Bridging the Representation Gap Between Models and Exemplars

Yakov Keselman
Department of Computer Science
Rutgers University
New Brunswick, NJ 08903, USA

Sven Dickinson
Department of Computer Science
University of Toronto
Toronto, Ontario M5S 3G4, Canada

Abstract

The recognition community has long avoided bridging the representational gap between traditional, low-level image features and generic models. Instead, the gap has been artificially eliminated by either bringing the image closer to the models, using simple scenes containing idealized, textureless objects, or by bringing the models closer to the images, using 3-D CAD model templates or 2-D appearance model templates. In this paper, we begin by examining this trend and track its evolution over the last 30 years. We argue for the need to bridge (not eliminate) this representational gap, and review our recent progress for the domain of model acquisition. Specifically, we address the problem of automatically acquiring a generic 2-D view-based class model from a set of images, each containing an exemplar object belonging to that class. We introduce a novel graph-theoretical formulation of the problem, and demonstrate the approach on real imagery.

1. Introduction

In the object recognition community, object representations have spanned a continuum ranging from prototypical models (often called class-based or generic models) to exemplar-based model (often called template-based or appearance-based models). Those advocating prototypical models address the task of recognizing novel (never before seen) exemplars from known classes, whose definitions strive to be invariant to changes in surface texture, color, part articulation, and minor deformation in shape. Those advocating exemplar models address the very different task of recognizing particular instances of objects, such as JFK's face or a can of Coke. In a completely orthogonal direction, prototypical models can be object-centered or viewer-centered (or both [8, 10, 9]), provided that the 3-D or 2-D features that comprise the model satisfy the above invariance goals. Similarly, exemplar models can be object-centered, specifying the exact 3-D geometry of an object, e.g., a rigid CAD model "template", or viewer-centered, specifying the exact appearance of an object.

Interestingly, the evolution of object recognition over the past 30 years has followed a path from prototypical models to exemplar models, as illustrated in Figure 1. Beginning in the 1970's, vision researchers aimed for prototypical vision systems, using complex volumetric parts, such as generalized cylinders (e.g., [5, 1, 23, 7]), and later superquadrics (e.g., [25, 13, 31, 15, 32, 19]) and geons (e.g., [4, 8, 10, 9, 3, 27, 6]). The main challenge to these early systems was the *representational gap* that existed between the low-level features that could be reliably extracted, and the abstract nature of the model components. Rather than addressing this representational gap, the community effectively eliminated it by bringing the images closer to the models. This was accomplished by removing object surface markings and structural detail, controlling lighting conditions, and reducing scene clutter. Edges in the image could then be assumed to map directly to the limbs and surface discontinuities of high-order volumetric parts making up the models. The results left many unsatisfied, as the images and objects were often contrived, and the resulting systems were unable to deal with real objects imaged under real conditions.

The 1980's ushered in 3-D models that captured the exact shape of the object. Such models, often in the form of CAD models, were effectively 3-D templates, e.g., [16, 20, 14]). Provided that such a model could be acquired for a real object, the community now found that it could build object recognition systems that could begin to recognize real (albeit restricted) objects. This time, the representational gap was eliminated by bringing the model closer to the imaged object, requiring the model to capture the exact geometry of the object. Moreover, since the presence of texture and surface markings seriously affected the computational complexity of these systems, they too selected objects which were texture-free – objects for which a salient image edge discontinuity mapped to a polyhedral edge. Again, there was dissatisfaction, as the resulting systems were unable to recognize complex objects with complex surface markings.

Recently, beginning in the 1990's, appearance models have replaced CAD models, and for the first time, recognition systems were constructed that could recognize arbitrarily complex objects, e.g., [33, 21, 18]). Storing a dense

