Matching Shapes

Serge Belongie, Jitendra Malik and Jan Puzicha
Department of Electrical Engineering and Computer Sciences
University of California, Berkeley, CA 94720, USA
{sjb,malik,puzicha}@cs.berkeley.edu

Abstract

We present a novel approach to measuring similarity between shapes and exploit it for object recognition. In our framework, the measurement of similarity is preceded by (1) solving for correspondences between points on the two shapes, (2) using the correspondences to estimate an aligning transform. In order to solve the correspondence problem, we attach a descriptor, the shape context, *to each point. The shape context at a reference point captures the distribution of the remaining points relative to it, thus offering a globally discriminative characterization. Corresponding points on two similar shapes will have similar shape contexts, enabling us to solve for correspondences as an optimal assignment problem. Given the point correspondences, we estimate the transformation that best aligns the two shapes; regularized thin-plate splines provide a flexible class of transformation maps for this purpose. Dissimilarity between two shapes is computed as a sum of matching errors between corresponding points, together with a term measuring the magnitude of the aligning transform. We treat recognition in a nearest-neighbor classification framework. Results are presented for silhouettes, trademarks, handwritten digits and the COIL dataset.*

1 Introduction

Consider the two 5's in Figure 1. Regarded as vectors of pixel brightness values and compared using L_2 norms, they are very different. However, regarded as *shapes* they appear rather similar to a human observer. Our objective in this paper is to operationalize a notion of shape similarity, with the ultimate goal of using that as a basis for category-level recognition. We approach this as a three stage process: (1) solve the correspondence problem between the two shapes, (2) use the correspondences to estimate an aligning transform, and (3) compute the distance between the two shapes as a sum of matching errors between corresponding points, together with a term measuring the magnitude of the aligning transformation.

We wish to solve the problem in considerable generality. Shapes are arbitrary 2D figures, e.g. derived from edges extracted in images of 3D objects, not just silhouettes. The family of aligning transforms include affine as well as non-rigid smooth transformations, parametrized using thin plate splines. Matching errors between corresponding points are computed using both shape and local appearance differences.

At the heart of our approach is a tradition of matching shapes by deformation that can be traced at least as far back as D'Arcy Thompson. In his classic work *On Growth and Form* [27], Thompson observed that related but not identical shapes can often be deformed into alignment using simple coordinate transformations. Fischler and Elschlager [9] operationalized this approach using energy minimization in a mass-spring model. Grenander et al. [13] developed these ideas in a probabilistic setting. Yuille's [31] version of the deformable template concept fitted hand-crafted parametrized models, e.g. for eyes, in the image domain using gradient descent. Von der Malsburg and collaborators [19] used elastic graph matching for aligning faces.

Our primary contribution is a simple and robust algorithm for finding correspondences between shapes. Shapes are represented by a set of points sampled from the shape contours (typically 100 or so pixel locations sampled from the output of an edge detector are used). There is nothing special about the points. They are *not* required to be landmarks or curvature extrema, etc.; as we use more samples we obtain ever better approximations to the underlying shape. We introduce a shape descriptor, the *shape context*, to describe the coarse distri-

0-7695-1143-0/01 $10.00 © 2001 IEEE

Figure 1. Examples of two handwritten digits.

bution of the rest of the shape with respect to a point on the shape. Finding correspondences between two shapes is then equivalent to finding for each sample point on one shape the sample point on the other shape that has the most similar shape context. Maximizing similarities and enforcing uniqueness naturally leads to a setup as a bipartite graph matching (equivalently, optimal assignment) problem. As desired, we can incorporate other sources of matching information readily, e.g. similarity of local appearance at corresponding points.

Given the correspondences at sample points, we extend the correspondence to the complete shape by estimating an aligning transformation that maps one shape onto the other. The transformations can be picked from any of a number of families – we have used Euclidean, affine and regularized thin plate splines in various applications. Once the shapes are aligned, computing similarity scores and recognition by k-NN classification is relatively straightforward.

We demonstrate object recognition in a wide variety of settings. We deal with 2D objects, e.g. the MNIST dataset of handwritten digits (Fig. 5), silhouettes, and trademarks (Fig. 7), as well as 3D objects from the Columbia COIL dataset, modeled using multiple views (Fig. 6). These are widely used benchmarks and our approach turns out to be the leading performer on all the problems for which there is comparative data.

The structure of this paper is as follows. We discuss related work in Section 2. In Section 3 we then describe our shape matching method in detail. Our transformation model is discussed in Section 4. We then discuss the problem of measuring shape similarity in Section 5 and demonstrate our proposed measure on a variety of databases including handwritten digits and pictures of 3D objects. Finally, we conclude in Section 6.

