Using the Inner-Distance for Classification of Articulated Shapes

Haibin Ling David W. Jacobs
Computer Science Department, University of Maryland, College Park
{hbling, djacobs}@ umiacs.umd.edu

Abstract

We propose using the inner-distance between landmark points to build shape descriptors. The inner-distance is defined as the length of the shortest path between landmark points within the shape silhouette. We show that the inner-distance is articulation insensitive and more effective at capturing complex shapes with part structures than Euclidean distance. To demonstrate this idea, it is used to build a new shape descriptor based on shape contexts. After that, we design a dynamic programming based method for shape matching and comparison. We have tested our approach on a variety of shape databases including an articulated shape dataset, MPEG7 CE-Shape-1, Kimia silhouettes, a Swedish leaf database and a human motion silhouette dataset. In all the experiments, our method demonstrates effective performance compared with other algorithms.

1 Introduction

Classification of complex shapes with part structures is an important problem in computer vision. However it is difficult to capture part structures. To attack this problem, we propose using the *inner-distance*, defined as the length of the shortest path within shape boundaries, to build shape descriptors. We show that the inner-distance is insensitive to shape articulations and it is often more discriminative than the Euclidean distance for complex shapes. For example, in Fig. 1, although points on (a) and (c) have similar spatial distributions, they are quite different in their part structures. On the other hand, (b) and (c) appear to be from same category with different articulations. The inner-distances between the two marked points are quite different in (a) and (b), while almost the same in (b) and (c).

The inner-distance is a natural replacement for the Euclidean distance in shape descriptors. We use it to extend shape contexts [2]. Based on the new descriptor, we design a dynamic programming method for silhouette matching that is fast and accurate. The proposed method is tested on a variety of shape databases. Excellent performance is achieved on all of them compared to other algorithms.

The rest of the paper is organized as follows. Sec. 2 discusses related works. Sec. 3 first gives a model for articulation and proves the articulation insensitivity of the inner-distance. Then the inner-distance's ability to capture part structures and computational issues are addressed. Sec. 4 describes the extension of the shape context using the inner-distance, and gives a framework for using dynamic programming for silhouette matching and comparison. Sec. 5 presents and analyzes all experiments. Sec. 6 concludes.

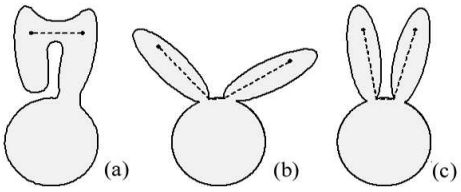

Figure 1: Three objects. The dashed lines denote shortest paths within shape boundary that connect landmark points.

2 Related Work

We will first discuss two works most closely related to this paper. One is the use of geodesic distances for bending invariant representation of 3D objects [5]. The other is the shape context [2] for 2D shapes. After that, some other work that handles part structures is discussed.

Our work is partly motivated by Elad and Kimmel's work [5] of using geodesic distances for 3D surface comparison through multidimensional scaling (MDS). Their key idea is that the geodesic distance is bending invariant, which is quite similar to the 2D articulation invariance in which we are interested. However, the direct counterpart of the geodesic distance in 2D reduces to the distances along the contours, which obviously is not useful. On the other hand, the inner-distance may also be extended to 3D shapes.

The *shape context* was introduced by Belongie et al. [2]. It describes the relative spatial distribution (distance and orientation) of landmark points around feature points. Combined with the thin-plate-spline [3], the shape context is demonstrated to be very discriminative. [13] extended the shape context by adding statistics of tangent vectors at landmark points. [22] suggested including a figural conti-

nuity constraint. [23] applied shape context and softassign [4] for fast and effective shape matching. In this paper, we extend the shape context by using the inner-distance to measure the spatial relations between points on shapes.

Roughly speaking, current methods for handling part structures fall into two categories, supervised and unsupervised. The supervised methods explicitly build models for part structures through training. Then the models are used for retrieval tasks. Examples can be found in [9, 6, 16]. The unsupervised methods do not depend on explicit part models. For example, [1] showed that similarities of part structure can be captured without the explicit computation of part structure. [20, 18] used shock graphs for shape comparison. Some other related work can be found in [8].

3 The Inner-Distance

Now we describe the inner-distance. Consider two points $x, y \in O$, where O is a shape defined as a connected and closed subset of R^2. The inner-distance between x, y, denoted as $d(x, y; O)$, is defined as the length of the shortest path connecting x and y within O. When O is convex, the inner-distance reduces to the Euclidean distance. However, this is not always true for non-convex shapes (e.g., Fig. 1). This suggests that the inner-distance is influenced by part structure to which the concavity of contours is closely related. In the following subsections, we will first show the inner-distance's insensitivity to articulation. Then, through examples and experiments, we show the inner-distance's ability to capture part structure.

