THÈSE PRÉSENTÉE À LA FACULTÉ DES SCIENCES
POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

Algorithms and VLSI Architectures for Low-Power Mobile Face Verification

par

Jean-Luc Nagel

Acceptée sur proposition du jury:
Prof. F. Pellandini, directeur de thèse
PD Dr. M. Ansorge, co-directeur de thèse
Prof. P.-A. Farine, rapporteur
Dr. C. Piguet, rapporteur

Soutenue le 2 juin 2005

INSTITUT DE MICROTECHNIQUE
UNIVERSITÉ DE NEUCHÂTEL

2006

To the best of my knowledge and belief, this thesis contains no material previously published or communicated by another person, that has not been duly referred to in the course of the manuscript.

Dissertation typeset with Adobe FrameMaker 6.0 by the author.

Université de Neuchâtel unine

IMPRIMATUR POUR LA THESE

Algorithms and VLSI Architectures for Low-Power-Mobile Face Verification

Jean-Luc NAGEL

UNIVERSITE DE NEUCHATEL

FACULTE DES SCIENCES

La Faculté des sciences de l'Université de Neuchâtel,
sur le rapport des membres du jury

MM. F. Pellandini (directeur de thèse),
P.-A. Farine, M. Ansorge et
C. Piguet (CSEM, Neuchâtel)

autorise l'impression de la présente thèse.

Neuchâtel, le 26 septembre 2005

La doyenne:

M. Rahier

Faculté des Sciences
■ Rue Emile-Argand 11 ■ CP 2 ■ CH-2007 Neuchâtel
■ Téléphone : +41 32 718 21 00 ■ Fax : +41 32 718 21 03 ■ E-mail : secretariat.sciences@unine.ch ■ www.unine.ch

A mes parents

Keywords

Biometrics; face authentication; image acquisition and processing; pattern recognition; mobile devices; low-power digital circuits and systems design.

Abstract

Among the biometric modalities suitable for mobile applications, face verification is definitely an interesting one, because of its low level of invasiveness, and because it is easily applicable from the user viewpoint. Additionally, a reduced cost of the identity verification device is possible when sharing the imaging device between different applications. This is for instance not possible with fingerprint verification, which asks for a specific sensor.

This Ph.D. thesis addresses the problem of conceiving a low-power face authentication system adapted to the stringent requirements of mobile communicators, including the study and design of the algorithms, and the mixed hardware and software implementation on a dedicated multi-processors platform. With respect to the anterior state-of-art, this thesis proposes several original algorithmic improvements, and an original architecture for a VLSI System-On-Chip realization, where the algorithmic and architectural aspects have been jointly optimized, so as to enable the execution of the face verification to be *entirely* processed on low-power mobile devices such as mobile phones, or personal digital assistants. At the time when this work was undertaken and completed, there were no similar results published either from the academic scientific community, or from the market in form of a product description / announcement.

The mobile environment is known to be characterized by large variations in image acquisition conditions with regard to the scene acquisition viewpoint and illumination, and it is therefore requiring a sound optimization of the algorithmic robustness. Moreover, face verification is a quite complex task that is usually performed on powerful - possibly

multi-processor - mainframe computers rather than on low-power digital signal processors. In the case of mobile applications, the algorithms need to be tightly optimized, and specific hardware architectures need to be developed in order to fulfill the power consumption constraints.

This thesis thus embraces several domains such as cognitive science, pattern recognition, image acquisition and processing, and low-power digital circuit design. The first part of the thesis presents a list of possible applications and of constraints raised by the mobile environment, and an extensive literature review. A class of algorithms is selected, characterized, and optimized with respect to the specificities of mobile face verification. The second part of the thesis describes a System-on-Chip implementation performed at system-level, including image acquisition and higher-level processing. Finally, the architecture of a specific embedded VLSI coprocessor is described and discussed complementarily to the corresponding design methodology used.

Mots-clés

Biométrie; vérification de visages; acquisition et traitement d'image; reconnaissance de formes; terminaux mobiles; conception de circuits et sytèmes numériques à faible consommation.

Résumé

Parmi les modalités biométriques convenant à la vérification d'identité sur un dispositif portable, la reconnaissance de visage est particulièrement intéressante de par sa faible invasivité, et de par sa simplicité d'usage (du moins du point de vue de l'utilisateur). De plus, le coût du dispositif de vérification d'identité peut être réduit lorsque le senseur d'image est partagé entre plusieurs applications. Ce n'est pas le cas de la reconnaissance d'empreintes digitales, par exemple, qui nécessite un senseur propre.

