Skeletal Shape Abstraction from Examples

M. Fatih Demirci, Ali Shokoufandeh, *Member*, *IEEE*, and Sven J. Dickinson, *Member*, *IEEE*

Abstract—Learning a class prototype from a set of exemplars is an important challenge facing researchers in object categorization. Although the problem is receiving growing interest, most approaches assume a one-to-one correspondence among local features, restricting their ability to learn true abstractions of a shape. In this paper, we present a new technique for learning an abstract shape prototype from a set of exemplars whose features are in many-to-many correspondence. Focusing on the domain of 2D shape, we represent a silhouette as a medial axis graph whose nodes correspond to "parts" defined by medial branches and whose edges connect adjacent parts. Given a pair of medial axis graphs, we establish a many-to-many correspondence between their nodes to find correspondences among articulating parts. Based on these correspondences, we recover the abstracted medial axis graph along with the positional and radial attributes associated with its nodes. We evaluate the abstracted prototypes in the context of a recognition task.

Index Terms—Shape abstraction, medial axis graphs, prototype learning, many-to-many graph matching.

1 Introduction

Object categorization requires prototypical models that are invariant to within-class changes of appearance and shape. In the domain of prototypical shape modeling, this translates to models that capture the articulation-invariant, coarse part structure of an object. If a set of exemplars belonging to a given class can be replaced by (or grouped hierarchically under) a single prototype, then the complexity of the recognition task can be greatly reduced through coarse-to-fine search/indexing techniques.

Early shape categorization (or generic object recognition) systems, e.g., [1], [2], constructed such models manually, a challenging and time-consuming task. Since then, a number of researchers have explored the problem of automatically learning a prototypical shape model from a set of exemplars, e.g., [3]. As powerful as these recent techniques are, most are restricted to categorical models whose exemplars share the same features in *one-to-one correspondence*. And although the relative positions of these features may vary slightly, as in the cases of such restricted categories as faces, motorcycles, or cars, such models are typically not invariant to part articulation, image rotation, or scale.

Invariance to deformation, articulation, occlusion, and image transformation can best be achieved through an object-centered structural description which captures the invariant relations among a set of deformation-invariant parts. Like many generations of recognition systems, including the early categorization systems cited above [1], [2], graphs represent a convenient structural abstraction of shape. Relaxing the one-to-one feature correspondence assumption across a category's set of exemplars therefore translates to a many-to-many node correspondence assumption across their corresponding exemplar graphs. If we can find these many-to-many correspondences, we can use them to generate the abstract features (nodes) and relations (edges) of a prototypical graph that meets our requirements for a categorical model.

In this paper, we propose a new framework for automatically learning a shape-based categorical model from a set of exemplar shapes. Assuming that each input exemplar can be represented as a graph, our framework begins by computing the many-to-many node correspondences between a pair of exemplar graphs. From these correspondences, we generate a prototypical graph whose nodes represent abstractions of corresponding feature collections and whose edges represent attachments between the abstractions. This pairwise abstraction process forms the heart of a hierarchical clustering procedure that partitions a set of shapes into classes and computes their abstracted class prototypes.

To ground the method in a particular shape description, we turn to the domain of generating prototypical models of 2D image regions, defined by the shapes of their bounding contours (silhouettes). The silhouette of a region can be decomposed into a *medial axis graph*, whose nodes correspond to "parts" defined by the branches of the region's *medial axis* [4] and whose edges connect adjacent parts. Given a pair of regions represented by their medial axis graphs, our pairwise shape abstraction procedure consists of two steps. First, a structural, many-to-many correspondence between the graphs is established—a challenging problem given that small variations in shape may result in significant variations in graph topology. Moreover, due to noise, or within-class deformation, fragments of one medial axis may not correspond to any fragments in the other. We therefore seek a partial, many-to-many correspondence between the medial axis graphs. Fig. 1 (columns one, two, four, and five) illustrates the many-to-many correspondences computed between two pairs of exemplar graphs (representing the regions shown immediately above).

The second step of the pairwise abstraction process generates the abstract description from the underlying many-to-many correspondences. We compute the abstracted medial axis graph by first computing the averages of the corresponding pairs of subgraphs (collections of skeletal branches) to yield the nodes in the abstracted graph, and then define the overall topology of the resulting abstract parts to yield the relations. Each matching pair of subgraphs corresponds to a single node in the abstracted graph and two abstracted nodes are connected by an edge if the corresponding subgraphs are adjacent in the original graphs. As mentioned earlier, the above two-step procedure forms the basis of an iterative framework in which pairs of similar medial axis graphs are clustered and abstracted, yielding a set of abstract medial axis graph class prototypes. Fig. 1 (columns three and six) illustrates the shape prototypes (and their medial axis graphs below) computed for two pairs of exemplars. The two shape prototypes reflect the main contribution of this work: From a set of examples whose part structure and part shape may differ, we compute a prototype whose part structure and part shape is an *abstraction* of the examples rather than an *intersection* of the examples.

2 Related Work

Shape abstraction (sometimes referred to as shape learning, shape averaging, or shape simplification) has been studied in many different contexts, including contours [5], [6], closed surfaces [7], structural descriptions [8], [9], active shape models [10], graphs [11], constellation models [3], and animation [12]; space allows us to only briefly sample some of the various approaches.

- *M.F. Demirci is with the Department of Computer Engineering, TOBB University of Economics and Technology, Sogutozu Cad. No.: 43, Ankara 06560, Turkey. E-mail: mfdemirci@etu.edu.tr.*
- *A. Shokoufandeh is with the Department of Computer Science, Drexel University, 3141 Chestnut St., Philadelphia, PA 19104. E-mail: ashokouf@cs.drexel.edu.*
- *S.J. Dickinson is with the Department of Computer Science, University of Toronto, 6 King's College Rd., Rm 283B, Pratt Building, Toronto, ON M5S 3G4, Canada. E-mail: sven@cs.toronto.edu.*

Manuscript received 20 Sept. 2007; revised 29 May 2008; accepted 22 Oct. 2008; published online 31 Oct. 2008.
Recommended for acceptance by M. Pelillo.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-2007-09-0615.
Digital Object Identifier no. 10.1109/TPAMI.2008.267.

