



BAYESIAN CLASSIFIER ALGORITHM FOR SHAPE CLASSIFICATION

DAS A. AND THOTE S.R.

Department of Electronics & Telecommunication Engineering, J.D.I.E.T., Yavatmal, MS, India.

*Corresponding Author: Email- amal48958@gmail.com, thotesaurabh@gmail.com

Received: February 21, 2012; Accepted: March 15, 2012

Abstract- This paper provides a new algorithm for shape matching of different objects. The main idea is to match skeleton graphs by comparing the skeleton paths which are formed between skeleton end nodes. In comparison with other methods just like MAT, Shock Graph, grassfire algorithm method, we do not consider the skeleton tree structure. Our approach is motivated by the fact that visually similar skeleton graphs may have completely different skeleton graph. The proposed comparison of skeleton paths between endpoints of skeleton graphs yields up to 100% correct matching of shapes. If there is noise in the image then its skeletons are pruned (discarding of unwanted portion) by skeleton pruning method. Our experimental results demonstrate that It is having low storage i.e. it doesn't require more than 1 bit/pixel for storing purpose. Also it is faster than other methods and also having 98.2% efficiency. This proposed method can be applied in different field like Image / document processing ,Pattern recognition, Biological part matching, Computer graphics, Computational geometry, Geographic information systems, Robotics, Education, Industrial inspection etc.

Keywords- Skeleton Graph, Skeleton end point, Skeleton path construction, Skeleton pruning, low storage, high storage, immense application.

Citation: Das A. and Thote S.R. (2012) Bayesian Classifier Algorithm for Shape Classification. Journal of Artificial Intelligence, ISSN: 2229-3965 & E-ISSN: 2229-3973, Volume 3, Issue 2, pp.-106-110.

Copyright: Copyright©2012 Das A. and Thote S.R. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Introduction

Skeleton of any object is nothing but the centre pixels of any object, is an important shape descriptor for object recognition. Shape similarity based on skeleton matching usually performs better than contour or other shape descriptors in the presence of partial occlusion and articulation of parts. However, it is a challenging task to automatically recognize objects using their skeletons due to skeleton sensitivity to boundary deformation. Usually, the skeleton branches have to be pruned for recognition. Moreover, another major restriction of recognition methods based on skeleton is a complex structure of obtained tree or graph representations of the skeletons. Graph edit operations are applied to the tree or graph structures, such as merge and cut operations, in the course of the matching process. Probably, the most important challenge for skeleton similarity is the fact that the topological structure of skeleton trees or graphs of similar objects may be

completely different. This fact is illustrated in Fig. 1. Although the skeletons of the two horses (Fig. 1a) and (Fig. 1b) are similar, their skeleton graphs (Fig. 1c) and (Fig. 1d) are very different. This example illustrates the difficulties faced by approaches based on graph edit operations in the context of skeleton matching. To match skeleton graphs or skeleton trees like the ones shown in Fig. 1, some edit operations (cut, merge, and so forth) are inevitable. On the other hand, skeleton graphs of different objects may have the same topology, as shown in Fig. 2. The skeletons of the brush in Fig. 2a and the pliers in Fig. 2b have the same topology, as shown in Fig. 2c. This paper presents a novel scheme for skeleton-based shape similarity measure. The proposed skeleton graph matching is based on the similarity of the shortest paths between each pair of endpoints of the pruned skeletons; for example, see the shortest paths (in red) in Fig. 3. The shortest paths between every pair of skeleton endpoints are rep-