3.1 A Model of Articulated Objects

Before discussing the articulation insensitivity of the inner-distance, we need to give a model of articulated objects. Intuitively, when a shape O is said to have articulated parts, it means 1) O can be decomposed into some parts, say, $O_1, O_2, ..., O_n$; 2) The junctions between parts are very small compared to the parts they connect; 3) The articulation on O as a transformation is rigid when limited to any part O_i, but can be non-rigid on the junctions; 4) The new shape O' achieved from articulation of O is again an articulated object and can articulate *back* to O.

Based on the above intuition, we define an articulated object $O \subset R^2$ of n parts together with an articulation f as: $O = \{\bigcup_{i=1}^{n} O_i\} \bigcup \{\bigcup_{i \neq j} J_{ij}\}$, where

1. $\forall i, 1 \leq i \leq n$, part $O_i \subset R^2$ is connected and closed, and $O_i \bigcap O_j = \emptyset, \forall i \neq j, 1 \leq i, j \leq n$.
2. $\forall i \neq j, 1 \leq i, j \leq n$, $J_{ij} \subset R^2$, connected and closed, is the junction between O_i and O_j. If there is no junction between O_i and O_j, then $J_{ij} = \emptyset$. Otherwise, $J_{ij} \bigcap O_i \neq \emptyset$, $J_{ij} \bigcap O_j \neq \emptyset$.
3. $diam(J_{ij}) \leq \epsilon$, where $diam(P)$ is defined as $diam(P) \doteq max_{x,y \in P}\{d(x, y; P)\}$ for a point set $P \subset R^2$. And $\epsilon \geq 0$ is very small compared to the size of the articulated parts. A special case is $\epsilon = 0$, which means all junctions degenerate to single points and O is called an *ideal articulated object*.

The articulation of object O is a one-to-one mapping f from O to $O' = f(O) \subset R^2$, such that:

1. O' is also an articulated object, with the decomposition $O' = \{\bigcup_{i=1}^{n} O'_i\} \bigcup \{\bigcup_{i \neq j} J'_{ij}\}$. Furthermore, $O'_i = f(O_i)$, $\forall i, 1 \leq i \leq n$ are parts of O' and $J'_{ij} = f(J_{ij})$, $\forall i \neq j, 1 \leq i, j \leq n$ are junctions in O'. This preserves the topology between the articulated parts.
2. f is rigid (rotation and translation only) on O_i, $\forall i, 1 \leq i \leq n$. This means inner-distances within each part will not change.

Notes: 1) In the above and following, we use notation $f(P) \doteq \{f(x) : x \in P\}$ for short. 2) It is obvious from the above definitions that f^{-1} is an articulation which maps O' to O, together with the parts and junctions.

Fig. 2 gives some examples of articulated shapes.

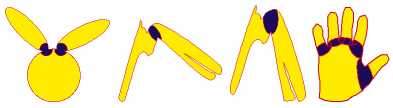

Figure 2: Examples of articulated objects. The separated yellow segments are parts and the blue ones are junctions.

3.2 Articulation Insensitivity of the Inner-Distance

We are interested in how the inner-distance varies under articulation. From Sec. 3.1 we know that changes of the inner-distance are due to deformations of junctions. Intuitively, this means the change is very small compared to the size of parts. Since most pairs of points have inner-distances comparable to the sizes of parts, the relative change of the inner-distances during articulation are small. This roughly explains why the inner-distances are articulation insensitive.

We use the following notation: 1) $C(x_1, x_2; P)$ denotes a shortest path from $x_1 \in P$ to $x_2 \in P$ for a closed and connected point set $P \subset R^2$ (so $d(x_1, x_2; P)$ is the length of $C(x_1, x_2; P)$). 2) $'$ indicates the image of a point or a point set under f, e.g., $P' \doteq f(P)$, $p' \doteq f(p)$. 3) "[" and "]" denote the concatenation of paths.

Let us first point out two facts about the inner-distance within a part or crossing a junction. Both facts are direct results from the definitions in sec. 3.1.

$$d(x, y; O_i) = d(x', y'; O'_i), \forall x, y \in O_i, 1 \leq i \leq n \quad (1)$$

$$|d(x, y; O) - d(x', y'; O')| \leq \epsilon, \forall x, y \in J_{ij}, \quad \forall i \neq j, 1 \leq i, j \leq n, J_{ij} \neq \emptyset \quad (2)$$

Note that (2) does not require the shortest path between x, y to lie within the junction J_{ij}. Now for general cases, $x, y \in O$, we have the following theorem:

Theorem: Let O be an articulated object and f be an articulation of O as defined in sec. 3.1. $\forall x, y \in O$, suppose the shortest path $C(x, y; O)$ goes through m different junctions in O and $C(x', y'; O')$ goes through m' different junctions in O', then

$$|d(x, y; O) - d(x', y'; O')| \leq max\{m, m'\}\epsilon \quad (3)$$

Proof: The proof uses the intuition mentioned above. First we decompose $C(x, y; O)$ into segments. Each segment is either within a part or across a junction. Then, applying (1) and (2) to each segment leads to the theorem. In the proof we assume all shortest paths are unique. This does not affect the result since only lengths of paths are concerned.

