Classification of Contour Shapes Using Class Segment Sets

Kang B. Sun and Boaz J. Super

Department of Computer Science
The University of Illinois at Chicago
{kbs, super}@cs.uic.edu

Abstract

Both example-based and model-based approaches for classifying contour shapes can encounter difficulties when dealing with classes that have large nonlinear variability, especially when the variability is structural or due to articulation. This paper proposes a part-based approach to address this problem. Bayesian classification is performed within a three-level framework which consists of models for contour segments, for classes, and for the entire database of training examples. The class model enables different parts of different exemplars of a class to contribute to the recognition of an input shape. The method is robust to occlusion and is invariant to planar rotation, translation, and scaling. Furthermore, the method is completely automated. It achieves 98% classification accuracy on a large database with many classes.

1. Introduction

Shape classes that have a large nonlinear variability of global shape, due to structural variation, articulation, or other factors, present a challenge for several existing shape recognition approaches. Approaches that match the target shape to stored example shapes require a large number of stored examples to capture the range of variability [1,10,12,14]. Approaches based on generative models require a large number of parameters, which renders them significantly more expensive computationally, and also increases the possibility of converging to non-optimal local minima [4,5,9]. Furthermore, existing example-based and model-based approaches cannot handle object classes that have different parts or numbers of parts without splitting the class into separate subclasses. This type of structural variation can be handled by approaches that represent part relationships explicitly and match shapes syntactically; however, these structural approaches are computationally expensive [8,13].

This paper proposes a parts-based approach that can recognize classes with large global shape variability. In contrast with example-based approaches, the input shape is not compared to stored example shapes. In contrast with model-based approaches, there is no need to compute a deformation of a shape model or its parts; only planar similarity transformations of model contour segments are needed. In contrast with structural approaches, no explicit information about the relationships among parts is used, just the parts themselves. By using redundant and overlapping sets of parts, high classification accuracy can be achieved without the computational cost of syntactic matching methods.

This paper proposes a three-level statistical framework including distinct models for database, class, and part. Bayesian inference is used to perform classification within this framework. The current implementation classifies shapes consisting of single closed contours. However, there is nothing intrinsic to the approach which limits the method to single closed contours: the class model described below can be constructed from shapes consisting of multiple open and closed contours. An extension of the implementation to multi-contour shapes is in progress.

The class model is key to the method. Each class is modeled not by a set of shape examples but by a set of segments aggregated from multiple shape examples. We call this a *class segment set*. It extends the advantage of the example-based approach by allowing information from different parts of multiple example shapes of a class to contribute to the classification of an input shape of that class (Fig. 1). This enables recognition of input shapes that are partially similar to more than one exemplar but are not similar to any one class exemplar as a whole. It also supports classification of occluded shapes.

The contour segment model is a simple generative model in an invariant segment space. Each segment is normalized with respect to planar similarity transformations (translation, rotation, and scaling), and represented in a high-dimensional segment space of dimension d. The variability around the segment is modeled by an imposed d-dimensional equivariate normal dis-

