Matching and Recognition of Shapes using Chord-based Point Density Graphs

Xiaolei Huang, Dimitris N. Metaxas
Division of Computer and Information Sciences
Rutgers University - New Brunswick, NJ, USA
{xiaolei, dnm}@cs.rutgers.edu

Abstract

We present a novel approach to matching and recognition of 2D shapes, which uses a chord-based shape representation and matches shapes by finding correspondences between chords. Given a 2D shape, we choose its basis chord such that both end points of the chord have high probability of being present under occlusion. Then the shape is transformed and represented in an internal reference frame uniquely defined by the basis chord. In this chord-based reference frame, we capture the coordinate distribution of all shape points using a point density graph. When a second shape is being matched against the first shape, we pursue a chord on the second shape based on which the second shape's point density graph is the closest to that of the first shape. To avoid exhaustive search of all chords on the second shape, we use the Chord Length Distributions of the two shapes to prune a dominant portion of the search space. The distance between two point density graphs is measured using a symmetrized Kullback-Leibler divergence. Then when the correspondence between a pair of chords is established, a unique similarity transform is determined to match the two shapes so that the corresponding chords are aligned. Finally, we employ a hierarchical approach to extend our method to include Affine transformations. Matching and Recognition results from the Brown SIID project shapes, the MNIST dataset of handwritten digits, and the SQUID fish database, demonstrate our algorithm's performance, its invariance properties and its robustness to occlusion, articulation, missing gaps, and spurious structures on shapes.

1. Introduction

Matching 2D shapes and measuring the similarity between shapes are important problems in Computer vision. Our main target application domains are shape-based recognition, indexing and retrieval, and matching training shapes of the same class for building shape prior models. In this paper, we aim to develop a shape matching algorithm that is fast, invariant to rotation/translation/scaling, applicable to shapes of arbitrary topology, and robust to partial occlusion, missing gaps, and spurious boundary structures.

Our shape representation is inspired by the geometric hashing framework proposed by Wolfson and collaborators [1], [3]. In the geometric hashing paradigm, the coordinates of points on a model serve as a natural geometric characteristic of the model. While coordinates depend on a reference frame, it is important to use an internal reference frame of the model that will remain present under partial occlusion and remain unchanged if the model undergoes rotation, translation, and scaling. In our approach, a shape is represented by a set of points. Then a pair of points on the shape is chosen to be a basis, and an internal reference frame is defined based on the chord joining the pair of points. The pair of points are chosen in such way that the probability of these basis points being occluded is small. In the internal reference frame, the basis chord is aligned with the x-axis. The midpoint of the chord is the origin, and the length of the chord is normalized to be one. Finally, when all points on the shape are transformed into this chord-based reference frame, their coordinates in this frame easily give rise to a model-based shape representation that is invariant to rotation, translation and scaling.

When two shapes are being matched, we need to find correspondences between them and estimate an alignment transform to match them as closely as possible. The main difference between our approach to finding correspondences and previous approaches is that, instead of finding correspondence between individual points on the two shapes, we pursue the correspondence between chords, which is in essence the correspondence of two pairs of points simultaneously. Given a shape, we choose its basis chord, and transform it to the normalized reference frame defined by the chord. Then the coordinate distribution of points on the shape with respect to the basis chord in this reference frame is captured using a point density graph, which is a robust and discriminative global shape descriptor that resembles the Shape Context proposed by Belongie, Malik et. al. [4]. On the second shape, we aim to find a chord based on which the second shape's point density graph is the closest to the first shape's point density graph. To avoid expensive exhaustive search, we use the Chord Length Distribution, proposed by Jain and collaborators [6], [7], to prune a dominant portion of the search space instantly. The simi-

larity measure between two point density graphs is defined using a symmetrized Kullback-Leibler divergence, which is widely used to measure how different two probability distributions are. The result is a fast algorithm to find the closest matching chord on the second shape to the basis chord on the first shape. Once we have the correspondence between two chords, one from each shape, a unique similarity transformation is determined to match the two shapes such that the two corresponding chords are aligned. To increase accuracy and robustness to occlusion, we can find correspondences between multiple pairs of chords. Then we can compute a transformation either based on the pair that is most closely matched, or use the correspondences between all the chord endpoints. Finally, we extend our matching algorithm and shape similarity measure in a hierachy of transformations, which consists of the Similarity (rotation, translation and isotropic scaling) and Affine (Similarity, nonisotropic scaling and shearing) transformations.

We tested our algorithm using several widely used benchmark shape datasets, including a shape database (99 shapes) from the Brown SIID project [8], the MNIST database of handwritten digits [2], and the SQUID fish shape database [9]. The experimental results demonstrate several very attractive aspects of our approach: (1) it is very fast, with a worst-case complexity of $O(N^2)$, where N is the number of points on the shape with the smaller number of points, (2) it is intrinsically invariant under rotation, translation and scaling, (3) it can handle shapes of arbitrary topology, and the shapes do not need to represented using the same number of points, and (4) it is robust to occlusion, articulation, missing gaps and spurious structures on shapes.

