Efficient Shape Matching Using Shape Contexts

Greg Mori, *Member*, *IEEE*,
Serge Belongie, *Member*, *IEEE*, and
Jitendra Malik, *Senior Member*, *IEEE*

Abstract—We 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.

Index Terms—Shape, object recognition, optical character recognition.

1 Introduction

We are interested in the use of shape for recognizing 3D objects, represented by a collection of multiple 2D views. A satisfactory theory of shape representation would have a number of desirable attributes:

1. It should support recognition based on exquisitely fine differences, e.g., distinguishing faces of twins.
2. At the same time, it should support making coarse discriminations very quickly. Thorpe et al. [1] showed that people, when presented with an image, can answer coarse queries such as the presence or absence of an animal in as little as 150ms.
3. The approach should scale to deal with a large number of objects. Biederman [2] has argued that humans can distinguish on the order of 30,000 different objects.
4. It should be possible to acquire a representation of an object category from relatively few examples, i.e., there should be a good generalization ability.

In this paper, we further develop an approach based on the representation of *shape contexts*, introduced in Belongie et al. [3], which arguably satisfies criteria 1 and 4 above while 3 is yet only a distant possibility. The techniques we develop here can be used for 2 and attempt to address the issues involved in 3, scaling to large numbers of objects.

The basic idea of shape contexts is illustrated in Fig. 1. A shape is represented by a discrete set of points sampled from the internal and external contours on the shape. These can be obtained as locations of edge pixels as found by an edge detector, giving us a set $\mathbb{P} = \{p_1, \dots, p_n\}$, $p_i \in \mathbb{R}^2$, of n points. Consider the set of vectors originating from a point to all other sample points on a shape. These $n-1$ vectors express the configuration of the entire shape relative to the reference point. One way to capture this information is as the *distribution* of the relative positions of the remaining $n-1$ points in a spatial histogram. Concretely, for a point p_i on the shape, compute a coarse histogram h_i of the relative coordinates of the remaining $n-1$ points,

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

This histogram is defined to be the *shape context* of p_i. We use bins that are uniform in log-polar space, making the descriptor more sensitive to positions of nearby sample points than to those of points farther away. In the absence of background clutter, the shape context of a point on a shape can be made invariant under uniform scaling of the shape as a whole. This is accomplished by normalizing all radial distances by the mean distance α between the n^2 point pairs in the shape.

As illustrated in Fig. 1, shape contexts will be different for different points on a single shape S; however, corresponding (homologous) points on similar shapes S and S' will tend to have similar shape contexts. By construction, the shape context at a given point on a shape is invariant under translation and scaling. Shape contexts are not invariant under arbitrary affine transforms, but the log-polar binning ensures that, for small locally affine distortions due to pose change, intracategory, variation, etc., the change in the shape context is correspondingly small. In addition, since the shape context descriptor gathers coarse information from the entire shape, it is relatively insensitive to the occlusion of any particular part.

In contrast to the original work [3] on shape contexts, which used the χ^2 distance to compare shape contexts, we now treat them as feature vectors and compare them using the L^2-norm. The results using the L^2-norm are comparable to those using the χ^2 distance and the L^2-norm is marginally faster to compute.

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. The present paper will introduce 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, multiscale search could be performed, or scale-invariant interest point detection or segmentation could be used to estimate scale.

The thrust of this paper is in Section 4, where we develop two different algorithms for fast pruning based on shape contexts, resulting in a short list of likely candidate shapes to be evaluated later by a more accurate and expensive procedure [3]. This is preceded by Section 2 on past work and Section 3 on the structure of our matching framework. In Section 5, we show experimental results on the ETH-80 object database [4], the Snodgrass and Vanderwart drawings [5], and the EZ-Gimpy CAPTCHA [6]. We conclude in Section 6.

2 Past Work

Past work on object recognition has developed the use of two major cues: appearance and shape. The first group of work, on appearance-based recognition, makes direct use of pixel brightness values. The work of Turk and Pentland [7] is a prime example of

- *G. Mori is with the School of Computing Science, Simon Fraser University, Burnaby, BC, Canada V5A 1S6. E-mail: mori@cs.sfu.ca.*
- *S. 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.*
- *J. Malik is with the Electrical Engineering and Computer Science Division, University of California at Berkeley, Berkeley, CA 94720. E-mail: malik@cs.berkeley.edu.*

Manuscript received 27 May 2004; revised 30 Mar. 2005; accepted 31 Mar. 2005; published online 14 Sept. 2005.
Recommended for acceptance by R. Basri.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-0267-0504.

