to the ground plane in the chair's stable orientation. Some allowable ranges of orientation are more lenient than others. For example, the back support of a lounge chair can take on a large range of orientations relative to the sittable surface.

The PKP *dimensions* tests the potential functional element using multiple metrics. For example, the sittable surface of the chair is expected to be within a certain size range (depth and width) and to be situated within a set range above the ground (height).

The PKP *stability* is required for all subcategories of chair. For the sittable surface or seat rest to be maintained in its required orientation, the chair must *provide stable support*. Stable support is established by finding the convex hull of the contact points of the object with the ground plane in a given orientation. If a vector from the center of mass of the object perpendicular to the ground plane projects within the convex hull of the contact points, then the object is considered to be stable. To test if the object can act as a chair, the system applies weight to a distribution of points on the candidate sittable surface. This simply shifts the center of mass of the object, and therefore, the same stability test can be reapplied.

The *proximity* PKP tests to make sure two surfaces are in the proper proximity. For example, for a functional element to act as a back support, it must be close to the sittable surface and opposite an accessible area (i.e., the front of the seat). The surface must also be above the level of the sittable surface and be approximately centered relative to the sittable surface.

The PKP *clearance* is simple but extremely important. The functional elements may all be of the proper dimensions and be situated in the proper orientation to perform the functional requirements, but if the elements are not accessible by the user, they cannot be considered valid. Clearance is established by specifying the area that is expected to be accessible by the user and making sure there are no obstructions present. For example, the sittable surface must be clear above and "in front of" so that there is room for the person's torso and legs.

PKP's are invoked in a sequence dependent on the subcategory functional plan. All PKP's return an association measure that reflects how well the functional requirements are met.

B. Structure of the Class Definition for Chair

The functional representation of each category is organized in a hierarchical graph (Fig. 2). This graph is also a control structure for the evaluation process. Each node of the graph is represented by a frame having four fields: *Name, Type, Realized By*, and *Functional Plans*. The Name field holds a unique identifier. Nodes are one of three types: *Category, Subcategory*, or *Function*. The root node in Fig. 2 is of type *Category*, being a basic-level category. The Functional Plans field has as many arcs as there are subcategories defined for that node. For example, in our current implementation, we have defined four subcategories: Conventional Chair, Balans Chair, Lounge Chair, and Highchair.

The graph structure of Fig. 2 represents our function-based description of the category *Chair*. Each subgraph formed with a subcategory frame as its root denotes a separate functional plan. Therefore, the function-based description of the subcategory Lounge Chair is realized by a totally different functional plan than that of the Balans Chair.

The final field of the frame is the *Realized By* field. This field points to an ordered list of function labels. The applicability of a given function label is evaluated by the sequence of PKP invocations associated with the function label node. For example, Conventional Chair requires the functions *provides sittable surface* and *provides stable support*. Both of these function labels must be satisfied at some threshold association measure in order to consider the object to be falling within the subcategory of Conventional Chair. It should be noted that there may be multiple potential results for a given object, each with its own association measure.

Each function label has its own specified constraint values for each PKP invocation depending on the functional requirement being evaluated. These values are stored in a constraint list that is associated to the category definition. The constraint list is made up of unique constraint identifiers, along with minimum, maximum, and average values for each. These constraint values have been gathered from sources that summarize the results of ergonomic design research [5].

