EURASIP Journal on Applied Signal Processing 2004:4, 510–521

A Probabilistic Model for Face Transformation with Application to Person Identification

Florent Perronnin
Multimedia Communications Department, Institut Eurécom, BP 193, 06904 Sophia Antipolis Cedex, France
Email: perronni@eurecom.fr

Jean-Luc Dugelay
Multimedia Communications Department, Institut Eurécom, BP 193, 06904 Sophia Antipolis Cedex, France
Email: dugelay@eurecom.fr

Kenneth Rose
Department of Electrical and Computer Engineering, University of California, Santa Barbara, CA 93106-9560, USA
Email: rose@ece.ucsb.edu

Received 30 October 2002; Revised 23 June 2003

A novel approach for content-based image retrieval and its specialization to face recognition are described. While most face recognition techniques aim at modeling faces, our goal is to model the *transformation* between face images of the same person. As a global face transformation may be too complex to be modeled directly, it is approximated by a collection of local transformations with a constraint that imposes consistency between neighboring transformations. Local transformations and neighborhood constraints are embedded within a probabilistic framework using two-dimensional hidden Markov models (2D HMMs). We further introduce a new efficient technique, called turbo-HMM (T-HMM) for approximating intractable 2D HMMs. Experimental results on a face identification task show that our novel approach compares favorably to the popular eigenfaces and fisherfaces algorithms.

Keywords and phrases: face recognition, image indexing, face transformation, hidden Markov models.

1. INTRODUCTION

Pattern classification is concerned with the general problem of inferring classes or "categories" from observations [1]. The success of a pattern classification system is largely dependent on the quality of its stochastic model, which generally models the *generation* of observations, to capture the *intraclass* variability.

Face recognition is a challenging pattern classification problem [2, 3] as face images of the same person are subject to variations in facial expression, pose, illumination conditions, presence or absence of eyeglasses and facial hair, and so forth. Most face recognition algorithms attempt to build for each person P a face model $\mathcal{M}_p$ (the stochastic source of the system) which is designed to describe as accurately as possible his/her intraface variability.

This paper introduces a novel approach for content-based image retrieval, which is applied to face identification and whose stochastic model focuses on the *relation* between observations of the same class rather than the generation process. Here we attempt to model a *transformation* between face images of the same person. If $\mathcal{F}_T$ and $\mathcal{F}_Q$ are, respectively, template and query images and if $\mathcal{M}$ is the probabilistic transformation model, then our goal is to estimate $P(\mathcal{F}_T|\mathcal{F}_Q, \mathcal{M})$. An important assumption made here is that the intraclass variability is the same for all classes and thus, $\mathcal{M}$ can be shared by all individuals. As the global face transformation may be too complex to be modeled directly, the basic idea is to split it into a set of local transformations and to ensure neighborhood consistency of these local transformations. Local transformations and neighboring constraints are embedded within a probabilistic framework using two-dimensional hidden Markov models (2D HMMs). A similar approach for general content-based image retrieval appeared first in [4] and preliminary results were presented on a database of binary images.

The remainder of this paper is organized as follows. Our probabilistic model of face transformation based on 2D HMMs will be detailed in Section 2. In Section 3, we introduce turbo-HMMs (T-HMMs), a set of interdependent horizontal and vertical 1D HMMs that are exploited to approximate the computationally intractable 2D HMMs. T-HMMs are one of the main contributions of this paper and one of the keys of the success of our approach as we derive efficient formulas to compute $P(\mathscr{F}_T|\mathscr{F}_Q,\mathscr{M})$ and to train automatically all the parameters of the face transformation model $\mathscr{M}$. In Section 4, we conceptually compare our novel algorithm to two different face recognition approaches that are particularly relevant: modeling faces with HMMs [5, 6] and elastic graph matching (EGM) [7]. In Section 5, we give experimental results for a face identification task on the FERET face database [8] showing that the proposed approach can significantly outperform two popular face recognition algorithms, namely eigenfaces and fisherfaces. Finally, we outline future work.

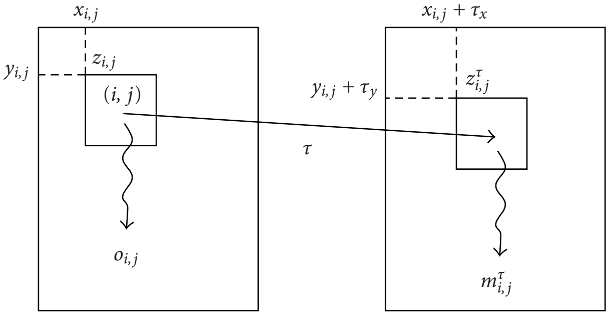


Figure 1: Local matching.

2. MODELING FACE TRANSFORMATION

In this section, we model the transformation between two face images of the same person using a probabilistic framework based on local mapping and neighborhood consistency.

2.1. *Framework*

Our assumption is that a global transformation between two face images of the same person may be too complex to be modeled directly and that it should be approximated with a set of *local transformations*. They should be as simple as possible for an efficient implementation but such that the composition of all local transformations, that is, the global transformation, should be rich enough to model a wide range of transformations between faces of the same person. However, if we allow any combination of local transformations, the model could be over flexible and capable of patching together very different faces. This naturally leads to the second component of our framework: a *neighborhood coherence constraint*. The purpose of the neighborhood constraint is to provide context information and to impose consistency requirements on the combination of local transformations. It must be emphasized that such neighborhood consistency rules produce dependence in the local transformation selection for all image regions and the optimal solution must therefore involve a global decision. To combine the local transformation and consistency costs, we propose to embed the system within a probabilistic framework using 2D HMMs.

At any location on the face, the system is considered to be in one of a finite set of states. Assuming that the 2D HMM is first-order Markovian, the probability of the system to enter a particular state at a given position, that is, the *transition probability*, depends on the state of the system at the adjacent positions in both horizontal and vertical directions. At each position, an observation is emitted by the state according to an *emission-probability distribution*. In our framework, local transformations can be viewed as the states of the 2D HMM and emission probabilities model the local mapping cost. These transformations are "hidden" and information on them can only be extracted through the observations. Transition probabilities relate states of neighboring regions and implement the consistency rules. In the following, we specify the local transformations and neighborhood constraints.

2.2. *Local transformations*

A local transformation maps a region in a template image $\mathscr{F}_T$ to a cell in a query image $\mathscr{F}_Q$. In the simplest setting, regions are obtained by tiling $\mathscr{F}_T$ into possibly overlapping blocks. However, one could envision a more complex tiling scheme where regions may be irregular cells, for example, the outcome of a segmentation algorithm. There are two possible types of transformations: *geometric* and *feature* transformations. Translation, rotation, and scaling are examples of simple geometric transformations and may be useful to model local deformations of the face. In the simple case where features are the pixel values, gray level shift or scale would be examples of simple feature transformations and could be used to compensate for illumination variations. The difference between geometric and feature transformations is not as clearcut as it may first seem and is dependent on the domain of the feature vectors. For instance, while a scaling was previously classified as geometric transformation, it could also be interpreted as a feature transformation in the Fourier domain. In the remainder of this paper, the only geometric transformation we used was the translation (if blocks are small enough, one can approximate a slight global affine transformation with a set of local translations). Hence, cells of $\mathscr{F}_Q$ are blocks of the same size as the blocks of $\mathscr{F}_T$. As we chose Gabor features (cf. Section 5.2) which are robust to small variations in illumination, we did not implement any feature transformation.

We now explicate the emission probability which models the cost of a local transformation. An observation $o_{i,j}$ is extracted from each block (i, j) of $\mathscr{F}_T$ (cf. Figure 1). Let $q_{i,j}$ be the state associated with block (i, j). The probability that at position (i, j), the system emits observation $o_{i,j}$ knowing that it is in state $q_{i,j} = \tau$, where $\tau = (\tau_x, \tau_y)$ is a translation vector, and knowing λ, the set of parameters of the HMM, is $b_\tau(o_{i,j}) = P(o_{i,j}|q_{i,j} = \tau, \lambda)$. Let $z_{i,j} = (x_{i,j}, y_{i,j})$ denote the coordinates of block (i, j) (i.e., the coordinates of its upper left pixel) in $\mathscr{F}_T$. Let $z^{\tau}_{i,j}$ be the coordinates of the matching block in $\mathscr{F}_Q$: $z^{\tau}_{i,j} = z_{i,j} + \tau$. The emission probability $b_\tau(o_{i,j})$ represents the cost of matching these two blocks.