0162-8828/05/$20.00 © 2005 IEEE Published by the IEEE Computer Society

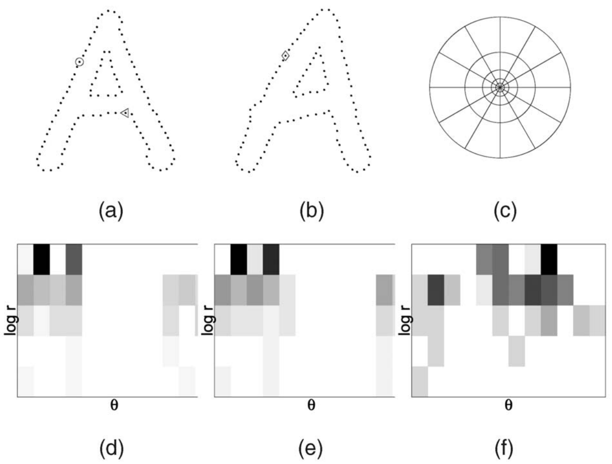

Fig. 1. Shape contexts. (a) and (b) Sampled edge points of two shapes. (c) Diagram of log-polar histogram bins used in computing the shape contexts. We use five bins for $\log r$ and 12 bins for θ. (d)-(f) Example shape contexts for reference samples marked by $\circ, \diamond, \triangleleft$ in (a) and (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.)

this genre. Several other approaches in this vein [8], [9] first attempt to find correspondences between the two images before doing the comparison. This turns out to be quite a challenge as differential optical flow techniques do not cope well with the large distortions that must be handled due to pose/illumination variations. Errors in finding correspondence will cause downstream processing errors in the recognition stage. As an alternative, there are a number of methods that build classifiers without explicitly finding correspondences. In such approaches, one relies on a learning algorithm having enough examples to acquire the appropriate invariances. These approaches have been used for handwritten digit recognition [10], [11], face recognition [12], and isolated 3D object recognition [13].

In contrast, techniques that perform recognition based on shape information attempt to capture a global structure of extracted edge or silhouette features. Silhouettes have been described (and compared) using Fourier descriptors [14], skeletons derived using Blum's medial axis transform [15], or directly matched using dynamic programming. Although silhouettes are simple and efficient to compare, they are limited as shape descriptors for general 3D objects because they ignore internal contours and are difficult to extract from real images. Other approaches [16], [17], [18] treat the shape as a set of points in the 2D image, extracted using, say, an edge detector. Another set of methods compute correspondences between edge points, such as the work of Carlsson [19], which uses *order structure*, and the work of Johnson and Hebert [20] and Chui and Rangarajan [21].

Recent years have seen the emergence of hybrid approaches [22], [23], [24], [25] that capture appearance information through a collection of local image patches. 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 [23] or histograms of image gradients [22], [24], termed SIFT descriptors (Scale Invariant Feature Transform). This line of work has been demonstrated to be effective in detecting relatively small numbers of categories. However, the problem of scaling to large databases of known objects with these methods remains open. Of the approaches mentioned above, the work by Lowe [22] has gone the furthest in addressing the issues of large data sets. The approach involves efficiently matching features by searching k-d trees with an algorithm called "Best-Bin-First" [26].

The algorithms we develop in this paper will be based on the shape context point descriptor. In particular, the *representative shape contexts* (Section 4.1) algorithm is related to the above work on local patch models. The major differences 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 representative shape contexts are placed at randomly selected edge points spread over the entire shape, as opposed to the interesting points selected in the other approaches.

Other work on efficient shape-based retrieval includes that by Sebastian et al. [27], who improved the efficiency of shock-graph shape matching using a coarse-level matching phase. Shakhnarovich et al. [28] used a variant of the "Locality Sensitive Hashing" (LSH) of Indyk and Motwani [29] to quickly retrieve human body shapes. Frome et al. [30] also used LSH to perform efficient retrieval of shapes; their work involved 3D shape information obtained from laser range scanners.

