THE ASIA-OCEANIA TOP UNIVERSITY LEAGUE ON ENGINEERING - 2011, 10−12OCTOBER 2011, BEIJING, CHINA

Using Geodesic Information in Shape Matching for Object Recognition

Vittal Premachandran* **and Ramakrishna Kakarala**

School of Computer Engineering, Nanyang Technological University, Singapore
*corresponding author, E-mail: vittalp@pmail.ntu.edu.sg

Abstract

Object recognition via shapes is an interesting area of Computer Vision, which has gained a lot of momentum in the recent past. The ability of humans to detect objects from just their shapes has been a major motivation for further research in this area. In order to detect objects from just their shapes, some sort of a shape-matching algorithm should be developed, which can correctly identify similar shapes and reject dissimilar shapes.

In this paper, we introduce the use of different metrics for use in shape matching. We propose to use geodesic shape information in the formulation of the features depicting the shape of an object. The use of geodesic shape descriptor helps in capturing more of the shape information than the use of other metrics, say, the Euclidean distance. This new metric is then used to describe the shape characteristics of any given shape by developing a shape context at various sampled points along the shape contour.

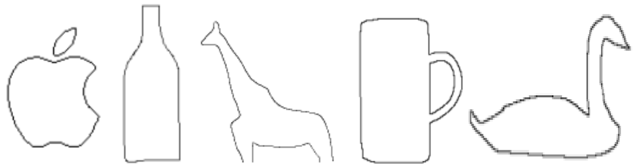
We then show how the shape context, built using the geodesic information, can be used to match other shape contexts, of sampled points obtained from the query shapes. This method is then followed by the use of a dynamic programming step to find the best alignment of the two sets of points, which concludes the shape matching.

We show how the use of such a shape context outperforms some of the commonly used algorithms, which are believed to have state-of-the-art results in shape matching.

1. Introduction

Object recognition is an important part of Computer Vision. It directly contributes towards image understanding, which is a growing field in computer vision and multimedia processing. Finding good representations of objects, finding good matching algorithms, etc, are central issues in various applications.

Object recognition in computer vision is usually done by the extraction of features from the query object, which is then matched to similar features that are extracted from reference objects. Feature extraction and feature matching are topics that have been majorly researched. They depend on the problem that is being addressed. The features could be some form of representation of the object's colour, texture, or shape, or a combination of one or more of them. However, in order to recognize the class of the object, usually, just the object's contour (shape) information should be enough. Certain psychophysical studies [1] show that humans can recognize the object's class using just the object's contour. Fig. 1a shows that certain objects have a characteristic shape and can be recognized only form their contours. In addition, we humans are able to recognize the objects that have a characteristic shape, even if they have different colours and/or textures. Fig. 1b shows multiple instances of a coffee mug, each having a different colour and texture, but all having a characteristic shape. It is this property, which we humans have, that acts as a motivation to research further in this field.



(a)

(b)

Figure 1: (a) Examples of objects, which can be recognized purely from their contours. The classes from left to right are, Apple Logo, Bottle, Giraffe, Coffee Mug, and Swan. (b) Examples of coffee mugs, each mug has a different colour and texture, but all having a unique and characteristic shape [2].

In order to perform object recognition based on shapes, one has to first extract the shape information from the objects, perform some sort of shape matching between the shapes extracted from the reference object and the query object, and finally make a decision as to whether the query object belongs to the same class as the reference objects or not. The decision is taken based on the cost of performing the match. The *bag-of-words* matching technique, used by many [3][4], is a very successful technique for shape matching. However, the use of the *bag-of-words* technique throws away all the spatial information. Clearly, it is the relative spatial location of the contour segments, which help us recognise the object based on its shape. Hence, we make use of features, which incorporate spatial information, while performing shape matching.

Each of these steps will be explained in detail, in the following sections. In Section 2, we look at how the shape information can be effectively extracted from a given object with a brief review of previous work in the field of shape matching. In Section 3, we go on to explain how the use of geodesic information will help in capturing the shape information in a better way than the currently used methods. Later, in Section 4, we validate our claim by showing the results from some of the experiments that we have performed, and conclude the paper in Section 5.

