Active Object Recognition Integrating Attention and Viewpoint Control*

Sven J. Dickinson
Department of Computer Science and
Center for Cognitive Science
Rutgers University
New Brunswick, NJ 08903

Henrik I. Christensen
Laboratory of Image Analysis, IES
Aalborg University
DK-9220 Aalborg, Denmark

John K. Tsotsos
Department of Computer Science
University of Toronto, 6 King's College Rd.
Toronto, Ontario, Canada M5S 1A4

Göran Olofsson
Computational Vision and Active Perception Laboratory
Dept. of Numerical Analysis and Computing Science
Royal Institute of Technology, S-100 44 Stockholm, Sweden

Abstract

We present an active object recognition strategy which combines the use of an attention mechanism for focusing the search for a 3-D object in a 2-D image, with a viewpoint control strategy for disambiguating recovered object features. The attention mechanism consists of a probabilistic search through a hierarchy of predicted feature observations, taking objects into a set of regions classified according to the shapes of their bounding contours. We motivate the use of image regions as a focus-feature and compare their uncertainty in inferring objects with the uncertainty of more commonly used features such as lines or corners. If the features recovered during the attention phase do not provide a unique mapping to the 3-D object being searched, the probabilistic feature hierarchy can be used to guide the camera to a new viewpoint from where the object can be disambiguated. The power of the underlying representation is its ability to unify these object recognition behaviors within a single framework. We present the approach in detail and evaluate its performance in the context of a project providing robotic aids for the disabled.

*To appear in *Computer Vision and Image Understanding*. An earlier, condensed version of this paper was presented at the 1994 European Conference on Computer Vision (ECCV), in Stockholm.

1 Introduction

Active vision can be defined as the use of image interpretation to intelligently change the intrinsic and extrinsic sensor parameters to more effectively solve a particular vision task. One aspect of active vision is the use of an attention mechanism to decide where in the image to search for a particular object. Template matching schemes which move an object template throughout the image offer no attention mechanism since all positions in the image are treated equally. However, any recognition scheme that preprocesses the image to extract some set of features provides a basis for an attention mechanism. Assuming that the recovered image features correspond to model features, object search can be performed at those locations in the image where the features are recovered.

For an attention mechanism to be effective, the features must be distinguishing, i.e., have low entropy. If the recovered features are common to every object being searched, they offer little in the way of focusing the search for an object. This is typical in object recognition systems which match simple image features like corners or zeroes of curvature to model features [30, 19, 39, 26]. Although invariant to viewpoint, there may be an abundance of such features in the image, leading to a combinatorial explosion in the number of possible correspondences between image and model features that must be verified. In the first part of this paper, we will argue that regions, characterized by the shapes of their bounding contours, provide a more effective attention mechanism than simple linear features. We go on to present a Bayesian attention mechanism which maps objects into volumetric parts, maps volumetric parts into aspects, and maps aspects to component faces. Face predictions are then matched to recovered regions with a goodness-of-fit providing an ordering of the search locations.

An effective attention system is not enough to overcome problems such as region segmentation errors, heavy object occlusion, or ambiguous views of the object. At best, such a system could exploit knowledge of the object to possibly recover from some of these problems. For example, in the case of segmentation errors, domain-dependent knowledge could be used to correct under- and over-segmentation of regions, or to group disconnected lines in the image into salient structures. However, in the case of limited information due to occlusion or ambiguous view, the best we could hope for is an object hypothesis based on partial information.

We can enhance the power of our attention mechanism through viewpoint control. In the case of region segmentation problems, the camera could be moved to a viewpoint in which, for example, a given object surface projects to a higher contrast region, or an object edge projects to a higher gradient in the image. Or, if a particular view of an object (or one of its parts) is ambiguous, perhaps due to occlusion, the camera could be moved to disambiguate the object. In the second part of this paper, we extend our object representation for attention to support active viewpoint control. We will introduce a representation, called the *aspect prediction graph*, which is based on the aspect graph. Given an ambiguous view of an object, i.e., a view in which the object cannot be uniquely identified, the representation will first tell us if there is a view of the object which is more discriminating. If so, the representation will tell us in which direction should we move the camera to encounter that view. Finally, the representation will tell us what visual events (the appearance or disappearance of features on the object) we should encounter while moving the camera to the new viewpoint.

Following a brief review of relevant related work, we will first describe the probabilistic object representation used by the attention system. Next, we describe the attention mechanism in detail, while motivating the use of regions as focus features. We will then introduce extensions to the object representation that will support our viewpoint control strategy, followed by its integration with the attention mechanism. Finally, we test both aspects of the system as they apply to the domain of object recognition for robotics aids for the disabled.

2 Related Work

In previous work, we presented a bottom-up approach to the recovery and recognition of objects composed of qualitative 3-D volumetric parts from a single 2-D image (Dickinson, Pentland, and Rosenfeld [14]). The approach is based on a hybrid object representation in which objects are composed of a set of chosen 3-D object-centered volumetric parts; the parts, in turn, are mapped to a set of 2-D viewer-centered aspects. The part recovery problem was formulated as a heuristically guided search through the various groupings of image regions into aspects, each representing a view of a volumetric part. A system called OPTICA (**O**bject recognition using **P**robabilistic **T**hree-dimensional **I**nterpretation of **C**omponent **A**spects) was built to demonstrate the approach, and it was successfully applied to the problem of unexpected object recognition from real images (Dickinson, Pentland, and Rosenfeld [13]).

A major limitation of the approach was both its dependency on a complete and consistent covering of the image regions in terms of a set of aspects, and its assumption that all objects visible in the image are made up of the chosen volumes. OPTICA was first extended to the problem of top-down, expected object recognition, by using knowledge of the target object to focus the various search procedures inherent in OPTICA's unexpected object recognition paradigm (Dickinson and Pentland [11]). However, as a starting point, it still required complete aspect and volume coverings of the image. When dealing with noisy images of less constrained scenes, along with shadows and poor lighting, such coverings of the image are not only very costly, but overly ambitious. More intuitively, is it really necessary to completely recover all high-order shape information in the scene in order to locate a particular object? OPTICA provided no attention mechanism to decide what features to look for in the image or where to look for them.

The importance of incorporating attention mechanisms into an interpretation framework was argued by Tsotsos in [40]. There, both psychological as well as computational evidence was presented. In addition, a model for recognition was described and was applied by way of experimental example to a time-varying medical image domain. In the system of Tsotsos [40], attention was tied to the limiting of search for candidate interpretations. This relationship was subsequently formalized in [41]. A focus of attention during search is derived from the "best guesses" for the solution of the problem at hand. However, search in vision can take many forms. In order to conjoin features (such as "red" and "the letter B") into a single percept, search for corresponding features in different portions of processing hierarchies may be required. It may be that the feature being searched for has no corresponding instance and thus a visual search using eye motion must be initiated. This would be accompanied by establishing expectations as to what the attentive system was looking for, thus biasing the computation. These biases facilitate the computation of particular concepts. Yet a different form of search is that for features that may help to distinguish between two competing interpretations. Biases on computation may be obtained by default mechanisms or by a priori frequency of occurrence (a probability of some form) data.

At the earlier end of visual processing, methods exist for the localization of salient image features. A biologically plausible scheme to solve the problem of selective visual attention appears in Tsotsos [42] and Culhane and Tsotsos [5, 6]. It proposes a method that solves the problem of locating and localizing items in the visual field and shows how to implement the idea of an inhibitory attention beam. The scheme is based on the foundation laid by Koch and Ullman [24], but incorporates several novel changes and additions which permit a proof of convergence with constant time convergence properties. Furthermore, it addresses the issue of saliency maps and the binding across representations, and includes much tighter comparisons to biology. Experiments show that luminance, edge, and motion fields (regions of common flow) can be used equally well as input representations.

At the higher end of visual interpretation, Tsotsos [40] showed how to use a number of different organizational axes of a model database to limit search through a set of default heuristics. The present work has a similar flavor, but adds to the abstract framework a priori probability measures for 3-D object recognition.

Several other researchers have addressed the problem of attention in the context of computer vision systems. Rimey and Brown [36] used Bayesian networks for selection of preprocessing modules and spatial attention regions, but the approach is based on explicit and accurate modeling of the domain. Furthermore, the approach focused more on scene context than on how a particular 3-D object may be viewed. Califano et al. [4] used a heuristic method for the selection of operators based on discriminative power; the attention region is defined in terms of the region of ambiguity. Kittler et al. [31] used a rule based method for the selection of operators to facilitate verification of objects based on 2-D information. The method also includes the ability to define spatial attention regions based on temporal context. Stark et al. [38] used functional verification procedures in combination with relation specification for the selection of operators and the definition of attention regions; the reasoning is based on fuzzy logic for evidence combination. Brunnström, Lindeberg, and Eklundh [3] describe a foveation system in which blobs in a scale-space representation of the image are used to guide a foveation mechanism which recovers junction information.

Although some heuristics were introduced to accommodate over-segmentation in OPTICA, lighting conditions had to be favorable for a successful interpretation to be generated [14]. Furthermore, there was no support for viewpoint control to offset the effects of poor region segmentation or ambiguous views of the object. Wilkes and Tsotsos [45] offer a solution to polyhedral object recognition, whereby the camera is moved to a canonical viewpoint of the object based on maximizing the projected lengths of two non-parallel edges in the image. With viewpoint control, they effectively reduce the 3-D recognition problem to a 2-D recognition problem. Hutchinson and Kak [18] describe a system for disambiguating objects recovered from range images. Based on a set of current hypotheses about the identity and position of an object, they evaluate candidate sensing operations with regard to their effectiveness in minimizing ambiguity. Maver and Bajcsy [32] describe an approach to choosing the next view in order to resolve occluded regions in a range image. Based on height information at the polygonally-approximated border of an occluded region, a sequence of views is planned.

Kim, Jain, and Volz [23] explore an approach which determines both optimal camera distance from the object as well as viewing direction. Camera distance was chosen to maximize object surface visibility (or minimize the number of views required to cover the surface of the object) while maintaining a lower bound on feature size in the image. Two approaches to determining camera position are presented. Assuming that the distinguishing feature has been selected, the *visual aspect graph (VAG)* method moves to a position on the viewing sphere belonging to an aspect containing the feature. The aspects can be ranked according to some "goodness" measure. To account for feature visibility when occluding objects are present, the feature(s) being sought are projected onto a spherical or cylindrical screen, effectively partitioning the viewing sphere (or cylinder) into occluded and unoccluded regions. This work differs significantly from our approach in that no object recognition framework is incorporated, no methods for deciding which feature to search for are provided, camera movement appears to be a single viewpoint change with no continuous visual event tracking, and no results are reported.

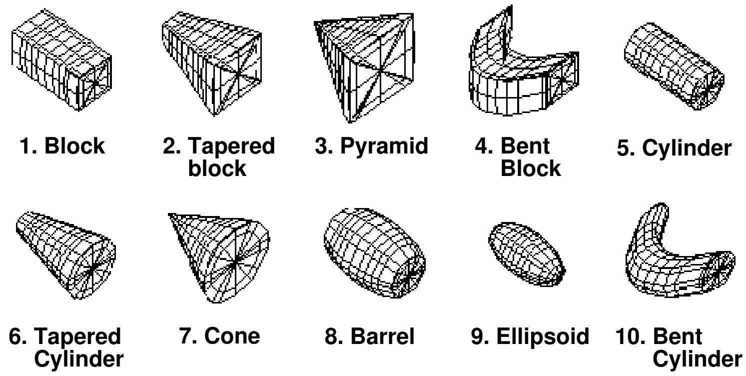


Figure 1: The Ten Modeling Primitives

In our approach, we use a set of viewing probabilities to rank the possible directions in which we can move in order to disambiguate an object (part) hypothesis. An aspect representation, called the aspect prediction graph, of an object's parts encodes the visual events encountered as the sensor moves from one viewpoint to the next. The attention mechanism not only provides initial placement in this graph, but is used in the process of verifying the visual events.

