Matching with Shape Contexts

Serge Belongie[1], Greg Mori[2], and Jitendra Malik[3]

[1] University of California, San Diego sjb@cs.ucsd.edu
[2] Simon Fraser University mori@cs.sfu.ca
[3] University of California, Berkeley malik@cs.berkeley.edu

Summary. 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. We also demonstrate that shape contexts can be used to quickly prune a search for similar shapes. We present two algorithms for rapid shape retrieval: *representative shape contexts*, performing comparisons based on a small number of shape contexts, and *shapemes*, using vector quantization in the space of shape contexts to obtain prototypical shape pieces. Results are presented for silhouettes, handwritten digits and visual CAPTCHAs.

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 chapter 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* [39], 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. [18] developed these ideas in a probabilistic setting. Yuille [42] 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 [24] 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 [35, 36]. They have the advantage that a vector space structure is not required–only a pairwise dissimilarity measure.

We demonstrate object recognition in a wide variety of settings. Results are presented on the MNIST dataset of handwritten digits (Fig. 8), silhouettes (Fig. 9), the Snodgrass and Vanderwart line drawings (Fig. 10), and the EZ-Gimpy CAPTCHA (Fig. 12).

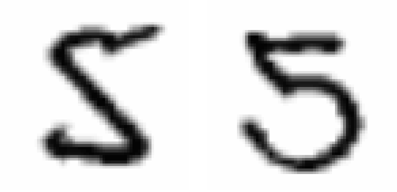


Fig. 1. Examples of two handwritten digits. In terms of pixel-to-pixel comparisons, these two images are quite different, but to the human observer, the shapes appear to be similar.

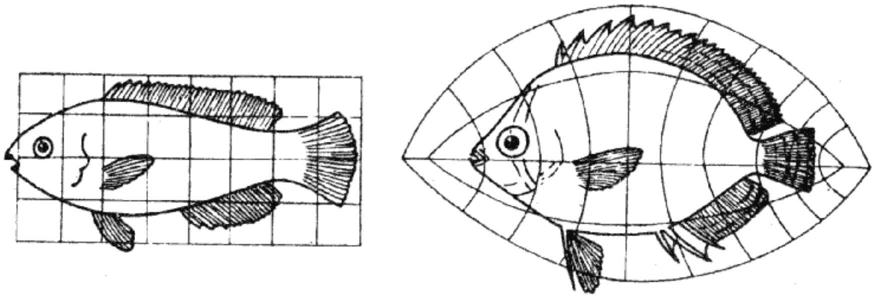


Fig. 2. Example of coordinate transformations relating two fish, from D'Arcy Thompson's *On Growth and Form* [39]. Thompson observed that similar biological forms could be related by means of simple mathematical transformations between *homologous* (i.e. corresponding) features. Examples of homologous features include center of eye, tip of dorsal fin, etc.

The structure of this chapter is as follows. We begin by introducing the shape context descriptor in Section 2. In Section 3 we develop the shape context based matching framework. We provide experimental results in a variety of application areas in Section 4. Finally, we conclude in Section 5.

2 The Shape Context

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. 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.

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 [22], 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 plays 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,

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

This histogram is defined to be the *shape context* of p_i. We use bins that are uniform in log-polar[4] 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 $C_{ij} = C(p_i, q_j)$ denote the cost of matching these two points. As shape

[4] 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.

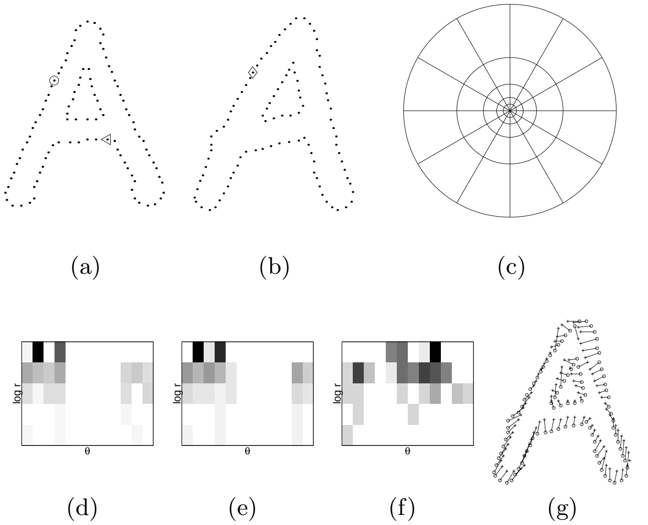


Fig. 3. Shape context computation and matching. (a,b) Sampled edge points of two shapes. (c) Diagram of log-polar histogram bins used in computing the shape contexts. We use 5 bins for $\log r$ and 12 bins for θ. (d-f) Example shape contexts for reference samples marked by $\circ, \diamond, \triangleleft$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

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.[5] The set of costs C_{ij} over all i and j provide us with a matrix that can be used as the input to a variety of bipartite matching algorithms, to be discussed in Section 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

[5] Alternatives include Bickel's generalization of the Kolmogorov-Smirnov test for 2D distributions [7], which does not require binning, or treating the shape contexts as vectors and comparing them using an L_p norm.