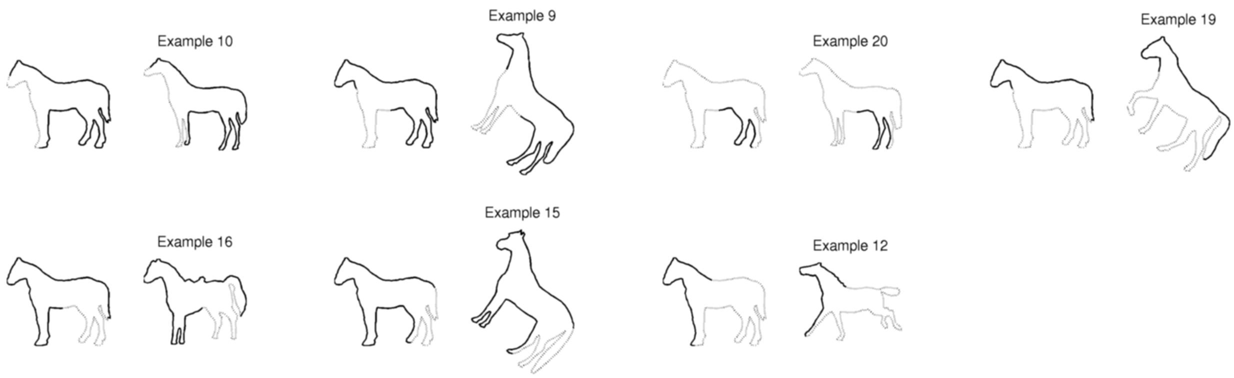

Figure 1. Each panel shows a match of one segment of the same input shape to a segment of a different training example. The method aggregates information from these matches (and others, not shown) to classify the input shape.

tribution. Each such distribution in the invariant segment space corresponds to a specific contour segment from one example shape of one of the classes to be recognized. The segments from each example shape are redundant and overlapping. Each class model can be viewed as a mixture of Gaussian distributions in invariant segment space, derived from all of the segments of all shapes of that class.

The database model is constructed from the union of the class segment sets; i.e., from the set of all segments of all training examples of all classes. From the statistics of this large set, we estimate a mean segment and a variation around the mean. This yields the Mahalanobis distance metric in the invariant segment space. We also compute an ordered set of basis segments by principal component analysis (PCA). These basis segments provide an interpretable model of the variation of contour segments across all classes.

Although this technique has some similarity with statistical models of shapes such as point distribution models (PDMs) [4], our approach applies to segments rather than to shapes, to the entire database rather than to a single class, and it does not require strict landmark correspondences. One consequence of the last point is that the basis segments of the shape database are generated automatically.

This framework has several important advantages. First, both learning and classification are completely automated. Second, it can classify shapes in classes with high variability, as well as partial and occluded shapes. Third, it has few parameters, all of which can be set within broad ranges. Fourth, it is computationally efficient and can handle a large number of shape classes. The empirical evaluation presented in this paper demonstrates that this approach has a low rate of classification errors.

Although the proposed method already has high accuracy, it can also be used as a fast first stage of a multi-stage classifier. The second stage could be a more costly method that makes use of additional information, such as the relationships among the parts (e.g., [13]).

The rest of this paper is organized as follows. The background of the method is discussed in Section 2. The method is described in Section 3 and evaluated experimentally in Section 4.

2. Background

This section briefly reviews recently proposed shape matching methods that have been developed for several different problems, including classification, detection, and retrieval. A general survey of the literature on shape matching can be found in [2].

A number of recent methods for matching shapes have been demonstrated for classification and/or retrieval using various forms of example-based pattern recognition. Belongie et al. [1] proposed the use of "shape contexts," which are log-polar histograms of shape points relative to a given point on the shape. A k-nearest neighbor classification method using automatic selection of class prototypes was demonstrated. Grigorescu et al. [10] used sets of distances to labeled shape points relative to a given point. Sebastian et al. [13] proposed a structural shape representation and matching technique based on shock graphs. Example-based classification is performed using hand-selected prototype shapes from each class. Shock-graph matching is expensive, so a two-stage matching procedure is used: fast, approximate matching followed by full shock-graph matching. Latecki and Lakämper [11] use a structural matching approach to match segments of contours using a discrete curve evolution formalism.

In contrast with the methods just discussed, our approach does not match shapes to shapes. Instead, it

Figure 2. An example class with significant nonlinear variability.

matches the set of segments of the input shape to the class segment sets, each of which consists of the segments from all of the shapes in a class. Thus, our approach is an example-based approach based on parts rather than whole shapes, and based on aggregating parts across the shapes in a class. The individual training shapes are not used at run-time. This strategy extends the example-based approach to be able to recognize input shapes that are not similar to any one training shape, but that may have parts that are similar to different training shapes of a class. It provides a way to handle classes with significant nonlinear variability, caused by, e.g., structural variation or articulation (Fig. 2).

