A PROBABILISTIC MODEL OF FACE MAPPING APPLIED TO PERSON RECOGNITION

THÈSE N$^\circ$ 3136 (2004)

PRÉSENTÉE À LA FACULTÉ D'INFORMATIQUE ET COMMUNICATIONS

Institut Eurécom

SECTION DES SYSTÈMES DE COMMUNICATION

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Florent PERRONNIN

Ingénieur de l'Ecole Nationale Supérieure des Télécommunications, Paris, France
de nationalité française

acceptée sur proposition du jury:

Directeur:	Prof. Jean-Luc Dugelay
Rapporteurs:	Prof. Touradj Ebrahimi
	Dr. Jean-Claude Junqua
	Prof. Raymond Knopp
	Henri Sanson

Lausanne, EPFL
2004

A Probabilistic Model of Face Mapping Applied to Person Recognition

November 17, 2004

Abstract

Within the field of pattern recognition, biometrics is the discipline which is concerned with the automatic recognition of a person based on his/her physiological or behavioral characteristics. Face recognition, a central area in biometrics, is a very challenging task and is still largely considered an open problem. However, it is worthwhile to note that most face recognition algorithms focus on the feature extraction issue, and that much less attention has been given to the classification stage.

In this dissertation, we introduce a novel measure of "distance" between faces which involves the estimation of the *set of possible transformations between face images of the same person*. The global transformation, which is assumed too complex for direct modeling, is approximated with a set of local transformations under a constraint imposing consistency between neighboring local transformations. The proposed local transformations and neighboring constraints are embedded within the probabilistic framework of the two-dimensional hidden Markov model (2-D HMM) in the case of discrete states and of the two-dimensional state-space model (2-D SSM) in the case of continuous states.

To make the proposed face recognition approach practical, we also consider novel efficient approximations of the intractable 2-D HMM and 2-D SSM: the turbo HMM and the turbo SSM respectively. They consist of a set of inter-connected horizontal and vertical 1-D Markov chains that communicate through an iterative process.

Once a proper measure of distance has been defined, we turn to the problem of face image retrieval in large databases. To reduce the computational cost, the face space is partitioned through a clustering of the data. The main challenge that we address is the computation of a cluster centroid which is consistent with the proposed measure of distance.

Finally, we consider the problem of identity verification which requires a robust confidence measure. The issue is the accurate modeling of wrongful claims. For a distance such as the one introduced in this dissertation, we can model either the set of possible transformations between face images of different persons or directly the impostor distribution. We show that the latter approach leads to the best classification.

Résumé

Dans le domaine de la reconnaissance des formes, la biométrie est la discipline qui consiste à identifier une personne à partir de ses caractéristiques physiques ou comportementales. La reconnaissance de visages, qui tient une place centrale en biométrie, est une tâche particulièrement difficile et est généralement considérée comme un problème ouvert. Cependant, il convient de noter que la plupart des algorithmes de reconnaissance de visages se concentrent sur le problème de l'extraction des vecteurs caractéristiques et que l'étape de classification a reçu une attention moindre.

Nous introduisons dans cette dissertation une nouvelle mesure de "distance" entre visages qui nécessite d'estimer *l'ensemble des transformations possibles entre images de visages d'une même personne.* La transformation globale, que nous supposons trop complexe pour être modélisée directement, est approximée par un ensemble de transformations locales, sous la contrainte que des transformations voisines doivent rester cohérentes entre elles. Transformations locales et contraintes de voisinage sont incorporées dans le cadre probabiliste d'un modèle de Markov caché bi-dimensionel (MMC 2-D) dans le cas d'états discrets ou d'un modèle espace-état bi-dimensionnel (MEE 2-D) dans le cas d'états continus.

Pour que cette approche soit utilisable en pratique, nous considérons aussi de nouvelles approximations performantes des MMC 2-D et MEE 2-D: les turbo MMC et turbo MME respectivement. Ils consistent en un ensemble de chaînes de Markov 1-D inter-connectées qui communiquent au travers d'un processus itératif.

Après avoir défini cette mesure de distance, nous nous tournons vers le problème de la recherche d'images de visages dans de grande bases de données. De manière à réduire le temps de calcul, l'espace des images est partitionné à l'aide d'un algorithme de regroupement des données. La problème principal que nous nous attachons à résoudre est le calcul d'un centroïde qui soit cohérent avec la mesure de distance proposée.

Finalement, nous nous intéressons au problème de la vérification des identités, ce qui nécessite une mesure de confiance robuste. La difficulté est alors de modéliser les transactions frauduleuses. Pour une distance telle que celle introduite dans cette dissertation, nous avons le choix de modéliser la transformation entre images de personnes différentes ou la distribution des imposteurs. Nous montrons que la seconde approche conduit à une meilleure classification.

Acknowledgments

First, I would like to express my gratitude to Jean-Claude Junqua and Roland Kuhn, two researchers I admire and whose immense talent is only matched by their genuine humility. It was my good fortune to work with them during the two years I spent at the Panasonic Speech Technology Laboratory (PSTL) in Santa Barbara, California. They helped me develop my taste for research in general, and pattern recognition in particular. They also taught me how to conduct proper research and their lessons have been invaluable to me for the past three years.

I would like to thank Jean-Luc Dugelay, my Ph.D. advisor, for hosting me in the image processing group and for finding the financial resources which were necessary to carry out my research. My thanks also go to France Telecom Research and Development, and particularly to Henri Sanson, for funding my research activities for three years at a time when most companies were drastically reducing their research spendings.

I am indebted to Kenneth Rose, a most revered professor at the University of California in Santa Barbara (UCSB). Our collaboration on several aspects of this work was extremely fruitful. Had I not received his guidance during the early stages of my Ph.D., this thesis would have been very different, without any doubt for the worse.

I am also extremely grateful to the members of my Ph.D. defense committee for accepting the responsibility of reviewing my work and for accommodating my very tight schedule.

I would like to thank my fellow Ph.D. students and colleagues in the multimedia communications department with whom I shared numerous discussions, but also friendship. Among the others I would like to thank Gwenaël Doërr, Christian Rey, Luca Brayda, Federico Matta, Fabrice Souvannavong, Emmanuel Garcia, Benoit Huet, Ana Andrés del Valle, Nicolas de Saint Aubert, Joakim Jiten, Vivek

Tyagi and Jihane Bennour. I am especially thankful to Christian Wellekens, one of the most established European experts in the field of speech processing and a very talented professor, and Fabio Valente, the VB virtuoso, for sharing with me some of their vast knowledge on speech recognition and hidden Markov models. Special thanks also to Caroline Mallauran for turning my research (which, I have to admit, can sound very boring to the non-specialist) into an attractive demonstrator.

Last, but most certainly not least, I would like to express my deepest gratitude to my wife Florence, for showing a genuine interest in my research (although her area of expertise – performance evaluation – is significantly different from pattern recognition), for reading and commenting on my articles with great care, and for her ceaseless encouragements during these three years. She shaped this work in more ways she could possibly imagine.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

List of Abbreviations

AFR	Automatic Face Recognition
ASR	Automatic Speech Recognition
BIC	Bayesian Intra-/Extra-personal Criterion
DET	Detection Error Trade-off
EER	Equal Error Rate
EM	Expectation Maximization
FAR	False Acceptance Rate
FD	Face Dependent
FIT	Face Independent Transformation
FRR	False Rejection Rate
Gpm	Gaussians per mixture
HMM	Hidden Markov Model
MAP	Maximum a Posteriori
ML	Maximum Likelihood
OCR	Optical Character Recognition
PCA	Principal Component Analysis
PMLGFT	Probabilistic Mapping with Local Grid and Feature Transformations
PMLGT	Probabilistic Mapping with Local Grid Transformations
PMLT	Probabilistic Mapping with Local Transformations
SAT	Speaker Adaptive Training
SSM	State-Space Model
T-HMM	Turbo HMM
T-SSM	Turbo SSM

1

Introduction

1.1 Motivation

Let us consider the general pattern classification problem where a sample x is to be assigned to one of a set of possible classes $\{\omega_i\}$. Within the Bayesian decision framework, the optimal classifier commonly referred to as the *minimum risk classifier* employs the following decision rule: assign the observed pattern x to the class ω_i that minimizes the *conditional risk* given by:

$$R(\omega_i|x) = \sum_j \lambda(\omega_i|\omega_j) P(\omega_j|x) \tag{1.1}$$

where the *loss function* $\lambda(\omega_i|\omega_j)$ quantifies the loss incurred for selecting ω_i when the true class of x is ω_j, and where $P(\omega_j|x)$ is the (posterior) probability of class ω_j given that sample x was observed, which is computed in practice from $P(\omega_j)$ – the *class prior probabilities* – and $P(x|\omega_j)$ – the *class-conditional probability density functions* (pdf). For a detailed review see, e.g., [DHS00]. Typically, the loss functions are determined by the application and are hence assumed known, whereas the class priors and class-conditional pdf's need to be estimated given a training set of labeled samples. In practice, the more challenging task is the estimation of class-conditional pdf's that characterize intra-class variability, and its accuracy is a primary determining factor for the classifier performance. The quality of these estimates hinges on two crucial factors: the correctness of the chosen model and the availability of a sufficiently large training set to estimate the model parameters.

Obviously, these two considerations are inter-related as the fewer parameters of a compact model will require less training data to be robustly estimated.

The discipline of *biometrics* is concerned with the automatic recognition of a person based on his/her physiological or behavioral characteristics [JRP04]. Biometric applications involve pattern classification systems where the samples are biometric data from a person under consideration, and need to be classified into categories whose nature depends on the specific task at hand. For the *identification* task, a new biometric sample is assigned to the most likely identity from a predefined set of identities. In this case, the classes are the possible identities. For the *verification* task, the system is probed with a biometric sample and a claimed identity. The goal is to decide whether the sample indeed corresponds to the claimed identity. Verification is thus a two-class decision problem where the classes correspond to the acceptance/rejection decision.

The focus of this dissertation is on *automatic face recognition* (AFR) [CWS95, WS99], a central area in biometrics. It is a very challenging task, as faces of different persons share global shape characteristics, while face images of the same person are subject to considerable variability, which might overwhelm the measured inter-person differences. Such variability is due to a long list of factors including facial expressions, illumination conditions, pose, presence or absence of eyeglasses and facial hair, occlusion and aging. Although much progress has been made over the past three decades, AFR is largely considered an open problem, as observed during the FERET evaluation [PMRR00] and the facial recognition vendor tests (FRVT) 2000 [BBP01] and 2002 [PGM$^+$03], and is a highly active research topic.

Data scarcity is often a problem of paramount importance in biometric applications. When a new user first enrolls in a system, only a few instances of the considered biometrics are typically captured in order to reduce the duration of enrollment and minimize inconvenience to the user (as well as maximize user cooperation). Hence, very little intra-class variability can be observed during the enrollment session. If only one sample is provided, intra-class variability is obviously impossible to assess. In the case of AFR, the image which is provided (or its representation) is thus directly used as a template and the likelihood $P(x|\omega_i)$ can be interpreted as a possible measure of similarity between the query and enrollment images. More generally, we note that the main issue is the ability to define a distance between images which is meaningful for the task at hand.

While most algorithms focus on the problem of representation, i.e. feature extraction, less attention has been given to the derivation and computation of an appropriate distance. For instance, the popular eigenfaces [TP91] and Fisherfaces algorithms [EC96, SW96, BHK97] employ low dimensional coding of face images. The distance between faces in the face subspace is based on simple metrics such as L_1, L_2, cosine and Mahalanobis distances [BSDG01] (c.f. appendix A). Combinations thereof such as the Mahalanobis-L_1, -L_2 and -cosine [BBTD03] have been proposed. Variations, such as the "Moon" [MP98] and "Yambor" [YDR00] distances for eigenfaces or the weighted euclidean distance for Fisherfaces [EC96, Zha99], have also been suggested. The candidate distance that yields the best results in a given set of experiments is simply chosen. However, it is often difficult to ascertain why one distance measure performs better than another. Moreover, as outlined in [BBTD03], it is even difficult to describe in precise terms what some of these distances are actually computing.

To define a meaningful distance, it is beneficial to formalize the *relationship* between observations of the same class, i.e., between face images of the same person. Due to the scarcity of data, we have to assume (or postulate) the existence of a *"universal" distance measure* that can be applied to different classes, i.e., that the intra-class variability is similar in the various classes. Thus, the parameters of the distance measure can be estimated from a larger training set which is not restricted to images of persons that are enrolled in the system. If I_t denotes the template image for class ω_i, I_q a query image and $\mathcal{R}$ the relationship between images of the same class, then the class-conditional probability is expressed as:

$$P(I_q|\omega_i) = P(I_q|I_t, \mathcal{R}) \tag{1.2}$$

A distance based on the above expression has already been used by the Bayesian intra/extra-personal classifier (BIC) [MP97, MWP98] that aims at estimating the distribution of image differences and by related approaches such as [MNP01]. Note that, while the Elastic Graph Matching (EGM) [LVB$^+$93] also defines a distance between face images, it does not make use of a probabilistic framework. However, other approaches related to EGM such as [VDR99], have made an attempt to define a probabilistic distance.

1.2 Contributions and Outline

In this dissertation, we introduce a novel measure of "distance" between images. This measure involves the estimation of the *set of possible transformations between face images*. The global transformation, which is assumed too complex for direct modeling, is approximated with a set of local transformations under a constraint imposing consistency between neighboring local transformations. The proposed local transformations and neighboring constraints are embedded within the probabilistic framework of a two-dimensional hidden Markov model (2-D HMM) in the case of discrete states and of the two-dimensional state-space model (2-D SSM) in the case of continuous states. This orginal approach will latter be referred to as the *probabilistic mapping with local transformations* (PMLT).

The outline of the dissertation is as follows. In the next two chapters, we provide a brief introduction to the discipline of biometrics and review the literature on AFR, respectively. The following 5 chapters correspond to original contributions:

- In chapter 4, we introduce the turbo hidden Markov model (T-HMM) and the turbo state-space model (T-SSM) as efficient approximations of the intractable 2-D HMM and 2-D SSM respectively. They consist of a set of inter-connected horizontal and vertical 1-D Markov chains that communicate through an iterative process. We attempt to provide efficient approximate answers to the three fundamental problems of HMM design [Rab89]. While the work on the T-HMM and the T-SSM is not the focus of this dissertation, it was necessary to make the face recognition algorithms developed in the course of this thesis tractable.

- In chapter 5, we first describe more extensively the proposed framework based on local transformations and neighboring coherence constraints. We then specialize it to the problem of face identification in the case of elastic facial distortions, due, for instance, to expressions using discrete grid transformations.

- In chapter 6, we specialize the proposed framework to the case of illumination variations using continuous feature transformations. We also consider the case where we model both facial distortions and illumination variations.

- In chapter 7, we consider the problem of clustering face images using the proposed measure of distance. The primary motivation is to partition the face space to reduce the number of comparisons when a query is made on a database that contains a large number of templates. We first address the issue of the update step, which is obvious for simple metrics such as the Euclidean distance, but which is much more challenging in the case of a complex measure of distance. We then address the problem of multiple clusters assignment of an observation.

- In chapter 8, we consider the problem of score normalization for robust identity verification. The issue is to model accurately the distribution of wrongful claims. However, for a distance such as the one introduced in this dissertation, we have two different ways to model such claims. The first approach models the set of possible transformations between face images of different persons. The second one models the distribution of impostors. Which of these two approaches to score normalization is the more robust is not obvious and this question is the focus of this chapter.

Finally, in chapter 9, we conclude this thesis.

2

An Introduction to Biometrics

2.1 Introduction

This chapter introduces the generic biometric system. Although such systems are based on a wide variety of technologies, much can be said about them. We start in section 2.2 by defining the term biometric and by discussing the properties of the ideal biometric. We then present in section 2.3 the two operational modes of a biometric system. In section 2.4, we describe the architecture of the generic biometric system. We also explain in section 2.5 how to evaluate the performance of a biometric system. Finally, we conclude by presenting potential applications of biometrics in section 2.6. Although multimodality is a very promising research direction to improve on the performance of individual biometric systems, it falls out of the scope of this dissertation and thus, it will not be discussed in this chapter.

2.2 Definition and Properties

There exists a wealth of applications that require reliable person identification or identity verification [JRP04]. The two traditional approaches to automatic person recognition, namely the *knowledge-based* approaches which rely on something that one knows such as a password, and the *token-based* approaches which rely on something that one has such as a badge, have obvious shortcomings: passwords might be forgotten or guessed by a malicious person while badges might be lost or stolen.

Biometrics recognition, which can be defined as "the automatic identification or identity verification of an individual based on *physiological* and *behavioral* characteristics" [Way00a] is an alternative to these traditional approaches as a biometric attribute is inherent to each person and thus cannot be forgotten or lost and might be difficult to forge. The face, the fingerprint, the hand/finger geometry, the iris or the retina are examples of physiological characteristics while the signature, the gait or the keystroke are examples of behavioral characteristics. It should be underlined that there is no clear cut between physiological and behavioral characteristics and that all biometric devices have both physiological and behavioral components [Way00b]. For instance, while the voice biometric is generally classified as behavioral it is dependent on physiological characteristics of the person under consideration such as his/her vocal tract length. On the other hand, every biometric, even those traditionally classified as physiological, has to be presented to the system and the presentation itself is a behavior.

Ideally a biometric should be [JRP04, Way00b]:

- *universal:* all the persons should have the characteristic.
- *permanent:* the characteristic should not vary over time.
- *distinctive:* samples corresponding to different persons should be as different as possible, i.e. the inter-class variability should be as large as possible.
- *robust:* samples corresponding to the same person should be as close as possible, i.e. the intra-class variability should be as small as possible.
- *accessible:* the sample should be easy to present to the sensor.
- *acceptable:* it should be perceived as non-intrusive by the user.

For instance the face biometric is universal, very easily accessible and it is generally considered as well accepted by users. However, it scores low on the permanence, distinctiveness and robustness.

2.3 Operational Mode

It is of utmost importance to distinguish between the two operational modes of a biometric system:

- In the *identification* mode, the user makes no claim of identity and the system has to perform a search over the entire database to find the most likely identity (one-to-many comparisons). A *close-set* is generally assumed which means that all the trials are supposed to be from persons who are registered in the database.

- In the *verification* mode, the user claims an identity and the system has to decide whether the sample indeed corresponds to the claimed identity (one-to-one comparison). An *open-set* is generally assumed, which means that the input samples may correspond to persons who are not registered in the database.

In the following, we use the generic term *recognition* when we do not want to make the distinction between identification and verification.

2.4 Architecture

Biometric applications involve typical pattern classification systems as explained in the introductory section (c.f. section 1.1). The architecture of the generic biometric system is depicted on Figure 2.1. It is composed of at least two mandatory modules, the *enrollment* and the *recognition* modules, and an optional one, the *adaptation* module.

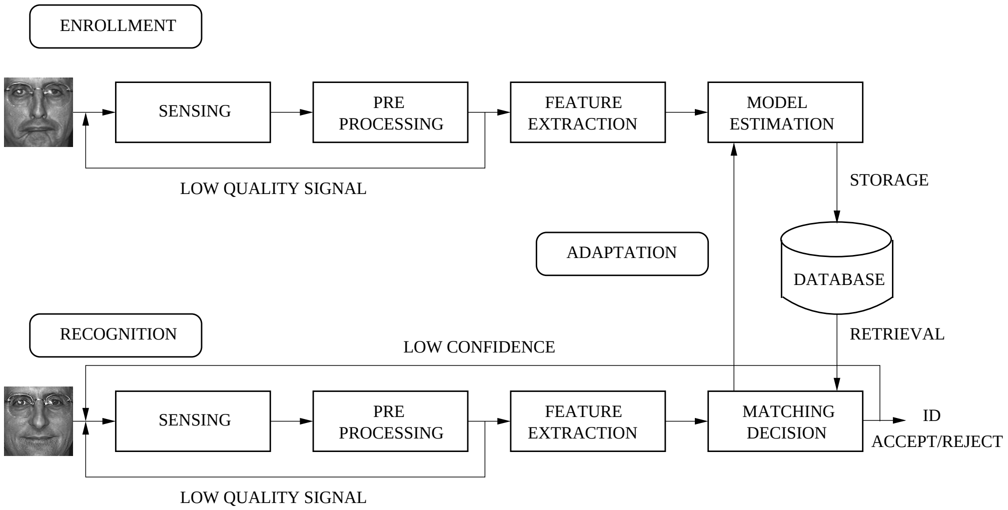


Figure 2.1: Architecture of a typical biometric system.

Enrollment is performed when a person registers in a biometric system. The typical stages of the enrollment are as follows. The biometric of interest is first captured by a *sensing* device. A series of *pre-processing* steps is then applied to the obtained signal. For the problem of AFR, such pre-processing operations may include face detection/segmentation, geometric or photometric normalization, etc. A very important component of many pre-processors is the *quality checker*: if the quality of the input signal is too poor, the system may require another sample from the user. Then features are extracted from the signal. The goal of the *feature extrac-*

tion step is to extract the unique features that characterize the considered person while discarding irrelevant information. Thus, feature extraction can be generally understood as a form of non-reversible compression. Finally, a *model can be estimated* with the available features. It is subsequently stored, for instance on a smart card or in a centralized database.

The first steps of the recognition are generally similar to the ones of the enrollment: sensing, pre-processing and feature extraction. Then one or multiple templates are retrieved from the database, depending on the operational mode. The extracted set of features is then compared with the template(s). Based on the outcome of the matching and the decision policy of the biometric system, a decision is taken. In the verification mode, the system can take an acceptance or rejection decision or, in a case of uncertainty, request additional data from the user.

During the enrollment phase, a user friendly system generally captures only a few instances of the biometric which is insufficient to describe with great accuracy the characteristics of this attribute. Moreover certain biometrics such as the face or the voice are not permanent. The goal of the adaptation module is hence to maintain or even improve the performance of the system over time by updating the model after each access to the system.

Note that the focus of this dissertation will be on the matching module but that we will also consider the issue of template retrieval.

2.5 Performance Evaluation

The technical performance evaluation of biometric systems is a very challenging issue which is too often overlooked. During the experimental evaluation we followed some guidelines, referred to as “best scientific practices”, for conducting technical performance testing [MW02]. Especially, we systematically used disjoint datasets to train our face classifier and to evaluate its performance. We never used the same persons to train and test the system. When carrying out verification experiments, we also assumed that the impostors were unknown to the system. That being said, we now briefly describe the performance measures of a biometric system and how to evaluate the uncertainty of these performance estimates.

2.5.1 Performance measures

As the identification and verification are two different operational modes, they require different measures of performance.

Identification

The *identification rate* is generally used to report the performance of a biometric system in the identification mode. If the top match corresponds to the identity of the person who submitted the query, then a success is declared. The identification rate is the percentage of such successful requests. Another measure of performance is the *cumulative match score*. A success is declared if the identity of the person who submitted the query is among the top N matches. The performance of a system can be represented by drawing the cumulative match score as a function of N.

When a search has to be performed over a very large database of templates, strategies have to be devised to reduce the number of comparisons. The traditional approach is to partition the templates into a number of datasets or classes which are meaningful for the biometric under consideration. For instance, for the fingerprint recognition problem, these sets correspond to the global patterns at the center of fingerprints such as the arch, the loop, the whorl, etc. When a query is probed, the first step consists in associating the query to one of the datasets and then to match the query with the templates associated with this class. When such a partitioning of the database is defined, two other measures of performance can be considered. The *penetration rate* can be defined as the expected proportion of the template data to be searched under the rule that the search proceeds through the entire partition, regardless of whether a match is found [MW02]. A binning error occurs if the template and a subsequent sample from the same user are placed in different partitions and the *binning error rate* is the expected number of such errors [MW02]. Obviously, the larger the number of classes, the lower the penetration rate but the greater the binning error rate.

Verification

When a biometric system works in the verification mode, it can make two types of errors. It can either reject a person that made a rightful identity claim, also referred to as a *client*, or accept a person that made a wrongful identity claim, also referred to as an *impostor*. The *false rejection rate* (FRR) is the expected proportion of transactions with truthful claims of identity that are incorrectly denied. This is also referred to as a *Type-I* error or a *miss*. The *false acceptance rate* (FAR) is the expected proportion of transactions with wrongful claims of identity that are incorrectly confirmed. This is also sometimes referred to as a *Type-II* error or a *false alarm*. Note that the FAR and FRR are defined over transactions. To avoid ambiguity with systems that allow multiple attempts or that have multiple templates per user, the *false match rate* (FMR) and the *false non-match rate* (FNMR) have been defined for a single comparison of a query against a single enrolled template.

However, as we will always consider the case of a single comparison against a single template, in this dissertation we do not make a distinction between the FAR and FMR on one hand and the FRR and the FNMR on the other hand.

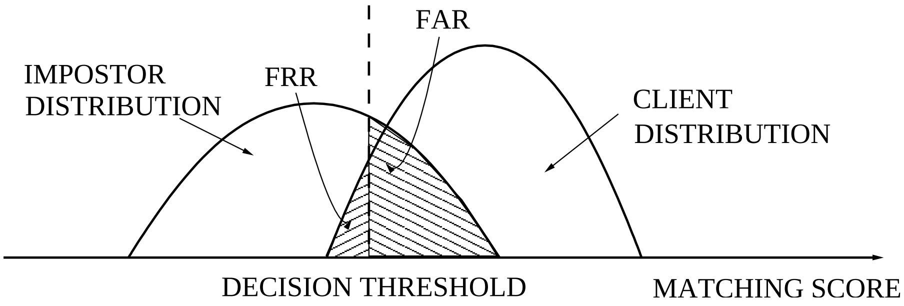


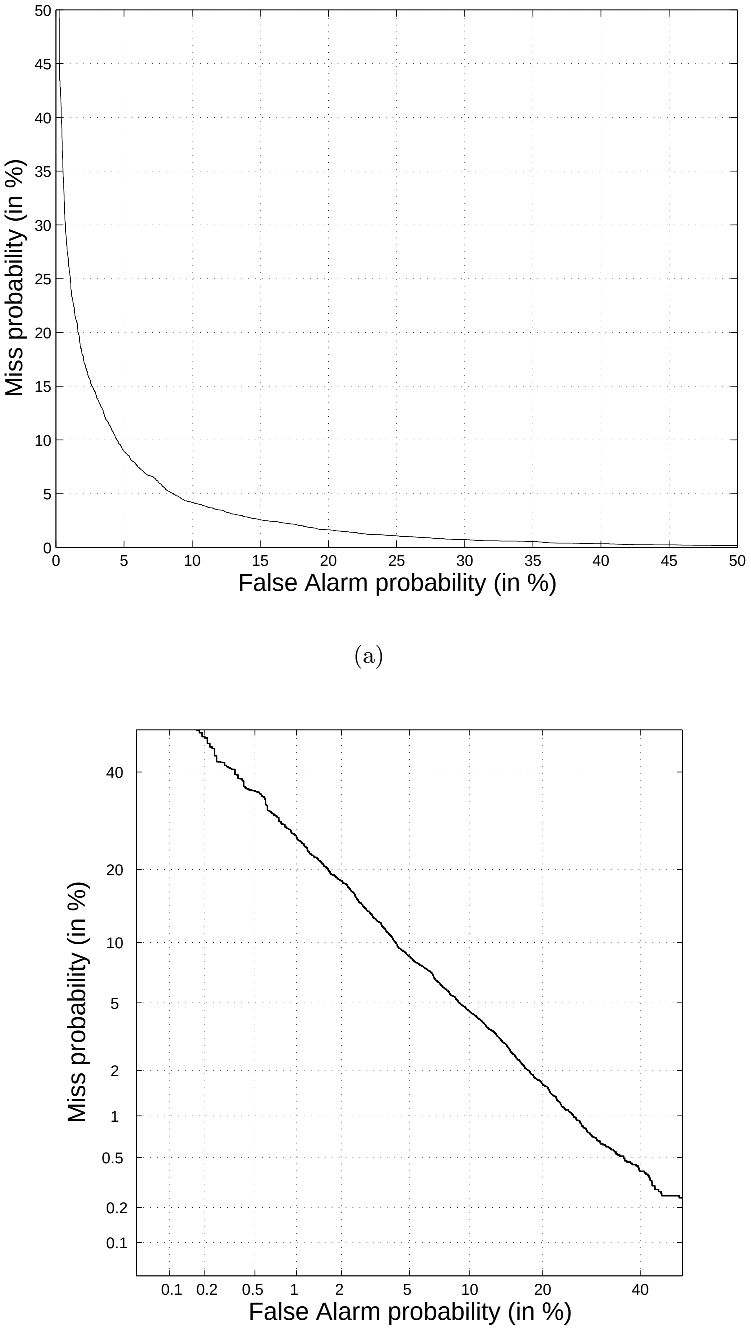
Figure 2.2: Impostor and client distributions for a typical biometric system.

To take an acceptance/rejection decision, a biometric system typically compares the matching score to a decision threshold θ. If the matching score falls below θ, then the claim is considered wrongful. If the matching score is higher than θ, then the claim is considered rightful. Obviously, the FAR and FRR are conflicting types of errors. The higher the decision threshold θ, the lower the FAR but the higher the FRR. On the other hand, the lower θ, the lower the FRR but the higher the FAR (c.f. Figure 2.2). The system performance can be depicted in the form of a *receiver operating characteristic* (ROC) curve. It plots, parametrically as a function of θ, the FAR against the FRR (c.f. Figure 2.3 (a)). For a given application, θ should be set according to the desired level of security.

Note also that the *detection error trade-off* (DET) curve has been proposed as an alternative to the ROC curve [MDK$^+$97]. In the DET curve a normal deviate scale is used to spread out the plot and distinguish better systems that have similar performances. If the client and impostor distributions are close to Gaussians, then the DET curve is almost linear (c.f. Figure 2.3 (b)). Moreover, a shift or the scaling of the client or impostor distributions can be readily interpreted as a shift or a tilt of the DET curve [ACLT00].

The *equal error rate* (EER), which corresponds to the point where $FAR = FRR$, is often used to report the performance of a system. A *decision cost function* (DCF) may also be used to summarize the performance of a system with one unique figure for a given threshold θ:

$$DCF(\theta) = C_{fr} P_{clt} P_{fr}(\theta) + C_{fa} P_{imp} P_{fa}(\theta) \tag{2.1}$$



(a)

(b)

Figure 2.3: (a) ROC curve and (b) DET curve for the same client and impostor distributions.

where C_{fr} and C_{fa} are respectively the costs of a false rejection and of a false acceptance, P_{clt} and P_{imp} are respectively the prior probabilities of client and impostor attempts and $P_{fr}(\theta)$ and $P_{fa}(\theta)$ are respectively the FRR and FAR for a given threshold θ. Note that the definition of the DCF is the direct interpretation of the conditional risk for the two-class decision problem (c.f. equation 1.1). The DCF is especially useful as an objective target for setting and measuring the goodness of a priori thresholds [RH01].

Interestingly, the FAR and FRR which are measures used in the *detection theory* can be directly related to the *recall* and *precision* which are *information retrieval* measures [SRSC01]. The recall is the proportion of relevant material retrieved from the database of templates. The precision is the proportion of retrieved templates which is relevant. Assuming that the probability P_t of each template (also referred to as the *richness* of the database) is uniform, then the recall R and the precision P can be written as functions of P_t, $P_{fr}(\theta)$ and $P_{fa}(\theta)$:

$$R(\theta) = 1 - P_{fr}(\theta) \tag{2.2}$$

$$P(\theta) = \frac{P_t \times (1 - P_{fr}(\theta))}{P_t \times (1 - P_{fr}(\theta)) + (1 - P_t) \times P_{fa}(\theta)} \tag{2.3}$$

A typical example of a precision versus recall curve is shown on Figure 2.4.

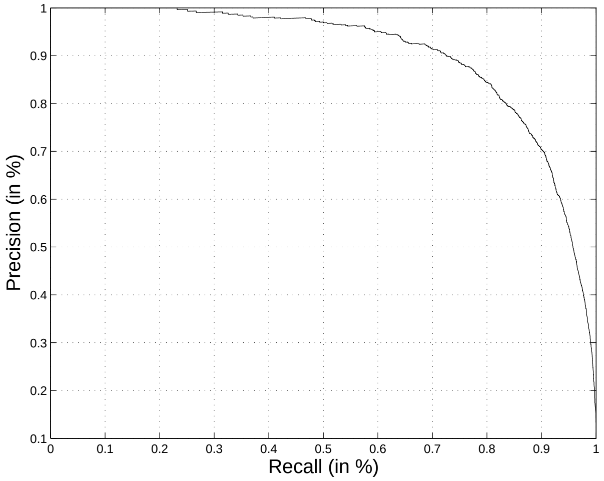


Figure 2.4: Precision versus recall curve. The client and impostor scores used to draw this curve are the same as the ones used in Figure 2.3. $P_t = 0.1$

2.5.2 Uncertainty of estimates

We briefly introduce the two types of uncertainties on the performance estimates: the *systematic errors* and *random errors* [MW02].

Systematic errors

Systematic errors are those due to a bias in the test procedure. This bias may be due to the fact that certain categories of the population, e.g. based on the age or the ethnicity, are either over- or under-represented. A solution to mitigate this bias, is to carry out experiments on as large a number of varied databases as possible. Another potential bias may arise if we train a system and assess its performance on the same database, i.e. if we consider perfectly matched conditions, which will generally give an overly optimistic performance estimate. To make sure also that our system is not too sensitive to a mismatch between the training and test conditions, whenever possible, we train and assess the performance of our system on different databases.

Random errors

Random errors which are due to the limited number of trials will reduce as the size of the test increases and they can be estimated using statistical tools.

It is possible to derive error bounds on the estimated error probability for a biometric system with a given confidence. Note that these confidence intervals do not represent a priori estimates of performance in different applications or with different populations. One generally assumes that trials are independent and that the error probability does not vary across the population [Way00b]. Thus, the probability distribution for the number of errors can be considered binomial. Let N be the number of trials, p be the error probability of the system and e be the number of observed errors. Then the maximum likelihood (ML) estimate of p is given by:

$$\hat{p} = \frac{e}{N} \tag{2.4}$$

If Np is sufficiently large (typically $Np > 10$), then the binomial distribution can be approximated with great accuracy by the normal distribution. Let σ^2 be the variance of the error rate. Under the assumption of normality, the true error rate will be with $1 - \alpha$ confidence in the interval:

$$[\hat{p} - z(1-\alpha/2)\sigma, \hat{p} + z(1-\alpha/2)\sigma] \tag{2.5}$$

where $z()$ is the inverse of the standard normal cumulative distribution. For instance, for a 95% confidence interval, we have $z(1-\alpha/2) = 1.96$ and for a 99% confidence interval, $z(1-\alpha/2) = 2.58$. As the standard deviation σ is unknown, it is replaced by its estimate $\hat{\sigma}$:

$$\hat{\sigma} = \sqrt{\frac{\hat{p}(1-\hat{p})}{N-1}} \tag{2.6}$$

If we want to compare two algorithms A_1 and A_2 on the same database, one may also determine directly whether the observed difference can be considered significant. If the errors are assumed independent, a simple approach is to perform *McNemar's test*. Let m be the number of samples for which A_1 made an error while A_2 was correct and let n be the number of samples for which A_2 made an error while A_1 was correct. Assuming for instance that $m < n$, then the probability that the observed difference happened purely by chance is given by:

$$P = 2 \sum_{i=n}^{m+n} \left(\begin{array}{c} m+n \\ i \end{array} \right) \left(\frac{1}{2} \right)^{i} \tag{2.7}$$

and thus A_1 can be declared to outperform A_2 with a $100 \times (1 - P)\%$ confidence.

2.6 Applications

In this section, we first give a classification of biometric applications and then provide a few examples of such applications.

2.6.1 Classifying applications

All applications can be partitioned according to the following seven categories [Way00b]:

- *cooperative/non-cooperative*: This terminology refers to the behavior of the deceptive user. In applications verifying a positive claim of identity, the deceptive user cooperates with the system in the attempt to be recognized. On the other hand, in systems verifying a negative claim of identity, the deceptive user will be non-cooperative in the attempt not to be recognized.

- *overt/covert*: A system is said to be overt if the user is aware that one of his biometrics is being measured. If not, the system is covert.

- habituated/non-habituated: This refers to the frequency of the interaction of a user with the biometric system.

- *Attended/non-attended*: This refers to whether the use of the biometric device is observed or guided by a person. Nearly all systems supervise at least the enrollment process.

- *Standard/non-standard* environment: This refers to the conditions of operation of a biometric system. For instance, outdoor systems will generally be considered as non-standard environment applications.

- *Public/private*: This refers to whether the users of the system will be customers or employees of the system management.
- *Open/closed*: This refers to whether the system will be required to exchange data with other biometric applications.

2.6.2 Example applications

There are mainly four areas of applications for biometrics: *access control*, *transaction authentication*, *law enforcement* and *personalization*.

Access control can be subdivided into two categories: *physical* and *virtual* access control [LS01]. The former one controls the access to a location. An example is the Immigration and Naturalization Service's Passenger Accelerated Service System (INSPASS) deployed in major US airports which enables frequent travelers to use an automated immigration system that authenticates their identity through their hand geometry [INS]. The latter one enables the access to a resource or a service such as a computer or a network.

Transaction authentication represents a huge market as it includes transactions at an automatic teller machine (ATM), electronic fund transfers, credit card and smart card transactions, transactions on the phone or on the Internet, etc. Mastercard estimates that a smart credit card incorporating finger verification could eliminate 80% of fraudulent charges [O'S97]. For transactions on the phone, biometric systems have already been deployed. For instance, the speaker recognition technology of Nuance is used by the clients of the Home Shopping Network and Charles Schwab [RH01].

Law enforcement has been one of the first applications of biometrics. Fingerprint recognition has been accepted for more than a century as a means of identifying a person. Automatic face recognition can also be very useful for searching through large mugshot databases.

Finally, personalization through person authentication is very appealing in the consumer product area. For instance, Siemens allows to personalize one's vehicle accessories, such as mirrors, radio station selections, seating positions, etc. through fingerprint recognition [SIE].

3

Face Recognition: a State of the Art

3.1 Introdcution

In this chapter we present a survey of the literature on AFR from still intensity images. An exhaustive review is out of the scope of this dissertation due to the large body of existing work. Indeed AFR has been a very active research topic for the past three decades. Thus we will focus on those approaches that we see as the most significant ones. The reader can also refer to other surveys [CWS95, WS99, Gru00].

As explained in the previous chapter, a typical AFR system first detects and segments a face, then extracts features and finally performs a matching with one or multiple templates. In this chapter, we will not consider the problem of face detection or segmentation and the interested reader can refer to [HL01, YKA02]. Instead, we will focus on the core of the pattern recognition system: the *feature extraction* and the *classification*. However, we will not review these two components separately (as done for instance in [CWS95]) since the classifier is generally heavily dependent on the extracted features.

This chapter will be split into two parts. In the first one, we will review *global approaches* to AFR, i.e. the ones that consider the face as a whole. In the second part, we will review *local approaches*, i.e. the ones that consider local features. While the latter approaches are generally more robust to variabilities in the face appearance such as rotation of the head, occlusion and gross variations due for instance to the presence or absence of facial hair, they also generally require significantly more computation than the former ones.

During this review, we will also show that, for both global and local approaches, the vast majority of the research has focused on the issue of face representation but that relatively little effort has been devoted to the matching problem.

3.2 Global Approaches

In this section we review those approaches to AFR that consider the face pattern as a whole. We first describe the basic *correlation* approach. We then present approaches based on the *singular value decomposition (SVD)*. Next we describe four classes of *subspace* approaches: the popular *eigenfaces* and *Fisherfaces*, the subspace approaches that go *beyond the second order statistics* and the *Bayesian intra-/extra-personal criterion (BIC)*. Then we will briefly present the *matching pursuit filters*. Also we will consider *neural networks* which have been used for both feature extraction and classification. Finally, we will very briefly present the *support vector machines (SVM)*.

3.2.1 Correlation

Although correlation-based methods are now rarely used, they are still very interesting to review because of their simplicity and because many algorithms that were developed for AFR can be seen as extensions of this basic method.

The technique consists in using the whole gray-level image as the template. Although the Euclidean distance could be used directly to compute the matching score between two face images, the cross-correlation is often preferred [Bar81, BP93] as it rescales the template and query images energy distributions so that their averages and variances match, thus making the score more robust to different ambient illuminations or characteristics of the digitizing device. In [BM95] Brunelli and Messelodi compare the standard correlation coefficient to three measures of similarity which are based on the L_2 and L_1 norms. These estimators were investigated both statistically, in the estimation of the correlation parameter of a bivariate normal distribution, and experimentally on a face recognition task. The similarity measures based on the L_1 norm were shown to be more robust.

To compensate for scale variation Burt proposes a hierarchical approach [Bur88]. Such an approach also speeds-up the computationally intensive correlation estimation. This work was also extended to the recognition of faces under varying pose by Beymer [Bey94]. When a query image is probed, features such as the eyes, nose and mouth are first located through a template matching approach and the pose of the face is then estimated (in [Bey94] the pose estimator can only distinguish between face turned left and right). Then, the query image is matched with the template images corresponding to the same general pose (left/right).

3.2.2 Singular value decomposition

In [Hon91] image features are divided into four classes: visual features, statistical features of pixels, transform coefficient features and *algebraic* features. Algebraic features are *intrinsic* to the image but *not necessarily visible.* The singular values (SVs) of an image, considered as a matrix of pixels, are examples of algebraic features.

Let us remind that if A is a real $m \times n$ matrix of rank r, then there exists two orthonormal matrices $U = [u_1, u_2, ..., u_m]$ of size $m \times m$ and $V = [v_1, v_2, ...v_n]$ of size $n \times n$ and a diagonal matrix $\Sigma = \text{diag}(\lambda_1, \lambda_2, ...\lambda_r, 0, ..., 0)$ of size $m \times n$ such that:

$$A = U \Sigma V^T \tag{3.1}$$

The λ_i's are the singular values of A. Formula 3.1 can be rewritten as:

$$A = \sum_{i=1}^{r} \lambda_i u_i v_i^T \tag{3.2}$$

which can be understood as a decomposition of the image A on the orthogonal basis $u_i v_i^T$. The larger the corresponding singular value, the greater the contribution to the reconstruction of the image.

The foundation of the characterization of images with their SVs is based on the properties of the SVD. We cite here the ones that are particularly relevant to face recognition [Hon91]:

- stability properties and, hence, insensitivity to image noise or small changes of gray values incurred from different illumination conditions;
- proportionality to the variance of the image intensity;
- invariance to rotation, translation and mirror transform.

Tian et al. recently argued in [TTWF03] that the SVs of an image contain only partial useful information about the face and that much information is carried by the orthonormal matrices U and V. This is directly linked to the fact that SVs represent information that is not necessarily visible. This claim was supported by a series of simple experiments. The authors swapped the SVs of different persons and reconstructed the corresponding image with good accuracy. The only noticeable difference was in the change of the gray level distributions. They even showed that face images could be reconstructed with the SVs of a non-facial image.

The idea of using the information contained in the orthonormal matrices appeared also in a much earlier paper [CLYW92] but without the justification provided by [TTWF03]. Let A and B be two matrices. Let $u_i v_i^T$ be the orthogonal basis of the SVD of A. If $\overline{\lambda}_i = u_i^T B v_i$, then $\overline{B} = \sum_i \overline{\lambda}_i u_i v_i^T$ is called the *projective image* of B on A. If $A_{i,j}$ is a set of training images for person i, the idea is to perform the SVD of the average image $\overline{A}_i$ to obtain the orthonormal matrices U_i and V_i. (small SVs are discarded in [CLYW92]). Training and test face images are then projected on U_i and V_i.

3.2.3 Eigenfaces

In this section, we first present the basic eigenfaces approach and we then consider one of its extension to multiple spaces.

The basic eigenface approach

Eigenfaces are based on the notion of dimensionality reduction. Kirby and Sirovich first outlined that the dimensionality of the face space, i.e. the space of variation between images of human faces, is much smaller than the dimensionality of a single face considered as an arbitrary image [KS90]. 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. The idea of the *Principal Component Analysis* (PCA) [Jol86], also known as the *Karhunen-Loeve Transform* (KLT), is to find the subspace which best accounts for the distribution of face images within the whole space.

Let $\{x_1, ..., x_N\}$ be a set of reference or training faces, $\overline{x}$ be the average face and $\delta_i = x_i - \overline{x}$. δ_i is sometimes referred to as a *caricature* image. Finally, if $\Delta = [\delta_1, ..., \delta_N]$, the *scatter* matrix S is defined as:

$$S = \sum_{i=1}^{N} \delta_i \delta_i^T = \Delta \Delta^T \tag{3.3}$$

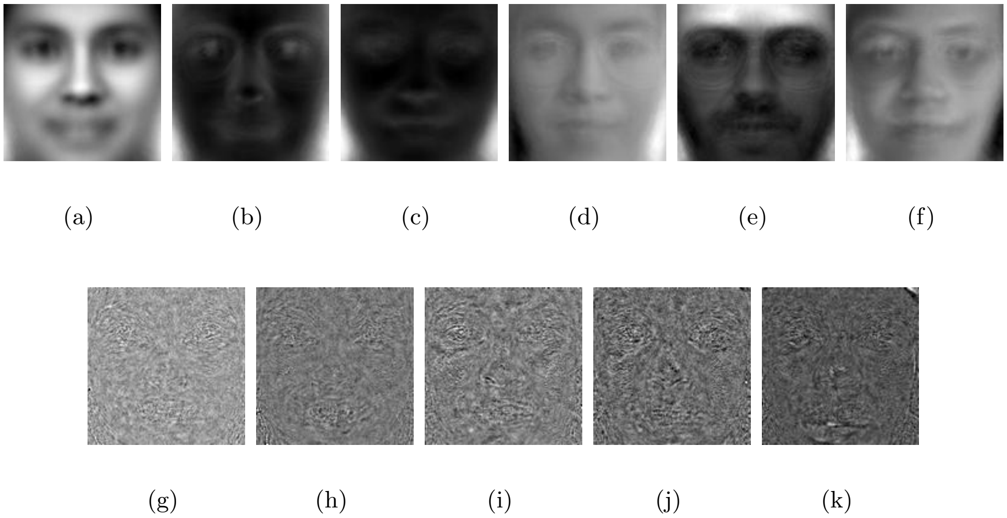
The optimal subspace P_{PCA} is the one that maximizes the scatter of the projected faces:

$$P_{PCA} = \arg\max_P |PSP^T| \tag{3.4}$$

where $|.|$ is the determinant operator. The solution to problem 3.4 is the subspace spanned by the eigenvectors $[e_1, e_2, ...e_K]$ corresponding to the K largest eigenvalues of the scatter matrix S:

$$Se_k = \lambda_k e_k \qquad \text{k= 1,...,K} \tag{3.5}$$

As the number of images in the training set is generally lower than the dimension of the image space, i.e. the number of pixels in an image, the number of non-zero eigenvalues is $N - 1$. Due to the size of the scatter matrix S, the direct estimation of its eigenvalues and eigenvectors is difficult. They are generally estimated either through a SVD of the matrix Δ or by computing the eigenvalues and eigenvectors of $\Delta^T \Delta$. It should be underlined that eigenfaces are not themselves usually plausible faces but only directions of variation between face images (c.f. Figure 3.1).


(a) (b) (c) (d) (e) (f)

(g) (h) (i) (j) (k)

Figure 3.1: (a) Eigenface 0 (average face), (b)-(f) eigenfaces 1 to 5 and (g)-(k) eigenfaces 995 to 999 as estimated on a subset of 1,000 images of the FERET face database.

Each face image x_i is represented by a point w_i in the K-dimensional space: $w_i = [w_i^1, w_i^2, ...w_i^K]^T = P_{PCA} \times \delta_i$. Each coefficient w_i^k is the projection of the face image on the k-th eigenface e_k and represents the contribution of e_k in reconstructing the input face image. PCA guarantees that, for the set of training images, the mean-square error introduced by truncating the expansion after the K-th eigenvector is minimized.

The eigenfaces were applied by Turk and Pentland to the problem of AFR [TP91]. 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 image yielding the smallest distance. The distance between faces in the face subspace is generally based on simple metrics such as L_1 (city-block), L_2 (Euclidean), cosine and Mahalanobis distances [BSDG01] (c.f. appendix A). Combinations thereof such as the Mahalanobis-L_1, -L_2 and -cosine [BBTD03] have been proposed. Variations, such as the "Moon" [MP98] and "Yambor" [YDR00] distances have also been suggested. Based on the work of others [BBTD03] and our own experience, the Mahalanobis-cosine distance is the one that yields the best results. However, the reasons for this superior performance compared to the other metrics are not clear.

Brunelli and Poggio argued that, from a theoretical point of view, eigenfaces cannot achieve better results than the simple correlation based method but that it may be able to reach a comparable performance with a much smaller computational effort [BP93]. Note however that, as the feature extraction makes use only of the first few eigenfaces, some irrelevant information contained in the higher eigenfaces is discarded (c.f. Figure 3.1) and thus, in practice eigenfaces can outperform the correlation approach.

Extension to multiple spaces

The eigenfaces approach was extended to multiple spaces. When a large amount of training data is available, one can either pool all the data to train one unique eigenspace or split the data into multiple training sets and train multiple eigenspaces. The first approach which was introduced by Murase and Nayar, and which was inspired by research on general 3-D object recognition, is referred to as the *parametric approach* [MN93]. The idea is to take pictures of objects under different views and lightning conditions and to build a *universal eigenspace* with the whole set of images.

The alternative method, which is known as the *view-based approach*, consists in building a separate eigenspace for each possible view. The approach followed by Pentland et al. in [PMS94] is the following one. For each new target image, its orientation is first estimated by projecting it on each eigenspace and choosing the one that yields the smallest distance from face to space. One can use pruning strategies to reduce the computational load incurred from projecting the face image in multiple eigenspaces. The performance of the parametric and view-based approaches were compared in [PMS94] and the latter one performs better. The problem with the view-based approach is that it requires large amounts of *labeled* training data to train each separate eigenspace.

More recently *mixtures of principal components* (MPC) were proposed by Kim et al. [KKB02] and by Turaga and Chen [TC02] to extend the traditional PCA. An iterative procedure based on the EM algorithm was derived in both cases to train automatically the mixture of principal components. However, while [KKB02] represents a face by the best set of features corresponding to the closest set of eigenfaces, in [TC02] a face image is projected on each component eigenspace and these individual projections are then linearly combined. [TC02] tested MPC on a database of face images that exhibit large variabilities in poses and illumination conditions. Each eigenspace converges *automatically* to varying poses and the first few eigenvectors of each component eigenspace seem to capture lightning variations.

3.2.4 Fisherfaces

In this section, we first present the basic Fisherfaces approach and we then consider possible extensions.

The basic Fisherfaces approach

While PCA is optimal with respect to data compression [KS90], in general it is sub-optimal for a recognition task. Actually, PCA confounds *intra-personal* and *extra-personal* sources of variability in the total scatter matrix S. Illumination conditions are a source of large variabilities between face images of the same person and the extra-personal variability due to lightning can even be larger than the intra-personal variability. Thus eigenfaces can be contaminated by non-pertinent information. It has been suggested that by throwing out the first several (typically 3) principal components, the variation due to lightning would be reduced [BHK97]. However, it is unlikely that these components model solely variation in lightning and relevant information might also be discarded.

However, this does not necessarily mean that one should give up on linear dimensionality reduction techniques. The argument in [BHK97] is the following one: all the images of a *Lambertian surface* [1] without self-shadowing and taken from a fixed view point lie in a 3-D linear sub-space. Therefore, under the ideal conditions listed above, the classes are linearly separable. This is a strong argument in favor of using linear methods for dimensionality reduction in the AFR problem, at least when one is concerned with compensating for illumination variations.

For a classification task, a dimension reduction technique such as *Fisher's Linear Discriminant* (FLD) should be preferred to PCA. The idea of FLD is to select

[1] A Lambertian surface is a surface whose radiance is independent of direction, which means that it adheres to Lambert's cosine law.

a subspace that maximizes the ratio of the inter-class variability and the intra-class variability. Whereas PCA is an *unsupervised* feature extraction method, FLD uses the category information associated with each training observation and is thus categorized as *supervised*. The application of the FLD to the problem of AFR is mainly due to Etemad and Chellappa [EC96] Swets and Weng [SW96] and Belhumeur et al. [BHK97].

Let $x_{i,k}$ be the k-th picture of training person i, N_i be the number of training images for person i, $\overline{x_i}$ be the average face for person i and C be the number of persons in the training set. S_B and S_W, respectively the *between-* and *within-class scatter matrices*, are given by:

$$S_B = \sum_{i=1}^{C} N_i(\overline{x_i} - \overline{x})(\overline{x_i} - \overline{x})^T \tag{3.6}$$

$$S_W = \sum_{i=1}^{C} \sum_{k=1}^{N_i} (x_{i,k} - \overline{x_i})(x_{i,k} - \overline{x_i})^T \tag{3.7}$$

The optimal subspace P_{LDA} is the one that maximizes the between-scatter of the projected face images while minimizing the within-scatter of the projected faces:

$$P_{LDA} = \arg\max_{P} \frac{|PS_BP^T|}{|PS_WP^T|} \tag{3.8}$$

The solution to equation 3.8 is the sub-space spanned by $[e_1, e_2, ...e_K]$, the generalized eigenvectors corresponding to the largest eigenvalues of the generalized eigenvalue problem:

$$S_Be_k = \lambda_k S_W e_k \qquad \text{k= 1,...,K} \tag{3.9}$$

However, if the dimensionality of the feature space is higher than the number of training individuals, which is generally the case, then S_W is singular and this principle cannot be applied in a straightforward manner. To overcome this issue, generally one first applies PCA to reduce the dimension of the feature space and then performs the standard FLD [BHK97, SW96]. The eigenvectors that form the discriminant subspace are often referred to as *Fisherfaces* [BHK97]. In [SW96], the space spanned by the first few Fisherfaces is called the *most discriminant features* (MDF) classification space while PCA features are referred to as *most expressive features* (MEF).

The distance between faces in the face subspace is generally based on simple metrics as is the case for the eigenfaces approach. However, as the eigenvalues associated to each eigenvector are directly related to the discriminatory power of the

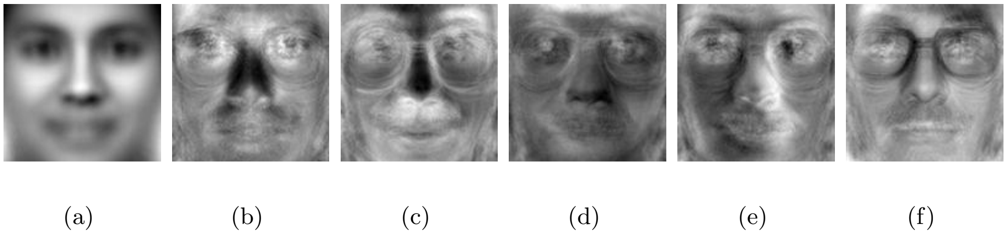


Figure 3.2: (a) Fisherface 0 (average face) and (b)-(f) Fisherfaces 1 to 5 as estimated on a subset of 1,000 images of the FERET face database.

considered dimension, this information can be used. For instance, Zhao suggested a weighted Euclidean distance where the weight associated to dimension k is λ_k^α and α is a parameters which has to be hand-tuned [Zha99] (c.f. appendix A). Also, in [KLM00] the *gradient direction metrics* was proposed for the verification problem. The distance between a probe image and a model is measured in the gradient direction of the a posteriori probability of the hypothesized client identity. A mixture of Gaussian distributions with identity covariance matrix is assumed as the density function of the possible impostors. In [SK04], the previous approach is extended to the case of a general shared covariance matrix for the components of the GMM.

Note that, while Fisherfaces usually perform significantly better than eigenfaces, Martìnez and Kak, showed experimentally that when the training dataset is small or when there is a significant mismatch between the training and test conditions, then eigenfaces can outperform Fisherfaces [MK01].

Alternative approaches

Other solutions to equation 3.8 were suggested.

The FLD procedure involves the simultaneous diagonalization of the two within- and between-class scatter matrices which is stepwise equivalent to two operations [Fuk90]: first whitening the within-class scatter matrix, and second applying PCA on the between-class scatter matrix using the transformed data. As during whitening the eigenvalues of the within-class scatter matrix appear in the denominator, the small eigenvalues cause the whitening step to fit for irrelevant variations and thus lead to poor generalization. The Enhanced FLD Model (EFM) proposed by Liu and Wechsler improves on the basic FLD by retaining an optimal number of components in the reduced PCA space [LW02]. The goal is to maintain a balance between the need that the selected eigenvalues account for most of the spectral energy and the requirement that the eigenvalues of the within-class scatter matrix (in the reduced

PCA space) are not too small.

Chen et al. suggested that the null space of S_W, i.e. the space spanned by the vectors x that satisfy $S_W x = 0$, carries most of the discriminative information as in such a space perfect classification of the training data can be performed [CLK+00]. Therefore, the construction of a discriminant subspace is done in two steps: 1) the null space of S_W is estimated 2) the vector set that maximizes the between-class scatter matrix of the transformed samples (in the null space of S_W) are chosen.

Yang and Yang also propose to make use of the information contained in the null space of S_W [YY03]. Let $\tilde{S}_W$ and $\tilde{S}_B$ the transformed versions of S_B and S_W in the PCA space. In the null space of $\tilde{S}_W$, the Fisher criterion is replaced by the maximization of the scatter matrix in the projection space which is similar to the approach of [CLK+00]. In the orthogonal complement, the projection of $\tilde{S}_W$ is positive and the optimal discriminant vectors can be directly extracted from the Fisher criterion.

3.2.5 Beyond the second order statistics

While the eigenfaces and Fisherfaces have been successfully applied to AFR, both approaches rely on *second order statistics*. However, it has been argued that in a task such as AFR, much of the important information is contained in the high-order statistics of the images [BS97, Yan02]. Therefore, a representation where the high-order statistics are decorrelated may be more powerful for AFR than one in which only the second order statistics are decorrelated. In this section, we will thus consider two such representations: *kernel* approaches and *independent component analysis* (ICA).

Kernel approaches

The basic idea underlying kernel approaches is to transform the vector space into a higher dimensional space. The justification stems from Cover's theorem which states that non-linearly separable patterns in an input space are linearly separable with high probability if the input space is transformed non-linearly to a high dimensional feature space [Hay99]. One can thus extend the eigenfaces and Fisherfaces approaches to *kernel eigenfaces* and *kernel Fisherfaces* [Yan02].