3 Review of the Object Representation

3.1 Object-Centered Modeling

To demonstrate our approach to attention, we have selected an object representation similar to that used by Biederman [2], in which the Cartesian product of contrastive shape properties gives rise to a set of volumetric primitives called geons. Since the introduction of Biederman's geons to the vision community, a number of other researchers have developed computational models for geon recovery resulting in a number of geon-based recognition systems (e.g., Dickinson et al. [8], Bergevin and Levine [1], Hummel and Biederman [17], Munck-Fairwood [15], Jacot-Descombes and Pun [20], and Narayan and Jain [35]). However, unlike these approaches, which are typically applied to manually segmented line drawings, our approach is applied to real images and is not dependent on the choice of geons as modeling primitives. For our investigation, we have chosen three properties including cross-section shape, axis shape, and cross-section size variation (Dickinson, Pentland, and Rosenfeld [12]). The cartesian product of the dichotomous and trichotomous values of these properties give rise to a set of ten volumes (a subset of Biederman's geons), modeled using Pentland's SuperSketch 3-D modeling tool [34], and illustrated in Figure 1. To construct objects, the volumes are attached to one another with the restriction that any junction of two volumes involves exactly one distinct surface from each volume.

3.2 Viewer-Centered Modeling

Traditional aspect graph representations of 3-D objects model an entire object with a set of aspects, each defining a topologically distinct view of an object in terms of its visible surfaces (Koenderink and van Doorn [25]). Our approach differs in that we use aspects to represent a (typically small) set of volumetric parts from which each object in our database is constructed, rather than representing an entire object directly. Consequently, our goal is to use aspects to recover the 3-D volumetric

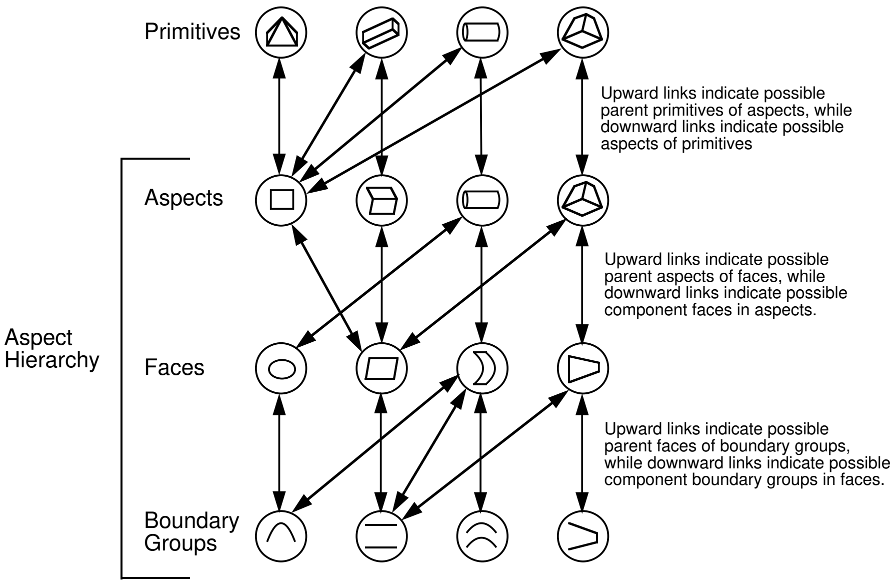


Figure 2: The Augmented Aspect Hierarchy

parts that make up the object in order to carry out a recognition-by-parts procedure, rather than attempting to use aspects to recognize entire objects. The advantage of this approach is that since the number of qualitatively different volumes is generally small, the number of possible aspects is limited and, more important, *independent* of the number of objects in the database. The disadvantage is that if a volumetric part is occluded from a given 3-D viewpoint, its projected aspect in the image will also be occluded. Thus we must accommodate the matching of occluded aspects, which we accomplish by use of a hierarchical representation we call the *aspect hierarchy.*

The aspect hierarchy consists of three levels, consisting of the set of *aspects* that model the chosen volumes, the set of component *faces* of the aspects, and the set of *boundary groups* representing all subsets of contours bounding the faces. The ambiguous mappings between the levels of the aspect hierarchy were originally captured in a set of upward conditional probabilities (Dickinson et al. [7]), mapping boundary groups to faces, faces to aspects, and aspects to volumes. However, for the attention mechanism described in this paper, the aspect hierarchy was augmented to include the downward conditional probabilities mapping volumes to aspects, aspects to faces, and faces to boundary groups.[1] Figure 2 illustrates a portion of the augmented aspect hierarchy.

To generate the conditional probabilities of the aspects given the shapes, we employ the following procedure, as described in [12]. We first model our 3-D volumetric primitives using the Supersketch modeling tool (Pentland [34]). Supersketch models each shape using a superquadric surface subject to stretching, bending, twisting, and tapering deformations. The superquadric with length, width, and breadth a_1, a_2, and a_3 is described (adopting the notation $\cos\eta = \mathbf{C}_\eta$, $\sin\omega = \mathbf{S}_\omega$) by the following equation:

$$\mathbf{X}(\eta,\omega) = \begin{pmatrix} a_1 \mathbf{C}_\eta^{\epsilon_1} \mathbf{C}_\omega^{\epsilon_2} \\ a_2 \mathbf{C}_\eta^{\epsilon_1} \mathbf{S}_\omega^{\epsilon_2} \\ a_3 \mathbf{S}_\eta^{\epsilon_1} \end{pmatrix} \tag{1}$$

where $\mathbf{X}(\eta,\omega)$ is a three-dimensional vector that sweeps out a surface parameterized in latitude η

[1] For the probabilistic search process described in section 5, the augmented aspect hierarchy is actually represented by two acyclic graphs, one capturing the upward conditional probabilities and the other capturing the downward conditional probabilities.

and longitude ω, with the surface's shape controlled by the parameters ϵ_1 and ϵ_2.

The next step in generating the conditional probabilities involves rotating different instances of each primitive about its internal x, y, and z axes in 10° intervals.[2] The resulting quantization of the viewing sphere gives rise to 648 views per shape; however, by exploiting shape symmetries, we can reduce the number of views for the entire set of ten shapes to 688. For each view, we orthographically project the shape onto the image plane, and classify the view in terms of one of the aspects.[3] The resulting frequency distribution gives rise to a set of bottom-up (found in [7]) and top-down conditional probability matrices.

4 Preattentive Feature Extraction

4.1 A Case for Focusing on Regions

Given the various levels of the augmented aspect hierarchy, the question arises: At which recovered features from the image do we focus our search for a particular object? Many CAD-based recognition systems (e.g., Lowe [30], Huttenlocher and Ullman [19], Thompson and Mundy [39], and Lamdan, Schwartz and Wolfson [26]) advocate extracting simple features like corners, high curvature points, or zeroes of curvature. Although robustly recoverable from the image, there may be many such features in the image offering marginal utility for directing a search. Such features are analogous to the boundary group level of features in the augmented aspect hierarchy. By examining the conditional probabilities in the augmented aspect hierarchy, we can compare the relative utility of boundary groups and faces in inferring the identity of a volumetric part.[4]

To compare the utility of boundary groups versus faces in recovering volumes, we will use the conditional probabilities captured in the augmented aspect hierarchy to define a measure of *average inferencing uncertainty*, or the degree to which uncertainty remains in volume identity given a recovered boundary group or face. More formally, we define average inferencing uncertainty for boundary groups, U^{BG}_{Avg}, and for recovered faces, U^{F}_{Avg}, as follows:[5]

$$U^{BG}_{Avg} = -\frac{1}{N_{BG}} \sum_{i=1}^{N_{BG}} \sum_{j=1}^{N_V} Prob(V_j \mid BG_i) \log Prob(V_j \mid BG_i) \qquad (2)$$

$$U^{F}_{Avg} = -\frac{1}{N_{FA}} \sum_{i=1}^{N_F} \sum_{j=1}^{N_V} Prob(V_j \mid F_i) \log Prob(V_j \mid F_i) \qquad (3)$$

where:

N_{BG} = number of boundary groups in the augmented aspect hierarchy
N_{FA} = number of faces in the augmented aspect hierarchy
N_V = number of volumes in the augmented aspect hierarchy

Table 1 compares the average inferencing uncertainty for the boundary groups and faces. Clearly, faces offer a more powerful focus feature for the recovery of volumetric parts than do the simpler features that make up the boundary groups. However, this advantage is only realizable if the cost of

[2] All spatial orientations of the shapes are assumed to be equally likely.

[3] Note that this procedure also yields the individual faces and aspects associated with the shapes.

[4] Since aspect recovery first requires the recovery of component faces, we will examine the choice between recovering simple contour-based features (boundary groups) and regions (faces).

[5] We have suppressed the zero-probability terms in this and remaining expressions for notational simplicity.

Feature	Avg. Uncertainty
boundary groups	0.74
faces	0.23

Table 1: Average Uncertainty in Inferring Volumes from Boundary Groups and Faces

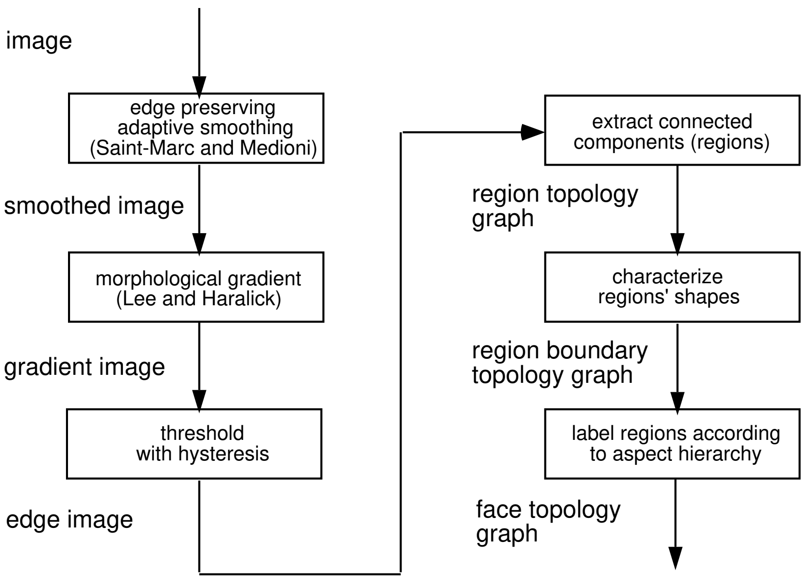


Figure 3: Face Recovery

extracting the two types of features is comparable. By using simple region segmentation techniques whose complexity is comparable to common edge detection techniques, we can avoid the complexity of grouping lines into faces [30]. We can accommodate the segmentation errors associated with a cheap region grower by using partial information to intelligently guide viewpoint control to improve interpretation, the subject of the second part of this paper.

4.2 Face Recovery

Having decided to use labeled faces as a focus-of-attention feature, we can proceed to outline the process of face recovery, as outlined in Figure 3. Face recovery consists of first extracting a set of regions from an image, describing the shapes of the regions' bounding contours, and classifying the regions' shapes. In the following sections, we discuss these processes in greater detail.

4.2.1 Region Segmentation

There are two approaches to recovering closed contours representing image regions. In a region-based approach, pixel homogeneity is used to cluster similar pixels together to form a region. Tracing the boundary of a region yields the bounding contour. In an edge-based approach, edges are extracted and grouped to form closed sets of contours. Inevitable gaps in the edges make the grouping process computationally complex, as was demonstrated by Lowe [30]. We avoid this grouping complexity by simply performing a connected component labeling of an edge image. If a gap exists in a line, then the regions on either side of the line will get the same component label. The result is that there is significant region undersegmentation in the image, but the computational complexity is comparable to simple region-based approaches.

In our implementation, we begin by applying Saint-Marc, Chen, and Medioni's edge-preserving adaptive smoothing filter to the image [37], followed by a morphological gradient operator (Lee et al. [27]). A hysteresis thresholding operation is then applied to produce a binary image from which a set of connected components is extracted. Edge regions are then thinned and assigned to neighboring regions, resulting in a *region topology graph* in which nodes represent regions and arcs specify region adjacencies. In future work, our goal is to move towards a true region-based segmentation method, capturing both the properties of the region's boundaries (or edges) as well as the region's internal composition.

