

SHAPE ANALYSIS USING CURVATURE-BASED DESCRIPTORS AND PROFILE HIDDEN MARKOV MODELS

Rui Huang, Vladimir Pavlovic, and Dimitris N. Metaxas

Department of Computer Science, Rutgers University, Piscataway, NJ 08854, USA
{ruihuang, vladimir, dnm}@cs.rutgers.edu

ABSTRACT

This paper presents a new framework for shape modeling and analysis. A shape instance is described by a curvature-based shape descriptor. A Profile Hidden Markov Model (PHMM) is then built on such descriptors to represent a class of similar shapes. PHMMs are a particular type of Hidden Markov Models (HMMs) with special states and architecture that can tolerate considerable shape contour perturbations, including rigid and non-rigid deformations, oclusions, and missing parts. The sparseness of the PHMM structure also provides efficient inference and learning algorithms for shape modeling and analysis. Our experimental results on corpus callosum images show the effectiveness and robustness of this new framework.

Index Terms— Image shape analysis, hidden Markov models

1. INTRODUCTION

Shape analysis is an important process for many medical imaging applications. For example, it is often a topic of wide interests if there is strong correlation between the shape of some biological structures (e.g., corpus callosum) and age, sex, handedness, or the presence of diseases [1, 2, 3]. Also, a good shape model can be helpful to image segmentation because medical images are often noisy while the desired structures usually have a regular shape [4]. Many shape modeling techniques have been developed with different concerns and respective advantages [5, 6].

Contour-based shape analysis methods mainly exploit shape boundary information, which in many applications is both effective and efficient. Shape contour can be further discretized into a set of landmark points. Statistical shape models based on landmarks (e.g., active shape models [7]) have been successfully used in many domains. However the performance of most of these landmark-based models relies on the accurate landmark correspondence across a population of shape instances, which is not easy to acquire, especially when the data set is large and unlabeled. In addition, when there are oclusions or missing parts on the shape contours, a one-to-one landmark correspondence between a pair of shape instances may not exist.

If we describe a shape contour by a sequence of shape attributes (e.g., curvature, radius, orientation, etc.) computed at the contour points, Hidden Markov Models (HMMs) are an ideal probabilistic sequence modeling method for shape representation [8, 9]. HMMs provide not only robust inference algorithms but also a probabilistic framework for training and building the model [10].

In this paper, we propose a new framework for shape analysis. Our new model is a combination of the curvature descriptor and Profile Hidden Markov Models (PHMMs). PHMMs are strongly linear, left-right HMMs. Thus, they can model a shape more specifically than general ergodic HMMs. This special architecture contains *insert* and *delete* states, in addition to the regular match states, resulting in robustness to considerable shape contour perturbations, including rigid and non-rigid deformations, oclusions and missing contour parts. The adopted framework also leads to a computationally efficient set of algorithms for shape analysis and modeling.

2. SHAPE DESCRIPTION

Adopting the terminology of [5], we use *shape description* to denote the numerical feature vector extracted from a given shape instance using a certain method, and *shape representation* the non-numerical, high-level representation of the shape (e.g., a graphical model) which preserves the important characteristics of the shape. We introduce the shape description part of our model in this section and the shape representation in the next section.

2.1. Feature Extraction

In this work we employ the *curvature* descriptor. Assuming the shape contour has been extracted into an ordered list of points, the shape can then be described by the sequence of the curvatures computed at the contour points. To compute the curvature accurately, one may need to upsample the point set. A Gaussian filter may be applied to the point coordinates before computing the curvatures to reduce the noise impact. Given three consecutive points P_{i-1} , P_i and P_{i+1} on the contour, we define $\vec{a} = \overrightarrow{P_{i-1}P_i}$ and $\vec{b} = \overrightarrow{P_iP_{i+1}}$, then the bending angle at P_i which represents the local curvature is

$$\theta_i = \text{sign}(\vec{a} \times \vec{b}) \arccos\left(\frac{\vec{a} \cdot \vec{b}}{|\vec{a}||\vec{b}|}\right) \quad (1)$$

2.2. Feature Selection

After computing the curvature sequence, one can downsample a dense curvature sequence to reduce the model complexity. A good strategy to downsample the sequence is to keep all the local extremes in the sequence (because the high curvature parts of the shape contour are usually more informative, i.e., they are the salient features), and then choose equally spaced points in between. The spatially equal-distance sampling is also important to the reconstruction of the shape contour from the curvature descriptor.