The emission-probability $b_\tau(o_{i,j})$ is modeled with a mixture of Gaussians (linear combinations of Gaussians have the ability to approximate arbitrarily shaped densities):

$$b_\tau(o_{i,j}) = \sum_k w_{i,j}^{\tau,k} b_{\tau,k}(o_{i,j}), \qquad (1)$$

where $\{b_{\tau,k}(o_{i,j})\}$ are the component densities and $\{w_{i,j}^{\tau,k}\}$ are the mixture weights and must satisfy the constraint: $\forall(i,j)$ and $\forall\tau$, $\sum_k w_{i,j}^{\tau,k} = 1$. Each component density is an N-variate Gaussian function of the form

$$b_{\tau,k}(o_{i,j}) = \frac{1}{(2\pi)^{N/2}\left|\Sigma_{i,j}^{\tau,k}\right|^{1/2}} \times \exp\left\{-\frac{1}{2}\left(o_{i,j}-\mu_{i,j}^{\tau,k}\right)^T \Sigma_{i,j}^{\tau,k(-1)}\left(o_{i,j}-\mu_{i,j}^{\tau,k}\right)\right\}, \qquad (2)$$

where $\mu_{i,j}^{\tau,k}$ and $\Sigma_{i,j}^{\tau,k}$ are, respectively, the mean and covariance matrix of the Gaussian, N is the size of the feature vectors, and $|\cdot|$ is the determinant operator. This HMM is nonstationary as Gaussian parameters depend on the position (i,j).

The choice of notation $P(\mathcal{F}_T|\mathcal{F}_Q,\mathcal{M})$ suggests that we should separate Gaussian parameters into face-dependent (FD) parameters, that is, parameters that depend on a particular query image, and face-independent transformation (FIT) parameters, that is, the parameters of $\mathcal{M}$ that are shared by all individuals. The benefits of such a separation are discussed in Section 4.1. Let $m_{i,j}^\tau$ be the feature vector extracted from the matching block in $\mathcal{F}_Q$. We use a bipartite model which separates the mean into additive FD and FIT parts:

$$\mu_{i,j}^{k,\tau} = m_{i,j}^\tau + \delta_{i,j}^{\tau,k}, \qquad (3)$$

where $m_{i,j}^\tau$ is the FD part of the mean and $\delta_{i,j}^{\tau,k}$ is an FIT offset. Intuitively, $b_{\tau,k}(o_{i,j})$ should be approximately centered and maximum near $m_{i,j}^\tau$. The parameters we need to estimate are the FIT parameters, that is, $\{w\}$, $\{\delta\}$, and $\{\Sigma\}$.

2.3. Neighborhood consistency

The neighborhood consistency of the transformation is ensured via the transition probabilities of the 2D HMM. If we assume that the 2D HMM is first-order Markovian in a 2D sense, the transition probabilities are of the form $P(q_{i,j}|q_{i,j-1},q_{i-1,j},\lambda)$. However, we show in Section 3 that a 2D HMM can be approximated by a turbo-HMM (T-HMM): a set of horizontal and vertical 1D HMMs that "communicate" through an iterative process. The transition probabilities of the corresponding horizontal and vertical 1D HMMs are given by

$$\begin{aligned} a_{i,j}^{\mathcal{H}}(\tau;\tau') &= P(q_{i,j}=\tau|q_{i,j-1}=\tau',\lambda), \\ a_{i,j}^{\mathcal{V}}(\tau;\tau') &= P(q_{i,j}=\tau|q_{i-1,j}=\tau',\lambda), \end{aligned} \qquad (4)$$

where $a_{i,j}^{\mathcal{H}}$ and $a_{i,j}^{\mathcal{V}}$ model, respectively, the horizontal and vertical elastic properties of the face at position (i,j) and are part of the face transformation model $\mathcal{M}$. Figure 2 represents the neighborhood consistency between adjacent vertical blocks.

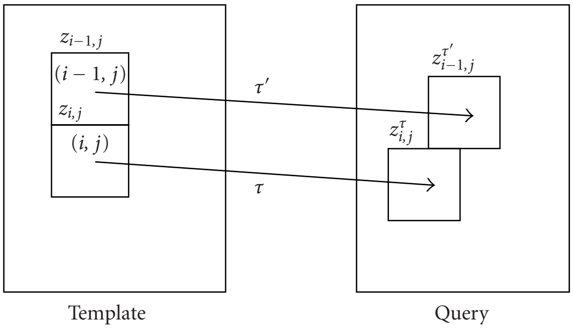


Figure 2: Neighborhood consistency.

As we want to be insensitive to global translations of face images, we choose $a^{\mathcal{H}}$ and $a^{\mathcal{V}}$ to be of the form

$$a_{i,j}^{\mathcal{H}}(\tau;\tau') = a_{i,j}^{\mathcal{H}}(\delta\tau), \qquad a_{i,j}^{\mathcal{V}}(\tau;\tau') = a_{i,j}^{\mathcal{V}}(\delta\tau), \qquad (5)$$

where $\delta\tau = \tau - \tau'$. We can apply further constraints on the transition probabilities to reduce the number of free parameters in our system. We can assume, for instance, separable transition probabilities:

$$\begin{aligned} a_{i,j}^{\mathcal{H}}(\delta\tau) &= a_{i,j}^{\mathcal{H}x}(\delta\tau_x) \times a_{i,j}^{\mathcal{H}y}(\delta\tau_y), \\ a_{i,j}^{\mathcal{V}}(\delta\tau) &= a_{i,j}^{\mathcal{V}x}(\delta\tau_x) \times a_{i,j}^{\mathcal{V}y}(\delta\tau_y). \end{aligned} \qquad (6)$$

We can also assume parametric transition probabilities. If $\mathcal{F}_T$ and $\mathcal{F}_Q$ have the same scale and orientation, then $a_{i,j}^{\mathcal{H}}$ and $a_{i,j}^{\mathcal{V}}$ should have two properties: they should preserve both *local distance*, that is, τ and τ' should have the same norm, and *ordering*, that is, τ and τ' should have the same direction. A horizontal separable parametric transition probability that satisfies the two previous constraints is

$$\begin{aligned} a_{i,j}^{\mathcal{H}x}(\delta\tau_x) &= c\left(\sigma_{i,j}^{\mathcal{H}x}\right)\exp\left\{-\frac{1}{2}\left(\frac{\delta\tau_x}{\sigma_{i,j}^{\mathcal{H}x}}\right)^2\right\}, \\ a_{i,j}^{\mathcal{H}y}(\delta\tau_y) &= c\left(\sigma_{i,j}^{\mathcal{H}y}\right)\exp\left\{-\frac{1}{2}\left(\frac{\delta\tau_y}{\sigma_{i,j}^{\mathcal{H}y}}\right)^2\right\}, \end{aligned} \qquad (7)$$

where c is a normalization factor such that $\sum_{\delta\tau_x} a_{i,j}^{\mathcal{H}x}(\delta\tau_x) = 1$ and $\sum_{\delta\tau_y} a_{i,j}^{\mathcal{H}y}(\delta\tau_y) = 1$. Similar formulas can be derived for vertical transition probabilities.

In this section, we specified and derived emission and transition probabilities but have not introduced another traditional HMM parameter: the initial occupancy probability distribution. We assume in the remainder that the initial occupancy probability is uniform to ensure invariance to global translations of face images. In the next section, we derive efficient formulas to compute $P(\mathcal{F}_T|\mathcal{F}_Q,\mathcal{M})$ and to train automatically all the parameters of the face transformation model $\mathcal{M}$, that is, $\{w\}$, $\{\delta\}$, $\{\Sigma\}$, and transition probabilities $\{a^{\mathcal{H}}\}$ and $\{a^{\mathcal{V}}\}$.

3. TURBO-HMMs

While the HMM has been extensively applied to one-dimensional problems, the complexity of its extension to two dimensions grows exponentially with the data size and is intractable in most cases of interest. Many approaches to solve the 2D problem consist of approximating the 2D HMM with one or many 1D HMMs. Perhaps the simplest approach is to trace a 1D scan that takes into account as much of the neighborhood relationship of the data as possible, for example, the Hilbert-Peano scan [9]. Another approach is the so-called pseudo 2D HMM [10] which assumes that there exists a set of "super" states which are Markovian and which subsume a set of simple Markovian states. Finally, the path-constrained variable state Viterbi algorithm [11] considers sequences of states on a row (or a column, a diagonal, etc.) as states of a 1D HMM. However, this 1D HMM has such a huge number of states that the direct application of the Viterbi algorithm is often unpractical. Hence the central idea is to consider only the N sequences with the largest posterior probabilities.