An alternative to the example-based approach is the generative model-based approach, in which a single parameterized class model is learned from a set of training shapes of that class. The class model consists of a typical shape (often a mean shape) and a probability distribution describing departures from the prototypical shape (e.g., [4,5,9]). The advantages of the model-based approach include compactness (it is not necessary to store individual training shapes) and interpretability. However, finding the model and pose parameters that fit the model to the data is computationally expensive, typically requiring iterative methods. It is also prone to getting trapped in local minima. Because of the computational cost, model-based approaches are not well suited for classification when there are many classes. Finally, the training requires sets of corresponding landmarks on the training shapes; missing landmarks and incorrect correspondences can cause failure. These methods are thus not applicable when a class varies sufficiently to have different sets of landmarks on different shapes, although partial progress has been made in addressing this problem [7]. A consequence of this limitation is that the training of many model-based systems is not completely automated.

In contrast with the model-based approach, the parts-based approach presented here classifies shapes without solving for pose and deformation parameters. Furthermore, instead of statistical models of individual shape classes, the statistical models in the current approach are for segments instead of shapes and for the whole database instead of individual classes. In addition, the only landmarks required on a segment are its two endpoints. Due to this simplification of the landmark identification and correspondence problem, the method proposed here is completely automated.

3. Classification Method

This section describes the components of the method: the class segment sets, the invariant segment space representation, the statistical database model, and the Bayesian classifier.

3.1 Class segment sets

Suppose there are M object classes to be recognized, and let c_i denote the ith object class. Let $C_i = \{\omega_{i1},\ldots,\omega_{iN_i}\}$ be a set of training objects for class c_i, where in the current implementation each ω_{ij} is a closed planar contour. Let $\omega(t)=(x(t),y(t))$ denote a contour parameterized by $t \in [0,1]$. For each $\omega_{ij} \in C_i$, let $S(\omega_{ij})$ denote a set of contour segments generated from ω_{ij} by the procedure described below; thus, $S(\omega_{ij}) = \{s_{ij,1},\ldots,s_{ij,m(i,j)}\}$, where $m(i,j)$ is the number of segments on ω_{ij}. We will denote the set of all segments of all contours in a class c_i by $S(C_i)$. Thus,

$$S(C_i) = \bigcup_{j=1}^{N_i} S(\omega_{ij}). \qquad (1)$$

The set $S(C)$ for any class C will be called a *class segment set*. Finally, let

$$S = \bigcup_{i=1}^{M} S(C_i) \qquad (2)$$

denote the set of all segments generated from all example contours of all classes.

3.2 Segment space

3.2.1 Segments. For any contour ω, let $\kappa_\omega(t)$ be the curvature of $\omega(t)$ at t, and let $\Gamma(\omega) = \{u_i\}$, $u_i \in [0,1]$, be the set of parameter values of the extremal points of $\kappa_\omega(t)$. In this paper, the $\{u_i\}$ will be called *critical points*.

The segments of $\omega(t)$ are defined between all ordered pairs of points (u_i,u_j) from Γ, where u_i and u_j do not have to be adjacent. Note that on a closed contour, the union of the segment from u_i to u_j and the segment from u_j to u_i is the whole contour. Segments may be simple or complex depending on the number of critical points they contain. The *complexity* of segment s is defined as $\left|[u_i,u_j] \cap \Gamma(\omega)\right| - 1$, which is one less than the number of critical points in s.

Many contours have critical points that are not significant for analyzing the shape of the contour. These are often due to discretization artifacts and to small-

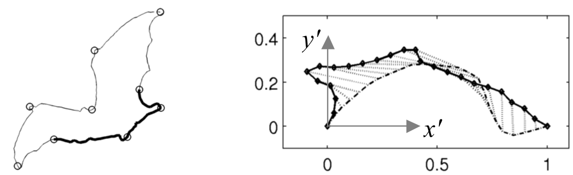

Figure 3. Left: an example of a complex segment of a bat shape. Right: the highlighted segment compared to another segment in the invariant segment space.

scale details on the object. A two step procedure is used to remove less-significant critical points. First, the functions $x(t)$ and $y(t)$ are smoothed by convolving them with a Gaussian (we used $\sigma = 11$). Second, critical points are filtered using a simple procedure, as follows. Let T_i be the triangle formed by $\omega(u_{i-1})$, $\omega(u_i)$, and $\omega(u_{i+1})$, where the indices wrap around the starting point of the contour if necessary. Then, we remove from Γ the critical points for which the area of T_i is less than a threshold that depends on the contour length. This procedure is applied repeatedly until no further points are removed from Γ. An example output is shown on the left side of Figure 3.

