Robust 2D Shape Correspondence using Geodesic Shape Context

Varun Jain Richard (Hao) Zhang
GrUVi Lab, School of Computing Science
Simon Fraser University, Burnaby, BC, Canada
E-mail: vjain, haoz@cs.sfu.ca

Abstract

A meaningful correspondence and similarity measure between shapes is particularly useful in applications such as morphing, object recognition, shape registration and retrieval. In this paper, we present a robust shape descriptor for points along a 2D contour, based on the curvature distribution collected over bins arranged geodesically along the contour. Convolution, binning and hysteresis thresholding of curvatures are applied to render the descriptor more robust against noise and non-rigid shape deformation. Once the shape descriptor is computed for every point or feature vertex of the two shapes to be matched, a one-to-one correspondence can be quickly established through best matching of the descriptors, aided by a proximity heuristic.

Our approach does not rely on the linear ordering of the points along a contour, facilitating its 3D generalization. It is also capable of matching all the points along the contour, not just a specified set of feature vertices. Our shape descriptor is intuitive, fast to compute, shape distinguishing, and easy to implement. The performance of our approach, when applied to shape correspondence and shape retrieval on the Brown database and the articulated shapes database of Ling et al., shows that it is robust against both rigid and common non-rigid transformations such as bending and moderate stretching.

Keywords: *Geodesic shape context, shape correspondence, shape descriptors, object recognition*

1. Introduction

The problem of finding a point-to-point correspondence between two shapes has numerous applications in computer vision, graphics, and medical image analysis. Finding a meaningful correspondence is usually the first step for shape morphing, which is particularly useful in animation and entertainment applications. Also, it is typically the initial step for shape retrieval. Shape searching and retrieval in 2D has been well studied, but with the recent advances in 3D scanning and acquisition technology, the need for a reliable 3D shape retrieval system has also increased [13].

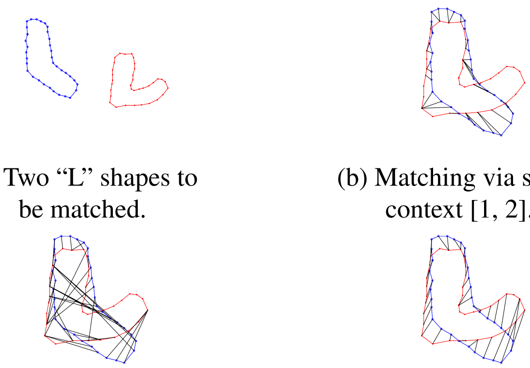
An effective way to find a correspondence between the points of two shapes is to describe every point in a higher dimensional feature space. This description encodes the shape information from the perspective of that particular point. A correspondence is then obtained by matching the descriptors of individual points. Such an inclusion of *shape context* usually yields better correspondence as compared to matching points based solely on their absolute coordinates.

The goal of our work is to develop a descriptor based on shape context for 2D contours which is robust against shape noise and non-rigid deformations and is more meaningful (with respect to human perception) than existing approaches. Ultimately, we would like to extend this work to 3D shapes, thus, while designing the 2D descriptor we take special care so that it is more natural to generalize to 3D.

Our shape descriptor computes the geodesic shape context at a point. We define the shape descriptor by recording average thresholded curvatures over appropriately sized geodesic bins about a point. As we will show, this technique is robust in establishing correct correspondences under noise and small shape variations. We further refine our descriptor by appropriately thresholding the curvatures and using a heuristic based on pairwise proximity to make the descriptor more invariant to non-rigid shape deformations such as bending and moderate stretching.

1.1. Previous Work

Several techniques have been proposed as robust shape descriptors. Among the earlier ones are those based on statistical moments, Fourier descriptors, Hausdorff distance and medial axis transforms. These techniques represent the shape by its global properties and are unsuitable to use for applications, such as shape morphing, where a point-to-point correspondence is desired. A detailed survey of shape correspondence techniques can be found in [14]. Among



(a) Two "L" shapes to be matched.

(b) Matching via shape context [1, 2].

(c) Matching via 2D curvature map.

(d) Matching via geodesic shape context.

Figure 1. Matching of two hand-drawn "L" shapes. (a) Original shapes; (b) matching using 2D shape contexts [1, 2]; (c) matching using 2D curvature map; (d) matching using our descriptor. Note that we use a simple best matching strategy to find the final matching in all cases, as we wish to compare the qualities of the descriptors only.

some recent work are spin image descriptor of [7, 8] and shape contexts of [1, 2]. The basic idea for both approaches is that the descriptor for a point encodes the relative position of all the other points on the shape. In other words, a point is parameterized by its distance to all the other points on the shape. They rely on the fact that two similar points will have similar pair-wise point-to-point distances. However, they are not invariant to non-rigid deformations. For example, the shapes in Figure 1 are similar as they differ only by a bending of the lower part. But the bending transformation disturbs the pair-wise distance configuration between the points and hence the shape context or spin image representation for similar points would vary greatly.