4.2.2 Shape Description

From the region topology graph, each region is characterized according to the qualitative shapes of its bounding contours. The steps of partitioning the bounding contour and classifying the resulting contours are performed simultaneously using a minimal description length algorithm due to Li [29]. From a set of initial candidate contour breakpoints (derived from a polygonal approximation), the algorithm considers all possible groupings of the inter-breakpoint contours according to a minimum description length measure based on how well lines and elliptical arcs can be fit to the segment groups in terms of the cost of coding the various segments. The partitioned segments of the bounding contour are represented as labeled nodes in a *region boundary graph*, with arcs between the nodes representing adjacency (co-termination), parallelism, or symmetry. Two non-coterminating lines are considered parallel if the angle between their fitted lines is small, while two non-coterminating curves are considered parallel if one is convex, one is concave, and the angle between their directions is small.[6] Two non-coterminating, non-parallel lines are considered symmetric if there is sufficient overlap when one line is projected onto the other.[7]

4.2.3 Face Classification

Once the regions have been extracted and their shapes described, we must classify each region's shape according to the faces in the augmented aspect hierarchy. The classification of an image region consists of matching its region boundary graph to those graphs representing the faces in the augmented aspect hierarchy using an interpretation tree search (Grimson and Lozano-Pérez [16]). If there is an exact match, as shown in Figure 4, then we immediately generate a *face hypothesis* for that image region, identifying the label of the face. If for any reason (e.g., occlusion, segmentation errors, noise, etc.) there is no match, we must descend to the boundary group level of the augmented aspect hierarchy, as shown in Figure 5. We then compare *subgraphs* of the region boundary graph describing the image region to those graphs at the boundary group level of the augmented aspect hierarchy. For each subgraph that matches, we generate a face hypothesis with a probability determined by the appropriate entry in the conditional probability matrix (in the augmented aspect hierarchy) mapping boundary groups to faces and the proportion of the region's

[6] The direction of a curve is computed as the vector whose head is defined by the midpoint of the line joining the two endpoints of the curve, and whose tail is defined by the point on the curve whose distance to the line joining the endpoints is greatest.

[7] Two non-parallel vectors will have an intersection point. When one vector is rotated about that point, it can be brought into correspondence with the other. If the resulting overlap of the two lines is a large portion of the smaller of the two lines, the lines are said to be symmetric.

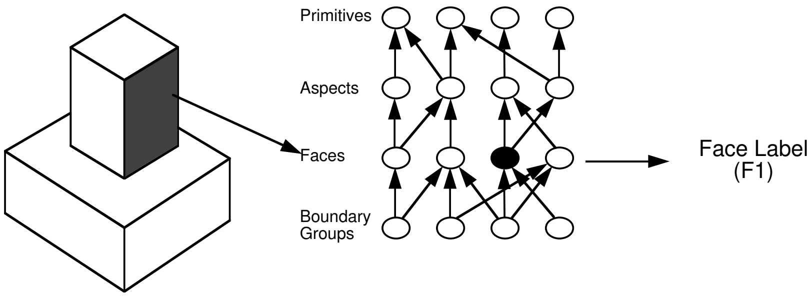


Figure 4: Labeling an unoccluded region

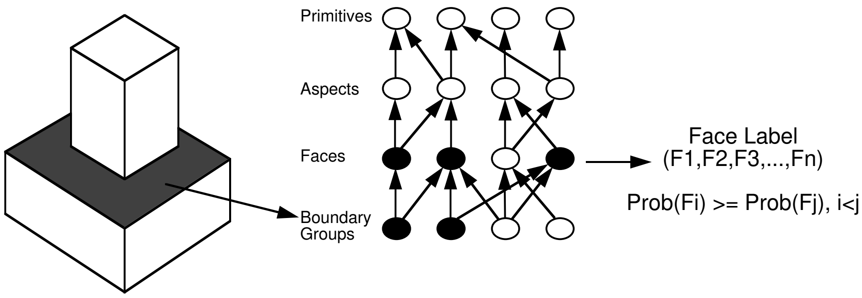


Figure 5: Labeling an occluded region

bounding contour covered by the boundary group. The labeled face hypotheses for all regions in the image are captured in a *face topology graph*.

5 Attention-Driven Recognition

5.1 Focusing the Search

The goal of the attention mechanism is to limit search, both in the image and in the model database (augmented aspect hierarchy). Our first task is to decide what features in the image we are searching for. There is a very important trade-off here which is critical to the problem of choosing features to attend to. On one hand, we wish to expend as little effort as possible in recovering a set of features that might belong to the object and hence give us a place to initiate the search for the object. However, if the recovered features are not discriminative enough, we will be faced with an abundance of features located throughout the image, all attempting to draw the attention mechanism to their location. Thus we have a trade-off between uniqueness, or indexing power, and cost of recovery.

In Section 4.1, we concluded that for 3-D modeling primitives which resemble the commonly used generalized cylinders, superquadrics, or geons, the most appropriate image features for recognition appear to be image regions, or faces. Moreover, the utility of a face description can be improved by grouping the faces into the more complex aspects, thus obtaining a less ambiguous mapping to the volumes and further constraining their orientation. Only when a face's shape is altered due to volume occlusion or intersection should we descend to analysis at the contour or boundary group level. For our attention mechanism, we will essentially reverse this process. Starting from the object, we will generate predictions down to the level of labeled faces. Since our face recovery preprocessing step recovers labeled faces, our attention can be drawn to those recovered labeled faces which match predictions. Furthermore, since each face label has a corresponding probability, that probability can be used to rank-order candidate image faces for search.

In selecting which recovered face to focus our attention on, we utilize a decision theoretic approach using a Bayesian framework. A similar approach was reported by Levitt et al. [28], who use Bayesian networks for both model representation and description of recovered image features. Specifically, they use Bayesian networks for both data aggregation and selection of actions and feature detectors based on expected utility. The approach is thus centered around the use of a Bayesian approach to integration and control. Similar techniques have also been reported by Rimey and Brown [36], and Jensen et al. [21], where both regions of interest and feature detectors are selected according to utility/cost strategies.

To select a region of interest, i.e., attend to a particular face, the augmented aspect hierarchy may be considered as a Bayesian network, allowing us to utilize decision theory as described, for example, by Pearl [33]. To apply such a strategy, it is necessary to define both utility and cost measures. The utility function, U, specifies the power of a given feature at one level of the augmented aspect hierarchy, e.g., volumes, aspect, faces, and boundary groups, to discriminate a feature at a higher level. The cost function, C, specifies the cost of extracting a particular feature. The subsequent planning is then aimed at optimizing the benefit, $\max B(U, C)$; profit, e.g., $utility - cost$, is often maximized in this step. For the system described in this paper, the face recovery algorithm was chosen to support a simple implementation on a real-time, pipeline architecture. The cost of face recovery is assumed to be constant and equal for all types of faces. Given such an implementation, the selection of which face to consider next should simply optimize the utility function.

The process of mapping an object to a candidate face is outlined in Figure 6. Given a target object, $object_T$, the first step is to choose a target volume, $volume_T$, to search for. Next, given a target volume, $volume_T$, we choose a target aspect, $aspect_T$, to search for. Finally, given a target aspect, $aspect_T$, we choose a target face, $face_T$, to search for. Given a target face, $face_T$, we then examine the face topology graph for labeled faces which match $face_T$. If there is more than one, they are ranked in descending order according to their probabilities.

The above top-down sequence of predictions represents a depth-first search of a tree defined by each object; the root of the tree represents the target object, while the leaf nodes of the tree represent target faces. The target volume subtrees for each object tree are independent of the object database and can be specified at compile time. The branching factor at a given node in any object tree can be reduced by specifying a probability (or utility) threshold on a prediction.

The heuristic we use to guide the search is based on the power of an object's features, e.g., volumes, aspects, and faces, to identify the object. For example, to determine how discriminative a particular volume, $volume_i$, is in identifying the target object, $object_T$, we use the following function:

$$D(volume_i, object_T) = \frac{Prob(object_T|volume_i)}{\sum_j Prob(object_j|volume_i)} * Prob(volume_i) \tag{4}$$

The numerator specifies how discriminative $volume_i$ is for $object_T$, while the ratio specifies the "voting power" of $volume_i$ for the object of interest. $Prob(object_j|volume_i)$, for any given i and j, is computed directly from the contents of the object database. The last term specifies the likelihood of finding the volume, and is included to discourage the selection of a volume which is highly discriminative but very unlikely. The $Prob(volume)$ may be calculated as follows:

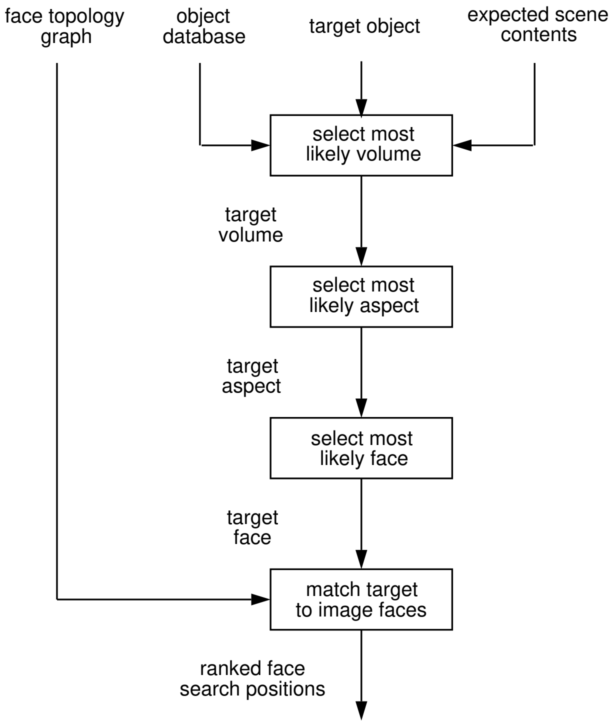


Figure 6: Attention Mechanism

$$Prob(volume_i) = \sum_k (Prob(volume_i|object_k) * Prob(object_k)) \tag{5}$$

where $Prob(volume_i|object_k)$, for any given i and k, is computed directly from the object database, and $Prob(object_k)$ represents a priori knowledge of scene content. In a similar fashion, we define $D(aspect_i, volume_T)$ as follows:

$$D(aspect_i, volume_T) = \frac{Prob(volume_T|aspect_i)}{\sum_j Prob(volume_j|aspect_i)} * Prob(aspect_i) \tag{6}$$

$$Prob(aspect_i) = \sum_k (Prob(aspect_i|volume_k) * Prob(volume_k)) \tag{7}$$

and $D(aspect_T, face_i)$ as follows:

$$D(face_i, aspect_T) = \frac{Prob(aspect_T|face_i)}{\sum_j Prob(aspect_j|face_i)} * Prob(face_i) \tag{8}$$

$$Prob(face_i) = \sum_k (Prob(face_i|aspect_k) * Prob(aspect_k)) \tag{9}$$

To determine the best target volume to search for in order to recognize the target object, the following utility function is evaluated:

$$U(volume_T, object_T) = \max_a \left(\frac{Prob(object_T|volume_a)}{\sum_j Prob(object_j|volume_a)} * Prob(volume_a)\right) \tag{10}$$

To determine the best target aspect to search for in order recover the target volume, the following utility function is evaluated:

$$U(aspect_T, volume_T) = \max_a\left(\frac{Prob(volume_T|aspect_a)}{\sum_j Prob(volume_j|aspect_a)} * Prob(aspect_a)\right) \tag{11}$$

To determine the best target face to search for in order recover the target aspect, the following utility function is evaluated:

$$U(face_T, aspect_T) = \max_a\left(\frac{Prob(aspect_T|face_a)}{\sum_j Prob(aspect_j|face_a)} * Prob(face_a)\right) \tag{12}$$

When we descend the search tree to a given target face, we search for matching face candidates in the face topology graph. We focus our attention on the best face matching the target face, and proceed to verify the object, as described in the next section. If a target face, target aspect, or target volume cannot be verified, the search algorithm backtracks, applying the above utility functions to remaining faces, aspects, and volumes in the search tree.

5.2 Verification

From a target face in the image which matches a target face in the object tree, we next proceed to recognize the object using the process shown in Figure 7. Recognition is the process by which we move from a matched target face node in the search tree back up to an object. Once we have a matched face leaf node, our next step is to verify its parent (target) aspect [14]. This entails searching the vicinity of the target face for faces whose labels and configuration match the target aspect using an interpretation tree search (Grimson and Lozano-Pérez [16]). Note that the resulting verified aspect has a score associated with it which can be compared to a score threshold to terminate the search from a particular target face.[8] The score of a recovered aspect is calculated as follows:

$$AspectScore = \frac{1}{N}\sum_{k=1}^{N} Prob(Face_k) * \frac{Length(BG_k)}{Length(Region_k)} \tag{13}$$

where: N is the number of faces in model aspect, $Length(BG_k)$ is the length of boundary group, and $Length(Region_k)$ is the perimeter of the region. Note that if the region boundary graph recovered for the shape *exactly* matches some face in the augmented aspect hierarchy, its probability will be 1.0 and the length of its boundary group will be the perimeter of the entire region.

