A combined distance measure for 2D shape matching

Geetha Ramachandran
Institute of Telecommunications
Vienna University of Technology
Gusshausstrasse 25/389, 1040 Vienna Austria
geetha.ramachandran@tuwien.ac.at

***Abstract*—We present a method for 2D shape matching using a combination of distance functions and discrete curvature. The eccentricity transform computes the longest geodesic distance across the object. This transform is invariant to translation and rotation. The maximal eccentricity points yield diameters across the image. We compute the Euclidean distances from the boundary to the diameter to characterize the curvature of the shape. Our shape descriptor is comprised of the best matches retrieved from the normalized histogram of the eccentricities, the Hausdorff distance between the set of distances to the diameter and a measure of the number of points lying on either side of the diameter along with the peak values. We evaluate this descriptor on 2D image databases consisting of rigid and articulated shapes by ranking the number of matches. In almost all cases, the shapes are matched with at least one shape from the same class.**

I. Introduction

Shape matching in 2D images plays a key role in object recognition and tracking. Shapes are matched based on their similarity, the similarity being defined by the unique signature each shape provides. The key challenge in shape matching is, hence, to derive such a signature from the image of an object. This in turn helps categorize or differentiate between categories of shapes. Thus the signature that defines the shape should be invariant to scale, rotation and to a certain degree perturbations caused by noise.

Shape matching for articulated objects is more complicated than rigid objects. An articulated object is defined as a union of rigid parts and joints. A transformation that is rigid with respect to the parts, but non-rigid when the whole object is taken into consideration is an articulation. An example of this is the dataset shown in Figure 5. Shape matching in these instances entails identifying articulations of an object to belong to the same class. The same object may have different shapes in different poses. Hence, the signature needs to be consistent across these articulations.

There exists a vast amount of literature on shape matching using feature selection. These methods range from applying textural features on satellite imagery [3] to applying deep learning to features for image classification[10]. Distance functions provide an alternative method to this. One of the first instance of a distance function on images is the distance transform proposed in [9] that associates to each point the length of the shortest path to the closest boundary point in the shape. The weakness of this method arises from the locality of the distance measurements and thus its susceptibility to error in case of slight perturbations in the shape. The eccentricity [7] measure, instead, looks at the longest geodesic distance between two points on an object and these remain the same irrespective of the transformation, unless there occurs a morphological change. It is also invariant to translation and rotation. This provides a very strong basis for a shape descriptor.

The problem of articulated 2D shape matching in binary images using the eccentricity transform was studied in [5]. The eccentricity transform of a shape assigns to each point of the shape the distance to the point farthest away. The eccentricities are measured based on geodesic distances, making it robust against articulation. Two measures were combined here to match the shapes: the histogram of the eccentricity values and a histogram of the connected components of the discrete level-sets of the eccentricity transform. This method proves successful, but faces drawbacks because of the low dimensionality of the histograms. Also, another key factor that is overlooked here is the curvature of the shapes.

To overcome the lower dimensionality problem, we factor in the curvature of the image. We apply the chords-to-points distance accumulation measure on the shape. This was introduced as a means of measuring planar curvature in [2]. Here, the distance from all the points on the image to multiple chords are accumulated. However, instead of applying multiple chords to measure the curvature, we identify the maximum eccentricity values and plot diameters over the image. The set of distances to these diameters are used as a measure to compare shapes. In addition to this, we identify the following key attributes namely: the number of points lying on each side of the diameter and the distances to the peaks on each side. This gives us a clear picture of how the shape is formed. We combine these measures to successfully match images belonging to the same class with each other.

In Section II, we explain in more detail the ideas we apply in this paper. Following this in Section III, we explain how we implement and combine these concepts to classify images. We present our evaluation of our method in Section IV and our conclusions follow in Section V.