First, $C(x, y; O)$ is decomposed into l segments:

$$C(x,y;O)=[C(p_0,p_1;R_1),C(p_1,p_2;R_2),...,C(p_{l-1},p_l;R_l)]$$

by point sequence $p_0 p_1, ..., p_l$ and regions $R_1, ..., R_l$ achieved via the following steps:

1)$p_0 \leftarrow x$, $i \leftarrow 0$
2)WHILE $p_i \neq y$, DO
  $i \leftarrow i + 1$
  $R_i \leftarrow$ the region (a part or a junction) $C(x, y; O)$ enters after p_{i-1}
  IF $R_i = O_k$ for some k (R_i is a part):
   Set p_i as a point in O_k such that:
    1) $C(p_{i-1}, p_i; O_k) \subseteq C(x, y; O)$
    2) $C(x, y; O)$ enters a new region (a part or a junction) after p_i or terminate at p_i $(= y)$
  ELSE $R_i = J_{rs}$ for some r,s (R_i is a junction):
   Set p_i as the point in $J_{rs} \bigcap C(x, y; O)$ such that $C(x, y; O)$ never reenters J_{rs} after p_i.
  $R_i \leftarrow$ the union of all the parts and junctions $C(p_{i-1}, p_i; O)$ passes through (note $J_{rs} \subseteq R_i$).
3) $l \leftarrow i$

An example of this decomposition is shown in Fig. 3 (a). With this decomposition, $d(x, y; O)$ can be written as:

$$d(x, y; O) = \sum\nolimits_{1 \leq i \leq l} d(p_{i-1}, p_i; R_i)$$

Suppose m_1 of the segments cross junctions (i.e., not contained in any single part), then obviously $m_1 \leq m$.

In O', we construct a path from x' to y' corresponding to $C(x, y; O)$ as follows (e.g. Fig. 3 (b)):

$$\widetilde{C}(x',y';O')=[C(p'_0,p'_1;R'_1),C(p'_1,p'_2;R'_2),...,C(p'_{l-1},p'_l;R'_l)]$$

Denote $\widetilde{d}(x', y'; O')$ as the length of $\widetilde{C}(x', y'; O')$, it has the following property due to (1), (2):

$$|d(x, y; O) - \widetilde{d}(x', y'; O')| \leq m_1\epsilon \leq m\epsilon \quad (4)$$

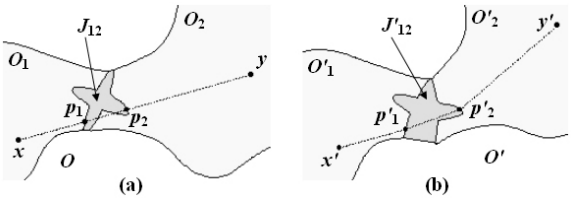

Figure 3: (a) Decomposition of $C(x, y; O)$ (the dashed line) with $x = p_0, p_1, p_2, p_3 = y$. Note that a segment can go through a junction more than once (e.g. $p_1 p_2$). (b) Construction of $\widetilde{C}(x', y'; O')$ in O' (the dashed line). Note that $\widetilde{C}(x', y'; O')$ is not the shortest path.

On the other hand, since O can be articulated from O' through f^{-1}, we can construct $\widetilde{C}(x, y; O)$ from $C(x', y'; O')$ in the same way as constructing $\widetilde{C}(x', y'; O')$ from $C(x, y; O)$. Then, similar to (4), there is

$$|d(x', y'; O') - \widetilde{d}(x, y; O)| \leq m'\epsilon \quad (5)$$

Combining (4) and (5),

$$\begin{aligned} d(x, y; O) - m'\epsilon \leq \widetilde{d}(x, y; O) - m'\epsilon \leq d(x', y'; O') \\ \leq \widetilde{d}(x', y'; O') \leq d(x, y; O) + m\epsilon \end{aligned}$$

This implies (3). #

From (3) we can make the following remarks concerning the changes of inner-distances under articulation:

1. The inner-distance is strictly invariant for ideal articulated objects. This is obvious since $\epsilon = 0$.
2. Since ϵ is very small, for most pairs of x, y, the relative change of inner-distance is very small. This means the inner-distance is insensitive to articulations.

3.3 Inner-Distances and Part Structures

In addition to articulation insensitivity, we believe that the inner-distance captures part structures better than the Euclidean distance. This is hard to prove because the definition of part structure remains unclear. Instead we support the idea with examples and experiments. Figures 1, 4 and 7 show examples where the inner-distance distinguishes shapes with parts while the Euclidean distance meets trouble.

During retrieval experiments using several shape databases, the inner-distance based descriptors all achieve excellent performance. Through observation we have found that some databases (e.g., MPEG7) are difficult for retrieval mainly due to the complex part structures in their shapes, though they have little articulation. These experiments show that the inner-distance is effective at capturing part structures (see Sec. 5.2 and Figures 7 and 10 for details).