In Section 2, we review some of the relevant literature. In Section 3, we present our chord-based shape representation. In Section 4, we describe our complete chord-based shape matching algorithm. The results of our algorithm on several databases are demonstrated in Section 5, and we conclude in Section 6.

2. Previous Work

A large body of research has been devoted to shape matching, comparison and recognition. The most commonly used shape representation primitives are curves, point sets, and medial axes.

Many traditional curve matching approaches [13], [14], [15] use local invariant features (e.g., curvature) as descriptors or footprints, and find correspondences between points or curve segments with close footprints. They mostly account for rotation and translation, and have not been proven to be robust. The Geometric Hashing paradigm proposed by Lamden and Wolfson [1] is a more robust model-based technique for matching geometric features. It can match features that undergo rigid or affine transformations. The approach in [18] used dynamic programming to find an edit transformation that maps one curve onto another based on their syntactic representations.

Matching shapes using point sets is a popular approach. In [4], Belongie, Malik et. al introduced Shape Context, a decriptor at every point on the shape that captures the coarse distribution (histogram) of the rest of the shape with respect to that point. Then an optimal one-to-one assignment problem is solved to establish correspondences between points on two shapes such that the total matching cost is minimized. Spin Image [10] is a similar point distribution histogram and it is used to describe the global shape of a 3D object. Another significant work in robust and non-rigid point set matching is by Rangarajan and collaborators [16], [17]. They developed an iterative optimization algorithm to determine the correspondences and non-rigid transformation between two point sets jointly using a combination of techniques such as deterministic annealing and softassign. The Iterative Closest Point (ICP) [11], [12], is another shape registration method with good performance. The method does need a good initial transformation estimate between two shapes, then it converges quickly to a close-to-optimal solution. In [22], Duta et al. also described a 2D shape registration method to find the matching subsets of points on a pair of shapes.

Shapes are also recognized and compared using their medial axis graphs [21] or shock graphs [8], [19]. Under the context of shape-based indexing and retrieval, people also use many shape representations based on shape descriptors. A good review on visual shape descriptors can be found in [20].

Our chord-based shape matching algorithm achieves the invariance to scaling, rotation and translation in a natural way. Also because it is fast, it is suitable for solving an initial alignment transformation between shapes, and it can be used for matching, indexing, retrieval and recognition of shapes in large databases.

3. The Chord-based Shape Representation using Point Density Graphs

3.1. The basis chord for a shape

Given a shape, S, represented by a set of n points, a chord is a line segment joining two points on the shape. Among the total $\binom{n}{2}$ chords, the ideal basis chord would be the one whose two end points have the least probability of being occluded when the object corresponding to the shape appears in new images. In practice, we propose two rules for choosing and computing the basis chord.

The first rule uses the chord passing through the shape centroid, and in the direction of the major axis of variation. A fast algorithm based on Singular Value Decomposition (SVD) can be used to find the two end points of this chord. Considering the x and y coordinates of the shape points as two random variables, a symmetric covariance matrix C of

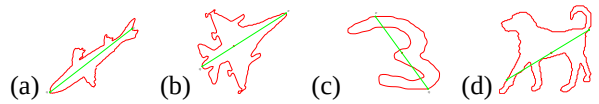

Figure 1: Examples of the chosen basis chords using the first rule.

size 2 by 2 is defined by:

$$C = \frac{1}{n}\begin{pmatrix} \tilde{X}^T\tilde{X} & \tilde{X}^T\tilde{Y} \\ \tilde{X}^T\tilde{Y} & \tilde{Y}^T\tilde{Y} \end{pmatrix}$$

where $\tilde{X}$ and $\tilde{Y}$ are the deviation vectors from the shape points to the shape centroid. By applying SVD, we can get the eigenvector v_1 corresponding to C's largest eigenvalue, and v_1 is the direction of the major axis of variation. We then draw a line through the shape centroid with direction v_1, and choose the two extrema intersection points between the shape and the line, P and Q, to be the end points of our basis chord. Examples of the chosen chords on a number of shapes using this rule are shown in Figure 1.