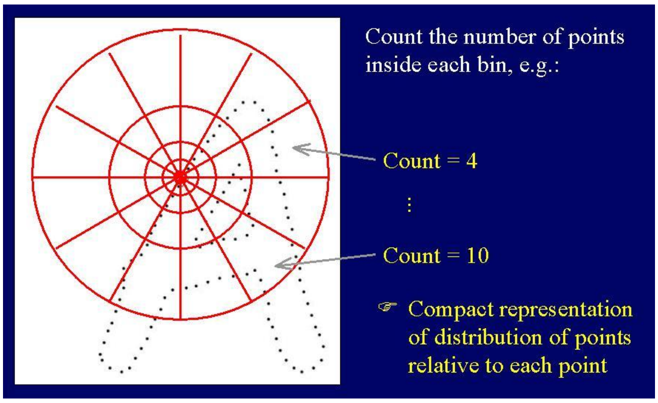
2. Shape Extraction and Literature Review

The object's shape information is stored in the object's contour, i.e. the object's boundary. Hence, it makes sense to extract the boundaries of the object before performing any kind of shape matching. The boundary is usually extracted by giving the image as an input to an edge detector, which would then output the edge-map of the input grayscale image. The edge-map contains only the information from the object's shape and discards other information such as the colour and texture of the object. An edge-map is usually a binary image with pixels having a value of either '1', if they belong to the object's boundary, or having a value '0', if they do not belong the object's boundary. Any edge detector can be used to perform the task of edge detection. Most famous one is the Canny edge detector [5] and the most recent and well-performing edge detector is the Pb edge detector [6].

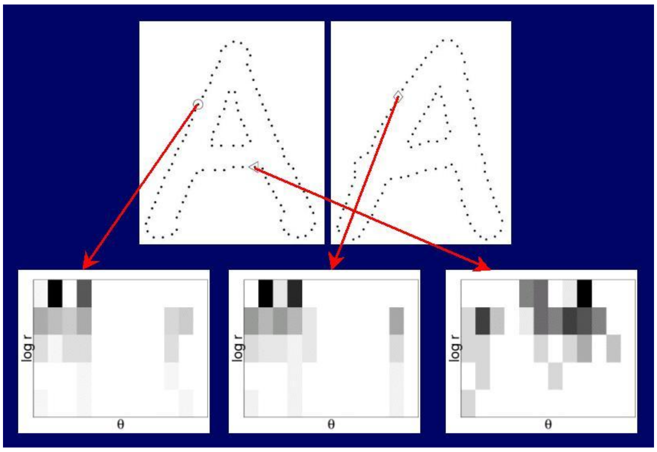
After the extraction of the object's contour, we are now required to represent the contour in a manner that can be used for comparison with other contours. Different authors use the contours in different ways. Some match the contours directly by using distance metrics such as the Hausdorff Distance [7][8] or the Chamfer Distance [9][10]. The use of such matching techniques have a low threshold because, the metrics, Hausdorff distance and Chamfer distance are extremely sensitive to clutter and rotation. Some others ways to match contours involve the detection of corners on the contours, splitting the long contour, at the corner points, into smaller sub-contours, and then matching the sub-contours [11]. Such methods are as robust as the corner detector that is used [12]. In addition, the definition of a corner itself is a very subjective. What appears as a corner to one corner detector might appear as just a slight bend in the contour to another corner detector.

Because of the above-mentioned problems in certain shape matching techniques, a different technique of representing contours has come into existence. The method involves the sampling of contours into a set of uniformly spaced points and then using this set of sampled points to represent the object's contour in a succinct manner. This set of points will be used to define a representation, which would capture the complete shape characteristics of the object.