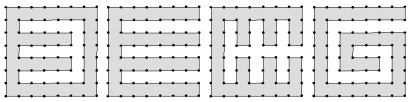

Figure 4: With the same sample points, the distributions of Euclidean distances between all pair of points are virtually indistinguishable for the four shapes, while the distributions of the inner-distances are quite different.

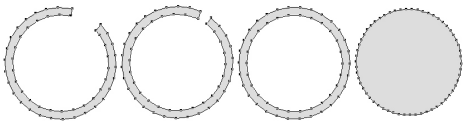

Figure 5: With about the same number of sample points, the four shapes are virtually indistinguishable using Euclidean distances, as in Fig. 4. However, their distributions of the inner-distances are quite different except for the first two shapes. Note: 1) None of the shapes has (explicit) parts. 2) More sample points will not affect the above statement.

Aside from part structures, examples in Fig. 5 show cases where the inner-distance can better capture the topology of shapes without parts. We expect further studies on the relationship between inner-distances and shape in the future.

3.4 Computing the Inner-Distance

A natural way to compute the inner-distance is using shortest path algorithms. This consists of two steps: 1) Build a graph on the sample points. For each pair of sample points x, y, if the line segment connecting x and y falls entirely within the object, then build an edge between x and y with the weight equal to the Euclidean distance $\|x - y\|$. 2) Apply a shortest path algorithm to the graph.

4 Matching and Retrieval

Now that the inner-distance is ready, we apply it to extend the shape context [2] for shape matching and comparison. There are other ways to use the inner-distance of course. One way is to apply MDS as in [5]. Another way is to use the *shape distribution* [14]. We choose shape context because it is highly discriminative and it is naturally extended with the inner-distance.

4.1 Previous Work on Shape Context

Given n sample points $x_1, x_2, ..., x_n$ on a shape, the shape context [2] at point x_i is defined as a histogram h_i of the relative coordinates of the remaining $n - 1$ points

$$h_i(k) = \#\{x_j : j \neq i, x_j - x_i \in bin(k)\} \quad (6)$$

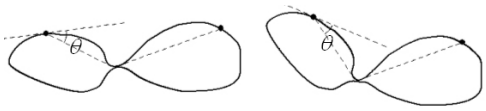

Figure 6: The inner-angle θ between two boundary points.

Where the bins uniformly divide the log-polar space. The distance between two shape context histograms is defined using the χ^2 statistic as in (9).

For shape comparison, [2] used a framework combining shape context and thin-plate-spline[3] (SC+TPS). Given the points on two shapes A and B, first the point correspondences are found through a weighted bipartite matching. Then, TPS is used iteratively to estimate the transformation between them. After that, the similarity D between A and B is measured as a weighted combination of three parts

$$D = 1.6D_{ac} + D_{sc} + 0.3D_{be} \quad (7)$$

Where D_{ac} measures the appearance difference. D_{be} measures the bending energy. The D_{sc} term, named the *shape context distance* in [2], measures the average distance between a point on A and its most similar counterpart on B (in the sense of (9)). The SC+TPS is shown to be very effective for shape matching by tests [2] on the MNIST database [11], MPEG7 CE-Shape-1, and others.

4.2 Extension of Shape Context

To extend the shape context defined in (6), we redefine the bins with the inner-distance. The Euclidean distance is directly replaced by the inner-distance. For the orientation bins, the relative orientation between two points can be defined as the tangential direction at the starting point of the shortest path. However, this tangential direction *is* sensitive to articulation. Fortunately, for boundary points, the angle between the contour tangent at the start point and the tangential direction of the shortest path from it is insensitive to articulation (invariant to ideal articulation). We call this angle the *inner-angle* (e.g., see Fig. 6) and use it for the orientation bins. This is equivalent to using the relative frame, i.e., the local coordinate system is rotated to align with the tangent at the sample point. This is suggested in [2] to get rotation invariance. Fig. 7 shows examples of the shape context computed by the two different methods.

In the following the shape context in [2] is called SC and the extension with inner-distance IDSC.

4.3 Shape Matching Through Dynamic Programming

We are interested in contour matching in this paper. The matching problem is formulated as follows: Given two

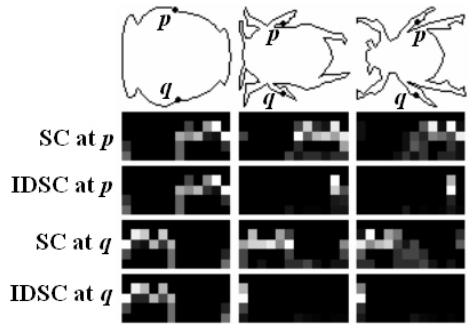

Figure 7: Shape context (SC) and inner-distance shape context (IDSC). The top row shows three objects from the MPEG7 database (Sec. 5.2), with two marked points p, q on each shape. The next rows show (from top to bottom), the SC at p, the IDSC at p, the SC at q, the IDSC at q. Both the SC and the IDSC use local relative frames. In the histograms, the x axis denotes the orientation bins and the y axis denotes log distance bins.