Belongie et al [1, 2] also suggest an iterative solution to improve the robustness of shape contexts. They first find a point-to-point correspondence between two shapes and then transform one shape into another using thin plate splines. Then they iterate between these two steps. However, such iterative solutions are known to converge to local minima if the initial correspondence is not reasonable. Also, designing a transformation model that incorporates non-rigid shape deformations is usually difficult.

In addition, shape context of Belongie et al [1, 2] is also not invariant to rotations. To fix this problem, they suggest that the shape context for a point be computed with respect to the orientation of the tangent of the contour at that point. This requires a robust estimation of tangents and as we noticed in our experiments, the matching obtained with shape contexts can be problematic even with slight variations in tangent estimation.

Recently, Gatzke et al [4] proposed *curvature map*, a shape descriptor based on sampling a number of points on the shape over the geodesic neighborhood of a point and recording the curvature of the shape at these samples. This type of shape descriptor is again not invariant to non-rigid shape deformations and as suggested by our experiments, recording curvature over sample points is non-robust to noise and moderate bending in a shape. Note that Gatzke et al. only describe their descriptor for 3D shape matching. We have implemented and analyzed a 2D version of it.

Liu et al [10] propose a shape descriptor using the spectral properties of a point's immediate neighborhood. The local nature of the descriptor achieves better robustness against non-rigid deformations but the descriptor becomes less descriptive. Therefore, they rely on the second step of their matching algorithm, which computes the matching by formulating and solving a constrained dynamic programming problem, to compensate for any inadequacy in their shape descriptor.

Indeed, this type of formulation heavily relies on the fact that the given points are ordered along the contour. Hence, it is not easily applicable to 3D shape, as there is no obvious ordering of points over a surface. In fact, having the constraint of order preservation while finding the matching greatly simplifies the problem. Scott et al [11] present an algorithm for solving an "order preserving assignment problem". Using this algorithm in our experiments, we noticed that it gives excellent matchings for all the shape descriptors mentioned so far. This shows that considering the points to be ordered while finding the matching places less emphasis on the quality of the descriptor.

Order-preserving matching typically involves expensive optimizations, as in Liu et al. [10] and Scott et al. [11]. Since this is often far too expensive for large point sets, these approaches can only be applied to a small number of feature points along the shape; this would then require robust feature detection.

There are other ways of finding a correspondence without using shape descriptors. Sederberg et al [12] and Zhang [15] present two similar physically based approaches for correspondence of 2D polygons which fit one polygon over another by optimizing certain energy functional and then find a correspondence. Such global optimization is rather expensive. Hence, the method cannot be used for applications such as shape searching, where a shape has to be matched with many other shapes.

1.2. Paper Outline

In Section 2 we give an overview of our shape descriptor, followed by its detailed description and the explaination of various techniques used for making it robust to non-rigid shape deformations in Section 3. Section 4 contains experimental results, and in Section 5, we quote some concluding remarks and future prospects.

2. Overview

While designing a shape descriptor for 2D contours, we had the following goals:

1. The descriptor should be invariant to ordinary geometric transformations such as rotation, translation and uniform scaling.
2. It should be robust against shape noise.
3. It should be robust to non-rigid deformations; especially shape bending.
4. It should be scalable to 3D shapes.
5. It should capture the shape information sufficiently, so that the correspondence obtained from a (fast) best matching procedure is adequate. In this way, expensive bipartite-matching (as in Belongie et al's shape contexts, [1, 2]) or dynamic programming formulations (as in the method of Liu et al [10]) for matching the resulting shape descriptors, can be avoided.

The inspiration for our work is the shape context of Belongie et al [1, 2]. The 2D shape context for a point is constructed by first dividing the plane into concentric circular bins and then collecting the number of points in each bin as the shape descriptor. The reader is referred to [1] and [2] for details regarding shape contexts. We have noticed in our experiments that shape contexts are as such quite robust. However, they fail when confronted with shapes having non-rigid deformations, such as bending. As we are aiming for robustness to bending, it is intuitive to ask the question: what property of a shape remains unchanged when sub-parts of the shape are bent. It is easy to see that the geodesic distance (distance along the contour) between two points on the shape remains constant when shapes undergo bending transformations. Hence, a natural generalization of shape contexts to include bending transformations would be to construct geodesic bins instead of bins according to Euclidean distances.

We perform a geodesic traversal along the shape and collect geometric information, hence the name *geodesic shape contexts*. The information that we collect is the average curvature inside these geodesically formed bins. We refer to this curvature information as the basic geodesic shape context for a point. We then refine this basic descriptor using appropriate techniques to achieve robustness against non-rigid deformation.