Let $\{x_i, i = 1...N\}$ be a set of training vectors and let Φ be the transformation from the input space to the higher dimensional space. If S^Φ is the total scatter matrix in the transformed space, then the eigenvalue problem for the kernel eigenface is:

$$S^{\Phi} e_k^{\Phi} = \lambda_k^{\Phi} e_k^{\Phi} \qquad \text{k= 1,...,K} \tag{3.10}$$

As the vectors e_k^{Φ} lie in the span of $[\Phi(x_1), ...\Phi(x_N)]$, there exists coefficients $\alpha_{k,i}$'s such that:

$$e_k^{\Phi} = \sum_{i=1}^{N} \alpha_{k,i} \Phi(x_i) \tag{3.11}$$

Let the kernel function $k(.,.)$ and the $N \times N$ kernel matrix K be defined as:

$$k(x_i, x_j) \quad = \quad \Phi(x_i)^T \Phi(x_j) \tag{3.12}$$

$$K_{ij} \quad = \quad k(x_i, x_j) \tag{3.13}$$

The eigenvalue problem turns into:

$$N \lambda_k \alpha_k = K \alpha_k \tag{3.14}$$

where α_k is the vector $[\alpha_{k,1}, ..., \alpha_{k,N}]^T$. The projection of a new vector $\Phi(x)$ into the kernel eigenfaces space is obtained in the following manner:

$$\Phi(x)^T e_k^{\Phi} = \sum_{i=1}^{N} \alpha_{k,i} \Phi(x)^T \Phi(x_i) = \sum_{i=1}^{N} \alpha_{k,i} k(x_i, x) \tag{3.15}$$

One of the most interesting properties of kernel approaches is that all the calculation can be achieved by using $k(.,.)$ only and we do not need to know Φ. In [YAK00], Yang, et al. use polynomial kernels. The conventional eigenfaces approach is a special case of kernel eigenfaces with a polynomial kernel of first order. Note that the distance computation in the kernel eigenface space is based on the same simple metrics used for the eigenfaces. The Fisherfaces approach can be extended in a similar manner [Yan02].

Independent component analysis

Independent component analysis (ICA) is a generalization of PCA which decorrelates the high-order statistics of images. In [BS97] Bartlett and Sejnowski calculate the independent components through an unsupervised learning algorithm that maximizes the mutual information between the input and the output of a non-linear transformation. Various approaches for selecting a subset of the independent components were discussed. The best performance is obtained when the components with the highest between-class to within-class variability are chosen. The Euclidean distance is used to estimate the distance between face representations in the lower dimensional space. In [BS97] the authors reported that ICA outperformed PCA, especially in the case of pose or lighting variations. However, in [Mog99] Moghaddam did not report any improvement for ICA over PCA.

3.2.6 The Bayesian intra-/extra-personal criterion

In [MP97], Moghaddam and Pentland argue in favor of a probabilistic measure of similarity in contrast to simpler measures of similarity such as the Euclidean distance. The focus of [MP97, MWP98, Mog02] is on modeling the difference δ between face images. The observed variability can be explained by two mutually exclusive classes of variability: the intra-personal variability Ω_I (equivalent to our notation $\mathcal{R}$) and the extra-personal variability Ω_E. The chosen measure of similarity between two face images is $P(\Omega_I|\delta)$ which, using Bayes rule, can be evaluated as follows:

$$P(\Omega_I|\delta) = \frac{P(\delta|\Omega_I)P(\Omega_I)}{P(\delta|\Omega_I)P(\Omega_I) + P(\delta|\Omega_E)P(\Omega_E)} \tag{3.16}$$

This measure of similarity is referred to as the MAP classifier in [MWP98].

The difference between face images of the same person is assumed to be a normally distributed random variable:

$$P(\delta|\Omega_I) = \frac{1}{(2\pi)^{N/2}|S|^{1/2}} \exp\left\{-\frac{1}{2}\delta^T S^{-1}\delta\right\} \tag{3.17}$$

Due to the high dimensionality of δ the direct estimation of the parameters of this probability density function, i.e. of the covariance matrix S, is difficult. Moreover, estimating $P(\delta|\Omega_I)$ can be very computationally intensive. In the following, we summarize the optimal approach for estimating high-dimensional Gaussian densities described in [MP97].

Using a PCA, one can write $\Lambda = \Phi^T S \Phi$ where $\Phi = [\Phi_1, ...\Phi_N]$ is the matrix of eigenvectors of S (c.f. Figure 3.3), and $\Lambda = \text{diag}\{\lambda_1, ..., \lambda_N\}$ is the corresponding diagonal matrix of eigenvalues. If $y = [y_1, ...y_N]^T$ is the projection of δ on the basis defined by Φ^T, i.e. $y = \Phi^T\delta$, and if we denote $d(\delta) = \delta^T S^{-1}\delta$, we have:

$$d(\delta) = y^T\Lambda^{-1}y = \sum_{i=1}^{N} \frac{{y_i}^2}{\lambda_i} \tag{3.18}$$

The image difference space can be split into a principal subspace $F = \{\Phi_1, ...\Phi_M\}$ and its orthogonal complement $\overline{F} = \{\Phi_{M+1}, ..., \Phi_N\}$. $d(\delta)$ can thus be separated into $d_F(\delta) + d_{\overline{F}}(\delta)$ and $P(\delta|\Omega_I)$ into $P_F(\delta|\Omega_I)P_{\overline{F}}(\delta|\Omega_I)$. While the terms of $d_F(\delta)$ are easy to compute, the terms of $d_{\overline{F}}(\delta)$ are difficult to estimate due to the high dimensionality of the problem. Hence $d_{\overline{F}}(\delta)$ (resp. $P_{\overline{F}}(\delta|\Omega_I)$) is approximated with $\hat{d}_{\overline{F}}(\delta)$ (resp. $\hat{P}_{\overline{F}}(\delta|\Omega_I)$):

$$d_{\overline{F}}(\delta) = \sum_{i=M+1}^{N} \frac{y_i^2}{\lambda_i} \approx \hat{d}_{\overline{F}}(\delta) = \frac{1}{\rho} \sum_{i=M+1}^{N} y_i^2 = \frac{\epsilon^2(\delta)}{\rho} \tag{3.19}$$

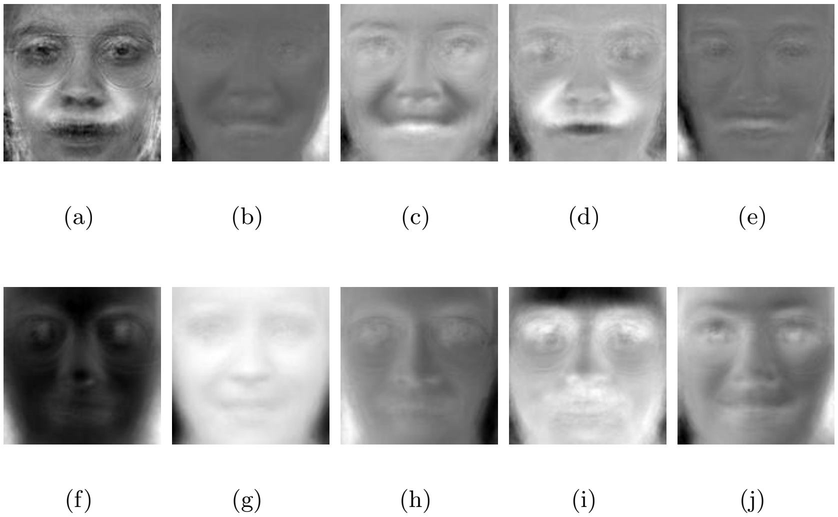


Figure 3.3: (a)-(e) Intra-personal eigenfaces and (f)-(j) extra-personal eigenfaces as estimated on a subset of 1,000 images of the FERET face database.

where ρ is a constant and $\epsilon^2(\delta)$ is the distance of δ to the face difference space which can be computed as follows:

$$\epsilon^2(\delta) = ||\delta||^2 - \sum_{i=1}^{M} y_i^2 \tag{3.20}$$

The optimal value of ρ which minimizes the Kullback-Leibler divergence between $P(\delta|\Omega_I)$ and its estimate $P_F(\delta|\Omega_I)\hat{P}_{\overline{F}}(\delta|\Omega_I)$ is given by:

$$\rho = \frac{1}{N-M} \sum_{i=M+1}^{N} \lambda_i \tag{3.21}$$

Using all the previous derivations, we finally obtain:

$$P(\delta|\Omega_I) \approx \frac{\exp\left\{-\frac{1}{2}\sum_{i=1}^{M}\frac{y_i^2}{\lambda_i}\right\}}{(2\pi)^{M/2}\prod_{i=1}^{M}\lambda_i^{1/2}} \frac{\exp\left\{-\frac{1}{2}\frac{\epsilon^2(\delta)}{\rho}\right\}}{(2\pi\rho)^{(N-M)/2}} \tag{3.22}$$

In the case where the distribution of the difference images in the F-space is multimodal, the estimation of $P_F(\delta)$ can be improved by the use of a mixture of Gaussians instead of a single Gaussian.

Assuming that the distance between face images of different persons is also Gaussian [Mog02], a similar approach can be used to estimate $P(\delta|\Omega_E)$. However, a simple ML formulation which makes use only of the intra-personal similarity $P(\delta|\Omega_I)$ is often preferred to the more complex MAP classifier as it requires twice as less computation with very little, if no, degradation of the performance [?]. The reason for the very small observed difference between the ML and MAP classifiers is explained in [WT03]. As the extra-personal subspace is similar to the PCA eigenspace, it does not contribute significantly to separating intra- and extra-personal variabilities.

While the approach devised in [MP97, MWP98] uses a simple intensity difference, Moghaddam, et al. have shown experimentally in [MNP01] that a deformable XYI-warping method for obtaining pixel to pixel correspondence does lead to an improved representation for the "difference" between face images.

3.2.7 Matching pursuit filters

The matching pursuit filter technique is an adaptive wavelet expansion. A wavelet expansion of an image is said to be adaptive if the choice of the wavelet basis depends on the image under consideration. The original matching pursuit idea of Mallat and Zhang uses a greedy heuristic to iteratively construct a best-adapted decomposition of a function f [MZ93]. At each iteration, the next wavelet in the expansion is chosen by minimizing the error between the original image and the reconstructed image. Let $\mathcal{D}$ be a dictionary of non-necessarily orthogonal wavelets. Let $R^i(f)$ denote the residue of f after the i-th iteration with $R^0(f) = f$. Then at the i-th iteration, g_i is chosen such that:

$$g_i = \arg\max_{g \in \mathcal{D}} |R^{i-1}(f)^T g_i| \text{ , for } i \geq 1 \tag{3.23}$$

and the residual is updated in the following manner:

$$R^i(f) = R^{i-1}(f) - \alpha_i g_i \tag{3.24}$$

where α_i is the projection of the residual image $R^{i-1}(f)$ onto the basis element g_i:

$$\alpha_i = R^{i-1}(f)^T g_i \tag{3.25}$$

In [Phi98], Phillips applies the matching pursuit filter to the problem of face detection and recognition. To extend this technique to the problem of pattern recognition the right hand side in equation 3.23 is replaced with a cost function C_g which allows 1) the simultaneous expansion of multiple templates and 2) to incorporate knowledge of the given pattern recognition problem. Indeed, while for the face detection problem, the cost function measures how well coefficient vectors cluster, for the AFR problem, we search for a basis that separates the coefficients as much as possible.

For his expansion, Phillips uses a dictionary composed of 2-D directional wavelets. The dictionary is derived from the second partial derivatives of Gaussian densities and their Hilbert transforms, which were selected because they are directional edge detectors. High frequency wavelets were excluded to reduce the effect of high-frequency noise. Low-frequency wavelets were also excluded for computational considerations but also to avoid encoding information in the background. Let $\{x_1, ..., x_N\}$ be the set of templates used to design the matching pursuit filter, let α_i^k be the coefficient estimated for template k at the i-th iteration and $\Lambda_{i-1} = \{\alpha_1^1, ..., \alpha_{i-1}^1, ..., \alpha_1^N, ..., \alpha_{i-1}^N\}$. The cost function for the identification task is:

$$\begin{aligned} C_g(R^{i-1}(x_1), ..., R^{i-1}(x_N), \Lambda_{i-1}) = & -\sum_k \max_{j \neq k} d_\theta(k,j) \\ & +\lambda \sum_k ||[\alpha_1^k, ..., \alpha_{i-1}^k, R^{i-1}(x_k)^T g]|| \end{aligned} \quad (3.26)$$

where the function $d_\theta(k,j)$ equals the cosine of the angle between the vectors $[\alpha_1^k, ..., \alpha_{i-1}^k, R^{i-1}(x_k)^T g]$ and $[\alpha_1^j, ..., \alpha_{i-1}^j, R^{i-1}(x_j)^T g]$. Thus, the first term in C_g forces the coefficients vectors to separate and the second term searches for a set of coefficients vectors with the largest average magnitude. The parameter λ sets the relative importance of the two terms. While the focus in [Phi98] is on the case where only one example image is available per person, the extension to multiple images per person is straightforward.

As for the similarity measure between two faces, a simple angle distance between their coefficient vectors was chosen.

3.2.8 Neural networks

Artificial Neural Networks (ANNs) have been applied to AFR for both feature extraction and classification.

Feature extraction

While the popular back-propagation (BP) neural net may be trained to recognize face images, the direct application of this principle is often impossible due to the size of the input features [CF90, LGTB97, ZYL97] as it would lead to a complex network which would be difficult to train. Therefore, before classification is performed, a dimension reduction technique should be applied.

To perform dimension reduction, Cotrell and Fleming use in [CF90] a Multi Layer Perceptron (MLP) that works in the *auto-association* mode where the input units communicate their values to the output units through a hidden layer. While the first part of the network compresses the n redundant measurements into a smaller number of characteristics p (usually $p \ll n$) which should convey the essential information, the second part of the network works in the opposite way, and uses the compressed information to regenerate the n original inputs as accurately as possible. It was proved that the optimal solution of the MLP is, under the best circumstances (*linearity* of the network), strictly equivalent to the Karhunen-Loeve Transform (KLT) and that the network projects the input onto the sub-space spanned by the first p principal components [Bou00].

Lawrence et al. suggested in [LGTB97] another approach based on a *Self-Organizing Map* (SOM), introduced by Kohonen [Koh89]. The SOM is an unsupervised learning process for arranging high-dimensional data without any class information by performing simultaneously projection and clustering. The performance of this dimensionality reduction technique was compared to KLT and it was shown that, while both frontends exhibited similar performance, the use of KLT produced slightly worse results.

Classification

WISARD (Wilkie, Aleksander and Stonham's Recognition Device) [Sto84] is a single layer neural network which is composed of discriminators, one for each class of object (face) that needs to be recognized, each discriminator comprising a set of function nodes. Classification is achieved by determining the classifier that gives the highest response for a given input image. The quality of the data used during the training phase is a major factor in the performance level of this system. While in [Sto84] invalid training material had to be rejected manually, in [MK90] Krin and Stonham propose a network that can control its own training. For this purpose, the training material is partitioned into sub-sets using a neural network with clustering capabilities.

In [CF90], after performing dimension reduction, the features (i.e. the outputs of the hidden layer of the MLP in auto-associative mode) are fed into a second network that works in the *classification* mode. In [LGTB97], the features are fed into a *Convolutional Network* (CN) which provide for partial invariance to global transformations such as translation, rotation and scale and to deformation. The performance of this network was compared to a MLP and experimental results showed that the MLP performed very poorly compared to the CN.

Lin, Kung and Lin introduced in [LKL97] a *Probabilistic Decision-Based Neural Networks* (PDBNNs) for AFR. PDBNNs have the merits of both neural networks and statistical approaches. They inherit the modular structure from its predecessor, the Decision Based Neural Network (DBNN) and the discriminant function of PDBNNs is in a form of a probability density. This approach was shown to have a performance comparable to the system combining SOMs and CNs while requiring a much smaller amount of computation at both training and classification time.

3.2.9 Support vector machines

The support vector machine (SVM) is a *binary classification* method that finds the optimal linear decision surface based on the concept *of structural risk minimization* [Vap95, Bur98]. Let $\{(x_i, y_i), i = 1, ..., N\}$ be a set of labeled training data where x_i is the data to be classified and $y_i \in \{+1, -1\}$ is the label. In the simple linearly separable case, the linear decision surface has the form:

$$w^T x + b = 0 \tag{3.27}$$

where w, the normal to the decision surface, is written as:

$$w = \sum_{i=1}^{N} \alpha_i y_i x_i \tag{3.28}$$

Among the hyperplanes minimizing the empirical risk, we search for the one that minimizes the expected risk. It can be shown that this optimal hyperplane is obtained by minimizing $||w||^2$ with respect to the α_i's subject to a set of constraints. This is a quadratic optimization problem. There exists a simple geometric interpretation: the optimal hyperplane is the one that maximizes the sum of the distances to the closest positive and negative training samples, *the support vectors*. This sum is called the *margin* of the separating hyperplane. This framework can be extended to non-separable training sets but also to non-linear decision surfaces using kernel functions.

In [Phi99], Phillips reformulated the AFR problem as a problem in difference space, which models dissimilarities between faces of different people. The two classes are: dissimilarities between faces of the same person and dissimilarities between faces of different persons as is the case in [MWP98]. Experimental results indicate that, for both identification and verification, the SVM is superior to a simple eigenface approach where a Euclidean distance is used to compute the distance in the face subspace.

While Phillips used generic support vector machines in [Phi99], in [JKLM99, JMKL00] Jonsson et al. use client specific support vectors. They tried to apply the SVM both to PCA and LDA representations of the face. Their study supports the hypothesis that the SVM approach is able to extract the relevant discriminatory information from the training data. Thus, when the representation space already captures and emphasizes the discriminatory information as is the case for the LDA, SVM are not superior to simple Euclidean or correlation distances anymore.

3.3 Local Approaches

In this section, we consider those approaches to AFR which view the face as a collection of features. We first very briefly review the *geometric feature-based* approaches. We then describe the *component-based* algorithms which are generally direct extensions of global approaches. Next we present approaches based on the *Gaussian mixture model (GMM)* and the *hidden Markov model (HMM)*. We also consider the graph-based approaches such as the *elastic graph matching (EGM)* and the *elastic bunch graph matching (EBGM)*. Finally, we present the *local feature analysis (LFA)* which is the core of Visionics FaceIt's AFR module.

3.3.1 Geometric features

The earliest techniques used for the problem of AFR are the geometric feature-based approaches. The idea of such systems is to extract relative positions and other parameters of distinctive features such as eyes, mouth, nose and chin. For instance, in [BP93] a set of 35 geometrical features are extracted automatically:

- eyebrow thickness and vertical position at the eye center position
- a coarse description of the left eyebrow-s arches
- nose vertical position and width
- mouth vertical position, width, height, upper and lower lips
- radii describing the chin
- face width at nose position
- face width halfway between nose tip and eyes

These features are matched using a Mahalanobis distance. However, it is generally difficult to extract such features with great accuracy and it was shown in [BP93] that this approach was outperformed by a component-based approach.

3.3.2 Component-based approaches

Many global approaches to AFR were extended to consider different components of the face, such as the eyes, the mouth or the nose instead of the whole face pattern. The design of a component-based approach thus relies mainly on the choice of the components, the measure of similarity between two components and the way the outputs of the different matchers are fused. In the following, we give several examples of such approaches.

Elaborating on the work of Baron [Bar81], Brunelli and Poggio extended the correlation approach to multiple templates [BP93]. Four components were considered: the eyes, the nose, the mouth and the face (from eyebrows downward). Each feature is matched separately and the combination is done through the use of a HyperBF network. The experimental analysis shows that the most discriminating features are the eyes, then the nose, the mouth and that the whole face is the least discriminating one.

In [PMS94] Pentland et al. proposed an extension of the eigenfaces, called *modular* eigenfaces. As is the case in [BP93], the global representation of the face is augmented by local prominent features such as the eyes, the nose or the mouth. The notion of eigenface is thus extended to eigeneyes, eigennose and eigenmouth. Note that the notion of *eigenfeatures* was first applied by Welsh and Shah to the problem of image coding [WS92]. For a small number of eigenvectors, the eigenfeatures approach outperformed the eigenface approach and the combination of eigenfaces and eigenfeatures outperformed each algorithm taken separately.

In [Phi98] Phillips used matching pursuit filters to encode the interior of the face, but also the tip and the bridge of the nose, and the left and right eyes. For each template, a total score is computed which is a weighted sum of the scores of the individual features. The weighting scheme is very simple: the weight for the interior of the face is set to 0.5 and the weights for the remaining features are set to 1.0.

Finally in [HHWP03] Heisele et al. extended the global SVM approach to the component-based approach. In a first step, 10 facial components are detected. Then to generate the input of the face classifier, each component is first normalized in size and their gray values are combined into a single feature vector which is subsequently fed to an SVM classifier.

3.3.3 Gaussian mixture model

The application of the Gaussian mixture model (GMM) to the problem of AFR by Sanderson and Paliwal [SP02, San02] is directly inspired by the work of Reynolds et al. on speaker recognition [RQD00].

In [Rey95], Reynolds introduced the GMM for the speaker recognition problem. Let $O = \{o_1, ..., o_T\}$ be a sequence of observations emitted by a GMM whose parameters are denoted λ. If we assume that these observations are independent, then the log-likelihood of the sequence of observations is:

$$\log P(O|\lambda) = \sum_{t=1}^{T} \log P(o_t|\lambda) \tag{3.29}$$

where $P(o_t|\lambda)$ is given by:

$$P(o_t|\lambda) = \sum_{k=1}^{K} w_k \mathcal{N}(o_t; \mu_k, \Sigma_k) \tag{3.30}$$

$\lambda = \{w_k, \mu_k, \Sigma_k, k = 1...K\}$, where w_k, μ_k and Σ_k are respectively the mixture weights, Gaussian means and covariance matrices. The model parameters λ_C of a given client C were trained through a ML estimation using only data from client C.

However, Reynolds et al. showed in [RQD00] that an improved performance could be obtained by adapting the client model from a *universal background model* (UBM). This UBM is a large GMM (it contains typically between 512 and 2,048 Gaussians) that represents the speaker-independent distribution of features. Thus, the UBM parameters, denoted λ_{UBM}, should be trained with a large amount of data corresponding to different persons or conditions to reflect as much acoustic diversity as possible. The model of client C is estimated through a MAP adaptation of the UBM using client specific data. For the verification problem, if θ is the decision threshold, then the following test is performed:

$$\frac{1}{T} \left[\log P(O|\lambda_C) - \log P(O|\lambda_{UBM})\right] \gtrless \theta \tag{3.31}$$

In [SP02, San02] the features used are DCT-based. The downside of using a GMM based classifier is that much information about the face structure is lost. Therefore, in order to increase the performance of the GMM approach without sacrificing its simplicity, in [CSB04] Cardinaux et al. propose to augment features with their position of extraction. An improved performance is obtained at the expense of a modest increase of the computational cost.

3.3.4 Hidden Markov model

In [Sam94], Samaria pioneered the use of the hidden Markov model (HMM) for the problem of AFR. The starting point of this work is the observation that a face can be segmented into a number of regions which contain facial landmarks such as the eyes, nose, mouth, etc. If these regions could be reliably located, then standard pattern matching techniques could be applied to each region individually as done for instance in [BP93]. However, Samaria argues that the accurate location of such points is a difficult problem and that the boundaries between adjacent regions are unclear. A potential solution to the problem of traditional pattern matching techniques is to associate facial regions with the states of a HMM. Samaria [Sam94] and Nefian [Nef99] both started with a simple 1-D HMM and then considered the more complex case of the 2-D HMM.

1-D HMM

If a face is in an upright, frontal position, one can assume that regions of the face will appear in a predictable order: forehead, eyes, nose, mouth, chin. This natural ordering suggests the use of a *top-to-bottom* model, similar to the traditional *left-to-right* model used in speech recognition, where the states of the model correspond to the five facial landmarks previously listed. The observations emitted by each state are blocks of consecutive lines or their compressed version through a DCT. Models are estimated using the *Baum-Welch* algorithm based on the EM principle and at test time the *Viterbi* algorithm is performed to determine the best model.

However such an approach is limited by the 1-D modeling of a 2-D object. While it can compensate for vertical deformations of the face, it cannot deal with the same variabilities in the horizontal direction, such as in-depth rotation.

2-D HMM

As the complexity of the Baum-Welch and Viterbi algorithms for a true 2-D HMM is exponential in the size of the data, and thus intractable for the problem of interest, [Sam94] and [Nef99] used the *pseudo 2-D HMM* (P2D HMM) also sometimes referred to as *planar HMM* or *embedded HMM*, introduced for the problem of optical character recognition (OCR) by Agazzi et al. [AKLP93, KA94]. The assumption of the P2D HMM is that there exists a set of "super" states which are Markovian and which subsume a set of simple Markovian state. Hence, the network of simple Markovian states is not fully connected in 2-D. The super states correspond to the same facial regions as in the 1-D case ordered in a *top-to-bottom* fashion and they contain simple left-to-right 1-D HMMs. Observations are blocks of pixels or their compressed version through DCT. It was shown in [EMR00, CSB04] that an improved performance could be obtained if the individual HMM models were derived from a generic HMM face model using, for instance, through MAP adaptation.

Recently, Nefian introduced the embedded Bayesian network (EBN) which generalizes the P2D HMM by allowing each HMM to be replaced by any arbitrary Bayesian network, and applied it successfully to the problem of AFR [Nef02].

3.3.5 Graph-based approaches

In this section, we will review the *elastic graph matching* (EGM) and the *elastic bunch graph matching (EBGM).* We will also consider the extensions of both algorithms to incorporate discriminatory information.

Elastic graph matching

The *Elastic Graph Matching* algorithm (EGM), which has roots in the neural network community, was introduced by Lades et al. [LVB+93]. Given a template face image I_t, one first derives a face model from this image. A grid is placed on the face image and the face model is a *vector field* $O = \{o_{i,j}\}$ where $o_{i,j}$ is the feature vector extracted at position (i, j) of the grid which summarizes local properties of the face (c.f. Figure 3.4). Gabor coefficients are generally used but other features, like morphological feature vectors, have also been considered and successfully applied to the EGM problem [KTP00]. The lattice formed by the vector field is generally much coarser than the "natural" lattice formed by the vector field of pixels. Given a query image I_q, one also derives a vector field $X = \{x_{k,l}\}$ but on a finer grid than the template face (c.f. Figure 3.4).

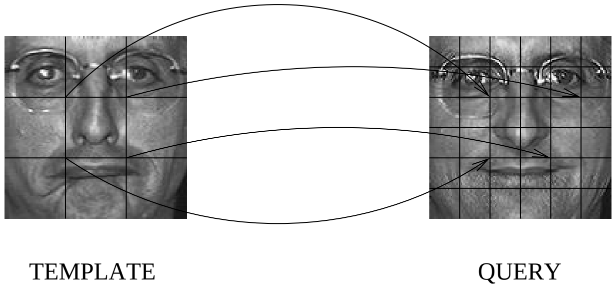


Figure 3.4: Possible mapping between a template image and a query image.

In the EGM approach, the distance between the template and query images is defined as a best mapping $\mathcal{M}^*$ among the set of all possible mappings $\{\mathcal{M}\}$ between the two vector fields O and X. The optimal mapping depends on the definition of the cost function $\mathcal{C}$. Such a function should keep a balance between the local matching of features and the requirement to preserve spatial distance. Therefore, a proper cost function should be of the form:

$$\mathcal{C}(\mathcal{M}) = \mathcal{C}_v(\mathcal{M}) + \rho\mathcal{C}_e(\mathcal{M}) \tag{3.32}$$

where $\mathcal{C}_v$ is the cost of local matchings, $\mathcal{C}_e$ the cost of local deformations and ρ is a parameter which controls the *rigidity* of the elastic matching and has to be hand-tuned. $\mathcal{C}_v$ is the sum of all local matchings and the measure of similarity between a vector $o_{i,j}$ in I_t and a vector $x_{k,l}$ in I_q is a simple cosine distance.

As the number of possible mappings is extremely large, even for lattices of moderate size, an exhaustive search is out of the question and an approximate solution has to be found. Toward this end, a two steps procedure was designed:

- *rigid matching*: the whole template graph is shifted around the query graph. This corresponds to $\rho \to \infty$. We obtain an initial mapping $\mathcal{M}_0$.
- *deformable matching*: the nodes of the template lattice are then stretched through random local perturbations to reduce further the cost function until the process converges to a locally optimal mapping $\mathcal{M}^*$, i.e. once a predefined number of trials have failed to improve the mapping cost.

The previous matching algorithm was later improved. For instance, in [KTP00] the authors argue that the two-stage coarse-to-fine optimization is sub-optimal as the deformable matching relies too much on the success of the rigid matching. The two stage optimization procedure is replaced with a probabilistic hill-climbing algorithm which attempts to find at each iteration both the optimal global translation and the set of optimal local perturbations. In [TKP01], they further drop the $\mathcal{C}_e$ term in equation 3.32. However, to avoid unreasonable deformations, local translations are restricted to a neighborhood.

Elastic bunch graph matching

The Elastic Bunch Graph Matching (EBGM) approach elaborates on the simple EGM [WFKvdM97]. One of the major innovations of the EBGM is to associate graph nodes to facial landmarks. Thus the same graph nodes correspond to the same facial features for different faces.

The face representation of a given image is built using a *face bunch graph* (FBG). A FBG is a general representation of the face. All vectors in a FBG referring to the same facial feature (called fiducial point) are bundled together in a bunch. Each fiducial point is represented by several alternatives to account for as many possible variations of that feature. For instance, an eye bunch may include features (called jets) from close, open, female and male eyes, etc. Such an FBG has to been initiated by locating manually the fiducial points on a set of reference images. The EBGM is used to extract the fiducial points on the face and thus to generate the graph representation of an image. The cost function which is used to estimate the similarity between an image graph and the FBG is very similar to the cost function of the EGM (c.f. equation 3.32). However, a difference is in the use of the phase information in the EBGM. It is used to disambiguate features which have a similar magnitude but also to estimate local translations. To find the face graph which minimizes this cost function, a coarse to fine approach similar to the one used for the EGM is employed: first the algorithm compensates for global translation, scale or aspect ratio and then for local distortions.

Since the nodes of the graphs correspond to the same facial features, the matching of two face graphs is greatly simplified. The similarity function between two face graphs is defined as the average over the similarities between pairs of corresponding jets. The similarity between corresponding jets is measured with a simple cosine distance.

It should be noted that the idea of associating the nodes of a graph to salient features of the face appeared in earlier work by Majunath et al. [MCvdM92]. Feature points are detected without assuming any knowledge of the face structure. The feature extraction consists of two basic steps. During the first step, one performs a Gabor wavelet decomposition to extract information at different scales and orientations. The second step makes use of local scale interactions between oriented features. Typically, 35 to 50 points are obtained in this manner and form the face graph. They generally correspond to salient facial features such as the eyes or the nose. Once features are extracted from the face, a topological graph is built to model the interaction between features. To compare two face graphs, a two-stage matching similar to the one suggested in [LVB$^+$93] is developed. One first compensates for a global translation of the graphs and then performs local deformations for further optimization. However, another difference with [LVB$^+$93] is that the cost of local deformations (also called *topology cost*) is only computed after the features are matched which results in a very fast algorithm. However, one advantage of [WFKvdM97] over [MCvdM92] is in the use of the bunch graph which provides a *supervised* way to extract salient features.

Incorporating discriminatory information

An obvious shortcoming of both EGM and EBGM is that the cost of local matchings is a simple sum of all local matchings. This contradicts the fact that certain parts of the face contain more discriminant information and that this distribution of the information across the face may vary from one person to another. Hence, the cost of local matchings at each node should be weighted according to their discriminatory power.

A general method for combining linearly multiple experts was developed and applied to the weighting of the nodes of and EBGM system in [Krü97]. The weighting is defined by a *parameterization function* (the same for all nodes) and the parameters of the function are estimated on a training set by maximizing an evaluation function using the simplex method. [KTP00] derived coefficients based on the between- and within-class variability at each node of an EGM graph. Fisher's Linear Discriminant has also been used in [DFB99, KTP00] to project the feature vectors in maximally discriminant sub-spaces. Finally, [TKP01] reformulates the classical Fisher's Discriminant ratio to a quadratic optimization problem subject to a set of inequality constraints. The optimal separating hyperplanes are found at each location using SVMs [Vap95].

3.3.6 Local feature analysis

Visionics FaceIt's AFR module is based on *local feature analysis (LFA)* introduced by Penev and Atick [PA96, Pen00]. LFA aims at addressing two shortcomings of PCA. The PCA representation is *non-local*, i.e. the support of the eigenfaces, which can be viewed as global filters, extend over the entire image, as is the case for all global approaches. Moreover, the PCA representation is *non-topographic*, which means that nearby values in the feature representation do not possess any relationship among each other.

This does not mean that interesting information cannot be retrieved from the eigenfaces. Let us denote by $[e_1, ..., e_N]$ the first N eigenfaces, i.e., if σ_n^2 is the eigenvalue associated with e_n, $\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_N$. LFA uses a set of local analysis filters $K_{i,j}$ which are different at each position (i, j) where the face is analyzed:

$$K_{i,j}^N(k,l) = \frac{1}{\sigma_k} \sum_{n=1}^{N} e_n(i,j) e_n(k,l) \tag{3.33}$$

Let I be a face image whose projection on the PCA space gives the following vector $[w_1, ..., w_N]^T$. Then the feature vector $o_{i,j}$ extracted with LFA at position (i, j) is given by:

$$o_{i,j} = \sum_{k,l} K^N_{i,j}(k,l) I(k,l) = \sum_{n=1}^{N} \frac{w_n}{\sigma_n} e_n(i,j) \tag{3.34}$$

and the original image can be reconstructed as follows:

$$I(i,j) \approx \sum_{n=1}^{N} \sigma_n \left(\sum_{k,l} e_n(k,l) o_{k,l} \right) e_n(i,j) \tag{3.35}$$

It turns out that the filters $K^N_{i,j}$ are feature detectors (and not edge detectors as would be the case for instance with Gabor filters) different from each other and matched to the feature that is expected near their respective centers. The properties of these filters are extensively discussed in [PA96, Pen00]. One of the most important ones is that, for a given dimension N, the outputs at different positions on the grid are maximally decorrelated. Another important property is that the reconstruction error for the LFA representation is exactly equal to that for the PCA representation.

However, a problem with this approach (and more generally, with all local approaches) compared to PCA is that the output is no longer low-dimensional, a desirable property. To remove the local correlation between features at adjacent positions, a sparsification algorithm is used, i.e. we search for the set of points that best reconstructs the face. A greedy heuristic is used to find incrementally the best points. The algorithm is initialized with an empty set of points. Now let us assume that the best set of points to reconstruct the face was obtained at the i-th step. At the $i+1$-th step of the algorithm, one first calculates the residual error at each position of the grid and then looks for the point (i,j) which has the largest residual error. This point is added to the set of points and the procedure is repeated until the residual error falls below a predefined threshold or a maximum number of points was obtained.

Very little information is provided about the way such features are matched to perform recognition.

3.4 Conclusion

In this chapter, we have reviewed the major approaches to AFR which can be separated into global and local approaches. Note that, far from being exclusive, global and local techniques can be combined to make the best out of their complementary natures. For instance, in [LW02] the eigenfaces and Fisherfaces algorithms are applied on a Gabor representation of the face and the performances of these algorithms are greatly improved compared to the case where they are applied on a gray level representation.

As underlined in the introductory section of this dissertation, the majority of the work on AFR has focused on the problem of representation and that relatively little work has been devoted to the matching problem. Indeed, for the classification step most approaches employ simple measures of similarity such as the L_1, L_2, cosine or Mahalanobis distances without any guarantee that they are optimal for the problem at hand.

One noticeable exception is BIC which focuses on the modeling of image differences (c.f. section 3.2.6). Note that this algorithm was one of the top performers during the FERET evaluation [PMRR00]. Due to its relative simplicity and its excellent performance, BIC will serve as a baseline for the comparison with our novel approach.

4

Turbo HMM and Turbo SSM

4.1 Introduction

The one-dimensional hidden Markov model (1-D HMM) is a class of stochastic signal model which has a long history of success in various problem domains, perhaps most notably in the field of automatic speech recognition (ASR). This success is largely due to the development of computationally efficient algorithms to solve the three fundamental problems of HMM design [Rab89]. Given a sequence of observations $O = \{o_1, ..., o_T\}$ and the model parameters λ, these problems are the following ones:

- Problem 1: How to efficiently compute $P(O|\lambda)$, the probability of the observation sequence given the model parameters?
- Problem 2: How to choose the sequence of states $Q = \{q_1, ..., q_T\}$ which is optimal in some meaningful sense?
- Problem 3: How to adjust the model parameters λ to maximize $P(O|\lambda)$?

The solution to these three problems are respectively the forward-backward, Viterbi and Baum-Welch algorithms.

The Markov random field (MRF) is the 2-D counterpart of the 1-D Markov chain where the natural ordering of past, present and future is replaced by the spatial concept of neighborhood. The MRF modeling process generally consists of the following steps [Li94]: defining a neighborhood system, defining cliques, defining the prior clique potentials, deriving the likelihood energy and deriving the posterior energy. In this dissertation, we will consider a subclass of MRF models, the Markov mesh random field (MMRF), which reintroduces the notion of past, present and future thanks to the raster scan [AHK65]. More precisely, we will focus on the first order MMRF. Let $Q = \{q_{i,j}, i = 1, ..., I, j = 1, ..., J\}$ be a $I \times J$ array of states and let $Q_{i,j}$ be the set of states to the left or above $q_{i,j}$: $Q_{i,j} = \{q_{m,n}, m < i \text{ or } n < j\}$. Then the first order MMRF can be defined by the following property (c.f. also Figure 4.1):

$$P(q_{i,j}|Q_{i,j}) = P(q_{i,j}|q_{i,j-1}, q_{i-1,j}) \tag{4.1}$$

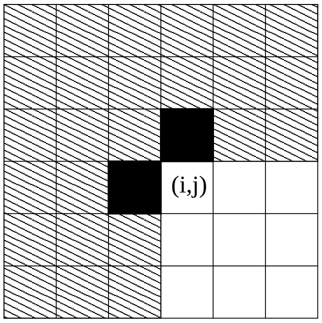


Figure 4.1: Markovian property of transitions among states for the first order Markov mesh random field (c.f. equation 4.1).

Reintroducing the notion of past and future is beneficial as it allows to develop the joint distribution of states $P(Q|\lambda)$, as is the case for the 1-D HMM. Thus, the forward-backward, Viterbi and Baum-Welch algorithms developed for the 1-D case can be extended to the 2-D case. However, even with the simple first-order Markovian model considered, the direct extension of these algorithms to the 2-D case is exponential in the size of the data [KA94], and hence intractable for most applications of practical value. Thus, approximations are required.

The approach we pursue here is to start from a hypothetical 2-D HMM and to first convert it into a turbo HMM (T-HMM): a set of inter-connected horizontal and vertical 1-D HMMs that “communicate” through inducing prior probabilities on each other. The solutions to the three problems of HMM design rely on a modified version of the forward-backward algorithm. It is performed successively on rows and columns and the process is iterated until convergence. We will also consider the continuous state HMM, generally referred to as the state-space model (SSM) [TKH00]. We will thus introduce the turbo state-space model (T-SSM).

In this chapter, we will first very briefly review previous approximations of the 2-D HMM. In section 4.3, we will derive the approximation of the likelihood function that will next be used to provide efficient approximate answers to the three problems of HMM design in sections 4.4 to 4.6. As the modified-forward backward algorithm is iterative, we will also consider convergence issues in section 4.7. Finally, we will compare experimentally the proposed approximate solution to the ML solution in section 4.8 before outlying future work.

4.2 Previous Approximations of the 2-D HMM

The goal of this section is not to provide an extensive review of the literature on approximations of the 2-D HMM but to give an idea of the approaches which have been pursued. It seems that most approximations attempt to replace the 2-D HMM with a 1-D HMM or a set of 1-D HMMs whose properties are well understood.

Perhaps the simplest approach is to trace a 1-D scan that takes into account as much of the neighborhood relationship (or 2-D structure) of the data as possible such as the *Hilbert-Peano* scan (c.f. Figure 4.2), as done by Abend et al. [AHK65] .

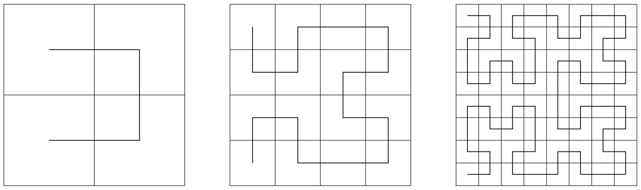


Figure 4.2: Examples of Hilbert-Peano scans for various array sizes.

A more recent approach is the *path constrained variable state Viterbi* (PCVSV) introduced by Li et al. [LNG00] which considers a sequence of states on a row as the states of a 1-D HMM. However, such a 1-D HMM has such a huge number of states that the direct application of the algorithms designed for the 1-D HMM is unpractical. The central idea of PCVSV is thus to consider only the N sequences with the largest prior probabilities. A fast algorithm is designed to avoid the calculation of posterior probabilities for all state sequences. It separates the blocks on a row from other blocks by neglecting their statistical dependencies. Columns or diagonals could also be considered instead of rows. In [LNG00], diagonals are chosen since blocks on diagonals are more geometrically distant than blocks on rows or columns and are therefore expected to exhibit less correlation.

Certainly the most famous approximation of the 2-D HMM that makes use of a set of 1-D HMMs is the pseudo 2-D HMM (P2D HMM) of Kuo and Agazzi [KA94], also sometimes referred to as planar or embedded HMM, which has been applied to the problem of OCR and face recognition [Sam94, Nef99]. The assumption of the P2D HMM is that there exists a set of "super" states which are Markovian and which subsume a set of simple Markovian states. Hence, the network of simple Markovian states is not fully connected in 2-D.

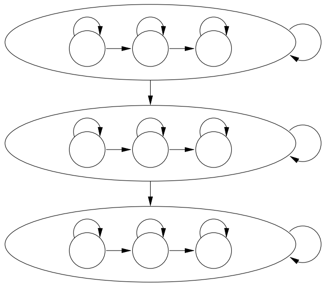


Figure 4.3: A Pseudo 2-D HMM

Another approach is to consider *independent* horizontal and vertical 1-D HMMs. In [HLSS02] Hallouli et al. explore two different fusion schemes for the problem of OCR: decision fusion and data fusion. In the decision fusion scheme, the classifiers are assumed independent which enables to derive an approximation of the joint likelihood function. On the contrary, in the data fusion scheme line and column features occurring at the same spatial index are considered highly correlated.

In [MHNM97], Miller et al. consider *inter-dependent* horizontal and vertical 1-D HMMs and focus on the decoding problem for binary image reconstruction. The decoding algorithm is based on the following intuitive heuristic. If the horizontal and vertical passes agree on a bit at a given position, then it is fixed for the subsequent iterations and the process is repeated until a cost function based on sum of squared errors decreases.

Finally, in [TC01, Tok01] Tokuyasu and Chou introduce the turbo recognition (TR), which is an approach to layout analysis of scanned document images inspired by the turbo decoding from communication theory. The TR algorithm is based on a generative model of image production in which two finite state grammars simultaneously describe the structure in horizontal and vertical directions. The decoding algorithm used by TR is derived from graphical models as applied in particular to turbo codes. Note that a similar approach has also been applied to the problem of solving crossword puzzles [SLK99].

4.3 Approximation of the Likelihood Function

We assume in the following that the reader is familiar with 1-D HMMs (see e.g., [Rab89]). 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 i-th row and j-th 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 i-th row and j-th column of states. Finally, let λ be the set of all model parameters, and let $\lambda_i^{\mathcal{H}}$ and $\lambda_j^{\mathcal{V}}$ be the respective rows and columns of parameters.

The joint likelihood of observations O and states Q given λ is:

$$\begin{aligned} P(O, Q|\lambda) &= P(O|Q, \lambda)P(Q|\lambda) \\ &= \prod_{i,j} P(o_{i,j}|q_{i,j}, \lambda)P(q_{i,j}|q_{i,j-1}, q_{i-1,j}, \lambda). \end{aligned} \tag{4.2}$$

Note that the conditional probability $P(q_{i,j}|q_{i,j-1}, q_{i-1,j}, \lambda)$ reduces to $P(q_{1,j}|q_{1,j-1}, \lambda)$ if $i = 1$, to $P(q_{i,1}|q_{i-1,1}, \lambda)$ if $j = 1$ and to $P(q_{1,1}|\lambda)$ if $i = j = 1$.

In the next two subsections, we will present the approximations of $P(O, Q|\lambda)$ underlying the T-HMM framework. The first step is to separate the 2-D HMM into a set of horizontal and vertical 1-D HMMs. As solving the three problems of interest is still too computationally intensive, an additional approximation has to be made.

4.3.1 Separating a 2-D HMM into horizontal and vertical 1-D HMMs

We will assume from now on that $P(q_{i,j}|q_{i,j-1}, q_{i-1,j}, \lambda)$ is *separable*, i.e. that it can be decomposed into the product of horizontal and vertical components. This approximation will allow us to run the forward-backward algorithm on rows and columns. Hence:

$$P(q_{i,j}|q_{i,j-1}, q_{i-1,j}, \lambda) = \nu(q_{i,j-1}, q_{i-1,j}) f^{\mathcal{H}}(q_{i,j}, q_{i,j-1}) f^{\mathcal{V}}(q_{i,j}, q_{i-1,j}) \tag{4.3}$$

where $\nu(q_{i,j-1}, q_{i-1,j})$ is a normalization factor:

$$\nu(q_{i,j-1}, q_{i-1,j}) = \frac{1}{\sum_{q_{i,j}} f^{\mathcal{H}}(q_{i,j}, q_{i,j-1}) f^{\mathcal{V}}(q_{i,j}, q_{i-1,j})} \tag{4.4}$$

Here we derive equations for the optimal horizontal and vertical components. We then show that, if $f^{\mathcal{H}}(q_{i,j}, q_{i,j-1})$'s and $f^{\mathcal{V}}(q_{i,j}, q_{i-1,j})$'s are optimal, then they effectively approximate $P(q_{i,j}|q_{i,j-1}, \lambda)$ and $P(q_{i,j}|q_{i-1,j}, \lambda)$, respectively.

Let us first consider the problem generally. Consider a conditional distribution $p_{i|jk}$ where $\sum_i p_{i|jk} = 1, \forall(j, k)$. We want to approximate $p_{i|jk}$ into the product

$a_{ij}b_{ik}$, where a_{ij} and b_{ik} are non negative and satisfy the requirement: $\sum_i a_{ij} = 1, \forall j$ and $\sum_i b_{ik} = 1, \forall k$. A positive normalization factor n_{jk} is needed to ensure: $\sum_i n_{jk}a_{ij}b_{ik} = 1, \forall (j,k)$. Since for all (j,k), both $p_{i|jk}$ and $n_{jk}a_{ij}b_{ik}$ are probability distributions, we can define the divergence [CT93] (see also appendix B):

$$\mathcal{D}_{jk} = \sum_i p_{i|jk} \log \left(\frac{p_{i|jk}}{n_{jk}a_{ij}b_{ik}} \right) \tag{4.5}$$

Our goal is to minimize $\sum_{j,k} \mathcal{D}_{jk}$ subject to the above constraints. We hence minimize the Lagrangian:

$$\mathcal{L} = \sum_{j,k} \mathcal{D}_{jk} + \sum_j \lambda_j (\sum_i a_{ij} - 1) + \sum_k \mu_k (\sum_i b_{ik} - 1) \tag{4.6}$$

We obtain the following formulas:

$$\frac{\partial \mathcal{L}}{\partial a_{ij}} = 0 \quad \Rightarrow \quad a_{ij} = \frac{\sum_k p_{i|jk}}{\sum_{i,k} p_{i|jk}} \;, \forall (i,j) \tag{4.7}$$

$$\frac{\partial \mathcal{L}}{\partial b_{ik}} = 0 \quad \Rightarrow \quad b_{ik} = \frac{\sum_j p_{i|jk}}{\sum_{i,j} p_{i|jk}} \;, \forall (i,k) \tag{4.8}$$

Since index j and k run from 1 to J and K, respectively, we can simplify the formulas for a_{ij} and b_{ik}:

$$a_{ij} = \frac{\sum_k p_{i|jk}}{K} \qquad\qquad b_{ik} \quad = \quad \frac{\sum_j p_{i|jk}}{J}$$

Now to interpret the result we observe that in general $p_{i|k} = \sum_j p_{i|jk} p_{j|k}$. If we further assume that $p_{j|k}$ is maximally non-informative, i.e., uniformly distributed then we obtain

$$p_{i|k} = \sum_j \frac{p_{i|jk}}{J},$$

which is exactly the formula we derived for b_{ik} above. A similar observation can be made regarding a_{ij}.

Next we specialize to the problem of interest, hence, $p_{i|jk}$ is replaced with $P(q_{i,j}|q_{i,j-1}, q_{i-1,j}, \lambda)$, a_{ij} with $f^{\mathcal{H}}(q_{i,j}, q_{i,j-1})$ and b_{ik} with $f^{\mathcal{V}}(q_{i,j}, q_{i-1,j})$. So, when $f^{\mathcal{H}}(q_{i,j}, q_{i,j-1})$'s (resp. $f^{\mathcal{V}}(q_{i,j}, q_{i-1,j})$'s) are chosen optimally, they approximate $P(q_{i,j}|q_{i,j-1}, \lambda_i^{\mathcal{H}})$'s (resp. $P(q_{i,j}|q_{i-1,j}, \lambda_j^{\mathcal{V}})$'s) assuming no prior information on $P(q_{i-1,j}|q_{i,j-1}, \lambda)$ (resp. $P(q_{i,j-1}|q_{i-1,j}, \lambda)$). Hereafter, we will replace the notations $f^{\mathcal{H}}(q_{i,j}, q_{i,j-1})$ and $f^{\mathcal{V}}(q_{i,j}, q_{i-1,j})$ with the more intuitive notations $P(q_{i,j}|q_{i,j-1}, \lambda_i^{\mathcal{H}})$ and $P(q_{i,j}|q_{i-1,j}, \lambda_j^{\mathcal{V}})$.

Moreover, we propose an additional simplifying assumption. To avoid the complexity due to terms that depend on states that are both on different rows and columns we assume that $\nu(q_{i,j-1}, q_{i-1,j})$ is (approximately) constant, i.e. does not depend on $q_{i,j-1}$ and $q_{i-1,j}$. We therefore obtain:

$$P(O,Q|\lambda) = \prod_{i,j} P(o_{i,j}|q_{i,j},\lambda)P(q_{i,j}|q_{i,j-1},\lambda_i^{\mathcal{H}})P(q_{i,j}|q_{i-1,j},\lambda_j^{\mathcal{V}}) \tag{4.9}$$

4.3.2 Approximation of the conditional probability

Note that equation 4.8.2 can be re-written as follows:

$$\begin{aligned} P(O,Q|\lambda) &= \prod_j P(o_j^{\mathcal{V}}|q_j^{\mathcal{V}},\lambda_j^{\mathcal{V}})P(q_j^{\mathcal{V}}|\lambda_j^{\mathcal{V}}) \prod_i P(q_{i,j}|q_{i,j-1},\lambda_i^{\mathcal{H}}) \\ &= \prod_j P(o_j^{\mathcal{V}},q_j^{\mathcal{V}}|\lambda_j^{\mathcal{V}}) \prod_i P(q_{i,j}|q_{i,j-1},\lambda_i^{\mathcal{H}}) \end{aligned} \tag{4.10}$$

where each term $P(o_j^{\mathcal{V}}|q_j^{\mathcal{V}},\lambda_j^{\mathcal{V}})$ corresponds to a vertical 1-D HMM. This formula still does not allow an efficient computation of $P(O|\lambda)$ and an additional approximation is required. The complexity of the estimation is due to the horizontal transition probabilities $P(q_{i,j}|q_{i,j-1},\lambda_i^{\mathcal{H}})$ which relate adjacent vertical 1-D HMMs. To simplify $P(O,Q|\lambda)$ we should replace the conditional probability $P(q_{i,j}|q_{i,j-1},\lambda_i^{\mathcal{H}})$ by a term which does not depend on $q_{i,j-1}$ but which still conveys horizontal context information. A possible solution is to perform the following substitution:

$$P(q_{i,j}|q_{i,j-1},\lambda_i^{\mathcal{H}}) \rightarrow P(q_{i,j}|o_i^{\mathcal{H}},\lambda_i^{\mathcal{H}}) \tag{4.11}$$

Hence, when performing the substitution 4.11 in equation 4.10 one obtains the following quantity:

$$P^{\mathcal{V}}(O,Q|\lambda) = \prod_j \left[P(o_j^{\mathcal{V}},q_j^{\mathcal{V}}|\lambda_j^{\mathcal{V}}) \prod_i P(q_{i,j}|o_i^{\mathcal{H}},\lambda_i^{\mathcal{H}}) \right] \tag{4.12}$$

where 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. Hence, horizontal and vertical HMMs can "communicate". We have to assume that the quantity $P(q_{i,j}|o_i^{\mathcal{H}},\lambda_i^{\mathcal{H}})$ is known, i.e., that it was obtained during a previous horizontal step. Obviously, an iterative procedure is required to compute these horizontal and vertical quantities. Note that one can derive a similar formula $P^{\mathcal{H}}(O,Q|\lambda)$ where horizontal 1-D HMMs communicate through the use of vertical priors.

4.4 Solution to Problem 1

In this section, we first derive the computation of $P^{\mathcal{V}}(O|\lambda)$ from the approximation of $P^{\mathcal{V}}(O,Q|\lambda)$. We then provide the equations for the modified forward-backward iterations, first in the case of discrete states and then in the case of continuous states. Finally, we consider the modified forward-backward operationally.

4.4.1 Computation of the likelihood function

If we sum (or integrate) over all possible paths, we obtain the marginal:

$$\begin{aligned} P^{\mathcal{V}}(O|\lambda) &= \sum_{Q} P^{\mathcal{V}}(O,Q|\lambda) \\ &= \sum_{q_1^{\mathcal{V}} \ldots q_J^{\mathcal{V}}} \prod_{j} \left[P(o_j^{\mathcal{V}}, q_j^{\mathcal{V}} | \lambda_j^{\mathcal{V}}) \prod_{i} P(q_{i,j} | o_i^{\mathcal{H}}, \lambda_i^{\mathcal{H}}) \right] \\ &= \prod_{j} \sum_{q_j^{\mathcal{V}}} \left[P(o_j^{\mathcal{V}}, q_j^{\mathcal{V}} | \lambda_j^{\mathcal{V}}) \prod_{i} P(q_{i,j} | o_i^{\mathcal{H}}, \lambda_i^{\mathcal{H}}) \right] \qquad (4.13) \end{aligned}$$

Let us note $P_j^{\mathcal{V}} = \sum_{q_j^{\mathcal{V}}} \left[P(o_j^{\mathcal{V}}, q_j^{\mathcal{V}} | \lambda_j^{\mathcal{V}}) \prod_i P(q_{i,j} | o_i^{\mathcal{H}}, \lambda_i^{\mathcal{H}}) \right]$. $P_j^{\mathcal{V}}$'s can be computed independently with a modified version of the forward-backward algorithm. The only difference with the forward-backward for the 1-D HMM is that we have to perform the following substitution for the emission probability:

$$b_{q_{i,j}}(o_{i,j}) \rightarrow b_{q_{i,j}}^{\mathcal{H}}(o_{i,j}) = b_{q_{i,j}}(o_{i,j}) \gamma_{i,j}^{\mathcal{H}}(q_{i,j}) \qquad (4.14)$$

For a summary of HMM notations, the reader can refer to Table 4.1.

notation	definition
$\pi_{q_{1,j}}^{\mathcal{V}}$	$P(q_{1,j} \mid \lambda_j^{\mathcal{V}})$
$a_{q_{i,j},q_{i-1,j}}^{\mathcal{V}}$	$P(q_{i,j} \mid q_{i-1,j}, \lambda_i^{\mathcal{V}})$
$b_{q_{i,j}}(o_{i,j})$	$P(o_{i,j} \mid q_{i,j}, \lambda)$
$\alpha_{i,j}^{\mathcal{V}}(q_{i,j})$	$P(o_{1,j}, \ldots o_{i,j}, q_{i,j} \mid \gamma_{1,j}^{\mathcal{H}}, \ldots \gamma_{i,j}^{\mathcal{H}}, \lambda_j^{\mathcal{V}})$
$\beta_{i,j}^{\mathcal{V}}(q_{i,j})$	$P(o_{i+1,j}, \ldots o_{I,j} \mid q_{i,j}, \gamma_{i+1,j}, \ldots \gamma_{I,j}, \lambda_j^{\mathcal{V}})$
$\gamma_{i,j}^{\mathcal{H}}(q_{i,j})$	$P(q_{i,j} \mid o_i^{\mathcal{H}}, \gamma_i^{\mathcal{V}}, \lambda_i^{\mathcal{H}})$

Table 4.1: HMM notation summary.

4.4.2 The modified forward-backward: discrete case

Forward α variable

- Initialization:

$$\alpha_{1,j}^{\mathcal{V}}(q_{1,j}) = \pi_{q_{1,j}}^{\mathcal{V}} b_{q_{1,j}}^{\mathcal{H}}(o_{1,j}) \qquad (4.15)$$

- Recursion:

$$\alpha^{\mathcal{V}}_{i+1,j}(q_{i+1,j}) = \left[\sum_{q_{i,j}} \alpha^{\mathcal{V}}_{i,j}(q_{i,j}) a^{\mathcal{V}}_{q_{i,j},q_{i+1,j}}\right] b^{\mathcal{H}}_{q_{i+1,j}}(o_{i+1,j}) \tag{4.16}$$

- Termination:

$$P^{\mathcal{V}}_j = \sum_{q_{I,j}} \alpha^{\mathcal{V}}_{I,j}(q_{I,j}) \tag{4.17}$$

Backward β variable

- Initialization:

$$\beta^{\mathcal{V}}_{I,j}(q_{I,j}) = 1 \tag{4.18}$$

- Recursion:

$$\beta^{\mathcal{V}}_{i,j}(q_{i,j}) = \sum_{q_{i+1,j}} a^{\mathcal{V}}_{q_{i,j},q_{i+1,j}} b^{\mathcal{H}}_{q_{i+1,j}}(o_{i+1,j}) \beta^{\mathcal{V}}_{i+1,j}(q_{i+1,j}) \tag{4.19}$$

Occupancy probability γ

$$\gamma^{\mathcal{V}}_{i,j}(q_{i,j}) = \frac{\alpha^{\mathcal{V}}_{i,j}(q_{i,j})\beta^{\mathcal{V}}_{i,j}(q_{i,j})}{\sum_{q_{i,j}} \alpha^{\mathcal{V}}_{i,j}(q_{i,j})\beta^{\mathcal{V}}_{i,j}(q_{i,j})} \tag{4.20}$$

Similar formulas can be derived for the horizontal pass. We can see that the interaction between horizontal and vertical processing, which is based on the occupancy probability γ, is more elaborate than the one used in [MHNM97]. It is worthwhile also to note that our re-estimation equations are similar to the ones derived for the page layout problem in [Tok01] based on the graphical model formalism.

