Graph Matching for Object Recognition and Recovery

Lei He, Chia Y. Han, Xun Wang, Bryan Everding, William G. Wee

Department of ECECS, University of Cincinnati, Cincinnati, OH 45221-0030, USA

Abstract

A robust skeleton-based graph matching method for object recognition and recovery applications is presented. The object model uses both a skeleton model and contour segment models, for object recognition and recovery. Initially, the skeleton representation is created from the input contour that is provided by a deformable contour method (DCM). The skeleton is then matched against a set of object skeleton models to select a candidate model. Corresponding feature points or landmarks on the input and model contours are determined from their skeletons automatically. Based on the landmarks, the input and model contours are broken into contour segments. The input contour segments are then matched against the corresponding model segments for error analysis. For any large error in the segment mismatch, a fine-tuning process is performed to enhance the final segmentation result. The skeleton-based shape matching algorithm is illustrated by using a set of animal silhouette examples. Experiments of object recovery using real biomedical image samples, such as MRI knee and brain's corpus callosum images, have shown satisfactory results.

1 Introduction

Image understanding plays an important role in image analysis and computer vision applications. Generally it includes two key interrelated components: image segmentation and object recognition. As shown in Figure 1, the segmentation process yields contours, either exact or approximate, of objects of interest in images for object recognition. The object recognition process performs shape matching and the results can be fedback into the image segmentation process to increase the accuracy of the segmentaton results. This is normally referred to as object recovery. The work of this paper is focused primarily on the object recognition.

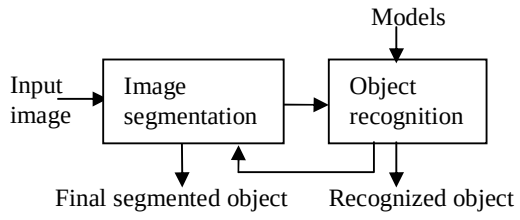


Figure 1: *Image understanding process*

The object recognition is realized through shape matching, by matching the skeleton graph of the input contour from a DCM with those of the models. The "best" matching skeleton pair determines the correct model and constructs the contours' landmark correspondences. The correspondences of the contours' segments follow automatically. Then, for the contour segments with a large error when compared with the matched model segments, a fine-tuning process, which is formulated as a maximization of a posteriori probability [4], is performed for final result. Two kinds of representations: skeleton model and contour segments model are involved. The skeleton model is constructed for shape matching and object recognition, and the contour segments model is for fine-tuning the final result. Both models are used in a complementary fashion.

The objective of this paper is to present a new skeleton-based shape matching approach, incorporating the correspondence of the skeleton edges to contour segments for shape matching and object recognition, yielding a robust and efficient model-based object recovery. This also provides a good alternate means to alleviate difficulties encountered in segmentation problems and to refine the results obtained from DCMs for complex image segmentation problems. Section 2 briefly reviews the DCMs for segmentation, object shape representations, and shape matching and object recognition methods. Section 3 gives the algorithm description. Experimental results on matching animal profiles for recognition and recovering shapes in biomedical images, are provided in Section 4. Section 5 draws the conclusions.

2 Background

One of the most advanced and popular image segmentation methods is the DCMs [1][2][12]. Recent progress in DCMs has advanced the state of the art significantly. However, contour extraction for detailed object recovery still remains a challenge when the contour is not readily present due to gaps, blur contour segments, or complex shapes. Furthermore, many challenges for the DCMs such as the selection of the initial contour, the geometric invariance for using models, the stopping criteria, parameter tuning, and other challenges remain. Nevertheless, DCMs can in most cases extract the desired object boundary, or a good approximation, which can then be used as the input contour for object recognition. Model-based DCMs can be formulated as snakes [13][14], level set methods [15][16][17] and parametric

deformable models [4][5][6][11]. The a priori knowledge of object shape is either embedded into the snake energy function [13][14] or the contour deformation velocity [15][16][17] to constrain the admissible deformation range. Parametric deformable models are commonly used when the a priori shape knowledge of the object of interest is available and can be represented by a small number of parameters with certain distributions.

There are several well-established object shape representations [27]. They include chain code (CC), B-spline (BS), Fourier descriptors (FD), Wavelet descriptors (WD), and skeleton (SK). The skeleton representation consists of the locus of centers of maximal disks (CMDs) that can be inscribed within the object and a maximal disk is not completely contained in any other disk totally included in the object. Different from the boundary-based object representation methods (CC, BS, FD and WD), skeleton is a region-based method, which emphasizes the structural properties that go beyond simple boundaries, e.g., location of convex-parts, width and length of each part. A variety of methods [18][19][20] were proposed to compute the object skeleton, and the thinning [18] and distance transformations (DT) based methods [19][20] are the most popular.

Selecting an appropriate object shape representation is a crucial problem for object recognition and recovery applications. The desired representational properties include: uniqueness (UNI), invariance to object geometric transformations (translation (T), rotation (R), and scaling (S)), noise robustness (NR), complexity (CM), reversibility to the original shape (RV), and multi-scale (MS). Table 1 compares these representations in terms of the desired properties. As this table indicates, skeleton provides many significant features suitable for our purpose except the low noise robustness. However, this problem can be easily solved without adding a significant number of additional computations to our algorithm. Moreover, skeleton is the only representation method providing object structural information as mentioned above, which is important for object shape matching. Therefore, skeleton was selected for our approach.