3. Basic geodesic shape context

Given, two contours S_1 and S_2, we first construct the basic geodesic shape context for every point on both contours. For a point P on a contour, the geodesic shape context is constructed as follows:

1. Divide the geodesic neighborhood of P into bins of non-decreasing lengths ending at point Q furthest away from P along the contour, as shown in Figure 2.
2. For each bin, find the average of the curvatures of all the points lying inside the bin.
3. If C_i is the average curvature of the i^{th} bin, then the basic geodesic shape context of the point is defined as the vector $\{C_1, C_2, C_3, \ldots C_n\}$, where n is the number of bins.

Instead of increasing the bin size strictly exponentially, as in the original shape context, we repeat bins of equal size to capture local information better, e.g., in Figure 2.

If $\{P_1, P_2, P_3, \ldots\}$ are points along the contour, we calculate the curvature at a given point P_i by fitting a cubic parametric curve $(x(t), y(t))$ to the set of points $(P_{i-w}, P_{i-w+1}, \ldots P_i, \ldots P_{i+w-1}, P_{i+w})$ for some neighborhood size w. Typically, we choose $3 \leq w \leq 5$. The curvature at P_i is then given by:

$$C_{P_i} = \frac{||X'(t) \times X''(t)||}{||X'(t)||^3}, \text{ where } X(t) = \begin{bmatrix} x(t) \\ y(t) \end{bmatrix}.$$

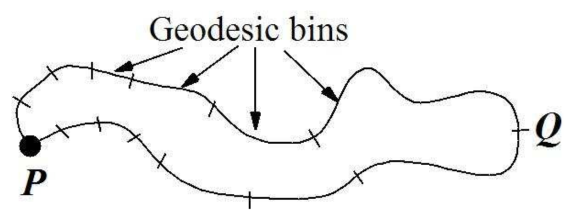


Figure 2. The basic geodesic shape context for point P is the vector $\{C_1, C_2, C_3, \ldots C_n\}$ where n is the number of bins and C_i is the average curvature of the points in the i^{th} bin.

Once the descriptor is formed for every point on the two shapes, we can directly compare a point P on S_1 to a point Q on S_2 with the Euclidean metric:

$$DisSim(P, Q) = \sum_{i=1}^{n} ||C_P(i) - C_Q(i)||^2,$$

where, $C_P(i)$ and $C_Q(i)$ are the average curvatures of the i^{th} bin for P and Q, respectively.

A match for point P from all points in S_2 is the point which has the minimum dissimilarity with P. In other words:

$$match(P) = \underset{Q \in S_2}{\operatorname{argmin}}(DisSim(P, Q)).$$

The averaging scheme we use, in contrast to curvature map [4], makes the descriptor robust against surface noise and irregularities. This can be explained from a signal processing perspective. Consider the curvature plot for a contour as a signal. Curvature map samples this signal at regular intervals. It is easy to see that if the samples are not selected carefully, it is quite probable that some features of the signal are not sampled. Taking averages, on the other hand, is equivalent to first convolving this signal with a box filter and then recording samples at regular intervals on the convolved signal. A sample on the convolved signal contains information about its neighborhood which makes the sampling more feature sensitive. At the same time, box filtering also achieves noise removal.

3.1. Hard and hysteresis thresholding

Although the averaging scheme renders our descriptor robust against noise and small shape deformations, it doesnot handle more drastic bending in the shape well, as the curvature of corner vertices can drastically change with bending. To handle this, we propose thresholding of curvatures. Specifically, we give more weight to negative curvatures, corresponding to concavity in a shape, and attenuate the positive curvatures.

The intuition behind this is based on human perception. It is generally believed that humans match shapes based on the similarity of their parts. We simulate this by placing more emphasis on concavities following the "Minima Rule" [5], which stipulates that part boundaries are found at negative curvature minima. This can be accomplished by the following simple thresholding scheme: if C is the curvature at a point, the thresholded curvature is given by:

$$\hat{C} = \begin{cases} C/C', & \text{if } C \geq 0 \\ -(1 - e^{C}), & \text{if } C < 0, \end{cases} \tag{1}$$

where, C' is some large constant we select.

However, our experiments suggested that such a hard thresholding can be rather sensitive to slight variations in the shape. We therefore suggest to use a *hysteresis* thresholding scheme instead, which is commonly used for edge detection in image processing [6]. For all the examples given in this paper, we have used the following thresholding scheme: if C_i is the curvature at point P_i, $i = 1, \ldots, n$, the thresholded curvature is given by:

$$\hat{C}_i = \begin{cases} C_i/C', & \text{if } C_i \geq 0 \\ -(2 - e^{C_i}), & \text{if } C_i < \tau. \\ -(2 - e^{C_i}), & \text{if } \tau < C_i < 0 \text{ and } C_{i-1}, C_{i+1} < 0 \\ C_i, & \text{otherwise,} \end{cases} \tag{2}$$

where, $\tau < 0$ is the threshold for negetive curvature.

