Conformation-Based Hidden Markov Models: Application to Human Face Identification

Djamel Bouchaffra, *Senior Member, IEEE*

***Abstract*—Hidden Markov models (HMMs) and their variants are capable to classify complex and structured objects. However, one of their major restrictions is their inability to cope with *shape or conformation* intrinsically: HMM-based techniques have difficulty predicting the n-dimensional shape formed by the symbols of the visible observation (VO) sequence. In order to fulfill this crucial need, we propose a novel paradigm that we named conformation-based hidden Markov models (COHMMs). This new formalism classifies VO sequences by embedding the nodes of an HMM state transition graph in a Euclidean vector space. This is accomplished by modeling the noise contained in the shape composed by the VO sequence. We cover the one-level as well as the multilevel COHMMs. Five problems are assigned to a multilevel COHMM: 1) sequence probability evaluation, 2) statistical decoding, 3) structural decoding, 4) shape decoding, and 5) learning. We have applied the COHMMs formalism to human face identification tested on different benchmarked face databases. The results show that the multilevel COHMMs outperform the embedded HMMs as well as some standard HMM-based models.**

***Index Terms*—Dual-tree wavelet transform, embedded hidden Markov models, face identification, hidden Markov models (HMMs), object contour representation, shape decoding, structural decoding.**

I. INTRODUCTION

THE traditional HMM model represents a powerful machine learning formalism for observation sequence prediction. This Bayesian framework has been applied in several research areas with success. However, its broad spectrum of implementation still remains scarce. The main reason behind this limitation is explained by the fact that HMMs are unable to: 1) account for long-range dependencies which unfold structural[1] information, and 2) intrinsically unravel the shape[2] formed by the symbols of the visible observation (VO) sequence. Because the traditional HMMs modeling is based on the hidden state conditional independence assumption of the visible observations, therefore, HMMs make no use of structure. Furthermore, the fact that the HMM state transition graph is not embedded in a Euclidean vector space, therefore HMMs make no use of shape information. This lack of structure inherent to standard HMMs represents a major challenge to the machine learning and pattern recognition community. It has drastically limited the shape recognition task of complex objects.

To face this challenge, a few approaches have been proposed. The hierarchical hidden Markov models (HHMMs) introduced in [1] and [2] are capable to model complex multiscale structures which appear in many natural sequences. However, the HHMMs algorithm have difficulty modeling shape information. The structural hidden Markov models (SHMMs) introduced in [3] and [4] propound a methodology that identifies the different constituents of a VO sequence. These constituents known as "local structures" are computed via an equivalence relation defined in the space of the VO subsequences. However, the problem of shape determination of these local structures is not examined. Other graphical models such as embedded HMMs (EHMMs) [5], coupled HMMs (CHMMs) [6], factorial HMMs (FHMMs) [7], event-coupled HMMs (ECHMMs) [8] and input–output HMMs (IOHMMs) [9] that depict different architectures have also been put forward in the machine learning community. Their missions consists of boosting the capabilities of the standard HMMs. Nevertheless, this generalization of the HMMs to capture local structures did not address the shape modeling problem of the VO sequence. As far as we are aware, the embedding of HMMs (or dynamic Bayesian networks in general) in a Euclidean vector space which allows shape features to be exploited has not been the object of investigation by the machine learning and pattern recognition community.

We propose a novel machine learning paradigm that embeds the nodes of an HMM state transition graph in a Euclidean vector space [10]. This HMM extension entitled conformation-based hidden Markov models (COHMMs) extends the traditional concept of HMMs by: 1) segmenting the entire VO sequence to reveal its constituents, 2) clustering these segments, and 3) applying wavelet transforms to capture their shapes. Classification is performed by filtering the Gaussian noise embedded in a VO sequence shape representation captured by its external contour.

There exist several applications where COHMMs can be applied. A first application would consist of predicting a protein 3-D fold known as tertiary structure given its primary structure. This problem is one of the most challenging ones in the area of proteomics. A second application would be biometrics for face, or fingerprint identification.

Section II clarifies the notion of a VO sequence. Section III describes briefly the traditional and the embedded HMMs.

Manuscript received September 23, 2008; revised December 15, 2009; accepted December 28, 2009.

The author is with the Mathematics and Computer Science Department, Grambling State University, Grambling, LA 71245 USA (e-mail: dbouchaffra@ieee.org).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNN.2009.2039875

[1] From "structure" which is the way in which parts are arranged, or put together to form a whole.

[2] A shape is any subset $S \subset \Re^n$ with a boundary ∂S, usually restricted to those subsets homeomorphic to a ball.

1045-9227/$26.00 © 2010 IEEE

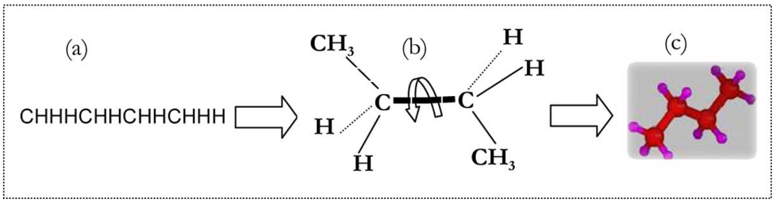

Fig. 1. Butane molecule. (a) Its VO sequence O = CHHHCHHCHHCHHH. Each symbol is either a carbon or a hydrogen atom. (b) VO sequence instantiation into UNIFs. (c) UNIF shapes captured by their external contours.

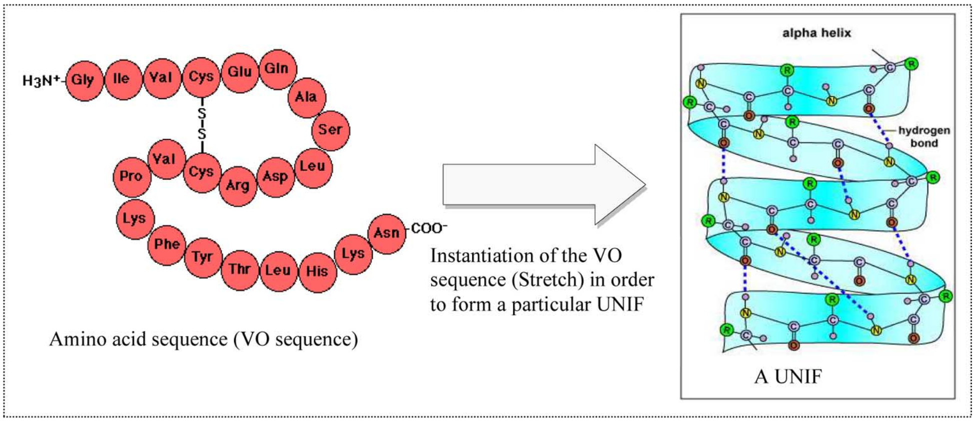
H3N+
Gly
Ile
Val
Cys
Glu
Gln
Ala
Ser
S
S
Leu
Asp
Arg
Cys
Val
Pro
Lys
Phe
Tyr
Thr
Leu
His
Lys
Asn
COO-
Amino acid sequence (VO sequence)
Instantiation of the VO sequence (Stretch) in order to form a particular UNIF
alpha helix
hydrogen bond
A UNIF

Fig. 2. VO sequence of amino acids (protein primary structure) is mapped to its UNIF (Alpha Helix: protein secondary structure) via a particular instantiation (through stretching). “Hydrophobicity” is thought to be one of the primary forces driving the folding of secondary structures.

Section IV explains the shape mapping between the VO sequence and the shape it depicts. Section V introduces the concept of COHMMs. We cover the one-level COHMMs, the optimal segmentation problem, and the multilevel COHMMs. The application of the COHMMs to human face identification is presented in Section VI and the conclusion is laid out in Section VII.

II. The Visible Observation Sequence

We define a VO sequence as a flow of symbols ordered by time. Likewise, we define a unit of information (UNIF) as a shape formed by a group of symbols. If the entire VO sequence has a shape, therefore its shape represents a UNIF that we call *object*. However, if the VO sequence is made of subsequences that possess shapes, therefore each shape is by itself a UNIF. In this case, the sequence of UNIFs obtained represents an entire object. A UNIF can unfold only through a *meaningful* organization of the VO sequence. Not all VO sequences constitute a UNIF but only those which disclose structural components of the observed object. For example, the butane gas formula “CHHHCHHCHHCHHH” represents a VO sequence. However, the same formula can be written in a more informative way as a sequence of UNIFs: “$CH_3 - CH_2 - CH_2 - CH_3$.” In this representation, the shapes of the structural parts of the butane which are “CH_3” and “CH_2” are UNIFs. In other words, the UNIFs are certain rearrangements of their constituent symbols that produce shapes (see Fig. 1).

A second application would consist of predicting the protein 3-D fold (or conformation) given its primary structure. The sequence of amino acids “GLY, ILE, VAL, CYS, GLU, GLN, ALA\ldots ” known as the primary structure is the VO sequence $O = o_1, o_2, \ldots, o_T$ of the protein 3-D fold (see Fig. 2). The UNIFs are the protein secondary structures: *Alpha-Helix*, *Beta-Sheet*, *Beta-Turn*, and others. They represent 3-D forms of local segments of proteins. However, they do not describe specific atomic positions in 3-D space, which are considered tertiary structures. One possible instantiation of the amino acid sequence produces a protein secondary structure sequence.

III. Standard and Embedded Hidden Markov Models

To better understand the contribution of the COHMMs, we provide a summarized description of the traditional HMMs [11] and the embedded HMMs [5].

A. Traditional HMMs

Definition 3.1: A HMM is a doubly embedded stochastic process with an underlying stochastic process that is not observable (it is hidden), but can only be observed through another set of stochastic processes that produce the sequence of observations.

B. Elements of an HMM

We now introduce the elements of an HMM, and explain how the model generates observation sequences. An HMM is characterized by the following parameters:

- $\mathbf{N}$, the number of hidden states q_i in the model;
- $\mathbf{R}$, the number of distinct observation o_i per hidden state, i.e., the size of the discrete alphabet;
- the initial state distribution $\pi = \{\pi_i\}$, where $\pi_i = P(q_0 = e_i)$, $1 \leq i \leq N$, and $\sum_i \pi_i = 1$;

- the state transition probability matrix $\mathcal{A} = \{a_{ij}\}$, where $a_{ij} = P(q_{t+1} = e_j \mid q_t = e_i)$, $1 \leq i, j \leq N$ and $\sum_j a_{ij} = 1$;
- the emission (or state) probability matrix, $\mathcal{B} = \{b_j(k)\}$, where $b_j(k) = P(o_k \text{ at time } t \mid q_t = e_j)$, $1 \leq k \leq R$ and $1 \leq j \leq N$, and $\sum_k b_j(k) = 1$. In a continuous density HMM, the states are characterized by continuous observation density functions. Generally, the model probability density function is taken as a sum of c mixtures of the form $b_j(k) = \sum_{i=1}^{i=c} c_{ji} N(o_k, \mu_{ji}, \Sigma_{ji})$, where N is the Gaussian distribution. An HMM is usually represented as $\lambda = [\pi, \mathcal{A}, \mathcal{B}]$.