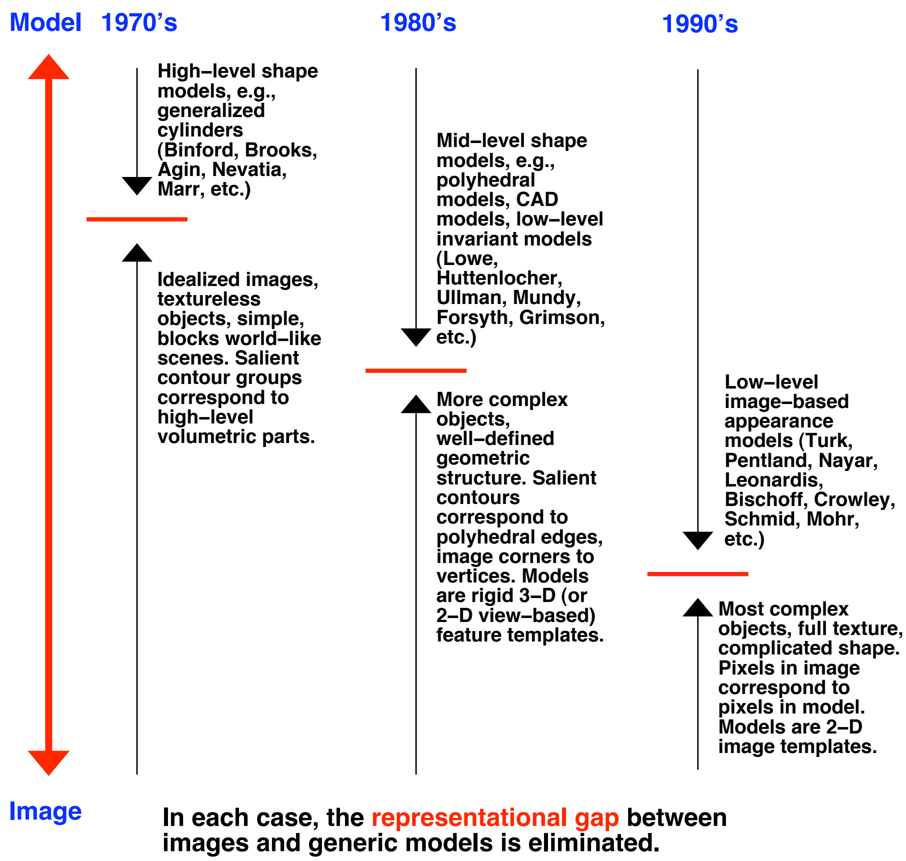


Figure 1: Evolution of Object Recognition

set of images of the object from all possible viewpoints, the appearance models were not limited by object geometric complexity, texture, or surface markings. In this case, the representational gap was eliminated by bringing the models all the way down to the image. The resulting systems could therefore recognize only exemplar objects – specific objects that had been seen at training time. Despite a number of serious limitations of this approach, including difficulties in dealing with background clutter, object translation, rotation, and scaling, the approach has gained tremendous popularity.

It is important to note that in bringing the model closer to the image, the appearance-based and CAD-based approaches have altered the problem definition from generic to exemplar object recognition. The systems developed in the 70's cannot be compared to those developed today, for their target domains are different. We must acknowledge the efficacy of appearance-based recognition systems for exemplar recognition; provided that the above limitations can be overcome, this technique may emerge as the best method for recognizing exemplars. However, it is important to acknowledge that the prototypical recognition problem is an important one, and despite the vision community's distraction towards (and fascination with) appearance-based methods, the hypothesis that these (or their analogous 3-D template-based) methods can scale up to perform prototypical object recognition is dubious. What, then, has led us away from the important problem of generic object recognition?

Over the past 30 years, the three approaches to eliminating the representational gap (shown in Figure 1) are driven by the same limiting assumption: there exists a one-to-one correspondence between a "salient" feature in the image (e.g., a long, high-contrast line or curve, a well-defined homogeneous region, a corner or curvature discontinuity or, in the case of an appearance-based model, the values of a set of image pixels) and a feature in the model. This assumption is fundamentally flawed, for saliency in the image does *not* imply saliency in the model. Under this assumption, object recognition will continue to be exemplar-based, and generic recognition will continue to be rather contrived.

To make real progress on the problem of generic object recognition, we must address the representational gap outlined in Figure 1. Not only must we continue to push the technologies of segmentation and perceptual grouping, we must be able to generate image *abstractions* that may not exist explicitly in the image, but which capture the salient, invariant shape properties of a generic model.[1] We argue that for the purpose of generic or prototypical object (or, more specifically, shape) recognition, the process of image abstraction must recover a set of "metaregions" that map to the coarse surfaces (or volumetric primitives) on a prototypical model.

In light of this goal, the difficult challenge of image abstraction can therefore be cast as a two-fold problem, as shown in Figure 2. First, we seek a (compile-time) method for automatically acquiring a generic model from a set of images (containing exemplars belonging to a known class) that bridges the representational gap between the output of an image segmentation module and the "parts" of a generic model. Although in Figure 2, this is exemplified by a generic, view-based model (a coffee cup), those advocating object-centered models could easily map the parts of this view into a set of object-centered components, e.g., [8, 10, 9], leading to an object-centered model. Although we make no claim as to the final form of the model (2-D or 3-D), we believe that a parts-based approach to viewer-centered representations can better accommodate the intractable complexity associated with "whole object" views of complex, articulating objects [10].