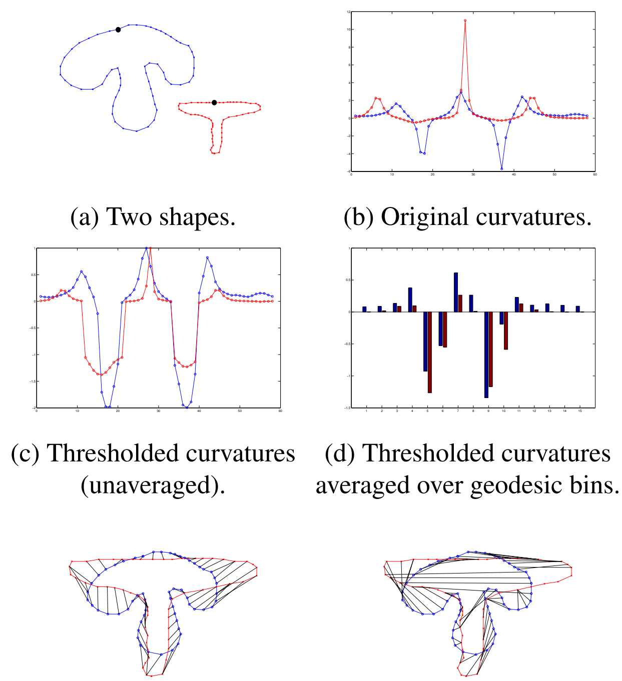
As before, we attenuate all the positive curvature values. For negative curvatures, in the hard thresholding scheme before, we were amplifiying any curvature that is less than zero, which can be sensitive to minor shape variations and noise. In the new scheme, we amplify any negative curvature which is less than the threshold τ. These are the curvatures that we consider as shape features. But for negative curvatures greater than the threshold τ, we only amplify them if we find that they indeed define a feature. We decide whether a negative curvature represents noise or feature by looking at the curvature of its neighbors where the neighborhood size can be increased when appropriate. This soft thresholding generally works well since it uses selective amplification and hence, does not amplify noise. We found from our experiments that setting C' to be the maximum curvature along the contour and τ to be a quater of the average of all the negative curvatures along the contour generally works well for our correspondence tasks.

To further explain the need for thresholding, consider the shapes given in Figure 3. It is evident that the red contour is similar to the blue except that the top portion is bent downwards. Also shown are the plots of the original curvature and their thresholded values. As is clear from the curvature plots, the concavities on both the shapes match well with the thresholded curvatures than with the original curvatures. The latter can be thought of as containing detailed shape information which is non-robust to various shape transformations. This information is conveniently filtered out by our thresholding.

3.2. Guiding heuristic using proximity

Due to the global nature of our descriptor, it can be hard to distinguish between nearby points on a shape and points that have similar shape descriptors. To guide the matching process so as to obtain a consistent (with respect to the linear ordering of the contours) matching we propose a proximity heuristic. Note that this is independent of the linear ordering of the contour points and hence is extendable to 3D. As our shape descriptor is robust and descriptive, using this simple proximity heuristic already leads to excellent results. We now describe the proximity heuristic.

The proximity guidence scheme relies on one or more "anchor" points, chosen on both shapes, that are well matched by similarity between point shape descriptors



(a) Two shapes. (b) Original curvatures.

(c) Thresholded curvatures (unaveraged). (d) Thresholded curvatures averaged over geodesic bins.

(e) Matching using geodesic shape contexts. (f) Matching using 2D curvature map.

Figure 3. (a) Two hand drawn mushroom shapes. (b) Curvature plots starting from the black point along the contour in counter-clockwise direction. (c) Thresholded curvatures. (d) Thresholded curvatures averaged over geodesic bins. (e) Matching obtained from our shape descriptor. (f) Matching obtained from mapping of unthresholded curvatures.

alone. The matching of any remaining points is based on both descriptor similarity and the point's proximity to the anchor points. In our experiments, we select only two anchor points per shape. The first point is chosen to be the best matched pair, min_i and min_j, on the two shapes using only descriptor similarity. To choose the second point, we add a penalty for choosing a point near the first point. This requirement is aimed at improving the accuracy of the heuristic, since, if the anchor points are selected close to each other, their impact on nearby points would be much less than that on geodesically far points.

Let A_S denote the dissimilarity matrix where the ij^{th} entry is the dissimilarity between the i^{th} point in S_1 and j^{th} point in S_2. That is,

$$A_S(i,j) = DisSim(i,j).$$

Let $A_S(min_i^1, min_j^1)$ be the minimum entry of A_S, where min_i^1 and min_j^1 are points of S_1 and S_2 respectively. min_i^1 and min_j^1 are the first anchor points for the two shapes. We then find the geodesic distance of all points in S_1 from point min_i^1 and the geodesic distance of all points of S_2 from the point min_j^1. Let $dist_i^1(P)$ be the geodesic distance of $P \in S_1$ from min_i^1, and $dist_j^1(Q)$ be the geodesic distance of $Q \in S_2$ from min_j^1. We define a distance-dissimilarity matrix A_{SD}^1 as follows:

