Embedded Profile Hidden Markov Models for Shape Analysis

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

An ideal shape model should be both invariant to global transformations and robust to local distortions. In this paper we present a new shape modeling framework that achieves both efficiently. 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, occlusions, and missing parts. The sparseness of the PHMM structure provides efficient inference and learning algorithms for shape modeling and analysis. To capture the global characteristics of a class of shapes, the PHMM parameters are further embedded into a subspace that models long term spatial dependencies. The new framework can be applied to a wide range of problems, such as shape matching/registration, classification/recognition, etc. Our experimental results demonstrate the effectiveness and robustness of this new model in these different settings.

1. Introduction

Shape analysis is an important process for many computer vision applications, including image classification, recognition, retrieval, registration, segmentation, etc. Many shape modeling techniques have been developed over years with different concerns and respective advantages [16, 24].

Contour-based shape analysis methods mainly exploit the shape boundary information, which in many applications is both effective and efficient. For example, in image retrieval, the object contour is arguably the most convenient query that can be easily input by a user; in image segmentation, many successful boundary-based methods can be improved by a prior model of the boundary shape [17, 9]. Shape contour can be further discretized into a set of landmark points. If we describe a shape contour by a sequence of shape attributes (e.g., curvature, radius, orien-

tation, etc.) computed at these landmarks, Hidden Markov Models (HMMs) are an ideal probabilistic sequence modeling method for shape representation. HMMs provide not only robust inference algorithms but also a probabilistic framework for training and building the model [18]. One of the earliest works on HMM-based shape model [14] used the autoregressive model parameters derived from the radius sequence of the contour points for shape description, and stationary or non-stationary HMMs with 2 to 6 states for shape representation. Promising results were presented. In [12], a shape was described by an 8-directional differential chain code. Both fully connected and left-right HMMs were used for shape representations, and the left-right HMM achieved slightly higher classification rates than the fully connected HMM. A similar chain code description was also used in [2], but with a topologically different HMM for representation. This circular HMM-based shape model is insensitive to scaling and rotation to some extent. [7] used a Fourier spectral feature descriptor, and proposed a specially designed HMM topology and parameter re-estimation procedure to directly deal with these type of features. Recently, Bicego et al. [4] combined a curvature descriptor with an ergodic HMM for shape classification. Curvatures are treated as mixtures of Gaussians, and the Bayesian inference criterion was applied to select optimum number of the HMM states. This work was improved in [22], which used similar representation and description while focuses more on the design of the classification function by combining HMMs with generalized probabilistic descent method. However, most of these HMM-based methods concentrated on shape classification/recognition, while ignored another important aspect of shape modeling: shape matching/registration [23].

In this paper, we propose a new framework for shape modeling. 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 the entire shape profile 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. The adopted framework also leads to a computationally efficient set of algorithms for shape analysis. A further embedding of the model parameters can capture more global characteristics of a class of shapes. Unlike previous HMM-based methods, the new framework is more comprehensive and comparable in performance to alternative state-of-the-art approaches. We will later point out the differences between our model and those works.

2. Shape Description

Adopting the terminology of [16], 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 class. We introduce the shape description part of our model in this section and the shape representation next.

2.1. Feature Extraction

In this work we employ the *curvature* descriptor. Assuming that 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 these points. To compute the curvature accurately, one may need to upsample the point set via interpolation. A Gaussian filter may be applied to the point coordinates before computing the curvatures to reduce the noise effect. Given three consecutive points $\mathbf{x}_{i-1}$, $\mathbf{x}_i$ and $\mathbf{x}_{i+1}$ on the contour, we define $\vec{\mathbf{s}}_{i-1} = \overrightarrow{\mathbf{x}_{i-1}\mathbf{x}_i}$ and $\vec{\mathbf{s}}_i = \overrightarrow{\mathbf{x}_i\mathbf{x}_{i+1}}$, and the bending angle at $\mathbf{x}_i$ which represents the local curvature is

$$\theta_i = \text{sign}(\vec{\mathbf{s}}_{i-1} \times \vec{\mathbf{s}}_i) \arccos(\frac{\vec{\mathbf{s}}_{i-1} \cdot \vec{\mathbf{s}}_i}{|\vec{\mathbf{s}}_{i-1}||\vec{\mathbf{s}}_i|}) \qquad (1)$$

