We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900
Open access books available

185,000
International authors and editors

200M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



1

Automatic Face Recognition System for Hidden Markov Model Techniques

Peter M. Corcoran and Claudia Iancu
College of Engineering & Informatics,
National University of Ireland Galway,
Ireland

1. Introduction

Hidden Markov Models (HMMs) are a class of statistical models used to characterize the observable properties of a signal. HMMs consist of two interrelated processes: (i) an underlying, unobservable Markov chain with a finite number of states governed by a state transition probability matrix and an initial state probability distribution, and (ii) a set of observations, defined by the observation density functions associated with each state.
In this chapter we begin by describing the generalized architecture of an automatic face recognition (AFR) system. Then the role of each functional block within this architecture is discussed. A detailed description of the methods we used to solve the role of each block is given with particular emphasis on how our HMM functions. A core element of this chapter is the practical realization of our face recognition algorithm, derived from EHMM techniques. Experimental results are provided illustrating optimal data and model configurations. This background information should prove helpful to other researchers who wish to explore the potential of HMM based approaches to 2D face and object recognition.

2. Face recognition systems

In this section we outline the basic architecture of a face recognition system based on Gonzalez's image analysis system [Gonzalez & Woods 1992] and Costache's face recognition system [Costache 2007]. At a top-level this is represented by the functional blocks shown in Figure 1.

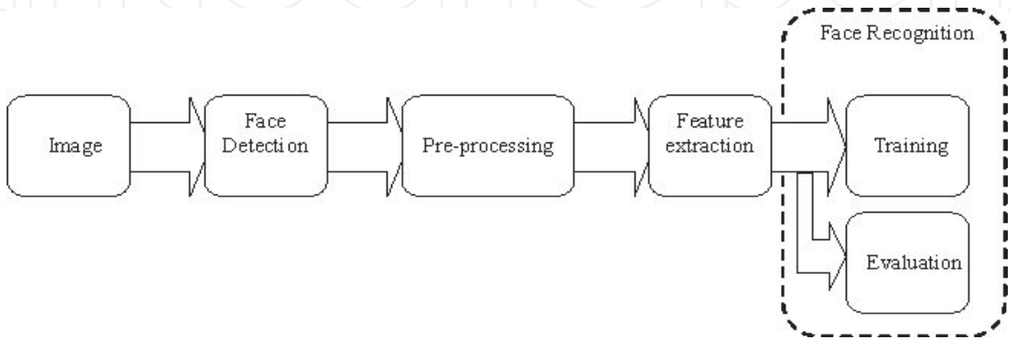


Fig. 1. The architecture of a face recognition system

1. Face detection and cropping block: this is the first stage of any face recognition system and the key difference between a semi-automatic and a fully automatic face recognizer. In order to make the recognition system fully automatic, the detection and extraction of faces from an image should also be automatic. Face detection also represents a very important step before face recognition, because the accuracy of the recognition process is a direct function of the accuracy of the detection process [Rentzeperis *et. al.* 2006, Corcoran *et. al.* 2006].
2. Pre-processing block: the face image can be treated with a series of pre-processing techniques to minimize the effect of factors that can adversely influence the face recognition algorithm. The most critical of these are *facial pose* and *illumination*. A discussion on these factors and their significance w.r.t. HMM techniques is given in Section 3.
3. Feature extraction block: in this step the features used in the recognition phase are computed. These features vary depending on the automatic face recognition system used. For example, the first and most simplistic features used in face recognition were the geometrical relations and distances between important points in a face, and the recognition 'algorithm' matched these distances [Chellappa *et. al.* 1992]; the most widely used features in face recognition are KL or eigenfaces, and the standard recognition 'algorithm' uses either the Euclidian or Mahalanobis distance [Chellappa *et. al.* 1992, 1995] to match features. Our features and the extraction method used are described in Section 4.
4. Face recognition block: this consists of 2 separate stages: a *training process*, where the algorithm is fed samples of the subjects to be learned and a distinct model for each subject is determined; and an *evaluation process* where a model of a newly acquired test subject is compared against all exisiting models in the database and the most closely corresponding model is determined. If these are sufficiently close a recognition event is triggered.

3. Face detection and cropping

As mentioned in the previous section, face detection is one of the most important steps in a face recognition system and differentiates between semi-automatic and fully automatic face recognizer. The goal of an automatic face detector is to search for human faces in a still image and, if found, to accurately return their locations. In order to make the detection fully automatic the system has to work without input from the user. Many attempts to solve the problem of face detection exist in the literature beginning with the basic approach of [Kanade 1977] and culminating with the method of [Viola & Jones 2000, 2001]. Comprehensive surveys of face detection techniques can be found in [Yang *et. al.* 2002] and [Costache 2007]. In this section we underline the main challenges an automatic face detector has to tackle, and we briefly describe the face detector used in our experiments.
Face detection methods were classified by [Yang *et. al.* 2002] into four principle categories: (i) knowledge-based, (ii) feature invariant, (iii) template matching and (iv) appearance-based methods. According to [Gonzalez & Woods 1992], the main disadvantage presented by the majority of these methods is the time required to detect all the faces in an image. State-of-the-art face detection methods provide real-time solutions. The best known of these methods, and the gold standard for face detection was originally proposed by [Viola & Jones 2001]. The original algorithm was, according to its authors, 15 times faster than any previous approach. The algorithm has been well proved in recent years as being one of the fastest and most accurate face detection algorithms reported and is presently the gold standard against which other face detection techniques are benchmarked. For these reasons we adopted it to implement our face detection subsystem.

Our implementation of the Viola & Jones detection algorithm is provided in the Intel digital image processing C++ library [OpenCV 2006]. This can be used both for face detection and subsequent cropping of confirmed facial images. The OpenCV face detector has been pre-trained using a very comprehensive database of face/non-face examples and is widely used in the literature.

4. Pre-processing techniques

Automatic face detection is influenced by a number of key factors [Costache 2007]: facial *orientation* or *pose*: the appearance of the face varies due to relative camera-face pose, between full frontal images and side-profile images; *in-situ occlusions* such as facial hair (e.g. beard, moustache), eye-glasses and make-up; facial *expressions* can significantly influence the appearance of a face image; *overlapping occlusions* where faces are partially occluded by other faces present in the picture or by objects such as hats, or fans; *conditions of image acquisition* where the quality of the picture, camera characteristics and in particular the *illumination conditions* can strongly influence the appearance of a face.
For our system to perform better in the recognition stage, we apply a set of pre-processing techniques: the first step in pre-processing is to bring all images into the same color space and to normalize the size of face regions. This normalization process is critical to improving the final face recognition rate and we will later present some experimental results for our HMM-specific AFR.

4.1 Color to grayscale conversion

In most face recognition applications the images are single or multiple views of 2D intensity data [Zhao *et. al.* 2003], and many databases built for face recognition applications are available as grayscale images. From the four databases used in our experiments, 3 contained grayscale images (BioID, Achermann, UMIST) and one contained RGB images (FERET). Practical images will, naturally, be acquired in color as modern image acquisition systems are practically all color and so we need to convert from color to grayscale, or intensity images of the selected face regions. In practice the intensity data may be available from the imaging system – many camera system employ YCC data internally and the Y component can be utilized directly. In other cases we may need to perform an explicit conversion of RGB data. Here a set of red, green and blue integer values characterize an image pixel. The effective luminance, Y of each pixel is calculated with the following formula [Pratt 1991]:

$$Y = 0.3 \times \text{Red} + 0.59 \times \text{Green} + 0.11 \times \text{Blue} \quad (1)$$

4.2 Image resizing

For a HMM-based face recognition system having a consistently sized face region is particularly important because the HMM requires regional analysis of the face with a scanning window of fixed size. A straightforward approach is to resize all determined face regions to a common size. To facilitate more efficient computation we seek the smallest sized face region possible without impacting the overall system recogntion rate. Some empirical data will be presented later to illustrate how different factors, including the size of normalized face regions, affect recognition rate.

There are many techniques that can be used to enlarge or reduce the size of an image. These methods generally realize a trade-off between speed and the degree to which they reduce the occurrence of visual artifacts in the resulting image. The most commonly used resize method is called bicubic interpolation and has the advantage that the interpolated image is smoother than images obtained using simpler interpolation techniques and has fewer artifacts [Lehmann *et. al.* 1999]. In our work we have used bicubic spline interpolation using bicubic polynomials. More details of how to calculate bicubic spline interpolation functions can be found in [Hummel 1977].

4.3 Illumination normalization

One of the most important factors that influence the recognition rate of a system is illumination variation. In was shown in [Adini *et al.* 1997, Gokemen *et al.* 2007] that variations in illumination can be more relevant than variations between individual characteristics. Such variations can induce an AFR system to decide that two different individuals with the same illumination characteristics are more similar than two instances of the same individual taken in different lighting conditions. Thus normalizing illumination conditions across detected face regions is crucial to obtaining accurate, reliable and repeatable results from an AFR. One approach suitable for face models which combine both facial geometry and facial texture such as active appearance models (AAM) is described by [Ionita 2008]. However as HMM techniques do not explicitly rely on facial geometry or textures it is not possible to integrate the illumination normalization within the structure of the model itself. Instead we must rely on a discrete illumination normalization process. Fortunately most AFR systems employ a similar prefiltering stage and we can draw on a wide range of techniques from the literature.