3 Matching Framework

The work by Belongie et al. [3] resulted in extremely good performance, e.g., 99.4 percent accuracy on the MNIST handwritten digit set, as well as on a variety of 3D object recognition problems. However, applying this deformable matching algorithm to a large database of models would be computationally prohibitive. To deal with this problem, we will use a two-stage approach to object recognition: fast pruning followed by detailed matching.

In the following sections, we first describe a new descriptor that is an extension of shape contexts and then develop fast pruning techniques based upon this descriptor.

3.1 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. This same structure can be applied to construct a richer descriptor based on oriented edges. In this work, 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:

$$\hat{v}_i = \langle \hat{h}_i^{1,x}, \hat{h}_i^{1,y}, \hat{h}_i^{2,x}, \hat{h}_i^{2,y}, \ldots, \hat{h}_i^{d,x}, \hat{h}_i^{d,y} \rangle$$
$$d_{GSC}(\hat{h}_i, \hat{h}_j) = ||\hat{v}_i - \hat{v}_j||_2,$$

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

We call these extended descriptors *generalized shape contexts*. Note that generalized shape contexts reduce to the original shape contexts if all tangent angles are clamped to zero. Our experiments in Section 5 will compare these new descriptors with the original shape contexts.

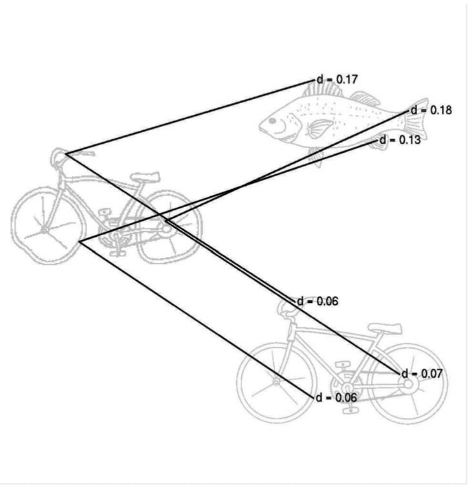

Fig. 2. Matching individual shape contexts. Three points on the query shape (left) are connected with their best matches on two known shapes. L^2 distances are given with each matching.

4 Fast Pruning Using Shape Contexts

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 short list 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 short list. We leverage the descriptive power of shape contexts toward this goal of quick pruning.

We propose 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, *shapemes*, uses vector quantization to reduce the complexity of the shape contexts from 60-dimensional histograms to quantized classes of shape pieces.

4.1 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. Fig. 2 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 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:

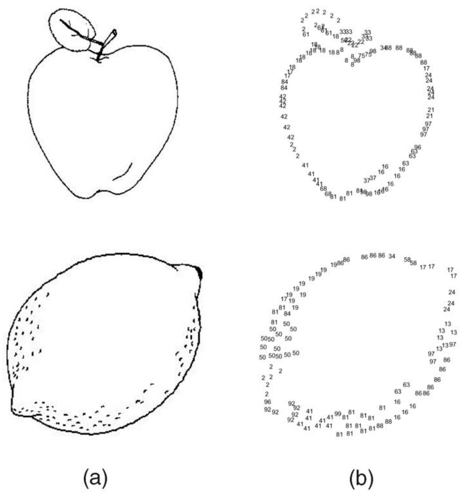

Fig. 3. (a) Line drawings. (b) Sampled points with shapeme labels. $k = 100$ shapemes were extracted from a known set of 260 shapes (26,000 generalized shape contexts). Note the similarities in shapeme labels (2, 41 on the left side, 24, 86, 97 on the right side) between similar portions of the shapes.

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

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

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

where $\mathbb{S}$ is the set of all shapes. We determine the short list by sorting these distances.

4.2 Shapemes

The second matching method uses vector quantization on the shape contexts. With $|\mathbb{S}|$ known shapes, and shape contexts computed at s sample points on these shapes, the full set of shape contexts for the known shapes consists of $|\mathbb{S}| \cdot s$ $d\text{-dimensional}$ 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. Fig. 3 shows the representation of sample points as shapeme labels.