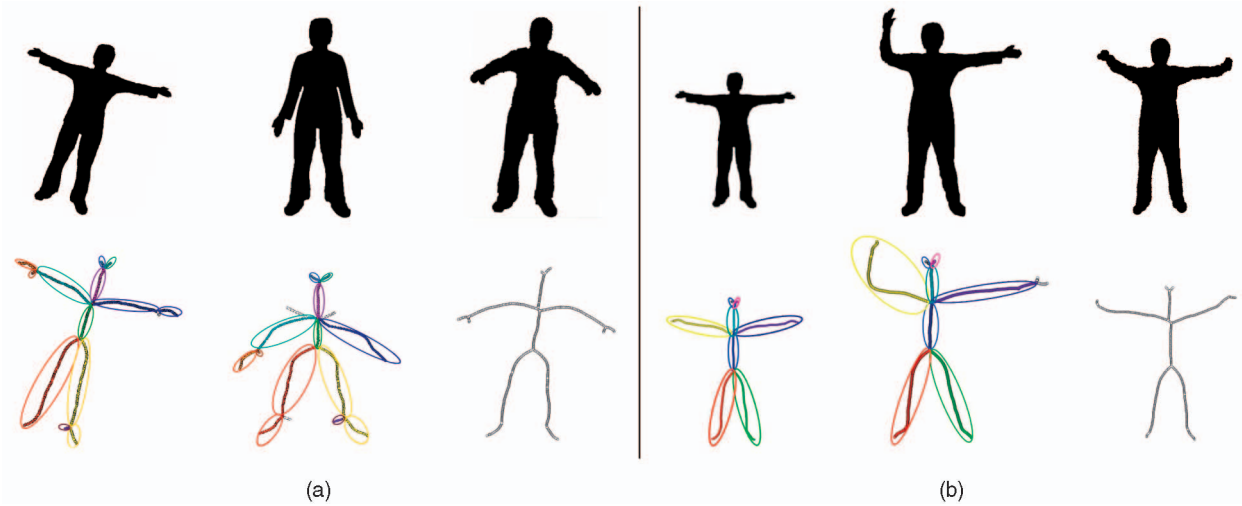


Fig. 1. Computing the shape prototype from two exemplars. The first two columns show two exemplar regions, with varying image rotation and part articulation, and their corresponding medial axis graphs below. Medial branches are partitioned (ellipses) to form the nodes in the graph, and node correspondences computed by the matching are colored (both ellipses and their component medial axis branches) the same. Correspondences between nodes may be many-to-many, as illustrated by the right leg of the person (facing reader) in the first column matching the two right leg parts in the second column. Similarly, the two left arm parts in column one match the left arm in column two. A second example, illustrating invariance to scale and part deformation (bent arm), is illustrated in columns four and five. Columns three and six in each row illustrate the abstract shape prototypes generated from the pairs of exemplars, rendered from the computed abstract medial axis graphs shown below.

In the context of contours, the shape abstraction framework developed by Ueda and Suzuki [5] uses a multiscale dynamic-programming curvature-based approach to find matching curve fragments. The authors form an abstracted curve by constructing it from those points of high curvature that are common across the initial curves, with model curve fragments representing averages of corresponding curve fragments in the initial curves. Active shape models [10] offer a powerful statistical framework for shape abstraction by computing distributions for contour point locations on exemplars belonging to a class. However, like the curve fragment approach above, the technique relies on a one-to-one feature correspondence across exemplars. Moreover, the shapes must be correctly aligned at training time.

Appearance-based models (both global and local) have evolved from exemplar-specific models, due to the specificity of their underlying features, to restricted categorical models, when classes include features that are invariant to within-class variability. For example, Mohan et al. [13] have reported promising results in learning component-based, appearance-based models of humans, supporting their detection in cluttered outdoor scenes. Weber et al. [14] have learned categorical models for such restricted categories as heads, leaves, and cars, while Fei-Fei et al. [15] (and Fergus et al. [3]) learn a generative probabilistic model in the form of a constellation of scale-invariant image patches belonging to an object class. Leibe and Schiele [16] combine both the appearance and shape (contour) of sets of class exemplars to model and categorize isolated objects, while Winn and Jojic [17] also combine shape and appearance, learning a deformable probabilistic index map that captures a notion of parts.

The above appearance-based approaches are very impressive in their ability to cope with real objects in real images. However, although they attempt to learn categorical descriptions, the categorical features are typically not true abstractions of the input exemplar features, but rather consistently appearing local features in one-to-one correspondence. Moreover, the granularity of such local features often lies below that of a category's high-level part structure, limiting the articulation invariance of such models. Finally, most appearance-based modeling approaches are not object-centered, often precluding their invariance to significant changes in image position, rotation, scale, or part articulation.

In the domain of graph algorithms and computer vision, there have been efforts to generate a prototypical graph from a set of exemplars. Jiang et al. introduced the concept of the *median graph*, defined as a graph, drawn from the set, whose sum distance to the other members (graphs) of the set is minimized, and the *set median graph*, a more general concept, which is not constrained to come from the set [11]. Jiang et al. proposed a genetic algorithm for computing the generalized median, while Luo et al. have explored the related problem of graph clustering using a spectral embedding of graphs [18]. It is important to note that these approaches assume that graphs belonging to the same class are structurally similar and do not accommodate the many-to-many correspondences that often reflect significant within-class shape variation.

Keselman and Dickinson [19] overcome this limitation in an approach to shape-based generic model abstraction from exemplars. Starting with a set of region adjacency graphs, the algorithm searches for isomorphic partitions, resulting in a set of abstract (meta)regions that may not have appeared in any input graph. Although effective, the resulting abstraction is not entirely satisfactory as it is obtained directly from the region adjacency graph of one of the exemplars, i.e., the algorithm abstracts only the structure of the region adjacency graphs without abstracting the shapes of its corresponding regions. Levinshtein et al. [20] attempt to abstract a decompositional blob/ridge graph model from examples, again abstracting only the part structure and not the shapes of the parts.

Another form of graph abstraction attempts to explicitly encode structural variability within a class. Specifically, Torsello and Hancock [21] capture the within-class variation in tree structure using a union tree, from which class members can be derived using node/edge removal operations. The union tree is a generative mixture model that is learned from the class members by optimizing a minimum description length criterion. Todorovic and Ahuja [22] apply the union tree to learning a region-based hierarchical object model from a set of exemplar region segmentations. While not as powerful as a union tree in capturing variation, Lozano and Escolano [23] learn a probabilistic graph from a set of example graphs, in which node and edge probabilities reflect relative frequencies of nodes/edges in the example graphs.

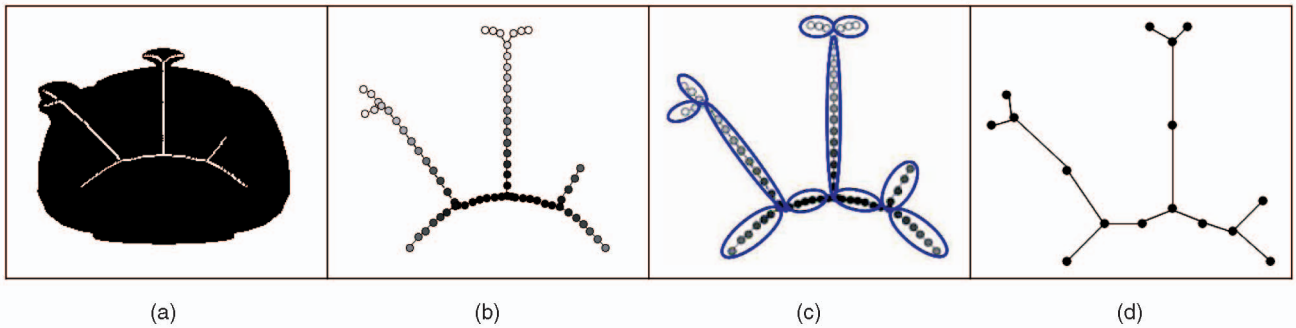


Fig. 2. The construction of the medial axis graph from a region. (a) The silhouette of a teapot and its medial axis. (b) The unrooted tree representation of the medial axis; shock points with larger radii are shown darker. (c) Groupings formed using points of degree 3 or larger as splitting nodes. (d) The medial axis graph formed from the groups.