2.3. Shape Reconstruction

One can reconstruct the shape contour from a curvature descriptor. According to the definition of the bending angle from Eq. (1),

$$\mathbf{x}_{i+1} - \mathbf{x}_i = \vec{b} = R\vec{a} = R(\mathbf{x}_i - \mathbf{x}_{i-1}) \quad (2)$$

where R is the rotation matrix $[\cos \theta_i, -\sin \theta_i; \sin \theta_i, \cos \theta_i]$. Then we have a linear system

$$\begin{cases} R\mathbf{x}_n - (R+I)\mathbf{x}_1 + \mathbf{x}_2 = 0 \\ R\mathbf{x}_{i-1} - (R+I)\mathbf{x}_i + \mathbf{x}_{i+1} = 0, \quad i = 2, \dots, n-1 \\ R\mathbf{x}_{n-1} - (R+I)\mathbf{x}_n + \mathbf{x}_1 = 0 \end{cases} \quad (3)$$

Given a set of boundary conditions, e.g., $\mathbf{x}_1 = [x_1, y_1]^T$, $\mathbf{x}_2 = [x_2, y_2]^T$, one can solve the system using the least squares method. Note that the constants x_1, y_1, x_2, y_2 actually decide the translation, rotation and scaling of the reconstructed shape contour. This shows some attractive properties of the curvature descriptor. It is invariant to object translation. The curvature computed at each contour point is rotationally invariant, so the descriptor is also invariant to object rotation if the starting point is given. Otherwise, the object rotation implies a change in the starting point, which can be handled by the PHMM-based representation method. The curvature descriptor is not strictly invariant to object scaling since a change in the contour length usually leads to a change in the curvature sequence length. One possible solution is to normalize all the shape contours to the same length or, equivalently, sample the contours to a fixed number of points. However, when there are missing parts on the contour, the length of the contour may not be proportional to the actual object scale. Fortunately, PHMMs can address the scaling problem, as well as the occlusions and missing contour parts.

3. SHAPE REPRESENTATION

A curvature descriptor can capture the characteristic of a given shape instance. However, two similar shape of the same class can still have quite different curvature descriptors. To model a class of shapes, one needs a higher-level model to take into consider all the variations within the class.

3.1. Profile hidden Markov models - Elements

PHMMs are a particular type of HMMs well suited for describing general sequence profiles and multiple alignments. PHMMs have been successfully used in bioinformatics and

molecular biology for modeling of DNA and protein sequences [11, 12].

As shown in Fig. 1, a PHMM is a left-right HMM with three different types of states:

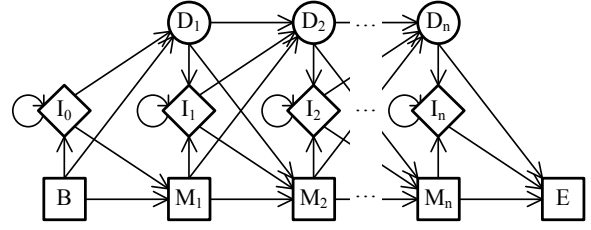


Fig. 1. Profile Hidden Markov Model.

Match states M_1, \dots, M_n are regular states of a left-right HMM with certain emission models $e_{M_i}(O_j)$;

Insert states I_0, \dots, I_n are used to model the portions of the observation sequences that do not correspond to any match states in the model (e.g., stretched parts in the observations). They have emission distributions $e_{I_i}(O_j)$;

Delete states D_1, \dots, D_n are used to handle the portions of the model that do not appear in the observation sequences (e.g., occluded or missing parts in the observations). These situations can also be handled by forward jump transitions between non-neighboring match states. However, to allow for arbitrary deletions the match states need to be completely forward connected. Introducing delete states is an alternative way to get from any match state to any later one with fewer transitions. These states are silent states, which do not emit any observations. Another two silent states *Begin* and *End* are introduced for modeling both ends of a sequence.

3.2. Profile hidden Markov models - Algorithms

Even though PHMMs have different types of states from traditional HMMs, they inherit most of the HMM algorithms [10] with simple adaptations.