The base values for the accumulation of the association measure originate with the PKP invocations. For a given PKP invocation, a qualitative decision is first made as to whether there is any functional element of the input object that satisfies the specified constraint range. If not, then a measure of zero is returned for the PKP invocation; otherwise, a list of functional elements with measures between zero and one is returned. This list of elements may then be input to another PKP invocation. If a required function label for a given (sub)category has no possible elements, then the association measure for the (sub)category may go to zero and further analysis for that (sub)category discontinued. The association measure is passed back to the current (sub)category, and the association measures of the

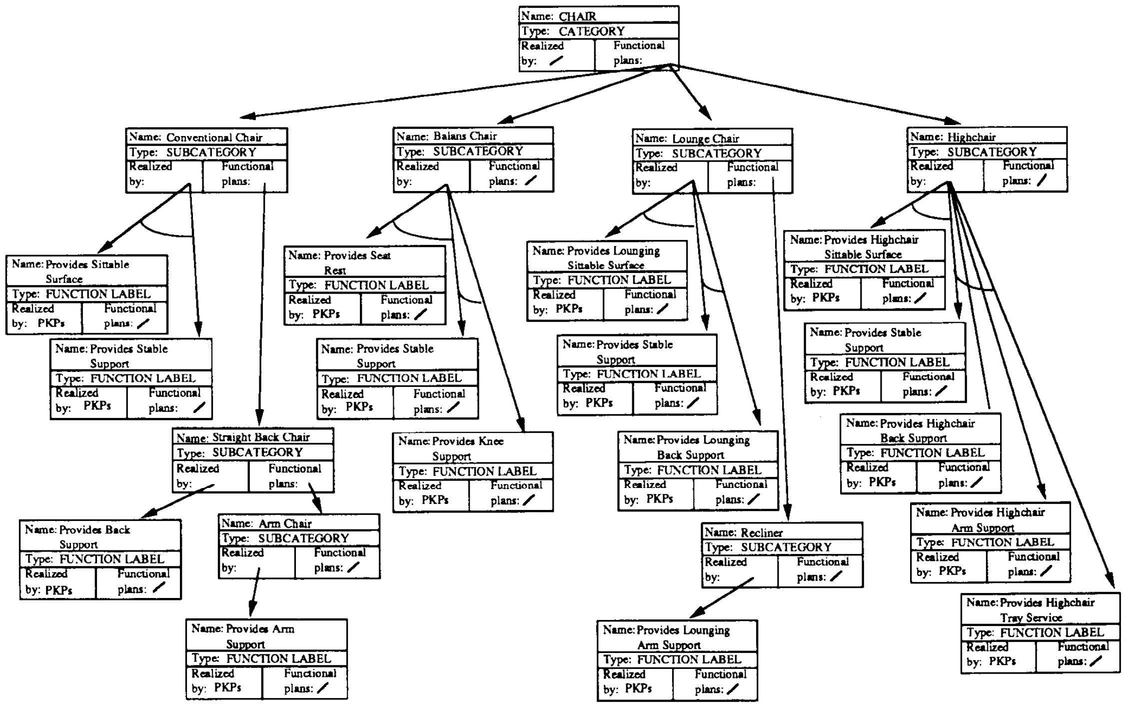

Fig. 2. Category representation graph.

different function labels are combined to determine the cumulative association measure for the (sub)category (see [12] for more details).

The category representation graph is the control structure for input object analysis. As the graph is traversed in depth-first fashion, if the (sub)category node has associated functional requirements, then those requirements are evaluated. If it is found that the requirements can be met by some portion(s) of the structure within some threshold association measure, then the functional elements are formed into a list. When applicable, the proper orientation for the object is also saved in the list.

The subcategory nodes are constrained by the information acquired from the parent subcategory nodes. This restriction is called *structural constraint propagation*. Many functional elements have an implied association that will constrain their possible structure and position. For example, the functional element that acts as the back of a chair for the subcategory Straight Back Chair must be situated above and approximately perpendicular to the functional element, found at the Conventional Chair level, which acts as the sittable surface.

If more than one function label is associated with a single (sub)category node, then the function label nodes are evaluated in a left to right manner. Therefore, referencing the functional requirements defined for the Conventional Chair (Fig. 2), the function label *provides sittable surface* must be fulfilled before initiating the procedural knowledge associated with *provides stable support*. This implies that *structural constraint propagation* exists between sibling function labels as well as between subcategory function labels.