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.

2.1 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 [4].

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. 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 in an iterative fashion. 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.

2.2 Generalized Shape Contexts

The spatial structure of the shape context histogram bins, with central bins smaller than those in the periphery, results in a descriptor that is more precise about the location of nearby features, and less precise about those farther away. When additional features, such as local edgel orientations, are available, this same structure can be applied to construct a richer descriptor. We call these extended descriptors *generalized shape contexts*.

We have experimented with an instantiation of generalized shape contexts based on edgel orientations. To each edge point q_j we attach a unit length tangent vector t_j that is the direction of the edge at q_j. In each bin we sum the tangent vectors for all points falling in the bin. The descriptor for a point p_i is the histogram $\hat{h}_i$:

$$\hat{h}_i^k = \sum_{q_j \in Q} t_j, \text{ where } Q = \{q_j \neq p_i, (q_j - p_i) \in \text{bin}(k)\}$$

Each bin now holds a single vector in the direction of the dominant orientation of edges in the bin. When comparing the descriptors for two points, we convert this d-bin histogram to a $2d$-dimensional vector $\hat{v}_i$, normalize these vectors, and compare them using the L_2 norm:

$$\hat{v}_i = \langle \hat{h}_i^{1,x}, \hat{h}_i^{1,y}, \hat{h}_i^{2,x}, \hat{h}_i^{2,y}, ..., \hat{h}_i^{d,x}, \hat{h}_i^{d,y} \rangle$$

$$d_{GSC}(\hat{h}_i, \hat{h}_j) = ||\hat{v}_i - \hat{v}_j||_2$$

where $\hat{h}_i^{j,x}$ and $\hat{h}_i^{j,y}$ are the x and y components of $\hat{h}_i^j$ respectively.

Note that these generalized shape contexts reduce to the original shape contexts if all tangent angles are clamped to zero. Our experiments in Section 4 will compare these new descriptors with the original shape contexts.

2.3 Shapemes: Vector Quantized Shape Contexts

Another extension we have explored uses vector quantization on the shape contexts. Given a set $|\$|$ of shapes, and shape contexts computed at s sample points on these shapes, the full set of shape contexts for the known shapes consists of $|\$| \cdot s$ d-dimensional vectors. A standard technique in compression for dealing with such a large amount of data is vector quantization. Vector quantization involves clustering the vectors and then representing each vector by the index of the cluster that it belongs to. We call these clusters *shapemes* – canonical shape pieces.

To derive these shapemes, all of the shape contexts from the known set are considered as points in a d-dimensional space. We perform k-means clustering to obtain k shapemes. Figure 5 shows the representation of sample points as shapeme labels.

2.4 Related Descriptors

Local patch models

Recent years have seen the emergence of local patch models as approaches [1, 12, 14, 27] for object recognition. These approaches capture appearance information through a collection of local image patches, while shape information is encoded via spatial relationships between the local patches. The locations for the local patches are selected with various interest point operators, and are represented either as raw pixel values [14] or histograms of image gradients [12, 27], termed SIFT descriptors (Scale Invariant Feature Transform).

The major differences between our work using shape contexts and the above methods are in the scope of the descriptor and the locations at which they are computed. Shape contexts are a relatively large scale point descriptor. With a radius of approximately half the diameter of an object each shape context captures information from almost the entire shape. Second, the shape contexts are computed at a dense set of locations spread over the entire shape, as opposed to the interest points selected in the other approaches.

Extensions to 3D

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 [21] 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.

Frome et al. [16] have extended the original 2D shape contexts for use in matching 3D point sets such as those obtained via laser range finders. The extension is a natural one – an oriented sphere centered at each point in 3D is divided into bins with equally spaced boundaries in the azimuth and elevation dimensions, and logarithmically spaced boundaries in the radial dimension. Frome et al. present results showing that these 3D shape contexts outperform spin images in 3D object recognition tasks.

Extension to the Continuous Case

Berg and Malik [6] developed a descriptor which is akin to a shape context for grayscale images. Their features are based on a spatially varying smoothing of edge energy, termed "geometric blur", which increases along with the distance from the center of the descriptor. This variation in smoothing level is similar to the increase in radial width of the shape context bins as one moves away from the center of the shape context descriptor.