represented as sequences of radii of the maximal disks at corresponding skeleton points. We also benefit from the fact that the skeleton endpoints inherit a cyclic order from the contours. This is possible, since the skeletons are pruned based on contour partitioning with discrete curve evolution (DCE), which guarantees that all endpoints of skeleton branches lie on the contour. For example, in Fig. 4, all the endpoints (denoted by 1, 2 . . . 6) of the horse's skeleton are vertices of the DCE simplified polygon (in red). The DCE was introduced in. An important property of the DCE-based pruning is its stability in that it is able to remove spurious branches while preserving structurally relevant branches. The proposed skeleton graph matching method is described in Section 4. In contrast to the existing approaches to skeleton similarity, we do not explicitly consider the topological structure of the skeleton trees or graphs. Instead, we focus on the similarity of paths connecting the skeleton endpoints. We use the similarity of the shortest paths between each pair of skeleton endpoints to establish a correspondence relation of the endpoints in different graphs. For example, vertex 1 in Fig. 1a corresponds to vertex 1 in Fig. 1b since their shortest paths to vertices 2, 3, 4, 5, and 6 are similar. Finally, the dissimilarity value between graphs is easily estimated by the distances between the corresponding endpoints. Thus, the basic idea of our method is to determine the similarity of complex structures (graphs or trees) by examining the shortest paths between their endpoints. As we will show in Section 7, the proposed method yields successful recognition results and is faster than the existing graph and tree matching methods. The usage of shortest geodesic paths in skeleton graph and in shape similarity is not new; in particular, many-to-many matching in Demirci and the Inner-Distance in Ling and Jacobs use the shortest paths. However, there are substantial differences in our approach. Consider a shortest path between all skeleton nodes and considers the shortest paths between all contour points. We only consider the shortest skeletal paths between skeleton end nodes, which allow us to avoid the problem of the instability of the skeleton junction points and makes our approach more robust to contour deformations. Moreover, we use skeletal shortest paths in a different and novel way to define shape similarity. In our approach, we use a two-layer structure. In the first layer, skeletal shortest paths emanating from a given skeleton endpoint form its shape descriptor. In the second layer, we compute the similarity of two shapes by matching the shape descriptors of the skeleton endpoints. Since similar skeletons may have different number of endpoints, we have to allow for a partial correspondence of the endpoints. This is possible with a recently introduced method for partial similarity of sequences, which we extend and describe in next Section. By employing this method, we are able to also match skeletons of object parts to the skeletons of complete objects and match parts to parts, which is a necessary requirement for robust object recognition. The proposed skeleton graph matching is based on the assumption that similar skeletons have similar structure of their end nodes (measured by the similarity of the shortest paths to other end nodes). This assumption is significantly weaker than a standard assumption that a structure of the whole skeleton graph (based on both end nodes and junction nodes) is similar. Usually, the structure of both end nodes and junction nodes is weighted and edited since, as pointed out above (Fig. 1), it is common that skeletons of similar shapes have a different structure of junction

nodes. Moreover, as described in above, many approaches to match skeleton graphs require that the graphs are converted to trees prior to finding the correspondence. However, as we will illustrate in next Section, such a conversion may result in loss of important structural information and, consequently, negatively influence the object recognition result. The proposed method computes dissimilarity values for graphs that do not have to be trees. The geodesic skeletal paths are represented as sequences of radii of maximal disks in our approach. Although we do not explicitly consider the topological structure of the skeleton graphs, we do not completely ignore this structure. It is implicitly represented by the fact that overlapping parts of the geodesic skeletal paths are similar, since their overlapping parts have the same subsequences of radii. For our example in Figs. 1a and 1b, it means that paths from 6 to 1 and from 5 to 2 overlap. The fact that the overlapping segments are slightly different in Figs. 1a and 1b does not affect the similarity of corresponding sequences of radii in Figs. 1a and 1b. Therefore, our approach is flexible enough to perform extremely well on articulated shapes, but it is not too flexible to confuse dissimilar shapes. This fact is also confirmed by our experimental results next.

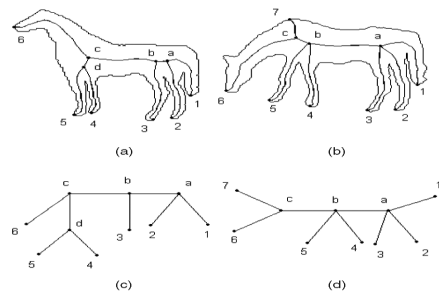


Fig.1- Visually similar shapes in (a), (b) have very different skeleton graph in (c), (d)

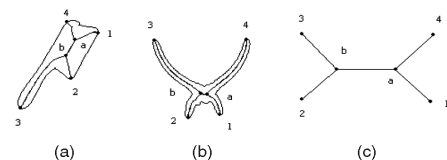


Fig.2- dissimilar shapes in (a), (b) can have the same skeleton graph in (c).

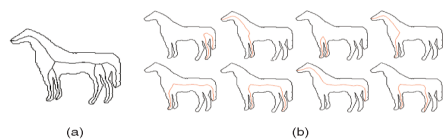


Fig.3- (a) the horse's skeleton. (b) The shortest paths (in red) between the pairs of endpoints on the skeleton

Skeleton graph

This section describes the initial steps for building the skeleton graphs. The following definitions apply to continuous skeletons, as well as to skeletons in digital images (composed of pixels).

Definition 1

A skeleton point having only one adjacent point is an endpoint (the

skeleton endpoint); a skeleton point having three or more adjacent points is a junction point. If a skeleton point is not an endpoint or a junction point, it is called a connection point. (Here, we assume that the skeleton curve is one pixel wide.)

Definition2

The sequence of connection points between two directly connected skeleton points is called a skeleton branch. A standard way to build a skeleton graph is as follows: both the endpoints and junction points are chosen as the nodes for the graph, and all the skeleton branches between the nodes are the edges between the nodes. For example, Figs. 1c and 1d are graphs representing the skeletons in Figs. 1a and 1b, respectively.

Definition3

The endpoint in the skeleton graph is called an end node, and the junction point in the skeleton graph is called a junction node.

Matching of skeleton graph

We match skeleton graphs by establishing a correspondence of their end nodes only, since these nodes are the salient points on the contour, and all skeleton branches ending on the contour can be seen as visual parts of the original shape. Thus, the proposed representation does not involve any junction nodes.

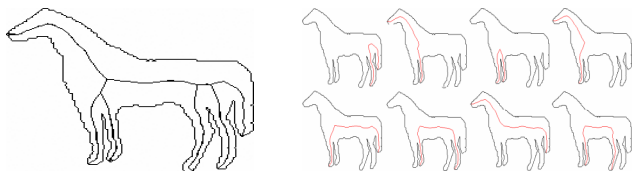


Fig. 4- (a) the horse's skeleton. (b) The shortest paths (in red) between the pairs of endpoints.

Suppose there are N end nodes in the skeleton graph G to be matched, and let v_i ($i = 1, 2, \dots, N$) denote the i th end node along the shape contour in the clockwise direction. Let $sp(m, n)$ denote the shape path from v_m to v_n . We sample $sp(m, n)$ with M equidistant points, which are all skeleton points. Let $R_{m,n}(t)$ denote the radius of the maximal disk at the skeleton point with index t of $sp(m, n)$. Let $R_{m,n}$ denote a vector of the radii of the maximal disks centered at the M sample skeleton points on $sp(m, n)$:

$$R_{m,n} = (R_{m,n}(t))_{t=1, \dots, M} = (r_1, r_2, \dots, r_M) \tag{1}$$

Thus, the shortest paths between every pair of skeleton endpoints are represented as Sequences of radii of the maximal disks at corresponding skeleton points. In this paper, the radius $R_{m,n}(s)$ is approximated with the values of the distance transform $DT(s)$ at each skeleton point s . Suppose there are N_0 pixels in the original shape S . To make the proposed method invariant to the scale, we normalize $R_{m,n}(s)$ in the following way:

$$R_{m,n}(s) = \frac{DT(s)}{N_0} \tag{2}$$

Where s_i ($i = 1, 2, \dots, N_0$) varies over all N_0 pixels in the shape. The shape dissimilarity between two shape paths is called a path distance. If R and R' denote the vectors of radii of two shape paths sp and sp' respectively, the path distances pd between sp and sp' is:

$$pd(sp, sp') = \sum_{i=1}^M \frac{(r_i - r'_i)^2}{|r_i + r'_i|} \tag{3}$$

The main motivation for Eq. (3) is the fact that similar shapes will have similar radii sequences on their corresponding skeleton paths. Formula (3) differs from the squared Euclidean norm by the scaling factor in the denominator, which has the effect of weighting the radii difference with respect to the radii values, e.g. if both radii are large, their difference must be significant. This is motivated by human perception, since the difference in thicker parts of objects must be more significant in order to be noticed. Path distance can also be used for finding the correspondence between two similar shapes.

Bayesian Classifier

Compared to the method in which uses contour segments and Bayesian classification to perform a recognition task, our method uses paths instead of contour segments. Since paths are normalized, our method does not require any invariant reference frame, and consequently the process of PCA26 can be removed. For a given query shape and a given shape class, we compute the probability that the shape belongs to the class. This step is repeated for all shape classes, and the query shape is then assigned to the class with the highest probability. Given a shape ω' that should be classified by Bayesian classifier, we build the skeleton graph $G(\omega')$ of ω' and input $G(\omega')$ as the query. For a skeleton graph $G(\omega')$, if the number of end nodes is n , the corresponding number of paths is $n(n - 1)/2$ compared to the number of parts $n!$. Then, the Bayesian classifier computes the posterior probability of all classes for each path sp' belongs to $G(\omega')$. By accumulating the posterior probability of all paths of $G(\omega')$, the system automatically yields the ranking of class hypothesis for the query shape ω' . We use Gaussian distribution to compute the probability p that two skeleton paths are similar:

(4)

$$p(sp' | c_i) = \sum_{sp \in G(c_i)} p(sp' | sp) p(sp | c_i) \tag{4}$$

For example, this probability is high for two different paths with small pd value.

For different datasets, α should be different. In our experiments, for the dataset of Aslan and Tari. $\alpha = 0.15$ and $\alpha = 0.05$ for Kimia dataset.²³ The class-conditional probability of observing sp' given that ω' belongs to class c_i is:

$$p(sp' | c_i) = 1 / |G(c_i)| \tag{5}$$

We assume that all paths within a class path set are equiprobable, therefore:

$$p(sp | c_i) = 1 / |G(c_i)| \tag{6}$$

According to the probability that the query shape belongs to a given class, the posterior probability of a class given that path $sp' \in G(\omega')$ is determined by Bayes rule:

$$p(c_i | sp') = \frac{p(sp' | c_i)p(c_i)}{p(sp')} \tag{7}$$

Similar to the above assumption, $p(c_i) = 1/M$. The probability of sp' is equal to

$$p(sp') = \sum_{i=1}^M p(sp' | c_i)p(c_i) \tag{8}$$

Through the above formulas, we can get the posterior probability of all paths of $G(\omega)$. By summing the posterior probabilities of a class over the set of paths in the query shape, we obtain the probability that it belongs to a given class. Obviously, the biggest one, C_m , is the class that input shape belongs to

$$C_m = \arg \max_{i=1, \dots, M} \sum_{sp' \in G(\omega')} p(c_i | sp') \tag{9}$$

Experiment and Result

In this section, we evaluate the performance of the proposed method based on the database of Aslan and Tari. Aslan and Tari database includes 14 classes of articulated shapes with 4 shapes in each class. We use each shape in this database as a query, and show the classification result of our system in Fig.5. We used leave-one-out classification, i.e. the query shape was excluded from its class:

class	query	result	query	result	query	result	query	result
1		1		1		11		9
2		2		2		2		2
3		3		3		6		3
4		4		4		4		4
5		5		5		5		5
6		6		6		6		6
7		7		7		7		7
8		8		8		8		8
9		9		9		9		9
10		10		10		10		10
11		11		11		11		11
12		12		12		12		12
13		13		8		13		13
14		14		14		14		14

Fig. 5- Results of the proposed method on Aslan and Tari database, since each class is composed of four shapes, the class of query and the result should be the same. Red numbers mark the results where this is not the case.

The table in Fig. 5 is composed of 14 rows and 9 columns. The first column of the table represents the class of each row. For each row, there are four experimental results which belong to the same class. Each experimental result has two elements. The first one is the query shape and the second one is the classification result of our system. If the result is correct, it should be equal to the first column of the row. The red numbers mark the wrong classes assigned to query objects. Since there is only one error in 56 classification results, the classification accuracy in percentage by this measure is 98.2%. In fact, the only error is reasonable. Even a human can misclassify it. The query shape is very similar to the star, which is class 8. Therefore, in some sense, we can conclude that all of our results are correct. We compared our method to the method presented by Sun and Super. In their method used the same Bayesian classifier but based on contour parts. Their meth-

od yields four wrong results for 56 query shapes, which yields the classification accuracy of only 92.8%. Since Aslan and Tari did not present results on their entire database, we cannot directly compare the recognition rate of our method to Ref. 3. However, we were able to compare our method to the inner distance18 on this dataset. Inner distance18 obtains only 94.64% through the nearest neighborhood classification, though it can solve the articulated shape classification very well. The classification time for all 56 shapes with the proposed method takes only 5 min on the PC with 1.5GHZ CPU and 512M RAM. In comparison, Sun and Super's method took 13 min on the same computer.

Applications

The 2D shape matching and retrieval has its applications in the following fields

- 2D Shape recognition
- Image / document processing
- Pattern Recognition
- Biological part matching

2D Shape Recognition

In the field of 2 D shape matching & recognition, an application can be developed so that any image may be captured using web camera. Then the captured image may further be processed followed by binarization, skeletonization, and skeleton segmentation & labeling in the computer to obtain its tree. Finally the image recognition can be done using tree matching technique. The database containing a huge number of 2D shapes may undergo this type of recognition.