shapes A and B, describe them by point sequences on their contour, say, $p_1p_2...p_n$ for A with n points, and $q_1q_2...q_m$ for B with m points. Without loss of generality, assume $n \geq m$. The matching π between A and B is a mapping from $1, 2, ..., n$ to $1, 2, ..., m$, where p_i is matched to $q_{\pi(i)}$ if $\pi(i) \neq 0$ and otherwise left unmatched. π should minimize the match cost $H(\pi)$ defined as

$$H(\pi) = \sum\nolimits_{1 \leq i \leq n} C(i, \pi(i)) \tag{8}$$

where $C(i, 0) = \tau$ is the penalty for leaving p_i unmatched, and for $1 \leq j \leq m$, $C(i, j)$ is the cost of matching p_i to q_j. This is measured using the χ^2 statistic as in [2]

$$C(i,j) \equiv \frac{1}{2} \sum\nolimits_{1 \leq k \leq K} \frac{[h_{A,i}(k) - h_{B,j}(k)]^2}{h_{A,i}(k) + h_{B,j}(k)} \tag{9}$$

Here $h_{A,i}$ and $h_{B,j}$ are the shape context histograms of p_i and q_j respectively, and K is the number of histogram bins.

Since the contours provide orderings for the point sequences $p_1p_2...p_n$ and $q_1q_2...q_m$, it is natural to restrict the matching π with this order. To this end, we use dynamic programming to solve the matching problem. Dynamic programming is widely used for contour matching. Details can be found in [1, 15] for example.

By default, the above method assumes the two contours are already aligned at their start and end points. Without this assumption, one simple solution is to try different alignments at all points on the first contour and choose the best one. The problem with this solution is that it raises the matching complexity from $O(n^2)$ to $O(n^3)$. Fortunately, for the comparison problem, it is often sufficient to try aligning a fixed number of points, say, n_s points. Usually n_s is much smaller than m and n (with $n, m = 100$, our experiments show that $n_s = 4$ or 8 is good enough and larger n_s does not demonstrate significant improvement). The complexity is still $O(n_s n^2) = O(n^2)$.

Bipartite graph matching is used in [2] to find point correspondence π. Bipartite matching is more general since it minimizes the matching cost (8) without additional constraints. For example, it works when there is no ordering constraint on the sample points (while dynamic programming is not applicable). For sequence points along silhouettes, however, dynamic programming matching is more efficient and accurate since it uses the ordering information.

4.4 Shape Distances

Once the matching is found, we use the matching cost $H(\pi)$ as in (8) to measure the similarity between shapes. One thing to mention is that dynamic programming is also suitable for shape context. In the following, we use IDSC+DP to denote the method of using dynamic programming matching with the IDSC, and use SC+DP for the similar method with the SC.

In addition to the excellent performance demonstrated in the experiments, the IDSC+DP framework is simpler than the SC+TPS framework (7) [2]. First, besides the size of shape context bins, IDSC+DP has only two parameters to tune: 1) The penalty τ for a point with no matching, usually set to 0.3, and 2) The number of start points n_s for different alignments during the DP matching, usually set to 4 or 8. Second, IDSC+DP is easy to implement, since it does not require the appearance and transformation model as well as the iteration and outlier control. Furthermore, the DP matching is faster than bipartite matching, which is important for shape retrieving in large shape databases.

The time complexity of the IDSC+DP consists of three parts. First, the computation of inner-distances can be achieved in $O(n^3)$ with Johnson or Floyd-Warshall's shortest path algorithms, where n is the number of sample points. Second, the construction of the IDSC histogram takes $O(n^2)$. Third, the DP matching costs $O(n^2)$, and only this part is required for all pairs of shapes. In our experiment using partly optimized Matlab code on a regular Pentium IV 2.8G PC, a single comparison of two shapes with $n = 100$ takes about 0.31 second.

5 Experiments

5.1 Articulated Database

To show the articulation insensitivity of the inner-distance, we test the proposed method IDSC+DP on an articulated shape data set we collected. The dataset contains 40 images from 8 different objects. Each object has 5 images articulated to different degrees (see Fig. 8). The dataset is

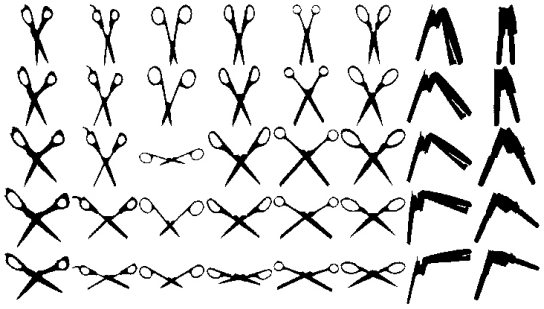

Figure 8: Articulate shape database. This dataset contains 40 images from 8 objects with articulation. Each column contains five images from the same object.