	UNI	CM	INVARIANCE			NR	RV	MS
			T	R	S			
CC		LOW	√			LOW	√	√
BS		MID	√	√	√	HIGH		
FD	√	HIGH	√	√	√	MID	√	√
WD	√	HIGH		√	√	HIGH	√	√
SK	√	MID	√	√	√	LOW	√	√

Table 1: *The comparison of shape representation methods*

Traditionally there are three kinds of shape matching methods: template matching, statistical classification and structural classification [27]. Template matching is the simplest shape identification method by comparing the input shape to a list of stored shape representations (templates), which is usually formulated as a parameterization problem, with a cost function to be minimized [21][22]. It generally can handle only simple cases where there is only a small geometric variation between the input shape and templates. Statistical approaches use only a set of selected shape measures or features that are more resilient to shape variations [6][23][24] to match shapes. Our previous work [6] proposed a curvature based contour matching approach that is simple to implement and invariant to object geometric transformations. However, it can't match complex shapes because curvature is a local description and sensitive to noise. Sclaroff et al. [23] proposed a modal matching method to establish the correspondence of contour feature points and to recognize objects based on the eigenmode description. However, as pointed out in [24], without the connectivity information of the contour, the algorithm is not guaranteed to generate a legal set of correspondences. Hill et al. [24] proposed a three-step algorithm for automatically identifying the landmarks on two contours and constructing their correspondences. The results are good but the computing complexity is high due to its iterative processes in all the three steps. Structural classification methods select the correct model for the input shape by comparing their structure, i.e., their respective ordered composition of simple sub-patterns or shape primitives. Shapes are represented by such a composition of shape primitives. The selection and segmentation of shape primitives are not easy, and generally depend on the preference and experience of the user. Zhu et al. [8] constructed skeleton models for object recognition. The algorithm computes the similarity between the input and the model for all possible matches. The primitive segmentation and parameter tuning is sensitive to noise, and it has a high computation complexity with multiple parameters to be tuned. Siddiqi [9] proposed a theory of shock graphs for shape representation and matching. The matching algorithm is to find the graph nodes correspondences based on the topological and geometrical similarity of the corresponding nodes. The shock segmentation algorithm is rather complex and sensitive to noise. Both above algorithms are for object recognition purpose, and not well suited for shape recovery applications since correspondences on contour landmarks can't be determined from their skeleton edges.

The proposed skeleton-based shape matching method is a combination of both the structural and the statistical methods, in which the object skeleton is represented as an ordered tree (graph without loop) or a string of skeleton edges (shape primitives). The connectivity relationship among skeleton edges and the geometrical features of skeleton edges are used for skeleton matching and will be explained in the next section.

3 Proposed Approach

This skeleton-based shape matching and recovery approach can be described in four major steps: (I) skeleton processing, (II) skeleton model construction,

(III) skeleton matching and model detection, and (IV) contour segment correction, as shown in Figure 2.

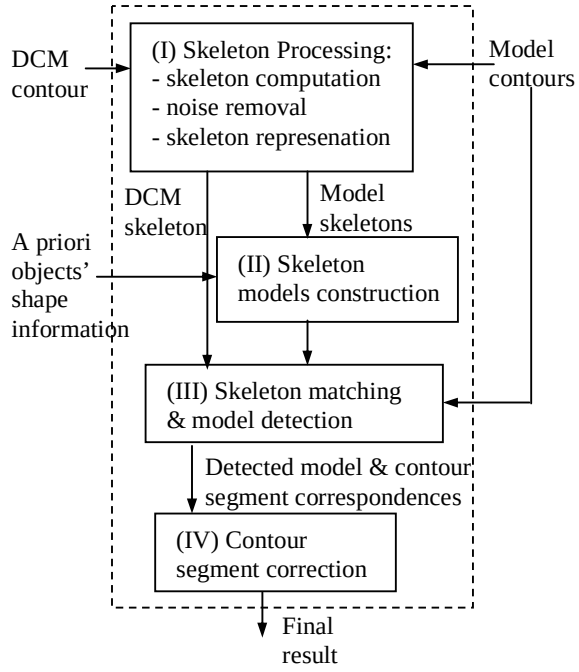


Figure 2: *Object recognition and recovery flowchart*

(I) Skeleton Processing

The first step consists of determining the skeleton, removing noise or unwanted information, and representing the skeleton. Skeleton computation is normally based on the thinning or distance transformation (DT). In general, thinning methods produce connected skeletons that can't be used to reconstruct the original object shape, while skeletons generated by DT-based methods can. However, the latter generally are not connected and are sensitive to object boundary noise. To keep the advantages and overcome the shortfalls of these methods, Svensson et al. [7] proposed a new method that is based on the idea of iteratively thinning the distance transformation of an object. The resulting skeleton is connected, invariant to object geometric transformations and capable of reconstructing the object shape, which is suitable for object shape matching and recovery applications. Therefore, this method is used to compute the skeletons for both the input contour and the model contours.