II. Related Concepts

In graph theory, for a connected graph, the *eccentricity* of a vertex is the measure of the shortest length of the paths to any other vertex in the graph. For a shape S, if d^s is the geodesic distance, then the eccentricity of a point $\mathbf{p} \in S$ is defined as

$$ECC(S, \mathbf{p}) = \max(d^s(\mathbf{p}, \mathbf{q}))|q \in S \quad (1)$$

978-1-4799-7186-2/15/$31.00 ©2015 IEEE

The eccentricity of $\mathbf{p}$ is the length of the longest geodesic that has $\mathbf{p}$ as one of its end points in the same connected region.

The eccentricity transform was introduced in [7] as a means of applying the eccentricity as a distance transform over a graph or a 2D image. The application of the transform associates with each point $\mathbf{p}$ in the shape its eccentricity. The maximal eccentricity points form the diameter of the shape. It is shown in [7] that the transform performs well in the presence of salt and pepper noise.

The chord-to-point distance accumulation (CPDA) is a discrete curvature measure. It applies the idea that the flatness of a curve can be measured by comparing a segment of the curve with a straight line. The measure accumulates the distance from a point in the boundary to a chord that is specified by moving points on the boundary. Consider the boundary $B = \{p_0, p_1, ..., p_{N-1}\}$ of an object. A line l_i can be defined from a point p_i to p_{i+l}. The perpendicular distance from a point $p_k \in P$ to l_i defines the distance to L_i. The distance is positive or negative depending on where p_k is located with respect to l_i. The distance accumulation for p_k and a chord of length L is the sum h_L of the distances as i moves from $k - L$ to k.

$$h_L(k) = \sum_{i=k-L}^{k} D_{ik} \tag{2}$$

The scale-space image of the distance accumulation show that the locations of the zero crossings are stable through a range of values of L.

III. Proposed Method

We apply the concepts described above to derive a shape descriptor.

A. Computing the eccentricity transform

As described in Section II, the eccentricity is the measure of the longest of the shortest paths from one vertex to another. A common approach to the computation of the shortest path over a graph is the Dijkstra's shortest path algorithm. From a given point, the distances to all the unvisited adjacent vertices is computed. If any of the distances is lesser than previously known, then the vertex is updated to reflect this new information. This process is continued until all vertices are updated.

To apply the Dijkstra's shortest path to a 2D image, every point on the image is assumed to be connected to its four adjacent neighbors. As shown in Figure 1, to move from point A to point B, the shortest path is two-step. This is inconsistent with the Euclidean distance between the two points. This inconsistency can be resolved by applying the Fast Marching methods for distance computation [6]. Fast marching resolves the distance computation by simulating wavefront propagation. The key difference between fast marching and Dijkstra's method lie in the 'update' step. Instead of proceeding step-by-step, fast marching looks at the closest Euclidean distance points. The details of this method can be found in [1]. Here, since we are dealing with 2D images, assuming uniform sampling, we safely approximate fast marching distance computation by applying 8-connectivity and using Dijkstra's shortest path algorithm. This is illustrated in Figure1. As shown, we can move from point A to B in a single step. The distance thus computed is equivalent to the Euclidean distance between the points.

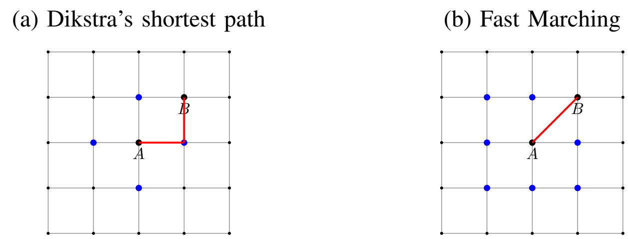

Fig. 1: Fast Marching path traversal can be approximated using an 8-connected graph

The eccentricity transform ideally requires the computation of the eccentricity at every point of the image. This is a naive approach, and is computationally expensive. An iterative approach is applied in [5]. The eccentricity is first computed at any random point. From the furthest end point of the path, the eccentricity is re-computed so that a diameter is established. All the points in the shape are now assigned the longest of the shortest paths to either of the end points of the diameter. The centre points of the shape are estimated to be the points with minimum eccentricity values. Eccentricity local maxima are computed from each of these central points and the distances to all the points in the image are calculated until no local maxima remain. This method is further refined to grow clusters to define eccentricity regions.