Once a target aspect is found, we then proceed up the tree one level to the target volume, defining a mapping between the faces in the target aspect and the surfaces on the target volume. The score of a volume is calculated as follows:

$$VolumeScore = AspectScore * Prob(ModelVolume|ModelAspect) \tag{14}$$

where:

$Prob(ModelVolume|ModelAspect)$ = probability of volume given aspect (from the augmented aspect hierarchy)

Moving back one level to the object, we must then decide whether or not we have enough information confirming the target object. If so, the recognition process is complete. If not, we must then decide which volume to search for next.

[8]The aspect score is a function of the scores of its component faces.

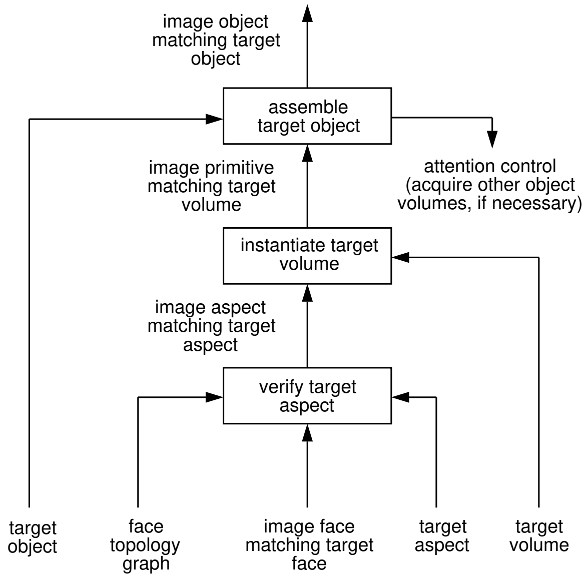


Figure 7: Object Verification

5.3 Searching for Multi-part Objects

If the target object has more than one part, then verification may involve searching for multiple parts of an object. The constraint we rely on is that if two parts belonging to the object are connected in 3-D, then to verify these two parts in the image, their corresponding part aspects must be topologically connected in 2-D. Furthermore, a specification of how the parts are attached in 3-D can be mapped to a specification as to which regions in their respective aspects are adjacent in 2-D [14]. Note that this constraint is simply a heuristic used in our search. Due to occlusion, the aspects corresponding to two two parts connected in 3-D may not be topologically adjacent in 2-D. Conversely, the aspects corresponding to two disconnected parts in 3-D may be topologically adjacent in 2-D. Absolute verification of a connection would require recovering the geometry and pose of the two parts, which we address in [10].

Given our connectivity constraint, a target volume should be chosen as the most discriminating volume *among those that are connected in 3-D to a volume already verified.* Since the new target volume is connected to a verified volume, we can focus our search for the target face on only those image faces that are topologically adjacent to the faces belonging to the verified volume.

5.4 Computational Complexity

In this section, we briefly outline the complexity of our search process. Since the complexity of the region segmentation and boundary segmentation tasks are entirely dependent on the choice of segmentation algorithms, we only address the complexity of the face labeling. Recall that each recovered face can be represented as a graph. Since the size of the largest face and the number of faces in the augmented aspect hierarchy are fixed, the complexity of face matching is $O(n^{c_{max}})$, where n is the number of contours making up a recovered image face, and c_{max} is the maximum number of contours making up an augmented aspect hierarchy face or boundary group.

When searching for an object with k parts, we can end up searching for each of its k component volumes. Each of those volumes can appear as a maximum of a_{max} aspects, where a_{max} is the maximum number of aspects modeling any volume in our augmented aspect hierarchy. At the next level, we have have to search for f_{max} different faces, where f_{max} is the maximum number of faces making up any aspect in the augmented aspect hierarchy. If there are r regions in the image, then each region, in the worst case, could have a face hypothesis for each type of model face. Thus, for a given object with k parts and an image with r regions, in which the maximum number of contours making up a region is n, the complexity of the recognition process, including preprocessing, is $O(ka_{max}f_{max}rn^{c_{max}})$ which, given that a_{max} and f_{max} are constants, reduces to $O(krn^{c_{max}})$.

6 Viewpoint Control

6.1 The Role of Active Vision in Object Recognition

During the volume recovery process, we may not be able to recover a volume in its most likely view. In fact, the likelihood is significant that some viewpoint degeneracy will occur (see Wilkes, Dickinson, and Tsotsos [44]). Unfortunately, in examining the conditional probabilities inherent in the augmented aspect hierarchy, we discover that less likely views of a volume may not be unique to that volume. For example, the least likely view of the block volume (volume 1 in Figure 1) is the aspect consisting of a single parallelogram face. This same face, in fact, represents a valid aspect for *all* the other volumes except the ellipsoid, the barrel, the cone, and the truncated cone. Furthermore, less likely views of a volume often underconstrain an attempt to fit a quantitative shape model to the recovered qualitative shape (Dickinson and Metaxas [10]).

Clearly, given a low probability view of a volume, we would like to use its projected aspect, along with knowledge of the volume's possible aspects and their probabilities (from the augmented aspect hierarchy), to predict not only which aspect represents a "better" view of the volume, but how the camera should be moved in order to find it. Conversely, given a recovered aspect of a volume, along with a direction of motion, we might wish to know into what aspect the current aspect will transform. In the next section, we present a new representation, called the *aspect prediction graph*, which supports these two queries.

6.2 Extending the Representation

Consider a monocular camera system actively observing a scene containing static objects. Furthermore, assume that our camera can fixate on a given object while moving around it. As mentioned earlier, we would like to predict the object's appearance in a new view, as well as propose a direction of camera movement to obtain a "better" view of the object's volumetric parts. To begin with, let us assume that the observer presently perceives an aspect of one of the object's volumetric parts. For the given volume, we would like a representation that not only specifies the possible aspects of the volume, but also the transitions or relations between its aspects. Each relation should represent a qualitatively distinct change in viewpoint, and should specify how the faces of the involved aspects are related under this change in viewpoint.

To capture these relationships, we have constructed an *aspect prediction graph* for each of the ten volumes. The aspect prediction graph (APG) is derived from two sources. The first is a traditional aspect graph (Koenderink and van Doorn [25]) in which nodes represent topologically distinct views of an object and arcs specify transitions between the views. The APG is a more compact version of the aspect graph in which topologically equivalent nodes are grouped regardless of whether their faces map to different surfaces on the object. For example, the APG for a block encodes 3 aspects for a block (volume 1 in Figure 1) while a traditional aspect graph encodes 26 aspects. Next, the APG specifies the visual events in terms of which faces appear/disappear when moving from one aspect to another. Furthermore, the position of such a face appearance/disappearance from a source aspect to a target aspect is specified with respect to particular contours of faces in the source aspect (event contours). In addition, the transition between two nodes (aspects) encodes the direction(s) relative to the event contours that one must move in the image plane in order to observe the visual event. Finally, the APG borrows from the augmented aspect hierarchy both the $Prob(volume|aspect)$ and $Prob(aspect|volume)$ conditional probabilities, and assigns them to the nodes in the APG. Note that the downward conditional probabilities from a given volume to its possible aspects are independent of the other volumes. However, the upward conditionals in an APG's nodes are a function of the other volumes; hence the collection of APG's corresponding to the set of volumes are linked by the upward conditional probabilities at their nodes.

Given an aspect of a volume, the observer can usually move in more than one direction to get to some other aspect. For example, given a frontal view of a block with only a single visible face, we could move left, right, up, or down (assuming that you can move underneath the block) to an aspect containing two visible faces. To cover all these alternatives, the APG encodes multiple arcs between the aspects, each representing a qualitatively distinct view change direction. In addition, associated with each of these arcs are one or more *face events*. Each face event specifies what face will appear or disappear under the change of view corresponding to the arc, and where the event will occur relative to the source aspect. In the example with the frontally viewed block, a face will appear on the left of the original face if the observer moves left, on the right when moving right, or above when moving up. A movement towards the upper-left or upper-right would bring a new aspect and two new faces into view: one face above and one to the left of the original face if moving towards the upper-left, and one above and one to the right if moving towards the upper-right.

To illustrate the above concepts, Figure 8 presents the APG for the block volume, illustrating the three possible aspects of the block. Between every two nodes (aspects) in the aspect prediction graph are a pair of directional arcs. The directional arc between aspect 1 and aspect 2 in Figure 8 is expanded in Figure 9. From aspect 1 in Figure 8, there are three ways to move to a view in which aspect 2 will be visible. Movement relative to contours 0 and 1 on face 2 will cause a visual event in which face 2 disappears at contour 1 on face 0 and at contour 3 on face 1. Or, movement relative to contours 0 and 1 on face 0 will cause a visual event in which face 0 will disappear at contour 0 on face 1 and contour 0 on face 2. Finally, movement relative to contours 0 and 3 on face 1 will cause a visual event in which face 1 will disappear at contour 0 on face 0 and contour 1 on face 2.

It should be noted that in the augmented aspect hierarchy, each aspect has an indexing of its component faces, and each component face has a similar indexing of its bounding contours. By referring to the normals of such well-defined contours in a recovered aspect, we can qualitatively specify direction rules with respect to an aspect-centered coordinate system. The direction of view change (in the image plane or on the surface of a viewing sphere) is specified as a vector sum of the normals to particular contours of the recovered aspect corresponding to the current APG aspect.[9] The face events are also defined with respect to these specified contours. For example, we can predict along which contour in the current aspect a new face will appear or disappear when moving towards a new aspect.

6.3 A Strategy for Moving the Camera

Using the attention mechanism described earlier in section 5, the search for an object includes a search for its component volumes. Each recovered volume is characterized by the aspect in which it is viewed. For a given aspect of a volume, we can use the volume-to-aspect mappings in the

[9] For concave and convex curve segments, the normal at the midpoint is used.

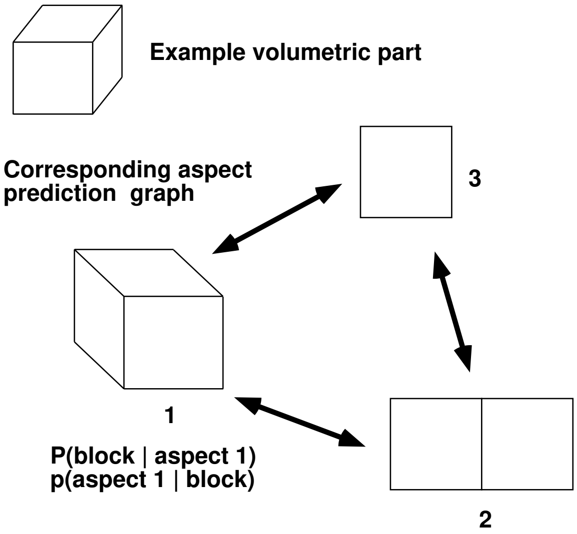


Figure 8: Aspect Prediction Graph (APG) for Volume 1 (Block)

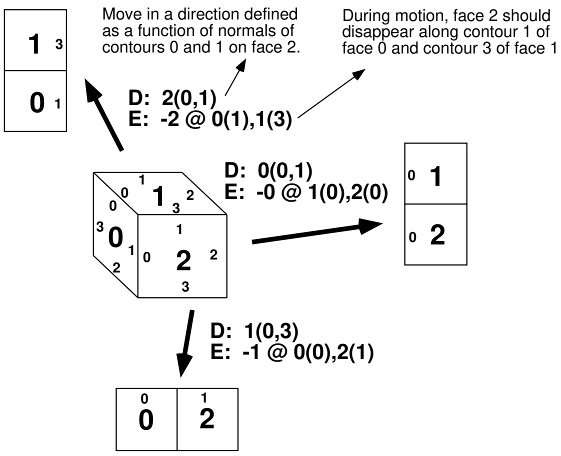


Figure 9: APG Transitions from Aspect 1 to Aspect 2 in Figure 8

aspect prediction graph to determine which aspects (if any) are more probable (or stable) than the current one, by maintaining an ordered list of aspects for each volume, ranked in decreasing order of their downward conditional probabilities. Conversely, if we have an ambiguous aspect whose mapping to the hypothesized volume is weak, we can use the aspect-to-volume mappings in the aspect prediction graph to determine which aspects offer a less ambiguous mapping to that volume. These aspects, ranked in decreasing order of their upward conditional probabilities, offer an effective means of disambiguating a given view of a volume.

When we want to move the camera in a direction to get a "better" view, we first check the APG to see which aspects (neighboring nodes) can be reached from the current aspect (node). The probabilities associated with the APG nodes tell us to which aspect to move in order to achieve a more likely view of the volume or to disambiguate it. The arc to this "best" neighbor node encodes the view change direction (in the image plane) in terms of a function of the normals of selected aspect contours. We calculate the values of these normals in the image and get a direction for camera movement with respect to the current aspect.

