Viewpoint-Invariant Indexing for Content-Based Image Retrieval

Sven Dickinson
Dept. of Comp. Sci. and
Center for Cognitive Science
Rutgers University
New Brunswick, NJ 08903

Alex Pentland
Vision and Modeling Group
Media Laboratory
MIT
Cambridge, MA 02139

Suzanne Stevenson
Dept. of Comp. Sci. and
Center for Cognitive Science
Rutgers University
New Brunswick, NJ 08903

Abstract

Current methods for shape-based image retrieval are restricted to images containing 2-D objects. We propose a novel approach to querying images containing 3-D objects, based on a view-based encoding of a finite domain of 3-D parts used to model the 3-D objects appearing in images. To build a query, the user manually identifies the salient parts of the object in a query image. The extracted views of these parts are then used to hypothesize the 3-D identities of the parts which, in turn, are used to hypothesize other possible views of the parts. The resulting set of part views, along with their spatial relations (constraints) in the query image, form a composite query that is passed to the image database. Images containing objects with the same parts (in any view) with similar spatial relations are returned to the user. The resulting viewpoint invariant indexing technique does not require training the system for all possible views of each object. Rather, the system requires only knowledge of the possible views for a finite vocabulary of 3-D parts from which the objects are constructed.

1 Introduction

Current methods for shape-based image retrieval are restricted to 2-D shape, e.g., [1, 11, 17, 8, 18, 19]. For example, consider a user who is observing an image of an automobile where the automobile is viewed from the side. Next, the user outlines the automobile in the image and asks the system to return images containing similar objects. The system will interpret similar as meaning "looks the same", and will return (at best) only images containing cars that are viewed from the side. Images of cars viewed from the front, back, top, or bottom will *not* be returned. Although the objects are similar, the views are not. The system has no comprehension that similar 3-D objects can appear differently, depending on viewpoint.

For content-based image retrieval to be viewpoint-invariant requires that we have some way of associating together the various views of an object. In our automobile example, the user-selected view of the car should somehow be associated with other views of the car, so that indexing is effectively performed with a set of view classes of the car. This would mean that the system would somehow have to know about all possible views of a car, so that the user-selected view could index to the set of car views. Each of the car views could then be passed as a separate query to the image database.

This approach is impractical for two reasons. First, how is the system to acquire the various views of the car, or any other object for that matter? Such an approach would mean training the system on all possible views of each object that might be contained in a query image. Although possible, this would be an extremely tedious and time-consuming process. Furthermore, novel objects selected by the user would therefore not map to a set of views with which to index the image database, precluding a search for objects which might be similar in shape. The second problem is due to the large number of views required to encode a 3-D object. For example, in the system of Murase and Nayar, 72 views are required to sample a single line of latitude around the viewing sphere centered on an object [16]. Even if we were somehow able to encode a set of definitive views for each object, the number of queries to the image database that we would need to make (one for each view) would be intractable for complex objects.

An alternative approach to storing all possible 2-D views of a 3-D object is to store a 3-D model of the object. Using a more traditional 3-D from 2-D recognition framework, e.g., [15, 10, 13], the user-selected view of the object could be recognized from a database of 3-D objects. Preprocessing of the image database would yield a table indexing objects to images, so that a recognized object in the query image would quickly

0-8186-8329-5/97 $10.00

return a set of images containing that object. Unfortunately, this approach is equally infeasible, for it requires that a 3-D model exist for every possible query object. How is such a model acquired, and what form should such a model take?

In this paper, we offer a novel viewpoint-invariant approach to shape indexing, which supports indexing using a set of view classes, but avoids the training and complexity issues outlined above. The technique is based on our previous work in object representation, recovery, and recognition using part-based aspect matching [5, 7, 6, 4]. The underlying assumption is that views of objects can be broken down into smaller, simpler views of a finite, albeit large, set of volumetric part classes. The number of part classes, as well as their views, is therefore independent of the number of objects that are expected to appear. The views are precomputed off-line and stored in a secondary, part-view database.

Imagine the following scenario. Instead of outlining the entire object in a query image, the user would identify two or three of the object's constituent parts. The views of these parts would then be matched to the part view database containing the view classes for a large set of simple, volumetric parts. A match between a user-identified view and a view from the part view database yields the 3-D identity of the part, from which its other possible views can be enumerated. The resulting set of view hypotheses are then passed as a composite query to the image database. If a user-identified part view is ambiguous, i.e., if it maps to more than one volumetric part, then multiple sets of view classes, one set per part hypothesis, can be matched to the image database.

To account for the fact that objects are composed of multiple parts, we must encode the spatial relationships between the user-selected parts on the object, and relate them to the results of the image database query. Simple connectedness in the image is sufficient, possibly resulting in false positive matches being returned to the user. For example, if two user-selected parts are connected on the object, then we will assume that for any database image in which both parts are visible, their respective (part) views will be adjacent in the image.

Finally, one essential requirement of the view-based parts representation is that the views be invariant to minor deformations in the shapes of the parts. For example, if a part's dimensions (or relative dimensions) change, or the curvature of the part changes, the set of views describing the part should be invariant to such deformations. Therefore, the views must be defined qualitatively to support the expected variability of the parts appearing in the images.