4.4.3 The modified forward-backward: continuous case

Note that the equations that were derived in the previous sub-section for the discrete state HMM are valid in the continuous case, with the exception that we have to replace sums by integrals. However, while in the discrete case the forward and backward equations relate the forward and backward probabilities at adjacent positions, in the continuous case, these quantities are pdf's and we are interested in the equations that relate their parameters.

To keep the mathematical analysis tractable, we choose the emission probabilities, horizontal and vertical transition probabilities and initial occupancy probabilities to be Gaussians. In this dissertation, we will only present results for the case where the observations and states are uni-dimensional. Using the state-space formalism, the emission probability (or *measurement model*) can be expressed by the following equation:

$$o_{i,j} = f_{i,j} q_{i,j} + u_{i,j} \ , \ i = 1, \ldots, I, j = 1 \ldots J \tag{4.21}$$

where $u_{i,j} \sim \mathcal{N}(0, {\sigma_{i,j}}^2)$ is the *measurement noise.* The horizontal and vertical transition probabilities (or *process models*) can be written as:

$$\begin{aligned} q_{i,j} &= g^{\mathcal{H}}_{i,j} q_{i,j-1} + v^{\mathcal{H}}_{i,j} \ , \ i = 1, \ldots, I, j = 2 \ldots J && (4.22) \\ q_{i,j} &= g^{\mathcal{V}}_{i,j} q_{i-1,j} + v^{\mathcal{V}}_{i,j} \ , \ i = 2, \ldots, I, j = 1 \ldots J && (4.23) \end{aligned}$$

where $v^{\mathcal{H}}_{i,j} \sim \mathcal{N}(0, {s^{\mathcal{H}}_{i,j}}^2)$ and $v^{\mathcal{V}}_{i,j} \sim \mathcal{N}(0, {s^{\mathcal{V}}_{i,j}}^2)$ are respectively the horizontal and vertical *process noises*. Finally, we introduce the horizontal and vertical initial occupancy probabilities:

$$\begin{aligned} q_{i,1} &= v^{\mathcal{H}}_{i,1} \ , \ i = 1, \ldots, I && (4.24) \\ q_{1,j} &= v^{\mathcal{V}}_{1,j} \ , \ j = 1, \ldots, J && (4.25) \end{aligned}$$

As the emission, transition and initial occupancy probabilities are Gaussians, if we also initialize the occupancy probabilities $\gamma^{\mathcal{H}}_{i,j}$'s in a Gaussian manner (assuming that we start with a column operation), it is straightforward to show using equations 4.15-4.20 that $\alpha^{\mathcal{V}}_{i,j}$ and $\beta^{\mathcal{V}}_{i,j}$ have a Gaussian shape:

$$\begin{aligned} \alpha^{\mathcal{V}}_{i,j}(q_{i,j}) &= \frac{c^{\alpha\mathcal{V}}_{i,j}}{\sigma^{\alpha\mathcal{V}}_{i,j} (2\pi)^{\frac{1}{2}}} \exp\left\{ -\frac{(q_{i,j} - \mu^{\alpha\mathcal{V}}_{i,j})^2}{2{\sigma^{\alpha\mathcal{V}}_{i,j}}^2} \right\} \\ \beta^{\mathcal{V}}_{i,j}(q_{i,j}) &= \frac{c^{\beta\mathcal{V}}_{i,j}}{\sigma^{\beta\mathcal{V}}_{i,j} (2\pi)^{\frac{1}{2}}} \exp\left\{ -\frac{(q_{i,j} - \mu^{\beta\mathcal{V}}_{i,j})^2}{2{\sigma^{\beta\mathcal{V}}_{i,j}}^2} \right\} \end{aligned}$$

and that $\gamma^{\mathcal{V}}_{i,j}$ is a Gaussian with mean $\mu^{\gamma\mathcal{V}}_{i,j}$ and variance ${\sigma^{\gamma\mathcal{V}}_{i,j}}^2$. Introducing the following notations $\mu^{b\mathcal{H}}_{i,j}$, ${\sigma^{b\mathcal{H}}_{i,j}}^2$ and $c^{b\mathcal{H}}_{i,j}$:

$$\begin{aligned} \mu^{b\mathcal{H}}_{i,j} &= \frac{f_{i,j} o_{i,j} {\sigma^{\gamma\mathcal{H}}_{i,j}}^2 + \mu^{\gamma\mathcal{H}}_{i,j} \sigma^2_{i,j}}{{f_{i,j}}^2 {\sigma^{\gamma\mathcal{H}}_{i,j}}^2 + \sigma^2_{i,j}} && (4.26) \\ {\sigma^{b\mathcal{H}}_{i,j}}^2 &= \frac{{\sigma^{\gamma\mathcal{H}}_{i,j}}^2 \sigma^2_{i,j}}{{f_{i,j}}^2 {\sigma^{\gamma\mathcal{H}}_{i,j}}^2 + \sigma^2_{i,j}} && (4.27) \\ c^{b\mathcal{H}}_{i,j} &= \frac{\exp\left\{ -\frac{1}{2} \frac{(o_{i,j} - f_{i,j} \mu^{\gamma\mathcal{H}}_{i,j})^2}{({\sigma_{i,j}}^2 + {f_{i,j}}^2 {\sigma^{\gamma\mathcal{H}}_{i,j}}^2)} \right\}}{(2\pi)^{\frac{1}{2}} ({\sigma_{i,j}}^2 + {f_{i,j}}^2 {\sigma^{\gamma\mathcal{H}}_{i,j}}^2)^{\frac{1}{2}}} && (4.28) \end{aligned}$$

we can compute $\mu^{\alpha\mathcal{V}}_{i,j}$, $\mu^{\beta\mathcal{V}}_{i,j}$, $\mu^{\gamma\mathcal{V}}_{i,j}$, ${\sigma^{\alpha\mathcal{V}}_{i,j}}^2$, ${\sigma^{\beta\mathcal{V}}_{i,j}}^2$, ${\sigma^{\gamma\mathcal{V}}_{i,j}}^2$, $c^{\alpha\mathcal{V}}_{i,j}$ and $c^{\beta\mathcal{V}}_{i,j}$ using equations 4.15-4.20 [1].

[1] The following formulas were derived using the software Maple.

Forward α variable

- Initialization:

$$\mu_{1,j}^{\alpha\mathcal{V}} = \frac{\mu_{1,j}^{b\mathcal{H}} {s_{1,j}^{\mathcal{V}}}^2}{{s_{1,j}^{\mathcal{V}}}^2 + \sigma_{1,j}^{b\mathcal{H}2}} \tag{4.29}$$

$$\sigma_{1,j}^{\alpha\mathcal{V}2} = \frac{{s_{1,j}^{\mathcal{V}}}^2 \sigma_{1,j}^{b\mathcal{H}2}}{{s_{1,j}^{\mathcal{V}}}^2 + \sigma_{1,j}^{b\mathcal{H}2}} \tag{4.30}$$

$$c_{1,j}^{\alpha\mathcal{V}} = \frac{c_{i,j}^{b\mathcal{H}} \exp\left\{-\frac{1}{2}\frac{(\mu_{1,j}^{b\mathcal{H}} - \mu_j^{\mathcal{V}})^2}{\sigma_j^{\mathcal{V}2} + \sigma_{1,j}^{b\mathcal{H}2}}\right\}}{(2\pi)^{\frac{1}{2}}({s_{1,j}^{\mathcal{V}}}^2 + \sigma_{1,j}^{b\mathcal{H}2})^{\frac{1}{2}}} \tag{4.31}$$

- Recursion:

$$\mu_{i+1,j}^{\alpha\mathcal{V}} = \frac{g_{i+1,j}^{\mathcal{V}} \mu_{i,j}^{\alpha\mathcal{V}} {\sigma_{i+1,j}^{b\mathcal{H}}}^2 + \mu_{i+1,j}^{b\mathcal{H}}({s_{i+1,j}^{\mathcal{V}}}^2 + {g_{i+1,j}^{\mathcal{V}}}^2 \sigma_{i,j}^{\alpha\mathcal{V}^2})}{{\sigma_{i+1,j}^{b\mathcal{H}}}^2 + {s_{i+1,j}^{\mathcal{V}}}^2 + {g_{i+1,j}^{\mathcal{V}}}^2 \sigma_{i,j}^{\alpha\mathcal{V}2}} \tag{4.32}$$

$${\sigma_{i+1,j}^{\alpha\mathcal{V}}}^2 = \frac{{\sigma_{i+1,j}^{b\mathcal{H}}}^2({s_{i+1,j}^{\mathcal{V}}}^2 + {g_{i+1,j}^{\mathcal{V}}}^2 \sigma_{i,j}^{\alpha\mathcal{V}2})}{{\sigma_{i+1,j}^{b\mathcal{H}}}^2 + {s_{i+1,j}^{\mathcal{V}}}^2 + {g_{i+1,j}^{\mathcal{V}}}^2 \sigma_{i,j}^{\alpha\mathcal{V}2}} \tag{4.33}$$

$$c_{i+1,j}^{\alpha\mathcal{V}} = \frac{c_{i,j}^{\alpha\mathcal{V}} c_{i+1,j}^{b\mathcal{H}} \exp\left\{-\frac{1}{2}\frac{(\mu_{i+1,j}^{b\mathcal{H}} - g_{i+1,j}^{\mathcal{V}} \mu_{i,j}^{\alpha\mathcal{V}})^2}{{\sigma_{i+1,j}^{b\mathcal{H}}}^2 + {s_{i+1,j}^{\mathcal{V}}}^2 + {g_{i+1,j}^{\mathcal{V}}}^2 \sigma_{i,j}^{\alpha\mathcal{V}2}}\right\}}{(2\pi)^{\frac{1}{2}}({\sigma_{i+1,j}^{b\mathcal{H}}}^2 + {s_{i+1,j}^{\mathcal{V}}}^2 + {g_{i+1,j}^{\mathcal{V}}}^2 \sigma_{i,j}^{\alpha\mathcal{V}2})^{\frac{1}{2}}} \tag{4.34}$$

- Termination:

$$P_j^{\mathcal{V}} = c_{I,j}^{\alpha\mathcal{V}} \tag{4.35}$$

Backward β variable

- Initialization:

$$\mu_{I,j}^{\beta\mathcal{V}} = 0 \tag{4.36}$$

$$\sigma_{I,j}^{\beta\mathcal{V}2} \to \infty \tag{4.37}$$

$$c_{I,j}^{\beta\mathcal{V}2} = 1 \tag{4.38}$$

- Recursion:

$$\mu_{i,j}^{\beta\mathcal{V}} = \frac{1}{g_{i+1,j}^{\mathcal{V}}} \left(\frac{\mu_{i+1,j}^{b\mathcal{H}} {\sigma_{i+1,j}^{\beta\mathcal{V}}}^2 + \mu_{i+1,j}^{\beta\mathcal{V}} {\sigma_{i+1,j}^{b\mathcal{H}}}^2}{{\sigma_{i+1,j}^{\beta\mathcal{V}}}^2 + {\sigma_{i+1,j}^{b\mathcal{H}}}^2} \right) \tag{4.39}$$

$$\sigma_{i,j}^{\beta\mathcal{V}2} = \frac{1}{{g_{i+1,j}^{\mathcal{V}}}^2} \left({s_{i+1,j}^{\mathcal{V}}}^2 + \frac{{\sigma_{i+1,j}^{b\mathcal{H}}}^2 {\sigma_{i+1,j}^{\beta\mathcal{V}}}^2}{{\sigma_{i+1,j}^{b\mathcal{H}}}^2 + {\sigma_{i+1,j}^{\beta\mathcal{V}}}^2} \right) \tag{4.40}$$

$$c_{i,j}^{\beta\mathcal{V}} = \frac{c_{i+1,j}^{\beta\mathcal{V}} c_{i+1,j}^{b\mathcal{H}} \exp\left\{ -\frac{1}{2} \frac{(\mu_{i+1,j}^{b\mathcal{H}} - \mu_{i+1,j}^{\beta\mathcal{V}})^2}{{\sigma_{i+1,j}^{b\mathcal{H}}}^2 + {\sigma_{i+1,j}^{\beta\mathcal{V}}}^2} \right\}}{(2\pi)^{\frac{1}{2}} |g_{i+1,j}^{\mathcal{V}}| ({\sigma_{i+1,j}^{b\mathcal{H}}}^2 + {\sigma_{i+1,j}^{\beta\mathcal{V}}}^2)^{\frac{1}{2}}} \tag{4.41}$$

Occupancy probability γ

$$\mu_{i,j}^{\gamma\mathcal{V}} = \frac{\mu_{i,j}^{\alpha\mathcal{V}} \sigma_{i,j}^{\beta\mathcal{V}2} + \mu_{i,j}^{\beta\mathcal{V}} \sigma_{i,j}^{\alpha\mathcal{V}2}}{\sigma_{i,j}^{\alpha\mathcal{V}2} + \sigma_{i,j}^{\beta\mathcal{V}2}} \tag{4.42}$$

$$\sigma_{i,j}^{\gamma\mathcal{V}2} = \frac{\sigma_{i,j}^{\alpha\mathcal{V}2} \sigma_{i,j}^{\beta\mathcal{V}2}}{\sigma_{i,j}^{\alpha\mathcal{V}2} + \sigma_{i,j}^{\beta\mathcal{V}2}} \tag{4.43}$$

Similar formulas can be derived for horizontal quantities.

4.4.4 The modified forward-backward operationally

In Table 4.2 we consider the steps of the algorithm which are very similar in the discrete and continuous cases. 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. Whether the iterative process is initialized with row or column operation may theoretically impact the performance.

Note that we do not obtain one estimate of $P(O|\lambda)$ but two: a horizontal one $P^{\mathcal{H}}(O|\lambda) = \prod_i P_i^{\mathcal{H}}$ and a vertical one $P^{\mathcal{V}}(O|\lambda) = \prod_j P_j^{\mathcal{V}}$. Relating $P^{\mathcal{H}}(O|\lambda)$ or $P^{\mathcal{V}}(O|\lambda)$ to the true likelihood function $P(O|\lambda)$ is not obvious because of the substitution 4.11. However, we can consider the combination of these two scores as a classical problem of decision fusion. One can show that the optimal estimate $\hat{P}(O|\lambda)$ based on a divergence criterion is (c.f. appendix B):

$$\hat{P}(O|\lambda) \propto \sqrt{P^{\mathcal{H}}(O|\lambda) P^{\mathcal{V}}(O|\lambda)} \tag{4.44}$$

Of course, we could have considered more elaborate fusion schemes but, as we will see in the next chapter, this approach yielded acceptable results for the problem of interest.

$\boxed{1}$ Initialize the horizontal occupancy probability.
Assuming no prior information:
discrete case: initialize $\gamma_{i,j}^{\mathcal{H}}$'s uniformly, $\forall(i,j)$.
continuous case: set $\sigma_{i,j}^{\gamma\mathcal{H}} \to \infty$, $\forall(i,j)$.

$\boxed{2}$ Apply the modified forward-backward on the vertical 1-D HMMs.

$\boxed{3}$ Apply the modified forward-backward on the horizontal 1-D HMMs.

$\boxed{4}$ Go back to step 2 until convergence of the horizontal and vertical priors.

Table 4.2: Steps of the modified forward-backward iterations, assuming that we start with a vertical pass.

4.5 Solution to Problem 2

Problem 2 is concerned with the issue of finding the sequence of states Q that "best" explains in some sense the sequence of observations O. This is generally understood as the Q that best explains O *globally*:

$$Q^* = \arg\max_Q P(Q|O,\lambda) = \arg\max_Q P(O,Q|\lambda) \tag{4.45}$$

However, due to the complexity of this problem for both the discrete and continuous states cases, we will consider the states that best explain O *locally*:

$$q_{i,j}^* = \arg\max_{q_{i,j}} P(q_{i,j}|O,\lambda) \tag{4.46}$$

Obviously, we do not have a direct access to $P(q_{i,j}|O,\lambda)$ but to its estimates $\gamma_{i,j}^{\mathcal{H}}$ and $\gamma_{i,j}^{\mathcal{V}}$. Using one more time a criterion based on the Kullback-Leibler divergence, the optimal estimate $\hat{P}(q_{i,j}|O,\lambda)$ is:

$$\hat{P}(q_{i,j}|O,\lambda) \propto \sqrt{\gamma_{i,j}^{\mathcal{H}}(q_{i,j})\gamma_{i,j}^{\mathcal{V}}(q_{i,j})} \tag{4.47}$$

In the continuous case, as $\gamma_{i,j}^{\mathcal{H}}$ and $\gamma_{i,j}^{\mathcal{V}}$ are Gaussian, $\hat{P}(q_{i,j}|O,\lambda)$ will be Gaussian with mean:

$$\hat{\mu}_{i,j}^{\gamma} = \frac{\sigma_{i,j}^{\gamma\mathcal{V}^2}\mu_{i,j}^{\gamma\mathcal{H}} + \sigma_{i,j}^{\gamma\mathcal{H}^2}\mu_{i,j}^{\gamma\mathcal{V}}}{\sigma_{i,j}^{\gamma\mathcal{V}^2} + \sigma_{i,j}^{\gamma\mathcal{H}^2}} \tag{4.48}$$

and thus $q_{i,j}^* = \hat{\mu}_{i,j}^{\gamma}$.

Choosing the globally optimal sequence of states is equivalent to choosing the locally optimal states if and only if:

$$P(Q|O,\lambda) = \prod_{i,j} P(q_{i,j}|O,\lambda) \tag{4.49}$$

i.e., in the case where there is no context information (uniformly distributed transition probabilities in the discrete case). In the case where there is some context information, it is also interesting to determine when this approximation is valid. Using the following set of inequalities:

$$\prod_{i,j} P(q_{i,j}|O,\lambda) \leq P(Q|O,\lambda) \leq P(q_{i,j}|O,\lambda) \tag{4.50}$$

we see that finding the locally optimal states is equivalent to finding globally optimal states in the case where the distribution $P(Q|O,\lambda)$ is sharply peaked, i.e. when one path Q^* accounts for most of the total probability: $P(Q^*|O,\lambda) \approx 1$. In this latter case, we can provide an alternative solution to problem 1. Indeed, we have:

$$P(O|\lambda) = \sum_Q P(O,Q|\lambda) \approx P(O,Q^*|\lambda) \tag{4.51}$$

However, there exists a potential problem if the HMM is not fully connected. Indeed, in such a case the “optimal” state sequence may, in fact, not even be a valid state sequence [Rab89], i.e. $P(O,Q^*|\lambda) = 0$.

4.6 Solution to Problem 3

In the 1-D case, the ML estimate of the HMM parameters λ are generally derived using the Baum-Welch algorithm which is based on the EM principle. Let $\mathcal{Q}$ be Baum’s auxiliary function defined as:

$$\mathcal{Q}(\lambda|\lambda') = \sum_Q P(Q|O,\lambda') \log P(O,Q|\lambda) dQ \tag{4.52}$$

where λ' represents the current estimate of the HMM parameters. It has been proved that the maximization of $\mathcal{Q}(\lambda,\lambda')$ with respect to λ will lead to an increased likelihood [DLR77]:

$$\hat{\lambda} = \arg\max_{\lambda} \mathcal{Q}(\lambda|\lambda') \Rightarrow P(O|\hat{\lambda}) \geq P(O|\lambda') \tag{4.53}$$

In our case, as we do have not one but two estimates of $P(O,Q|\lambda)$, a horizontal and a vertical one, we have two $\mathcal{Q}$ functions:

$$\begin{aligned} \mathcal{Q}^{\mathcal{H}}(\lambda^{\mathcal{H}}|\lambda') &= \sum_Q P^{\mathcal{H}}(Q|O,\lambda') \log P^{\mathcal{H}}(O,Q|\lambda) dQ & (4.54) \\ \mathcal{Q}^{\mathcal{V}}(\lambda^{\mathcal{V}}|\lambda') &= \sum_Q P^{\mathcal{V}}(Q|O,\lambda') \log P^{\mathcal{V}}(O,Q|\lambda) dQ & (4.55) \end{aligned}$$

If we refer to equation 4.12, then clearly $\mathcal{Q}^{\mathcal{H}}$ (resp. $Q^{\mathcal{V}}$) can be written as the sum of I terms (resp. J terms) where each term corresponds to the i-th horizontal (resp. j-th vertical) 1-D HMM:

$$\mathcal{Q}^{\mathcal{H}}(\lambda^{\mathcal{H}}|\lambda') = \sum_{i=1}^{I} \mathcal{Q}_i^{\mathcal{H}}(\lambda_i^{\mathcal{H}}|\lambda') \tag{4.56}$$

$$\mathcal{Q}^{\mathcal{V}}(\lambda^{\mathcal{V}}|\lambda') = \sum_{j=1}^{J} \mathcal{Q}_j^{\mathcal{V}}(\lambda_j^{\mathcal{H}}|\lambda') \tag{4.57}$$

Moreover, these horizontal and vertical terms can be subdivided in the sum of 3 terms which correspond respectively to the initial occupancy, transition and emission probabilities. Note that emission probabilities are both horizontal and vertical parameters and thus that it may not be possible to optimize $\mathcal{Q}^{\mathcal{H}}$ and $Q^{\mathcal{V}}$ separately in the case where the horizontal and vertical occupancy probabilities do not reach agreement. To address this issue, we simply maximize:

$$\mathcal{Q}(\lambda|\lambda') = \mathcal{Q}^{\mathcal{H}}(\lambda^{\mathcal{H}}|\lambda') + \mathcal{Q}^{\mathcal{H}}(\lambda^{\mathcal{V}}|\lambda') \tag{4.58}$$

which is equivalent to accumulating the horizontal and vertical statistics for the emission probabilities.

Note that, obviously, the re-estimation formulas for the parameters are dependent on the HMM model and thus, on the problem of interest. Therefore, these formulas will be derived in the next chapters once our face recognition model based on the 2-D HMM is introduced.

4.7 Convergence Issues

The solutions to the 3 problems of HMM design for the T-HMM and the T-SSM rely on the modified forward-backward iterations. It is thus of interest to determine whether the horizontal and vertical passes converge in some well-defined sense and, if possible, to improve the convergence.

4.7.1 A measure of convergence

Let $\gamma^{\mathcal{H}}$ and $\gamma^{\mathcal{V}}$ be respectively the joint distributions of the $\gamma_{i,j}^{\mathcal{H}}$'s and $\gamma_{i,j}^{\mathcal{V}}$'s. $\mathcal{D}(\gamma^{\mathcal{H}}, \gamma^{\mathcal{V}})$, the symmetric divergence (c.f. appendix B), is a measure of how well the horizontal and vertical passes agree over the entire image. If we further assume independence of $\gamma_{i,j}^{\mathcal{H}}$'s and similarly of $\gamma_{i,j}^{\mathcal{V}}$'s, then:

$$\mathcal{D}(\gamma^{\mathcal{H}}, \gamma^{\mathcal{V}}) = \sum_{i,j} \mathcal{D}(\gamma_{i,j}^{\mathcal{H}}, \gamma_{i,j}^{\mathcal{V}}) \tag{4.59}$$

This measure of convergence is useful as a stopping criterion. After the n-th iteration, we compute $\mathcal{D}^{(n)}(\gamma^{\mathcal{H}}, \gamma^{\mathcal{V}})$ and stop iterating if $\mathcal{D}^{(n)} - \mathcal{D}^{(n-1)}$ falls below a pre-defined threshold θ.

In the case of discrete states, we have not been able to prove yet that the horizontal and vertical priors agree, i.e. that $\mathcal{D}(\gamma^{\mathcal{H}}, \gamma^{\mathcal{V}})$ converges to zero. However, we have observed experimentally that in practice $\gamma^{\mathcal{H}}$ and $\gamma^{\mathcal{V}}$ converge.

In the continuous case, as $\gamma^{\mathcal{H}}_{i,j}$'s and $\gamma^{\mathcal{V}}_{i,j}$'s are Gaussians, there exists a closed form solution (c.f. appendix B):

$$\mathcal{D}(\gamma^{\mathcal{H}}, \gamma^{\mathcal{V}}) = \frac{1}{2}\sum_{i,j}\left[\frac{\sigma^{\gamma\mathcal{H}^2}_{i,j}}{\sigma^{\gamma\mathcal{V}^2}_{i,j}} + \frac{\sigma^{\gamma\mathcal{V}^2}_{i,j}}{\sigma^{\gamma\mathcal{H}^2}_{i,j}} - 2 + \left(\frac{1}{\sigma^{\gamma\mathcal{H}^2}_{i,j}} + \frac{1}{\sigma^{\gamma\mathcal{V}^2}_{i,j}}\right)(\mu^{\gamma\mathcal{H}}_{i,j} - \mu^{\gamma\mathcal{V}}_{i,j})^2\right] \tag{4.60}$$

which is equal to zero if and only if $\mu^{\gamma\mathcal{H}}_{i,j} = \mu^{\gamma\mathcal{V}}_{i,j}$ and $\sigma^{\gamma\mathcal{H}^2}_{i,j} = \sigma^{\gamma\mathcal{V}^2}_{i,j}$, $\forall(i,j)$. Since $\mu^{\gamma\mathcal{H}}_{i,j} - \mu^{\gamma\mathcal{V}}_{i,j}$ converges to zero and $\sigma^{\gamma\mathcal{H}^2}_{i,j} / \sigma^{\gamma\mathcal{V}^2}_{i,j}$ converges to one (c.f. appendix C), for the T-SSM $\mathcal{D}(\gamma^{\mathcal{H}}, \gamma^{\mathcal{V}})$ converges to zero. Note that we have not yet been able to prove that $\mu^{\gamma\mathcal{H}}_{i,j}$ and $\mu^{\gamma\mathcal{V}}_{i,j}$ actually converge. However, we have observed experimentally that it was the case.

4.7.2 Annealing

Whether the iterative process is initialized with row or column operation may theoretically impact the performance. To soften the influence of one direction on the other, especially during the first few passes where much of the "decisions" are taken, Tokuyasu applied turbo iterative updates in a graduated fashion [Tok01]. Such an approach is inspired by the use of deterministic annealing for HMM design [MRC94].

In the discrete case, we can for instance raise the horizontal and vertical priors $\gamma^{\mathcal{H}}_{i,j}(q_{i,j})$'s and $\gamma^{\mathcal{V}}_{i,j}(q_{i,j})$'s to the power of a variable that we will call τ in the forward-backward equations. We then multiply them by a normalization constant such that the new horizontal and vertical priors sum to one. In the continuous case, we can divide the variance $\sigma^{\gamma\mathcal{H}}_{i,j}$ by τ. In both cases, τ will first be close to zero, which means that, loosely speaking, there will be little communication between both directions during the first few iterations. Then, τ is raised according to an annealing schedule, and the interaction between the horizontal and vertical passes becomes stronger.

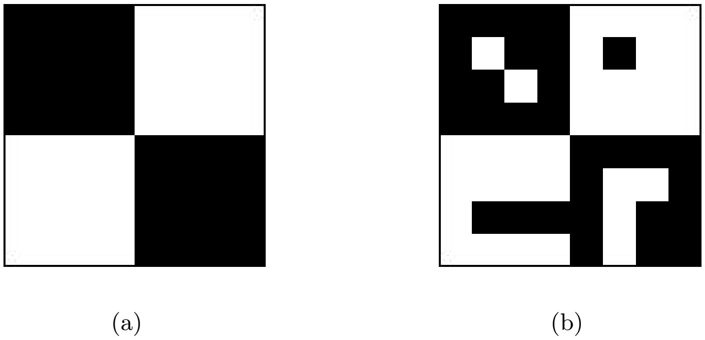
4.8 Experimental Validation

In this section, we will show the potential of the T-HMM and the T-SSM on simple problems. In the following, we will focus on the decoding problem. In the discrete and continuous cases, we will first describe the problem of interest. Note that in both cases, we are not interested in modeling the following problems with a true 2-D HMM but with a set of horizontal and vertical 1-D HMMs. Then we present the ML solution and finally compare the turbo solution to the ML solution. For a fast prototyping, we decided to implement the following experiments in Matlab.

4.8.1 Discrete case

Description of the problem

To validate the T-HMM we consider the problem of decoding a binary image corrupted by a bit-flip noise (c.f. Figure 4.4).



(a) (b)

Figure 4.4: Examples of (a) a 8×8 binary image and (b) a corrupted version of the same image (bit-flip noise).

The states of our system are the possible values of a pixel in the original image (0 or 1). The transition probabilities describe the statistics of the original image. We will assume the following transition probabilities:

$$\begin{aligned} a^{\mathcal{H}}_{0,0} = a^{\mathcal{H}}_{1,1} = a^{\mathcal{V}}_{0,0} = a^{\mathcal{V}}_{1,1} &= 1 - a && (4.61) \\ a^{\mathcal{H}}_{0,1} = a^{\mathcal{H}}_{1,0} = a^{\mathcal{V}}_{0,1} = a^{\mathcal{V}}_{1,0} &= a && (4.62) \end{aligned}$$

The observations emitted by our system are the possible values of a pixel in the corrupted image (0 or 1). Let b be the probability of a bit flip. Then the emission probabilities are given by:

$$\begin{aligned} b_0(0) = b_1(1) &= 1 - b && (4.63) \\ b_0(1) = b_1(0) &= b && (4.64) \end{aligned}$$

We assume that the initial occupancy probability is uniformly distributed to keep the number of parameters in our system as small as possible:

$$\pi^{\mathcal{H}}_0 = \pi^{\mathcal{H}}_1 = \pi^{\mathcal{V}}_0 = \pi^{\mathcal{V}}_1 = \frac{1}{2} \qquad (4.65)$$

To summarize, our HMM has two parameters: a and b. In the following we will assume that the values of these parameters are known as our focus is on decoding.

Description of the ML solution

To find the ML solution, we assume that a set of states on a row, a column or a diagonal is a state of a 1-D HMM. The Viterbi algorithm can thus be applied in a straightforward manner on this 1-D HMM. Obviously, this is still computationally very intensive as such an HMM has a very large number of states. However, depending on the chosen isolating element, the amount of computation can vary.

In the following we will use square images of size $N \times N$. Thus, the complexity of the decoding is the same if we consider rows or columns. If we consider a sequence of states on a row to be a single state, then the corresponding 1-D HMM has 2^N states. The number of operations required by the Viterbi algorithm is hence on the order of:

$$(N-1) \times 2^N \times 2^N = (N-1) \times 4^N \tag{4.66}$$

In the case where we choose the diagonal as the isolating element, the number of operations required by the Viterbi algorithm is on the order of:

$$2 \times (2^1 \times 2^2 + 2^2 \times 2^3 + ... + 2^{N-1} \times 2^N) = 2^4 \times \sum_{i=0}^{N-2} 4^i \approx \frac{4}{3} \times 4^N \tag{4.67}$$

Although the complexity is still exponential in the size of the data, the decrease is significant. Thus, in the following experiments, we use the diagonal as an isolating element to find the ML solution.

Experimental results

All the results we present in the following are for the 8×8 image considered on Figure 4.4(a). We carried out experiments with different images but, as we obtained similar results, they will not be presented here. Even for this simple system which contains two states and for the small images considered the time required to find one ML solution was approximately 6 sec. on a 2 GHz Pentium 4 with 1 GB Ram.

We now have to set the value of the parameter a. For the original pattern under consideration the length of a white or a black region is 4 pixels. The *duration* of a state is defined as the expected number of observations emitted successively by this state. The duration of the states of the considered system is equal to $\frac{1}{1-(1-a)} = \frac{1}{a}$ [Rab89]. We thus set $a = \frac{1}{4}$.

The first results are presented on Figure 4.5 as the difference between the log-likelihood of the ML solution and the T-HMM solution without annealing. Obviously, this is a positive quantity and the lower this value the better the T-HMM solution approximates the ML solution. To obtain meaningful results, for a given bit-flip rate we generated 1,000 corrupted images and averaged the log-likelihoods. For this particular set of experiments, only limited improvement was obtained after 25 iterations.

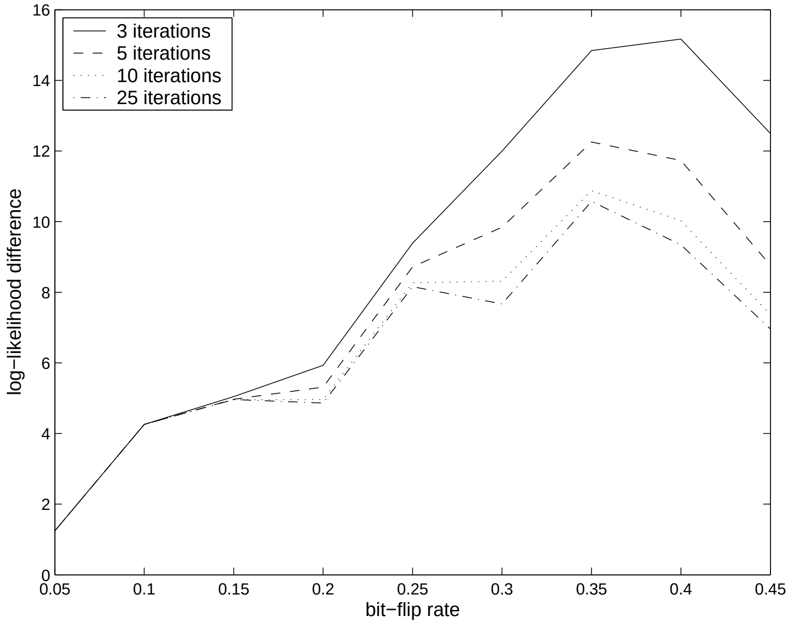


Figure 4.5: Simulation results for the pattern considered in Figure 4.4(a) for various numbers of iterations for the T-HMM. Difference between the log-likelihood of the ML solution and T-HMM solution without annealing versus bit-flip rate.

On Figure 4.6, we compare the results of the T-HMM solutions without and with annealing. It it interesting to note that, for a small number of iterations, annealing decreases the performance. This is not surprising as the annealing softens the communication process between the horizontal and vertical passes and thus, a greater number of iterations is required to reach agreement. On the other hand, for a large number of iterations annealing greatly improves the performance, especially for a high bit-flip rate.

On Figure 4.7, we present the result of the decoding process of the corrupted pattern considered on Figure 4.4. For this example, no annealing was applied. Even for this difficult configuration, the T-HMM manages to find the ML solution which,

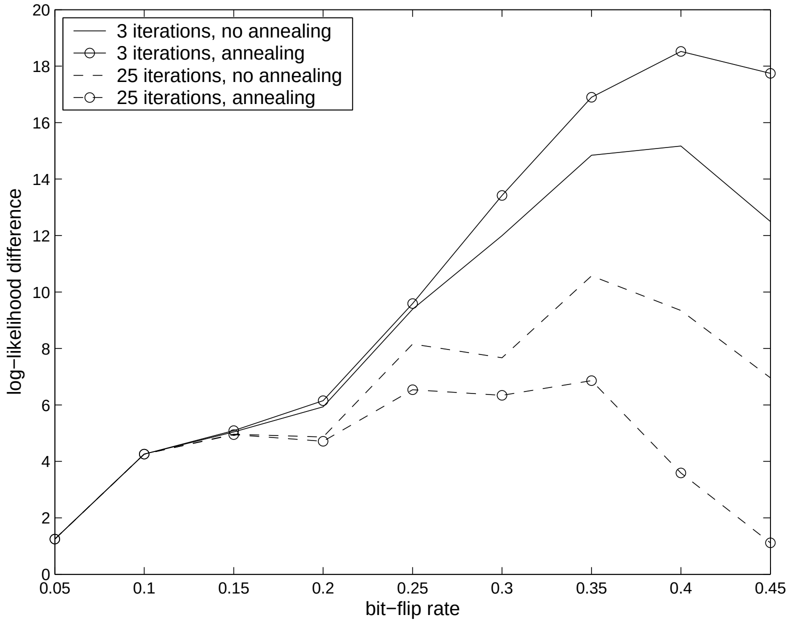


Figure 4.6: Comparison between the T-HMM solutions with and without annealing for two different numbers of iterations. Difference between the log-likelihood of the ML and T-HMM solutions versus bit-flip rate.

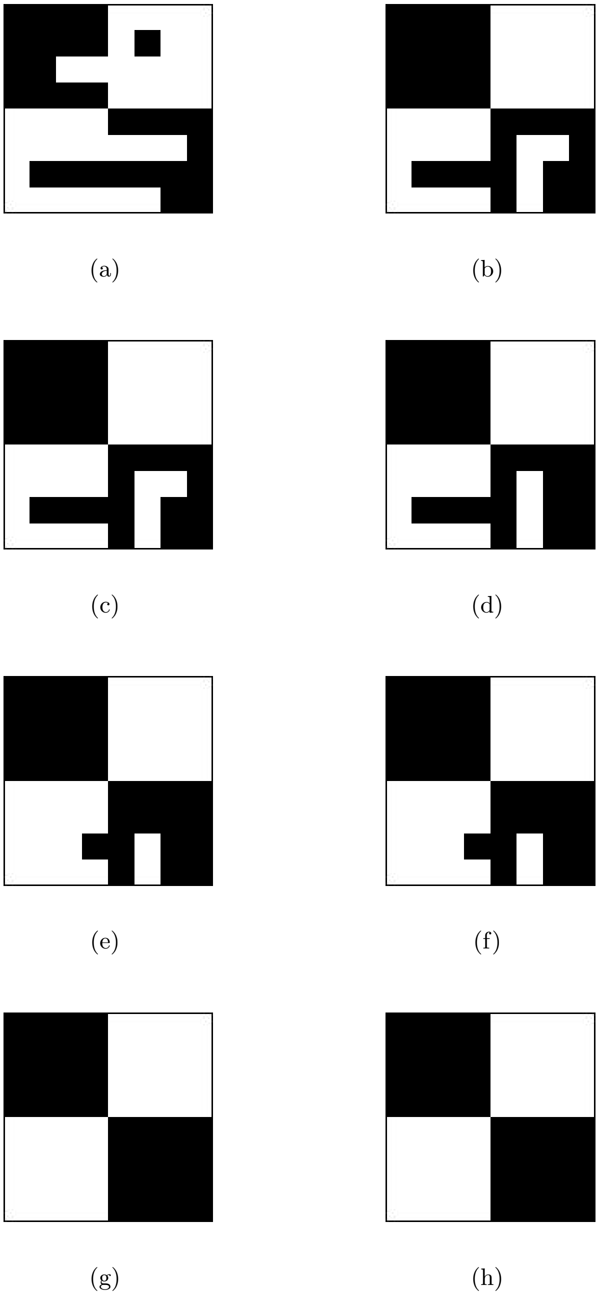
in this case, is the original image. Note that a fairly large number of iterations is required to reach the ML solution (40). However, even with 40 iterations, the amount of computation of the T-HMM is still very low compared to the amount of computation required by the ML solution as the time required for one horizontal and one vertical iteration is on the order 3 ms.

Finally, we compare on Figure 4.8 the convergence properties of the T-HMM solutions without and with annealing when decoding the image of Figure 4.4. While the evolution of the symmetric divergence is irregular when no annealing is applied it is much more regular with annealing. The down-side is that, in the latter case, the convergence is also much slower.

4.8.2 Continuous case

Description of the system

To validate the T-SSM we consider the problem of decoding a signal embedded in additive white Gaussian noise. The states of our system are the possible values of the original signal at a given position. The transition probabilities describe the



(a) (b)

(c) (d)

(e) (f)

(g) (h)

Figure 4.7: Example of a decoding for the corrupted pattern considered on Figure 4.4. The left and right columns correspond respectively to the results after the horizontal and vertical passes respectively. (a) and (b) 1 iteration, (c) and (d) 2 iterations, (e) and (f) 10 iterations, (g) and (h) 40 iterations.

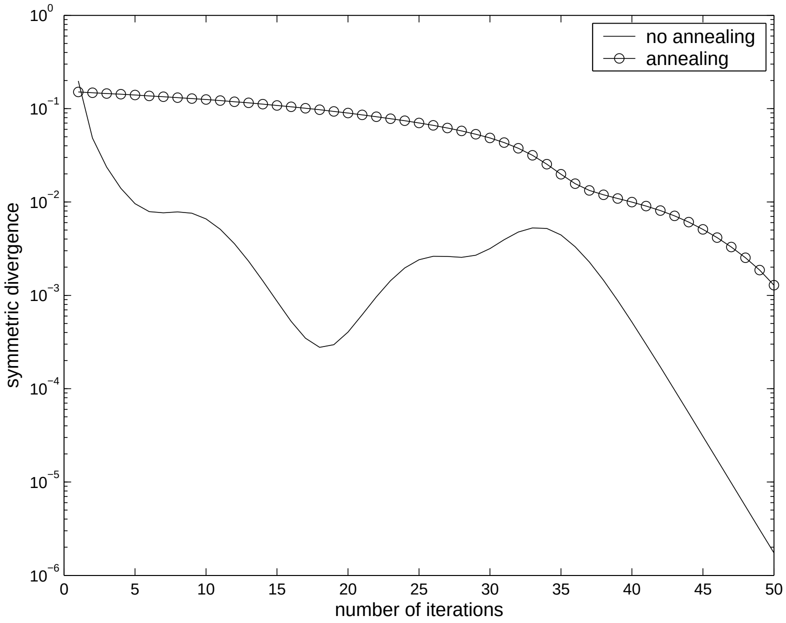


Figure 4.8: Comparison of the convergence properties of the T-HMM solutions with and without annealing for the decoding problem of Figure 4.4. Symmetric divergence (c.f. equation 4.60) versus number of iterations.

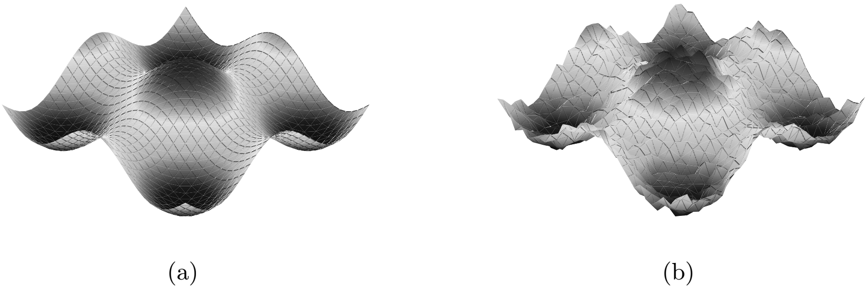


Figure 4.9: Examples of (a) a 32x32 plane wave and (b) a corrupted version of the same plane wave (additive Gaussian noise).

statistics of the original signal. We will assume the following transition probabilities:

$$s_{i,j}^{\mathcal{H}\ 2} = s_{i,j}^{\mathcal{V}\ 2} = s^2 \ , \forall (i,j) \text{ with } i \neq 1 \text{ or } j \neq 1 \tag{4.68}$$

As each position, the system emits an observation. We choose the emission probability to be of the following form:

$$f_{i,j} = 1 \text{ and } {\sigma_{i,j}}^2 = \sigma^2 \ , \forall (i,j) \tag{4.69}$$

We also assume that the initial occupancy probability is non-informative, i.e.:

$$s_{i,1}^{\mathcal{H}\ 2} \rightarrow \infty \ , \forall i \text{ and } s_{1,j}^{\mathcal{H}\ 2} \rightarrow \infty \ , \forall j \tag{4.70}$$

To summarize, the considered T-SSM has only two parameters: s^2 and σ^2. In the following, we will also assume that the values of these parameters are known as our focus is on decoding.

Description of the ML solution

Using equation , the joint likelihood is given by:

$$\log P(O, Q|\lambda) = -\frac{1}{2} \sum_{i,j} \left[\frac{(q_{i,j} - o_{i,j})^2}{\sigma^2} + \frac{(q_{i,j} - q_{i,j-1})^2}{s^2} + \frac{(q_{i,j} - q_{i-1,j})^2}{s^2} \right] + C \tag{4.71}$$

with obvious boundary conditions for $i = 1$ or I and $j = 1$ or J. C is a constant which is independent of the $q_{i,j}$'s. To find the best sequence of states Q^*, we set $\partial \log P(O, Q|\lambda)/\partial q_{i,j} = 0$, $\forall (i,j)$ and obtain the following system of $I \times J$ equations with $I \times J$ unknowns:

$$q_{i-1,j} + q_{i+1,j} + q_{i,j-1} + q_{i,j+1} - q_{i,j} \left(\frac{s^2}{\sigma^2} + 4 \right) = -o_{i,j} \left(\frac{s^2}{\sigma^2} \right), \forall (i,j) \tag{4.72}$$

The solution to this system does not depend on s^2 and σ^2 separately but on the ratio s^2/σ^2.

In the following, we consider that $I = J = N$. The complexity of solving a linear system of N^2 equations with N^2 unknowns in the general case is on the order of N^6 operations. However, if we order equations properly, this system is *banded* with bandwidth N. Hence, the complexity of solving this system is on the order of N^4 operations [QSS00]. While this is much lower than the complexity of the general case, this might be too demanding if N is large.

Finally, if $s^2 \ll \sigma^2$ this system of equations is *ill-conditioned* [QSS00], i.e. a very small perturbation on the observations $o_{i,j}$ (due to noise) or on the parameters (due to estimation errors) might lead to completely different solutions, an unwanted effect.

Experimental results

All the results we present in the following are for the 32×32 image considered on Figure 4.8.2 (a).

We now have to the set the value of the parameter s^2. The original signal under consideration is the following plane wave.

$$S(i,j) = \cos\left[\frac{3\pi}{2}\left(-1+2\frac{i-1}{N-1}\right)\right] + \cos\left[\frac{3\pi}{2}\left(-1+2\frac{j-1}{N-1}\right)\right] \qquad (4.73)$$

$$i = 1,..,32, i = 1,..,32$$

A reasonable estimate of s^2 is:

$$\hat{s}^2 = \frac{1}{2\pi}\int_0^{2\pi}\left[\cos\left(\frac{3\pi t}{N-1}\right) - \cos\left(\frac{3\pi(t+1)}{N-1}\right)\right]^2 dt \qquad (4.74)$$

$$= 1 - \cos\left(\frac{3\pi}{N-1}\right) \qquad (4.75)$$

The first results are presented on figure 4.10 as the difference between the log-likelihood of the ML solution and the T-HMM solution without annealing. To obtain meaningful results, for a given ratio s^2/σ^2 we generated 1,000 corrupted images and averaged the log-likelihoods. For this problem, little improvement was obtained after 25 iterations. On Figure 4.11 we compare the results of the T-SSM solutions without and with annealing. A difference with the T-HMM is that, in our experiments, the performance did not decrease for a small number of iterations (typically 3). In the case where the variance of the measurement noise is very large, the performance is greatly improved with the annealing, as was the case for the T-HMM.

Finally we compare on Figure 4.12 the convergence properties of the T-SSM solutions without and with annealing. The evolution of the symmetric divergence is regular for both cases which is to be contrasted with the T-HMM. We believe that this is due to the continuous nature of the states of the T-SSM while the states of the T-SSM are discrete. Obviously, the convergence of the T-SSM with annealing is much slower than the T-SSM without annealing.

4.9 Conclusion

In this chapter, we introduced the turbo hidden Markov model (T-HMM) as an efficient approximation of the intractable 2-D HMM. The T-HMM can be defined as a set of interconnected horizontal and vertical 1-D HMMs that communicate through an iterative process by inducing prior probabilities on each other. We also considered the turbo state-space model (T-SSM) which is the extension of the T-HMM

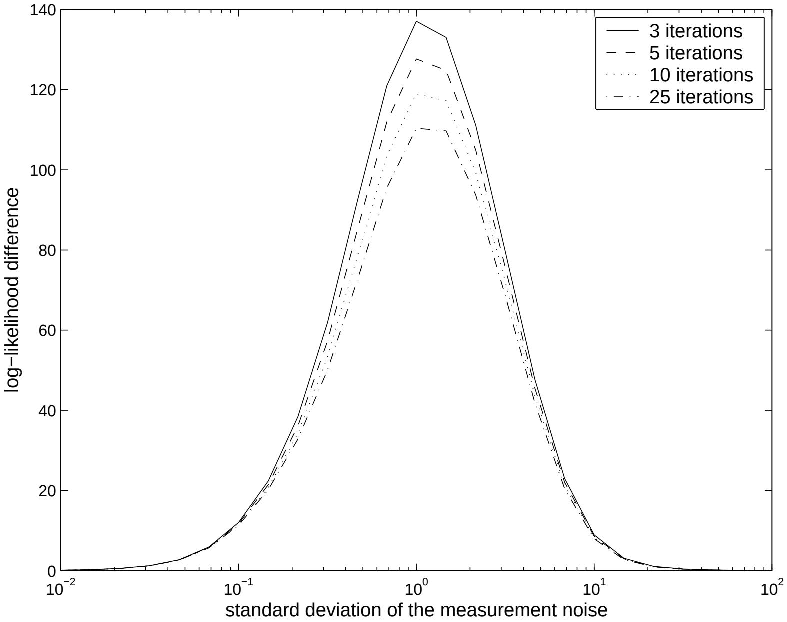


Figure 4.10: Simulation results for the signal considered on Figure 4.8.2 for various numbers of iterations for the T-SSM. Difference between the log-likelihood of the ML solution and T-SSM solution without annealing versus standard deviation of the measurement noise.

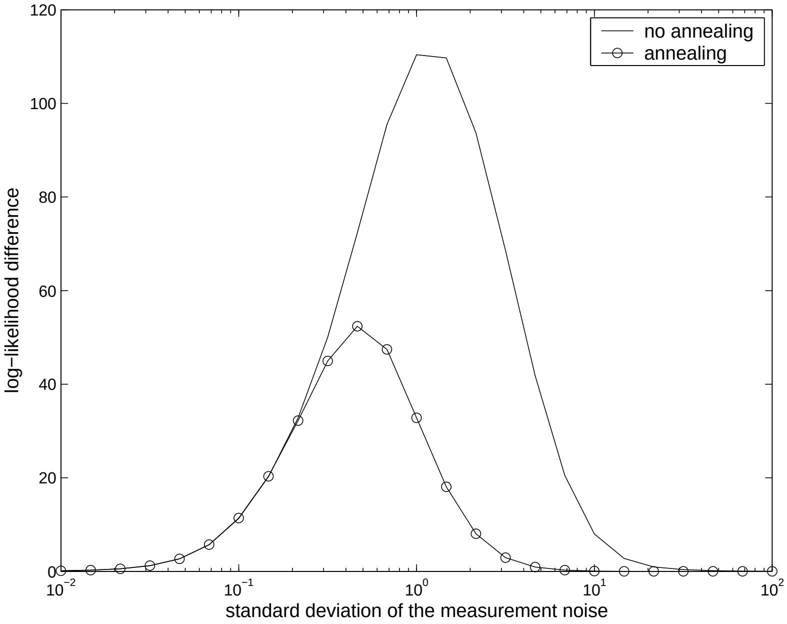


Figure 4.11: Comparison between the T-SSM solutions with and without annealing for 25 iterations. Difference between the log-likelihood of the ML and T-SSM solutions versus standard deviation of the measurement noise.

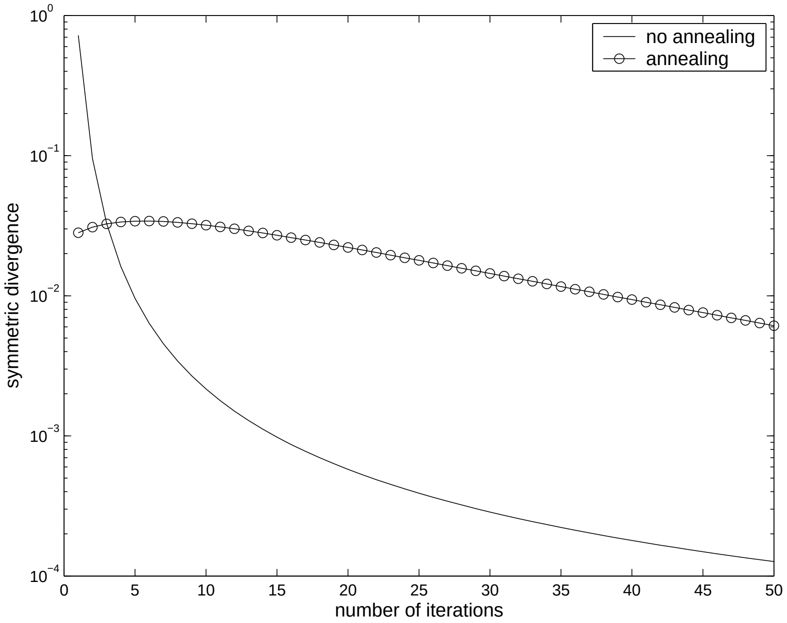


Figure 4.12: Comparison of the convergence properties of the T-SSM solutions with and without annealing for the decoding problem of Figure 4.8.2. Symmetric divergence (c.f. equation 4.60) versus number of iterations.

to continuous states. For both the T-HMM and T-SSM we attempted to provide efficient approximate answers to the three problems of HMM design. We also considered the convergence properties of the iterative process. Finally, we showed the potential of the T-HMM and the T-SSM on simple problems.

However, many convergence properties were observed rather than proved formally, especially in the case of the discrete state T-HMM. As the forward-backward iterations of the T-HMM seem to bear some similarity with those derived for the turbo decoding problem using the graphical model formalism [Fre98], we believe that we could certainly benefit from the large body of work on turbo codes to prove some of these properties.

As for the T-SSM, we assumed a very simple model with linear Markovian dynamics and Gaussian pdf's. Note that linear dynamics represent a first order fit to any true underlying dynamical system, however complex, and often capture most of the salient features of the underlying system. Therefore, we believe that the Gaussian assumption is more restrictive. As an arbitrary pdf can be approximated with a linear combination of Gaussians, we could use mixtures of Gaussians and the equations we derived could be extended in a straightforward manner. However in such a case the complexity of the modified forward-backward iterations would be exponential in the size of the data. The generalized pseudo-Bayesian (GBP) algorithm was proposed to address this issue [BSL93]. The assumption is that it is not important to keep track of distinct mixture histories whose differences occurred more than a given number of observations in the past.

Note that for both the T-HMM and the T-SSM, we use a very simple technique to fuse the horizontal and vertical scores obtained during the modified forward-backward iterations. Obviously, we believe that more elaborate fusion schemes could be envisioned.

In the following chapters, we will consider the application to the T-HMM and the T-SSM to the problem of AFR.

5

Modeling Elastic Facial Distortions

5.1 Introduction

In this chapter, we will first elaborate on our probabilistic model of image mapping which is based on local transformations and neighborhood consistencies (section 5.2). We will then specialize this very general framework to the problem of modeling elastic facial distortions. The goal is to derive a measure of distance between face images which is robust to facial expressions. In section 5.3 we describe the components of our HMM-based transformation model. In section 5.4 we explain how to perform the matching with this model and in section 5.5 we explain how to train it. In section 5.6 we discuss the choice of local features. In section 5.7 we first evaluate the influence of a degradation of the image resolution or an imprecise segmentation of the face on the recognition rate of the proposed approach. We then assess its robustness with respect to facial expressions, but also to a degradation of the image resolution, an imprecise segmentation, illumination, pose and occlusion. In both cases we compare the performance of the proposed approach with BIC (c.f. section 3.2.6). In section 5.8 we provide an analysis of our transformation model. Finally, in section 5.9, we will draw conclusions.

5.2 Framework

Our premise is that a global transformation between two face images may be too complex to be modeled directly and that it should be approximated with a set of *local transformations*. These local transformations should be as simple as possible for efficient implementation but the composition of all local transformations (i.e., the global transformation) should be rich enough to model a wide range of variabilities between face images of the same person. However, if we do not restrict the set of admissible combinations of local transformations, the model might become over-flexible and “succeed” to patch together very different faces.

This observation naturally leads to the second component of our framework: the *neighborhood coherence constraint* whose purpose 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 introduce dependencies in the local transformation selection for the various 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 the probabilistic framework of the turbo hidden Markov model (T-HMM) introduced in chapter 4, and whose complexity is much lower than the 2-D HMM. As discussed in section 3.3.4, the HMM has already been successfully applied to the problems of face detection and face recognition. However, the approach we propose is fundamentally different as our focus is on modeling a transformation between face images while the goal of [Sam94, Nef99] is on modeling the face.

We will now explicate the probabilistic framework. Let us assume that feature vectors are extracted on a grid from the query image I_q. At any location on I_q, the system is assumed to be in some unknown state. If we assume that the horizontal and vertical 1-D HMMs which form the T-HMM are first-order Markovian, the state of the system at a given position depends on the states at the adjacent positions in both horizontal and vertical directions, as quantified by the *transition probabilities*. At each position, an observation is emitted according to the state-conditional *emission probabilities*. In our framework, local transformations are identified with the states of the HMM, and emission probabilities model the local mapping cost. These transformations are “hidden” and information on them can only be extracted from the observations. Transition probabilities relate states of neighboring regions and implement the consistency rules.

The set of possible global transformations, and hence the resulting distance, primarily depends on the allowed local transformations. In this dissertation we consider in particular two types of local transformations: *grid* transformations and *feature* transformations. A grid transformation consists in a local deformation of the feature extraction lattice of the query image. A feature transformation consists in transforming the extracted features directly through the application of a meaningful operator. Note that, if we work in a transform domain, a feature transformation can reflect both geometric or photometric transformations in the pixel domain.

In the following sections of this chapter, we will specialize this framework to compensate for facial expressions using grid transformations. Feature transformations will be the focus of the next chapter.