3 Many-to-Many Matching of Medial Axis Graphs

The shape (silhouette) of a region can be represented as a *medial axis graph*, whose nodes represent branches of the region's medial axis and whose edges represent branch adjacency. The medial axis [4] captures the symmetries of a region, and its branches can be thought of as the region's "parts." More formally, the medial axis is the locus of centers of maximal circles (touching the boundary of the region in at least two places) contained in the region. The center of a maximal circle, whose attributes include position and radius, is called a "shock point" [24]. To compute the set of shock points and their connectivity, we begin with an input shape (Fig. 2a) and apply the algorithm due to Siddiqi et al. [25]. Since we assume that our shape has no holes, this yields an unrooted tree of shock points (Fig. 2b), representing a discrete sampling of the continuous medial axis. Here, each medial axis branch is sampled and represented by a set of discrete medial loci with position and radius encoded. To represent branches as nodes in the medial axis graph, we use the shock points of degree 1 or ≥ 3, which represent the end points of a branch, as illustrated in Fig. 2c. The branches and their connectivities map to nodes in the medial axis graph such that each node either corresponds to a branch in the tree or a point where two or more branches meet (Fig. 2d). This construction ensures that neighboring branches are connected through a node.

Having defined our medial axis graph, we now turn to the problem of finding a matching between the nodes of two such graphs. Most previous approaches [26], [27], [28], [29] to graph matching focus on finding a one-to-one matching between the nodes of two graphs. However, as illustrated in Fig. 1 (top), correspondence is often not one-to-one, but rather many-to-many. Even when part correspondences should be one-to-one, noise and segmentation errors may yield many-to-many correspondences among their features. For medial axis graphs, this means that a single branch (node) in one graph may match a collection of "oversegmented" short branches (nodes) in another graph.

Our algorithm for establishing many-to-many part correspondences among medial axis graphs is an extension of Demirci et al.'s work on many-to-many matching [30]. In that approach, the nodes of two graphs to be matched are embedded into a fixed-dimension Euclidean space, followed by a many-to-many matching between the embedded nodes (points). The first step of the procedure is accomplished using the low-distortion embedding technique of [31], [32], while the second step of the procedure is performed using the Earth Mover's Distance (EMD) [33] under transformation.

The adaptation of the framework [30] to many-to-many part matching of medial axis graphs is based on several key observations. First, we are attempting to match regions at the level of their "parts." Since a part represents a connected subgraph of the medial axis graph, the resulting many-to-many matching between two medial axis graphs must map a connected subgraph into a connected subgraph. Second, the abstraction of two regions should ideally have as many "parts" in common between the regions as possible. As a result, we seek a matching that maximizes the number of pairings of connected subgraphs.[1] Finally, matched "parts" should ideally be as similar to each other as possible. Therefore, we would like to establish a maximal, partial many-to-many matching between connected subgraphs of the medial axis graphs such that the collections of medial axis fragments corresponding to the nodes of the matched subgraphs are maximally similar. The resulting many-to-many matching framework is designed to be robust against minor translation, scale, image rotation, articulation, and within-class deformation, as shown in Fig. 1. Moreover, flow constraints can be incorporated into the EMD algorithm phase to support matching in the presence of occlusion, yielding a partial matching between two medial axis graphs.

Intuitively, for each node (branch) in one medial axis graph, we want to identify the set of zero or more nodes in the second graph to which the node should be assigned. Unfortunately, the correspondences may exist at the level of partial nodes, e.g., two nodes and half of a third node in one graph may map to a single node in another graph. Alternatively, small subsets of shock points associated with one node (in the first graph) may be incorrectly mapped to (i.e., "spread across") shock points belonging to multiple nodes (in the second graph). Our part matching algorithm therefore takes a fine-to-coarse approach, first computing a many-to-many matching between two shock point trees. The solution is a many-to-many mapping from connected shock point subtrees in one graph to connected shock point subtrees in the other graph such that the (normalized) masses of corresponding subtrees is similar and the total work is minimized. It is from these finer granularity (i.e., partial medial axis) correspondences that we compute the final many-to-many correspondences between the coarser nodes in the two medial axis graphs.

The algorithm for mapping many-to-many shock point correspondences (between two shock point trees) to many-to-many medial branch correspondences is based on a simple thresholding procedure. Medial axis graph node A is mapped into medial axis graph node B if: 1) One or more shock points of node A are mapped into one or more shock points of node B and 2) the relative mass of the matched shock points of node A is not negligible ($\geq$ a threshold τ). The thresholding procedure is asymmetric in that a "light" node A may be mapped into a "heavy" node B, but not vice versa. To make the overall matching procedure symmetric, A and B are matched if either A is mapped

1. Given a correspondence between a connected subgraph in one medial axis graph and a connected subgraph in another, the two corresponding subgraphs may not share a single node in correspondence; rather, the correspondence exists at a more abstract level.

procedure mtm-matching($G_1(V, E_1)$, $G_2(U, E_2)$)
  Let A_1 and A_2 be the medial axes corresponding to G_1 and G_2, respectively.
  Compute a many-to-many matching between A_1 and A_2 using EMD (see text).
  for all $v \in V$ **do**
    Let U_v be the set of nodes in G_2 into which a considerable portion of shock points (by mass) represented by v is mapped ($\geq \tau$).
  end for
  for all $u \in U$ **do**
    Let V_u be the set of nodes in G_1 into which a considerable portion of shock points (by mass) represented by u is mapped.
  end for
  Let $G(W, E)$ be a bipartite graph, where $W = V \uplus U$, and $v \in V$ is connected to $u \in U$ iff $u \in U_v$ or $v \in V_u$.
  Let $M = \{W_i\}_i$ be the set of connected components of G ($W_i = V_i \uplus U_i$).
  return M.
end procedure

(a)

procedure abstraction(R_1, R_2)
  Let $G_1(V, E_1)$ and $G_2(U, E_2)$ be the medial axis graphs of R_1 and R_2, respectively.
  Let M be the partial many-to-many matching, computed according to Figure 3(a), that maps $V_i \subset V$ into $U_i \subset U$.
  Let $H(W, E)$ be the resulting abstraction graph (see text).
  for all $W_i = (V_i, U_i) \in W$
    Let $s(W_i)$ be the average of $s(V_i)$ and $s(U_i)$ (see text).
  end for
  Let $A = \cup_i s(W_i)$ be the resulting medial axis-based abstraction.
  Let $R(A)$ be the region corresponding to the medial axis.
  return R.
end procedure

(b)

Fig. 3. Algorithms for (a) many-to-many matching of medial axis and (b) computing the abstraction of two regions.

into B or B is mapped into A. Thus, if several nodes representing small medial axis fragments are mapped into a single node representing a large medial axis fragment (but not vice versa), the large node and the small nodes will be matched to each other. Note that a node in one graph may have no matching nodes in the other graph, yielding a partial matching. The many-to-many matching corresponds to the set of connected components on the bipartite graph whose partitions are the two sets of nodes and whose edges are established according to the node mapping described above. We summarize our approach to partial many-to-many matching of two medial axis graphs in the algorithm given in Fig. 3a.