The second rule for choosing the basis chord of a shape is biologically inspired. By observing the shapes of many animals, we can see that in general, thin limbs have higher probability of being occluded than the thick torso in images of the same object [1]. So we would prefer a basis chord across the thick torso part of a shape. However, using the first rule, it is possible that one or both of the end points of the computed chord is located on a thin limb, instead of the thick torso. One such example is shown in Figure 1(d). Our proposed solution to this problem utilizes the medial axes (skeleton) of a shape. For the dog shape S in Figure 1(d), its skeleton is shown in Figure 2(a). Then the skeleton is segmented into branches at the junction points. In Figure 2(b), each branch is rendered using a different color, and the junction points are marked with blue asterisks. Next, we evaluate a significance value for each branch. Centering at each medial axis point p, we use the radius of the maximal disc that fits in the shape solid, r_p, to measure the thickness of the shape around that point. Let us denote the branches on the skeleton as $B_1, B_2, ..., B_m$. We then evaluate the branch B_i's significance value as:

$$S(B_i) = w_1\frac{\bar{r}_{B_i}}{\bar{r}} + w_2\frac{l_{B_i}}{\bar{l}} + w_3\frac{n_{B_i}}{\bar{n}}$$

In the above formula, w_k (k=1, ..., 3) are weights, $\bar{r}_{B_i}$ is the average maximal radius of all skeleton points on branch B_i, $\bar{r}$ is the average maximal radius of all skeleton points, l_{B_i} is the length of branch B_i, $\bar{l}$ is the average length of all branches, n_{B_i} is the number of other branches connected to B_i, and $\bar{n}$ is the average number of other branches connected to a branch. Basically, a branch's significance value depends on its average maximal radius (thickness), its length, and the number of other branches connected to it. Once we recognize the branch with the largest significance value, we fit a line to that branch, and the two extrema intersection points between the line and the original shape are chosen to be the end points of the basis chord using our second rule. We show the resulting basis chord for the dog shape in Figure 2(c).

[1] here limb and torso are both used in a general sense. Limbs denote the thin protruded subparts of the body, and torso is the central thick part of the body.

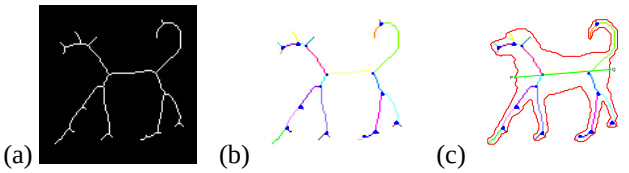

Figure 2: Derivation of the basis chord using the second rule. (a) the shape skeleton. (b) the skeleton is segmented into branches at junction points. Each branch is rendered in a different color, and the junction points are marked with blue asterisks. (c) The basis chord $\overline{PQ}$ is found by fitting a line (in green) to the most significant skeleton branch (in yellow). The weights used for evaluating the branch significance values are: $w_1 = 0.4, w_2 = 0.3, w_3 = 0.3$.

Among the two rules, the second rule fits our least probability of occlusion criterion better, especially for complex articulated shapes. But the first rule is highly efficient to compute and also effective on many shapes. In practice, in order to choose a good basis chord automatically, we first compute two candidate chords, one from each rule. Then if the two candidates roughly agree with each other, we can use either candidate as the chosen basis chord. If the two candidates divert from each other, we check whether the most significant skeleton branch of the shape approximates a straight line, if so, we choose the chord by the second rule, otherwise, we use the chord using the first rule.

To achieve better robustness to occlusion, instead of one basis chord, we can keep multiple basis chords of a shape and represent the shape in a coordinate system based on each chord. We can choose the first basis chord using the above rules, then the other chords can be picked either radially distributed around the first chord, or orthogonal to the first chord and equally spaced.

3.2. The Chord-based Transformation

Once the basis chord of a shape is computed, we transform the shape into an internal reference frame defined by the chord. In this reference frame, the basis chord is aligned with the x-axis. The midpoint of the chord is the origin, and the length of the chord is normalized to be one.[2] The result is a chord-based shape representation that is invariant under rotation, translation, and isotropic scaling of the original shape. For example, a fish shape, as shown in Figure

[2] This similarity transformation matrix is easy to derive, so we ignore the derivation here.

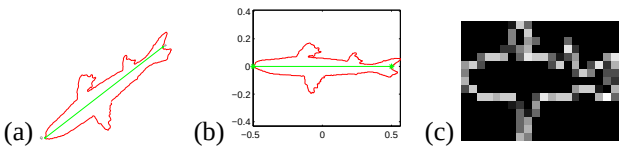

Figure 3: The chord-based transformation and the point density graph. (a) A fish shape with its basis chord marked. (b) The transformed shape in the chord-based reference frame. (c) The Point Density Graph in the normalized chord-based reference frame. Here the quantization parameters $M = 20, N = 15$. The intensity of each bin encodes the point density at that bin. The brighter, the higher density.

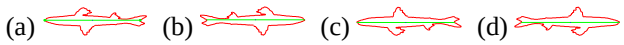

Figure 4: The four possible transformations based on the chord. (a) The first transform, same as that in Figure 3(b). (b) The reflection of (a) relative to the y-axis. (c) The reflection of (a) relative to the x-axis. (d) The reflection of (b) relative to the x-axis.

3(a), is transformed based on its chosen basis chord $\overline{PQ}$, and the transformed shape is shown in Figure 3(b). Note, however, that when a basis chord is fixed, there are in fact totally four possible ways of orienting the axes in the normalized reference frame based on the chord, thus there are four possible chord-based transformations. In Figure 4, all four transformations are shown. Since the four transformations are reflections of each other, it is enough for us to derive the transformation for only one case. In practice, we keep all four possible transformations for matching.