2 Prior Work on Shape Matching

An extensive survey of shape matching in computer vision can be found in [28]. Broadly speaking, there are two approaches: (1) feature-based, and (2) brightness-based.

Feature-based approaches involve the use of spatial arrangements of extracted features such as edges or junctions. Silhouettes have been described (and compared) using Fourier descriptors, e.g. [32], skeletons derived using Blum's medial axis transform [26], or directly matched using dynamic programming e.g. [11]. Since silhouettes are limited as shape descriptors for general objects[1], other approaches[14, 10] treat the shape as a set of points in the 2D image, extracted using, say, an edge detector. Amit and Geman [1] find key points or landmarks, and recognize objects using the spatial arrangements of point sets. However not all objects have distinguished key points (think of a circle for instance), and using key points alone sacrifices the shape information available in smooth portions of object contours. Most closely related to our approach is the work of Rangarajan and collaborators [12, 7], which is discussed in Section 3.2.

Brightness-based approaches make more direct use of pixel brightness values. Several approaches[19, 29, 8] first attempt to find correspondences between the two images, before doing the comparison. This turns out to be quite a challenge as differential optical flow techniques do not cope well with the large distortions that must be handled due to pose/illumination variations. Errors in finding correspondence will cause downstream processing errors in the recognition stage. As an alternative, there are a number of methods that build classifiers without explicitly finding correspondences. In such approaches, one relies on a learning algorithm having enough examples to acquire the appropriate invariances. Some examples include [21, 6] for handwritten digit recognition, [22] for face recognition, and isolated 3D object recognition [24].

3 Matching with Shape Contexts

In our approach, a shape is represented by a discrete set of points sampled from the internal or external contours on the shape. These can be obtained as locations of edge pixels as found by an edge detector, giving us a set $\mathcal{P} = \{p_1, \ldots, p_n\}$, $p_i \in \mathbb{R}^2$, of n points. They need not, and typically will not, correspond to key-points such as maxima of curvature or inflection points. We prefer to sample the shape with roughly uniform spacing, though this is also not critical. Fig. 2(a,b) shows sample points for two shapes. Assuming contours are piecewise smooth, we can obtain as good an approximation to the underlying continuous shapes as desired by picking n to be sufficiently large.

For each point p_i on the first shape, we want to find the "best" matching point q_j on the second shape. This

[1] They ignore internal contours and are difficult to extract from real images.

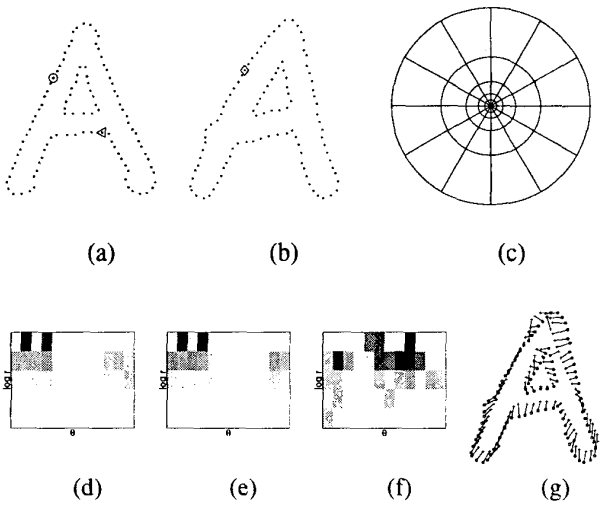

Figure 2. Shape context computation and matching. (a,b) Sampled edge points of two shapes. (c) Diagram of log-polar histogram bins used in computing the shape contexts. We use 5 bins for $\log r$ and 12 bins for θ. (d-f) Example shape contexts for reference samples marked by $\circ, \diamond, \triangleleft$ in (a,b). Each shape context is a log-polar histogram of the coordinates of the rest of the point set measured using the reference point as the origin. (Dark=large value.) Note the visual similarity of the shape contexts for $\circ$ and $\diamond$, which were computed for relatively similar points on the two shapes. By contrast, the shape context for $\triangleleft$ is quite different. (g) Correspondences found using bipartite matching, with costs defined by the χ^2 distance between histograms.

is a correspondence problem similar to that in stereopsis. Experience there suggests that matching is easier if one uses a rich local descriptor, e.g. a gray scale window or a vector of filter outputs, instead of just the brightness at a single pixel or edge location. Rich descriptors reduce the ambiguity in matching.