It is worth noting that, in its ability to match collections of branches many-to-many, our matching framework implicitly groups together the members of a collection. A single part may yield a collection of branches due to various forms of skeletal instability, including ligature [34], or it may be a higher order grouping (abstraction) of a set of stable skeletal branches. If an input medial axis graph is preprocessed to remove skeletal over and undersegmentation due to ligature instability [35], abstractions computed by the many-to-many matching framework would represent higher order groupings and not include lower level skeletal regularization.

4 Pairwise Region Abstraction

The many-to-many matching algorithm described above yields a set of corresponding subgraphs from which an abstract medial axis graph must be computed. Each pair of corresponding medial axis subgraphs defines a node in the abstract medial axis graph. But, we still face the challenge of defining the topology (connectivity) of the nodes to generate a complete description of the abstract structure. Moreover, for each node in the abstract structure, we also need to compute appropriate radial attributes which allow a reconstruction of an associated boundary.

Specifically, let M denote the partial many-to-many matching among the nodes of two graphs, $G_1(V, E_1)$ and $G_2(U, E_2)$, i.e., M maps $V_i \subset V$ to $U_i \subset U$, for $i = 1, \ldots, |M|$. Furthermore, let v_{ij} denote the shortest path distance (in G_1) between V_i and V_j and, similarly, u_{ij} be the shortest path distance (in G_2) between U_i and U_j. Let G be the complete weighted (product) graph whose vertices correspond to pairs (V_i, U_i) $i = 1, \ldots, |M|$ and whose edge weights are defined by $d_{ij} = \min(v_{ij}, u_{ij})$, $i, j \in \{1, \ldots, |M|\}$. The nodes of G capture the set of possible pairings between parts in G_1 and G_2, while the edges in G reflect the proximity between the parts. To obtain the topology of the abstraction between G_1 and G_2, we seek a minimal substructure of G that spans all of its nodes. Specifically, the abstraction H of G_1 and G_2 is computed as the minimum connected spanning subgraph on the complete graph G.

The above procedure is illustrated in Fig. 4. The many-to-many graph matching of G_1 and G_2 in Fig. 4a yields a set of many-to-many node correspondences (similarly colored), along with a set of unmatched nodes (uncolored). Each pair of matched nodes in a given many-to-many correspondence yields a node in the complete graph shown in Fig. 4b. For example, the many-to-many correspondence mapping $\{v_3, v_4\} \in G_1$ to $\{u_4\} \in G_2$ yields two nodes in the complete graph, corresponding to vertices v_3u_4 and

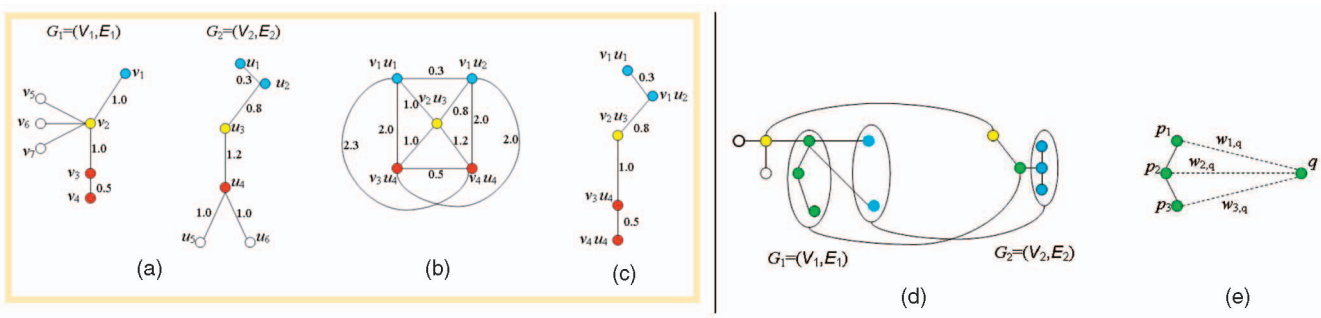


Fig. 4. Defining the structural (left) and medial (right) abstractions. The pairs of matched vertices of graphs G_1 and G_2, shown in (a), will form the nodes of the complete graph G, shown in (b). The minimum spanning tree H, shown in (c), defines the topology (connectivity) of the resulting nodes of the abstract structure. The ordered subsets of matched shock points P and Q (green vertices in (d)) will result in (e) a mapping for the ordered sets. We will use this mapping for computing the abstraction Z of P and Q. The weight $w_{i,q}$ of each edge in the mapping is determined by the many-to-many matching algorithm.

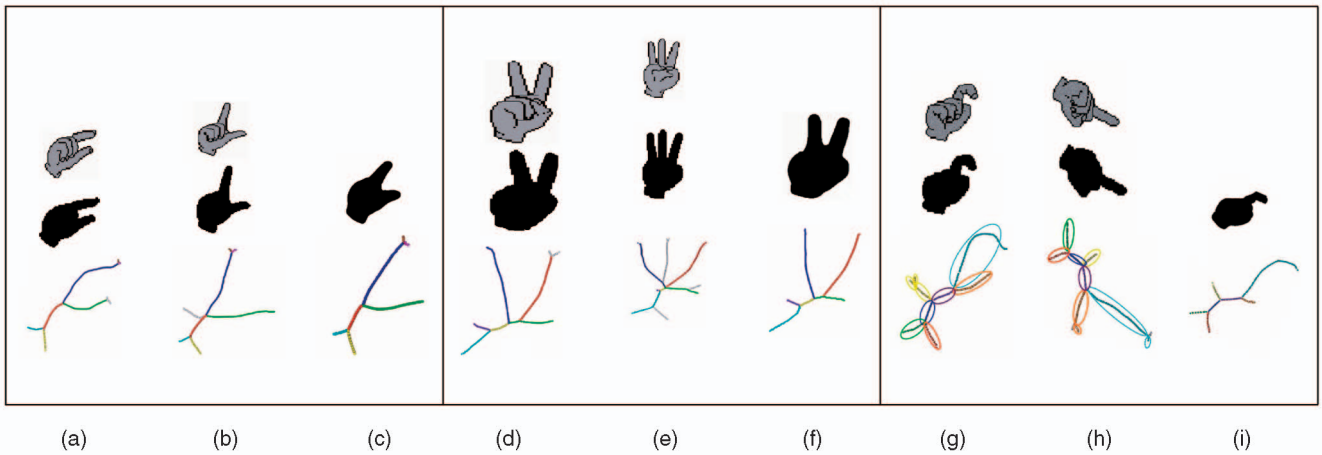


Fig. 5. Computing the abstractions of hand gestures. The abstraction (c) of gestures (a) and (b) illustrates the algorithm's ability to cope with articulation. The top row illustrates the two gestures, the middle row their corresponding silhouettes, and the bottom row their medial axis graphs, colored according to the computed many-to-many shock point correspondences. The abstraction (f) of gestures (d) and (e) illustrates the algorithm's ability to extract the common, salient coarse structure. Finally, abstraction (i) of (g) and (h) illustrates the algorithm's ability to cope with image rotation, part articulation, as well as significant structural differences in medial axis graph topology.

v_4u_4. The resulting minimum spanning tree H of G, shown in Fig. 4c, defines the topological structure of the final medial axis graph.

