COMPUTER VISION AND IMAGE UNDERSTANDING
Vol. 62, No. 2, September, pp. 164–176, 1995

Recognition by Functional Parts

EHUD RIVLIN

Department of Computer Science, Technion—Israel Institute of Technology, Haifa, Israel

SVEN J. DICKINSON

*Rutgers Center for Cognitive Science (RuCCS), and Department of Computer Science, Rutgers University,
P.O. Box 1179, Piscataway, New Jersey 08855*

AND

AZRIEL ROSENFELD*

Center for Automation Research, University of Maryland, College Park, Maryland 20742

Received August 1, 1994; accepted May 12, 1995

We present an approach to function-based object recognition that reasons about the functionality of an object's intuitive parts. We extend the popular "recognition by parts" shape recognition framework to support "recognition by functional parts," by combining a set of functional primitives and their relations with a set of abstract volumetric shape primitives and their relations. Previous approaches have relied on more global object features, often ignoring the problem of object segmentation and thereby restricting themselves to range images of unoccluded scenes. We show how these shape primitives and relations can be easily recovered from superquadric ellipsoids which, in turn, can be recovered from either range or intensity images of occluded scenes. Furthermore, the proposed framework supports both unexpected (bottom-up) object recognition and expected (top-down) object recognition. We demonstrate the approach on a simple domain by recognizing a restricted class of hand-tools from 2-D images. © 1995 Academic Press, Inc.

1. INTRODUCTION

The problem of object recognition from sensory data is defined in the literature as the association of visual input with a name or symbol. In the absence of distinguishing properties such as color, texture, or motion, object recognition first requires the visual recovery of shape, followed by the matching of the recovered shape to a database of known objects [18]. Although much research on the topic has been published, the community still lacks vision systems that can recognize in real time a large number of objects (natural or man-made). Full recovery has been difficult to achieve while matching suffers from combinatorial explosion.

Model-based recognition has been suggested as a remedy to these problems. Many such 3-D object recognition systems take a single object model and attempt to locate it in the image, e.g., [15, 16, 35]. Furthermore, the object models are commonly CAD-like, capturing the *exact* geometry of the object. Although very effective for certain robot vision tasks in constrained environments, where a known target must be accurately localized for manipulation or inspection, these techniques are inadequate when addressing less constrained environments like a robot vision system moving about a factory or house.

Consider, for example, a robot vision system whose goal is to move through a handicapped person's household, retrieving and manipulating everyday objects such as books, cups, and chairs. How can we avoid having to provide the system with detailed CAD specifications of each object that the system is to recognize? One way of making object models more flexible is to parameterize geometric models, as proposed by Brooks in his ACRONYM system [4]. For example, the legs of a chair model could have lengths that fall in some specified range, or the number of chair legs could be variable. Object recognition systems using parameterized models have also been proposed by Huttenlocher [14] and by Lowe [17]. However, all three of the above systems are very top-down, requiring not only knowledge of what object is in the image, but in some cases a good initial guess as to the orientation of the object.

A different approach to the problem is to consider the recognition process in the context of an agent interacting

* The third author acknowledges the support of the U.S. Airforce Office of Scientific Research under grant F49620-93-1-0039.

1077-3142/95 $12.00

with its environment [24]. The recognition process is subordinate to the agent's intentions and behavior in its environment. Recognition is equivalent to the process that checks if an object suits a particular purpose. If an object is perceived to fulfill a function necessary to carry out a certain behavior or action, then it is recognized. Gibson's theory of affordances [12], i.e., properties that are defined with reference to an observer, was a major step in this direction. Winston *et al.* [37] emphasized how much easier it is to describe what objects are used for, rather than to describe what objects look like. They tried to show how recognition could be performed using functional definitions, and how physical models could be learned using functional definitions and specific acts of identification.

When we consider recognition from a functional point of view, we leave the concept of shape-alone-based recognition for a more general and flexible concept. For example, if we model four chairs, each having a different configuration of differently shaped parts but all functioning as chairs, we would require four different object shape models. Alternatively, recognition based on functionality would enable our agent to possess knowledge of the needed function of a chair without explicitly specifying the possible shapes of a chair. The seminal work of Stark and Bowyer *et al.* [26–32] has addressed function-based object recognition, focusing on domains including chairs and dishes. In their work, they define a set of functional primitives specific to each object class. For example, in their system that recognizes chairs, they have functional primitives for support, sitting height, stability, etc. From a CAD representation of an object, they can compute these primitives and categorize the object. Although their system has been tested mainly with CAD data, it has also been applied to complete range images of an object acquired through an Odetics range scanner.

Despite the success of this approach, it has some limitations. To begin with, the approach assumes a 3-D representation of the image from which the functional primitives can be computed. Furthermore, the approach assumes an image of an isolated object; object occlusion in the image cannot be supported since there is no object segmentation performed on the image data. It is important to note that the work of Stark and Bowyer takes a *global* approach to functional recognition, making it sensitive to occlusion and partial views. Due to the nature of their functional reasoning, it does not extend to function-based recognition from 2-D imagery containing multiple occluded objects.[1] It should be noted, however, that for certain tasks, such as the determination of object stability, such global representations are sufficient.

In this paper, we present a theory of function-based recognition which is a natural extension of part-based shape recognition. Instead of focusing on global properties such as stability, height, and existence of large horizontal surfaces, we will reason about the functionality of an object's parts. Moreover, those parts are the *same* parts that we recover from the image for shape recognition. Thus, instead of reasoning about the functionality of a collection of 3-D points or planar surfaces, we propose to reason about a more intuitive notion of an object's parts (Pentland [20]). Although we will not index using part shape, we can use knowledge of part shape to help segment the image into parts. Given a set of recovered volumetric parts, we can then reason both about the functionalities of individual parts *and* the interactions between the parts. Such interactions can include relative orientation, size, shape, or even motion!

Although the idea of reasoning about the function of an object's parts has been proposed by other researchers, there has been little concern about dealing with real image data. In Winston *et al.* [37], the vision component was replaced by a linguistic interface which provided English descriptions of scene content. In Vaina and Jaulent's compatibility model [36], shape attributes of an object, e.g., length and relative part orientation, were provided as input; neither a shape description nor a recovery scheme was presented. In Brady *et al.*'s Mechanics Mate [3], a mapping from curvature primal sketch (CPS) and smoothed local symmetries (SLS) features in a 2-D image to a set of higher order geometrical structures was proposed. These higher order structures were then mapped to a set of functional parts belonging to a set of handtools. The extension to 3-D shape was proposed but never implemented.

Our goal in this work is not only to propose an object representation which integrates function and shape, but addresses the problem of recovering shape and function from either 2-D or 3-D image data. We will outline an approach which first segments an image containing multiple objects into a set of volumetric parts, supporting part recovery from incomplete views of the object and supporting object occlusion. Following part grouping by object, the approach will infer the possible functionalities of individual parts and collections of parts. The robot can check if the needed functionality for a certain action is consistent with the recovered functionality. Comparing this approach to that of Stark and Bowyer for the problem of searching the image for a "chair kind of support," we would like to reason about a set of chair legs, a seat, and a back, rather than a set of simple planar surfaces or 3-D points.

The paper is organized as follows. First, we outline our theory of functionality for objects in Section 2 and introduce a representation for volumetric parts from which we reason about functionality. Section 3 discusses the recovery of the volumetric parts from both 3-D range and 2-D inten-

[1] In recent work, Stark and Bowyer [29] reason about the functionality of objects that are partially visible in range images.