As a key contribution we propose a descriptor, the *shape context*, that could play such a role in shape matching. Consider the set of vectors originating from a point to all other sample points on a shape. These vectors express the configuration of the entire shape relative to the reference point. Obviously, this set of $n-1$ vectors is a rich description, since as n gets large, the representation of the shape becomes exact.

The full set of vectors as a shape descriptor is much too detailed since shapes and their sampled representation may vary from one instance to another in a category. We identify the *distribution* over relative positions as a more robust and compact, yet highly discriminative descriptor. For a point p_i on the shape, we compute a coarse histogram h_i of the relative coordinates of the remaining $n-1$ points,

$$h_i(k) \quad = \quad \#\{q \neq p_i \ : \ (q-p_i) \in \text{bin}(k)\} \quad .(1)$$

This histogram is defined to be the *shape context* of p_i. The descriptor should be more sensitive to differences in nearby pixels. We thus propose to use a log-polar coordinate system. An example is shown in Fig. 2(c).

Consider a point p_i on the first shape and a point q_j on the second shape. Let $C_{ij} = C(p_i, q_j)$ denote the cost of matching these two points. As shape contexts are distributions represented as histograms, it is natural[2] to use the χ^2 test statistic:

$$C_{ij} = \frac{1}{2}\sum_{k=1}^{K}\frac{[h_i(k)-h_j(k)]^2}{h_i(k)+h_j(k)}$$

where $h_i(k)$ and $h_j(k)$ denote the K-bin normalized histogram at p_i and q_j, respectively.

Given the set of costs C_{ij} between all pairs of points i on the first shape and j on the second shape we want to minimize the total cost of matching subject to the constraint that the matching be one-to-one. This is an instance of the square assignment (or weighted bipartite matching) problem, which can be solved in $O(N^3)$ time using the Hungarian method. In our experiments, we use the more efficient algorithm of [17]. The input to the assignment problem is a square cost matrix with entries C_{ij}. The result is a permutation $\pi(i)$ such that the sum $\sum_i C_{i,\pi(i)}$ is minimized.

When the number of samples on two shapes is not equal, the cost matrix can be made square by adding "dummy" nodes to each point set with a constant matching cost of ϵ_d. The same technique may also be used even when the sample numbers are equal to allow for robust handling of outliers. In this case, a point will be matched to a "dummy" whenever there is no real match available at smaller cost than ϵ_d. Thus, ϵ_d can be regarded as a threshold parameter for outlier detection.

The cost C_{ij} for matching points can include, an additional term based on the local *appearance similarity* at points p_i and q_j. This is particularly useful when we are comparing shapes derived from gray-level images instead of line drawings. For example, one can add a cost based on color or texture similarity, SSD between small gray-scale patches, distance between vectors of filter outputs, similarity of tangent angles, and so on.

[2] Alternatives include Bickel's generalization of the Kolmogorov-Smirnov test for 2D distributions [4], which does not require binning.

3.1 Invariance and Robustness

A matching approach should be (1) invariant under scaling and translation, and (2) robust under small affine transformations, occlusion and presence of outliers. In certain applications, one may want complete invariance under rotation, or perhaps even the full group of affine transformations. We now evaluate shape context matching by these criteria.

Invariance to translation is intrinsic to the shape context definition since all measurements are taken with respect to points on the object. To achieve scale invariance we normalize all radial distances by the mean distance α between the n^2 point pairs in the shape.

Since shape contexts are extremely rich descriptors, they are inherently insensitive to small perturbations of parts of the shape. While we have no theoretical guarantees here, robustness to small affine transformations, occlusions and presence of outliers is evaluated experimentally in Sect. 4.1.

In the shape context framework, we can provide for complete rotation invariance if this is desirable for an application. Instead of using the absolute frame for computing the shape context at each point, one can use the tangent vector at each point as the positive x-axis. In this way the reference frame turns with the tangent angle, and the result is a completely rotation invariant descriptor. In the extended version of this paper [3] we demonstrate this experimentally using the dataset from Kimia and collaborators[26].

3.2 Related work

The most comprehensive body of work on shape correspondence in this general setting is the work of Rangarajan and collaborators [12, 7]. They developed an iterative optimization algorithm to determine point correspondences and underlying image transformations jointly, where typically some generic transformation class is assumed, e.g. affine or thin plate splines. The cost function that is being minimized is the sum of Euclidean distances between a point on the *transformed* first shape and the second shape. This sets up a chicken-and-egg problem: the distances make sense only when there is at least a rough alignment of shape. Joint estimation of correspondences and shape transformation leads to a difficult, highly non-convex optimization problem, which is addressed using deterministic annealing [12]. The *shape context* is a very discriminative point descriptor, facilitating easy and robust correspondence recovery by incorporating global shape information into a local descriptor.