The notation used for the major skeleton entities is illustrated in Figure 3(a) with a skeleton example of a side view of a quadruped animal. The curve segments of actual skeleton edges are simplified as line segments in a *skeleton graph*. The attributes of a skeletal point on the derived skeleton graph are its distance value and if it is a CMD or not. A skeletal point having only one skeletal point in its eight neighbors is defined as an *ending node* (E-node) (e.g., A, B, E), and a skeletal point having at least three skeletal points in its eight neighbors is defined as a *bifurcation node* (B-node) (e.g., C, D, J). All others are normal nodes. After the skeleton computation, skeletal points can be linked to form skeleton edges. A skeleton edge (SE) includes all the skeletal points between a B-node (or an E-node) and another B-node. A SE that has two B-nodes is defined as a *primary* SE (e.g. CD and DJ), otherwise it's a *normal* SE (e.g. AC, BC, ED).

Skeleton can be pruned based on the importance of the skeleton edges. For each normal skeleton edge, we can determine its importance by comparing the original shape with the shape reconstructed [10] from the skeleton without the edge. A large difference in the shapes indicates that the edge is important; otherwise, it is due to shape noise and can be cut out from the skeleton graph.

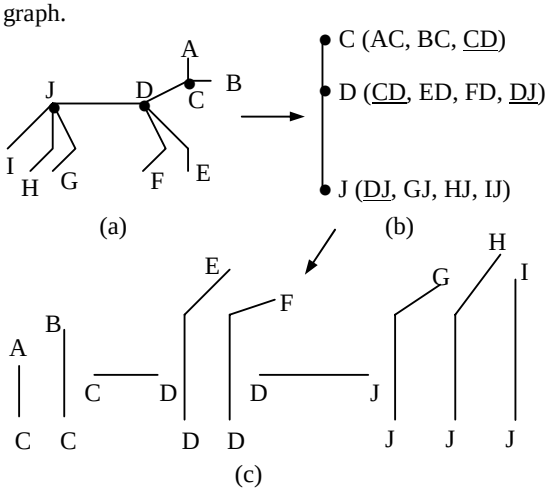


Figure 3: *A skeleton graph (a), its tree representation (b) and SE string (c)*

Only one type of primitive - skeleton edge is used, instead of multiple primitives, such as the worm, circle, and shocks used in [8][9]. The complexity and sensitivity to noise of primitive segmentation algorithm is then greatly reduced. For effective processing, the skeleton graph is further represented by a tree and a string, as shown in Figure 3(b) and 3(c), respectively. A B-node tracing algorithm to generate a skeleton tree is applied by recording SEs sequentially, starting from the B-node connected with the shortest normal SE. The tree nodes are the B-nodes, and the tree edges are the primary SEs and are underlined in the figure. .

(II) Skeleton model construction

After the skeleton is computed or constructed, the following three major types of information – rigid transformations, nonrigid deformations, and model specific information - can be gathered into the model and used in the next matching phase.

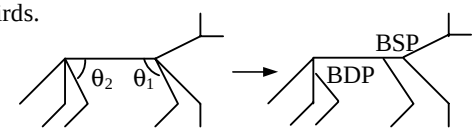
1). Rigid transformations

a). The admissible connectivity relationship among the SEs. This relationship can be parameterized by the bifurcation angles, the angles subtended by the adjacent SEs (e.g., θ_1, θ_2 in Figure 4(a)).

b). The admissible translations of SEs caused by the bifurcation delay (BDP)/splitting (BSP) phenomena [8] (Figure 4(b)) in the configuration of SEs.

2). Nonrigid deformations: The object shape variations can be described as the Gaussian distribution on the distance values of skeletal points.

3). Model specific rules: A typical example would be that the limbs should be the protrusions on the same side of body for any side-view of quadruped mammals and birds.



(a) (b)

Figure 4: *Skeleton model and bifurcation delay/splitting*

(III) Skeleton matching and model detection

The skeleton matching and model detection algorithm encompasses three major steps: searching for possible matches or correspondence between the SEs of the input skeleton and the skeleton models; checking the validity of identified matches based on the model information; and computing similarity measure to determine the correct model and SE correspondences.

As a skeleton can be represented as a tree or a string comprising all the SEs of the skeleton, the skeleton matching problem can be formulated as a graph (tree) matching [8][9] or string matching problem [13][25][26]. A branch-and-bound string matching algorithm is used in our application. In essence, the string matching algorithms apply the minimum edit operations (substitution, deletion and insertion) to transform one string to another string and the associated transformation cost can be used to measure the similarity between the two strings. The larger the cost, the less similar the two strings are. The algorithm searches over all possible matches between the two strings for the same number and connectivity relationship of B-nodes, and the same numbers of primary and normal SEs in each matched B-node group pair. There could be several matches due to the different SE recording order within a B-node group. This process repeats between the input skeleton and all models.