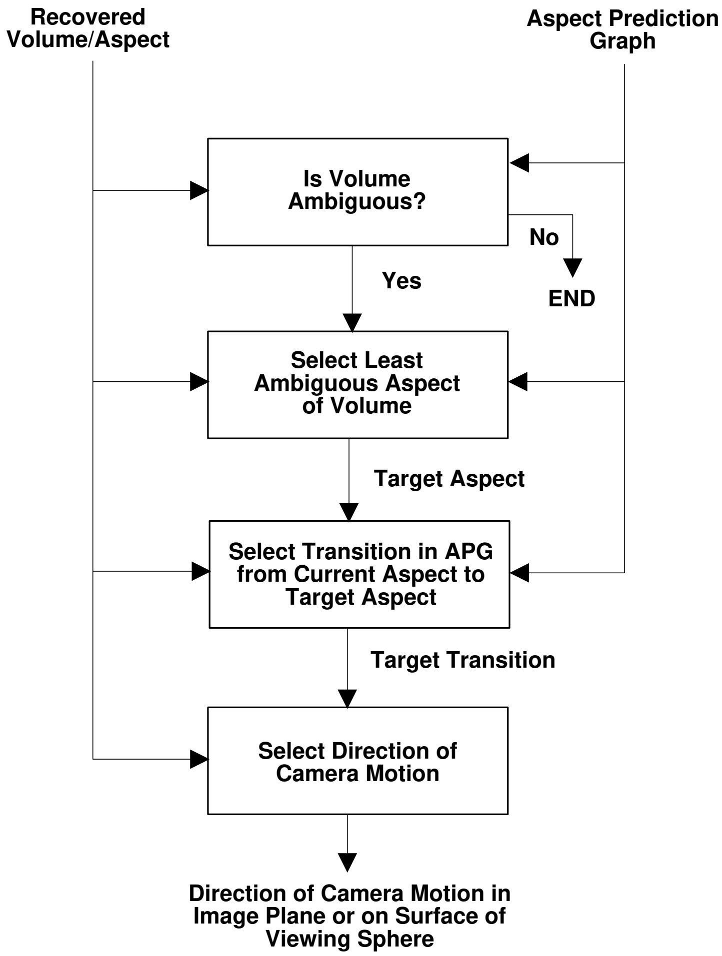


Figure 10: The Strategy for Moving the Camera

Our strategy for moving the camera is summarized in Figure 10. Note that if a recovered volume (and its associated recovered aspect) is unambiguous, no camera motion is computed. If the recovered aspect is determined to be ambiguous (by looking at the upward conditional probabilities mapping the recovered aspect to the target volume), then we select from among the aspects belonging to the target volume that aspect, called the target aspect, whose conditional probability to the target volume is maximized. Given the recovered aspect and the target aspect, we can extract from the aspect prediction graph the transition that takes us from the recovered aspect to the target aspect. Due to the compactness of our part-based aspect graphs, a single aspect transition, called the target transition, is sufficient. Finally, given the target transition and the recovered aspect, we compute the direction of camera motion in the image plane or on the surface of the viewing sphere (given an estimate of the distance to the object). This direction is specified by the transition as a function of the normal directions of the contours comprising the faces in the recovered aspect.

Finally, it should be noted that our strategy for moving the camera is based on disambiguating a single object part. Due to occlusion by other parts of the object or even other objects, the ambiguous part may not even be visible from the new viewpoint. In this case, the next least ambiguous view should be chosen from the APG, and movement to that view should be planned. If, in the worst case, the ambiguous view is the only unoccluded view of the object, then the attention mechanism must choose another volume to verify the presence of the hypothesized object.

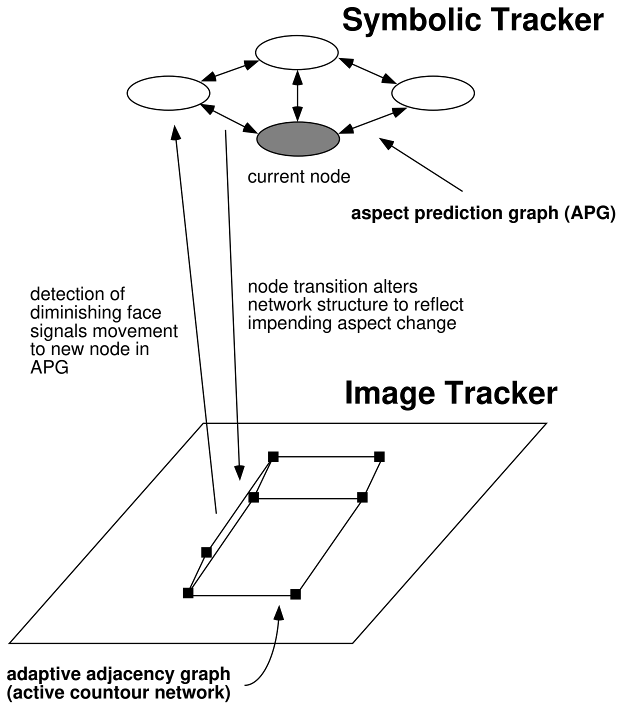


Figure 11: Qualitative Object Tracking

6.4 Tracking a Volume Across Views

While moving the camera, we must track the aspect from one frame to the next so that we can verify the visual events as specified in the APG. Since we have not recovered the part's 3-D geometry, we need some way of qualitatively tracking the volume in the image. Our approach to qualitative object tracking, as shown in Figure 11, combines a symbolic tracker and an image tracker, which we briefly discuss below; details can be found in [9].

The symbolic tracker tracks movement from one node to another in the aspect prediction graph. For our viewpoint control strategy, we begin at the node in the APG representing the ambiguous view (current node). As we move the camera, we will compare the visual events detected by the image tracker to those events predicted to appear as we move from the current node to the APG node chosen to disambiguate the volume (target node). From the visual event specification defined by the arc spanning the current and target nodes, we will add or delete structure from the image tracker. If new face events predicted to appear by the symbolic tracker cannot be verified by the image tracker, or the image tracker detects disappearing face events not predicted by the symbolic tracker, the recovered ambiguous aspect does not represent a target volume in the image.

The image tracker employs a representation called an adaptive adjacency graph, or AAG. The AAG is initially created from the recovered (ambiguous) aspect, and consists of a network of active contours (snakes) [22]. In addition, the AAG encodes the topology of the network's regions, as defined by minimal cycles of contours. Contours in the AAG can deform subject to both internal and external (image) forces while retaining their connectivity at nodes. Connectivity of contours is achieved by imposing constraints (springs) between the contour endpoints. If an AAG detected in one image is placed on another image that is slightly out of registration, the AAG will will be "pulled" into alignment using local image gradient forces.

The basic behavior of the AAG is to track image features while maintaining connectivity of the contours and preserving the topology of the graph. This behavior is maintained as long as the positions of active contours in consecutive images do not fall outside the zones of influence of tracked image features. This, in turn, depends on the number of active contours, the density of features in the image, and the disparity between successive images. If either the tracked object or the camera moves between successive frames, the observed scene may change due to disappearance of one of the object faces. The shape of the region corresponding to the disappearing face will change and eventually the size of the region will be reduced to zero. The image tracker monitors the sizes and shapes of all regions in the AAG and detects such events. When such an event is detected, a signal describing the event is sent to the symbolic tracker.

Conversely, if the symbolic tracker has predicted the appearance of a new face, it will add active contours to the AAG to pick up the expected face. These contour will form a new cycle where the new face is expected to appear. If the contours do not lock on to an appearing face, i.e., its area does not increase due to decreasing face foreshortening, then the image tracker will signal to the symbolic tracker that the event could not be verified.

7 Results

We test the attention and viewpoint control strategies in the context of a multidisciplinary research effort exploring active vision in the domain of robotic aids for a disabled child [43]. Through a touch-screen interface, a child can instruct a mobile robot vision system to identify, localize, and manipulate 3-D objects in its environment. One of the ways the child can select an object for manipulation is through a set of object icons on the touch-screen. It is for this particular task that the attention and viewpoint control strategies are aimed. Once an object is found, it is highlighted in the image for the child to confirm. If the child rejects the highlighted object, implying that he/she wanted some other instance in the image, the search must continue for the next best object of the chosen class. To support simple manipulation of the objects, the domain of objects that the system can visually identify consists of the ten volumetric shapes outlined in Figure 1; more complex objects, modeled as constructions of the ten shapes, will be supported in the future. Thus, each object consists of one component volume. Finally, each of the ten objects is assumed to be equally likely.

7.1 Attention

In Figure 12, we present the results of applying the attention mechanism to a scene containing single-volume objects. In this case, the child has selected the "block" icon, instructing the system to find the best instance of a block in the image; once found, the instance is displayed to the child. The child can then confirm that instance as the one they desire, or command the system to continue looking for the next best instance, and so on. Moving top to bottom and left to right, the first image shows the results of the region segmentation step; recall that the face topology graph constructed from the region topology graph is the input to the attention mechanism. The next three images show the three best instances of the block viewed in its most likely aspect containing three faces. The faces in the aspect are highlighted in the image. Furthermore, only those contours (boundary group) used in defining the face are highlighted in the face.

Using Equation 14, the first three volumes received the score of 1.0, 1.0, and 0.86, respectively. In the third case, region undersegmentation results in the merging of regions from two blocks. The resulting region does not exactly match the component face of the block aspect. However, a strong inference to that face can be made from the boundary group which is highlighted. In the next four figures, we search for the block given its next best aspect, i.e., that consisting of two faces. The scores of the best three volumes are 0.48, 0.48, and 0.48, respectively. In the fourth case, we show a lower-scoring volume (score = 0.33), which is clearly incorrect. During the search process, both low probability predictions and low scoring recovered features (faces, aspects, and volumes) can be pruned, resulting in only high quality volumes being recovered. The next three images show the three best volumes given the lowest probability aspect, i.e., the aspect containing one face. The scores of these three volumes are 0.22, 0.22, and 0.22, respectively. Finally, in the last figure, we show the result of searching for the best instance of a cylinder (the only other volume in the image); the score of the cylinder is 0.31.

Figure 13(b) illustrates the results of applying the attention algorithm to find the best instance of the truncated pyramid (volume 2 in Figure 1) from the segmented region image in Figure 13(a). In, Figure 13(c) the best instance of the cylinder (volume 5) is shown. Finally, in Figure 13(d), the attention mechanism has been directed to find the best instance of the barrel (volume 8). Due to region undersegmentation, the end face of the volume was merged with the body face. Although the body face was matched to the most likely aspect of the volume, the end face is assumed to be occluded at the bottom of the recovered volume.

In Figure 14, we illustrate the search for a multi-part object. Figure 14(a) shows the original image of a coffee cup, while Figure 14(b) shows the results of the region segmentation. The most discriminating part of the cup (given a small database containing a cup, a hammer, and the ten single-part objects corresponding to the ten volumes) is the bent cylinder used to model the handle. The search algorithm then searches for the best instance of the bent cylinder in the image, shown in Figure 14(c). Then, at regions adjacent to the regions encompassed by the bent cylinder, the search algorithm searches for the best instance of the remaining part of the cup, i.e., cylinder, as shown in Figure 14(d). If, for example, bent cylinders were found at many locations in the image, then subsequent search for remaining object parts would take place at each of these locations using a breadth-first search.

Figure 15 gives another example of multi-part object search. Figure 15(a) shows the original image of a hammer, while Figure 15(b) shows the region segmented image. Since both parts of our hammer (handle and head) are modeled as cylinders, we have no choice but to search for a cylinder in the image. The best cylinder instance is shown in Figure 15(c), with only a portion of the cylinder being recovered in its most likely aspect. Search for the remaining object part is focused only at regions adjacent to the first volume. The best connected cylinder is shown in Figure 15(d), recovered in its second most likely aspect (parallelogram). Note that what we've found is the best instance of a pair of connected cylinders in the image. Until we reason about the connections between the two cylinders, e.g., which one is connected at its side and which is connected at its end, we don't know at this point which is the handle and which is the head.