3.3. The Point Density Graph

We represent the transformed shape in the chord-based coordinate system using a Point Density Graph. The graph captures the coordinate distribution of shape points with respect to the basis chord and is a shape representation with global characteristics.

For a transformed shape, as shown in Figure 3(b), we first get the minimum and maximum x and y coordinates of all the points on the shape. Suppose the x coordinates are within interval $[xl, xu]$, and the y coordinates are within $[yl, yu]$, we then quantize the x interval into M bins, and quantize the y interval into N bins, so we totally have $M*N$ bins in the center region.

At any bin, let the number of shape points that lie in the bin be h_{ij}, where $i = 1, ..., M$ and $j = 1, ..., N$. The point density graph value at the $ijth$ bin is defined by the normalized value:

$$b_{ij} = \frac{h_{ij}}{\sum_{k=1}^{M}\sum_{l=1}^{N} h_{kl}}$$

The graph corresponding to the fish shape in Figure 3(a) is shown in Figure 3(c).

The point density graph is in essence a coarse 2D distribution (normalized histogram) of the shape point coordinates

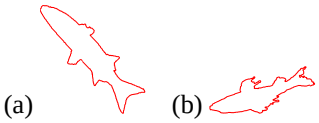

Figure 5: Two fish shapes with arbitrary orientation, position and scale.

in the chord-based reference frame. The higher density at a bin, the higher probability a point on the shape appears in that bin. The shape matching performance of our algorithm is not sensitive to the choice of the quantization parameters M and N. But too small values induce large quantization errors, and too large values are not robust to spurious structures along the shape contour. For shapes represented using several hundred points, the values between $10 \sim 20$ are good. We can also use the aspect ratio of a transformed shape's bounding box to determine a proper ratio between M and N.

4. The Chord-based 2D Shape Matching Algorithm

We now present our complete algorithm for matching 2D shapes of arbitrary position, orientation and scale. The basic idea is that, given a pair of shapes, we choose a basis chord on the first shape using the rules proposed in Section 3.1, and compute the shape's point density graph based on that chord. Then for the second shape, we compute its point density graph based on every chord in a small subset of all chords (the subset is the result of an effective candidate pruning process using chord length distribution). The chord, based on which the second shape's point density graph is the closest to that of the first shape, is considered the closest matching chord (correspondence) to the first shape's basis chord. Then based on the pair of matched chords, we can compute the translation, rotation and scaling parameters to align the two shapes. The dissimilarity between two shapes is measured using the symmetrized Kullback-Leibler distance between their point density graphs based on the pair of corresponding chords.

As an example, we use the two fish shapes, S_1 and S_2, as shown in Figure 5 to describe the algorithm. Suppose S_1 is represented by m points $(x_1^{(1)}, y_1^{(1)}), ..., (x_m^{(1)}, y_m^{(1)})$, and S_2 is represented by n points $(x_1^{(2)}, y_1^{(2)}), ..., (x_n^{(2)}, y_n^{(2)})$. First, we get the first shape's basis chord, $\overline{PQ}^{(1)}$, using the rules described in section 3.1, and transform S_1 into a reference frame based on the chord, as described in section 3.2. Then the first shape's four point density graphs can be computed as in section 3.3.

Next, in order to efficiently find the closest matching chord $\overline{PQ}^{(2)}$ on the second shape S_2, the steps in our algorithm are:

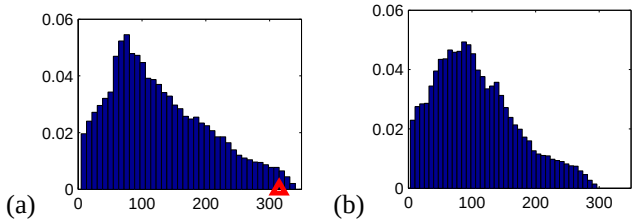

Figure 6: The two chord length distributions presented as normalized histograms. (a) The chord length distribution of S_1. The length of the basis chord $\overline{PQ}^{(1)}$ is pointed using a red upper triangle. (b) The chord length distribution of S_2.