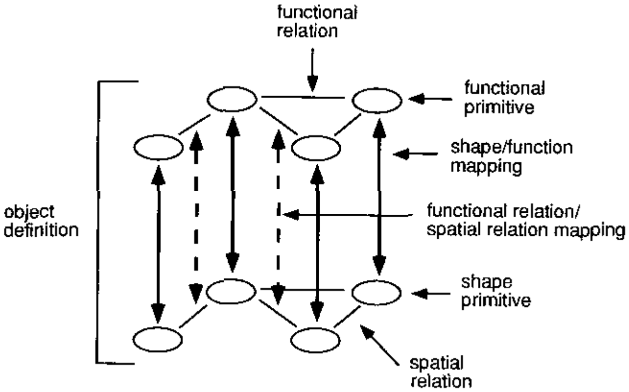

FIG. 1. Representing object functionality.

sity images. Next, in Section 4, we describe our recognition algorithm as it applies to both expected (top-down) and unexpected (bottom-up) object recognition. Finally, in Section 5, we demonstrate the technique applied to the domain of hand tools.

2. REPRESENTING OBJECT FUNCTIONALITY

Our theory of function-based object recognition is a natural extension of part-based shape recognition. That is, we reason about the functionality of an object's parts and their interrelations. Figure 1 illustrates the concept. At the shape level, objects are constructions of coarse volumetric primitives with spatial relations between the primitives. At the function level, the shape primitives map to a set of functional primitives and the spatial relations map to a set of functional relations. At the functional level, objects are not represented in terms of shape, but in terms of a set of functional primitives and relations. In the following sections, we describe this hierarchical representation in more detail. We begin by describing the coarse shape representation and follow with the functional representation. Finally, we illustrate the representation by means of an example.

2.1. *Representing Shape*

2.1.1. *Shape Primitives.* Our shape representation models objects as constructions of coarse volumetric shapes belonging to four classes: sticks, strips, plates, and blobs. The representation is an extension to the generalized blob models (sticks, plates, and blobs) proposed by Mulgaonkar *et al.* [19]. Our four classes are distinguished by their relative dimensions. Letting a_1, a_2, and a_3 represent length, width, and height, respectively, of a volumetric part, we can define the four classes as

$$\text{Stick: } a_1 \simeq a_2 \ll a_3 \vee a_1 \simeq a_3 \ll a_2 \vee a_2 \simeq a_3 \ll a_2 \quad (1)$$

$$\text{Strip: } a_1 \neq a_2 \wedge a_2 \neq a_3 \wedge a_1 \neq a_3 \quad (2)$$

$$\text{Plate: } a_1 \simeq a_2 \gg a_3 \vee a_1 \simeq a_3 \gg a_2 \vee a_2 \simeq a_3 \gg a_2 \quad (3)$$

$$\text{Blob: } a_1 \simeq a_2 \simeq a_3. \quad (4)$$

Intuitively, if all three dimensions are about the same, we have a blob. If two are about the same and the third is very different, we have two cases: if the two are bigger than the third, we have a plate, while if the two are smaller than the third, we have a stick. Finally, when no two dimensions are about the same, we have a strip. For example, a knife blade is a strip, because no two of its dimensions are similar.

Figure 2 shows examples of these four types of primitives. The ratios of the three dimensions for the stick are 1:1:10; for the strip, 48:1:7; for the plate, 16:16:1; and for the blob, 1:1:1. Note that our taxonomy of primitives is

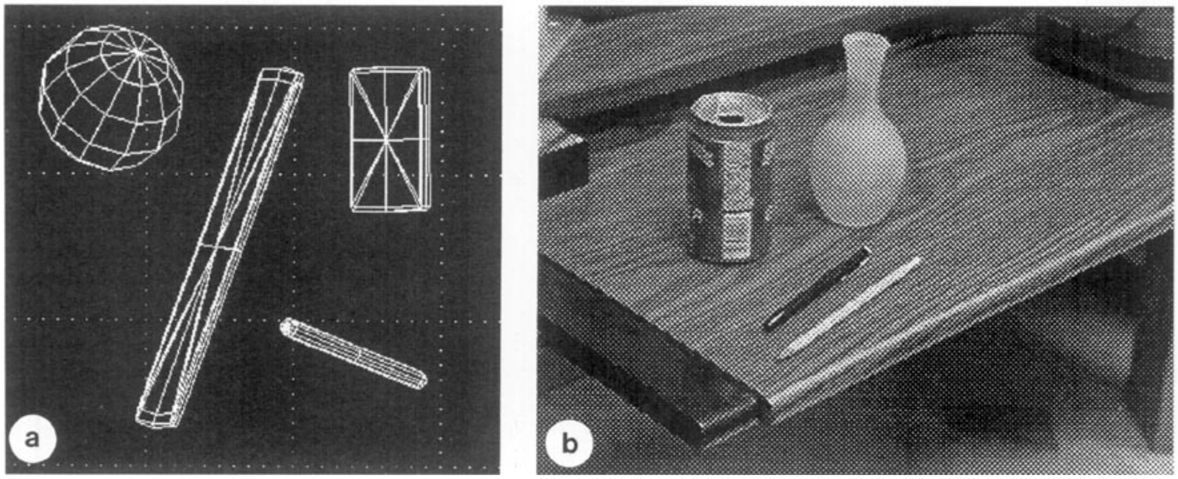

FIG. 2. Shape primitives: (a) The four primitives modeled by superquadrics using Pentland's Thingworld [20]. (b) Two blobs (can and vase), two sticks (pencil and pen), and a plate (table).

qualitative; a cylinder, for example, can be a stick or a plate. Variations of the basic primitives, such as tapering, pinch, and curvature, can also be considered in the context of functionality. It should also be noted that the ratio of shape dimensions used to separate the four shape classes is specified as a function of the object definition.

2.1.2. *Spatial Relations.* We can qualitatively describe the ways in which two shape primitives can be combined. For example, we can attach two shapes end to end, end to side, or side to side, as proposed by Biederman when building objects out of geons [2]. To further specify these attachments, we adopt the convention of labeling each volumetric primitive's attachment surfaces [9]. For example, a square plate has six attachment surfaces, while a cylindrical stick has three attachment surfaces. For simplicity, we shall require any junction of two primitives to involve exactly one attachment surface from each primitive. Figure 3 illustrates a simple object composed of a blob and a stick; note that in this example, both objects are cylinders.

In addition to specifying the two attachment surfaces participating in the junction of two primitives, we can also consider the angles at which they join. Furthermore, we can classify the joints as perpendicular, oblique, tangential, etc. Another refinement would be to qualitatively describe the position of the joint on each surface. For example, an attachment can be near the middle, near the side, near the corner, or near the end of a surface. Figure 4 illustrates six different concatenations of a strip and a stick. A stick has three attachment surfaces, but when symmetry is taken into account it has only two distinguishable attachment surfaces. Similarly, a strip has three distinguishable attachment surfaces. Therefore, there are six ways of combining the two primitives in terms of attachment surfaces if we ignore where on a surface an attachment is made as well as the angle of the attachment.