7.2 Viewpoint Control

Figure 16(a) presents the results of searching for a block (volume 1) in the image. Although the most probable aspect could not be recovered, the second most probable aspect (containing two faces) was recovered. This aspect is ambiguous (projection of block (volume 1) and bent block (volume 4)). Since we are searching for the block, we use the recovered aspect to position ourselves in the aspect prediction graph. Using the aspect probabilities encoded in the aspect prediction graph, the system knows which aspect should be recovered to disambiguate the aspect. In addition, the arc between the recovered aspect and the target aspect (in the aspect prediction graph) encodes in which direction the sensor should be moved (in the image plane) in order to encounter the target

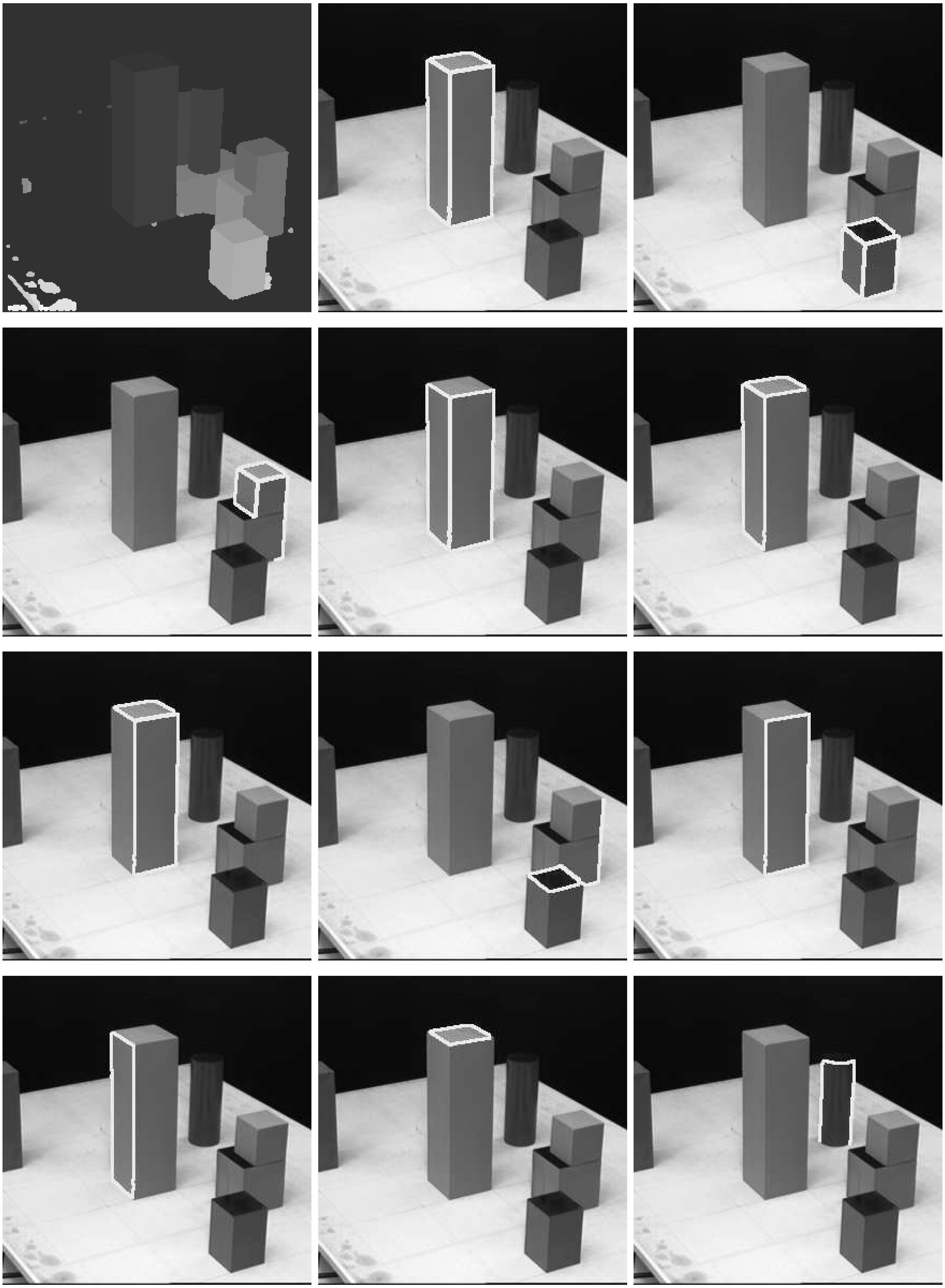


Figure 12: Searching for Volumes 1 (block) and 5 (cylinder) (see text for explanation)

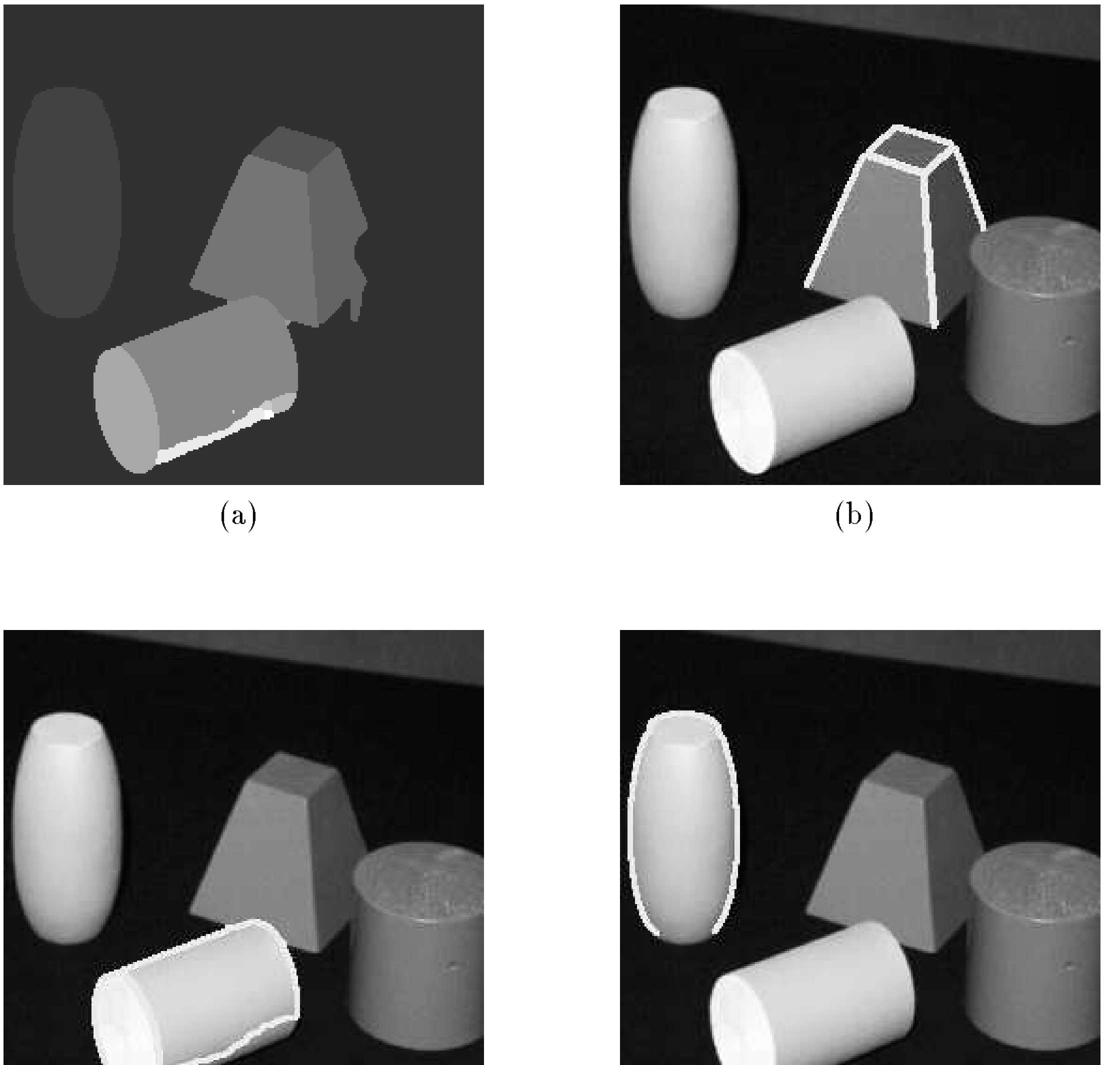


Figure 13: Searching for Volumes 2 (truncated pyramid), 5 (cylinder), and 8 (barrel)

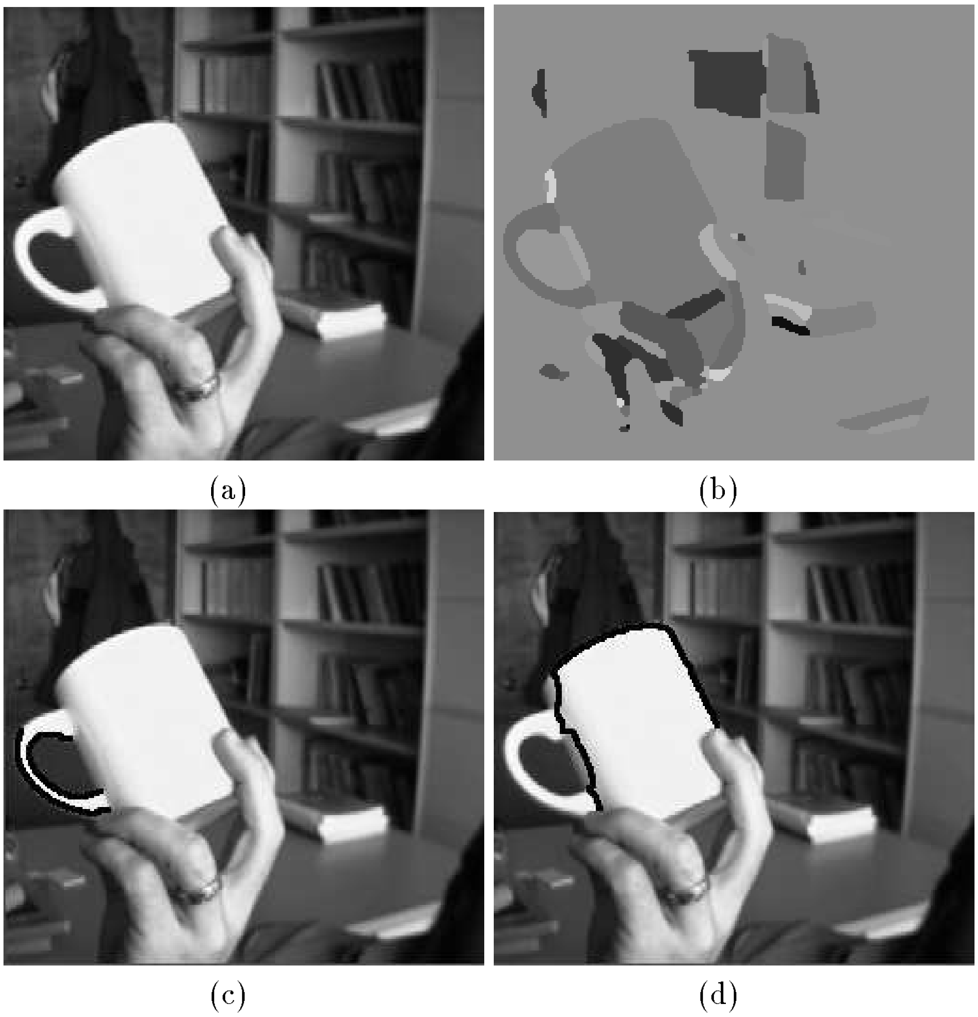
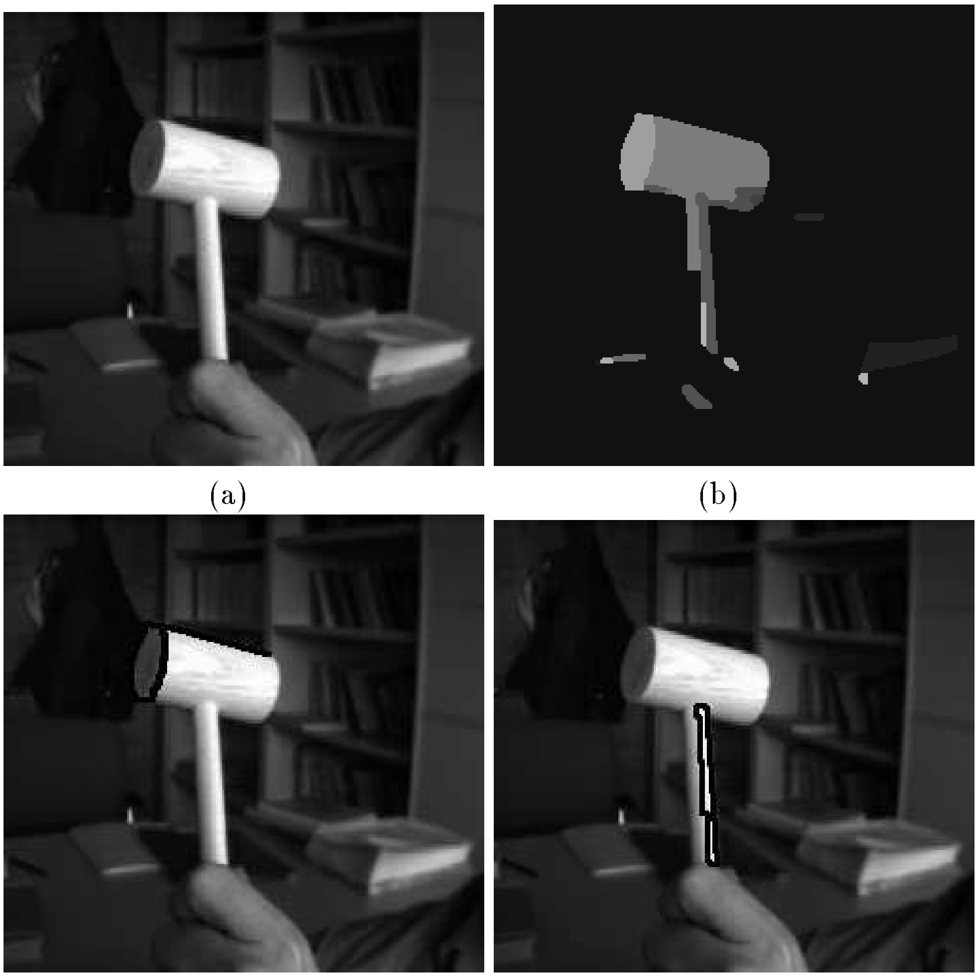
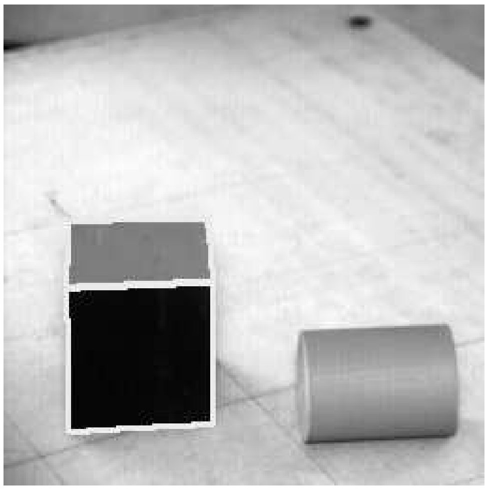