$$A_{SD}^1(P,Q) = ||dist_i^1(P) - dist_j^1(Q)||^2.$$

Now we construct a new dissimilarity matrix for choosing the second anchor. As we would like to prevent the second anchor from being too close to the first one, the new dissimilarity matrix is defined as:

$$A_S^{new}(P,Q) = A_S(P,Q) - \frac{1}{2}[dist_i^1(P) + dist_j^1(Q)].$$

The second term on the right hand side is the penalty so that the next best match found is generally not close to the first anchor. Let $A_S^{new}(min_i^2, min_j^2)$ be the minimum entry of A_S^{new}. min_i^2 and min_j^2 are the second anchor points for both shapes. If $dist_i^2(P)$ and $dist_j^2(Q)$ are the goedesic distances of points P and Q from min_i^2 and min_j^2 respectively, we define the second distance-dissimilarity matrix as:

$$A_{SD}^2(P,Q) = ||dist_i^2(P) - dist_j^2(Q)||^2.$$

Finally, we redefine the dissimilarity between two points P and Q as:

$$\begin{aligned} DisSim(P,Q) &= A_S(P,Q) + \\ & c_1\ A_{SD}^1(P,Q) + \\ & c_2\ A_{SD}^2(P,Q), \end{aligned}$$

where c_1 and c_2 are free parameters which can be set by the user.

This heuristic can be extended to fixing more matching pairs. However, it should be noted that fixing more matching pairs will make the process non-robust against stretching since this heuristic depends completely on geodesic distances which change drastically with streching. Hence, we restrict ourselves to using only two anchor points.

4. Results

For the purpose of evaluating our shape descriptor, we report its performance when applied to shape retrieval on two shape databases, and shape correspondence of various contours. For shape retrieval experiments, we have used the Brown database [3] of 95 binary images and Ling et al's [9] articulated shapes database. The lattar consists of 40 images from 8 different objects. Each object has 5 images articulated to different degrees. Both the databases are illustrated in Figure 7. Note that for all our experiments we first extract the contour of the image and then apply our method on this contour. Also, for shapes in Ling et al's [9] database, only the outer contour was extracted.

Figure 4 shows some results obtained from our shape searching experiments. The first column in the figure shows the shape that was used as a query to the database. The next five columns show the top five matches returned from our searching algorithm. We use our shape descriptor to find dissimilarity between two shapes as follows: the dissimilarity between two shapes is defined as $\sum_{P,Q} DisSim(P, Q)$ where, P and Q are matching points found by our correspondence algorithm.

Figure 5(a) and 5(b) show an image representation of the dissimilarity matrices between the images of Brown database and Ling et al.'s database respectively. The ij^{th} entry of the matrix is the dissimilarity between image i and image j. A dark spot represents less dissimilarity and a bright spot represents greater dissimilarity. The images were ordered so that similar images appear together in the matrix. The block diagonal structure of both the matrices show that the dissimilarity, obtained using our shape description, between similar images is low.

Next we evaluate our descriptor when applied to shape correspondence. Given two contours, we use our shape descriptor to find a point-to-point matching. Matching obtained for six different pairs of contours is shown in Figure 6. We show matching points by drawing a black line between the two. The first column shows the shapes to be matched. The second and the third columns show the matching obtained from shape contexts [1, 2] and 2D curvature maps respectively. The fourth column shows matching obtained from the our descriptor.

It can be easily seen from these experiments that the shape context [1, 2] descriptor fails to retrieve proper matching when the two shapes differ by some level of bending. For example, the arms and legs of the human shapes in 6(c) and 6(d) are poorly matched although the torso is matched fairly well. Note that our descriptor is robust against such shape variations, as well as moderate stretching deformations (e.g., Figure 6(b)). Figure 6(e) and 6(f) are contours of images taken from Ling et al's database which have a great deal of articulation. However, our shape descriptor still performs a good job of retrieving the right correspondence. These examples demonstrate that our shape descriptor contains more shape information than shape contexts.

For all the examples shown in this paper, after computing the shape descriptor for every point, the point-to-point matching is computed by a simple best match strategy, that is, point $P \in S_1$ is matched with $Q \in S_2$ if $DisSim(P, Q) = min_{i \in S_2}(DisSim(P, i))$. No extra measures are being taken to force the correspondence to be one-to-one although, as suggested by Figure 6, our descriptor performs better in terms of recovering a one-to-one correspondence.