Forward-backward algorithm: the forward variable of the PHMM is defined as the probability of the partial observation sequence O_1, \dots, O_j and state x at time j , i.e., $F_x(j) = P(O_1, \dots, O_j, S_j = x | \Theta)$. It can be computed inductively:

$$\begin{aligned} F_{M_i}(j) &= [F_{M_{i-1}}(j-1)a_{M_{i-1}M_i} + F_{I_{i-1}}(j-1)a_{I_{i-1}M_i} \\ &\quad + F_{D_{i-1}}(j-1)a_{D_{i-1}M_i}]e_{M_i}(O_j) \\ F_{I_i}(j) &= [F_{M_i}(j-1)a_{M_iI_i} + F_{I_i}(j-1)a_{I_iI_i} \\ &\quad + F_{D_i}(j-1)a_{D_iI_i}]e_{I_i}(O_j) \\ F_{D_i}(j) &= F_{M_{i-1}}(j)a_{M_{i-1}D_i} + F_{I_{i-1}}(j)a_{I_{i-1}D_i} \\ &\quad + F_{D_{i-1}}(j)a_{D_{i-1}D_i} \end{aligned} \quad (4)$$

where a_{xy} denotes the transition probability from state x to y . The forward variable can be used to compute the likelihood of a sequence given the model parameters:

$$P(O|\Theta) = \sum_x F_x(t) \quad (5)$$

where t is the length of the observation.

The backward variable is defined as the probability of the partial observation sequence O_{j+1}, \dots, O_t given state x at time j , i.e.,

$B_x(j) = P(O_{j+1}, \dots, O_t | S_j = x, \Theta)$. It can be computed in the same manner as the forward variable, but in the opposite direction.

Combining the forward and backward variables, one can compute the probability of being in state x at time j , given the observation sequence O :

$$P(S_j = x | O, \Theta) = \frac{F_x(j)B_x(j)}{\sum_x F_x(j)B_x(j)} \quad (6)$$

which can be used to measure the certainty of a specific matching.

Viterbi algorithm has similar recurrence equations to the forward algorithm, but with the sum operation replaced by maximization. This algorithm can be used to find the single best state sequence given the observation and the model parameters.

Note that the transitions of PHMMs are very sparse, i.e., there are only three transitions (to and from) each state. Hence, the computational complexity of both algorithms is only $O(nt)$ time (in contrast to $O(n^2t)$ of ergodic HMMs) and $O(nt)$ space for a model of n states and an observation sequence of length t . This may lead to significant computational savings when dealing with complex shapes.

3.3. Shape model construction

When aligned sequences are available, a PHMM can be easily learned from the transition and emission counts. However, training a PHMM from multiple initially unaligned sequences is a difficult problem, usually tackled with local optimizers [12]. A viable strategy is to start building a PHMM from only one sequence, align the other sequences to the initial model, and finally refine the model parameters with all the aligned sequences. In this case, the starting sequence should be that of a representative shape, i.e., no dramatic deformations, occlusions or missing parts, and the initial transition and emission probabilities can be chosen based on expert knowledge of the object.

More precisely, given a curvature sequence, $\theta_1, \dots, \theta_n$, n match states in the PHMM are assigned Gaussian emission models

$$e_{M_i}(O_j) = N(O_j; \theta_i, \sigma_i) \quad (7)$$

σ_i is manually selected according to our knowledge of the deformation capability of the specific part of the shape and can be later adapted. The insert states also have Gaussian emission distributions

$$e_{I_i}(O_j) = N(O_j; 0, \sigma) \quad (8)$$

The zero mean suggests that the insert states are simply an extension of the current contour, which is useful for modeling the scaling effects and stretched shape parts. The σ is chosen to control the rigidity of such extensions.

The transitions involving match states usually dominate those between insert and delete states, signifying importance of match states for modeling the shape. Once the initial model is built, the training sequences are aligned to it, and the model parameters are fitted to the data.

4. EXPERIMENTS

We applied the new shape modeling framework to a set of corpus callosum shapes with several different tasks to show its effectiveness and robustness. The data set contains 65 corpus callosum images collected from some previous work [1, 3, 4]. Fig. 2 shows a real corpus callosum image and four extracted contour images from the data set, and the mean shape we trained from the whole data set according to Sec. 3.3 and reconstructed according to Sec. 2.3.



Fig. 2. Shape database.