The famous paper by Belongie et al. [13] used the shape context for concisely representing the object's contour, and then used this representation for shape matching. The shape context is made of two key ingredients; the distances and the angles between the sampled points. The distances were obtained by calculating the Euclidean distance between the point under consideration and every other sampled point. Similarly, the angular information between any two points was obtained by finding the angle between the tangent drawn at point under consideration and the displacement vector between the two points. Armed with the distance and the angular information, a histogram was then created at each sampled point, which was the shape context of that point. Similar such shape contexts are generated for every sampled point. Furthermore, the binning was done in such a way that points that were close to each other were given more importance than the points that were far away from each other. To specify such importance, instead of uniform binning, the authors adopted a log-polar style of binning. Figure 2 gives an example of the shape context at a particular sampled point on the character 'A'. The red Circular disc is a log-polar histogram with twelve angular orientations and five distance bins. The histogram is placed over the point under consideration to calculate the point's shape context. The value in bin (i,j) is the number of points that fall in the intersection of the *i*th distance bin and the *j*th angular bin. Figure 2b shows how points that are at similar locations on different characters, belonging to the same class, have similar shape contexts, as compared to the shape contexts of the points at different locations.



(a)



(b)

Figure 2: (a) The red circular disc is a log-polar histogram with twelve angular bins and five distance bins. (b) The shape context at points in similar locations in different characters belonging to the same class have similar looking shape contexts, as compared to the shape contexts of the points at different locations [14].

Formally, we could define the shape context as follows. Suppose we are given a shape contour *C*, we first sample the contour into a set of *n* equally spaced points, $P = \{p_1, p_2,...,p_n\}$. Each point $p_i \in P$, is a two dimensional vector, with the x-coordinate and the y-coordinate as its dimensions. The histogram at point p_i is defined as h_i and is given in Equation (1), below.

$$h_i(k) = \#\{p_j : j \neq i, p_j \in bin(k)\} \quad (1)$$

Here, $h_i(k)$ is the histogram at the *i*th point, for *k*th bin, whose value is the number of points, $p_j, 1 \leq i, j \leq n$, for all *j* not equal to *i*, falling into *bin(k)*.

In the next section, we describe the modifications that we made to the original shape context, which help in obtaining better results while matching.

3. Geodesic Information as Shape Descriptor

In the previous section, we saw how a shape context was built to represent the shape globally. The two main ingredients of building the shape context was the distance information and the angular information. While the original authors of the shape context suggested the use of Euclidean distance for calculating the distance between the two points, Ling and Jacobs [15] pointed out that such a distance measurement was not sufficient in capturing the complete shape information, from a given shape. By using the Euclidean distance, we lose information about the shape of the contour in joining the two points. Hence, Ling and Jacobs proposed a new shape context, namely, the Inner Distance Shape Context (IDSC). The IDSC at a point is also made of the same two ingredients, the distance and the angle. However, it differs in how the distance and the angle values are calculated. For any two given points, *p* and *q*, the inner distance between the two points is the shortest distance along the path joining the two points, such that the path lies completely within the boundary of the object. The angle between any two points, *p* and *q*, is the angle between the tangent drawn at point *p*, and the first part of the path leading to point *q*. The authors name the angle as the inner angle. Figure 3 gives an example of how the inner distance and the inner angle is calculated.

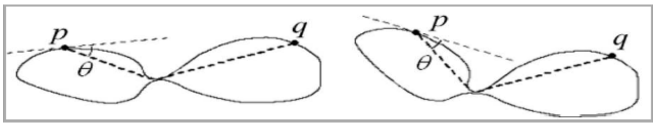


Figure 3: The inner distance between any two points, *p* and *q*, is the shortest distance along the path joining the two points, such that the path lies completely within the boundary of the object. The inner angle is the angle between the tangent at the first point and the first part of the path leading from the first point, *p*, to the second point, *q*. [15]

While the inner distance captures much more of the geodesic information about the object than the Euclidean distance, the usage of the inner angle leads to loss of vital information. Take for example the objects given in Figure 3. If we can say that the object can be divided into two parts, say, the left lobe and the right lobe, then the inner angle between a particular point, say *p*, on the left lobe, and every point on the right lobe is one and the same. This results in a loss in information regarding the exact location of the points in the right lobe, with respect to point *p*. Figure 4 gives an illustration of the same.