We recently introduced a novel approach that transforms a 2D HMM into a turbo-HMM (T-HMM): a set of horizontal and vertical 1D HMMs that "communicate" through an iterative process. Similar approaches have been proposed in the image processing community, mainly in the context of image restoration [12] or page layout analysis [13]. The term "turbo" was also used in [13] in reference to the now celebrated turbo error-correcting codes. However, in [13], the layout of the document is preformulated with two orthogonal grammars and the problem is clearly separated into horizontal and vertical components in distinction with the more challenging case of general 2D HMMs.

While [14] focused on decoding, that is, searching the most likely state sequence, in this section, we provide efficient formulas to (1) compute the likelihood of a set of observations given the model parameters and (2) train the model parameters.

3.1. The modified forward-backward

We assume in the following that the reader is familiar with 1D HMMs (see, e.g., [15]). Let $O = \{o_{i,j},\ i = 1,\ldots,I,\ j = 1,\ldots,J\}$ be the set of all observations. For convenience, we also introduce the notations $o_i^{\mathcal{H}}$ and $o_j^{\mathcal{V}}$ for the ith row and jth column of observations, respectively. Similarly, $Q = \{q_{i,j},\ i = 1,\ldots,I,\ j = 1,\ldots,J\}$ denotes the set of all states, while $q_i^{\mathcal{H}}$ and $q_j^{\mathcal{V}}$ denote the ith row and jth column of states. Finally, let λ be the set of all HMM parameters and let $\lambda_i^{\mathcal{H}}$ and $\lambda_j^{\mathcal{V}}$ be the respective rows and columns of parameters.

The goal of this section is to compute $P(O|\lambda)$ using the quantities introduced in Table 1. It was shown in [14] that the joint likelihood of O and Q, given λ, can be approximated by

$$P(O,Q|\lambda) \approx \prod_j \left[P\left(o_j^{\mathcal{V}}, q_j^{\mathcal{V}} \,|\, \lambda_j^{\mathcal{V}}\right) \prod_i P\left(q_{i,j} \,|\, o_i^{\mathcal{H}}, \lambda_i^{\mathcal{H}}\right) \right], \quad (8)$$

where each term $P(o_j^{\mathcal{V}}, q_j^{\mathcal{V}} | \lambda_j^{\mathcal{V}})$ corresponds to a 1D vertical HMM and the term $\prod_i P(q_{i,j}|o_i^{\mathcal{H}}, \lambda_i^{\mathcal{H}})$ is, in effect, a horizontal prior for column j. We assume that the quantity $P(q_{i,j}|o_i^{\mathcal{H}}, \lambda_i^{\mathcal{H}})$ is known, that is, it was obtained during the previous horizontal step.

Table 1: HMM notation summary.

Notation	Definition
$\pi_{q_{1,1}}$	$P(q_{1,1}\|\lambda)$
$b_{q_{i,j}}(o_{i,j})$	$P(o_{i,j}\|q_{i,j},\lambda)$
$\alpha_{i,j}^{\mathcal{H}}(q_{i,j})$	$P(o_{i,1},\ldots,o_{i,j},q_{i,j}\|\lambda_i^{\mathcal{H}})$
$\beta_{i,j}^{\mathcal{H}}(q_{i,j})$	$P(o_{i,j+1},\ldots,o_{i,J}\|q_{i,j},\lambda)$
$\gamma_{i,j}^{\mathcal{H}}(q_{i,j})$	$P(q_{i,j}\|o_i^{\mathcal{H}},\lambda_i^{\mathcal{H}})$
$\gamma_{i,j}(q_{i,j})$	$(\gamma_{i,j}^{\mathcal{H}}(q_{i,j}) + \gamma_{i,j}^{\mathcal{V}}(q_{i,j}))/2$

If we sum over all possible paths, we obtain the following marginal:

$$\begin{aligned} P(O|\lambda) &= \sum_Q P(O,Q|\lambda) \\ &\approx \sum_{q_1^{\mathcal{V}}\cdots q_J^{\mathcal{V}}} \prod_j \left[P\left(o_j^{\mathcal{V}}, q_j^{\mathcal{V}} \,|\, \lambda_j^{\mathcal{V}}\right) \prod_i P\left(q_{i,j} \,|\, o_i^{\mathcal{H}}, \lambda_i^{\mathcal{H}}\right) \right] \\ &\approx \prod_j \sum_{q_j^{\mathcal{V}}} \left[P\left(o_j^{\mathcal{V}}, q_j^{\mathcal{V}} \,|\, \lambda_j^{\mathcal{V}}\right) \prod_i P\left(q_{i,j} \,|\, o_i^{\mathcal{H}}, \lambda_i^{\mathcal{H}}\right) \right]. \end{aligned} \quad (9)$$

We introduce the compact notation

$$P_j^{\mathcal{V}} = \left[P\left(o_j^{\mathcal{V}}, q_j^{\mathcal{V}} \,|\, \lambda_j^{\mathcal{V}}\right) \prod_i P\left(q_{i,j} \,|\, o_i^{\mathcal{H}}, \lambda_i^{\mathcal{H}}\right) \right]. \quad (10)$$

$\{P_j^{\mathcal{V}}\}$ can be computed using a modified version of the forward-backward algorithm which we describe next after introducing one last notation:

$$b_{q_{i,j}}^{\mathcal{H}}(o_{i,j}) = \begin{cases} b_{q_{i,j}}(o_{i,j}) & \text{if } j = 1, \\ b_{q_{i,j}}(o_{i,j})\gamma_{i,j}^{\mathcal{H}}(q_{i,j}) & \text{if } j > 1. \end{cases} \quad (11)$$

The forward α variables

(i) Initialization:

$$\alpha_{1,j}^{\mathcal{V}}(q_{1,j}) = \begin{cases} \pi_{q_{1,1}} b_{q_{1,1}}(o_{1,1}) & \text{if } j = 1, \\ b_{q_{1,j}}^{\mathcal{H}}(o_{1,j}) & \text{if } j > 1. \end{cases} \quad (12)$$

(ii) Recursion:

$$\alpha_{i+1,j}^{\mathcal{V}}(q_{i+1,j}) = \left[\sum_{q_{i,j}} \alpha_{i,j}^{\mathcal{V}}(q_{i,j}) a_{q_{i,j},q_{i+1,j}}^{\mathcal{V}} \right] b_{q_{i+1,j}}^{\mathcal{H}}(o_{i+1,j}). \quad (13)$$

(iii) Termination:

$$P_j^{\mathcal{V}} = \sum_{q_{I,j}} \alpha_{I,j}^{\mathcal{V}}(q_{I,j}). \quad (14)$$

The backward β variables

(i) Initialization:

$$\beta^{\mathcal{V}}_{I,j} = 1. \tag{15}$$

(ii) Recursion:

$$\beta^{\mathcal{V}}_{i,j}(q_{i,j}) = \sum_{q_{i+1,j}} a^{\mathcal{V}}_{q_{i,j},q_{i+1,j}} b^{\mathcal{H}}_{q_{i+1,j}}(o_{i+1,j})\beta^{\mathcal{V}}_{i+1,j}(q_{i+1,j}). \tag{16}$$

Occupancy probability γ

$$\gamma^{\mathcal{V}}_{i,j}(q_{i,j}) = \frac{\alpha^{\mathcal{V}}_{i,j}(q_{i,j})\beta^{\mathcal{V}}_{i,j}(q_{i,j})}{\sum_{q_{i,j}} \alpha^{\mathcal{V}}_{i,j}(q_{i,j})\beta^{\mathcal{V}}_{i,j}(q_{i,j})}. \tag{17}$$

Similar formulas can be derived for the horizontal pass. It is worthwhile to note that our reestimation equations are similar to the ones derived for the page layout problem in [13] based on the graphical model formalism. Also, we can see that the interaction between horizontal and vertical processing, which is based on the occupancy probability γ, is not as simple as the one used in [12].

Next, we consider the steps of the algorithm. We first initialize γ's uniformly (i.e., assuming no prior information). Then, the modified forward-backward algorithm is applied successively and iteratively on the rows and columns. Whether the iterative process is initialized with row or column operation may theoretically impact the performance. However, this choice had a very limited impact in our experiments and we always started with a horizontal pass. This algorithm is clearly linear in the size of the data and can be further accelerated with a parallel implementation, simply by running the modified forward-backward for each row or column on a different processor.