Next, from an image of a real exemplar, we seek a (run-time or recognition-time) method that will recover a high-level "abstraction" that contains the coarse features that make up some model. In this paper, we will briefly review our approach (described in [17]) to the first problem, that of generic model acquisition, and speculate briefly on our direction towards solving the second, and more difficult problem of generic object recognition. Some sections of this paper have been taken directly from [17], while others summarize algorithmic details given in [17]. Generic object recognition is an essential task, whose lack of solution confines object recognition to the laboratory, where such conditions as lighting, clutter, occlusion, and object domain can be tightly controlled. Granted, the generic object recognition torch has continued to burn, albeit dimly, with a number of researchers committed to the problem, e.g., [29, 36, 2, 30, 24], to name just a few. We believe the time has come for the pendulum to swing back towards solving the problem of generic, prototypical, or class-based modeling and recognition.

[1] Although a number of approaches extract regions or perceptual groups as input to a recognition system, they typically assume that corresponding regions or groups exist on the model.

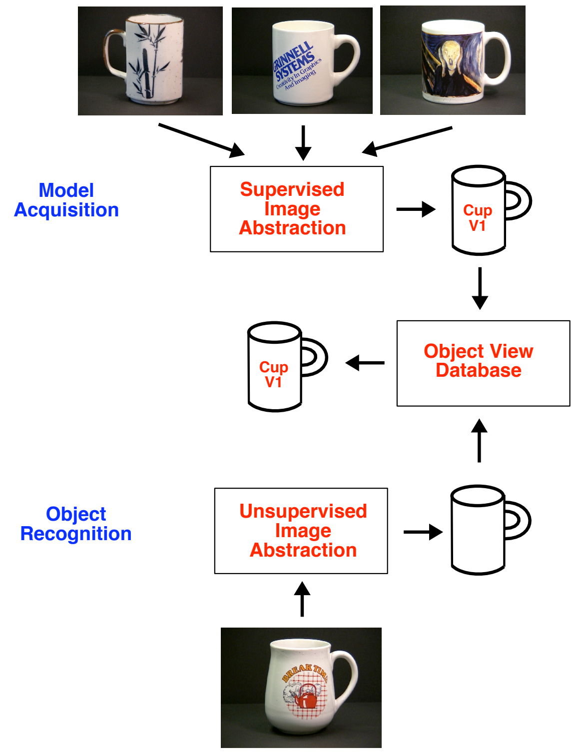
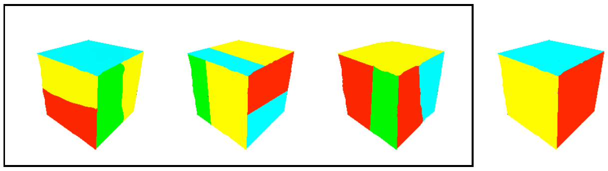


Figure 2: The Role of Image Abstraction in Both Model Acquisition and Object Recognition

2. An Illustrative Example of Generic Model Acquisition

Assume that we are presented with a collection of images, such that each image contains a single exemplar, all exemplars belong to a single known class, and that the viewpoint with respect to the exemplar in each image is similar. Fig. 3(a) illustrates a simple example in which three different images, each containing a block in a similar orientation, are presented to the system (we will return to this example throughout the paper to illustrate the various steps in our algorithm). Our task is to find the common structure in these images, under the assumption that structure that is common across many exemplars of a known class must be definitive of that class. Fig. 3(b) illustrates the class "abstraction" that is derived from the input examples. In this case, the domain of input examples is rich enough to "intersect out" irrelevant structure (or appearance) of the block. However, had many or all the exemplars had vertical stripes, the approach would be expected to include vertical stripes in that view of the abstracted model.

Any discussion of model acquisition must be grounded in image features. In our case, each input image will be region-segmented to yield a region adjacency graph. Similarly, the output of the model acquisition process will yield



(a) (b)

Figure 3: Illustrative Example of Generic Model Acquisition: (a) input exemplars belonging to a single known class; (b) generic model abstracted from examples.

a region adjacency graph containing the "meta-regions" that define a particular view of the generic model. Other views of the exemplars would similarly yield other views of the generic model. The integration of these views into an optimal partitioning of the viewing sphere, or the recovery of 3-D parts from these views, is beyond the scope of this paper. For now, the result will be a collection of 2-D views that describe a generic 3-D object. This collection would then be added to the view-based object database used at recognition time.