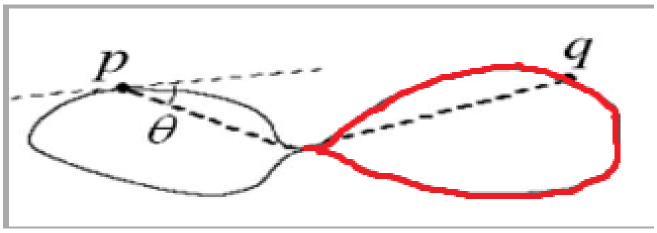


Figure 4: The inner angle between the point, *p*, and all other sampled points from the contour segment marked in red will have the same inner angle as that between the two points, *p* and *q*.

Hence, the first modification that we propose includes the change in the way the angle is measured. The angle between two points, say *p* and *q*, in our implementation, is calculated as the angle between the tangent at point *p*, and the displacement vector between the two points, starting from point *p* and ending at point *q*. We name this angle as the Euclidean Angle. Figure 5, gives an illustration of how the Euclidean Angle is calculated.

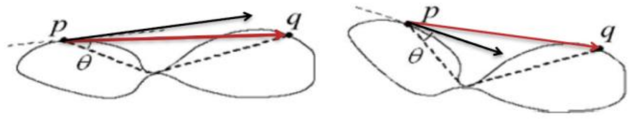


Figure 5: The Euclidean angle is calculated as the angle between the tangent (shown by the black arrow) at a point p, and the displacement vector (shown by the red arrow), starting at point p and ending at point q.

The second modification that we make is in the way the distance metric is calculated. We propose to replace the use of either the Euclidean distance, or the use of the inner distance, with the geodesic distance between two points. The geodesic distance between any two points is the shortest distance between them, along the contour of the object. The use of such a distance metric allows us to capture more of the shape information. The geodesic distance between two points, p and q, is approximated to be the sum of the linear distances between consecutive sampled points, starting from point p and ending at point q.

3.1. Shape Matching

Now that we have a new distance metric and a new angular metric, we create the shape contexts at each sampled points as given in Equation (1). In order to match two shapes, A and B, we first sample both the shapes, and generate shape contexts at each sampled point in both shape A and shape B. With the histograms of both the shapes now available, we use certain histogram matching techniques to compare the "distance" between the two shapes. The histogram of each sampled point, p_i, from shape A, is compared to the histogram of each sampled point, q_j, from shape B, using the χ^2 statistic as in [13]. The cost of comparing point p_i, from shape A, to point q_j, from shape B, is given in Equation (2).

$$c(i,j) = \frac{1}{2} \sum_{1 \le k \le K} \frac{\left[h_{A,i}(k) - h_{B,j}(k)\right]^2}{h_{A,i}(k) + h_{B,j}(k)} \qquad (2)$$

Where c(i,j) is the cost, K is the maximum number of bins in the histogram, and k refers to a bin number between 1 and K. By default, the above method assumes that the sampled points from shape A, $p_1, p_2, \ldots, p_n$, are aligned with the sampled points from shape B, $q_1, q_2, \ldots, q_n$. However, this might not be necessarily true. Hence to find the best matching we compare every point in shape A with every point in shape B. Such a comparison provides us with an nxn matrix, where n is the number of sampled points on each contour. The job, now, is to find the best alignment of the two sets of points. This can be performed quickly using dynamic programming. Detailed description to perform this match can be found in [11].

In the next section, we show some experimental results, which compares our method with other similar shape matching methods.

4. Experiments and Results

In order to test our descriptor, we perform our experiments on the well-known MPEG-7 dataset [16]. The MPEG-7 database consists of 1400 images in total. The database consists of images from 70 different classes. Each class consists of 20 images. The MPEG-7 dataset is a challenging dataset as the objects in each class not only differ by translation, scale and rotation, but are also deformed and occluded in many cases. Figure 6 shows an example from each of the 70 classes, in the database.