IV. Implementation

The system is implemented in C on a Sun workstation. Over 100 test objects, defined by a number of different individuals, have been analyzed. Each object definition is composed of a face file and a vertex file.[1] The recognition system reads each of these files along with the category definition file. This file holds the information in a format that can be read to construct the category representation graph.

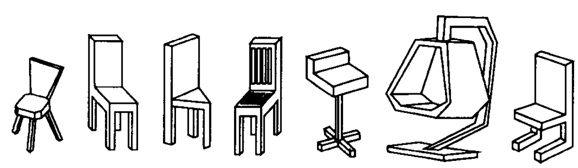

Fig. 3. Example objects recognized as straight back chairs.

The extent of how "generic" the function-based representation scheme actually is can best be seen in a sample of the objects that the system was capable of correctly categorizing. All of the objects appearing in Fig. 3 (along with many others) were categorized as straight back chairs.

Each fulfills the functional requirements of *provide sittable surface*, *provide stable support* and *provide back support* in its own way. In order to gain a better understanding of the reasoning process, a trace of the analysis of a simple example is now given. Fig. 4 depicts the input of an Arm Chair and the labeled output produced by the system.

The ground plane is considered to be parallel to the X-Y plane. It is also assumed that gravity acts in the $-Z$ direction. As seen in Fig. 4, input objects do not have to be in "upright" orientation. The system's first step is to evaluate the shape of the input object. This consists of enumerating the surfaces and modules that can act as functional

[1] The collection of object descriptions used is available to interested researchers through anonymous ftp on figment.csee.usf.edu under $pub/errors\_stuff/Objects$.

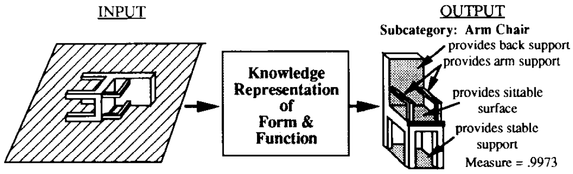

Fig. 4. Example input and output of system evaluation.

elements. Individual surfaces are listed, along with all surfaces that can be formed by grouping *essentially coplanar* surfaces. The object is further evaluated by subdivision into a set of convex 3-D modules, which are found directly from the object geometry. The center of mass of the whole object is calculated, along with the area of each of the surface functional elements.

A. Evaluation of 3-D Shape

Evaluation begins with the category associated with the root node. Since there are no function labels associated with the node *Chair*, processing passes to the first *Subcategory Conventional Chair*. The list of PKP's invoked to realize the first function label *provides sittable surface* is shown in Fig. 5(a). The *dimensions* PKP finds all functional elements of the input object that are of the proper size range to be a sittable surface. This ensures that the "seat" of the chair is large enough to support the seat of a normal person and not so large that it could be a couch or table top. The surface, or group of surfaces, must also provide the proper amount of contiguous surface area. Surfaces that survive this test include what we would think of as the back of the chair, the seat of the chair, and the bottom of the chair.

The list of potential sittable surfaces found in the first procedure is passed to the next PKP *relative orientation*. This procedure attempts to confirm that the potential sittable surface is essentially parallel to the ground plane. If it is not, a transformation that will orient the potential sittable surface parallel to the ground plane is calculated and stored with the surface.

The next PKP uses information from the prior PKP's to test whether each potential sittable surface, when positioned parallel to the ground, can be within the proper height range. The potential sittable surface has been transformed such that the normal of the surface is aligned in the $+Z$ direction. The dimensions test finds the greatest distance spanned by the object in the $-Z$ direction. This gives a tentative height for the potential sittable surface. The back is eliminated in this test because there is no structure that can support the back in the proper height range. Two surfaces remain as potential sittable surfaces: the seat and the bottom of the chair.

The tests performed to this point are computationally simple tests that are used to prune the list of possible functional elements. The next two tests ensure that the surviving surfaces are clear and accessible for use.