As far as we are aware of, the shape context descriptor and its use for matching 2D shapes is novel. The most closely related idea in past work is that due to Johnson and Hebert [16] in their work on range images. They introduced a representation for matching dense clouds of oriented 3D points called the "spin image". A spin image is a 2D histogram formed by spinning a plane around a normal vector on the surface of the object and counting the points that fall inside bins in the plane.

4 Modeling Transformations

Given a set of correspondences between two shapes, one can proceed to estimate a transformation that maps the model into the target. For this purpose there are several options; perhaps most common is the affine model. In this work, we use the thin plate spline (TPS) model, which is commonly used for representing flexible coordinate transformations [30, 25]. Bookstein [5], for example, found it to be highly effective for modeling changes in biological forms. The thin plate spline is the 2D generalization of the cubic spline. In its regularized form, which is discussed below, the TPS model includes the affine model as a special case. We will now provide some background information on the TPS model.

Let v_i denote the target function values at corresponding locations $p_i = (x_i, y_i)$ in the plane, with $i = 1, 2, \ldots, n$. In particular, we will set v_i equal to x_i' and y_i' in turn to obtain one continuous transformation for each coordinate. We assume that the locations (x_i, y_i) are all different and are not collinear. The TPS interpolant $f(x, y)$ minimizes the bending energy $I_f = \iint f_{xx}^2 + 2f_{xy}^2 + f_{yy}^2 dxdy$ and has the form:

$$f(x,y) = a_1 + a_x x + a_y y + \sum_{i=1}^{n} w_i U\left(\|(x_i, y_i) - (x, y)\|\right) \quad (2)$$

where $U(r) = r^2 \log r$. In order for $f(x, y)$ to have square integrable second derivatives, we require that

$$\sum_{i=1}^{n} w_i = 0 \quad \text{and} \quad \sum_{i=1}^{n} w_i x_i = \sum_{i=1}^{n} w_i y_i = 0 \quad (3)$$

Together with the interpolation conditions, $f(x_i, y_i) = v_i$, this yields a linear system for the TPS coefficients:

$$\left(\begin{array}{c|c} K & P \\ \hline P^T & 0 \end{array}\right)\left(\begin{array}{c} w \\ \hline a \end{array}\right) = \left(\begin{array}{c} v \\ \hline 0 \end{array}\right) \quad (4)$$

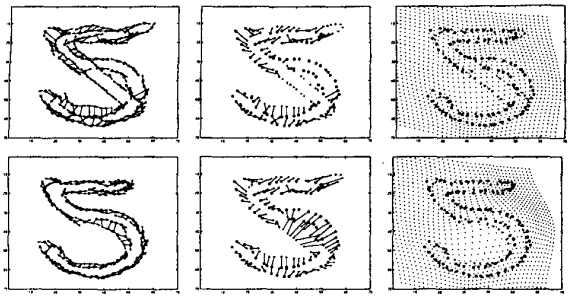

Figure 3. Illustration of the matching process applied to the example of Fig. 1. Top row: 1st iteration. Bottom row: 5th iteration. Left column: estimated correspondences shown relative to transformed model, with tangent vectors shown. Middle column: estimated correspondences shown relative to untransformed model. Right column: result of transforming the model based on the current correspondences; this is the input to the next iteration. The grid points illustrate the interpolated transformation over $\mathbb{R}^2$. Here we have used a regularized TPS model with $\lambda_o = 1$.

where $K_{ij} = U(||(x_i, y_i) - (x_j, y_j)||)$, the ith row of P is $(1, x_i, y_i)$, w and v are column vectors formed from w_i and v_i, respectively, and a is the column vector with elements a_1, a_x, a_y. We will denote the $(n+3) \times (n+3)$ matrix of this system by L. As discussed e.g. in [25], L is nonsingular and we can find the solution by inverting L. If we denote the upper left $n \times n$ block of L^{-1} by A, then it can be shown that $I_f \propto v^T A v = w^T K w$.

When there is noise in the specified values v_i, one may wish to relax the exact interpolation requirement by means of regularization. This is accomplished by minimizing $H[f] = \sum_{i=1}^{n} (v_i - f(x_i, y_i))^2 + \lambda I_f$. The *regularization parameter* λ, a positive scalar, controls the amount of smoothing; the limiting case of $\lambda = 0$ reduces to exact interpolation. As demonstrated in [30], we can solve for the TPS coefficients in the regularized case by replacing the matrix K by $K + \lambda I$, where I is the $n \times n$ identity matrix. It is interesting to note that the highly regularized TPS model degenerates to the least-squares affine model.

