Shape Matching and Object Recognition Using Shape Contexts

Serge Belongie, Jitendra Malik and Jan Puzicha

Keywords: shape, object recognition, digit recognition, correspondence problem, MPEG7, image registration, deformable templates

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. The dissimilarity between the 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 as the problem of finding the stored prototype shape that is maximally similar to that in the image. Results are presented for silhouettes, trademarks, handwritten digits and the COIL dataset.

Author Affiliation Data

This work was carried out while the authors were with the Electrical Engineering and Computer Science Division, University of California at Berkeley, Berkeley, CA 94720.

Serge Belongie is with the Department of Computer Science and Engineering, University of California at San Diego, La Jolla, CA, 92093. E-mail: sjb@cs.ucsd.edu.

Jitendra Malik is with the Electrical Engineering and Computer Science Division, University of California at Berkeley, Berkeley, CA 94720. E-mail: malik@cs.berkeley.edu.

Jan Puzicha is with RecomMind, Inc., 1001 Camelia Street, Berkeley, CA 94710.

1 Introduction

Consider the two handwritten digits 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 this notion of shape similarity, with the ultimate goal of using it 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.

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* [55], Thompson observed that related but not identical shapes can often be deformed into alignment using simple coordinate transformations, as illustrated in Fig. 2. In the computer vision literature, Fischler and Elschlager [15] operationalized such an idea by means of energy minimization in a mass-spring model. Grenander et al. [21] developed these ideas in a probabilistic setting. Yuille [61] developed another variant of the deformable template concept by means of fitting hand-crafted parametrized models, e.g. for eyes, in the image domain using gradient descent. Another well-known computational approach in this vein was developed by von der Malsburg and collaborators [31] using elastic graph matching.

Our primary contribution in this work is a robust and simple 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 better approximations to the underlying shape. We introduce a shape descriptor, the *shape context*, to describe the coarse distribution of the rest of the shape with respect to a given 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. A classic illustration of this idea is provided in Fig. 2. 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. Aligning shapes enables us to define a simple, yet general, measure of shape similarity. The dissimilarity between the two shapes can now be computed as a sum of matching errors between corresponding points, together with a term measuring the magnitude of the aligning transform.

Given such a dissimilarity measure, we can use nearest neighbor techniques for object recognition. Philosophically, nearest neighbor techniques can be related to prototype-based recognition as developed by Rosch and collaborators [47, 48]. They have the advantage that

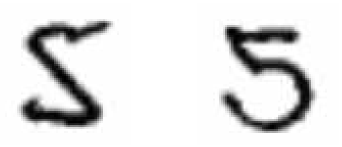

Figure 1. Examples of two handwritten digits. In terms of pixel-to-pixel comparisons, these two images are quite different, but to the human observer, the shapes appear to be similar.

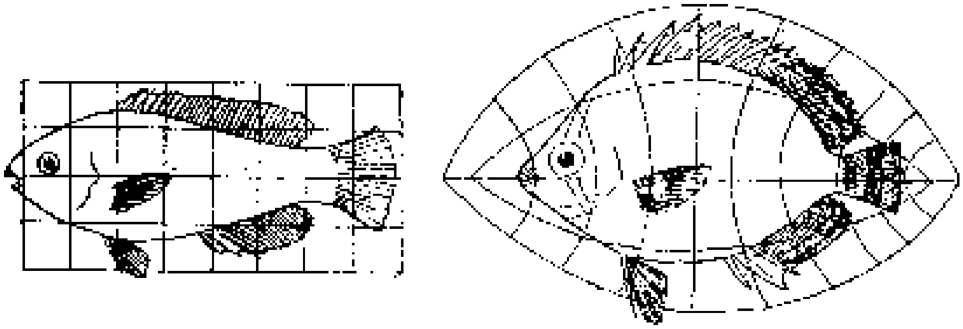

Figure 2. Example of coordinate transformations relating two fish, from D'Arcy Thompson's *On Growth and Form* [55]. Thompson observed that similar biological forms could be related by means of simple mathematical transformations between *homologous* (i.e. corresponding) features. Examples of homologous features include center of eye, tip of dorsal fin, etc.

a vector space structure is not required–only a pairwise dissimilarity measure.

We demonstrate object recognition in a wide variety of settings. We deal with 2D objects, e.g. the MNIST dataset of handwritten digits (Fig. 8), silhouettes (Figs. 11 and 13) and trademarks (Fig. 12), as well as 3D objects from the Columbia COIL dataset, modeled using multiple views (Fig. 10). 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.

We have also developed a technique for selecting the number of stored views for each object category based on its visual complexity. As an illustration, we show that for the 3D objects in the COIL-20 dataset, one can obtain as low as 2.5% misclassification error using only an average of 4 views per object (see Figs. 9 and 10).

The structure of this paper is as follows. We discuss related work in Section 2. In Section 3 we describe our shape matching method in detail. Our transformation model is presented 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 in Section 6. We conclude in Section 7.

2 Prior Work on Shape Matching

Mathematicians typically define shape as an equivalence class under a group of transformations. This definition is incomplete in the context of visual analysis. This only tells us when two shapes are exactly the same; we need more than that for a theory of shape similarity or shape distance. The statistician's definition of shape, e.g. Bookstein [6] or Kendall [29], addresses the problem of shape distance, but assumes that correspondences are known. Other statistical approaches to shape comparison do not require correspondences – e.g. one could compare feature vectors containing descriptors such as area or moments – but such techniques often discard detailed shape information in the process. Shape similarity has also been studied in the psychology literature, an early example being Goldmeier [20].

An extensive survey of shape matching in computer vision can be found in [58, 22]. Broadly speaking, there are two approaches: (1) feature-based, which involve the use of spatial arrangements of extracted features such as edge elements or junctions, and (2) brightness-based, which make more direct use of pixel brightnesses.

2.1 Feature-Based Methods

A great deal of research on shape similarity has been done using the boundaries of *silhouette* images. Since silhouettes do not have holes or internal markings, the associated boundaries are conveniently represented by a single closed curve which can be parametrized by arclength. Early work used Fourier descriptors, e.g. [62, 43]. Blum's medial axis transform has led to attempts to capture the part structure of the shape in the graph structure of the skeleton by Kimia, Zucker and collaborators, e.g. Sharvit et al. [53]. The 1D nature of silhouette curves leads naturally to dynamic programming approaches for matching, e.g. [17], which uses the edit distance between curves. This algorithm is fast and invariant to several kinds of transformation including some articulation and occlusion. A comprehensive comparison of different shape descriptors for comparing silhouettes was done as part of the MPEG-7 standard activity [33], with the leading approaches being those due to Latecki et al. [33] and Mokhtarian et al. [39].

Silhouettes are fundamentally limited as shape descriptors for general objects; they ignore internal contours and are difficult to extract from real images. More promising are approaches that treat the shape as a set of points in the 2D image. Extracting these from an image is less of a problem – e.g. one can just use an edge detector. Huttenlocher et al. developed methods in this category based on the Hausdorff distance [23]; this can be extended to deal with partial matching and clutter. A drawback for our purposes is that the method does not return correspondences. Methods based on Distance Transforms, such as [16], are similar in spirit and behavior in practice.

The work of Sclaroff and Pentland [50] is representative of the eigenvector- or modal-matching based approaches; see also [52, 51, 57]. In this approach, sample points in the image are cast into a finite element spring-mass model and correspondences are found by comparing modes of vibration. Most closely related to our approach is the work of Rangarajan and collaborators [19, 9], which is discussed in Section 3.4.