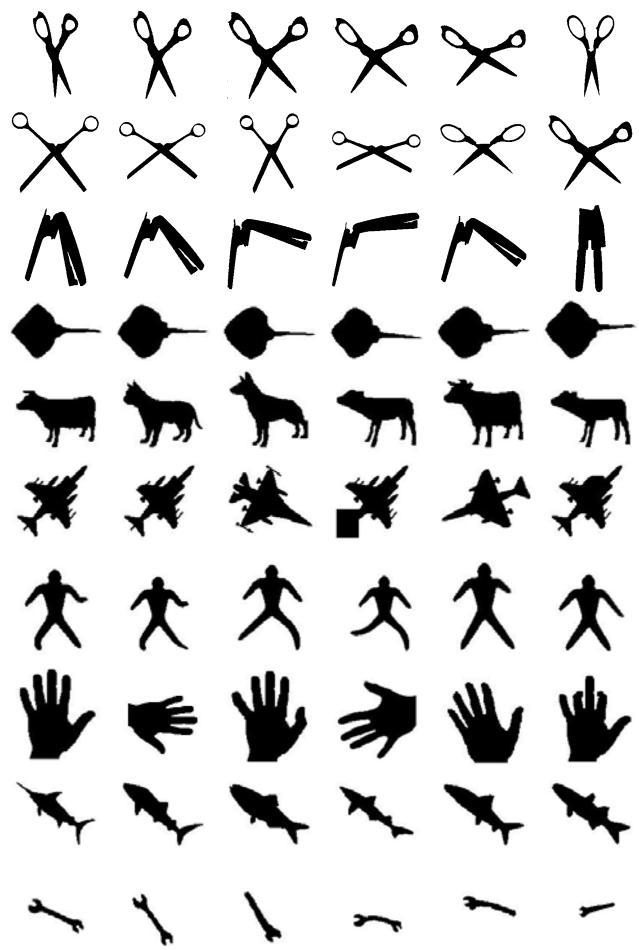
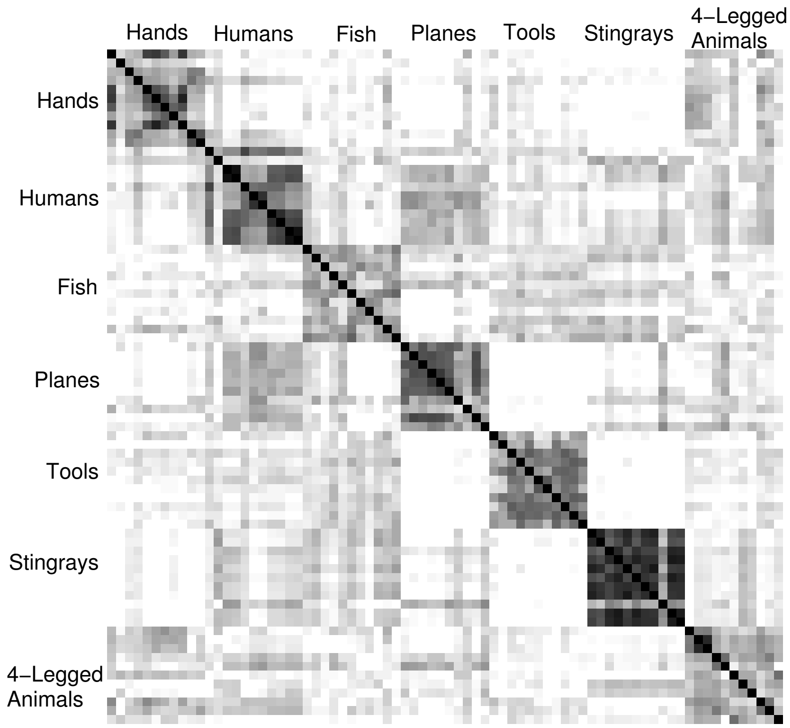


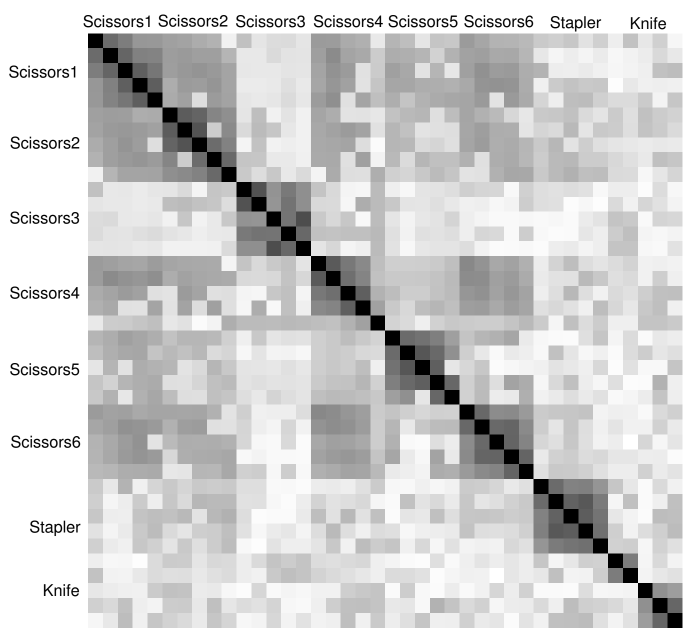
Figure 4. Shape searching results: First column shows the input query shapes. The next five columns show the top five matches from the database.