In comparison, here, we start with the assumption that the eccentric points of the shape lie on the boundary. This follows from the properties of eccentricity listed in [4]. For a simply connected shape S, all the eccentric points are located on the boundary. There can occur shapes where this is not the case. For instance, in case of shapes with holes, if the shape has more than one hole, eccentric points could exist inside the shape. For the datasets we consider, we are able to obtain good results despite this assumption.

Since, the shape is assumed to be 8-connected, the boundary is identified as any point with less than 8 adjacent neighbors. The image is assumed to be uniformly sampled, hence each edge is given equal weights. The eccentricity for all points is thus computed. The points with maximum eccentricity help identify the diameters across the image.

B. Forming the shape descriptor

To match shapes, a unique shape descriptor is created. The shape descriptor is formed of three components: the eccentricity histogram $\mathbf{h}$, a Hausdorff distance comparator $\mathbf{c}$ and a set of attributes $\mathbf{a}$.

The eccentricity histogram $\mathbf{h}$ for a shape S is defined as follows [5],$\forall i = 1, ..., k_h$:

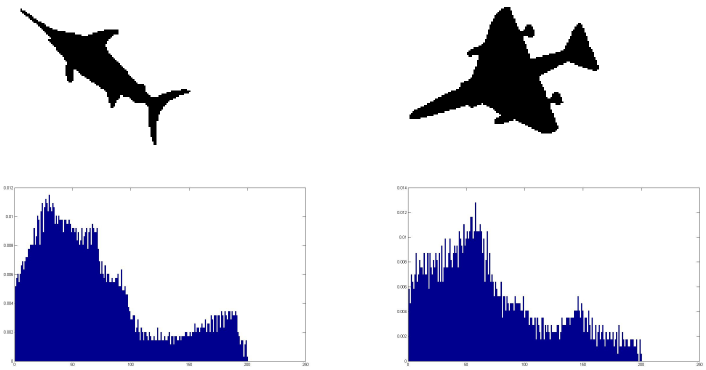

Fig. 2: Eccentricity histograms for an image of the category 'fish' and 'airplanes'. The histograms are similar in structure, hence making it difficult to separate the categories.

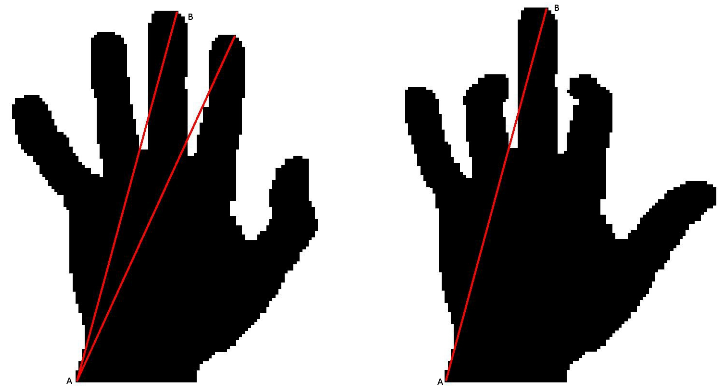

Fig. 3: The first hand shape has two diameters, while the second shape has only one. Using the minimum Hausdorff distance, the diameter AB is found to match.

$$\mathbf{h}(S,i) = \frac{1}{|S|}\#\left\{\mathbf{p} \in S | \frac{i-1}{k_h} \leqslant \frac{ECC(S,\mathbf{p})-m}{M-m} < \frac{i}{k_h}\right\} \tag{3}$$

where $|S|$ is the number of pixels in S and m and M are the smallest and largest of the eccentricity values over S. This histogram gives an idea of how the eccentricities are distributed over the image. The histogram gives a fair set of matches for the images. We use the L2-norm to compare the histograms. However, different images can have identical eccentricity distributions, as shown in Figure 2. It is a linear descriptor and fails to give an idea of the structure of the image.

