

Shape Classification Using Shape Context and Dynamic Programming

Prof Muzameel Ahmed, Rashmi, Shravya.M, Sindhu.N, Supriha.Shet

Abstract: The suggested algorithm for shape classification described in this paper is based on several steps. The algorithm analyzes the contour of pairs of shapes. Their contours are recovered and represented by a pair of N points. Given two points p_i and q_j from the two shapes the cost of their matching is evaluated by using the shape context and by using dynamic programming, the best matching between the point sets is obtained. Dynamic programming not only recovers the best matching, but also identifies occlusions, i.e. points in the two shapes which cannot be properly matched. From dynamic programming we obtain the minimum cost for matching pairs of shapes. After computing pair wise minimum cost between input and all reference shapes in the given database, we sort based on the minimum cost in ascending order and select first two shapes to check if it belongs to the input class. If it belongs to the input class, then we say that the shape is classified as a perfect match, else it is a mismatch. The algorithm has been tested on a set of shape databases like Kimia-25, Kimia-99, Kimia-216 and MPEG-7 providing good performances for shape classification.

Key words: Shape Context, Dynamic Programming, contours, log-polar histogram, classifier, outlier.

I. Introduction:

Computer vision aims at building machines able to view and to recognize objects and features as humans are able to do. In recent years, computer vision and psychology have extensively studied visual perception and object recognition and made significant progress, but current techniques for object recognition and shape classification do not yet provide entirely satisfactory solutions. Objects have several properties that can be used for

recognition and categorization, like shape, color, texture and brightness. The property of shapes is used to provide a well performing algorithm for

shape classification. In machine learning, pattern recognition is the assignment of a label to a given input value. An example of pattern recognition is classification, which attempts to assign each input value to one of a given set of classes (for example, determine whether a given email is "spam" or "non-spam"). There are two major approaches for shape representation in the literature: one approach is

boundary based and uses contour information, and the second approach needs a holistic representation, requiring general information about the shape. Shape-based methods primarily analyze silhouettes, i.e. boundaries of objects present in images. Silhouettes do not have holes or internal markings, and therefore are represented by a single closed curve, parameterized by its arc-length. Shape context is a descriptor developed for finding correspondences between point sets. Given a set of points from an image (e.g. extracted from a set of detected edges), the shape context captures the relative distribution of points in the plane relative to each point on the shape. A shape is represented

by a discrete set of points sampled from the internal or external contour on the object. The shape contexts have been used as attributes for a weighted bipartite matching problem. Attalla and Siy have presented a polygonal approximation of shape contours that divide it into equal segments and all segments will serve as local features that will represent the shape. The method of Fourier descriptor is extended to object recognition to produce a set of normalized coefficients which are invariant under any affine transformation. The method is based on a parameterized boundary description which is transformed into the Fourier domain and normalized to eliminate dependencies on the affine transformation and on the starting point. The inner-distance between landmarks, i.e. the shortest path between these landmarks, has also been used to build shape descriptors. Dynamic Programming (DP) approach has been used for shape classification. The basic idea behind this approach is to represent each shape by a sequence of convex and concave segments using the inflection points extracted from the curvature and to allow the matching of merged sequences of small segments in a noisy shape with larger segments in the other shape. This procedure is obtained by a Dynamic Programming (DP) algorithm searching for the least cost match in a Dynamic Programming (DP) table.

Some of the object recognition methods rely on holistic representation of shapes. Geometric invariants are shape descriptors that remain unchanged under geometric transformations such as projections or changes in point of view. These invariant descriptors can be obtained locally or globally. Another global way to represent shapes is obtained by using PCA-based methods.

In the algorithm described here, we use contour of shapes and shape context descriptor. Using Dynamic Programming (DP), we classify the shapes. We have tested

for set of databases like kimia-25, kimia-99, kimia-216 and also for complex database like MPEG-7.

II. Problem Statement:

Suppose there is a set of shapes in a database, which is divided into N classes and n shapes in each class. Given a input shape from the database, the algorithm should match the input shape to the class to which it belongs.