There have been several approaches to shape recognition based on spatial configurations of a small number of keypoints or landmarks. In geometric hashing [32], these configurations are used to vote for a model without explicitly solving for correspondences. Amit et al. [1] train decision trees for recognition by learning discriminative spatial configurations of keypoints. Leung et al. [35], Schmid and Mohr [49], and Lowe [36] additionally use gray level information at the keypoints to provide greater discriminative power. It should be noted that 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.

2.2 Brightness-Based Methods

Brightness-based (or appearance-based) methods offer a complementary view to feature-based methods. Instead of focusing on the shape of the occluding contour or other extracted features, these approaches make direct use of the gray values within the visible portion of the object. One can use brightness information in one of two frameworks.

In the first category we have the methods that explicitly find correspondences/alignment using grayscale values. Yuille [61] presents a very flexible approach in that invariance to certain kinds of transformations can be built into the measure of model similarity, but it suffers from the need for human-designed templates and the sensitivity to initialization when searching via gradient descent. Von der Malsburg and colleagues [31] use elastic graph matching, an approach that involves both geometry and photometric features in the form of local descriptors based on Gaussian derivative jets. Vetter et al. [59] and Cootes et al. [10] compare brightness values but first attempt to warp the images onto one another using a dense correspondence field.

The second category includes those 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. In the area of face recognition good results were obtained using principal components analysis (PCA) [54, 56] particularly when used in a probabilistic framework [38]. Murase and Nayar applied these ideas to 3D object recognition [40]. Several authors have applied discriminative classification methods in the appearance-based shape matching framework. Some examples are the LeNet classifier [34], a convolutional neural network for handwritten digit recognition, and the Support Vector Machine (SVM)-based methods of [41] (for discriminating between templates of pedestrians based on 2D wavelet coefficients) and [11, 7] (for handwritten digit recognition). The MNIST database of handwritten digits is a particularly important dataset as many different pattern recognition algorithms have been tested on it. We will show our results on MNIST in Section 6.1.

3 Matching with Shape Contexts

In our approach, we treat an object as a (possibly infinite) point set and we assume that the shape of an object is essentially captured by a finite subset of its points. More practically, a shape is represented by a discrete set of points sampled from the internal or external contours on the object. 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.[1] Fig. 3(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.

3.1 Shape Context

For each point p_i on the first shape, we want to find the "best" matching point q_j on the second shape. This 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 [27], 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 novel 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

[1] Sampling considerations are discussed in Appendix B.

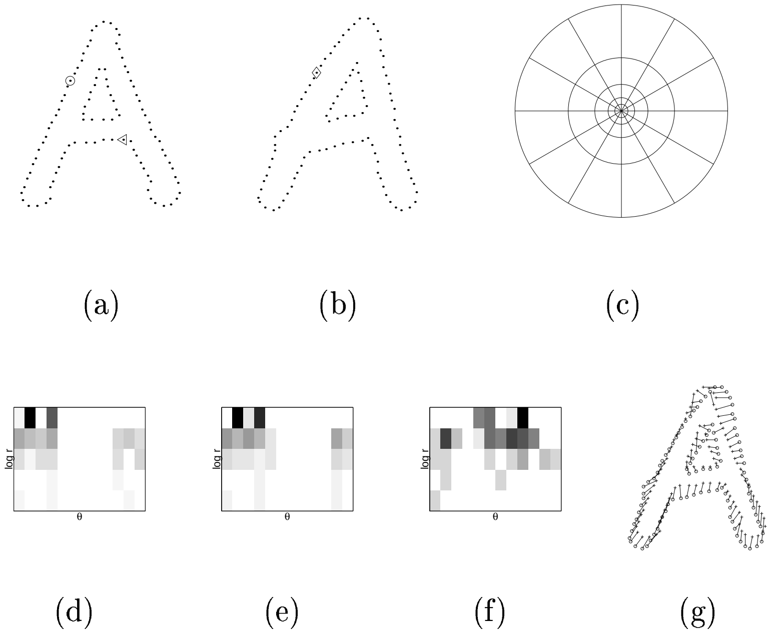

Figure 3. Shape context computation and matching. (a,b) Sampled edge points of two shapes. (c) Diagram of log-polar histogram bins used in computing the shape contexts. We use 5 bins for $\log r$ and 12 bins for θ. (d-f) Example shape contexts for reference samples marked by $\circ, \diamond, \triangleleft$ 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.

coordinates of the remaining $n-1$ points,

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

This histogram is defined to be the *shape context* of p_i. We use bins that are uniform in log-polar[2] space, making the descriptor more sensitive to positions of nearby sample points than to those of points farther away. An example is shown in Fig. 3(c).

Consider a point p_i on the first shape and a point q_j on the second shape. Let $C_{ij} =$

[2] This choice corresponds to a linearly increasing positional uncertainty with distance from p_i, a reasonable result if the transformation between the shapes around p_i can be locally approximated as affine.

$C(p_i, q_j)$ denote the cost of matching these two points. As shape contexts are distributions represented as histograms, it is natural to use the χ^2 test statistic:

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

where $h_i(k)$ and $h_j(k)$ denote the K-bin normalized histogram at p_i and q_j, respectively.[3]

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 normalized correlation scores between small gray-scale patches centered at p_i and q_j, distances between vectors of filter outputs at p_i and q_j, tangent orientation difference between p_i and q_j, and so on. The choice of this appearance similarity term is application dependent, and is driven by the necessary invariance and robustness requirements, e.g. varying lighting conditions make reliance on gray-scale brightness values risky.

3.2 Bipartite Graph Matching

Given the set of costs C_{ij} between all pairs of points p_i on the first shape and q_j on the second shape we want to minimize the total cost of matching,

$$H(\pi) = \sum_i C\left(p_i, q_{\pi(i)}\right) \tag{2}$$

subject to the constraint that the matching be one-to-one, i.e. π is a permutation. 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 [42]. In our experiments, we use the

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

more efficient algorithm of [28]. The input to the assignment problem is a square cost matrix with entries C_{ij}. The result is a permutation $\pi(i)$ such that (2) is minimized.

In order to have robust handling of outliers, one can add "dummy" nodes to each point set with a constant matching cost of ϵ_d. 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. Similarly, when the number of sample points on two shapes is not equal, the cost matrix can be made square by adding dummy nodes to the smaller point set.

3.3 Invariance and Robustness

A matching approach should be (1) invariant under scaling and translation, and (2) robust under small geometrical distortions, 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 nonlinear transformations, occlusions and presence of outliers is evaluated experimentally in Sect. 4.2.

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 a relative frame, based on treating 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 Appendix A we demonstrate this experimentally. It should be emphasized though that in many applications complete invariance impedes recognition performance, e.g. when distinguishing 6 from 9 rotation invariance would be completely inappropriate. Another drawback is that many points will not have well defined or reliable tangents. Moreover, many local appearance features lose their discriminative power if they are not measured in the same coordinate system.

Additional robustness to outliers can be obtained by excluding the estimated outliers from the shape context computation. More specifically, consider a set of points that have been labeled as outliers on a given iteration. We render these points “invisible” by not allowing them to contribute to any histogram. However, we still assign them shape contexts, taking into account only the surrounding inlier points, so that at a later iteration they have a chance of re-emerging as an inlier.

3.4 Related work

The most comprehensive body of work on shape correspondence in this general setting is the work of Rangarajan and collaborators [19, 9]. 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 first shape and the *transformed* 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 solved using deterministic annealing [19]. 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, 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 [26] 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. As the size of this plane is relatively small, the resulting signature is not as informative as a shape context for purposes of recovering correspondences. This characteristic, however, might have the tradeoff of additional robustness to occlusion. In another related work, Carlsson [8] has exploited the concept of *order structure* for characterizing local shape configurations. In this work, the relationships between points and tangent lines in a shape are used for recovering correspondences.