One should be aware that we do not end up with one score but with one horizontal score $P(O|\lambda^{\mathcal{H}})$ and one vertical score $P(O|\lambda^{\mathcal{V}})$. Combining these two scores is a classical problem of decision fusion. As experiments showed that these scores were generally very close, we simply averaged them to obtain a global score. Although this simple heuristic may not be optimal, it provided good results.

3.2. The modified Baum-Welch algorithm

We now estimate the parameters of the T-HMM. Generally, the maximum likelihood (ML) reestimation formulas can be derived directly by maximizing Baum's auxiliary function [16]

$$Q(\lambda|\bar{\lambda}) = \sum_q \log P(O,q|\lambda)P(O,q|\bar{\lambda}). \tag{18}$$

Here, the problem is that we obtain two equations

$$\begin{aligned} Q(\lambda^{\mathcal{H}}|\bar{\lambda}^{\mathcal{H}}) &= \sum_{q\in Q} \log P(O,q|\lambda^{\mathcal{H}})P(O,q|\bar{\lambda}^{\mathcal{H}}), \\ Q(\lambda^{\mathcal{V}}|\bar{\lambda}^{\mathcal{V}}) &= \sum_{q\in Q} \log P(O,q|\lambda^{\mathcal{V}})P(O,q|\bar{\lambda}^{\mathcal{V}}) \end{aligned} \tag{19}$$

that may be incompatible in the case where $\gamma^{\mathcal{H}}$'s and $\gamma^{\mathcal{V}}$'s do not converge. So a simple combination rule is to maximize

$$Q(\lambda|\bar{\lambda}) = Q(\lambda^{\mathcal{H}}|\bar{\lambda}^{\mathcal{H}}) + Q(\lambda^{\mathcal{V}}|\bar{\lambda}^{\mathcal{V}}). \tag{20}$$

To train the system, we provide a set of pairs of pictures. Each pair contains a template and a query image that belong to the same person. We now provide formulas for reestimating the Gaussian parameters and transition probabilities. Index p in the sums of the following formulas is for the pth pair of pictures. Although each quantity $o_{i,j}$, $m^{\tau}_{i,j}$, $\gamma_{i,j}$, and $\xi_{i,j}$ should be indexed with p in the following equations, we omitted this index on purpose to simplify notations.

Let $\gamma^{\mathcal{H}}_{i,j}(\tau,k)$ (resp., $\gamma^{\mathcal{V}}_{i,j}(\tau,k)$) be the probability of being in state $q_{i,j} = \tau$ at position (i,j) during the horizontal (resp., vertical) pass with the kth mixture component accounting for $o_{i,j}$:

$$\begin{aligned} \gamma^{\mathcal{H}}_{i,j}(\tau,k) &= \gamma^{\mathcal{H}}_{i,j}(\tau)\frac{w^{\tau,k}_{i,j}b_{\tau,k}(o_{i,j})}{\sum_k w^{\tau,k}_{i,j}b_{\tau,k}(o_{i,j})}, \\ \gamma^{\mathcal{V}}_{i,j}(\tau,k) &= \gamma^{\mathcal{V}}_{i,j}(\tau)\frac{w^{\tau,k}_{i,j}b_{\tau,k}(o_{i,j})}{\sum_k w^{\tau,k}_{i,j}b_{\tau,k}(o_{i,j})}, \\ \gamma_{i,j}(\tau,k) &= \frac{\gamma^{\mathcal{H}}_{i,j}(\tau,k)+\gamma^{\mathcal{V}}_{i,j}(\tau,k)}{2}. \end{aligned} \tag{21}$$

We also introduce

$$\xi^{\mathcal{H}}_{i,j}(\tau,\tau+\delta\tau) = \sum_{\tau} \frac{\alpha^{\mathcal{H}}_{i,j-1}(\tau)a^{\mathcal{H}}_{i,j}(\delta\tau)b^{\mathcal{V}}_{\tau}(o_{i,j})\beta^{\mathcal{H}}_{i,j}(\tau+\delta\tau)}{P(o^{\mathcal{H}}_i|\lambda^{\mathcal{H}}_i)}. \tag{22}$$

We assume diagonal covariance matrices and general transition matrices. The reestimation formulas are as follows (the update for a single dimension is shown for δ and σ):

$$\delta^{\tau,k}_{i,j} = \frac{\sum_p \gamma_{i,j}(\tau,k)(o_{i,j}-m^{\tau}_{i,j})}{\sum_p \gamma_{i,j}(\tau,k)}, \tag{23}$$

$$\left(\sigma^{\tau,k}_{i,j}\right)^2 = \frac{\sum_p \gamma_{i,j}(\tau,k)\left(o_{i,j}-m^{\tau}_{i,j}-\delta^{\tau,k}_{i,j}\right)^2}{\sum_p \gamma_{i,j}(\tau,k)}, \tag{24}$$

$$w^{\tau,k}_{i,j} = \frac{\sum_p \gamma_{i,j}(\tau,k)}{\sum_p \gamma_{i,j}(\tau)}, \tag{25}$$

$$a^{\mathcal{H}}_{i,j}(\delta\tau) = \frac{\sum_{p,\tau}\xi^{\mathcal{H}}_{i,j}(\tau,\tau+\delta\tau)}{\sum_{p,\tau}\gamma^{\mathcal{H}}_{i,j}(\tau)}. \tag{26}$$

A formula similar to (26) can be derived for vertical transition probabilities.

4. RELATED WORK

The goal of this section is not to provide a full review of the literature on face recognition (the interested reader can refer, for instance, to [2, 3]) but to compare the proposed approach to two different algorithms from a conceptual point of view. The first one consists in modeling faces with HMMs [5, 6]. The interesting point is that, although we use the same mathematical framework (HMMs), the philosophy is different as [5, 6] *model a face* while our algorithm models *a transformation between faces*. The second algorithm, elastic graph matching (EGM) [7], is particularly relevant to this paper as its philosophy, based on local similarity and neighborhood consistency, is similar to the philosophy of the proposed algorithm.

4.1. Modeling faces with HMMs

Modeling faces with HMMs was pioneered in [5] and later improved in [6]. While early work involved a simple top-bottom 1D HMM, a model based on pseudo 2D HMMs (P2D HMMs) [10] proved to be more successful. The assumption of P2D HMMs is that there exists a set of "super" states which are Markovian and which themselves contain a set of simple Markovian states. In the following, we do not compare approaches in terms of their mathematical frameworks, that is, we do not compare P2D HMMs to T-HMMs, but in terms of the philosophies of both methods.

While our HMM models a face transformation, HMMs in [5, 6] model faces. In our framework, the parameters of the HMM can be clearly separated into FD parameters (the features extracted from $\mathcal{F}_Q$) and FIT parameters (δ's, Σ's, w's, and transition probabilities $a^{\mathcal{H}}$'s and $a^{\mathcal{V}}$) as seen in Section 2.2. These transformation parameters are shared by all persons as we assume that they have similar facial properties. The intraclass variability, due, for instance, to different facial expressions, can therefore be estimated reliably by pooling the data of all training individuals. Of course, if one had large amounts of enrollment material for each person, one could envision to train one set of face transformation parameters per individual but the amount of enrollment data is generally scarce.

One major drawback of the approach in [5, 6] is that the separation of parameters cannot be done as easily and, generally, these HMMs *confound all sources of variability*. For instance, each HMM face has to model variations due to facial expressions. Therefore, to train the mixture of Gaussians that would correspond to the mouth, one should provide for each person an example image with the mouth in various states, open, smiling, and so forth, and it is conceivable that in each HMM face, a fair number of Gaussians models the various facial expressions. Hence, one has to train a large number of Gaussians using large amounts of training data from the same individual to get a good performance.

One drawback of our method is that we do not have a probabilistic model of the face. $m_{i,j}^{\tau}$ is directly extracted from a face image and is not the result of a training process. Nevertheless, as we efficiently separate parameters, only a small number of template images should be required to train $m_{i,j}^{\tau}$'s.