2 A Parts-Based View Representation

The hybrid representation we use to describe objects draws on two prevalent object representation schools in the computer vision community. The first school is called object-centered modeling, wherein 3-D object descriptions are invariant to changes in their position and orientation with respect to the viewer. The second school is called viewer-centered modeling, wherein an object description consists of the set of all possible 2-D views of an object, often linked together to form an aspect graph. Object-centered models are compact, but their recognition from 2-D images requires making 3-D inferences from 2-D features. Viewer-centered models, on the other hand, reduce the recognition problem from three dimensions down to two, but incur the cost of having to store many different views for each object.

In order to meet the goals of qualitative object modeling and matching, we first model objects as object-centered constructions of qualitatively-defined volumetric parts chosen from some arbitrary, finite set [5]. It is at the volumetric part modeling level that we invoke the concept of viewer-centered modeling. Traditional aspect graph representations of 3-D objects model an entire object with a set of aspects (or views), each defining a topologically distinct view of an object in terms of its visible surfaces [12]. Our approach differs in that we use aspects to represent a finite set of volumetric parts from which objects appearing in our image database are constructed, rather than representing the entire object directly. The resulting set of part aspects is therefore *independent* of the number of objects in the database.

By having a sufficiently large set of volumetric part building blocks, and by assuming that objects appearing in the image database can be composed from this set, our training phase which computes the part views is independent of the contents of the image database. It is important to note that the part vocabulary need not be complete, but only sufficient to describe interesting portions of most interesting objects. There is no reason to describe the whole object as long as we can describe enough of it to perform indexing.

A potential problem with our hybrid representation is that, if a volumetric part is occluded from a given 3-D viewpoint, its projected aspect in the image will also be occluded. We must therefore accommodate the matching of occluded aspects, which we accomplish by use of a hierarchical representation that we call the *aspect hierarchy*. The aspect hierarchy consists of

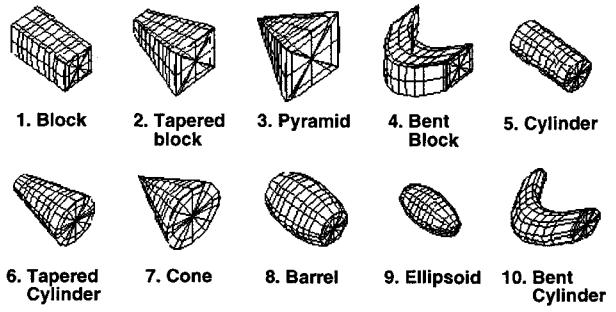


Figure 1: The ten modeling primitives.

three levels: 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 are captured in a set of upward and downward conditional probabilities [3], mapping boundary groups to faces, faces to aspects, and aspects to volumes. The probabilities are estimated from a frequency analysis of features viewed over a sampled viewing sphere centered on each of the part classes.

To illustrate the use of our hybrid representation in searching image databases, we begin with a small vocabulary of ten volumetric part classes, as shown in Figure 1 [5, 7, 6]. The set of all aspects that model the ten volumes constitutes the first level of the aspect hierarchy, part of which is shown in Figure 2; each aspect is represented by a graph in which nodes represent faces and arcs represent face adjacencies. The set of all faces that make up the aspects constitutes the second level of the aspect hierarchy, part of which is shown in Figure 2. It is important to note that a face captures *qualitative* relationships among *qualitatively-described* contours. The boundary of a face is partitioned at curvature discontinuities into a set of contours. Furthermore, each contour is classified as either straight, concave, or convex with respect to the face; exact curvature, contour lengths, and angles between adjacent contours are not represented. Each face can thus be represented as a graph in which nodes represent bounding contours of a face, and arcs represent certain nonaccidental contour relations, including parallelism, symmetry, and intersection.

The third and lowest level of the aspect hierarchy contains the boundary groups (see Figure 2), representing all subsets of the contours comprising the faces at the second level of the aspect hierarchy, i.e., all possible subgraphs of the graphs representing the faces. The boundary groups support the inference of faces

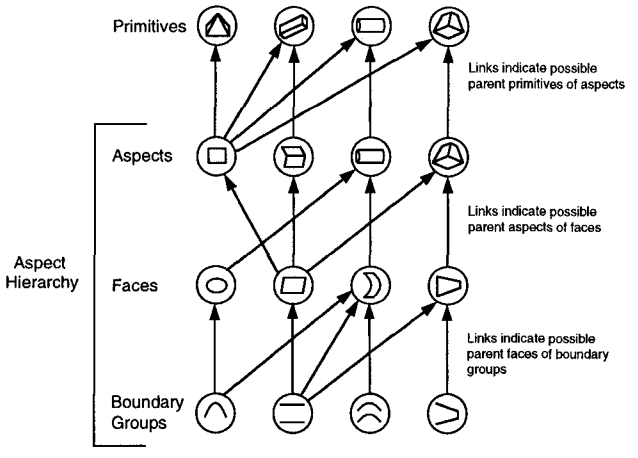


Figure 2: The aspect hierarchy.

from incomplete image data, e.g., due to noise or occlusion, and therefore play a fundamental role in the recovery of volumetric shapes from image data.