Since the input to the system consists of digital contours, each contour is represented by an ordered sequence of sample points. Each input contour is resampled with n_c points at intervals of uniform arc length; i.e., at $\omega(t_i)$, where $t_i = (i-1)/n_c$, for $i = 1,...,n_c$.

3.2.2 Invariant segment representation. In order to achieve invariance to planar similarity transformations (2-D translation, rotation, and uniform scaling), segments are represented in an invariant reference frame.

First, each segment s is sampled with n points at equal intervals of arc length, i.e., $\{\mathbf{x}_1,...,\mathbf{x}_n\}$. Second, the segment s is transformed to another segment s' in an invariant reference frame (x',y'). The two endpoints $(\mathbf{x}_1,\mathbf{x}_n)$ of s provide a natural basis for computing the transformation of s to s': $\mathbf{x}_1$ is mapped to $\mathbf{x}'_1 = (0,0)$ and $\mathbf{x}_n$ is mapped to $\mathbf{x}'_n = (1,0)$, as shown in Figure 3. The coordinates of the remaining segment points in the new reference frame, $\mathbf{x}'_2,...,\mathbf{x}'_{n-1}$, are invariant to translation, rotation, and scaling.

We define a $2n$-dimensional *segment space*, contained in $\mathbf{R}^{2n}$, such that each segment s' is represented by a *segment vector* $\mathbf{v} = (x'_1,...,x'_n,y'_1,...,y'_n)^T$ in that space. Finally, let $V = [\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_{|S|}]$ denote the matrix of the segment vectors $\mathbf{v}_i$ corresponding to all segments in S.

3.3 Statistical database model

We construct a statistical model for S, the entire database of segments. The model yields a measure of the mean and variation of the segments together with a set of decorrelated basis segments; it also provides dimensionality reduction. This is done by performing a principal component analysis (PCA) [6]. First, a mean segment $\mu_\mathbf{v}$ is computed:

$$\mu_\mathbf{v} = \frac{1}{|S|}\sum_{i=1}^{|S|}\mathbf{v}_i . \tag{3}$$

Let $V' = [\mathbf{v}_1-\mu_\mathbf{v}, \mathbf{v}_2-\mu_\mathbf{v},...,\mathbf{v}_{|S|}-\mu_\mathbf{v}]$. Then the eigenvectors and eigenvalues of the $2n\times 2n$ matrix $V'V'^T$ are computed:

$$\Lambda = B'^T V'V'^T B', \tag{4}$$

where B' is the matrix of eigenvectors and Λ is the diagonal matrix of eigenvalues $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_{2n}$. Each eigenvalue λ_i corresponds to the variance σ_i^2 along the ith eigenvector direction. Finally, let $B = [\,\mathbf{b}_1, ..., \mathbf{b}_K\,]$ denote the $2n \times K$ matrix formed from the first K columns of B', where $K \leq 2n$. For any segment vector $\mathbf{v}$, its coefficients in the K-dimensional eigenspace are

$$\mathbf{a} = B^T(\mathbf{v} - \mu_\mathbf{v}). \tag{5}$$

Let $A = [\,\mathbf{a}_1, ..., \mathbf{a}_{|S|}\,]$ denote the *coefficient vectors* of the segments in S. For convenience, we will use the notation $\varphi(s) = \mathbf{a}$ to represent the mapping from a segment to its coefficient vector. The best K-dimensional reconstruction of $\mathbf{v}_i$ (in a least-squares sense) is given by

$$\mathbf{v}_i \approx B\mathbf{a}_i + \mu_\mathbf{v} \tag{6}$$

This gives us the dimensionality reduction, which can increase the speed of classification when $K << 2n$.