4.2. Elastic graph matching

EGM stems from the neural network community. Its basic principle is to match two face graphs in an elastic manner [7, 17]. The quality of a match is evaluated with a cost function $\mathcal{C}$:

$$\mathcal{C} = \mathcal{C}_v + \rho\mathcal{C}_e, \tag{27}$$

where $\mathcal{C}_v$ is the cost of the local matchings, $\mathcal{C}_e$ is the cost of local distortions, and ρ is a rigidity parameter which controls the balance between $\mathcal{C}_v$ and $\mathcal{C}_e$. The matching is generally a two-step procedure: the two faces are first mapped in a rigidly manner and then elastic matching is performed through iterative random perturbations of the nodes of the graph. Both optimization steps correspond to a simulated annealing (SA) at zero temperature [7].

Wiskott et al. [18] elaborated on the idea with the elastic bunch graph matching (EBGM) which can be used for face recognition and also face labeling. Both algorithms were later improved, especially to incorporate local coefficients that weight the different parts of the face according to their discriminatory power using for instance fisher's linear discriminant (FLD) [19] or support vector machines (SVM) [20].

It is clear that the philosophies of EGM and of the proposed framework are distinct but bear obvious similarities. In our approach, the joint log-likelihood of observations and states $\log P(O, Q|\lambda)$ can be separated into

$$\log P(O, Q|\lambda) = \log P(O|Q, \lambda) + \log P(Q|\lambda). \tag{28}$$

The first term, which depends on emission probabilities, corresponds to the local matchings cost $\mathcal{C}_v$ and the second term, which depends on transition probabilities, corresponds to the local distortions cost $\mathcal{C}_e$. Moreover, in the simple case where we use one Gaussian mixture, for the whole face, with a single Gaussian in the mixture ($\Sigma_{i,j}^{\tau,k} = \Sigma$) and where there is, for the whole face, one unique transition probability which is separable and parametric (cf. Section 2.3), the formula for the joint log-likelihood $\log P(O, Q|\lambda)$ would be almost identical to $\mathcal{C}$ in [7]. The main advantages of our novel approach are in: (1) the use of the well-developed HMM framework and (2) the use of a shared deformable model of the face.

First, as shown in Section 3.1, one can use a modified version of the forward-backward algorithm to compute the likelihood of the observations knowing the set of parameters. In EGM, the quality of the matching is generally assessed using a *best match* which, in the HMM framework, is equivalent to the *Viterbi* algorithm, whose aim is to find the best path in a trellis. Our score, which takes into account all paths, should be more robust.

Another advantage is the existence of simple formulas to train *automatically all the parameters* of the system (cf. Section 3.2). This is not the case with EGM as the parameter ρ is generally set manually. Duc et al. [19] showed experimentally that ρ only has a small impact on the final performance. However, as different parts of the face have different elastic properties, it would be natural to use different elastic coefficients for each part of the face. Hence, ρ may have a limited influence either because $\mathcal{C}_e$ is noninformative, which is implicitly suggested by [20], for instance, where $\mathscr{C}_\nu$ is discarded, or because the elastic properties of the face are poorly modeled with one unique parameter ρ. Using multiple elasticity coefficients is only possible if these coefficients can be trained automatically. To the best of our knowledge, it has never been investigated in the EGM framework and it is evaluated in Section 5.

Finally, while different methods have been proposed to weight the different parts of the face according to their discriminatory power [19, 20], they all suggest to train one set of parameters per person. To train these parameters, one should have a reasonable amount of enrollment data. The interpretation of "reasonable" is application dependent but at least two images should be provided by each person at enrollment time. In our case, as the model of face transformation is shared, its parameters can be trained offline and do not need to be reestimated each time a new user is enrolled. Thus, we are able to weight the different parts of the face even when one unique image is available at enrollment time.

5. EXPERIMENTS

In this section, we assess the performance of our novel algorithm on a face identification task and compare it to two popular algorithms: eigenfaces and fisherfaces.

5.1. *The database*

All the following experiments were carried out on a subset of the FERET face database [8]. We used 1,000 individuals: 500 for training the system and 500 for testing the performance. We use two images (one target and one query image) per training and test individual. This means that test individuals are enrolled with one unique image. Target faces are FA images extracted from the gallery and query images are extracted from the FB probe. FA and FB images are frontal views of the face that exhibit large variabilities in terms of facial expressions. Images are preprocessed to extract normalized facial regions. For this purpose, we used the coordinates of the eyes and the tip of the nose provided with each image. First, each image was rotated so that both eyes were on the same line. Then a square box, twice the size of the interocular distance, was centered around the nose. Finally the corresponding region was cropped and resized to 128 × 128 pixels. See Figure 5 for an example of normalized face image.

5.2. *Gabor features*

We used Gabor features that have been successfully applied to face recognition [7, 18, 19, 21] and facial analysis [22]. Gabor wavelets are defined by the following equation:

$$\psi_{\mu,\nu}(z) = \frac{\|k_{\mu,\nu}\|^2}{\sigma^2} \exp\left(-\frac{\|k_{\mu,\nu}\|^2\|z\|^2}{2\sigma^2}\right) \times \left[\exp\left(ik_{\mu,\mu}z\right) - \exp\left(-\frac{\sigma^2}{2}\right)\right], \tag{29}$$

where

(i) $\exp(ik_{\mu,\mu}z)$ is a plane wave, $k_{\mu,\nu}$, the center frequency of the filter, is of the form $k_{\mu,\nu} = k_\nu \exp(i\phi_\mu)$, and μ and ν define, respectively, the orientation and scale of $k_{\mu,\nu}$. Let $k_{\max}$ be the maximum frequency and let f be the spacing factor. Then $k_\nu = k_{\max}/f^\nu$. If M be the number of orientations, $\phi_\mu = \pi\mu/M$;

(ii) $\exp(-\|k_{\mu,\nu}\|^2\|z\|^2/2\sigma^2)$ is a Gaussian envelope which restricts the plane wave and σ determines the ratio of window width to wavelength. We should underline that, in our experiments, the plane wave is also restricted by the size of the blocks (cf. Section 2.2);

(iii) $\exp(-\sigma^2/2)$ is a term that makes the filter DC free;

(iv) $\|k_{\mu,\nu}\|^2/\sigma^2$ compensates for the frequency-dependent decrease of the power spectrum in natural images.

Each kernel $\psi_{\mu,\nu}$ exhibits properties of spatial frequency, spatial locality, and orientation selectivity. Gabor responses are obtained through the convolution of the face image and the Gabor wavelet and we use the modulus of these responses as feature vectors.

After preliminary experiments, the block size was fixed to 32 × 32 pixels and we chose the following set of parameters for the Gabor wavelets: five scales, eight orientations, $\sigma = 2\pi$, $k_{\max} = \pi/4$, and $f = \sqrt{2}$. Finally, for each image, we normalized the feature coefficients to zero mean and unit variance which performed a divisive contrast normalization [22].

5.3. *The baseline: eigenfaces and fisherfaces*

For comparison purpose, we implemented the eigenfaces and fisherfaces algorithms. We should note that both methods are examples of techniques where one attempts to build a model of the face.

Eigenfaces are based on the notion of dimensionality reduction. Kirby and Sirovich [23] first outlined that the dimensionality of the face space, that is, the space of variation between images of human faces, is much smaller than the dimensionality of a single face considered as an arbitrary 2D image. As a useful approximation, one may consider an individual face image to be a linear combination of a small number of face components or *eigenfaces* derived from a set of reference face images. One calculates the covariance or correlation matrix between these reference images and then applies principal component analysis (PCA) [24] to find the eigenvectors of the matrix: the eigenfaces. To find the best match for an image of a person's face in a set of stored facial images, one may calculate the distances between the vector representing the new face and each of the vectors representing the stored faces, and then choose the stored image yielding the smallest distance [25].