To address the dependence of λ on the data scale, suppose (x_i, y_i) and (x'_i, y'_i) are replaced by $(\alpha x_i, \alpha y_i)$ and $(\alpha x'_i, \alpha y'_i)$, respectively, for some positive constant α. Then it can be shown that the parameters w, a, I_f of the optimal thin plate spline are unaffected if λ is replaced by $\alpha^2 \lambda$. This simple scaling behavior suggests a normalized definition of the regularization parameter. Let α again represent the scale of the point set as estimated by the mean edge length between two points in the set. Then we can define λ in terms of α and λ_o, a scale-independent regularization parameter, via the simple relation $\lambda = \alpha^2 \lambda_o$.

The complete matching algorithm is obtained by alternating between the steps of recovering correspondences and estimating transformations (see Fig. 3). We usually employ a fixed number of iterations, typically three in large scale experiments, but more refined schemes are possible. On a regular Pentium III 500 MHz workstation this process takes roughly 200ms when the shapes have 100 sample points each.

4.1 Empirical Robustness Evaluation

In order to study the robustness of our proposed method, we performed the synthetic point set matching experiments described in [7]. The experiments are broken into three parts designed to measure robustness to deformation, noise, and outliers. (The latter tests each include a "moderate" amount of deformation.) In each test, we subjected the model point set to one of the above distortions to create a "target" point set. We then ran our algorithm to find the best warping between the model and the target. Finally, the performance is quantified by computing the average distance between the coordinates of the warped model and those of the target. The results are shown in Fig. 4. More details of the experiments may be found in [2].

5 Shape Similarity and Recognition

We define the shape distance $D(\mathcal{P}, \mathcal{Q})$ between shapes $\mathcal{P}$ and $\mathcal{Q}$ as a weighted sum of three terms: shape context distance, image appearance distance and bending energy. We will demonstrate the use of this distance for recognition in a nearest-neighbor classifier for a number of different object recognition problems.

We measure shape context distance between shapes $\mathcal{P}$ and $\mathcal{Q}$ as the symmetric sum of shape context matching costs over best matching points, i.e. $D_{\text{sc}}(\mathcal{P}, \mathcal{Q}) = \frac{1}{n} \sum_{p \in \mathcal{P}} \arg\min_{q \in \mathcal{Q}} C(p, T(q)) + \frac{1}{m} \sum_{q \in \mathcal{Q}} \arg\min_{p \in \mathcal{P}} C(p, T(q))$ where $T(\cdot)$ denotes the estimated TPS shape transformation.

Often there is additional appearance information available that is not captured by our notion of shape, e.g. local image patches, textural information, color, etc. As a key benefit of the shape matching framework, the distorted image can be warped back into a normal form after recovery of the underlying 2D image transformation, thus correcting for distortions of the image appearance. We used a term $D_{\text{ac}}(\mathcal{P}, \mathcal{Q})$ for appearance cost

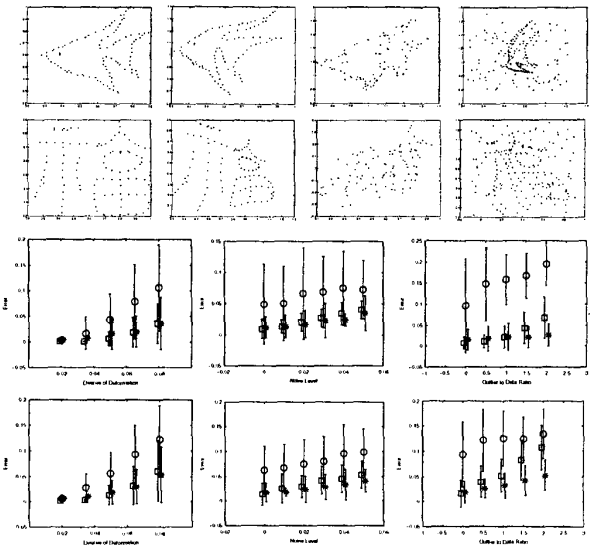

Figure 4. Empirical robustness evaluation, following [7]. Two model pointsets are shown in the first column of rows 1 and 2. Columns 2-4 show examples of point sets for the deformation, noise, and outlier tests. Row 3 shows error as a function of the deformation, noise, or outlier to data ratio for our method (□), [7]'s method (∗) and iterated closest point (○) for the fish shape in row 1. Row 4 shows the results for the Chinese character in row 2. The error bars indicate the std. dev. of the error over 100 random trials.

which is the sum of squared differences in Gaussian windows around corresponding points.