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 2D shape contour from a 1D curvature descriptor. According to the definition of the bending angle (Eq. (1)), $\mathbf{x}_{i+1} - \mathbf{x}_i = \vec{\mathbf{s}}_i = \mathbf{R}_i\vec{\mathbf{s}}_{i-1} = \mathbf{R}_i(\mathbf{x}_i - \mathbf{x}_{i-1})$ where $\mathbf{R}_i = [\cos\theta_i, -\sin\theta_i; \sin\theta_i, \cos\theta_i]$ is the rotation matrix. We then have a linear system (for a closed contour)

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

Given a set of boundary conditions, e.g., $\mathbf{x}_1 = (x_1, y_1)^{\mathbf{T}}$, $\mathbf{x}_2 = (x_2, y_2)^{\mathbf{T}}$, one can solve the system using the least squares method. Note that the choice of the constants x_1, y_1, x_2, y_2 actually determines the translation, rotation and scaling of the reconstructed shape contour. This also shows some attractive properties of the curvature descriptor. First, it is invariant to the object translation. Second, the curvature computed at each contour point is rotationally invariant, so the descriptor is also invariant to the object rotation if the starting point is given. Otherwise, the object rotation causes a circular shift of the curvature sequence, which can be handled by the PHMM-based representation. Finally, the curvature descriptor is not strictly invariant to the object scaling since a change of the contour length usually leads to a change of 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 nonrigid or local deformations or missing parts on the contour, the contour length may not be proportional to the actual object scale. Fortunately, PHMMs can also address the scaling problem, as well as the nonrigid deformations and missing contour parts.

3. Shape Representation

A curvature descriptor can capture the characteristic of a given shape instance. However, two similar shapes 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. Most previous HMM-based methods model the feature sequences with ergodic HMMs. As a consequence, several potential problems may arise. First, most of the states may be used to explain multiple observations along the shape profile. This potentially makes shape matching a complex problem. Second, the training procedure in general ergodic models is typically plagued by sensitivity to the model structure selection, model initialization, and local optima of parameter estimation. We show that these problems can be effectively address in the new PHMM framework.

3.1. Profile hidden Markov models

PHMMs are a particular type of HMMs well suited for describing general sequence profiles and sequence matching. PHMMs have shown outstanding success in computa-
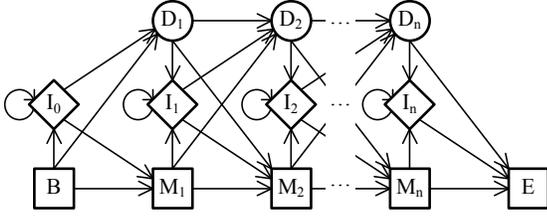
Figure 1. Profile Hidden Markov Model.

tional molecular biology for modeling of DNA and protein sequences [10, 11].

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

Match states $M_1, \cdots, M_n$ are regular states of a left-right HMM with emission models $e_{M_i}(O_j)$. Note that the match states can not be revisited, hence each match state is used to explain no more than one observation segment, a *salient* feature of the modeled shape. This significantly simplifies the matching problem.

Insert states $I_0, \cdots, I_n$ are used to model the portions of the observation sequences that do not correspond to any match states in the model (e.g., the stretched parts on the observed shape contour). 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 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 model transitions from any match state to any subsequent state with fewer transitions in the model. 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.

A profile model may appear more complex than the alternative ergodic model as it typically contains more states (the number of match states is close to the number of salient features in a typical shape instance). However, the transitions of a PHMM are very sparse: there are at most three transitions to and from each state. This significantly reduces the complexity of its algorithms. More importantly, it also addresses a number of problems that the ergodic model may have. For example, with the model structure fixed, the number of states can be easily determined by the number of salient features in the curvature descriptors, and the model parameters are much easier to learn (see Sec. 3.3).

3.2. Inference

Even though PHMMs have different types of states from traditional HMMs, they inherit most of the HMM algorithms [18] with simple adaptations.