4 Modeling Transformations

Given a finite set of correspondences between points on two shapes, one can proceed to estimate a plane transformation $T : \mathbb{R}^2 \longrightarrow \mathbb{R}^2$ that may be used to map arbitrary points from one shape to the other. This idea is illustrated by the warped gridlines in Fig. 2, wherein the specified correspondences consisted of a small number of landmark points such as the centers of the eyes, the tips of the dorsal fins, etc., and T extends the correspondences to arbitrary points.

We need to choose T from a suitable family of transformations. A standard choice is the affine model, i.e.

$$T(x) = Ax + o \tag{3}$$

with some matrix A and a translational offset vector o parameterizing the set of all allowed transformations. Then the least squares solution $\hat{T} = (\hat{A}, \hat{o})$ is obtained by

$$\begin{aligned} \hat{o} &= \frac{1}{n} \sum_{i=1}^{n} \left(p_i - q_{\pi(i)} \right) \quad , & (4) \\ \hat{A} &= (Q^+ P)^t & (5) \end{aligned}$$

where P and Q contain the homogeneous coordinates of $\mathcal{P}$ and $\mathcal{Q}$, respectively, i.e.

$$P = \begin{pmatrix} 1 & p_{11} & p_{12} \\ \vdots & \vdots & \vdots \\ 1 & p_{n1} & p_{n2} \end{pmatrix} \quad . \tag{6}$$

Here, Q^+ denotes the pseudo–inverse of Q.

In this work, we mostly use the thin plate spline (TPS) model [14, 37], which is commonly used for representing flexible coordinate transformations. Bookstein [6] found it to be highly effective for modeling changes in biological forms. Powell applied the TPS model to recover transformations between curves [44]. 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.

We start with the 1D interpolation problem. 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

$$I_f = \iint_{\mathbb{R}^2} \left(\frac{\partial^2 f}{\partial x^2}\right)^2 + 2\left(\frac{\partial^2 f}{\partial x \partial y}\right)^2 + \left(\frac{\partial^2 f}{\partial y^2}\right)^2 dxdy$$

and has the form:

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

where the kernel function $U(r)$ is defined by $U(r) = r^2 \log r^2$ and $U(0) = 0$ as usual. In order for $f(x,y)$ to have square integrable second derivatives, we require that

$$\sum_{i=1}^{n} w_i \;=\; 0 \quad \text{and} \quad \sum_{i=1}^{n} w_i x_i \;=\; \sum_{i=1}^{n} w_i y_i \;=\; 0 \ . \tag{7}$$

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

$$\left(\begin{array}{c|c} K & P \\ \hline P^T & 0 \end{array}\right) \left(\begin{array}{c} w \\ \hline a \end{array}\right) = \left(\begin{array}{c} v \\ \hline 0 \end{array}\right) \tag{8}$$

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 [44], 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

$$I_f \;\propto\; v^T A v \;=\; w^T K w \ . \tag{9}$$

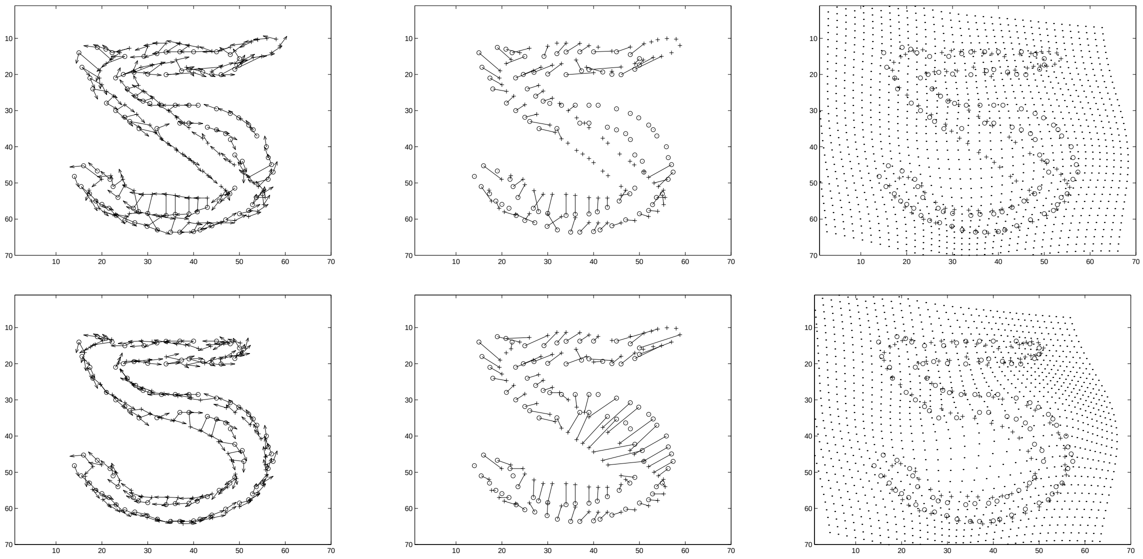

Figure 4. Illustration of the matching process applied to the example of Fig. 1. Top row: 1st iteration. Bottom row: 5th iteration. Left column: estimated correspondences shown relative to the transformed model, with tangent vectors shown. Middle column: estimated correspondences shown relative to the 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 $\lambda_o = 1$.

4.1 Regularization and Scaling Behavior

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

$$H[f] \;=\; \sum_{i=1}^{n}(v_i - f(x_i, y_i))^2 + \lambda I_f \ . \tag{10}$$

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 [60, 18], 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 median 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 $\lambda = \alpha^2 \lambda_o$.

We use two separate TPS functions to model a coordinate transformation,

$$T(x, y) \quad = \quad (f_x(x, y), f_y(x, y)) \tag{11}$$

which yields a displacement field that maps any position in the first image to its interpolated location in the second image.

In many cases, the initial estimate of the correspondences contains some errors which could degrade the quality of the transformation estimate. The steps of recovering correspondences and estimating transformations can be iterated to overcome this problem. We usually use a fixed number of iterations, typically three in large scale experiments, but more refined schemes are possible. However, experimental experiences show that the algorithmic performance is independent of the details. An example of the iterative algorithm is illustrated in Fig. 4.

4.2 Empirical Robustness Evaluation

In order to study the robustness of our proposed method, we performed the synthetic point set matching experiments described in [9]. 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; see Fig. 5. 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. 6. In the most challenging part of the test – the outlier experiment – our approach shows robustness even up to a level of 100% outlier-to-data ratio.

In practice we will need robustness to occlusion and segmentation errors which can be explored only in the context of a complete recognition system, though these experiments provide at least some guidelines.

4.3 Computational demands

In our implementation on a regular Pentium III /500 MHz workstation a single comparison including computation of shape context for 100 sample points, set-up of the full matching matrix, bipartite graph matching, computation of the TPS coefficients, and image warping for three cycles takes roughly 200ms. The run-time is dominated by the number of sample points for each shape, with most components of the algorithm exhibiting between quadratic and cubic scaling behaviour. Using a sparse representation throughout, once the shapes are roughly aligned, the complexity could be made close to linear.