Figure 14: Searching for a coffee cup in an image: (a) original image; (b) region segmented image; (c) best instance of the cup's most discriminating part (bent cylinder); and (d) best instance of the cup's body (cylinder) adjacent to the handle.

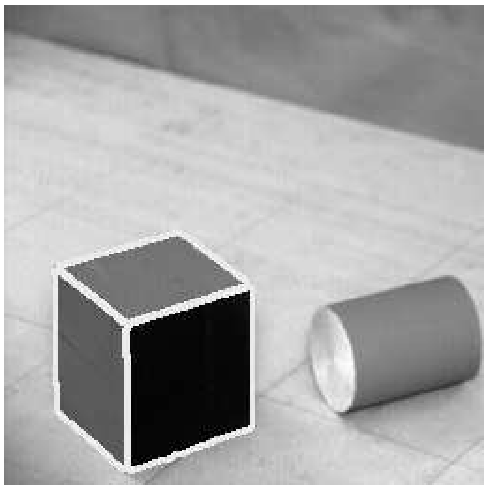


(a) (b) (c) (d)

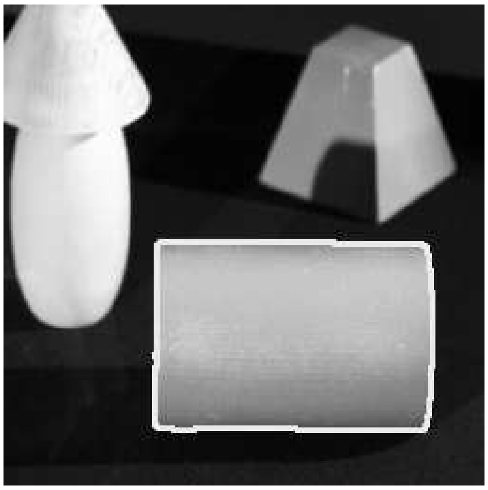
Figure 15: Searching for a hammer cup in an image: (a) original image; (b) region segmented image; (c) best instance of a hammer part; (d) best instance of a second hammer part adjacent to the part in (b).



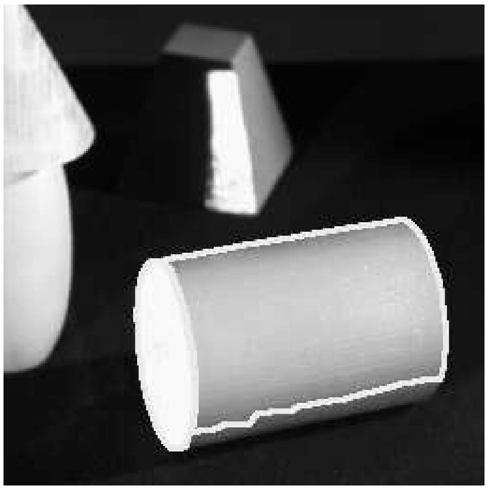
(a)



(b)



(c)



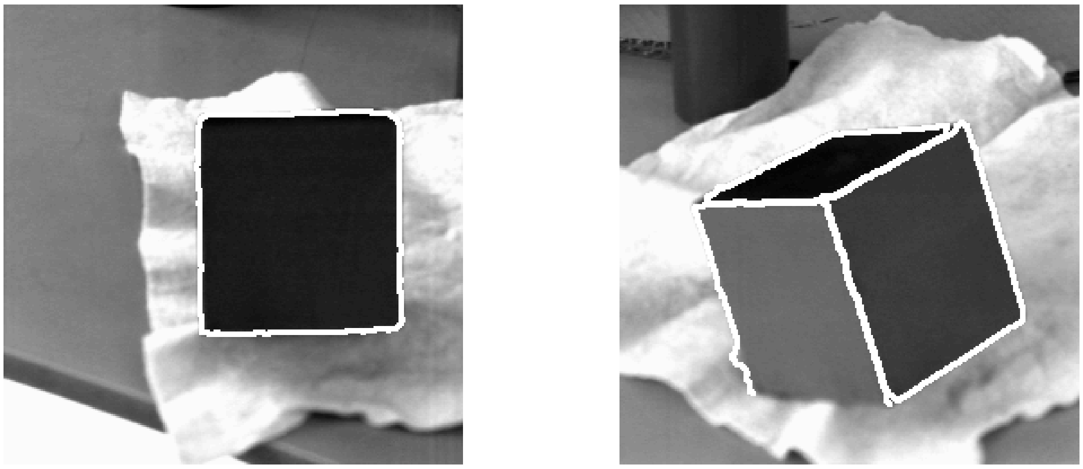
(d)

Figure 16: Moving the Sensor to Disambiguate Volumes 1 (block) and 5 (cylinder)

aspect. In this case, we can move either to the left or the right to bring a new face into view. We arbitrarily move to the left along the surface of a fixed-size view sphere and obtain the view shown in Figure 16(b).

For these experiments, a much simpler tracking mechanism was used, in which we assume that the position of the aspect in the image does not change significantly in relation to the sizes of its regions. To verify the target aspect (highlighted in the image), we invoked the attention mechanism and restricted it to those regions in the new image that intersect with those regions in the old image defining the ambiguous aspect. In future work, we will integrate the active contour tracker reported in [9]. In a second example, shown in Figure 16(c), a cylinder is recovered in its second most likely aspect (common to volumes 1, 2, 3, 4, 5, and 10). Guided by the aspect prediction graph, the camera is moved to the left and the attention scheme is guided to disambiguate the volume by searching for its most likely aspect, as is shown in Figure 16(d).

A third example is shown in Figure 17. Searching for the block volume yields a recovered aspect containing a single face (Figure 17(a)). Since the aspect is ambiguous, the system, in trying to



(a) (b)

Figure 17: Moving to the Least Ambiguous Aspect

verify the block volume, can choose to move either towards one of the sides or towards one of the corners. Moving to the side would would reveal the two-faced aspect of the block volume, which is still ambiguous. The system therefore moves towards a corner to reveal the unambiguous aspect of the block (Figure 17(b)).

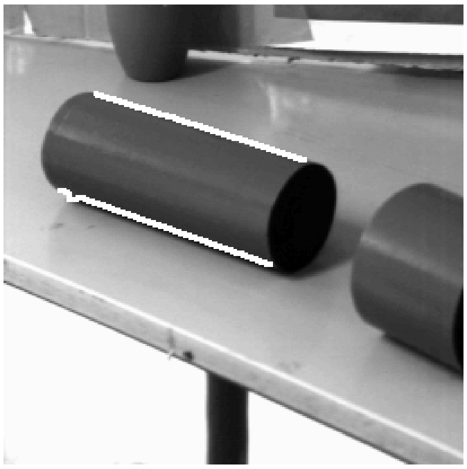
In the final example, shown in Figure 18, we again begin by searching for a block volume. Once more, the best recovered block appears as the single face aspect which is ambiguous, as shown in Figure 18(a). Moving in the direction of a corner, the system attempts to verify the unambiguous aspect of the block. As shown in Figure 18(b), the results of the verification are very weak, with only a portion of the original face contributing towards the unambiguous aspect being sought. Since the score of the recovered aspect falls far short, the verification fails, and it is concluded that the highlighted object is not a block. At this point, two options are available. The bottom-up shape recovery strategy, as outlined in [14], can be applied to both frames, with the added constraint that the recovered aspect in the two frames must be consistent with a single volume. Alternatively, the attention strategy can be applied using volumes whose aspects include the aspect recovered in Figure 18(a). Figures 18(c) and (d) show the search for the cylinder volume in the second frame; two different groupings give rise to the verified cylinder.

In choosing which alternative volumes to search for when invoking the attention mechanism, one can order the search according to the conditional probabilities mapping aspects to volumes. Using this ordering, both the tapered block and pyramid have a higher aspect-to-volume mapping (between the single parallelogram face and their respective volumes) than the cylinder. The algorithm would therefore attempt to verify these two volumes before attempting to verify the cylinder volume. The cost of such a sequential search must be compared to the cost of a bottom-up interpretation focused at the face where the original aspect was recovered.

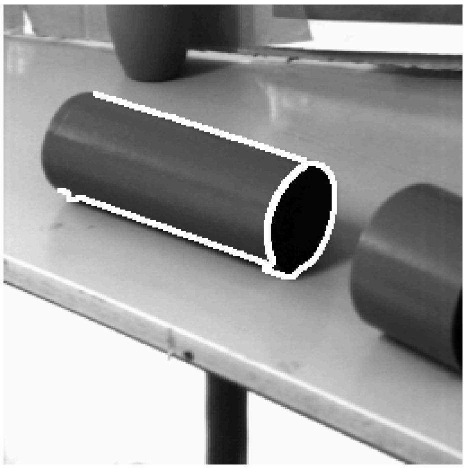
8 Limitations

Although the augmented aspect hierarchy can be computed for any set of volumetric parts which project to collections of faces (aspects), our representation does assume that objects can be represented as constructions of volumetric parts. Although this does cover a large class of man-made and natural objects, there are many object classes for which this modeling strategy is inappropriate. In

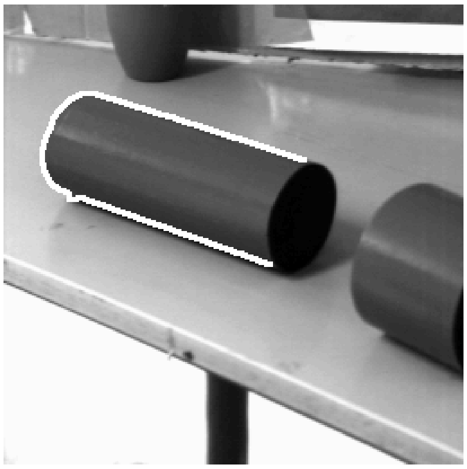
(a)



(b)



(c)



(d)

Figure 18: Failing to Verify a Volume Through Viewpoint Control

addition, the Bayesian formulation used in the attention mechanism assumes a closed world consisting of objects constructed only from the set of volumetric parts defining the augmented aspect hierarchy. Thus, in a domain where objects not contained in our database appear in the image, the bottom-up conditional probabilities used in computing feature utility may be inaccurate. The active contour network tracking mechanism does not currently support occluded parts. For real objects composed of multiple parts, part occlusion is inevitable, and we are currently extending our tracker to handle region deformations due to occlusion. Perhaps the most significant limitation of our system is its reliance on salient homogeneous regions in the image that can be easily segmented. Real objects contain a plethora of detail that must somehow be abstracted out when recovering the face components of an aspect. We are currently investigating methods by which such detail can be filtered out of the face topology graph.