The sets of upward (and downward) conditional probabilities linking together the layers of the aspect hierarchy are derived through an empirical procedure. For a given part class, we model the part using a CAD system, step through all possible deformations of the part within the class, and for each resulting instance, generate the set of views of the part over a tesselated view sphere centered on the part. Counting each feature in each view gives rise to a set of frequency distributions which are used to estimate the conditional probabilities [7]. In previous work, the aspect hierarchy served as the backbone of both bottom-up and top-down recognition models [6, 4]. As we will show in the next section, this same representational framework can be applied to a part-based query mechanism that will ultimately support viewpoint-invariant shape indexing.

3 Generating a Search Index

At query time, the user outlines complex, unusual, or otherwise discriminating portions of a target object, with the goal being to return images from the image database that contain similarly-shaped objects. Using the mouse, the user will interactively draw the salient contours of an object's part over the image being displayed. If the part has multiple component faces (forming the aspect of the part), the user must outline each component face. For occluded or intersecting parts, the user may draw either the entire part (extrapolating to fill in what they believe to be the shape of the part) or simply that portion of the part which is visible.

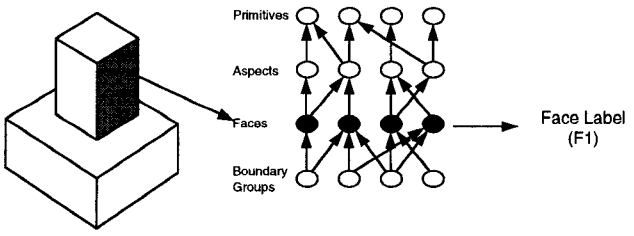


Figure 3: Labeling an unoccluded face.

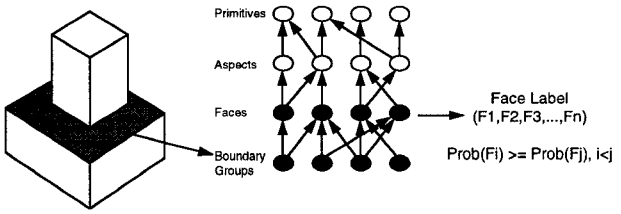


Figure 4: Labeling an occluded face.

For the selected part, the interactively-drawn part aspect will then be matched to the aspect hierarchy. Once a face is drawn, its shape is first described by the best-fitting set of straight lines, convex elliptical curves, and concave elliptical curves, according to a minimum description length algorithm [14]. The result is a graph in which nodes represent qualitatively described bounding contour sections, and arcs represent relations including cotermination, parallelism, and symmetry. The graph is then matched to the faces in the aspect hierarchy using an interpretation tree search [9]. Figure 3 illustrates an example where an identified face exactly matches a face in the aspect hierarchy.

If, due to occlusion or poorly-drawn contours, an outlined face doesn't match an aspect hierarchy face, then parts of the drawn face are matched to the boundary group level of the aspect hierarchy. From the bottom-up conditional probabilities in the aspect hierarchy, each matching boundary group can be used to infer one or more faces, weighted by the conditional probabilities. The result of face matching, therefore, is one or more face interpretations, ranked in decreasing order of probability. Figure 4 illustrates an example where, due to occlusion, an identified face does not match an aspect hierarchy face. We therefore descend to the boundary group level and match "subfaces" to the boundary groups, with each matching boundary group yielding one or more face inferences, each with an associated probability.

For each face in the user-identified part aspect, we have a set of face hypotheses. From the aspect hierarchy, as shown in Figure 5, each face can be used to infer a set of aspect hypotheses, ranked in decreasing order of probability. If we accumulate the aspect hypotheses for all the faces in the aspect, and weight them according to the face to aspect probabilities, we can define an ordering on the aspect hypotheses. Using an interpretation tree search [9], the hypotheses are verified, in turn, until a correct (i.e., high-scoring) hypothesis is found.

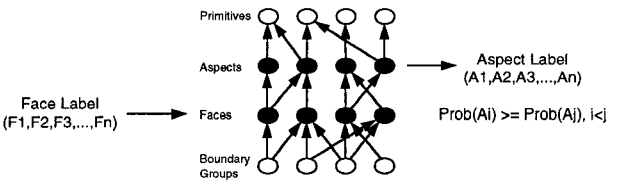


Figure 5: Generating the aspect labels of a face.

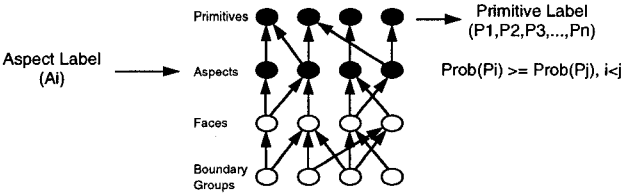


Figure 6: Generating the primitive labels of an aspect.