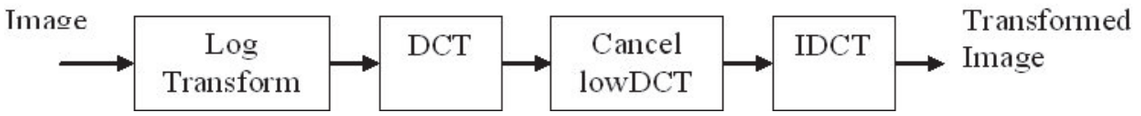


Fig. 2. Block scheme of logDCT algorithm

Algorithms used for performing the normalization vary from a simple histogram equalization (HE) to more complex techniques such as albedo maps [Smith & Hancock 2005] and contrast limited adaptive histogram equalization (CLAHE) [Zuiderveld 1994, Pizer *et al* 2006, Corcoran *et al* 2006]. These algorithms perform well when the variations in illumination are small but there is no commonly adopted method for illumination normalization in images which performs well for every type of illumination. Some tests have been conducted to determine the robustness of face recognition algorithms to changes in lighting [Phillips *et al* 2000, O'Toole *et al* 2007]. Also, numerous illumination normalization techniques have been developed. Some of the more widely used of these - *histogram equalization*, *histogram specification* and *logarithm transformation* - have been compared in [Du *et al* 2005] with more recently proposed methods, *gamma intensity* correction and *self-quotient image*. The results are interesting: both HE and logarithmic transform improved recognition rates over face regions that were not normalized, compared favorably to the other techniques.

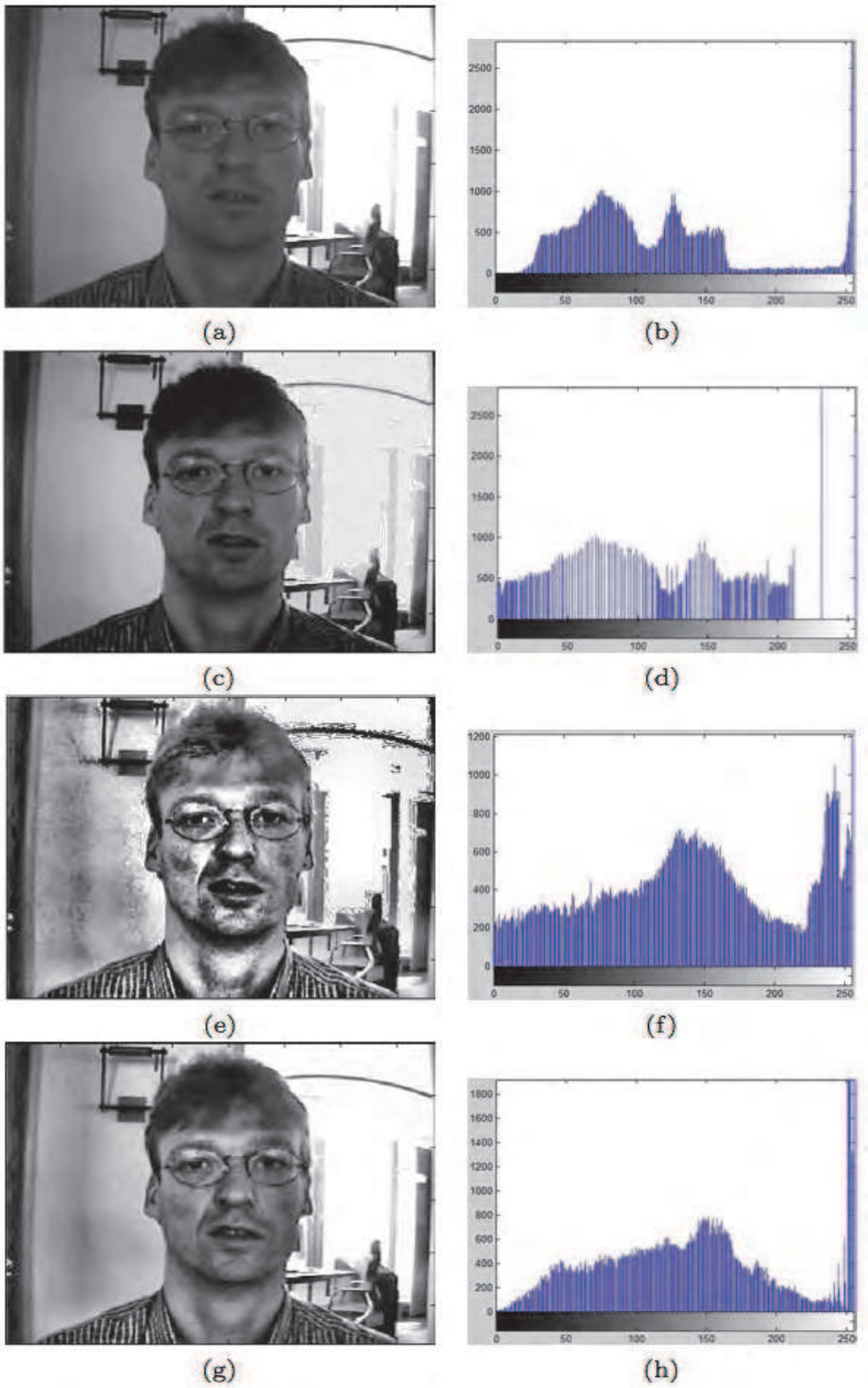


Fig. 3. Examples of illumination normalization techniques – details in the text.

To tackle the problem of illumination variations we implemented the following three illumination normalization algorithms: (i) *histogram equalization* (HE) based on [Gonzalez & Woods 1992], (ii) *contrast limited adaptive histogram equalization* (CLAHE) based on [Zuiderveld 1994], and (iii) the relative new method of DCT in the logarithm domain - *logDCT* based on [Chen *et al* 2006]. In figure 3 above we show some examples of a face image processed by different normalization algorithms: (a) shows the unprocessed image with (b) the original luminance histrogram; (c) is the same image normalized with simple HE and (d) the effect of HE on the image histogram; (e) is the image with *adaptive* HE applied and (f) the effect of AHE on the histogram, in particular note the high frequency blow-up of the histogram; finally (g) shows how CLAHE eliminates the high-frequency artifacts of AHE and (h) reduces the high-frequency blow-up when compared with (f).

5. Feature extraction

Feature extraction for both 1D and 2D HMMs was originally described by [Samaria 1994]. His method was subsequently adopted in the majority of HMM-based face recognition papers. This feature extraction technique is based on scanning the image with a fixed-size window from left-to-right and top-to-bottom. A window of dimensions $h \times w$ pixels begins scanning each extracted face region from the left top corner sub-dividing the image into a set number of $h \times w$ sized blocks.

On each of these blocks a transformation is applied to extract the characterizing features which represent the observation vector for that particular region. Then the scanning window moves towards right with a step-size of n pixels allowing an overlap of o pixels, where o = w − n. Again features are extracted from the new block. The process continues until the scanning window reaches the right margin of the image. When the scanning window reaches the right margin for the first row of scanned blocks, it moves back to the left margin and down with m pixels allowing an overlap of v pixels vertically. The horizontal scanning process is resumed and a second row of blocks results, and from each of these blocks an observation vector is extracted. The scanning process and extraction of blocks is depicted in Figure 4.



Fig. 4. Blocks extraction from a face image

In our work we have used two types of features to describe the images: 2D DCT coefficients and Daubechies wavelets.

5.1 Overview of features used with HMM in face recognition

The first features used in face recognition performed with HMM were pixel intensities [Samaria & Fallside 1993, Samaria 1994, Samaria & Harter 1994]. The recognition rates obtained by Samaria using pixel intensities with a P2D-HMM were up to 94.5% on the ORL database. However the use of pixel intensities as features has some disadvantages [Nefian & Hayes 1999]: firstly they cannot be regarded as robust features since: (i) the intensity of a pixel is very sensitive to the presence of noise in the image or to illumination changes; (ii) the use of all the pixels in the image is computationally complex and time consuming; and (iii) using all image pixels does not eliminate any redundant information and is thus a very inefficient form of feature. Another example of features used with EHMM for face recognition are KLT features used by [Nefian & Hayes 1998, Nefian & Hayes 2000] with recognition rates of up to 98% on ORL database. The main advantage of using KLT features instead of pixel intensities is their capacity to reduce redundant information in an image. The disadvantage is their dependence of the database of training images from which they are derived [Costache 2009].

The most widely used features for HMM in face recognition are 2D-DCT coefficients. These DCT coefficients combine excellent decorrelation properties with energy compaction. Indeed, the more correlated the image is, the more energy compaction increases. Thus a relatively small number of DCT coefficients contain the majority of information encapsulated in an image. A second advantage is the speed with which they can be computed since the basis vectors are independent of the database and are often pre-computed and stored in an imaging device as part of the JPEG image compression standard. Recognition rates obtained when using 2D DCT with HMM can achieve 100% success on smaller databases such as ORL. In our research we also introduce the use of Daubechies wavelets. Apart from the work of [Le & Li 2004] wavelets have not been previously used with HMMs for face recognition applications.

6. Face recogntion

In the earlier sections of this chapter we have described the main pre-filtering blocks for our AFR system. We nest focus on the actual HMM itself and the various operational processes required to implement the training and recognition phases of our AFR.