III. Proposed Method:

A. Outline Of Proposed Approach:

The proposed algorithm for shape retrieval and recognition is based on several steps summarized in Fig1. The algorithm is based on the analysis of the contour of the pair of shapes under consideration. Its contour is recovered and mapped into a pair of N points. The cost of matching between points p_i and q_j from the two shapes is evaluated by the shape context which we will briefly describe further. Having the cost of matching each point from one shape with all the points in the other shape, by using dynamic programming we obtain the best matching between the point sets of the two shapes. Dynamic programming not only recovers the best matching between points, but also identifies outliers, i.e. points in the two shapes which cannot be properly matched. This step is essential for identifying partial occlusions in the two shapes. The cost of matching obtained from dynamic programming is used for classification. The proposed approach has been tested on most public shape database.

B. Shape Context

Shape context has been introduced by Belongie. An object is represented by a discrete set of points sampled regularly along the contour. For every point, a log-polar histogram, the shape context is computed approximating the distribution of adjacent point locations relative to a reference point. In order to achieve scale invariance, the outer radius for the histograms is set equal to the mean distance between all the pair points.

For a point $p_i, i=1, \dots, N$ of the shape, the shape context is a coarse histogram h_i of the relative coordinates of the remaining $q_j, j=1, \dots, N-1$ points:

$$h_i(k) = \#\{q \neq p_i : (q - p_i) \in \text{bin}(k)\}. \quad (1)$$

The bins are uniform in log-polar space, making the descriptor more sensitive to positions of nearby points than those of more distant points. The cost of matching a pair of points p_i and q_j from two shapes is computed as

$$C(p_i, q_j) = \frac{1}{2} \sum_{k=1}^K \frac{[h_i(k) - h_j(k)]^2}{h_i(k) + h_j(k)}, \quad (2)$$

where $h_i(k)$ and $h_j(k)$ denote the K -bin normalized histogram at p_i and q_j , respectively. This definition of shape context is not rotationally invariant, but can be easily modified to be a completely rotational invariant descriptor. There are several ways to obtain rotational invariance. One possibility is to make the reference angle for shape context equal to the local tangent angle on the shape. Another option is to align the shape context axis with the principal axis of the shape. Shape context is not invariant for mirror transformation; therefore we consider also the mirrored shape when computing the shape context.

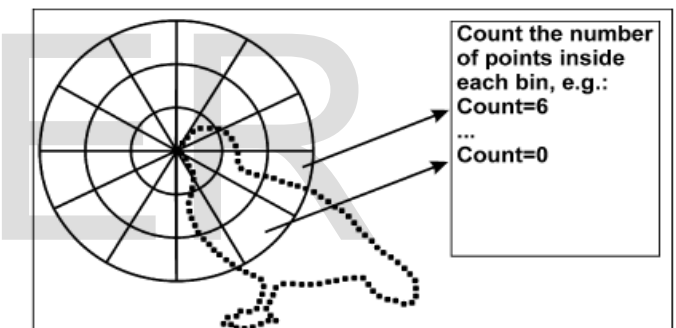


Fig 3.2 A scheme of representation of point sets using shape context .A shape is represented by a discrete set of points sampled regularly along the contours. For every point, a log-polar histogram—the shape context—is computed which approximates the distribution of adjacent point locations relative to the reference point.