4.1. Shape matching

Shape matching is a first step of many shape analysis applications. In this section we assume that the curvature sequences have been obtained and, if necessary, downsampled according to Sec. 2. The input to the shape matching algorithm is two curvature sequences O^1 and O^2 , and the output is the point correspondence.

First we compute the PHMM model Θ of sequences O^1 using the method described in Sec. 3.3. We then use this model to find the alignment of the second sequence O^2 to the model as

$$S^* = \arg \max_S P(O^2, S | \Theta) \quad (9)$$

Here S denotes the sequence of states under model Θ and it depicts an optimal correspondence between the two sequences. This formulation requires that the initial correspondence between the two shapes be known, i.e., both sequences start from the same part of the objects, then the problem can be solved by the Viterbi algorithm straightforwardly.

However this is rarely the case due to the rotation, and one often needs to compute the initial correspondence first, i.e.,

$$j^* = \arg \max_j P(O_j^2 O_{j+1}^2, \dots, O_t^2 O_1^2, \dots, O_{j-1}^2 | \Theta) \quad (10)$$

The brutal force approach needs $O(nt^2)$ time to evaluate the likelihood of all the t sequences starting from O_1^2, \dots, O_t^2 respectively, using the Forward algorithm.

One approximate but efficient way of accomplishing this, as well as aligning the two shapes, is to modify the emission and transition models involving the model states I_0 and I_n , which then act like two “don’t-care” states, with broad distributions of contour features. The Viterbi search is then run on the sequence (O^2, O^2) , a twice concatenated original sequence. In this manner we can reduce the complexity of matching two shapes to $O(2nt)$ in most cases.

To test our algorithm, we performed matching experiments on the corpus callosum data set. One of the most difficult example is shown in Fig. 3. The upper image is the shape used to build the model (O^1), where the numbers below the red points are the match state id’s. The middle image is treated as the observations (O^2), where the numbers below the blue points indicate which match state this observation is matched to according to our algorithm. We also show some of the observation id’s (after the alignment) around the matching labels for reference purposes. Note that the original starting point of the observation sequence was not the same as the model starting point. Besides the starting points, the lengths of the two sequences are also different. These situations pose both the rotation and scaling problems, which are successfully solved by our algorithm. The splenium (to the right in the figure) of the observation sequence is larger than that of the model sequence, so there are insertions between some of the points (e.g., both 24 and 25 were matched twice, etc.). We also noticed some other insertions in the middle part because the total length of the observation is larger. On the other hand, the genu of the observation sequence (to the left in the figure) is remarkably smaller than that of the model, where we observed deletions (e.g., 64 jumped to 67, and 71 jumped to 75, etc.). Note that the matching only takes seconds on a normal PC, even with the brutal force alignment algorithm.

In the lower graph of Fig. 3, we show the matching certainty for each observation computed according to Eq. (6). One can interpret it as a measurement of how good a specific match is, individually. For example, our algorithm accurately matches O_{83} to M_{64} . However, the local deformation on shape O around O_{83} causes it to be significantly different from the model shape M around M_{64} . The low matching certainty score $P(S_{83} = M_{64}|O, \Theta)$ points to this discrepancy. Similarly, other points of low matching score correspond to changes in local shape O away from the original shape M . This information can be particularly useful for detection of abnormalities.

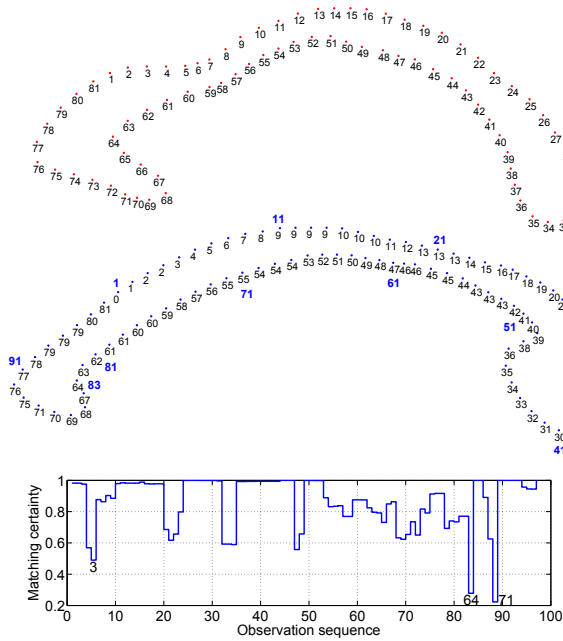


Fig. 3. Shape matching

4.2. Shape Similarity Measure

In this section, we show some results of measuring similarity between shapes, which is important to classification and recognition problems. When the similarity scores between each pair of shapes are calculated, the classification/recognition is simply a problem of choosing classifiers and strategies.