6.1 Background to embedded hidden markov models in face recognition

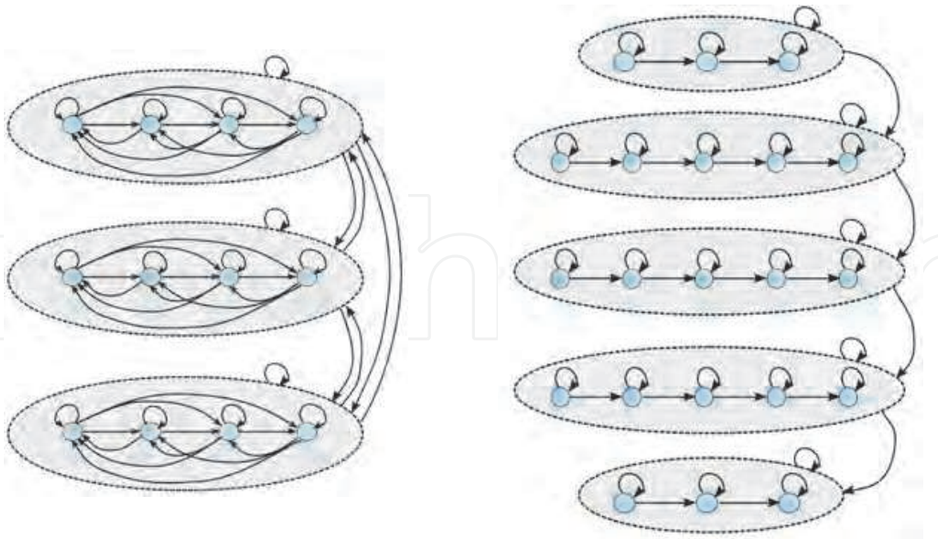
After their introduction in the late 60's by [Baum *et al* 1966, 1970] and the more detailed description in the late 80's by [Raibner & Juang 1986, Rabiner 1989] HMMs have been widely used in speech recognition applications. In this field of application very high recognition rates are obtained due to the specific capacity of HMM to cope with variations in the timing and duration human speech patterns [Juang and Rabiner 2005]. HMMs have also been used successfully in other applications such as OCR and handwriting recognition. Thus it was no surprise that researchers began to consider their use for problems such as face recognition where adaptability of HMMs might offer solutions to some of the underlying problems of accurately recognizing a 2D face region.

Note that the application of HMM techniques to the face recognition problem implies the use of an inherently 1D method of pattern matching to solve an inherently 2D problem. So why did researchers think this might work? Well, as the most significant facial features of a frontal face image occur in a natural order, from top to bottom, and this sequence is

immutable, even if the face is moderately rotated. The first attempts to use HMMs for face recognition and detection were made by [Samaria & Fallside 1993, Samaria & Harter 1994] who used a left-to-right HMM and divided the face in a fixed succession of regions (observantion states) such as eyes, nose, & mouth. This early work by Samaria was essentially 1D in scope and the first attempt to implement a more appropriate 2D models was *Pseudo 2D HMM*, introduced by [Kuo & Agazzi 1994] for character recognition, subsequently adapted by [Samaria 1994] for the face recognition problem. The idea was later taken forward and improved by [Nefian & Hayes 1999, 2000]. These researchers changed the name to *Embedded HMM* (EHMM).
There have been several alternative 2D versions of HMM used for face recognition in the literature. However EHMM has become the standard method employed by researchers working in the field of HMM face recognition. As a result this algorithm has been implemented in the Intel digital image processing C++ library [OpenCV 2006] which was also employed to implement our face detector, described in section 2 above.

6.2 An overview of EHMMs

The embedded HMM is a generalization of the classic HMM, where each state in the one dimensional HMM is itself an HMM. This enables a generalization of the 1D HMM techniques to a second dimension while simplifying the dependencies and transitions between states. Thus, an embedded HMM consists of a set of super states each of which envelopes a set of embedded states. The super states model the two dimensional data in a first dimension, while the embedded HMMs model the data in the other dimension.
The structure of an EHMM with 3 superstates and 4 embedded states is shown in Figure 5(a). This EHMM is unrestricted, meaning that transitions between all the states in the embedded HMMs and between all the superstates are allowed.



(a): An unrestricted EHMM (b): Restricted EHMM for face recognition

Fig. 5. EHMM for face recognition

The elements of an embedded HMM are:

- A set of N_0 superstates $S_0 = S_{0,1}$, $1 \le i \le N_0$

- The initial probabilities of the super states $\Pi_0 = \pi 0,i$ where $\pi 0,i$ is the probability of being in superstate i at time zero.
- The transition probability matrix $A_0 = a_{0,ij}$, where $a_{0,ij}$ is the probability of transitioning from super state i to superstate j.
- The parameters of the embedded HMM for the superstate k, $1 \le k \le N_0$ and where $\Lambda^k = (\Pi_1^k, A_1^k, B^k)$ which includes: (i) the number of embedded states in the k^{th} super state, N_1^k, and the set of embedded states, $S_1^k = S_{1,i}^k$ with $1 \le i \le N_1^k$; (ii) the initial state distribution, $\Pi_1^k = \pi_{1,i}^k$, where $\pi_{1,i}^k$ is the probability of being in state i of the superstate k at time zero; (iii) the state transition probability matrix $A_1^k = a_{1,ij}^k$, where $a_{1,ij}^k$ is the transition probability from state i to state j; (iv) the probability distribution matrix of the observations, B^k; these observations are characterized by a set of continuous probability density functions, considered finite *Gaussian mixtures* of the form:

$$b_i^k(O_{t0,t1}) = \sum_{m=1}^{M_i^k} c_{im}^k N(O_{t0,t1}, \mu_{im}^k, U_{im}^k) \tag{2}$$

where c_{im}^k is the mixture coefficient for the m^{th} mixture in state i of super state k, and $N(O_{t0,t1}, \mu_{im}^k, U_{im}^k)$ is a Gaussian density with a mean vector μ_{im}^k and covariance matrix U_{im}^k.

In shorthand notation, an embedded HMM is defined as the triplet $\lambda = (\Pi_0, A_0, \Lambda)$ where $\Lambda = \Lambda^1, \Lambda^2, ..., \Lambda^{N_0}$. This model is appropriate for facial images since it exploits an important characteristic of these: frontal faces preserve the structure of "superstates" from top to bottom, and also the left-to-right structure of 'embedded states' within each "superstate" [Nefian & Hayes 1999, 2000]. An example of the state structure of the face model and the non-zero transition probabilities of the embedded HMM are shown in Figure 5(b). The configuration presented is 5 super states, each with 3, 6, 6, 6, 3 states respectively. Each state in the overall top-to-bottom HMM is assigned to a left-to-right 1D HMM. In this case transitions are allowed only from left-to-right or self-transitions and only between *consecutive states* both for the embedded HMMs within each superstate, and for the main superstates of the top-level HMM as well.

6.3 The training process for an EHMMs

The training of HMM, as shown by [Rabiner 1989] is accomplished using the Baum-Welch algorithm. While EHMM exhibits a more complex structure than the simple 1D HMM, the training algorithm follows the same steps. The main difference in training is the use of a doubly embedded Viterbi for segmentation. The training structure is depicted in Figure 6, and the role of each block is described next:

Step 1. PrototypeHMM: the first step is defining the prototype EHMM: *parameters:* N_i^k the numbers of states, N_0 the number of superstates , K the number of Gaussians used to model the probability density for the observation vectors in each state of an embedded HMM; *conditions:* which transitions are allowed ($a_{1,ij}^k > 0$) and which are not ($a_{1,ij}^k = 0$); in our left-to-right HMM the only transitions allowed are self-transitions and transitions to the next state, so the probability of transition to previous states is 0. For a numerical example we choose $N_0 = 5$, $N_1^k = 3,6,6,6,3$ where $k = 1, 2, ..., 5$, and $K = 3$.

Step 2. Uniform segmentation: the image is uniformly segmented. Firstly the observation vectors extracted from the entire image are uniformly divided into $N_0 = 5$ vertical super states, or image strips, for the overall top-to-bottom HMM. Next the data corresponding to each of these vertical super states is now horizontally segmented from left to right into N_i^k uniform states. For a 128 × 128 pixel facial region with scanning window 12 × 12 with 8 pixels overlap we obtain 30 observation vectors per scanning row both horizontally and vertically, thus 900 observation vectors in total. In a uniform segmentation, the observation vectors are first divided between $N_0 = 5$ superstates: 30 observation vectors per row/5 superstates, so 6 observation vectors per row in each superstate $\Rightarrow$ 6 × 30 = 180 observation vectors per superstate. Then horizontally these 180 observation vectors are uniformly divided in states as follows: for the superstates 1 and 5 with only 3 states: there will be 60 observation vectors per state; for superstates 2, 3, 4 with 6 states each: there will be 30 observation vectors per state.

Step 3. Parameter initialization: after segmentation, the initial estimates of the model parameters are obtained using the concept of counting event occurrences for the initial probabilities of the states and the transition probabilities. In order to compute the observation probabilities, for each state of the embedded HMMs a K-means clustering algorithm, where *K* is the number of Gaussians per state, is applied. All the observation vectors extracted from each state are used to obtain a corresponding mixture of Gaussians describing the observation probability density function. From these initial values we then begin to iterate. In the example given above the initial probabilities of the states in each superstate are determined from the system constraints as follows: first state in each embedded HMM has initial probability equal to 1.0, all the other states have initial probability of zero. Transition probabilities are then obtained by counting transition occurrences. For example in the first state of the first superstate: there are 60 observation vectors distributed across 6 horizontal rows of scanning implying 6 possibilities of transition from state 1 into state 2: probability of transition from state 1 into state 1 is $P_{1,1} = 54/60$; probability of transition from state 1 into state 2 is $P_{1,2} = 6/60$; transition probabilities for the other states can be calculated in the same way.

Step 4. Embedded Viterbi segmentation: in the first step of the iteration, a doubly embedded Viterbi algorithm replaces the uniform segmentation. With the new segmentation and again counting event occurrences, a set of new values for initial and transition probabilities are found. This process is described in detail in section 5.4 below.

Step 5. Segmental K-means: according to the new segmentation performed at step 4, another K-means is applied to the current set of observation vectors corresponding to each new state, and new observation probability density functions are computed. On the next iteration, these new values are introduced into the doubly embedded Viterbi and a new segmentation is initiated.

Step 6. Convergence: Steps 4 and 5 are repeated until the difference on consecutive iterations is below a set threshold. If convergence is not achieved after a certain number of iterations the training is considered to have failed for the current input face region. Typically we have set the convergence threshold at 0.01 and the maximum number of iterations at 40. Once convergence is achieved, further iterations are stopped and the EHMM is output and stored in a reference database.