3. Related Work

Automatic model acquisition from images has long been associated with object recognition systems. One of the advantages of appearance-based modeling techniques, e.g., [21], is that no segmentation, grouping, or abstraction is necessary to acquire a model. An object is simply placed on a turntable in front of a camera, the viewing sphere is sampled at an appropriate resolution, and the resulting images (or some clever representation thereof) are stored in a database. Others have sought increased illumination-, viewpoint-, or occlusion-invariance by extracting local features as opposed to using raw pixel values, e.g., [26, 28, 22, 34]. Still, the resulting models are very exemplar-specific due to the extreme locality at which they extract and match features (e.g., one pixel or at best, a small neighborhood around one pixel). The resulting models are as far from generic as one can get.

In the domain of range images, greater success has been achieved in extracting coarse models. Generic shape primitives, such as restricted generalized cylinders, quadrics, and superquadrics have few parameters and can be robustly recovered from 3-D range data [25, 31, 32]. Provided the range data can be segmented into parts or surfaces, these generic primitives can be used to model the coarse shapes of the parts, effectively abstracting away structural detail. Unlike methods operating on 2-D data, these methods are insensitive to perceived structure in the form of surface markings or texture.

In the domain of generating generic models from 2-D data, there has been much less work. The seminal work of Winston [35] pioneered learning descriptions of 3-D objects from structural descriptions of positively or negatively labeled examples. Nodes and edges of graph-like structures were annotated with shapes of constituent parts and their relations. As some shapes and relations were abstractions and decompositions of others, the resulting descriptions could be organized into a specificity-based hierarchy. In the 2-D shape model domain, Ettinger learned hierarchical, structural descriptions from images, based on Brady's curvature primal sketch features [11]. The technique was successfully applied to traffic sign recognition and remains one of the more elegant examples of feature abstraction and generic model acquisition.

4. Problem Formulation

Returning to Fig. 3, let us now formulate our problem more concretely. As we stated, each input image is processed to form a region adjacency graph (we employ the region segmentation algorithm of Felzenzwalb and Huttenlocher [12]). Let us now consider the region adjacency graph corresponding to one input image. We will assume, for now, that our region adjacency graph represents an oversegmentation of the image. (In Section 8, we will discuss the problem of undersegmentation, and how our approach can accommodate it.) The space of all possible region adjacency graphs formed by any sequence of merges of adjacent regions will form a lattice, as shown in Fig. 4. The lattice size is exponential in the number of regions obtained after initial oversegmentation.[2]

Each of the input images will yield its own lattice. The bottom node in each lattice will be the original region adjacency graph. In all likelihood, if the exemplars have different shapes (within-class deformations) and/or surface markings, the graphs forming the bottom of their corresponding lattices may bear little or no resemblance to each other. Clearly, similarity between the exemplars cannot be ascertained at this level, for there does not exist a one-to-one correspondence between the "salient" features (i.e., regions) in one graph and the salient features in another. On the other hand, the top of each exemplar's lattice, representing a silhouette of the object (where all regions have been merged into one region), carries little information about the salient surfaces of the object.

We can now formulate our problem more precisely, recalling that a lattice consists of a set of nodes, with each node corresponding to an entire region adjacency graph. Given N input image exemplars, $E_1, E_2, \ldots, E_N$, let $L_1, L_2, \ldots, L_N$ be their corresponding lattices, and for a given lattice, L_i, let $L_i n_j$ be its constituent nodes, each

[2] Indeed, considering the simple case of a long rectangular strip subdivided into $n+1$ adjacent rectangles, the first pair of adjacent regions able to be merged can be selected in n ways, the second in $n-1$, and so on, giving a lattice size of $n!$.