To compensate for this, we modify upon the chords-to-points distance accumulation method to form a second component $\mathbf{c}$ in our descriptor. The points with maximum eccentricity are selected to form one or more diameters of the image. The distance is calculated from all the boundary points of the image to the each of the diameters. For an image I_k, this forms a set of distance measures $D_k(x,b)$ where x is the number of diameters plotted across the image and b is the number of boundary points. There exist multiple diameters due to one of the following reasons. First, the object is symmetrical across multiple axes. Second, there lie adjacent points near the diameter extremities which are at the same distance from each other. Third, there could be noise in the image. Hence, we take into consideration all possible diameters.

To compute the best match, we calculate the minimum of the Hausdorff distances among these sets of diameters. The Hausdorff distance between two point sets $A = a_0, a_1..., a_p$ and $B = b_0, b_1...b_q$ is defined as

$$H(A,B) = max\left(h(A,B), h(B,A)\right) \tag{4}$$

where

$$h(A,B) = \max_{a\in A}\min_{b\in B} \parallel a-b \parallel \tag{5}$$

In this case, $\parallel \cdot \parallel$ represents the Euclidean distance. $h(A,B)$ identifies the point $a \in A$ that is farthest away from any point in B and measures the distance from to the nearest point. Essentially, it identifies the most mismatched point in A. $h(B,A)$ identifies the most mismatched point in B. Thus, $H(A,B)$ measures the degree of mismatch between the two finite point sets.

We use this measure to identify the diameters that divide the images consistently. This is indicated in Figure 3. Here, the minimum of the Hausdorff distances between the sets of measured distances from the boundary to the diameters is given by AB, that is consistent across the both the images of the deformed hands.

The third component of the shape descriptor consists of a set of attributes that gives us an idea of how the points are distributed across the chosen diameter. The first and second attributes are the number of points above and below the diameter - indicated by the positive and negative distances from the boundary. The third and fourth attributes indicate the maximum and minimum distances from the diameter. This is used to indicate the highest peak on both sides of the diameter, thus giving an idea of the scale of the image in a direction orthogonal to the diameter. We calculate the L2-norm between the sets of attributes. These three components provide complementary matches. Once we have these three values, we use the measure that provides the highest number of matches in a given class.

Our results on the Kimia 25 dataset are shown in Figure 4. Here, the first column is the test image and is being matched against the rest of the images in the dataset.

IV. EVALUATION

We conducted experiments on three datasets: Kimia 25 [12], Kimia 99 [11] and the Ling articulated dataset [8]. The Kimia 25 dataset is comprised of five classes of four images each and one class with five images. Hence $q = 25$ and $l_{max} = 6$. The Kimia 99 dataset consists of nine classes of eleven images each. The Ling dataset, Figure 5 consists of a set of articulated objects. Each column indicates a different articulation of 8 classes of objects.