3 Matching Framework

We turn now to the use of shape contexts as part of a theory of object recognition based on shape matching. As stated earlier, it is desirable for such a theory to support both accurate fine discrimination, as well as rapid coarse discrimination. This suggests a two stage approach to shape matching, namely:

1. *Fast pruning:* Given an unknown 2D query shape, we should be able to quickly retrieve a small set of likely candidate shapes from a potentially very large collection of stored shapes. We have developed two algorithms for this problem.

2. *Detailed matching:* Once we have a small set of candidate shapes, we can perform a more expensive and more accurate matching procedure to find the best matching shape to the query shape.

In this work we will not address the problem of scale estimation. Shapes will be presented in a setting that allows for simple estimation of scale via the mean distance between points on a shape. In a natural setting, multi-scale search could be performed, or scale-invariant interest point detection or segmentation could be used to estimate scale.

3.1 Fast Pruning

Given a large set of known shapes the problem is to determine which of these shapes is most similar to a query shape. From this set of shapes, we wish to quickly construct a shortlist of candidate shapes which includes the best matching shape. After completing this coarse comparison step one can then apply a more time consuming, and more accurate, comparison technique to only the shortlist. We leverage the descriptive power of shape contexts towards this goal of quick pruning.

We have developed two matching methods that address these issues. In the first method, *representative shape contexts* (RSCs), we compute a few shape contexts for the query shape and attempt to match using only those. The second method uses the *shapemes* defined above to efficiently compare the entire set of shape contexts for a query shape to the set of known shapes.

Representative Shape Contexts

Given two easily discriminable shapes, such as the outlines of a fish and a bicycle, we do not need to compare every pair of shape contexts on the objects to know that they are different. When trying to match the dissimilar fish and bicycle, none of the shape contexts from the bicycle have good matches on the fish – it is immediately obvious that they are different shapes. Figure 4 demonstrates this process. The first pruning method, *representative shape contexts*, uses this intuition.

In concrete terms, the matching process proceeds in the following manner. For each of the known shapes S_i, we precompute a large number s (about

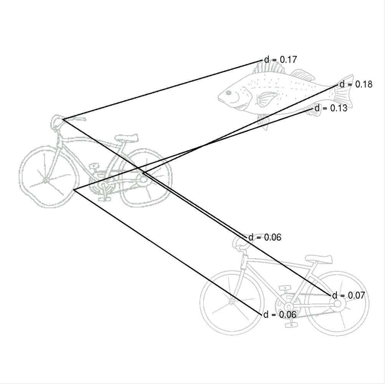


Fig. 4. Matching individual shape contexts. Three points on the query shape (left) are connected with their best matches on two known shapes. L_2 distances are given with each matching.

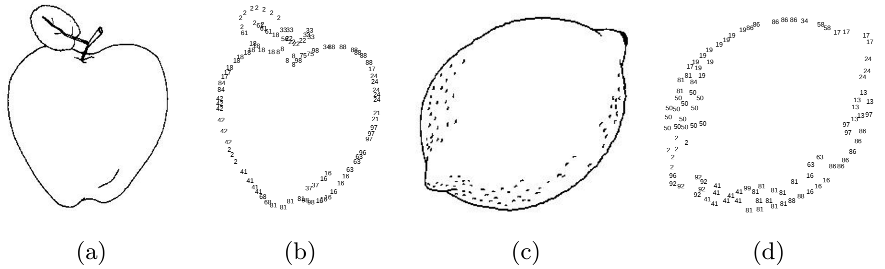


Fig. 5. (a,c) Line drawings. (b,d) Sampled points with shapeme labels. $k = 100$ shapemes were extracted from a known set of 260 shapes (26000 generalized shape contexts). Note the similarities in shapeme labels (2,41 on left side, 24,86,97 on right side) between similar portions of the shapes.

100) of shape contexts $\{SC_i^j : j = 1, 2, \ldots, s\}$. But for the query shape, we only compute a small number r ($r \approx 5-10$ in experiments) of shape contexts. To compute these r shape contexts we randomly select r sample points from the shape via a rejection sampling method that spreads the points over the entire shape. We use all the sample points on the shape to fill the histogram bins for the shape contexts corresponding to these r points. To compute the distance between a query shape and a known shape, we find the best matches for each of the r RSCs.