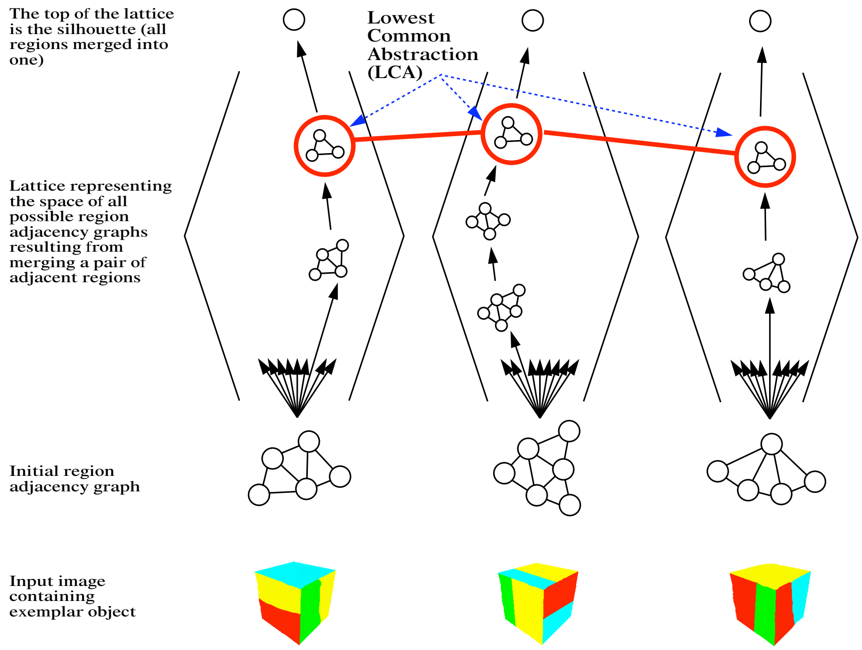


Figure 4: The Lowest Common Abstraction of a Set of Input Exemplars

representing a region adjacency graph, G_{ij}. We define a *common abstraction*, or CA, as a set of nodes (one per lattice) $L_1 n_{j_1}, L_2 n_{j_2}, \ldots, L_N n_{j_N}$ such that for any two nodes $L_p n_{j_p}$ and $L_q n_{j_q}$, their corresponding graphs G_{pj_p} and G_{qj_q} are isomorphic. Thus, the root node (whose graph consists of one node representing the silhouette region) of each lattice is a common abstraction. We define the *lowest common abstraction*, or LCA, as the common abstraction whose underlying graph has maximal size (in terms of number of nodes). Given these definitions, our problem can be simply formulated as finding the LCA of N input image exemplars.

Intuitively, we are searching for a node (region segmentation) that is common to every input exemplar's lattice and that retains the maximum amount of structure common to all exemplars. Unfortunately, the presence of a single heavily undersegmented exemplar (a single-node silhouette in the extreme case) will drive the LCA towards the trivial silhouette CA. In a later section, we will relax our LCA definition to make it less sensitive to such outliers.

5. The LCA of Two Examples

For the moment, we will focus our attention on finding the LCA of two lattices, while in the next section, we will accommodate any number of lattices. Since the input lattices are exponential in the number of regions, actually computing the lattices is intractable. Clearly, we need a means for focusing the search for the LCA that avoids significant lattice generation. Our approach will be to restrict the search for the LCA to the *intersection* of the lattices. Typically, the intersection of two lattices is much smaller than either lattice (unless the images are very similar), and leads to a tractable search space. But how do we generate this new "intersection" search space without enumerating the lattices?

Our solution is to work top-down, beginning with a node known to be in the intersection lattice – the root node, representing a single region (silhouette). If the intersection lattice contains only this one node, i.e., one or both of the region segmented images contain a single region, then the process stops and the LCA is simply the root (silhouette). However, in most cases, the root of each input lattice is derived from an input region adjacency graph containing multiple regions. So, given two silhouettes, each representing the apex of a separate, non-trivial lattice, we have the opportunity to search for a lower abstraction (than the root) common to both lattices. Our approach will be to find a decomposition of each silhouette region into two subregions, such that: 1) the shapes of the corresponding subregions are similar, and 2) the relations among the corresponding regions are similar. Since there are an infinite number of possible decompositions of a region into two component regions, we will restrict our search to the space of decompositions along region boundaries in the original region adjacency graphs. Note that there may be multiple 2-region decompositions that are common to both lattices; each is a member of the intersection set.

Assuming that we have some means for ranking the matching decompositions (if more than one exists), we pick the best one (the remainder constituting a set of backtracking points), and recursively apply the process to each pair of isomorphic component subregions.[3] The process continues in this fashion, "pushing" its way down the intersection lattice, until no further decompositions are found. This lower "fringe" of the search space represents the LCA of the original two lattices.

The specific algorithm for choosing the optimal pair of decompositions is given in [17], and can be summarized as follows:

1. Map each region adjacency graph to its dual *boundary segment graph*, in which boundary segments become nodes and edges capture segment adjacency.

2. Form the product graph (or association graph) of the two boundary segment graphs. Nodes and arcs in the product graph correspond to pairs of nodes and arcs, respectively, in the boundary segment graphs. A path in the product graph therefore corresponds to a pair of paths in the boundary segment graphs which, in turn, correspond to a pair of decompositions of the region adjacency graphs.