We define the similarity score between two shapes as

$$\begin{aligned}
 P(O^1, O^2) &= \sum_{\Theta} P(O^1|\Theta)P(O^2|\Theta)P(\Theta) \\
 &\approx P(O^1|\Theta_1^*)P(O^2|\Theta_1^*)P(\Theta_1^*) + \\
 &\quad P(O^1|\Theta_2^*)P(O^2|\Theta_2^*)P(\Theta_2^*) \quad (11)
 \end{aligned}$$

where $\Theta_i^* = \arg \max_{\Theta} P(\Theta|O^i) = \arg \max_{\Theta} P(O^i|\Theta)$ for uninformative model priors.

We tested our algorithm on the same data set with the image query task. Fig. 4 shows the results of three different image queries in rows. In each of the three rows, the first image is the query image. We then show the three most similar and three most dissimilar images from the whole data set found by our algorithm (similarities decrease from left to right in each row).

5. DISCUSSIONS

In this paper we proposed a new 2D shape modeling framework based on curvature descriptors and profile hidden Markov models.

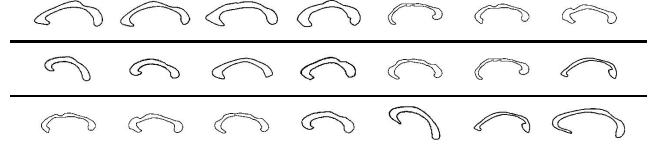


Fig. 4. Shape similarity measure

There are several advantages of this model. The structure and sparseness of PHMMs allows for a set of computationally efficient algorithms to handle the rotations and achieve automatic alignment, hence there is no need to manually label the training data, and different training sequences can even have different length. It is also robust to other rigid and non-rigid deformations, occlusions and missing contour parts. Future work will focus on other applications, such as, object recognition, abnormality detection and recovery, as well as using the proposed model as a shape prior to image segmentation.

6. REFERENCES

- [1] William Byne, Ruth Bleier, and Lanning Houston, "Variations in human corpus callosum do not predict gender: A study using magnetic resonance imaging," *Behavioral Neuroscience*, vol. 102, no. 2, 1988.
- [2] S. F. Witelson, "Hand and sex differences in the isthmus and genu of the human corpus callosum: A postmortem morphological study," *Brain*, vol. 112, 1989.
- [3] Fabrice Robichon, "Abnormal callosal morphology in male adult dyslexics: Relationships to handedness and phonological abilities," *Brain and Language*, vol. 62, pp. 127–146, 1998.
- [4] M. B. Stegmann, R. H. Davies, and C. Ryberg, "Corpus callosum analysis using MDL-based sequential models of shape and appearance," in *SPIE*, February 2004, pp. 612–619.
- [5] Sven Loncaric, "A survey of shape analysis techniques," *Pattern Recognition*, vol. 31, no. 8, 1998.
- [6] Dengsheng Zhang and Guojun Lu, "Review of shape representation and description techniques," *Pattern Recognition*, vol. 37, no. 1, 2004.
- [7] T.F. Cootes, D. Cooper, C.J. Taylor, and J. Graham, "Active shape models - their training and application," *Computer Vision and Image Understanding*, vol. 61, no. 1, pp. 38–59, January 1995.
- [8] Manuele Bicego and Vittorio Murino, "Investigating hidden Markov models' capabilities in 2D shape classification," *IEEE TPAMI*, vol. 26, no. 2, 2004.
- [9] Ninad Thakoor and Jean Gao, "Shape classifier based on generalized probabilistic descent method with hidden Markov descriptor," in *ICCV*, 2005, vol. 1.
- [10] Lawrence R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, February 1989.
- [11] Richard Durbin, Sean R. Eddy, Anders Krogh, and Graeme Mitchison, *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*, Cambridge University Press, 1998.
- [12] Sean R. Eddy, "Profile hidden Markov models," *Bioinformatics*, vol. 14, no. 9, 1998.