The *forward* variable of the PHMM is defined as the probability of the partial observation sequence $O_1 \cdots O_j$ and the state X at time j, i.e., $F_X(j) = P(O_1 \cdots O_j, S_j = X|\Theta)$, where Θ are the model parameters. It can be computed inductively:

$$
\begin{aligned}
F_{M_i}(j) &= [F_{M_{i-1}}(j-1)a_{M_{i-1}M_i} + F_{I_{i-1}}(j-1)a_{I_{i-1}M_i} \\
&\quad + F_{D_{i-1}}(j-1)a_{D_{i-1}M_i}]e_{M_i}(O_j) \\
F_{I_i}(j) &= [F_{M_i}(j-1)a_{M_iI_i} + F_{I_i}(j-1)a_{I_iI_i} \\
&\quad + F_{D_i}(j-1)a_{D_iI_i}]e_{I_i}(O_j) \\
F_{D_i}(j) &= F_{M_{i-1}}(j)a_{M_{i-1}D_i} + F_{I_{i-1}}(j)a_{I_{i-1}D_i} \\
&\quad + F_{D_{i-1}}(j)a_{D_{i-1}D_i}
\end{aligned} \tag{3}
$$

The forward variable can be used to compute the likelihood of a sequence given the model parameters: $P(O_1 \cdots O_t|\Theta) = \sum_X F_X(t)$ where t is the length of the observation sequence.

The *backward* procedure for computing $B_X(j) = P(O_{j+1} \cdots O_t|S_j = X, \Theta)$ can be derived in a similar manner. Given the forward and backward estimates the posterior state distributions can be computed in a usual manner as

$$
P(S_j = X|O, \Theta) = \frac{F_X(j)B_X(j)}{\sum_X F_X(j)B_X(j)} \tag{4}
$$

The optimal state sequence $\arg\max_S P(S|O, \Theta)$ *Viterbi* recursions have the same form as Eq. (3) with the summations replaced by $\max$ operator.

It is important to note that the computational complexity of all three 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. In practice, most of these algorithms take seconds to run on a normal PC, with a magnitude of 10^2 for n and t.

3.3. Learning

As mentioned above, the building of the PHMM is much easier than that of the ergodic HMM. The model structure is fixed, we only need to specify the number of the match states, which can be the number of the salient features in a typical training sequence (the rest of the features will be modeled by the insert states). One can also use more states for better performance, since the complexity of most algorithms only increase linearly. The model parameters are simple to estimate and the variance in their estimates does not significantly impact the model performance [10].

In our implementation, we consider a homogeneous transition model, $a_{X_iM_{i+1}} = \alpha$, $a_{X_iI_i} = \beta$, $a_{X_iD_{i+1}} = \gamma =$

$1 - \alpha - \beta$, where $i = 1, \cdots, n$, $X \in \{M, I, D\}$, and α usually dominates β and γ, signifying the importance of match states for modeling the shape. One typical choice for the observation models $e_{M_i}(O_j)$ and $e_{I_i}(O_j)$ are Gaussian models. Insert states use a single zero-mean Gaussian model as they are usually used to model the smoothly stretched contour parts. This ultimately results in the small set of parameters, $\Theta = (\mu_1, \sigma_1, \cdots, \mu_n, \sigma_n, \sigma_I)$.

The PHMM parameters Θ can be estimated in the traditional EM formalism without the need for labeled state correspondences. However, training a PHMM from multiple initially unaligned shape sequences is a difficult problem, usually tackled with local optimizers [11]. Fortunately, unlike general ergodic models, PHMMs allow a viable strategy starting with a PHMM initialized from a single sequence. This model is subsequently reestimated by matching the remaining training sequences to the initial model, and finally refined with all the aligned sequences.

Estimation of ergodic shape models typically shows significant dependency on initial model estimates. In PHMMs this dependency is reduced due to a simpler model structure. Our initialization procedure relies on a set of general steps.

1. Given a curvature sequence, $\theta_1, \cdots, \theta_n$, the sequence may be downsampled to keep mostly the salient features. However, since we start building the model from a single sequence, we usually keep all to be match states.

2. The n match states in the PHMM are assigned Gaussian emission models $e_{M_i}(O_j) = N(O_j; \theta_i, \sigma_i)$. σ_i can be initialized uniformly to some constant, or more specifically according to our knowledge about the deformation capability of different parts of the contour. These parameters can easily control the flexibility of the model.