A list of possible seat surfaces has now been identified (see Fig. 5(b)). If the list were empty, then it would be decided at this point that the object in question is not a conventional chair. An association measure of zero would be returned, and processing would continue with the next subcategory node Balans Chair. The association measure for each functional element found to this point is a function of the area and the potential height. Since the list is not empty, a list of potential sittable surfaces has been accumulated. This completes the tests associated with the procedural knowledge of *provides sittable surface*. The list of potential sittable surfaces is passed to the next function label node.

The second function to confirm is that the object has a base structure that *provides stable support*. The only PKP associated to this function label is *stability*. The procedure tests each potential result in its specified orientation. The object must be able to be placed in a stable position and still maintain the sittable surface in its proper orientation. To test for stability, each potential sittable surface is oriented in the X-Y plane with the surface normal in the $+Z$ direction. The maximum $-Z$ displacement is found, and all vertices at this level are accumulated. These are potential points of contact with the ground to give support to the object. One of three conditions must exist: 1) Only a single point is in contact; 2) multiple collinear points are in contact; 3) at least three noncollinear points are in contact. In order to have sufficient contact, there must be at least three noncollinear points. Hence, if one of the first two conditions is found, then the object must be rotated such that at least three noncollinear points are in contact. This can lead to multiple possible new orientations to test. For each possible orientation, a list of contact points is accumulated. The convex hull of these points is then calculated to be used in the test for stability. It is assumed that the object has homogeneous density. Therefore, the force exerted downward can be represented with a single vector from the center of mass of the object pointing in the $-Z$ direction. If the force vector projects into the ground plane within the convex hull of the contact points, then the object is "self-stable." It is only considered "self-stable" because a force applied by the weight of a person does not have to be exerted directly over the center of mass of the object. This force can be applied in different positions downward on the sittable surface and tested to make sure that each resultant force (object plus applied weight) projects inside the convex hull.

Evidence is accumulated at the Conventional Chair node in support of the current hypothesis. The only surviving surface is, in fact, the seat of the chair (Fig. 5(d)). Face #20 (the bottom of the seat) was eliminated because stable support could not be verified.

The parsing of the object continues by checking the Straight Back Chair's associated function label. The list of PKP's used to confirm *provides back support* is given in Fig. 5(e). Each surface or group of surfaces that is *essentially orthogonal* to the potential sittable surface is tested. The *proximity* test checks to make sure the surface is close to and centered relative to the sittable surface. Clearance is also tested for the proposed back support relative to the potential sittable surface. There is only one surviving orientation at this point that provides all specified functions (Fig. 5(f)). This result is passed to the Arm Chair subcategory.

The list of PKP's used to realize *provides arm support* is depicted in Fig. 5(g). For a surface to act as an arm support, it must be oriented essentially parallel to the sittable surface. The arm support surfaces must be close and at the sides of the sittable surface. The surface must also be clear above for accessibility. One pair is found: one surface on each side of the sittable surface. These functional elements are labeled, and a new association measure is calculated.

Since there are no subcategories left in this subgraph, processing continues at the subcategory node Balans Chair. An association measure of zero is returned because the functional requirements of *provides seat rest* and *provides knee support* cannot be fulfilled by the structure of the arm chair. Association measures of zero are also found for the subcategory Lounge Chair and the subcategory Highchair, though for different reasons.