6.4 The decoding process for an EHMM (Doubly embedded Viterbi)

In the description of the training process above we have seen that step 4 consists in the re-segmentation of the states in the 1D HMMs and of the superstates in the overall HMM. Re-segmentation means finding the most probable sequence of states given a certain sequence of observation vectors and we can solve this problem by applying the Viterbi algorithm. We can easily apply Viterbi algorithm in the embedded 1D HMMs for which we have determined all the probabilities at step 3 above. However for the overall HMM after step 3 we only have the initial and transition probabilities, without the observations probabilities.

In order to solve this problem a method based on the Viterbi algorithm known as *double embedded Viterbi* was developed [Kuo & Agazzi 1994]. It involves applying the Viterbi algorithm to both the embedded HMMs and to the global, or top-level HMM, hence the name. A detailed description may be found in [Nefian 1999]. As the algorithm is mathematically complex and the formulas are challenging to understand and even more so to implement. For this reason we will next provide a detailed practical (as opposed to theoretical) description of our step by step implementation of the algorithm. The underlying concept is illustrated in Figure 6 and comprises the following steps:

Step 1. After the parameters initialization step No. 3 of the previous section we have: initial probabilities, transition probabilities and observation probabilities for each embedded HMM, and initial and transition probabilities for the top-level HMM. In the first step of the double Viterbi, each scanned row of observation vectors $O_0^i = v_j^i$ with $1 \le i \le (H-v)/(h-v)$ and $1 \le j \le (W-o)/(w-o)$ within each of the embedded 1D HMMs has the conventional Viterbi algorithm applied. After this step the optimal state distribution is obtained for each row of observation vectors based on the relevant 1D HMM $(Q^i \left| Q_0^i, \Lambda_k \right.)$ and also the probability of each row of observation for the top-level HMM, or superstate $P(O_0^i, Q^i \left| \Lambda^k \right.)$, where $1 \le i \le (H-v)/(h-v)$ and $1 \le k \le N_0$.

Step 2. After the first application of the Viterbi algorithm we have: initial and transition probabilities for the superstates as determined at step 1 of the training algorithm described above, and the observations probability distributions for the top-level HMM, that is: the probabilities of each horizontal row of observations given each

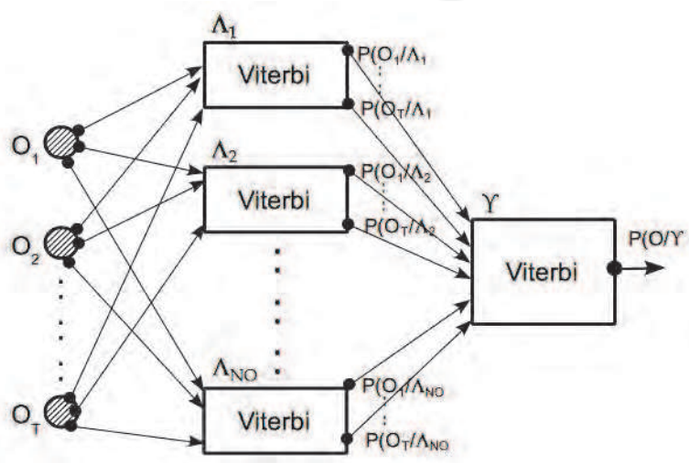


Fig. 6. Doubly embedded Viterbi

superstate. Now Viterbi is applied on the top-level HMM and the optimal sequence of superstates is obtained given the sequence of rows of observation vectors (vertical re-segmentation) and the probability of the entire sequence of observations (which characterizes the entire image) given the EHMM model created. This probability is compared at each iteration with the same probability obtained on the previous iteration (step 6 in Section 5.3).

Step 3. Vertical re-segmentation means reassignment of each row of observation vectors to the corresponding superstate (embedded 1D HMM). After we determine to which embedded 1D HMM each row of observation vectors appertains. Then using the findings at the first step of the *double embedded* Viterbi algorithm horizontal re-segmentation can be achieved.

In Figure 7 below we give an example of states and superstates segmentation for a given face image. Each color represents a different state in the superstates. As one can see, the 5 superstates found in this image are: forehead region, ends right above the eyebrows and is divided into 3 states, eye region, starts just above eyebrows and ends just after the pupil, is divided into 6 states, the nose region, starts just after the pupil and ends just before the nostrils, is divided into 6 states, the nostrils region, starts just under the nose region and ends at the middle between mouth and tip of the nose, is divided into 6 states, and finally the mouth region starts after the nostrils region and ends at the bottom of the image, is divided into 3 states. Also from the image we can see that the rows of observation vectors inside one superstate are distributed unevenly between states.

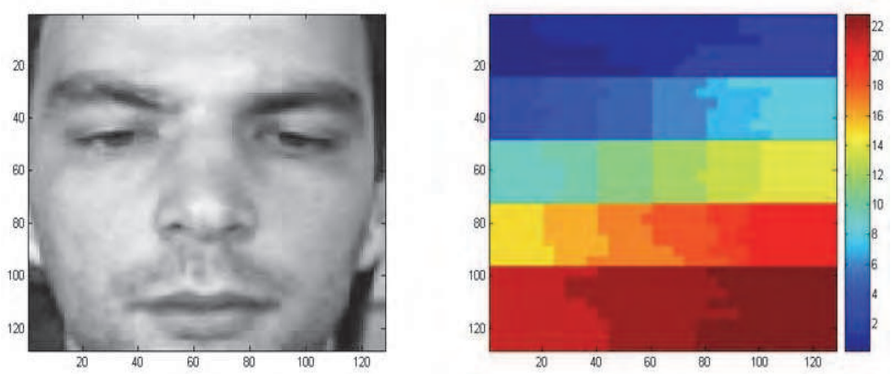


Fig. 7. State distribution after applying doubly embedded Viterbi

6.5 The evaluation process for an EHMM

In the training process described previously we have shown how an EHMM model is built for a subject in the database. After building separate EHMMs for all subjects in the database we can move to the recognition step where the likelihood of a face test image is evaluated against all models. The evaluation process comprises of the following steps:

Step 1. first the face image is scanned with a rectangular window from left-to-right and top-to-bottom and the observation vectors are extracted.

Step 2. then the sequence of observation vectors is introduced in each EHMM model and its corresponding likelihood is computed. Theoretically the probability of a sequence of observation vectors, given a model, is found using the forward-backward *evaluation algorithm*. However, in practice it was argued [Raibner 1989,

Kuo & Agazzi 1994] that the Viterbi algorithm can successfully replace the evaluation algorithm. For our EHMM we use a doubly embedded Viterbi algorithm. At the end of this step we have the probabilities of the test image to match each of the EHMM models in the recognition database.

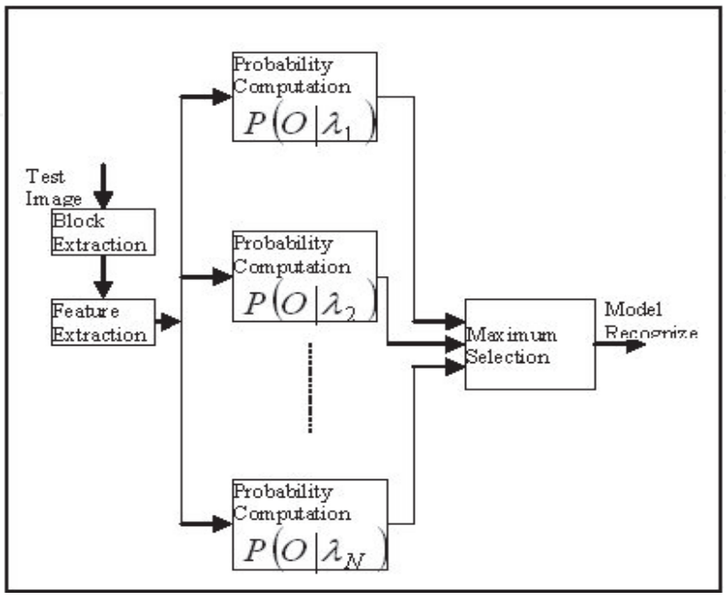


Fig. 8. HMM recognition scheme (N is the number of subjects in the database)

Step 3. the final step consists in comparing all the probabilities computed at the previous step and choosing as winner the model which returns the highest probability. The evaluation process is depicted graphically in Figure 8.

7. Implementation details

In order to implement our AFR system two different software programs were designed: one for the face detection and normalization processes and one to support the HMM based face recognition process. Many functions for face detection and recognition were based on a well known open source image processing library [OpenCV 2006]. Some details on each of these software workflows are given to facilitate other researchers.

7.1 Face detection

For the detection and cropping of all faces in the test databases we employed a well-known face detection algorithm [Viola & Jones 2000, 2001], described in section 2 above. In order to implement detection and cropping of all faces in all images in a single step, a tool was required to operate batch processes. This is implemented using Matlab. Such an approach allows additional high-level filters and other image processing techniques, also implemented in Matlab, to be easily linked with the OpenCV based face detection process. Thus the speed and efficiency of the OpenCV routines are coupled with the flexibility to incorporate supplemental Matlab filters into our test and evaluation process.

The Matlab program takes as input a folder of images, automatically loading each image, calling the face detection function from OpenCV and returning all face rectangles detected

in the given images. These are then cropped and saved on disk as new JPG image files. Note that this process facilitates a manual inspection or supplemental testing of a set of images to determine if they are correctly and uniformly cropped. To achieve the functionality of this program, there are two principal stages:

i. the declaration of the face detection function in OpenCV has to be modified in order to provide Matlab access to the detection function from the detection library using the standard mex (Matlab-C) interface[1];
ii. the OpenCV library is compiled with the Matlab-C compiler. The mex command was used to build the detection function adding all OpenCV dependencies.