3. The insert states also have Gaussian emission distributions $e_{I_i}(O_j) = N(O_j; 0, \sigma_I)$. 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. σ_I is chosen to control the rigidity of such extensions, and usually smaller than σ_i.

Once the initial model is built, the remaining training sequences are aligned to it, and the model parameters are fitted to the data. In this way, we avoid labeling the training data.

3.4. Embedded Profile hidden Markov models

The regular PHMM is both efficient and effective for many shape analysis tasks, as we show in Sec. 4. However, inconsistent shape matching may occur because of the *lack* of global shape constraints in PHMMs. Such constraints impose global dependencies between spatially distant parameters (e.g., match states) in an otherwise local model.

One way to impose global constraints is to embed the PHMM parameters Θ into a lower dimensional subspace that spans the range of Θ. While there are numerous ways to achieve this embedding, both linear and nonlinear, we here adopt the embedding via Probabilistic Principal Component Analysis (PPCA). PPCA formalism matches our probabilistic formulation and is typically a first stage of many nonlinear embedding techniques, such as the nonlinear or kernel PCA. Unlike the nonlinear methods, PPCA is efficient and often produces satisfactory results in practice.

Let $\Theta = (\mu_1, \cdots, \mu_n)^{\mathbf{T}}$ be the vector of the model parameters whose embedding we seek, then $P(\Theta|h) = N(\Theta; Wh, \delta\mathbf{I})$ where W[1] is the principal component matrix, h is the principal factor, and $\delta\mathbf{I}$ is the covariance of the noise model. It is common to assume a prior distribution over the latent variable h, e.g., a Gaussian prior $P(h) = N(h; 0, \lambda\mathbf{I})$ where $\lambda\mathbf{I}$ is also learned from the training data.

The latent factors h correlate the otherwise, in the regular PHMM, uncorrelated match states. The complete model can now be expressed as

$$P(O, S, h|W) = \int_\Theta P(O|S, \Theta)P(S|\Theta)P(\Theta|W, h)P(h)d\Theta \tag{5}$$

This Embedded PHMM (EPHMM) is now parameterized by a new set of global shape parameters W, λ. Latent factor h represents the global deformation of the shape and needs to be estimated during the inference stage, in addition to hidden states S. We accomplish this using the following coordinate ascent fixed point equations that amounts to an approximate MAP inference:

$$
\begin{aligned}
S^* &= \arg\max_S P(O, S|\Theta^*, h^*, W) \\
(h^*, \Theta^*) &= \arg\max_{h, \Theta} P(O, S^*|\Theta, h, W) \tag{6}
\end{aligned}
$$

The first task is the PHMM Viterbi algorithm. The second is the PPCA inference (solved in the manner similar to active shape models). Given the above inference procedure, estimation of the embedding matrix W is carried out in the standard EM framework. λ is typically treated as a hyperparameter.

4. Applications and Results

In this section, we show how our new framework can be applied to shape classification and matching. We demonstrated its effectiveness and robustness on several data sets.

4.1. Starting point detection

As we mentioned before, the curvature descriptor is not strictly rotation invariant when the starting point is not given. In most cases, we start extracting features from the

---

[1] W includes the offset term $\bar{h}$.

leftmost point on the boundary and following the boundary in a clockwise manner, so the object rotation leads to a circular shift of the curvature sequence. Given model Θ and observation O, the starting point can be computed as follows, $j^* = \arg\max_j P(O_j O_{j+1} \cdots O_t O_1 O_2 \cdots O_{j-1}|\Theta)$. The brute force approach needs $O(nt^2)$ time to evaluate the likelihood of all the t sequences starting from $O_1, \cdots, O_t$ respectively using the Forward algorithm.

One approximate but efficient way of accomplishing the same task is to modify the model parameters involving the states I_0 and I_n with broad distributions of contour features (i.e., to allow higher variances), which then act like two "don't-care" states. The Viterbi algorithm is then run on the sequence (O, O), a twice concatenated original sequence. Ideally, I_0 and I_n will absorb most of the repeated observations from the two ends, and the sub-sequence started from the actual starting point will be matched to the original model. In this manner we may reduce the complexity of starting point detection to $O(nt)$ in many cases.