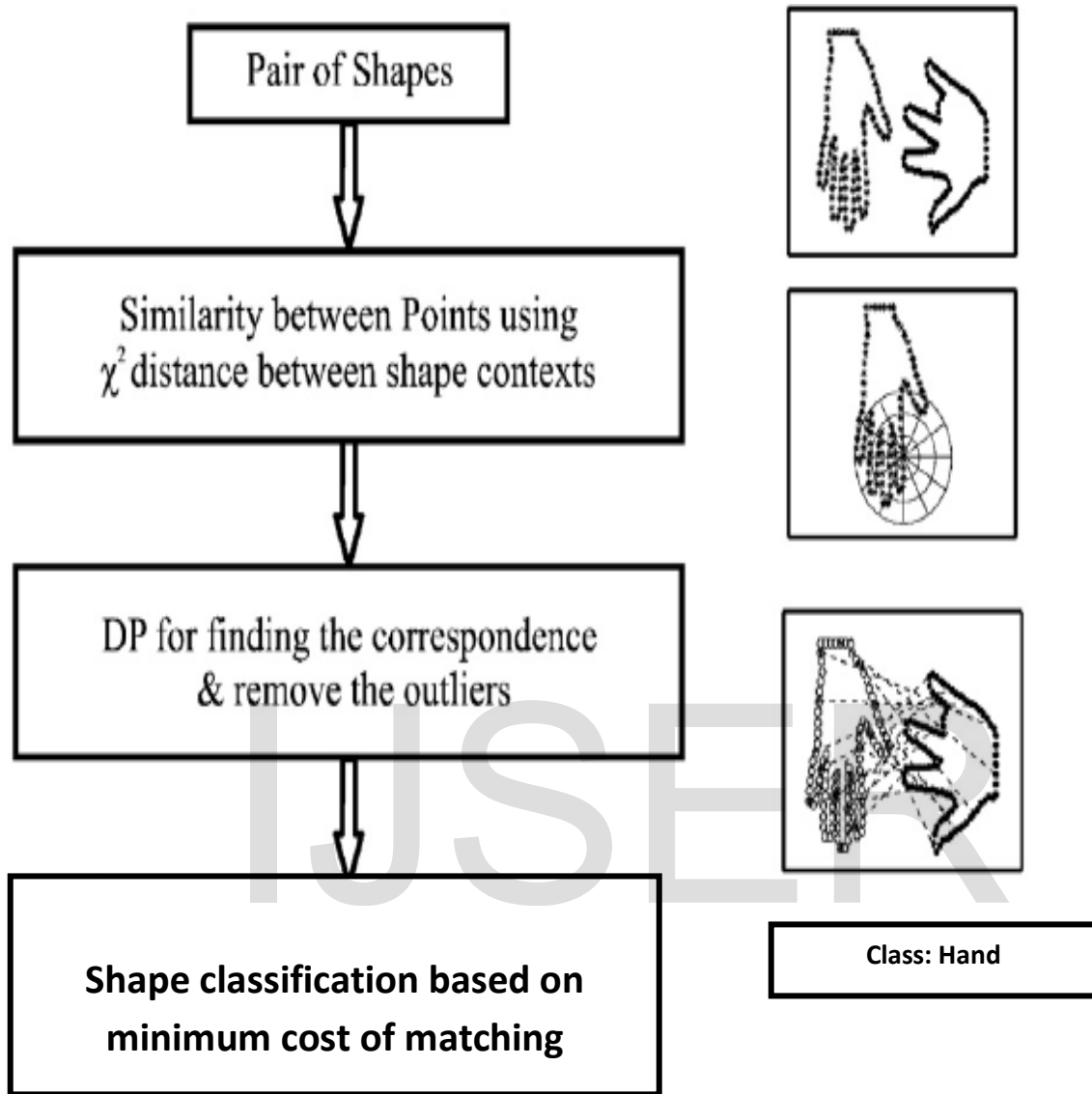


Fig.1 Outline of proposed approach

C. Dynamic Programming For Point Correspondence:

Dynamic Programming (DP) has already been used for many applications especially for contour matching. These applications consider sequences of convex/concave segments—separated by inflection points—forming the two shapes under consideration with the goal of finding the best match of segments in shapes A and B. This is formulated as a minimization problem which is solved efficiently by Dynamic Programming (DP). The algorithm builds a Dynamic Programming (DP) table or matrix, where rows and columns correspond to inflection points of the two shapes A and B, respectively.