Cette thèse traite le problème de la conception d'un sysème de vérification de visage à faible consommation, adapté aux besoins pointus des terminaux mobiles. Ce travail comprend l'étude et la conception d'algorithmes, ainsi qu'une implantation mixte matérielle-logicielle sur une plate-forme dédiée comprenant plusieurs processeurs. Par rapport à l'état de l'art, cette thèse propose plusieurs améliorations algorithmiques originales, et une architecture VLSI originale pour la réalisation d'un système sur puce, dans laquelle les aspects algorithmiques et architecturaux ont été conjointement optimisés, de sorte que l'exécution de la reconnaissance de visage peut être *entièrement* traitée sur le dispositif mobile (téléphone portable ou un agenda personnel). Au moment de la réalisation et de la clôture de ce travail de thèse, aucun résultat équivalent n'avait été publié par la communauté scientifique, ou n'était disponible sur le marché sous la forme d'une description ou annonce de produit.

L'environnement mobile est caractérisé par de larges variations des conditions d'acquisition des images, en ce qui concerne l'angle de vue et l'illumination, ce qui nécessite une conséquente optimisation de la

robustesse des algorithmes. De plus, la vérification de visages est une tâche relativement complexe, généralement effectuée sur des stations comportant un ou plusieurs processeurs performants, plutôt que sur des processeurs de traitement du signal à faible consommation. Dans le cas des applications portables, les algorithmes doivent donc être également optimisés en termes de complexité, et des architectures matérielles spécialisées doivent être développées pour arriver à respecter les contraintes de consommation électrique.

Cette thèse couvre donc différents domaines, tels que les sciences cognitives, la reconnaissance de formes, l'acquisition et le traitement d'image, et la conception de circuits intégrés numériques à faible consommation. La première partie du rapport décrit une liste d'applications possibles, et les contraintes imposées par l'environnement mobile, ainsi qu'une revue de l'état de l'art étendue. Une catégorie d'algorithmes est sélectionnée, caractérisée, et optimisée en fonction des spécificités de la vérification d'identité sur un appareil portable. La deuxième partie de la thèse décrit au niveau système une implantation sous la forme d'un système-sur-puce contenant aussi bien le senseur d'image qu'un micro-contrôleur et des blocs de traitement spécialisés (coprocesseurs). Finalement, une architecture spécifique de coprocesseur embarqué est décrite et discutée, complémentairement à la méthodologie de conception utilisée.

Contents

Chapter 4: Elastic Graph Matching Algorithm Applied to Face Verification

Chapter 5: Further Use of Elastic Graph Matching

Chapter 6: A Low-Power VLSI Architecture for Face Verification

List of Figures

Chapter 1

Introduction

1.1 Motivations

This thesis report presents several algorithm improvements and a dedicated VLSI system-on-chip architecture, which jointly allow the execution of face verification algorithms that are *entirely* operated on a low-power mobile device (e.g. a mobile phone or a personal digital assistants, PDA).

The motivations driving this work were multifold: firstly, one of the major problems in face recognition is clearly its sensitivity to lighting variations, and this phenomenon is even emphasized for mobile devices because of the strongly varying and uncontrolled environmental conditions. Therefore, the challenge of understanding the effects of lighting variation on algorithms in order to improve their robustness, while maintaining a reasonable computational complexity, was very stimulating.

A further strong motivation was given by the fact that – to the best of our knowledge – no commercial solution providing true mobile face verification, in the sense that the verification is being performed on the device itself and not in a base station, existed at time of the realization and closure of this Ph.D. work.

Finally, considering the design of low-power digital or mixed circuits, one observes that pushing the limits of power consumption requires integrating increasingly more functionalities on the same chip, which leads to the so-called System-on-Chip (SoC) paradigm, and requires a careful co-design of software and hardware. The realization of a single Integrated Circuit dedicated to face verification, and embedding e.g. a digital camera, a general purpose micro-controller and several specific coprocessors, represents a challenging example of SoC design.

1.2 Organization of the thesis

Chapter 2 introduces the biometrics domain and illustrates the need of enhanced security on mobile devices. In addition, a list of possible applications and of constraints raised by the mobile environment is proposed.

Chapter 3 is then presenting an extensive but non-exhaustive review of the literature dealing with face authentication. Several criteria to select proper algorithms are defined and used to classify the discussed algorithms. Additionally, existing commercial products and patents are reviewed. Finally, a set of algorithms fulfilling the requirements of mobile face authentication is selected.

Chapter 4 describes in detail the selected set of algorithms, which is based on the so-called Elastic Graph Matching method, and includes the characterization and the optimization of several parameters with respect to the specificities of mobile face verification.