Having defined the topology of the abstract medial axis graph, we now proceed to define the contents of its nodes, including shock points and their positions and radii. Recall that, for a given node, the many-to-many matching of two medial axis graphs implicitly defines two sets of corresponding shock points in the graphs' underlying medial axes (Fig. 3). It is from these corresponding point sets that we create a new set of points for the node, and define their attributes as a weighted function of the attributes of the points in the two point sets. More formally, given two shock sequences $P = \langle p_1, \ldots, p_n \rangle$ and $Q = \langle q_1, \ldots, q_m \rangle$, with $n \leq m$, our goal is to compute a new shock sequence $Z = \langle z_1, \ldots, z_\ell \rangle$, specifying the number ℓ of points in Z and the coordinates x, y and radius r for each $z_i \in Z$.

To form the abstraction Z from sequences P and Q, we will use an *oriented averaging* procedure that is similar to the Lagrangian model for interpolating real-valued functions [36]. In this procedure, a new point is added to Z for each correspondence between points in P and Q. This ensures that the abstraction is constructed through each correspondence between the input sequences. More precisely, in order to generate the average Z, we begin by forming a mapping between the points in P and Q by computing an EMD-based many-to-many matching between them [33].[2] Assume, without loss of generality, that $P_q = \langle p'_1, \ldots, p'_k \rangle$ denotes an ordered subset of points in P that are matched to a single point $q \in Q$. We note that, for each $p \in P_q$ corresponding to $q \in Q$, a weight $w_{p,q}$ will also be computed by the EMD mapping. An illustration of this step for the two subsets of nodes in Fig. 4d is shown in Fig. 4e. Since there are three mappings from P to Q, Z will contain three medial points, each of whose attributes will be a normalized, weighted average of the attributes of the two points involved in the mapping, with the weights, in turn, based on the flows ($w_{p,q}$) defined by the mapping.

Specifically, to form the average ordered set, for each correspondence (P_q, q), we add one point z to Z whose attributes r_z, x_z, y_z are calculated as follows:

$$r_z = \frac{1}{2w_z}\left[\sum_{p \in P_q} r_p \times w_{p,q} + r_q\right], \quad x_z = \frac{1}{2w_z}\left[\sum_{p \in P_q} x_p \times w_{p,q} + x_q\right],$$

$$y_z = \frac{1}{2w_z}\left[\sum_{p \in P_q} y_p \times w_{p,q} + y_q\right],$$

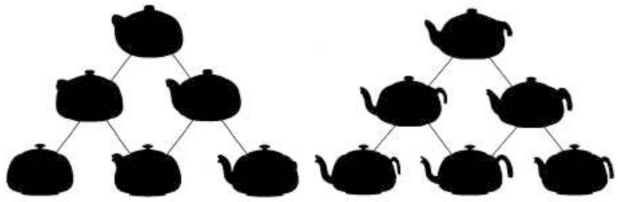
where $w_z = \sum_{p \in P_q} w_{p,q}$. In general, the number of points ℓ in the ordered set Z is bounded by $n \times m$ and is a function of the number of points in $p \in P$ and $q \in Q$ that participate in pairings (small weight pairings may be pruned). The overall algorithm for computing the abstraction of two regions is presented in Fig. 3b. Columns three and six in Fig. 1 illustrate averages computed for the pairs of shapes shown in columns one and two and columns four and five, respectively. We should note that the question of ensuring that the skeletal abstraction is equal to the skeleton generated by the abstraction's reconstructed shape is open. Although the derived representation, in our averaging procedure, could be enforced to be a true medial axis by incorporating appropriate constraints, this is beyond the scope of this paper.

We illustrate our framework for pairwise shape abstraction on pairs of shapes representing human gestures. Fig. 5 illustrates the computed abstraction for two articulations of the same gesture. Figs. 5a and 5b show the gestures, their corresponding silhouettes, and their computed medial axis graphs, respectively. Fig. 5c shows the abstracted medial axis graph (below) and the shape generated from it (above). The abstracted shape clearly represents an intermediate state of the finger articulation, illustrating the articulation invariance of the framework. Figs. 5d, 5e, and 5f illustrate the ability of the framework to abstract the salient common structure from two shapes, yielding a shape that captures the common two-finger structure. Note the significant differences in topology between the two input shapes and the many-to-many matching algorithm's ability to find corresponding groups of skeletal fragments. Figs. 5g, 5h, and 5i illustrate the algorithm's rotation invariance, articulation invariance, and the ability of the many-to-many matcher to compute many-to-many correspondences (individual branches are delineated by ellipses, and many-to-many branch correspondences are colored the same).

2. Note that this is not the original mapping between P and Q computed by the many-to-many matching algorithm, which may include flows directed from/toward points not in P and Q. Instead, a new EMD mapping is computed using just the points in P and Q.

procedure prototypes($\{v_1, \ldots, v_n\}, \tau$)
  Let $T = (V(T), E(T))$ denote the hierarchical structure of prototypes.
  set $\mathcal{L}_1 \leftarrow \{v_1, ..., v_n\}$.
  set $\ell \leftarrow 0$.
  repeat
    $\ell \leftarrow \ell + 1$.
    set $V(T) \leftarrow V(T) \cup \mathcal{L}_\ell$.
    set $\mathcal{L}_{\ell+1} \leftarrow \mathcal{L}_\ell$.
    Let $\delta(u, v)$ denote the dissimilarity score between views u and v, computed by many-to-many matching algorithm.
    for all $v \in \mathcal{L}_\ell$
      Let mate$(v) \leftarrow \text{Argmin}_{u \in \mathcal{L}_l, u \neq v} \delta(u, v)$.
      Let $w \leftarrow$abstraction $(v, \text{mate}(v))$.
      if $(\delta(v, w) < \tau)$ **and** $(\delta(\text{mate}(v), w) < \tau)$
        set $\mathcal{L}_{\ell+1} \leftarrow \mathcal{L}_{\ell+1} - \{v, \text{mate}(v)\} \cup \{w\}$.
        set $E(T) \leftarrow E(T) \cup \{\langle w, v \rangle, \langle w, \text{mate}(v) \rangle\}$.
      end if
    end for
  until $(|\mathcal{L}_{\ell+1}| = 1)$ **or** $(\mathcal{L}_{\ell+1} = \mathcal{L}_\ell)$ **or** $(|\mathcal{L}_{\ell+1}| > |\mathcal{L}_\ell|)$
  if $(|\mathcal{L}_{\ell+1}| = 1)$
    set $V(T) \leftarrow V(T) \cup \mathcal{L}_\ell$.
  return T.
end procedure

(a)



(b)

Fig. 6. Computing the shape prototypes. (a) The algorithm for Computing Class Prototypes for Views $\{v_1, \ldots, v_n\}$ and Threshold τ. (b) Example of computing shape prototypes for a set of teapot images. The original images are shown at the bottom (leaf level), while their averages define the internal nodes. Pairwise averaging continues until the prototype is sufficiently dissimilar from one of its descendants. In this case, six views give rise to two prototypes.