The Dynamic Programming (DP) algorithm has two main steps providing the correspondence between the point sets and removing the outliers (and occlusions). The first step consists in filling cell by cell a Dynamic Programming (DP) table identifying possible outliers and occlusions. The second step tracks the minimum path over the Dynamic Programming (DP) table. In our case, we have two point sets describing the contour of the two shapes and we want to find the best correspondence between these two point sets and identify outliers. Given two shapes A and B, each described by a set of N points, given p_i in shape A and q_j in shape B, the cost of matching p_i and q_j is given by Eq. (2), which are the entries of a suitable cost matrix $C(i, j)$. This step is based on a DP algorithm based on the following procedure:

Let us suppose we have a penalty threshold p for an occlusion (i.e. an outlier) and we have another suitable $DP(i, j)$ matrix with the same size of the cost matrix C with zero entries. We fill each cell in the table by considering the penalty value of p and three previous entries in the DP table, namely left cell, upper cell and the diagonal cell, and we fill it. During this process, at each step, we keep track of the minimum among the three previous entries so as to find the minimum path after filling the matrix. Indeed, we start from the last entry and, based on the conditions that have given the minimum values at each step, we track back and find the complete path that gives us the corresponding points and outlier.

The basic idea for dynamic programming is as given below: Let DP be the dynamic programming table and C be the cost matrix obtained after shape context. The entries to Dynamic Programming (DP) table is as shown in below fig.3. Entries $DP(i-1, j-1)$, $DP(i-1, j)$, and $DP(i, j-1)$ are needed to fill in $DP(i, j)$.

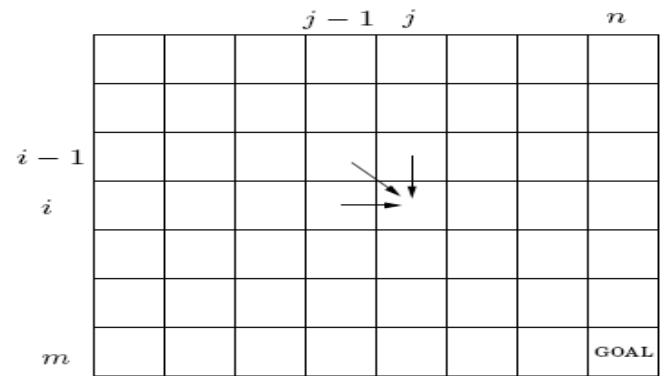


Fig.3 DP Table

Any order is fine, as long as $DP(i-1, j)$, $DP(i, j-1)$, and $DP(i-1, j-1)$ are handled before $DP(i, j)$. For instance, we could fill in the table one row at a time, from top row to bottom row, and moving left to right across each row. Or alternatively, we could fill it in column by column. Both methods would ensure that by the time we get around to computing a particular table entry, all the other entries we need are already filled in. After filling all the entries, the goal i.e $DP(m, n)$ represents the minimum cost of matching. Shape classification is done using this data for all point sets.

D. Shape classification:

Calculate DP table for input and all reference shape. The reference shape which gives minimum cost for input shape is considered as the best match and classified to the class to which the reference shape belongs.

IV. Experimental results:

In this section we checked the performance for different databases including complex database MPEG-7.

A. Choosing the parameters:

In our experiments we consider the longest arc of the contour of each shape then it is represented by 100 points. During the computation of the shape context, we used 8 and 12 bins for computing the histograms for distances and angles, respectively. The threshold penalty value for detecting the outliers was set equal to 0.6 of the mean cost C.

B. Kimia-25 shape data set:

Kimia-25 provided by Sharvit consists of 25 shapes from six different categories as shown in Fig.4. In each category we have four images. Since our algorithm works for equal number of objects in each class, we have taken only 24 images. Out of 24 images, 20 images have properly matched which gives a performance of 83.33%.

The fig 4 shows kimia-25 data set.