Finally, the spatial relations must encode the relative sizes of the spatial primitives. For example, although a stick attached to a blob in a certain manner might constitute a hammer, the blob's size cannot be an order of magnitude larger than the stick's. Consequently, for each spatial relation between two spatial primitives, we specify a ratio of particular features. For a blob, any dimension is chosen; for a stick, its length is chosen; for a plate, its surface width (or length) dimension is chosen; and for a strip, its length is chosen. Thus, for example, a relative size specification between a stick and a blob of 4:1 would indicate that the length of the stick is 4 times the dimension of the blob. Tolerances would, of course, need to be specified.

2.2. *Representing Function*

2.2.1. *Functional Primitives.* Functional primitives represent the building blocks of a functional representation of an object. For example, the functional primitives defining a coffee cup would include a handle and a container; a chair would include a seat, a base, and a back [26, 27]. In the remainder of this paper, we will illustrate our approach to functional object recognition by focusing on a class of manipulation tasks. Bearing in mind that a manipulation task involves an agent and its environment, we will define a class of objects that have an *end-effector* (that part which delivers the action) and a *handle* (that part which provides the interface between the agent and the end-effector). Examples of such objects might include simple hand tools like a screwdriver or a hammer, or other handled objects such as a cup, a frying pan, or a briefcase.

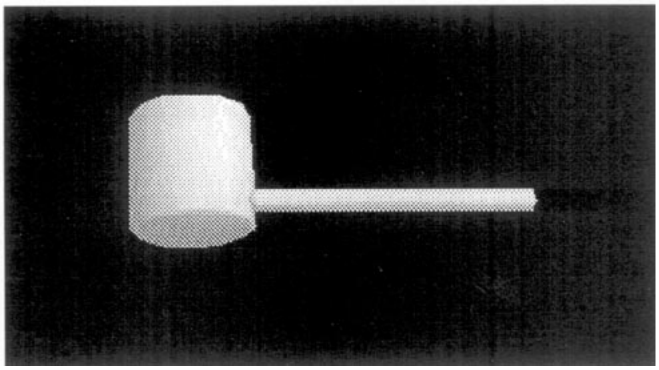

FIG. 3. Concatenation: a blob and a stick.

2.2.2. *Functional Relations.* A given set of parts might independently satisfy the needs for an end-effector or a handle. However, they must be joined in a particular way so as to satisfy the needs of a particular task. The set of functional relations linking the functional primitives describes the function of the interaction between the functional primitives. Consider, for example, the functional model for a hammer specifying an end-effector and a handle. The functional relation linking the handle and end-effector specifies that the handle is used to swing the end-effector in a direction which maximizes the force tangential to the swing arc while maximizing striking stability.

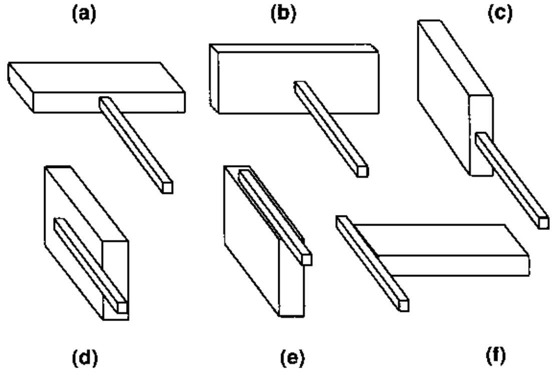

FIG. 4. Six different objects composed of a strip and a stick.

2.3. *The Relationship between Shape and Function*

2.3.1. *Mapping Functional Primitives to Shape Primitives.* In general, the mapping between shape primitives (and their relations) and a particular functional primitive is many-to-one. For example, one, two, three, or more chair legs may satisfy the functional primitive of chair base, provided they satisfy constraints for height, stability, and support.[2] Furthermore, a given set of recovered shapes may satisfy multiple functional primitives. For example, a large flat surface may serve as a supporting surface or it may serve as a covering surface (roof). For simplicity, we will restrict ourselves to object models with a one-to-one mapping between shape primitives and functional primitives.

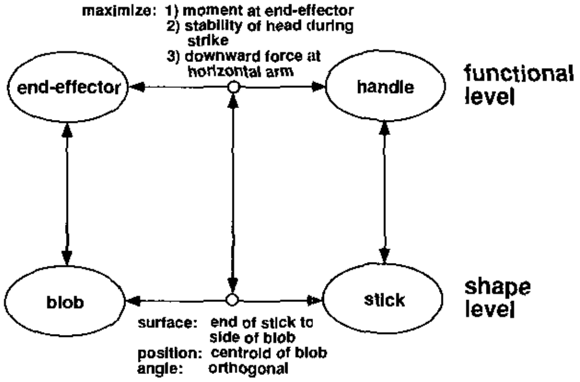
maximize: 1) moment at end-effector
2) stability of head during strike
3) downward force at horizontal arm
end-effector
handle
functional level
blob
stick
shape level
surface: end of stick to side of blob
position: centroid of blob
angle: orthogonal

FIG. 5. Functional model for a hammer.

Returning to the hammer example, the end-effector should be blob-like, ensuring that the dimensions of the striking surface are roughly equal (rotationally symmetric to allow striking error in any direction).[3] If the end-effector were stick-like, the distance between the handle junction and the striking surface would be large, making it more difficult to locate the nail. If the end-effector were plate-like, it would have insufficient momentum for driving a nail. The handle, on the other hand, should be stick-like, small enough so that it can be grasped by a human hand, and long enough to provide a high moment at its junction with the end-effector.

It is important to note that in choosing a simple domain which affords a one-to-one mapping between shape and function, we do not demonstrate the potential complexity of the mapping between shape and function. This complexity can take two forms. The first is the complexity in mapping between shape and shape, i.e., grouping shape primitives to form larger shape primitives. For example, consider a patio chair with a seat made out of multiple slats. The mapping between a flat surface and a chair seat is still one to one, but the mapping between physically distinct parts (the slats) and the flat surface is many to one. Such a grouping of parts on the basis of, for example, coplanarity, collinearity, or cocurvilinearity, must be addressed and is beyond the scope of this paper.

The second type of mapping complexity that we do not address is the mapping between multiple shape primitives and a single functional primitive. Returning to our chair leg example, a set of multiple legs on a chair may differ enough in shape and orientation to defy their being grouped into a single shape primitive. However, the collection serves the single function of chair base. By restricting our discussion to a one-to-one mapping between shape primitives and functional primitives, we focus only on the concept of integrating both functional and shape descriptions. In future work, we hope to explore this mapping beyond the simple one-to-one mapping to include both the grouping of multiple shape primitives into single shape primitives, and the mapping of multiple shape primitives to a single functional primitive.

2.3.2. *Mapping Functional Relations to Spatial Relations.* The specification as to how the functional components defining an object are combined is captured by a set of functional relations. These functional relations are then mapped to a set of spatial relations linking the shape primitives. In the hammer example, the functional relation maps to an attachment between the stick (handle) and the blob (end-effector) such that the axis of the stick is orthogonal to the (principal) axis of the blob and is attached to the centroid of the blob. The complete model for the hammer, including functional and shape primitives, functional and shape relations, and the mapping from functional shapes and relations to spatial shapes and relations is outlined in Fig. 5.