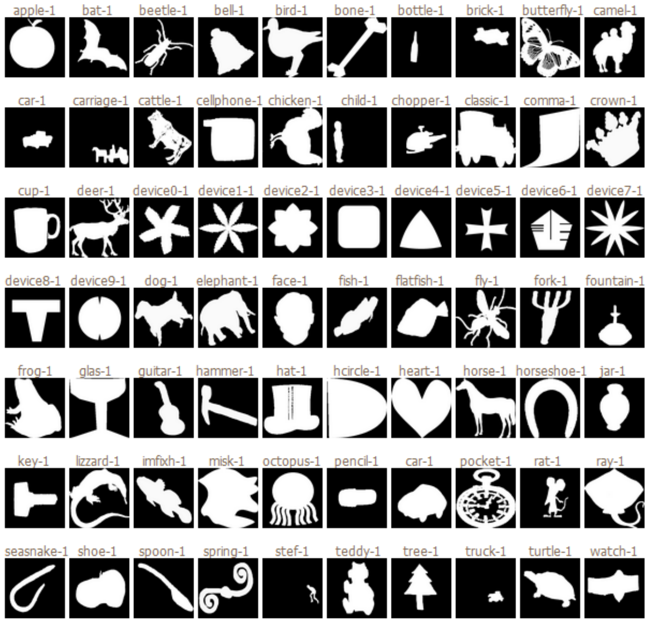
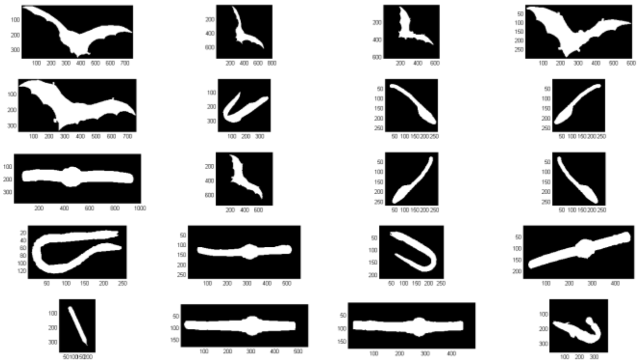


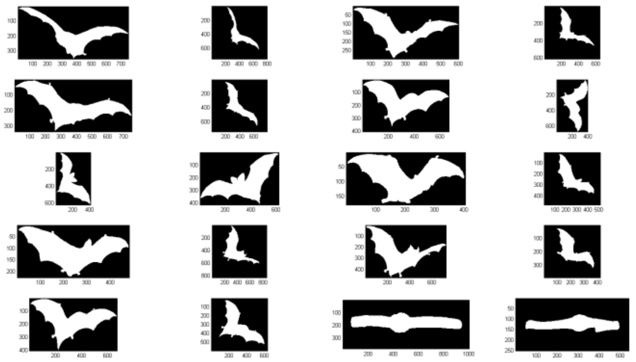
Figure 6: A collection of images from the MPEG-7 database. There are a total of 70 images. Each sub image shows one example from one of the 70 classes [17].

The recognition rate, for the MPEG-7 database, is measured by the score called the Bullseye score. The bullseye test involves the computation of the bullseye score for the whole database. To compute the bullseye score, each image is matched with every other image in the database. The images are then sorted in ascending order, based on the cost of matching. The top 40 images i.e. 40 images with the least cost of matching, are counted. At most, 20 of these 40 images can belong the same class as the query image. If all 20 images from the same class as the query image are present in the top 40 best matches, then the bullseye score for that query image is said to be 100%. However, if the top 40 best matches did not contain all 20 images from the same class, the bullseye score is (k/20)*100, where k is the number of objects from the same class as the query image, in the top 40 best matches.

We set the following parameters in our experiments, which is the same as that set in [15]. The number of sample points, $n = 100$, the number of distance bins, $n_d = 8$, the number of angular bins, $n_\theta = 12$, and the number of locations to search for the points alignment is, $k = 8$. We also account for the handling of the mirror image examples by comparing each image to every other image as well as the mirror image of every other image in the database. Figure 7 shows the comparison of retrieval results of our method alongside the method of [15].