5. Conclusions and future work

We have presented a shape descriptor for finding a correspondence between two 2D contours. The main feature of our shape descriptor is that by performing a geodesic traversal, it captures the shape of the contour more effectively and thus finds a meaningful correspondence. It is simple and intuitive to understand and implement and is efficient since the final matching is quickly extracted by reading off the dissimilarity matrix. We experimentally show that our descriptor is robust against noise and non-rigid shape defor-



(a) Dissimilarity matrix for Brown database.



(b) Dissimilarity matrix for Ling et al's database.

Figure 5. Image plot of the dissimilarity matrix for the Brown shape database and Ling et al's articulated shape database, obtained using our shape descriptor as a means for shape retrieval. Note that darker spots represent less dissimilarity, hence a better match.

mations such as bending and moderate amounts of stretching.

In future, we would like to generalize the descriptor for 3D shape correspondence and build a framework for 3D shape searching and retrieval based on our descriptor. Also, we would like to explore the possibility of designing a shape descriptor that is invariant to stretching.

References

[1] S. Belongie, J. Malik, J. Puzicha. Shape Context: A new descriptor for shape matching and object recognition. *NIPS*, 2000.

[2] S. Belongie, J. Malik, J. Puzicha. Matching Shapes. In *IEEE International Conference on Computer Vision*, 2001.

[3] J. Chan, H. Tek, B. Kimia. Symmetry-based indexing of image databases. In *J. Visual Communication and Image Representation*, 1998, vol. 9, no. 4, pages 268-278.

[4] T. Gatzke, C. Grimm, M. Garland and S. Zelinka. Curvature maps for local shape comparison. In *Proceedings of Shape Modeling International*, 2005.

[5] D. D. Hoffman and W. A. Richards, “Parts of Recognition,” *Cognition*, Vol. 18, pp. 65–96, 1984.

[6] A. Jain, *Fundamentals of Digital Image Processing*, Prentice Hall, 1989.

[7] A. Johnson and M. Hebert. Recognizing objects by matching oriented points. In *Proc. Computer Vision and Pattern Recognition (CVPR)*, 1997, pages 684-689.

[8] A. Johnson and M. Hebert. Using Spin-Images for efficient multiple model recognition in cluttered 3-D scenes. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1999, pages 433-449.

[9] H. Ling, D. W. Jacobs. Using the Inner-Distance for Classification of Articulated Shapes. In *Proc. Computer Vision and Pattern Recognition (CVPR*, 2005 Vol. 2, pages 719-726.

[10] L. Liu, G. Wang, B. Zhang, B. Guo, H-Y. Shum. Perceptually Based Approach for Planar Shape Morphing. In *Proc. Pacific Graphics*, October, 2004, pages 111-120.

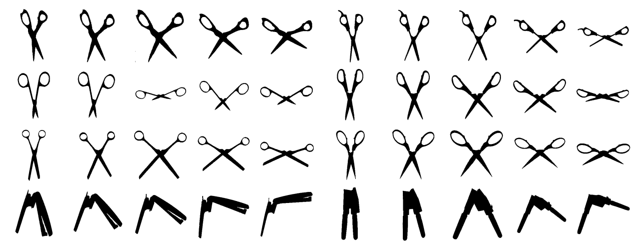
[11] C. Scott and R. Nowak. Robust contour matching via the order preserving assignment problem. In Technical Report TREE 0406, Department of Electrical and Computer Engineering, Rice University., May, 2004.

[12] T. W. Sederberg, E. Greenwood. A physically based approach to 2D shape blending. In *Proc. SIGGRAPH*, 1992, pages 25-34.

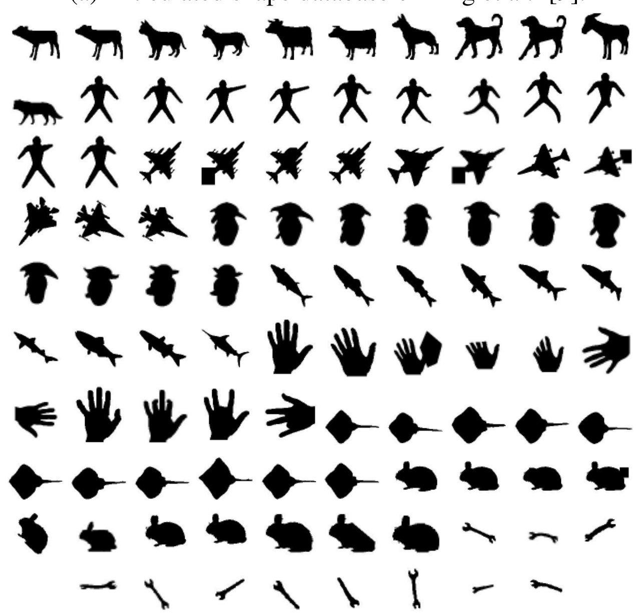
[13] P. Shilane, P. Min, M. Kazhdan, and T. Funkhouser. The Princeton Shape Benchmark. In *Shape Modeling International*, June, 2004.