3. With appropriate edge weights, along with a suitable objective function, the optimal pair of corresponding decompositions corresponds to the shortest path in the product graph.

4. The optimal pair of decompositions is verified in terms of satisfying the criteria of region shape similarity and region relation consistency.

6. The LCA of Multiple Examples

So far, we've addressed only the problem of finding the LCA of two examples. How, then, can we extend our approach to find the LCA of multiple examples? Furthermore, when moving towards multiple examples, how do we prevent a "noisy" example, such as a single, heavily undersegmented silhouette, from driving the solution towards the trivial silhouette? To extend our two-exemplar LCA solution to a robust, multi-exemplar solution, we begin with two important observations. First, the LCA of two exemplars lies in the intersection of their abstraction lattices. Thus, both exemplar region adjacency graphs can be transformed into their LCA by means of sequences of region merges. Second, the total number of merges required to transform the graphs into their LCA is minimal among all elements of the intersection lattice, i.e., the LCA lies at the lower fringe of the lattice.

[3]Each subregion corresponds to the union of a set of regions corresponding to nodes belonging to a connected subgraph of the original region adjacency graph.

Our solution begins by constructing an approximation to the intersection lattice of multiple examples. Consider the closure of the set of the original region adjacency graphs under the operation of taking pairwise LCA's. In other words, starting with the initial region adjacency graphs, we find their pairwise LCA's, then find pairwise LCA's of the resulting abstraction graphs, and so on (note that duplicate graphs are removed). We take all graphs, original and LCA, to be nodes of a new *closure* graph. If graph H was obtained as the LCA of graphs G_1 and G_2, then directed arcs go from nodes corresponding to G_1, G_2 to the node corresponding to H in the closure graph.

Next, we will relax the first property above to accommodate "outlier" exemplars, such as undersegmented, input silhouettes. Specifically, we will not enforce that the LCA of multiple exemplars lie in the intersection set of all input exemplars. Rather, we will choose a node in our approximate intersection lattice that represents a "low abstraction" for many (but not necessarily all) input exemplars. More formally, we will define the LCA of a set of exemplar region adjacency graphs to be that element in the intersection of two or more abstraction lattices that minimizes the total number of edit operations (merges or splits) required to obtain the element from *all* the given exemplars. If a node in the intersection lattice lies along the lower fringe with respect to a number of input exemplars, then its sum distance to all exemplars is small. Conversely, the sum distance between the silhouette outlier (in fact, the true LCA) and the input exemplars will be large, eliminating that node from contention. Our algorithm for computing this "median" of the closure graph, along with an analysis of its complexity, is given in [17].

7. Experiments

In Figures 5 and 6, we illustrate the results of our approach applied to two sets of three coffee cup images, respectively. In each case, the lower row represents the original images, the next row up represents the input region segmented images (with black borders), while the LCA is shown with an orange border. In each case, the closure graph consists of only four members, with the same pairwise LCA emerging from all input pairs. While in Fig 5, the solution captures our intuitive notion of the cup's surfaces, the solution in Fig 6 is less intuitive. A strip along the bottom is present in each exemplar, and understandably becomes part of the solution. However, due to region segmentation errors, the larger blue region in the middle cup extends into the handle. Consequently, a cut along its handle (as is possible on the other cups) is not possible for this exemplar, resulting in a "stopping short" of the recursive decomposition at the