From an aspect hypothesis for the user-identified part, we can then use the aspect hierarchy to generate a volume hypothesis, as shown in Figure 6. If the mapping from the aspect to the volume is not unique ($P(volume|aspect) < 1.0$), then we can first prompt the user to select the correct 3-D shape for the outlined part from among the possible inferences (each displayed as a small rotating 3-D shape on the user's console).[1] Finally, using the probabilities mapping volumes to aspects in the aspect hierarchy, we can hypothesize all possible views for the selected volumetric part, ranked in decreasing order of probability. These views (part aspects) constitute a set of ranked search indices to be sent as queries to the image database.

4 Preprocessing the Image Database

Preprocessing a database image to recover the aspects of parts appearing in the image is the most challenging component of the system and the major limiting factor of our approach. In our prior work, e.g., [7, 6, 4], simple region growing operators were applied to an image, resulting in a graph-based description of a region's shape in terms of the shapes of (and relations

[1] Alternatively, if the true shape of the part is not known, then all possible volumetric inferences that exceed a given threshold, as defined by the user, are retained.

between) its bounding contours. However, these techniques were applied to very simple scenes containing objects without significant texture. For images containing cluttered scenes of complex, textured objects, such techniques will not properly segment an image into a set of salient regions.

Even if we had a perfect region segmentation module, it is unlikely that the resulting segmented regions would map one-to-one to the faces in an aspect in the aspect hierarchy. Particular regions (and other image features) may be salient in the *image* without being salient in the *world*. For example, the individual regions resulting from a set of high-contrast stripes on a coffee cup are not salient in terms of the cup's structure. Perhaps the most important aspect of segmentation is *region abstraction*, or the grouping of regions together to form "meta-regions" that correspond to salient (either real or abstract) surfaces on an object.

We are currently developing methods for region abstraction that take, as a starting point, a color region segmentation of the image [2]. However, to continue to pursue the ideas outlined in this paper, we have greatly simplified the problem by having the user annotate the images in the database. Unlike keyword annotation of an image, which can be very subjective, we have the user annotate the major regions of prominent part structure in the image in the same manner as the query image is annotated. From these user-defined regions, captured in a *region topology graph*, a set of aspects is hypothesized and verified. The result is a set of verified aspects, with each aspect encoding the particular regions it encompasses as well as a goodness of fit.

Whatever method is used to recover the regions, the result is a table indexed by aspect number (type) which lists, in decreasing order by score, all aspects of that type appearing in the database. Each such aspect specifies the image in which it appears, along with the nodes in that image's region topology graph that it encompasses. Furthermore, an adjacency matrix is built for all pairs of aspects in a given image, indicating whether the aspects are touching, overlapping, or disjoint. When a query aspect is received by the database, a collection of candidate aspects is returned by the aspect table. When a connectivity constraint between two aspects in an image must be tested, their connectivity is read directly from the adjacency matrix.

5 Matching the Query

As mentioned earlier, a given volumetric part (identified in the query image) maps to a set of views, ordered by probability. Each of these views is then passed to the database for image retrieval. Recall that preprocessing of each database image results in a covering of the image by a set of possibly overlapping aspects, indexed by a table. When a search aspect is generated from a user query, its location in the aspect table reveals those images (and locations) where the aspect was found in the database. When the aspects belonging to the first part selected by the user are sent as queries, the matching images are immediately ranked according to a similarity measure based on the probability of the query aspect given the volumetric part class and the quality (score) of the image database aspect. If the number of matching images is small, all can be displayed as thumbnail images on the user display.

In all likelihood, a single part query will result in a prohibitively high number of images that contain one or more matching parts. The user will then be asked to return to the query image and select, in the same manner that the first part was selected, a second part belonging to the object, subject to the constraint that the second part must be *connected* to the first part. This second query is then applied only to the set of images matching the first query. Furthermore, the connectedness constraint in the query image is passed on to the database images satisfying both queries, under the assumption that if parts are visible and connected in one view, they are visible and connected in any other view of the object. This process is repeated until a sufficiently small satisfying set of matching images is found. Images containing a subset of the user-selected parts can also be returned, albeit with a lower overall score.

6 Demonstration of the Approach

We demonstrate our approach to viewpoint-invariant shape indexing using a prototype system based on the 10-volume shape vocabulary depicted in Figure 1. In Figure 7, we show an image of a table lamp, representing the query image with which the user will interact to form a 3-D query. The goal will be to find instances of a lamp in a set of three images (obtained over the Web) of furniture showrooms, some of which contain lamps; these images are shown in Figure 8. The salient regions in each database image were all interactively outlined when the image was entered into the database. We stress that this is only a test in principle; obviously, a much larger database is required for real testing.