5 Constructing Class Prototypes

The pairwise abstraction model presented in Section 4 is used as a building block to generate representative abstractions of a set of views associated with an object class. Since the many-to-many matching algorithm provides a quantitative measure of dissimilarity between two views, one can repeatedly select a subset of maximally similar pairs and compute their corresponding abstracted views. The resulting hierarchical structure, called the *abstraction hierarchy*, captures the maximally representative prototypes for subsets of views in an object class.

To formalize this construction for a given object class consisting of n views $\{v_1, \ldots, v_n\}$, we first compute their medial axis graphs. To initialize the abstraction hierarchy, the graphs corresponding to the original views $\{v_1, \ldots, v_n\}$ populate the leaf level. Using our many-to-many matching algorithm outlined in Section 3, we compute an $n \times n$ distance matrix where each entry (i, j) defines the dissimilarity score between views (graphs) v_i and v_j. For each graph, we determine its most similar graph (mate) according to the distance matrix and use our framework to compute the average views between graphs and their mates. We then place these average views at the next level up in the abstraction hierarchy, with edges between the levels indicating which graph pairs are used to compute the averages. Repeating the same process at every level results in a hierarchical structure with the root corresponding to the prototype shape for the entire object class.

Due to subsampling and partial matching of the medial axes, some average views in the abstraction hierarchy may not reflect the topological similarities of the original graphs. To ensure that the graphs are properly represented by their averages, we use our matching algorithm to compute the dissimilarity score between an average and its two children. If the dissimilarity score is greater than a predefined threshold, the average graph is not inserted into the tree; instead, both of its children become class prototypes. Like any hierarchical clustering algorithm, the choice of threshold affects the number and sizes of clusters (in our case, the number of prototypes and number of descendants). The algorithm for computing the prototypes for a set of n views and a threshold τ is presented in Fig. 6a. Fig. 6b illustrates the construction of two shape prototypes from six views of a teapot.

6 Experiments

In this section, we evaluate our shape prototypes in the context of an object recognition experiment. We begin with a set of 900 object silhouettes, representing distinct views of a collection of nine 3D objects. Every second view is removed to form a set of 450 query views and a remaining database of 450 views. For the 450 database views, we compute a set of class prototypes and their underlying abstraction hierarchies.[3]

Two types of experiments are run. In a recognition trial, the query is compared to each prototype. If the parent 3D object of the closest prototype is the same as the parent of the query view, then the trial is successful. In a pose estimation trial (essentially a more stringent form of recognition trial), the closest prototype is found (as in the recognition trial) and its underlying abstraction hierarchy is searched (following the path defined by closest matching internal nodes) in a top-down manner until the closest leaf is found. If this leaf represents a neighboring view of the query view (on the same 3D object), then the trial is successful. Finally, for each experiment, we compute performance as a function of sampling resolution of the views in the database.

To evaluate the efficacy of our shape prototypes, we compare their performance in the above experiment to two competing frameworks. In the first framework, we replace each prototype with the median view drawn from the leaves of its underlying abstraction hierarchy, i.e., from the set of all views used to construct the prototype, we choose that view whose sum distance to all other views is minimum. In the second framework, we eliminate the prototypes and compare the query directly to each of the database views (leaves). Our prediction is that, when the view sampling resolution is sparse, our computed prototypes should yield better performance than the median prototypes since, on average, they are closer to the queries.[4] When the sampling resolution increases, we

3. Note that, for any given prototype, its descendants (views) are constrained to belong to the same 3D object.

4. This effect is particularly pronounced in such degenerate cases as two samples or a set of samples forming a circle.

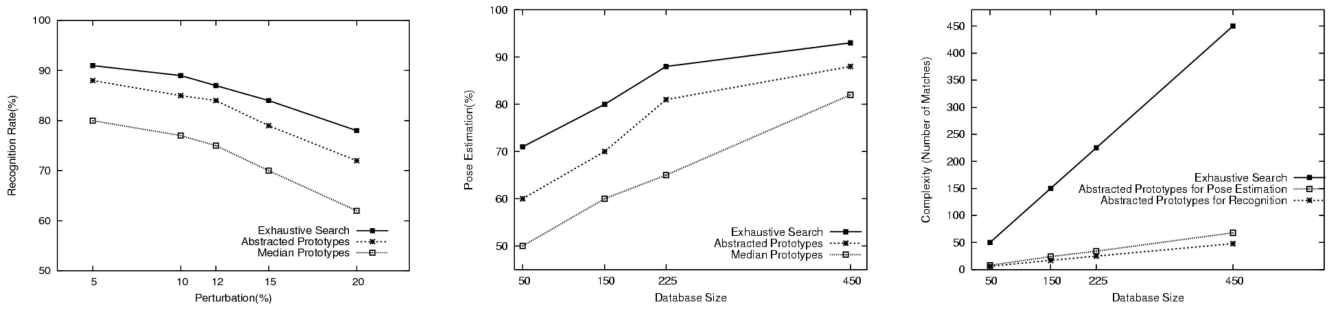


Fig. 7. Experimental results. (a) Recognition rates as a function of increasing database size for exhaustive search, abstracted prototypes, and median prototypes. (b) Correct pose estimation rates as a function of increasing database size for exhaustive search, abstracted prototypes, and median prototypes. (c) The complexity (number of matches) of the recognition and pose estimation algorithms as a function of increasing database size. Since both prototype algorithms are based on the same number of prototypes and the same descendant leaves, we compare exhaustive search only to recognition and pose estimation using our abstracted prototypes. Note that the exhaustive search has the same complexity when applied to either recognition or pose estimation; hence, only one curve is shown.

expect the median prototypes' performance to improve as they move closer to the means (our prototypes). However, the medians are still exemplars and not abstractions and we therefore expect their performance to be consistently inferior to that of our computed prototypes. For the flat, "prototype-less" database, we expect performance to exceed either prototype method (queries will always be closer to their neighboring views than to any type of prototype) at the cost of increased search complexity.

The results are consistent with our predictions. The recognition and pose estimation rates as a function of increasing database size for each method are reported in Figs. 7a and 7b, respectively. As view sampling resolution increases, the performance gain of the abstracted prototypes decreases, although it does remain superior. As predicted, maximum recognition and pose estimation rates are achieved with the flat database. However, as shown in Fig. 7c, which plots the number of model comparisons (matches) computed by each method, both prototype frameworks perform 85-90 percent fewer matches, yielding far greater efficiency at the cost of roughly a 10 percent drop in correctness.