To derive these shapemes, all of the shape contexts from the known set are considered as points in a $d\text{-dimensional}$ space. We perform $k\text{-means}$ clustering to obtain k shapemes.

We represent each known view as a collection of shapemes. Each $d\text{-bin}$ shape context is quantized to its nearest shapeme and replaced by the shapeme label (an integer in $\{1, \ldots, k\}$). A known view is then simplified into a histogram of shapeme frequencies. No spatial information among 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 the query shape. We then find nearest neighbors in the space of histograms of shapemes.

5 Results

We use the ETH-80 Object Database, the Snodgrass and Vanderwart line drawings, and the EZ-Gimpy CAPTCHA as

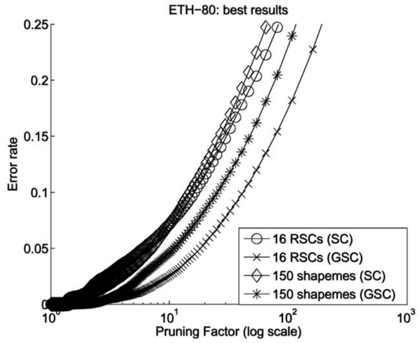

Fig. 4. Error rate versus pruning factor on ETH-80 data set, averaged across 10 runs. Comparison of results for best parameter settings for each of the four methods is shown. Error bars are omitted for clarity, but the standard deviation is small. The maximum standard deviation over all runs is 0.54 percent.

our test sets. In the following sections, we present graphs showing the performance of the two methods on these test sets. The graphs plot error rate versus 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 short list 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 short list has 10 shapes in it.

In general, the representative shape contexts method performs better at large pruning factors—particularly when dealing with occlusion. Missing a couple of shape contexts won't spoil the matching. However, the vector quantization used in shapemes does buy us computational speed and using all of the shape contexts in this manner allows low error rates to be obtained.

5.1 ETH-80

The first experiment involves the ETH-80 database [4]. The database consists of 80 unique objects, each from one of eight classes. Each object is represented by 41 views spaced evenly over the upper viewing hemisphere. We prepared a set of known shapes by selecting one object from each of the eight classes and using all of its views in the training set, a total of 328 images. The images of the remaining nine objects from each class were used as a test set. This experiment was repeated 10 times, each time selecting a different set of training objects. We use an edge detector [31] to extract line features from the images. These edges are then sampled to create point features for use in shape contexts.

We ran experiments using the two pruning methods. Representative shape contexts pruning was done using 4, 8, 12, and 16 shape contexts and generalized shape contexts. In each of these experiments, the best $\frac{3}{4}$ of the RSCs (i.e., 3, 6, 9, and 12) were used to compute the matching cost. Shapeme pruning was performed with quantization to 25, 50, 75, 100, 125, and 150 shapemes, again using both types of shape contexts. Results are presented in Fig. 4.

Both of the pruning methods are successful: For example, a pruning factor of approximately 40 (short list of length 8) can be obtained with an error rate of 10 percent for the representative shape contexts method (16 RSCs using generalized shape contexts) and 14 percent for the shapeme method (150 shapemes using generalized shape contexts).

Fig. 5 shows some short lists on the ETH-80 data set using the representative shape contexts pruning method. Many of the errors on this data set involve objects that have the same coarse shape. For example, the shape matching process deems the tomatoes and apples to be very similar. Relying solely on coarse shape, without cues such as color and texture, it is difficult to differentiate between the members of these groups of objects.

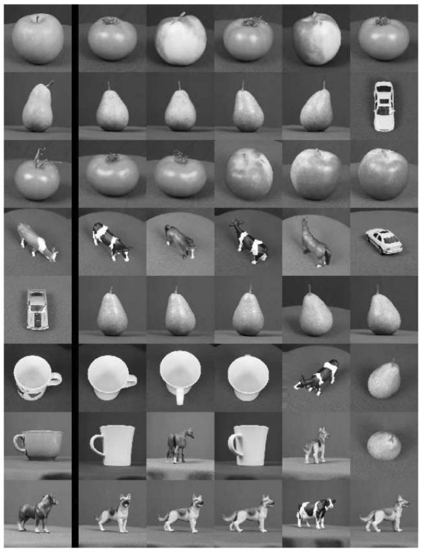

Fig. 5. Short lists for the ETH-80 data set using the representative shape contexts method. The first column is the query object. The remaining five columns show closest matches to each query object.