Using a mouse, the user first interacts with the query image, outlining the various regions belonging to a single part. In Figure 9(a), the user has drawn the contours of the lamp shade. At this point, the

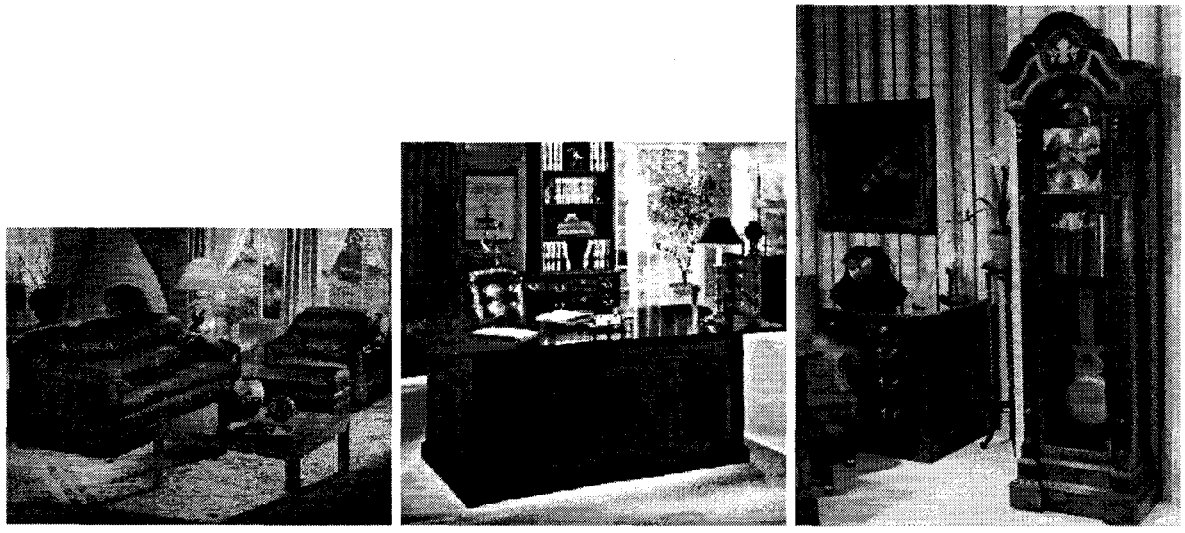


Figure 8: Sample image database images consisting of three furniture scenes.

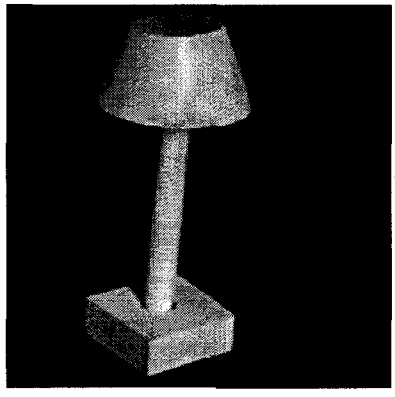


Figure 7: Query image containing a table lamp.

Part	Volume	Score
Shade	Tapered Cylinder	0.69

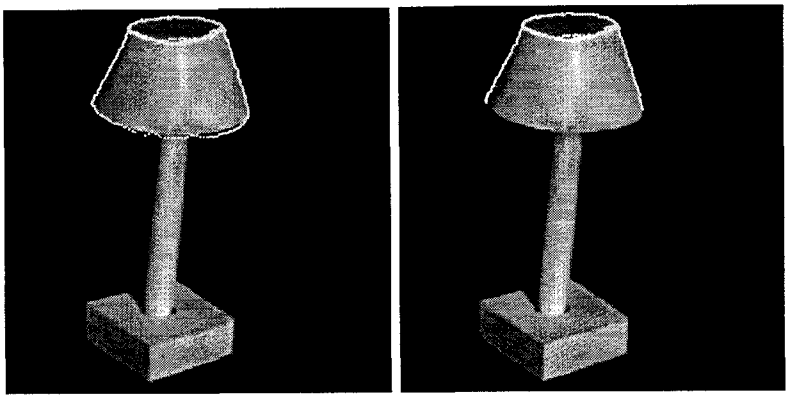
Table 1: Results of automatic recognition of user-drawn lamp shade.

system takes those contours and attempts to recover a 3-D part class corresponding to the user-identified part view. Table 1 shows the results of interpreting the contours based on our aspect hierarchy. By matching the drawn part aspect to our aspect hierarchy, the shape was unambiguously identified as volume 6 (tapered cylinder), appearing in aspect 12 in the aspect hierarchy, and having a score of 0.69.[2] Those contours used in the interpretation are shown in Figure 9(b). The noisy contour representing the bottom of the lamp shade was not used in the inference, resulting in a matching score less than 1.0.

From the inferred 3-D identity of the part, we can now begin to build our composite query to the image database. First, from the aspect hierarchy, we can use the conditional probabilities mapping volumes to aspects to generate the complete set of possible views of the volume. There are five such views, as shown in Figure 10. These views can be ranked in decreasing order of probability, and represent a set of ordered queries to the image database.

There are two options available to the user. The first is to send the query based on the single identified part. Without additional constraints, however, this would likely return many false positive matches from a normally large database. For illustration purposes, let us examine the results of generating this single-part query, as shown in Figure 11. The volumes in the images that matched the query are highlighted in the image. Two instances of the "trapezoidal" view of the tapered cylinder were found in the database, each corresponding to a lamp shade, and both having the same

[2]The score of a volume is proportional to the score of the aspect used to infer the volume and the conditional probability mapping the aspect to the volume. The score of an aspect is proportional to the score of the faces(s) used to infer the aspect and the conditional probabilities mapping the faces to the aspect. Finally, the score of a face is 1.0 if the entire face is matched correctly; otherwise, the score is proportional to the conditional probability mapping the matched boundary group to the face and the proportion of region boundary pixels defining the boundary group.