We subsequently assume that the curvature sequences have been obtained and if necessary, downsampled according to Sec. 2 and aligned to the same starting point using one of the above methods.

4.2. Shape similarity measure

Similarity measure is fundamental to many pattern recognition problems. In this section, we define the shape similarity measure based on our framework, and apply it to several different problems. The similarity score between two shapes is defined as

$$
\begin{aligned}
P(O^1, O^2) &= \sum_\Theta P(O^1|\Theta)P(O^2|\Theta)P(\Theta) \\
&\approx P(O^1|\Theta_1^*)P(O^2|\Theta_1^*)P(\Theta_1^*) + \\
&\quad P(O^1|\Theta_2^*)P(O^2|\Theta_2^*)P(\Theta_2^*) \quad (7)
\end{aligned}
$$

where $\Theta_i^* = \arg\max_\Theta P(\Theta|O^i) = \arg\max_\Theta P(O^i|\Theta)$ for uniformative model priors. This is solved by the Forward algorithm.

We first test this measure on a corpus callosum data set with the image query task. The data set contains 65 corpus callosum images [5, 19, 21]. Fig. 2 shows a real corpus callosum image and five extracted contour images from the data set. Fig. 3 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).

Figure 2. Corpus callosum shape data set.

Figure 3. Shape similarity measure.

We also performed classification on the shape data set created by Sebastian et al. [20], which consists of 9 classes of objects, each having 11 images, bearing all the variances we mentioned. Fig. 4 shows some examples from the data set (top row: one shape from each class; bottom row: all the shapes in the class "hand"). In the most straightforward fashion, we measured the distance between each pair of shapes, and then used the nearest neighbor classifier and leave-one-out strategy to achieve a 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 computing distance between every pair of shapes, one can build a PHMM for each class, and then evaluate the likelihood of the test image given models of different classes. This usually requires more training images. More decent methods can be used to build the PHMMs, e.g., [22], to further improve the classification rate.

Figure 4. Sebastian shape data set.

4.3. Shape matching

Shape similarity measure is mainly related to a basic problem for HMMs, i.e., evaluating the likelihood of the observation sequence given the model. On the other hand, shape matching corresponds to another basic problem, finding the optimal explanation of the observation sequence. 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 build the PHMM model Θ of sequence O^1 (also known as the target) using the method described in Sec. 3.3. We then compute the optimal state path of the second sequence O^2 (the source) given this 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 is solved by the Viterbi algorithm straightforwardly.

To test our algorithm, we performed matching experiments on the above two data sets. Initial point detection is handled as outlined in Sec. 4.1. One of the typical
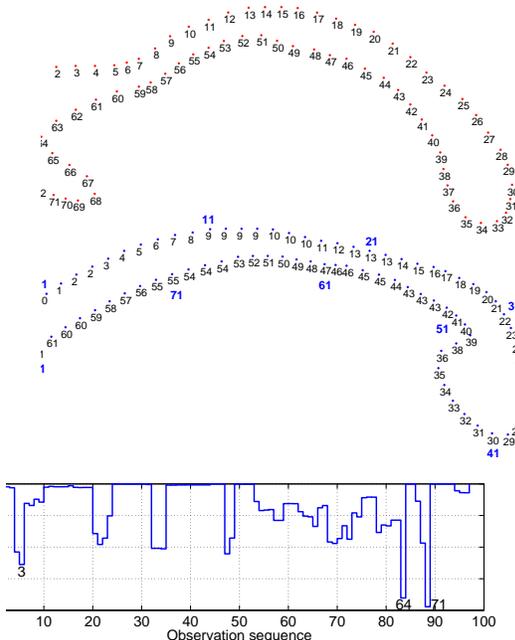
Figure 5. Shape matching using regular PHMMs.

