A Profile Hidden Markov Model Framework for Modeling and Analysis of Shape

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

In this paper we propose a new framework for modeling 2D shapes. A shape is first described by a sequence of local features (e.g., curvature) of the shape boundary. The resulting description is then used to build a Profile Hidden Markov Model (PHMM) representation of the shape. 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, occlusions and missing contour parts. Different from traditional HMM-based shape models, the sparseness of the PHMM structure allows efficient inference and learning algorithms for shape modeling and analysis. The new framework can be applied to a wide range of problems, from shape matching and classification to shape segmentation. Our experimental results show the effectiveness and robustness of this new approach in the three application domains.

1. Introduction

Shape analysis is an important problem in image processing with applications in image retrieval, object segmentation, classification/recognition, tracking, etc. Many shape modeling techniques have been developed with different concerns and respective advantages [9, 12]. Adopting the terminology of [9], we use *shape description* to denote the numerical feature vector extracted from a given shape using a certain method, and *shape representation* to denote the non-numerical, high-level representation of the shape (e.g., a graphical model) which preserves the important characteristics of the shape.

Contour-based shape analysis methods only exploit shape boundary information, which in many applications is effective and efficient. A shape contour can be simply described 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 [7, 6, 1, 3, 2, 11]. HMMs provide not only robust inference algorithms but also a probabilistic framework for training and building the model.

In this paper, we propose a new shape representation framework based on Profile Hidden Markov Models (PHMMs). PHMMs are strongly linear, left-right HMMs, thus 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, occlusions and missing contour parts. At the same time, the adopted framework leads to a computationally efficient set of algorithms for shape matching, classification and segmentation.

2. Shape Description

The shape description method generates a shape feature vector from a given shape. In this paper, 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 all the contour points. A Gaussian filter may be applied to the contour 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

$$\theta_i = \operatorname{sign}(\vec{a} \times \vec{b}) \arccos(\frac{\vec{a} \cdot \vec{b}}{|\vec{a}||\vec{b}|}) \tag{1}$$

The curvature descriptor has some attractive properties. 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 start point is given. Otherwise, the object rotation implies a change in the start point, which can be handled by the PHMM-based representation method. The curvature descriptor is not invariant to object scaling since a change in the contour length usually leads to a change in the sequence length. One possible solution is to normalize all the shape contours to the same length. However, when there are missing parts on the contour, the length of the contour may not be proportional to the object scale. Fortunately, PHMMs can address the scaling problem, as well as the occlusions and missing contour parts.

3. Shape Representation

3.1. Profile Hidden Markov Models

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 [4, 5].

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

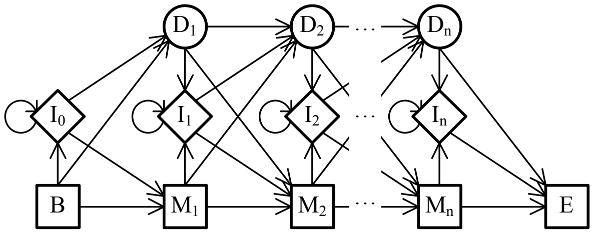

Figure 1. Profile Hidden Markov Model.

- *Match* states $M_1, \cdots, M_n$ are regular states of a left-right HMM with certain emission models $e_{M_i}(O_j)$;
- *Insert* states $I_0, \cdots, I_n$ are used to handle the portions of the observation sequences that do not correspond to any match states in the model. They have emission distributions $e_{I_i}(O_j)$;
- *Delete* states $D_1, \cdots, D_n$ 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 can usually 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 $B(egin)$ and $E(nd)$ are introduced for modeling both ends of a sequence.