5.3 The HMM-Based Transformation Model

If we examine our score $P(I_q|I_t, \mathcal{R})$, it is clear that the HMM parameters, denoted $\lambda_{t,\mathcal{R}}$ to reflect their dependence on both I_t and $\mathcal{R}$, may be conveniently separated into face dependent (FD) parameters λ_t, i.e,. parameters that are directly extracted from I_t, and face independent transformation (FIT) parameters $\lambda_{\mathcal{R}}$, i.e., the parameters of the shared transformation model $\mathcal{R}$ which can be reliably estimated by pooling together the training images of all available individuals.

In the following, we will consider respectively the emission and transition probabilities of our HMM. The issue of the initial occupancy probability will be very briefly discussed at the end of this section.

5.3.1 Emission probability

We assume that feature vectors are extracted from I_q on a sparse grid and from I_t on a dense grid. Let $o_{i,j}$ be the observation extracted from I_q at position (i, j) on the sparse grid and let O denote the set of all observations: $O = \{o_{i,j}, i = 1, \ldots, I, j = 1, \ldots, J\}$. Let $q_{i,j}$ be the associated state at position (i, j). Let $m_{k,l}$ be the feature vector extracted from I_t at position (k, l) on the dense grid. We thus have $\lambda_t = \{m_{k,l}, k = 1, \ldots, K, l = 1, \ldots, L\}$. λ_t will be later referred to as the *template*.

It $\tau = (\tau_x, \tau_y)$ is a translation vector, the emission probability, i.e., the probability that at position (i, j) the system emits observation $o_{i,j}$ given that it is in state $q_{i,j} = \tau$, is denoted $b^{\tau}_{i,j} = P(o_{i,j}|q_{i,j} = \tau, \lambda_t, \lambda_{\mathcal{R}})$. A translation τ maps a feature vector $o_{i,j}$ in I_q into a feature vector in I_t that will be denoted $m^{\tau}_{i,j}$ (c.f. Figure 5.1). The emission probability $b^{\tau}_{i,j}$ represents the cost of matching $o_{i,j}$ and $m^{\tau}_{i,j}$ and, thus, models the intra-class variability of the face around position (i, j) that cannot be explained purely by a translation.

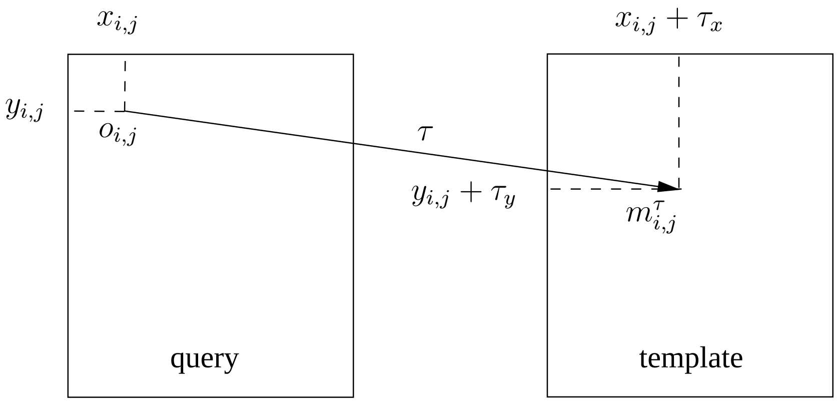


Figure 5.1: Local mapping of a feature vector $o_{i,j}$ in the query image into a feature vector $m^{\tau}_{i,j}$ in the template image.

We model $b^{\tau}_{i,j}$ with a mixture of Gaussians:

$$b^{\tau}_{i,j} = \sum_{k=1}^{K_{i,j}} w^{k}_{i,j} b^{\tau,k}_{i,j} \tag{5.1}$$

This choice is motivated by the fact that linear combinations of Gaussians can approximate arbitrarily shaped densities. $K_{i,j}$ is the number of components at position (i,j), $b^{\tau,k}_{i,j}$'s are the component densities and $w^{k}_{i,j}$'s are the mixture weights and must satisfy the following constraint:

$$\sum_{k=1}^{K_{i,j}} w^{k}_{i,j} = 1 \ , \ \forall (i,j) \tag{5.2}$$

Each component density is a D-variate Gaussian function of the form:

$$b^{\tau,k}_{i,j}(o_{i,j}) = \frac{1}{(2\pi)^{\frac{D}{2}} |\Sigma^{k}_{i,j}|^{\frac{1}{2}}} \exp\left\{ -\frac{1}{2}(o_{i,j} - \mu^{\tau,k}_{i,j})^{T} {\Sigma^{k}_{i,j}}^{(-1)} (o_{i,j} - \mu^{\tau,k}_{i,j}) \right\} \tag{5.3}$$

where $\mu^{\tau,k}_{i,j}$ and $\Sigma^{k}_{i,j}$ are respectively the mean and covariance matrix of the Gaussian, D is the dimensionality of the feature space and $|\cdot|$ denotes the determinant operator. This HMM is non-stationary as Gaussian parameters depend on the position (i,j). During the scoring process, this allows to weight automatically the different parts of the face according to their variability or their discriminatory power according to the considered training criterion.

We now relate $\mu^{\tau,k}_{i,j}$ to $m^{\tau}_{i,j}$ by writing $\mu^{\tau,k}_{i,j}$ as a function of $m^{\tau}_{i,j}$. In the following, we will consider that the observed variability can be modeled with an additive component $\delta^k_{i,j}$:

$$\mu^{\tau,k}_{i,j} = m^\tau_{i,j} + \delta^k_{i,j} \tag{5.4}$$

Note that we could also have considered the more general case where $\mu^{\tau,k}_{i,j}$ is obtained through an affine transformation of $m^\tau_{i,j}$:

$$\mu^{\tau,k}_{i,j} = W^k_{i,j}\zeta^\tau_{i,j} \tag{5.5}$$

where $\zeta^\tau_{i,j} = \begin{bmatrix} 1 \\ m^\tau_{i,j} \end{bmatrix}$ is a vector of size $D+1$ and $W^k_{i,j}$ is a $D \times (D+1)$ matrix. Obviously, this latter approach would enable a more accurate modeling of the variability at the expense of a higher computational cost.

Interestingly, similar equations have been written in the field of ASR for the class of speaker adaptive training (SAT) algorithms [AMSM96]. Especially in [AH96, Boc00], the authors make use of "bi-partite" models for the Gaussian means to separate variabilities. These models are made of two components: one models mostly the speaker dependent (SD) part of the acoustic variabilities, and the other the residual speaker independent (SI) variabilities. The Gaussian means are written as a function f of the SD parameters where the parameters of f are SI, which is exactly what is expressed by equations 5.4 and 5.5.

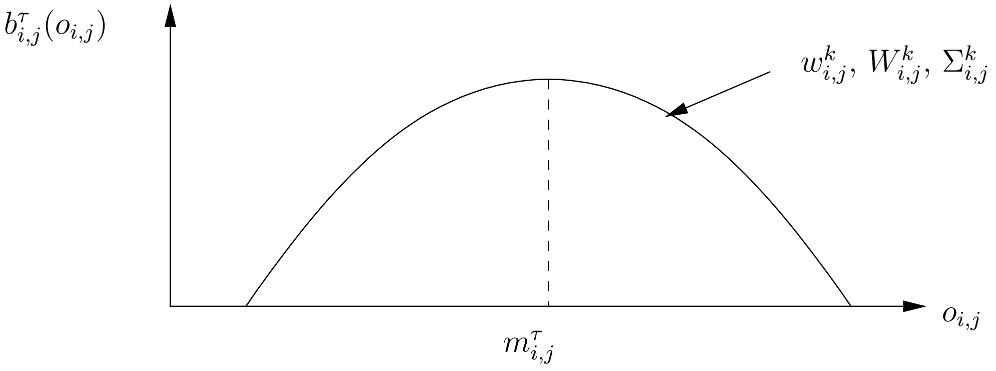


Figure 5.2: Separation of the emission probability parameters into FD parameters ($m^\tau_{i,j}$) and FIT parameters ($w^k_{i,j}$, $\delta^k_{i,j}$ and $\Sigma^k_{i,j}$).

It is interesting to understand the meaning of the previous equations and, especially, the impact of the separation of the emission probability parameters into FD parameters and FIT parameters. While the shape of $b^\tau_{i,j}$ depends only on the FIT parameters $w^k_{i,j}$, $\Sigma^k_{i,j}$ and $W^k_{i,j}$, it should be maximum at $m^\tau_{i,j}$, a FD parameter (c.f. Figure 5.2).

Note that if we have some prior knowledge on the relative locations of the faces in I_q and I_t, e.g. if we know that they are approximately located at the same position, then it is not necessary to try to map a feature vector $o_{i,j}$ in I_q with all the feature vectors $m_{k,l}$ in I_t. Indeed considering all the possible local matchings would be unnecessarily costly. Hence, we should restrict the set of possible translations τ between $o_{i,j}$ and the $m_{k,l}$'s to a region of I_t. $T_{i,j}$ denotes the set of possible translations at position (i, j) and is characterized by the following property:

$$b^{\tau}_{i,j}(o_{i,j}) = 0 \text{ if } \tau \notin T_{i,j} \tag{5.6}$$

The shape and the extent of $T_{i,j}$ depend on several factors such as the accuracy of the face segmentation or the elasticity of the different parts of the face. While the first factor has the same impact on all the different parts of the face, the second one has a different impact as different parts of the face have different elastic properties. Thus, $T_{i,j}$ depends on (i, j) and should be learned during the training phase.

5.3.2 Transition probability

The neighborhood consistency of the transformation is ensured via the transition probabilities of the horizontal and vertical 1-D HMMs that compose the T-HMM. The horizontal and vertical transition probabilities are denoted: $a^{\mathcal{H}}_{i,j}(\tau;\tau') = P(q_{i,j} = \tau'|q_{i,j-1} = \tau)$ and $a^{\mathcal{V}}_{i,j}(\tau;\tau') = P(q_{i,j} = \tau'|q_{i-1,j} = \tau)$. Note that this corresponds to a fairly simple model of the elasticity of the face where each part of the face is linked to its horizontal and vertical neighbors with springs.

Invariance to global shift in face images is a desirable property. Hence, if $\tau' = \tau + \delta\tau$, we choose $a^{\mathcal{H}}$ and $a^{\mathcal{V}}$ to be of the form:

$$\begin{aligned} a^{\mathcal{H}}_{i,j}(\tau;\tau+\delta\tau) &= a^{\mathcal{H}}_{i,j}(\delta\tau) && (5.7) \\ a^{\mathcal{V}}_{i,j}(\tau;\tau+\delta\tau) &= a^{\mathcal{V}}_{i,j}(\delta\tau) && (5.8) \end{aligned}$$

We can apply further constraints on the transition probabilities to reduce the number of free parameters in our system. For instance, we can assume separable transition probabilities. If $\delta\tau = (\delta\tau_x, \delta\tau_y)$, then:

$$\begin{aligned} a^{\mathcal{H}}_{i,j}(\delta\tau) &= a^{\mathcal{H}x}_{i,j}(\delta\tau_x) \times a^{\mathcal{H}y}_{i,j}(\delta\tau_y) && (5.9) \\ a^{\mathcal{V}}_{i,j}(\delta\tau) &= a^{\mathcal{V}x}_{i,j}(\delta\tau_x) \times a^{\mathcal{V}y}_{i,j}(\delta\tau_y) && (5.10) \end{aligned}$$

We can also assume parametric transition probabilities. If I_t and I_q have the same scale and orientation, then the horizontal transition probabilities could have the following form:

$$a^{\mathcal{H}}_{i,j}(\delta\tau) \propto \exp\left\{-\frac{1}{2}\left[\left(\frac{\delta\tau_x}{\sigma^{\mathcal{H}x}_{i,j}}\right)^2 + \left(\frac{\delta\tau_y}{\sigma^{\mathcal{H}y}_{i,j}}\right)^2\right]\right\} \tag{5.11}$$

A similar formula can be derived for vertical transition probabilities. Another idea to reduce the number of transition probability parameters would be to use the face symmetry. If (i, j') is the symmetric of (i, j) with respect to the line of symmetry of the face, then we can impose the following constraints:

$$a^{\mathcal{H}}_{i,j}(\delta\tau_x, \delta\tau_y) = a^{\mathcal{H}}_{i,j'}(-\delta\tau_x, \delta\tau_y) \tag{5.12}$$

$$a^{\mathcal{V}}_{i,j}(\delta\tau_x, \delta\tau_y) = a^{\mathcal{V}}_{i,j'}(-\delta\tau_x, \delta\tau_y) \tag{5.13}$$

$a^{\mathcal{H}}_{i,j}$ and $a^{\mathcal{V}}_{i,j}$ model respectively the horizontal and vertical elastic properties of the face at position (i, j) and are part of the face transformation model $\mathcal{R}$. Note that using multiple horizontal and vertical transition probabilities at different locations enables to model the different elastic properties of the various parts of the face.

To limit the number of possible output transition probabilities at each state we discard unlikely transitions, i.e. unreasonable distortions of the face. The set of possible transition probabilities $\delta\theta$ between positions $(i, j - 1)$ (resp. $(i - 1, j)$) and (i, j) is denoted $\Delta T^{\mathcal{H}}_{i,j}$ (resp. $\Delta T^{\mathcal{V}}_{i,j}$). Thus, $\Delta T^{\mathcal{H}}_{i,j}$ and $\Delta T^{\mathcal{V}}_{i,j}$ are characterized by the following equations:

$$a^{\mathcal{H}}_{i,j}(\delta\tau) = 0 \text{ if } \delta\tau \notin \Delta T^{\mathcal{H}}_{i,j} \quad a^{\mathcal{V}}_{i,j}(\delta\tau) = 0 \text{ if } \delta\tau \notin \Delta T^{\mathcal{V}}_{i,j} \tag{5.14}$$

As was the case for the set of permissible translations $T_{i,j}$, $\Delta T^{\mathcal{H}}_{i,j}$ and $\Delta T^{\mathcal{V}}_{i,j}$ depend on the elastic properties of the face at the different positions and thus should be learned during the training phase.

Finally, we consider the remaining set of HMM parameters – the initial occupancy probabilities. We assume herein that the initial occupancy probability distribution is uniform, to ensure invariance to global translations of face images.

5.4 Recognition with the Transformation Model

In this section, we address the issue of estimating the score $P(O|\lambda_t, \lambda_{\mathcal{R}})$ with our transformation model based on the T-HMM. As discussed in the previous chapter (c.f. sections 4.4 and 4.5) there exists two possible solutions to solve this problem. The first approach computes two scores, a horizontal one $P^{\mathcal{H}}(O|\lambda_t, \lambda_{\mathcal{R}})$ and a vertical one $P^{\mathcal{V}}(O|\lambda_t, \lambda_{\mathcal{R}})$, and fuses them. The second one looks for the best sequence of states Q^* and performs the following approximation: $P(O|\lambda_t, \lambda_{\mathcal{R}}) \approx P(O, Q^*|\lambda_t, \lambda_{\mathcal{R}})$.

However, the main problem with the second approach is that choosing a set of locally optimal states Q^*, as suggested in section 4.5, may not lead to a valid sequence of states in the case where the states are not fully connected. For the problem of interest, we are in such a case as we prune unlikely transition probabilities to reduce the computational cost (c.f. section 5.3.2). Note that we are all the more likely to end up with an invalid sequence of states during the first few iterations, i.e. when the horizontal and vertical passes have not had time yet to reach an agreement. Thus, this approach may require a fairly large number of iterations.

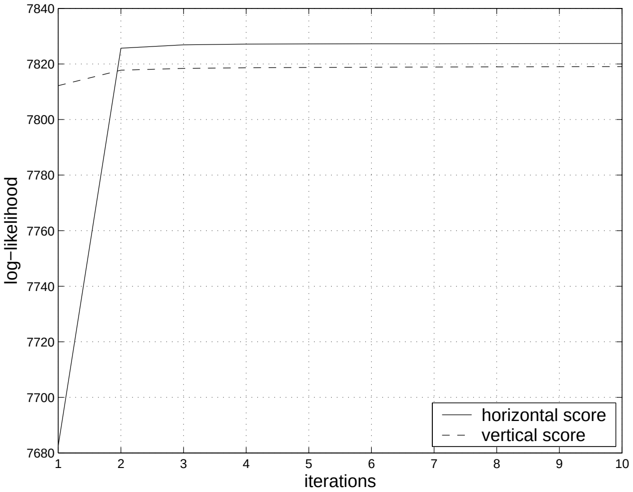


Figure 5.3: Typical case of convergence of the horizontal and vertical log-likelihoods as a function of the number of turbo iterations (starting with a horizontal pass and without annealing).

On the other hand, for the first approach the horizontal and vertical likelihoods are extremely fast to converge for the problem under consideration in the case where we do not perform annealing (c.f. Figure 5.3). Although annealing may favorably impact the recognition rate, it requires a larger number of iterations and therefore an increased computational cost.

Therefore, in all our experiments, to estimate $P(O|\lambda_t, \lambda_{\mathcal{R}})$ we chose the first approach and did not perform annealing.

5.5 Training the Transformation Model

In this section, we consider the issue of estimating the parameters $\lambda_{\mathcal{R}}$ of the face transformation model. We remind the reader that the $\lambda_{\mathcal{R}}$ parameters are the emission probability parameters $w_{i,j}^k$, $\delta_{i,j}^k$ and $\Sigma_{i,j}^k$, the set of permissible translations $T_{i,j}$, the transition probability parameters $a_{i,j}^{\mathcal{H}}$ and $a_{i,j}^{\mathcal{V}}$ and the set of permissible transitions $\Delta T^{\mathcal{H}}_{i,j}$ and $\Delta T^{\mathcal{V}}_{i,j}$. In this section, we first very briefly introduce the basic idea of the ML estimation of the HMM, as applied to our problem, and which is based on the EM principle. We then explicate the E-step and M-step of the estimation. Finally, we describe the training procedure operationally.

5.5.1 Maximum likelihood estimation

Training the transformation model requires a set of pairs of images $\{(I^p_t, I^p_q), p = 1, ..., P\}$ that belong to the same person. The goal of the ML estimation is to adjust $\lambda_{\mathcal{R}}$ to maximize the likelihood:

$$\prod_{p=1}^{P} P(I^p_q | I^p_t, \mathcal{R}) \tag{5.15}$$

Let us consider the case where we have one pair of images (I_t, I_q). The re-estimation formulas can be derived directly by maximizing Baum's auxiliary function, which, in the case of interest, takes the following form:

$$\mathcal{Q}(\lambda_{\mathcal{R}} | \lambda'_{\mathcal{R}}) = \sum_{Q} P(Q | O, \lambda_t, \lambda'_{\mathcal{R}}) \log P(O, Q | \lambda_t, \lambda_{\mathcal{R}}) \tag{5.16}$$

As explained in section 4.6, for the T-HMM we do not have one $\mathcal{Q}$ function but two: a horizontal one $\mathcal{Q}^{\mathcal{H}}$ and a vertical one $\mathcal{Q}^{\mathcal{V}}$. While this is not a problem for re-estimating the transition probabilities, this could be an issue for the emission probabilities in the case where the horizontal and vertical passes do not reach agreement as they are both horizontal and vertical parameters. The solution we suggested was simply to sum the horizontal and vertical statistics to re-estimate the emission probability parameters.

The Baum-Welch algorithm can be readily interpreted as an implementation of the EM algorithm in which the E-step is the calculation of the auxiliary function $\mathcal{Q}$, or more precisely of $\mathcal{Q}^{\mathcal{H}}$ and $\mathcal{Q}^{V}$ in our case, and the M-step corresponds to the maximization over $\lambda_{\mathcal{R}}$.

Note that the ML criterion can be shown to be optimal if certain conditions hold, such as model correctness and infinite training data. However, in our case, the true data source is unlikely to be an HMM and only limited training data is available. Therefore, other training objective functions could be considered, especially discriminative ones, such as the maximum mutual information (MMI) [Nor96] or the minimum classification error (MCE) [JCL96].

5.5.2 E-step

During the E-step, one performs the modified forward-backward iterations to estimate the horizontal and vertical occupancy probabilities $\gamma^{\mathcal{H}}_{i,j}(\tau) = P^{\mathcal{H}}(q_{i,j} = \tau|O, \lambda_t, \lambda_{\mathcal{R}})$ and $\gamma^{\mathcal{V}}_{i,j}(\tau) = P^{\mathcal{V}}(q_{i,j} = \tau|O, \lambda_t, \lambda_{\mathcal{R}})$ respectively. Let $k_{i,j}$ be the Gaussian mixture index at position (i, j). We also define $\gamma^{\mathcal{H}}_{i,j}(\tau, k) = P^{\mathcal{H}}(q_{i,j} = \tau, k_{i,j} = k|O, \lambda_t, \lambda_{\mathcal{R}})$ (resp. $\gamma^{\mathcal{V}}_{i,j}(\tau, k) = P^{\mathcal{V}}(q_{i,j} = \tau, k_{i,j} = k|O, \lambda_t, \lambda_{\mathcal{R}})$), the probability of being in state $q_{i,j} = \tau$ at position (i, j) during the horizontal (resp. vertical) pass with the k-th mixture component accounting for $o_{i,j}$. These quantities can be estimated as follows:

$$\gamma^{\mathcal{H}}_{i,j}(\tau, k) = \gamma^{\mathcal{H}}_{i,j}(\tau)\frac{w^{k}_{i,j} b^{\tau,k}_{i,j}(o_{i,j})}{b^{\tau}_{i,j}(o_{i,j})} \qquad \gamma^{\mathcal{V}}_{i,j}(\tau, k) = \gamma^{\mathcal{V}}_{i,j}(\tau)\frac{w^{k}_{i,j} b^{\tau,k}_{i,j}(o_{i,j})}{b^{\tau}_{i,j}(o_{i,j})} \tag{5.17}$$

The following quantities are also necessary to re-estimate the transition probabilities: $\xi^{\mathcal{H}}_{i,j}(\tau, \tau + \delta\tau) = P^{\mathcal{H}}(q_{i,j} = \tau, q_{i,j+1} = \tau + \delta\tau|O, \lambda_t, \lambda_{\mathcal{R}})$ and $\xi^{\mathcal{V}}_{i,j}(\tau, \tau + \delta\tau) = P^{\mathcal{V}}(q_{i,j} = \tau, q_{i+1,j} = \tau + \delta\tau|O, \lambda_t, \lambda_{\mathcal{R}})$. These horizontal and vertical quantities can be computed as follows:

$$\xi^{\mathcal{H}}_{i,j}(\tau, \tau + \delta\tau) = \frac{\alpha^{\mathcal{H}}_{i,j}(\tau) a^{\mathcal{H}}_{i,j}(\delta\tau) b^{\mathcal{V}}_{i,j+1}(o_{i,j+1}) \beta^{\mathcal{H}}_{i,j+1}(\tau + \delta\tau)}{\sum_{\tau}\sum_{\delta\tau} \alpha^{\mathcal{H}}_{i,j}(\tau) a^{\mathcal{H}}_{i,j}(\delta\tau) b^{\mathcal{V}}_{i,j+1}(o_{i,j+1}) \beta^{\mathcal{H}}_{i,j+1}(\tau + \delta\tau)} \tag{5.18}$$

$$\xi^{\mathcal{V}}_{i,j}(\tau, \tau + \delta\tau) = \frac{\alpha^{\mathcal{V}}_{i,j}(\tau) a^{\mathcal{V}}_{i,j}(\delta\tau) b^{\mathcal{H}}_{i+1,j}(o_{i+1,j}) \beta^{\mathcal{V}}_{i+1,j}(\tau + \delta\tau)}{\sum_{\tau}\sum_{\delta\tau} \alpha^{\mathcal{V}}_{i,j}(\tau) a^{\mathcal{V}}_{i,j}(\delta\tau) b^{\mathcal{H}}_{i+1,j}(o_{i+1,j}) \beta^{\mathcal{V}}_{i+1,j}(\tau + \delta\tau)} \tag{5.19}$$

5.5.3 M-step

As explained in section 4.6, the horizontal (resp. vertical) $\mathcal{Q}$ function can be written as the sum of I horizontal (resp. J vertical) $\mathcal{Q}$ functions that correspond to the I horizontal (resp. J vertical) 1-D HMMs. These $\mathcal{Q}$ functions can be themselves separated into the sums of 3 terms which correspond respectively to the initial occupancy, transition and emission probabilities. As we assumed the initial occupancy to be non-informative, we now only consider the emission and transition probability parameters. Note that the following equations are straightforward extensions of the derivations of [Rab89] and [Bil98].

Emission probabilities

As for emission probabilities we have to sum horizontal and vertical statistics, we use the following notation $\gamma_{i,j}(\tau, k) = \frac{1}{2}\left(\gamma^{\mathcal{H}}_{i,j}(\tau, k) + \gamma^{\mathcal{V}}_{i,j}(\tau, k)\right)$. For Gaussian mixtures, the part of the $\mathcal{Q}$ function which corresponds to the emission probability can be subdivided into two parts. The first one corresponds to the mixture weights while the second one depends on the component parameters.

For the mixture weights, we obtain the following quantity:

$$\sum_{\tau}\sum_{k}\gamma_{i,j}(\tau,k)\log w_{i,j}^{k} \tag{5.20}$$

Adding the Lagrange multiplier η and using the constraint that $\sum_k w_{i,j}^k = 1$, we have to maximize the following quantity:

$$\sum_{\tau}\sum_{k}\gamma_{i,j}(\tau,k)\log w_{i,j}^{k} + \eta\left(1-\sum_{k} w_{i,j}^{k}\right) \tag{5.21}$$

Taking the partial derivative with respect to $w_{i,j}^k$ and equating to zero, we get the following estimate $\hat{w}_{i,j}^k$ of $w_{i,j}^k$:

$$\hat{w}_{i,j}^{k} = \frac{1}{\eta}\sum_{\tau}\gamma_{i,j}(\tau,k) \tag{5.22}$$

Now summing over k, we obtain:

$$\hat{w}_{i,j}^{k} = \frac{\sum_{\tau}\gamma_{i,j}(\tau,k)}{\sum_{\tau}\sum_{k}\gamma_{i,j}(\tau,k)} = \sum_{\tau}\gamma_{i,j}(\tau,k) \tag{5.23}$$

For the Gaussian components, we obtain the following quantity:

$$\begin{aligned} &\textstyle\sum_{\tau}\sum_{k}\gamma_{i,j}(\tau,k)\log b_{i,j}^{\tau,k}(o_{i,j}) \\ = -\tfrac{1}{2}&\textstyle\sum_{\tau}\sum_{k}\gamma_{i,j}(\tau,k)\left(\log|\Sigma_{i,j}^{k}| + (o_{i,j}-\mu_{i,j}^{\tau,k})^{T}\Sigma_{i,j}^{k\,(-1)}(o_{i,j}-\mu_{i,j}^{\tau,k})\right) \end{aligned} \tag{5.24}$$

In the following, we consider the case where $\mu_{i,j}^{\tau,k} = m_{i,j}^{\tau} + \delta_{i,j}^{k}$. If we take the partial derivative with respect to $\delta_{i,j}^k$, we get:

$$\sum_{\tau}\gamma_{i,j}(\tau,k)\Sigma_{i,j}^{k\,(-1)}(o_{i,j}-m_{i,j}^{\tau}-\hat{\delta}_{i,j}^{k}) \tag{5.25}$$

and if we equate the previous quantity to zero, we obtain the following estimate $\hat{\delta}_{i,j}^k$:

$$\hat{\delta}_{i,j}^{k} = \frac{\sum_{\tau}\gamma_{i,j}(\tau,k)(o_{i,j}-m_{i,j}^{\tau})}{\sum_{\tau}\gamma_{i,j}(\tau,k)} \tag{5.26}$$

Finally, if we take the partial derivative of 5.24 with respect to $\Sigma_{i,j}^{k\,(-1)}$ we get:

$$2S - \mathrm{diag}(S) \tag{5.27}$$

with

$$S = \frac{1}{2}\sum_{\tau}\gamma_{i,j}(\tau,k)\left(\Sigma_{i,j}^{k} - (o_{i,j}-m_{i,j}^{\tau}-\delta_{i,j}^{k})(o_{i,j}-m_{i,j}^{\tau}-\delta_{i,j}^{k})^{T}\right) \tag{5.28}$$

Setting $2S - \mathrm{diag}(S) = 0$ implies that $S = 0$. Thus, if we replace $\delta^k_{i,j}$ by its estimate $\hat{\delta}^k_{i,j}$ we obtain the following estimate $\hat{\Sigma}^k_{i,j}$ of $\Sigma^k_{i,j}$:

$$\hat{\Sigma}^k_{i,j} = \frac{\sum_\tau \gamma_{i,j}(\tau,k)(o_{i,j} - m^\tau_{i,j} - \hat{\delta}^k_{i,j})(o_{i,j} - m^\tau_{i,j} - \hat{\delta}^k_{i,j})^T}{\sum_\tau \gamma_{i,j}(\tau,k)} \tag{5.29}$$

In the previous derivations, we assumed the general case of a full covariance matrix.

Finally, the set $T_{i,j}$ of permissible translations τ at position (i,j) is given by the following equation:

$$\tau \in T_{i,j} \text{ if } \gamma_{i,j}(\tau) > \theta \tag{5.30}$$

where θ is a predefined threshold.

Transition probabilities

In $\mathcal{Q}^{\mathcal{H}}$, the part corresponding to the horizontal transition probability $a^{\mathcal{H}}_{i,j}$ is:

$$\sum_\tau \sum_{\delta\tau} \xi^{\mathcal{H}}_{i,j}(\tau, \tau+\delta\tau) \log a^{\mathcal{H}}_{i,j}(\delta\tau) \tag{5.31}$$

Adding the Lagrange multiplier η and using the constraint that $\sum_{\delta\tau} a^{\mathcal{H}}_{i,j}(\delta\tau) = 1$, we have to maximize the following quantity:

$$\sum_\tau \sum_{\delta\tau} \xi^{\mathcal{H}}_{i,j}(\tau, \tau+\delta\tau) \log a^{\mathcal{H}}_{i,j}(\delta\tau) + \eta \left(1 - \sum_{\delta\tau} a^{\mathcal{H}}_{i,j}(\delta\tau)\right) \tag{5.32}$$

Taking the partial derivative with respect to $a^{\mathcal{H}}_{i,j}(\delta\tau)$ and equating to zero, we get the following estimate $\hat{a}^{\mathcal{H}}_{i,j}(\delta\tau)$ of $a^{\mathcal{H}}_{i,j}(\delta\tau)$:

$$\hat{a}^{\mathcal{H}}_{i,j}(\delta\tau) = \frac{1}{\eta} \sum_\tau \xi^{\mathcal{H}}_{i,j}(\tau, \tau+\delta\tau) \tag{5.33}$$

Now summing over $\delta\tau$, we obtain:

$$\hat{a}^{\mathcal{H}}_{i,j}(\delta\tau) = \frac{\sum_\tau \xi^{\mathcal{H}}_{i,j}(\tau, \tau+\delta\tau)}{\sum_\tau \sum_{\delta\tau} \xi^{\mathcal{H}}_{i,j}(\tau, \tau+\delta\tau)} = \sum_\tau \xi^{\mathcal{H}}_{i,j}(\tau, \tau+\delta\tau) \tag{5.34}$$

Similarly, the optimal estimate $\hat{a}^{\mathcal{V}}_{i,j}(\delta\tau)$ of $a^{\mathcal{V}}_{i,j}(\delta\tau)$ is:

$$\hat{a}^{\mathcal{V}}_{i,j}(\delta\tau) = \sum_\tau \xi^{\mathcal{V}}_{i,j}(\tau, \tau+\delta\tau) \tag{5.35}$$

In formulas 5.34 and 5.35 we assumed unconstrained non-separable non-parametric transition probabilities.

Finally, the sets $\Delta T^{\mathcal{H}}_{i,j}$ and $\Delta T^{\mathcal{V}}_{i,j}$ of permissible horizontal and vertical transitions $\delta\tau$ are given by the following equations:

$$\delta\tau \in \Delta T^{\mathcal{H}}_{i,j} \text{ if } \sum_{\tau} \xi^{\mathcal{H}}_{i,j}(\tau, \tau+\delta\tau) > \theta \quad \delta\tau \in \Delta T^{\mathcal{V}}_{i,j} \text{ if } \sum_{\tau} \xi^{\mathcal{V}}_{i,j}(\tau, \tau+\delta\tau) > \theta \quad (5.36)$$

where θ is a predefined threshold.

While the re-estimation formulas 5.23, 5.26, 5.29, 5.34 and 5.35 were provided for the unlikely case where HMM parameters are estimated with only one pair of images, their extension to the case of multiple pairs of images is straightforward. Indeed, we just have to accumulate the statistics for all pairs of images.

5.5.4 The training operationally

In this section, we describe the operational training of the HMM-based transformation model. The training procedure we used, which is similar to the one implemented in the HMM toolkit (HTK) [YEK$^+$01], was inspired by the vector quantization algorithm [LBG80]. We start with a simple model which contains one Gaussian per mixture (Gpm) and then increment progressively the number of Gpm.

One Gaussian per mixture model

As the EM-based training is an iterative procedure, we have to find a reasonable initial estimate of the HMM parameters before applying the EM iterations. Let us denote $\delta_{i,j}$ and $\Sigma_{i,j}$ the parameters of the Gaussian at position (i,j). As we want $b^{\tau}_{i,j}$ to be maximum when $o_{i,j} = m^{\tau}_{i,j}$, we set $\delta_{i,j} = 0$ for the 1 Gpm system. To obtain a reasonable estimate of $\Sigma_{i,j}$ at each position (i,j), we assume that the query and template images are perfectly aligned and that there is no local distortion and we perform a rigid matching of the template and query images. We denote $m^{\tau}_{i,j} = m_{i,j}$ when $\tau = 0$. Thus, formula 5.29 simplifies in the following manner:

$$\hat{\Sigma}_{i,j} = (o_{i,j} - m_{i,j})(o_{i,j} - m_{i,j})^T \quad (5.37)$$

Note that one more time, this formula is for the case where we have only one couple of training images and that to extend this formula to multiple training images we just have to accumulate the statistics. Note that the idea to perform a rigid matching to initialize $\Sigma_{i,j}$ was somewhat inspired by the forced alignment which is traditionally used in ASR. As for the transition probabilities, we initialized them uniformly. Note that we have tried other initialization schemes for transition probabilities but that the convergence of the training shows very little sensitivity with respect to the initial choice of transition probabilities.

Once the HMM parameters have been initialized, one can re-estimate the parameters using the Baum-Welch algorithm.

Mixture incrementing

Once we have obtained a good estimate for the 1 Gpm system, we can increase the number of Gaussians progressively. All the Gaussians which have been estimated with more than a given number of samples are split into two Gaussians by introducing a small perturbation in the mean. Let $\delta_{i,j}^{-}$ and $\delta_{i,j}^{+}$ be the mean parameters of the two resulting Gaussians. If we assume the diagonal covariance matrix $\Sigma_{i,j}$ to be diagonal, with $\Sigma_{i,j} = \text{diag}\{\sigma_{i,j}[1], ..., \sigma_{i,j}[D]\}$, then the splitting is performed by slightly perturbing the δ offset:

$$\delta_{i,j}^{-}[d] = \delta_{i,j}[d] - \epsilon\sigma_{i,j}[d] \qquad \delta_{i,j}^{+}[d] = \delta_{i,j}[d] + \epsilon\sigma_{i,j}[d] \tag{5.38}$$

where ϵ is the parameter which controls the strength of the perturbation. The weights of the resulting Gaussians are set equal to the weight of the initial Gaussian divided by two and the covariance matrices are left unchanged.

Once the splitting has been performed for each mixture of Gaussians, the model can be re-estimated using the Baum-Welch algorithm. The splitting and re-training can be repeated until the desired number of Gaussians is obtained or as long as there is enough data to split Gaussians. An advantage of increasing progressively the number of Gaussians is that it allows to monitor the recognition performance to find the optimum number of Gaussians per mixture [YEK+01].

5.6 Gabor Features

The choice of the feature set that will be extracted from the query and template images is an issue of paramount importance. However, as our focus in this dissertation is on the classifier, we will rely on already existing local features (c.f. section 3.3). *Gabor* features seem to be among the most popular ones. They have long been successfully applied to the problems of face recognition and detection [MCvdM92, LVB+93, WFKvdM97, Krü97, AMU97, LW02] and facial analysis [Wis97, DBH+99].

Gabor wavelets, which are plane waves restricted by a Gaussian envelope, were introduced to image analysis due to their biological relevance and computational properties [LW02]. Indeed, Gabor wavelets kernels are similar to the 2-D receptive fields profiles of the mammalian cortical simple cells. Moreover, they exhibit desirable characteristics of spatial locality and orientation selectivity, and are optimally localized in the space and frequency domains.

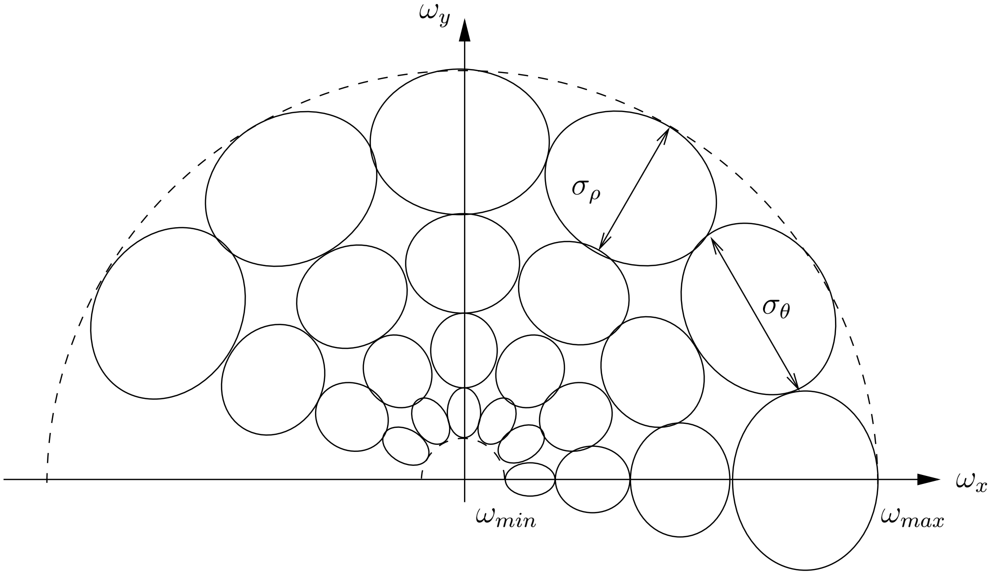


Figure 5.4: Gabor decomposition of the Fourier domain. The lines show the inflexion points of the 2-D Gaussian-shaped filters.

To define a bank of Gabor wavelets, [DFB99] suggests to partition the spectral half plane into M frequency and N orientation bands (c.f. Figure 5.4). The set of filters is defined as follows in the Fourier domain:

$$G_{i,j}(\omega_u, \omega_v) = \exp\left\{-\frac{1}{2}\left[\frac{\omega_u^2}{\sigma_{\rho_i}^2} + \frac{\omega_v^2}{\sigma_{\theta_i}^2}\right]\right\} \quad i = 1, ..., M, j = 1, ..., N \tag{5.39}$$

with:

$$\left(\begin{array}{c} \omega_u \\ \omega_v \end{array}\right) = \left[\begin{array}{cc} \cos(\omega_{\theta_j}) & \sin(\omega_{\theta_j}) \\ -\sin(\omega_{\theta_j}) & \cos(\omega_{\theta_j}) \end{array}\right]\left(\begin{array}{c} \omega_x \\ \omega_y \end{array}\right) - \left(\begin{array}{c} \omega_{\rho_i} \\ 0 \end{array}\right) \tag{5.40}$$

ω_{ρ_i} and σ_{ρ_i} are respectively the radial center and bandwidth and ω_{θ_j} and σ_{θ_i} are respectively the angular center and bandwidth. These parameters are defined as follows:

$$\omega_{\rho_i} = \omega_{min} + \sigma_0 \frac{(f+1)f^{i-1} - 2}{f - 1} \tag{5.41}$$

$$\sigma_{\rho_i} = \sigma_0 f^{i-1} \tag{5.42}$$

$$\omega_{\theta_j} = \frac{(j-1)\pi}{N} \tag{5.43}$$

$$\sigma_{\theta_i} = \frac{\pi \omega_{\rho_i}}{2N} \tag{5.44}$$

with σ_0 given by:

$$\sigma_0 = \frac{\omega_{max} - \omega_{min}}{2} \left(\frac{f-1}{f^M - 1} \right) \tag{5.45}$$

Therefore, to define a bank of Gabor wavelets, one has to set five parameters: ω_{min}, ω_{max}, f, M and N (c.f. Figure 5.5).

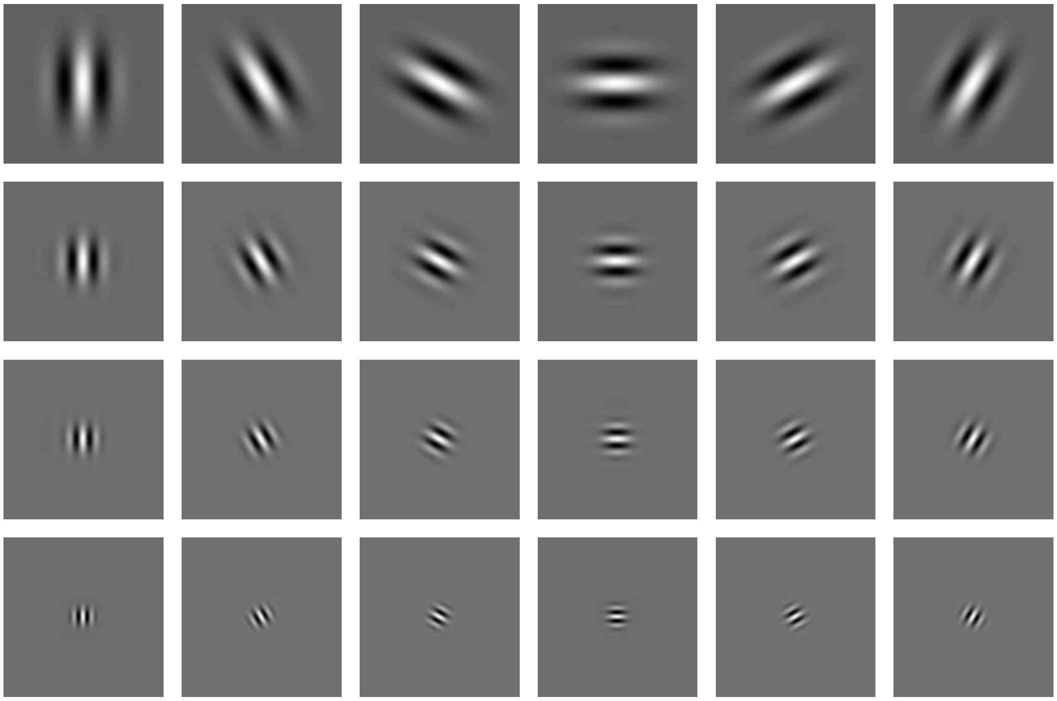


Figure 5.5: The real part of the Gabor kernels at 4 scales and 6 orientations with the following set of parameters: $\omega_{min} = \pi/24$, $\omega_{max} = \pi/3$, $f = \sqrt{2}$.

Gabor responses are obtained through the convolution of an image with the Gabor wavelets. This can be performed in a fast manner in the Fourier domain. If I is the image to be filtered and $g_{i,j}$ the Gabor kernel, then we have:

$$O_{i,j} = I * g_{i,j} \tag{5.46}$$

where $*$ denotes the convolution operator and $O_{i,j}$ is the convolution result. Thus applying the fast Fourier transform operator $\mathcal{F}$, we get:

$$\mathcal{F}\{O_{i,j}\} = \mathcal{F}\{I\}\mathcal{F}\{g_{i,j}\} \tag{5.47}$$

with $\mathcal{F}\{g_{i,j}\} = G_{i,j}$. Now applying the inverse fast Fourier transform operator $\mathcal{F}^{-1}$, we obtain:

$$O_{i,j} = \mathcal{F}^{-1}\left\{\mathcal{F}\{I\}G_{i,j}\right\} \tag{5.48}$$

We use the modulus of these responses as feature vectors.

After preliminary experiments, we chose the following set of parameters for the problem of interest: $M = 4$ scales, $N = 6$ orientations, $\omega_{min} = \pi/24$, $\omega_{max} = \pi/3$, $f = \sqrt{2}$. Features were extracted every 4 pixels in both horizontal and vertical directions from the template images and every 16 pixels from the query images. Note that we tried a finer resolution for both the template and query images but did not obtain any significant increase of the performance at the expense of a much greater computational cost.

5.7 Experimental Validation

In this section, we will first explain our choice of training and test data. We will then briefly discuss the issue of face segmentation. Next we will fine tune on a development set BIC and the proposed approach which will be latter referred to as PMLGT for probabilistic mapping with local grid transformations. BIC was chosen for the baseline as it is one of the only approaches to AFR which focuses on the relationship between face images and as it is one of the most successful approaches to AFR to date (c.f. section 3.2.6). First, we evaluated the influence of a degradation of the image resolution or an imprecise segmentation of the face on the recognition rate of the proposed approach. As our goal is on comparing the robustness of PMLGT and BIC in conditions which are as close to reality as possible, we also assessed the performance of both approaches in various conditions. We assessed the performance of PMLGT and BIC with respect to facial expressions, which is the focus of this chapter, but also illumination, pose and occlusion. Indeed, it is interesting to see how PMLGT and BIC will cop with variabilities which have not been learned.

5.7.1 Choice of the training and test data

To make sure that BIC and PMLGT are not unduly sensitive to a reasonable mismatch between the training and test conditions, we always trained and tested both systems on different databases. For our experiments, we used four face databases: FERET [PMRR00], AR [AR], Yale B [GBK01]and PIE [SBB02] (c.f. appendix D for a brief description of these databases and sample images).

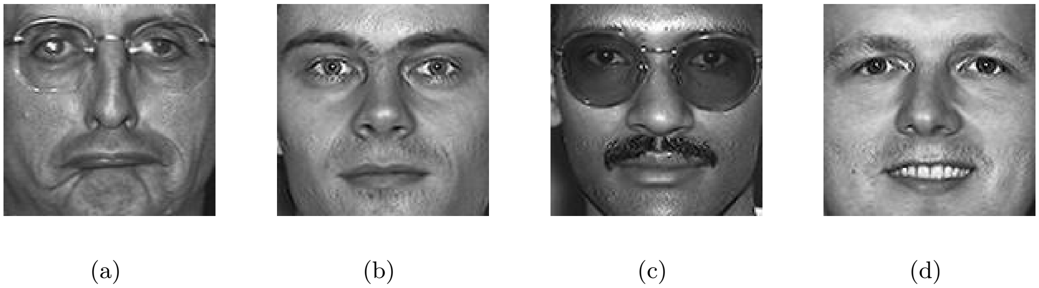
In all the following experiments, we used the FERET face database to train our transformation model. Indeed, this database contains a large number of persons (1,199). As both BIC and PMLGT make the assumption that the intra-class variability is the same for all classes, i.e. for all persons, training these systems with a large number of persons should make them more robust to new individuals. For the training, we used 695 persons. 200 of them are those persons who have an additional

FC image and the other 495 persons were chosen randomly among the remaining persons. We used for each of these persons 2 images: one FA and one FB image. The FA and FB images were successively used as query and template and thus, our system was trained with 1,390 pairs of images. We also used an additional 500 persons as a development set to fine tune the PMLGT and the BIC systems. For these 500 persons, we used their FA and FB images.

In most of the following experiments the test data was extracted from the AR, Yale B and PIE databases.

5.7.2 Face segmentation

The face segmentation is a very important pre-processing step before recognition and the performance of the face segmentation can greatly impact the performance of the subsequent recognition. This is particularly true for global approaches to AFR such as BIC. In the following, we assume that the locations of the centers of the eyes and the tip of the nose are known. These positions were either provided with the database, as is the case for FERET, or manually located. They are used to localize and normalize geometrically face images. First, each image was rotated so that both eyes were on the same line. Then a square box twice the size of the inter-eye distance was centered around the nose. Finally the corresponding region was cropped and resized to 128x128 pixels. (see Figure 5.6 for a few examples of normalized face images).



(a) (b) (c) (d)

Figure 5.6: A few examples of normalized FERET face images.

Note that in section 5.7.5 we will evaluate the impact of an imprecise location of the facial features.

5.7.3 Training BIC

We implemented the BIC approach which is described in section 3.2.6. The main issue is the estimation of the ρ parameter. Indeed, if N is the size of the feature space and M the number of training difference images, estimating the parameter ρ with equation 3.21 requires the set of eigenvalues $\{\lambda_i, i = M+1, ..., N\}$, which is not available. To address this issue, we extrapolated the values λ_i by fitting a function of the form $1/i$ as suggested in [MP97]. For our experiments, we used $\lambda_i \approx a/i$ [1]. The mean square estimate of the value a is given by:

$$a = \frac{1}{M} \sum_{i=1}^{M} i\lambda_i \tag{5.52}$$

As our goal in this section is on comparing the PMLGT and BIC *classifiers*, for a fair comparison we applied BIC to a Gabor representation of the face and not directly on the gray level images. Note that the idea to combine a local representation of the face, such as a Gabor representation, with a global approach to AFR has already been successfully applied in [LW02]. For the problem of interest, the Gabor representation consists of the concatenation of the feature vectors extracted every 4 pixels in both horizontal and vertical directions, which is equivalent to the representation of template images for the proposed approach.

The results are presented on Figure 5.7. The performance increases extremely fast for a small number E of features, reaches a maximum identification rate of 94.6% for $E = 50$ and then decreases slowly. Once the number of features was fixed, we tried to improve the performance by training a mixture of Gaussians for P_F (c.f. section 3.2.6). However, while the likelihood increased significantly, we did not observe any improvement of the performance. Therefore, we used a single Gaussian mixture in the following experiments.

[1] To avoid the explicit summation over i in the evaluation of ρ, we can make use of the following inequality:

$$\frac{1}{n+1} \leq \int_{n}^{n+1} \frac{dx}{x} = \log(n+1) - \log(n) \leq \frac{1}{n} \tag{5.49}$$

By summing from M to $N-1$ on the left and from $M+1$ to N on the right, this yields to:

$$\log(N+1) - \log(M+1) \leq \sum_{i=M+1}^{N} \frac{1}{i} \leq \log(N) - \log(M) \tag{5.50}$$

We thus obtain the following approximation by averaging the lower and upper bounds:

$$\sum_{i=M+1}^{N} \frac{1}{i} \approx \frac{1}{2} \log\left(\frac{N(N+1)}{M(M+1)}\right) \tag{5.51}$$

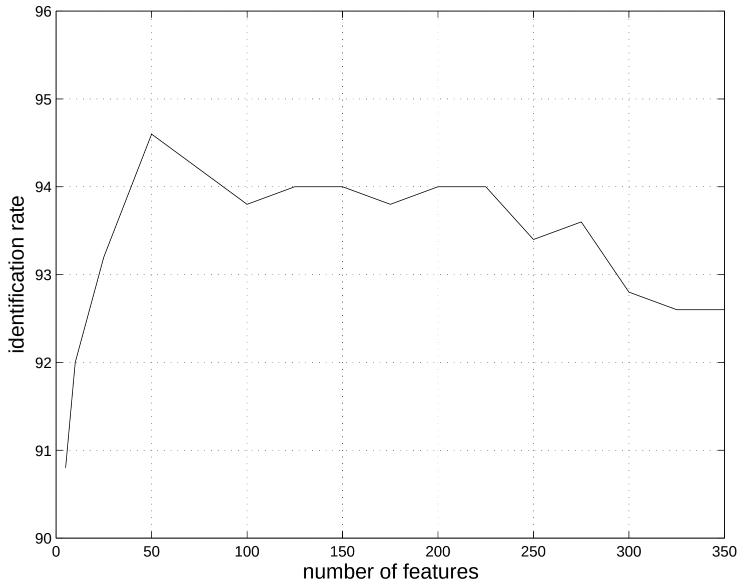


Figure 5.7: Performance of BIC on the FERET development set. Identification rate versus number of features.

5.7.4 Training PMLGT

For emission probabilities, we used diagonal covariance matrices to reduce the computational cost. We used general transition matrices but reduced the number of parameters to estimate by using the face symmetry. Initially, the set of permissible translations $T_{i,j}$ was initialized to a maximum of 8 pixels horizontally and vertically. As the precision of the feature extraction grid is 4 pixels for template images, the maximum number of possible translations (and thus states) at each position is $5 \times 5 = 25$. In the same manner, the sets $\Delta T_{i,j}^{\mathcal{H}}$ and $\Delta T_{i,j}^{\mathcal{V}}$ of permissible transitions was initialized to a maximum of 8 pixels horizontally and vertically. Thus, the maximum number of possible transitions at each position is $5 \times 5 = 25$. For the modified forward-backward iterations, we started with a horizontal pass, then performed a vertical pass and ended with a final horizontal pass, as experimentally, we found that no more than 3 passes were required for the horizontal and vertical scores to converge (c.f. Figure 5.3). This results in a fast matching algorithm. Indeed, running our non-optimized code on a 2 GHz Pentium 4 with 1 GB Ram, it takes on the order of 5 ms to compare two face images with an HMM that contains 16 Gpm.

Training the transformation model was done exactly as described in section 5.5. To re-estimate the 1 Gpm model, we performed 4 training iterations. A Gaussian could only be split if it had been estimated with at least 50 observations. The perturbation factor ϵ was set to 0.01. To re-estimate the models with multiple Gpm, we performed 12 training iterations.

We measured the impact of using multiple mixtures of Gaussians 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. Hence, we tried one mixture for the whole face ($\Sigma^k_{i,j} = \Sigma^k$, $\delta^k_{i,j} = \delta^k$ and $w^k_{i,j} = w^k$) and one mixture at each position (i.e. 49 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 is drawn on Figure 5.8 as a function of the maximum number of Gpm.

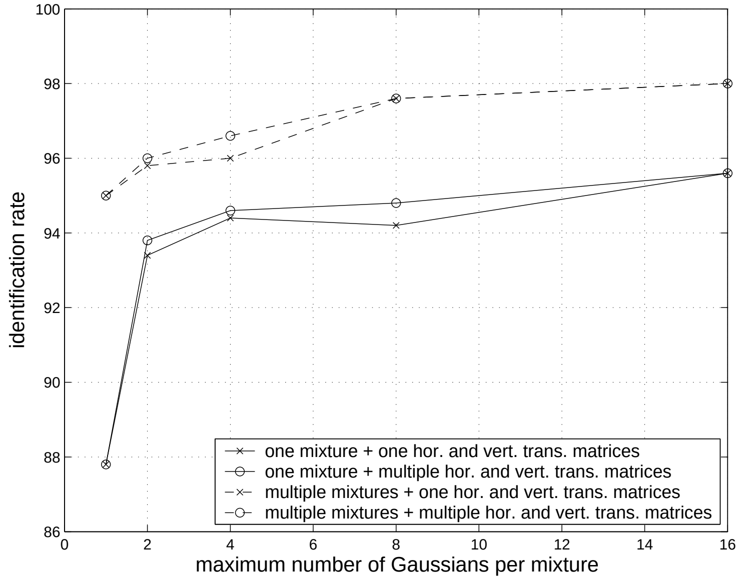


Figure 5.8: Performance of the proposed PMLGT on the FERET development set. Identification rate versus maximum number of Gpm.

We can see that using multiple mixtures has a great impact on the performance compared to the case where we use only one mixture for the whole face, especially for a small number of Gpm. Note that this is not a completely fair comparison as in one case we have the same number of Gpm in both systems but that in the former case, we have 49 mixtures and thus, approximately 49 times as many Gaussians. The latter system could be of particular interest in the case where only limited memory resources are available. On the other hand, using multiple horizontal and vertical transition matrices only has a very limited impact on the performance compared to the case where we use only one horizontal and one vertical transitions matrices for the whole face. Note that it is well-known also in ASR that transition probabilities have little impact on the performance of HMM-based systems. This is due to the relatively high dimensionality of feature vectors (in our case 24). Emission probabilities are generally several orders of magnitude smaller than transition probabilities and thus have a much greater impact on the likelihood.

The best performance we obtained was a 98% identification rate. To carry out our experiments on the AR, Yale B and PIE databases, we used our very best system with multiple mixtures, multiple transition matrices, and a maximum of 16 Gpm. We made sure with McNemar's test that the observed difference between PMLGT and BIC on this development set was significant with more than 99% confidence.

Note that we carried out preliminary experiments to determine whether a more elaborate model of the face variability, such as the one of equation 5.5, could improve the performance. Therefore, we tried an affine model of the form $W_{i,j}^k = (\delta_{i,j}^k \vdots \Pi_{i,j}^k)$ where $\delta_{i,j}^k$, the additive component of the variability, is a vector of size D and $\Pi_{i,j}^k$, the multiplicative component, is a matrix of size $D \times D$. To reduce the number of parameters to estimate, we chose $\Pi_{i,j}^k$ to be diagonal. With such a model, the identification rate was increased up to 99.4%.

5.7.5 Results

We carried out two sets of experiments to evaluate the robustness of BIC and PMLGT with respect to a degradation of the image resolution or a failure of the face segmentation system due to an imprecise location of facial features. We then performed four sets of experiments to compare the performance of BIC and PMLGT in the presence of facial expressions, illumination or pose variations and occlusion. For each set of experiments, we carried out the tests on the database(s) that, we thought, would be the most interesting for the considered variability.

Image resolution

It was shown in [Mog02] that BIC performed very well even for face images with a very coarse resolution (down to 21×12 pixels). Therefore, we carried out a set of experiments on FERET to know how the proposed approach depends on the resolution. We used the same enrollment and test images as in the previous section with the difference that the resolution of face images was downgraded to 64×64, 32×32 and 16×16. The BIC and PMLGT systems were trained exactly as described in sections 5.7.3 and 5.7.4 respectively. Results are presented on Figure 5.9.