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

[15] Y. Zhang. A fuzzy approach to digital image warping. In *IEEE Computer Graphics and Applications*, (16)2, 1996, pages 33-41.



(a) Articulated shape database of Ling et al. [9].



(b) Brown database [3].

Figure 7. Image databases used for experiments.

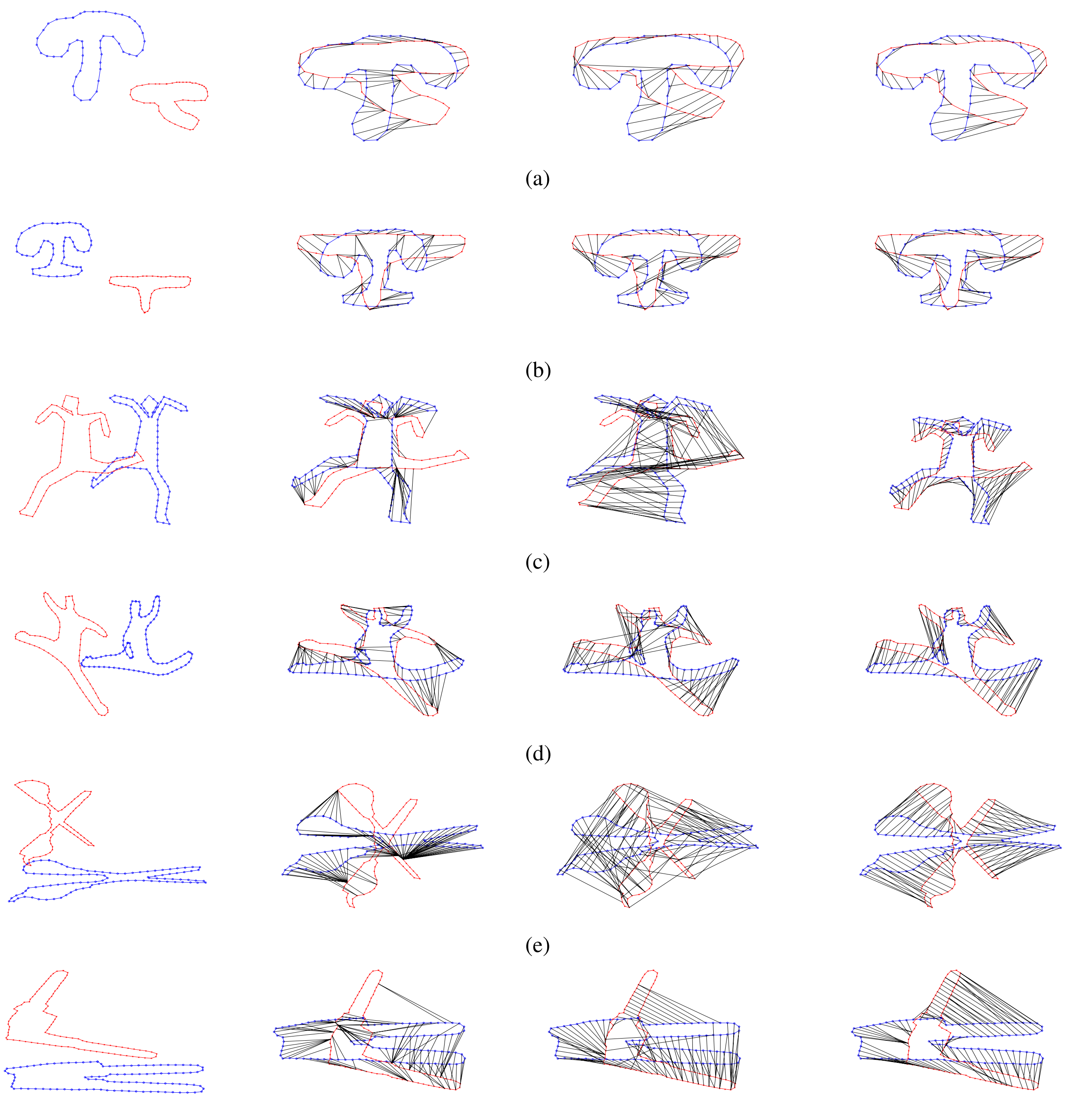


Figure 6. Comparsion between point correspondence results, where corresponding points are linked by a line segment in black. The leftmost column shows the two shapes to be matched. Then from left to right, correspondence results using the original shape context [1, 2], results using sampled curvatures and results using our descriptor. As we can see, the original shape context performs surprisingly well on the human figures with a great deal of bending, except for a few bad mismatches at the arms and legs; they are not as robust on the "mushroom" shapes. Sampling curvatures, even with robust curvature estimation, does not work well. Our shape descriptor, on the other hand, exhibits a high degree of robustness throughout our experiments.