The third term corresponds to the 'amount' of transformation necessary to align the shapes. In the TPS case the bending energy $D_{\text{be}}(\mathcal{P}, \mathcal{Q}) = w^T K w$ is a natural measure.

5.1 Digit Recognition

We begin with results on the well-known MNIST dataset of handwritten digits, which consists of 60,000 training and 10,000 test digits[21]. Matching used 100 point samples selected from the Canny edges of each digit image. We employed a TPS transformation model and used 3 iterations of shape context matching and TPS re-estimation. We used a nearest neighbor classifier with $D(\mathcal{P}, \mathcal{Q})$ as defined above.

Nearest neighbor classifiers have the property that as the number of examples n in the training set $\rightarrow \infty$, the 1-NN error converges to a value $\leq 2E^*$, where E^* is the Bayes Risk (for k-NN, by making $k \rightarrow \infty$ and $k/n \rightarrow 0$, the error $\rightarrow E^*$). However, what matters in practice is the performance for small n, and this gives us

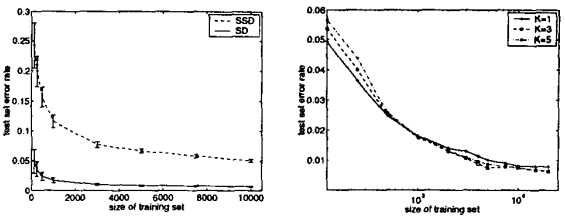

Figure 5. Handwritten digit recognition on the MNIST dataset. Left: Test set errors of a 1-NN classifier using SSD and Shape Distance (SD) measures. Right: Detail of performance curve for Shape Distance, including results with training set sizes of 15,000 and 20,000. Results are shown on a semilog-x scale for $K = 1, 3, 5$ nearest neighbors.

a way to compare different similarity/distance measures. In Fig. 5, our shape distance is compared to SSD (sum of squared differences between pixel brightness values of images regarded as vectors).

On the MNIST dataset nearly 30 algorithms have been compared (http://www.research.att.com/~yann/exdb/mnist/index.html). The lowest test set error rate published at this time is 0.7% for a boosted LeNet-4 with a training set of size $60,000 \times 10$ synthetic distortions per training digit. Our error rate using 20,000 training examples and 3-NN is 0.63%.

5.2 MPEG-7 Shape Silhouette Database

Our next experiment involves the MPEG-7 shape silhouette database, specifically Core Experiment CE-Shape-1 part B, which measures performance of similarity-based retrieval [15]. The database consists of 1400 images: 70 shape categories, 20 images per category. The performance is measured using the so-called "bullseye test," in which each image is used as a query and one counts the number of correct images in the top 40 matches.

As this experiment involves intricate shapes we increased the number of samples from 100 to 300. In some categories the shapes appear rotated and flipped, which we address using a modified distance function. The distance $\text{dist}(R, Q)$ between a reference shape R and a query shape Q is defined as

$$\text{dist}(Q, R) = \min\{\text{dist}(Q, R^a), \text{dist}(Q, R^b), \text{dist}(Q, R^c)\}$$

where R^a, R^b and R^c denote three versions of R: unchanged, vertically flipped, and horizontally flipped.

With these changes in place but otherwise using the same approach as in the MNIST digit experiments, we

obtain a retrieval rate of 76.51%. Currently the best published performance is achieved by Latecki et al. [20], with a retrieval rate of 76.45%, followed by Mokhtarian et al. [23] at 75.44%.

5.3 Columbia COIL-20 Database

Our next experiment involves the 20 common household objects from the COIL-20 database [24]. Each object was placed on a turntable and photographed every 5° for a total of 72 views per object. We prepared our training sets by selecting a number of equally spaced views for each object and using the remaining views for testing. The matching algorithm and shape distance are **exactly** the same as for digits.

Fig. 6(a) shows the performance of a 1-NN classifier using our shape distance as well as SSD (sum of squared differences). SSD performs very well on this easy database due to the lack of variation in lighting [14] (PCA just makes it faster).

In a companion paper [2] we recently developed a novel *editing* algorithm based on shape context similarity and k-medoid clustering. The editing algorithm is illustrated in Fig. 6(b). More views are chosen for visually complex categories. This idea is related to the "aspect" concept as discussed in [18]. The curve marked SC-proto in Fig. 6(a) shows the improved classification performance using this prototype selection strategy instead of equally-spaced views. Note that we obtain a 2.4% error rate with an average of only 4 two-dimensional views for each three-dimensional object, thanks to the flexibility provided by the matching algorithm.