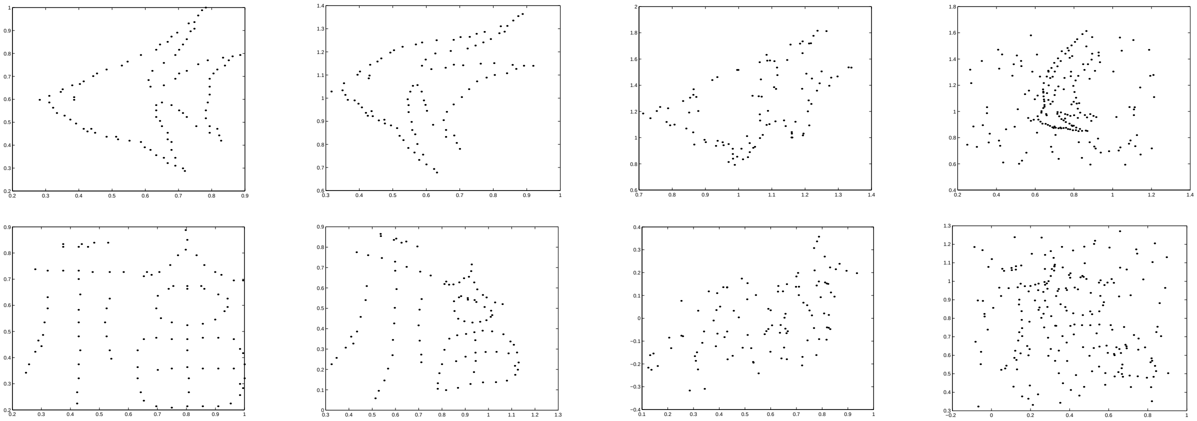

Figure 5. Testing data for empirical robustness evaluation, following Chui and Rangarajan [9]. The model pointsets are shown in the first column. Columns 2-4 show examples of target point sets for the deformation, noise, and outlier tests, respectively.

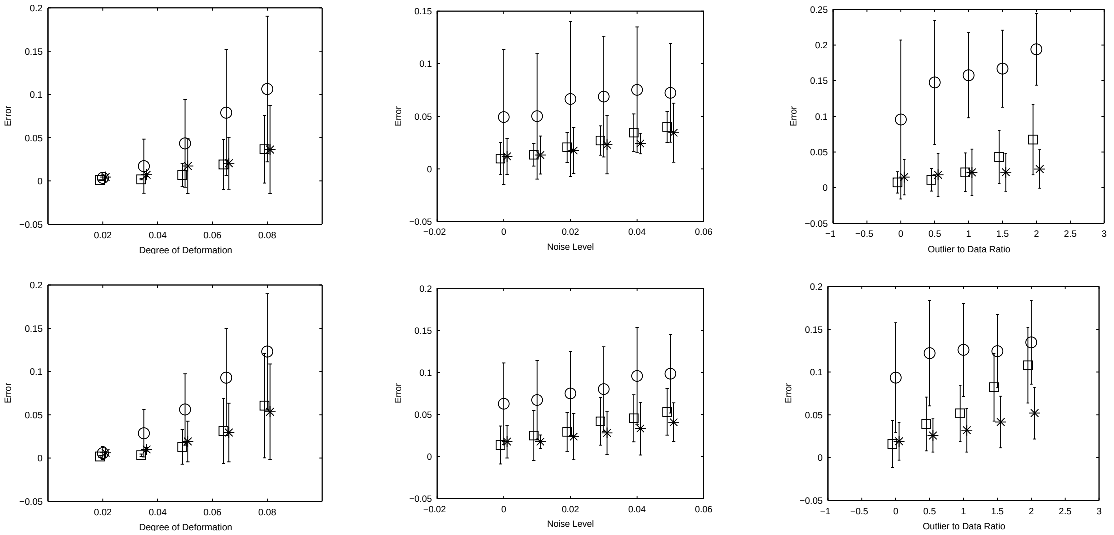

Figure 6. Comparison of our results (□) to Chui and Rangarajan ($*$) and iterated closest point ($\circ$) for the fish and Chinese character, respectively. The error bars indicate the standard deviation of the error over 100 random trials. Here we have used 5 iterations with $\lambda_o = 1.0$. In the deformation and noise tests no dummy nodes were added. In the outlier test, dummy nodes were added to the model point set such that the total number of nodes was equal to that of the target. In this case the value of ϵ_d does not affect the solution.

5 Object Recognition and Prototype Selection

Given a measure of dissimilarity between shapes, which we will make precise shortly, we can proceed to apply it to the task of object recognition. Our approach falls into the category of prototype-based recognition. In this framework, pioneered by Rosch and collaborators [48], categories are represented by ideal examples rather than a set of formal logical rules. As an example, a sparrow is a likely prototype for the category of birds; a less likely choice might be an penguin. The idea of prototypes allows for soft category membership, meaning that as one moves farther away from the ideal example in some suitably defined similarity space, one's association with that prototype falls off. When one is sufficiently far away from that prototype, the distance becomes meaningless, but by then one is most likely near a different prototype. As an example, one can talk about good or so-so examples of the color red, but when the color becomes sufficiently different, the level of dissimilarity saturates at some maximum level rather than continuing on indefinitely.

Prototype-based recognition translates readily into the computational framework of nearest neighbor methods using multiple stored views. Nearest neighbor classifiers have the property [46] that as the number of examples n in the training set goes to infinity, the 1-NN error converges to a value $\leq 2E^*$, where E^* is the Bayes Risk (for K-NN, $K \to \infty$ and $K/n \to 0$, the error $\to E^*$). This is interesting because it shows that the humble nearest neighbor classifier is asymptotically optimal, a property not possessed by several considerably more complicated techniques. Of course, what matters in practice is the performance for small n, and this gives us a way to compare different similarity/distance measures.

5.1 Shape Distance

In this section we make precise our definition of shape distance and apply it to several practical problems. We used a regularized TPS transformation model and 3 iterations of shape context matching and TPS re-estimation. After matching, we estimated shape distances as the weighted sum of three terms: shape context distance, image appearance distance and bending energy.

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.

$$D_{\mathrm{sc}}(\mathcal{P}, \mathcal{Q}) \quad = \quad \frac{1}{n}\sum_{p\in\mathcal{P}} \arg\min_{q\in\mathcal{Q}} C\left(p, T\left(q\right)\right) + \frac{1}{m}\sum_{q\in\mathcal{Q}} \arg\min_{p\in\mathcal{P}} C\left(p, T\left(q\right)\right) \tag{12}$$

where $T(\cdot)$ denotes the estimated TPS shape transformation.

In many applications there is additional appearance information available that is not captured by our notion of shape, e.g. the texture and color information in the grayscale image patches surrounding corresponding points. The reliability of appearance information often suffers substantially from geometric image distortions. However, after establishing image correspondences and recovery of underlying 2D image transformation the distorted image can be warped back into a normal form, thus correcting for distortions of the image appearance.

We used a term $D_{\mathrm{ac}}(\mathcal{P}, \mathcal{Q})$ for appearance cost, defined as the sum of squared brightness differences in Gaussian windows around corresponding image points,

$$D_{\mathrm{ac}}(\mathcal{P}, \mathcal{Q}) \quad = \quad \sum_{i=1}^{n} \sum_{\Delta\in Z^2} G(\Delta)\left[I_{\mathcal{P}}\left(p_i+\Delta\right) - I_{\mathcal{Q}}\left(T\left(q_{\pi(i)}\right)+\Delta\right)\right]^2 \tag{13}$$

where $I_{\mathcal{P}}$ and $I_{\mathcal{Q}}$ are the grey-level images corresponding to $\mathcal{P}$ and $\mathcal{Q}$, respectively. Δ denotes some differential vector offset and G is a windowing function typically chosen to be a Gaussian, thus putting emphasis to pixels nearby. We thus sum over squared differences in windows around corresponding points, scoring the weighted grey-level similarity.

This score is computed *after* the thin plate spline transformation T has been applied to best warp the images into alignment.

The third term $D_{\mathrm{b}e}(\mathcal{P}, \mathcal{Q})$ corresponds to the 'amount' of transformation necessary to align the shapes. In the TPS case the bending energy (9) is a natural measure (see [5]).