B. Experimental Results

Each of the 101 input objects was designated as either CHAIR or NONCHAIR (see Figs. 6 and 7), based on the intuitive feelings of the designer. The objective was to compare the system's categorization to the intuitive categorization assigned by the designers. Table I summarizes the number of objects evaluated, the number categorized as CHAIR/NON-CHAIR by the designer, and corresponding numbers

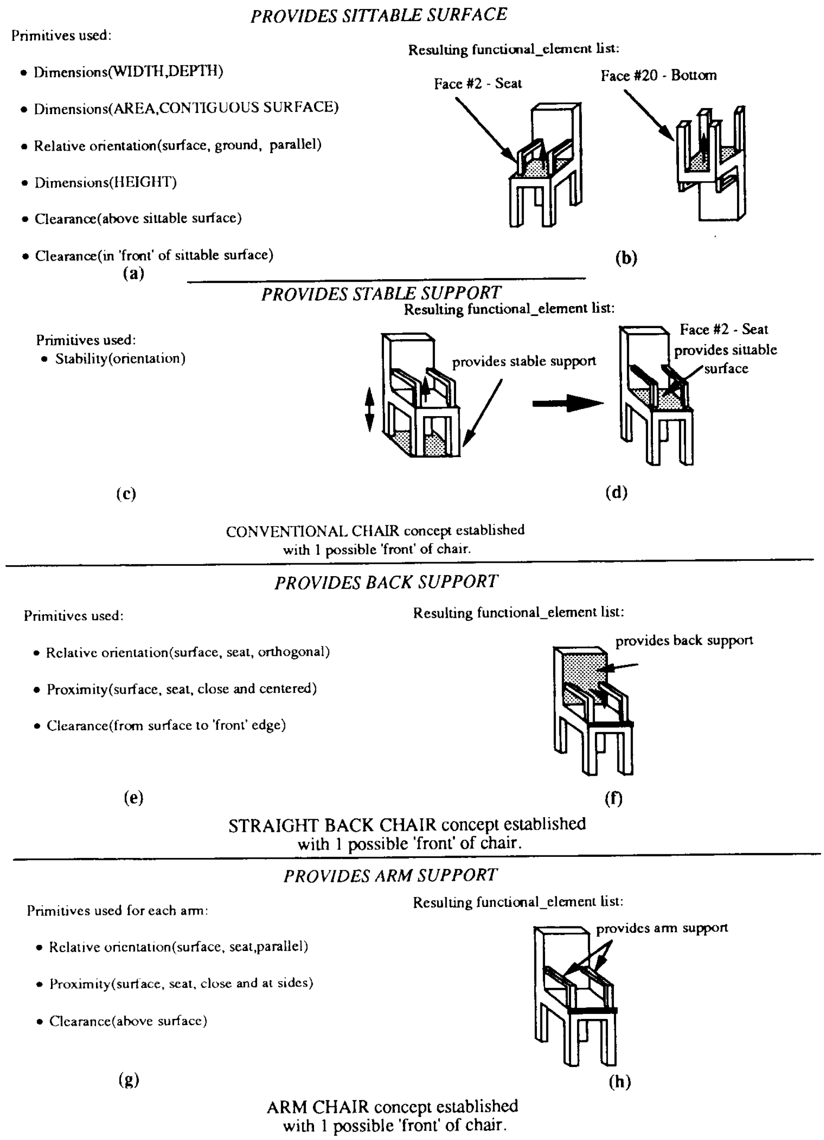

Fig. 5. Evaluation process of arm chair.

for the system. There is only one input object intuitively categorized by its designer as a chair but not recognized as such by the system. This object (see Fig. 8(a)) was not categorized as a chair due to the fact that the system could not identify a contiguous sittable surface within the proper width/depth size range. The greatest discrepancy occurred with intuitively NONCHAIR objects that the system evaluated as being capable of functioning as a chair. Fig. 8(b) depicts all objects that were counter-intuitively identified by the system as falling into the Straight Back Chair subcategory. All of these objects have in common that they have some orientation in which they can provide a sittable surface, provide stable support, and provide a back support. They can all, therefore, *function* as Straight Back Chairs. Fig. 8(c) depicts the set of objects found to be capable of functioning as a Conventional Chair (i.e., provides sittable surface and provides stable support). One example of this is the ***trash can*** (object #2) in Fig. 8(c). By turning the trash can over, a person could use the bottom as a sittable surface.