C. The Three Basic Problems of an HMM

There are three basic problems that are assigned to an HMM.

1) Evaluation: Given the observation sequence $O = o_1, o_2 \ldots, o_T$ and a model $\lambda = [\pi, \mathcal{A}, \mathcal{B}]$, determine the probability that this observation sequence was generated by the model λ.
2) Decoding: Suppose we have an HMM λ and a VO sequence O, then determine the most likely sequence of hidden states $q_1, q_2, \ldots, q_T$ that generated O.
3) Learning: Suppose we are given a coarse structure of a model (the number of hidden states and the number of observations symbols) but not the probabilities a_{ij} nor b_{jk}. Given a limited set of training observation sequences, determine these parameters. In other words, the goal is to search for the model λ that is most likely to have produced these observation sequences.

We first focus on the evaluation problem. Let $O = (o_1, o_2, \ldots, o_T)$ be the VO sequence of length T and $q = (q_1, q_2, \ldots, q_T)$ the state sequence with q_0 as an initial state. The evaluation problem is expressed as follows. Given a model λ, and the observation sequence O, evaluate the match between λ and the observation sequence O by computing

$$P(O \mid \lambda) = \sum_q P(O, q \mid \lambda) = \sum_q P(O \mid q, \lambda) \times P(q \mid \lambda). \tag{1}$$

Using the state conditional independence assumption of the visible observation sequence O, that is, $P(o_1, o_2, \ldots, o_T \mid q) = \prod_{t=1}^{T} P(o_t \mid q_t)$, and assuming a first-order Markov chain, we derive

$$P(O \mid \lambda) \approx \sum_q \prod_{t=1}^{T} P(o_t \mid q_t) \times P(q_t \mid q_{t-1}). \tag{2}$$

The evaluation problem is based on the state conditional independence assumption of the VO sequence symbols. However, there are several scenarios where a long-range dependency between visible observations is needed. Besides, this dependency would be much more informative if it were not only temporal but shape as well. In other words, we would be more advanced if we knew what shape these related observations are forming. Unfortunately, the notion of UNIF that calls for shape features is intrinsically absent in the HMM-based formalisms. It has been proven that standard HMMs perform well in recognizing amino acids and consequent construction of proteins from the first level structure of DNA sequences [12], however, they are inadequate for predicting a tertiary structure of a protein. The reason for this inadequacy comes from the fact that the same order of amino acid sequences might have different protein folding modes in natural circumstances [13]. In other words, it is only the shape information that enables the discrimination between these different folding modes.

D. Embedded HMMs

A traditional HMM is considered as 1-D. However, if each hidden state is viewed as a traditional HMM itself then we obtain a 2-D structure which is known in the literature as an "embedded HMM" (EHMM). Therefore, we can state the following.

Definition 3.2: An embedded hidden Markov model is a traditional HMM in which each hidden state is by itself an HMM. These hidden states are called "super states," whereas each state inside the super state is called an "embedded state."

The elements of an embedded HMM are as follows:

- the number of super states N and the set of super states S_N;
- the initial super state distribution at time 0, Π_0;
- the super state transition probability matrix A_0;
- the parameters of the embedded HMMs, which are:
  1) the number of embedded states in the kth super state, and the set of embedded states;
  2) the initial state distribution $\Pi_1^{(k)}$ within the super state k at time 0;
  3) the state transition probability matrix $A_1^{(k)}$ within the super state k;
- the emission probability matrix $B^{(k)} = \{b_i^{(k)}\}$ from each state (i, k) in the 2-D grid of the embedded HMM.

Finally, an embedded HMM is defined as the triplet $\lambda = [\Pi_0, A_0, \Lambda]$, where $\Lambda = \{\Lambda^{(k)}\} = [\Pi_1^{(k)}, A_1^{(k)}, B^{(k)}]$.

IV. The Shape Mapping: Projection Onto a Euclidean Space

This section brings forward the *shape mapping* between a VO sequence and the shape (or conformation) it forms. External contour points assigned to UNIFs capture the shape of objects such as a 3-D mineral structure, or a protein 3-D fold. The thrust in this task is to investigate how the observation symbols are seamlessly tied together and transformed to form a meaningful structure of an object. In the protein fold application, 3-D coordinate points of amino acid atoms in the protein are available in dedicated repositories, and therefore protein shape extraction becomes feasible by computing the protein external contour. However, in order to consider the shape information during the prediction task of any visible observation sequence O, one has to determine a mapping between a segment of the VO sequence and the contour of its UNIF. We first map through a function f the VO sequence to its *UNIF element*: this mapping is called "sequence instantiation," since a UNIF is an instantiation of a VO sequence. We further map through a function g the UNIF to its shape using a contour representation technique. Hence, a *Fourier* or a *wavelet* coefficient vector $[a_0, a_1, \ldots, a_j]^T$ representing the external contour is computed. This mapping is called "shape representation." The composite function $(g \circ f)$ maps the VO sequence $O = o_1, o_2, \ldots, o_T$ to its shape vector which is defined in a Euclidean space. This mapping allows the HMM-based models to be ingrained in a Euclidean space. Therefore, distance between shapes can be exploited within the HMMs framework. The composite mapping is depicted as follows:

$$\begin{tikzcd}
\underset{\{o_1o_2...o_k\}}{\mathcal{O}} \arrow[r, "f\text{: Instantiation}"] \arrow[dr, "(g\circ f)"'] & \underset{Unif_{\{o_1o_2...o_k\}}}{\mathcal{U}\subset\Re^n} \arrow[d, "g\text{: Representation}"] \\
& \underset{[a_0, a_1, ..., a_j]^T}{\mathcal{T}}
\end{tikzcd}$$

V. Conformation-Based Hidden Markov Models

The goal of the COHMMs is to classify VO sequences made of symbols that when grouped together and deformed in a certain manner may exhibit shapes. It is noteworthy that not all sequences of symbols encountered in nature possess this pattern of disclosing shapes.

Furthermore, the shapes assigned to UNIFs are captured by their external contours. A contour can be viewed as a discrete signal that consists of low-frequency and high-frequency contents. The low-frequency content is the most important part of the signal, since it provides the signal with its identity. This part is known as *the pure signal.* However, the high-frequency signal conveys flavor or nuance. This part is usually *associated with noise.* For example, the Fourier transform $c(k)$ of a function $f(t)$ is computed for only a limited number of k values which cover lower and higher frequency terms. Similarly, the wavelet analysis uses two technical terms which are *approximations A* (low resolution view of the image: low-frequency components) and *details D* (details of the image at different scales and orientations: high-frequency components). Approximations refer to the high-scale factor; these components of the signal are matched with the stretched wavelets. However, details represent the low-scale factor; these components of the signal are matched with the compressed wavelets. The thrust behind the concept of COHMMs is to express the probability distribution assigned to the shape of the pure signal as a function of the Gaussian distribution assigned to the shape of the signal noise. Therefore, the tasks in the COHMMs consist of: 1) representing the shape formed by the VO sequence through any state-of-the-art shape analysis technique, and 2) modeling the noise uncertainty assigned to the shape via a Gaussian distribution.

A. UNIF Shape Representation

Let $O = o_1, o_2, \ldots, o_T$ be a VO sequence of length T made of symbols o_i. Let $X(t) = \{x(t)\}_{t=1}^{t=m}$ be the closed contour representation of length m that captures the shape of its UNIF. Each n-dimension point of this contour is designated by $x(t) = [x_1(t), x_2(t), \ldots, x_n(t)]^T$. For the sake of simplicity, we focus in this paper on 3-D objects $(n = 3)$. Object shape representation can be performed in the spatial domain or in the transform domain. Our goal is to extract the noisy part of a signal during the shape analysis of the object. If we adopt the 3-D Fourier descriptor (FD) method to efficiently discriminate the external contour of an object, therefore the contour $X(t)$ (regarded as a 2π periodic function) is approximated using an infinite sum of *sine* and *cosine* functions. In a 3-D space, if $\omega = (2\pi \times t/T)$ (T is the total contour length), then using Lin and Hwangs' direct scheme FD representation [14], we can estimate (using a hat notation) each point $x(t)$ of the external contour $X(t)$ as

$$\begin{aligned}\hat{x}(t) &= \begin{bmatrix}\hat{x}_1(t)\\ \hat{x}_2(t)\\ \hat{x}_3(t)\end{bmatrix}\\ &= \begin{bmatrix}a_0\\ c_0\\ e_0\end{bmatrix} + \sum_{k=1}^{k=N}\begin{bmatrix}a_k & b_k\\ c_k & d_k\\ e_k & f_k\end{bmatrix}\begin{bmatrix}\cos(k\omega)\\ \sin(k\omega)\end{bmatrix}\\ &\quad + \sum_{k\geq N+1}\begin{bmatrix}a_k & b_k\\ c_k & d_k\\ e_k & f_k\end{bmatrix}\begin{bmatrix}\cos(k\omega)\\ \sin(k\omega)\end{bmatrix}\end{aligned}$$

where a_k, b_k, c_k, d_k, e_k, and f_k are the Fourier coefficients corresponding to the kth harmonics. Practically, we are often satisfied with a finite number N of these functions. The inherent presence of noise in the raw data o_t warrants the use of FDs. In the scenario where N is large, a random noise is added during the external contour reconstruction. We assume that the noisy part in (3) starts from $k = N + 1$ and any other term is part of the pure signal.