Table 1: Retrieval result on the articulate dataset.

Distance Type	Top 1	Top 2	Top 3	Top 4
SC+DP	20/40	10/40	11/40	5/40
IDSC+DP	**40/40**	**34/40**	**35/40**	**27/40**

very challenging because of the similarity between different objects (especially the scissors). The holes of the scissors make the problem even more difficult.

For each image, we sample 200 points along its outer contour. For the SC and IDSC, 5 log-distance bins and 12 orientation bins are used. Since all the objects are at the same orientation, we align the contours by forcing them to start from the bottom-left points and then set $n_s = 1$ for DP matching. For comparison, we also applied the SC+DP method with the same parameters.

For each image, the 4 most similar matches are chosen from other images in the dataset. The retrieval result is summarized as the number of 1st, 2nd, 3rd and 4th most similar matches that come from the correct object. Table 1 shows the retrieval results. It demonstrates that our method is very effective for objects with articulated parts, while the shape context is not very suitable for this data set.

5.2 MPEG7 Shape Database

The widely tested MPEG7 CE-Shape-1 [10] database consists of 1400 silhouette images from 70 classes. Each class has 20 different shapes (see Fig. 9 for some examples). The recognition rate is measured by the so called Bullseye test: For every image in the database, it is matched with all other images and the top 40 most similar candidates are counted. At most 20 of the 40 candidates are correct hits. The score of the test is the ratio of the number of correct hits of all images to the best possible number of hits (which is 20x1400).

In our experiment, we use 5 distance bins and 12 orientation bins as in [2], but only 100 sample points (300 were used in [2]) on each contour. 8 different start points

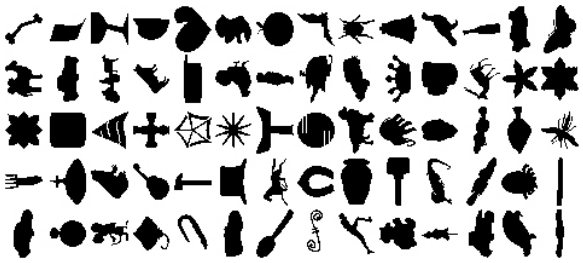

Figure 9: Typical shape images from the MPEG7 CE-Shape-1, one image from each class.

Table 2: Retrieval rate (bullseye) of different methods for the MPEG7 CE-Shape-1.

Algorithm	CSS [12]	Visual Parts[10]	SC+TPS[2]
Score	75.44%	76.45%	76.51%
Algorithm	Curve Edit[17]	Gen. Model[23]	**IDSC+DP**
Score	78.17%	80.03%	**85.40%**

($n_s = 8$) are used in the DP matching and the penalty factor τ is set to be 0.3. To handle mirrored shapes, we compare two point sequences (corresponding to shapes) with the original order and reversed order. Table 2 lists reported results from different algorithms. It shows that our algorithm outperforms all the alternatives. The speed of our algorithm is in the same range as those of shape contexts [2], curve edit distance [17] and generative model [23].

To help understand this performance, we did two other experiments in the same settings where the only difference is the descriptors used: one uses SC, another IDSC. The parameters in both experiments are: 64 sample points on each silhouette, 8 distance bins and 8 orientation bins. To avoid the matching effect, shapes are compared using the simple shape context distance measure D_{sc} (see Sec. 4.1 or [2]). The Bullseye score with SC is 64.59%, while IDSC get a higher score of 68.83%. Fig. 10 shows some retrieval results, where we see that the IDSC is good for objects with parts while the SC favors global similarities. Examination of the MPEG7 data set shows that the complexity of shapes are mainly due to the part structures but not articulations, so the good performance of IDSC shows that the inner-distance is more effective at capturing part structures.

5.3 Kimia's database

The IDSC+DP is tested on two shape databases provided by Kimia's group [19, 18]. The first database [19] contains 25 images from 5 categories (Fig. 11). It has been tested by [2, 19, 7]. In our experiment, 100 sample points are used for each silhouette, 5 distance bins and 12 orientation bins are used in IDSC, and $n_s = 4, \tau = 0.3$ are used in DP matching. The retrieval result is summarized as the number of 1st,

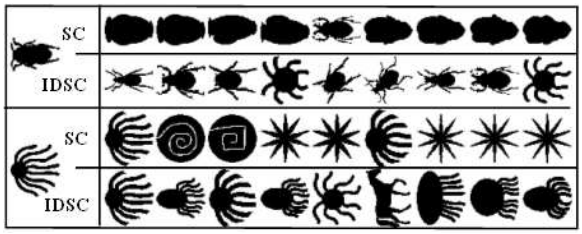

Figure 10: Two retrieval examples for comparing SC and IDSC on the MPEG7 data set. The left column show two shapes to be retrieved: a beetle and an octopus. The four right rows show the top 1 to 9 matches, from top to bottom: SC and IDSC for the beetle, SC and IDSC for the octopus.