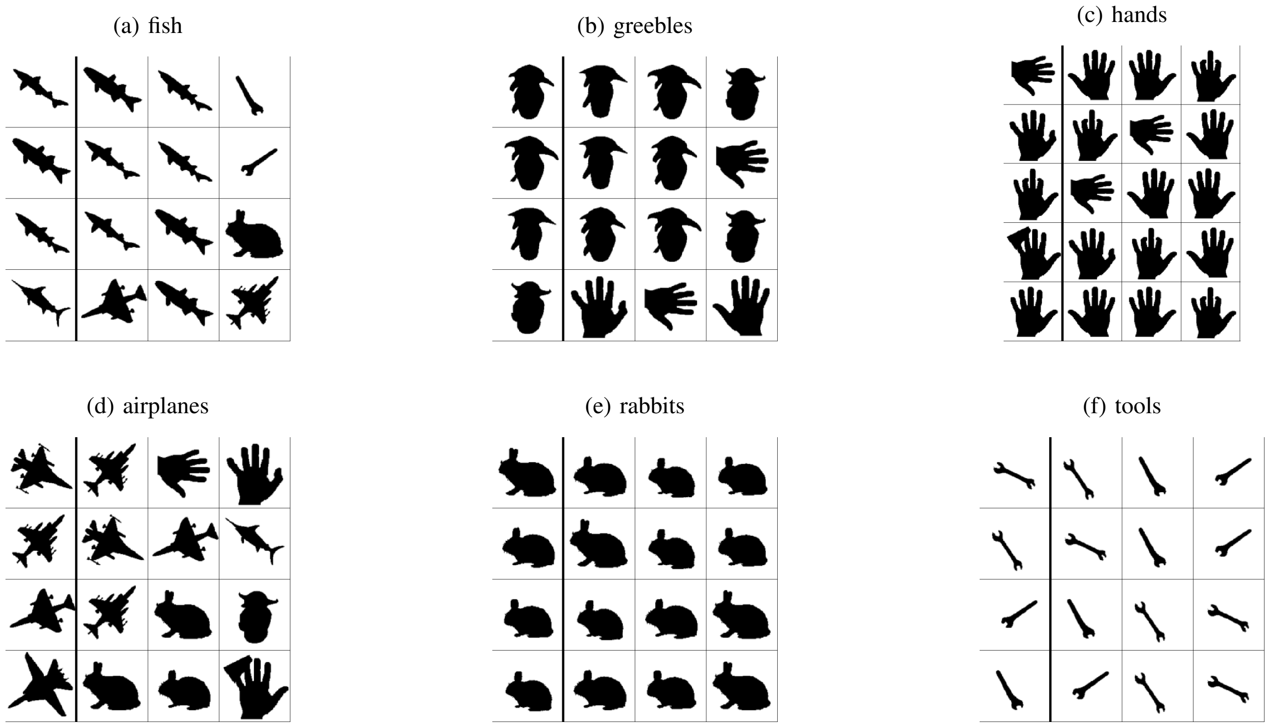

Fig. 4: We match the images in the first column against all the images in the dataset. The next three columns indicate the best matches retrieved, omitting the test image itself.

Algorithm	r=1	r=2	r=3
ECCobj2D, s only	18	19	17
ECCobj2D, h only	20	16	14
ECCobj2D	22	20	17
ECCCurv, h only	15	13	10
ECCCurv, c only	19	15	9
ECCCurv, a only	21	18	6
ECCCurv	22	20	15

TABLE I: Match results on the Kimia 25 dataset.

Algorithm	r=1	r=2	r=3	r=4	r=5	r=6
ECCobj2D, s only	84	68	65	67	56	57
ECCobj2D, h only	87	74	66	64	49	52
ECCobj2D	94	85	81	73	81	73
ECCCurv, h only	72	48	44	39	30	22
ECCCurv, c only	61	51	40	31	18	14
ECCCurv, a only	73	55	41	24	15	12
ECCCurv	93	84	76	68	49	38

TABLE II: Match results on the Kimia 99 dataset.

A shape database is composed of q shapes. Each shape S_i in the database has a label L_i that indicates the class to which the image belongs. The purpose of a shape matching algorithm is to assign to each shape the best matches. The efficiency of matching algorithms is measured by the number of correct matches:

$$Match_r(\phi) = \sum_{i=1}^{q} 1_{L(\phi_i(r))=L(i)} \leqslant q \tag{6}$$

$Match_r(\phi)$ gives the number of matches retrieved for a given category. In the tables $r = 1$ shows the number of instances where the first match retrieved is of the same category, omitting the image itself, $r = 2$ gives the number of instances where the first and second matches were of the same category and so on.

Algorithm	r=1	r=2	r=3	r=4
ECCobj2D	40	33	29	22
ID-shape context + DP [8]	40	34	35	27
ECCCurv	40	29	15	5

TABLE III: Match results on the Ling dataset. We also show the numbers from the original work done on the dataset.