Even though PHMMs have different types of states from traditional HMMs, it is easy to extend most HMM algorithms to PHMMs. For the Forward algorithm, the recurrence equations become:

$$\begin{aligned}
F_{M_i}(j) &= [F_{M_{i-1}}(j-1)a_{M_{i-1}M_i} + F_{I_{i-1}}(j-1)a_{I_{i-1}M_i} \\
&\quad +F_{D_{i-1}}(j-1)a_{D_{i-1}M_i}]e_{M_i}(O_j) \\
F_{I_i}(j) &= [F_{M_i}(j-1)a_{M_iI_i} + F_{I_i}(j-1)a_{I_iI_i} \\
&\quad +F_{D_i}(j-1)a_{D_iI_i}]e_{I_i}(O_j) \\
F_{D_i}(j) &= [F_{M_{i-1}}(j)a_{M_{i-1}D_i} + F_{I_{i-1}}(j)a_{I_{i-1}D_i} \\
&\quad +F_{D_{i-1}}(j)a_{D_{i-1}D_i}] \qquad (2)
\end{aligned}$$

where $F_x(j)$ is the probability of the partial observation sequence $O_1, \cdots, O_j$ and state x at time j. a_{xy} denotes the transition probability from state x to y. The Viterbi algorithm has similar recurrence equations, but with the sum operation replaced by maximization.

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 a sequence of length t. This may lead to significant computational savings when dealing with complex shape models.

3.2. Shape model construction

Training a PHMM from initially unaligned sequences is a difficult problem, usually tackled with local optimizers [5]. When aligned sequences are available, a PHMM can be easily learned from the transition and emission counts. More conveniently, we can build a PHMM from only one sequence. In this case, the sequence should be that of a representative shape, i.e., no dramatic deformations, occlusions or missing parts. Transition and emission probabilities need to be carefully chosen based on expert knowledge of the object.

A common strategy used later in this paper is as follows: given a curvature sequence, $\theta_1, \cdots, \theta_n$, n match states in the PHMM are assigned the emission model

$$e_{M_i}(O_j) = N(O_j; \theta_i, \sigma_i). \qquad (3)$$

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 have the emission model

$$e_{I_i}(O_j) = N(O_j; 0, \sigma). \qquad (4)$$

The zero mean suggests that the insert states are simply an extension of current contour, which is useful for modeling scaling effects. The manually picked σ controls the rigidness 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.

The number of match states n need not correspond to every sample in a dense curvature sequence. A reasonable way to downsample the sequence is to keep all the larger values since they correspond to the higher curvature features on the shape contour, and then sample equally spaced points in between.

4. Applications

We applied the new shape modeling framework to several applications to show its effectiveness and robustness.

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. 3.2. 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. 3.2. We then use this model to find the alignment of the second sequence O^2 to the model as $S^* = \arg\max_S P(O^2, S|\Theta)$. 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, in which case the above problem can be simply solved by the Viterbi algorithm. Because this is rarely the case, one needs to compute the initial correspondence, i.e., $j^* = \arg\max_j P(O^2_j O^2_{j+1}, \cdots, O^2_t O^2_1, \cdots, O^2_{j-1}|\Theta)$. The brutal force approach needs $O(nt^2)$ time to evaluate the likelihood of all the t sequences starting from $O^2_1, \cdots, O^2_t$ respectively, using the Forward algorithm.

One 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 shape database created by Sebastian et al. [10], which consists of 9 classes of objects, each having 11 images, bearing all the issues we mentioned. Fig. 2 shows some examples of the database (top row: one shape from each class; bottom row: all the shapes in one class). We first tested the robustness of our algorithm to rotation, scaling and missing shape parts (note that the curvature description is already invariant to translation). The result is shown in Fig. 3, with some representative points highlighted. The left image is the shape used to build the model, and the right one is treated as observations. The red points labeled with numbers are the match states in the model and the blue points labeled with "M" are observations matched to the corresponding model states. Both sequences started from the leftmost contour point ("1" and "M44" respectively). The algorithm successfully detected the corresponding start point on the observation. The index finger shows an example of the effect of the insert states. There are 18 observations, but only 8 of them are matched to the match states 47 to 54, since the model is shorter than the observation sequence. The other 10 observations are matched to the insert states. The effect of the delete states is shown on the third finger, which is missing in the observations. The 5 observations were able to jump from M6 to M10, and M13, and so on, which finally jumped over 16 match states (from M6 to M21). In Fig. 4 we show the matching of two shapes from two different objects, which can be considered non-rigid deformations. Only those states with high curvature are labeled in the figure. In the top row, the cat shape is used to build model and the donkey as the observation, and vice versa in the bottom row. Note the correct correspondence between those points. While the tail and one of the ears of the donkey cannot be seen, they don't affect the correct matching of the rest of the shape.