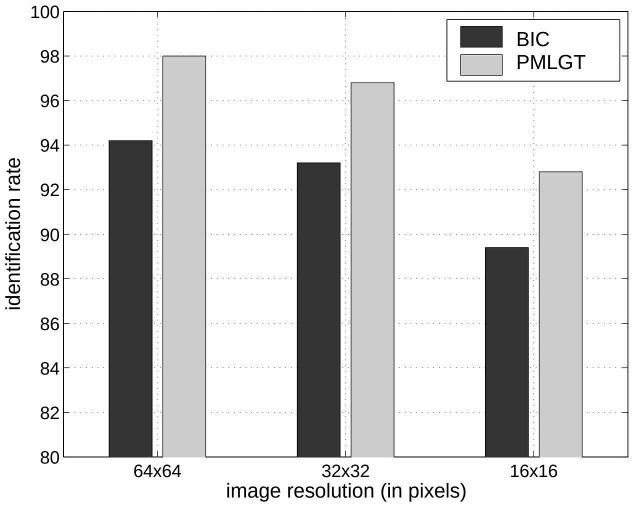


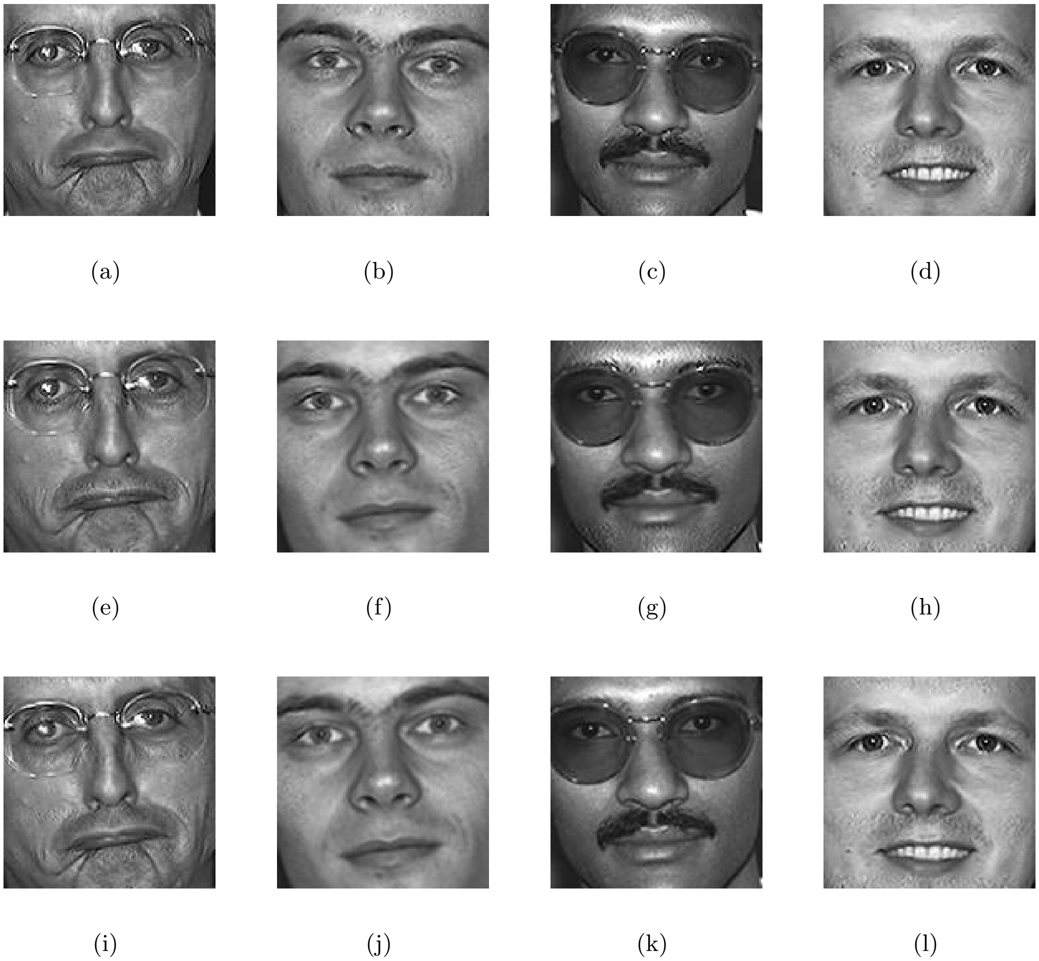
Figure 5.9: Influence of the image resolution. Results on the FERET database.

We can see that the dependence of BIC and PMLGT on the image resolution is similar. Indeed, there is little degradation of the performance down to 32×32 pixels and a significant degradation for 16×16 pixels.

Imprecise segmentation

The robustness of BIC and PMLGT with respect to an imprecise segmentation was evaluated on the FERET database. We used the same enrollment and test images as in the previous section. The only difference is that, at test time the localization of facial features on query images was perturbated with an additive Gaussian noise with mean zero and a varying standard deviation (c.f. Figure 5.10). The localization of facial features for enrollment images was not perturbated. The rational behind this choice is the fact that enrollment is often supervised and thus, an imprecise location of features can be manually corrected. Results are presented on Figure 5.11.

Obviously, PMLGT is much more robust to an imprecise location of facial features than BIC. With McNemar's test we observed with more than 99% confidence that, compared to the case of a perfect segmentation, the decrease of performance for BIC is already significant for a standard deviation of 2 pixels while for PMLGT it is only significant for 3 pixels. We believe that the robustness of PMLGT is due to the local grid transformations which allow more flexibility in the matching. Therefore, we forced the PMLGT to perform a rigid matching by constraining the system to be at each position (i, j) in the state $\tau_{i,j} = (0, 0)$. The results we obtained with this rigid PMLGT for a standard deviation of 1, 2 and 3 pixels were 94.4%, 89.8% and 85.6% respectively, which validates our claim.



(a) (b) (c) (d)

(e) (f) (g) (h)

(i) (j) (k) (l)

Figure 5.10: A few examples of normalized FERET face images. The localization of facial features was perturbated with an additive Gaussian noise with mean zero and standard deviation (a)-(d) 1 pixel, (e)-(h) 2 pixels and (i)-(l) 3 pixels.

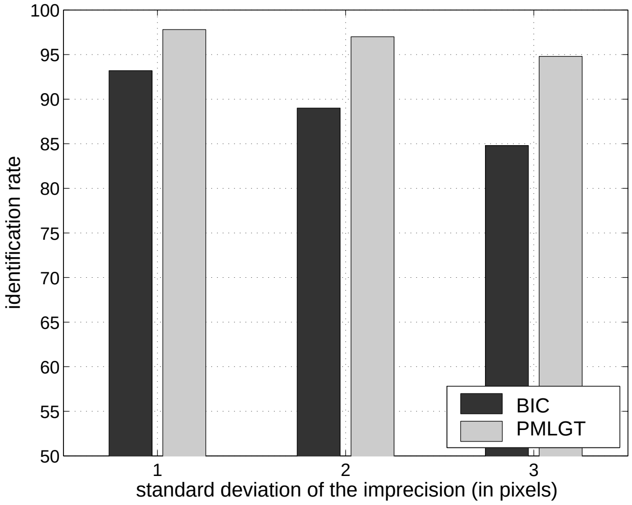


Figure 5.11: Imprecise location of facial features. Results on the FERET database.

Facial expressions

The robustness of BIC and PMLGT with respect to facial expressions was evaluated on the AR face database. All the available persons were used. The images labeled 01, which correspond to the neutral expression, were used as enrollment data and the images 02, 03 and 04, which correspond respectively to the smile, anger and scream expressions, were used as test images (see Figure D.4).

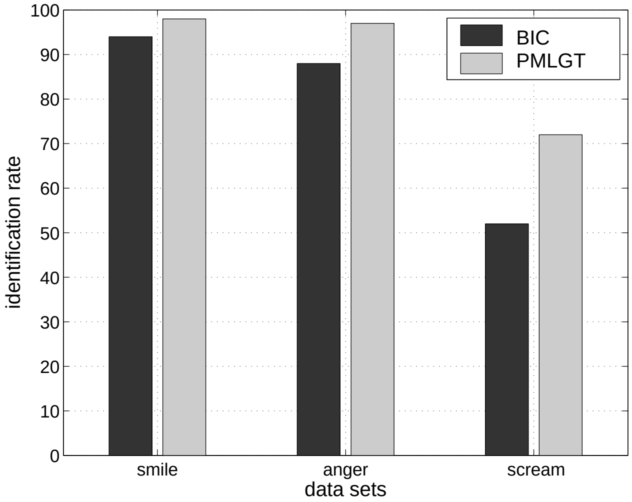


Figure 5.12: Facial expression results on the AR database.

Results are presented on Figure 5.12. The PMLGT algorithm outperforms the BIC algorithm for all expressions. Both BIC and PMLGT perform fairly poorly for extreme facial expressions such as the scream. We however point out that the training data extracted from FERET does not contain such radical expression variations. The average performance on the 3 facial expressions is 78% for BIC and 89% for PMLGT. With McNemar's test, we made sure with more than 99% confidence that the observed difference was significant.

We wanted to assess the impact of local grid transformations on the performance of PMLGT. Hence, we forced the PMLGT to perform a rigid matching as was the case for the previous set of experiments. The average performance of this rigid PMLGT is 87% and thus, there is only a slight degradation compared to the PMLGT. If we perform McNemar's test of significance, we can see that the observed difference cannot be considered significant. Hence, for this particular set of experiments, we conclude that local translations only have a very limited impact on the performance for facial expressions, which is kind of surprising. We will see however in the next chapter that we will draw a different conclusion for different features.

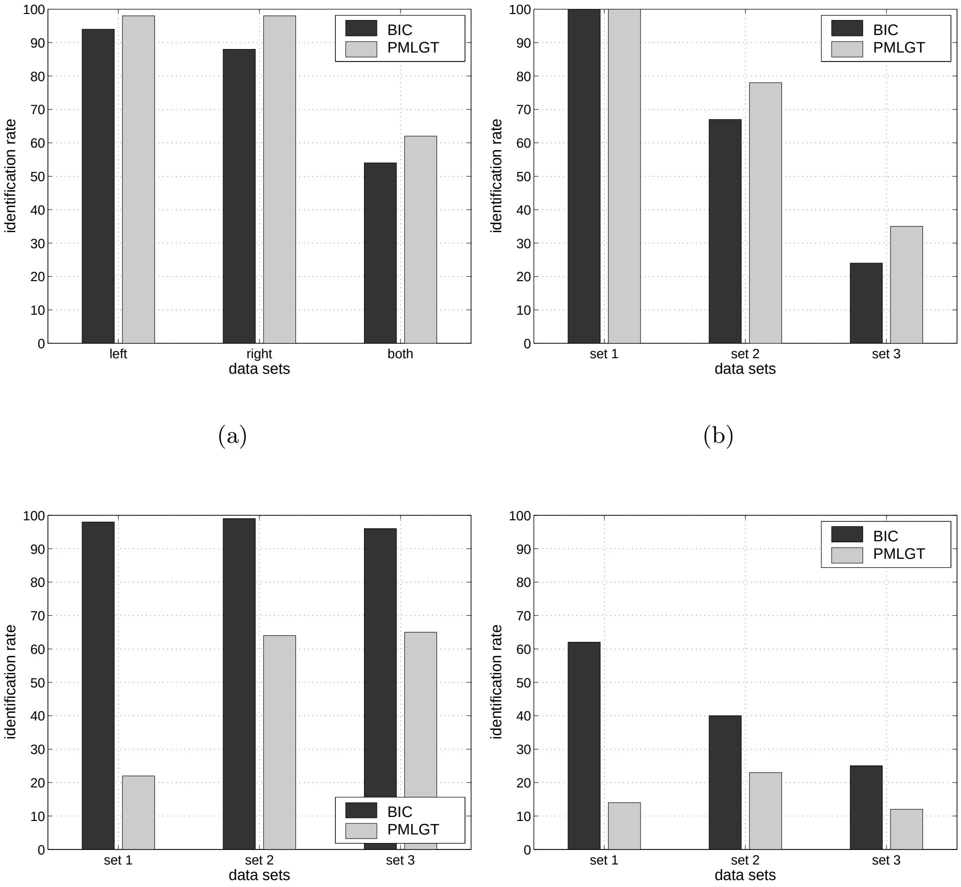
Illumination

The robustness of BIC and PMLGT with respect to illumination variations was evaluated on the AR, PIE and Yale B databases:

- For the AR database, sets 05, 06 and 07, which correspond respectively to the left light on, the right light on and both lights on, were used as test data (see Figure D.4). The neutral expression was used as enrollment image.

- For Yale B, we used those images which correspond to the frontal camera. The image which corresponds to the flash which is directly in the optical axis of the camera was chosen as enrollment image and we used as test data 38 images which correspond to flashes which make an angle between 20^o and 77^o with the optical axis (see Figure D.2). Moreover, these 38 images were subdivided into three data sets according to the angle between the flash and the optical axis of the camera: $20^o \leq \theta \leq 25^o$ for "set 1", $35^o \leq \theta \leq 50^o$ for "set 2" and $60^o \leq \theta \leq 77^o$ for "set 3".

- For the PIE database, experiments were carried out on the sets with and without ambient lighting, which will be later referred to as PIE 1 and PIE 2 (see Figure D.3). Only the images corresponding to the frontal camera were used. For each of the 68 persons, an image corresponding to the pure ambient lighting of PIE 1 was used as enrollment image and the 2×21 other conditions were used as test data. As was the case for Yale B, the 21 images were subdivided into three data sets, according to the angle between the flash and the optical axis of the camera: $\theta \leq 25^o$ for "set 1", $25^o \leq \theta \leq 40^o$ for "set 2" and $40^o \leq \theta \leq 70^o$ for "set 3".



(a) (b) (c) (d)

Figure 5.13: Illumination results on (a) AR, (b) Yale B (c) PIE 1 and (d) PIE2.

Results are presented on Figure 5.13. They are contrasted as PMLGT seems to outperform BIC on AR and Yale B while BIC clearly outperforms PMLGT on PIE 1 and PIE 2. We will now elaborate on these results.

On AR, both algorithms exhibit a good performance for sets 05 and 06, i.e. when half of the face is illuminated but a low performance on set 07, i.e. when the face is illuminated from both sides. This is not surprising as, when only half of the face is corrupted by illumination, the other half can be reliably used to perform recognition. Note that the poor performance of BIC and PMLGT on set 07 can also be explained by the fact that many images seem to be over-illuminated and have a very low contrast. The average performance over the three sets is 79% for BIC and 86% for PMLGT.

On Yale B, the average performance is 61% for BIC and 69% for PMLGT. While both algorithms exhibit a perfect recognition rate for set 1, i.e. when the angle between the optical axis is small, for set 3, i.e. for extreme illumination variations, the performance is poor for both BIC and PMLGT, considering the fact that this database contains only 10 persons.

For PIE 1, the average performance is 98% for BIC and only 46% for PMLGT. To explain the observed difference in performance, we ran the rigid version of our algorithm on PIE 1 and obtained on the average a 69% recognition rate. Thus, in the case where there is a pure illumination variation, the rigid mapping outperforms PMLGT. Indeed, the greater flexibility of PMLGT is a hindrance in this case as, apparently, PMLGT tries to compensate for the illumination variation with local translations. However, we can see that there is still a huge gap in performance between BIC and the rigid mapping and using a global representation of the face seems to provide a certain amount of invariance to illumination.

Finally, on PIE 2 the average identification rate is 45% for BIC and 16% for PMLGT. Note that the fact that BIC outperforms PMLGT on PIE 2 might first be surprising considering that PIE 2 and Yale B images are fairly similar and that PMLGT outperforms BIC on Yale B. A possible explanation is the difference in the choices of the templates. While for Yale B, the template image corresponds to the case where the flash is directly in the optical axis of the camera, for PIE 2 the template image corresponds to an image with pure ambient lighting.

For for all the previous experiments, we performance McNemar's test of significance and, each time a difference in performance was observed on the average identification rates, it could be declared to be significant with more than 99% confidence.

Pose

To assess the robustness of BIC and PMLGT with respect to pose variation, we carried out a set of experiments on the PIE database. We chose the images with neutral expressions from 6 cameras: 05, 07, 09, 29 and 37. These sets were grouped as follows: 07 and 09, 05 and 29, 11 and 37 and correspond approximately to up or down rotations of the head of $\pm 15^o$ and to left or right rotations of $\pm 22^o$ and $\pm 45^o$

respectively (see Figure D.3).

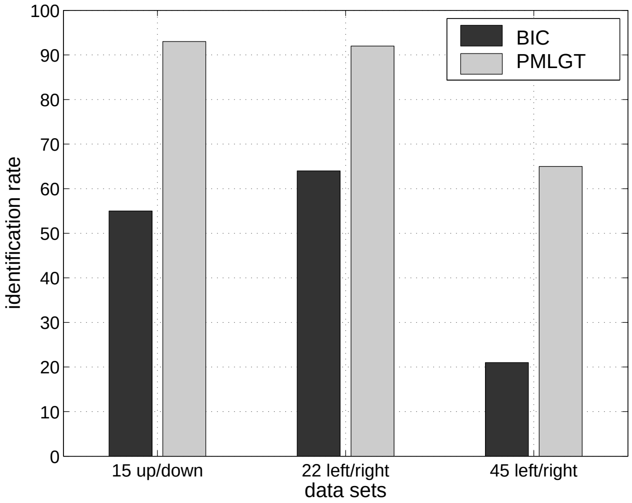


Figure 5.14: Pose results on the PIE database.

Results are presented on Figure 5.15. The PMLGT algorithm outperforms very significantly (i.e. with more than 99% confidence) the BIC algorithm for all poses. The average identification rates are 46% and 82% for BIC and PMLGT respectively. Since we suspected that the difference in performance was primarily due to the grid transformations of the PMLGT, we ran the rigid version of PMLGT and obtained an average identification rate of 60% which seems to validate our claim.

Occlusion

To assess the robustness of BIC and PMLGT with respect to occlusion, we carried out a set of experiments on the AR database. We used as test images the sets 08 and 11 which correspond to an occlusion of the face due to sunglasses and a scarf respectively (see Figure D.3). The image 01 which corresponds to the neutral expression was chosen as the template.

Results are presented on Figure 5.15. Both algorithms exhibit a very poor performance for an occlusion due to sunglasses (on the order of 10% identification rate). For an occlusion due to a scarf, PMLGT clearly outperforms BIC (90% versus 75%) and, using McNemar's test, the observed difference could be considered significant with more than 99% confidence. The reason for the difference in performance for the PMLGT between an occlusion due to sunglasses or a scarf will be explained in the next section.

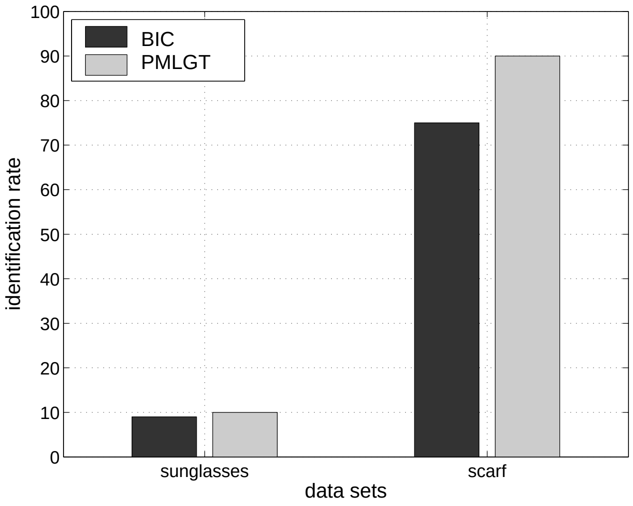


Figure 5.15: Occlusion results on the AR database.

5.8 Analysis

The goal of this section is to analyze which parts of the face are the more variable ones and which parts are the most elastic ones. This analysis was done on the system with multiple mixtures, 1 GpM and multiple transition probabilities.

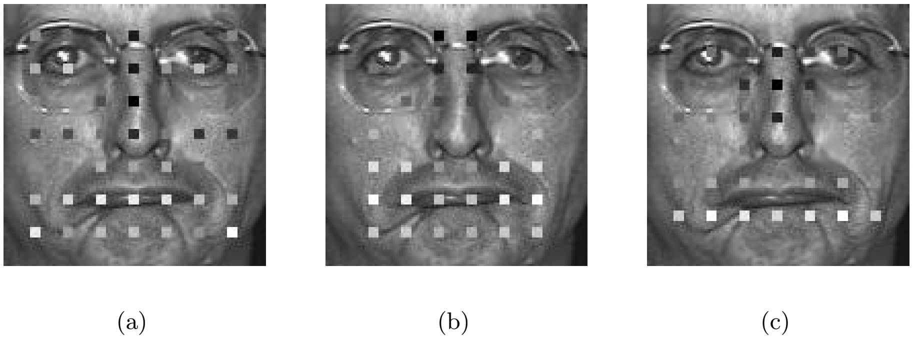
To measure the variability of one part of the face, we computed the entropy of the emission probability. In the case where we have a single Gaussian with a diagonal covariance matrix $\Sigma_{i,j} = \text{diag}\{\sigma_{i,j}[1]^2, ..., \sigma_{i,j}[D]^2\}$, the entropy at position (i, j) is given by (c.f. also appendix B):

$$H_{i,j} = \sum_{d=1}^{D} \log(\sigma_{i,j}[d]) + \frac{D}{2}(1 + \log(2\pi)) \tag{5.53}$$

The greater $H_{i,j}$, the more variable the face around position (i, j). To measure the horizontal and vertical elasticity of a given part of the face, we computed the quantities $\log({\sigma^{\mathcal{H}}_{i,j}}^2)$ and $\log({\sigma^{\mathcal{V}}_{i,j}}^2)$ where ${\sigma^{\mathcal{H}}_{i,j}}^2$ and ${\sigma^{\mathcal{V}}_{i,j}}^2$ are respectively defined as:

$${\sigma^{\mathcal{H}}_{i,j}}^2 = \sum_{\delta\tau} ||\delta\tau||^2 a^{\mathcal{H}}_{i,j}(\delta\tau) \qquad {\sigma^{\mathcal{V}}_{i,j}}^2 = \sum_{\delta\tau} ||\delta\tau||^2 a^{\mathcal{V}}_{i,j}(\delta\tau) \tag{5.54}$$

The results are presented on Figure 5.16. It is clear that the lower part of the face is both the most variable and the most elastic one. This means that the upper part of the face will have a higher contribution during the scoring when comparing two images. This is consistent with findings from other researchers [CWS95] and this explains our own results on occlusion (c.f. the previous section). This suggests



(a) (b) (c)

Figure 5.16: (a) Variability of the different parts of the face: the brighter a dot, the more variable the corresponding part of the face. (b) and (c) Horizontal and vertical elasticity of the different parts of the face respectively: the brighter a dot, the more elastic the corresponding part of the face.

the possibility to discard the lower part of the face for the purpose of AFR, as done for instance by Sanderson and Paliwal in [SP02, San02].

5.9 Conclusion

In this chapter, we elaborated on our framework based on local transformations and neighborhood coherence constraints and specialized it to the problem of modeling facial distortions incurred, for instance, from facial expressions. We detailed our HMM-based model and explained how to train it and how to use it for the recognition problem. It is worthwhile to note that the proposed PMLGT bears some similarity with motion estimation algorithms and especially with MAP estimation of dense motion [Bov00]. We also discussed the important issue of the choice of features. The performance of the PMLGT was assessed using a large dataset of four databases (FERET, Yale B, PIE and AR). A comparison was carried out with the BIC classifier which is one of the most successful approaches to AFR to date.

It was shown that PMLGT compared favorably to BIC for an imprecise segmentation, facial expressions, pose variations and occlusion of the lower part of the face. While grid transformations have a very positive impact on the recognition rate for pose variations, they had only a very limited impact for facial expressions. We will show however in the next chapter that grid transformations may also have a significant for facial expression variations using different features. As for illumination variations, we obtained contrasted results as no algorithm outperforms the other one under all conditions.

However, even for reasonable illumination variations, such as those of PIE with ambient lighting, the drop of performance of PMLGT can be very significant. For radical illumination variations, such as those of Yale B or PIE without ambient lighting, the performance collapses. Obviously grid transformations cannot cop with certain types of variabilities such as the illumination and may even impact unfavorably the performance as the classifier may wrongly try to compensate for illumination variations with small translations.

Clearly, to be robust to illumination, we should compensate for the variation during the pre-processing or incorporate some knowledge about illumination variation into our classifier. In the next chapter, we will consider an approach which uses both strategies.

6

Modeling Illumination Variation

6.1 Introduction

It was shown in the previous chapter that the proposed approach could not cop in a satisfactory manner with a wide range of illumination conditions if we considered only local grid transformations. It was even shown experimentally that, in the case of a pure illumination variation, grid transformations may try to wrongfully "explain" the observed variability and thus decrease the performance. Indeed, for any AFR system, illumination remains one of the most challenging variabilities to cope with as demonstrated during the FERET evaluation [PMRR00] and the facial recognition vendor tests 2000 [BBP01] and 2002 [PGM$^+$03].

It is possible to deal with the illumination at the different stages of the recognition: during the *pre-processing*, the *feature extraction* or the *classification*. Note that the focus of this dissertation is on still intensity images but that it is also possible to cope with the illumination at the *sensing* level by considering other modalities such as infra-red or range images.

Pre-processing algorithms for illumination compensation include general image processing tools such as histogram equalization and gamma correction [Bov00]. Another approach, which is based on Weber's law [1], consists in applying a logarithm

[1] Weber's law states that the change in a stimulus that will be "just noticeable" is a constant ratio of the original stimulus.

transform to the image intensity [AMU97, SK03]. Finally, many pre-processing algorithms consist in separating an image into its reflectance and illumination fields. The assumption is that the luminance varies slowly across the image while sharp changes can occur in the reflectance. The homomorphic filtering [GW92] or the approach suggested in [GB03] are examples of such algorithms.

At the feature extraction stage, the goal is to derive features that are invariant to illumination. Edge maps, derivatives of the gray level and Gabor features were compared in [AMU97] and an empirical study showed that none of these features was sufficient to overcome the variations due to changes in the direction of illumination. Another idea is to *learn* features which are insensitive to illumination variations such as the Fisherfaces [BHK97] (see also section 3.2.4).

Finally, various algorithms have been proposed to cope with the illumination variation at the classification stage. The idea underlying [GBK01] is that the set of images of an object in fixed pose, but under all possible illumination conditions, is a convex cone in the space of images that can be approximated by low dimensional linear subspaces. [BRV02] proposed an approach based on 3-D morphable models which encode both shape and texture information and an algorithm that recovers these parameters from a single face image.

An alternative way of classifying illumination compensation algorithms is to distinguish those that do not require any learning from those that need to learn the illumination variability. One advantage of the former class of algorithms is that they do work on all types of images and not only face images. This is of particular interest for the problem of face detection for instance, when one does not even know whether a face is present in an image or not. On the other hand, such algorithms are limited in the sense that they do not make use of some knowledge about the specific problem at hand. However, the issue of learning-based algorithms is their ability to generalize on novel data. This is especially true for illumination as it is clearly impossible to learn all the possible illumination conditions. So it is of particular interest to know if a certain type of illumination variability is learned, whether the algorithm will be able to generalize on novel illumination conditions.

In this chapter, we consider an approach to illumination compensation which works both at the pre-processing and classification stages. In section 6.2, we first show how to transform, thanks to an additional pre-processing step, the illumination into an additive variability in the feature domain. In section 6.3, we introduce *feature transformations* to compensate for this variability and we explicate the HMM-based transformation model. In section 6.4, we explain how to perform recognition with this model and in section 6.5, we explain how to train it. In section 6.6, we present a series of experiments which demonstrate a dramatically improved recognition rate in the case of illumination variations with no significant degradation of the performance for other variabilities.

6.2 Modeling Illumination

The starting point for modeling illumination is the well-known assumption that an image I can be seen as the product of a reflectance R and an illumination L [Hor86]:

$$I(x,y) = R(x,y) \times L(x,y) \tag{6.1}$$

Applying the logarithm operator, we obtain:

$$\log I(x,y) = \log R(x,y) + \log L(x,y) \tag{6.2}$$

and the illumination turns into an additive term in the pixel domain. If the feature extraction operator $\mathcal{E}$ is linear, such as the convolution, then we obtain:

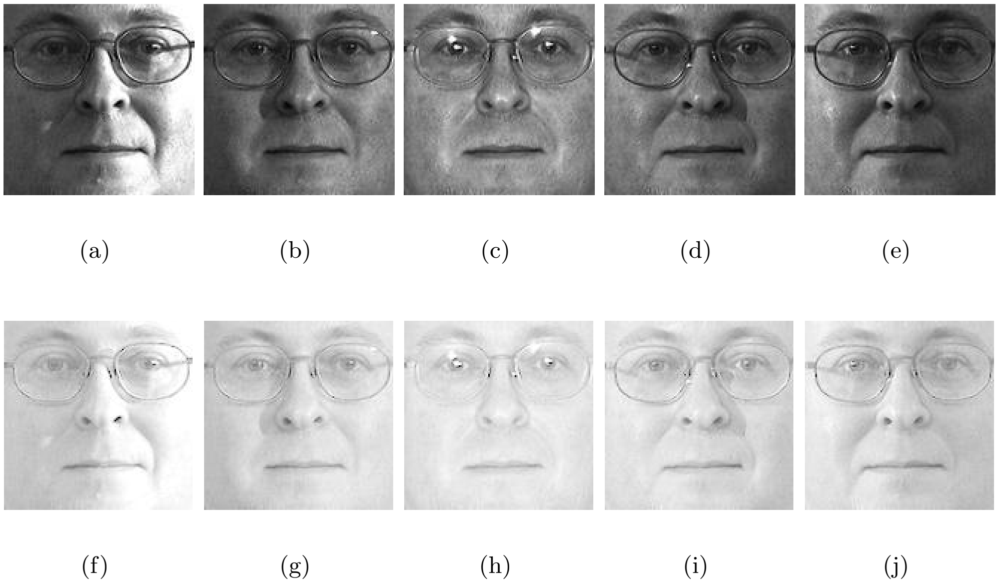
$$\mathcal{E}\{\log I(x,y)\} = \mathcal{E}\{\log R(x,y)\} + \mathcal{E}\{\log L(x,y)\} \tag{6.3}$$

and the illumination remains additive in the feature domain. Note that, as the features we use are the modulus of Gabor responses, the illumination cannot be considered as a perfectly additive term in the feature domain.

As explained in the introductory section, applying the log operator in the pixel domain has been shown to be a particularly efficient pre-processing step to mitigate the influence of illumination. Example face images pre-processed with the log transform are shown on Figures 6.1 and 6.2. We can see, especially on PIE 2 images that the results of this very simple approach can be very impressive as it enables to perfectly distinguish features that were previously barely visible. In the section on experimental results, we will evaluate the influence of the log transform alone on both BIC and PMLGT.

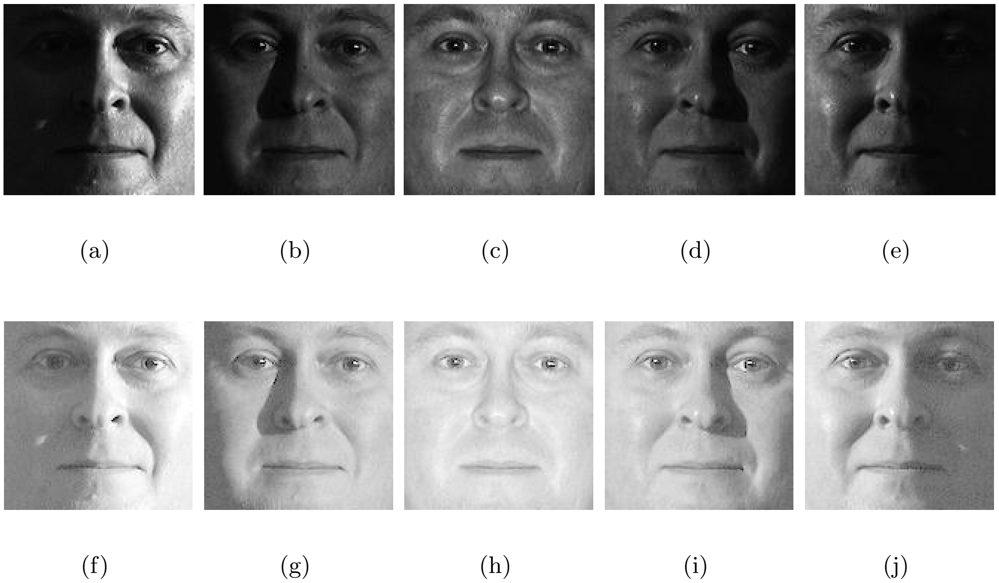
Since the system described in the previous chapter can model additive variabilities, as expressed by equation 5.4, a first idea would be to train the Gaussian mixtures parameters, i.e. w's, δ's and Σ's, not only to model the facial expression variations, but also the various possible illumination conditions. Although this approach might first sound appealing, we believe it is suboptimal for two main reasons:

- The “choice” of Gaussians at adjacent positions would be unconstrained, which is not satisfying as the illumination cannot vary in an arbitrary manner over the face.


(a) (b) (c) (d) (e)

(f) (g) (h) (i) (j)

Figure 6.1: Sample PIE 1 images (i.e. with ambient lighting): (a)-(e) before the log transform and (f)-(j) after the log transform.


(a) (b) (c) (d) (e)

(f) (g) (h) (i) (j)

Figure 6.2: Sample PIE 2 images (i.e. without ambient lighting): (a)-(e) before the log transform and (f)-(j) after the log transform.

- We would confound all sources of variabilities, an unwanted effect as explained in [KPJ01]. Indeed, consider a system that models N independent variabilities, such that the i-th variability can be reasonably modeled with n_i Gaussians. If we manage to separate sources of variability, an efficient estimation requires to estimate the parameters of $\sum_{i=1}^{N} n_i$ Gaussians. If we confound sources of variability, however, estimation of the parameters of $\prod_{i=1}^{N} n_i$ Gaussians is necessary. In the latter case, while performance improvements will be logarithmic in the amount of data, memory requirements will go up linearly. Eventually, there is no clear sense of what types of variabilities are modeled by which Gaussian parameters.

The idea is hence to introduce feature transformations to model the illumination and to enforce consistency between feature transformations at adjacent positions, in the same manner we enforced consistency between grid transformations, to constrain the illumination variation.

6.3 The HMM-based Transformation Model

Our states which represent both local grid and feature transformations are now doubly indexed: $q_{i,j} = (\tau_{i,j}, \phi_{i,j})$. $\tau_{i,j}$ and $\phi_{i,j}$ are respectively the grid and feature transformation parts of the state. In the remainder of this section, we consider the emission and transition probabilities of the HMM-based transformation model and briefly discuss the issue of the initial occupancy probability.

6.3.1 Emission probability

If $q_{i,j} = (\tau, \phi)$, the emission probability $b_{i,j}^{\tau,\phi}$ is modeled with a mixture of Gaussians, as was previously the case for grid transformations only:

$$b_{i,j}^{\tau,\phi} = \sum_{k=1}^{K_{i,j}} w_{i,j}^{k} b_{i,j}^{\tau,\phi,k} \tag{6.4}$$

where $b_{i,j}^{\tau,\phi,k}$'s are D-variate Gaussians with means $\mu_{i,j}^{\tau,\phi,k}$ and covariance matrices $\Sigma_{i,j}^{k}$. If the "feature" state ϕ also denotes the additive contribution of the illumination in the feature domain, the Gaussian means are of the form:

$$\mu_{i,j}^{\tau,\phi,k} = \mu_{i,j}^{\tau,k} + \phi = m_{i,j}^{\tau} + \delta_{i,j}^{k} + \phi \tag{6.5}$$

In the previous chapter, we separated the variability into inter-class variability and intra-class variability. In this chapter, we go one step further by separating the intra-class variability into grid transformation variability caused by facial expressions variations, and feature transformation variability, caused by illumination variations.

6.3.2 Transition probability

If we assume that grid and feature transformations model respectively differences in facial expression and illumination, and that facial expression and illumination *variations* are mostly independent (i.e. a facial expression or a pose change between two adjacent positions has a limited impact on the illumination change between the same positions and vice versa), then the horizontal and vertical transition probabilities can be separated as follows:

$$a_{i,j}^{\mathcal{H}} = P(q_{i,j+1}|q_{i,j}) \quad = \quad P(\tau_{i,j+1}|\tau_{i,j}) \times P(\phi_{i,j+1}|\phi_{i,j}) \tag{6.6}$$

$$a_{i,j}^{\mathcal{V}} = P(q_{i+1,j}|q_{i,j}) \quad = \quad P(\tau_{i+1,j}|\tau_{i,j}) \times P(\phi_{i+1,j}|\phi_{i,j}) \tag{6.7}$$

From now on, we will denote $a_{i,j}^{\tau,\mathcal{H}}$ (resp. $a_{i,j}^{\tau,\mathcal{V}}$) the part of the horizontal (resp. vertical) transition probability which corresponds to the grid state and $a_{i,j}^{\phi,\mathcal{H}}$ (resp. $a_{i,j}^{\phi,\mathcal{V}}$) the part which corresponds to feature state.

While the choice of a discrete number of grid transformations is natural due to the discrete nature of the feature extraction grid of the template image, it is easier to deal with the illumination with an *infinite continuous* set of illumination states. We choose the horizontal and vertical illumination components of the transition probabilities to be D-variate Gaussians:

$$a_{i,j}^{\phi,\mathcal{H}}(\phi, \phi + \delta\phi) = a_{i,j}^{\phi,\mathcal{H}}(\delta\phi) \quad = \quad \frac{1}{(2\pi)^{\frac{D}{2}} |S_{i,j}^{\mathcal{H}}|^{\frac{1}{2}}} \exp\left\{ -\frac{1}{2} \delta\phi^T S_{i,j}^{\mathcal{H}\,(-1)} \delta\phi \right\} \tag{6.8}$$

$$a_{i,j}^{\phi,\mathcal{V}}(\phi, \phi + \delta\phi) = a_{i,j}^{\phi,\mathcal{V}}(\delta\phi) \quad = \quad \frac{1}{(2\pi)^{\frac{D}{2}} |S_{i,j}^{\mathcal{V}}|^{\frac{1}{2}}} \exp\left\{ -\frac{1}{2} \delta\phi^T S_{i,j}^{\mathcal{V}\,(-1)} \delta\phi \right\} \tag{6.9}$$

Obviously, $a_{i,j}^{\phi,\mathcal{H}}(\delta\phi)$ and $a_{i,j}^{\phi,\mathcal{V}}(\delta\phi)$ are maximum when $\delta\phi = 0$, i.e. when there is no illumination variation. The choice of such a form of transition probability is primarily motivated by its computational tractability. To reduce even more the complexity, in the following we assume that the covariance matrices $S_{i,j}^{\mathcal{H}}$ and $S_{i,j}^{\mathcal{V}}$ are diagonal and therefore, that the components of the feature vectors are independent from each other. $S_{i,j}^{\mathcal{H}}$'s and $S_{i,j}^{\mathcal{V}}$'s are the only FIT parameters of our model of illumination variation. Intuitively, they model the speed of the horizontal and vertical variations of the illumination in each feature component at position (i, j).

We do not make use of initial occupancy probabilities to ensure invariance to a global shift of the energy in each frequency band.

6.4 Recognition with the Transformation Model

We denote $Q = (T, \Phi)$ a sequence of states where T is a sequence of grid states: $T = \{\tau_{i,j}, i = 1, ...I, j = 1, ..., J\}$ and Φ is a sequence of feature states: $\Phi = \{\phi_{i,j}, i = 1, ..., I, j = 1, ..., J\}$. In the case where we attempt to model facial expressions and illumination variations, our similarity measure between face images is:

$$P(O|\Phi^*, \lambda_t, \lambda_{\mathcal{R}}) \tag{6.10}$$

where Φ^* is the sequence of feature states that best explains the illumination variation, i.e.:

$$\Phi^* = \arg\max_{\Phi} P(\Phi^*|O, \lambda_t, \lambda_{\mathcal{R}}) \tag{6.11}$$

In the case where the system can be in only one grid state at each position (rigid matching) and where emission probabilities are Gaussian, we can make direct use of the modified forward-backward as applied to the T-SSM to find the best sequence of states Φ^* (c.f. section 4.4.3). During the modified forward-backward, we can estimate $\gamma_{i,j}(\phi) = P(\phi_{i,j} = \phi|O, \lambda_t, \lambda_{\mathcal{R}})$ and then choose the sequence of locally optimal states:

$$\phi^*_{i,j} = \arg\max_{\phi} \gamma_{i,j}(\phi) \tag{6.12}$$

As discussed in section 4.5, although choosing the sequence of locally optimal states may not lead to the sequence of globally optimal states, this approximation is valid in the case where the best sequence of states accounts for most of the total probability.

In the case where we perform an elastic matching and where emission probabilities are mixtures of Gaussians, a direct application of the modified forward-backward would be exponential in the size of the data. Instead we propose to apply iterative passes to find successively the grid states, Gaussian indexes and feature states that best explain the transformation between two images. Let $k_{i,j}$ be the Gaussian index in the emission probability at position (i, j) and let K be the sequence of Gaussian indexes: $K = \{k_{i,j}, i = 1, ..., I, j = 1, ..., J\}$. Let us also denote respectively $T^{(n)}$, $K^{(n)}$ and $\Phi^{(n)}$ the best set of grid states, Gaussian indexes and illumination states after the n-th iteration. The iterative procedure is described in Table 6.1.

Although this iterative procedure is not guaranteed to lead to the optimal solution or even to converge, it does provide acceptable results. Once Φ^* is obtained, the computation of the score $P(O|\Phi^*, \lambda_t, \lambda_{\mathcal{R}})$ is done with the modified forward-backward for the T-HMM where we replace the features $o_{i,j}$ with their illumination compensated version $(o_{i,j} - \phi^*_{i,j})$.

$\boxed{1}$	Initialize Φ_0: $\forall(i,j)$, $\phi_{i,j} = 0$, i.e., we assume that there is no illumination variation between I_q and I_t.
$\boxed{2}$	$T^{(n)} = \arg\max_T \log P(T \vert O, \Phi^{(n-1)}\lambda_t, \lambda_{\mathcal{R}})$: during the forward-backward, one estimates the occupancy probabilities $\gamma_{i,j}(\tau)$ and chooses at each position (i,j) the state $\tau^*_{i,j}$ such that: $\tau^*_{i,j} = \arg\max_\tau \gamma_{i,j}(\tau)$.
$\boxed{3}$	$K^{(n)} = \arg\max_K \log P(K \vert O, \Phi^{(n-1)}, T, \lambda_t, \lambda_{\mathcal{R}})$: during the forward-backward, one can also estimate $\gamma_{i,j}(\tau, k)$. The optimal Gaussian index $k^*_{i,j}$ is chosen such that $k^*_{i,j} = \arg\max_k \gamma_{i,j}(\tau^*_{i,j}, k)$.
$\boxed{4}$	$\Phi^{(n)} = \arg\max_\Phi \log P(\Phi \vert O, T^{(n)}, K^{(n)}, \lambda_t, \lambda_{\mathcal{R}})$: we apply the modified forward-backward for the T-SSM, estimate $\gamma_{i,j}(\phi)$ and choose at each position (i,j) the state $\phi^*_{i,j}$ such that $\phi^*_{i,j} = \arg\max_\phi \gamma_{i,j}(\phi)$.
$\boxed{5}$	Iterate: Go back to step 2 until $T^{(n)}$, $K^{(n)}$ and $\Phi^{(n)}$ converge.

Table 6.1: Iterative procedure to find the set of grid states T^*, Gaussian indexes K^* and feature states Φ^* which best explain the observed variability between two face images.

6.5 Training the Transformation Model

In this section,we focus on the estimation of the feature transformation parameters, i.e. the $S^{\mathcal{H}}_{i,j}$'s and $S^{\mathcal{V}}_{i,j}$'s. We train these parameters with a set of pairs of template and query images in a ML fashion using the Baum-Welch algorithm as described in 5.5.1. In the following, we explicate the E-step and M-step of the Baum-Welch algorithm and then we explicate how to train these parameters operationally.

6.5.1 E-step

During the training, for each pair of images (I^p_t, I^p_q), we perform the algorithm described in the previous section to estimate Φ^*. During step 4, we compute the following quantities: $\mu^{\gamma\mathcal{H}}_{i,j}$, $\mu^{\gamma\mathcal{V}}_{i,j}$, $\sigma^{\gamma\mathcal{H}^2}_{i,j}$, $\sigma^{\gamma\mathcal{V}^2}_{i,j}$, $\sigma^{\alpha\mathcal{H}^2}_{i,j}$, $\sigma^{\alpha\mathcal{V}^2}_{i,j}$.

6.5.2 M-step

As $S^{\mathcal{H}}_{i,j}$'s and $S^{\mathcal{V}}_{i,j}$'s are diagonal, with $S^{\mathcal{H}}_{i,j} = \text{diag}\{s^{\mathcal{H}}_{i,j}[1]^2, ..., s^{\mathcal{H}}_{i,j}[D]^2\}$ and $S^{\mathcal{V}}_{i,j} = \text{diag}\{s^{\mathcal{V}}_{i,j}[1]^2, ..., s^{\mathcal{V}}_{i,j}[D]^2\}$, the parameters $s^{\mathcal{H}}_{i,j}[d]^2$ and $s^{\mathcal{V}}_{i,j}[d]^2$ can be optimized separately in each dimension. In the following, to simplify notations, we discard the dimension index $[d]$. We also assume that we train these parameters with one pair of images as was the case in the previous chapter. The extension to multiple pairs of images is straightforward as it simply consists in accumulating the statistics. If we denote $\xi^{\mathcal{H}}_{i,j}(\phi, \phi + \delta\phi) = P^{\mathcal{H}}(\phi_{i,j} = \phi, \phi_{i,j+1} = \phi + \delta\phi | O, \lambda_t, \lambda_{\mathcal{R}})$, the part of $\mathcal{Q}^{\mathcal{H}}$ (the horizontal Baum auxiliary function) which contains the parameter ${s^{\mathcal{H}}_{i,j}}^2$ is:

$$\begin{aligned} & \int_{\phi,\delta\phi} \xi^{\mathcal{H}}_{i,j}(\phi, \phi + \delta\phi) \log a^{\phi,\mathcal{H}}_{i,j} d\phi d\delta\phi \\ = & -\frac{1}{2} \int_{\phi,\delta\phi} \xi^{\mathcal{H}}_{i,j}(\phi, \phi + \delta\phi) \left(\log \left({s^{\mathcal{H}}_{i,j}}^2 \right) + \frac{\delta\phi^2}{{s^{\mathcal{H}}_{i,j}}^2} \right) d\phi d\delta\phi \end{aligned} \tag{6.13}$$

If we take the partial derivative with respect to ${s^{\mathcal{H}}_{i,j}}^2$, we get:

$$-\frac{1}{2} \int_{\phi,\delta\phi} \xi^{\mathcal{H}}_{i,j}(\phi, \phi + \delta\phi) \left(\frac{1}{{s^{\mathcal{H}}_{i,j}}^2} - \frac{\delta\phi^2}{{s^{\mathcal{H}}_{i,j}}^4} \right) d\phi d\delta\phi \tag{6.14}$$

and, if we equate it to zero, we finally obtain the following estimate $\hat{s}^{\mathcal{H}}_{i,j}{}^2$ of ${s^{\mathcal{H}}_{i,j}}^2$:

$$\hat{s}^{\mathcal{H}}_{i,j}{}^2 = \int_{\phi,\delta\phi} \xi^{\mathcal{H}}_{i,j}(\phi, \phi + \delta\phi) \delta\phi^2 d\phi d\delta\phi \tag{6.15}$$

Similarly, if we define $\xi^{\mathcal{V}}_{i,j}(\phi, \phi + \delta\phi) = P^{\mathcal{V}}(\phi_{i,j} = \phi, \phi_{i+1,j} = \phi + \delta\phi | O, \lambda_t, \lambda_{\mathcal{R}})$, the optimal estimate $\hat{s}^{\mathcal{V}}_{i,j}{}^2$ of ${s^{\mathcal{V}}_{i,j}}^2$ is:

$$\hat{s}^{\mathcal{V}}_{i,j}{}^2 = \int_{\phi,\delta\phi} \xi^{\mathcal{V}}_{i,j}(\phi, \phi + \delta\phi) \delta\phi^2 d\phi d\delta\phi \tag{6.16}$$

$\xi^{\mathcal{H}}_{i,j}$ and $\xi^{\mathcal{V}}_{i,j}$ are given, in the discrete case, by equations 5.18 and 5.19. In the continuous case, we just have to replace in the previous equations τ with ϕ, $\delta\tau$ with $\delta\phi$ and the sum with the integral. Introducing the notations $\rho^{\alpha\mathcal{H}}_{i,j} = s^2/(s^2 + \sigma^{\alpha\mathcal{H}^2}_{i,j})$ and $\rho^{\alpha\mathcal{V}}_{i,j} = s^2/(s^2 + \sigma^{\alpha\mathcal{V}^2}_{i,j})$, we obtain for $\hat{s}^{\mathcal{H}}_{i,j}{}^2$ and $\hat{s}^{\mathcal{V}}_{i,j}{}^2$ the following closed form solutions [2]:

$$\begin{aligned} \hat{s}^{\mathcal{H}}_{i,j}{}^2 &= (\mu^{\gamma\mathcal{H}}_{i,j+1} - \mu^{\gamma\mathcal{H}}_{i,j})^2 + \rho^{\alpha\mathcal{H}}_{i,j} \sigma^{\alpha\mathcal{H}^2}_{i,j} + \rho^{\alpha\mathcal{H}^2}_{i,j} {\sigma^{\gamma\mathcal{H}}_{i,j+1}}^2 & (6.17) \\ \hat{s}^{\mathcal{V}}_{i,j}{}^2 &= (\mu^{\gamma\mathcal{V}}_{i+1,j} - \mu^{\gamma\mathcal{H}}_{i,j})^2 + \rho^{\alpha\mathcal{V}}_{i,j} \sigma^{\alpha\mathcal{V}^2}_{i,j} + \rho^{\alpha\mathcal{V}^2}_{i,j} {\sigma^{\gamma\mathcal{V}}_{i+1,j}}^2 & (6.18) \end{aligned}$$

The first parts of the previous formulas correspond to the horizontal and vertical estimates of the best path Φ^* respectively. The additional terms are due to the fact that we do not only consider the best path Φ^* but that we integrate over all paths in the modified forward-backward algorithm.

[2] The following formulas were derived using the software Maple.

6.5.3 The training operationally

The training of our HMM-based transformation model with discrete grid transformations and continuous feature transformations is carried out in two steps.

During the first stage, we train solely the parameters which correspond to the grid transformations. To train these parameters, we only make use of pairs of images (I_t^p, I_q^p) which do not exhibit any illumination variation. Thus, the training is carried out exactly as described in the previous chapter (c.f. section 5.5).

In the second stage, starting from this model, we train the covariance matrices $S_{i,j}^{\mathcal{H}}$ and $S_{i,j}^{\mathcal{V}}$ which are the only parameters of the illumination transformation model. This training is performed using a set of pairs of images that exhibit illumination variation. The assumption is that, as the transformation model trained on the data without illumination variations already accounts for variations due to facial expressions and, in a lesser extent, pose, all the variability that remains unexplained is purely due to illumination. The diagonal elements of $S_{i,j}^{\mathcal{H}}$'s and $S_{i,j}^{\mathcal{V}}$'s are initialized to values close to 0, i.e. we first assume that there is little if no illumination variation. Then the diagonal elements of $S_{i,j}^{\mathcal{H}}$'s and $S_{i,j}^{\mathcal{V}}$'s are re-estimated using the Baum-Welch algorithm. Note that the second stage of the training is similar to the SAT training used in ASR [AMSM96].

6.6 Experimental Validation

In this section, we carry out two sets of experiments. In the first one, we evaluate the influence of the log transform in the pixel domain on the identification rate of BIC and PMLGT. In the second set of experiments, we evaluate the performance of the proposed approach which will be later referred to as PMLGFT for probabilistic mapping with local grid and feature transformations

6.6.1 Evaluation of the logarithm transform

In this section, we evaluate the impact of the log transform on the recognition rate for both BIC and PMLGT, in the case where no illumination variation is observed during the training. While we expect this pre-processing to improve the performance of both algorithms for illumination variations, we wanted also to see its impact on the other types of variabilities. Therefore, we repeated exactly the same facial expression, illumination, pose and occlusion experiments as the ones we carried out in the previous chapter 5.7.5. The only difference was in the use of the log transform on images as a pre-processing step. Both BIC and PMLGT were trained as described in sections 5.7.3 and 5.7.4 respectively (same data, same parameters).

Facial expressions

Facial expression results are presented on Figure 6.3 which can be compared to Figure 5.12 page 101.

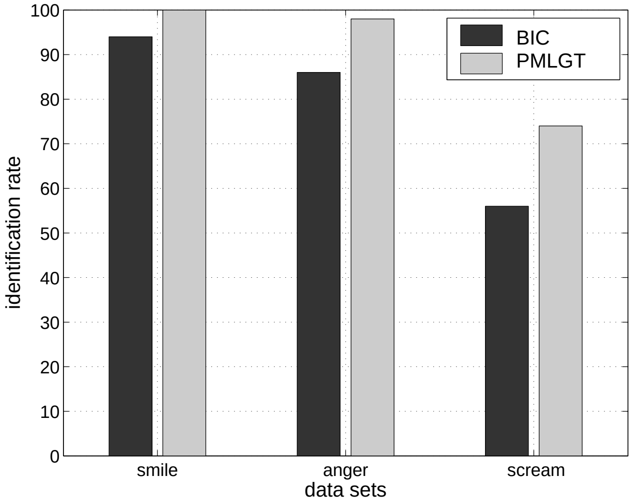


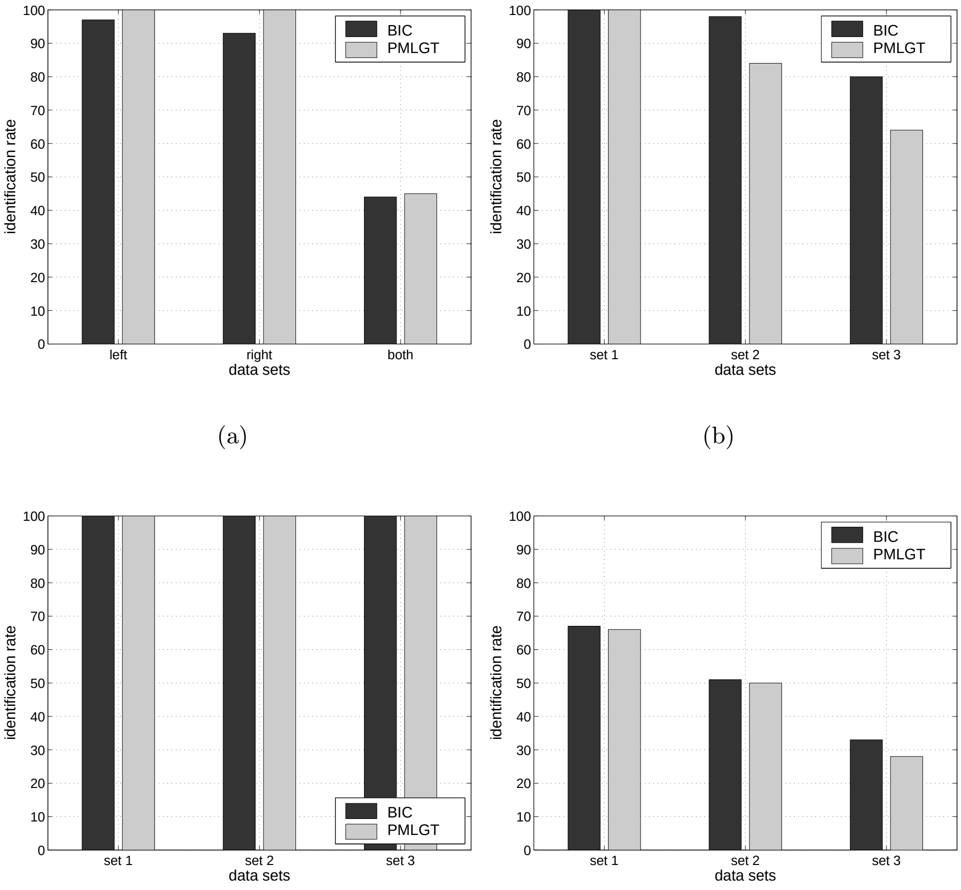
Figure 6.3: Facial expression results on the AR database.

For both BIC and PMLGT the impact of the log transform is very limited: for BIC the identification rate is 78% without the log transform and 79% with the log transform, and for PMLGT it is 89% without the log transform and 90% with the log transform. On the average, if we perform a McNemar's test, the small observed increase of performance cannot be declared significant. However, PMLGT can still be declared to outperform BIC with more than 99% confidence.

In the previous chapter, we assessed the influence of grid transformations on the recognition rate for variations in facial expressions and observed that they had no significant impact on the performance (c.f. section 5.7.5). We repeated this analysis in the case where we apply the log transform. The performance of the rigid version of PMLGT is 84% on the average. If we perform a McNemar's test, we can make sure that the difference between PMLGT and the rigid matcher is significant with more than 99% confidence. Thus, if we first apply a log transform in the pixel domain, grid transformations have a very positive impact on the performance for facial expressions.

Illumination

Illumination results are presented on Figure 6.3 which can be compared to Figure 5.13 page 103. The log transform has a positive impact on both BIC and PMLGT for almost all data sets.



(a) (b)

(c) (d)

Figure 6.4: Illumination results on (a) AR, (b) Yale B (c) PIE 1 and (d) PIE 2.

On the AR face database there is only a small increase of performance on datasets 05 and 06, i.e. when faces are illuminated by one light on the left or on the right, but a significant decrease for set 07, i.e. when faces are illuminated by both lights, especially for PMLGT. As discussed in section 5.7.5, the images of set 07 of AR seem to have a very low contrast. As the log transform reduces even more the contrast of an image (c.f. Figures 6.1 and 6.2), it will have a negative impact on the identification rate.

On Yale B, the log transform has a very significant impact for both BIC and PMLGT. On set 2, the performance increases from 67% to 98% for BIC and from 79% to 84% for PMLGT. On the particularly challenging set 3, the performance increases from 24% to 80% for BIC and from 35% to 64% for PMLGT. Thus, while PMLGT outperformed BIC with more than 99% confidence without the log transform, if we perform a McNemar's test, BIC can now be said to outperform PMLGT with more than 99% confidence.

The log transform has a spectacular impact on the performance of PMLGT on PIE 1. Indeed, while the average identification rate was respectively 98% and 46% without the log transform for BIC and PMLGT, it is now a perfect 100% for both algorithms.

Finally, the log transform has a significant impact (with more than 99% confidence) on the performance of BIC and especially on the performance of PMLGT on PIE 2. Now, BIC cannot be declared to outperform PMLGT. Note that the performance of both algorithms is still fairly low. On the very challenging set 3, the identification rate of BIC and PMLGT is on the order of 30%.