5.2 Choosing Prototypes

In a prototype based approach, the key question is: what examples shall we store? Different categories need different numbers of views. For example, certain handwritten digits have more variability than others, e.g. one typically sees more variations in fours than in zeros. In the category of 3D objects, a sphere needs only one view, for example, while a telephone needs several views to capture the variety of visual appearance. This idea is related to the "aspect" concept as discussed in [30]. We will now discuss how we approach the problem of prototype selection.

In the nearest neighbor classifier literature, the problem of selecting exemplars is called *editing*. Extensive reviews of nearest neighbor editing methods can be found in Ripley [46] and Dasarathy [12]. We have developed a novel editing algorithm based on shape distance and K-medoid clustering. K-medoids can be seen as a variant of K-means that restricts prototype positions to data points. First a matrix of pairwise similarities between all possible prototypes is computed. For a given number of K prototypes the K-medoid algorithm then iterates two steps: (i) For a given assignment of points to (abstract) clusters a new prototype is selected for that cluster by minimizing the average distance of the prototype to all elements in the cluster, and (ii) given the set of prototypes, points are then reassigned to clusters according to the nearest prototype. More formally, denote by $c(\mathcal{P})$ the (abstract) cluster of shape $\mathcal{P}$, e.g. represented by some number $\{1,\ldots,k\}$ and denote by $p(c)$ the associated prototype. Thus we have a class map

$$c \ : \ \mathcal{S}_1 \subset \mathcal{S} \ \longrightarrow \ \{1,\ldots,k\} \tag{14}$$

and a prototype map

$$p \ : \ \{1,\ldots,k\} \ \longrightarrow \ \mathcal{S}_2 \subset \mathcal{S} \tag{15}$$

Here, $\mathcal{S}_1$ and $\mathcal{S}_2$ are some subsets of the set of all potential shapes $\mathcal{S}$. Often, $\mathcal{S} = \mathcal{S}_1 = \mathcal{S}_2$. K–medoids proceeds by iterating two steps:

1. group $\mathcal{S}_1$ into classes given the class prototypes $p(c)$, and

2. identify a representative prototype for each class given the elements in the cluster.

Basically, 1. is solved by assigning each shape $\mathcal{P} \in \mathcal{S}_1$ to the nearest prototype, thus

$$c(\mathcal{P}) \ = \ \arg\min_k D(\mathcal{P}, p(k)) \ . \tag{16}$$

For given classes, in 2. new prototypes are selected based on minimal mean dissimilarity, i.e.

$$p(k) \ = \ \arg\min_{p \in \mathcal{S}_2} \sum_{\mathcal{P}:c(shape)=k} D(\mathcal{P}, p) \ . \tag{17}$$

Since both steps minimize the same cost function

$$H\left(c,p\right) \ = \ \sum_{\mathcal{P}\in\mathcal{S}_1} D\left(\mathcal{P}, p\left(c(\mathcal{P})\right)\right) \tag{18}$$

the algorithm necessarily converges to a (local) minimum.

As with most clustering methods, with k-medoids one must have a strategy for choosing k. We select the number of prototypes using a greedy splitting strategy starting with

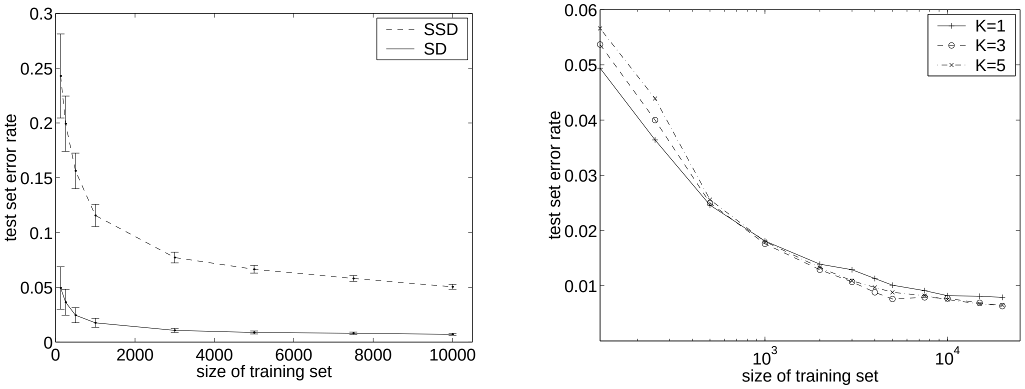

Figure 7. Handwritten digit recognition on the MNIST dataset. Left: Test set errors of a 1-NN classifier using SSD and Shape Distance (SD) measures. Right: Detail of performance curve for Shape Distance, including results with training set sizes of 15,000 and 20,000. Results are shown on a semilog-x scale for $K = 1, 3, 5$ nearest neighbors.

one prototype per category. We choose the cluster to split based on the associated overall misclassification error. This continues until the overall misclassification error has dropped below a criterion level. Thus the prototypes are automatically allocated to the different object classes, thus optimally using available resources. The application of this procedure to a set of views of 3D objects is explored in Section 6.2 and illustrated in Fig. 10.

6 Case Studies

6.1 Digit Recognition

Here we present results on the MNIST dataset of handwritten digits, which consists of 60,000 training and 10,000 test digits [34]. In the experiments, we used 100 points sampled from the Canny edges to represent each digit. When computing the C_{ij}'s for the bipartite matching, we included a term representing the dissimilarity of local tangent angles. Specifically, we defined the matching cost as $C_{ij} = (1-\beta)C_{ij}^{sc} + \beta C_{ij}^{tan}$, where C_{ij}^{sc} is the shape context cost, $C_{ij}^{tan} = 0.5(1-\cos(\theta_i - \theta_j))$ measures tangent angle dissimilarity, and $\beta = 0.1$. For recognition, we used a K–NN classifier with a distance function

$$D = 1.6D_{\mathrm{ac}} + D_{\mathrm{sc}} + 0.3D_{\mathrm{be}} \quad . \tag{19}$$

The weights in (19) have been optimized by a leave–one–out procedure on a 3000 × 3000 subset of the training data.

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%. The 63 errors are shown in Fig. 8.[4]

As mentioned earlier, what matters in practical applications of nearest neighbor methods is the performance for small n, and this gives us a way to compare different similarity/distance measures. In Fig. 7 (left) our shape distance is compared to SSD (sum of squared differences between pixel brightness values). In Fig. 7 (right) we compare the classification rates for different K.

[4] DeCoste and Schölkopf [13] report an error rate of 0.56% on the same database using Virtual Support Vectors (VSV) with the full training set of 60,000. VSVs are found as follows: (1) obtain SVs from the original training set using a standard SVM, (2) subject the SVs to a set of desired transformations (e.g. translation), (3) train another SVM on the generated examples.