1. Compute the distribution of lengths of all chords, i.e. the chord length distribution, for both S_1 and S_2. For each shape, we first get the histogram of its chord lengths, then the histogram is normalized so that the sum of all bins is equal to 1.[3] This normalized chord length histogram is a good approximation to the shape's chord length density function. Let us denote S_1's chord length distribution as H_1, and S_2's chord length distribution as H_2. Figure 6 shows the two histograms.
2. Let the length of $\overline{PQ}^{(1)}$ be l_1, and suppose its correspondence $\overline{PQ}^{(2)}$'s length is l_2. If we assume the two shapes are similar and can be matched sufficiently closely, then the percentile corresponding to l_1 on H_1 should be close to the percentile corresponding to l_2 on H_2, i.e., the percentage of chords with lengths less than l_1 on the first shape should be close to the percentage of chords with lengths less than l_2 on the second shape. Based on this theory, we can prune a dominant portion of the search space for $\overline{PQ}^{(2)}$ instantly, by restricting its candidates in a small subset of all chords on S_2. More formally, let the chord length random variable in S_1 be V_1, and let the chord length random variable in S_2 be V_2. Then the candidate chords for $\overline{PQ}^{(2)}$ are in the small subset C:

$$C(\overline{PQ}^{(2)}) = \{\, \overline{p_i p_j} \mid p_i = (x_i^{(2)}, y_i^{(2)}), \; p_j = (x_j^{(2)}, y_j^{(2)}), \; and \; -\alpha < [\, Pr\,(\, V_2 < length(\overline{p_i p_j})\,) - Pr\,(\, V_1 < length(\overline{PQ}^{(1)})\,)\,] < \alpha \,\}$$

   Here α is a small fractional number that accounts for the allowed variability. Typically, we choose $\alpha < 0.1$. The greater is α, the more candidate chords on S_2 we need to evaluate using the subsequent steps.
3. Based on every chord in $C(\overline{PQ}^{(2)})$, we transform S_2 into an internal reference frame, and compute its point density graph in that frame. Note, however, that using the method for computing point density graphs in section 3.3, the graphs of S_2 based on parallel chords of different lengths will actually be the same, since the quantized region always covers the whole shape. To solve this problem, for a transformed shape $TS_2^{\overline{p_i p_j}}$ based on the chord $\overline{p_i p_j}$ of S_2, instead of quantizing the whole bounding box region of the transformed shape, denoted as $A_2 = [xl^{(2)}, xu^{(2)}] \times [yl^{(2)}, yu^{(2)}]$, we only quantize the region equivalent to the quantized region of the transformed first shape $TS_1^{\overline{PQ}^{(1)}}$, denoted as $A_1 = [xl^{(1)}, xu^{(1)}] \times [yl^{(1)}, yu^{(1)}]$. After the region A_1 of $TS_2^{\overline{p_i p_j}}$ is quantized to $M * N$ bins, let the number of $TS_2^{\overline{p_i p_j}}$ shape points that lie in each bin be h_{kl} ($k = 1, ..., M$, $l = 1, ..., N$). Then in order to make the density graph comparable to the first shape's density graphs, we define the normalized graph value at the klth bin as: $b_{kl} = \frac{h_{kl}}{n_r}$, where n_r ($n_r \leq n$)) is the total number of shape points in region A_1 of $TS_2^{\overline{p_i p_j}}$. One more thing to mention is that, for S_2, among the four possible chord-based transformations, we only need to keep the point density graph for any one of them, since we will compare this graph against all four possible density graphs of S_1.
4. We then measure the distances between all the point density graphs we acquired for S_2 based on different chords, and the four point density graphs of S_1. And $\overline{PQ}^{(1)}$'s closest matching chord $\overline{PQ}^{(2)}$ is chosen to be the chord that generates the closest point density graph of S_2 to one of S_1's point density graphs. Considering a point density graph as the coordinate *probability distribution* of all points on a shape in its natural reference frame, we use the Kullback-Leibler (K-L) divergence (also known as relative entropy, cross entropy) to compare two point density graphs. Since K-L divergence is asymmetric, in this paper, we instead use a symmetrized K-L divergence - the Bhattacharyya "distance" [5]. Let us denote two density graphs as g_1 and g_2, both represented as normalized 2D histograms of size $M * N$. Also let the density graph value at the $ijth$ bin in g_1 be $b_{ij}^{(1)}$, and $b_{ij}^{(2)}$ in g_2. Then the Bhattacharyya distance between the two graphs is:

$$D(g_1, g_2) = -\log(\sum_{i=1}^{M}\sum_{j=1}^{N}\sqrt{b_{ij}^{(1)}}\sqrt{b_{ij}^{(2)}})$$

Now we have a pair of matching chords: $\overline{PQ}^{(1)}$ on S_1 and $\overline{PQ}^{(2)}$ on S_2, a unique similarity transformation is determined to match the two shapes such that the two matching chords are aligned. The transformation matrix applied on S_2 to align it with S_1 is: $M = M_1^{-1} M_2$, where M_1

[3] At each bin, its normalized value is the number of chords whose lengths lie in the range of the bin, divided by the total number of chords.