At the end of this stage all detected faces were manually separated by subject in different folders. Throughout our tests we used different numbers of pictures per person in the training stage, and a fixed number of pictures in the testing stage. More exactly, we trained the system successively with 1 and up to 5 pictures per person, and we tested with the remaining 5 pictures that were not used in any training. In future tests we will denote the number of faces used for training as Nvs5: 1vs5, 2vs5, etc.

7.2 Face recognition

The second step in implementing the face recognition system was to build a program that would perform the main face recognition processes. The face recognition implementation was done in the C language using Microsoft Visual Studio. The implementation consists of three main components:

1. Top-level component is the first component and has the purpose of (i) reading multiple images from the disc, (ii) saving the output of the training stage (which is represented by the models) and (iii) analyzing the output of the testing stage. A detailed description of the training and testing stages can be found in Chap. 4;
2. Mid-level component: the second component which processes (pre-processing: illumination normalization, resize, filtering etc) the faces, computes observation vectors, builds and stores HMM models and computes likelihoods between faces and models. Please see Chapter 4 for a description of the HMM stages;
3. Low-level component is the third component which contains the basic routines of the EHMM algorithm (feature extraction, segmentation, Viterbi, state probability distribution etc) and uses functions implemented in the OpenCV library.

A detailed description of the processes taking place in each of these components and how they are interfaced is given in [Iancu 2010]. Figure 9 presents a visual explanation.

7.3 Databases and training datasets

For the experiments presented in the next sections of this chapter we used a mix of subjects from 3 standard databases: BioID, Achermann and UMIST. Each database provides some of our desired variations: BioID exhibits high variations in illumination, some expression variations and slight pose variations; Achermann presents some head rotations and slight illumination variations; UMIST covers a range of poses from frontal to semi-profile. A short description of each of these databases is given next.

BioID database: BioID[2] is a dataset consisting of 1521 gray level images with a resolution of 384 × 286 pixels, containing 23 different test persons with frontal views with variations in

[1]http://www.mathworks.com/support/compilers/interface.html
[2]http://www.bioid.com/support/downloads/software/bioid-face- database.html

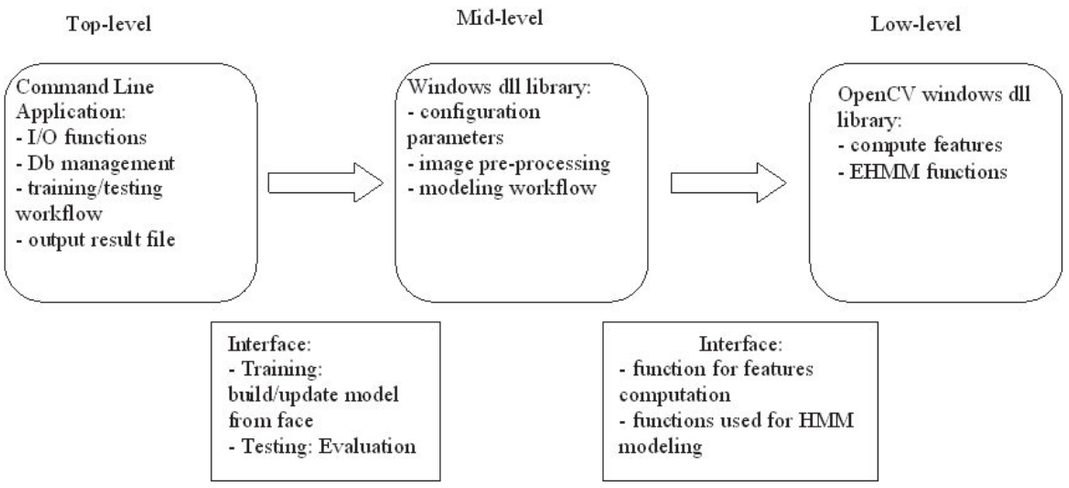


Fig. 9. Main program workflow

facial expression and illumination. The actual size of the face inside the picture is on average 128 × 128. From the entire database 200 pictures of 20 different persons were selected. Faces were selected to maximize pose and illumination variations as far as possible in the selected picture.

Achermann database: The Achermann database[3] contains 260 images of 26 people, each with 10 images, with 143×143 pixel in size. Unlike many other databases which contain only frontal views, the faces in this database span a range of pose angles.

UMIST database: UMIST database[4] consists of 564 images of 20 people, each covering a range of poses from profile to frontal views. Subjects cover a range of race/-sex/appearance. The files are all in PGM format, approximately 220 × 220 pixels in 256 shades of grey. From this database we extracted 100 pictures of 10 subjects, with pictures ranging from frontal up to 30 degrees right and left turn.

Tests were performed using the software tools described in section previously in Section 6.2 on a database formed by combining the 3 databases described above. This comvined database consisted of 560 pictures of 56 subjects, for each subject 10 pictures being selected. Unless stated otherwise, pictures were resized to 128 × 128 grayscale pixels then each face was scanned using a 12 × 12 window from left to right and top to bottom, with an overlap of 8 pixel both vertically and horizontally, in order to extract the observation vectors. The first 3×3 2D DCT coefficients corresponding to the low frequencies are retained. The EHMM depicted in Figure 5(b) is used, with 5 super states and 3-6-6-6-3 embedded states respectively, representing forehead-eyes-nose-mouth-chin areas. A standard number of 3 Gaussians were used in each state to model the feature vectors for most of the following experiments. In the training step, 1 to 5 pictures for each subject are used for building the corresponding EHMM model. After training, in order to check the algorithm, *verification* is performed, where the images used for training are also used for testing. If the algorithm is correct than the recognition rate should be 100%. In the recognition phase, the 5 images per person not used for training are utilized.

[3]http://iamwww.unibe.ch/ fkiwww/staff/achermann.html
[4]http://www.sheffield.ac.uk/eee/research/iel/research/face.html

8. Experimental results

As structure for the EHMM we used Samaria's thesis as starting point. According to Samaria's experiments presented in his PhD thesis [Samaria 1994], the most efficient structure allowing the smallest error rate (5.5%) consists in 5 super states with 3-6-6-6-3 embedded states respectively and was tested employing as observation vectors the pixel intensities of a face region and the dataset of face regions was drawn from the ORL database. The same EHMM structure was used by [Nefian 1999] 2D DCT coefficients as observa- tion vectors and the dataset of face regions was drawn from the ORL database. Nefian obtained an improved error rate of 2%.

8.1 Test data 1 – different sizes of face region

A first set of experiments is directed towards determining an optimal size of the detected and cropped face regions used for recognition. The sizes of the faces available in research databases can vary significantly. As examples, it varies between 128 × 128 for BioID, 143 × 143 for Achermann and 220 × 220 for UMIST. As our experiments draw images from these, and other data sources, a consistent normalization between face regions is important. Downscaling will result in information loss, but this is still preferable to upscaling which can introduce artifacts and create false information. There is also a reduction in computational demands for smaller pictures – another reason to favour downscaling. For example it takes 75% less time to process a 64×64 image region than for 128 × 128 sized regons. Thus processing smaller pictures is clearly desirable for algorithms to run efficiently in handheld devices such as digital cameras. In this set of experiments we tested sizes between 64 × 64 and 196 × 196. The bicubic spline interpolation technique was used for image downscaling where required. The results are given in the Table 2 and depicted in Figure 10.

Picture size	1vs5	2vs5	3vs5	4vs5	5vs5
64 × 64	42.86	56.79	73.57	83.57	89.64
96 × 96	52.14	64.64	76.43	86.07	87.86
128 × 128	66.43	71.07	81.07	83.57	86.07
192 × 192	61.79	69.29	77.86	83.93	83.57

Table 2. Recognition rates (%) for different picture sizes

8.2 Test data 2 – different numbers of gaussians

One of the underlying assumptions of our HMM model is that a random signal can be modeled by one or more Gaussian probability functions. If the signal is simple, it clusters around one mean value and it can be modeled by a single Gaussian. For more complex signals there may be more than one clustering center, and a more accurate model will match the number of Gaussians to the number of primary clusters. The objective of this experiment is to test how a varying number of Gaussians will influence the underlying recognition rates. We used face regions of size 128 × 128 pixels. This size of images should provide close to an optimal recognition rate as detailed in section 8.1.