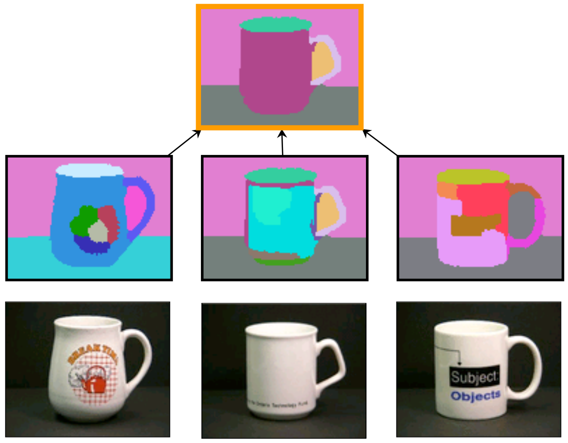


Figure 5: Computed LCA (orange border) of Three Examples

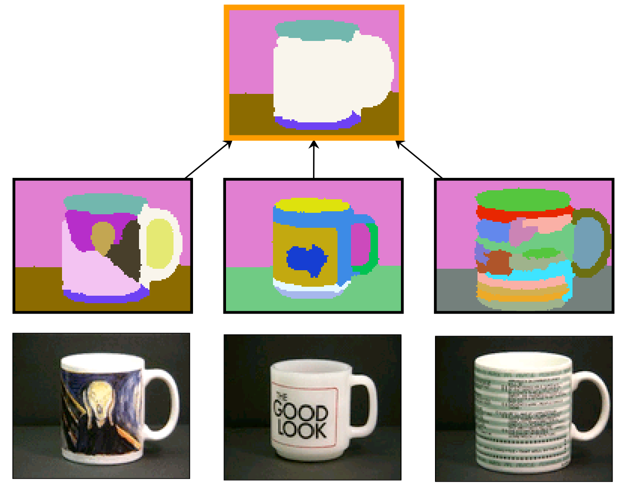


Figure 6: Computed LCA (orange border) of Three Examples

large white region in the solution (LCA).

In Figure 7, we again present three exemplars to the system. In this case, the closure graph has many nodes. Unlike Figures 5 and 6, in which all pairwise LCA's were equal (leading to a somewhat trivial solution to our search for the global LCA), each pair of input exemplars leads to a different LCA which, in turn, leads to additional LCA's. Continuing this process eventually results in the inclusion of the silhouette in the closure graph. The solution, according to our algorithm, is again shown in orange, and represents an effective model for the cup.

8. Conclusions

The quest for generic object recognition hinges on an ability to generate abstract, high-level descriptions of input data. This process is essential not only at run-time, for the recognition of objects, but also at compile time, for the automatic acquisition of generic object models. In this paper, we address the latter problem – that of generic model acquisition from examples. We review our novel formulation of the problem, in which the model is defined as the lowest common abstraction of a number of segmentation lattices, representing a set of input image exemplars [17]. To manage the intractable complexity of this formulation, we focus our search on the intersection of the lattices, reducing complexity by first considering pairs of lattices, and later combining these local results to yield an approximation to the global solution.

We have shown some very preliminary results that compute a generic model from a set of example images belonging to a known class. Although these results are encouraging, further experimentation is necessary and a number of limitations need to be addressed. For example, we currently assume an oversegmented image, thereby requiring only region merge operations. However, our region representation explicitly encodes a finite number of region split points [30], allowing us to accommodate region splitting within our framework.

Our next major step is the actual recognition of the derived models from a novel exemplar. Our efforts are currently focused on the analysis of the conditions under which two regions are merged. If we can derive a set of rules for the perceptual grouping of regions, we will be able to generate abstractions from images. Given a rich set of training data derived from the model acquisition process (recall that the LCA of two examples yields a path of region merges), we are applying machine learning methods to uncover these conditions. Combined with our model acquisition procedure, we can close the loop on a system for generic object recognition which addresses a representational gap that has been long ignored in computer vision.

References

[1] G. Agin and T. Binford. Computer description of curved objects. *IEEE Transactions on Computers*, C-25(4):439–449, 1976.

[2] R. Basri, L. Costa, D. Geiger, and D. Jacobs. Determining the similarity of deformable shapes. In *Proceedings, ICCV Workshop on Physics-Based Modeling in Computer Vision*, pages 135–143, 1995.

[3] R. Bergevin and M. D. Levine. Part decomposition of objects from single view line drawings. *CVGIP: Image Understanding*, 55(1):73–83, January 1992.

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

[5] T. Binford. Visual perception by computer. In *Proceedings, IEEE Conference on Systems and Control*, Miami, FL, 1971.