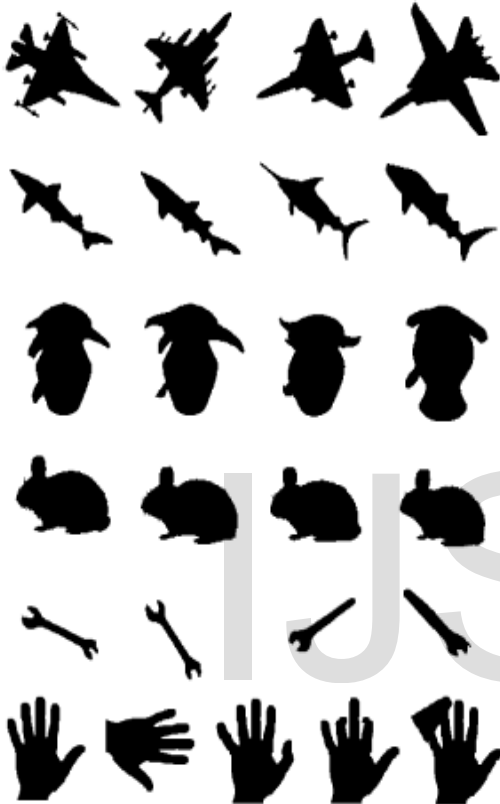


Fig.4 The kimia-25 data set consists of 25 shapes from six different categories: each row shows a different object category.

C. Kimia-99 data set:

This database consists of 99 shapes from nine different categories as shown in Fig.5. In each category we have eleven images. Out of 99 images, 91 images have properly matched which gives a performance of 91.9%.

Kimia-99 dataset is shown in Fig.5.

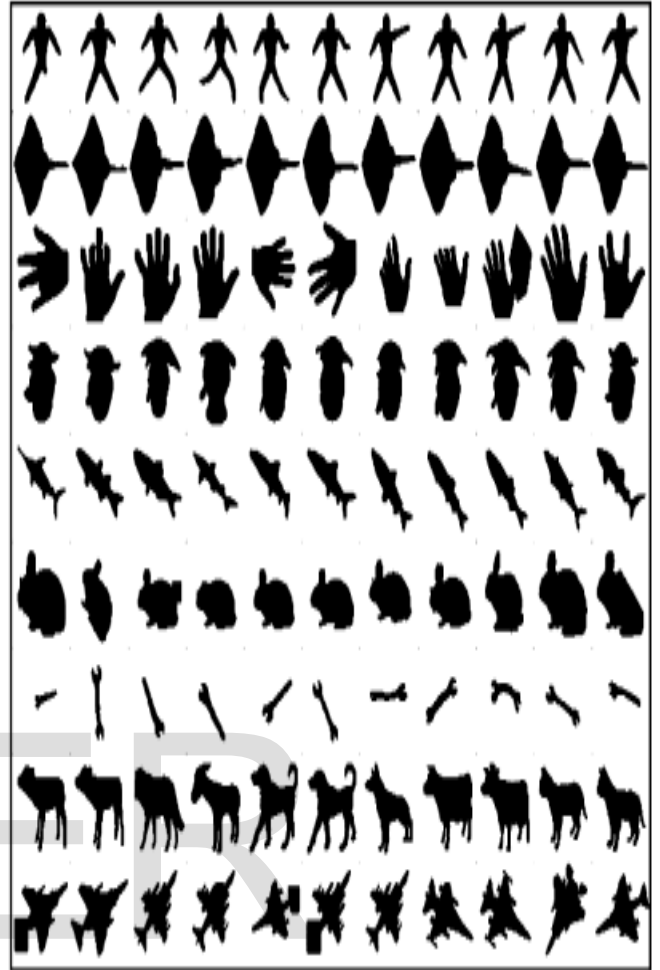


Fig.5 Kimi-99 database: Each row shows a different object category.

D. Kimia-216 shape data set:

This database consists of 216 shapes from eighteen different categories as shown in Fig.6. In each category we have twelve images. Out of 216 images, 201 images have properly matched which gives a performance of 93%.

The fig.6 shows the examples of shapes in Kimia-216 data set.

F. Final summary of performance:

Databases	Number of images taken	Number of images matched	Performance
Kimia-25	24	21	83.33%
Kimia-99	99	91	91.9%
Kimia-216	216	201	93%
MPEG-7	1400	1269	90.64%

Table 8 Performance measure

The threshold The penalty value for detecting the outliers was set equal to 0.6 of the mean cost C as computed in Section 2.2.