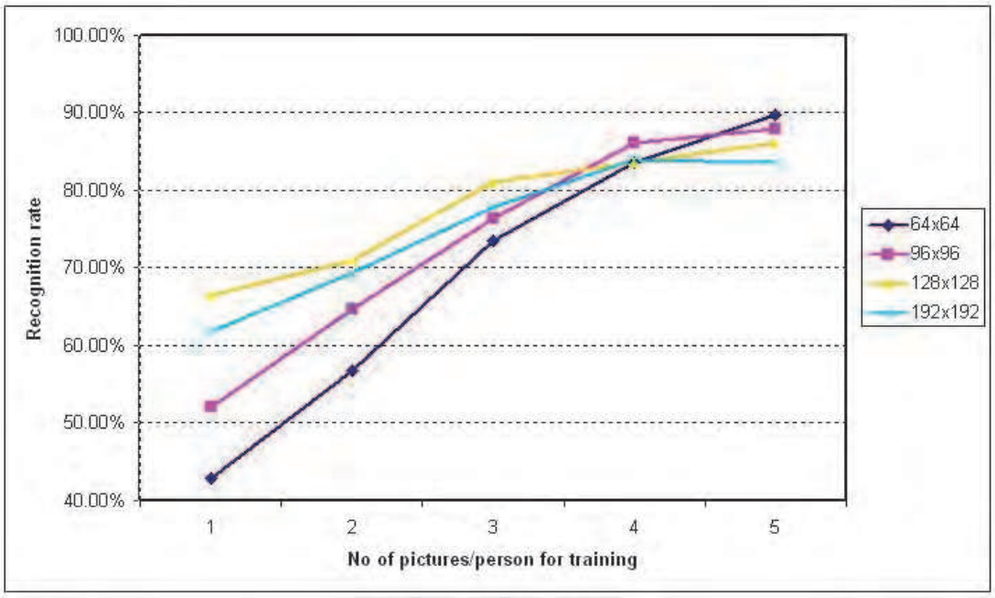


Fig. 10. Recognition rates for different picture sizes

No. of Gaussians	1vs5	2vs5	3vs5	4vs5	5vs5
1	58.93	65.71	72.14	81.07	82.14
2	58.21	67.86	79.29	82.86	86.07
3	66.43	71.07	81.07	83.57	86.07
4	55.00	67.14	78.21	84.29	88.21

Table 3. Recognition rates (%) for different numbers of Gaussians

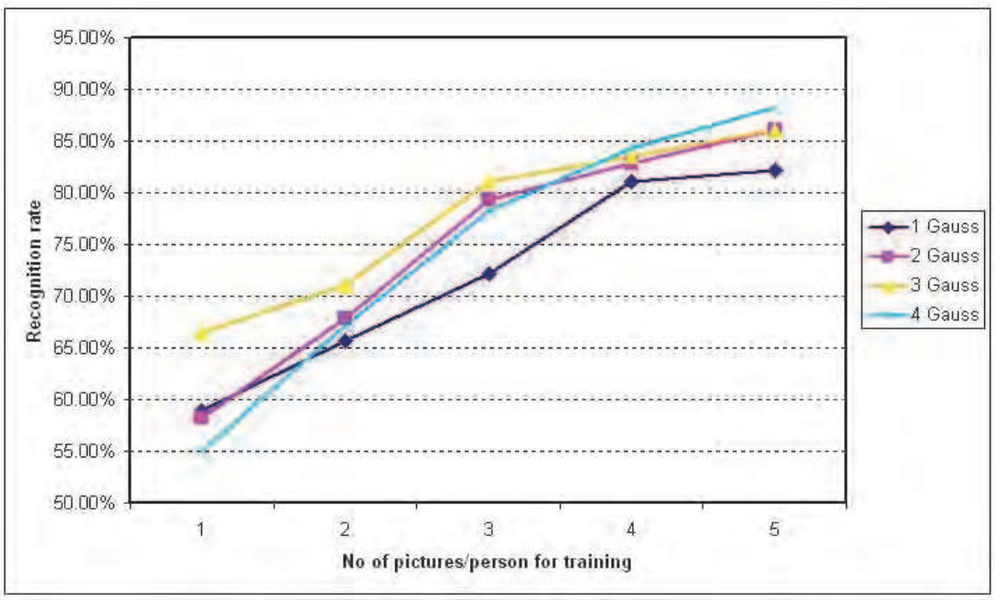


Fig. 11. Recognition rates for different numbers of Gaussian functions

The best results are obtained for mixtures of 3 Gaussians per HMM. For 2 or 4 Gaussians per HMM good results are also obtained, particularly when more training images per person are available. Note, however, that 4 Gaussians per HMM requires significantly more computation time and the benefits are marginal. Because the difference in processing time between modeling with 2 and 3 Gaussians is not significant (the recognition process with 2 Gaussians is in average 15% faster than for 3 Gaussians), it was decided to use a mixture of 3 Gaussians for our later experiments.

Note that for more complex or unusual model characteristics it is expected that the improvement in general recognition rates for 3 Gaussians could be more significant than is shown here. For now we can state that increased computation from using 3 Gaussians is likely to be balanced by the adaptability that is available to the model to handle more extreme variations in facial characteristics. No investigation of varying the underlying number of Gaussians for each superstate was undertaken. However it seems likely that superstates which cover complex features, such as the eye, nose and mouth regions are likely to benefit from increasing the underlying number of Gaussians in the more complex face regions.

8.3 Experimental results for different numbers of DCT coefficients

Again, the idea behind using fewer features to represent a signal is to speed up the recognition process, without adversely affecting the recognition rates. The number of 2D DCT coefficients we used previously was 3 × 3. In Figure 12 we show the results for 2×2, 3×3 and 5×5 DCT coefficients used. The results are more clearly presented in Table 4. We can see that on average 3×3 coefficients perform marginally better than 5 × 5 when up to 3 training pictures are used. However as more pictures are used in training, 5×5 coefficients seem to characterize the image better although in many applications this improvement is may not be sufficient to justify the additional computational effort for handheld imaging devices.

DCT	1vs5	2vs5	3vs5	4vs5	5vs5
2×2	51.79	63.57	75.71	82.86	83.21
3×3	66.43	71.07	81.07	83.57	86.07
5×5	62.14	70.36	79.64	86.07	90

Table 4. Recognition rates (%) for different numbers of DCT features

8.4 Experimental results for simplified topologies

The final tests of this section involve using a different topology for the EHMM. The core model was drawn from the work of Samaria and Nefian employs 5 superstates with an internal organization of 3-6-6-6-3 embedded states respectively, on images resized at 128 × 128. Here a simplified version of the classic EHMM is tested: the face is segmented into 4 superstates and 2-4-4-2 embedded states. Smaller face regions are used to further reduce the memory and computation required. Face regions were resized to 32 × 32 and 64 × 64. Details of the tests and results are presented in Figures 13 & 14. The numerical results can be found in Tables 5 and 6.

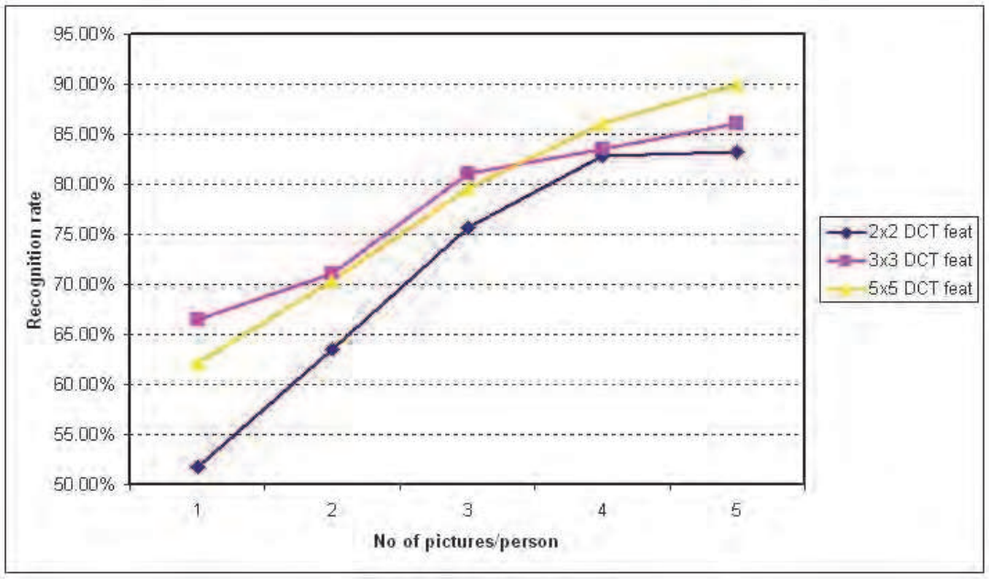


Fig. 12. Recognition rates for different numbers of features

Window size / No Gaussian	1vs5	2vs5	3vs5	4vs5	5vs5
8 × 8/1 Gauss	57.86	67.50	72.86	80.00	82.14
12 × 12/1 Gauss	51.43	51.79	64.29	76.43	83.21
8 × 8/2 Gauss	58.57	64.64	76.43	84.29	86.43
12 × 12/2 Gauss	55.36	49.64	61.07	74.64	82.14

Table 5. Recognition rates (%) for picture size 32 × 32

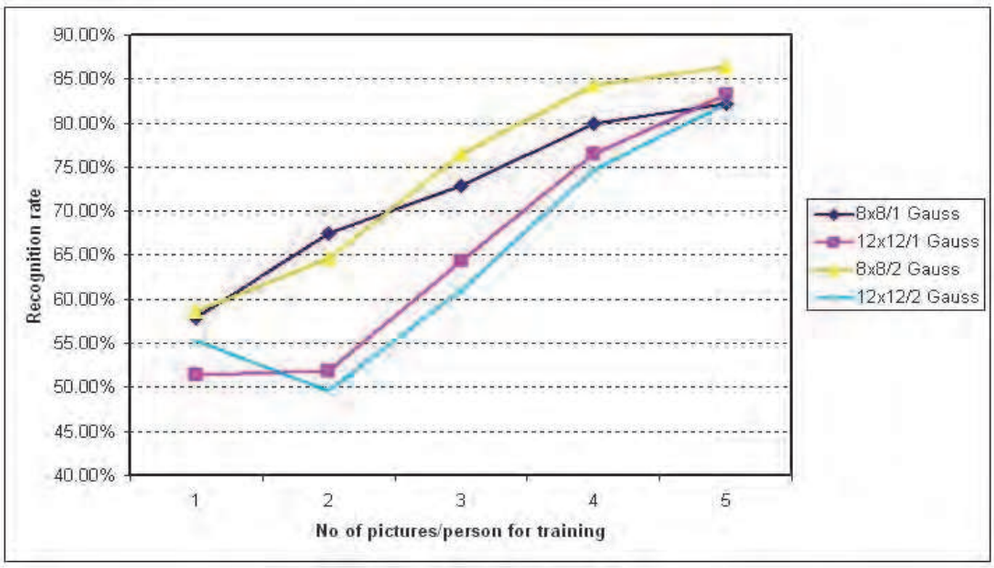


Fig. 13. Recognition rates for 32 × 32 picture size

There were other parameters we had to change in the tests in order to increase the recognition rates. Originally we started the tests on 32 × 32 pixels images scanned with a 12 × 12 pixels window with 8 pixels overlap (75% of the size of the scanning window). However a simple calculation shows that in this case we only have 6 scanning steps both vertically and horizontally so there will be superstates with only 6 observation vectors, that meaning that some states will have only 1 or 2 observation vectors to be modeled with Gaussian mixtures. Obviously modeling with 2 Gaussian mixtures a single observation vector does not make sense. Furthermore having a single observation vector per state cannot offer a detailed description of that state.