Pose

Pose results are presented on Figure 6.5 and can be compared to Figure 5.14 page 105. While the log transform has no significant impact on the performance of the PMLGT (1% decrease of the average identification rate), it does have, surprisingly, a significant impact on the performance of BIC. Indeed, the identification rate increases from 46% to 56% on the average. However, it is difficult to explain this difference.

Occlusion

Occlusion results are presented on Figure 6.6 which can be compared to Figure 5.15 page 106. As expected, there is no significant difference in performance for both BIC and PMLGT.

6.6.2 Evaluation of the proposed approach

In this section, we consider the addition of pairs of images with illumination variations to the training data to improve the performance of the classifiers. These images are extracted form the FAFC set of FERET [PMRR00] which contains 200 persons (c.f. also appendix D). More precisely, we make use of those frontal images

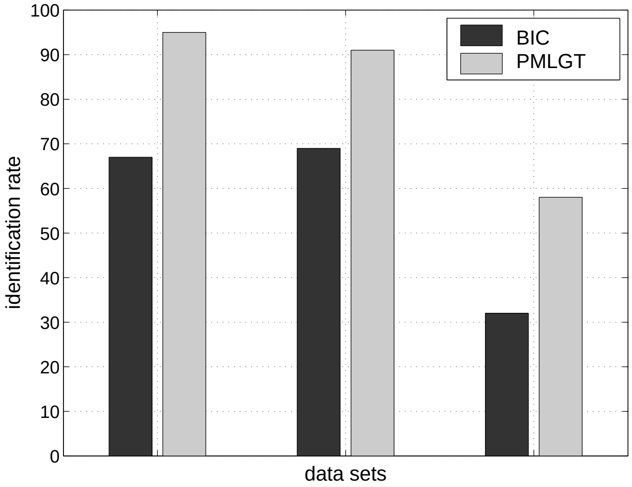


Figure 6.5: Pose results on the PIE database.

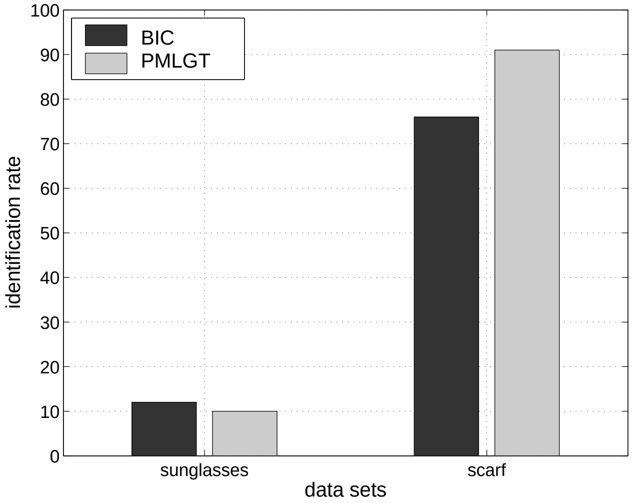


Figure 6.6: Occlusion results on the AR database.

which are referred to as BA, BJ and BK. BA and BJ images correspond to different facial expressions in the same illumination condition and BK images correspond to a different illumination condition. Thus the additional training material amounts to 600 images and 400 pairs of images as we consider (BA,BK) and (BJ,BK) pairs.

We evaluate the performance of the proposed approach that will be later referred to as PMLGFT for probabilistic mapping with local grid and feature transformations. We outline that for PMLGFT, it is necessary to apply the log transform in the pixel domain before the feature extraction (c.f. section 6.2). To train the HMM-based transformation model, we use the two-stage strategy described in section 6.5.3. We start from the model trained in the previous section on FAFB images. Then we train the $S^{\mathcal{H}}_{i,j}$ and $S^{\mathcal{V}}_{i,j}$ covariance matrices with the Baum-Welch algorithm on the FAFC data. Note that we have tried to use different horizontal and vertical covariance matrices at different positions. The rational is that the illumination may vary differently at different positions on the face. For instance, around the nose sharp variations are likely to happen due to self shadowing. However using multiple $S^{\mathcal{H}}_{i,j}$ and $S^{\mathcal{V}}_{i,j}$ did not improve the performance. This may be either due to our rather simple model of illumination variation or to the relatively modest amount of training data. Therefore, we assume in the following that $S^{\mathcal{H}}_{i,j} = S^{\mathcal{V}}_{i,j} = S$, $\forall(i,j)$. The parameters we used to estimate Φ^*, as described in Table 6.1, are the following ones. To estimate the best grid transformations during step 2, we performed one horizontal and one vertical pass. To estimate the best feature transformations during step 4, we performed 5 horizontal and 5 vertical passes. Steps 2 to 4 of the algorithm were repeated 3 times to reach a reasonable convergence. To train S, we carried out 3 Baum-Welch iterations. Note that we used the same parameters at recognition time. This incurs a very significant increase of the computational cost. Indeed, running our non-optimized code on a 2 GHz Pentium 4 with 1 GB Ram, it takes on the order of 25 ms to compare two face images with an HMM that contains 16 Gpm with both grid and feature transformations.

We compare PMLGFT to BIC and PMLGT. BIC and PMLGT are trained exactly as described in the previous chapter (c.f. section 5.7.4). The only differences are that we apply the log transform in the pixel domain, as is the case in the previous section, and that we add the FAFC images to the FAFB data. Thus, we model the illumination in the straightforward manner described in section 6.2. We believe that such an approach is suboptimal and that PMLGFT should outperform PMLGT.

Our goal is to evaluate the performance of the BIC, PMLGT and PMLGFT classifiers in different illumination conditions, but also on other types of variabilities. Therefore, we repeated exactly the same experiments as the ones we carried out in the previous chapter and in the previous section. Moreover, we carried out one additional set of experiments to show the impact of both pose and illuminations variations.

Facial expressions

Facial expression results are presented on Figure 6.7 which can be compared to Figure 6.3. There is no significant difference in performance, neither for BIC, nor for PMLGT and PMLGFT, which could be expected as there is no illumination variation to compensate for. Note that this is a very important for PMLGFT. Indeed, PMLGFT makes use of two types of transformations, grid and feature transformations, which "compete" to explain the observed variability. Thus, the performance of PMLGFT could have been lower than the performance of PMLGT when there is no illumination variation. This shows that, even if no illumination variation is observed, the PMLGFT does not try to interpret facial expression variations as illumination variations.

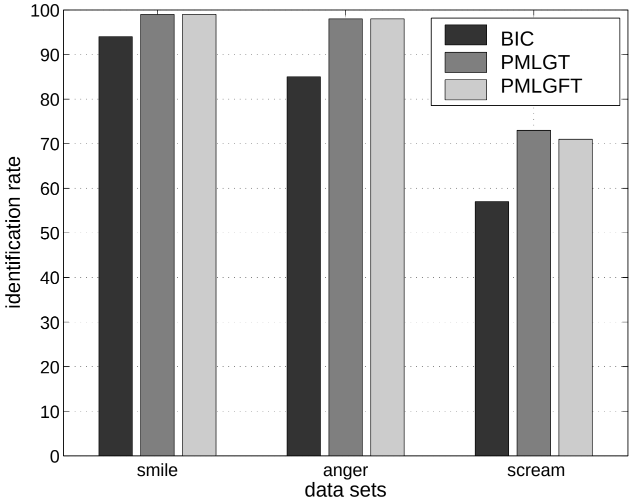
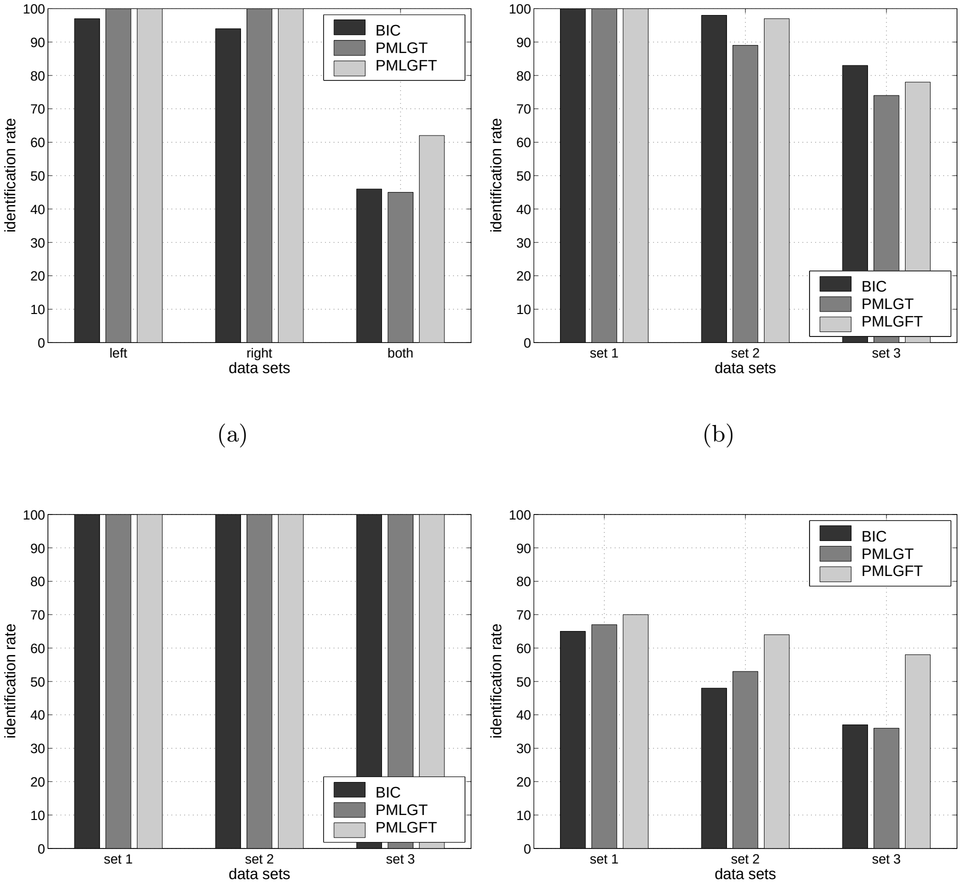


Figure 6.7: Facial expression results on the AR database.

Illumination

Illumination results are presented on Figure 6.8 which can be compared to Figure 6.4. Adding the illumination variation in the training data has no significant impact on BIC and has an impact on PMLGT only on Yale B. On the other hand, it does have a significant impact on PMLGFT on AR, Yale B and PIE 2. If we perform McNemar's test of significance, we can now say that PMLGFT outperforms BIC on AR and PIE 2 and that no algorithm can be declared to outperform the other one on Yale B and PIE 1.



(a)

(b)

(c)

(d)

Figure 6.8: Illumination results on (a) AR, (b) Yale B (c) PIE 1 and (d) PIE2.

The average identification rate of PMLGT without the log transform, of PMLGT with the log transform and PMLGFT (necessarily with the log transform) are summarized in Table 6.6.2.

Pose

Pose results are presented on Figure 6.9 which can be compared to Figure 6.5. No significant difference can be observed, neither for BIC, nor for PMLGT or PMLGFT. This is not surprising as there is no illumination variation in the considered images. It demonstrates that PMLGFT does not attempt to interpret the pose variability,

	PMLGT (no log)	PMLGT (log)	PMLGFT (log)
AR	86%	82%	87%
Yale B	69%	87%	91%
PIE 1	46%	100%	100%
PIE 2	16%	54%	65%

Table 6.2: Illumination variation results: average identification rate of PMLGT without the log transform, PMLGT with the log transform and PMLGFT (necessarily with the log transform) on AR, Yale B, PIE 1 and PIE 2.

even if it was not observed during the training, as an illumination variability.

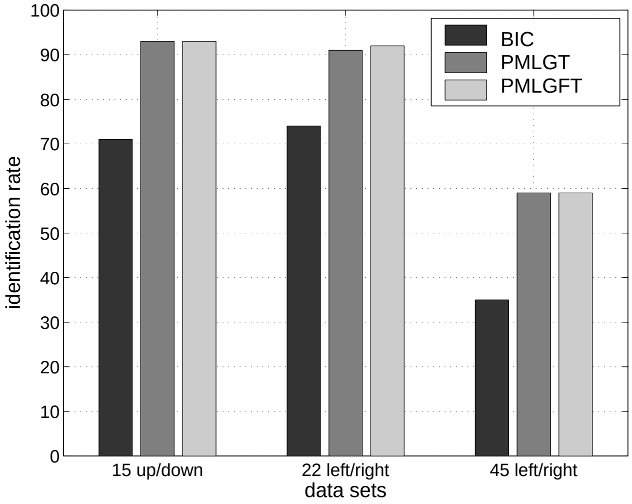


Figure 6.9: Pose results on the PIE database.

Occlusion

Occlusion results are presented on Figure 6.10 which can be compared to Figure 6.6. There is no significant difference, neither for BIC nor for PMLGT on both data sets. However, for PMLGFT, there is a very significant increase of the identification rate on data set 08, which corresponds to an occlusion of the upper part of the face with sunglasses. We currently have no explanation for this increased performance.

Pose and illumination

Finally, we evaluate the ability of the three classifiers to deal with pose and illumination variations. Experiments were carried out on the Yale B database (c.f.

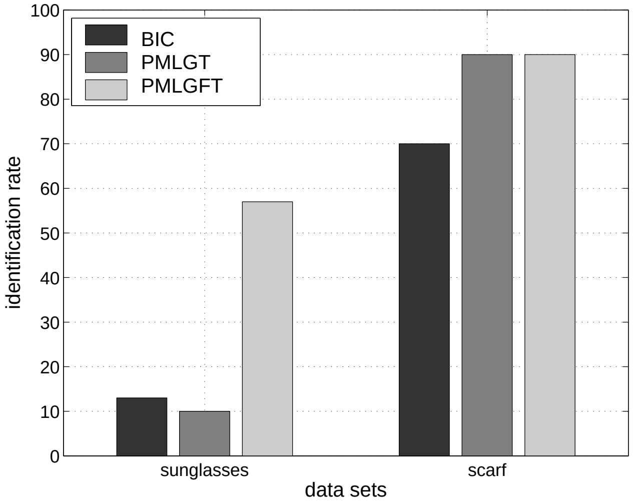


Figure 6.10: Occlusion results on the AR database.

appendix D). We used the data from the 9 cameras for this set of experiments and images were divided into three sets according to the angle θ between the flash and the optical axis of the frontal camera: $20^o \leq \theta \leq 25^o$ for set 1, $35^o \leq \theta \leq 50^o$ for set 2 and $60^o \leq \theta \leq 77^o$ for set 3. Results are presented on Figure 6.11.

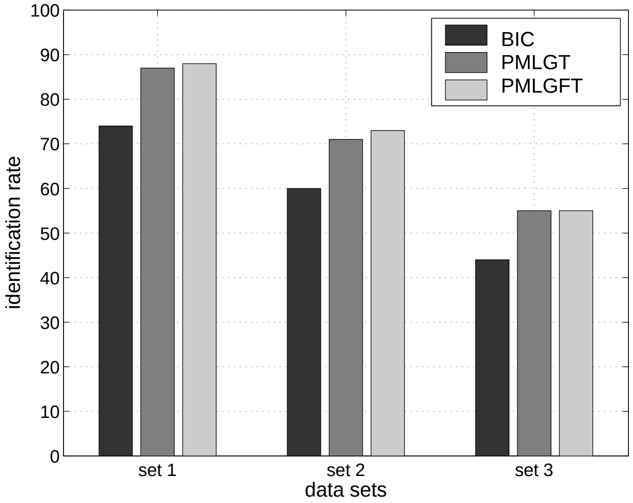


Figure 6.11: Pose and illumination results on the Yale B database.

The average identification rates over the three data sets are 59% for BIC, 70% for PMLGT and 71% for PMLGFT. While both PMLGT and PMLGFT outperform BIC with more than 99% confidence, PMLGFT seems to loose its advantage over

PMLGT. Indeed, a McNemar's test proves that PMLGFT can only be considered to outperform PMLGT with approximately 90% confidence. Considering that Yale B contains only 10 persons, this shows how difficult it is to cope with multiple variabilities simultaneously.

6.7 Conclusion

In this chapter, we enriched the transformation model derived in the previous chapter by allowing feature transformations with the goal to compensate for illumination variations. We first showed how to transform the illumination variation into an additive variability in the feature domain, simply by applying a logarithm transform in the pixel domain. We then explicated the emission and transition probabilities of the HMM-based transformation model with both grid and feature transformations. We explained how to perform recognition with the considered model and how to train its parameters.

We first evaluated the influence of the log transform in the pixel domain and showed that it had a very significant impact on the performance of both BIC and PMLT for illumination variations. We also evaluated our novel approach to illumination compensation and showed that an additional increase of the performance could be obtained for extreme illumination conditions such as a flash without ambient lighting. If we compare the results obtained in this chapter and in the previous one, the performance of the proposed approach is dramatically improved for illumination variations and is almost unchanged for other types of variabilities. Note however that when there are multiple sources of variabilities (e.g. pose and illumination), the performance of the system degrades very significantly.

We would like to outline that very recent work in the ASR community for the problem of noise estimation and compensation bears a lot of similarity with our approach to illumination compensation [SR03, DA04]. Indeed, in [SR03] the noise is modeled as a sequence of states of a dynamical system with a continuum of states. Observations generated by such a system are assumed to be related to the state of the system by a functional relation which models clean speech as the corrupting influence of noise. In our case, we assume that variations due to facial expressions corrupt the illumination signal. The work of [DA04] brings important differences, one of which is to perform a joint noise and speech tracking. This is fairly similar with the joint grid and feature transformations estimation used in our approach.

An advantage of ASR over AFR is that the statistics of the noise model can be estimated either during the first or last few frames. We believe that one limitation of our current approach is the fact that the covariance matrix S in our illumination transformation model is fixed for all pairs of images. Indeed, we think that S should incorporate both some a priori knowledge learned off-line through a training phase, as is currently the case, but also some information which is dependent on the pairs of images that need to be compared.

Finally, we would like to point out that, while our model of illumination compensation has been introduced in the context of AFR, it could benefit to other research areas. As our original approachhas a lot in common with motion estimation algorithms, and especially MAP estimation of dense motion [Bov00], we think that our approach could be applied to the difficult problem of motion estimation in the presence of illumination variations.

As it was shown that the log transform in the pixel domain could cop with most of the illumination variations and that feature transformations had an impact mainly on extreme illumination variations, in the following chapters, we will consider only grid transformations.

7

Face Image Retrieval in Large Databases

7.1 Introduction

As underlined in the introductory chapter of this dissertation, defining a meaningful distance between images for the problem of AFR is a very challenging issue due to the long list of variabilities that can affect face images of the same person. A complex measure of distance is therefore required to accommodate for all possible variabilities. Although a more elaborate distance may improve the performance, it will also generally lead to an increase in the complexity. For instance, while the addition in chapter 6 of feature transformations to our probabilistic model of image mapping had a positive impact on the identification rate in the case of a strong illumination variation, it also incurred a very significant increase of the computational cost as the time required to compare two face images was multiplied by a factor of 5. Hence, it is difficult to design a measure which is both *accurate* and *computationally efficient*.

However, both properties are required to tackle the very challenging task of automatic retrieval of face images in large databases that can potentially contain from a few thousands to several millions of images. Mainly, two approaches have been suggested to address this problem.

The first approach makes use of two (or even more) measures of distance and cascades them. Generally, the first distance has a low accuracy but requires little computation while the second one has a high accuracy but requires significantly more computation. The first algorithm is run on the whole set of data and the N-best candidates are retained. The second algorithm is then run on this subset of images. Such an approach has already been used for multimodal biometrics person authentication for instance [HJ98].

The second approach is to perform a partitioning of the image space through a clustering of the data. When a new target image is added to the database, one computes the distance between this image and all clusters and the image is associated to its nearest cluster. When a query image has to be classified, the first step consists in determining the nearest cluster and the second step involves the computation of the distance between the test image and the subset of images of the considered cluster. Note that both the target and query images can be assigned to more than one cluster. Indeed, if face images of a given person are close to the "boundary" between two or more clusters, different images of the same person may be assigned to different clusters in the case where large variabilities are not fully handled by the distance measure, as depicted on Figure 7.1. To solve this problem, target and query images can be assigned to the K nearest clusters or to all the clusters whose distance falls below a predefined threshold. Obviously, the incurred increase in accuracy is obtained at the expense of a higher computational cost.

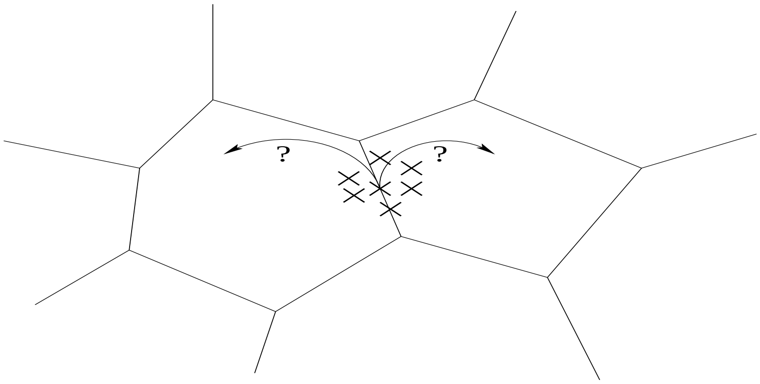


Figure 7.1: Uncertainty in cluster assignment.

Many clustering algorithms, especially those based on a probabilistic framework, can be directly interpreted as an application of the EM algorithm [DLR77]. During the E-step, the distance between each observation and each cluster centroid is computed and each observation is assigned to its nearest cluster (or probabilistically to all clusters). During the M-step, the cluster centroid is updated using the assigned observations. The update step also depends on the chosen distance since the centroid is defined as the point which minimizes the average distance between the assigned observations and the centroid.

Until now our work has focused on the issue of distance computation and the "missing" stage to be able to perform clustering with the proposed distance is the update step. When using simple metrics the update step is greatly simplified. For instance, for the Euclidean distance, the update step is a simple averaging of the assigned observations. In the case of complex distances, such as the distance induced by the PMLT, computing the centroid is much more challenging.

The remainder of this chapter is organized as follows. In the next three sections, we will focus on the robust estimation of cluster centroids. In section 7.2.1, we explicate the update step for our non-trivial distance. In section 7.3, we discuss the issue of cluster centroid initialization and propose a procedure for initializing centroids which is tailored to the problem of interest. In section 7.4, we address the problem of data scarcity to estimate cluster centroids. Finally, in section 7.5, we provide an alternative to the multiple cluster assignment paradigm and show that instead of assigning and image to multiple clusters, it is more efficient to assign it to all clusters probabilistically.

7.2 EM-based clustering

In this section, we first present the theory of EM-based clustering and apply it to our problem. A special emphasis is put on the update step. We then present our first experimental results.

7.2.1 Theory

Our goal is, given a set of N images $\{I_1, ..., I_N\}$, to estimate C clusters $\{\mathcal{C}_1, ..., \mathcal{C}_C\}$. The measure of distance between an image I_n and a cluster is the probability that this image was generated by the cluster centroid knowing $\mathcal{R}$, the model of relationship between images of the same person. Therefore *images $\{I_1, ..., I_N\}$ are naturally treated as query images* and *cluster centroids as templates*. In the following, we will denote by O_n the "query representation" of I_n, i.e. the set of feature vectors extracted on a sparse grid from I_n. O will denote the set of all observations: $O = \{O_1, ..., O_N\}$. Following the notation used in the two previous chapters for the parameters of template images, we will denote by λ_c the centroid of cluster $\mathcal{C}_c$. Thus, the measure of distance between I_n and cluster $\mathcal{C}_c$ is $P(O_n|\lambda_c, \lambda_{\mathcal{R}})$.

As our measure of distance is probabilistic, it is natural to use a ML framework to perform clustering [DHS00]. We assume that the distribution of the data can be modeled with a mixture of C components, where each component corresponds to one of the C clusters:

$$P(O_n|\lambda, \lambda_{\mathcal{R}}) = \sum_{c=1}^{C} w_c P(O_n|\lambda_c, \lambda_{\mathcal{R}}) \tag{7.1}$$

with $\lambda = \{w_1, ..., w_C, \lambda_1, ..., \lambda_C\}$. The mixture weights w_c are subject to the following constraint:

$$\sum_{c=1}^{C} w_c = 1 \tag{7.2}$$

We also assume that samples are drawn independently from the previous mixture.

$$P(O|\lambda) = \prod_{n=1}^{N} P(O_n|\lambda) \tag{7.3}$$

Our goal is to find the parameters $\{w_1, ..., w_C\}$ and $\{\lambda_1, ..., \lambda_C\}$ which maximize $P(O|\lambda)$. This problem cannot be solved directly and an iterative procedure based on the EM algorithm is generally used. The application of the EM algorithm to the problem of the estimation of mixture densities is based on the computation (E-step) and maximization (M-step) with respect to λ of Baum's auxiliary $\mathcal{Q}$ function. In this case, the hidden variable must include not only the state sequence Q, but also a variable Θ that indicates the mixture component (i.e. the cluster). Therefore, the $\mathcal{Q}$ function takes the following form:

$$\mathcal{Q}(\lambda|\lambda') = \sum_{Q} \sum_{\Theta} P(Q, \Theta|O, \lambda') \log P(O, Q, \Theta|\lambda) \tag{7.4}$$

If we split $\log P(O, Q, \Theta|\lambda)$ into $\log P(O, Q|\Theta, \lambda) + \log P(\Theta|\lambda)$, the $\mathcal{Q}$ function can be written as:

$$\mathcal{Q}(\lambda|\lambda') = \sum_{c=1}^{C} \sum_{n=1}^{N} \gamma_n^c \log(w_c) + \sum_{c=1}^{C} \sum_{n=1}^{N} \gamma_n^c \sum_{Q} \log P(O_n, Q|\lambda_c, \lambda_{\mathcal{R}}) \tag{7.5}$$

where the probability γ_n^c for image I_n to be assigned to cluster $\mathcal{C}_c$ is given by:

$$\gamma_n^c = P(\lambda'_c|O_n, \lambda_{\mathcal{R}}) = \frac{w'_c P(O_n|\lambda'_c, \lambda_{\mathcal{R}})}{\sum_{i=1}^{C} w'_i P(O_n|\lambda'_i, \lambda_{\mathcal{R}})} \tag{7.6}$$

We remind the reader that the T-HMM framework does not provide one value $P(\lambda'_c|O_n, \lambda_{\mathcal{R}})$ but a horizontal one $P^{\mathcal{H}}(\lambda'_c|O_n, \lambda_{\mathcal{R}})$ and a vertical one $P^{\mathcal{V}}(\lambda'_c|O_n, \lambda_{\mathcal{R}})$.

As was done in the previous chapters the horizontal and vertical statistics are averaged. From now on γ_n^c denotes this average.

To maximize $\mathcal{Q}(\lambda|\lambda')$, we can maximize independently the two terms. To find the optimal estimate $\hat{w}_c$ of w_c, we maximize the first term under the constraint 7.2 and obtain (the mathematical computations are similar to the ones derived for $w_{i,j}^k$ in section 5.5.3):

$$\hat{w}_c = \frac{1}{N} \sum_{n=1}^{N} \gamma_n^c \tag{7.7}$$

Maximizing the second term is one of the central issues of this chapter and is now addressed. Let $O_n = \{o_{i,j}^n, i = 1, ..., I, j = 1, ..., J\}$ and $\lambda_c = \{m_{k,l}^c, k = 1, ..., K, l = 1, ..., L\}$. In the following, we will use slightly different notations compared to the two previous chapters as our focus has been until now on the estimation of parameters indexed by the position (i, j) in the query image (e.g., $w_{i,j}^k$, $\delta_{i,j}^k$, $\Sigma_{i,j}^k$, etc.) while the focus is now on the estimation of parameters indexed by the location (k, l) in the template image. We denote by $\tau_{i,j}^{k,l}$ the translation vector which maps $o_{i,j}^n$ into $m_{k,l}^c$. $\gamma_{i,j}^{k,l}(n, c)$ is the probability of being in state $q_{i,j} = \tau_{i,j}^{k,l}$ at position (i, j) when matching O_n with λ_c and $\gamma_{i,j}^{k,l}(n, c, g)$ is the probability of being in state $q_{i,j} = \tau_{i,j}^{k,l}$ when matching O_n with λ_c with the g-th mixture component accounting for $o_{i,j}^n$. One more time, we do not have direct access to these quantities but to their horizontal and vertical statistics. In the following, $\gamma_{i,j}^{k,l}(n, c)$ and $\gamma_{i,j}^{k,l}(n, c, g)$ denote the arithmetic averages of the corresponding horizontal and vertical statistics.

The part of $\log P(O_n, Q|\lambda_c, \lambda_{\mathcal{R}})$ which contains $m_{k,l}^c$ is:

$$-\frac{1}{2} \sum_{i,j} \sum_{g} \gamma_{i,j}^{k,l}(n, c, g)(o_{i,j}^n - m_{k,l}^c - \delta_{i,j}^g)^T {\Sigma_{i,j}^g}^{(-1)} (o_{i,j}^n - m_{k,l}^c - \delta_{i,j}^g) \tag{7.8}$$

and therefore, the second part of equation 7.5 can be written as:

$$-\frac{1}{2} \sum_{n=1}^{N} \sum_{c=1}^{C} \gamma_n^c \sum_{i,j} \sum_{k,l} \sum_{g} \gamma_{i,j}^{k,l}(n, c, g)(o_{i,j}^n - m_{k,l}^c - \delta_{i,j}^g)^T {\Sigma_{i,j}^g}^{(-1)} (o_{i,j}^n - m_{k,l}^c - \delta_{i,j}^g) + cte \tag{7.9}$$

where cte is independent of the $m_{k,l}^c$'s. To maximize the $\mathcal{Q}$ function we take the partial derivative with respect to $m_{k,l}^c$:

$$-\sum_{n=1}^{N} \gamma_n^c \sum_{i,j} \sum_{g} \gamma_{i,j}^{k,l}(n, c, g) {\Sigma_{i,j}^g}^{(-1)} (o_{i,j}^n - m_{k,l}^c - \delta_{i,j}^g) \tag{7.10}$$

and equate it to zero to obtain the following estimate $\hat{m}^c_{k,l}$ of $m^c_{k,l}$:

$$\begin{aligned}\hat{m}^c_{k,l} \quad = \quad & \left(\sum_{n=1}^{N} \gamma^c_n \sum_{i,j} \sum_{g} \gamma^{k,l}_{i,j}(n,c,g) {\Sigma^g_{i,j}}^{(-1)}\right)^{-1} \times \\ & \left(\sum_{n=1}^{N} \gamma^c_n \sum_{i,j} \sum_{g} \gamma^{k,l}_{i,j}(n,c,g) {\Sigma^g_{i,j}}^{(-1)} (o^n_{i,j} - \delta^g_{i,j})\right) \end{aligned} \qquad (7.11)$$

The steps of the clustering algorithm are displayed in table 7.1.

```
1   Initialize C clusters.
2   For each observation $O_n$,
        for each cluster $c$,
            compute the distance $P(O_n|\lambda_c, \lambda_{\mathcal{R}})$ and accumulate statistics.
3   For each cluster c,
        update $w_c$ with equation 7.7,
        update $\lambda_c$ using equation 7.11.
4   Go back to step 2 until the likelihood $P(O|\lambda)$ converges.
```

Table 7.1: Basic clustering algorithm.

7.2.2 First experimental results

In this section, we present our first clustering results. All the experiments in this chapter were carried out on FERET as it was the only available database which contained a large enough number of persons to train our system and to assess its performance.

Cluster centroids were trained on the same data that was used to estimate the parameters of our face transformation model with local grid transformations only (c.f. section 5.7.1). We remind that this data consists of 695 persons with 2 images per person, which makes a total of 1,390 images. The performance of our clustering algorithm was tested on the same data that was used in section 5.7.4 to assess the performance of our system. We remind that it consists of 500 persons with 2 images per person, which makes a total of 1,000 images. Both training and test images exhibit mainly variations in facial expression. Each image was chosen successively as the query and the 999 remaining images were used as templates. The transformation model used to measure the distance between two face images is the one trained in section 6.6.1. The baseline performance of our system is the identification rate when each query is compared to all the templates, which is 95.7%. On a 2 GHz

Pentium 4 with 1 GB RAM, this set of comparisons takes on the order of 5 seconds. The goal is now to reach a similar performance but with a number of comparisons which is significantly smaller than $N = 999$.

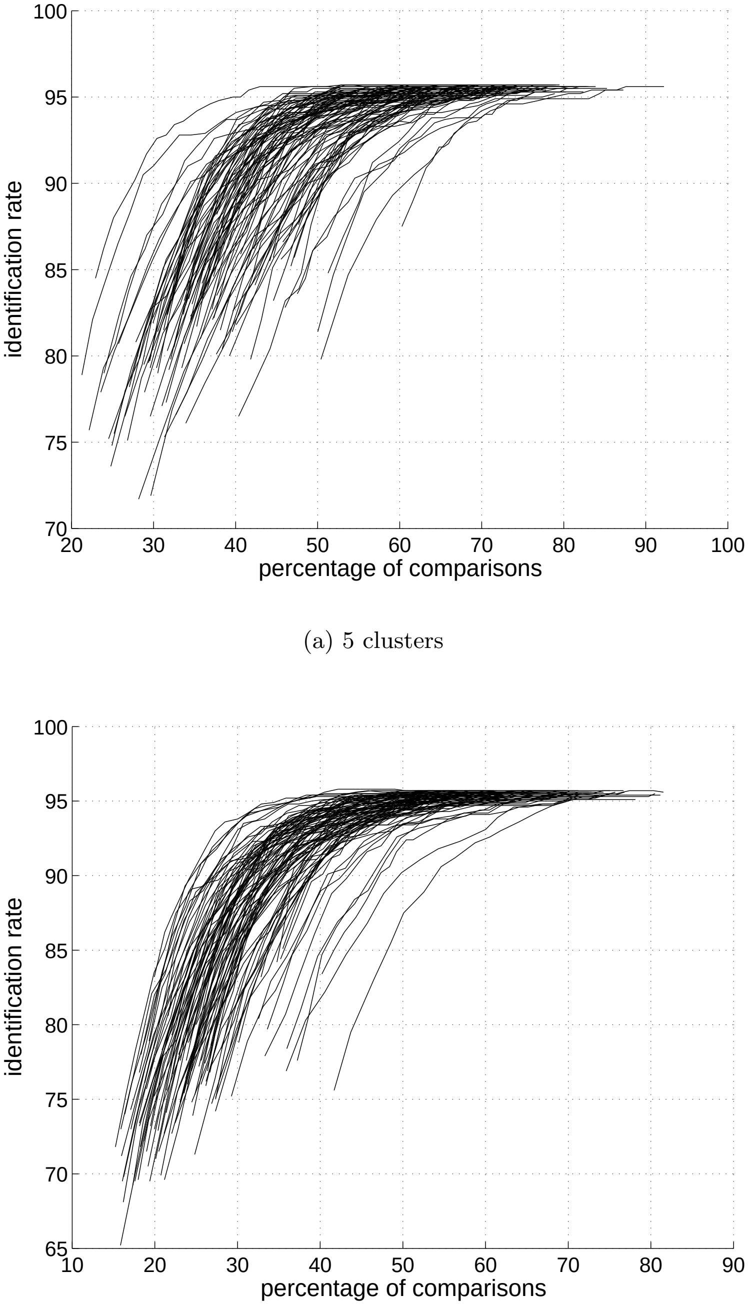
Clusters were trained exactly as described in Table 7.1. To initialize the C clusters we chose randomly a set of C images from the set of training images, making sure that no two images of the same person would be chosen as two cluster centroids. We performed 6 EM iterations to train clusters. The assignment of an enrollment or test image to a cluster or a set of clusters was done in the following manner. One computes $\gamma_n^c = P(\lambda_c|O_n, \lambda_{\mathcal{R}})$ for $c = 1, .., C$. Then the image I_n is assigned to cluster $\mathcal{C}_c$ if:

$$\gamma_n^c \geq \theta \tag{7.12}$$

θ is a threshold which has to be set according to the desired identification rate and the constraints on the computation time. The lower θ, the smaller the number of clusters to which an image is assigned and, thus, the lower the computational cost and the identification rate.

Consequently, to measure the performance we vary the value θ and plot the identification rate as a function of the percentage of comparisons which has to be performed compared to the case where we compare the query to the N templates. We have to take into account the comparisons with all cluster centroids and the comparisons with all the templates associated to the assigned cluster(s). As we consider in this chapter a flat clustering approach, if C is the number of clusters and N is the number of templates, then in the ideal case where clusters are perfectly balanced and where two images of the same person are always assigned to the same cluster, the number of comparisons is $C + N/C$. Note that this function reaches a minimum for $C = \sqrt{N}$ and thus the minimum number of comparisons is $2\sqrt{N}$. As in our case, $N = 999$, we know that we should not expect any improvement for more than 30 clusters approximately. We would like to outline that this is just an upper bound on the number of clusters.

The focus of this set of experiments is on the impact of the initialization on the performance of the system. Therefore, for a given number of clusters, we repeated 100 times the random initialization. Results are presented on Figures 7.2 (a) and (b) for 5 and 10 clusters respectively. Clearly, the choice of the initial cluster centroids has a huge impact on the performance in both cases. For instance, for 5 clusters the identification rate varies between approximately 75% and more than 90% if we perform 30% of the comparisons. Hence, a robust initialization procedure is required. This will be the focus of the next section.



(a) 5 clusters

(b) 10 clusters

Figure 7.2: Influence of the initialization on the performance.

7.3 Initializing cluster centroids

In this section, we first present an approach to cluster centroid initialization which is based on agglomerative clustering and which is specially tailored to suit our needs. We then provide experimental evidence that this approach leads to good initial centroid estimates.

7.3.1 An agglomerative clustering approach

While the EM procedure is bound to reach a local optimum, it is by no means guaranteed to reach the global one. The quality of the optimum which is found depends on several factors, one of which is the initialization of cluster centroids. Indeed, selecting the initial centroids in a random manner may lead to very different solutions as shown in the previous section.

One possible approach to solve this problem is to perform *cross-validation*. The development data set is split into a training set and an evaluation set. Different systems, corresponding to different initializations of cluster centroids, are trained on the training set and evaluated on the evaluation set. The system that performs the best on the evaluation set is then subsequently chosen. However, this approach has two noticeable shortcomings. It requires a large initial training set so that after the splitting there is enough data to train robustly cluster centroids and to carry out a statistically significant cross-validation. It requires also a very significant amount of computation as multiple systems have to be trained.

A simple procedure we employed to alleviate this problem was to perform as an initialization step a *hierarchical agglomerative* clustering [DHS00]. The goal is not to obtain the C best possible clusters but to obtain with a fast procedure reasonable seed centroids that can be subsequently fed to the EM procedure described in the previous section. The steps of a typical agglomerative clustering algorithm are described in table 7.2.

$\boxed{1}$	Initialize N clusters: $\mathcal{C}_n = \{O_n\}$.
$\boxed{2}$	While the number of clusters $> C$, find nearest clusters $\mathcal{C}_i$ and $\mathcal{C}_j$, merge $\mathcal{C}_i$ and $\mathcal{C}_j$.

Table 7.2: Hierarchical agglomerative clustering

Now we still have to define a distance between clusters. Let $d(x, y)$ be a measure of distance between two observations x and y. Various distances between clusters have been successfully used [DHS00]:

$$\begin{aligned}
\mathcal{D}_{min}(\mathcal{C}_i, \mathcal{C}_j) &= \min_{x \in \mathcal{C}_i, y \in \mathcal{C}_j} d(x, y) & (7.13)\\
\mathcal{D}_{max}(\mathcal{C}_i, \mathcal{C}_j) &= \max_{x \in \mathcal{C}_i, y \in \mathcal{C}_j} d(x, y) & (7.14)\\
\mathcal{D}_{avg}(\mathcal{C}_i, \mathcal{C}_j) &= \frac{1}{|\mathcal{C}_i||\mathcal{C}_j|} \sum_{x \in \mathcal{C}_i} \sum_{y \in \mathcal{C}_j} d(x, y) & (7.15)\\
\mathcal{D}_{mean}(\mathcal{C}_i, \mathcal{C}_j) &= d(\lambda_i, \lambda_j) & (7.16)
\end{aligned}$$

where $|\mathcal{C}_i|$ is the cardinality of cluster $\mathcal{C}_i$ and we remind that λ_i is the centroid of $\mathcal{C}_i$. Note that the first three distances only employ measures of distance between observations and require a similar amount of computation. Indeed, once distances between any pair of images have been precomputed the subsequent merging requires very little computation. On the other hand D_{mean} distance requires to estimate a cluster centroid at each merging step. While all these distances have a "minimum variance flavor" [DHS00] and thus are related to the ML criterion, which is the criterion of interest in our case, none of these criteria maximizes directly the likelihood.

Let $\{I_n\}$ be a set of images assigned to $\mathcal{C}_i$. The likelihood $\mathcal{L}(\mathcal{C}_i)$ of $\mathcal{C}_i$ is given by:

$$\mathcal{L}(\mathcal{C}_i) = \sum_{n : I_n \in \mathcal{C}_i} P(O_n | \lambda_i, \lambda_{\mathcal{R}}) \tag{7.17}$$

As we want a fast initialization procedure, we do not want to have to use the EM procedure to estimate λ_i. Thus we make use of the concept of *medoid* [KR90]: one chooses the most likely observation among the set of observations assigned to $\mathcal{C}_i$. Thus, if λ_{I_m} is the set of features extracted from I_m on a dense grid (template representation), then:

$$\lambda_i = \arg \max_{m : I_m \in \mathcal{C}_i} \sum_{n : I_n \in \mathcal{C}_i} P(O_n | \lambda_{I_m}, \lambda_{\mathcal{R}}) \tag{7.18}$$

Let us remind that, during the initialization step, the goal is to find the C cluster centroids which maximize the likelihood of the set of observations. After each merging stage, the likelihood of the set of observations will decrease. While the distances D_{min}, D_{max}, D_{mean} and D_{avg} search for the two clusters that, after merging, will lead to a cluster with the smallest possible variance, at each step, our goal is not to merge the two clusters that lead to the highest likelihood (or the minimum variance) but the clusters that lead to the *smallest decrease* of the likelihood. Hence, the distance between two clusters $\mathcal{C}_i$ and $\mathcal{C}_j$ is defined as the decrease in likelihood after the merging:

$$\mathcal{D}_{like}(\mathcal{C}_i, \mathcal{C}_j) = \mathcal{L}(\mathcal{C}_i) + \mathcal{L}(\mathcal{C}_j) - \mathcal{L}(\mathcal{C}_i \cup \mathcal{C}_j) \tag{7.19}$$

Note that this is similar to the criterion which is often used by Gaussian merging algorithms [San98]. While at each step we are guaranteed to obtain the smallest decrease in likelihood, we are not guaranteed that the sequence of steps leads to the global maximum.

If we apply directly this procedure, the clusters we obtain may be highly unbalanced, i.e. some clusters may be assigned a large number of data items while others may contain only a small number of data items. This is a problem as a cluster centroid cannot be robustly estimated with a too small number of data items. Moreover, for the problem under consideration clusters should be as balanced as possible to perform the minimum possible number of comparisons at test time. When we looked at the steps of the clustering algorithm, we noticed that after a few steps one cluster was getting bigger and bigger by merging with other clusters containing only a few data items (generally one). This was rather surprising as $\mathcal{D}_{like}(\mathcal{C}_i, \mathcal{C}_j)$ is automatically weighted by the the number of data items assigned to $\mathcal{C}_i$ and $\mathcal{C}_j$ and this should prevent a large cluster of getting blindly bigger. However, this statement is true only in the case where the cluster centroids of $\mathcal{C}_i$, $\mathcal{C}_j$ and $\mathcal{C}_i \cup \mathcal{C}_j$ are estimated exactly. In our case, we use an approximate estimate of the cluster centroid: the medoid. Let n_i and n_j be respectively the number of images assigned to $\mathcal{C}_i$ and $\mathcal{C}_j$ and let us assume that $n_i \gg n_j$. Let λ_i and λ_j be the centroids of $\mathcal{C}_i$ and $\mathcal{C}_j$ respectively. When searching for the centroid of $\mathcal{C}_i \cup \mathcal{C}_j$, as $n_i \gg n_j$, the observations of $\mathcal{C}_j$ will have very little influence and it is likely that λ_i will be the new centroid for $\mathcal{C}_i \cup \mathcal{C}_j$. In such a case we obtain:

$$\begin{aligned}
\mathcal{D}_{like}(\mathcal{C}_i, \mathcal{C}_j) &= \sum_{n:I_n \in \mathcal{C}_i} P(O_n|\lambda_i, \lambda_{\mathcal{R}}) + \sum_{n:I_n \in \mathcal{C}_j} P(O_n|\lambda_j, \lambda_{\mathcal{R}}) - \sum_{n:I_n \in \mathcal{C}_i \cup \mathcal{C}_j} P(O_n|\lambda_i, \lambda_{\mathcal{R}}) \\
&= \sum_{n:I_n \in \mathcal{C}_j} P(O_n|\lambda_j, \lambda_{\mathcal{R}}) - \sum_{n:I_n \in \mathcal{C}_j} P(O_n|\lambda_i, \lambda_{\mathcal{R}}) \qquad (7.20)
\end{aligned}$$

which is independent of n_i.

Hence, we should find a way to penalize the previous distance in order to take into account the balance between clusters. Let n_i be the number of data items in cluster $\mathcal{C}_i$ and let N be the total number of data items. We also introduce $p_i = n_i/N$. Clearly, the entropy [CT93] (see also appendix B):

$$\mathcal{H} = -\sum_{i=1}^{N} p_i \log(p_i) \qquad (7.21)$$

is a measure of balance as, the larger $\mathcal{H}$, the more balanced is the set of clusters. Let $\mathcal{H}$ be the entropy for the set of clusters $\{\mathcal{C}_1, ...\mathcal{C}_C\}$. If we merge clusters $\mathcal{C}_i$ and

$\mathcal{C}_j$, then the delta entropy will be:

$$\Delta\mathcal{H}(\mathcal{C}_i, \mathcal{C}_j) = p_i \log(p_i) + p_j \log(p_j) - (p_i + p_j)\log(p_i + p_j) \tag{7.22}$$

which is a negative quantity. The closer is this quantity to zero, the smaller the reduction of entropy, and thus the smaller the reduction of the "balance" in our system.

Hence, we use as a measure of distance between two clusters $\mathcal{C}_i$ and $\mathcal{C}_j$:

$$\mathcal{D}(\mathcal{C}_i, \mathcal{C}_j) = \mathcal{D}_{like}(\mathcal{C}_i, \mathcal{C}_j) - \rho\Delta\mathcal{H}(\mathcal{C}_i, \mathcal{C}_j) \tag{7.23}$$

where ρ is a parameter that keeps the balance between the two possibly competing criteria: the minimum likelihood decrease versus the maximum entropy decrease.

7.3.2 Experimental results

We now evaluate the proposed initialization algorithm. We first address the choice of a proper value ρ. For this set of experiments, we did not perform the subsequent ML training described in section 7.2.1. On Figure 7.3, we draw the performance of the system with 20 clusters for different values of ρ. The optimum seems to be for $\rho = 100$ and this is the value we will use for all the following experiments in this chapter. Note that the identification rate seems to vary smoothly with large variations of ρ which indicates that it is a robust parameter.

We now evaluate the impact of the ML training. Results are presented on Figure 7.4 for $C = 20$ clusters. The ML training only brings a moderate improvement. In the next section, we will attempt to improve the quality of the cluster centroid estimates.

7.4 Dealing with Data Scarcity

We believe that the main reason for the rather small improvement brought by the ML training is the lack of training data. Indeed, as feature vectors are extracted on a sparse grid from query images and on a dense grid from template images, one may need a very large number of images to estimate robustly $m_{k,l}^c$, $\forall c$ and $\forall(k, l)$. If little data is available, some parameters $m_{k,l}^c$ may even not be updated at all. A potential solution to this problem would be to extract feature vectors from query images on a finer grid but this would increase the computational cost, an unwanted effect as our goal is to speed-up the comparison between a query image and a set of template images. A more appropriate solution would be to adapt the cluster centroid parameters rather than training them in an ML fashion. In this section, we first provide a very brief review of the literature on adaptation techniques for continuous

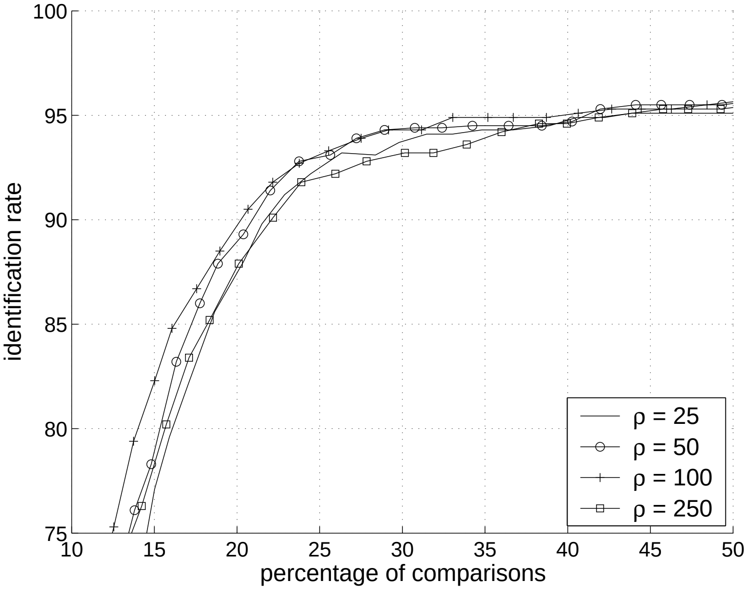


Figure 7.3: Performance of the system without ML training for various values of the parameter ρ for $C = 20$ clusters.

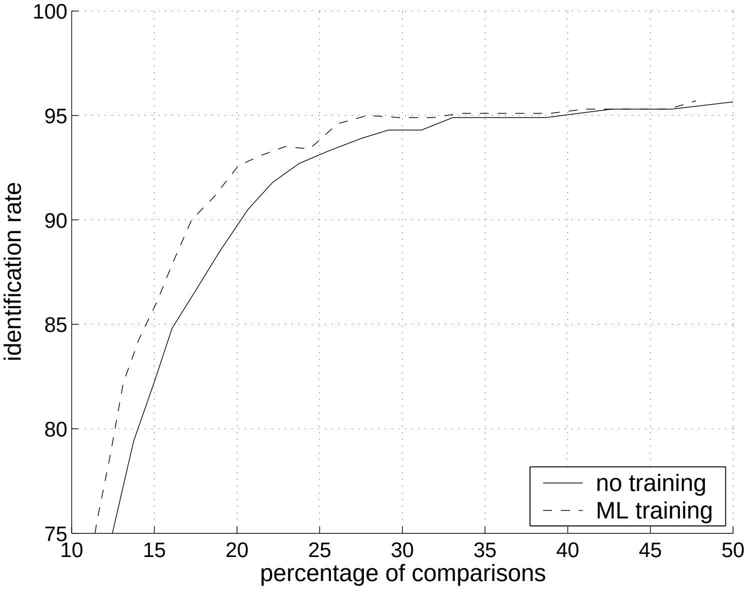


Figure 7.4: Impact of the training for $C = 20$ clusters.

density HMMs. We then explain why the *eigenvoice* technique is chosen and how to apply it to the problem of interest. Finally, we present experimental results to assess the impact of the eigenvoice approach on the estimation of the centroids.

7.4.1 Adaptation techniques for continuous density HMMs

There exists three main classes of adaptation techniques for continuous density HMMs: *MAP* adaptation, *linear transforms* and *clustering / eigenvoices* [Woo01]. In the following, we will focus our attention on the problem of Gaussian mean estimation as this is the problem of interest.

In MAP parameter estimation, the set of parameters λ is chosen to maximize $P(O|\lambda)P(\lambda)$ where $P(\lambda)$ is the prior distribution of parameters. The use of a prior distribution alleviates the need of a very large data set to get robust parameter estimates. MAP estimation is relatively easy if the prior density is from the same family as the posterior distribution (conjugate prior) if it exists. For HMMs with mixtures of Gaussians densities, such a conjugate prior of finite dimension does not exist and an alternative approach is generally used [GL94]. For a particular Gaussian mean μ, the MAP estimate is:

$$\hat{\mu} = \frac{\tau \mu_0 + \sum_t \gamma_t o_t}{\tau + \sum_t \gamma_t} \tag{7.24}$$

where τ is the parameter which gives the bias between the ML estimate of the mean and the prior mean μ_0, o_t is the training vector at time t and γ_t is the occupancy probability of the considered Gaussian at time t. A key advantage of MAP is that the MAP estimate converges to the ML estimate as the amount of training data increases to infinity. Its main drawback is that MAP is a local approach to updating the parameters, i.e. only parameters that are observed in the adaptation data will be re-estimated. To address this issue, several techniques have been proposed. For instance, the structural MAP (SMAP) organizes Gaussians into a tree structure and parameters are re-estimated starting from the root, using the distribution from the node above as a prior [SL97].

An alternative approach is to estimate a linear transformation of model parameters. The popular maximum likelihood linear regression (MLLR) [LW95] updates a mean according to the following equation:

$$\hat{\mu} = A\mu + b \tag{7.25}$$

where A is a $n \times n$ matrix, b is a n dimensional vector and n is the dimensionality of the feature vectors. It can be shown that there exists a closed form solution to the ML estimation of A and b using the EM algorithm. A trade-off has to be found between robust adaptation via a global transform and using specific transforms that apply to a smaller group of Gaussians. While MLLR is faster than MAP it can lead to a decrease of the performance compared to a non-adapted model for very small amounts of adaptation data.

The two previous approaches do not have any information about the correlation between the parameters of the HMM. Clustering strategies such as the cluster adaptive training (CAT) [Gal00] and the eigenvoice approach [KJNN00] apply constraints on the location of parameters in the whole parameter space to drastically reduce the amount of data required for a robust estimation. While the goal of CAT is to represent a model as a weighted sum of reference models, the aim of eigenvoices is to represent a model as an average model plus a weighted sum of vectors which represent the principal directions of variabilities, the eigenvoices. In both cases the mean update can be written as:

$$\hat{\mu} = \mu(0) + \sum_{e=1}^{E} w_e \mu(e) \tag{7.26}$$

In the case of CAT, $\mu(0)$ is a bias model and the $\mu(e)$'s are the reference models. In the case of eigenvoices, $\mu(0)$ is the average model and the $\mu(e)$'s are the eigenvoices. As was the case for MLLR, there exists a closed form solution to the ML estimation of the weights $\{w_1, ..., w_E\}$ using the EM algorithm. While eigenvoices and CAT require very little data, their performance reaches a plateau for large amounts of adaptation data.

Note that these algorithms can be combined to make the most out of their complementary natures. For instance, as MAP is fairly slow compared to MLLR or eigenvoices, the MLLR or eigenvoice estimates can serve as a prior for MAP.

7.4.2 Adapting cluster centroids using eigenvoices

In the following, we will choose the eigenvoice approach to estimate cluster centroids. This choice is motivated by the fact that we have little data to estimate cluster centroids (some of the $m_{k,l}^c$ parameters may not be re-estimated) and by the strong correlation that exists between the $m_{k,l}^c$'s, especially at adjacent positions.

For the problem of interest, the parameters we want to adapt are $\{\lambda_1, ..., \lambda_C\}$. We assume that we have performed PCA on a set of template images. $\mu(0) = \{\mu_{k,l}(0), k = 1, ..., K, l = 1, ..., L\}$ denotes the average vector and $\mu(e) = \{\mu_{k,l}(e), k = 1, ..., K, l = 1, ..., L\}$ is the e-th eigenvector which represents the e-th direction of variation. If we constrain the cluster centroid λ_c to lie in the subspace spanned by the first E principal components, then we can write:

$$m_{k,l}^c = \mu_{k,l}(0) + \sum_{e=1}^{E} w_e^c \mu_{k,l}(e) \qquad \forall(k,l) \tag{7.27}$$

The goal is now to estimate the weights w_e^c. This procedure is known as the maximum likelihood eigen-decomposition (MLED). If we incorporate equation 7.27 into equation 7.9, and take the partial derivative with respect to w_e^c, we obtain:

$$-\sum_{n=1}^{N} \gamma_n^c \sum_{i,j} \sum_{g} \gamma_{i,j}^{k,l}(n,c,g) \mu_{k,l}(e)^T {\Sigma_{i,j}^g}^{-1} \left(o_{i,j}^n - \delta_{i,j}^g - \mu_{k,l}(0) - \sum_{f=1}^{E} w_f^c \mu_{k,l}(f) \right) \tag{7.28}$$

Now if we equate this quantity to zero, we obtain the following estimate $\hat{w}_e^c$ of w_e^c:

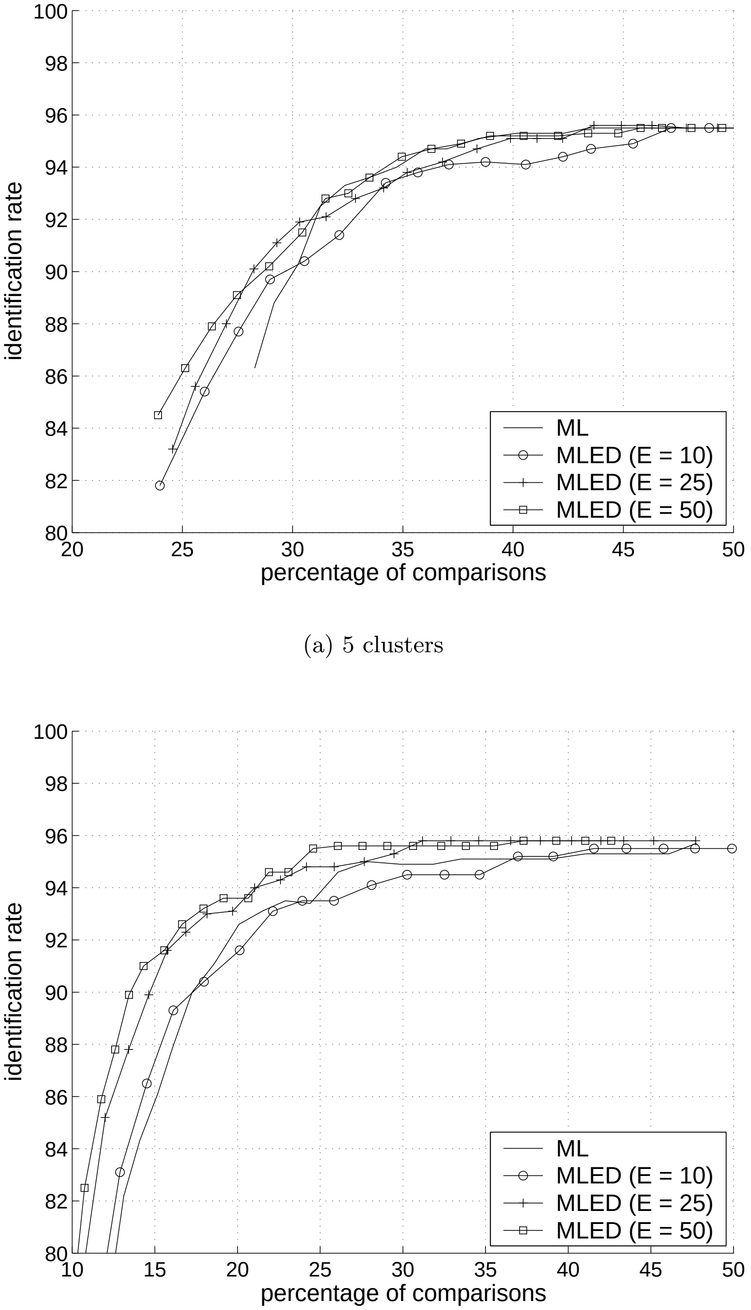
$$\sum_{f=1}^{E} \hat{w}_f^c \left[\sum_{k,l} \mu_{k,l}(e)^T \left(\sum_{n=1}^{N} \gamma_n^c \sum_{i,j,g} \gamma_{i,j}^{k,l}(n,c,g) {\Sigma_{i,j}^g}^{-1} \right) \mu_{k,l}(f) \right] =$$
$$\sum_{k,l} \mu_{k,l}(e)^T \sum_{n=1}^{N} \gamma_n^c \sum_{i,j,g} \gamma_{i,j}^{k,l}(n,c,g) {\Sigma_{i,j}^g}^{-1} \left(o_{i,j}^n - \delta_{i,j}^g - \mu_{k,l}(0) \right) \tag{7.29}$$

Thus,we obtain C linear systems of E equations with E unknowns (one per cluster). Generally, E is fairly small and thus inverting the corresponding matrix requires little computation.