5.4 Trademark Retrieval

The automatic identification of trademark infringement is of commercial interest. Currently, trademarks are broadly classified according to the Vienna code, and infringements are detected by manually looking for close perceptual similarity in an appropriate category. Shape, together with text and texture, is key in defining perceptual similarity. Using our notion of shape distance, Fig. 7 depicts nearest neighbor retrieval results from a database of 300 trademarks. We experimented with eight different query trademarks for each of which the database contained at least one potential infringement. It is clearly seen that the potential infringements are easily detected and appear as most similar on the top ranks despite substantial variation of the actual shapes. It has been manually verified that no visually similar trademark has been missed by the algorithm.

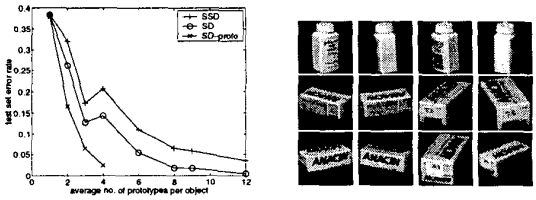

Figure 6. Left: 3D object recognition using the COIL-20 dataset. Comparison of test set error for SSD, Shape Distance (SD), and Shape Distance with k-medoid prototypes (SD-proto) vs. number of prototype views. For SSD and SD, we varied the number of prototypes uniformly for all objects. For SD-proto, the number of prototypes per object depended on the within-object variation as well as the between-object similarity. Right: K-medoid prototype views for two different 3D objects, using an average of 4 views per object. With this approach, resources are allocated adaptively depending on the visual complexity of an object. In this example we observe that the Anacin box requires twice as many views as the baby powder bottle.

6 Conclusion

We have presented a new approach to the analysis of shape. A key characteristic of our approach is the estimation of shape similarity and correspondences based on a novel descriptor, the shape context. In our experiments we have demonstrated excellent performance on a wide variety of datasets, both of 2D and 3D objects.

Acknowledgments This research is supported by (ARO) DAAH04-96-1-0341, the Digital Library Grant IRI-9411334, an NSF graduate Fellowship for S.B and the German Research Foundation by DFG grant PU-165/1. We wish to thank H. Chui and A. Rangarajan for providing the synthetic testing data used in 4.1.

References

[1] Y. Amit, D. Geman, and K. Wilder. Joint induction of shape features and tree classifiers. *IEEE Trans. PAMI*, 19(11):1300–1305, November 1997.

[2] S. Belongie, J. Malik, and J. Puzicha. Shape context: A new descriptor for shape matching and object recognition. In *NIPS*, November 2000.

[3] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. Technical Report UCB//CSD-00-1128, UC Berkeley, January 2001.

[4] P. J. Bickel. A distribution free version of the Smirnov two-sample test in the multivariate case. *Annals of Mathematical Statistics*, 40:1–23, 1969.

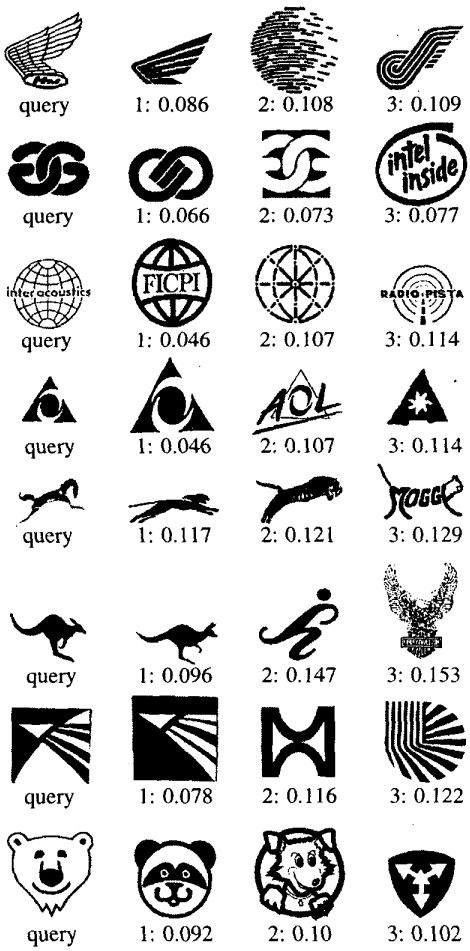

Figure 7. Trademark retrieval results based on a database of 300 different real-world trademarks. We used an affine transformation model and a weighted combination of shape context similarity D_{sc} and the sum over local tangent orientation differences.