(a) (b)

Figure 9: Interpretating the lamp shade: a) User-specified contours; b) Contours used to infer the correct shape of the lamp shade.

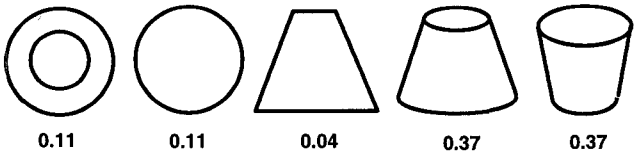


Figure 10: All possible aspects of the tapered cylinder (volume 6), along with their respective values of $Prob(aspect|volume6)$.

Part	Volume	Score
Neck	Cylinder	0.15
	Bent Cylinder	0.03
	Bent Block	0.01

Table 2: Results of automatic recognition of user-drawn lamp neck.

score due to the fact that the entire bounding contour of each region was used in its aspect interpretation. This example illustrates the viewpoint invariance of our matching strategy, since the viewpoint for which the part was identified by the user (4th view from left in Figure 10) does *not* match the viewpoints of the two instances found in the database (3rd view from left in Figure 10).

The second option is to build a more discriminating query by adding more parts which, in turn, adds more constraints to the database query. For example, Figure 12(a) shows a second part (the neck of the lamp) being added to the original query, while Table 2 shows the results of interpretating the second part. In this case, there are three interpretations for the part. The first is a cylinder (volume 5 in Figure 1) whose end face is occluded. The second is an extremely short, thick bent cylinder (volume 10) whose concave side is facing upwards in the image, and whose concave occluding contour is occluded by the lamp shade.[3] The third interpretation is a bent block (volume 4) oriented identically to the bent cylinder with the concave face occluded. Since the latter two interpretations have a very low score, they can be left out of the query, leaving only the aspects corresponding to the cylinder.

When the two parts (the lamp shade and neck) are combined, we now have an index representing a conjunction of two 3-D parts. If each of the two parts appears in a particular database image *and* their two respective views in that image are adjacent, then the image will be returned to the user, ordered with the other returned images according to the scores of the matched aspects. In Figure 13, we show the single image returned from the database as a result of our two-volume query. Of the two lamps returned in our first query, only one survives the second query due to the added constraint of the lamp neck. Note also that the view of the lamp neck in the query (two parallel lines bridged at one end by a convex curve and at the other (occluded) end by a concave curve) is different from that in the returned database instance (two parallel lines bridged at both ends by a straight line). Note that in the query image and database image lamp necks, a portion (one end) of each region was not grouped into its defining aspect, due to contour segmentation and/or grouping errors.

Instead of querying the image database with the

[3] Note that the bent cylinder volume has parallel end faces.

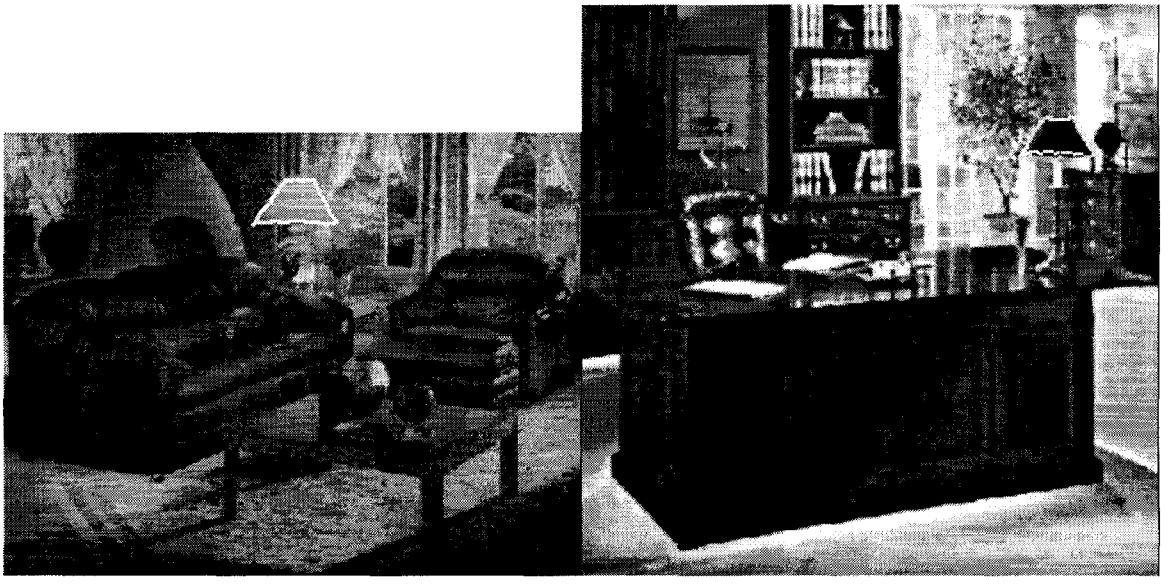


Figure 11: Results of querying the image database using a single-part query based on the lamp shade. Two instances are returned having identical scores. Note that the view of the lamp shade in the query image is different from the view found in both database images, illustrating the viewpoint-invariant nature of the search.