For each match found, several validity checks based on the rules and information collected in Step (II) can be used. First, we can identify the BDP/BSP in the input skeleton according to the model specific information. The model specific information is then used to check the validity of the input skeleton string representation, which is to detect the error of selecting the starting B-node incorreclty. When an incorrect SE string for the input skeleton is generated due to BDP/BSP, the algorithm continues with other possible string representations in order to satisfy the validity checking. Finally, after removing the BDP/BSP, the rigid information such as bifurcation angles in the input skeleton can be computed. The validity of both the BDP/BSPs and the bifurcation angles in the input skeleton are then checked by comparing them with those of the model. This process ensures that the SEs and B-nodes recording order won't affect the matching result, thus enhances the algorithm robustness.

The last task in this step is to detect the correct model from a set of models passed above checking and determining the correct SE correspondences based on a similarity function between the input and a model. Given an input shape skeleton string $D=(d_1, d_2, \ldots, d_R)$ and its corresponding model skeleton string $M=(m_1, m_2, \ldots, m_R)$, their similarity function is defined as:

$$S(D,M) = e^{P}/(E_1 E_2 E_3) \quad (1)$$

, where P is the thickness similarity of the input shape D and a model M, weighted by the shape error weight E_1, the thickness error weight E_2, and the length error weight E_3. Here $P = \frac{1}{R}\sum_{i=1}^{R} p(d_i, m_i)$, with

$p(d_i, m_i) = \prod_{j=1}^{N} \frac{1}{\sigma_j \sqrt{2\pi}} e^{-\frac{(rd_{ij} - rm_{ij})^2}{2\sigma_j^2}}$, rd_{ij} and rm_{ij} are the normalized distance values of the jth corresponding points on the skeleton edges d_i and m_i, respectively, after normalizing the skeleton edges d_i and m_i to be the same length N. rd_{ij} and rm_{ij} are normalized with respect to the largest distance value on d_i and m_i. σ_j is the variance of the normalized rm_{ij}. In the experiments, we use $\ln(p(d_i, m_i))$ instead of $p(d_i, m_i)$ to avoid a very small P. To compute the E_1, which is to measure the shape difference between D and M, the input SE d_i should be transformed first to the coordinate system of its corresponding model SE m_i as d_i'. Then they should be normalized to have the same length. After the transformation and normalization, E_1 can be computed as: $E_1 = \frac{1}{R}\sum_{i=1}^{R} e_1(d_i, m_i)$, with

$e_1(d_i, m_i) = \frac{1}{N}\sum_{j=1}^{N} \sqrt{(xd_{ij}' - xm_{ij}')^2 + (yd_{ij} - ym_{ij})^2}$,

(xd_{ij}', yd_{ij}') and (xm_{ij}, ym_{ij}) are the coordinates of the jth corresponding points on d_i' and m_i, respectively. The E_2 is to measure the difference of the average thickness ratio between D and M, and it is formulated as: $E_2 = \frac{1}{R}\sum_{i=1}^{R} e_2(d_i, m_i)$, with

$e_2(d_i, m_i) = \frac{|\overline{rd_i}/\overline{rd} - \overline{rm_i}/\overline{rm}|}{\overline{rm_i}/\overline{rm}}$, $\overline{rd_i}$ and $\overline{rm_i}$ are the average distance values of d_i and m_i, $\overline{rd}$ and $\overline{rm}$ are the average distance values of all the skeletal points on D and M. The E_3 is to measure the difference of the length ratio between D and M, and it is formulated as: $E_3 = \frac{1}{R}\sum_{i=1}^{R} e_3(d_i, m_i)$, with

$e_3(d_i, m_i) = \frac{|ld_i/ld - lm_i/lm|}{lm_i/lm}$, ld_i and lm_i are the lengths of d_i and m_i, ld and lm are the lengths of all the skeleton edges of D and M.

The similarity value is computed for all possible matches, and the model that has the largest value with certain SE correspondences to the input shape is selected as the correct model. Meanwhile, the SE corre-

spondences producing the largest similarity value is considered as the correct match between the model and the input. When the skeleton of the input shape has additional or missing SEs, we only compute the similarity function between the matched SEs.

(IV) Contour segment correction

The contour landmarks are determined by the E-nodes of its SEs. By reconstructing the shape from the skeleton with the E-nodes removed, the landmarks are then identified from the missing points in the reconstructed shape with respect to the original shape. These landmarks are then used to determine the contour segment correspondences. The corresponding contour segments on the input and model contours are compared for errors. The error computation is the same as the computation of the shape error weight E_1, in which the skeleton edges are replaced as the contour segments.

The final contour segment correction for the segments with large errors is the same as in [4][6]. Given a set of samples, many methods [15][16][17] are proposed to construct the object contour model. Our contour model is obtained in a similar fashion. First a domain expert draws the object boundaries on a set of model images and gives a rough alignment on the contours. Then the contours are linearly normalized to be the same length to compute the mean shape of the model. The model variance can also be derived from the sample contours using the principal component analysis as in [16][17]. However, due to the lack of samples, the contour model is obtained by assigning a Gaussian distribution on the mean contour which is represented by an Elliptical Fourier descriptor as in [4][6].