5.2 Snodgrass and Vanderwart

The second experiment uses the Snodgrass and Vanderwart line drawings [5]. This data set contains line drawings of 260 commonly occurring objects. They are a standard set of objects that have been frequently used in the psychophysics community for tests with human subjects. Since the images are line drawings, no preprocessing phase of edge extraction is needed. We sample points from the line drawings directly and use elongated oriented filters to estimate local tangent directions.

The Snodgrass and Vanderwart data set has only one image per object. We use these original images as the known set and create a synthetic distorted set of images for querying. The thin plate spline (TPS) model [32] is used to create these distortions. In a 2D view of a class of 3D objects, there are two sources of variation: pose change and intraclass change. We use the nonlinear TPS model to simulate both of these types of variation simultaneously. We apply a random TPS warp of fixed bending energy to a reference grid and use this warp to transform the edge points of a line drawing.

In addition to distortions, we test the ability of our pruning methods to handle occlusion. We take the set of TPS-distorted objects and subject them to random occlusions. The occlusions are generated using a linear occluding contour. The query objects in Fig. 6 show some distorted and occluded Snodgrassand Vanderwart images. Note that the occluding contour is included—we will sample points from it when creating the shape contexts.

The 260 original Snodgrass and Vanderwart images were used as the training set. We generated 5,200 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. The same set of test parameters as in the experiments on the ETH-80 data set was used. Figs. 6 and 7 show the results for our two pruning methods.

In the low occlusion setting ($\leq$ 10 percent occlusion), the shapeme method can achieve a pruning factor of ≈ 100 (short list of length three out of 260 images) with an error rate of 10 percent (150 shapemes, original shape contexts), while the representative shape contexts method has an error rate of 4 percent (16 RSCs, original shape contexts).

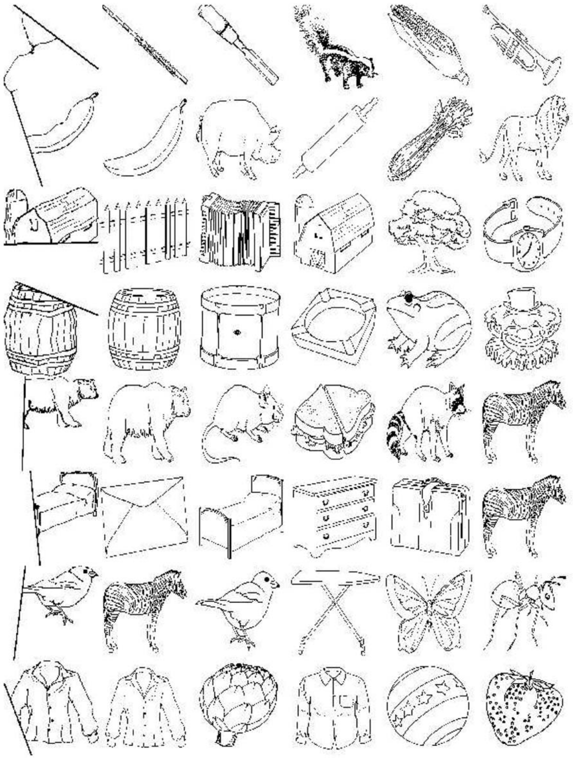

Fig. 6. Short lists for the distorted and occluded Snodgrass and Vanderwart data set using the representative shape contexts method. The first column is the query object. The remaining five columns show closest matches to each query object.

With extremely difficult levels of occlusion (20 percent to 30 percent and 30 percent to 40 percent), RSCs can obtain large amounts of pruning with reasonable error rates, while shapemes are able to operate at low error rates with moderate pruning as they efficiently use all shape contexts on a query shape.

Note that, on this data set, 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.

5.3 EZ-Gimpy

A CAPTCHA is a program [6] 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." Blum's group has designed a number of different CAPTCHAs. EZ-Gimpy (Fig. 8) 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, that has been distorted and placed in a cluttered image.