V. Conclusions

In this paper, we have presented a method for 2D shape matching for articulated shapes. We use the eccentricity transform, which is based on the measurement of maximal geodesic distances in a shape. The shape descriptor is composed of three parts: a normalized histogram of the eccentricity values, the Hausdorff distance between the sets of distances measured from the boundary of the shape to the diameter and a measure

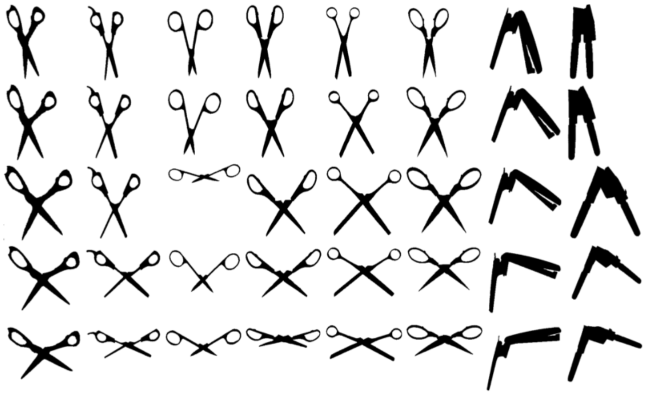

Fig. 5: Articulated Ling dataset [8]

of the number of points lying on either side of the diameter along with the peak values. This combined measure gives us information about the connectivity, compactness and structure of the image.

We also showed that using 8-connectivity for establishing the adjacency graph combined with Dijkstra's shortest path algorithm serves as a good approximation to the Fast Marching method for 2D images, under the assumption of uniform sampling. We provide experimental results to show that our method provides results that are comparable to similar approaches on well known image databases.

An extension of this method is to apply the method to 3D single and multiple object scenes. However then, Fast Marching cannot be approximated in instances where the triangular meshes are not uniform. We will also investigate extending the chords-to-points distance accumulation to 3D.

REFERENCES

[1] Bronstein, A., Bronstein, M., and Kimmel, R. (2008). *Numerical Geometry of Non-Rigid Shapes*. Springer Publishing Company, Incorporated, 1 edition.

[2] Han, J. H. and Poston, T. (2001). Chord-to-point distance accumulation and planar curvature: A new approach to discrete curvature. *Pattern Recogn. Lett.*, 22(10):1133–1144.

[3] Haralick, R., Shanmugam, K., and Dinstein, I. (1973). Textural features for image classification. *Systems, Man and Cybernetics, IEEE Transactions on*, SMC-3(6):610–621.

[4] Ion, A. (2009). *The Eccentricity Transform of n-Dimensional Shapes with and without Boundary*. PhD thesis, Vienna Univ. of Technology, Faculty of Informatics.

[5] Ion, A., Artner, N., Peyre, G., Kropatsch, W., and Cohen, L. (2011). Matching 2d and 3d articulated shapes using the eccentricity transform. *Computer Vision and Image Understanding*, 115(6):817 – 834.

[6] Kimmel, R. and Sethian, J. A. (1998). Computing geodesic paths on manifolds. In *Proc. Natl. Acad. Sci. USA*, pages 8431–8435.

[7] Kropatsch, W., Ion, A., Haxhimusa, Y., and Flanitzer, T. (2006). The eccentricity transform (of a digital shape). In *Discrete Geometry for Computer Imagery*, volume 4245 of *Lecture Notes in Computer Science*, pages 437–448. Springer Berlin Heidelberg.

[8] Ling, H. and Jacobs, D. W. (2007). Shape classification using the inner-distance. *IEEE Trans. Pattern Anal. Mach. Intell*, 29:286–299.

[9] Rosenfeld, A. (1983). A note on geometric transforms of digital sets. *Pattern Recognition Letters*, 1(4):223 – 225.

[10] Salakhutdinov, R., Tenenbaum, J. B., and Torralba, A. (2013). Learning with hierarchical-deep models. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(8):1958–1971.

[11] Sebastian, T. B., Klein, P. N., and Kimia, B. B. (2004). Recognition of shapes by editing their shock graphs. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(5):550–571.

[12] Sharvit, D., Chan, J., Tek, H., and Kimia, B. B. (1998). Symmetry-based indexing of image databases. *J. Visual Communication and Image Representation*, 9:366–380.