Window size/No. of Gaussians	1vs5	2vs5	3vs5	4vs5	5vs5
8 × 8/1 Gauss	55.71	61.79	69.64	75.36	76.43
12 × 12/1 Gauss	60.71	65.36	72.50	78.57	82.50
8 × 8/2 Gauss	59.64	63.93	71.79	79.64	82.86
12 × 12/2 Gauss	57.14	65.71	70.36	81.07	85.00

Table 6. Recognition rates (%) for picture size 64 × 64

Because of this, low recognition rates were obtained for 32 × 32 pixels images scanned with 12×12 pixels windows. However, for images sized 64 × 64 pixels, scanning with an 8 × 8 pixels window could mean taking too much detail and missing larger facial features. This may explain why the best results for 64 × 64 size images are obtained when scanning with 12 × 12 pixels window.

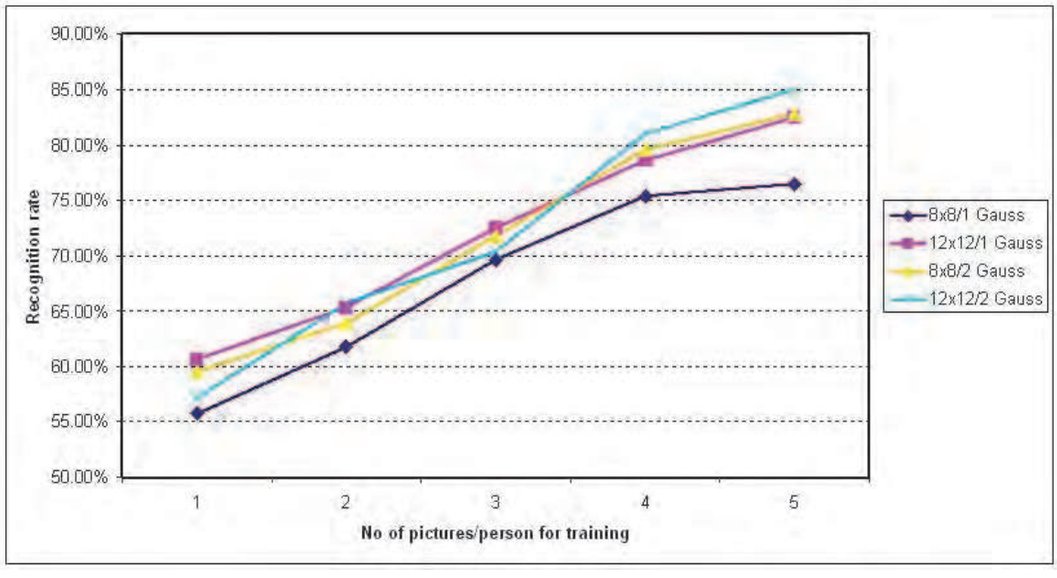


Fig. 14. Recognition rates for 64 × 64 picture size

8.5 Experiments using different illumination normalization techniques

For our experiments to tackle the problem of illumination variations a study was undertaken of the effectiveness of the following three illumination normalization algorithms: (i) *histogram equalization* (HE) based on [Gonzalez & Woods 1992], (ii) *contrast limited adaptive histogram equalization* (CLAHE) based on [Zuiderveld 1994], and (iii) the relative new method of DCT in the logarithm domain - *logDCT* based on [Chen *et al* 2006]. In addition paired combinations of several of these techniques were also evaluated. To the best of our knowledge this is the first attempt in the literature to compare different illumination normalization techniques in the context of EHMM.

Illum Norm Tech	1vs5	2vs5	3vs5	4vs5	5vs5
No normalization	66.43	71.07	81.07	83.57	86.07
HE	65.00	74.64	85.71	90.71	92.50
CLAHE	68.21	75.71	82.86	88.21	90.57
LogDCT	52.50	58.21	70.71	76.43	77.86
LogDCT+CLAHE	60.00	70.36	75.00	82.14	84.64
CLAHE+LogDCT	68.21	77.50	84.64	90.00	92.86
LogDCT+HE	66.43	72.86	81.43	86.79	90.00
HE+LogDCT	53.57	64.29	76.79	85.00	88.57
HE+CLAHE	68.57	78.21	86.07	92.14	95.36
CLAHE+HE	71.07	79.29	87.14	92.86	95.71

Table 7. Recognition rates(%) for illumination normalization techniques

We tested these three illumination normalization techniques, *viz*, HE, CLAHE, LogDCT and compared the recognition results with the results obtained when no illumination normalization was performed. After visually observing the results of illumination normalization on each image in the database we concluded that LogDCT flattens the facial features excessively whereas CLAHE enhances them. Thus the idea of combining these two normalization methods, was mooted. It was then decided to investigate a number of other combinations of these techniques and the results are presented below. The number of DCT components that are canceled for logDCT depends on the size of the image and the level of the illumination variation present in the image. For our tests we used the first 4 DCT coefficients for 128 × 128 images. The recognition rates for each illumination normalization technique and the combinations we tested are given in Table 7.

In Figure 15 a set of face images affected by various degrees of illumination are shown together with the results of applying each illumination normalization technique or combinations thereof to the original images. Making a parallel between the recognition rates given in Table 7 and the visual results shown in Figure 15, a few observations can be provided:

- the best recognition rates are obtained when combining HE and CLAHE regardless of the order.

- Better recognition rates are achieved using the simple HE technique when compared to the more sophisticated CLAHE and LogDCT techniques.
- LogDCT returns very poor results even when compared to the rates obtained when no illumination normalization technique is used.
- From a human's perceptual point of view, the most efficient visually illumina- tion normalization technique appears to be LogDCT+CLAHE but this gives recognition rates poorer than the original pictures

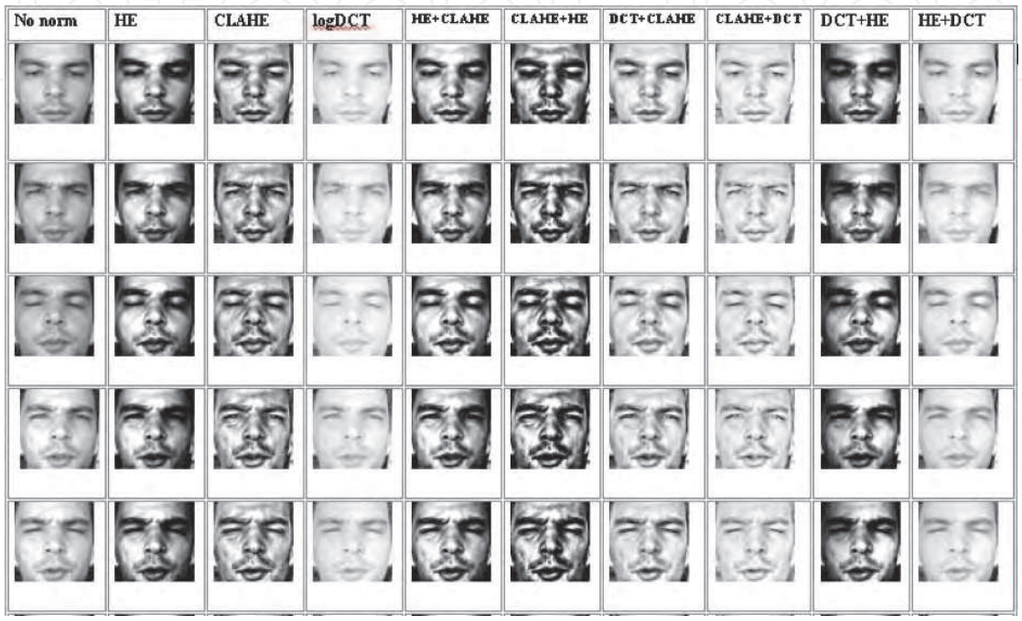


Fig. 15. Illustrative examples of various illumination normalization techniques applied to a representative subset of face data.

- The best recognition rates are obtained for illumination normalization which enhances the facial features: HE, CLAHE and combinations of these two.

There are two exceptions that contradict this last rule: (i) LogDCT+HE: despite the fact that from Figure 15 this combination appears to enhance facial features, it also appears to be unable to eliminate the influence of illumination; (ii) The second exception is CLAHE+LogDCT which gives quite good results despite looking somewhat flatter than the other techniques.

9. Discussion of our experiments

Throughout this chapter we described a series of tests performed with the purpose of finding the optimal combination of factors that influence the recognition process: size of face image, topology of the model, that is number of superstates, number of states for each super state and number of Gaussians to model the observation vectors, illumination normalization technique to diminish the differences in illuminations, and coefficients to describe the information contained in the image.