3. RECOVERING SHAPE

In the last section, we described a set of functional primitives defined on a set of shapes consisting of sticks, strips, plates, and blobs. Since these four shape classes are defined according to their relative dimensions, we need to not only segment an input image into parts, but recover 3-D (dimensional) information from those parts. In this section, we describe an approach to recovering sticks, strips, plates,

[2] A simple chair could be constructed by attaching a back support to a box. In this case, the chair could be thought of as having a single chair leg consisting of the box (minus the top which serves as the seat). The most general specification of the base of a chair, as articulated by Stark and Bowyer [26], would specify only that the seat was supported at the proper height and that the structure was stable. The exact number, type, and orientation of the shape primitives making up the base need not be specified. However, one might argue that the collection of shapes comprising the base could be thought of as a single blob, thereby providing stability.

[3] The notion of a part being blob-like is invariant to minor tapering and bending deformations, so that hammers with tapered or bent heads would fall into this class.

and blobs from an image. The approach consists of recovering a superquadric from the image, providing explicit dimensions which we can then use to classify our shape. Superquadrics offer a compact, coarse, volumetric description of an object's parts [20]. If finer shape modeling is required, deformable superquadrics can be used to capture both global part shape (using a superquadric) and local shape (using a deformable mesh) [33]. Since superquadrics capture more shape attributes than just the x, y, and z dimensions of a part, they provide us with a foundation from which to recover a richer vocabulary of qualitative shapes with which to reason about function. For example, we may decide to distinguish among curved- and straight-axis shapes or between tapering and constant cross-sectional sweep rules [2].

Recently, several researchers have proposed various segmentation techniques to partition image or range data, in order to automate the process of fitting superquadric volumetric primitives to the data. Most of those approaches are applied to range data only [10, 11, 13, 25], while Pentland [21] describes a two-stage algorithm to fit superquadrics to image data. In the first stage, he segments the image using a filtering operation to produce a large set of potential object "parts," followed by a quadratic optimization procedure that searches among these part hypotheses to produce a maximum likelihood estimate of the image's part structure. In the second stage, he fits superquadrics to the segmented data using a least-squares algorithm. Pentland's approach is only applicable in case of occluding boundary data under simple orthographic projection, as is true of earlier work of Terzopolous *et al.* [34], Terzopolous and Metaxas [33], and Pentland and Sclaroff [22], which address only the problem of model fitting. In a related approach, Raja and Jain [23] recover geons from range imagery and use superquad fitting to determine the axis of the geon.

The approach we take, due to Dickinson and Metaxas [6], is to use a qualitative segmentation of a 2-D image to provide strong constraints on the deformable model fitting procedure described in [33]. In addition, the technique has been recently extended to deformable model recovery from range data [7]. The result is a technique which allows us to recover certain classes of superquadrics from image or range data, under orthographic, perspective, and stereo projection [6]. Furthermore, the technique supports the recovery of occluded parts, allowing us to reason about the functionality of objects that are only partially visible. We will not describe the above recovery methods in this paper; details can be found in [6, 7]. We will, however, describe the geometry of a deformable superquadric and show how we classify a superquadric as a stick, strip, plate, or blob.

3.1. *Geometry of a Deformable Superquadric*

Geometrically, the models that we can recover from either range or image data are closed surfaces in space whose intrinsic (material) coordinates are $\mathbf{u} = (u, v)$, defined on a domain Ω [33, 6]. The positions of points on the model relative to an inertial frame of reference Φ in space are given by a vector-valued, time varying function of $\mathbf{u}$

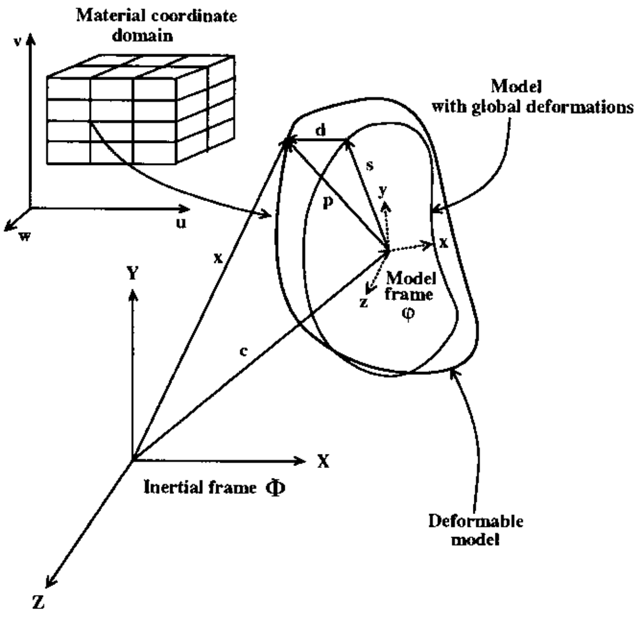

FIG. 6. Geometry of a deformable model.

$$\mathbf{x}(\mathbf{u}, t) = (x_1(\mathbf{u}, t), x_2(\mathbf{u}, t), x_3(\mathbf{u}, t))^{\mathrm{T}}, \tag{5}$$

where $^{\mathrm{T}}$ is the transpose operator. We set up a noninertial, model-centered reference frame ϕ, as shown in Fig. 6, and express these positions as

$$\mathbf{x} = \mathbf{c} + \mathbf{R}\mathbf{p}, \tag{6}$$

where $\mathbf{c}(t)$ is the origin of ϕ at the center of the model and the orientation of ϕ is given by the rotation matrix $\mathbf{R}(t)$. Thus, $\mathbf{p}(u, t)$ denotes the canonical positions of points on the model relative to the model frame. We further express $\mathbf{p}$ as the sum of a reference shape $\mathbf{s}(\mathrm{u}, t)$ and a displacement function $\mathbf{d}(\mathbf{u}, t)$:

$$\mathbf{p} = \mathbf{s} + \mathbf{d}. \tag{7}$$

The ensuing formulation can be carried out for any reference shape given as a parameterized function of $\mathbf{u}$. Based on the shapes we want to recover (sticks, strips, plates, and blobs with possible tapering and bending global deformations), we first consider the case of superquadric ellipsoids [1], which are given by the formula

$$\mathbf{e} = a\begin{pmatrix} a_1 C_u^{\varepsilon_1} C_v^{\varepsilon_2} \\ a_2 C_u^{\varepsilon_1} S_v^{\varepsilon_2} \\ a_3 S_u^{\varepsilon_1} \end{pmatrix}, \tag{8}$$

where $-\pi/2 \leq u \leq \pi/2$ and $-\pi \leq v\ \pi$, and where $S_w^\varepsilon = \operatorname{sgn}(\sin w)|\sin w|^\varepsilon$, and $C_w^\varepsilon = \operatorname{sgn}(\cos w)|\cos w|^\varepsilon$, respectively. Here, $a \geq 0$ is a scale parameter, $0 \leq a_1, a_2, a_3 \leq 1$ are aspect ratio parameters, and $\varepsilon_1, \varepsilon_2 \geq 0$ are "squareness" parameters.

We then combine linear tapering along principal axes 1 and 2, and bending along principal axis 3 of the superquadric $\mathbf{e}$[4] into a single parameterized deformation $\mathbf{T}$, and express the reference shape as

$$\mathbf{s} = \mathbf{T}(\mathbf{e}, t_1, t_2, b_1, b_2, b_3) = \begin{pmatrix} \left(\frac{t_1 e_3}{a a_3 w} + 1\right) e_1 + b_1 \cos\left(\frac{e_3 + b_2}{a a_3 w} \pi b_3\right) \\ \left(\frac{t_2 e_3}{a a_3 w} + 1\right) e_2 \\ e_3 \end{pmatrix}, \quad (9)$$

where $-1 \leq t_1, t_2 \leq 1$ are the tapering parameters in principal axes 1 and 2, respectively, and where b_1 defines the magnitude of the bending and can be positive or negative, $-1 \leq b_2 \leq 1$ defines the location on axis 3 where bending is applied and $0 < b_3 \leq 1$ defines the region of influence of bending. Our method for incorporating global deformations is not restricted to only tapering and bending deformations. Any other deformation that can be expressed as a continuous parameterized function can be incorporated as our global deformation in a similar way.