V. Future Research Directions

There are three areas we would like to investigate for extensions to the present system. First, the definition of more categories can be added to the knowledge base. We are completing the expansion of the system to include a number of basic level categories in the super-ordinate category "furniture." We also plan to add category representation from a different super-ordinate category, perhaps "dishes." This will allow us to test our assumption that the number

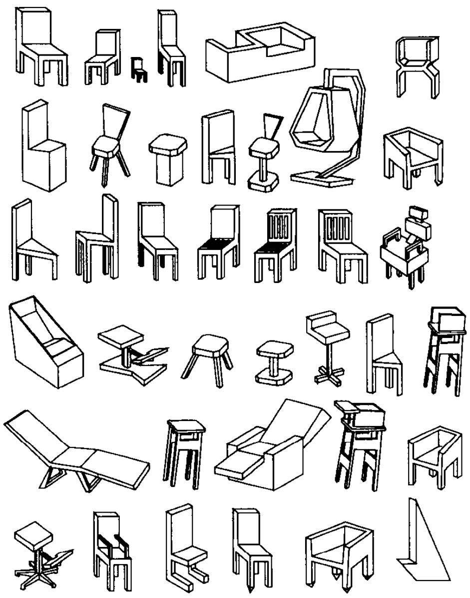

Fig. 6. Intuitive chair objects.

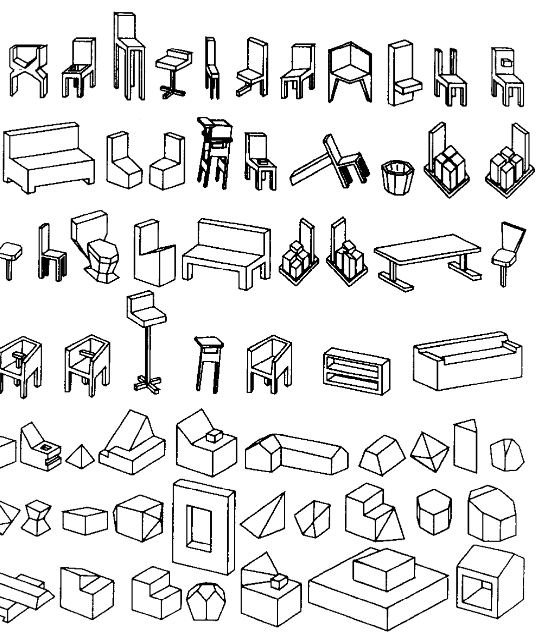

Fig. 7. Intuitive nonchair objects.

of PKP's required grows very slowly with the number of categories. This will also allow us to investigate the formation of heuristics in hypothesizing the categories to use in the evaluation process. The structural information attained during the enumeration of functional elements could provide cues for the choice of hypothesis.

Second, we plan to investigate using nonideal input. Currently, the input objects examined by the system are "ideal" in that they are the

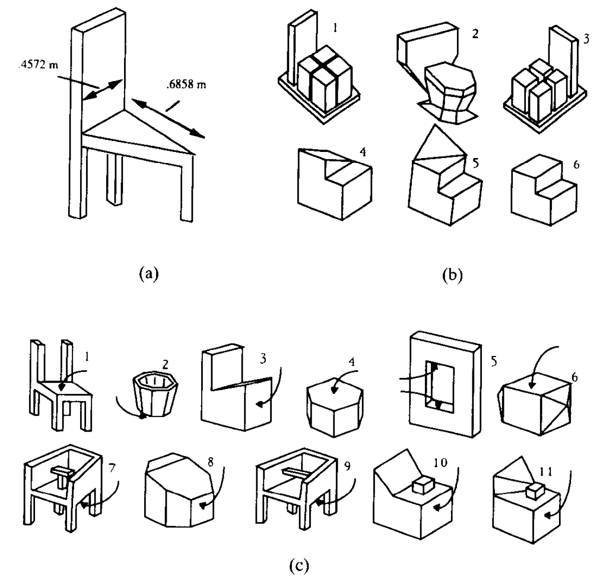

Fig. 8. Counter-intuitive chair results.