4 Experiments

In this section, two experiments are used to illustrate our algorithm: the first experiment uses a set of animal silhouette shapes to demonstrating the graph matching algorithm for object recognition; the second uses biomedical image samples, such as MRI knee and corpus callosum images to show the object recovery process.

4.1 Skeleton-based shape matching

Skeleton-based graph matching for model detection are demonstrated by selecting the most similar shape for the input test shapes, shown in Figure 5, from a set of animal model shapes shown in Figure 6. Both the input and the model shapes include several quadruped mammals and birds, which were obtained from some biology books, photographs, and the Brown University Stimuli (http://www.cog.brown.edu/~tarr/stimuli.html).

As described in Section 3, the skeleton-based shape matching process starts from the skeleton processing for both the input and model shapes. The skeletons are obtained after noise removal and are drawn within the shapes in Figure 5 and 6. The skeleton models are constructed directly on the model shapes by following the algorithm in Step (II) of Section 3. After skeleton matching is used to locate the corresponding skeleton edges between the input and all the possible models, the similarity functions are computed for all valid matches. The similarity function computation algorithm firstly normalizes the shapes such that the corresponding skeleton edges have the same length. Thus the input and model shapes can be exchanged without affecting the similarity measure between them. Table 2 shows the linearly normalized similarity values (0-10), with the highest values being shown in boldface.

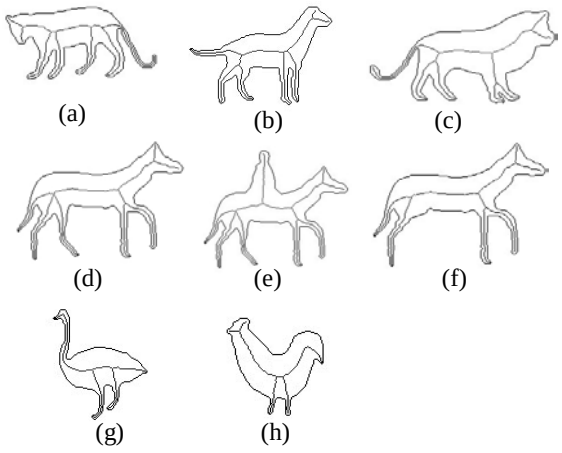


Figure 5: *The input animal shapes and their skeletons*

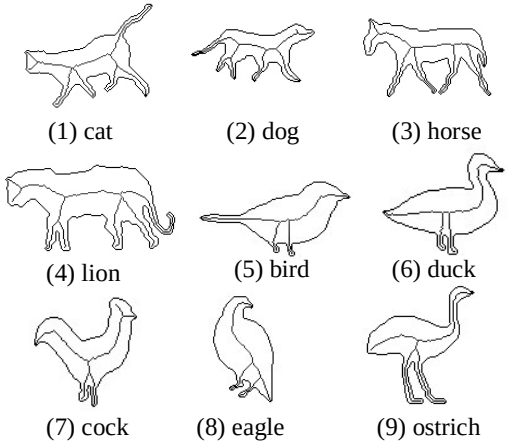


Figure 6:*The model animal shapes and their skeletons*

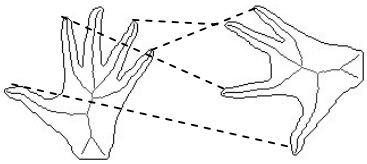
	(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)
(1)	**8.90**	2.79	0.18	0.65	0.64	**3.81**	-	-
(2)	0.01	**4.66**	0.01	2.22	1.98	2.05	-	-
(3)	4.45	3.27	0.55	**3.70**	**3.61**	0.97	-	-
(4)	5.12	0.28	**5.12**	0.08	0.06	0.88	-	-
(5)	-	-	-	-	-	-	0.44	2.49
(6)	-	-	-	-	-	-	1.74	0.86
(7)	-	-	-	-	-	-	3.94	**3.53**
(8)	-	-	-	-	-	-	3.17	3.30
(9)	-	-	-	-	-	-	**4.08**	1.70

Table 2: *Similarities between input and model shapes*

By looking at Table 2, it can be easily seen which model the input shape is most similar to. Also, we can see that the quadruped mammals don't match with birds

because their skeleton structures are not the same. This is a direct result from the matching algorithm in Step (I) of Section 3. This is different from most of the existing shape matching algorithms [8][9], which compute the shapes' similarity no matter what. This shows that the proposed algorithm provides a flexible choice: it can make decision to compute the similarity between two shapes or not according to their skeleton structure. If the two objects are detected to belong to the same species, i.e., they have the same skeleton structure, the similarity between them can be computed. Otherwise, the algorithm does not compute the similarity.

As stated earlier, the skeleton-based graph matching problem is solved by a string matching approach, which enables the proposed algorithm to handle the structure errors by performing string components (skeleton edges) deletion and insertion operations. An example for a single SE error (missing) is shown in Figure 7, in which an input hand shape, Figure 7(b), with the middle finger being cut out is matched with a normal hand model, Figure 7(a).