Part	Volume	Score
Base	Block	1.00

Table 3: Results of automatic recognition of user-drawn lamp base.

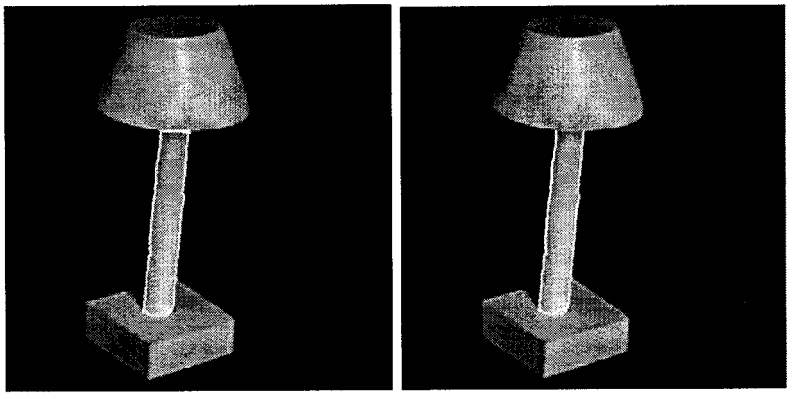
lamp shade, the user could query the database with the base of the lamp, a part shape which, in this case, is much less unique to the lamp. Figures 14(a) and (b) show the user-specified contours, along with the contours used to recognize the part, respectively. In this case the system used all the contours to interpret the shape as a block. Rather than drawing the occluding contours in the image, the user has extrapolated the view of the part to encompass that part of the base occluded by the neck. Alternatively, the user could have drawn on the visible portion of the base, and the system would still have correctly interpreted the part, albeit with a score less than that shown in Table 3.

If we generate a single-part query based only on the lamp base (block), then the best three instances of the block in the database are shown in Figure 15. Note that for the best instance of the block, the region representing the front of the desk was occluded while both the top and side were intact. In the second instance of the block, two sides of the desk were incomplete in the sense that not all the bounding contours of the two regions were used in the aspect interpretation (e.g., bottom contour of leftmost face and right contour of topmost face). In the third instance of the block, one whole face of the block was not visible due to perspective effects. The diminishing evidence contributing to the interpretations of the three database instances is responsible for their relative ordering.

Since the indexing power of a single part is low, particularly for a commonly occurring part class, more parts, along with their connectivity constraints, are needed to reduce the number of false positives returned to the user. In addition, it should be noted that none of the database lamps has block bases. If we allow query part subsets to match database images, then a query consisting of all three lamp parts would, for example, yield all database images containing all three connected parts, any two connected parts, or any one part.

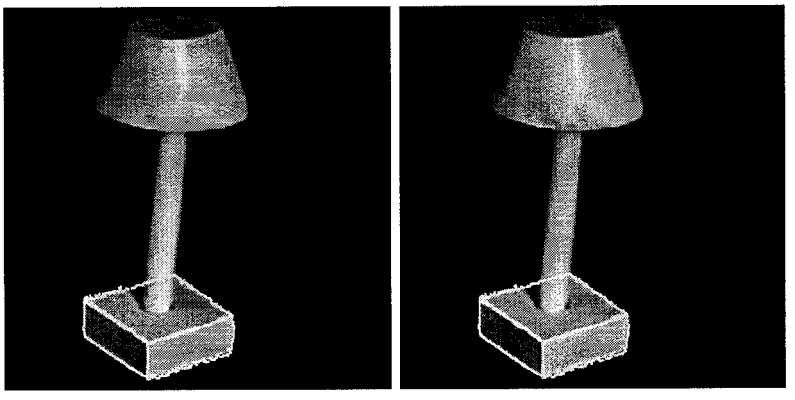
7 Towards a More Complete Part Vocabulary

Ideally, since we can't anticipate what objects will be contained in our images, we need a much richer set of volumetric parts that cover a wide range of objects, i.e., a limited universe of shapes that we expect can model any object (or significant portion thereof) we anticipate to be in the image. Work has already begun on the automatic construction of a much larger



(a) (b)

Figure 12: Interpretating the lamp neck: a) User-specified contours; b) Contours used to infer the correct shape of the lamp neck.



(a) (b)

Figure 14: Interpretating the lamp base: a) User-specified contours; b) Contours used to infer the correct shape of the lamp base.

aspect hierarchy containing hundreds, if not thousands, of shapes. Working with a deformable superquadric model with many global deformation parameters (e.g., pinching, twisting, tapering, bending, shearing, etc.), we will sample the entire parameter space of the model. Each part instance will be centered in a tesselated view sphere, and a view of the part will be synthesized from each position. For a given view, the resulting synthetic image will be region segmented, and the region shapes and adjacencies will be used to derive an aspect representation of the view.