Chapter 5 extends the use of Elastic Graph Matching to face detection, and proposes the structure of a low-complexity color-based face detection algorithm, complemented by a further extension to fingerprint-based biometric authentication.

Chapter 6 deals with a System-on-Chip implementation at system-level including image acquisition and higher-level processing. Finally a specific embedded coprocessor is presented together with the applied associated design methodology and the resulting performance figures.

Finally, the conclusions provided in Chapter 7 are summarizing and discussing the different contributions reported in this thesis, and are sketching perspectives for future research and application developments.

1.3 Contributions

The major contributions reported in this thesis are the following:

Chapter 2: The identification of biometric applications that are implementable on mobile communicators, and the definition of the corresponding performance requirements.

Chapter 3: The determination of algorithm selection criteria, and their application to choose proper algorithms fulfilling as well the perform-

ance requirements established in Chapter 2. Furthermore, an additional contribution is given by the compilation of an updated review of the literature coming from both academic and commercial sources.

Chapter 4: A detailed analysis of the effects of illumination variations on the face authentication, and the review of several normalization techniques based on this analysis. A further contribution concerns the systematic comparison of feature extraction techniques combined to the Elastic Graph Matching (EGM) algorithm, including the introduction of a novel low-complexity feature extraction technique. A further contribution in this chapter consists in using the Simplex Downhill methods for the iterative optimization of a graph correspondence search, combined to a new scale adaptation and a new graph deformation penalty.

Chapter 5: A VLSI architecture for online color-based face detection combined with a model-based face detection that is reusing the EGM algorithm. A further contribution is given by the extension of the EGM usage to the fingerprint modality.

Chapter 6: The description of a low-power embedded face authentication system, including a dedicated coprocessor executing most of the EGM operations.

These major contributions are discussed in detail in the respective chapters, and they are also discussed in Chapter 7.

Chapter 2

Biometrics

This chapter introduces the necessary basic terminology and general concepts related to the biometric technology, which will be helpful in the rest of the thesis. Section 2.1 defines the term Biometrics and several associated expressions, and discusses the requirements of a biometric system. The different physiological traits that can be used in biometrics, i.e. the biometric modalities, are briefly presented in Section 2.2, followed by a discussion of verification vs. identification scenarios in Section 2.3. Additionally, potential applications of biometric systems are discussed in Section 2.4, from which particular applications benefiting from low-power mobile biometrics will be identified (Section 2.5). A discussion of privacy concerns related to the application of biometrics (in general and in the particular case of mobile personal applications) is proposed in Section 2.6. Finally, a discussion of the biometric performance evaluation will be carried out in Section 2.7, jointly to the description of a list of existing face databases and protocols that are commonly used to compare algorithms.

2.1 Introduction

A very general definition of the term Biometrics finds its root in biology:

> "*Biometrics is a branch of biology that studies biological phenomena and observations by means of statistical analysis*"

In this sense, biometrics covers equally the analysis of data from agricultural field experiments (e.g. comparing yields of different wheat varieties), the analysis of data from human clinical trials (e.g. evaluating the relative effectiveness of competing therapies), or the analysis of data from environmental studies (e.g. effects of air or water pollution on the appearance of human disease in a region or country) [2-7].

However, we are more concerned here with the information technology oriented definition, where biometrics usually refer to technologies for measuring and analyzing individual physiological characteristics such as fingerprints, eye retinas and irises, voice patterns, facial patterns, and hand measurements. A conventional [2-2] definition for *biometrics* is thus given as follows:

> "*Biometrics are automated methods of recognizing a person based on a physiological or behavioral characteristic*".

Several important observations can be readily made:

- the terms *physiological* or *behavioral* refer to properties individually characterising the *being* of a person, in contrast with an information known by the person (a password, a personal identification number) or an object possessed by the person (a key, a card, a token), which are two other pillars of security applications.
- only applications associating an *identity* to physiological or behavioral characteristics can be considered as biometric technologies. Some applications may perform a *recognition* of the behavior or of some physiological characteristics for other purposes not directly related to identity management e.g. human-computer interaction, HCI. Definitions and distinction between terms "verification", "authentication", "recognition" and "identification" will be discussed in Section 2.3.
- *uniqueness* (or *individuality* of the physiological or behavioral characteristics) is necessary to separate two persons. This means that the characteristics associated to two different identities shall be "sufficiently different" so that their identities can be distinguished, i.e. the probability that any two or more individuals may have sufficiently similar representations, so that they lead to a false correspondence, is very low. Yet otherwise stated, the *distinctiveness* of the features, i.e. the variations or differences in these features among the general population, should be sufficiently high to ensure their uniqueness.
  Note that in real systems, although the underlying physiological traits are unique, the *measured* characteristics may not be unique (see Subection 3.2.2).
- the term *automated* indicates that the decision of recognizing a person should not involve a human decision. Some systems are

called semi-automatic systems when they require some human action during enrolment (i.e. the creation of a set of physiological or behavioral reference characteristics and its association to an identity. This human interaction generally consists in a manual registration of characteristic points, so-called fiducial points, in an image, followed by the automatic extraction of biometric characteristics. However the testing stage is most of the time fully automatic.