Similarly, if we adopt the 3-D wavelets transform (constructed as separable products of 1-D wavelets by successively applying a 1-D analyzing wavelet in three spatial directions x_1, x_2, and x_3, therefore we can still approximate a 3-D signal using the *approximation* terms A and the *details* term D. These two component parts of the signal can be separately extracted using a filter bank. The original signal is the fusion of the A and D terms; they both contribute to the reconstruction process by revealing complementary characteristics of the signal. Mathematically stated, the inverse transform of a function $f(x) \in L^2$ with respect to some analyzing wavelet Ψ_{jk} (j: scale, k: position) is defined as $f(x) = \sum_j \sum_k c_{j,k}\Psi_{j,k}(x)$, where $c_{j,k} = \int_{-\infty}^{+\infty} f(x)\Psi_{j,k}(x)dx$ are coefficients known as discrete wavelet transform (DWT) of $f(x)$ [15]. A discrete parametrized closed curve that represents the shape of a 3-D object of interest is the vector $\hat{x}(t) = [\hat{x}_1(t), \hat{x}_2(t), \hat{x}_3(t)]^T$. If the wavelet transform is applied independently to each of the $\hat{x}_1(t)$, $\hat{x}_2(t)$ and $\hat{x}_3(t)$ functions, we can describe the 3-D curve in terms of a decomposition of $\hat{x}(t)$. However, the noise in the image is contained mostly in the details term D of each coordinate. This random noise which is part of the whole image signal in the transform domain is modeled probabilistically via a Gaussian distribution function. In conclusion, whatever image processing technique we intend to use, we can coarsely approximate the original signal $x(t)$ by decomposing it into a sum of a pure signal and a noisy signal. We can write $x(t) \approx \hat{x}(t) = \zeta(t) \oplus N(t)$ $(t = 1, \ldots, m)$, where $\zeta(t)$ is the 3-D pure signal vector (based on Fourier descriptors, or wavelet transform coefficients) assigned to low-frequency components and $N(t)$ is a 3-D Gaussian noise vector assigned to high-frequency components, with mean vector μ_t and covariance matrix Σ_t.

B. One-Level COHMMs: Mathematical Formulation

We introduce a mathematical expression of *the one-level COHMMs* and the different problems assigned to it. We also provide a definition of this model and the parameters involved. We assume that the external contour of the shape formed by the UNIF assigned to the VO sequence is computed using any state-of-the-arts shape analysis technique. We also assume that the same symbol of a VO sequence can be located at different n-dimension coordinates in the UNIF shape. For example, the same amino acid can be located at different position coordinates in a 3-D protein fold. In this scenario, the evaluation problem is stated as follows. Given a model λ, the VO sequence O, and an approximation of its UNIF external contour sequence $\hat{X}(t) = \{\hat{x}(t)\}_{t=1}^{t=m}$, evaluate the match between λ and this VO sequence O by computing $P[O \mid \lambda]$. If q stands for the hidden state sequence assigned to O, then

$$P(O \mid \lambda) = \sum_q P[O, \hat{X}(t), q \mid \lambda]. \tag{3}$$

Using the conditional probability rule, we have

$$P[O, \hat{X}(t), q \mid \lambda] = P[\hat{X}(t) \mid O, q, \lambda] \times P(O \mid q, \lambda) \times P(q \mid \lambda). \tag{4}$$

The product $P(O \mid q, \lambda) \times P(q \mid \lambda)$ expresses a traditional HMM. Finally, the term that remains to be computed is $P[\hat{X}(t) \mid O, q, \lambda]$. By replacing $\hat{x}(t)$ with its $(\zeta(t) \oplus N(t))$ decomposition, we obtain

$$\begin{aligned} P\left[\hat{X}(t) = \{\hat{x}(t)\}_{t=1}^{t=m} \mid O, q\right] \\ = P\left[\{N(t) = \hat{x}(t) \ominus \zeta(t)\}_{t=1}^{t=m} \mid O, q\right]. \end{aligned} \tag{5}$$

However, it is reasonable to assume that the random noise embedded in the contour 3-D points is independent of the hidden state sequence q, but depends only on the visible symbols representing the raw data O, therefore

$$\begin{aligned} P\left[\{N(t) = \hat{x}(t) \ominus \zeta(t)\}_{t=1}^{t=m} \mid O, q\right] \\ = P\left[\{N(t) = \hat{x}(t) \ominus \zeta(t)\}_{t=1}^{t=m} \mid O\right] \end{aligned} \tag{6}$$

where $N(t)$ is a multivariate Gaussian distribution. Finally, the one-level COHMM evaluation problem can be written as

$$\begin{aligned} P(O \mid \lambda) \approx \sum_q P\left[\{N(t) = \hat{x}(t) \ominus \zeta(t)\}_{t=1}^{t=m} \mid O\right] \\ \times P(O \mid q, \lambda) \times P(q \mid \lambda). \end{aligned} \tag{7}$$

Since each noise point on a contour depends only on its observation symbol data o_k, therefore (7) can be extended to

$$\begin{aligned} P(O \mid \lambda) \approx \sum_q \Bigg[\prod_{t=1}^{m} P[N(t) = \hat{x}(t) \ominus \zeta(t) \mid o_{\hat{x}(t)}] \\ \times P(q_0) \times P(o_t \mid q_t) \times P(q_t \mid q_{t-1}) \Bigg] \end{aligned} \tag{8}$$

where $o_{\hat{x}(t)}$ are the k symbols of the VO sequence such that $f(o) \supset \hat{x}(t)$

$$\begin{aligned} P[N(t) = \hat{x}(t) \ominus \zeta(t) \mid o_{\hat{x}(t)}] \\ = \frac{1}{(2\pi)^{(3/2)} |\Sigma_t|^{1/2}} \\ \times \exp\left[-\frac{1}{2}[N(t) - \mu(t)]^T \Sigma_t^{-1} [N(t) - \mu(t)]\right]. \end{aligned} \tag{9}$$

Both the noise $N(t) = \hat{x}(t) \ominus \zeta(t)$ and the mean $\mu(t)$ are 3-D vectors. The mean vector and the covariance matrix are respectively maximum-likelihood (ML)-estimated by the sample mean vector: $\hat{\mu} = (1/k)\sum_{t=1}^{t=k} N(t)$, and the sample covariance matrix $\hat{\Sigma} = (1/k - 1)\sum_{t=1}^{t=k}[N(t) - \hat{\mu}][N(t) - \hat{\mu}]^T$.

We now define the one-level COHMMs.

Definition 5.1: A one-level COHMM is a quadruple $\lambda = [\pi, \mathcal{A}, \mathcal{B}, \mathcal{T}]$, where:

- $\pi = \{\pi_i\} = P(q_0 = e_i)$ is the initial hidden state probability vector;
- $\mathcal{A} = \{a_{ij}\} = P(q_{t+1} = j \mid q_t = i)$ is the hidden state transition probability matrix;
- $\mathcal{B} = \{b_j(k)\} = P(o_k \text{ at time } t \mid q_t = e_j)$ is the emission probability matrix;
- $\mathcal{T} = P[N(t) = \hat{x}(t) \ominus \zeta(t) \mid o_{\hat{x}(t)}]$ is the probability distribution function assigned to the noise produced by the k contour points $\hat{x}(t)$ that belong to $f(o)$.

Conceptually, we view the generation mode of the COHMMs formalism as follows. Each symbol o_i of a VO sequence O is emitted from a hidden state $q_j \in \{1, 2, \ldots, n\}$ at each unit time. A sequence of symbols is therefore created and distorted via the mapping f to form a particular UNIF whose shape is produced via the mapping g. The one-level COHMMs formalism produces the conformation assigned to a traditional HMM.Fig. 3 depicts the state transition graph of a one-level COHMMs.

1) The Problems Assigned to a One-Level COHMM: Four problems are assigned to a one-level COHMMs.

- Probability Evaluation: Given a model λ and a VO sequence O and with its corresponding UNIF external contour points sequence $X(t) = \{x(t)\}_{t=1}^{t=m}$, the goal is to evaluate how well does λ match O.
- Statistical Decoding: In this problem, we attempt to find the best hidden state sequence. This problem is similar to problem 2 of a traditional HMMs; it can be solved using Viterbi algorithm.
- Shape Decoding: In this problem, the task consists of determining the "correct" shape of the UNIF assigned to the VO sequence O via the noise on its external contour.
- Learning: We determine the model parameters $\lambda = [\pi, \mathcal{A}, \mathcal{B}, \mathcal{T}]$ that maximize $P(O \mid \lambda)$.

C. Multilevel COHMMs

We introduce a mathematical description of *the multilevel COHMMs* that generalizes the one-level COHMMs in order to predict complex VO sequences made of several structural components.

Psychophysical studies [16] have shown that we can recognize objects using fragments of outline contour alone. In this context, a VO sequence $O = o_1, o_2, \ldots, o_T$ is viewed as made of constituents $O_1, O_2, \ldots, O_s$. Each O_i is a string of symbols $o_i \in \Sigma$ interrelated in some way. In other words, each VO sequence O is not only one sequence in which all symbols are conditionally independent, but a sequence that is divided into a series of s strings $O_i = o_{i_1} o_{i_2} \ldots o_{i_{r_i}}$ $(1 \leq i \leq s)$. The task within the multilevel COHMMs is threefold: 1) segment a VO sequence into s "meaningful" pieces, 2) determine the shape of each UNIF of a segment O_i by embedding it in a Euclidean space, and 3) compute the joint probability of the entire VO sequence O with its UNIF sequence.

1) Optimal Segmentation of the Entire VO Sequence: The goal is to determine a methodology that enables segmenting a T-element sequence into s "meaningful" segments (or strings) using a predefined criterion. This problem is known as the (s,s) segmentation problem. Let $\mathrm{Seg}_s(O)$ be the set of all segmentations of O into s segments. Therefore, the (s,s) segmentation problem can be stated as follows. Assume we are given a sequence $O = o_1, o_2, \ldots, o_T$, where $o_i \in \Sigma$. How can we determine the best segmentation $\Delta^* \in \mathrm{Seg}_s(O)$ among all possible segmentations of O into s segments? A segmentation $\Delta \in \mathrm{Seg}_s(O)$ is defined by $s+1$ segment boundaries $1 = b_1 < b_2 < \cdots < b_s < b_{s+1} = T+1$, generating segments $O_1, O_2, \ldots, O_s$ where $O_i = o_{b_i}, \ldots, o_{b_{i+1}-1}$. The best segmentation Δ^* is the one that creates *homogeneous* segments O_i with respect to some error measure. Depending on the nature of the data, different error measures can be investigated. We propose the following error measure: $E(O_i) = \sum_{o_i \in O_i} d^2(o_i, \bar{o}_i)$, where $\bar{o}_i$ is the *most representative* symbol of the segment O_i and d is a distance. If the data are real valued and defined in a Euclidean space, therefore the most representative symbol is the *mean* and the error measure in this case is simply the variance. Since there are several possible segmentations $\Delta \in \mathrm{Seg}_s(O)$, thus the global error measure is defined as $E(O, \Delta) = \sum_{O_i \in \Delta} \sum_{o_i \in O_i} d^2(o_i, \bar{o}_i)$. Finally, the optimal segmentation task consists of finding the segmentation $\Delta^* \in \mathrm{Seg}_s(O)$ that minimizes $E(O, \Delta)$. Dynamic programming approaches is used to solve this problem in a tractable and efficient manner [17]. However, the optimal solution may not be unique. There could be more than one segmentation Δ that minimize the error measure $E(O, \Delta)$. Our strategy consists of selecting the one that has the smallest number of segments s.