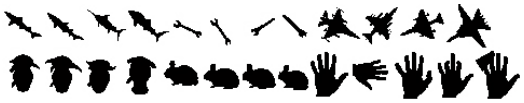

Figure 11: Kimia dataset 1 [19]: this dataset contains 25 instances from 5 categories.

Table 3: Retrieval result on Kimia dataset 1 [19] (Fig. 11).

Method	Top 1	Top 2	Top 3
Sharvit et. al [19]	23/25	21/25	20/25
Gdalyahu and Weinshall[7]	25/25	21/25	19/25
Belongie et. al [2]	25/25	24/25	22/25
IDSC+DP	**25/25**	**24/25**	**25/25**

Table 4: Retrieval result on Kimia dataset 2[18] (Fig. 12). Gen. model is due to [23] and shock edit is due to [18].

Algorithm	1st	2nd	3rd	4th	5th	6th	7th	8th	9th	10th
SC [18]	97	91	88	85	84	77	75	66	56	37
Gen. Model	99	97	99	98	96	96	94	83	75	48
Shock Edit	99	99	99	98	98	97	96	95	93	82
IDSC+DP	**99**	**99**	**99**	**98**	**98**	**97**	**97**	**98**	**94**	**79**

2nd and 3rd closest matches that fall into the correct category. Our result is 25/25,24/25,25/25, which outperforms the other three reported results shown in Table 3.

The second database [18] contains 99 images from 9 categories (Fig. 12) and has been tested by [18, 23]. In our experiment, 300 sample points are used for silhouettes, 8 distance bins and 12 orientation bins are used in IDSC, and $n_s = 4, \tau = 0.3$ are used in DP matching. Similar to results described above, the retrieval result is summarized as the number of top 1 to top 10 closest matches (the best possible result for each of them are 99). Table 4 lists the numbers of correct matches of several methods, which shows that our approach performs a little better than others.

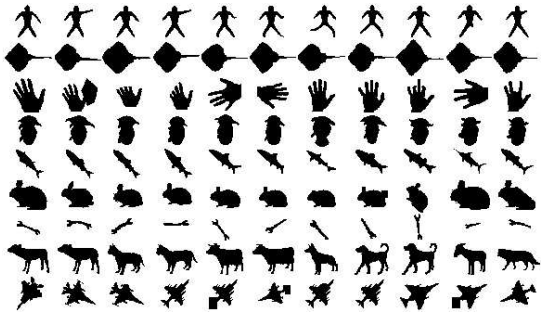

Figure 12: Kimia dataset 2 [18]: this dataset contains 99 instances from 9 categories.

Figure 13: Typical images from Swedish leaf data base, one image per species. Note that some species are quite similar, e.g. the 1st, 3rd and 9th species.

5.4 Swedish Leaf Database

Recently, foliage image retrieval has started to attract research efforts in computer vision and related areas. The large variation of leaf shapes and texture make the problem very challenging. We use the Swedish leaf dataset from a leaf classification project at Linköping University and the Swedish Museum of Natural History [21]. The dataset contains isolated leaves from 15 different Swedish tree species, with 75 leaves per species. Fig. 13 shows some silhouette examples. Some initial classification work has been done in [21] by combining simple features like moments, area and curvature etc. Using 25 training samples and 50 testing samples per species, an average classification rate of 82% is reported. We tested with Fourier descriptors, SC+DP and IDSC+DP with the same size of training and testing set and 128 points on each silhouette. For SC and IDSC, we use 8 log-distance bins and 12 orientation bins; for DP matching, we set $n_s = 1$ and $\tau = 0.3$. With 1-nearest-neighbor, the classification rates are 89.60% using Fourier descriptors, 88.12% using SC+DP and 94.13% using IDSC+DP.

5.5 Human body matching

In this experiment, we demonstrate the potential for using the proposed method on human body matching, which is important in human motion analysis. The dataset is a human motion sequence from a stationary camera (from the Keck lab). Silhouettes are extracted with background subtraction. Our task is to match the silhouettes from different frames. For adjacent frames, IDSC+DP performs very well, as demonstrated in the left of Fig. 14. For two silhouettes

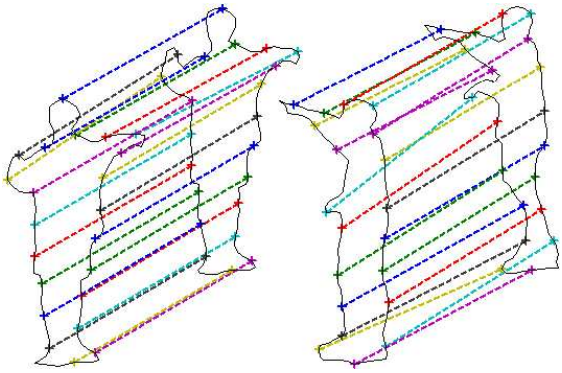

Figure 14: Human silhouettes matching. Left: between adjacent frames. Right: silhouettes separated by 20 frames. Only half of the matched pairs are shown for illustration.

separated by 20 frames, the articulation turns out to be large and the matching becomes challenging. The IDSC+DP also gives promising result, see the right part in Fig. 14 for example.