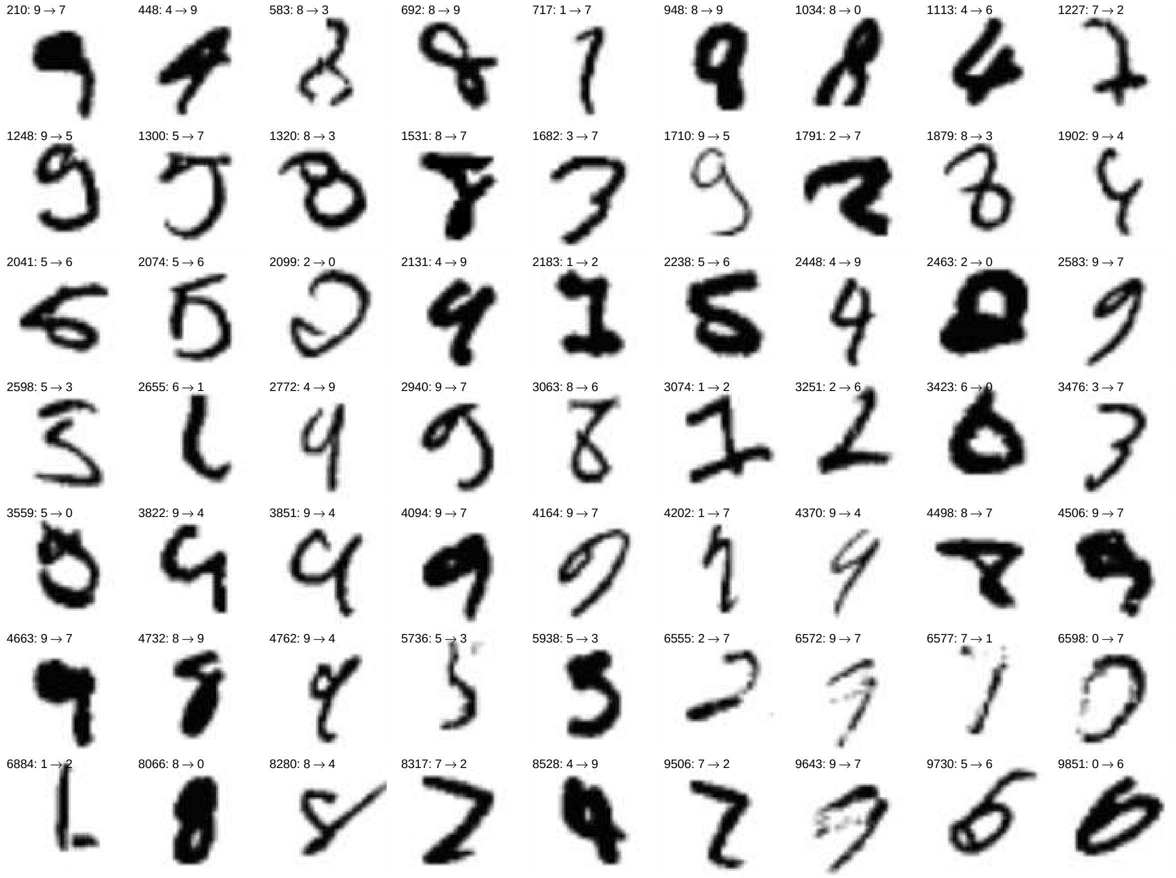

Figure 8. All of the misclassified MNIST test digits using our method (63 out of 10,000). The text above each digit indicates the example number followed by the true label and the assigned label.

6.2 3D Object Recognition

Our next experiment involves the 20 common household objects from the COIL-20 database [40]. 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 is **exactly** the same as for digits. Recall that the Canny edge detector responds both to external and internal contours, so the 100 sample points are not restricted to the external

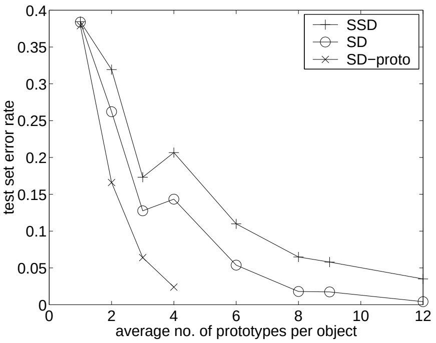

Figure 9. 3D object recognition using the COIL-20 dataset. Comparison of test set error for SSD, Shape Distance (SD), and Shape Distance with k-medoid prototypes (SD-proto) vs. number of prototype views. For SSD and SD, we varied the number of prototypes uniformly for all objects. For SD-proto, the number of prototypes per object depended on the within-object variation as well as the between-object similarity.

boundary of the silhouette.

Fig. 9 shows the performance using 1-NN with the distance function D as given in Equation 19 compared to a straightforward sum of squared differences (SSD). SSD performs very well on this easy database due to the lack of variation in lighting [24] (PCA just makes it faster).

The prototype selection algorithm is illustrated in Fig. 10. As seen, views are allocated mainly for more complex categories with high within class variability. The curve marked SC-proto in Fig. 9 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.

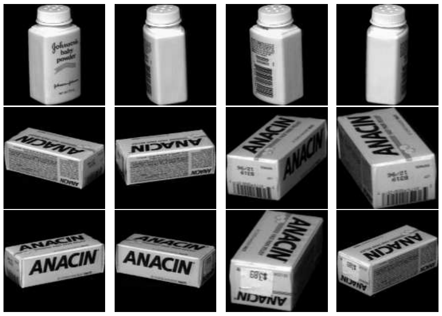

Figure 10. Prototype views selected for two different 3D objects from the COIL dataset using the algorithm described in Section 5.2. With this approach, views are allocated adaptively depending on the visual complexity of an object with respect to viewing angle.

6.3 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 [25]. 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

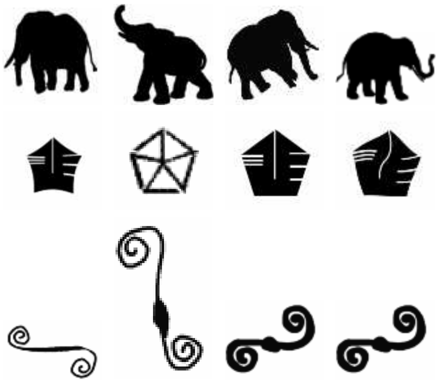

Figure 11. Examples of shapes in the MPEG7 database for three different categories.

is achieved by Latecki et al. [33], with a retrieval rate of 76.45%, followed by Mokhtarian et al. at 75.44%.

6.4 Trademark Retrieval

Trademarks are visually often best described by their shape information, and in many cases shape provides the only source of information. The automatic identification of trademark infringement has interesting industrial applications, since with the current state of the art trademarks are broadly categorized according to the Vienna code, and then manually classified according to their perceptual similarity. Even though shape context matching does not provide a full solution to the trademark similarity problem (other potential cues are text and texture), it still serves well to illustrate the capability of our approach to capture the essence of shape similarity. In Fig. 12 we depict retrieval results for a database of 300 trademarks. In this experiment, we relied on an affine transformation model as given by (3), and as in the previous case, we used 300 sample points.

We experimented with eight different query trademarks for each of which the database contained at least one potential infringement. We depict the top four hits as well as their similarity score. It is clearly seen that the potential infringements are easily detected and appear as most similar in 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.

7 Conclusion

We have presented a new approach to shape matching. A key characteristic of our approach is the estimation of shape similarity and correspondences based on a novel descriptor, the shape context. Our approach is simple and easy to apply, yet provides a rich descriptor for point sets that greatly improves point set registration, shape matching and shape recognition. In our experiments we have demonstrated invariance to several common image transformations, including significant 3D rotations of real-world objects.

A Completely Rotation Invariant Recognition

In this appendix we demonstrate the use of the relative frame in our approach as a means of obtaining complete rotation invariance. To demonstrate this idea, we have used the database provided by Kimia's group [53] shown in Fig. 13. In this experiment we used $n = 100$ sample points and, as mentioned above, we used the relative frame (Sect. 3.3) when computing the shape contexts. We used 5 bins for $\log(r)$ over the range 0.125α to 2α and 12 equally spaced radial bins in these and all other experiments in this paper. No transformation model at all was used. As a similarity score, we used the matching cost function $\sum_i C_{i,\pi(i)}$ after one iteration with no transformation step. Thus, this experiment is specifically designed solely to evaluate the power of the shape descriptor in the face of rotation.