Note that in cluttered images many of the RSCs contain noisy data, or are not located on the shape S_i. Hence, for each of the known shapes S_i we find the best k RSCs, the ones with the smallest distances. Call this set of indices G_i. The distance between shapes Q and S_i is then:

$$d_S(Q, S_i) = \frac{1}{k} \sum_{u \in G_i} \frac{d_{GSC}(SC_Q^u, SC_i^{m(u)})}{N_u}$$

$$\text{where } m(u) = \arg\min_j d_{GSC}(SC_Q^u, SC_i^j)$$

N_u is a normalizing factor that measures how discriminative the representative shape context SC_Q^u is:

$$N_u = \frac{1}{|\$|} \sum_{S_i \in \$} d_{GSC}(SC_Q^u, SC_i^{m(u)})$$

where $\$$ is the set of all shapes. We determine the shortlist by sorting these distances.

Pruning with Shapemes

The second pruning method makes use of the vector quantization process described earlier to reduce the complexity of comparing two shapes. We represent each shape as a collection of shapemes. Each d-bin shape context is quantized to its nearest shapeme, and replaced by the shapeme label (an integer in $\{1, \ldots, k\}$). A shape is then simplified into a histogram of shapeme frequencies. No spatial information amongst the shapemes is stored. We have reduced each collection of s shape contexts (d bin histograms) to a single histogram with k bins.

In order to match a query shape, we simply perform this same vector quantization and histogram creation operation on the shape contexts from each of the known shapes and the query shape. We then find nearest neighbours in the space of histograms of shapemes to construct a shortlist of potential matches.

3.2 Detailed Matching

The process of detailed matching consists for two basic steps, which we operationalize in an iterative fashion: (1) solving for correspondences and (2) transformation into alignment.

Correspondence

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 wish to determine the one-to-one correspondences between them. A number of algorithms can be used for this purpose. The simplest method is nearest neighbor, consisting of one arg min pass on the rows of C followed by another pass on the columns to break many-to-one mappings. This is fast, but will in general leave a number of points unassigned. A better approach is to find the permutation π that minimizes 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. 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 [32]. In our experiments, we use the more efficient algorithm of [23]. 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.

The above cost function can be augmented to incorporate mappings of pairs of correspondences, so that geometric distortion can be taken into account simultaneously with point-to-point matching cost. Berg et al. [5] take such an approach, for which they employ an approximate solution of the Integer Quadratic Programming problem.

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.

Transformation into Alignment

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

$$\hat{o} = \frac{1}{n}\sum_{i=1}^{n}\left(p_i - q_{\pi(i)}\right) \quad , \tag{4}$$

$$\hat{A} = (Q^+ P)^t \tag{5}$$

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

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

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

In this work, we mostly use the thin plate spline (TPS) model [13, 29], which is commonly used for representing flexible coordinate transformations. Bookstein [9] found it to be highly effective for modeling changes in biological forms. Powell applied the TPS model to recover transformations between curves [33]. Chui and Rangarajan [10] use TPS in their robust point matching algorithm. 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

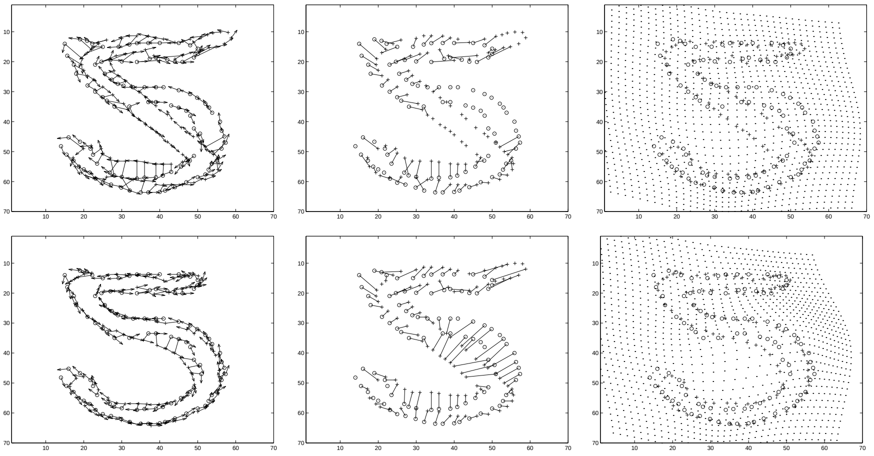


Fig. 6. Illustration of the matching process applied to the example of Fig. 1. Top row: 1st iteration. Bottom row: 5th iteration. Left column: estimated correspondences shown relative to the transformed model, with tangent vectors shown. Middle column: estimated correspondences shown relative to the 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 $\lambda_o = 1$.

this system by L. As discussed e.g. in [33], 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

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

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 [17,41], 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,

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

which yields a displacement field that maps any position in the first image to its interpolated location in the second image.[6]

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. 6.