It should be underlined that a space derived using PCA is not optimal for our problem as it minimizes a mean square error while the criterion of interest is ML. In [Ngu99] the maximum likelihood eigenspace (MLES) was proposed to address this issue. The basic idea is to incorporate equation 7.27 into equation 7.9, to take the partial derivative with respect to $\mu_{k,l}(e)$ and then to equate it to zero. However, in our case we cannot apply this principle as the amount of data required to train the eigenspace in a ML fashion is comparable to the amount of data required to train cluster centroids with ML.

7.4.3 Experimental results

We now compare the performance of MLED adaptation to ML training. Results are presented on Figures 7.5 (a) and (b) for 5 and 20 clusters respectively. For 5 clusters, MLED does not improve over ML. MLED may even result in a decrease of the performance if the number of eigenvectors E is chosen too small. It seems to indicate that there is enough data to train 5 clusters with ML. For 20 clusters MLED outperforms ML if the number of eigenvectors is large enough (i.e. $E \geq 25$),



(a) 5 clusters

(b) 20 clusters

Figure 7.5: Comparison of the ML training with the MLED adaptation.

especially when the percentage of comparisons is small. For instance, when the percentage of comparisons is 15% the performance of the ML system is approximately 86% while the performance of the MLED system with $E = 50$ is more than 91%. We attempted to perform a MAP post-smoothing but no improvement was obtained.

We believe that the increase in performance is due both to the use of the correlation between feature vectors but also to the smoothing effect of MLED. If enough data is available to train the cluster centroid of $\mathcal{C}$, then the intra-class variability will be averaged. However, in the case where only a small number of observations are assigned to a cluster $\mathcal{C}$, these variations may not cancel and irrelevant intra-class variability may "contaminate" the centroid. However, when using a small number of eigenfaces in MLED, the face is only imperfectly reconstructed. Thus, if irrelevant variabilities are not modeled by the first directions of variation of the PCA space, they will be canceled.

7.5 Alleviating the Multi-Class Assignment

In this section, we first show that the multi-class assignment is likely to result in wasteful comparisons and that to avoid this paradigm, one can assign each face image to all classes probabilistically. We then present experimental results to assess the performance of the probabilistic assignment.

7.5.1 Probabilistic assignment

A limitation of the current approach is that we do not make the most out of the available information. To make our argument clear, let us assume that the face space is partitioned as depicted on Figure 7.6. When the template image I_t is added to the database, it is likely to be assigned to clusters $\mathcal{C}_6$, $\mathcal{C}_7$ and $\mathcal{C}_8$. At test time,

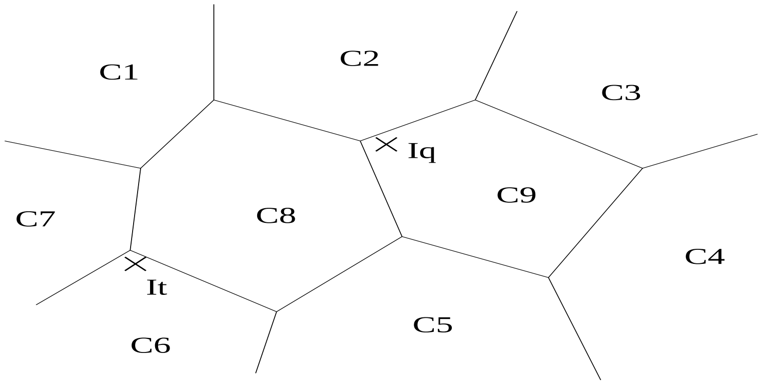


Figure 7.6: Uncertainty in cluster assignment.

the query image I_q is first assigned to $\mathcal{C}_2$, C_8 and C_9 and then compared to all the template images contained in one of these clusters, which includes I_t. However, I_t and I_q are fairly distant and, thus, unlikely to belong to the same person. Therefore, such a comparison will most likely be wasteful. The reason why I_t and I_q were compared while they should not have been is that, when assigning an image to one or multiple clusters, we throw away a lot of valuable information: the "distances" $P(O_n|\lambda_c, \lambda_{\mathcal{R}})$. Indeed, the vector $v = [P(O_n|\lambda_1, \lambda_{\mathcal{R}}), ..., P(O_n|\lambda_C, \lambda_{\mathcal{R}})]^T$ could be used to characterize the face image.

A similar approach has already been proposed in the field of speaker detection and indexing. In [SRSC01], a speech utterance s is scored against a set of models $\{A_1, ..., A_N\}$ referred to as *anchors* and the vector $v = [P(s|A_1), ..., P(s|A_N)]^T$ is used to characterize the speech utterance. This characterization vector can be considered as a projection of the target image into a speaker space. We propose two major improvements over the original anchor modeling approach:

- As in [SRSC01] the number of anchor models was large ($N = 668$ in their experiments), methods for reducing the size of the Euclidean distance comparison were investigated in an effort to increase performance by using only those anchor models that provide good characterizing information. However, such an approach does not reduce the cost of computing v which can also be significant. In the proposed approach, our anchors are not faces but the centroids which are obtained after clustering a set of face images. The clustering step should therefore perform a dimension reduction and drastically decrease the cost of computing v and of comparing it with other vectors.

- Instead of using a characterization vector v based on the likelihood, we propose to use posterior probabilities: $v = [P(\lambda_1|O_n, \lambda_{\mathcal{R}}), ..., P(\lambda_C|O_n, \lambda_{\mathcal{R}})]^T$. Such a vector should be more robust as it normalizes the likelihood.

Let v_q be the characterization vectors of I_q. Then at test time, we first compute the distance between v_q and the characterization vectors of all template images contained in the database. Although there are as many distances to compute as template images, this is very fast as these vectors are very low dimensional. Then I_q is compared with the template images I_t that are less than a given threshold distant from I_q.

Note that this approach can be seen as a special case of the first method described in the introductory section to reduce the computational cost. The characterization vectors are shrewd representations of the images and thus identification based purely on these vectors has a low accuracy. However, they are fairly fast to estimate and very fast to compare. An interesting property of such an approach is that the characterization vector retains the properties of the costly distance. Indeed, if the distance is robust to some variations, then the characterization vector should not be significantly affected by these variations, a property that is not discussed in [SRSC01].

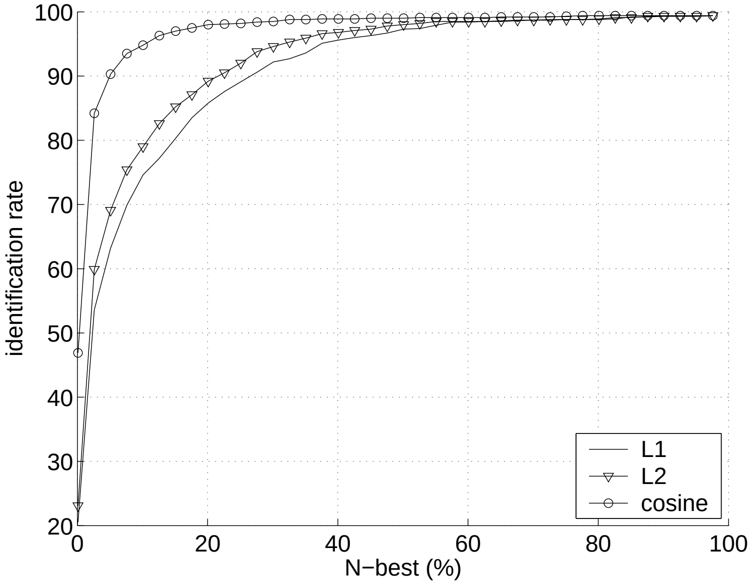
7.5.2 Experimental results

As opposed to the three previous sections, the focus is not on the estimation of cluster centroids but on the best way to use them. Therefore, we will make use of the best centroid estimates we have obtained, i.e. the ones that have been adapted with MLED (with $E = 50$ eigenvectors) in the previous section.

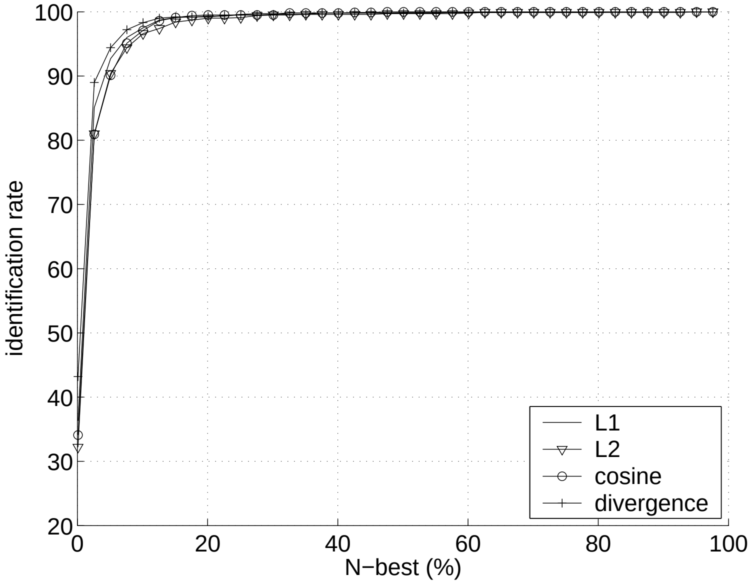
The goal of the first set of experiments is to determine 1) which distance is the most appropriate to measure the similarity of characterization vectors and 2) whether the characterization vector based on posteriors is superior to the one based on likelihoods. Thus, in this first set of experiments, we perform identification with the characterization vectors only. We tested the L_1, L_2 and cosine metrics on both types of characterization vectors. As a posterior-based characterization vector defines a discrete probability distribution, we also tried the symmetric divergence on this type of vectors (c.f. appendix B).

Note that the likelihoods $P(O_n|\lambda_c, \lambda_{\mathcal{R}})$ are extremely large (on the order of $10^{10,000}$) and thus they are difficult to compare directly. Therefore, in the following we did not use likelihood-based characterization vectors but characterization vectors based on the log-likelihood. In the same manner, $P(O_n|\lambda_c, \lambda_{\mathcal{R}})$'s are so large that the posteriors $P(\lambda_c|O_n, \lambda_{\mathcal{R}})$ are equal to 1 for the most likely centroid and 0 for the other ones. Thus, to increase the fuzziness of the assignment, we raised the posteriors to the power of a small positive factor β and then renormalized them so that they would sum to unity. In the following experiments we set $\beta = 0.01$.

Results are presented for $C = 20$ clusters on Figure 7.7 On Figure 7.7 (a), we compare the performance of the L_1, L_2 and cosine metrics for characterization vectors based on the log-likelihood. Clearly, the cosine is by far the best choice. On Figure 7.7 (b), we compare the performance of the L_1, L_2, cosine and symmetric divergence metrics for posterior-based characterization vectors. Results are improved for the first three metrics (especially for L_1 and L_2) compared to log-likelihood-based vectors. The four measures of distance exhibit a similar performance but the symmetric divergence seems to outperform the three other metrics by a slight margin. Hence, in the following experiments, we will use posterior-based characterization vectors and the similarity of two such vectors will be measured with the symmetric divergence.



(a)



(b)

Figure 7.7: Performance of a system with $C = 20$ clusters which makes use of (a) log-likelihood-based characterization vectors (b) posterior-based characterization vectors. Cumulative identification rate versus N-best (as a percentage of the database).

.

Now that we have chosen the type of characterization vector and the metric, we can evaluate the performance of our system when characterization vectors are used during a pre-processing step to find the most likely candidates. As was the case in the previous sections, we present results as the identification rate versus the percentage of comparisons for various numbers of clusters (Figure 7.8). While the increase of performance from 5 to 10 clusters is very significant, especially for a small number of comparisons, it is smaller when going from 10 to 20 clusters. No improvement could be obtained with more than 20 centroids. This shows that, for the problem of interest, clustering is very important as only a very small number of clusters is required to reach the best performance.

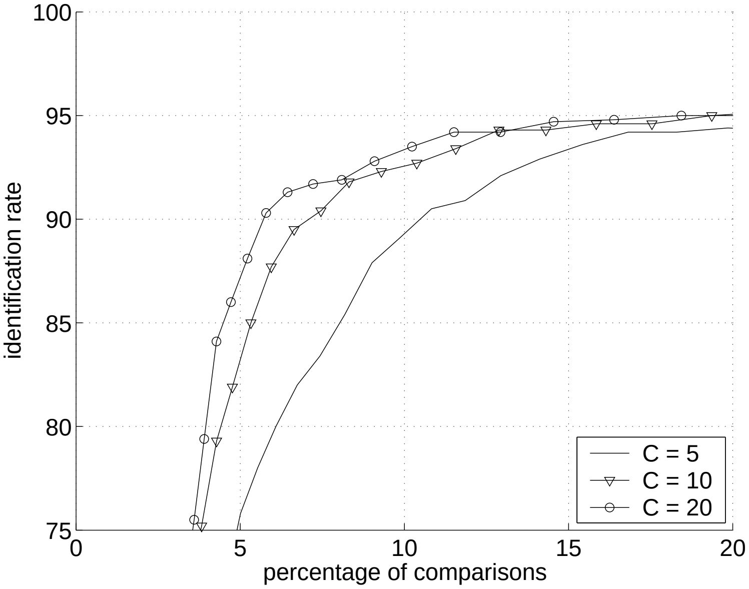


Figure 7.8: Performance of the system with probabilistic cluster assignment for a varying number C of clusters.

Now if we compare the results of Figure 7.8 with those of Figure 7.5, we can see that the probabilistic assignment leads to a much better performance, especially for a small number of comparisons. We remind that the performance of the baseline system is a 95.7% identification rate. We can see on Figure 7.8 that we can obtain a 95% identification rate with only 15% of the computation.

7.6 Conclusion

In this section, we have worked on the challenging problem of image clustering with the aim of partitioning the image space. Our work has first focused on the robust estimation of cluster centroids. We first applied the general EM framework to our problem. We then addressed two important issues. We first proposed a simple, but effective, initialization procedure to estimate the seed cluster centroids that will be subsequently fed to the EM procedure. We then addressed the problem of data scarcity by adapting the model parameters using the eigenvoice approach rather than estimating them with ML. We also discussed an alternative to the multiple cluster assignment paradigm which consists in assigning a face image to all clusters probabilistically. Note that we are aware that a similar approach has already been proposed but we have suggested two major improvements which have been shown to increase the performance of our system very significantly.

While this work has been applied to speed up the comparison of a query image with a set of template images, it has numerous applications. First of all, the ability to estimate a centroid may be very useful when multiple images are available to estimate a face model. These images may either be available all at enrollment time or they may be available later to adapt the face model. In the next chapter, we will estimate a centroid for the problem of identity verification. The ability to cluster images also has many applications. Multiple transformation models could be estimated, one per cluster for instance, to better model the intra-class variability of each class of persons.

8

Modeling Wrongful Claims for Robust Verification

8.1 Introduction

A biometrics authentication/verification system accepts or rejects a person based on a claimed identity and a sample of the considered biometrics. Hence, authentication is a two-class decision problem and the success of an authentication system is based on the accurate *modeling of both rightful and wrongful claims.* Although this framework has long been applied to other biometrics such as speaker verification [Rey95, RP96, RQD00], surprisingly, the issue of modeling wrongful claims seems to have drawn very little attention from the face verification community as outlined in [SP02].

Indeed, let M_C be the model of a client C and let I_q be a query image. Then a naïve thresholding of the score leads to the following test:

$$P(I_q|M_C) \gtrless \theta \tag{8.1}$$

However, it is impossible to find a threshold θ which can be used in all conditions. If the threshold is set too high, clients might be wrongfully rejected in the case of a mismatch between the training and test conditions, which will result in a high FRR. On the other hand, if it is set too low to accommodate for all possible mismatches, then the FAR will increase dramatically. A more robust approach to verification is to train an impostor model $\overline{M}_C$ for C, and to perform the following comparison:

$$\frac{P(M_C|I_q)}{P(\overline{M}_C|I_q)} \gtrless \theta \tag{8.2}$$

In the case of a mismatch between the training and test conditions, the decrease of the likelihood $P(M_C|I_q)$ is likely to be compensated by a similar decrease of $P(\overline{M}_C|I_q)$. Therefore, the ratio will be relatively unaffected by the mismatch. Note that this technique can be applied to relational approaches to AFR such as the BIC or the proposed PMLT if we consider that the combination of I_t and $\mathcal{R}$ is the model M_C of client C.

However, in the case of relational approaches, another approach has also been suggested in [MWP98, Mog02]. While the previous approach raises the question: *"Is this sample more likely to belong to C or to and impostor of C?"*, the relational approach to impostor modeling considers the following question: *"Is the observed variability between the template image I_t and the query image I_q more likely to be intra-class or inter-class variability?"*. Let $\overline{\mathcal{R}}$ be the relationship between face images of two different persons. The acceptance/rejection decision is therefore based on the following likelihood ratio:

$$\frac{P(\mathcal{R}|I_q, I_t)}{P(\overline{\mathcal{R}}|I_q, I_t)} \tag{8.3}$$

As we will see in the remainder of this chapter, although the questions raised by these two approaches to modeling wrongful claims are semantically very close, they lead in effect to two very different classifiers. Which of these two approaches to modeling wrongful claims is the more robust is not obvious and this question is the focus of this chapter.

The remainder of this chapter is organized as follows. In the next section, we describe with more details the two possible strategies to verification. In section 8.3, we compare them from a theoretical point of view. In section 8.4 we present an experimental comparison carried out on the FERET, AR and PIE databases. The focus this time is not on the comparison of BIC and PMLT but on the comparison of the two approaches to impostor modeling. Both the theoretical and experimental comparisons indicate that the latter approach results in a better performance, especially in the challenging case where variabilities that were not learned during the training phase are observed at test time.

8.2 Two Strategies to Impostor Modeling

To illustrate the two authentication strategies for relational approaches, we make use of the simple Gaussian classifier considered in [MP97, MWP98, Mog02]. The difference between face images of the same person is supposed to be a normally distributed random variable. If we denote $\delta = I_q - I_t$, then:

$$P(\delta|\mathcal{R}) = \frac{1}{(2\pi)^{N/2}|S|^{1/2}} \exp\left\{-\frac{1}{2}\delta^T S^{-1} \delta\right\} \tag{8.4}$$

where N is the dimension of the image space, i.e. the number of pixels in I_q or I_t. The covariance matrix S is the only parameter and is estimated with pairs of images of the same person. Although this classifier has little practical value due to the high dimensionality of the image space, the theoretical comparison of both strategies to authentication on this classifier is simple. Interestingly, $P(\delta|\mathcal{R}) \equiv P(I_q|I_t, \mathcal{R})$ with:

$$P(I_q|I_t, \mathcal{R}) = \frac{1}{(2\pi)^{N/2}|S|^{1/2}} \exp\left\{-\frac{1}{2}(I_q - I_t)^T S^{-1}(I_q - I_t)\right\} \tag{8.5}$$

The difference is that the notation $P(\delta|\mathcal{R})$ assumes that δ is emitted by a Gaussian with zero mean while the notation $P(I_q|I_t, \mathcal{R})$ assumes that I_q is emitted by a Gaussian with mean I_t. We will see in the following that the difference between $P(\delta|\mathcal{R})$ and $P(I_q|I_t, \mathcal{R})$ is not purely notational.

8.2.1 Relational approach

Generalizing the approach to authentication introduced in [MWP98, Mog02], acceptance/rejection should be based on the following test ratio:

$$\frac{P(\mathcal{R}|I_q, I_t)}{P(\overline{\mathcal{R}}|I_q, I_t)} \gtrless \theta \tag{8.6}$$

where θ is an application dependent threshold which is set according to the desired level of security / convenience. However, as $P(\mathcal{R}|I_q, I_t)$ and $P(\overline{\mathcal{R}}|I_q, I_t)$ are difficult to estimate, one uses Bayes' formula to rephrase the previous test ratio as follows:

$$\frac{P(I_q|I_t, \mathcal{R})}{P(I_q|I_t, \overline{\mathcal{R}})} \gtrless \theta' \tag{8.7}$$

where θ' now incorporates also $P(\mathcal{R}|I_t)$ and $P(\overline{\mathcal{R}}|I_t)$, respectively the probabilities of a client or an impostor trial on the template I_t. If the difference between face images of different persons is also assumed to be Gaussian [Mog02], then we can write:

$$P(I_q|I_t, \overline{\mathcal{R}}) = \frac{1}{(2\pi)^{N/2}|\overline{S}|^{1/2}} \exp\left\{-\frac{1}{2}(I_q - I_t)^T \overline{S}^{-1}(I_q - I_t)\right\} \tag{8.8}$$

where $\overline{S}$ is estimated on pairs of images of different persons.

The relational approach to score normalization will be later referred to as R-norm.

8.2.2 Generative approach

If I_t is the template image for client C, then in the expression $P(I_q|I_t, \mathcal{R})$, $(I_t, \mathcal{R})$ can be seen as a model of C: $M_C \equiv (I_t, \mathcal{R})$. Note that grouping I_t and $\mathcal{R}$ would not have been possible if we had kept the notation δ. Let $\overline{M}_C$ denote the anti-model of C, i.e. the model of all the impostors that could try to gain access to the system by claiming the identity of C. Then the classical approach to verification in this case is:

$$\frac{P(M_C|I_q)}{P(\overline{M}_C|I_q)} \gtrless \theta \tag{8.9}$$

Using one more time Bayes' formula, we get:

$$\frac{P(I_q|M_C)}{P(I_q|\overline{M}_C)} = \frac{P(I_q|I_t, \mathcal{R})}{P(I_q|\overline{I_t, \mathcal{R}})} \gtrless \theta' \tag{8.10}$$

where θ' now incorporates $P(M_C)$ and $P(\overline{M}_C)$, respectively the probabilities of a client and an impostor trials on the model M_C.

There exists two traditional approaches to model $\overline{M}_C$: cohort models and background models [RP96]. The first approach uses a set of *cohorts* $\{C_1, ..., C_K\}$ whose selection might depend on the client C and a score is obtained for each cohort. The normalization score is of the form:

$$P(I_q|\overline{M}_C) = \frac{1}{K} \sum_{k=1}^{K} P(I_q|M_{C_k}) \tag{8.11}$$

The second approach makes use of a single model constructed by pooling training utterances from more than one person, whose selection might also depend on C. It was shown in [Rey97] that a background model trained with a large amount of data and which is not specific to C, also referred to as universal background model (UBM), leads to a very robust normalization score. As the focus of this paper is not on the comparison between these two approaches (see [RQD00] for such a comparison in the case of face authentication), and as the UBM approach is simple and performs very well, we use a UBM denoted U.

In practice, if $P(I_q|\overline{I_t, \mathcal{R}}) = P(I_q|U)$ is also supposed to be Gaussian, the parameters of U, which include this time both the mean and the covariance matrix, are simply estimated with training data from a large number of people. Let μ and Σ denote respectively the mean and covariance of this distribution:

$$P(I_q|\overline{I_t,\mathcal{R}}) = \frac{1}{(2\pi)^{N/2}|\Sigma|^{1/2}} \exp\left\{-\frac{1}{2}(I_q-\mu)^T\Sigma^{-1}(I_q-\mu)\right\} \tag{8.12}$$

The “generative” approach to score normalization will be later referred to as G-norm.

8.3 A Theoretical Comparison

While the likelihood ratios of both strategies to authentication have the same numerator (see equations 8.7 and 8.11), i.e. while client distributions are modeled in the same manner, denominators are different as $\overline{M_C} = \overline{(I_t,\mathcal{R})} \neq (I_t,\overline{\mathcal{R}})$ and thus the distribution of wrongful claims is modeled differently. Since $\Sigma \approx \overline{S}$ (e.g. see [WT03]), for this Gaussian classifier the main difference between $P(I_q|I_t,\overline{R})$ and $P(I_q|\overline{I_t,\mathcal{R}})$ is in the means of the Gaussians. In effect, R-norm and G-norm are very different. R-norm measures the amounts of intra- and inter-class variabilities between two images while G-norm measures a sort of distance between the query image and the data used to train the universal model U.

We will now show with two arguments that, from a theoretical point of view, G-norm is superior to R-norm. The first argument holds for any relational approach. The validity of the second one is limited to the Gaussian classifier and, incidentally, to the BIC which is directly derived from this classifier.

8.3.1 First argument

If T is the set of all possible template images, then $\overline{I_t}$ is defined as $T - \{I_t\}$. We now rewrite $P(I_q|\overline{I_t,\mathcal{R}})$ as follows:

$$\begin{aligned} P(I_q|\overline{I_t,\mathcal{R}}) &= \frac{P(I_q,\overline{I_t,\mathcal{R}})}{P(\overline{I_t,\mathcal{R}})} \\ &= \frac{P(I_q,I_t,\overline{\mathcal{R}}) + P(I_q,\overline{I_t},\mathcal{R}) + P(I_q,\overline{I_t},\overline{\mathcal{R}})}{P(\overline{I_t,\mathcal{R}})} \\ &= \frac{P(I_q,I_t,\overline{\mathcal{R}}) + P(I_q,\overline{I_t})}{P(\overline{I_t,\mathcal{R}})} \\ &= P(I_q|I_t,\overline{\mathcal{R}})\frac{P(I_t,\overline{\mathcal{R}})}{P(\overline{I_t,\mathcal{R}})} + P(I_q|\overline{I_t})\frac{P(\overline{I_t})}{P(\overline{I_t,\mathcal{R}})} \end{aligned} \tag{8.13}$$

Hence $P(I_q|\overline{I_t,\mathcal{R}})$ takes into account $P(I_q|I_t,\overline{\mathcal{R}})$, the normalization score of R-norm, and an additional term $P(I_q|\overline{I_t})$. Note that in the special case where $I_q = I_t$, i.e. when no variability is observed between the template and query images, $P(I_q|I_t,\overline{\mathcal{R}})$

is maximum (c.f. equation 8.8) which intuitively is not satisfying as we would like the normalization score to be as small as possible in such a case. One can say, loosely speaking, that the negation on $\mathcal{R}$ impacts the covariance matrix of the Gaussian while the negation on I_t impacts its mean. Thus, the additional term in G-norm prevents this unwanted effect (c.f. equation 8.12). This first argument favors the choice of G-norm over R-norm.

8.3.2 Second argument

Until now, we have always assumed a shared model $\overline{\mathcal{R}}$ of anti-relationship. As $\overline{\mathcal{R}}$ is supposed to model all the possible transformations between face images of different persons, it should be described with a very large number of parameters and, for a robust estimation, these parameters should be estimated with a large amount of training data.

However, when comparing I_t and I_q one does not need to know the whole distribution of the difference between images that belong to two arbitrary persons. Instead, as we have access to the identity of the client C to be verified, we could concentrate on the distribution of the difference between I_t and all the images that do not belong to C. This would require an $\overline{\mathcal{R}_t}$, i.e. a specific anti-relationship model, for each template image I_t. Intuitively, using an $\overline{\mathcal{R}_t}$ should yield a better performance than a $\overline{\mathcal{R}}$ as we would then focus on the region of interest of the distribution.

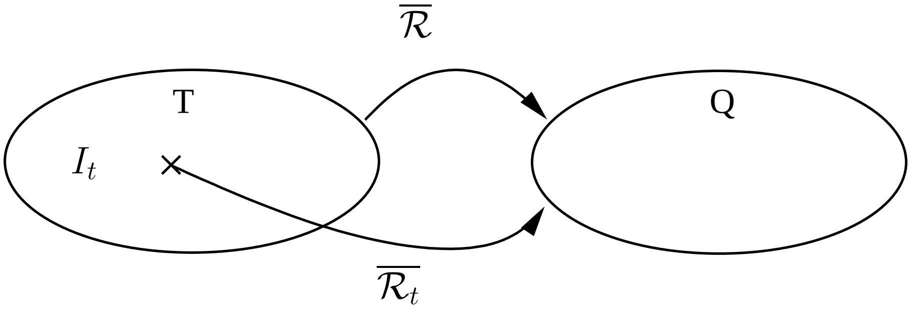


Figure 8.1: Modeling $\overline{\mathcal{R}}$ and $\overline{\mathcal{R}_t}$.

Let $\overline{\mathbf{I}}_{\mathbf{C}}$ denote the random variable which describes the emission of the query images I_q that do not belong to C. Theoretically, there is one such distribution for each client C and its parameters should be estimated with all the available images that do not belong to C. However, in practice, this distribution will be estimated on an independent training set that contains none of the template images of the evaluation set. Note that, even if this training set contained one or a few images from C, their influence would be negligible compared to all the other images. Hence, for all clients, it is reasonable to assume a shared random variable denoted $\bar{\mathbf{I}}$.

If $\bar{\mathbf{I}}$ is assumed normally distributed with mean μ and covariance Σ (assuming that we use the same data to train this distribution as in 8.2.2), then $\bar{\mathbf{I}} - I_t$ is also normally distributed with mean $\mu - I_t$ and covariance Σ. Hence we obtain:

$$\begin{aligned} P(I_q|I_t, \overline{\mathcal{R}}_t) &= \frac{\exp\left\{-\frac{1}{2}\left[(I_q - I_t) - (\mu - I_t)\right]^T \Sigma^{-1} \left[(I_q - I_t) - (\mu - I_t)\right]\right\}}{(2\pi)^{N/2}|\Sigma|^{1/2}} \\ &= \frac{1}{(2\pi)^{N/2}|\Sigma|^{1/2}} \exp\left\{-\frac{1}{2}(I_q - \mu)^T \Sigma^{-1} (I_q - \mu)\right\} \\ &= P(I_q|\overline{I_t, \mathcal{R}}) = P(I_q|U) \qquad (8.14) \end{aligned}$$

This means that, for the Gaussian classifier, it is equivalent to model the impostor distribution U (G-norm) and the relationship between images of different persons (R-norm) in the case where we make use of a template dependent relationship $\overline{\mathcal{R}}_t$. Keeping in mind that $P(I_q|I_t, \overline{R}_t)$ should yield a better performance than $P(I_q|I_t, \overline{\mathcal{R}})$, G-norm should theoretically outperform R-norm, at least for a given complexity of the impostor model (number of features for BIC, number of Gpm for PMLT).

Note that this argument is not exact but only approximate for PMLT which does not work directly on difference images.

8.4 An Experimental Comparison

The focus of this section will not be on the comparison of the BIC and PMLT but on the comparison of the two approaches to modeling wrongful claims. For both BIC and PMLT, we make use of the best client distributions we trained in section 6.6.1. For BIC, this corresponds to a model with 50 features and for PMLT to a model with a maximum of 16 Gpm. We first describe how to train our models of wrongful claims $\overline{R}$ and U. We then carry out two sets of experiments in matched conditions on the FERET database and then in mismatched conditions on the ARDB and PIE databases.

8.4.1 Training $\overline{R}$ and U

$\overline{R}$ and U were trained on the same data used to train $\mathcal{R}$. We remind that it consists of 695 persons, with two images per person extracted from the FAFB set, which makes a total of 1,390 images.

The impostor model for R-norm is trained in a similar manner to the client model, except that we use pairs of images that belong to different persons to model the inter-class variability. However, it is not necessary to consider all possible pairs of images to learn $\lambda_{\overline{R}}$ which would be very computationally intensive. In our experiments, for each of the 1,390 images, we chose randomly an image which corresponded to another person. We also tried more than one image to increase the number of pairs of images but this lead to a very significant increase of the computational cost but had no significant impact on the performance.

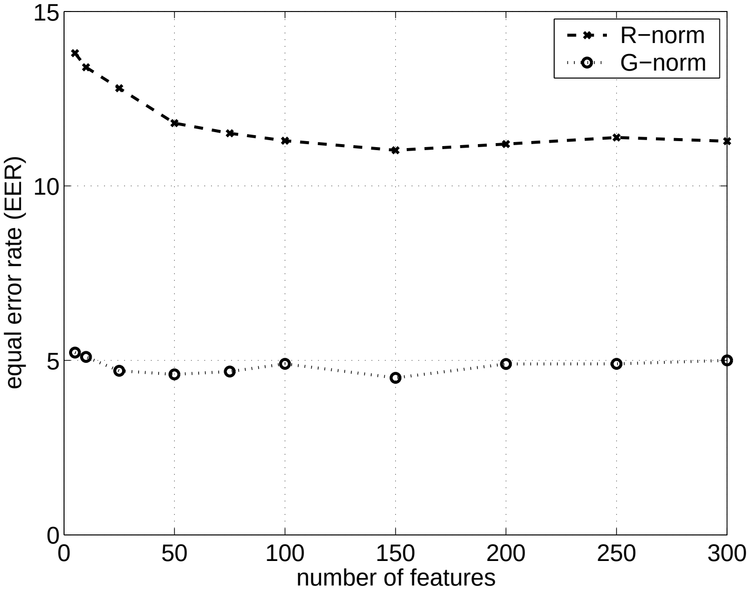
Training the impostor model for G-norm is very simple in the case of BIC as the only difference with the modeling of the client distribution is that we have to estimate the eigenvectors and eigenvalues of the matrix Σ, which is the correlation matrix of all images. Thus, we obtain the PCA eigenvalues and eigenvectors. As for PMLT, it is a little bit more complicated. If we draw a parallel between the simple Gaussian classifier and the PMLT and if we assume a unimodal distribution, then to estimate the impostor distribution, we should learn the transformation between an "average" image and all other images. This "average" image is trained as described in section 7.2.1. Note that, as we just have to train one centroid, there is no problem of data scarcity and we can thus use a simple ML training. Once the centroid is obtained, we learn the transformation between all 1,390 images and this average template. Thus, training the impostor distribution makes use of 1,390 pairs of images for both R-norm and G-norm, as was the case for the client distribution.

8.4.2 Experiments in matched conditions

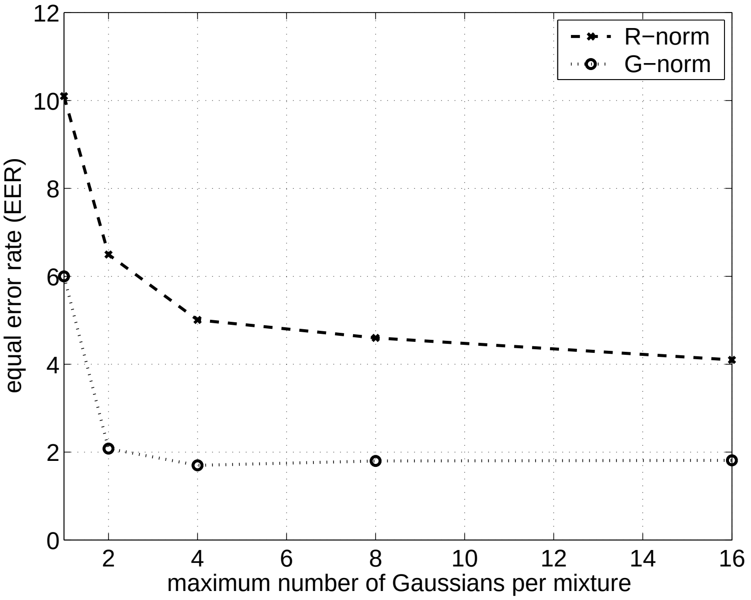
We tested BIC and PMLT on the same data set used to fine tune these systems in sections 5.7.3 and 5.7.4 respectively. Each image is successively used as a query and the remaining 999 images are used as templates. Thus the distribution of client scores was estimated with $1,000$ comparisons and the distribution of impostor scores with $998,000$ comparisons.

In the first set of experiments, we draw the EER as a function of the complexity of the model of wrongful claims: number of features for BIC and number of Gpm for PMLT. The baseline performance of the system without score normalization is 15.5% for BIC and 11.5% for PMLT which is very high considering that we are in matched conditions. This clearly shows that a system which has a very good performance in the identification mode may not achieve a reasonable performance in the verification mode.

Results are presented on Figure 8.2. Both R-norm and G-norm lead to a large decrease of the error rate but G-norm outperforms R-norm significantly. Indeed, for BIC, the best results for R-norm and G-norm are, with a 95% confidence interval, $11.0\% \pm 1.9\%$ and $4.6\% \pm 1.3\%$ respectively. For PMLT, the best results for R-norm



(a)



(b)

Figure 8.2: Performance of (a) BIC and (b) PMLT on a subset of the FERET FAFB set.

and G-norm are $4.1\% \pm 1.2\%$ and $1.8\% \pm 0.8\%$ respectively. Thus, G-norm can be said to outperform R-norm with more than 95% confidence in both cases.

It is very interesting to notice that, for both BIC and PMLT, G-norm reaches its best performance with a smaller number of features or Gpm than R-norm. This is consistent with our analysis in section 8.3.2. Indeed, it is much more efficient to model a specific relationship $\overline{R}_t$ than a global relationship $\overline{R}$ as the second approach results in the modeling of a lot of wasteful information.

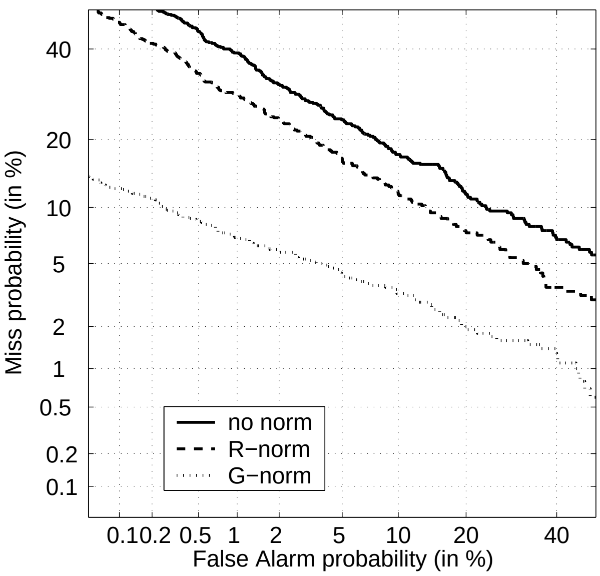
As the EER only represents the performance for a specific threshold θ, we also considered DET curves. On Figure 8.3 (a), we represent the performance of our best BIC systems without normalization, with R-norm and with G-norm. This corresponds to an impostor model with 150 features for R-norm and 50 features for G-norm. For all miss and false alarm probabilities, R-norm improves over the system without score normalization and G-norm improves very significantly over R-norm.

On Figure 8.3 (b), we represent the performance of our best PMLT systems. This corresponds for both R-norm and G-norm to an impostor model with 16 Gpm. For all miss and false alarm probabilities, R-norm and G-norm greatly improve over the system without score normalization and G-norm also greatly improves over R-norm, except for very small miss probabilities. Note however, that there is not enough data to estimate robustly the error rate of our system for very small miss probabilities. To clearly show the extent of the improvement, let us consider the false alarm probability of the three systems at a fix miss probability of 5%. For the system without score normalization, the corresponding false alarm probability is between 20% and 40%, which is useless for most applications of practical value. However, with R-norm it decreases between 2% and 5% and with G-norm it goes down to 0.1% approximately.

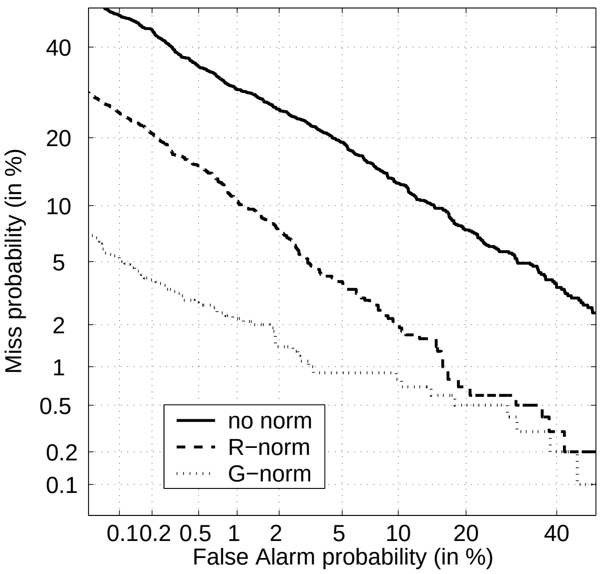
From this first set of experiments we can draw two important conclusions. The first one is that, as expected, G-norm outperforms R-norm. The second one is that, even in perfectly matched conditions, score normalization has a very significant impact on the performance. Indeed for BIC the EER, which was 15.5% without score normalization could be brought down to 4.6% and for PMLT from 11.5% to 1.8%. The reason for this much improved performance is that, in matched conditions the normalization score $P(I_q|U)$ indicates whether the neighborhood of I_q is densely populated with potential impostors.

8.4.3 Experiments in mismatched conditions

In this section, we present results in mismatched conditions on the AR and PIE databases. For both BIC and PMLT and for the system without score normalization,



(a)



(b)

Figure 8.3: DET curves for (a) BIC and (b) PMLT on a subset of the FERET FAFB set for the system without impostor modeling, with R-norm and with G-norm.

with R-norm impostor modeling and with G-norm impostor modeling, we used the best systems that were trained in the previous section.

Experiments on AR

In the following experiments, image 01 which corresponds to the face with neutral expression was used for the enrollment. Test images consisted of 7 subsets: 02, 03 and 04 which correspond to the three facial expressions, 05, 06 and 07 which correspond to the three illumination conditions and 11 which corresponds to an occlusion of the lower part of the face with a scarf. Note that, compared to the identification experiments, we discarded set 08 which corresponds to an occlusion of the upper part of the face with sunglasses. The reason for discarding this set is the inability of BIC and PMLT to deal with it and thus, including this set would unfavorably bias our results. The distributions of client and impostor scores are estimated respectively with 868 and $106,764$ comparisons.

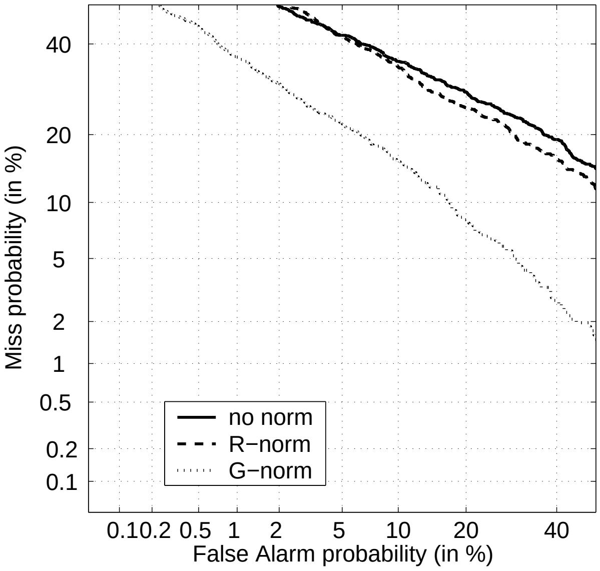
Results are presented on Figure 8.4. For both BIC and PMLT, R-norm has a smaller impact on the performance than was previously the case in matched conditions. It seems that it can even result in a degradation of the performance for a small false alarm probability. G-norm results in a large improvement of the performance, especially for BIC. The EER for BIC and PMLT for the three systems is summarized in Table 8.4. Thus G-norm can be said to outperform R-norm as expected.

	BIC	PMLT
no norm	$25.5\% \pm 2.9\%$	$22.7\% \pm 2.8\%$
R-norm	$23.4\% \pm 2.8\%$	$19.7\% \pm 2.6\%$
G-norm	$12.8\% \pm 2.2\%$	$13.1\% \pm 2.2\%$

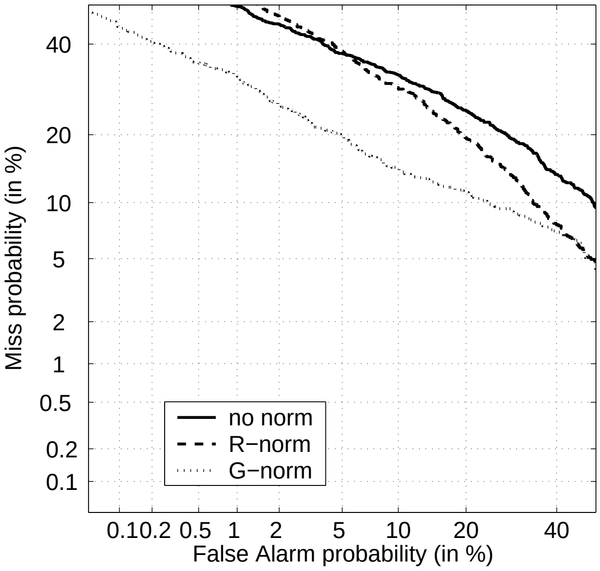
Table 8.1: EER and its associated 95% confidence interval for BIC and PMLT on the AR database for the system without impostor modeling, with R-norm and with G-norm

Experiments on PIE

In the following experiments, we used the image corresponding to an ambient lighting as enrollment image and the images corresponding to variations in pose and in illumination with ambient lighting as query images. Thus, compared to identification experiments, we discarded those images which correspond to variations in illumination without ambient lighting. The distributions of client and impostor scores were estimated respectively with $1,833$ and $122,811$ scores respectively.

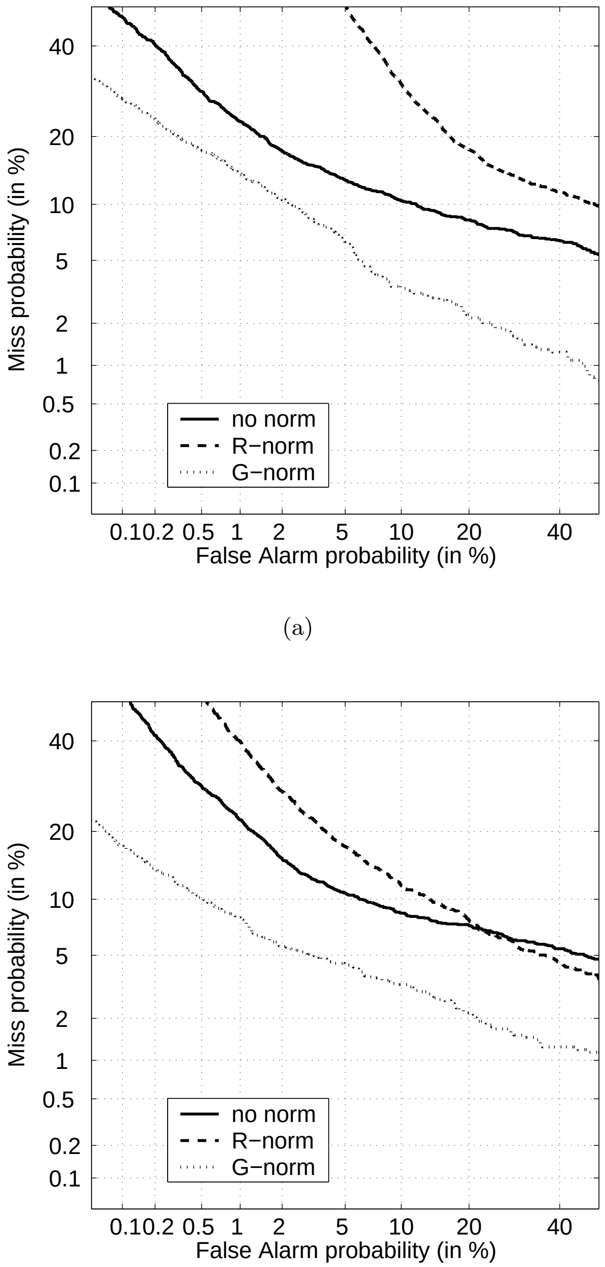


(a)



(b)

Figure 8.4: DET curves for (a) BIC and (b) PMLT on the AR database for the system without modeling, with R-norm and with G-norm



(a)

(b)

Figure 8.5: DET curves for (a) BIC and (b) PMLT on the PIE database for the system without impostor modeling, with R-norm and with G-norm.

Results are presented on Figure 8.5. We first notice that for both BIC and PMLT, R-norm results in a degraded performance compared to the simple system without impostor modeling especially for BIC. This seems to indicate that, when facing radically new conditions, R-norm is unable to distinguish between inter- and intra-class variabilities. Indeed it can so happen that this new condition will have a greater impact on $P(I_q|I_t, \mathcal{R})$ than on $P(I_q|I_t, \overline{\mathcal{R}})$. On the other hand, G-norm results in a large decrease of the error rate for all miss and false alarm probabilities. The EER for BIC and PMLT for the three systems is summarized in Table 8.5. Thus, we can see that the error rate of the system is approximately divided by a factor 2 with G-norm.

	BIC	PMLT
no norm	$10.3\% \pm 1.4\%$	$8.9\% \pm 1.3\%$
R-norm	$18.5\% \pm 1.8\%$	$11.1\% \pm 1.4\%$
G-norm	$5.6\% \pm 1.1\%$	$4.5\% \pm 0.9\%$

Table 8.2: EER and its associated 95% confidence interval for BIC and PMLT on the PIE database for the system without score impostor modeling, with R-norm and with G-norm

Finally, we present a last experiment on PIE where we attempt to obtain the best possible performance by training the client and impostor distributions with data which exhibits variations in facial expression and pose. We thus added to the training data the images that correspond to the BE, BF, BD, BG, BC and BH sets of FERET. Results are presented on Figure 8.6 for the approach without score normalization and with G-norm only. With a 95% confidence the performance of the system without score normalization decreases to $6.9\% \pm 1.2\%$ and the EER of the system with G-norm goes down to $3.1\% \pm 0.8\%$. Note that this later result is very competitive considering that the system has to deal with both pose and illumination variations.

8.5 Conclusion

In this chapter, we considered two strategies to score normalization for those approaches to AFR that attempt to model the relationship between images such as BIC and the proposed PMLT. The first strategy, which is specific to relational approaches and which consists in modeling the relationship between face images of different persons is a direct extension of the work of [MWP98, Mog02]. The second one, which consists in modeling the impostor distribution, is very general and can be applied to any face authentication system. These two techniques were first com-

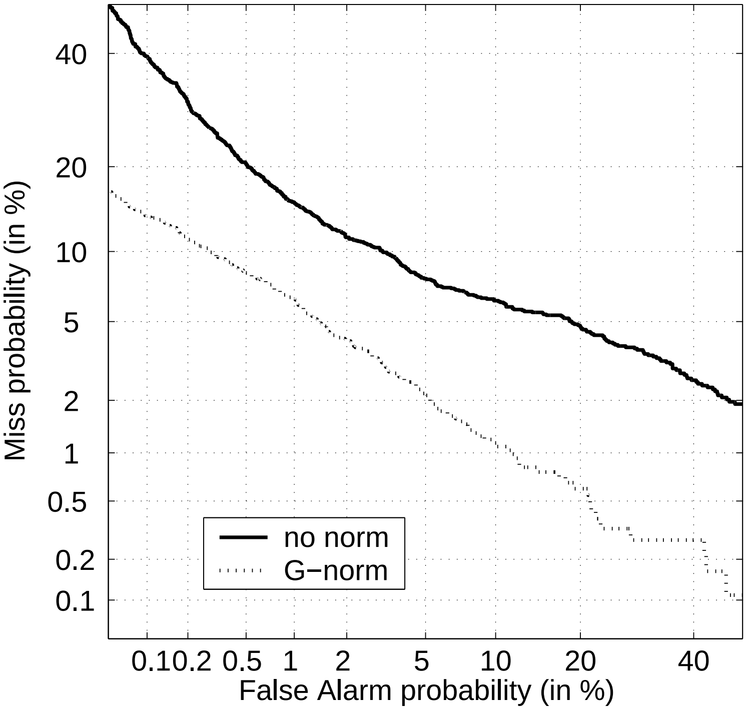


Figure 8.6: DET curves for PMLT on the PIE database for the system without impostor modeling and with G-norm. Case where $\mathcal{R}$ and U are trained with data that exhibits some pose variability.

pared from a theoretical and then from an experimental point of view on both BIC and PMLT. Both comparisons indicated that the general approach to score normalization results in a better performance, especially in the challenging but realistic case where there is a mismatch between the training and test conditions.

However, there is still a lot of room for improving our verification system. We especially expect to decrease the FAR and FRR using score normalization approaches, which were shown to lead to large improvements of the performance in the field of automatic speaker recognition [RQD00, ACLT00].

Also this work on verification may be applied to the problem of *image quality checking* (c.f. paragraph 2.4). Indeed, the normalization score $P(I|U)$ is a measure of distance between the image I and all the images that were used to train U. If $P(I|U)$ is low, this means that I is very different from the training images and that our system might fail on such an image. This may be due for instance to a mismatch between the training and test conditions, e.g. because the client provides a profile view while the system can handle only an in-depth rotation of $\pm 20^o$. If we rejected at enrollment or test time the images I such that $P(I|U) < \theta$, where θ is a predefined threshold, then we could increase the performance of our system.

9

Conclusion

9.1 Summary and Contributions

After introducing the discipline of biometrics in chapter 2, we briefly reviewed the literature on AFR in chapter 3. It was shown that many state-of-the-art AFR algorithms focus on the problem of representation, i.e. feature extraction, but that less attention has been given to the proper derivation and computation of an appropriate distance between face images. The BIC algorithm introduced by Moghaddam and Pentland [MP97] is a noticeable exception. Considering its excellent performance during the FERET evaluation, we chose it as the baseline for our experiments.

The main contribution of this dissertation is the introduction of a novel measure of "distance" between images. This measure involves the estimation of the *set of possible transformations between face images*. The global transformation, which is assumed too complex for direct modeling, is approximated with a set of local transformations under a constraint imposing consistency between neighboring local transformations. The proposed local transformations and neighboring constraints are embedded within the probabilistic framework of a 2-D HMM in the case of discrete states and of the 2-D SSM in the case of continuous states. This original approach was coined PMLT for probabilistic mapping with local transformations.

In chapter 4, we introduced the turbo hidden Markov model (T-HMM) and the turbo state-space model (T-SSM) as efficient approximations of the intractable 2-D

HMM and 2-D SSM respectively. They consist of a set of inter-connected horizontal and vertical 1-D Markov chains that communicate through an iterative process. We attempted to provide efficient approximate answers to the three fundamental problems of HMM design. We proved some convergence properties for the T-SSM and observed others for both the T-HMM and the T-SSM. The potential of the T-HMM and the T-SSM was demonstrated on simple problems which consisted in decoding a signal embedded in noise. While the work on the T-HMM and the T-SSM was not the focus of this dissertation, it was necessary to make the face recognition algorithms developed in the course of this thesis tractable.

In chapter 5, we specialized our framework to the problem of face identification in the case of elastic facial distortions (due, for instance, to facial expressions) using discrete grid transformations. We described the components of our HMM-based transformation model and explained how to perform the matching with this model and how to train it. The performance of this probabilistic mapping with local grid transformations (PMLGT) was evaluated on a large dataset including 4 databases (FERET, AR, PIE and Yale B). It was shown that for a face identification task PMLGT outperforms BIC for an imprecise segmentation of the face, for variations in facial expression or pose and for an occlusion of the lower part of the face. However, the results for illumination variations were not as clear.

Therefore, in chapter 6 we enriched our HMM-based model with continuous feature transformations to model illumination variations. We detailed our novel HMM-based transformation model and explained how to deal with both discrete and continuous states. Two sets of evaluations were carried out. We first measured the effect of the log transform on the identification rate and it was shown that it had a very positive impact on the performance for illumination variations but relatively little influence on other variabilities. We then evaluated our probabilistic mapping with local grid and feature transformations (PMLGFT) and it was shown that an additional gain in performance could be obtained for extreme illumination conditions, such as a face illuminated by a flash without ambient lighting.

Once a proper measure of distance had been defined, we turned in chapter 7 to the problem of face image clustering. The primary motivation was to partition the face space to reduce the number of comparisons when a query is made on a database that contains a large number of templates. We addressed two main issues in this chapter. We first considered the problem of the update step, which is obvious for simple metrics such as the Euclidean distance, but which is much more challenging in the case of a complex measure of distance such as the one induced by our PMLT. We then addressed the problem of multiple clusters assignment of an image. It was shown on the FERET database that we could divide the total number of comparisons by a factor 6 or 7 with little degradation of the performance.