The CAPTCHA data sets provide more than just a colorful toy problem to work on. They present challenging clutter since they are intended to be difficult for computer programs. More importantly, these data sets are large. There are 561 words that need to be recognized in EZ-Gimpy. Also, since the source code for generating these CAPTCHAs is available ("P" for *public*), we have access to a practically infinite set of test images. This is in contrast to many object recognition data sets in which the number of objects is limited and it is difficult to generate many reasonable test images. However, there are definitely limitations to this data set in terms of studying general object recognition. Most notably, these are 2D objects and there is no variation due to 3D pose. In addition, there are no shading and lighting effects in synthetic images of words.

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 (using the method of Belongie et al. [3]) to recognize the word in each EZ-Gimpy image. This algorithm is referred to as "Algorithm B" in our previous work on breaking CAPTCHAs [33]. Two details are different from those in the first two experiments. First, we constructed generalized shape contexts that are tuned to the shape of words: They are elliptical, with an outer radius of about four characters horizontally and $\frac{3}{4}$ of a character vertically. Second, the texture gradient operator [31] 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, nine 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. 8, the full set of test

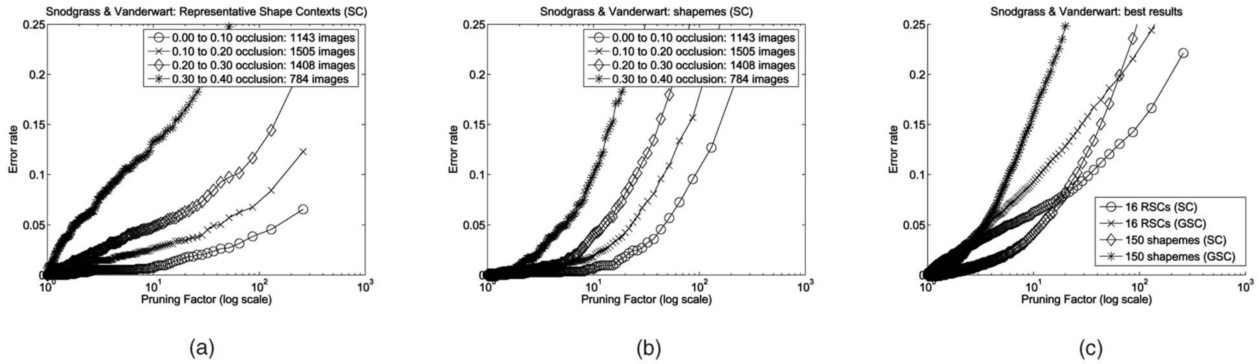

Fig. 7. Error rate versus pruning factor on the Snodgrass and Vanderwart data set. (a) and (b) Variation in performance with respect to the amount of occlusion in the test image. (c) Comparative results for all different methods. Results for the best parameter settings from each method are shown.

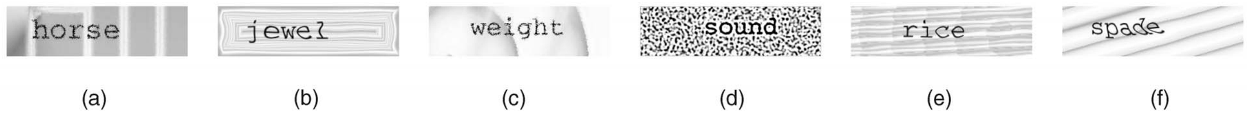

Fig. 8. Results on EZ-Gimpy images. The best matching words are as follows: (a) horse, (b) jewel, (c) weight, (d) sound, (e) rice, and (f) space.

images and results can be viewed at http://www.cs.sfu.ca/~mori/research/gimpy/ez/. In 92 percent (176/191) of these test cases, our method identified the correct word. This success rate compares favorably with that of Thayananthan et al. [34] who perform an exhaustive search using Chamfer matching with distorted prototype words.

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

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 percent success rate on the test set.