We collect the parameters in $\mathbf{s}$ into the parameter vector

$$\mathbf{q}_s = (a, a_1, a_2, a_3, \varepsilon_1, \varepsilon_2, t_1, t_2, b_1, b_2, b_3)^T. \quad (10)$$

Once we have recovered a superquadric from an image (range or intensity), it is a very simple matter to extract the dimensions of the superdradric. The width (x dimension) of the superquadric is given by

$$width = aa_1, \quad (11)$$

the height (y dimension) by

$$height = aa_2, \quad (12)$$

and the length (z dimension) by

$$length = aa_3. \quad (13)$$

Given the dimensions of the part, we can classify the part as either a stick, strip, plate, or blob according to the rules described in Section 2.

3.2. *An Alternative Approach to Recovering Qualitative Shape*

Before we leave the recovery section, it is worth noting that there is an approach to recovering sticks, strips, plates, and blobs directly from image data without first recovering superquads. The approach is based on the qualitative shape recovery work of Dickinson *et al.* [9] and relies only on a region segmentation of the input image. In that approach, a fixed set of volumes was analyzed over the viewing sphere, giving rise to a set of aspects. Shape recovery was therefore formulated as part-based aspect matching with a set of conditional probabilities associated with the aspects and their features used to guide both bottom-up inferencing of features and top-down prediction of features.

This approach could be extended by analyzing the views of a greater variety of parts sorted into the four shape classes. In this manner, image dimensions of image faces in a part aspect could be measured and, using a derived conditional probability distribution, used to suggest the class membership of the part. Although much less robust than inferring class membership from a superquadric fitted to the part due to the effects of foreshortening, the approach does avoid the process of fitting the superquad.

4. RECOVERING OBJECT FUNCTION

Our function-based object recognition strategy supports bottom-up (or unexpected) object recognition, whereby an object is presented to the system and the system identifies the object based on the functionalities of its parts. In addition, our strategy supports top-down (or expected) object recognition, whereby the system looks for a particular object in the image by mapping its functional parts to image feature predictions. In this section, we will describe both these strategies.

4.1. *Unexpected Object Recognition*

In an unexpected object recognition task, we first segment an input image into a set of homogeneous regions from which we recover a set of qualitative 3-D parts using local part-based aspect matching techniques [5, 8, 9]. Next, using the techniques of Dickinson and Metaxas [6], we use the recovered qualitative shape to constrain the fitting of a set of deformable superquadrics to the qualitative parts. From the resulting quantitative parts, we compare the dimensions of the parts to abstract a set of sticks, strips, plates, and blobs. Furthermore, we can recover the spatial relations spanning the recovered parts. For the experiments conducted in this paper, a spatial relation specification consists of an attachment specification and a relative orientation specification. A stick can be attached at its end or at its side; a strip at its flat surface, its short side, or its long side; a plate at its flat surface or its side; and a blob anywhere. Relative orientation is specified according to the angle between the z axes of the recovered primitives: orthogonal if the angle between the axes is less than 30° from perpendicular, and oblique otherwise.

If there is no a priori knowledge of what object is in the

[4] These coincide with the model frame axes x, y and z respectively.

image, then groups of spatial primitives and their spatial relations can be used to infer a set of functional primitives and relations. The recovered functional primitives and relations are then compared to a set of functional object models. In our simple domain of hand tools, we can map shape primitives to possible functional primitives and map shape relations to possible functional relations, providing a number of functional object hypotheses that are then compared to the object database. As an example, suppose we place a hammer in front of the camera and ask the system to identify the object. The recovery process would recover a stick and a blob in some spatial configuration. The blob then maps to end-effector as well as all other functions a blob could serve. Similary, the stick maps to a handle as well as all other functions that it could serve. Finally, the spatial relation between the stick and blob maps to all functional relations joining a stick and a blob in that configuration. Combining the various interpretations for the stick, the handle, and their relationship yields a number of object hypotheses which satisfy the recovered functionality.

4.2. *Expected Object Recognition*

In an expected object recognition task, we use knowledge of the target object's functional model to constrain our search in the image both in terms of what we look for and where we look for it. Given a functional object model, we first choose some functional primitive whose presence in the image would provide the least ambiguous mapping to the target object. For example, in looking for a cup on a table containing glasses and cups, we should look for a cup handle and not a container since the handle is unique to the cup. Next, the functional primitive is mapped to one of the four abstract shape primitives, i.e., sticks, strips, plates, and bobs. Finally, the shape primitive is mapped into an image region shape prediction in terms of extent or elongation. Like the unexpected object recognition algorithm, the image is first processed to extract a region topology graph. By examining the extent (or elongation) of an image region, along with that of its immediate neighbors, we can derive a simple heuristic for drawing attention to a particular image region. We can thus focus the recovery of the shape primitive and constrain the search for other primitives belonging to the object.

For example, if we are searching for blobs or plates, we can rank-order the image regions by increasing extent. Furthermore, regions whose immediate neighbors include a region with similar extent can be favored as being part of a blob, while regions whose neighbors do not include a region with similar extent can be favored as being part of a plate. Similarly, if searching for sticks or strips, we can rank-order the image regions by decreasing extent. Regions whose immediate neighbors include a region with similar extent can be favored as being part of a stick, while regions whose neighbors do not include a region with similar extent can be favored as being part of a strip. These rules can provide a useful ordering on the positions from which shape recovery is attempted.

From a candidate search position, the next step is to recover a superquadric from which the 3-D part dimensions and orientation can be recovered. This consists of first recovering the qualitative shape of the part [5, 8, 9], which is then used to constrain the fitting of a superquadric to the image data. Once the part is verified as a stick, strip, plate, or blob, the search for other object parts can be constrained to those image regions adjacent to or in the vicinity of any previously recovered volumes.

5. RESULTS

In this section, we apply the function-based expected object recognition algorithm to the image of the mallet shown in Fig. 7a. In Fig. 7b, we show the segmented region image. Without any a priori knowledge of scene content, each of the functional primitives, namely the end-effector and handle, are deemed equally likely to appear in the image. The algorithm arbitrarily chooses the end-effector (mallet head) and maps that to a search in the image for a blob. The algorithm rank-orders regions in the image according to the ratio of area to extent (computed from bounding box). The large region is chosen first and the bottom-up algorithm is used to recover the most likely interpretation of the region and its neighbors.

In Fig. 8a and 8b, we show the results of using the recovered qualitative shape to constrain the fitting of a superquad to each part; the parameters of the two superquads are given in Table 1. Since only a monocular image was used (a stereo pair was unavailable), depth or scale must be specified as a constraint; in this case, an arbitrary depth was chosen. When a stereo pair is available, true depth and scale can be recovered using the methods specified in [6]. When a stereo pair is unavailable, we can apply the simple heuristic that if two parts intersect in the image, they intersect in the world; in this case, since the qualitative parts intersect in the image, we infer that they belong to the same object.[5]

Finally, it should be noted in Fig. 8 that the fitted handle of the hammer does not quite extend along the full length of its corresponding recovered qualitative volume (stick). This is due to the fact that the fitting of the z dimension (or length) of the stick is a function of strong forces pulling from the end of the stick and weak forces pulling from the sides. Since the contour corresponding to the end of the stick (at its intersection with the blob) was not present,

[5] For a stronger heuristic that examines the nature of the parts' intersection in the image to determine intersection in the world, see [9].