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 in medical imaging applications.
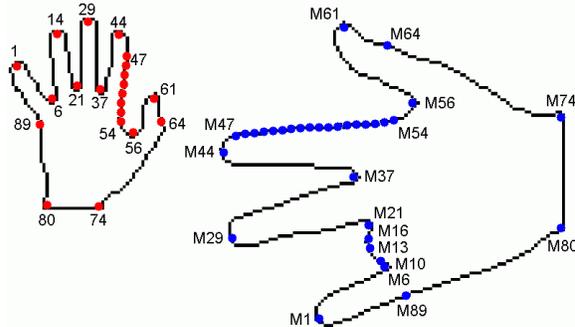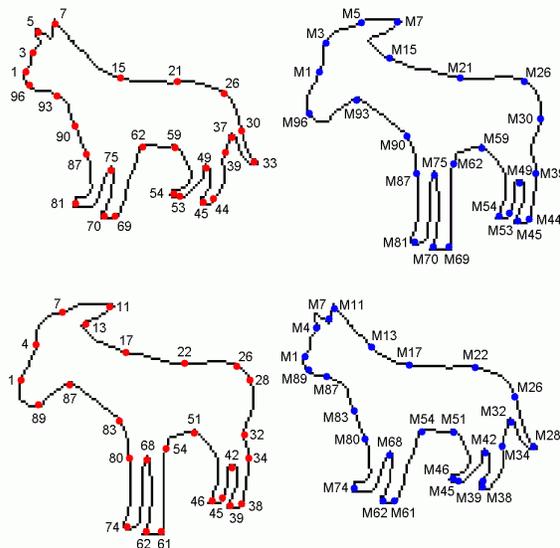

Figure 6. Matching of two hand shapes.

Figure 7. Matching of two animal shapes.

and difficult examples in the callosum data set is shown in Fig. 5. The upper image is the target shape used to build the model, where the numbers below the (red) points are the match state id's. The middle image is the source shape treated as the observation, where the numbers below the (blue) points indicate which match state this observation is matched to according to our algorithm (the repeated numbers are matched to corresponding insert states). We also show some of the observation id's (after the starting point detection) around the matching labels for reference purposes. Note that the original starting point of the observation sequence is not the same as the model starting point, and 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. As for the non-rigid deformations, 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., 24 and 25 are repeated, etc.). 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, 71 jumped to 75, etc.). In the lower graph of Fig. 5, we show the matching certainty for each observation computed according to Eq. (4). 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

Robustness. Fig. 6 shows examples in the Sebastian shape data set, which allows us to further examine the robustness of our algorithm to large rotation, scaling and missing parts. The left image is the shape used to build the model, and the right one is treated as observations. Some representative points are highlighted and labeled similar to previous figures. Both sequences start from the leftmost contour points ("1" and "M44" respectively). The algorithm successfully detected the corresponding start point on the observation. Note the insertions on the index finger (18 observations vs. 8 match states) and the deletions on the third finger, which is missing in the observations (5 observations vs. 16 match states). Fig. 7 is another example from this

data set where we match two shapes from two different objects, which can be considered non-rigid deformations and missing parts. In the top row, the cat shape is used to build the model and the donkey as the observation, and vice versa in the bottom row. Note the correct correspondence between the labeled 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 parts.

The above experiments show the power of the curvature+PHMM-based shape model. However we did observe some mismatching in the "aircraft" class of the Sebastian data set. The reason is that the shape contour of an aircraft often has multiple high curvature parts with similar feature values and distributed evenly along the contour. The regular PHMM may fail due to the strongly ambiguous features. Instead, we trained an embedded PHMM as described in Sec. 3.4 and used the inference procedure outlined in Eq. (6).

Role of embedding. Fig. 8 shows examples of the aircraft shapes collected from NASA Dryden online gallery. Note that auto-alignment on this data set is not easy. To avoid manually labeling the data, we used only 25 shapes in the data set that are easy to be aligned automatically. Fig. 9 shows the mean shape we trained from the 25 shapes according to Sec. 3.3 and Sec. 3.4 and reconstructed according to Sec. 2.3. The upper-left image of Fig. 10 shows the initial matching using the regular PHMM, where the regular PHMM model can hardly handle the strong ambiguities. The upper-right image is the reconstructed shape contour from the matched observations. It is far away from a reasonable shape of an aircraft, indicating a low probability of the latent variable h. After we regularize it using the PPCA, we get a new sets of parameters for the PHMM match states Θ^* (the reconstructed shape contour is shown in lower-right figure, which is reasonably like an aircraft), and rematch the observation sequence to the new adapted PHMM. This procedure usually converges in several iterations in our experiments. The lower-left image of Fig. 10 is the final matching result. Fig. 11 shows another example where the constrained PHMM outperforms the regular PHMM. It is worth noting that the seemingly simpler shapes actually causes more ambiguities and are often harder to match than others.
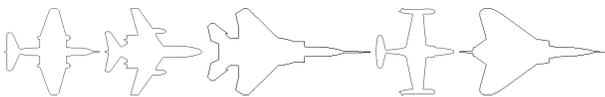