V. Conclusion

The algorithm for shape classification described has been tested on a variety of shape databases and for most of them provides better performances. Indeed for the Kimia-25, 99,216, MPEG-7 the proposed algorithm provides good results for classification. The algorithm here is based on two major properties: shape context i.e distribution of adjacent point locations relative to a reference point. Shape context descriptor is considered to give a better performance when compared to the previous fourier descriptor method. Second

property is the dynamic programming which illustrates the capability of best matching between two shapes using dynamic programming. Dynamic programming is considered as the best method for finding correspondences between two shapes and also in detecting outliers. The minimum cost of matching obtained from dynamic programming is used to classify shapes. We have tested for a few standard databases which gives performance approximately equal to 91. When tested for simple databases like kimia-99, kimia-216, a better performance is obtained compared to complex databases like MPEG-7.

References

- [1] Serge Belongie, Jitendra Malik and Jan Puzicha, "Shape Matching And Object Recognition Using Shape Contexts ", 2002.
- [2] Sven Loncaric, "A survey of shape analysis techniques".
- [3] Sameer Antani and L.Rodney Long, "Partial Shape Matching of Spine X-Ray Shapes Using Dynamic Programming",
- [4] Yang Mingqiang, Kpalma Kidiyo, Ronsin Joseph, "A survey of shape extraction techniques", july 2008.
- [5] Evangelos Milios and Euripides G.M. Petrakis, "Retrieval Based on Dynamic Programming", 2001
- [6] Amir A Amini , Terry E Weymouth, Ramesh C Jain, "Using Dynamic Programming for Solving Variational Problems on Vision.
- [7] Zusheng Rao, Euripides G.M. Petrakis, Evangelos Milios, "Retrieval Of Deformed and Occluded Shapes Using Dynamic Programming", 1999.
- [8] Noredinne Gherabi and Mohamed Bahaj, "Method For Shape Recognition Using Dynamic Programming".
- [9] T. Sebastian, P.N. Klein, B. Kimia, "Recognition of shapes by editing their shock graphs", IEEE Trans. Pattern Anal. Mach. Intell. 26(5) (2004) 550–571.
- [10] J.W. Gorman, O.R. Mitchell, F.P. Kuhl, "Partial shape recognition using dynamic programming", IEEE Trans. Pattern Anal. Mach. Intell. 10 (2)(1988) 257–266.
- [11] T.H. Cormen, C.E. Leiserson, R.L. Rivest, "Introduction to Algorithms", MIT Press, Cambridge, MA, 2001
- [12] B.Leibe, B.Schiele, "Analyzing appearance and contour based methods for object categorization", in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), vol. 2, 2003, 409–415.
- [13] M. Bober, "MPEG-7 visual shape descriptors", IEEE Trans. Circuits Syst. Video Technol. 11 (6) (2001) 716–719.

- [14] G.McNeill, S. Vijayakumar, "A probabilistic approach to robust shape matching", in: International Conference on Image Processing (ICIP), 2006
- [15] T. Sebastian, P.N. Klein, B. Kimia, "Recognition of shapes by editing their shock graphs", IEEE Trans. Pattern Anal. Mach. Intell. 26(5) (2004) 550–571.
- [16] H. Ling, D.W. Jacobs, "Shape classification using the inner-distance", IEEE Trans. Pattern Anal. Mach. Intell. 29 (2) (2007) 286–299.
- [17] H. Freeman, "Computer processing of line-drawing images", Comput. Surv. 6 (1) (1974) 57–97.
- [18] Biederman, G.Ju, "Surface versus edge-based determinants of visual recognitions", Cognit. Psychol. 20 (1988) 38–64.
- [19] S. Ullman, "High-Level Vision", MIT Press, Cambridge, MA, 1996.
- [20] E. Rivlin, I. Weiss, "Local invariants for recognition", IEEE Trans. Pattern Anal Mach. Intell. 17 (3) (1995) 226–238.

IJSER