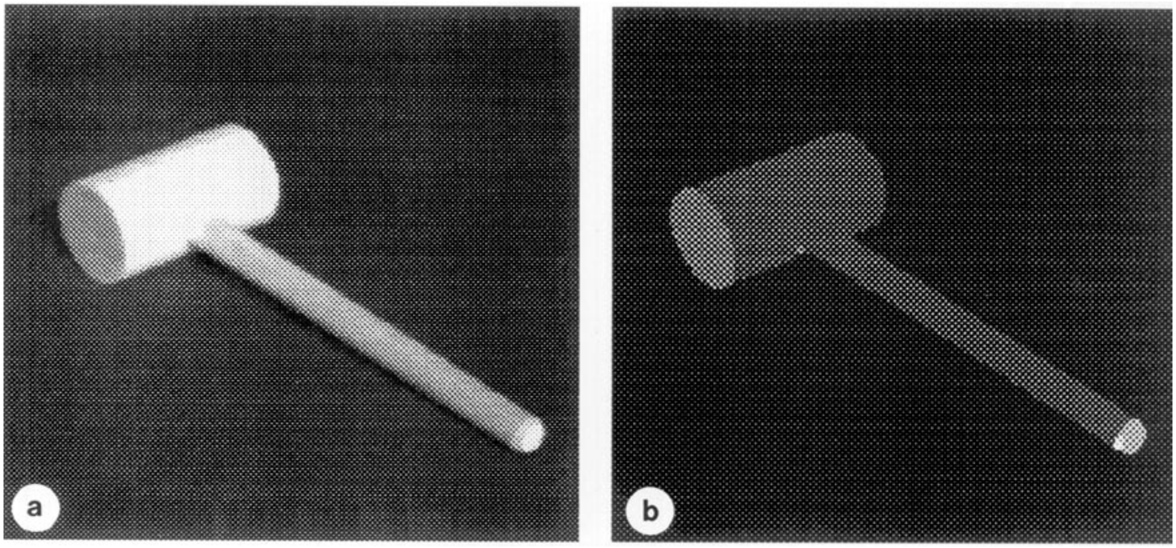

FIG. 7. (a) Original image. (b) Segmented region image.

the forces pulling the end of the fitted stick toward its intersection with the blob were weakened, resulting in the fitted stick stopping slightly short along its migration toward the intersection. A simple scaling of the forces exerted by the side of the stick could be used to extend the fitted stick a bit further, but the fitted stick will never extend beyond its longest side (contour) appearing in the image.

From the recovered superquad parameters in Table 1, we can proceed to classify each part as either a stick, a strip, a plate, or a blob according to Eqs. (11), (12), and (13) in Section 2.1.1; the results are shown in Table 2. Using Equations (1), (2), (3), and (4), defining two dimensions as similar if the ratio of the biggest to the smallest is within 4:1 (width:height:length ratios for the two parts are: 1:1:1.53 for the head and 1:1:5.18 for the handle), and specifying their relative size ratio (stick length to blob dimension) as lying between 1:1 and 4:1, the mallet head is classified as a blob, while the mallet handle is classified as a stick.

Since our search procedure is looking for the mallet head (end-effector), it chooses the blob and proceeds to search for the handle in the vicinity of the recovered blob. Due to region undersegmentation, the regions corresponding to the body surfaces of the head and handle of the mallet were joined. However, those contours not used to recover the head but still belonging to the large region are free to be part of other recovered volumes. Since we have already recovered a stick *and* its defining contours were not used to infer the blob, we can instantiate the handle

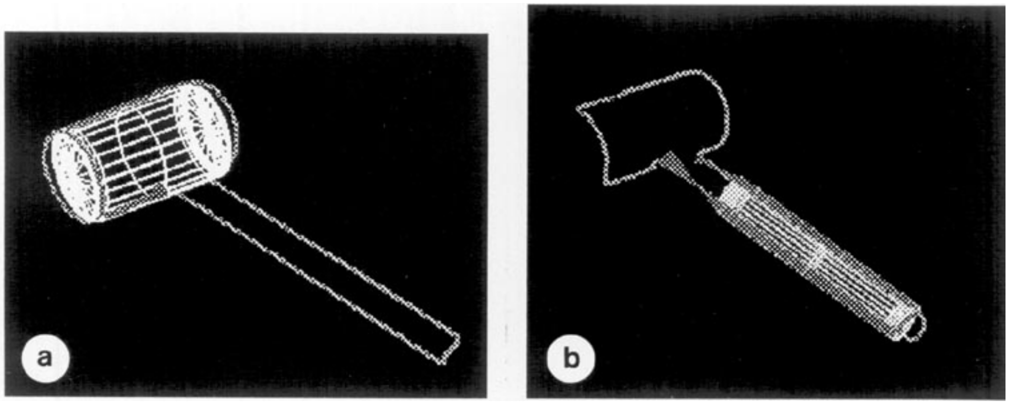

FIG. 8. Recovered superquads: (a) Mallet head, and (b) Mallet handle. Intermediate gray values along contour portions represent locations of image forces acting on the superquad. Note that due to the absence of forces at the junction between the two parts (no contours at the junction end of the handle), the fitted handle was not "pulled" all the way to the junction.

TABLE 1
Recovered Superquad Parameters for Mallet

Superquad parameter	Part	
	Head	Handle
a	37.19	37.19
a_1	0.45	0.22
a_2	0.45	0.22
a_3	0.69	1.14
t_x	−4.40	4.97
t_y	0.51	−3.88
t_z	−50.0	−50.0
r_{11}	0.49	0.54
r_{12}	−0.22	0.07
r_{13}	−0.84	0.84
r_{21}	−0.14	0.78
r_{22}	0.93	0.27
r_{23}	−0.33	−0.53
r_{31}	0.86	−0.26
r_{32}	0.28	0.96
r_{33}	0.42	0.09
ε_1	0.0	0.0
ε_2	1.0	1.0
$bend_z$	0.0	0.0
$taper_z$	0.0	0.0

TABLE 2
Recovered Dimensions for Mallet

Dimension	Part	
	Head	Handle
Width	16.74	8.18
Height	16.74	8.18
Length	25.66	42.40

in the image. The last step in recognizing the object is to satisfy the functional relation between the two parts which is mapped into a spatial constraint on the part junction. Since the computed relative orientation of the two parts is such that their z axes are orthogonal (>60° in our qualitative partitioning of angle), and since the junction occurs at the end of the handle and at the middle of the head, the algorithm successfully verifies the hammer in the image.

In the second example, we apply our function-based unexpected object recognition approach to a scene containing a short cylinder attached to the side of a block; the image is shown in Fig. 9a, while the segmented region image is shown in Fig. 9b. Only two qualitative volumes are recovered in which the conditional probability mapping the recovered aspect to the recovered volume is high (see [7–9]), corresponding to the block and the cylinder.[6] The fitted models for the two recovered qualitative volumes are shown in Figs. 10a and 10b, respectively.