Finally, in chapter 8 we focused on the verification problem which requires a robust confidence measure. Note that this is a topic which has been much studied in the field of automatic speaker recognition but which has received little attention in AFR. The issue is the accurate modeling of wrongful claims. For BIC or PMLT there exists two very distinct ways to model such claims. The first method raises the following question: "Is the observed difference between the template and query images more likely to be intra-personal or extra-personal?". The second approach attempts to answer this question: "Is the query image more likely to have been generated by the person under consideration or by one of its potential impostors?". Although semantically very close, these two questions lead in effect to two very different classifiers. These two approaches were compared from a theoretical and an experimental point of view, both in matched and mismatched conditions, and it was shown that the latter approach significantly outperformed the former one.

This work resulted in the following refereed conference and journal articles:

1. J.-L. Dugelay, J.-C. Junqua, C. Kotropoulos, R. Kuhn, F. Perronnin and I. Pitas, *Recent Advances in Biometric Person Authentication*, in Proc. of the IEEE Int. Conf. on Acoustics Speech and Signal Processing (ICASSP), 2002, vol. 4, pp. 4060-4063.

2. F. Perronnin and J.-L. Dugelay, *Introduction à la Biométrie*, in Revue Traitement du Signal, 2002, vol. 19, no. 4, pp. 253-265.

3. F. Perronnin and J.-L. Dugelay, *An Introduction to Biometrics and Face Recognition*, in Proc. of the workshop on IMAGE: Learning, Understanding, Information Retrieval, Medical, 2003.

4. F. Perronnin, J.-L. Dugelay and K. Rose, *Iterative Decoding of Two-Dimensional Hidden Markov Models*, in Proc. of the IEEE Int. Conf. on Acoustics Speech and Signal Processing (ICASSP), 2003, vol. 3, pp. 329–332.

5. F. Perronnin, J.-L. Dugelay and K. Rose, *Deformable Face Mapping for Person Identification*, in Proc. of the IEEE Int. Conf. on Image Processing (ICIP), 2003, vol. 1, pp. 661–664.

6. F. Perronnin and J.-L. Dugelay, *Discriminative Face Recognition*, in Proc. of the IAPR Int. Conf. on Audio- and Video-Based Person Authentication (AVBPA), 2003, pp. 446–453.

7. F. Perronnin, J.-L. Dugelay and K. Rose, *A Probabilistic Model of Face Transformation Applied to Person Identification*, EURASIP Journal on Applied Signal Processing (JASP), 2004, vol. 2004, no. 4, pp. 510–521.

8. F. Perronnin and J.-L. Dugelay, *A Model of Illumination Variation for Robust Face Recognition*, in Proc. of the workshop on Multimodal User Authentication (MMUA), 2003, pp. 157–164.

9. F. Perronnin and J.-L. Dugelay, *From Turbo Hidden Markov Models to Turbo State-Space Models*, in Proc. of the IEEE Int. Conf. on Acoustics Speech and Signal Processing (ICASSP), 2004, vol. 3, pp. 29–32.

10. F. Perronnin and J.-L. Dugelay, *Robust Score Normalization for Relational Approaches to Face Authentication*, Proc. of the EURASIP European Signal Processing Conf. (EUSIPCO), 2004.

11. F. Perronnin and J.-L. Dugelay, *Un Modèle Probabiliste de Transformation entre Images Appliqué à la Reconnaissance de Visages*, Proc. of Compression et Représentation des Signaux Audiovisuels (CORESA), 2004.

12. F. Perronnin, J.-L. Dugelay and K. Rose, *A Probabilistic Model of Face Mapping with Local Transformations and its Application to Person Recognition*, submitted to the IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI).

13. F. Perronnin and J.-L. Dugelay, *Clustering Face Images with Application to Image Retrieval in Large Databases*, submitted to the SPIE Conf. on Biometric Technology for Human Identification.

9.2 Future Work

At the end of chapters 4 to 8, we have tried to outline possibilities for future work. We now envision other research directions.

One could first try to improve on the proposed measure of distance. We have attempted to explicitly model facial expressions and illumination variations and have observed that PMLT was relatively robust to other variabilities such as pose or occlusion of the lower part of the face. However we could try to model explicitly such variabilities. Let us take the example of the pose. We could decide to model such variation using grid transformations as was the case for facial expressions. For instance, we tried to add FERET images which exhibit some pose variation (sets BC, BD, BE, BF, BG and BH) to the training data. Although this may not be the optimal way to proceed, we increased the average identification rate on the pose set of PIE from 81% (c.f. 6.6.1) to 91%. Note that we could also decide to introduce new local transformations to model pose variability.

Another approach to improve the accuracy of our measure of similarity would be to make use of richer statistical tools than the T-HMM or the T-SSM. Indeed, the choice of the T-HMM and the T-SSM, which are approximations of the simple first order MMRF with discrete and continuous states respectively, is primarily motivated by their very low computational complexity. However, we believe that an improved performance could be obtained if we went beyond the first order statistics or by dropping the causality assumption. Obviously, the improved accuracy would be obtained at the expense of an increase of the computational cost.

It would be also of interest to study whether PMLT could work on other face modalities such as infrared imagery or range (3-D) data.

We believe that PMLT could also be applied to other problems within the field of facial analysis. Especially, the current framework may be suited to the problem of facial expressions recognition [FL03]. To sustain this claim, we carried out preliminary experiments on the AR database which contains 4 facial expressions: neutral, smile, anger and scream (c.f. Figure D.4). 131 images from 74 persons were used to train a model for each of the 4 expressions. These models were trained in a similar manner to the UBM in 8.4.1 up to a maximum of 4 Gpm. No attempt was made to tune feature extraction or training parameters. Recognition was performed on a set of 400 images from 50 persons, all different from the training ones. We obtained a recognition rate of approximately 87%. Although this is far from perfect, these results are encouraging and we believe that an improved performance could be obtained by tuning parameters and by adding more training data.

Finally, we believe that PMLT could be applied to the retrieval of other types of images. For instance, within the field of biometrics, we believe that the same framework could be extended to the problem of automatic fingerprint recognition [MMJP03]. Indeed, the fingerprint acquisition introduces deformations of the fingerprint image that might change from one acquisition to another as they depend on the exact contact point but also on the pressure of the finger on the sensing device. Such elastic deformations may be modeled for instance with grid transformations.

A

Distance Measures

Let $x = [x_1, ..., x_D]^T$ and $y = [y_1, ..., y_D]^T$ be two D-dimensional vectors. The L_1, L_2 and cosine distances are defined as follows:

- L_1 (city-block):

$$d_{L_1}(x, y) = \sum_{d=1}^{D} |x_i - y_i| \tag{A.1}$$

- L_2 (Euclidean squared):

$$d_{L_2}(x, y) = ||x - y||^2 = \sum_{d=1}^{D} (x_i - y_i)^2 \tag{A.2}$$

- cosine:

$$d_{cos}(x, y) = 1 - \frac{x^T y}{||x|| ||y||} = 1 - \frac{\sum_{d=1}^{D} x_i y_i}{\sqrt{\sum_{d=1}^{D} x_i^2 \sum_{d=1}^{D} y_i^2}} \tag{A.3}$$

We now consider distances which are specific to eigenfaces. Let $\lambda_i = \sigma_i^2$ be the i-th eigenvalue corresponding to the i-th eigenvector and let u and v be the vectors defined by $u_i = x_i/\sigma_i$ and $v_i = y_i/\sigma_i$. Then the Mahalanobis-L_1, -L_2 and -cosine distances are defined as follows [BBTD03]:

- Mahalanobis-L_1:

$$d_{MahL_1}(x, y) = d_{L_1}(u, v) \tag{A.4}$$

- Mahalanobis-L_2:
$$d_{MahL_2}(x,y) = d_{L_2}(u,v) \tag{A.5}$$

- Mahalanobis-cosine:
$$d_{Mahcos}(x,y) = d_{cos}(u,v) \tag{A.6}$$

The "Moon" and "Yambor" distances, are defined as follows:

- Moon:
$$d_{Moon}(x,y) = \sum_{d=1}^{D} \sqrt{\frac{\lambda_i}{\lambda_i + \alpha^2}} x_i y_i \tag{A.7}$$
where α is a constant.

- Yambor:
$$d_{Yam}(x,y) = \sum_{d=1}^{D} \frac{1}{\sqrt{\lambda_i}} x_i y_i \tag{A.8}$$

We now consider a distance which has been specifically designed for Fisherfaces. If λ_i is the i-th eigenvalue corresponding to the i-th eigenvector, then the following soft distance was suggested [Zha99]:

- soft Fisherfaces distance:
$$d_{soft}(x,y) = \sum_{d=1}^{D} \lambda_i^{\alpha} (x_i - y_i)^2 \ , \ \alpha \in [0,1] \tag{A.9}$$

B

Entropy and Divergence

In this appendix, we briefly review the definition of the entropy, the relative entropy and the symmetric divergence. We then provide closed form solutions for these three quantities in the case of Gaussian distributions. Finally, we review a very simple approach to fuse two probability distributions which makes sense from an information theoretical point of view.

The *entropy* of a discrete probability distribution $p = \{p_i\}$ is defined as [CT93]:

$$H(p) = -\sum_i p_i \log p_i \tag{B.1}$$

If $q = \{q_i\}$ is also a discrete probability distribution, then the *relative entropy* also referred to as *Kullback-Leibler distance* or *divergence* is defined as:

$$D(p||q) = \sum_i p_i \log \frac{p_i}{q_i} \tag{B.2}$$

For a discrete random variable, the relative entropy is always non-negative and is equal to zero if and only if $p = q$. As $D(p||q) \neq D(q||p)$ we introduce the symmetric divergence:

$$D(p, q) = D(p||q) + D(q||p) = \sum_i (p_i - q_i) \log \frac{p_i}{q_i} \tag{B.3}$$

Note that these formulas can be extended to the continuous case, simply by replacing sums with integrals. Note however that the properties of the entropy and relative entropy are significantly different in the continuous case.

The Gaussian distribution is of particular interest to us. If p and q are Gaussian with means μ_p and μ_q respectively and variances σ_p^2 and σ_q^2 respectively then we have [Per00]:

$$\begin{aligned} H(p) &= \frac{1}{2}\left[1 + \log(2\pi\sigma_p^2)\right] && \text{(B.4)} \\ D(p||q) &= \frac{1}{2}\left[\frac{\sigma_p^2}{\sigma_q^2} - 1 + \log\left(\frac{\sigma_q^2}{\sigma_p^2}\right) + \frac{(\mu_p - \mu_q)^2}{\sigma_q^2}\right] && \text{(B.5)} \\ D(p,q) &= \frac{1}{2}\left[\frac{\sigma_p^2}{\sigma_q^2} + \frac{\sigma_q^2}{\sigma_p^2} - 2 + \left(\frac{1}{\sigma_p^2} + \frac{1}{\sigma_q^2}\right)(\mu_p - \mu_q)^2\right] && \text{(B.6)} \end{aligned}$$

Now we will consider the following problem. Given two discrete probability distributions p_i and q_i, how to “fuse” them into a unique distribution r_i. A possible solution is to choose r_i that minimizes $D(r_i||p_i) + D(r_i||q_i)$. Adding the Lagrange multiplier η and using the constraint $\sum_i r_i = 1$, we have to minimize the following quantity:

$$\sum_i r_i \log\frac{r_i}{p_i} + \sum_i r_i \log\frac{r_i}{q_i} + \eta(1 - \sum_i r_i) \tag{B.7}$$

Taking the partial derivative with respect to r_i and equating to zero we obtain:

$$\log r_i = \frac{1}{2}\log(p_i q_i) + \frac{\eta}{2} - 1 \tag{B.8}$$

Now summing over i we obtain:

$$r_i = \frac{\sqrt{p_i q_i}}{\sum_i \sqrt{p_i q_i}} \tag{B.9}$$

C

Convergence of the T-SSM

We will first show that ${\sigma_{i,j}^{\gamma\mathcal{H}}}^2$ and ${\sigma_{i,j}^{\gamma\mathcal{V}}}^2$ converge toward zero as the number of iterations increases and that ${\sigma_{i,j}^{\gamma\mathcal{H}}}^2/{\sigma_{i,j}^{\gamma\mathcal{V}}}^2$ converges to one. We will then show that $\mu_{i,j}^{\gamma\mathcal{H}} - \mu_{i,j}^{\gamma\mathcal{V}}$ converges toward zero. In the following derivations, we will use the notation $u[n]$ to denote the n-th value of a sequence u.

If we start with a vertical pass, we have the following set of inequalities:

- from equation 4.43

$$ {\sigma_{i,j}^{\gamma\mathcal{V}}}^2[n] \leq {\sigma_{i,j}^{\alpha\mathcal{V}}}^2[n] \tag{C.1} $$

- from equation 4.30 and 4.33:

$$ {\sigma_{i,j}^{\alpha\mathcal{V}}}^2[n] \leq {\sigma_{i,j}^{b\mathcal{H}}}^2[n-1] \tag{C.2} $$

- from equation 4.27

$$ {\sigma_{i,j}^{b\mathcal{H}}}^2[n-1] \leq {\sigma_{i,j}^{\gamma\mathcal{H}}}^2[n-1] \tag{C.3} $$

- from equation 4.43 (translated to horizontal quantities):

$$ {\sigma_{i,j}^{\gamma\mathcal{H}}}^2[n-1] \leq {\sigma_{i,j}^{\alpha\mathcal{H}}}^2[n-1] \tag{C.4} $$

- from equation 4.30 and 4.33 (translated to horizontal quantities):

$$ {\sigma_{i,j}^{\alpha\mathcal{H}}}^2[n-1] \leq {\sigma_{i,j}^{b\mathcal{V}}}^2[n-1] \tag{C.5} $$

- from equation 4.27 (translated to horizontal quantities):

$$\sigma_{i,j}^{b\mathcal{V}^2}[n-1] \leq \sigma_{i,j}^{\gamma\mathcal{V}^2}[n-1] \tag{C.6}$$

and thus $\sigma_{i,j}^{\gamma\mathcal{V}^2}$ decreases when the number of iterations n increases. Note that the same can be said about $\sigma_{i,j}^{\gamma\mathcal{H}^2}$, $\sigma_{i,j}^{\alpha\mathcal{V}^2}$, $\sigma_{i,j}^{\alpha\mathcal{H}^2}$, $\sigma_{i,j}^{b\mathcal{V}^2}$ and $\sigma_{i,j}^{b\mathcal{H}^2}$. All these sequences are decreasing and have a lower bound since they are positive. Thus they converge toward a finite value. Moreover from the previous set of inequalities we deduce that all these sequences converge toward the same limit $\lambda_{i,j}$. Now, using equation 4.27 we obtain:

$$\lambda_{i,j} = \frac{\lambda_{i,j}\sigma_{i,j}^2}{{f_{i,j}}^2\lambda_{i,j} + \sigma_{i,j}^2} \tag{C.7}$$

which implies that $\lambda_{i,j}^2 f_{i,j}^2 = 0$. If we exclude the pathological case where $f_{i,j}^2 = 0$, this is equivalent to $\lambda_{i,j} = 0$. We conclude that $\sigma_{i,j}^{\gamma\mathcal{H}^2}$ and $\sigma_{i,j}^{\gamma\mathcal{V}^2}$ converge toward 0. Moreover, using one more time equations 4.43, 4.30, 4.33 and 4.27, we obtain respectively: $\sigma_{i,j}^{\gamma\mathcal{H}^2}[n]/\sigma_{i,j}^{\alpha\mathcal{H}^2}[n] \rightarrow 1$, $\sigma_{i,j}^{\alpha\mathcal{H}^2}[n]/\sigma_{i,j}^{b\mathcal{V}^2}[n] \rightarrow 1$, and $\sigma_{i,j}^{b\mathcal{V}^2}[n]/\sigma_{i,j}^{\gamma\mathcal{V}^2}[n] \rightarrow 1$. Thus we conclude that $\sigma_{i,j}^{\gamma\mathcal{H}^2}/\sigma_{i,j}^{\gamma\mathcal{V}^2}$ converges to one.

To show that $\mu_{i,j}^{\gamma\mathcal{H}} - \mu_{i,j}^{\gamma\mathcal{V}}$ converges toward zero, we rewrite this difference:

$$\mu_{i,j}^{\gamma\mathcal{H}}[n] - \mu_{i,j}^{\gamma\mathcal{V}}[n] = (\mu_{i,j}^{\gamma\mathcal{H}}[n] - \mu_{i,j}^{\alpha\mathcal{H}}[n]) + (\mu_{i,j}^{\alpha\mathcal{H}}[n] - \mu_{i,j}^{b\mathcal{V}}[n]) + (\mu_{i,j}^{b\mathcal{V}}[n] - \mu_{i,j}^{\gamma\mathcal{V}}[n]) \tag{C.8}$$

We obtain the following set of limits:

- from equation 4.42 (translated to horizontal quantities) and knowing that $\sigma_{i,j}^{\alpha\mathcal{H}^2}[n] \rightarrow 0$:
$$\mu_{i,j}^{\gamma\mathcal{H}}[n] - \mu_{i,j}^{\alpha\mathcal{H}}[n] \rightarrow 0 \tag{C.9}$$

- from equations 4.29 and 4.32 (translated to horizontal quantities) and knowing that $\sigma_{i,j}^{b\mathcal{V}^2}[n] \rightarrow 0$:
$$\mu_{i,j}^{b\mathcal{H}}[n] - \mu_{i,j}^{\alpha\mathcal{V}}[n] \rightarrow 0 \tag{C.10}$$

- from equation 4.26 (translated to horizontal quantities) and knowing that $\sigma_{i,j}^{\gamma\mathcal{V}^2}[n] \rightarrow 0$:
$$\mu_{i,j}^{b\mathcal{H}}[n] - \mu_{i,j}^{\alpha\mathcal{V}}[n] \rightarrow 0 \tag{C.11}$$

and thus $\mu_{i,j}^{\gamma\mathcal{H}} - \mu_{i,j}^{\gamma\mathcal{V}}$ converges toward zero.

It should be underlined that the convergence properties that we have just proved are independent of the initialization of the horizontal and vertical priors.

D

Face Databases

D.1 FERET

The Facial Recognition Technology (FERET) database [PMRR00] contains over 14,000 images of size 256 x 384 taken from 1,199 individuals. For each individual, two frontal views were taken (FA and FB images) and a different facial expression was requested for the second frontal image. For 200 individuals, a third frontal image was taken with a different camera and different lighting (FC images) and a set of images was collected at various aspects ranging from right lo left profile. For some individuals, a second set of images was taken on a later date (duplicate sets).

D.2 Yale B

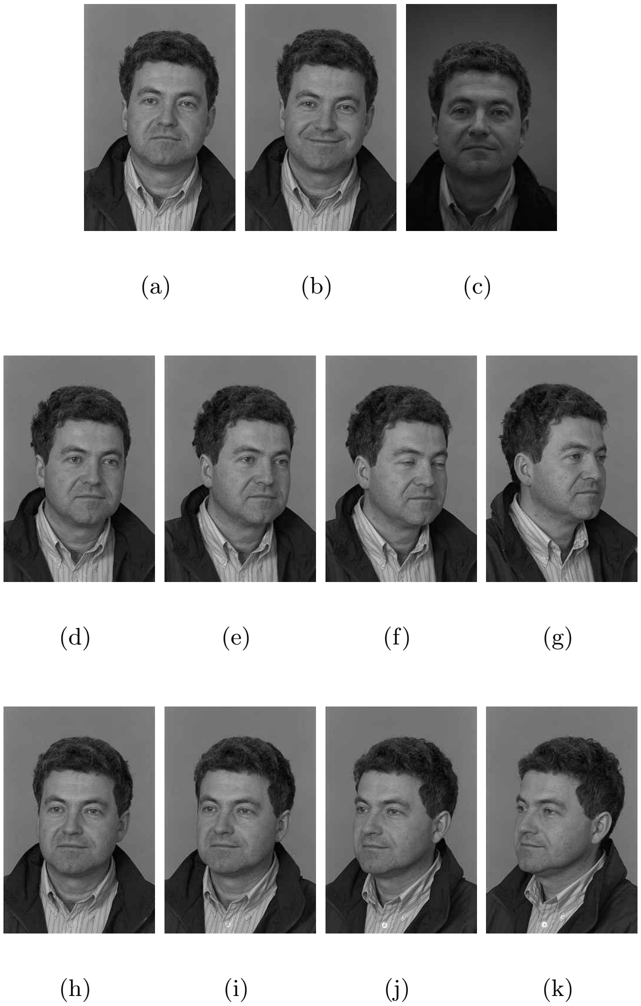
The Yale face database B (Yale B) [GBK01] contains 5,850 images of size 640 x 480 taken from 10 subjects. Some variation across pose was obtained by taking pictures simultaneously with 9 cameras. Pose 0 is frontal, poses 1, 2 , 3 , 4 and 5 are about 12 degrees from the optical axis while poses 6, 7 and 8 are about 24 degrees. To get wide illumination variations, the database was captured using a purpose-built illumination rig with 64 strobes. The 64 images of a subject in a particular pose were acquired in about 2 seconds, so there is only small change in head pose and facial expression for those 64 images. An additional set of images was captured with no strobe going off (ambient lighting).

D.3 PIE

The CMU Pose Illumination Expression (PIE) database [SBB02] contains over 40,000 images of size 640 x 486 taken from 68 individuals. To obtain large variations across pose, a set of 13 cameras was used. To obtain significant illumination variations, a flash system similar to the one constructed at the Yale university was used. The flash system consisted of 21 flashes. Since images were captured with and without background lighting and since one picture was taken with ambient lighting, $21 \times 2 + 1 = 43$ different illumination conditions were obtained.

D.4 AR

The Alex Martìnez-Robert Benavente (AR) face database [AR] contains over 4,000 images of size 768 x 576 taken from 126 subjects. Images feature frontal view faces with different facial expressions (neutral, smile, anger, scream), illumination conditions (left light on, right light on, both lights on) and occlusions (wearing sun glasses, wearing a scarf). Each person participated in two sessions separated by two weeks and the same set of pictures were taken in both sessions.

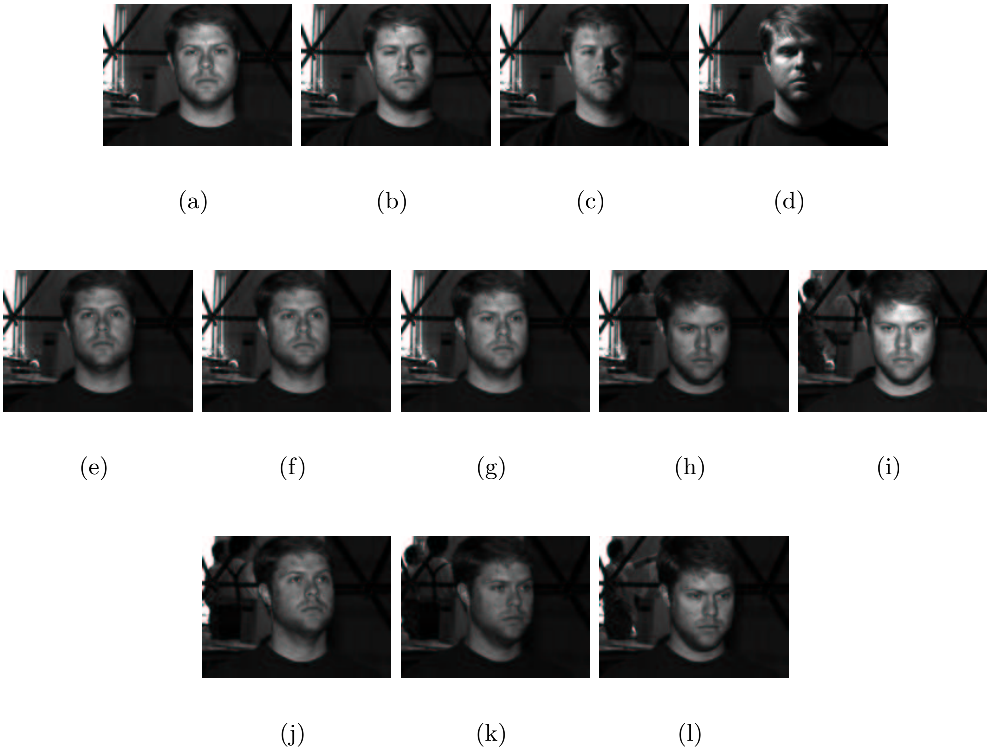


(a) (b) (c)

(d) (e) (f) (g)

(h) (i) (j) (k)

Figure D.1: Sample images of the FERET face database: (a) Frontal image, (b) alternative expression, (c) different camera and lighting, (d)-(g) rotation of the head to the left of 15^o, 25^o, 40^o and 60^o respectively, (h)-(k) rotation of the head to the right of 15^o, 25^o, 40^o and 60^o respectively.

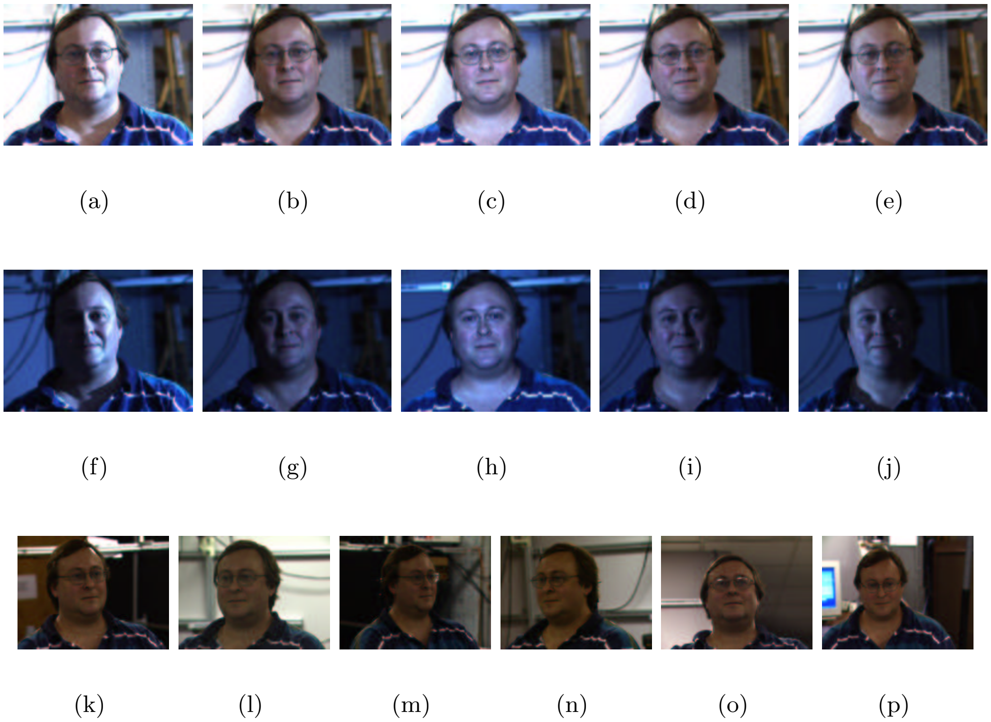


(a) (b) (c) (d)

(e) (f) (g) (h) (i)

(j) (k) (l)

Figure D.2: Sample images of the YALE B face database: (a)-(d) examples of illumination variations for the frontal pose, (e)-(l) examples of pose variations for the frontal illumination.

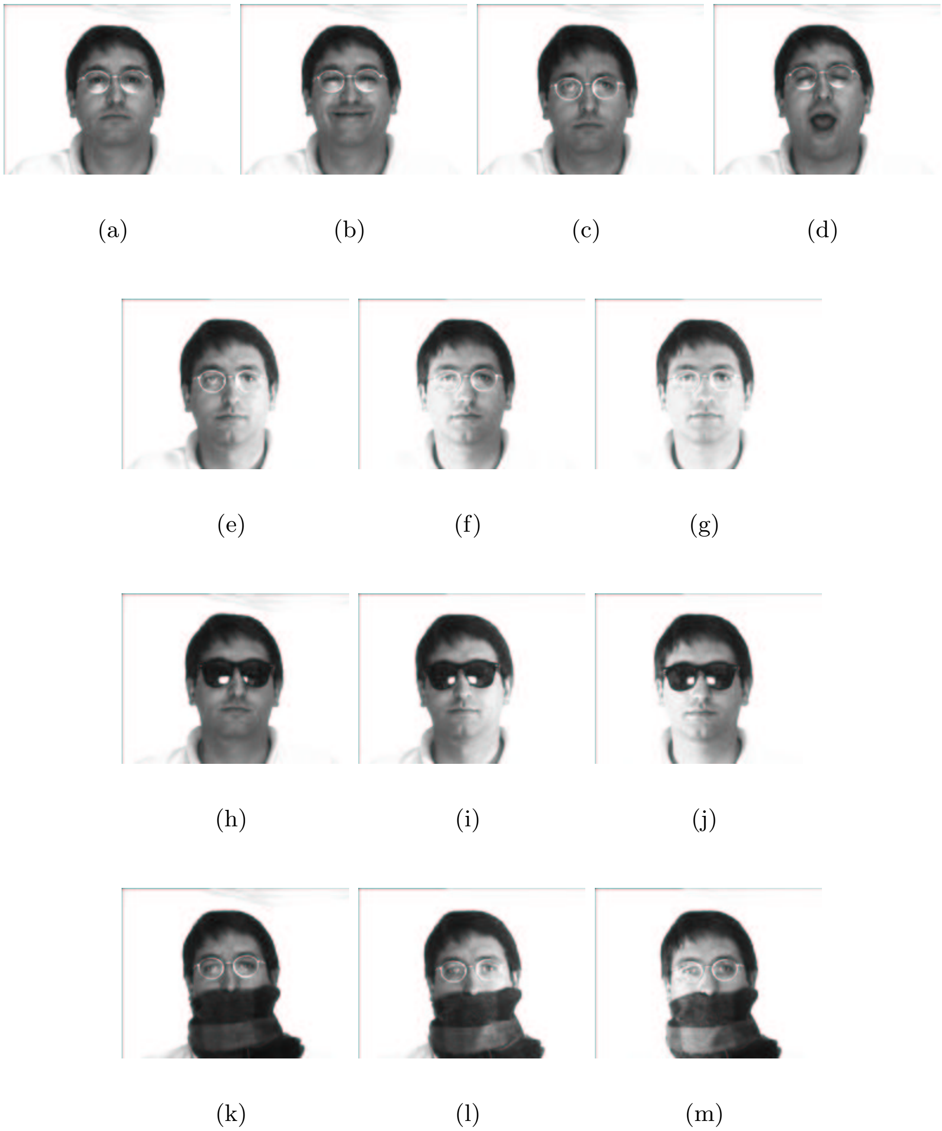


(a) (b) (c) (d) (e)

(f) (g) (h) (i) (j)

(k) (l) (m) (n) (o) (p)

Figure D.3: Sample images of the PIE database: (a)-(e) different illumination conditions with ambient lighting, (f)-(j) different illumination conditions without ambient lighting and (k)-(p) different poses: (k) and (l) (cameras 05 and 29 respectively) rotation of the head to the left or right of 22^o approximately, (m) and (n) (cameras 37 and 11 respectively) rotation of the head to the left or right of 45^o approximately, (o) and (p) (cameras 07 and 09 respectively) rotation of the head up or down of 15^o approximately.



(a) (b) (c) (d)

(e) (f) (g)

(h) (i) (j)

(k) (l) (m)

Figure D.4: Sample images of the AR face database: (a) neutral expression, (b) smile, (c) anger, (d) scream, (e) left light on, (f) right light on, (g) both lights on, (h) wearing sun glasses (i) wearing sun glasses and left light on (j) wearing sun glasses and right light on, (k) wearing scarf, (l) wearing scarf and left light on (m) wearing scarf and right light on.

BIBLIOGRAPHY

[ACLT00] R. Auckenthaler, M. Carey, and H. Lloyd-Thomas. Score normalization for text-independent speaker verification systems. *Digital Signal Processing*, 10(1-3):42–54, Jan 2000.

[AH96] A. Acero and X. Huang. Speaker and gender normalization for continuous-density hidden Markov models. In *Proc. of the IEEE Int. Conf. on Acoustics Speech and Signal Processing (ICASSP)*, volume 1, pages 342–345, 1996.

[AHK65] K. Abend, T. Harley, and L. Kanal. Classification of binary random patterns. *IEEE Trans. on Information Theory (IT)*, IT-11(4):538–544, Oct 1965.

[AKLP93] O. Agazzi, S. Kuo, E. Levin, and R. Pieraccini. Connected and degraded text recognition using planar hidden Markov models. In *Proc. of the IEEE Int. Conf. on Acoustics Speech and Signal Processing (ICASSP)*, volume 5, pages 113–116, 1993.

[AMSM96] T. Anastasakos, J. McDonough, R. Schwartz, and J. Makhoul. A compact model for speaker-adaptive training. In *Proc. of the ISCA Int. Conf. on Spoken Language Processing (ICSLP)*, volume 2, pages 1137–1140, 1996.

[AMU97] 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 (PAMI)*, 19(7):721–732, July 1997.

[AR] The AR face database, http://rvl1.ecn.purdue.edu/ aleix/aleix_face_db.html.

[Bar81] R. Baron. Mechanisms of human facial recognition. *International Journal of Man Machine Studies*, 15:136–178, 1981.

[BBP01] D. Blackburn, M. Bone, and P. Phillips. Face recognition vendor test 2000: evaluation report. Technical report, 2001.

[BBTD03] R. Beveridge, D. Bolme, M. Teixeira, and B. Draper. *The CSU face identification evaluation system users' guide version 5.0.* Computer science department, Colorado State University, May 2003.

[Bey94] D. Beymer. Face recognition under varying pose. *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 756–761, June 1994.

[BHK97] P. Belhumeur, J. Hespanha, and D. Kriegman. Eigenfaces vs. fisherfaces: recognition using class specific linear projection. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 19(7):711–720, July 1997.

[Bil98] J. Bilmes. A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models. Technical report, Department of Electrical Engineering and Computer Science, U.C. Berkeley, 1998.

[BM95] R. Brunelli and S. Messelodi. Robust estimation of correlation with applications to computer vision. *Pattern Recognition*, 28(6):833–841, 1995.

[Boc00] E. Bocchieri. Phonetic context dependency modeling by transform. In *Proc. of the ISCA Int. Conf. on Spoken Language Processing (ICSLP)*, volume 4, pages 179–182, 2000.

[Bou00] H. Bourlard. Auto-association by multilayer perceptrons and singular value decomposition. Technical report, IDIAP, 2000.

[Bov00] A. Bovik. *Handbook of image and video processing.* Academic Press, 2000.

[BP93] R. Brunelli and T. Poggio. Face recognition: Features versus templates. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 15(10):1042–1052, Oct 1993.

[BRV02] V. Blanz, S. Romdhani, and T. Vetter. Face identification across different poses and illuminations with a 3D morphable model. In *Proc. of the IEEE Int. Conf. on Automatic Face and Gesture Recognition (AFGR)*, 2002.

[BS97] M. Stewart Bartlett and T. Sejnowski. Independent components of face images: a representation for face recognition. In *Proc. of the 4th Annual Joint Symposium on Neural Computation*, 1997.

[BSDG01] J. Beveridge, K. She, B. Draper, and G. Givens. A nonparameteric statistical comparison of principal component and linear discriminant subspaces for face recognition. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 535–542, 2001.

[BSL93] Y. Bar-Shalom and X.-R. Li. *Estimation and tracking: principles, techniques and software.* Artech House, 1993.

[Bur88] P. Burt. Smart sensing within a pyramid vision machine. *Proc. of the IEEE, Invited paper*, 76(8):1006–1015, Aug 1988.

[Bur98] C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, (2):121–167, June 1998.

[CF90] G. Cotrell and M. Fleming. Face recognition using unsupervised feature extraction. *Proc. of the Int. Neural Network Conf.*, pages 322–325, 1990.

[CLK$^+$00] L.-F. Chen, H.-Y. Liao, M.-T. Ko, J.-C. Lin, and G.-J. Yu. A new LDA-based face recognition system which can solve the small sample size problem. *Pattern Recognition*, 33(10):1713–1726, Oct. 2000.

[CLYW92] Y.-Q. Cheng, K. Liu, J.-Y. Yang, and H.-F. Wang. A robust algebraic method for human face recognition. In *11-th Int. Conf. on Pattern Recognition Methodology and Systems*, volume 2, pages 221–224, 1992.

[CSB04] F. Cardinaux, C. Sanderson, and S. Bengio. Face verification using adapted generative models. In *Proc. of the IEEE Int. Conf. on Automatic Face and Gesture Recognition (AFGR)*, pages 825–830, 2004.

[CT93] T. Cover and J. Thomas. *Elements of Information Theory.* John Wiley & Sons, Inc., 1993.

[CWS95] R. Chellappa, C. Wilson, and S. Sirohey. Human and machine recognition of faces: a survey. *Proc. of the IEEE*, 83(5):705–740, May 1995.

[DA04] J. Droppo and A. Acero. Noise robust speech recognition with a switching linear dynamic model. In *Proc of the IEEE Int. Conf. on*

Acoustics Speech and Signal Processing (ICASSP), volume 1, pages 953–956, 2004.

[DBH$^+$99] G. Donato, M. Bartlett, J. Hager, P. Ekman, and T. Sejnowski. Classifying facial actions. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 21(10):974–989, Oct 1999.

[DFB99] B. Duc, S. Fischer, and J. Bigün. Face authentication with gabor information on deformable graphs. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 8(4):504–516, Apr 1999.

[DHS00] R. Duda, P. Hart, and D. Stork. *Pattern classification.* John Wiley & Sons, Inc., 2nd edition, 2000.

[DLR77] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 39(1):1–38, 1977.

[EC96] K. Etemad and R. Chellappa. Face recognition using discriminant eigenvectors. In *Proc. of the IEEE Int. Conf. on Acoustics Speech and Signal Processing (ICASSP)*, volume 4, pages 2148–2151, 1996.

[EMR00] S. Eickeler, S. Müller, and G. Rigoll. Recognition of JPEG compressed face images based on statistical methods. *Image and Vision Computing*, 18(4):279–287, March 2000.

[FL03] B. Fasel and J. Luettin. Automatic facial expression analysis: a survey. *Pattern Recognition*, 36(1):259–275, 2003.

[Fre98] B. Frey. *Graphical models for machine learning and digital communication.* The MIT Press, 1998.

[Fuk90] K. Fukunaga. *Introduction to Statistical Pattern Recognition.* Academic Press, 1990.

[Gal00] M. Gales. Cluster adaptive training of hidden Markov models. *IEEE Trans. on Speech and Audio Processing*, 8(4):417–427, July 2000.

[GB03] R. Gross and V. Brajovic. An image preprocessing algorithm for illumination invariant face recognition. In *Proc. of the IAPR Audio- and Video-Based Biometric Person Authentication (AVBPA)*, pages 10–18, 2003.

[GBK01] A. Georghiades, P. Belhumeur, and D. Kriegman. From few to many: illumination cone models for face recognition under variable lighting and pose. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 23(6):643–660, June 2001.

[GL94] J.-L. Gauvain and C.-H. Lee. Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains. *IEEE Trans. on Speech and Audio Processing*, 2(2):291–298, Apr 1994.

[Gru00] M. Grudin. On internal representations in face recognition systems. *Pattern Recognition*, 33(7):1161–1177, 2000.

[GW92] R. Gonzalez and R. Woods. *Digital image processing.* Addison-Wesley, 1992.

[Hay99] S. Haykin. *Neural networks: a comprehensive foundation.* Prentice Hall, second edition edition, 1999.

[HHWP03] B. Heisele, P. Ho, J. Wu, and T. Poggio. Face recognition: component-based versus global approaches. *Computer Vision and Image Understanding*, 91(1-2):6–21, July-Aug 2003.

[HJ98] L. Hong and A. Jain. Integrating faces and fingerprints for person identification. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 20(12):1295–1307, Dec 1998.

[HL01] E. Hjelmas and B. Low. Face detection: a survey. *Proc. of the IEEE Conf. on Computer Vision and Image Understanding*, 83(3):236–274, Sep 2001.

[HLSS02] K. Hallouli, L. Likforman-Sulem, and M. Sigelle. A comparative study between decision fusion and data fusion in Markovian printed character recognition. In *Proc. of the IEEE Int. Conf. on Pattern Recognition (ICPR)*, volume 3, pages 147–150, 2002.

[Hon91] Z.-Q. Hong. Algebraic feature extraction of image for recognition. *Pattern Recognition*, 24(3):211–219, 1991.

[Hor86] B. Horn. *Robot Vision.* Mc Graw-Hill, New-York, 1986.

[INS] Inspass, http://www.immigration.gov/graphics/howdoi/inspass.htm.

[JCL96] B.-H. Juang, W. Chou, and C.-H. Lee. *Automatic speech and speaker recognition – advanced topics*, chapter Statistical and discriminative methods for speech recognition, pages 109–132. Kluwer Academics, 1996.

[JKLM99] K. Jonsson, J. Kittler, Y. Li, and J. Matas. Support vector machines for face authentication. In *The British Machine Vision Conf. (BMVC)*, pages 543–553, 1999.

[JMKL00] K. Jonsson, J. Matas, J. Kittler, and Y. Li. Learning support vectors for face verification and recognition. In *Proc. of the IEEE Int. Conf. on Automatic Face and Gesture Recognition (AFGR)*, pages 208–213, 2000.

[Jol86] I. Joliffe. *Principal Component Analysis.* Springer-Verlag, 1986.

[JRP04] A. Jain, A. Ross, and S. Prabhakar. An introduction to biometric recognition. *IEEE Trans. on Circuits and Systems for Video Technology (CSVT)*, 14(1), Jan 2004.

[KA94] S. Kuo and O. Agazzi. Keyword spotting in poorly printed documents. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 16(8):842–848, Aug 1994.

[KJNN00] R. Kuhn, J.-C. Junqua, P. Nguyen, and N. Niedzielski. Rapid speaker adaptation in eigenvoice space. *IEEE Trans. on Speech and Audio Processing*, 8(6):695–707, Nov 2000.

[KKB02] H.-C. Kim, D. Kim, and S. Bang. Face recognition using the mixture-of-eigenfaces method. *Pattern Recognition Letters*, 23(13):1549–1558, Nov. 2002.

[KLM00] K. Kittler, Y.-P. Li, and J. Matas. On matching scores for LDA-based face verification. In *Proc. of the British Machine Vision Conf. (BMVC)*, pages 42–51, 2000.

[Koh89] T. Kohonen. *Self Organization and Associative Memory.* New York: Springer Verlag, 1989.

[KPJ01] R. Kuhn, F. Perronnin, and J.-C. Junqua. Time is money: why very rapid adaption matters. In *proc. of the ISCA Workshop on Adaptation Methods for Speech Recognition*, pages 33–36, 2001.

[KR90] L. Kaufman and P. Rousseeuw. *Finding groups in data: an introduction to cluster analysis*, chapter Partitioning around medoids. John Wiley & Sons, 1990.

[Krü97] N. Krüger. An algorithm for the learning of weights in discrimination functions using a priori constraints. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 19(7), Jul 1997.

[KS90] 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 (PAMI)*, 12(1):103–108, Jan 1990.

[KTP00] C. Kotropoulos, A. Tefas, and I. Pitas. Frontal face authentication using discriminant grids with morphological feature vectors. *IEEE Trans. on Multimedia*, 2(1):14–26, Mar. 2000.

[LBG80] Y. Linde, A. Buzo, and R. Gray. An algorithm for vector quantizer design. *IEEE Trans. on Communications*, COM-28(1):84–95, Jan 1980.

[LGTB97] S. Lawrence, C. Giles, A. Tsoi, and A. Back. Face recognition: A convolutional neural-network approach. *IEEE Trans. on Neural Networks (NN)*, 8(1):98–112, Jan. 1997.

[Li94] S.-Z. Li. Markov random field models in computer vision. In *Proc. of the IEEE European Conf. on Computer Vision (ECCV)*, volume B, pages 361–370, 1994.

[LKL97] S.-H. Lin, S.-Y. Kung, and L.-J. Lin. Face recognition/detection by probabilistic decision-based neural network. *IEEE Trans. on Neural Networks (NN)*, 8(1):114–132, Jan. 1997.

[LNG00] J. Li, A. Najmi, and R. Gray. Image classification by a two-dimensional hidden Markov model. *IEEE Trans. on Signal Processing*, 48(2):517–533, Feb 2000.

[LS01] S. Liu and M. Silverman. A practical guide to biometric security technology. *IT Professional*, 3(1):27–32, Jan-Feb 2001.

[LVB$^+$93] M. Lades, J. Vorbrüggen, J. Buhmann, J. Lange, C. v.d. Malsburg, R. Würtz, and W. Konen. Distortion invariant object recognition in the dynamic link architecture. *IEEE Trans. on Computers*, 42(3):300–311, Mar 1993.

[LW95] C. Leggetter and P. Woodland. Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models. *Computer Speech and Language*, 9(2):171–185, Apr 1995.

[LW02] C. Liu and H. Wechsler. Gabor feature based classification using the enhanced Fisher linear discriminant model for face recognition. *IEEE Trans. on Image Processing (IP)*, 11(4):467–476, Apr 2002.

[MCvdM92] B. Manjunath, R. Chellappa, and C. von der Malsburg. A feature based approach to face recognition. *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 373–378, 1992.

[MDK+97] A. Martin, G. Doddington, T. Kamm, M. Ordowski, and M. Przybocki. The DET curve in assessment of detection task performance. In *Proc. of the ISCA European Conf. on Speech Communication and Technology (EUROSPEECH)*, pages 1895–1898, 1997.

[MHNM97] C. Miller, B. Hunt, M. Neifeld, and M. Marcellin. Binary image reconstruction via 2-D Viterbi search. In *Proc. of the IEEE Int. Conf. on Image Processing (ICIP)*, volume 1, pages 181–184, 1997.

[MK90] T. Stonham M. Krin. Face recognition using a digital neural network with self organizing capabilities. *Proc. of the IEEE Int. Conf. on Pattern Recognition (ICPR)*, 1:738–741, 1990.

[MK01] A. Martìnez and A. Kak. PCA versus LDA. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 23(2):228–233, Feb 2001.

[MMJP03] D. Maltoni, D. Maio, A. K. Jain, and S. Prabhakar. *Handbook of Fingerprint Recognition.* Springer Verlag, 2003.

[MN93] H. Murase and S. K. Nayar. Learning and recognition of 3-d objects from appearance. *Proc. of the IEEE Workshop on Qualitative Vision*, pages 39–50, June 1993.

[MNP01] B. Moghaddam, C. Nastar, and A. Pentland. A Bayesian similarity measure for deformable image matching. *Image and Vision Computing*, 19:235–244, 2001.

[Mog99] B. Moghaddam. Principal manifolds and Bayesian subspaces for visual recognition. In *Proc. of the IEEE Int. Conf. on Computer Vision (ICCV)*, pages 1131–1136, 1999.

[Mog02] B. Moghaddam. Principal manifolds and probabilistic subspaces for visual recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 24(6):780–788, June 2002.

[MP97] B. Moghaddam and A. Pentland. Probabilistic visual learning for object representation. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 19(7):696–710, 1997.

[MP98] H. Moon and J. Phillips. *Empirical evaluation techniques in computer vision*, chapter Analysis of PCA-based face recognition algorithms. IEEE Computer Society Press, 1998.

[MRC94] D. Miller, K. Rose, and P. Chou. Deterministic annealing for trellis quantizer and HMM design using Baum-Welch re-estimation. In *Proc. of the IEEE Int. Conf. on Acoustics Speech and Signal Processing (ICASSP)*, volume 5, pages 261–264, 1994.

[MW02] A. Mansfield and J. Wayman. Best practices in testing and reporting performance of biometric devices, Aug 2002.

[MWP98] B. Moghaddam, W. Wahid, and A. Pentland. Beyond eigenfaces: probabilistic matching for face recognition. *Proc. of the IEEE Int. Conf. on Automatic Face and Gesture Recognition (AFGR)*, pages 30–35, 1998.

[MZ93] S. Mallat and Z. Zhang. Matching pursuits with time-frequency dictionaries. *IEEE Trans. on Signal Processing*, 41(12):3397–3415, Dec 1993.

[Nef99] A. Nefian. *A hidden Markov model-based approach for face detection and recognition*. PhD thesis, Georgia Institute of Technology, 1999.

[Nef02] A. Nefian. Embedded Bayesian networks for face recognition. In *Proc. of the IEEE Int. Conf. on Multimedia and Expo (ICME)*, volume 2, pages 25–28, 2002.

[Ngu99] P. Nguyen. Maximum likelihood eigenspace and MLLR for speech recognition in noisy environments. In *Proc. of the ISCA European Conf. on Speech Communication and Technology (EUROSPEECH)*, pages 2519–2522, 1999.

[Nor96] Y. Normandin. *Automatic speech and speaker recognition – advanced topics*, chapter Maximum mutual information estimation of hidden Markov models, pages 57–81. Kluwer Academics, 1996.

[O'S97] O. O'Sullivan. Biometrics comes to life. *Banking Journal*, Jan 1997.

[PA96] P. Penev and J. Atick. Local feature analysis: a general statistical theory for object representation, 1996.

[Pen00] P. Penev. Redundancy and dimensionality reduction in sparse-distributed representations of natural objects in terms of their local features. In *Advances in Neural Information Processing Systems*, pages 901–907, 2000.

[Per00] F. Perronnin. Improving acoustic models for large-vocabulary systems. Master's thesis, Institut Eurécom, June 2000.

[PGM$^+$03] P. Phillips, P. Grother, R. Micheals, D. Blackburn, E. Tabassi, and M. Bone. Face recognition vendor test 2002: evaluation report. Technical report, 2003.

[Phi98] P. Phillips. Matching pursuit filters applied to face identification. *IEEE Trans. on Image Processing (IP)*, 7(8):1150–1164, Aug 1998.

[Phi99] P. Phillips. Support vector machines applied to face recognition. In M. Kearns, S. Solla, and D. Cohn, editors, *Advances in Neural Information Processing Systems*, pages 803–809, 1999.

[PMRR00] P. Phillips, H. Moon, S. Rizvi, and P. Rauss. The FERET evaluation methodology for face recognition algorithms. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 22(10):1090–1104, Oct 2000.

[PMS94] A. Pentland, B. Moghaddam, and T. Starner. View-based and modular eigenspaces for face recognition. *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 84–91, June 1994.

[QSS00] A. Quarteroni, R. Sacco, and F. Saleri. *Numerical Mathematics*. Springer-Verlag, 2000.

[Rab89] L. Rabiner. A tutorial on hidden Markov models and selected applications. *Proc. of the IEEE*, 77(2):257–286, Feb 1989.

[Rey95] D. Reynolds. Speaker identification and verification using gaussian mixture speaker models. *Speech Communication*, 17:91–108, 1995.

[Rey97] D. Reynolds. Comparison of background normalization methods for text-independent speaker verification. In *Proc. of the ISCA European Conf. on Speech Communication and Technology (EUROSPEECH)*, volume 2, pages 963–966, 1997.

[RH01] D. Reynolds and L. Heck. Speaker verification: from research to reality. In *Proc. of the IEEE Int. Conf. on Acoustics Speech and Signal Processing (ICASSP): Tutorial*, 2001.

[RP96] A. Rosenberg and S. Parthasarathy. Speaker background models for connected digit password speaker verification. In *Proc. of the IEEE Int. Conf. on Acoustics Speech and Signal Processing (ICASSP)*, volume 1, pages 81–84, 1996.

[RQD00] D. Reynolds, T. Quatieri, and R. Dunn. Speaker verification using adapted gaussian mixture models. *Digital Signal Processing*, 10:19–41, 2000.

[Sam94] F. S. Samaria. *Face recognition using hidden Markov models.* PhD thesis, University of Cambridge, Cambridge, UK, 1994.

[San98] A. Sankar. Experiments with a Gaussian merging-splitting algorithm for HMM training for speech recognition. In *Proc. of the 1997 DARPA Broadcast News Transcription and Understanding Workshop*, pages 99–104, 1998.

[San02] C. Sanderson. *Automatic Person Verification Using Speech and Face Information.* PhD thesis, Griffith University, School of Microelectronic Engineering, 2002.

[SBB02] T. Sim, S. Baker, and M. Bsat. The CMU pose, illumination, and expression (PIE) database. In *Proc. of the IEEE Int. Conf. on Automatic Face and Gesture Recognition (AFGR)*, 2002.

[SIE] Siemens automotive, http://media.siemensauto.com.

[SK03] M. Savvides and V. Kumar. Illumination normalization using logarithm transforms for face authentication. In *Proc. of the IAPR Audio- and Video-Based Biometric Person Authentication (AVBPA)*, pages 549–556, 2003.

[SK04] M. Sadeghi and J. Kittler. Decision making in the LDA space: generalized gradient direction metric. In *Proc. of the IEEE Int. Conf. on Automatic Face and Gesture Recognition (AFGR)*, pages 248–253, 2004.

[SL97] K. Shinoda and C.-H. Lee. Structural MAP speaker adaptation using hierarchical priors. In *Proc. of the IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pages 381–388, 1997.

[SLK99] N. Shazeer, M. Littman, and G. Keim. Solving crossword puzzles as probabilistic constraint satisfaction. In *Proc. of the National Conf. on Articial Intelligence*, pages 156–162, 1999.

[SP02] C. Sanderson and K. Paliwal. Likelihood normalization for face authentication in variable recording conditions. In *proc. of the IEEE Int. Conf. on Image Processing (ICIP)*, volume 1, pages 301–304, 2002.

[SR03] R. Singh and B. Raj. Tracking noise via dynamical systems with a continuum of states. In *Proc. of the IEEE Int. Conf. on Acoustics Speech and Signal Processing (ICASSP)*, volume 1, pages 396–399, 2003.

[SRSC01] D. Sturim, D. Reynolds, E. Singer, and J. Campbell. Speaker indexing in large audio databases using anchor models. In *Proc. of the IEEE Int. Conf. on Acoustics Speech and Signal Processing (ICASSP)*, volume 1, pages 429–432, 2001.

[Sto84] T. Stonham. *Aspect of Face Processing*, chapter Practical Face Recognition and Verification with WISARD, pages 426–441. Dordrecht:Nijhoff, 1984.

[SW96] D. Swets and J. Weng. Using discriminant eigenfeatures for image retrieval. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 18(8):831–836, Aug 1996.

[TC01] T. Tokuyasu and P. Chou. Turbo recognition: a statistical approach to layout analysis. In *Proc. of the SPIE Electronic Imaging Conf. on Document Recognition and Retrieval*, volume 4307, pages 123–129, 2001.

[TC02] D. S. Turaga and T. Chen. Face recognition using mixtures of principal components. *Proc. of the IEEE Int. Conf. on Image Processing (ICIP)*, 2:101–104, 2002.

[TKH00] A. H. Sayed T. Kailath and B. Hassibi. *Linear estimation.* Prentice Hall, 2000.

[TKP01] A. Tefas, C. Kotropoulos, and I. Pitas. Using support vector machines to enhance the performance of elastic graph matching for frontal face recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 23(7):735–746, July 2001.

[Tok01] T. Tokuyasu. *Turbo Recognition: an Approach to Decoding page Layout.* PhD thesis, University of California, Berkeley, 2001.

[TP91] M. Turk and A. Pentland. Face recognition using eigenfaces. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 586–591, 1991.

[TTWF03] Y.-A. Tian, T.-N. Tan, Y.-H. Wang, and Y.-C. Fang. Do singular values contain adequate information for face recognition? *Pattern Recognition*, 36:649–655, 2003.

[Vap95] V. Vapnik. *The nature of statistical learning theory.* Springer, New York, 1995.

[VDR99] M. Vissac, J.-L. Dugelay, and K. Rose. A novel indexing approach for multimedia image databases. In *IEEE Workshop on Multimedia Signal Processing (MMSP)*, pages 97–102, 1999.

[Way00a] J. Wayman. A definition of ”biometrics”. In J. Wayman, editor, *National Biometric Test Center Collected Works 1997-2000*, pages 21–23. Aug 2000.

[Way00b] J. Wayman. Fundamentals of biometric authentication technologies. In J. Wayman, editor, *National Biometric Test Center Collected Works 1997-2000*, pages 1–19. Aug 2000.

[WFKvdM97] L. Wiskott, J. Fellous, N. Krüger, and C. von der Malsburg. Face recognition by elastic bunch graph matching. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 19(7):775–779, July 1997.

[Wis97] L. Wiskott. Phantom faces for face analysis. *Pattern Recognition*, 30(6):837–846, 1997.

[Woo01] P. Woodland. Speaker adaptation: techniques and challenges. In *Proc. of the ISCA Workshop on Adaptation Methods for Speech Recognition*, 2001.

[WS92] W. J. Welsh and D. Shah. Facial feature image coding using principal components. *IEEE Electronic Letters*, 28(22):2066–2067, Oct 1992.

[WS99] J. Weng and D. Swets. *Biometrics: personal identification in networked society*, chapter Face recognition. Kluwer, 1999.

[WT03] X. Wang and X. Tang. Unified subspace analysis for face recognition. In *Proc. of the IEEE Int. Conf. on Computer Vision (ICCV)*, pages 679–686, 2003.

[YAK00] M.-H. Yang, N. Ahuja, and D. Kriegman. Face recognition using kernel eigenfaces. In *Proc. of the IEEE Int. Conf. on Image Processing (ICIP)*, volume 1, pages 37–40, 2000.

[Yan02] M.-H. Yang. Kernel eigenfaces vs. kernel fisherfaces: Face recognition using kernel methods. In *Proc. of the IEEE Int. Conf. on Automatic Face and Gesture Recognition (AFGR)*, pages 215–220, 2002.

[YDR00] W. Yambor, B. Draper, and J. Ross. Analyzing PCA-based face recognition algorithms: eigenvector selection and distance measures. In *Workshop on Empirical Evaluation in Computer Vision*, 2000.

[YEK$^+$01] S. Young, G. Evermann, D. Kershaw, G. Moore, J. Odell, D. Ollason, V. Valtchev, and P. Woodland. *The HTK book*. Cambridge University Engineering Department, Dec 2001.

[YKA02] M-H. Yang, D. Kriegman, and N. Ahuja. Detecting faces in images:a survey. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 24(1):34–58, Jan 2002.

[YY03] J. Yang and J.-Y. Yang. Why can LDA be performed in PCA transformed space? *Pattern Recognition*, 36(2):563–566, Feb. 2003.

[Zha99] W. Zhao. *Robust image based 3-D face recognition*. PhD thesis, University of Maryland, Department of Electrical and Computer Engineering, 1999.

[ZYL97] J. Zhang, Y. Yan, and M. Lades. Face recognition: Eigenface, elastic matching, and neural nets. *Proc. of the IEEE*, 85(9), Sep. 1997.