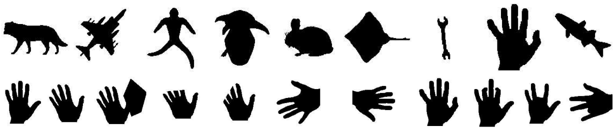

Figure 2. Shape database.

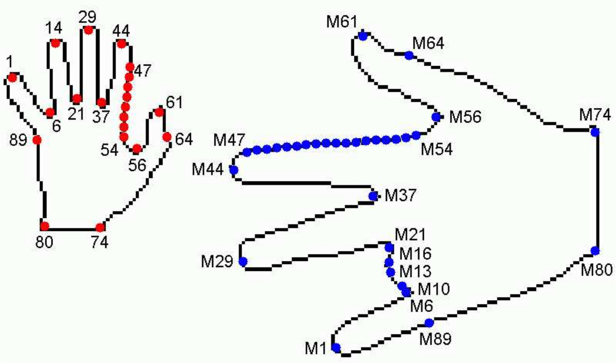

Figure 3. Matching of two hands.

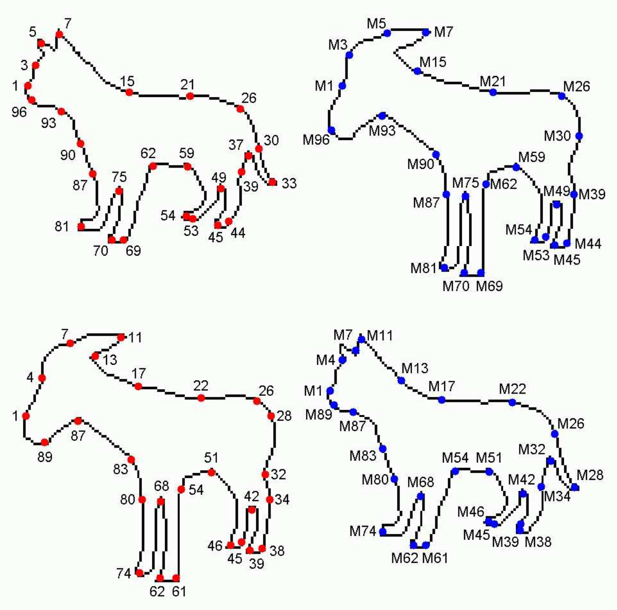

Figure 4. Matching of two animal shapes.

4.2. Shape classification

In this section, we apply our model to shape classification problem. Good similarity measure is critical to classification. We define the similarity score between two shapes as

$$P(O^1, O^2) = \sum_{\Theta} P(O^1|\Theta)P(O^2|\Theta)P(\Theta)$$
$$\approx P(O^1|\Theta_1^*)P(O^2|\Theta_1^*)P(\Theta_1^*) + P(O^1|\Theta_2^*)P(O^2|\Theta_2^*)P(\Theta_2^*)$$

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

When the similarity scores between each pair of shapes are calculated, shape classification is simply a problem of choosing classifiers and strategies. We tested our algorithm on the same database mentioned above with all the 9 classes of 11 images each. We used the nearest neighbor classifier and leave-one-out strategy and obtained 100% classification rate. This is not trivial considering the large in-class variance and a general set of parameters were used for all the images. Instead of general classifiers like the nearest neighbor classifier, more complicated and specific classifiers can be used, e.g., [11] exploited the HMM itself to help design the classifiers.