In [53] and [17], the authors summarize their results on this dataset by stating the number of 1st, 2nd, and 3rd nearest neighbors that fall into the correct category. Our results are 25/25, 24/25, 22/25. In [53] and [17] the results quoted are 23/25, 21/25, 20/25 and 25/25, 21/25, 19/25, respectively.

B Sampling Considerations

In our approach a shape is represented by a set of sample points drawn from the internal and external contours of an object. Operationally, one runs an edge detector on the grayscale image and selects a subset of the edge pixels found. The selection could be uniformly at random, but we have found it to be advantageous to ensure that the sample points have a certain minimum distance between them as this makes sure that the sampling along the contours is somewhat uniform. (This corresponds to sampling from a point process which is a hard-core model [45].)

Since the sample points are drawn randomly and independently from the two shapes, there is inevitably jitter noise in the output of the matching algorithm which finds correspondences between these two sets of sample points. However, when the transformation between the shapes is estimated as a regularized thin plate spline, the effect of this jitter is smoothed away.

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. Parts of this work have appeared in [3, 2] We wish to thank H. Chui and A. Rangarajan for providing the synthetic testing data used in 4.2. We would also like to thank them and various members of the Berkeley computer vision group, particularly A. Berg, A. Efros, D. Forsyth, T. Leung, J. Shi and Y. Weiss, for useful discussions.

References

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

[2] S. Belongie, J. Malik, and J. Puzicha. Matching shapes. In *Proc. 8th Int'l. Conf. Computer Vision*, pages 454–461, July 2001.

[3] S. Belongie, J. Malik, and J. Puzicha. Shape context: A new descriptor for shape matching and object recognition. In T. K. Leen, T. G. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems 13: Proceedings of the 2000 Conference*, pages 831–837, 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.

[5] F. L. Bookstein. Principal warps: thin-plate splines and decomposition of deformations. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 11(6):567–585, June 1989.

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

[7] C. Burges and B. Schölkopf. Improving the accuracy and speed of support vector machines. In D. S. Touretzky, M. C. Mozer, and M. E. Hasselmo, editors, *Advances in Neural Information Processing Systems 9: Proceedings of the 1996 Conference*, pages 375–381, 1997.

[8] S. Carlsson. Order structure, correspondence and shape based categories. In *International Workshop on Shape, Contour and Grouping*, number 1681 in Springer Lecture Notes in Computer Science, May 1999.

[9] H. Chui and A. Rangarajan. A new algorithm for non-rigid point matching. In *Proc. IEEE Conf. Comput. Vision and Pattern Recognition*, pages 44–51, June 2000.

[10] 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.

[11] C. Cortes and V. Vapnik. Support vector networks. *Machine Learning*, 20:273–297, 1995.

[12] B. V. Dasarathy, editor. *Nearest neighbor (NN) norms : NN pattern classification techniques.* IEEE Computer Society, 1991.

[13] D. DeCoste and B. Schölkopf. Training invariant support vector machines. *Machine Learning*, 2002. to appear.

[14] J. Duchon. Splines minimizing rotation-invariant semi-norms in Sobolev spaces. In W. Schempp and K. Zeller, editors, *Constructive Theory of Functions of Several Variables*, pages 85 –100. Berlin: Springer-Verlag, 1977.

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

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

[17] Y. Gdalyahu and D. Weinshall. Flexible syntactic matching of curves and its application to automatic hierarchical classification of silhouettes. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 21(12):1312–1328, 1999.

[18] F. Girosi, M. Jones, and T. Poggio. Regularization theory and neural networks architectures. *Neural Computation*, 7(2):219–269, 1995.

[19] 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.

[20] E. Goldmeier. Similarity in visually perceived forms. *Psychological Issues*, 8(1):1–135, 1936/1972.

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

[22] M. Hagedoorn. *Pattern matching using similarity measures*. PhD thesis, Universiteit Utrecht, 2000.

[23] D. Huttenlocher, G. Klanderman, and W. Rucklidge. Comparing images using the Hausdorff distance. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 15(9):850–863, Sept. 1993.

[24] D. Huttenlocher, R. Lilien, and C. Olson. View-based recognition using an eigenspace approximation to the Hausdorff measure. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 21(9):951–955, Sept. 1999.

[25] 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.

[26] A. E. Johnson and M. Hebert. Recognizing objects by matching oriented points. In *Proc. IEEE Conf. Comput. Vision and Pattern Recognition*, pages 684–689, 1997.

[27] D. Jones and J. Malik. Computational framework to determining stereo correspondence from a set of linear spatial filters. *Image and Vision Computing*, 10(10):699–708, Dec. 1992.

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

[29] D. Kendall. Shape manifolds, procrustean metrics and complex projective spaces. *Bull. Lond. Math. Soc.*, 16:81–121, 1984.

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

[31] 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.

[32] Y. Lamdan, J. Schwartz, and H. Wolfson. Affine invariant model-based object recognition. *IEEE Trans. Robotics and Automation*, 6:578–589, 1990.

[33] L. J. Latecki, R. Lakämper, and U. Eckhardt. Shape descriptors for non-rigid shapes with a single closed contour. In *Proc. IEEE Conf. Comput. Vision and Pattern Recognition*, pages 424–429, 2000.

[34] 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.

[35] T. K. Leung, M. C. Burl, and P. Perona. Finding faces in cluttered scenes using random labelled graph matching. In *Proc. 5th Int'l. Conf. Computer Vision*, pages 637–644, 1995.

[36] D. G. Lowe. Object recognition from local scale-invariant features. In *Proc. 7th Int'l. Conf. Computer Vision*, pages 1150–1157, September 1999.

[37] J. Meinguet. Multivariate interpolation at arbitrary points made simple. *J. Appl. Math. Phys. (ZAMP)*, 5:439–468, 1979.

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

[39] 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.

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

[41] M. Oren, C. Papageorgiou, P. Sinha, E. Osuna, and T. Poggio. Pedestrian detection using wavelet templates. In *Proc. IEEE Conf. Comput. Vision and Pattern Recognition*, pages 193–199, Puerto Rico, June 1997.

[42] C. Papadimitriou and K. Stieglitz. *Combinatorial Optimization: Algorithms and Complexity.* Prentice Hall, 1982.

[43] E. Persoon and K. Fu. Shape discrimination using Fourier descriptors. *IEEE Trans. Systems, Man and Cybernetics*, 7(3):170–179, Mar. 1977.

[44] 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.

[45] B. D. Ripley. Modelling spatial patterns (with discussion). *Journal of Royal Statistical society, Series B*, 39:172–212, 1977.

[46] B. D. Ripley. *Pattern Recognition and Neural Networks.* Cambridge Univ. Press, 1996.

[47] E. Rosch. Natural categories. *Cognitive Psychology*, 4(3):328–350, 1973.

[48] E. Rosch, C. B. Mervis, W. D. Gray, D. M. Johnson, and P. Boyes-Braem. Basic objects in natural categories. *Cognitive Psychology*, 8(3):382–439, 1976.

[49] C. Schmid and R. Mohr. Local grayvalue invariants for image retrieval. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 19(5):530–535, May 1997.

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

[51] G. Scott and H. Longuet-Higgins. An algorithm for associating the features of two images. *Proc. of the Royal Soc. London*, B-244:21–26, 1991.

[52] L. S. Shapiro and J. M. Brady. Feature-based correspondence: an eigenvector approach. *Image and Vision Computing*, 10(5):283–288, June 1992.

[53] D. Sharvit, J. Chan, H. Tek, and B. Kimia. Symmetry-based indexing of image databases. *J. Visual Communication and Image Representation*, 9(4):366–380, December 1998.

[54] L. Sirovich and M. Kirby. Low dimensional procedure for the characterization of human faces. *J. Opt. Soc. Am. A*, 4(3):519–524, 1987.