2) UNIF Formation Through Unsupervised Clustering: So far, we have defined a UNIF as a shape that can unfold after a stretch (or a deformation) of a VO subsequence. However, we have not shown how this process can be achieved. The objective of this section is to unravel the formation of the UNIF entity. The UNIFs are built through an unsupervised clustering algorithm applied to a set $\mathcal{S}$ of vectors representing shapes. Each cluster gathers the shapes (formed by the constituents O_i's) that are similar in some sense. The organization of the symbols o_i contributes to the production of the UNIF U_j. For example, a cloud of points O_i representing a VO sequence forms a circle or an ellipse U_j with a certain probability $P(U_j \mid O_i)$. This circular (or elliptical) shape is viewed as a cluster that gathers all round shapes with respect to some metric distance and a fixed threshold. Therefore, we define the notion of UNIFs as follows.

Definition 5.1: By partitioning the set $\mathcal{S}$ into a set of clusters. Each cluster U is a UNIF that describes piecewise the global shape formed by the entire VO sequence O.

Fig. 4 depicts examples of two objects that are decomposed into several UNIFs (or structures).

3) Mathematical Formulation of the Multilevel COHMMs: We present the mathematical expression of the multilevel COHMMs. We also give a definition of this model and the parameters involved. Let $O = O_1, O_2, \ldots, O_s = o_{1_1}\ o_{1_2}\ \ldots\ o_{1_{r_1}},\ o_{2_1}\ o_{2_2}\ \ldots o_{2_{r_2}}, \ldots, o_{s_1}, o_{s_2}, \ldots, o_{s_{r_s}}$ (where r_1 is the number of observations in subsequence O_1 and r_2 is the number of observations in subsequence O_2, etc., such that $\sum_{i=1}^{i=s} r_i = T$). Let $U = U_1, U_2, \ldots, U_s$ be the UNIF sequence assigned to the subsequences O_i's, and $X(t) = X_1(t)X_2(t)\ldots X_s(t)$ be the sequence of all external contours assigned to the UNIF sequence. The length of $X(t)$ is equal to $m = m_1 + m_2 + \cdots m_s$, where m_j is the length of the subcontour $X_j(t)$. Each O_i is mapped to its contour $X_j(t)$ using the mapping $(g \circ f)$ defined in Section IV. Let $\hat{X}_i(t)$ be the series of 3-D points of each $X_i(t)$, and $\hat{X}(t) = \{\hat{X}_i(t)\}_{i=1}^{i=s}$ be the series of the 3-D points of the entire contour $X(t)$. The probability of the observation sequence O with its external contour $X(t)$ (defined piecewise) given a model λ can be written

$$P(O \mid \lambda) = \sum_U P[O, \hat{X}(t), U \mid \lambda]. \tag{10}$$

Since the model λ is implicitly present during the evaluation of this joint probability, therefore it is omitted. We first need to evaluate $P[O, \hat{X}(t), U]$. It is reasonable to assume that the series $\hat{X}(t)$ depends only on the observation sequence O. Thus, using Bayes' formula first and then conditional independence of the $\{\hat{X}_i(t)\}_{i=1}^{i=s}$, we can write the following:

$$P[O, \hat{X}(t), U] \approx \prod_{i=1}^{i=s} P[\hat{X}_i(t) \mid O_i] \times P(O, U). \tag{11}$$

We evaluate each term separately. We first start by computing the first term of (11), which is $P[\hat{X}_i(t) \mid O_i]$. Given the fact that the vector $N_i(t) = \hat{x}_i(t) \ominus \zeta_i(t)$, we can write

$$\begin{aligned} P[\hat{X}_i(t) \mid O_i] &\\ &= \prod_{t=1}^{t=r_i} P\left[N_i(t) = \hat{x}_i(t) \ominus \zeta_i(t) \mid O_i\right] \\ &= \prod_{t=1}^{t=r_i} P\left[N_i(t) = \hat{x}_i(t) \ominus \zeta_i(t) \mid o_{\hat{x}_i(t)}\right] \qquad (12) \\ &= \prod_{t=1}^{t=r_i} \frac{1}{(2\pi)^{(3/2)}|\Sigma_{i,t}|^{1/2}} \\ &\quad \times \exp\left[-\frac{1}{2}[N_i(t) - \mu_i(t)]^T \Sigma_{i,t}^{-1} [N_i(t) - \mu_i(t)]\right] \\ &\equiv \Phi_i. \qquad (13) \end{aligned}$$

The second term of (11) is computed as follows. For the sake of simplicity, we assume that O_i depends only on U_i, and the UNIF probability distribution is a Markov chain of order 1 (illustrated by Fig. 5). Finally, we can recursively approximate the second term of (11)

$$P(O_1, \ldots, O_s, U_1, \ldots, U_s) \approx \prod_{i=1}^{i=s} P(O_i \mid U_i) \times P(U_i \mid U_{i-1}) \tag{14}$$

where $P(U_1 \mid U_0) \equiv P(U_1)$ since the form U_0 does not exist. If

$$\frac{P(U_i \mid O_i) \times P(O_i) \times P(U_i \mid U_{i-1})}{P(U_i)} \equiv \Psi_i \tag{15}$$

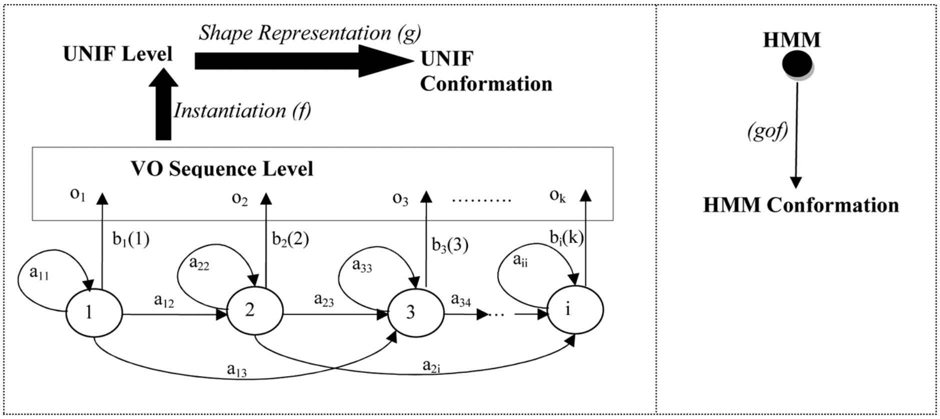

Fig. 3. State transition graph of a one-level conformation HMM. The digits 1, 2, 3, and 4 represent hidden states. The nodes o_i are emitted symbols that form a UNIF whose shape representation is projected onto a Euclidean space.

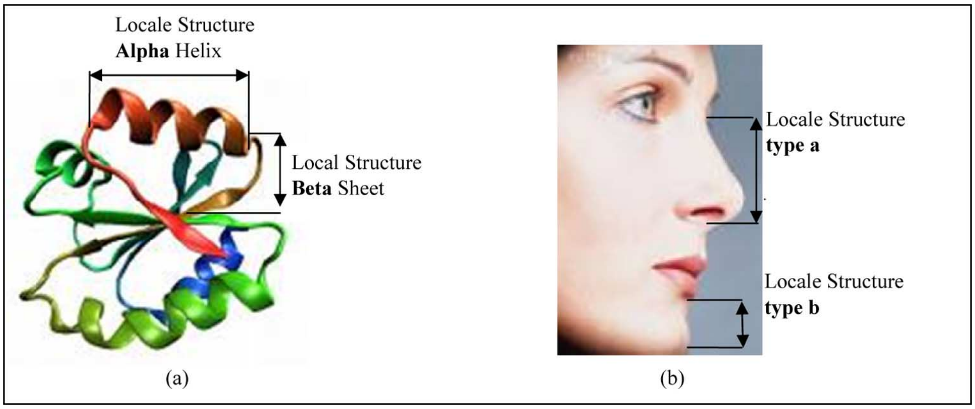

Fig. 4. Organization of constituents and their shapes in two different objects. (a) 3-D protein fold with α helix and β sheet as UNIFs. (b) Human face depicting different types of UNIFs corresponding to the facial regions.

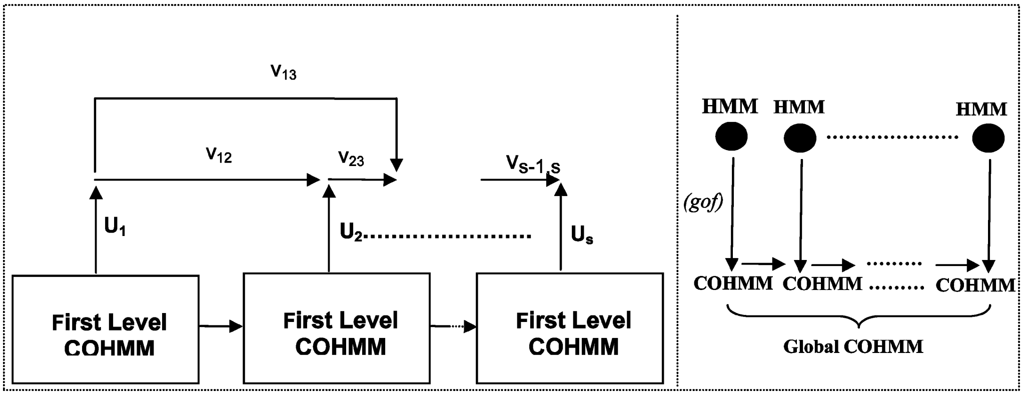

Fig. 5. State transition graph of a multilevel COHMM. The global shape of the entire VO sequence is captured piecewise through the UNIFs U_i extracted from each local one-level COHMM.

therefore, by regrouping the expressions of all the terms involved in (11), we obtain the final expression of the multilevel COHMMs

$$P(O \mid \lambda) \approx \sum_{U_1 U_2 \ldots U_s} \prod_{i=1}^{i=s} [\Phi_i] \times [\Psi_i]. \qquad (16)$$

The uncertainty about the shapes of the external contour formed by the observation sequence $O = (O_1, O_2, \ldots, O_s)$ is captured by the Gaussian noise probability distribution. The UNIF U_i assigned to O_i is introduced via the term $P(U_i \mid O_i)$. Besides, the term $P(O_i)$ of (16) is viewed as a one-level COHMM. Therefore, we can state the following.