6 Conclusion

Previous work on shape matching via a deformable template-based framework has been very successful for object recognition. However, these methods are too expensive computationally to be used on a large scale object database. We have shown how a shape context-based pruning approach can assist by constructing an accurate short list in order to reduce this computational expense. We proposed two methods of matching—one using a small number of representative shape contexts and the other based on vector quantization of shape contexts into shapemes.

We also presented generalized shape contexts (GSCs), an extension to shape contexts which makes use of local tangent information at point locations. These descriptors contain more detailed information about the shape and, when the local tangent can be reliably estimated, they outperform the original shape contexts.

The GSC is similar to Lowe's SIFT descriptor, which also aggregates edge orientations into a histogram. However, the spatial structure of the histogram bins of GSCs is very different from that of SIFT features. GSCs are large in scale, while SIFT features are local descriptors. SIFT features use a regular grid for histogram bins and disregard information far away from the center of the descriptor, using a Gaussian weighting to discount sample points. In contrast, the outermost bins of GSCs are largest in size, reflecting positional uncertainty of useful coarse shape cues.

We demonstrated the effectiveness of representative shape contexts and shapemes, two efficient pruning mechanisms based on shape contexts and GSCs, in experiments on the ETH-80, Snodgrass and Vanderwart, and EZ-Gimpy data sets.

Acknowledgments

Portions of this paper appeared in previous publications [35], [33]. Greg Mori was supported by grants from NSERC (RGPIN-312230) and the SFU President's Research Fund. Serge Belongie was supported by US National Science Foundation CAREER #0448615, The Alfred P. Sloan Research Fellowship, and through subcontracts B542001 and B547328 under the auspices of the US Department of Energy by the Lawrence Livermore National Laboratory under contract No. W-7405-ENG-48. The authors thank the anonymous reviewers for their useful suggestions.

References

[1] S. Thorpe, D. Fize, and C. Marlot, "Speed of Processing in the Human Visual System," *Nature,* vol. 381, pp. 520-522, 1996.

[2] I. Biederman, "Recognition-by-Components: A Theory of Human Image Understanding," *Psychological Rev.,* vol. 94, no. 2, pp. 115-147, 1987.

[3] S. Belongie, J. Malik, and J. Puzicha, "Shape Matching and Object Recognition Using Shape Contexts," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 24, no. 4, pp. 509-522, Apr. 2002.

[4] B. Leibe and B. Schiele, "Analyzing Appearance and Contour Based Methods for Object Categorization," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* vol. 2, pp. 409-415, 2003.

[5] J.G. Snodgrass and M. Vanderwart, "A Standardized Set of 260 Pictures: Norms for Name Agreement, Familiarity and Visual Complexity," *J. Experimental Psychology: Human Learning and Memory,* vol. 6, pp. 174-215, 1980.

[6] L. von Ahn, M. Blum, and J. Langford, "Telling Humans and Computers Apart (Automatically)," CMU Technical Report CMU-CS-02-117, Feb. 2002.

[7] M. Turk and A. Pentland, "Eigenfaces for Recognition," *J. Cognitive Neuroscience,* vol. 3, no. 1, pp. 71-96, 1991.

[8] 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,* vol. 42, no. 3, pp. 300-311, Mar. 1993.

[9] T. Cootes, D. Cooper, C. Taylor, and J. Graham, "Active Shape Models—Their Training and Application," *Computer Vision and Image Understanding,* vol. 61, no. 1, pp. 38-59, Jan. 1995.

[10] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-Based Learning Applied to Document Recognition," *Proc. IEEE,* vol. 86, no. 11, pp. 2278-2324, Nov. 1998.

[11] C. Burges and B. Schölkopf, "Improving the Accuracy and Speed of Support Vector Machines," *Advances in Neural Information Processing Systems,* pp. 375-381, 1997.

[12] B. Moghaddam, T. Jebara, and A. Pentland, "Bayesian Face Recognition," *Pattern Recognition,* vol. 33, no. 11, pp. 1771-1782, Nov. 2000.

[13] H. Murase and S. Nayar, "Visual Learning and Recognition of 3-D Objects from Appearance," *Int'l J. Computer Vision,* vol. 14, no. 1, pp. 5-24, Jan. 1995.