(a) Model (b) Input

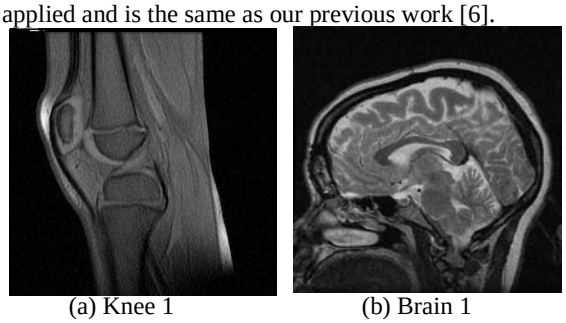
Figure 7: *An example of SE missing matching*

The fingers can be easily identified in the skeleton according to their lengths relative to other normal SEs. For this example, there are five possible matches between the input and the model. In each match a finger, called "*finger*", is missing in the model shape, we can compute the similarity value $S(D, M_{finger})$. Therefore, each time only four fingers in the model shape are matched with the four fingers in the input shape, the computed similarities are $S(D, M_{thumb})$= 4.4962, $S(D, M_{forefinger})$=7.2026, $S(D, M_{middle\ finger})$=7.5757, $S(D, M_{ring\ finger})$=6.5569, and $S(D, M_{small\ finger})$=5.5557. The correct match is detected with the largest similarity value when the middle finger in the model was not included in the matching. The resulting finger correspondences are drawn in the Figure 7. Three other examples for the input animals that include a single SE error are shown in the Table 2, Figure 5(e), (f) and (h). Similarly, multiple SE errors can also be handled with our algorithm. The ability of the algorithm to handle SE errors is a tradeoff between the admissible match and user requirements, i.e., the stricter the user requirements, the fewer number of the admissible matches.

4.2 Shape recovery

In this section, two experiments on two MRI knee and corpus callosum images are used to illustrate how to apply the skeleton-based graph matching algorithm for object recovery. To save space, only one sample of MRI knee and brain image is shown in Figure 8. The image in Figure 8(a) is an example of midline sagittal MRI knee images of size 256 by 256. To segment the femoral condyle (top portion of the knee), the goal is to delineate the top segment of the contour that separates the semicircular portion of the femur from the stem. The challenge is that there is a blurred edge segment along the middle top boundary, while the left and right sides of the femoral condyle are rather darker than the middle region. This prevents the deformable contour to reach the real boundary on the two sides before it flows out from the top.

The landmarks on the contours can be detected using the approach stated in Step (IV) of Section 3. These landmarks are then used to determine the contour segment correspondences. For each contour segment that has a large error compared to the corresponding model contour segment, the correction process is applied and is the same as our previous work [6].

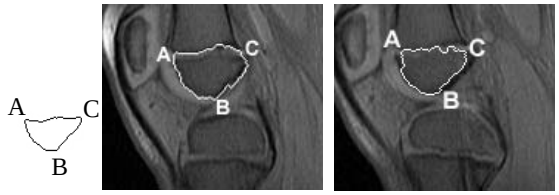


(a) Knee 1 (b) Brain 1

Figure 8: One sample of *the original MRI knee and brain image*

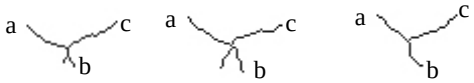
In the first experiment, two input knee shapes, Figure 9(b), (c), obtained by the DCM method [3] are used to match the knee model, Figure 9(a). The knee model contour was obtained as described in the step (IV) of Section 3, from a set of eight sample images. The skeletons of the knees after noise removal are shown in Figure 10. In the experiment, the E-nodes of the SEs correspond to the corner points on the original shape, which are used as the landmarks dividing the whole contour into three contour segments. For example, the *a*, *b*, and *c* E-nodes on the model skeleton, Figure 10, correspond to A, B and C landmarks on the model shape, Figure 9. Thus the contour landmark and then the segment correspondences are constructed following the SE E-nodes correspondences.

The contour segment error computation and final local contour segment correction procedure for the segments with large errors are obtained as stated in Step (IV) of Section 3. For the first input contour, in Figure 9(b), two large error segments of BC and CA, which may indicate the segmentation difficulties mentioned earlier, have to be further refined. The constructed final result is shown in Figure 11(a) with both segments BC and CA being corrected and shown as segment B'C' and C'A'. Similary, the two large error segments of AB and CA in Figure 9(c) are corrected and shown as segment A'B' and C'A' in Figure 11(b). The contour segments errors before and after segments correction are listed in Table 3. We can see the final shapes have less segment errors than those before correction.