To test or framework on a second domain, we apply it to the MPEG-7 CE-Shape-1 (Part B) database, consisting of 1,400 shapes clustered into 70 classes with 20 shapes per class. Again, correct recognition is achieved when the closest shape to the query belongs to the same class; since the shapes in a class represent different exemplars, computing a pose estimation rate is not applicable. The results, shown in Table 1, once again reflect the increased recognition rate afforded by our abstract prototypes over their median counterparts, again falling slightly short of an exhaustive search over the entire set of exemplars (at approximately an order of magnitude greater cost).

To evaluate the sensitivity of our algorithm to noise, we perturb our input shapes (in the first experiment) by adding noise to the graphs from which we construct the prototypes. Specifically, we add nodes to the graph amounting to 5, 10, 12, 15, and 20 percent of the original size of the graph. To add K percent noise, we randomly choose a leaf node of the medial axis graph (tree) and add K/10 percent nodes as children of the leaf. If the mass of the parent is m, the mass of each added child is $1/m$. The process is repeated, selecting another leaf at random, until K percent of the size of the original graph has been added. As shown in Fig. 8, recognition and pose estimation rates for the abstracted prototypes continue to exceed that for the median prototypes, but still fall slightly short of the exhaustive search.

TABLE 1
Recognition Results on MPEG-7 CE-Shape-1 (Part B)

Experiment	Abstract	Median	Exhaustive
recognition	81.6%	74.7%	88.4%

Our recognition experiment on MPEG-7 CE-Shape-1 indicates that the abstracted prototypes continue to outperform the median prototypes, yet still fall short of an exhaustive search (which requires approximately an order of magnitude greater search complexity).

On the issue of scalability, there are two issues. The first concerns the scalability of the abstraction process, i.e., how the process scales as new exemplars are added to a training set. The second concerns the recognition component, i.e., how recognition performance scales with the number of classes. The latter issue is a function of the particular indexing/recognition strategy, and not the abstraction process. The former issue affects the complexity of the pairwise abstraction process. For N input exemplars, the number of pairwise abstractions is bounded by the size of the tree with N leaves, which is $O(N)$.

7 Limitations

While our approach to computing shape prototypes from exemplars offers invariance to translation, rotation, scale, and articulation, and supports many-to-many part correspondences, working with silhouettes assumes that they have been correctly segmented or presented in isolation. This is a very strong assumption, and not the one shared by those, for example, attempting to recover restricted part-based models (e.g., constellations) from natural images [3]. As the constraints on an object's location in an image are reduced, by allowing it to articulate, change size, translate/rotate in the image, etc., and as the constraints on an object's structure are reduced, by not insisting that the model's parts map one-to-one to image parts, the model becomes weaker and less like a template. As a result, it becomes more difficult to use the model to perform figure/ground segmentation in cluttered or natural scenes. By making our model more flexible and less constrained, we therefore give up the ability (until such time as region segmentation methods can correctly separate figure from ground) to learn a shape prototype from exemplars appearing as real objects in real scenes. While our method is designed to abstract a shape prototype from a set of examples which may not share a subset of input features, it does assume that conditions like occlusion, clutter, and noise (i.e., noise that is due to poor segmentation rather than within-class variation) can be avoided.

Our abstracted medial axis graphs are designed to support object categorization (including both indexing and matching) and not visualization or compression. Thus, no attempt has been made to optimize reconstruction error (of the abstracted shape with respect

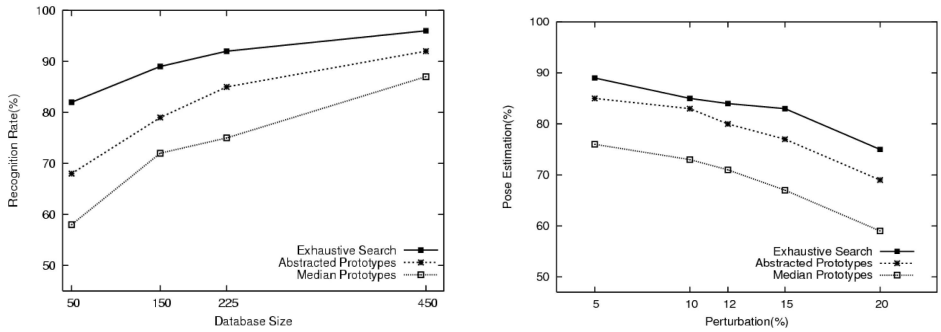


Fig. 8. Noise sensitivity. (a) Recognition rates as a function of increasing additive noise (percent of original graph size) for exhaustive search, abstracted prototypes, and median prototypes. (b) Correct pose estimation rates as a function of increasing additive noise (percent of original graph size) for exhaustive search, abstracted prototypes, and median prototypes.

to the input exemplars). Rather, our goal is optimize recognition performance which, in a recognition-by-parts paradigm, means optimizing structural similarity. While it is certainly true that reconstruction error is implicitly included in our mechanism for defining the medial points in a part, it should be noted that optimizing purely on the basis of reconstruction error may, in fact, work against our goal of optimizing recognition performance. On a related fidelity note, no attempt has been made to ensure that the skeletal abstraction is equal to the skeleton generated by the abstraction's reconstructed shape, or to ensure that the object angles around a branch point sum to 360 degrees. Even though the reconstructed shape is prototypical, by definition, it is still an exemplar, and like all exemplars, will yield skeletal instabilities. Across a set of exemplars, it is precisely these instabilities that we attempt to avoid through abstraction.

8 Conclusions

Generic object recognition (or object categorization) systems must be able to recognize objects based on their prototypical shape. Moreover, the matching of image and model features should be invariant to image translation, image rotation, image scaling, object articulation, and object occlusion. However, if such systems assume that a salient feature in the image maps one-to-one to a salient feature on the model, then object models are constrained to be flexible, object-centered templates of local image features. Overcoming this restriction requires a relaxing of the one-to-one assumption, allowing image-model feature correspondence to be many-to-many.

The medial axis graph is a representation framework that satisfies the invariance criteria outlined above, and the many-to-many matching framework overcomes the one-to-one feature matching restriction common to most recognition frameworks. Together, they provide the foundation for a framework for learning a skeletal shape abstraction from a set of exemplars. The computed prototypes are shown to outperform prototype exemplars, and offer an effective means for organizing a set of shapes in a coarse-to-fine manner. Shock graph-based object recognition is a mature subcommunity, e.g., [29], [37], [27], and the ability to abstract a set of prototypes from examples offers a powerful tool for improving search efficiency.

Acknowledgments

F. Demirci gratefully acknowledges the support of TÜBİTAK grant no. 107E208, A. Shokoufandeh the support of the US Office of Naval Research and US National Science Foundation (NSF), and S. Dickinson the support of NSERC, PREA, NSF, and CITO.