Definition 5.2: A multilevel COHMM is a sextuple $\lambda = [\pi, \mathcal{A}, \mathcal{B}, \mathcal{U}, \mathcal{D}, \mathcal{T}]$, where we have the following.

- π is the initial hidden state distribution within a constituent O_i, where $\pi_i = P(q_0 = i)$ and $1 \leq i \leq N$, $\sum_i \pi_i = 1$.

- $\mathcal{A}$ is the hidden state transition probability distribution matrix within a constituent O_i, $\mathcal{A} = \{a_{ij}\}$, where $a_{ij} = P(q_{t+1} = j \mid q_t = i)$ and $1 \leq i, j \leq N$, $\sum_j a_{ij} = 1$.
- $\mathcal{B}$ is the emission probability matrix within a constituent O_i, $\mathcal{B} = \{b_j(k)\}$, in which $b_j(k) = P(o_k \mid q_j)$, $1 \leq k \leq R$ and $1 \leq j \leq N$, $\sum_k b_j(k) = 1$.
- $\mathcal{U}$ is the posterior probability matrix of a UNIF U_i given its corresponding constituent $O_i, \mathcal{U} = P(U_j \mid O_i) = u_i(j)$, subject to $\sum_j u_i(j) = 1$.
- $\mathcal{V}$, the UNIF transition probability matrix, where $\mathcal{V} = \{v_{ij}\} = P(U_{t+1} = j \mid U_t = i)$, $\sum_j v_{ij} = 1$, $1 \leq i$, $j \leq F$.
- $\mathcal{T}$ is the Gaussian probability density function of the noise contained in the representation of the shape $X_i(t)$ formed by the subsequence O_i; it is written as

$$P[N_i(t) = \hat{x}_i(t) \ominus \zeta_i(t) \mid O_i] = \left(\frac{1}{(2\pi)^{(3/2)}|\Sigma_{i,t}|^{(1/2)}}\right) \times \exp\left[-\frac{1}{2}[N_i(t) - \mu_i(t)]^T \Sigma_{i,t}^{-1} [N_i(t) - \mu_i(t)]\right]$$

.

- $\mathbf{N}$ is the number of hidden states in the model. We label the individual states as $1, 2, \ldots, N$, and denote the state at time t as q_t, $\mathbf{R}$, the number of points in an external contour $X_i(t)$, and $\mathbf{F}$, the number of distinct UNIFs.

Fig. 5 depicts the state transition graph of a multilevel COHMM.

4) Problems Assigned to a Multilevel COHMM: There are five problems that arise in the context of a multilevel COHMM.

- Probability Evaluation: Given a model $\lambda = [\pi, \mathcal{A}, \mathcal{B}, \mathcal{U}, \mathcal{V}, \mathcal{T}]$ and a sequence of observations $O = (O_1, \ldots, O_s)$, we evaluate how well does the model λ match O. This problem has been discussed in Section V-C3. It can be implemented using the forward procedure as in the traditional HMMs.
- Statistical Decoding: The statistical decoding problem consists of determining the optimal hidden state sequence $q^* = \arg\max_q [P(O_i, q \mid \lambda)]$ that best "explains" a constituent O_i. This process is repeated for each constituent of the entire sequence O. This task can be implemented using Viterbi algorithm.
- Structural Decoding: The structural decoding problem consists of determining the optimal UNIF sequence $U^* = \langle U_1^*, U_2^*, \ldots, U_s^* \rangle$ such that

$$U^* = \arg\max_U P(O, U \mid \lambda). \tag{17}$$

We define

$$\delta_t(i) = \max_{\mathcal{U}} [P(O_1, \ldots, O_t, U_1, \ldots, U_t = i \mid \lambda)] \tag{18}$$

that is, $\delta_t(i)$ is the highest probability along a single path, at time t, which accounts for the first t strings and ends in form i. Then, using induction, we obtain

$$\delta_{t+1}(j) = \left[\max_i \delta_t(i) v_{ij}\right] u_{t+1}(j) \frac{P(O_{t+1})}{P(U_j)}. \tag{19}$$

Similarly, this latter expression can be computed using *Viterbi* algorithm. However, we estimate δ in each step through the UNIF transition probability matrix. For example, a sequence such as $\langle \mathrm{round}, \mathrm{curved}, \mathrm{straight}, \mathrm{zigzag}, \ldots, \mathrm{convex} \rangle$ can be derived to describe the global shape formed by the VO sequence.
- Shape Decoding: The task consists of determining the "correct" shapes of the UNIFs assigned to $O_i's$ via the noise embedded in their external contours $X_i(t)$. For example, the UNIF sequence $\langle \mathrm{round}, \mathrm{straight}, \mathrm{zigzag}, \ldots, \mathrm{convex} \rangle$ is decoded in terms of its contour vector sequence. Likewise, in the protein fold mapping application, because of the shape consideration, it becomes possible to differentiate between low energy state levels of two protein secondary structures such as "CompressedHelix" and "ElongatedHelix" that are often considered to be the same. This difference is fundamental in proteomics since the folding mode is related to the energy state level. The COHMM formalism is inherently suitable for this application: The "structural" decoding would be the equivalent of protein secondary structure identification, and the "shape" decoding would relate to the protein tertiary structure identification.

5) Parameter ReeEstimation: Learning: Many algorithms have been proposed to reestimate the parameters for traditional HMMs. For example, Djuric *et al.* [18] used "Monte Carlo Markov chain" sampling scheme. In the COHMM formalism, we have used the standard "forward–backward" (variant of the generalized expectation maximization algorithm) algorithm to reestimate the model parameters $\lambda = [\pi, \mathcal{A}, \mathcal{B}, \mathcal{U}, \mathcal{V}, \mathcal{T}]$. The goal is to iteratively search for a set of parameters of the COHMMs that maximize $P(O \mid \lambda)$. This is equivalent to maximizing the auxiliary function

$$Q(\lambda, \lambda^j) = \frac{1}{P(O \mid \lambda)} \times \sum_U P[O, X(t), U \mid \lambda] \log P[O, X(t), U \mid \lambda^j]. \tag{20}$$

We have used a bottom-up strategy that consists of reestimating $\{\pi_i\}$, $\{a_{ij}\}$, $\{b_j(k)\}$ in a first phase and then reestimating $\{u_j(k)\}$, and $\{v_{ij}\}$, $\mu_i(t)$, and $\Sigma_{i,t}$ in a second phase.

The estimation of the density function $P(U_j \mid O_i) \propto P(O_i \mid U_j)$ is established through a weighted sum of c Gaussian mixtures. The mathematical expression of this estimation is

$$P(O_i \mid U_j) \approx \sum_{r=1}^{r=c} \alpha_{j,r} N(\mu_{j,r}, \Sigma_{j,r}, O_i) \tag{21}$$

where $N(\mu_{j,r}, \Sigma_{j,r}, O_i)$ is a Gaussian distribution with mean $\mu_{j,r}$ and covariance matrix $\Sigma_{j,r}$, $\alpha_{j,r}$ is the mixture coefficient for the rth mixture in U_j. The mixture coefficients are subjects to the constraint $\sum_{r=1}^{r=c} \alpha_{j,r} = 1$. This *Gaussian mixture* posterior probability estimation technique obeys the exhaustivity and exclusivity constraint $\sum_j u_i(j) = 1$. This estimation enables the entire matrix $\mathcal{U}$ to be built. The forward–backward algorithm is used to estimate the matrix $\mathcal{V}$. The other parameters

$\pi = \{\pi_i\}$, $\mathcal{A} = \{a_{ij}\}$, and $\mathcal{B} = \{b_j(k)\}$ were estimated like in traditional HMMs [11].

Let us define the following.

- $\xi^{(r)}(u,v)$ as the probability of being at UNIF u at time r and UNIF v at time $(r+1)$ given the model λ and the observation sequence O. We can write

$$\begin{aligned}\xi^{(r)}(u,v) &= P(U_r = u, U_{r+1} = v \mid \lambda, O)\\ &= \frac{P(U_r = u, U_{r+1} = v\ , O \mid \lambda)}{P(O \mid \lambda)}.\end{aligned} \tag{22}$$

Using Bayes formula, we can write (23), shown at the bottom of the page. Then, we define the following probabilities:

- $\alpha_r(u) = P(O_1O_2\ldots O_r, U_r = u \mid \lambda)$;
- $\beta_r(u) = P(O_{r+2}\ldots O_T \mid U_r = u, \lambda)$;
- $\delta_v(O_{r+1}) = P(U_{r+1} = v \mid O_{r+1})$;
- $P_v(O_{r+1}) = \delta_v(O_{r+1}) \times (P(O_{r+1})/P(U_{r+1} = v))$;

therefore

$$\xi^{(r)}(u,v) = \frac{\alpha_r(u)v_{uv}\delta_v(O_{r+1})P(O_{r+1})\beta_{r+1}(v)}{P(O_1O_2\ldots O_T \mid \lambda)P(U_{r+1} = v)}. \tag{24}$$

We need to compute the following.