[14] C. Zahn and R. Roskies, "Fourier Descriptors for Plane Closed Curves," *IEEE Trans. Computers,* vol. 21, no. 3, pp. 269-281, Mar. 1972.

[15] D. Sharvit, J. Chan, H. Tek, and B. Kimia, "Symmetry-Based Indexing of Image Databases," *J. Visual Comm. and Image Representation,* June 1998.

[16] G. Borgefors, "Hierarchical Chamfer Matching: A Parametric Edge Matching Algorithm," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 10, no. 6, pp. 849-865, 1988.

[17] D. Huttenlocher, R. Lilien, and C. Olson, "View-Based Recognition Using an Eigenspace Approximation to the Hausdorff Measure," *Trans. Pattern Analysis and Machine Intelligence,* vol. 21, no. 9, pp. 951-955, Sept. 1999.

[18] D. Gavrila and V. Philomin, "Real-Time Object Detection for Smart Vehicles," *Proc. Seventh Int'l Conf. Computer Vision,* pp. 87-93, 1999.

[19] S. Carlsson, "Order Structure, Correspondence and Shape Based Categories," *Shape Contour and Grouping in Computer Vision,* pp. 58-71, 1999.

[20] A.E. Johnson and M. Hebert, "Recognizing Objects by Matching Oriented Points," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 684-689, 1997.

[21] H. Chui and A. Rangarajan, "A New Algorithm for Non-Rigid Point Matching," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* vol. 2, pp. 44-51, June 2000.

[22] D.G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *Int'l J. Computer Vision,* vol. 60, no. 2, pp. 91-110, 2004.

[23] R. Fergus, P. Perona, and A. Zisserman, "Object Class Recognition by Unsupervised Scale-Invariant Learning," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* vol. 2, pp. 264-271, 2003.

[24] G. Dorko and C. Schmid, "Selection of Scale Invariant Neighborhoods for Object Class Recognition," *Proc. Ninth Int'l Conf. Computer Vision,* pp. 634-640, 2003.

[25] Y. Amit, D. Geman, and K. Wilder, "Joint Induction of Shape Features and Tree Classifiers," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 19, no. 11, pp. 1300-1305, Nov. 1997.

[26] J. Beis and D. Lowe, "Shape Indexing Using Approximate Nearest-Neighbour Search in Highdimensional Spaces," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 1000-1006, 1997.

[27] T. Sebastian, P.N. Klein, and B.B. Kimia, "Shock-Based Indexing into Large Shape Databases," *Proc. European Conf. Computer Vision,* vol. 3, pp. 731-746, 2002.

[28] G. Shakhnarovich, P. Viola, and T. Darrell, "Fast Pose Estimation with Parameter Sensitive Hashing," *Proc. Ninth Int'l Conf. Computer Vision,* vol. 2, pp. 750-757, 2003.

[29] P. Indyk and R. Motwani, "Approximate Nearest Neighbors: Towards Removing the Curse of Dimensionality," *Proc. ACM Symp. Theory of Computing,* pp. 604-613, 1998.

[30] A. Frome, D. Huber, R. Kolluri, T. Bulow, and J. Malik, "Recognizing Objects in Range Data Using Regional Point Descriptors," *Proc. Eighth European Conf. Computer Vision,* vol. 3, pp. 224-237, 2004.

[31] D. Martin, C. Fowlkes, and J. Malik, "Learning to Find Brightness and Texture Boundaries in Natural Images," *Advances in Neural Information Processing Systems,* 2002.

[32] F.L. Bookstein, "Principal Warps: Thin-Plate Splines and Decomposition of Deformations," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 11, no. 6, pp. 567-585, June 1989.

[33] G. Mori and J. Malik, "Recognizing Objects in Adversarial Clutter: Breaking a Visual CAPTCHA," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* vol. 1, pp. 134-141, 2003.

[34] A. Thayananthan, B. Stenger, P.H.S. Torr, and R. Cipolla, "Shape Context and Chamfer Matching in Cluttered Scenes," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* vol. 1, pp. 127-133, June 2003.

[35] G. Mori, S. Belongie, and J. Malik, "Shape Contexts Enable Efficient Retrieval of Similar Shapes," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* vol. 1, pp. 723-730, 2001.