9 Conclusions

A major trade-off in object recognition systems is the balance between effort expended in feature recovery and effort expended in verification. Many approaches to recognition shift this balance towards verification, requiring accurate pose estimation in order to verify weak indexing features. Furthermore, since the recovered features are so common in the image, many possible correspondences between image and model features must be hypothesized leading to high computational complexity. In this paper, we have shown that regions offer much less uncertainty in hypothesising objects. Through a set of probabilities derived from a statistical analysis of a set of volumetric parts over the viewing sphere, we have presented an attention mechanism which can focus the search for an object at a much smaller set of locations in the image. Moreover, these locations can be evaluated and ranked, allowing a search to begin at more likely locations.

In examining the balance between recovery and verification, we have clearly moved towards recovery. Although more discriminating features mean less uncertainty and lower search complexity, there is a cost in attempting to recover more complex features (in our case, a set of regions and their bounding shapes). Our solution to this problem is to pass along this cost to a dynamic sensor. We assume that some relatively unoccluded, fronto-parallel surfaces will project into regions that can be quickly and cheaply extracted using simple region segmentation techniques, as is demonstrated in section 7. We use this limited knowledge to intelligently guide the sensor to a position where the object can be disambiguated.

The ability of a vision system to move to a new location in order to disambiguate a view of an object enhances its ability to recover and recognize objects. To provide a vision system with this capability, we must address a number of important questions, including: How do we know that a given view is ambiguous?; What view is less ambiguous?; How do we move the camera system to encounter the less ambiguous view?; and finally: What should we look for as we move? The approach proposed in this paper addresses the above four questions by combining a probabilistic augmented aspect hierarchy, encoding object (part) views and their likelihoods, with a highly compact aspect graph, encoding aspect transitions and visual event specifications. The resulting representation effectively unifies the processes of attention and viewpoint control, providing a more integrated approach to active object recognition.

10 Acknowledgements

Sven Dickinson and John Tsotsos would like to thank Gene Amdur, Lars Olsson, Winky Wai, Lars Olsson, James Maclean, Sean Culhane, Suzanne Stevenson, Martin Martin, Yiming Ye, and Feng Lu for assisting in the implementation of these ideas. John Tsotsos is the CP-Unitel Fellow of the Canadian Institute for Advanced Research. This research was funded the Institute for Robotics and Intelligent Systems, a Network of Centers of Excellence of the Government of Canada, and the Natural Sciences and Engineering Research Council of Canada. Henrik Christensen would like to thank Steen Kristensen for assisting in the implementation, and Finn V. Jensen for a number of constructive discussions. The work by Henrik Christensen was partly funded by the ESPRIT BRA Project "Vision as Process" (BR-7108); this funding is gratefully acknowledged. The work by Göran Olofsson has been performed within the ESPRIT-BRA project "Vision As Process, VAP" (BR 7108). The support from The Swedish Board for Technical and Industrial Development, NUTEK is also gratefully acknowledged. Sven Dickinson gratefully acknowledges the support of the National Science Foundation (NSF CAREER IRI-9623913).

References

[1] R. Bergevin and M. D. Levine. Generic object recognition: Building coarse 3D descriptions from line drawings. In *Proceedings, IEEE Workshop on Interpretation of 3D Scenes*, pages 68–74, Austin, TX, 1989.

[2] I. Biederman. Human image understanding: Recent research and a theory. *Computer Vision, Graphics, and Image Processing*, 32:29–73, 1985.

[3] K. Brunnström, T. Lindeberg, and J. Eklundh. Active detection and classification of junctions by foveation with a hand-eye system guided by the scale-space primal sketch. Technical Report ISRN KTH/NA/P-91/31-SE, Computer Vision and Active Perception Laboratory (CVAP), Royal Institute of Technology, Stockholm, Sweden, 1991.

[4] Andrea Califano, Rick Kjeldsen, and Ruud M. Bolle. Data and model driven foviation. In *Proceedings, IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–7. IEEE CS Press, June 1990.

[5] S. Culhane and J. Tsotsos. An attentional prototype for early vision. In G. Sandini, editor, *Second European Conference on Computer Vision*, LNCS-Series Vol. 588, pages 551–560, Santa Margherita Ligure, Italy, May 1992. Springer-Verlag.

[6] S. Culhane and J. Tsotsos. A prototype for data-driven visual attention. In *11th ICPR*, pages 36–40, The Hague, August 1992.

[7] S. Dickinson. The recovery and recognition of three-dimensional objects using part-based aspect matching. Technical Report CAR-TR-572, Center for Automation Research, University of Maryland, 1991.

[8] S. Dickinson, I. Biederman, A. Pentland, J.-O. Eklundh, R. Bergevin, and R. Munck-Fairwood. The use of geons for generic 3-D object recognition. In *Proceedings, International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1693–1699, Chambery, France, August 1993.

[9] S. Dickinson, P. Jasiobedzki, H. Christensen, and G. Olofsson. Qualitative tracking of 3-D objects using active contour networks. In *Proceedings, IEEE Conference on Computer Vision and Pattern Recognition*, Seattle, June 1994.

[10] S. Dickinson and D. Metaxas. Integrating qualitative and quantitative shape recovery. *International Journal of Computer Vision*, 13(3):1–20, 1994.

[11] S. Dickinson and A. Pentland. A unified approach to the recognition of expected and unexpected geon-based objects. In *Proceedings, SPIE Applications of AI X: Machine Vision and Robotics, special session on "Recognition by Components"*, Orlando, FL, April 1992.

[12] S. Dickinson, A. Pentland, and A. Rosenfeld. A representation for qualitative 3-D object recognition integrating object-centered and viewer-centered models. In K. Leibovic, editor, *Vision: A Convergence of Disciplines.* Springer Verlag, New York, 1990.

[13] S. Dickinson, A. Pentland, and A. Rosenfeld. From volumes to views: An approach to 3-D object recognition. *CVGIP: Image Understanding*, 55(2):130–154, 1992.

[14] S. Dickinson, A. Pentland, and A. Rosenfeld. 3-D shape recovery using distributed aspect matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):174–198, 1992.

[15] R. Fairwood. Recognition of generic components using logic-program relations of image contours. *Image and Vision Computing*, 9(2):113–122, 1991.

[16] W. Grimson and T. Lozano-Pérez. Model-based recognition and localization from sparse range or tactile data. *International Journal of Robotics Research*, 3(3):3–35, 1984.

[17] J. Hummel and I. Biederman. Dynamic binding in a neural net model for shape recognition. *Psychological Review*, 99:480–517, 1992.

[18] S. Hutchinson and A. Kak. Planning sensing strategies in a robot work cell with multi-sensor capabilities. *IEEE Transactions on Robotics and Automation*, 5(6):765–783, December 1989.

[19] D. Huttenlocher and S. Ullman. Recognizing solid objects by alignment with an image. *International Journal of Computer Vision*, 5(2):195–212, 1990.

[20] A. Jacot-Descombes and T. Pun. A probabilistic approach to 3-D inference of geons from a 2-D view. In *Proceedings, SPIE Applications of Artificial Intelligence X: Machine Vision and Robotics*, pages 579–588, Orlando, FL, 1992.

[21] F. Jensen, H. Christensen, and J. Nielsen. Baysian methods for interpretation and control in multiagent vision systems. In K. Bowyer, editor, *SPIE Applications of AI X: Machine Vision and Robotics*, volume 1708, pages 536–548, Orlando, FL, April 1992.

[22] M. Kass, A. Witkin, and D. Terzopolous. Snakes: Active contour models. *Internation Journal of Computer Vision*, 1(4):321–331, 1988.

[23] H.-S. Kim, R. Jain, and R. Volz. Object recognition using multiple views. In *Proceedings, IEEE International Conference on Robotics and Automation*, pages 28–33, St. Louis, MO, March 1985.

[24] C. Koch and S. Ullman. Shifts in selective visual attention. *Human Neurobiology*, 4:219–227, 1985.

[25] J. Koenderink and A. van Doorn. The internal representation of solid shape with respect to vision. *Biological Cybernetics*, 32:211–216, 1979.

[26] Y. Lamdan, J. Schwartz, and H. Wolfson. On recognition of 3-D objects from 2-D images. In *Proceedings, IEEE International Conference on Robotics and Automation*, pages 1407–1413, Philadelphia, PA, 1988.

[27] J. Lee, R. Haralick, and L Shapiro. Morphologic edge detection. *IEEE Journal of Robotics and Automation*, RA-3(2):142–155, 1987.

[28] T. Levitt, J. Agosta, and T. Binford. Model based influence diagrams for machine vision. In M. Herion, R. Shacter, L. Kanal, and J. lemmer, editors, *Uncertainty in Artificial Intelligence 5*, volume 10 of *Machine Intelligence and Pattern Recognition Series*, pages 371–388. North Holland, 1990.

[29] M. Li. Minimum description length based 2-D shape description. Technical Report CVAP114, Computational Vision and Active Perception Lab, Royal Institute of Technology, Stockholm, Sweden, October 1992.

[30] D. Lowe. *Perceptual Organization and Visual Recognition.* Kluwer Academic Publishers, Norwell, MA, 1985.

[31] J. Matas, P. Remnagnino, J. Kittler, and J. Illingworth. Control of scene interpretation. In J. Crowley and H. Christensen, editors, *Vision as Process.* Springer-Verlag, January 1995.

[32] J. Maver and R. Bajcsy. How to decide from the first view where to look next. In *Proceedings, DARPA Image Understanding Workshop*, pages 482–496, Pittsburgh, PA, September 1990.

[33] J. Pearl. *Probabilistic Reasoning in Intelligent Systems.* Morgan Kaufmann Publishers, Inc., 1987.

[34] A. Pentland. Perceptual organization and the representation of natural form. *Artificial Intelligence*, 28:293–331, 1986.

[35] N. Raja and A. Jain. Recognizing geons from superquadrics fitted to range data. *Image and Vision Computing*, 10(3):179–190, April 1992.

[36] R. Rimey and C. Brown. Where to look next using a bayes net: Incorporating geometric relations. In G. Sandini, editor, *European Conference on Computer Vision (ECCV)*, volume 588 of *Lecture Notes in Computer Science*, pages 542–550. Springer-Verlag, May 1992.

[37] P. Saint-Marc, J.-S. Chen, and G. Medioni. Adaptive smoothing: A general tool for early vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(6):514–529, 1991.

[38] L. Stark and K. Bowyer. Achieving generalized object recognition through reasoning about association of function to structure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(10):1097–1104, 1991.

[39] D. Thompson and J. Mundy. Model-directed object recognition on the connection machine. In *Proceedings, DARPA Image Understanding Workshop*, pages 93–106, Los Angeles, CA, 1987.

[40] J. Tsotsos. Representational axes and temporal cooperative processes. In M. Arbib and A. Hanson, editors, *Vision, Brain and Cooperative Computation*, pages 361–418. MIT Press / Bradford Books, 1987.

[41] J. Tsotsos. A complexity level analysis of vision. *Behavioral and Brain Sciences*, 13(3):423–455, 1990.

[42] J. Tsotsos. An inhibitory beam for attentional selection. In L. Harris and M. Jenkin, editors, *Spatial Vision in Humans and Robots*. Cambridge University Press, 1993.

[43] J. Tsotsos, S. Dickinson, M. Jenkin, E. Milios, A. Jepson, B. Down, E. Amdur, S. Stevenson, M. Black, D. Metaxas, J. Cooperstock, S. Culhane, F. Nuflo, G. Verghese, W. Wai, D. Wilkes, and Y. Ye. The playbot project. In *Workshop on AI Applications for Disabled People (held in conjunction with the 14th International Joint Conference on Artificial Intelligence (IJCAI))*, Montreal, August 1995.

[44] D. Wilkes, S. Dickinson, and J. Tsotsos. A quantitative analysis of view degeneracy and its application to active focal length control. In *Proceedings, International Conference on Computer Vision*, Cambridge, MA, June 1995.

[45] D. Wilkes and J. Tsotsos. Active object recognition. In *Proceedings, Computer Vision and Pattern Recognition '92*, Urbana, IL, June 1992.