6 Conclusions

We proposed using the inner-distance to build shape descriptors. We show that the inner-distance is articulation insensitive and is good for complicated shapes with part structures. We extended the shape context with the inner-distance to form a new descriptor, and designed a dynamic programming based method for shape matching and comparison. In retrieval experiments on several data sets, our approach demonstrated excellent retrieval results in comparison with several other algorithms. In addition, the test on sequential human silhouettes matching shows the potential of using inner-distances in tracking problems.

Acknowledgements: This work is supported by NSF (ITR-03258670325867). We would like to thank B. Kimia for the Kimia data set, O. Söderkvist for the Swedish leaf data, Z. Yue and Y. Ran for the Keck sequence.

References

[1] R. Basri, L. Costa, D. Geiger, and D. Jacobs, "Determining the Similarity of Deformable Shapes", *Vision Research* 38:2365-2385, 1998.

[2] S. Belongie, J. Malik and J. Puzicha. "Shape Matching and Object Recognition Using Shape Context," *PAMI*, 24(24):509-522, 2002.

[3] F. Bookstein. "Principal Warps: Thin-Plate-Splines and Decomposition of Deformations", *PAMI*, 11(6):567-585, 1989.

[4] H. Chui and A. Rangarajan. "A New Point Matching Algorithm for Non-rigid Registration", *Computer Vision and Image Understanding*, 89(2-3):114-141, 2003.

[5] A. Elad(Elbaz) and R. Kimmel. "On Bending Invariant Signatures for Surfaces", *PAMI*, 25(10):1285-1295, 2003.

[6] P. F. Felzenszwalb and D. P. Huttenlocher. "Pictorial Structures for Object Recognition", *IJCV*, 61(1):55-79, 2005.

[7] Y. Gdalyahu and D. Weinshall. "Flexible Syntactic Matching of Curves and Its Application to Automatic Hierarchical Classification of Silhouettes", *PAMI*, 21(12):1312-1328, 1999.

[8] L. Gorelick, M. Galun, E. Sharon, R. Basri and A. Brandt, "Shape Representation and Classification Using the Poisson Equation", CVPR, 61-67, 2004.

[9] W. E. L. Grimson, "Object Recognition by Computer: The Role of Geometric Constraints", MIT Press, Cambridge, MA, 1990.

[10] L. J. Latecki, R. Lakamper, and U. Eckhardt, "Shape Descriptors for Non-rigid Shapes with a Single Closed Contour", CVPR, I:424-429, 2000.

[11] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. "Gradient-based Learning Applied to Document Recognition", *Proceedings of the IEEE*, 86(11):2278-2324, 1998.

[12] F. Mokhtarian, S. Abbasi and J. Kittler. "Efficient and Robust Retrieval by Shape Content through Curvature Scale Space," in A. W. M. Smeulders and R. Jain, editors, *Image Databases and Multi-Media Search*, 51-58, World Scientific, 1997.

[13] G. Mori and J. Malik, "Recognizing Objects in Adversarial Clutter: Breaking a Visual CAPTCHA", CVPR, I:1063-6919, 2003.

[14] R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin. "Shape Distributions", *ACM Trans. on Graphics*, 21(4):807-832, 2002.

[15] E. G. M. Petrakis, A. Diplaros and E. Milios. "Matching and Retrieval of Distorted and Occluded Shapes Using Dynamic Programming", *PAMI*, 24(11):1501-1516, 2002.

[16] H. Schneiderman and T. Kanade. "Object Detection Using the Statistics of Parts", *IJCV*, 56(3):151-177, 2004.

[17] T. B. Sebastian, P. N. Klein and B. B. Kimia. "On Aligning Curves", *PAMI*, 25(1):116-125, 2003.

[18] T. B. Sebastian, P. N. Klein and B. B. Kimia. "Recognition of Shapes by Editing Their Shock Graphs", *PAMI*, 26(5):550-571, 2004.

[19] D. Sharvit J. Chan, H. Tek, and B. Kimia. "Symmetry-based Indexing of Image Database", *J. Visual Communication and Image Representation*, 9(4):366-380, 1998.

[20] K. Siddiqi, A. Shokoufandeh, S. J. Dickinson and S. W. Zucker. "Shock Graphs and Shape Matching", *IJCV*, 35(1):13-32, 1999.

[21] O. Söderkvist. "Computer Vision Classification of Leaves from Swedish Trees", Master Thesis, Linköping Univ. 2001.

[22] A. Thayananthan, B. Stenger, P. H. S. Torr and R. Cipolla, "Shape Context and Chamfer Matching in Cluttered Scenes", CVPR, I:1063-6919, 2003.

[23] Z. Tu and A. L. Yuille. "Shape Matching and Recognition-Using Generative Models and Informative Features", *ECCV*, 3:95-209, 2004.