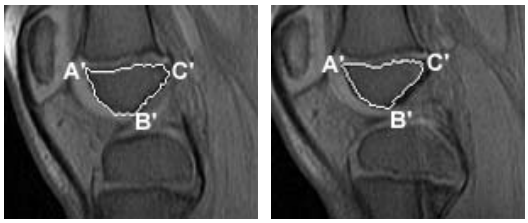
(a) Model (b) Knee 1 result (c) Knee 2 result

Figure 9 *The knee model and input shapes obtained by a DCM* [3]



(a) Knee model skeleton (b) Knee 1 skeleton (c) Knee 2 skeleton

Figure 10: *The knee skeletons of the knee model and shapes in* **Figure 9** *after noise removing*



(a) Knee 1 final result (b) Knee 2 final result

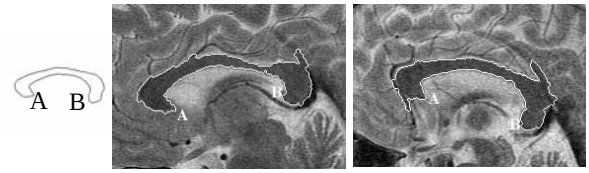
Figure 11: *The resulting knee contour after contour segments corrections*

Before Correction (**Figure 9**(b))		After Correction (**Figure 11**(a))	
Distance Errors	Segment points #	Distance Errors	Segment points #
0.624	42 (AB)	0.333	39 (A'B')
4.217	40 (BC)	1.629	32 (B'C')
2.458	48 (CA)	1.109	52 (C'A')
Before Correction (**Figure 9**(c))		After Correction (**Figure 11**(b))	
Distance Errors	Segment points #	Distance Errors	Segment points #
2.435	43 (AB)	1.18	39 (A'B')
1.483	31 (BC)	1.073	31 (B'C')
2.115	45 (CA)	1.267	47 (C'A')

Table 3: *The contour segments errors comparison*

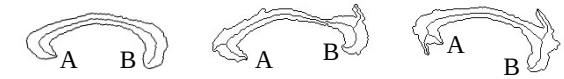
The image in Figure 8(b) is an example of midline sagittal MRI brain images of size 512 by 512. The objective in this experiment is to segment the corpus callosum. The challenge is to extract the contour segments with gaps at the lower left and upper right corners.

The two input corpus callosum shapes (Figure 12(b), (c)) obtained by the DCM [3] are used to match with the corpus callosum model, Figure 12(a). The model is derived from a set of six brain sample images as described in the step (IV) of Section 3. Their skeletons after noise removal are shown in Figure 13. The landmarks are determined from the skeleton E-nodes and the skeleton matching algorithm is applied to construct the correspondence, as shown in Figure 12 (A-A, B-B). Compared with the corresponding model contour segments, the contour segments AB (top segments) in both input contours have large error. The contour segment correction approach is then applied to fine tuning the result. The final contours are shown in Figure 14. The contour segment errors before and after segment corrections are listed in Table 4.



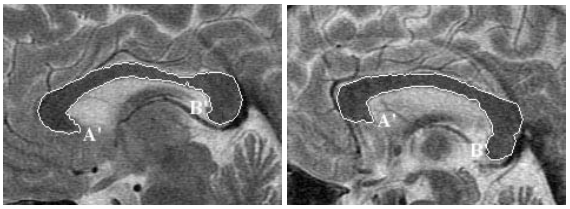
(a) Corpus callosum model (b) Brain 1 result of a DCM (c) Brain 2 result of a DCM

Figure 12 *The corpus callosum model and input shapes obtained by a DCM* [3]



(a) Corpus callosum model skeleton (b) Brain 1 corpus callosum skeleton (c) Brain 2 corpus callosum skeleton

Figure 13: *The corpus callosum skeletons after removing noise curve segments*



(a) Brain 1 final result (b) Brain 2 final result

Figure 14: *The resulting corpus callosum contour after contour segments correction*

Before Correction (**Figure 12**(b))		After Correction (**Figure 14**(a))	
Dist. Errors	Number of Segment points	Dist. Errors	Number of Segment points
20.689	319 (AB) (Top)	15.743	275 (A'B') (Top)
6.292	163 (BA) (Bot.)	6.292	163 (B'A') (Bot.)
Before Correction (**Figure 12**(c))		After Correction (**Figure 14**(b))	
Dist. Errors	Number of Segment points	Dist. Errors	Number of Segment points
10.332	342 (AB) (Top)	1.793	265 (A'B') (Top)
1.006	157 (BA) (Bot.)	1.006	157 (B'A') (Bot.)

Table 4: *The contour segments errors comparison*

5 Conclusion

In this paper, a robust and efficient skeleton-based shape matching method is presented to solve the object recognition and recovery problems in challenging and difficult conditions of image understanding. In the proposed method, a skeleton-based shape matching method uses a combination of both structural and statistical methods that are applied in a sequential manner. The

shape of the input and the models are first represented as an ordered tree and strings of the shape primitives, i.e. skeleton edges. The matching algorithm computes the similarity between the input and the models for all possible string matches. The connectivity relationship among skeleton edges and the geometrical features of the skeleton edges are used for skeleton matching and structural classification. To resolve multiple matches with the same structures, statistical methods are used to select the best match. This sequential approach can largely reduce the matching space and thereby lower the number of computations when compared with other previous works.

Different from many other model-based segmentation methods, the shape recovery by using the skeleton-based shape matching approach presented in this paper is invariant to object translation, rotation and scaling. Thus, the initial condition requirements and the searching space for shape recovery are reduced significantly. Also, by using shape information to guide the segmentation, the object recovery capability provides improvements in robustness and accuracy of the segmentation results as shown in the experiments. The experiments with the animal shape silhouette matching and recognition and the MRI knee and brain shape recovery demonstrate the capability and potential of this new approach.