References

[1] R. Brooks, "Model-Based 3D Interpretations of 2D Images," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 5, no. 2, pp. 140-150, 1983.
[2] R. Nevatia and T. Binford, "Description and Recognition of Curved Objects," *Artificial Intelligence,* vol. 8, pp. 77-98, 1977.
[3] R. Fergus, P. Perona, and A. Zisserman, "Object Class Recognition by Unsupervised Scale Invariant Learning," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* vol. 2, pp. 264-271, June 2003.
[4] H. Blum, "A Transformation for Extracting New Descriptors of Shape," *Models for the Perception of Speech and Visual Form,* W. Wathen-Dunn, ed., pp. 362-380, MIT Press, 1967.
[5] N. Ueda and S. Suzuki, "Learning Visual Models from Shape Contours Using Multiscale Convex/Concave Structure Matching," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 15, no. 4, pp. 337-352, Apr. 1993.
[6] N. Duta, A. Jain, and M.P. Dubuisson-Jolly, "Learning 2D Shape Models," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* vol. 2, pp. 8-14, June 1999.
[7] E. Chen and R. Parent, "Shape Averaging and Its Applications to Industrial Design," *IEEE Computer Graphics and Applications,* vol. 9, no. 1, pp. 47-54, Jan. 1989.
[8] P. Winston, "Learning Structural Descriptions from Examples," *The Psychology of Computer Vision,* chap. 5, pp. 157-209, McGraw-Hill, 1975.
[9] G. Ettinger, "Large Hierarchical Object Recognition Using Libraries of Parameterized Model Sub-Parts," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 32-41, 1988.
[10] T.F. Cootes, C.J. Taylor, D.H. Cooper, and J. Graham, "Active Shape Models—Their Training and Application," *Computer Vision and Image Understanding,* vol. 61, no. 1, pp. 38-59, 1995.
[11] X. Jiang, A. Munger, and H. Bunke, "On Median Graphs: Properties, Algorithms, and Applications," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 23, no. 10, pp. 1144-1151, Oct. 2001.
[12] N. Burtnyk and M. Wein, "Interactive Skeleton Techniques for Enhancing Motion Dynamics in Key Frame Animation," *Comm. ACM,* vol. 19, no. 10, pp. 564-569, 1976.
[13] A. Mohan, C. Papageorgiou, and T. Poggio, "Example-Based Object Detection in Images by Components," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 23, no. 4, pp. 349-361, Apr. 2001.
[14] M. Weber, M. Welling, and P. Perona, "Unsupervised Learning of Models for Recognition," *Proc. Sixth European Conf. Computer Vision,* vol. 1, pp. 18-32, citeseer.nj.nec.com/weber00unsupervised.html, 2000.
[15] L. Fei-Fei, R. Fergus, and P. Perona, "Learning Generative Visual Models from Few Training Examples: An Incremental Bayesian Approach Tested on 101 Object Categories," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* June 2004.
[16] B. Leibe and B. Schiele, "Analyzing Appearance and Contour Based Methods for Object Categorization," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2003.
[17] J. Winn and N. Jojic, "Locus: Learning Object Classes with Unsupervised Segmentation," *Proc. 10th IEEE Int'l Conf. Computer Vision,* vol. 1, pp. 756-763, 2005.
[18] B. Luo, R. Wilson, and E. Hancock, "Spectral Embedding of Graphs," *Pattern Recognition,* vol. 36, no. 10, pp. 2213-2230, 2003.
[19] Y. Keselman and S. Dickinson, "Generic Model Abstraction from Examples," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 27, no. 7, pp. 1141-1156, July 2005.
[20] A. Levinshtein, C. Sminchisescu, and S.J. Dickinson, "Learning Hierarchical Shape Models from Examples," *Proc. Int'l Workshop Energy Minimization Methods in Computer Vision and Pattern Recognition,* pp. 251-267, 2005.
[21] A. Torsello and E.R. Hancock, "Learning Shape-Classes Using a Mixture of Tree-Unions," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 28, no. 6, pp. 954-967, June 2006.

[22] S. Todorovic and N. Ahuja, "Unsupervised Category Modeling, Recognition, and Segmentation in Images," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 30, no. 12, pp. 2158-2174, Dec. 2008.
[23] M. Lozano and F. Escolano, "Protein Classification by Matching and Clustering Surface Graphs," *Pattern Recognition,* vol. 39, no. 4, pp. 539-551, 2006.
[24] B.B. Kimia, A.R. Tannenbaum, and S.W. Zucker, "Shapes, Shocks, and Deformations I: The Components of Two-Dimensional Shape and the Reaction-Diffusion Space," *Int'l J. Computer Vision,* vol. 15, no. 3, pp. 189-224, 1995.
[25] K. Siddiqi, S. Bouix, A. Tannenbaum, and S.W. Zucker, "Hamilton-Jacobi Skeletons," *Int'l J. Computer Vision,* vol. 48, no. 3, pp. 215-231, 2002.
[26] L.G. Shapiro and R.M. Haralick, "Structural Descriptions and Inexact Matching," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 3, pp. 504-519, 1981.
[27] M. Pelillo, K. Siddiqi, and S. Zucker, "Matching Hierarchical Structures Using Association Graphs," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 21, no. 11, pp. 1105-1120, Nov. 1999.
[28] S. Gold and A. Rangarajan, "A Graduated Assignment Algorithm for Graph Matching," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 18, no. 4, pp. 377-388, Apr. 1996.
[29] K. Siddiqi, A. Shokoufandeh, S. Dickinson, and S. Zucker, "Shock Graphs and Shape Matching," *Int'l J. Computer Vision,* vol. 30, pp. 1-24, 1999.
[30] M.F. Demirci, A. Shokoufandeh, Y. Keselman, L. Bretzner, and S. Dickinson, "Object Recognition as Many-to-Many Feature Matching," *Int'l J. Computer Vision,* vol. 69, no. 2, pp. 203-222, 2006.
[31] R. Agarwala, V. Bafna, M. Farach, M. Paterson, and M. Thorup, "On the Approximability of Numerical Taxonomy (Fitting Distances by Tree Metrics)," *SIAM J. Computing,* vol. 28, no. 2, pp. 1073-1085, 1999.
[32] J. Matoušek, "On Embedding Trees into Uniformly Convex Banach Spaces," *Israel J. Math.,* vol. 237, pp. 221-237, 1999.
[33] Y. Rubner, C. Tomasi, and L.J. Guibas, "The Earth Mover's Distance as a Metric for Image Retrieval," *Int'l J. Computer Vision,* vol. 40, no. 2, pp. 99-121, 2000.
[34] J. August, K. Siddiqi, and S.W. Zucker, "Ligature Instabilities in the Perceptual Organization of Shape," *Computer Vision and Image Understanding,* vol. 76, no. 3, pp. 231-243, 1999.
[35] D. Macrini, K. Siddiqi, and S. Dickinson, "From Skeletons to Bone Graphs: Medial Abstraction for Object Recognition," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* June 2008.
[36] E. Whittaker and G. Robinson, *The Calculus of Observations: A Treatise on Numerical Mathematics,* fourth ed. Dover, 1967.
[37] T. Sebastian, P.N. Klein, and B. Kimia, "Recognition of Shapes by Editing Their Shock Graphs," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 26, no. 5, pp. 550-571, May 2004.