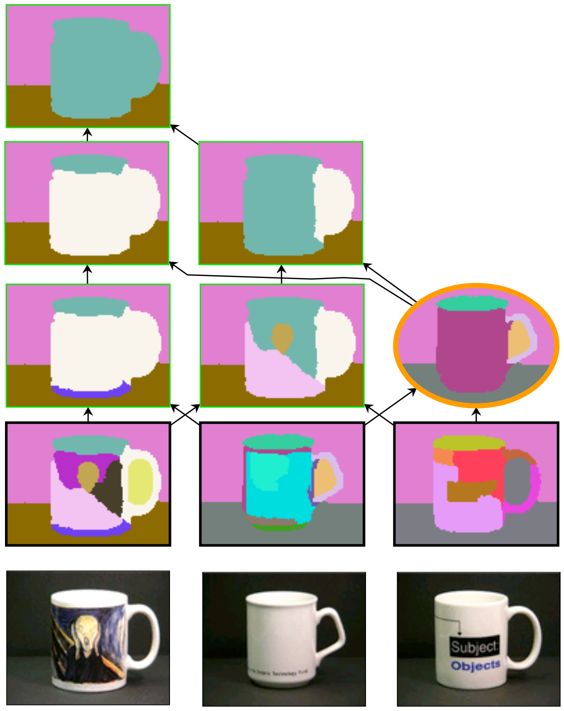


Figure 7: Computed LCA (orange border) of Three Examples

[6] D. Borges and R. Fisher. Class-based recognition of 3d objects represented by volumetric primitives. *Image and Vision Computing*, 15(8):655–664, 1997.

[7] R. Brooks. Model-based 3-D interpretations of 2-D images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5(2):140–150, 1983.

[8] 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.

[9] 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.

[10] 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.

[11] G. Ettinger. Large hierarchical object recognition using libraries of parameterized model sub-parts. In *Proceedings, IEEE Conference on Computer Vision and Pattern Recognition*, pages 32–41, Ann Arbor, MI, 1988.

[12] P. Felzenszwalb and D. Huttenlocher. Image segmentation using local variation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 98–104, Santa Barbara, CA, 1998.

[13] F. Ferrie, J. Lagarde, and P. Whaite. Darboux frames, snakes, and superquadrics. In *Proceedings, IEEE Workshop on Interpretation of 3D Scenes*, pages 170–176, 1989.

[14] D. Forsyth, J. Mundy, A. Zisserman, C. Coelho, A. Heller, and C. Rothwell. Invariant descriptors for 3d object recognition and pose. *IEEE PAMI*, 13:971–992, October 1991.

[15] A. Gupta. Surface and volumetric segmentation of 3D objects using parametric shape models. Technical Report MS-CIS-91-45, GRASP LAB 128, University of Pennsylvania, Philadelphia, PA, 1991.

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

[17] Y. Keselman and S. Dickinson. Generic model abstraction from examples. In *Proceedings, CVPR 2001, Kauai*, December 2001.

[18] A. Leonardis and H. Bischoff. Dealing with occlusions in the eigenspace approach. In *Proceedings, IEEE Conference on Computer Vision and Pattern Recognition*, pages 453–458, San Francisco, CA, June 1996.

[19] A. Leonardis, F. Solina, and A. Macerl. A direct recovery of superquadric models in range images using recover-and-select paradigm. In *Proceedings, Third European Conference on Computer Vision (Lecture Notes in Computer Science, Vol 800)*, pages 309–318, Stockholm, Sweden, May 1994. Springer-Verlag.

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

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

[22] R. Nelson and A. Selinger. A cubist approach to object recognition. In *Proceedings, IEEE International Conference on Computer Vision*, Bombay, January 1998.

[23] R. Nevatia and T. Binford. Description and recognition of curved objects. *Artificial Intelligence*, 8:77–98, 1977.

[24] M. Pelillo, K. Siddiqi, and S. Zucker. Matching hierarchical structures using association graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(11):1105–1120, November 1999.

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

[26] A. Pope and D. Lowe. Learning object recognition models from images. In *Proceedings, IEEE International Conference on Computer Vision*, pages 296–301, Berlin, May 1993.

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

[28] C. Schmid and R. Mohr. Combining greyvalue invariants with local constraints for object recognition. In *Proceedings, IEEE Conference on Computer Vision and Pattern Recognition*, pages 872–877, San Francisco, CA, June 1996.

[29] 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.

[30] K. Siddiqi, A. Shokoufandeh, S. Dickinson, and S. Zucker. Shock graphs and shape matching. *International Journal of Computer Vision*, 30:1–24, 1999.

[31] F. Solina and R. Bajcsy. Recovery of parametric models from range images: The case for superquadrics with global deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(2):131–146, 1990.

[32] D. Terzopoulos and D. Metaxas. Dynamic 3D models with local and global deformations: Deformable superquadrics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(7):703–714, 1991.

[33] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86, 1991.

[34] M. Weber, M. Welling, and P. Perona. Unsupervised learning of models for recognition. In *European Conference on Computer Vision*, volume 1, pages 18–32, 2000.

[35] P. Winston. Learning structural descriptions from examples. In *The Psychology of Computer Vision*, chapter 5, pages 157–209. McGraw-Hill, 1975.

[36] S. Zhu and A. L. Yuille. Forms: a flexible object recognition and modelling system. *International Journal of Computer Vision*, 20(3):187–212, 1996.