Figure 8. NASA Dryden aircraft shape data set.

5. Discussions and Future Work

In this paper we proposed a new shape modeling framework based on curvature descriptors and profile hidden Markov models. The structure and sparseness of PHMMs
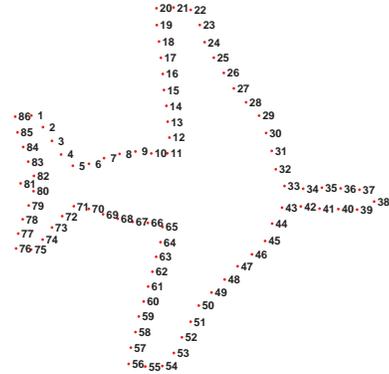

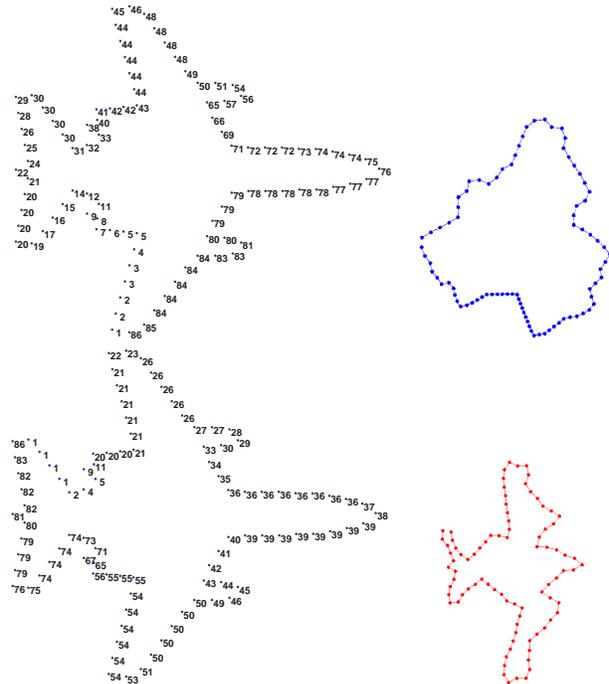Figure 9. Reconstructed mean shape.

Figure 10. Shape matching using embedded PHMMs 1.

allows for a set of computationally efficient algorithms to be developed for shape classification and matching tasks. We applied this framework to these problems and showed its robustness to rigid and non-rigid deformations, occlusions and missing contour parts.

From the model perspective, our work is most similar to the HMM-based methods [14, 12, 2, 7, 4, 22, 15]. Our model is easier to train and incorporated the global shape constraints that is critical for the shape matching, thus can be applied to more different and difficult tasks. With regard to more comprehensive shape models, [1] also provides a statistical framework, but no results on handling occlusions is reported. The model embedding part of our model is similar to active shape models [8], however the 1D nature and flexibility of our model makes it more efficient to
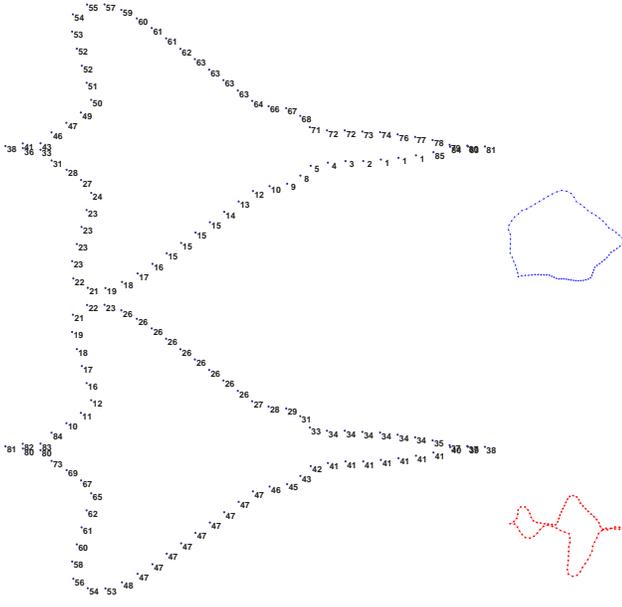
Figure 11. Shape matching using embedded PHMMs 2.