[55] D. W. Thompson. *On Growth and Form.* Cambridge University Press, 1917.

[56] M. Turk and A. Pentland. Eigenfaces for recognition. *J. Cognitive Neuroscience*, 3(1):71–96, 1991.

[57] S. Umeyama. An eigen decomposition approach to weighted graph matching problems. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 10(5):695–703, Sept. 1988.

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

[59] T. Vetter, M. J. Jones, and T. Poggio. A bootstrapping algorithm for learning linear models of object classes. In *Proc. IEEE Conf. Comput. Vision and Pattern Recognition*, pages 40–46, 1997.

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

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

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

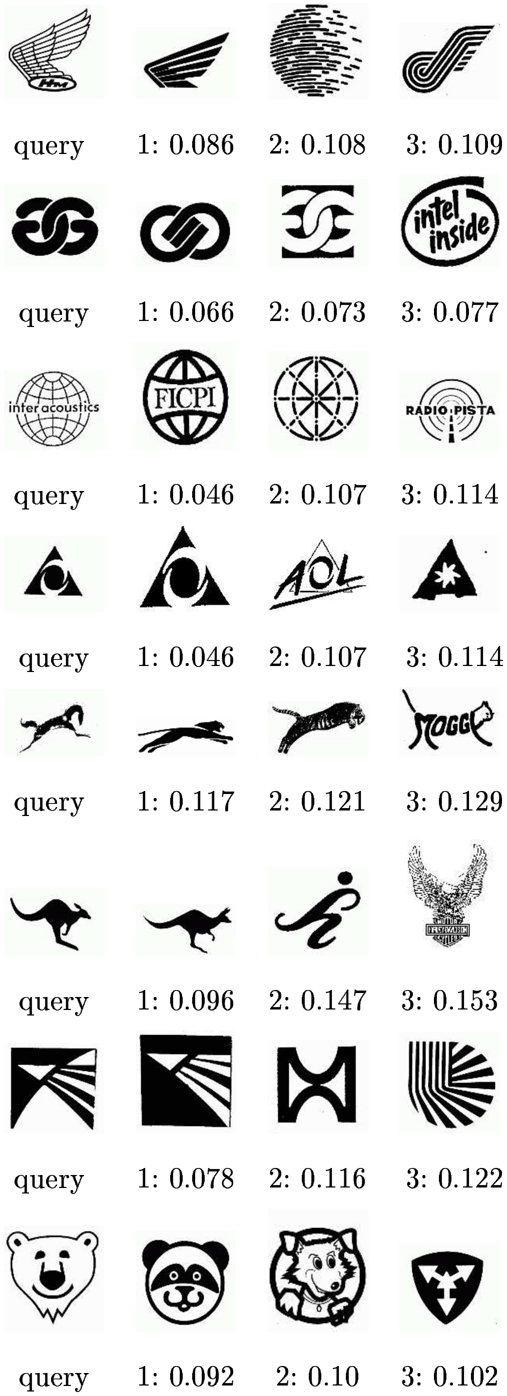

Figure 12. Trademark retrieval results based on a database of 300 different real-world trademarks. We used an affine transformation model and a weighted combination of shape context similarity D_{sc} and the sum over local tangent orientation differences.

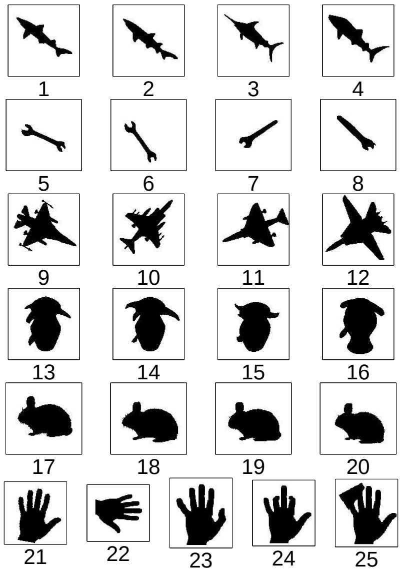

Figure 13. Kimia dataset: each row shows instances of a different object category. Performance is measured by the number of closest matches with the correct category label. Note that several of the categories require rotation invariant matching for effective recognition. All of the 1st ranked closest matches were correct using our method. Of the 2nd ranked matches, one error occurred in 1 vs. 8. In the 3rd ranked matches, confusions arose from 2 vs. 8, 8 vs. 1, and 15 vs. 17.

About the Authors

Serge Belongie was born in Sacramento, California, in 1974. He received the B.S. degree (with honor) in Electrical Engineering from the California Institute of Technology in 1995 and the M.S. and Ph.D. degrees in Electrical Engineering and Computer Sciences (EECS) at U.C. Berkeley in 1997 and 2000, respectively. While at Berkeley, his research was supported by a National Science Foundation Graduate Research Fellowship and the Chancellor's Opportunity Predoctoral Fellowship. He is also a co-founder of Digital Persona, Inc., and the principal architect of the Digital Persona fingerprint recognition algorithm. He is currently an assistant professor in the Computer Science and Engineering Department at U.C. San Diego. His research interests include computer vision, pattern recognition, and digital signal processing.

Jitendra Malik was born in Mathura, India in 1960. He received the B.Tech degree in Electrical Engineering from Indian Institute of Technology, Kanpur in 1980 and the PhD degree in Computer Science from Stanford University in 1986. In January 1986, he joined the faculty of the Computer Science Division, Department of EECS, University of California at Berkeley, where he is currently a Professor. During 1995-1998 he also served as Vice-Chair for Graduate Matters. He is a member of the Cognitive Science and Vision Science groups at UC Berkeley.

His research interests are in computer vision and computational modeling of human vision. His work spans a range of topics in vision including image segmentation and grouping, texture, stereopsis, object recognition, image based modeling and rendering, content based image querying, and intelligent vehicle highway systems. He has authored or co-authored more than a hundred research papers on these topics.

He received the gold medal for the best graduating student in Electrical Engineering from IIT Kanpur in 1980, a Presidential Young Investigator Award in 1989, and the Rosenbaum fellowship for the Computer Vision Programme at the Newton Institute of Mathematical Sciences, University of Cambridge in 1993. He received the Diane S. McEntyre Award for Excellence in Teaching from the Computer Science Division, University of California at Berkeley, in 2000. He is an Editor-in-Chief of the International Journal of Computer Vision.

Jan Puzicha received the Diploma degree in 1995 and the Ph.D. degree in computer science in 1999, both from the University of Bonn, Bonn, Germany. He was with the Computer Vision and Pattern Recognition Group, University of Bonn, from 1995 to 1999. In September 1999, he joined the Computer Science Department, University of California, Berkeley, as an Emmy Noether Fellow of the German Science Foundation, where he is currently working on optimization methods for perceptual grouping and image segmentation. His research interests include computer vision, image processing, unsupervised learning, data analysis, and data mining.

Author Contact Information

- **Serge Belongie** (corresponding author)
  Department of Computer Science and Engineering
  AP&M Building, Room 4832
  University of California, San Diego
  La Jolla, CA 92093-0114
  Phone: (858) 822-5163
  Fax: (858) 534-7029
  E-mail: sjb@cs.ucsd.edu

- **Jitendra Malik**
  725 Soda Hall
  Computer Science Division
  University of California at Berkeley
  CA 94720-1776
  Phone: (510) 642-7597
  Fax: (510) 643-1534
  E-mail: malik@cs.berkeley.edu

- **Jan Puzicha**
  RecomMind, Inc.
  1001 Camelia Street
  Berkeley, CA 94710
  Phone: (510) 558-7890
  Fax: (510) 525-2351
  E-mail: jan@recommind.com