- $P(O_{r+1}) = P\left(o^1_{r+1}\ldots o^k_{r+1} \mid \lambda\right) = \sum_{all\ q} P(O_{r+1} \mid q,\lambda)P(q \mid \lambda) = \sum_{q_1\ldots q_T} \pi_{q_1} b_{q_1}(o_1)a_{q1q2}\ldots b_{q_k}(o_k)$.
- $P(U_{r+1} = v) = \sum_j P(U_{r+1} = v \mid U_r = j)$.
- The term $P(O_1O_2\ldots O_T \mid \lambda)$ requires π, $\mathcal{A}$, $\mathcal{B}$, $\mathcal{U}$, $\mathcal{V}$, and $\mathcal{T}$. However, the parameters π, $\mathcal{A}$, and $\mathcal{B}$ can be reestimated as in traditional HMM; in order to reestimate $\mathcal{U}$ and $\mathcal{V}$, we define

$$\gamma_r(u) = \sum_{v=1}^{N} \xi_r(u,v). \tag{25}$$

We then compute the improved estimates of $u_v(r)$ and v_{uv} as

$$\hat{v}_{uv} = \frac{\sum\limits_{r=1}^{T-1} \xi^{(r)}(u,v)}{\sum\limits_{r=1}^{T-1} \gamma_r(u)} \tag{26}$$

$$\hat{u}_v(r) = \frac{\sum\limits_{r=1, O_r = v_r}^{T} \gamma_r(v)}{\sum\limits_{r=1}^{T} \gamma_r(v)}. \tag{27}$$

From (27), we derive

$$\hat{u}_r(v) = \hat{u}_v(r) \times \frac{\hat{P}(U_r = v)}{\hat{P}(O_r)}. \tag{28}$$

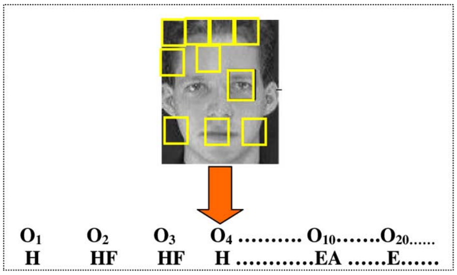

Fig. 6. Face O is viewed as an ordered visible observation sequence $O = O_1, O_2, \ldots, O_s$ of blocks. Each O_i is assigned to a facial region such as: "hair" (H), "both hair and forehead" (HF), "forehead only" (F), "ears" (EA), "eyes" (E), "nose" (N), "mouth" (M), "chin" (C), etc. These regions come in a natural order from top to bottom and left to right.

Finally, the parameters of $\mathcal{T}$ are estimated as

$$\begin{aligned}\hat{\mu}_i^{(r+1)}(t) &= \frac{1}{n}\sum_{j=1}^{j=n} N_i^j(t);\\ \hat{\Sigma}_{i,t}^{(r+1)} &= \frac{1}{n-1}\sum_{j=1}^{j=n}\left[N_i^j(t) - \hat{\mu}_i^{(r+1)}(t)\right]\\ &\quad \times \left[N_i^j(t) - \hat{\mu}_i^{(r+1)}(t)\right]^T.\end{aligned} \tag{29}$$

We calculate improved $\xi^{(r)}(u,v)$, $\gamma_r(u)$, $\hat{\mu}_i(t)$, $\hat{\Sigma}_{i,t}$, $\hat{v}_{uv}$, and $\hat{u}_r(v)$ repeatedly until some convergence criterion ε is achieved.

```
Begin initialize $\hat{v}_{uv}$, $\hat{u}_r(v)$, $\hat{\mu}_i(t)$ (sample mean vector), $\hat{\Sigma}_{i,t}$ (sample covariance matrix), training sequence $z$, and $\varepsilon$
repeat
    $z \leftarrow z + 1$
    compute $\hat{v}(z)$ from $v(z-1)$ and $u(z-1)$ using (26)
    compute $\hat{u}(z)$ from $v(z-1)$ and $u(z-1)$ using (27)
    compute $\hat{\mu}_i(t,z)$ from $\hat{\mu}_i(t,z-1)$; compute $\hat{\Sigma}_{i,t}(z)$ from $\hat{\Sigma}_{i,t}(z-1)$
    $v_{uv}(z) \leftarrow \hat{v}_{uv}(z-1)$; $u_{rv}(z) \leftarrow \hat{u}_{rv}(z-1)$; $\mu_i(t,z) \leftarrow \hat{\mu}_i(t,z-1)$; $\Sigma_{i,t}(z) \leftarrow \hat{\Sigma}_{i,t}(z-1)$
until $\max_{u,v,r,i}[v_{uv}(z) - v_{uv}(z-1), u_{rv}(z) - u_{rv}(z-1), \mu_i(t,z) - \mu_i(t,z-1), \Sigma_{i,t}(z) - \Sigma_{i,t}(z-1)] < \varepsilon$ (convergence achieved)
return $v_{uv} \leftarrow \hat{v}_{uv}(z)$; $u_{rv} \leftarrow \hat{u}_{rv}(z)$; $\mu_i(t) \leftarrow \hat{\mu}_i(t,z)$; $\Sigma_{i,t} \leftarrow \hat{\Sigma}_{i,t}(z)$
End
```

$$\xi^{(r)}(u,v) = \frac{P(O_1O_2\ldots O_r, U_r = u \mid \lambda)v_{uv}P(O_{r+1})P(O_{r+2}O_{r+3}\ldots O_T \mid U_r = u,\lambda)}{P(O_1O_2\ldots O_T \mid \lambda)}. \tag{23}$$

The convergence criterion that we have selected halts learning when no parameter values change more than a predetermined positive amount ε. Other popular stopping criteria (e.g., as the one based on overall probability that the learned model could have produced the entire training data) can also be used. However, these two criteria can produce only a local optimum of the likelihood function; they are far from reaching a global optimum.

Algorithm 1: The different steps involved in the multilevel COHMMs classifier.

```
• Training:
   1) Collect a training set containing VO sequences of
      arbitrary sizes.
   2) Break up each VO sequence into segments as
      explained in Section V-C1.
   3) Determine the UNIF element sequence assigned to
      these segments through instantiation (function $f$).
   4) Compute the shape representation vectors of these
      UNIF elements (function $g$).
   5) Partition these vectors into $F$ clusters labeled $U_i$
      ($i = 1, \ldots, F$).
   6) Extract the noise component in each shape
      representation vector using a filter bank.
   7) Compute the optimal model $\lambda^* = [\pi^*, \mathcal{A}^*, \mathcal{B}^*, \mathcal{U}^*,
      \mathcal{V}^*, \mathcal{T}^*]$ for each class $\omega_i$ ($i = 1, \ldots, c$).
• Testing:
   1) Classification
        Break up each VO sequence into segments
        Determine the UNIF elements assigned to these
        segments and their contours
        For each sequence $O$ of the test set Do
          Begin
            Compute $P(O \mid \lambda_i)$ ($i = 1, \ldots, c$).
            Select the best model and assign its class $\omega_i$
            to the test sequence $O$.
          End
   2) Compute the accuracy of the multilevel COHMMs
      using a reestimation method.
```

VI. Selected Application: Human Face Identification

In order to demonstrate the overall significance of the COHMMs paradigm, we have implemented some HMM-based models with different feature extraction techniques to perform human face identification. The comparison results between all these classifiers are reported in this section. We show how the concept of COHMMs can be used to model human faces for an identification task. We compare COHMMs with the results of the traditional and embedded HMMs when run on a same set of face databases.

A. COHMMs Face Modeling

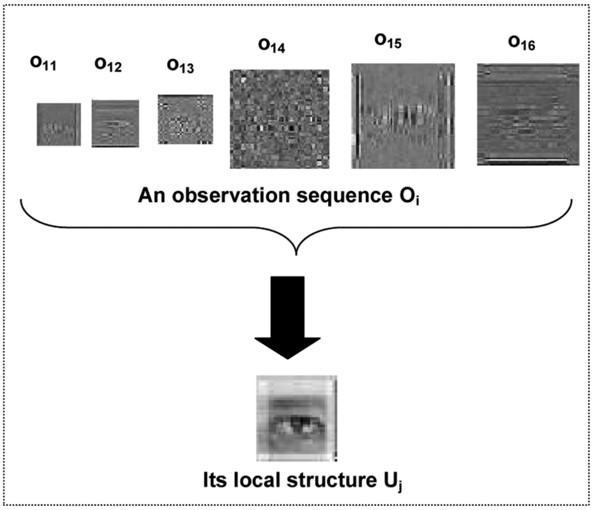

Fig. 7. Block O_i of the whole face O is assigned through instantiation to a sequence of norms assigned to the multiresolution detail images. This UNIF element belongs to the UNIF cluster (or local structure) "eyes."

COHMMs approach to face recognition consists of viewing a face as a visible sequence of blocks of information $O_i's$, where each block is a fixed-size 2-D window. Each block image undergoes DWT decomposition, producing an *average image* and a sequence of *detail images*. The subimage is then decomposed to a certain level and the energies of the subbands are selected to form a particular representation of a UNIF assigned to the block O_i. In this application, the DWT represents the instantiation function f, which produces a UNIF element vector $U_j = f(O_i)$ defined in a Euclidean space. However, the UNIF is a cluster made of these UNIF element vectors and is assigned to a human facial region using an unsupervised clustering algorithm. A UNIF element U_j belongs to some predefined facial regions as depicted in Fig. 6. The UNIF element vectors are obtained by scanning the image from left to right and top to bottom using the fixed-size 2-D window. In the case one uses Gabor filters, the original image will be convolved with a number of Gabor kernels, producing 24 output images. These images are furthermore divided into blocks using the same fixed-size 2-D windows as for DWT. The energies of these blocks are calculated. These energies correspond to the different resolutions of the block images of the face. A UNIF element vector assigned to the observation sequence O_i corresponds to the sequence of matrix norms of the detail images d_j^k. Therefore, each UNIF element is a multidimensional feature vector. The clusters are the UNIF $U_i's$ of the COHMMs; they represent the facial regions. These regions are: "hair," "forehead only," "both hair and forehead," "ears," "eyes," "nose," "mouth," etc. A subsequence $O_1, O_2, \ldots, O_k$ ($k \leq s$) representing a block image sequence of the face captures an entire facial region. Each block image is assigned one and only one facial region. Formally, a UNIF U_j is a cluster that gathers all "similar" UNIF elements $f(O_i)$. Two UNIF elements (two sets of detail images) are similar if they share the same individual facial region.Fig. 7 depicts an example of a UNIF and its visible sequence of observations. This modeling enables the COHMMs to be trained efficiently since several sets of detail images are assigned to the same facial region.

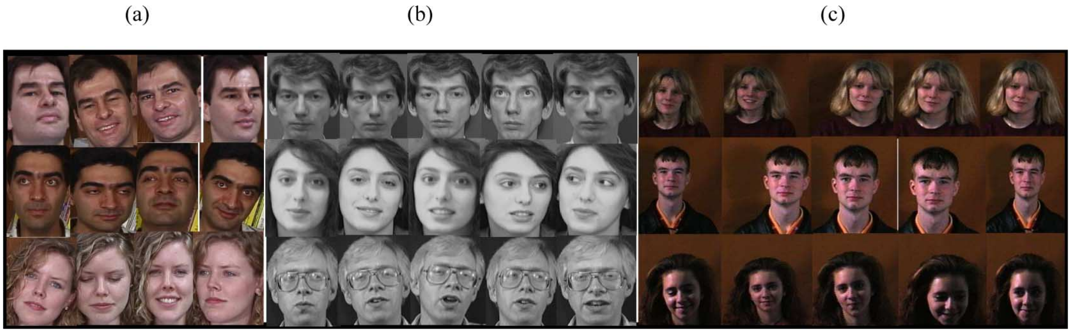

Fig. 8. Samples of faces from (a) the GeorgiaTech database [19], (b) the AT&T database [20], and (c) the Essex Faces95 database [21]. The images contain variation in pose, expression, scale, and illumination, as well as presence/absence of glasses.