The distance between two segments s_1 and s_2 is defined by the Mahalanobis distance in the eigenspace [6]:

$$D(s_1,s_2) = \sqrt{\sum_{j=1}^{K}\left[\left(a_{1j} - a_{2j}\right)/\sigma_j\right]^2}\,, \tag{7}$$

where a_{ij} is the jth component of $\mathbf{a}_i$ for $i = 1,2$.

The columns of B are decorrelated basis vectors; together with the mean vector $\mu_\mathbf{v}$, they provide a statistical model for general contour segments; i.e., segments from all classes. Figure 4 shows the variation of segments along the first few principal modes for the test database used in Section 4. An advantage of decorrelated basis vectors is that they make the model of a general segment more interpretable. The first four components approximately capture height, sideways shearing, diagonal stretching, and inflation/deflation of a segment, respectively. Higher-order components capture more detailed aspects of segment variability across S.

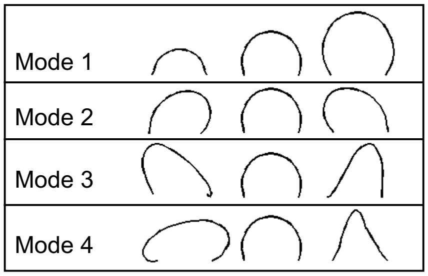

Figure 4. From left to right, each row shows one principal mode of variation over ±0.75 standard deviations. The center column is the mean shape.

3.4 Bayesian classifier

We use a Bayesian model within the three-level framework of segment, class, and database models. In contrast with [3], which also used contour segments and Bayesian classification to perform a recognition task, our method uses class models instead of individual objects, and uses an overlapping redundant set of segments of all complexities that are matched geometrically, instead of a nonoverlapping set of simple segments that are matched using a feature-based method.

Given an input shape ω' to be classified, Bayes' rule is used to compute the posterior probabilities of classes given segments $s' \in S(\omega')$. Then, the posterior probabilities due to all of the segments of ω' are aggregated to yield a rank ordering of the class hypotheses.

By Bayes' rule, the posterior probability is given by:

$$P(c_i \mid \varphi(s')) = \frac{p(\varphi(s') \mid c_i) P(c_i)}{p(\varphi(s'))} \tag{8}$$

where c_i is one of the M classes. The class-conditional probability for observing $\varphi(s')$ given that ω' belongs to class c_i is

$$P(\varphi(s') \mid c_i) = \sum_{s \in S(C_i)} p(\varphi(s') \mid \varphi(s)) P(s \mid c_i). \tag{9}$$

It is not feasible to compute $P(c_i|\varphi(s_1'),...,\varphi(s'_{|S(\omega')|}))$, because the segments of a contour are not independent or conditionally independent (segments can contain or overlap other segments). One possibility is to use the independence model as an ad hoc approximation. However, this introduces numerical problems due to small probabilities which can be handled only by introducing additional parameters; therefore, we use the simpler heuristic of summing the posterior probabilities of a class over the set of segments in the input shape. The class label $\hat{c}$ corresponding to the maximum of this sum is output by the algorithm:

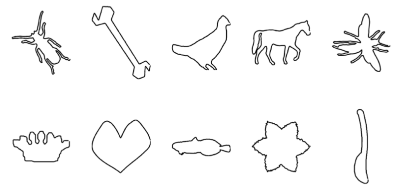

Figure 5. Example shapes from the evaluation dataset [12].

$$\hat{c} = \underset{i=1,\ldots,M}{\arg\max} \sum_{s' \in S(\omega')} P(c_i \mid \varphi(s')). \tag{10}$$

It remains to define specific models for the probabilities in (8) and (9). We assume that all classes are equiprobable, (i.e., $P(c_i) = 1/M$), and that all segments within a class segment set are equiprobable, (i.e., $P(s|c_i) = 1/|S(C_i)|$). (A more accurate but more complex model for $P(s|c_i)$ would assume that the example contours in a class are equiprobable and the segments of an example contour are equiprobable within that contour.) Finally, we assume an isotropic K-dimensional normal density around each coefficient vector in the database, with scale factor ρ:

$$p(\varphi(s') \mid \varphi(s)) = \frac{1}{(2\pi\rho^2)^{K/2}} \exp\left(-\frac{D^2(s', s)}{2\rho^2} \right). \tag{11}$$

Although these assumed distributions are not strictly true, they yield an easily computable approximation that appears to work well empirically.