- the level of *intrusiveness* (i.e. how the technology interferes with privacy concerns of a person) will generally define the *user acceptance* level (i.e. how users perceive and are willing to use biometric technology). User acceptance may be depending on several factors such as the ease of enrolment, a possible required contact between the user and a machine performing the verification, the required disclosure of otherwise unseen traits, or even the disclosure of more information than required for a simple identity recognition (e.g. DNA analysis might reveal details about the medical state of a person).
- *accuracy* is generally inversely proportional to the intrusiveness because a more constrained and more intrusive technique leaves less room to environmental variations. Depending on the targeted application and level of security, a very accurate albeit intrusive technique might be necessary. Alternatively if the required level of security is not drastically high a more “user-friendly” technique could be used.
- *environmental factors* (e.g. lighting or position of camera, i.e. viewpoint, for faces; moisture or dirt for fingerprints) generally have an effect on accuracy: the more sensitive physiological traits are to external factors, the less accurate will they generally be. Indeed in order not to falsely reject genuine users (i.e. users whose physiological traits are known from the system and who should therefore be recognized) in environmentally degraded conditions, the required correspondence between the measured traits and the know reference will have to be relaxed, thus allowing more impostors to gain access to the system.
- *liveness (or liveliness) detection:* the possibility for the system to assess that the sample comes from a live human is necessary for unsupervised (i.e. unmanned) check points, or they could be otherwise attacked with fake physiological “templates” (e.g. photo-

graphs in facial recognition systems, “gummy bear fingers” in fingerprint systems [2-13], voice records in speaker verification systems, or prosthesis in hand geometry verification systems).

- *vulnerability* refers to the risk for the biometric system and method (and for the associated data) to be compromised, generally as a result of a fraudulent activity.

Other factors such as cost, speed, ease of integration or template size can also be considered depending on the targeted application [2-15].

2.2 Biometric modalities

Biometric *modalities* represent the physiological or behavioral characteristics that are used for identification or verification. This section briefly presents the most popular modalities together with their advantages and inconveniences in order to select the best suited modality for *mobile* applications.

2.2.1 Hand geometry

One of the first biometric technologies to reach the practical application level was hand geometry as it was used in several U.S. airports in the frame of the United States Immigration and Naturalization Service (INS) Passenger Accelerated Service System (INSPASS). This systems is currently operational at Los Angeles, Miami, Newark, New York (JFK), and San Francisco airports in the U.S., as well as at Vancouver, and Toronto in Canada.

Advantages of hand geometry are mainly its ease of use compared to a relatively good performance. Hand geometry systems start from a 2.5D view of the palm using a digital camera and mirrors, from which a certain number of characteristics (length of fingers, thickness, etc.) are extracted and compared. Similarly to the situation occurring with fingerprints, the issue concerning the positioning of the hand on the platen is important and shall be consistent from one session to the other. Moreover, the uniqueness of this modality might be questionable, thus limiting the number of clients that may be enrolled in such a system. Finally, hand geometry requires a dedicated scanner and is obviously not usable in mobile applications.

2.2.2 Retina

Although sometimes confused with iris, the imaging of the retina relates to a different region of the eye[1]. Indeed, the blood vessels in the retina form patterns which are considered stable and unique and can be used for discriminating persons. The image acquisition is generally performed with an infra-rared light source.

This method was mainly used in the military domain and for very high security applications. Its high reliability is however balanced by its high level of intrusiveness: most of the time the user needs to align his face, possibly being in contact with a binocular, and he / she needs to fix a point in an image. Glasses shall be removed contrary to what happens with iris scanning. Clearly this method seems difficult to use in commercial applications and even impossible to use in mobile environments if precise alignment is required.

2.2.3 Iris

Experiments showed that the texture of the iris is unique among individuals - but also unique from one eye to the other - and stable across several decades. The very discriminating patterns of the iris make it a very interesting modality providing with a very high accuracy even in the presence of glasses or contact lenses.