4 Applications

Given a measure of dissimilarity between shapes, we can proceed to apply it to the task of object recognition. Specifically, we treat the problems of recognizing handwritten digits, shape silhouettes, line drawings of common objects, and visual CAPTCHAs (tests that most humans can pass, but that computers are meant not to). Our approach falls into the category of prototype-based recognition. In this framework, pioneered by Rosch and collaborators [36], 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

[6] One potential problem with the use of TPS is that it can admit local folds and reflections in the mapping, and it might not have an inverse. Guo et al. [19] employ an approach that addresses this problem by means of estimating a diffeomorphism between the corresponding pointsets.

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 [34] 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.

4.1 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.

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

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 $D_{\text{ac}}\left(\mathcal{P},\mathcal{Q}\right)$ for appearance cost, defined as the sum of squared brightness differences in Gaussian windows around corresponding image points,

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

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

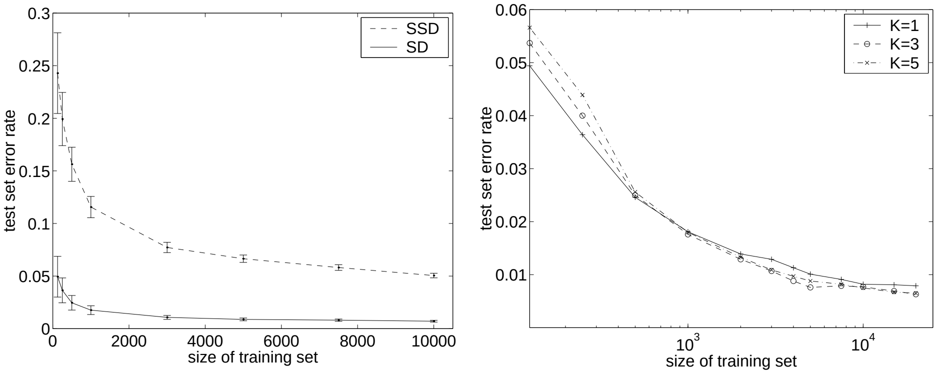


Fig. 7. Handwritten digit recognition on the MNIST dataset. Left: Test set errors of a 1-NN classifier using SSD and Shape Distance (SD) measures. Right: Detail of performance curve for Shape Distance, including results with training set sizes of 15,000 and 20,000. Results are shown on a semilog-x scale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

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 $D_{\mathrm{be}}(\mathcal{P}, \mathcal{Q})$ corresponds to the ‘amount’ of transformation necessary to align the shapes. In the TPS case the bending energy (9) is a natural measure (see [8]).

4.2 Digit Recognition

Here we present results on the MNIST dataset of handwritten digits, which consists of 60,000 training and 10,000 test digits [26]. 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

$$D = 1.6 D_{\mathrm{ac}} + D_{\mathrm{sc}} + 0.3 D_{\mathrm{be}} \ . \tag{14}$$

The weights in (14) 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://yann.lecun.com/exdb/mnist/`). The lowest test set error rate published at this time is 0.7% for a boosted LeNet-4 with a training set of size $60,000 \times 10$

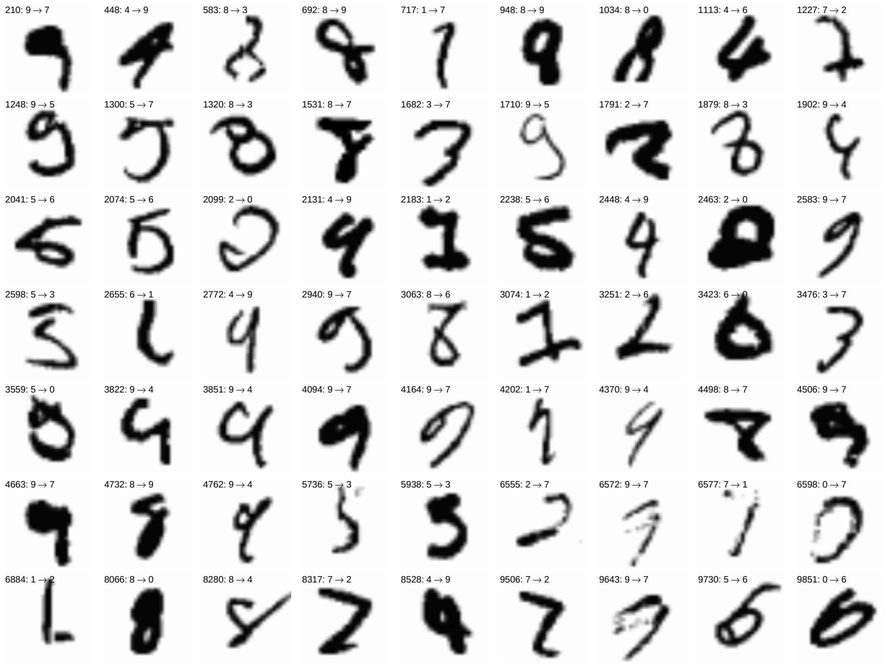


Fig. 8. All of the misclassified MNIST test digits using our method (63 out of 10,000). The text above each digit indicates the example number followed by the true label and the assigned label.