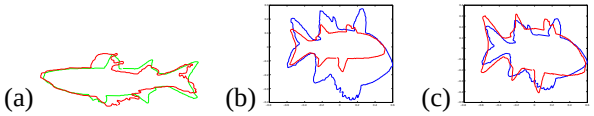

Figure 7: (a) The two fishes in Figure 5 are matched together by a similarity transformation. The quantization parameters $M = 20, N = 10$, and the chord length distribution pruning parameter $\alpha = 0.05$. (b)-(c) Matching shapes in the two-level hierachy of transformations. (b) A "fat" fish and a "thin" fish matched using a similarity transformation. (c) Using the transformation parameters from (b) as initial estimate, the two fish shapes are further matched using the Affine transformation, which allows for non-isotropic scaling and shearing.

is the transformation for transforming S_1 into the reference frame based on $\overline{PQ}^{(1)}$, and M_2 is the transformation for transforming S_2 into the reference frame based on $\overline{PQ}^{(2)}$. Note that, among the four possible transformations, we use the one that produces the matching point density graphs. Finally, the result of matching the two example fish shapes using a similarity transformation is shown in Figure 7(a).

To increase matching accuracy and achieve better robustness to occlusion, we can also find correspondences between multiple pairs of chords. We have described several ways of choosing multiple basis chords on the first shape in section 3.1. Then the closest matching chord on the second shape to each of the first shape's basis chords can be found as described in this section. Then we can either compute a transformation based on the closest pair of matching chords, or use the correspondences between all the chord endpoints to estimate a transformation using a robust estimator.

4.1. Shape Similarity and Shape Matching in a hierarchy of transformations

In this paper, we propose to measure shape similarity and to match shapes in a "hierachy of transformations." The hierachy we currently use has two levels: the Similarity Transformation as a lower level, and the Affine Transformation as a higher level. In the hierachy, shapes can be matched from the lowest level to the highest level, and the transformation parameters derived at a lower level are used to initialize the matching at a higher level. The dissimilarity, or distance, between two shapes is measured using a weighted sum of the distances at all levels. In our experiments, we have found that our fast matching algorithm using similarity transformation is sufficient for recognition and retrieval in large databases and matching training shapes of the same class. In order to compare two shapes and acquire a good dissimilarity measure, however, we need to consider the Affine transformation as well, since similar shapes often undergo non-isotropic scaling and shearing transformations (e.g. fishes, handwritten digits). In practice, we use the matching result from the similarity transformation to initialize another matching process using the higher level Affine transformation. The Affine matching is based on the correspondences between closest points on two shapes and a gradient descent optimization procedure. In Figure 7(b)-7(c), we show a matching example in our current two-level hierachy.

5. Experimental Results

We applied our chord-based shape matching algorithm on several widely used benchmark shape databases, to evaluate its performance, its invariance properties and its robustness to articulation, occlusion, missing gaps and spurious structures on shapes. During our experiments, it took about $10ms$ to match a pair of shapes ($1s$ to match 100 pairs of shapes) with each shape being represented by 200 points, on a 1GHz workstation.

5.1. The Brown Shape Database

Our first experiment uses a shape database (99 shapes) from the Brown Shape Indexing of Image Databases (SIID) project [8]. The shapes are from nine categories, and eleven shapes are included in each category to allow for variations in form, occlusion, articulation, and missing parts, etc. In our experiment, we pick nine query shapes, one from each category, then match all the remaining (90) test shapes against every query shape. For the sake of brevity, the matching and recognition results from three categories are shown in Figure 8. For each query shape, in the first row, we show its 10 closest matching test shapes using our shape similarity measure. Then in the second row, we show the test shapes and the query shape matched together, and the symmetrized K-L distances between them are also shown. At the end of each row, three additional test shapes are shown in the order of increasing distance from the query shape. Our algorithm performed very well on the task. Using the all nine categories and the K-Nearest-Neighbor (K-NN) method for recognition, we achieved a 98% recognition rate. The alignments between the query shape and the test shapes are also very good.

5.2. The MNIST Handwritten Digits

Our second experiment is on a subset of the MNIST database of handwritten digits [2], [4]. The subset we use has 200 digits of 10 categories (digits 0-9), and the shapes in a same category have large variability.[4] The matching and retrieval results using three query digits are presented in Figure 9. For each query digit, we match all the remaining 199 test digit shapes in the dataset against it, then its top ten closest test shapes are shown in the first row and the matching between the test shapes and the query is shown in the second