In order to make things more difficult for our recognizer, we tried to emulate as best possible the less that ideal appearance and diversity of a consumer collection by combining

three different standard databases in one big collection of images. The databases used were described briefly in Section 7.3. Also, throughout the tests we used different numbers of faces of the subjects for tests and training: from 1 to 5 faces for training and the remaining 5 faces for testing. From the results presented in Section 8 the following conclusions can be reached:

i. An optimal size for face image when using HMM is 128 × 128 pixels, although the recognition rates obtained for the smaller size of 96 × 96 pixels are quite close and may offer a better choice for applications where memory and computational efficiency are important, e.g. in handheld imaging devices;
ii. From experiments performed in section 8.2 the optimal number of Gaussian functions is 3, representing a trade-off between best precision and computational burden; but we remark that with 2 Gaussians a faster computation is achieved with almost the same recognition performance; the effects of varying the number of Gaussians across super-stages was not considered in this research;
iii. The optimal performance with 2D DCT is achieved by employing the first 9 coefficients, but we noted in section 8.3 that using the first 4 coefficients gives an acceptable result as well and may be preferable where speed of computation and memory efficiency are important; no significant improvement was noted when we used Daubechies wavelets in place of DCT;
iv. In section 8.4, very good results were obtained for a reduced 2-4-4-2 EHMM topology applied on very small images: these results improve when increasing the number of training images per person, and recognition rates as high as 86.43% were achieved in our experiments. As no illumination normalization was used and these tests were performed on a combined database rather than a single standard database, the results which were obtained may be regarded as highly promising for real-world applications.
v. Finally, in Section 8.5 three different illumination normalization techniques are used in the pre-processing phase of our recognizer. We also investigated some non-standard combinations of these techniques to determine their suitability for pre-processing data for a HMM face recognition algorithm. The best results were obtained for CLAHE and HE with over 95% recognition rates. Very good performances were obtained when using a combination of CLAHE and logDCT, with 92.87% recognition rate, but also when using the more basic HE, with up to 92.5%. This analysis of various image normalization filters should pro- vide a useful baseline for future researchers in face recognition. One aspect we would have liked to investigate was the potential of such combining of illumination normalization filters to improve the performance of other well-known face recognition techniques such as PCA, ICA and AAM methods.

10. References

Y. Adini, Y. Moses, and S. Ullman. Face recognition: the problem of compensating for changes in illumination direction. IEEE Trans. on Pattern Analysis and Machine Intelligence, 19(7):721–732, 1997.

L.E. Baum and T. Petrie. Statistical inference for probabilistic functions of finite state markov chains. Ann. Math. Stat, 37(6):1554–1563, 1966.

L.E. Baum, T. Petrie, G. Soules, and N. Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains. Ann. Math. Stat, 41(1):164–171, 1970.

R. Chellappa, B.S. Manjunath, and C.V.D. Malsburg. A feature based approach to face recognition. IEEE Conference on Computer Vision and Pattern Recognition, pages 373–378, 1992.

R. Chellappa, C. Wilson, and S. Sirohey. Human and machine recognition of faces: A survey. Proc. of IEEE, 83(5):705–741, May 1995.

W.L. Chen, M.J. Er, and S.Q. Wu. Illumination compensation and normalization for robust face recognition using discrete cosine transform in logarithm domain. Systems, Man and Cybernetics, Part B, 36:458–466, 2006.

P. Corcoran, C. Iancu, and G. Costache. Improved HMM based face recognition system. OPTIM, Brasov, Romania, 4:143–146, 2006.

P. Corcoran, M.C. Ionita, and G. Costache. Pose invariant face recognition using AAM models. Proceedings of the 10th International Conference on Optimization of Electrical and Electronic Equipment (OPTIM'06), 4:181– 184, 2006.

G.N. Costache. Advances in automated image categorization: Sorting images using person recognition techniques. PhD thesis, Dept of Electronic Engineering, NUI Galway, 2007.

G.N. Costache, P.M. Corcoran & P. Puslecki, *Combining PCA-based datasets without retraining of the basis vector set*, Pattern Recognition Letters, Volume 30, Issue 16, Pages 1441-1447, December 2009,

B. Du, S. Shan, L. Qing, and W. Gao. Empirical comparisons of sev- eral preprocessing methods for illumination insensitive face recognition. Acoustics, Speech, and Signal Processing, 2:981–984, 2005.

M. Gokmen E. Vucini and M.E. Groller. Face recognition under varying illumination. 15th WSCG, pages 57–64, 2007.

R.C. Gonzalez and R.E. Woods. Digital image processing. Addison-Wesley, Reading, MA, 1992.

R. Hummel. Image enhancement by histogram transformation. Computer Graphics and Image Processing, 6(2):184–195, 1977.

M.C. Ionita. Advances in the design of statistical face modelling techniques for face recognition. PhD thesis, Dept of Electronic Engineering, NUI Galway, 2008.

B.H. Juang and L.R. Rabiner. Automatic speech recognition - a brief history of the technology development. Elsevier Encyclopedia of Language and Linguistics, Second Edition, 2005.

T. Kanade. Computer recognition of human faces. Interdisciplinary Systems Research, 47, 1977.

S. Kuo and 0. Agazzi. Keyword spotting in poorly printed documents using pseudo 2-d hidden markov models. IEEE Transactions on Pattern Analysis and Machine Intelligence, 16:842–848, August 1994.

H.S. Le and H. Li. Face identification system using single hidden markov model and single sample image per person. IEEE International Joint Conference on Neural Networks, 1, 2004.

T.M. Lehmann, C. Gnner, and K. Spitzer. Survey: Interpolation methods in medical image processing. IEEE Transactions on Medical Imaging, 18:1049–1075, 1999.

A.V. Nefian. A hidden markov model based approach for face detection and recognition. PhD Thesis, 1999.

A.V. Nefian and M.H. Hayes III. Face detection and recognition using hidden markov models. Image Processing, ICIP 98, Proceedings. 1998 International Conference on,, 1:141–145, October 1998.

A.V. Nefian, and M.H. Hayes III, *Face recognition using an embedded HMM*, IEEE Conf. on A/V-based Biometric Person Authentication, pp. 19–24, 1999.

A.V. Nefian and M.H. Hayes III. Maximum likelihood training of the em- bedded hmm for face detection and recognition. International Conference on Image Processing, 1:33–36, 2000.

OpenCV *Online Reference Manual*, Intel Corporation 2006.

A.J. O'Toole, P.J. Phillips, F. Jiang, J. Ayyad, and N. Penardand H. Abdi. Face recognition algorithms surpass humans matching faces over changes in illumination. PAMI(29), 29(9):1642–1646, 2007.

P.J. Phillips, M. Hyeonjoon, S.A. Rizvi, and P.J. Rauss. The feret evaluation methodology for face-recognition algorithms. IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(10):1090–1104, 2000.

W. K. Pratt, *Digital Image Processing*, New York: John Wiley & Sons 1991.

L.R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. Proc. of IEEE, 77(2):257–286, February 1989.

L.R. Rabiner and B.H. Juang. An introduction to hidden markov models. IEEE ASSP Mag, 3(1):4–16, 1986.

E. Rentzeperis, A. Stergiou, A. Pnevmatikakis, and L. Polymenakos. Impact of face registration errors on recognition. Artificial Intelligence Applications and Innovations, 204:187–194, 2006.

F. Samaria. Face recognition with hidden markov models. Ph.D. thesis, Department of Engineering, Cambridge University, UK, 1994.

F. Samaria and F. Fallside. Face identification and feature extraction using hidden markov models. Image Processing: Theory and Applications, Elsevier, pages 295–298, 1993.

F. Samaria and A.C. Harter Parameterization of a stochastic model for human face identification. Applications of Computer Vision, 1994., Pro- ceedings of the Second IEEE Workshop on, 77:138–142, December 1994.

W.A.P. Smith and E.R. Hancock. Single image estimation of facial albedo maps. Lecture Notes in Computer Science, 3704:517–526, 2005.

E. Steinberg, P. Corcoran, Y. Prilutsky, P. Bigioi, M. Ciuc, S. Ciurel, & C. Vertran, *Classification system for digital images using workflow, face detection, normalization, and face recognition*, US Patent 7,555,148, June 2009.

P. Viola and M. Jones, *Robust real-time object detection*, presented at the 2nd International workshop on Statistical and Computational Theories of Vision, Vancouver, Canada, July 13th, 2001.

M.H. Yang, D. Kriegman, and N. Ahuja. Detecting faces in images: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence, 24(1):34–58, 2002.

W. Zhao, R. Chellappa, A. Rosenfeld, and P.J. Phillips. Face recognition: A literature survey. ACM Computing Surveys, pages 399–458, 2003.

K. Zuiderveld. Contrast limited adaptive histogram equalization. Graphics gems IV, Academic Press Professional, Inc., pages 474–485, 1994.

IntechOpen

IntechOpen



New Approaches to Characterization and Recognition of Faces
Edited by Dr. Peter Corcoran

ISBN 978-953-307-515-0
Hard cover, 252 pages
Publisher InTech
Published online 01, August, 2011
Published in print edition August, 2011

As a baby, one of our earliest stimuli is that of human faces. We rapidly learn to identify, characterize and eventually distinguish those who are near and dear to us. We accept face recognition later as an everyday ability. We realize the complexity of the underlying problem only when we attempt to duplicate this skill in a computer vision system. This book is arranged around a number of clustered themes covering different aspects of face recognition. The first section presents an architecture for face recognition based on Hidden Markov Models; it is followed by an article on coding methods. The next section is devoted to 3D methods of face recognition and is followed by a section covering various aspects and techniques in video. Next short section is devoted to the characterization and detection of features in faces. Finally, you can find an article on the human perception of faces and how different neurological or psychological disorders can affect this.

How to reference
In order to correctly reference this scholarly work, feel free to copy and paste the following:

Peter M. Corcoran and Claudia Iancu (2011). Automatic Face Recognition System for Hidden Markov Model Techniques, New Approaches to Characterization and Recognition of Faces, Dr. Peter Corcoran (Ed.), ISBN: 978-953-307-515-0, InTech, Available from: http://www.intechopen.com/books/new-approaches-to-characterization-and-recognition-of-faces/automatic-face-recognition-system-for-hidden-markov-model-techniques

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2011 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License, which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen