Skeleton Graph Matching Based on Critical Points Using Path Similarity

Yao Xu, Bo Wang, Wenyu Liu, and Xiang Bai

Department of Electronics and Information Engineering,
Huazhong University of Science and Technology, Wuhan, 430074, China
{quickxu,wangbo.yunze}@gmail.com, {liuwy,xbai}@hust.edu.cn

Abstract. This paper proposes a novel graph matching algorithm based on skeletons and applies it to shape recognition based on object silhouettes. The main idea is to match the critical points (junction points and end points) on skeleton graphs by comparing the geodesic paths between end points and junction points of the skeleton. Our method is motivated by the fact that junction points can carry information about the global structure of an object while paths between junction points and end points can represent specific geometric information of local parts. Our method yields the promising accuracy rates on two shape datasets in the presence of articulations, stretching, boundary deformations, part occlusion and rotation.

Keywords: Skeleton, skeleton graph, graph matching, shape recognition, path similarity.

1 Introduction

Image matching is a fundamental aspect of many problems in computer vision, including object or scene recognition, solving for 3D structure from multiple images, stereo correspondence, and moving tracking [1]. In this paper, we focus on shape matching based on skeletal path similarity. Recent few years have witnessed a popular way in which skeleton is involved in the image matching problems. Integrating geometrical and topological feature of the object, skeleton (or Medial Axis) [2] plays an important role as a shape descriptor for object recognition. However, the fact that the topological structure of skeleton trees or graphs of similar objects may be completely different probably remains the most challenging aspect due to the sensitivity of skeletonization. This fact is illustrated in Figure 1,the objects from the same class may have different skeleton graph because of the instability of the critical points (junction points and endpoints). Thus some nontrivial edit operations (cut, merge, et al.) are inevitable to match skeleton graphs or trees. This paper presents a novel scheme for skeleton-based shape similarity measure. The proposed method is based on the similarity of shortest paths between end points and junction points of the pruned skeletons [3] to overcome the limitations mentioned above.

H. Zha, R.-i. Taniguchi, and S. Maybank (Eds.): ACCV 2009, Part III, LNCS 5996, pp. 456–465, 2010.

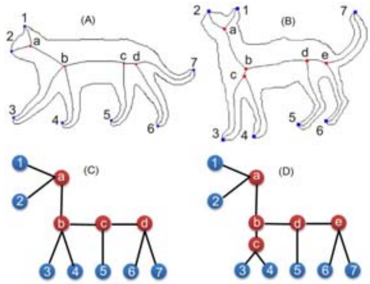

Fig. 1. Visually similar shapes in (A) and (B) have very different skeleton graph in (C) and (D)

As a preprocess for skeleton matching, we do the merge operation on the junction points of each skeleton based on their local context similarity. Then, the junction points and end points of different skeletons are matched in one-to-one correspondence with minimal cost, and the redundant junction points are not considered (cut operation on junction points). The penalty cost will be added for each redundant endpoint in order to compute the final shape similarity.

In section 2, the background of the related methods will be discussed. The way to match shapes based on the similarity of the skeleton paths between endpoints and junction points is introduced in section 3 and section 4. In section 5, the experimental results and analysis on two different datasets have been provided. At last, conclusion and future work are drawn out in section 6.

2 Related Work

The skeleton-based recognition methods are usually based on the graph or tree representation of the skeletons. Since the skeleton or medial axis is always organized into an Attributed-Relation Graph(ARG), the similarity between two objects can be measured by matching their ARGs. Zhu and Yuille [4] matched the skeleton graphs of objects using a branch-bounding method that was limited to motionless objects. Shock graph was a kind of ARG proposed by Siddiqi et al. [5], which was based on Shock Grammar. The distance between subgraphs was measured by comparing the eigenvalues of their adjacency matrices. Sebastian et al. have presented a scheme to compute the edit distance between the shock graphs [6]. Liu et al. [7,8] can deal with the problem when the two shapes have different amount of junction points in their skeleton graph. Demirci et al. transform weighted graphs into metric trees for accurate matching [9]. Aslan and Tari proposed an unconventional approach to shape recognition using unconnected skeletons in the course level [10]. Bai et al. proposed a method to match ARGs based on the shortest paths between endpoints [11]. The approach does not require any editing of the skeleton graph, however, only endpoints were used for matching in their framework without using the explicit structure of parts.

Motivated by the skeletal path representation [11], our proposed method utilizes the shortest paths between all the pairs of end points and junction points to represent a context of local structures. Utilization of merge or cut operation similar to [8] in matching phase is used for finding the optimal correspondence between the critical points on different shapes.

3 Shape Representation with Skeleton Paths

In this paper, all the skeletons for shape matching are extracted and pruned by the method introduced in [3].

A critical point (endpoint/junction point) can be called a node (end node /junction node) in a skeleton graph, and the shortest paths between every pair of nodes are represented as sequence of radii of the maximal disks at corresponding skeleton points [11]. If there is no other junction node on the path between an end node and a junction node, the end node and the junction node is said to be connected. The shortest path between a pair of end nodes on a skeleton graph is called a **end-to-end path**. The path between an end node and the nearest junction node on a skeleton graph is called a **junction-to-end path**. In addition,the path between different junction nodes is called a **junction-to-junction path**. We show a few example skeleton paths in Fig 2. Let *sp* denotes a skeleton path. We sample the path *sp* with M equidistant points, which are all skeleton points. Let $R(t)$ denotes the radius of the maximal disk at the skeleton point with index t in *sp*. Let L denotes the length of *sp*, R denotes a vector of the radius of the maximal disks centered at the M sample skeleton points on *sp*:

$$R = (R(t))_{t=1,2,\ldots,M} = (r_1, r_2, \ldots, r_M) \tag{1}$$

In our method, the radius $R(S)$ is approximated with the values of the distance transform $DT(s)$ at each skeleton point s. Suppose there are N_0 pixels in the

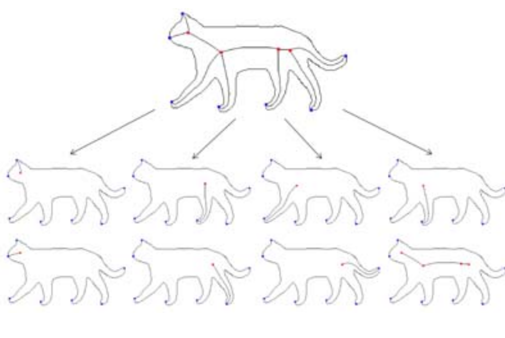

Fig. 2. Some of local paths (in red) in the cat's skeleton

original shape S. To make the proposed method invariant to the scale, *R(S)* is normalized in the following way:

$$R(S) = \frac{DT(s)}{\frac{1}{N_0}\sum_{i=1}^{N_0} DT(s_i)} \tag{2}$$

where $s_i(i = 1, 2, \ldots, N_0)$ varies over all N_0 pixels in the shape. The shape dissimilarity between two paths is called a path distance. If R and R' denote the vectors of radius of two paths *sp* and respectively, L and L' denote the lengths of the two paths *sp* and respectively, then the path distance pd between *sp* and textslsp' is:

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

Where α is a weight factor. In order to make the representation scale invariant, the path lengths are normalized.

4 Matching Nodes Using Skeleton Paths

Compared to the method in [11] that only used the end-to-end skeleton paths for matching the correspondence between end nodes, we match both junction nodes and end nodes using path similarity. The basic idea here is to match the junction nodes first using path similarity, then end nodes are matched using path similarity based on the correspondence of junction nodes. This is reasonable, since junction points always contain the important structure information for connecting the local meaningful parts of an object, and matching end nodes are easy when the correct correspondence of junction points are obtained. However, a challenging problem is the fact that junction nodes may not be stable, see example in Fig. 1. In order to solve this problem we do the merging operations based on the path contexts of junction nodes before matching process and the cut operations in matching process. In total, our method consists of two steps: mergence of junction nodes, matching critical nodes.

4.1 Mergence of Junction Nodes

We assume there are N junction nodes in a skeleton. The cost to merge two junction nodes V_i and V_j is defined as following:

$$cost(V_i, V_j) = \sum_{k=1}^{N} pd(sp_{i,k}, sp_{j,k}) \tag{4}$$

where $sp_{i,k}$, $sp_{j,k}$ denote the junction-to-end paths between every end node and junction nodes V_i and V_j, and k is the index of the end nodes in a counterclockwise direction. And the merging condition is as following:

$$cost(V_i, V_j) < N * \delta \tag{5}$$

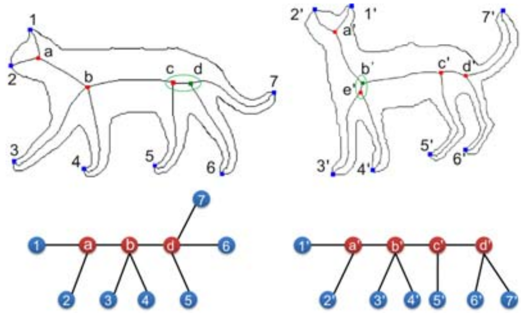

Fig. 3. The merge of junction nodes of two cats skeletons

where δ is a small value as a threshold. Any pair of junction nodes that satisfy the condition (5) are merged. Fig 3 illustrates an example of the merging process above. As Fig 3 shows, junction nodes c and d are merged as a single junction node d because they satisfy condition (5), so are b' *and* e'. In our implementation, we didn't merge the junction nodes c and d to one node actually. Instead only one of them will be selected for junction nodes matching.

4.2 Matching Critical Nodes

Let G and G' denote two graphs to be matched, and let the numbers of the junction nodes in G and G' be K and N, respectively. Here we assume $K \leqslant N$. It is easy to know that there are $C_N^K * K!$ kinds of matching cases and our aim is to obtain the optimal one-to-one matching with the minimal cost. In the case that the two graphs have different numbers of junction nodes, cut operation will be implemented by neglecting the redundant junction nodes. Specifically, we eliminate the junction nodes which are not matched. For example, there are one junction nodes V_1 in G, two junction nodes V_1', V_2' in G' , so 2 kinds of possible matching cases exist:

$$V_1 \longleftrightarrow V_1' \text{ or } V_1 \longleftrightarrow V_2'$$

In the former, V_2' is eliminated and in the latter one V_1' is eliminated. Of course, in most cases,more complex matching situations will occur. In Fig 4, after matching junction nodes and cut operation(in this case, the junction point d has been eliminated), critical points (in this case a,b,c) are obtained. Then, we get the common structure of the matched skeletons,and the critical nodes are in one-to-one correspondence.

For any pair of matched junction nodes V and V',suppose the numbers of end nodes adjacent to V and V' are m and n respectively. We assume $m \leqslant n$. Thus there are $C_n^m * m!$ kinds of matching choices. In this way, we can get U kinds of possible matching choices. Each matching choice has a matching cost and our aim is to obtain the one with the minimal cost. Assume there are P_k matched paths and Q_k unmatched paths in the kth matching choice(k is the index of matching choices), hence our model can be represented as following:

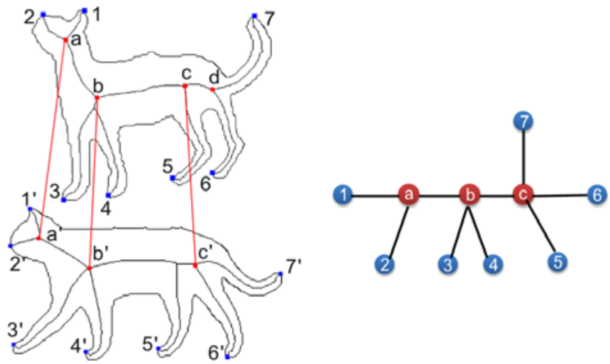

Fig. 4. Critical points achieved after junction node matching

$$\arg\min\,(cost_k), k = 1, 2, ..., U$$

$$cost_k = (1 + Q_k/P_k) \sum_{i=1}^{P_k} pd(sp_i, sp'_i) \tag{6}$$

where sp_i and sp'_i represent skeleton paths in the graphs to be matched, Q_k/P_k functions as a penalty factor if unmatched skeleton paths exist.

5 Experiments

In this section, we evaluate the performance of the proposed method in two parts: matching the critical nodes in the skeleton graphs, and the recognition performance of our method on two standard shape databases.

5.1 Correspondence Matching

To verify the accuracy of our method, shapes of various objects are matched and some representative results are shown. Besides the matching of two horses in Fig 5.(A) ,we test our method on several other examples. Since the structure of the horse is similar to the cat, our matching process finds the correct correspondence shown in Fig 5.(B). Fig 6 illustrates that the proposed method works well in the presence of articulation. Fig 7 also shows some matching results in the presence of occlusion or part missing. In Fig 7.(A) there is protrusion on the back of a cat, and in Fig 7.(B) two legs of a horse were removed. It demonstrates that the proposed method is able to obtain a correct correspondence even if parts of a shape are altered.