From the recovered superquad parameters in Table 3, we can proceed to classify each part as either a stick, a strip, a plate, or a blob according to Equations (11), (12), and (13) in Section 2.1.1; the results are shown in Table 4. Using Equations (1), (2), (3), and (4), and again defining

[6] All regions in the scene which do not touch the border of the image are examined by the algorithm in terms of their possible groupings with adjacent regions to form part-based aspects.

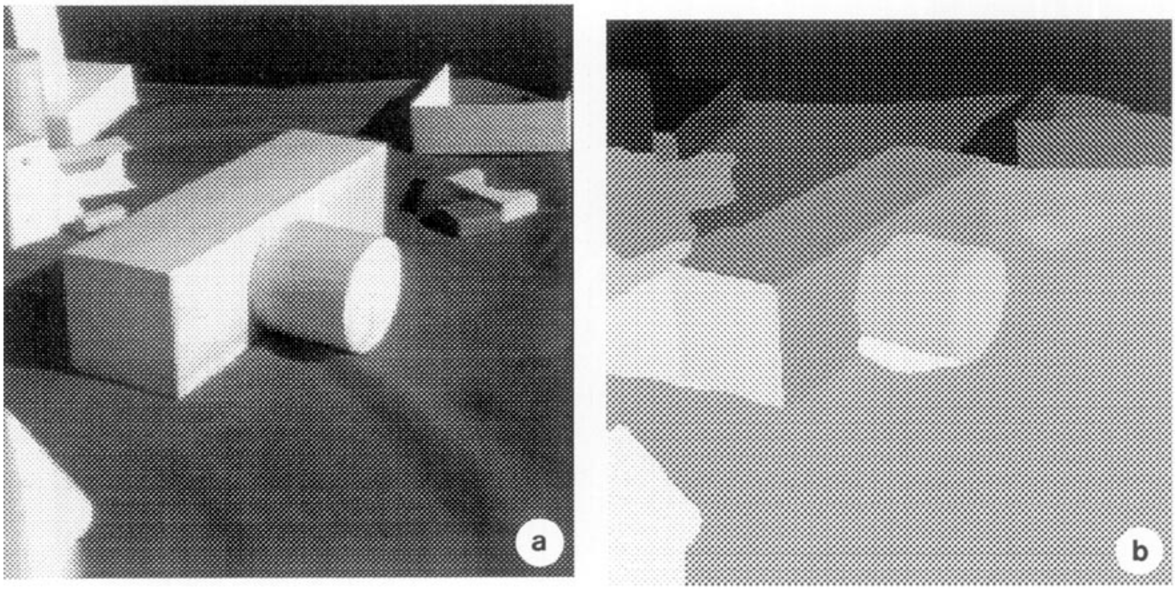

FIG. 9. (a) Original image. (b) Segmented region image.

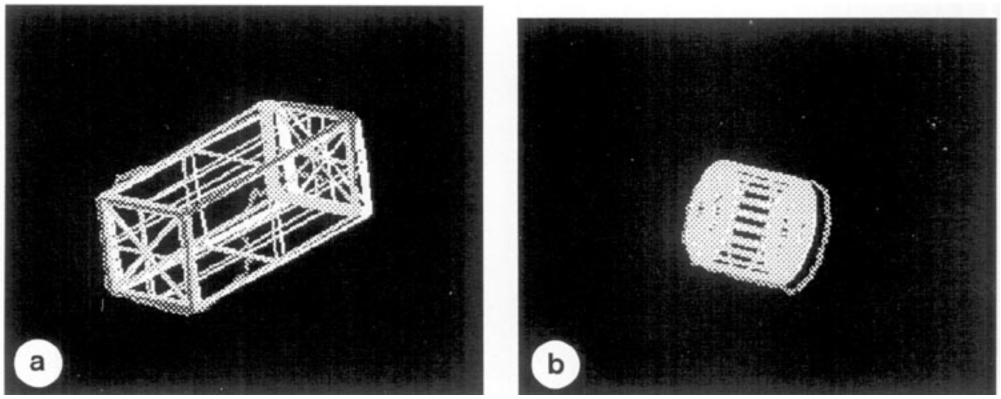

FIG. 10. Recovered superquads: (a) Block. (b) Cylinder. Intermediate gray values along contour portions represent locations of image forces acting on the superquad.

two dimensions as similar if the ratio of the biggest to the smallest is within 4:1 (width:height:length ratios for the two parts are 1:1:2.51 for the block and 1:1:0.89 for the cylinder), then both the block and the cylinder are classified as blobs. Although their connection position and orientation are consistent with the hammer model, the hammer model requires that its handle be a stick. In addition, the relative size of stick length to blob dimension does not fall in the required 1:1 to 4:1 range. The unknown object cannot, therefore, be classified as a hammer.

TABLE 3
Recovered Superquad Parameters for Unkown Object

Superquad parameter	Part	
	Head	Handle
a	37.19	37.19
a_1	0.54	0.45
a_2	0.54	0.45
a_3	1.36	0.40
t_x	−2.77	1.64
t_y	0.32	−0.06
t_z	−50.0	−50.0
r_{11}	0.78	0.39
r_{12}	−0.11	0.02
r_{13}	−0.62	0.92
r_{21}	−0.16	0.49
r_{22}	0.92	0.84
r_{23}	−0.36	−0.23
r_{31}	0.61	−0.78
r_{32}	0.38	0.54
r_{33}	0.70	0.32
ε_1	0.05	0.1
ε_2	0.05	1.0
$bend_z$	0.0	0.0
$taper_z$	0.0	0.0

6. LIMITATIONS

The domain of hand tools defines a simple, one-to-one mapping between an object's functional primitives and relations and their corresponding shape primitives and relations. In the more general case, shape primitives must be grouped to form larger shape primitives according to rules such as coplanarity, colinearity, and cocurvilinearity. The mapping from shape primitives to functional primitives is many-to-one, and a much more elaborate reasoning strategy is required to support the inference of a functional primitive from a collection of interacting shape primitives. Nevertheless, we strongly believe that such a reasoning mechanism must operate at the level of an object's coarse volumetric parts.

The method that we propose for shape recovery, i.e, the use of qualitative shape to constrain the recovery of quantitative shape, relies on a good region segmentation of an intensity image [6] or a range image [7], although a certain degree of both undersegmentation (as shown in the results) and oversegmentation can be tolerated. Furthermore, our region segmentation techniques rely on object surfaces being opaque and devoid of surface markings. In current work, we are trying to introduce a feedback mechanism between the qualitative and quantitative shape recovery processes to better handle region segmentation

TABLE 4
Recovered Dimensions for Unknown Object

Dimension	Part	
	Block	Cylinder
Width	20.08	16.74
Height	20.08	16.74
Length	50.58	14.88

problems and are extending our region segmentation strategy to include other measures of homogeneity such as texture and colour.

Finally, the object representation described in this paper is appropriate for objects composed of simple volumetric parts. Furthermore, we support only functionality that is defined in terms of an object's shape. Functions that are based on color, texture, or more importantly, motion, are not currently supported, although in current work we are enhancing our representation to include motions of an object's parts.