4.3. Segmentation

Another important application of the shape model is that of serving as the shape prior for image segmentation. For example, the traditional deformable model based segmentation often generate oversmooth boundaries, because the global internal energy term

$$E_{\text{int}}(C) = \sum_i [\alpha_i |P_i - P_{i-1}|^2 / 2h^2 + \beta_i |P_{i-1} - 2P_i + P_{i+1}|^2 / 2h^4] \qquad (5)$$

impose the same smooth effect over the whole contour. To capture the high curvature parts of the boundaries, one has to increase the density of the contour points. Another way to solve this problem is to use a shape prior to impose locally different internal energy terms. In the following experiment, we first use the method presented in [8] to get an initial segmentation. Then the segmented contour is aligned to a shape prior model. We then replace the original internal energy term in [8] with the following:

$$E_{\text{int}}(C) = \sum_i \omega_i |\theta_i - \hat{\theta}_i|^2 \qquad (6)$$

where θ_i is the bending angle at contour point P_i, while $\hat{\theta}_i$ is the bending angle given by the shape prior model. Once the "standard" internal energy terms is replaced with the one computed using the shape prior, we again run the segmentation algorithm of [8]. This way, the high curvature parts of the contour can be more precisely captured with less contour points. The shape model is built on a standard shape (Fig. 5b) and the testing image is generated by shearing the original shape, filling the foreground and background with different grey levels, and adding Gaussian noise (Fig. 5a). Note that the method with shape prior (Fig. 5d) segmented the high curvature contour better than the one without shape prior (Fig. 5c) (results are superimposed on ground truth image for clarity).

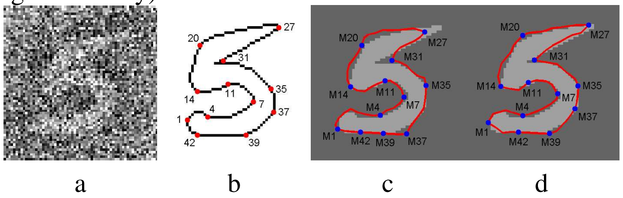

Figure 5. Segmentation with shape prior.

5. Discussions

In this paper we proposed a new 2D shape modeling framework based on curvature descriptors and profile hidden Markov models. The structure and sparseness of PHMMs allows for a set of computationally efficient algorithms to be developed for shape matching, classification and segmentation tasks. We applied this framework to several different applications and showed its robustness to rigid and non-rigid deformations, occlusions and missing contour parts. Future work will focus on automated learning of the shape model, study of its impact on classification problems, as well as a more tightly coupled combination of PHMMs with segmentation algorithms.

References

[1] N. Arica and F. T. Yarman-Vural. A shape descriptor based on circular hidden markov model. In *ICPR*, volume 1, 2000.

[2] M. Bicego and V. Murino. Investigating hidden markov models' capabilities in 2d shape classification.

[3] J. Cai and Z.-Q. Liu. Hidden markov models with spectral features for 2d shape recognition.

[4] R. Durbin, S. R. Eddy, A. Krogh, and G. Mitchison. *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*.

[5] S. R. Eddy. Profile hidden markov models. *Bioinformatics*, 14(9), 1998.

[6] A. L. N. Fred, J. S. Marques, and P. M. Jorge. Hidden markov models vs syntactic modeling in object recognition. In *ICIP*, volume 1, 1997.

[7] Y. He and A. Kundu. 2-d shape classification using hidden markov model.

[8] R. Huang, V. Pavlovic, and D. N. Metaxas. A hybrid framework for image segmentation using probabilistic integration of heterogeneous constraints. In *CVBIA*, pages 82–92, 2005.

[9] S. Loncaric. A survey of shape analysis techniques. *Pattern Recognition*, 31(8), 1998.

[10] T. B. Sebastian, P. N. Klein, and B. B. Kimia. Recognition of shapes by editing shock graphs. In *ICCV*, volume 1, 2001.

[11] N. Thakoor and J. Gao. Shape classifer based on generalized probabilistic descent method with hidden markov descriptor. In *ICCV*, volume 1, 2005.

[12] D. Zhang and G. Lu. Review of shape representation and description techniques. *Pattern Recognition*, 37(1), 2004.