References

[1] M. Kass, A. Witkin and D. Terzopoulos, "Snake: active contour models," *IJCV*, vol. 1, no. 4, pp. 321-331, 1988.

[2] R. Malladi, J.A. Sethian, and B.C. Vemuri, "Shape modeling with front propagation: a level set approach," *IEEE PAMI*, vol. 17, no. 2, pp.158-175, Feb. 1995.

[3] X. Wang, L. He, C.Y. Han and W.G. Wee, "Deformable contour method: a constrained optimization approach", pp. 183-192, *BMVC 2002*.

[4] L.H. Staib and J.S. Duncan, "Boundary finding with parametrically deformable models," *IEEE PAMI*, vol. 14, no. 11, pp. 1061-1075, Nov. 1992.

[5] A.K. Jain, Y. Zhong, and S. Lakshmanan, "Object matching using deformable templates," *IEEE PAMI*, vol. 18, no. 3, pp. 267-278, Mar. 1996.

[6] Y. Tang, L. He, X. Wang, and W. G. Wee, "A model based contour searching method," *Proc. IEEE Conf. BIBE*, pp. 347-354, Nov. 2000.

[7] S. Svensson and G. Borgefors, "On reversible skeletonization using anchor-points from distance transforms," *Journal of Visual Communication and Image Representation*, vol. 10, pp. 379-397, 1999.

[8] S. Zhu and A.L. Yuille, "FORMS: a flexible object recognition and modeling system," *IJCV*, vol. 20, no. 3, pp. 187-212, 1996.

[9] K. Siddiqi, A. Shokoufandeh, S.J. Dickinson, and S.W. Zucker, "Shock graphs and shape matching," *Sixth ICCV* (Bombay, India), pp. 222-229, January, 1998.

[10] C. Arcelli and G.S.D. Baja, "Finding local maxima in a pseudo-Euclidean distance transform," *CVGIP*, vol. 43, pp. 361-367, 1988.

[11] A. Chakraborty and J.S. Duncan, "Game-theoretic integration for image segmentation," *IEEE PAMI*, vol. 21, no. 1, pp. 12-30, Jan. 1999.

[12] V. Caselles, R. Kimmel and G. Sapiro, "Geodesic active contours," *IJCV*, Vol. 22, No. 1, pp. 61-79, 1997.

[13] B. Ostlad and A. Tonp, "Encoding of a priori information in active contour models," *IEEE PAMI*, vol. 18, no. 9, pp. 863-872, Sep. 1996.

[14] S.D. Fenster and J.R. Kender, "Sectored snakes: evaluating learned-energy segmentations," *IEEE PAMI*, vol. 23, no. 9, pp. 1028-1034, 2001.

[15] Y. Chen, S. Thiruvenkadam, F. Huang, H.D. Tagare, D. Wilson, E.A. Geiser, "On the incorporation of shape priors into geometric active contours," *Proc. of the IEEE Workshop on Variational and Level Set Methods*, pp. 145-152, 2001.

[16] M. Leventon, E. Grimson, and O. Faugeras, "Statistical shape influence in geodesic active contour," *IEEE CVPR*, pp. 316-322, 2000.

[17] M. Rousson and N. Paragios, "Shape priors for level set representations," *the seventh European Conference on Computer Vision*, pp78-92, 2002.

[18] C. Arcelli, "Pattern thinning by contour tracing," *Computer Graphics and Image Processing*, Vol. 17, pp.130-144, 1981.

[19] W. Niblack, D. Capson, and P.B. Gibbons, "Generating connected skeletons for exact and approximate reconstruction," *IEEE CVPR, Champaign*, Illinois, pp.826-828, 1992.

[20] F. Leymaric and M. D. Levine, "Simulating the grass-fire transform using an active contour model," *IEEE PAMI*, Vol. 14, No. 1, pp.56-75, 1992.

[21] J.S. Duncan, R. Owen, P. Anandan, L. Staib, T. McCauley, A. Salazar, and F. Lee, "Shape based tracking of left ventricular wall motion," *Computers in Cardiology 1990*, Chicago, Illinois, pp. 23-26.

[22] I. Cohen, N. Ayache and P. Sulger, "Tracking points on deformable curves," *Proc. Second Euro. Conf. Computer Vision 1992*, May 1992.

[23] S. Sclaroff and A.P. Pentland, "Modal matching for correspondence and recognition," *IEEE PAMI*, vol. 17, no. 6, pp. 545-561, Jun. 1995.

[24] A. Hill, C.J. Taylor, and A.D. Brett, "A framework for automatic landmark identification using a new method of nonrigid correspondence," *IEEE PAMI*, vol. 22, no. 3, pp. 241-251, Mar. 2000.

[25] Y. Wang and T. Pavlidis, "Optimal correspondence of string subsequences," *IEEE PAMI*, vol. 12, pp. 1080-1086, Nov. 1990.

[26] E.S. Ristad and P.N. Yianilos, "Learning string-edit distance," *IEEE PAMI*, vol. 20, pp. 522-532, May 1998.

[27] R.C. Gonzalez and R.E. Woods, *Digital image processing*, Addison-Wesley, 1992.