TABLE I
COMPARISON OF STANDARD HMMS FACE IDENTIFICATION ACCURACY WHEN PERFORMED IN THE SPATIAL DOMAIN AND IN THE TRANSFORM DOMAIN (WITH SELECTED WAVELET FILTERS) (IN PERCENT)

	AT&T	Essex95	FERET	GeorgiaTech
Spatial	87.5	71.9	31.1	71.5
Haar	95.75	84.2	35.8	75.4
Biorthogonal 9/7	93.5	78.0	37.5	73.3
Coiflet(3)	96.5	85.6	40.5	76.8
Gabor	96.8	85.9	42.9	77.2
2D-DCT	84	75.3	35.2	72.5

B. *Training and Testing COHMMs*

The training phase of the COHMMs consists of building a model $\lambda = [\pi, \mathcal{A}, \mathcal{B}, \mathcal{U}, \mathcal{V}, \mathcal{T}]$ offline, for each human face during a training phase. Each parameter of this model will be trained through the wavelet multiresolution analysis applied to each face image of a person. The *face structural decoding* undertaken during training consists of creating the set of clusters representing the UNIFs using an unsupervised clustering algorithm. The computation of centroids of each cluster made of multidimensional feature vectors represents the *face shape decoding*. The testing phase consists of decomposing each test image into blocks and automatically assigning a facial region to each one of them using the k-means clustering algorithm. The value of k corresponds to the number of facial regions (or local structures) selected *a priori*. The selection of this value was based in part upon visual inspection of the output of the clustering process for various values of k. When k is set to 6, the clustering process appeared to perform well, segmenting the face image into regions such as "forehead," "mouth," etc. Each face is expressed as a sequence of blocks O_i with their facial regions U_i. The recognition phase is performed by computing the model λ^* that maximizes the likelihood of a test face image.

C. *Experiments*

Several experiments were conducted using four different face databases: *GeorgiaTech* [19], *AT&T* [20], *Essex Faces95* [21], and *FERET* [22], [23]. The GeorgiaTech database is made of 450 images (15 color JPEG images at resolution 640×480 pixels) representing 50 people. These individuals are from different races and genders and with different age intervals. We have taken 15 images from each individual; these images contain different lighting illumination levels. The AT&T (formerly ORL) database of faces contains ten grayscale images each of 40 individuals. The images contain variation in lighting, expression, and facial details (for example, glasses/no glasses). The third database used was the Essex Faces95 database, which contains 20 color images each of 72 individuals. These images contain variation in lighting, expression, position, and scale. For the purposes of the experiments carried out, the Essex faces were converted to grayscale prior to training. The fourth database used was the facial recognition technology (FERET) grayscale database. Images used for experimentation were taken from the fa (regular facial expression), fb (alternative facial expression), ba (frontal "b" series), bj (alternative expression to ba) and bk (different illumination to ba) images sets. Those individuals with at least five images (taken from the specified sets) were used for experimentation. This resulted in a test set of 119 individuals. These images were rotated and cropped based on the known eye coordinate positions, followed by histogram equalization. Experimentation was carried out using Matlab on a 2.4-GHz Pentium 4 PC with 512 MB of memory. Fig. 8 shows some images taken from the GeorgiaTech database [Fig. 8(a)], the AT&T database [Fig. 8(b)], and the Essex database [Fig. 8(c)].

- **Face Identification Results using Wavelet/Standard HMMs.** In order to compare the COHMMs approach with the standard HMMs, we have carried out a set of experiments that uses different wavelet filters for feature extraction with HMMs-based face identification. A variety of DWT filters were used, including Gabor DWT/Haar, DWT/biorthogonal9/7, and DWT/Coiflet(3). The observation vectors were produced using the matrix norm sequence d_j^k (described in Section VI-A), with both height, j and width k of observation blocks set to 16, with overlap of 4 pixels. The blocks size was set so that significant structures/textures could be adequately represented within the block. The overlap value of 4 was deemed large enough to allow structures (e.g., edges) that straddled the edge of one block to be better contained within the next block. Wavelet decomposition was conducted to the fourth decomposition level allowing a complete decomposition of the image. In the case of Gabor filters, six scales and four orientations were used, producing an observation blocks of size 24. The experiments were implemented using fivefold cross validation. This involved splitting the set of training images for each person into 5 equally sized

TABLE II
COMPARISON OF FACE IDENTIFICATION ACCURACY USING WAVELET/HMM AND WAVELET/COHMMS (IN PERCENT)

	AT&T		**Essex**		**FERET**	
	DWT/HMM	DWT/COHMM	DWT/HMM	DWT/COHMM	DWT/HMM	DWT/COHMM
Haar	95.75	98.5	84.2	90.2	35.8	64.2
Biorth9/7	93.5	96.22	78.0	87.8	37.5	65.6
Coiflet(3)	96.5	98.1	85.6	92.7	40.5	68.2
Gabor	96.8	97.3	85.9	89.3	42.9	59.4

sets and using 4 of the sets for system training with the remainder being used for testing. The experiments were repeated five times with a different set used for testing each time, to provide more accurate error rate estimation. Therefore, with the AT&T database, eight images were used for training and two for testing during each run. When using the Essex Faces95 database, 16 images were used for training and four for testing during each run. For the FERET database, four images per individual were used for training, with the remaining image being used for testing. Finally, in the case of GeorgiaTech database, for each person we have used ten face images for training and the remaining five for testing.

One HMM was trained for each individual's face in the database. During testing, an image was assigned an identity according to the HMM that produced the highest likelihood value. As the task being performed was face identification, it was assumed that all testing individuals were known individuals. Accuracy of an individual run is thus defined as the ratio of correct matches to the total number of face images tested, with final accuracy set to the average accuracy of all the five cross-validation runs. The accuracy for the HMM-based face recognition performed in both the spatial and the transform domains are presented in Table I. As can be noticed from Table I, the use of DWT for feature extraction improves recognition accuracy in all face databases. One of the main reasons for this improvement is due to the compression and the decorrelation properties of these wavelet filters for natural images. With the AT&T database, accuracy increased from 87.5% when the observation vector was constructed in the spatial domain, to 96.5% when the Coiflet(3) wavelet was used. This jump represents a relative improvement classification rate of 10.3%. The increase in recognition rate is also appearing in the larger Essex95 database. Recognition rate increased from 71.9% in the spatial domain to 85.6% in the wavelet domain. As previously, the Coiflet(3) wavelet produced the best results. Recognition rate also increased for the FERET database, with the recognition rate increasing from 31.1% in the spatial domain to 40.5% in the wavelet domain. DWT has been shown to improve recognition accuracy when used in a variety of face recognition approaches, and clearly this benefit extends to HMM-based face recognition. Using Gabor filters increased recognition results even further. The identification rate for the AT&T database rose to 96.8% and the Essex figure jumped to 85.9%. A similar pattern is observed with the GerogiaTech face database. Finally, the recognition rate for the 2D-DCT (discrete cosine transform [24]) is rather lower than the wavelet domain but higher than the spatial domain except for the AT&T database.

TABLE III
ACCURACY RESULTS (IN PERCENT) ON AT&T DATABASE AND GEORGIATECH DATABASE USING DISCRETE COSINE TRANSFORM (2D-DCT) [24], INDEPENDENT COMPONENT ANALYSIS (ICA) [26], WEIGHTED PRINCIPAL COMPONENT ANALYSIS (PCA) [27], AND GABOR FILTERS/RANK CORRELATION (GFRC) [28]

Method	**AT&T**	**GeorgiaTech**
2D-DCT/HMM	84	72.5
ICA	85	75
Weighted PCA	88	73
Gabor Filters & Rank Correlation	91.5	75
2D-DCT (with six coefficients) / EHMM	**97.1**	**87**
DWT/COHMM (Haar)	**98.5**	**88.5**

- **Face Identification Results using Wavelet/COHMMs.** Other experiments were run in order to find out if the COHMMs provided a benefit over the standard and the embedded HMMs for face recognition. The same parameters (such as block size) were used for COHMMs as well as for standard and embedded HMMs. The experiments were conducted only in the wavelet domain, due to the satisfying results produced during the previous experiments. The recognition accuracy for COHMMs face recognition is presented in Table II. As can be seen from the results, the use of COHMMs increased recognition accuracy in all test cases. Indeed, the relative improvement rate from Haar/HMMs to Haar/COHMMs is around 3% when tested using the AT&T database. This is a significant increase in performance at this level of accuracy. The most significant increases in performance, however, were for the FERET data set. The use of fivefold cross-validation constrained options when it came to choosing images for experimentation. As the system was not designed to handle images with any significant degree of rotation, they were selected from those subsets which were deemed suitable—fa, fb, ba, bj, and bk. Within these subsets, however, was variation in illumination, pose, scale, and expression. Most significantly, the "b" set images were captured in different sessions from the images in the "f" sets. Coupled with the number of identities in the FERET data set that were used (119), the variation among the images made this a difficult task for a face identification system. This fact explains why the recognition rates for wavelet/HMM are rather low for this database, ranging from 35.8% when Haar was used to 42.9% for Gabor.

  Experiments were also conducted to allow comparison of the results with those reported in the literature such as the 2D-DCT/EHMM [19], [25]. Although the ability to compare works was an important consideration in the

TABLE IV
COMPARISON OF TRAINING AND CLASSIFICATION TIMES FOR AT&T DATABASE IMAGES (IN SECONDS)

	Training time per image	**Classification time per image**
Spatial/HMM	7.24	22.5
DWT/HMM	1.09	1.19
2D-DCT/EHMM	5.20	4.25
DWT/COHMM	5.15	4.20

creation of the FERET database, many authors use subsets from it that match their particular requirements. There are, however, many studies in the literature using the AT&T database that select 50% of the database images for training and the remaining 50% for testing. With this in mind, an experiment was performed with these characteristics. Table III shows that the DWT/COHMM approach performs well when compared with other state-of-the-art techniques that have used this data set. DWT/COHMM has outperformed the 2D-DCT/EHMM with a 1.4% improvement on the AT&T database and 1.5% on the GeorgiaTech database. These improvements are considered to be substantial as it is difficult to do better as we get closer to the 100% top accuracy.