[4] **We are submitting the complete datasets as Supplementary Material files.**

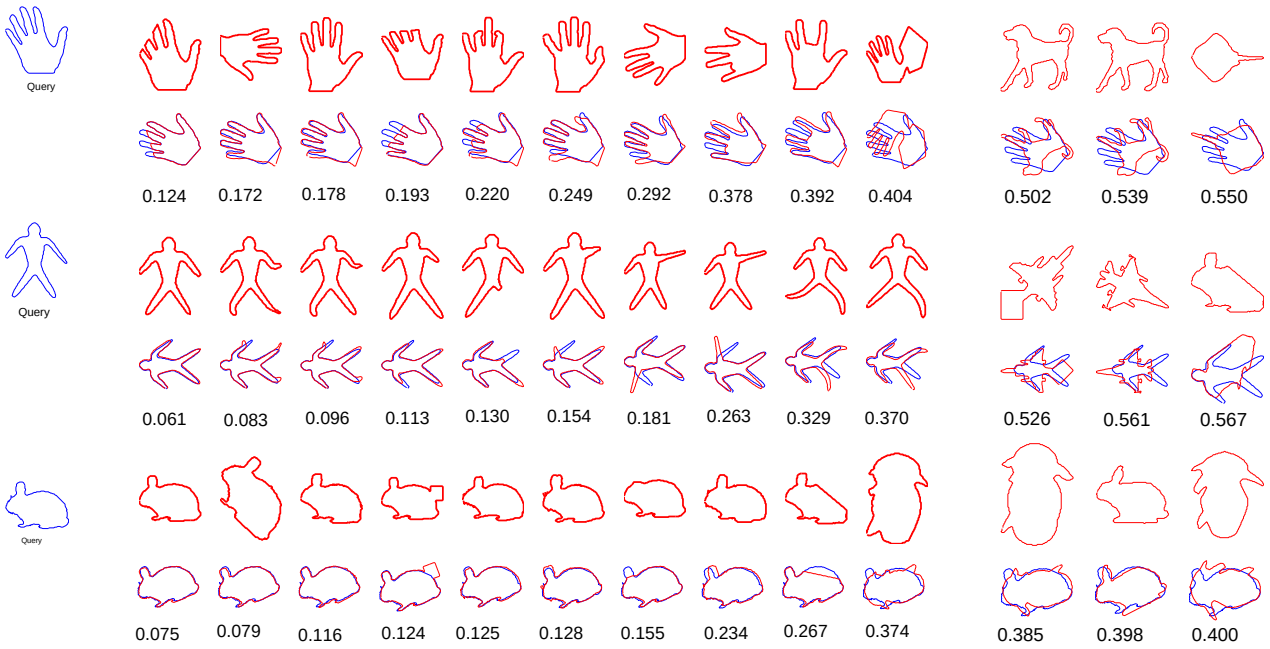

Figure 8: Recognition and matching results for shapes in the Brown small (99 shapes) database. The left-most blue shapes are the query shapes. For each query shape, we show its top 10 closest test shapes (red) in the first row, and show the test shapes (red) and the query shape (blue) matched together in the second row. At the end of each row, 3 additonal shapes that ranked the 11th-13th closest are also shown. The one false positive example in the bunny top ten matches is due to the toy shape's similarity to the bunny shape after rotation and scaling.

Figure 9: Recognition and Matching results on a set of 200 handwritten digit shapes. The left-most blue shapes are the query shapes, for each query shape, we show its top 10 closest test shapes in the first row, and show the test shapes (red) and the query shape (blue) matched together in the second row, the distance between each test shape and the query shape is also shown. In the top ten matches for digit 9, some digit 6 shapes appear because the two shapes are similar after rotation and scaling.

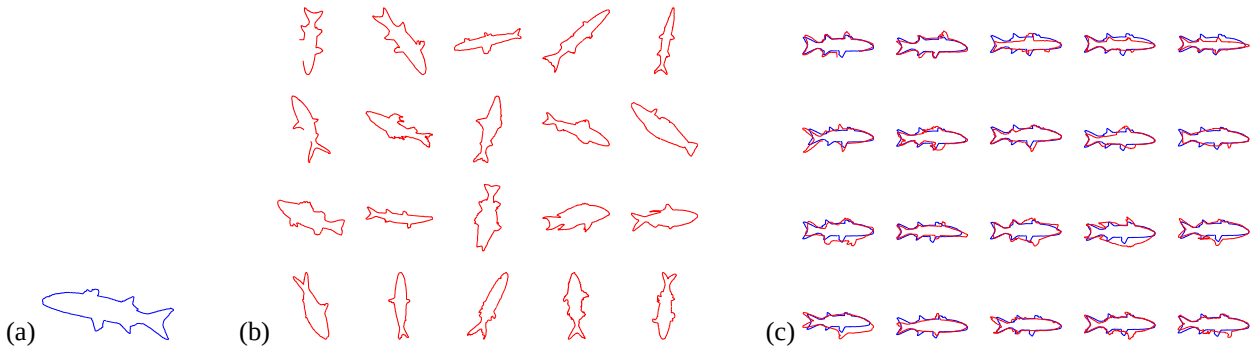

Figure 10: Matching fish shapes with missing gaps and spurious boundary structures. (a) The query fish shape. (b) The 20 test fish shapes in arbitrary orientation, scale and position. Note the missing gaps on the $1st$ and $6th$ shapes. (c) Each of the test shapes (in red) is matched with the query shape (in blue).

row. From the results, we can see that our algorithm performed very well on the recognition, retrieval and matching of complex handwritten digit shapes.