While PCA is optimal with respect to data compression [23], in general it is suboptimal for a recognition task. For such a task, a dimension-reduction technique such as FLD should be preferred to PCA. The idea of FLD is to select a subspace that maximizes the ratio of the interclass variability and the intraclass variability. However, the straightforward application of this principle is often impossible due to the high dimensionality of the feature space. A method called fisherfaces was developed to overcome this issue [26]. First,

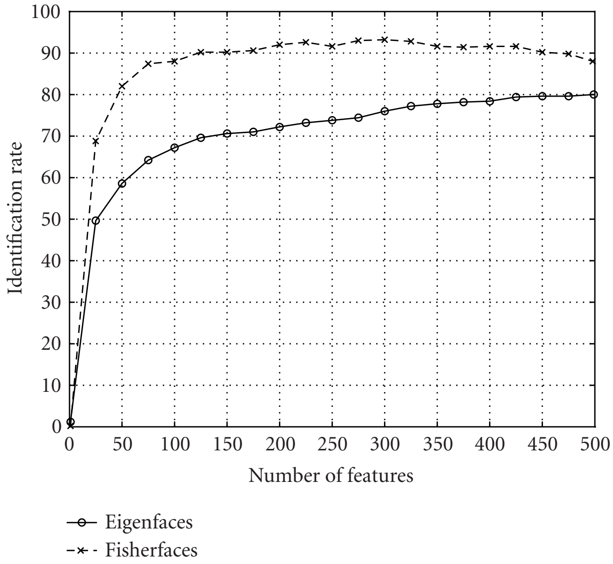


Figure 3: Identification rate of eigenfaces and fisherfaces as a function of the number of eigenfaces and fisherfaces.

one applies PCA to reduce the dimension of the feature space and then performs the standard FLD. A major similarity between our novel approach and fisherfaces is the fact that both algorithms assume that the intraclass variability is the same for all classes. The difference is in the way to deal with this variability; while fisherfaces try to cancel the intraface variability, we attempt to model it.

For a fair comparison, we did not apply directly eigenfaces and fisherfaces on the gray-level images but on the Gabor features as done, for instance, in [21]. A feature vector was extracted every four pixels in the horizontal and vertical directions (which means that there is a 28-pixels block overlap) and the concatenation of all these vectors formed the Gabor representation of the face. In [21], various metrics were tested to compute the distance between points in an eigenface or a fisherface spaces: the L_1, L_2 (Euclidean), Mahalanobis, and cosine distances. We chose the Mahalanobis metric which consistently outperformed all other distances. The performance was plotted on Figure 3 as a function of the number of eigenfaces and fisherfaces.

The best eigenfaces and fisherfaces identification rates are, respectively, 80% with the maximum possible number of eigenfaces and 93.2% with 300 fisherfaces. Fisherfaces were not guaranteed to perform so well due to the very limited number of elements per class in the training set (only two faces per person). However, in our experiments, they managed to generalize on novel test data.

5.4. Performance of the novel algorithm

Before showing experimental results of the proposed approach, we describe in detail the experimental setup. To reduce the computational load, and for a fair comparison with eigenfaces and fisherfaces, the precision of a translation vector τ was limited to 4 pixels in both horizontal and vertical directions and a feature vector m was extracted every 4 pixels of the query image. For each template image, a feature vector o was extracted every 16-pixels in both horizontal and vertical directions (which means that there is a 16-pixels block overlap) and it resulted in $7 \times 7 = 49$ observations per template image. We tried smaller step sizes for template images but this resulted in marginal improvements of the performance at the expense of a higher computational load.

We implemented traditional optimizations to speed up the algorithm at training and test time.

(i) *Windowing*: if we assume that $\mathscr{F}_T$ and $\mathscr{F}_Q$ are approximately aligned, then for each block in $\mathscr{F}_T$, one can limit the search for possible matching blocks in $\mathscr{F}_Q$ in a neighborhood (or window) of this block by setting $b_\tau(o_{i,j}) = 0$ if $|\tau_x| > T_x$ or $|\tau_y| > T_y$. While T_x and T_y should ideally be input dependent, based, for instance, on some a priori knowledge on the distortion between $\mathscr{F}_T$ and $\mathscr{F}_Q$, for simplicity, these parameters were constant in our system. After preliminary experiments, T_x and T_y were set to 8 pixels which limited the number of matching blocks, that is, of possible active states, to $5 \times 5 = 25$ at each position.

(ii) *Transition pruning*: to limit the number of possible output transition probabilities at each state, we discard unlikely transitions, that is, unreasonable deformations of the face. For the horizontal transition probabilities, we impose $a_{i,j}^{\mathscr{H}}(\delta\tau) = 0$ if $|\delta\tau_x| > \Delta_x^{\mathscr{H}}$ or $|\delta\tau_y| > \Delta_y^{\mathscr{H}}$. The same constraint can be applied to vertical transition probabilities. Similarly to the windowing parameters, while the Δ's should be input dependent, they were constant in our system. After preliminary experiments, Δ's were set to 8 pixels which limited the number of horizontal or vertical transition probabilities going out of a state to $5 \times 5 = 25$.

(iii) *Beam search*: the idea is to prune unlikely paths during the forward-backward algorithm [27]. During the forward pass, at each position (i, j), all α values that fall more than the beam width below the maximum α value at that position are ignored, that is, set to zero. Then, during the backward pass, β values are computed only if their associated α value is greater than zero. The beam size was set to 100.

The training and decoding algorithms based on T-HMMs are efficient as, once Gabor features are extracted, our non-optimized code compares two face images in less than 15 milliseconds on a 2 GHz Pentium 4 with 512M RAM.

We assume that $\Sigma_{i,j}^{\tau,k} = \Sigma_{i,j}^{k}$, $\delta_{i,j}^{\tau,k} = \delta_{i,j}^{k}$, and $w_{i,j}^{\tau,k} = w_{i,j}^{k}$ to reduce the number of the parameters to estimate. To train single Gaussian mixtures, we first align approximately $\mathscr{F}_T$ and $\mathscr{F}_Q$ and we match each block in $\mathscr{F}_T$ with the corresponding block in $\mathscr{F}_Q$. As for the transition probabilities, they are initialized uniformly. Then Σ's and $a_{i,j}$'s are reestimated. To train multiple Gaussians per mixture, we used an iterative splitting/retraining strategy inspired by the vector quantization algorithm [27, 28].

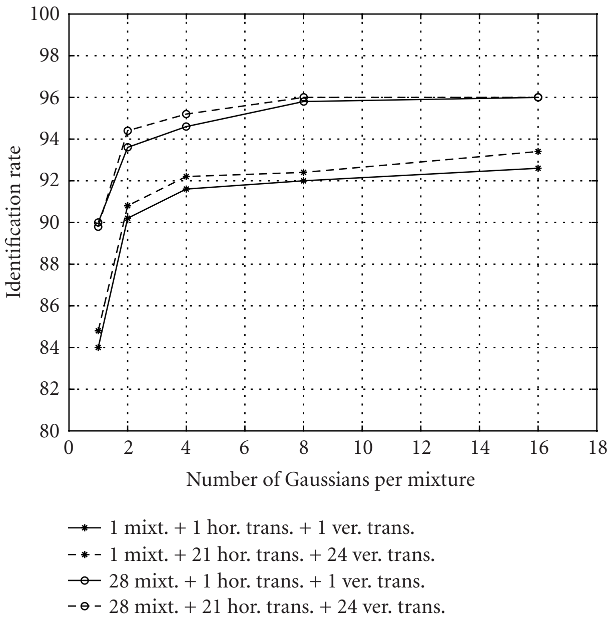


Figure 4: Performance of the proposed algorithm.

We measured the impact of using multiple Gaussian mixtures to weight the different parts of the face and using multiple horizontal and vertical transitions matrices to model the elastic properties of the various parts of the face. In both cases, we used face symmetry to reduce the number of the parameters to estimate. Hence, we tried one mixture for the whole face ($\Sigma_{i,j}^k = \Sigma^k$, $\delta_{i,j}^k = \delta^k$, and $w_{i,j}^k = w^k$) and one mixture for each position (using face symmetry, it resulted in $4 \times 7 = 28$ mixtures). We tried one horizontal and one vertical transition matrices for the whole face and one horizontal and one vertical transition matrices at each position (using face symmetry, it resulted in $3 \times 7 = 21$ horizontal and $4 \times 6 = 24$ vertical transition matrices). This made four test configurations. The performance was drawn on Figure 4 as a function of the number of Gaussians per mixture.

While applying weights to different parts of the face provides a significant increase of the performance, modeling the various elasticity properties of the face had a limited impact and resulted in marginal improvements. The best performance is 96.0% identification rate. We performed a McNemar's test of significance to determine whether the difference in performance between fisherfaces and the proposed approach is statistically significant [29]. Let K be the number of faces on which only one algorithm made an error ($K = 26$) and let M be the number of faces on which the proposed algorithm was correct while fisherfaces made an error ($M = 6$). The probability that the difference in performance between these algorithms would arise by chance is $P = 2\sum_{m=M}^{K}\binom{K}{m}(1/2)^K = 0.009$, which means we are 99% confident that this difference is significant.