Furthermore, there is an important factor in a face recognition application which is the time required for both training and testing. As is depicted by Table IV, this time is reduced substantially when DWT is being employed. Feature extraction and HMM training took approximately 7.24 s per training image when this was performed in the spatial domain using the AT&T database, as opposed to 1.09 s in the wavelet domain, even though an extra step was required (transformation to wavelet domain). This is a very substantial time difference and is due to the fact that the number of observations used to train the HMM is reduced by a factor of almost 30 in the wavelet domain. The time benefit realized by using DWT is even more obvious during the recognition stage, as the time required is reduced from 22.5 to 1.19 s. The COHMMs approach increases the time spent for both training and classification compared to the DWT/HMMs, although this is compensated by better results in recognition accuracy. Fortunately, the increase in time taken for classification is still a vast improvement on the time taken for HMM recognition in the spatial domain. Finally, COHMMs'time for training and testing is lower than the 2D-DCT/EHMM, but the difference is not very substantial since they are close in complexity.

- **Other Results for Face Identification.** Besides, other results of face identification tested on the GeorgiaTech (GT) database using "coupled HMM" (CHMM) and some combinations of "embedded and coupled HMM" (ECHMM) have been reported in the literature (refer to [25]). For a comparison purpose, we find it useful to depict these results in Table V.

TABLE V
FACE IDENTIFICATION ACCURACY (IN PERCENT) USING EHMM, TWO COMBINATIONS OF HMM/CHMM, AND ECHMM

Models	**Identification Accuracy**
EHMM	87.0
HMM-CHMM	89.0
CHMM-HMM	**92.2**
ECHMM	91.5

These results indicate that the combination CHMM-HMM represents the highest accuracy achieved so far when tested on the GT database. This improvement is due in part to the flexibility of the coupled HMMs towards face variations, scaling, and rotations predominant in the GT database. However, the "embedding effect" seems to slightly worsen the performance. Nevertheless, this drop in performance is not very significant and therefore may not be conclusive. Furthermore, it is worth to investigate whether the number of hidden layers (made of hidden states) in the EHMM/ECHMM framework is related to the shape decoding exhibited in the COHMMs formalism. Increasing the number of hidden layers will allow capturing face peculiarities but it will also raise the complexity of the ECHMM model (number of parameters to estimate). This is not an issue in the COHMMs formalism since the structural and shape decoding are both based on an unsupervised clustering. However, the validity of this clustering has to be tested adequately.

VII. CONCLUSION

We have devised a novel machine learning paradigm that extends an HMM state-transition graph to account for local structures as well as their shapes. This projection of a discrete structure onto a Euclidean space is needed in several pattern recognition tasks. Long-range structural dependencies with their metric information can therefore be explored. The results obtained from the selected application demonstrate the significance of the COHMMs formalism. The COHMMs classifier has outperformed both the traditional and the embedded HMMs classifiers. This achievement has revealed the predominant role of the local structure shape information assigned to a time series (sequence of data). An evaluation of further experiments with: 1) an increase of the number of images used in the training set which allow for a more robust estimation of the COHMM parameters, 2) a more discriminative mechanism between block windows, 3) an optimal number of UNIFs selected in the k-means clustering algorithm, and 4) a different shape representation technique, is underway. We believe that the embedment of shape properties within the HMM framework will leapfrog the mathematical modeling of spatial objects.

REFERENCES

[1] S. Fine, Y. Singer, and N. Tishby, "The hierarchical hidden Markov models: Analysis and applications," *Mach. Learn.*, vol. 32, no. 1, pp. 41–62, 1998.

[2] K. Murphy and M. Paskin, "Linear time inference in hierarchical HMMs," in *Proc. Neural Inf. Process. Syst.*, Boston, MA, Jul. 2001, pp. 833–840.

[3] D. Bouchaffra and J. Tan, "Structural hidden Markov models using a relation of equivalence: Application to automotive designs," *Data Mining Knowl. Disc. J.*, vol. 12, no. 1, pp. 79–96, 2006.

[4] D. Bouchaffra and J. Tan, "Introduction to structural hidden Markov models: Application to handwritten numeral recognition," *J. Intell. Data Ana.*, vol. 10, no. 1, pp. 67–79, 2006.

[5] A. V. Nefian and M. H. Hayes, "Face recognition using an embedded HMM," in *IEEE Conf. Audio Visual-Based Person Authenticat.*, 1999, pp. 1–6.

[6] M. Brand, "Coupled hidden Markov models for modeling interactive processes," Massachusetts Inst. Technol. (MIT) Media Lab., Cambridge, MA, Tech. Rep. 405, 1997.

[7] Z. Ghahramani and M. Jordan, "Factorial hidden Markov models," *Mach. Learn.*, vol. 29, no. 2–3, pp. 245–273, Dec. 1997.

[8] T. Kristjansson, B. Frey, and T. Huang, "Event-coupled hidden Markov models," in *Proc. IEEE Int. Conf. Multimedia Exposition*, 2000, vol. 1, pp. 385–388.

[9] Y. Bengio and P. Frasconi, "Input-output HMMs for sequence processing," *IEEE Trans. Neural Netw.*, vol. 7, no. 5, pp. 1231–1249, Sep. 1996.

[10] D. Bouchaffra, "Embedding HMM's-based models in a euclidean space: The topological hidden Markov models," in *Proc. 19th IEEE Int. Conf. Pattern Recognit.*, Tampa, FL, Dec. 2008, pp. 1–4, (Oral Presentation).

[11] L. Rabiner and B. Juang*, Fundamentals of Speech Recognition*. Englewood Cliffs, NJ: Prentice-Hall, 1993.

[12] S. Eddy, "Profile hidden Markov models," *Bioinformatics*, vol. 14, no. 9, pp. 755–63, 1998.

[13] K. Asai, S. Hayamizu, and H. Handa, "Prediction of protein secondary structures by hidden Markov models," *Comput. Appl. Biosci.*, vol. 9, no. 2, pp. 141–146, 1993.

[14] C. Lin and C. Hwang, "New forms of shape invariants from elliptic Fourier descriptors," *Pattern Recognit.*, vol. 20, no. 5, pp. 535–545, 1987.

[15] S.-T. Bow*, Pattern Recognition and Image Preprocessing*. New York: Marcel Dekker, 2002.

[16] I. Biederman and G. Ju, "Surface vs. edge-based determinants of visual recognition," *Cogn. Psychol.*, vol. 20, no. 1, pp. 38–64, 1988.

[17] R. Bellman, "On the approximation of curves by line segments using dynamic programming," *Commun. ACM*, vol. 4, no. 6, p. 284, 1961.

[18] P. M. Djuric and J. Chun, "An MCMC sampling approach to estimation of nonstationary hidden Markov models," *IEEE Trans. Signal Process.*, vol. 50, no. 5, pp. 1113–1123, May 2002.

[19] A. V. Nefian and M. H. Hayes, "Maximum likelihood training of the embedded HMM for face detection and recognition," in *Proc. IEEE Int. Conf. Image Process.*, 2000, pp. 33–36.

[20] F. Samaria, "Face recognition using hidden Markov models," Ph.D. dissertation, Eng. Dept., Cambridge Univ., Cambridge, U.K., 1994.

[21] D. Hond and L. Spacek, "Distinctive description for face processing," in *Proc. 8th British Mach. Vis. Conf.*, Essex, U.K., 1997, pp. 320–329.

[22] P. J. Phillips, H. Wechsler, J. Huang, and P. J. Rauss, "The FERET database and evaluation procedure for face recognition algorithms," *Image Vis. Comput.*, vol. 16, no. 5, pp. 295–306, 1998.

[23] P. J. Phillips, H. Moon, S. A. Rizvi, and P. J. Rauss, "The FERET evaluation methodology for face recognition algorithms," *IEEE Trans. Pattern Aanal. Mach. Intell.*, vol. 22, no. 10, pp. 1090–1104, Oct. 2000.

[24] A. V. Nefian and M. H. Hayes, "Hidden Markov models for face recognition," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, Seattle, WA, 1998, pp. 2721–2724.

[25] A. V. Nefian, "Embedded Bayesian networks for face recognition," in *Proc. IEEE Int. Conf. Multimedia Expo*, 2002, pp. 133–136.

[26] J. Kim, J. Choi, J. Yi, and M. Turk, "Effective representation using ICA for face recognition robust to local distortion and partial occlusion," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 12, pp. 1977–1981, Dec. 2005.

[27] H. Y. Wang and X. J. Wu, "Weighted PCA space and its application in face recognition," in *Proc. Int. Conf. Mach. Learn. Cybern.*, 2005, vol. 7, pp. 4522–4527.

[28] O. Ayinde and Y. H. Yang, "Face recognition based on rank correlation and Gabor filtered-images," *Pattern Recognit.*, vol. 35, no. 6, pp. 1275–1289, 2002.

Djamel Bouchaffra (M'99–SM'01) received the Ph.D. degree in computer science from University of Grenoble, France, in 1992.

Currently, he is an Associate Professor of Computer Science at the Department of Mathematics and Computer Science, Grambling State University, LA. He was a Research Scientist at the Center of Excellence for Document Analysis and Recognition, University of New York at Buffalo, and later held a position of Assistant Professor at Oakland University, MI. His field of research is in pattern recognition, machine learning, computer vision, and artificial intelligence. He is currently working on the development of mathematical models that have the ability to: 1) embed discrete structures into a Euclidean or a Riemannian space, 2) merge topology with statistics, and 3) use this fusion to perform adaptive classification of complex patterns. He has introduced both the structural and the topological hidden Markov models as two novel paradigms that attempt to implement this fusion. He has written several papers in peer-reviewed conferences and premier journals such as the IEEE Transactions on Pattern Analysis and Machine Intelligence and *Pattern Recognition* journal.

Prof. Bouchaffra was the Lead Guest Editor of a special issue in the journal of *Pattern Recognition* titled: "Feature extraction and machine learning for robust multimodal biometrics" published in March 2008 (vol. 41, no. 3). He chaired several sessions in conferences. He is among the reviewer panel of some governmental funding agencies such as NASA (ADP Program: Data Analysis and Astrophysics) and EPSRC in the United Kingdom. He was one of the general chairs of the conference ICSIT'05. He is an Editorial Board Member in several journals such as *Pattern Recognition* (Elsevier), *Advances in Artificial Intelligence* (Hindawi), etc. He is a member of the IEEE Computer Society.