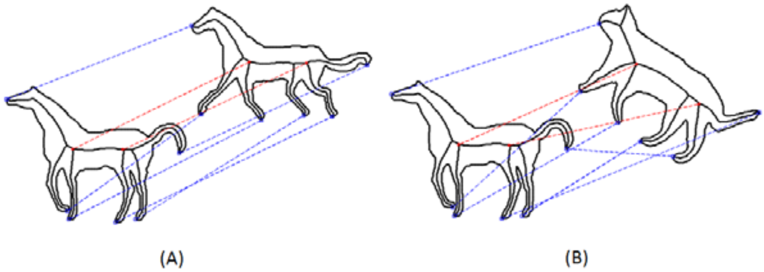

Fig. 5. Some representative results of correspondence matching

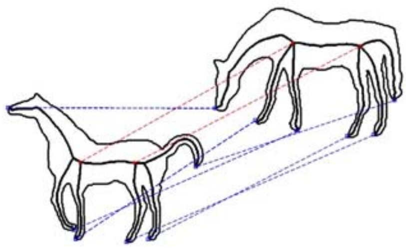

Fig. 6. The correspondence in the presence of articulation

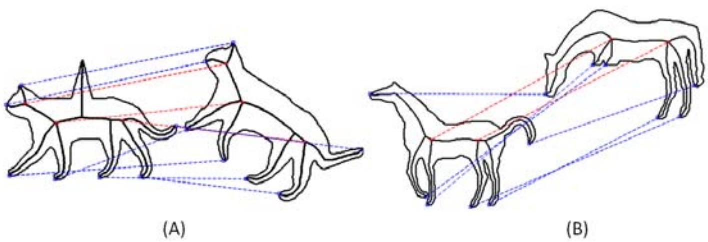

Fig. 7. The correspondence in the presence of occlusion or part missing

5.2 Robustness of Recognition

To evaluate the recognition performance of the proposed method, we test it on Aslan and Tari's two databases [10]. The first dataset includes 14 classes of articulated shapes with 4 shapes in each class, as shown in Fig 8. We use each shape in this database as a query. Several representative results are shown in Fig 9, where five most similar shapes are shown for the queries. Below each shape is the cost to match with the query. For each query, a perfect result should have three most similar shapes in the same class as the query. The distance in red marks an error where this is not the case.Encouragingly, the recognition rate on this dataset is 99.4% since there are only 1 errors in 168 query results. Moreover, we can easily observe that the wrong result is very similar to the query. For this dataset, we use parameters $M = 50, \alpha = 45$.

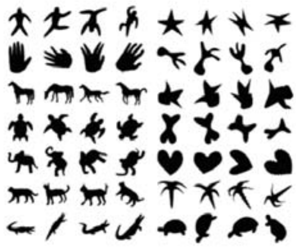

Fig. 8. Alan and Tari database [10] with 56 shapes

Table 1. Retrieval results on Alan and Tari 56 database [10]

Algorithm	1st	2nd	3rd
IDSC+DP [12]	53	51	38
Path Similarity [11]	55	55	53
Ours	56	56	55

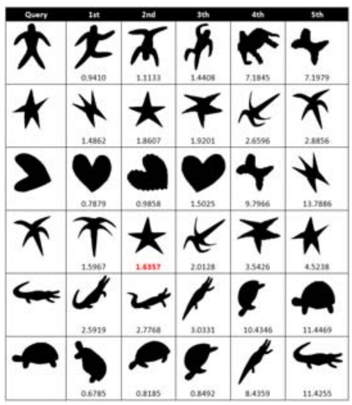

Fig. 9. Selected results of the proposed method on Alan and Tari database [10].Distance in red is the only error.

In Table 1, the result by the proposed method is compared to the result by other two recent shape matching methods. The proposed method performs better both than Inner Distance [12] in non-rigid deformations and Path Similarity [11], since we use the information of the junction nodes explicitly.