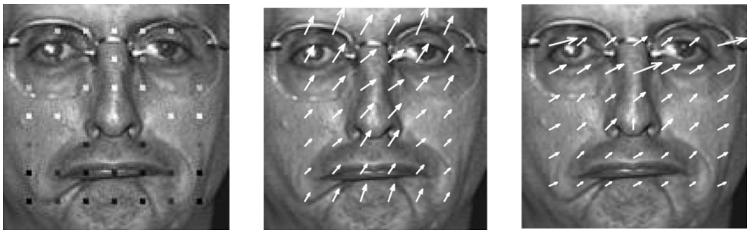
It is also interesting to compare our novel approach to EGM. As stated in Section 4.2, we think that the main advantages of our novel approach are (1) in the use of the well-developed T-HMM framework which provides efficient formulas to compute $P(\mathcal{F}_T|\mathcal{F}_Q, \mathcal{M})$ and to estimate all the parameters of M and (2) in the use of a shared deformable model of the face. Therefore, we will compare the benefits of these two improvements independently. Firstly, we can replace the T-HMM scoring with the SA scoring which is mostly used in the EGM framework. The iterative elastic matching step is generally stopped after a predefined number of iterations N have failed to increase the score. We fixed this figure N so that the amount of computation required by the SA scoring would be similar to the amount of computation required by the T-HMM scoring. We get approximately a 2.0% absolute increase of the performance for our best system with 16 Gaussians per mixture when we use the T-HMM scoring rather than the SA scoring which indicates that the former scoring procedure is more robust. Secondly, if we did not assume a shared transformation model, as we only have one image per person at enrollment time, we would not be able to train one set of parameters per person as is usually done in the EGM framework. Thus, in this case, an upper bound for the performance of EGM is the performance of our system in the simple case where we use one Gaussian mixture for the whole face, with a single Gaussian in the mixture, and where there is, for the whole face, one unique transition probability which is separable and parametric (cf. Section 4.2). The identification rate of such a system is approximately 84.0%, far below the performance of our best system with 16 Gaussians per mixture (cf. Figure 4).

5.5. *Analysis*

Finally, we visualize which parts of the face are the least variable, and thus, considered by our system the most reliable for face recognition (cf. Figure 5a), and which parts are the most elastic (cf. Figures 5b and 5c). The analysis was done on the system with 28 mixtures, 21 horizontal transition probabilities, and 24 vertical transition probabilities. In the case where there is only 1 GpM, $\log|\Sigma_{i,j}^{-1}|$ is a simple measure of local variability: the greater is this value, the fewer variability a face exhibits around position (i, j). It is interesting to note that the upper part of the face exhibits less variability than the lower part and thus, has a higher contribution during identification, which is consistent with other findings [2]. To visualize the elasticity information, we represented the horizontal, respectively, vertical, parametric transition probabilities as vectors $(\sigma_{i,j}^{\mathcal{H}x}, \sigma_{i,j}^{\mathcal{H}y})$, respectively, $(\sigma_{i,j}^{\mathcal{V}x}, \sigma_{i,j}^{\mathcal{V}y})$.

6. FUTURE WORK

A first improvement was suggested in Section 4.1. In our current implementation, we compute the distance between a template image and a query image using a face transformation model. In the case where we have multiple template images for person P, we should combine them into a single face model $\mathcal{M}_p$ (this would require a new formula for the face dependent part of the mean $m_{i,j}^{\tau}$). Hence we should model a transformation between a face model $\mathcal{M}_p$ and a query image $\mathcal{F}_Q$. If λ is the set of parameters of the transformation model, we should then estimate $P(\mathcal{M}_p|\mathcal{F}_Q, \lambda)$.



(a) (b) (c)

Figure 5: (a) The darker a dot, the more variability the corresponding part of the face exhibits, (b) horizontal transition probabilities represented as $(\sigma_{i,j}^{\mathcal{H}x}, \sigma_{i,j}^{\mathcal{H}y})$, and (c) vertical transition probabilities represented as $(\sigma_{i,j}^{\mathcal{V}x}, \sigma_{i,j}^{\mathcal{V}y})$.

A second possible improvement would be to use a discriminative criterion rather than an ML criterion to train the parameters of the face transformation model. If we assume that our HMM models perfectly the face transformation between faces of the same person and if we have infinite training data, then ML estimation can be shown to be optimal. However, as the underlying transformation is not a true HMM and as training data is limited, other training objective functions should be considered. During ML training, pairs of face images corresponding to the same individual were presented to our system and model parameters were adjusted to increase the likelihood of the template images, knowing the query images and the model parameters without taking into account the probability of other possible faces. In contrast to ML estimation, discriminative approaches such as minimum classification error (MCE) [30, 31] or maximum mutual information estimation (MMIE) [32, 33] would consider competing faces to reduce the probability of misclassification.

Although we have only presented face identification results, we should consider the extension of this work to face verification. While the first idea would be simply to threshold the score $(P(\mathcal{F}_Q|\mathcal{M}_p,\lambda) > \theta)$, this approach is known to lack robustness when there is a mismatch between training and test conditions [34]. Generally, a likelihood normalization of the following form has to be performed:

$$\frac{P(\mathcal{F}_Q \mid \mathcal{M}_p,\lambda)}{P(\mathcal{F}_Q \mid \mathcal{M}_{\bar{p}},\lambda)} > \theta, \tag{30}$$

where $\mathcal{M}_{\bar{p}}$ is an antiface model for individual P and $P(\mathcal{F}_Q|\mathcal{M}_{\bar{p}},\lambda)$ is the likelihood that $\mathcal{F}_Q$ belongs to an impostor. Two types of antimodels are generally used: background model set (BMS), where the set of background model for each client is selected from a pool of impostor models, and universal background model (UBM), where a unique background model is trained using all the impostor data [34, 35]. While the latter approach usually outperforms the first one, both score normalization methods should be tested on our novel approach.

While we showed that our system could model with great accuracy facial expressions with local geometric transformations, it is clear that geometric transformations cannot grab certain types of variability such as illumination variations which are known to greatly affect the performance of a face recognition system. In our system, small variations in illumination are compensated by Gabor features and the feature normalization step (cf. Section 5.2). However Gabor features, even combined with feature normalization, cannot fully compensate for large variations in illumination due, for instance, to the location of the light source. Hence, the idea would be to use *feature transformations* as suggested in Section 2.2. Our model of face transformation would thus not only compensate for variations due to facial expressions but also for changes in illumination conditions.

Finally, although our novel approach was tested on a face recognition task, we would like to outline that it was designed for the more general problem of *content-based image retrieval* and it has the potential to be extended to other biometrics such as fingerprint recognition.

7. SUMMARY

We presented a general novel approach for content based image retrieval and successfully specialized it to face recognition. In our framework, the stochastic source of the pattern classification system, which is a 2D HMM, does not directly model faces but a transformation between faces of the same person. We also introduced a new framework for approximating the computationally intractable 2D HMMs using turbo-HMMs (T-HMMs). T-HMMs are another major contribution of this paper and one of the keys of the success of our approach. We compared conceptually the proposed approach to two different face recognition algorithms. We presented experimental results showing that our novel algorithm significantly outperforms two popular face recognition algorithms: eigenfaces and fisherfaces. Also, a preliminary comparison of our probabilistic model of face transformation with the EGM approach showed great promise. However, to draw more general conclusions on the relative performance of approaches which model a face (such as eigenfaces and fisherfaces) and approaches which model the relation between face images (such as EGM and our novel approach), we would not only have to carry out more experiments but also to consider other algorithms for both classes of pattern classification methods.

ACKNOWLEDGMENTS

This work was supported in part by France Telecom, by the National Science Foundation (NSF) under Grants EIA-9986057 and EIA-0080134, and by the University of California MICRO program, Dolby Laboratories, Inc., Lucent Technologies, Inc., Mindspeed Technologies, and Qualcomm, Inc.

REFERENCES

[1] J. Schürmann, *Pattern Classification: A Unified View of Statistical and Neural Approaches*, John Wiley & Sons, NY, USA, 1996.

[2] R. Chellappa, C. Wilson, and S. Sirohey, "Human and machine recognition of faces: A survey," *Proceedings of the IEEE*, vol. 83, no. 5, pp. 705–740, 1995.

[3] W. Zhao, R. Chellappa, A. Rosenfeld, and P. J. Phillips, "Face recognition: A literature survey," Tech. Rep. CAR-TR948, University of Maryland, 2000.

[4] M. Vissac, J.-L. Dugelay, and K. Rose, "A novel indexing approach for multimedia image databases," in *Proc. IEEE Workshop on Multimedia Signal Processing*, pp. 97–102, Copenhagen, Denmark, September 1999.