Note that the Bayesian model connects the segment, class, and database models. Eqn. (11) connects the segment and database models via the estimated Mahalanobis distance D and the mapping φ, and Eqns. (8)-(10) connect the class and segment models.

4. Experimental evaluation

An evaluation of a method for classifying shape contours should satisfy several criteria. First, there should be overlapping classes in the evaluation dataset so that the discrimination problem is non-trivial. Second, there should be significant intra-class variation including, e.g., variation due to geometric differences, structural differences, and articulation. Third, to provide a good test of classification accuracy, the evaluation dataset should consist of many classes. Fourth, the evaluation dataset should be widely available.

A set of contours that meets these criteria is the MPEG-7 evaluation dataset used for testing shape matching [12]. It consists of 70 classes of shapes, with 20 examples per class, for a total of 1,400 shapes. Figure 5 shows a few example shapes from this database. Earlier papers used this database for evaluating shape-to-shape matching methods and reported retrieval accuracy [1,10,12,14]. However, since the method presented in this paper matches shapes to class models instead of shapes to shapes, the retrieval accuracy benchmark does not apply; in fact, it cannot be computed for our approach. Therefore, we report classification accuracy instead. We encourage other authors to also report classification accuracy on this dataset in order to facilitate comparisons of different methods with respect to classification.

A modified leave-one-out evaluation procedure was used. For each one of the 70 classes in the evaluation dataset, one of the 20 shapes in that class was reserved for a test set, and the other 19 shapes in that class were used for a training set. Thus, a complete training set consisted of 1,330 shapes and a complete test set consisted of 70 shapes. This procedure was repeated 20 times with the constraint that every one of the 1,400 shapes was included in the test set once and in the training set 19 times.

Each shape in a given test set was classified using the 70 class segment sets learned from the training set. The *classification accuracy* is defined as the average proportion of test shapes correctly classified.

The method was tested with $n = 25$ sample points per segment, $K = 15$ principal components, and $\rho = 0.01$. The classification accuracy is 97.93%, corresponding to only 29 errors out of 1,400 test inputs. Furthermore, the correct classification is in the top five class hypotheses 99.71% of the time; i.e., for all but 4 of the 1,400 test inputs (Fig. 6).

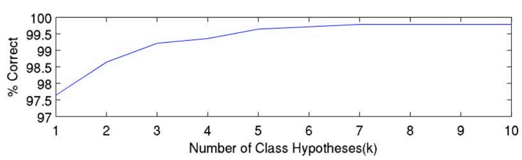

Figure 6. Percent of test shapes returning the correct class label in the top k class hypotheses.

The average time required to compute a classification is approximately 6 seconds, using Matlab on a 1.7 Ghz Xeon-based PC. The time required to set up the database initially is 17 minutes.

We evaluated the robustness of the method to the parameter values. The values of n and K, which control effective resolution, do not affect accuracy as long as the resolution is high enough to capture the shape of the segments. We measured the sensitivity of ρ over two orders of magnitude, and found that classification accuracy is stable over most of that range (Table 1).

Table 1. Classification accuracy vs. ρ.

ρ	0.001	0.005	0.01	0.05	0.1
Accuracy (%)	87.5	97.7	97.9	97.9	95.6

To demonstrate the benefit of using class segment sets, we compared our proposed method with one that was identical except that 'shape segment sets' were used instead of class segment sets. The query shape was compared with each database shape and classified according to the most similar database shape. This method yielded 96 classification errors. Thus, the use of class segment sets reduced the number of errors by more than a factor of three.

Robustness to occlusion is a primary reason for using a parts-based approach. To test performance under occlusion, we introduced random gaps into the test shapes and repeated the classification experiment. The test shapes frequently had multiple gaps. When the average fraction of deleted arc length is one-half, classification accuracy remains relatively high, above 80% (Fig. 7).