Our method is also tested on another bigger database provided by Aslan and Tari [13]. The database consists of 180 shapes which have 30 classes with 6 shapes in each class. For each shape, we check whether the 5 closest matches are in the same class as the query. Some typical results are shown in Fig 10. In the whole database, there are only 24 errors in 900 query results, so the recognition rate is 97.3%. The numbers of correct shapes for all 900 queries among the 1st,

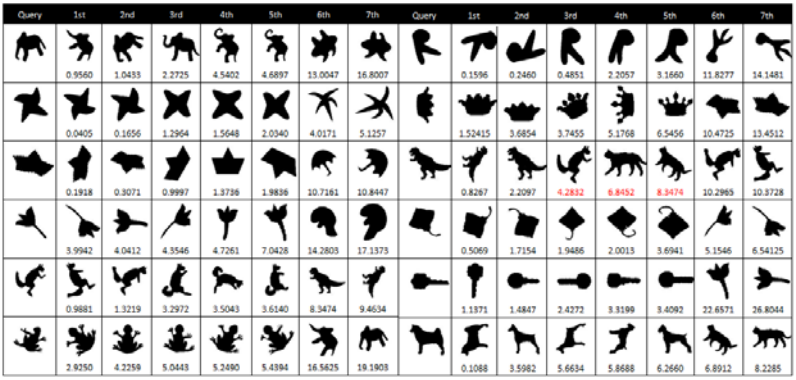

Fig. 10. Selected results of the proposed method on Tari'180 database [10]. Distance in red are errors

2nd, 3rd,4th, 5th closest matches are 180,179,174,175,168. Here, parameters are $M = 50, \alpha = 55$. We now analyze the computational complexity of the proposed method. Let M_i be the number of end nodes in the graph G_i, and let N_i be the number of junction nodes in G_i. Since the implementations in section 4.2 and 4.3 cost the most time, the time complexity of our method is approximately $O(M_i! * N_i!)$. However, since the number of junction nodes N_i has usually been significantly reduced to less than 7 after the merging process in section 4.1, the average time for matching per pair of shapes is very small. In this experiment, it is only 0.8 second.

6 Conclusion

In this paper, we propose a novel method to match skeleton graphs. The most important contribution is the merge and cut operation on junction nodes of skeleton graphs. The effect of these operations is the introduction of the structural information of the skeleton, which is very helpful in matching. As a result, our method is simple and efficient in correspondence matching even in the presence of occlusion and articulation. The experiment shows that the merge and cut process of junction nodes in our method have advantages over the method based only on path similarity. However, in our framework we didn't consider the case for many-to-one matching, which may be a limitation. In the future, our work will focus on classification based on the construction and unsupervised learning of tree union of skeletons.

Acknowledgement

This work was in part supported by Education Ministry Doctoral Research Foundation of China (No. 20070487028) and NSFC (No. 60873127). Xiang Bai was supported by MSRA Fellowship.

References

1. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. IJCV 60, 91–110 (2004)
2. Blum, H.: Biological shape and visual science. J. Theor. Biol. 38, 205–287 (1973)
3. Bai, X., Latecki, L.J., Liu, W.Y.: Skeleton pruning by contour partitioning with discrete curve evolution. IEEE Trans. PAMI 29, 449–462 (2007)
4. Zhu, S.C., Yuille, A.L.: Forms: A flexible object recognition and modeling system. IJCV 20, 187–212 (1996)
5. Siddiqi, K., Shkoufandeh, A., Dickinson, S., Zucker, S.: Shock graphs and shape matching. In: ICCV, pp. 222–229 (1998)
6. Sebastian, T.B., Klein, P.N., Kimia, B.B.: Recognition of shapes by editing their shock graphs. IEEE Trans. PAMI 26, 550–571 (2004)
7. Liu, T., Geiger, D.: Approximate tree matching and shape similarity. In: ICCV, pp. 456–462 (1999)
8. Geiger, D., Liu, T., Kohn, R.: Representation and self-similarity of shapes. IEEE Trans. PAMI 25, 86–99 (2003)
9. Demirci, F., Shokoufandeh, A., Keselman, Y., Bretzner, L., Dickinson, S.: Object recognition as many-to-many feature matching. IJCV 69, 203–222 (2006)
10. Aslan, C., Tari, S.: An axis based representation for recognition. In: ICCV, pp. 1339–1346 (2005)
11. Bai, X., Latecki, L.J.: Path similarity skeleton graph matching. IEEE Trans. PAMI 30, 1282–1292 (2008)
12. Ling, H., Jacobs, D.W.: Shape classification using inner-distance. IEEE Trans. PAMI 29, 286–299 (2007)
13. Baseski, E., Erdem, A., Tari, S.: Dissimilarity between two skeletal trees in a context. Pattern Recognition 42, 370–385 (2009)