[5] F. L. Bookstein. *Morphometric tools for landmark data: geometry and biology*. Cambridge Univ. Press, 1991.

[6] C. Burges and B. Schölkopf. Improving the accuracy and speed of support vector machines. In *NIPS*, pages 375–381, 1997.

[7] H. Chui and A. Rangarajan. A new algorithm for non-rigid point matching. In *CVPR*, volume 2, pages 44–51, June 2000.

[8] T. Cootes, D. Cooper, C. Taylor, and J. Graham. Active shape models - their training and application. *Computer Vision and Image Understanding (CVIU)*, 61(1):38–59, Jan. 1995.

[9] M. Fischler and R. Elschlager. The representation and matching of pictorial structures. *IEEE Trans. Computers*, C-22(1):67–92, 1973.

[10] D. Gavrila and V. Philomin. Real-time object detection for smart vehicles. In *Proc. 7th Int. Conf. Computer Vision*, pages 87–93, 1999.

[11] 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.

[12] S. Gold, A. Rangarajan, C.-P. Lu, S. Pappu, and E. Mjolsness. New algorithms for 2D and 3D point matching: pose estimation and correspondence. *Pattern Recognition*, 31(8), 1998.

[13] U. Grenander, Y. Chow, and D. Keenan. *HANDS: A Pattern Theoretic Study Of Biological Shapes*. Springer, 1991.

[14] D. Huttenlocher, R. Lilien, and C. Olson. View-based recognition using an eigenspace approximation to the Hausdorff measure. *PAMI*, 21(9):951–955, Sept. 1999.

[15] S. Jeannin and M. Bober. Description of core experiments for MPEG-7 motion/shape. Technical Report ISO/IEC JTC 1/SC 29/WG 11 MPEG99/N2690, MPEG-7, Seoul, March 1999.

[16] A. E. Johnson and M. Hebert. Recognizing objects by matching oriented points. In *CVPR*, pages 684–689, 1997.

[17] R. Jonker and A. Volgenant. A shortest augmenting path algorithm for dense and sparse linear assignment problems. *Computing*, 38:325–340, 1987.

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

[19] M. Lades, C. Vorbrüggen, J. Buhmann, J. Lange, C. von der Malsburg, R. Wurtz, and W. Konen. Distortion invariant object recognition in the dynamic link architecture. *IEEE Trans. Computers*, 42(3):300–311, March 1993.

[20] L. J. Latecki, R. Lakämper, and U. Eckhardt. Shape descriptors for non-rigid shapes with a single closed contour. In *CVPR*, pages 424–429, 2000.

[21] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, November 1998.

[22] B. Moghaddam, T. Jebara, and A. Pentland. Bayesian face recognition. *Pattern Recognition*, 33(11):1771–1782, November 2000.

[23] F. Mokhtarian, S. Abbasi, and J. Kittler. Efficient and robust retrieval by shape content through curvature scale space. In A. W. M. Smeulders and R. Jain, editors, *Image Databases and Multi-Media Search*, pages 51–58. World Scientific, 1997.

[24] H. Murase and S. Nayar. Visual learning and recognition of 3-D objects from appearance. *Int. Journal of Computer Vision*, 14(1):5–24, Jan. 1995.

[25] M. J. D. Powell. A thin plate spline method for mapping curves into curves in two dimensions. In *Computational Techniques and Applications (CTAC95)*, Melbourne, Australia, 1995.

[26] D. Sharvit, J. Chan, H. Tek, and B. Kimia. Symmetry-based indexing of image databases. *J. Visual Communication and Image Representation*, 1998.

[27] D. W. Thompson. *On Growth and Form*. Dover, 1917.

[28] R. C. Veltkamp and M. Hagedoorn. State of the art in shape matching. Technical Report UU-CS-1999-27, Utrecht, 1999.

[29] T. Vetter, M. J. Jones, and T. Poggio. A bootstrapping algorithm for learning linear models of object classes. In *CVPR*, pages 40–46, 1997.

[30] G. Wahba. *Spline Models for Observational Data*. SIAM, 1990.

[31] A. Yuille. Deformable templates for face recognition. *J. Cognitive Neuroscience*, 3(1):59–71, 1991.

[32] C. Zahn and R. Roskies. Fourier descriptors for plane closed curves. *IEEE Trans. Computers*, 21(3):269–281, March 1972.