7. CONCLUSIONS

We have presented an approach to function-based object recognition that reasons about the functionalities of an object's parts. Previous approaches have relied on more global object features, often ignoring the problem of object segmentation and thereby restricting themselves to range maps of unoccluded scenes. We extend the popular "recognition by parts" shape recognition framework to support "recognition by functional parts," by combining a set of functional primitives and their relations with a set of abstract volumetric shape primitives and their relations. We show how these shape primitives and relations can easily be recovered from superquadric ellipsoids which, in turn, can be recovered from either range or intensity images of occluded scenes. Furthermore, the proposed framework supports both unexpected (bottom-up) object recognition and expected (top-down) object recognition.

REFERENCES

1. A. Barr, Superquadrics and angle-preserving transformations, *IEEE Comput. Graphics Appl.* **1,** 1981, 11–23.
2. I. Biederman, Human image understanding: Recent research and a theory, *Comput. Vision Graphics Image Process.* **32,** 1985, 29–73.
3. M. Brady, P. Agre, D. Braunegg, and J. Connell, The mechanics mate, in *Advances in Artificial Intelligence* (T. O'Shea, Ed.), pp. 79–94, Elsevier, Amsterdam, 1985.
4. R. Brooks, Model-based 3-D interpretations of 2-D images, *IEEE Trans. Pattern Anal. Mach. Intell.* **5**(2), 1983, 140–150.
5. S. Dickinson, H. Christensen, J. Tsotsos, and G. Olofsson, Active object recognition integrating attention and viewpoint control, in *Proceedings, ECCV '94, Stockholm, Sweden, May 1994.*
6. S. Dickinson and D. Metaxas, Integrating qualitative and quantitative shape recovery, *Int. J. Comput. Vision* **13**(3), 1994, 1–20.
7. S. Dickinson, D. Metaxas, and A. Pentland, Constrained recovery of deformable models from range data, in *Proceedings, 2nd International Workshop on Visual Form, Capri, Italy, May 1994.*
8. S. Dickinson, A. Pentland, and A. Rosenfeld, From volumes to views: An approach to 3-D object recognition, *CVGIP: Image Understanding* **55,** 1992, 130–154.
9. S. Dickinson, A. Pentland, and A. Rosenfeld, 3-D shape recovery using distributed aspect matching, *IEEE Trans. Pattern Anal. Mach. Intell.* **14**(2), 1992, 174–198.
10. F. Ferrie, J. Lagarde, and P. Whaite, Darboux frames, snakes, and superquadrics, in *Proceedings, IEEE Workshop on Interpretation of 3D Scenes, 1989,* pp. 170–176.
11. F. Ferrie, J. Lagarde, and P. Whaite, Recovery of volumetric descriptions from laser rangefinder images, in *Proceedings, ECCV '90, Antibes, France, April 1990,* pp. 387–396.
12. J. Gibson, *The Ecological Approach to Visual Perception,* Houghton Mifflin, Boston, 1979.
13. A. Gupta, Surface and volumetric segmentation of 3D objects using parametric shape models, Technical Report MS-CIS-91-45, GRASP LAB 128, University of Pennsylvania, Philadelphia, PA, 1991.
14. D. Huttenlocher, Three-dimensional recognition of solid objects from a two-dimensional image, Technical Report 1045, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA, 1988.
15. D. Huttenlocher and S. Ullman, Recognizing solid objects by alignment with an image, *Internat. J. Comput. Vision* **5**(2), 1990, 195–212.
16. D. Lowe, *Perceptual Organization and Visual Recognition,* Kluwer Academic, Norwell, MA, 1985.
17. D. Lowe, Fitting parameterized three-dimensional models to images, *IEEE Trans. Pattern Anal. Mach. Intell.* **13**(5), 1991, 441–450.
18. D. Marr, *Vision,* W. H. Freeman, San Francisco, CA, 1982.
19. P. Mulgaonkar, L. Shapiro, and R. Haralick, Matching "sticks, plates and blobs" objects using geometric and relational constraints, *Image Vision Comput.* **2**(2), 1984, 85–98.
20. A. Pentland, Perceptual organization and the representation of natural form, *Artif. Intell.* **28,** 1986, 293–331.
21. A. Pentland, Automatic extraction of deformable part models, *Int. J. Comput. Vision* **4,** 1990, 107–126.
22. A. Pentland and S. Sclaroff, Closed-form solutions for physically based shape modeling and recognition, *IEEE Trans. Pattern Anal. Mach. Intell.* **13**(7), 1991, 715–729.
23. N. Raja and A. Jain, Recognizing geons from superquadrics fitted to range data, *Image Vision Comput.* **10**(3), 1992, 179–190.
24. E. Rivlin, Y. Aloimonos, and A. Rosenfeld, Purposive recognition: A framework, Technical Report CAR-TR-2811, Center for Automation Research, University of Maryland, College Park, MD, December 1991.
25. F. Solina, Shape recovery and segmentation with deformable part models, Technical Report MS-CIS-87-111, GRASP LAB 128, University of Pennsylvania, Philadelphia, PA, 1987.
26. L. Stark and K. Bowyer, Achieving generalized object recognition through reasoning about association of function to structure, *IEEE Trans. Pattern Anal. Mach. Intell.* **13**(10), 1991, 1097–1104.
27. L. Stark and K. Bowyer, Generic recognition through qualitative reasoning about 3-D shape and object function, in *Proceedings, IEEE Conference on Computer Vision and Pattern Recognition, Maui, HI, 1991,* pp. 251–256.
28. L. Stark and K. Bowyer, Indexing function-based categories for generic object recognition, in *Proceedings, IEEE Conference on Computer Vision and Pattern Recognition, Champaign, IL, 1992,* pp. 795–797.
29. L. Stark and K. Bowyer, Indexing function-based categories for generic object recognition, *CVGIP: Image Understanding,* to appear.
30. L. Stark, L. Hall, and K. Bowyer, An investigation of methods of combining functional evidence for 3-D object recognition, *Int. J. Pattern Recognit. Artif. Intell.,* to appear.
31. L. Stark, A. Hoover, D. Goldgof, and K. Bowyer, Function based recognition from incomplete knowledge of shape, in *Proceedings,*

IEEE Workshop on Qualitative Vision (P. Kahn, Y. Aloimonos, and D. Weinshall, Eds.), pp. 11–22, 1993.

32. M. Sutton, L. Stark, and K. Bowyer, Function-based generic recognition for multiple object categories, in *Three-Dimensional Object Recognition Systems* (A. Jain and P. Flynn, Eds.), Advances in Image Communication and Machine Vision Series, Elsevier, Amsterdam, 1993.
33. D. Terzopoulos and D. Metaxas, Dynamic 3D models with local and global deformations: Deformable superquadrics, *IEEE Trans. Pattern Anal. Mach. Intell.* **13**(7), 1991, 703–714.
34. D. Terzopoulos, A. Witkin, and M. Kass, Constraints on deformable models: Recovering 3d shape and nonrigid motion, *Artif. Intell.* **36,** 1988, 91–123.
35. D. Thompson and J. Mundy, Model-directed object recognition on the Connection Machine, in *Proceedings, DARPA Image Understanding Workshop, Los Angeles, CA, 1987,* pp. 93–106.
36. L. Vaina and M. Jaulent, Object structure and action requirements: A compatibility model for functional recognition, *Int. J. Intell. Systems* **6,** 1991, 313–336.
37. P. Winston, T. Binford, B. Katz, and M. Lowry, Learning physical description from functional descriptions, examples, and precedents, in *Proceedings, AAAI, Palo Alto, CA,* 1983, pp. 433–439.