5.3. The SQUID Fish Shape Database

We also use a set of fish shapes in the SQUID database [9], to demonstrate our algorithm's robustness to missing gaps and spurious structures on shapes. In Figure 10(a), the query fish shape is shown. In Figure 10(b), 20 test fish shapes which are different from the query are shown. Note that many test fish shapes have spurious boundary structures and the $1st$ and the $6th$ test shapes have missing gaps. In Figure 10(c), the 20 test shapes (in red) are shown matched with the query shape (in blue).

6. Conclusions

In this paper, we presented a novel chord-based 2D shape representation and a hierarchical chord-based shape matching algorithm. Our shape representation achieves the invariance to rotation, translation and scaling naturally, and our experiments on several widely used datasets demonstrate the shape matching algorithm's performance and its robustness to occlusion, articulation, missing gaps and spurious structures on shapes. Our future goal is to further validate our method on very large shape databases.

References

[1] Y. Lamdan, and H. J. Wolfson, "Geometric Hashing: a general and efficient model-based recognition scheme," *Proc. Int'l Conf. Computer Vision (ICCV'88)*, pp. 238-249, 1988.

[2] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based Learning applied to Document Recognition," *Proceedings of the IEEE*, Vol. 86, No. 11, pp. 2278-2324, November 1998.

[3] H. J. Wolfson, and I. Rigoutsos, "Geometric Hashing: An Overview," *IEEE Computational Science and Engineering*, Vol. 4, No. 4, pp. 10-21, 1997.

[4] S. Belongie, J. Malik, and J. Puzicha, "Matching Shapes," *Proc. Int'l Conf. Computer Vision*, Vol. 1, pp. 454-461, 2001.

[5] T. Kailath, "The divergence and Bhattacharyya distance measures in signal selection," *IEEE Trans. on Comm. Tech.*, Vol. 15, No. 1, pp. 52-60, 1967.

[6] S. P. Smith, and A. K. Jain, "Chord Distribution for Shape Matching," *Computer Graphics and Image Processing*, Vol. 20, pp. 259-271, 1982.

[7] Z. You, and A. K. Jain, "Performance Evaluation of Shape Matching via Chord Length Distribution, " *Computer Vision, Graphics and Image Processing*, Vol. 28, pp. 185-198, 1984.

[8] T. B. Sebastian, P. N. Klein, and B. B. Kimia, "Recognition of Shapes by Editing Shock Graphs," *Proc. Int'l Conf. Computer Vision (ICCV'01)*, pp. 755-762, 2001.

[9] F. Mokhtarian, S. Abbasi and J. Kittler, "Robust and Efficient Shape Indexing through Curvature Scale Space," *Proc. British Machine Vision Conference*, pp. 53-62, 1996.

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

[11] P. J. Besl, and N. McKay, "A Method for Registration of 3D shapes," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 14, No. 2, pp. 239-256, 1992.

[12] Z. Zhang, "Iterative Point Matching for Registration of Free-Form Curves and Surfaces," *Int'l J. Computer Vision*, Vol. 13, No. 2, pp. 119-152, 1994.

[13] H. J. Wolfson, "On Curve Matching," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 12, pp. 483-489, 1990.

[14] E. Kishon, T. Hastie, and H. J. Wolfson, "3-D Curve Matching using Splines," *J. of Robotic Systems*, Vol. 8, pp. 723-743, 1991.

[15] G. Barequet, and M. Sharir, "Partial Surface Matching by Using Directed Footprints," *Symposium on Computational Geometry*, pages C-9-C-10, 1996.

[16] S. Gold, A. Rangarajan, C. P. Lu, and S. Pappu, "New Algorithms for 2D and 3D Point Matching: Pose Estimation and Correspondence," *Pattern Recognition*, Vol. 31, No. 8, pp. 1019-1031, 1998.

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

[18] Y. Gdalyahu, and D. Weinshall, "Flexible Syntactic Matching of Curves and its Application to Automatic Hierachical Classification of Silhouettes," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 21, No. 12, pp. 1312-1328, 1999.

[19] K. Siddiqi, A. Shokoufandeh, S. Dickinson, and S. Zucker, "Shock Graphs and Shape Matching," *Int'l J. Computer Vision*, Vol. 35, No. 1, pp. 13-32, 1999.

[20] M. Bober, "MPEG-7 Visual Shape Descriptors," *IEEE Trans. on Circuits and Systems for Video Technology*, Vol. 11, No. 6, pp. 716-719, 2001.

[21] S. C. Zhu, and A. L. Yuille, "FORMS: A Flexible Object Recognition and Modeling System," *Int'l J. Computer Vision*, Vol. 20, No. 3, pp. 187-212, 1996.

[22] N. Duta, A. K. Jain, and M.-P. Dubuisson-Jolly, "Learning 2D Shape Models," *IEEE Conf. on Computer Vision and Pattern Recognition*, Vol. 2, pp. 8-14, 1999.