Image/Document Processing

Image processing and shape analysis are important and useful in many document processing applications, such as segmentation of connected characters, recognition of confusing character shapes, etc.

Pattern Recognition

In computer vision, patterns are typically defined by undirected graphs of features (graph vertices) and their relations (graph edges) and, so, graph matching is a core problem to solve in such tasks as character recognition, object identification, and shape analysis. In these areas, many methods have been developed for solving exact matching (one-to-one vertex matching between (pairs of graphs), but less so for establishing correspondences between sets of vertices of the two graphs ("inexact" graph matching).

Biological Part Matching

The proposed method of shape recognition can be extended to the field of biological parts matching. The finger print recognition (biometric machines) or any other part of a human body can be matched and recognized using this technique. The same can find a big scope in the field of biological part matching for detection and operation of human parts.

Conclusion

A novel object recognition method based on similarity of skeleton graphs is presented. The most significant contribution of this paper is the novel approach to skeleton graph matching. We represent a skeleton as a set of geodesic paths between skeleton endpoints. The paths are compared using sequence matching. The

proposed approach does not require any complicated strategies for tree/graph matching based on editing of topological structures and complicated weighting of branches/nodes. In addition to superior performance, the proposed method also reduces the time cost. Moreover, the fact that our representation of skeletons is based on their endpoints opens a possibility of new applications. We are able to compute the main symmetry axes of articulated objects by computing self similarity of skeleton divisions induced by pairs of endpoints. The performance of our method is limited in the presence of large protrusions, since they require skipping a large number of skeleton endpoints. However, we believe this limitation can be solved with partial matching, for example, when the dissimilarity is computed only for the pair of sub graphs that are most similar. Our method is not limited to skeleton graphs. Our future work will also focus on matching any weighted graphs.

References

- [1] Blum H. (1973) *J. Theoretical Biology*, 38, 205-287.
- [2] Sebastian T.B. and Kimia B.B. (2005) *Signal Processing*, 85 (2), 247-263.
- [3] Basri R., Costa L., Geiger D. and Jacobs D. (1998) *Vision Research*, 38, 2365-2385.
- [4] Huttenlocher D.P., Klanderman G.A. and Rucklidge W.J. (1993) *IEEE Trans. Pattern Analysis and Machine Intelligence*, 15(9), 850-863.
- [5] Belongie S., Puzhicha J. and Malik J. (2002) *IEEE Trans. Pattern Analysis and Machine Intelligence*, 24(4), 509-522.
- [6] Shaked D. and Bruckstein A.M. (1998) *Computer Vision and Image Understanding*, 69(2), 156-169.
- [7] Zhu S.C. and Yuille A.L. (1996) *Int'l J. Computer Vision*, 20 (3), 187-212.
- [8] Liu T. and Geiger D. (1999) *Int'l Conf. Computer Vision*, 456-462.
- [9] Geiger D., Liu T. and Kohn R.V. (2003) *IEEE Trans. Pattern Analysis and Machine Intelligence*, 25(1), 86-99.
- [10] Di Ruberto C. (2004) *Pattern Recognition*, 37(1), 21-31.
- [11] Pelillo M., Siddiqi K. and Zucker S.W. (1999) *IEEE Trans. Pattern Analysis and Machine Intelligence*, 21(11), 1105-1120.
- [12] Pelillo M. (2002) *IEEE Trans. Pattern Analysis and Machine Intelligence*, 24(11), 1535-1541.
- [13] Kimia B.B., Tannenbaum A.R. and Zucker S.W. (1995) *Int'l J. Computer Vision*, 15(3), 189-224.
- [14] Sebastian T.B., Klein P. and Kimia B.B. (2001) *Int'l Conf. Computer Vision*, 755- 762.
- [15] Sebastian T.B., Klein P.N. and Kimia B.B. (2004) *IEEE Trans. Pattern Analysis and Machine Intelligence*, 26(5), 550-571.
- [16] Siddiqi K., Shkoufandeh A., Dickinson S. and Zucker S. (1998) *Int'l Conf. Computer Vision*, 222-229.
- [17] Siddiqi K., Kimia B., Tannenbaum A. and Zucker S. (1999) *Image and Vision Computing*, 17(5- 6), 365-373.
- [18] Siddiqi K., Shokoufandeh A., Dickinson S. and Zucker S. (1999) *Int'l J. Computer Vision*, 35 (1), 13-32.