TABLE I
EXPERIMENTAL RESULTS

Total # of objects	101
# Intuitively Categorized as Chairs	38
# categorized correctly as chairs by system	37
# Intuitively Categorized as Non-chairs	63
# categorized correctly as non-chairs by system	46

output of a CAD tool. We hope to investigate the use of two forms of nonideal input. First, we want to explore the use of complete 3-D models constructed from multiple real images of an object. Second, we want to explore the use of incomplete 3-D models, as might be obtained from a single image and/or occluded views.

Third, we plan to investigate learning capabilities of the system. Through an interactive process, the system could question the user as to whether the structural differences found between objects categorized by the system have any functional significance. According to the user's response, new subcategories could be formed, and the control structure could be reorganized in such a way as to reflect the new functional plan. In this way, the system could learn by its experience.

REFERENCES

[1] M. Brady, P. E. Agre, D. J. Braunegg, and J. H. Connell, "The mechanics mate," in *Advances in Artificial Intelligence* (T. O'Shea, Ed.). New York: Elsevier, 1985, pp. 79–94.

[2] R. A. Brooks, and T. A. Binford, "Representing and reasoning about partially specified scenes," in *Proc. DARPA Image Understanding Workshop*, 1980, pp. 95–103.

[3] J. H. Connell and M. Brady, "Generating and generalizing models of visual objects," *Artificial Intell.* vol. 31, pp. 159–183, 1987.

[4] M. Di Manzo, E. Trucco, F. Giunchiglia, F. Ricci, "FUR: Understanding FUnctional Reasoning," *Int J. Intelligent Syst.*, vol. 4, pp. 431–457, 1989

[5] Eastman Kodak Co., *Ergonomic Design for People at Work.* New York: Van Nostrand Reinhold, 1983, vol. 1,

[6] R. R. Goldberg and D. G. Lowe, "Verification of 3-D parametric models in 2-D image data," in *Proc. IEEE Workshop Comput. Vision* (Miami, FL), 1987, pp. 255–257.

[7] W. E. L. Grimson, "On the recognition of parameterized 2D objects," *Int. J. Comput. Vision*, vol. 3, pp. 353–372, 1988.

[8] S. Ho, "Representing and using functional definitions for visual recognition," Ph. D. Dissertation, Univ. Wisconsin, Madison, 1987.

[9] M. Minsky, *The Society of Mind*. New York: Simon and Shuster, 1985.

[10] E. Rosch, C. B. Mervis, W. D. Gray, D. Johnson, P. Boyes-Braem, "Basic objects in natural categories," *Cognitive Psych.*, vol. 8, pp. 382–439, 1976.

[11] L. Stark, and K. W. Bowyer, "Achieving generalized object recognition through reasoning about association of function to structure," *AAAI-90 Workshop Qualitative Vision* (Boston, MA), July 1990, pp. 137–141.

[12] L. Stark, L. O. Hall, and K. W. Bowyer, "An investigation of methods of combining functional evidence for 3-D object recognition," in *Proc. SPIE 1381: Intell. Robots Comput. Vision IX: Algorithms Techn.* (Boston, MA), 1990.

[13] P. H. Winston, T. O. Binford, B. Katz, and M. Lowry, "Learning physical description from functional definitions, examples, and precedents," in *Proc. Int. Symp. Robotics Research* (M. Brady and R. Paul, Eds.). Cambridge, MA: MIT Press, 1984, vol. 1.

[14] P. H. Winston, "Learning structural descriptions from examples," in *The Psychology of Computer Vision*, P. H. Winston (Ed.). New York: McGraw-Hill, 1975.