[5] F. S. Samaria, *Face recognition using hidden Markov models*, Ph.D. thesis, University of Cambridge, Cambridge, UK, 1994.

[6] A. Nefian, *A hidden Markov model-based approach for face detection and recognition*, Ph.D. thesis, Georgia Institute of Technology, Atlanta, Ga, USA, 1999.

[7] M. Lades, J. Vorbrüggen, J. Buhmann, et al., "Distortion invariant object recognition in the dynamic link architecture," *IEEE Trans. on Computers*, vol. 42, no. 3, pp. 300–311, 1993.

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

[9] K. Abend, T. J. Harley, and L. N. Kanal, "Classification of binary random patterns," *IEEE Transactions on Information Theory*, vol. 11, no. 4, pp. 538–544, 1965.

[10] S.-S. Kuo and O. Agazzi, "Keyword spotting in poorly printed documents using pseudo 2-D hidden Markov models," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 16, no. 8, pp. 842–848, 1994.

[11] J. Li, A. Najmi, and R. M. Gray, "Image classification by a two-dimensional hidden Markov model," *IEEE Trans. Signal Processing*, vol. 48, no. 2, pp. 517–533, 2000.

[12] C. Miller, B. R. Hunt, M. A. Neifeld, and M. W. Marcellin, "Binary image reconstruction via 2-D Viterbi search," in *Proc. International Conference on Image Processing*, vol. 1, pp. 181–184, Washington, DC, USA, October 1997.

[13] T. A. Tokuyasu, *Turbo recognition: an approach to decoding page layout*, Ph.D. thesis, University of California, Berkeley, Calif, USA, 2001.

[14] F. Perronnin, J.-L. Dugelay, and K. Rose, "Iterative decoding of two-dimensional hidden Markov models," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, vol. 3, pp. 329–332, Hong Kong, April 2003.

[15] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[16] L. E. Baum, T. Petrie, G. Soules, and N. Weiss, "A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains," *Annals of Mathematical Statistics*, vol. 41, no. 1, pp. 164–171, 1970.

[17] J. Zhang, Y. Yan, and M. Lades, "Face recognition: Eigenface, elastic matching, and neural nets," *Proceedings of the IEEE*, vol. 85, no. 9, pp. 1423–1435, 1997.

[18] L. Wiskott, J. M. Fellous, N. Krüger, and C. von der Malsburg, "Face recognition by elastic bunch graph matching," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 775–779, 1997.

[19] B. Duc, S. Fischer, and J. Bigün, "Face authentication with Gabor information on deformable graphs," *IEEE Trans. Image Processing*, vol. 8, no. 4, pp. 504–516, 1999.

[20] A. Tefas, C. Kotropoulos, and I. Pitas, "Using support vector machines to enhance the performance of elastic graph matching for frontal face authentication," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 23, no. 7, pp. 735–746, 2001.

[21] C. Liu and H. Wechsler, "Gabor feature based classification using the enhanced fisher linear discriminant model for face recognition," *IEEE Trans. Image Processing*, vol. 11, no. 4, pp. 467–476, 2002.

[22] G. Donato, M. S. Bartlett, J. C. Hager, P. Ekman, and T. J. Sejnowski, "Classifying facial actions," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 21, no. 10, pp. 974–989, 1999.

[23] M. Kirby and L. Sirovich, "Application of the Karhunen-Loève procedure for the characterization of human faces," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 12, no. 1, pp. 103–108, 1990.

[24] I. T. Joliffe, *Principal Component Analysis*, Springer-Verlag, NY, USA, 1986.

[25] M. A. Turk and A. P. Pentland, "Face recognition using eigenfaces," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp. 586–591, Maui, Hawaii, USA, June 1991.

[26] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. fisherfaces: Recognition using class specific linear projection," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 711–720, 1997.

[27] "HTK, hidden Markov model toolkit," http://htk.eng.cam.ac.uk/.

[28] Y. Linde, A. Buzo, and R. Gray, "An algorithm for vector quantizer design," *IEEE Trans. Communications*, vol. 28, no. 1, pp. 84–95, 1980.

[29] L. Gillick and S. J. Cox, "Some statistical issues in the comparison of speech recognition," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, pp. 532–535, Glasgow, Scotland, UK, May 1989.

[30] A. Ljolje, Y. Ephraim, and L. R. Rabiner, "Estimation of hidden Markov model parameters by minimizing empirical error rate," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, vol. 2, pp. 709–712, Albuquerque, NM, USA, April 1990.

[31] B.-H. Juang and S. Katagiri, "Discriminative learning for minimum error classification," *IEEE Trans. Signal Processing*, vol. 40, no. 12, pp. 3043–3054, 1992.

[32] L. R. Bahl, P. F. Brown, P. V. de Souza, and R. L. Mercer, "Maximum mutual information estimation of hidden Markov model parameters for speech recognition," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, pp. 49–52, Tokyo, Japan, April 1986.

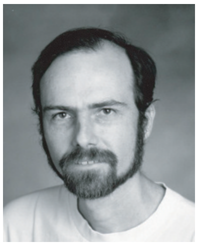
[33] Y. Normandin, *Hidden Markov models, maximum mutual information estimation and the speech recognition problem*, Ph.D. thesis, McGill University, Montreal, Canada, 1991.

[34] C. Sanderson and K. K. Paliwal, "Likelihood normalization for face authentication in variable recording conditions," in *Proc. International Conference on Image Processing*, vol. 1, pp. 301–304, Rochester, NY, USA, September 2002.

[35] D. Reynolds, "Comparison of background normalization methods for text independent speaker verification," in *Proc. 5th European Conference on Speech Communication and Technology (Eurospeech '97)*, vol. 2, pp. 963–966, Rhodes, Greece, September 1997.

Florent Perronnin received his Engineering degree in 2000 from the Ecole Nationale Supérieure des Télécommunications, Paris, France. From January 2000 to October 2001, he was with the Panasonic Speech Technology Laboratory, Santa Barbara, California, first as an Intern and then as a Research Engineer, working on speech and speaker recognition. In November 2001, he joined the Multimedia Communications Department, Institut Eurécom, Sophia Antipolis, France, where he is currently pursuing his Ph.D. degree. His research focuses on pattern recognition and, more specifically, on biometrics person authentication.

Jean-Luc Dugelay received his Ph.D. degree in computer science from the University of Rennes in 1992. He joined the Eurecom Institute, Sophia Antipolis, in 1992, where he is a Professor in charge of image and video research and teaching activities in the Department of Multimedia Communications. His research interests are in the area of multimedia signal processing and communications including security imaging (i.e., watermarking and biometrics), image/video coding, facial image analysis, virtual imaging, face cloning, and talking heads. He is an author or coauthor of more than 65 publications that have appeared as journal papers or proceeding articles, 3 book chapters, and 3 international patents. He gave several tutorials on digital watermarking (coauthored with F. Petitcolas from Microsoft Research, Cambridge) at major conferences, and invited talks on Biometrics. He was Technical Cochair and Organizer of the Fourth Workshop on Multimedia Signal Processing, Cannes, October 2001. He was Coorganizer and Session Chair of the special session on "Multimodal Person Authentication" (ICASSP 2002, May 13–17, Orlando). His group is involved in several national and European projects related to digital watermarking and biometrics. Jean-Luc Dugelay is a Senior Member of the IEEE Signal Processing Society. He is currently an Associate Editor for the IEEE Transactions on Multimedia and the IEEE Transactions on Image Processing.



Kenneth Rose received the Ph.D. degree in electrical engineering from Caltech in 1991. He then joined the Department of Electrical and Computer Engineering, University of California at Santa Barbara, where he is currently a Professor. His research activities are in the areas of information theory, signal compression, source-channel coding, image/video coding and processing, pattern recognition, content-based search and retrieval, and nonconvex optimization. He is particularly interested in application of information and estimation theoretic approaches to fundamental problems in signal processing. His optimization algorithms have been adopted by others in numerous disciplines besides electrical engineering and computer science, including physics, chemistry, biology, medicine, materials, astronomy, geology, psychology, linguistics, ecology, and economics. Dr. Rose was Technical Program Cochair of the 2001 IEEE Workshop on Multimedia Signal Processing, and currently serves as Area Editor for the IEEE Transactions on Communications. In 1990, he received (with A. Heiman) the William R. Bennett Prize-Paper Award from the IEEE Communications Society. He is a Fellow of the IEEE.