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.[7]

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.

4.3 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

[7] DeCoste and Schölkopf [11] 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.

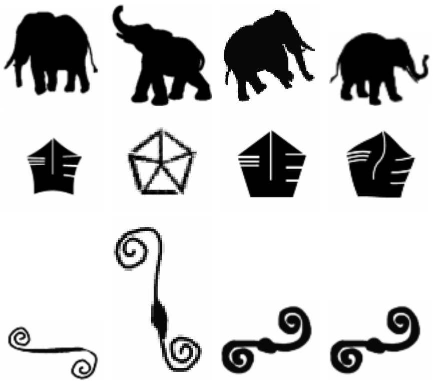


Fig. 9. Examples of shapes in the MPEG7 database for three different categories.

similarity-based retrieval [20]. 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 $\mathrm{dist}(R, Q)$ between a reference shape R and a query shape Q is defined as

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

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. [25], with a retrieval rate of 76.45%, followed by Mokhtarian et al. at 75.44%.

4.4 Snodgrass and Vanderwart

To illustrate our algorithms for fast pruning, we use the Snodgrass & Vanderwart line drawings [37]. They are a standard set of 260 objects that have been frequently used in the psychophysics community for tests with human subjects.

The Snodgrass & Vanderwart dataset has only one image per object. We use these original images as the training set, and create a synthetic set of distorted and partially occluded shapes for querying. We distort each shape by applying a random TPS warp of fixed bending energy to a reference grid, and use this warp to transform the edge points of the shape. Occlusions are then generated using a random linear occluding contour.

We generated 5200 distorted and occluded images (20 per original image) for use as a test set. The occluded images were split into levels of difficulty according to the percentage of edge pixels lost under occlusion. Figures 10 and 11 show the results for our two pruning methods. The graphs plot error rate vs. pruning factor (on a log scale). The error rate computation assumes a perfect detailed matching phase. That is, a query shape produces an error only if there is no correctly matching shape in the shortlist obtained by the pruning method. The abscissa on each of the graphs shows the pruning factor, defined to be $|\mathbb{S}|/length(Shortlist)$. For example, with $|\mathbb{S}| = 260$ known shapes, if the pruning factor is 26 then the shortlist has 10 shapes in it.

Note that on this dataset, the generalized shape contexts perform slightly worse than the original shape context descriptors. The reason for this is that the synthetic TPS distortions used to create the test set corrupt the tangent vectors used in generalized shape contexts. The random TPS distortions contain local scale warps that deform the tangent vectors greatly.

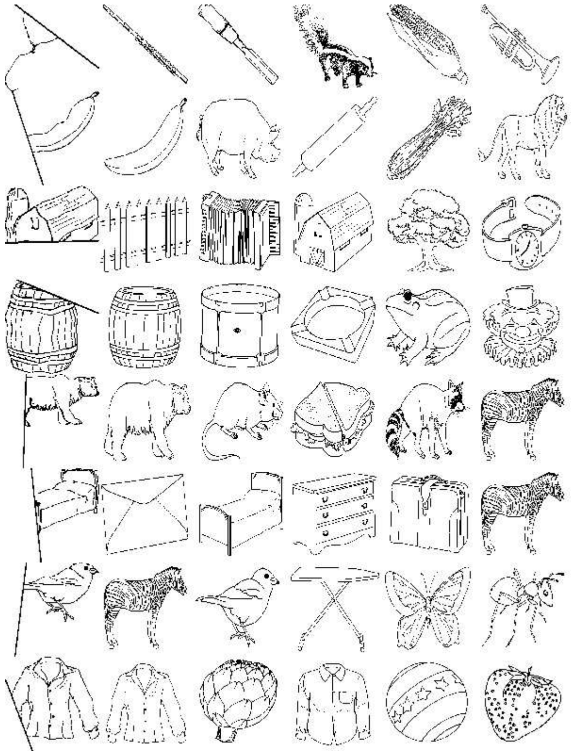
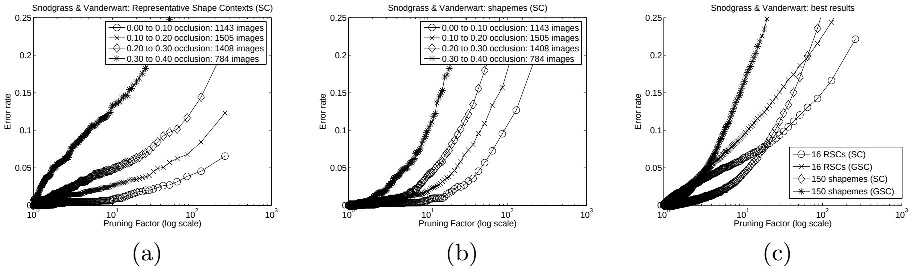