Once an aspect has been defined for each view, neighboring views having identical aspects will be merged, resulting in an aspect graph of the part instance. In a similar fashion, neighboring instances in the model's sampled parameter space will be clustered if their aspect graphs are identical. The result is an automatic partitioning of the parameter space into a set of volumetric part classes and their associated aspect graphs. From this information, an aspect hierarchy can be automatically constructed, including its conditional probabilities.

8 Conclusions

The simple demonstration in Section 6 serves only to illustrate the concept of viewpoint-invariant 3-D shape indexing; it is clearly not practical to require that each database image be annotated with a set of abstract regions. Until we can more effectively address the problems of both region segmentation and region abstraction, our idea will remain, at best, a promising concept realizable only in the long term. Nevertheless, we firmly believe that the kind of semantic indexing that users will demand cannot be restricted to 3-D objects appearing in the same view. We need some kind

Figure 15: Results of the single-part lamp base (block) query. The three best-scoring instances are shown left to right.

of model that can resolve two dissimilar views of the same object. It is impractical to assume that there exists a full 3-D model to relate the different views of the same object; web pages contain pictures, not 3-D models. And assuming that a complete collection of views exists for every query object is equally impractical.

Our approach relies on a large enough set of volumetric "building blocks" such that a significant portion of any query object can be constructed from members of the set. We believe that this set of building blocks not only models objects at the right granularity, but represents an intuitively appealing granularity for user interaction. We assume no knowledge of either query object or database object other than that such objects can be partially constructed from our set. This set, along with its resulting aspect hierarchy, is computed once off-line. Furthermore, we have outlined a procedure for its automatic construction which, in fact, can be applied to any parameterized class of shape.

The user interface that we have provided is crude, providing a set of snake-like interactive tools to outline regions in the image. Much work lies ahead in evaluating and improving this interface. For example, should the parts be outlined in the image, or perhaps selected from a vocabulary and fitted to the image data?[4] How should ambiguous inferences be presented for selection? How should the matching images be presented to the user, and how should matching objects be highlighted? Although this paper has not focused on user interface issues, we acknowledge the importance of conducting proper experiments in designing/evaluating the user interface.

Although much work lies ahead in, for example, designing a user interface, constructing a richer set of shapes, and devising more efficient algorithms for indexing (both into the database of part views and the database images), our major challenge will be improved region segmentation and abstraction. Such segmentation need not be perfect, as we have successively dealt with both region under- and over-segmentation in previous work [4]. Nevertheless, there has been little work to date on region and shape abstraction from 2-D images, and we believe that to be the critical component of such a system. We will therefore explore, in parallel, the representational, interaction, and indexing issues outlined in this paper, as well as the problem of region segmentation and abstraction. Together, these ideas represent a promising direction for overcoming the 2-D barrier of current shape-indexing methods.

[4] This suggestion was offered by an anonymous reviewer of the paper.

References

[1] Z. Chen and S.-Y. Ho. Computer vision for robust 3-D aircraft recognition with fast library search. *Pattern Recognition*, 24(5):375–390, 1991.

[2] D. Comaniciu and P. Meer. Robust analysis of feature spaces: Color image segmentation. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 750–755, 1997.

Figure 13: Results of querying the image database using a two-part query based on the lamp shade and neck. One instance is returned, consisting of views of each part that are different from those in the query image.

[3] 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.

[4] S. Dickinson, H. Christensen, J. Tsotsos, and G. Olofsson. Active object recognition integrating attention and viewpoint control. *Computer Vision and Image Understanding*, 67(3):239–260, September 1997.

[5] 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.

[6] 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.

[7] 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.

[8] C. Faloutsos, R. Barber, M. Flickner, W. Niblack, D. Petkovic, and W. Equitz. Efficient and effective querying by image content. *Journal of Intelligent Systems*, 3(3/4):231–262, July 1994.

[9] 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.

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

[11] H. Jagadish. A retrieval technique for similar shapes. In *International Conference on Management of Data, SIGMOD '91*, pages 208–217, Denver, CO, 1991.

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

[13] Y. Lamdan, J. Schwartz, and H. Wolfson. Affine invariant model-based object recognition. *IEEE Transactions on Robotics and Automation*, 6(5):578–589, October 1990.

[14] 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.

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

[16] H. Murase and S. Nayar. Visual learning and recognition of 3-D objects from appearance. *International Journal of Computer Vision*, 14:5–24, 1995.

[17] W. Niblack, R. Barber, W. Equitz, M. Flickner, E. Glasman, D. Petlovic, and P. Yanker. The qbic project: Querying images by content using color, texture, and shape. In *SPIE Conference 1908 on Storage and Retrieval for Image and Video Databases*, 1993.

[18] A. Pentland, R. Picard, and S. Sclaroff. Photobook: Tools for content-based manipulation of image databases. *International Journal of Computer Vision*, 18(3), June 1996.

[19] S. Sclaroff and A. Pentland. Modal matching for correspondence and recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(6):545–561, June 1995.