(a)



(b)

Figure 7: (a) The figure shows the retrieval results of the original IDSC algorithm [15] when a bat was given as an input. Of the 20 best matches, only 6 of them belong to the same class as the query image. (b) The figure shows the retrieval results of our algorithm when the same bat, as given to the original IDSC, was given as an input. Of the 20 best matches, 18 of them belong to the same object class as the query image.

Table 1 shows a comparison of the Bullseye scores obtained from other algorithms, which perform the same task, i.e. shape matching.

Algorithm	Bullseye Score
SC+TPS [13]	76.51%
Curve Edit [18]	78.17%
IDSC+DP [15]	85.40%
Ours	**86.73%**

Table 1: Comparison of the bullseye score obtained from our algorithm alongside the bullseye scores obtained from other algorithms.

5. Conclusion

In this paper, we have shown how useful geodesic information is, and how the geodesic information can be used to perform shape matching. We show the use of geodesic information captures more of the shape information compared to the metrics used in other algorithms. More detailed results and analysis will be available in a future paper of our, which is under preparation.

References

[1] I. Biederman, G. Ju. Surface vs. edge-based determinants of visual recognition. *Cognitive Psychology*, 20(1):38-64, January 1988.

[2] ETHZ Shape Classes Dataset. http://www.vision.ee.ethz.ch/datasets/downloads/ethz_shape_classes_v12.tgz

[3] G. Csurka, C.R. Dance, L. Fan, J. Williamowski, C. Bray. Visual categorization with bags of keypoints. *ECCV International Workshop on Statistical Learning in Computer Vision*, 2004.

[4] J. Sivic, A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *Proc. Int. Conf. on Computer Vision*, volume 2, pages 1470-1477, Nice, France, October 2003.

[5] J. Canny, A computational approach to edge detection. *Readings in computer vision: issues, problems, principles, and paradigms*, vol. 184, 1987.

[6] D. Martin, C. Fowlkes, J. Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *PAMI*, 2004.

[7] D. Huttenlocher, G. Klanderman, W. Rucklidge. Comparing images using the Hausdorff distance, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 850-863, 1993.

[8] D. Sim, R. Park. Two-dimensional object alignment based on the robust oriented Hausdorff similarity measure. *IEEE Transactions on Image Processing*, vol 10, no. 3, pp. 475-483, 2002.

[9] G. Borgefors. Hierarchical chamfer matching: A parametric edge matching algorithm. *IEEE Transactions on pattern analysis and machine intelligence*, vol. 10, no. 6, pp. 849-865, 1988.

[10] J. Shotton, A. Blake. R. Cipolla, Multiscale categorical object recognition using contour fragments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1270-1281, 2007.

[11] E.G.M. Petrakis, A. Diplaros, E. Milios. Matching and Retrieval of distorted and Occluded Shapes using Dynamic Programming, *IEEE Transactions on Pattern Analysis and Machine Intelligence*. pp. 1501-1516, 2002.

[12] P. Tissainayagam, D. Suter. Assessing the performance of corner detectors for point feature tracking

applications. *Image and Vision Computing*, vol. 22, no. 8, pp. 663-679, 2004.

[13] S.Belongie, J.Malik, J.Puzicha. Shape Matching and Object Recognition Using Shape Context, *PAMI*, 2002.

[14] S. Belongie, J. Malik, J. Puzicha. Matching with Shape Contexts. [Online]. Available: http://www.eecs.berkeley.edu/Research/Projects/CS/vision/shape/sc_digits.html

[15] H. Ling, D.W. Jacobs. Shape Classification Using Inner Distance, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 2, pp. 286-299, Feb. 2007.

[16] MPEG 7 Core Experiment CE-Shape-1 Test Set. [Online]. Available: http://knight.cis.temple.edu/~shape/MPEG7/MPEG7dataset.zip

[17] R. Ralph. MPEG 7 Shape Matching. [Online] Available: http://knight.cis.temple.edu/~shape/MPEG7/dataset.html

[18] T.B. Sebastian, P.N. Klein and B.B. Kimia. On Aligning Curves, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2003.