Fig. 10. Shortlists for the distorted and occluded Snodgrass & Vanderwart dataset using the representative shape contexts method. The first column is the query object. Remaining 5 columns show closest matches to each query object.



(a) (b) (c)

Fig. 11. Error rate vs. pruning factor on Snodgrass dataset. (a,b) Variation in performance with respect to amount of occlusion in test image. (c) Comparative results for all different methods. Results for best parameter settings from each method are shown.

4.5 Captcha

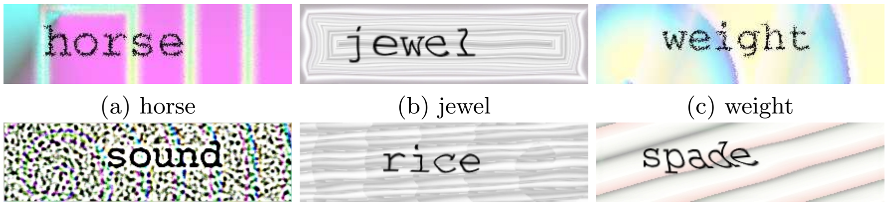
A CAPTCHA is a program [40] that can generate and grade tests that most humans can pass, but current computer programs can't pass. CAPTCHA stands for "Completely Automated Public Turing test to Tell Computers and Humans Apart". EZ-Gimpy (Figure 12) is a CAPTCHA based on word recognition in the presence of clutter. The task is to identify a single word, chosen from a known dictionary of 561 words, that has been distorted and placed in a cluttered image.

For our experiments, a training set of the 561 words, each presented undistorted on an uncluttered background, was constructed. We applied the representative shape contexts pruning method, using the 561 words as our objects, followed by detailed matching to recognize the word in each EZ-Gimpy image. This algorithm is referred to as "Algorithm B" in our previous work on breaking CAPTCHAs [31]. Two details are different from those in the other experiments. First, we constructed generalized shape contexts that are tuned to the shape of words: they are elliptical, with an outer radius of about 4 characters horizontally, and $\frac{3}{4}$ of a character vertically. Second, the texture gradient operator [28] was used to select the placement of the RSCs, while Canny edge detection is used to find edge pixels to fill the bins of the shape contexts.

We generated 200 examples of the EZ-Gimpy CAPTCHA. Of these examples, 9 were used for tuning parameters in the texture gradient modules. The remaining 191 examples were used as a test set. Examples of the EZ-Gimpy CAPTCHA images used, and the top matching words are shown in Fig. 12, the full set of test images and results can be viewed at `http://www.cs.sfu.ca/~mori/research/gimpy/ez/`. In 92% (176/191) of these test cases, our method identified the correct word. This success rate compares favourably with that of Thayananthan et al. [38] who perform exhaustive search using Chamfer matching with distorted prototype words.

Of the 15 errors made, 9 were errors in the RSC pruning. The pruning phase reduced the 561 words to a shortlist of length 10. For 9 of the test images the correct word was not on the shortlist. In the other 6 failure cases, the deformable matching selected an incorrect word from the shortlist.

The generalized shape contexts are much more resilient to the clutter in the EZ-Gimpy images than the original shape contexts. The same algorithm, run using the original shape contexts, attains only a 53% success rate on the test set.



(a) horse (b) jewel (c) weight

(d) sound (e) rice (f) space

Fig. 12. Results on EZ-Gimpy images. The best matching word is shown below each image.

5 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. To address the computational expense associated with large scale object databases, we have also shown how a shape context-based pruning approach can construct an accurate shortlist.