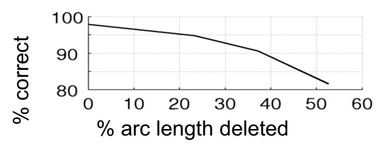

Figure 7. Classification accuracy versus occlusion.

We also tested performance under simultaneous occlusion and clutter. The set of segments from the query shape was augmented with randomly selected segments from other shapes. Table 2 shows the results for 25% average occlusion and different degrees of clutter. The results demonstrate robustness to clutter even in the presence of occlusion.

Table 2. Accuracy under clutter at 25% avg. occlusion.

Query composition: ratio of original query segments to additional clutter segments	Accuracy
1:0	94.8%
1:1	92.4%
1:1.5	91.9%
1:2	90.7%

5. Conclusion

This paper presented a shape classification method based on a three-level framework, with class models that consist of class segment sets, segment models that consist of vectors in an invariant segment space, and a database model consisting of a statistical description of the segments in the database. Bayesian classification is performed within this framework. The class segment sets provide the ability to classify input shapes that have large deviations from the training examples by matching them piecewise to one or more exemplars. The segment space representation provides a method for aligning and comparing segments invariant to planar similarity transformations. The statistical database model provides an interpretable contour segment basis set and a Mahalanobis distance. It also increases efficiency by dimensionality reduction. Empirical evaluation of classification accuracy on a large 70-class dataset demonstrated 98% accuracy and robustness to occlusion and clutter.

6. References

[1] S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape contexts," *IEEE Trans. PAMI* **24**(4) 509-522, 2002.

[2] R. Campbell and P. Flynn, "A survey of free-form object representation and recognition techniques," *Comp. Vision Image Understanding* **81**, 166-210, 2001.

[3] J. Chen and G. Stockman, "3D Free-Form Object Recognition Using Indexing by Contour Features," *Comp. Vision Image Understanding* **71**(3) 334-355, 1998.

[4] T. Cootes, C. Taylor, D. Cooper, and J. Graham, "Active Shape Models—Their Training and Application," *Comp. Vision Image Understanding* **61**(1) 38-59, 1995.

[5] R. Davies, C. Twining, T. Cootes, J. Waterton, and C. J. Taylor. "A Minimum Description Length Approach to Statistical Shape Modeling," *IEEE Trans. Med. Imag.* **21**(5) 525-537, 2002.

[6] R. Duda, P. Hart, and D. Stork, *Pattern Classification*. New York, NY: J. Wiley and Sons, 2001.

[7] N. Duta, A. Jain, and M. Dubuisson-Jolly, "Automatic Construction of 2D Shape Models," *IEEE Trans. PAMI* **23**(5), 433-446, 2001.

[8] Y. Gdalyahu and D. Weinshall, "Flexible Syntactic Matching of Curves and Its Application to Automatic Hierarchical Classification of Silhouettes," *IEEE Trans. PAMI* **21**(12), 1312-1328, 1999.

[9] S. Ghebreab and A. Smeulders, "Strings: Variational Deformable Models of Multivariate Continuous Boundary Features," *IEEE Trans. PAMI* **25**(11) 1399-1410, 2003.

[10] C. Grigorescu and N. Petkov, "Distance Sets for Shape Filters and Shape Recognition," *IEEE Trans. Image Processing* **12**(10) 1274-1286, 2003.

[11] L. Latecki and R. Lakämper, "Shape Similarity Measure Based on Correspondence of Visual Parts," *IEEE Trans. PAMI* **22**(10), 1185-1190, 2000.

[12] L. Latecki, R. Lakämper, and U. Eckhardt, "Shape Descriptors for Non-rigid Shapes with a Single Closed Contour," *Proc. CVPR*, 2000.

[13] T. Sebastian, P. Klein, and B. Kimia, "Shock-Based Indexing into Large Shape Databases," *Proc. ECCV*, pp. 731-746, 2002.

[14] T. Sebastian, P. Klein, and B. Kimia, "On Aligning Curves," *IEEE Trans. PAMI* **25**(1), 116-125, 2003.