handle large rotation and occlusions. From the application perspective, our work is more similar to [13]. Our work is different in that we can build fully probabilistic models over classes of shapes. Shape context [3] also proves to be a good descriptor for many shape analysis problems but it is not a high level representation. While such contourless shape models using 2D descriptors can handle topologically more difficult shapes, our contour-based shape modeling approach can be both computationally efficient and robust to significant deformations, and it is easier to be incorporated into contour-based segmentation methods. Disjoint region connecting methods such as [6] also allow our model to be applied to a wider range of 2D shapes.

Future work will focus on extending the current framework to 3D shape modeling. We are currently exploring different higher dimensional shape features/descriptors and the representations based on random fields. Another important application of our model is that of serving as the shape prior for image segmentation [17, 9]. In this case, our model's ability to handle large rotation and occlusions efficiently is especially helpful. We are working on establishing a more tightly coupled combination of our shape model with segmentation algorithms based on deformable models.

Acknowledgement

References

[1] W. M. A. Srivastava, S. H. Joshi and X. Liu. Statistical shape analysis: Clustering, learning and testing. *IEEE Trans.*

PAMI, 27(4), 2005.

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

[3] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Trans. PAMI*, 24(4), 2002.

[4] M. Bicego and V. Murino. Investigating hidden Markov models' capabilities in 2D shape classification. *IEEE Trans. PAMI*, 26(2), 2004.

[5] W. Byne, R. Bleier, and L. Houston. Variations in human corpus callosum do not predict gender: A study using magnetic resonance imaging. *Behavioral Neuroseience*, 102(2), 1988.

[6] J. Cai and Z.-Q. Liu. Integration of structural and statistical information for unconstrained handwritten numeral recognition. *IEEE Trans. PAMI*, 21(3), 1999.

[7] J. Cai and Z.-Q. Liu. Hidden Markov models with spectral features for 2D shape recognition. *IEEE Trans. PAMI*, 23(12), 2001.

[8] T. Cootes, D. Cooper, C. Taylor, and J. Graham. Active shape models - their training and application. *Computer Vision and Image Understanding*, 61(1), 1995.

[9] S. Dambreville, Y. Rathi, and A. Tannen. Shape-based approach to robust image segmentation using kernel pca. In *CVPR*, 2006.

[10] R. Durbin, S. R. Eddy, A. Krogh, and G. Mitchison. *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press, 1998.

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

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

[13] Y. Gdalyahu and D. Weinshall. Flexible syntactic matching of curves and its application to automatic hierarchical classification of silhouettes. *IEEE Trans. PAMI*, 21(12), 1999.

[14] Y. He and A. Kundu. 2-D shape classification using hidden Markov model. *IEEE Trans. PAMI*, 13(11), 1991.

[15] R. Huang, V. Pavlovic, and D. N. Metaxas. A profile hidden Markov model framework for modeling and analysis of shape. In *ICIP*, 2006.

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

[17] H. E. A. E. Munim and A. A. Farag. A shape-based segmentation approach: An improved technique using level sets. In *ICCV*, 2005.

[18] L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2), 1989.

[19] F. Robichon. Abnormal callosal morphology in male adult dyslexics: Relationships to handedness and phonological abilities. *Brain and Language*, 62:127–146, 1998.

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

[21] M. B. Stegmann, R. H. Davies, and C. Ryberg. Corpus callosum analysis using MDL-based sequential models of shape and appearance. In *SPIE*, 2004.

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

[23] R. C. Veltkamp and M. Hagedoorn. State of the art in shape matching. *Principles of visual information retrieval*, 2001.

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