Acknowledgments

This research was supported by (ARO) DAAH04-96-1-0341, the Digital Library Grant IRI-9411334, an NSF Graduate Fellowship, and the German Research Foundation by DFG grant PU-165/1 to our collaborator J. Puzicha, now at Recommind, Inc. Parts of this work have appeared in [2–4,30,31]. We would also like to thank H. Chui, A. Rangarajan, 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*, volume 1, 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. Serge Belongie, Jitendra Malik, and Jan Puzicha. Shape matching and object recognition using shape contexts. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 24(4):509–522, April 2002.
5. A. Berg, T. Berg, and J. Malik. Shape matching and object recognition using low distortion correspondences. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, San Diego, CA, June 2005. to appear.
6. A. Berg and J. Malik. Geometric blur for template matching. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 607–614, Kauai, HI, 2001.
7. 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.
8. 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.
9. F. L. Bookstein. *Morphometric tools for landmark data: geometry and biology.* Cambridge Univ. Press, 1991.
10. 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.
11. D. DeCoste and B. Schölkopf. Training invariant support vector machines. *Machine Learning*, 2002. to appear.
12. G. Dorko and C. Schmid. Selection of scale invariant neighborhoods for object class recognition. In *Proc. 9th Int. Conf. Computer Vision*, pages 634–640, 2003.
13. 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.
14. R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 264–271, 2003.
15. M. Fischler and R. Elschlager. The representation and matching of pictorial structures. *IEEE Trans. Computers*, C-22(1):67–92, 1973.
16. A. Frome, D. Huber, R. Kolluri, T. Bulow, and J. Malik. Recognizing objects in range data using regional point descriptors. In *Proc. 8th Europ. Conf. Comput. Vision*, volume 3, pages 224–237, 2004.
17. F. Girosi, M. Jones, and T. Poggio. Regularization theory and neural networks architectures. *Neural Computation*, 7(2):219–269, 1995.
18. U. Grenander, Y. Chow, and D.M. Keenan. *HANDS: A Pattern Theoretic Study Of Biological Shapes.* Springer, 1991.

19. H. Guo, A. Rangarajan, S. Joshi, and L. Younes. A new joint clustering and diffeomorphism estimation algorithm for non-rigid shape matching. In *IEEE Workshop on Articulated and Non-rigid motion (ANM)*, Washington, DC, 2004.
20. 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.
21. Andrew E. Johnson and Martial Hebert. Recognizing objects by matching oriented points. In *Proc. IEEE Conf. Comput. Vision and Pattern Recognition*, pages 684–689, 1997.
22. 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.
23. R. Jonker and A. Volgenant. A shortest augmenting path algorithm for dense and sparse linear assignment problems. *Computing*, 38:325–340, 1987.
24. M. Lades, C.C. Vorbrüggen, J. Buhmann, J. Lange, C. von der Malsburg, R.P. Wurtz, and W. Konen. Distortion invariant object recognition in the dynamic link architecture. *IEEE Trans. Computers*, 42(3):300–311, March 1993.
25. 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.
26. 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.
27. David G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. Journal of Computer Vision*, 60(2):91–110, 2004.
28. D. Martin, C. Fowlkes, and J. Malik. Learning to find brightness and texture boundaries in natural images. *NIPS*, 2002.
29. J. Meinguet. Multivariate interpolation at arbitrary points made simple. *J. Appl. Math. Phys. (ZAMP)*, 5:439–468, 1979.
30. G. Mori, S. Belongie, and J. Malik. Shape contexts enable efficient retrieval of similar shapes. In *Proc. IEEE Conf. Comput. Vision and Pattern Recognition*, December 2001.
31. G. Mori and J. Malik. Recognizing objects in adversarial clutter: Breaking a visual captcha. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, volume 1, pages 134–141, Madison, WI, 2003.
32. C. Papadimitriou and K. Stieglitz. *Combinatorial Optimization: Algorithms and Complexity.* Prentice Hall, 1982.
33. 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.
34. B. D. Ripley. *Pattern recognition and neural networks.* Cambridge University Press, Cambridge, 1996.
35. E. Rosch. Natural categories. *Cognitive Psychology*, 4(3):328–350, 1973.
36. 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.
37. J. G. Snodgrass and M. Vanderwart. A standardized set of 260 pictures: Norms for name agreement, familiarity and visual complexity. *Journal of Experimental Psychology: Human Learning and Memory*, 6:174–215, 1980.

38. A. Thayananthan, B. Stenger, P. H. S. Torr, and R. Cipolla. Shape context and chamfer matching in cluttered scenes. In *Proc. Conf. Computer Vision and Pattern Recognition*, volume I, pages 127–133, Madison, USA, June 2003.
39. D'Arcy Wentworth Thompson. *On Growth and Form.* Cambridge University Press, 1917.
40. Luis von Ahn, Manuel Blum, and John Langford. Telling humans and computers apart (automatically). *CMU Tech Report CMU-CS-02-117*, February 2002.
41. G. Wahba. *Spline Models for Observational Data.* SIAM, 1990.
42. A. Yuille. Deformable templates for face recognition. *J. Cognitive Neuroscience*, 3(1):59–71, 1991.