After detection of the eyelids and iris, the texture is usually analyzed with a set of Gabor filters. A measure of energy is performed leading to a scalar measure for circular regions. All these values are grouped into a vector called *Iris Code*. A simple comparison of codes is finally necessary to distinguish genuine from impostor irises.

The image acquisition is carried out with a digital camera assuming that the eye is relatively close to it although there is no direct contact. This is a clear advantage as compared to other modalities requiring such a contact (e.g. retina or fingerprint scanners).

This method is potentially usable in a mobile context but several pending patents are limiting the use of this modality [2-5][2-21].

1. The iris is the colored frontal part of the eye. It controls light levels inside the eye similarly to the aperture on a camera. After being focused by the crystalline, the light reaches the retina - at the back of the eye - that is the sensory tissue of the eye.

2.2.4 Voice

A relatively natural means to identify persons is by using their voice. Characteristics of the voice are different among individuals due to differences in vocal chords, vocal tract, palate, and teeth, which interact during phonemes formation. The dynamics of the sound can thus be used to discriminate people based on audio recordings.

There are nonetheless difficulties coming from the acoustic environment which may vary considerably with the use of mobile devices. However this solution seems interesting for mobile phones, for instance because they already contain a microphone the user is familiar with. Finally a continuous verification can be performed while the user is speaking.

2.2.5 Fingerprint

Fingerprint matching is perhaps the most widely known biometric modality due to its extensive use in the forensic domain as reported by popular thrillers. This also means that the disclosure of fingerprints for everyday applications may limit its user acceptance due to its possibly inherent connotation referring to forensics. Fingerprints can be considered as being characteristic (they are unique among individuals) and robust (they remain stable during adulthood).

Most of the early automatic fingerprint identification systems (AFIS) were based on standard methodologies used in forensic domain. They consisted mainly in extracting features called *minutiae* (e.g. end of ridges, bifurcations, etc.), in classifying the print (e.g. arch, whorl, etc.) and then in comparing these features (type, location, orientation). The principal task of these AFIS systems was then to locate the minutiae in binary or gray level images.

Unfortunately, the degradation of the image quality - due to damaged fingers or dirty scanners - can lead to the detection of a high number of false minutiae. For this reason recent systems also use pattern recognition techniques for registering two images. Typically, statistical information is extracted from each image through a process called feature extraction. Then, these features are compared in a multidimensional feature space, leading to the class allocation of the test sample to one of the trained reference classes. It is assumed that extracted features are less sensitive to local degradations than minutiae extraction. Combination of pattern-based and minutiae-based systems seems to be

the most robust [2-8] (see also Section 5.2 for a discussion on pattern-based fingerprint matching techniques).

The major problem with fingerprint recognition thus comes especially from the image acquisition and from the degradation of the fingers themselves. In the majority of the fingerprint scanners, the tip of the finger is placed in contact with a flat surface[2]. Depending on the angle, orientation and pressure of the finger, the image acquired during two different test sessions (e.g. one day apart) could be already significantly different. Consequently, the deployment of the system may become complicated in order to assure that clients place their fingers consistently on the device. The quality of the images is also dependent on the humidness of the finger and on the dirt present on the sensor. This last issue generally reduces the use of fingerprint authentication to certain categories of professions practiced by the user (e.g. well applicable to lawyer, but not to garage mechanic).

Finally a special image acquisition device is required for grabbing fingerprint images. Current techniques are mainly optical (CMOS, CCD) and solid-state (capacitive, temperature) but other solutions recently appeared (e.g. ultra-sonic) [2-20].

2.2.6 Face

The uniqueness of the facial pattern might be difficult to prove due to the *intrinsic* variations encountered for a single person, regarding for instance hairstyle, expression, glasses, beard. Moreover it is very sensitive to *extrinsic* variations such as lighting, view angle and scale. Still, it is probably one of the more user friendly modalities because the client only needs to look at a camera, i.e. to disclose an image of his face, an action done daily without having the impression to reveal critical information.

Additionally, facial recognition might be used in non-cooperative environments because the user is not required to precisely place his face in a given position and he might be identified without even knowing it. This could be however questionable in terms of legacy and/or privacy (see also Section 2.6).

It is possible to apply facial recognition to mobile environments (as this thesis will demonstrate it) at the cost of a higher computational com-

2. Some “contact less” optical scanners are however now available on the market.

plexity for the necessary increase in robustness. To the best of our knowledge this modality has thus never been implemented entirely in mobile devices because they do not provide sufficient computational power for these complex algorithms. However, this becomes feasible when designing dedicated low-power accelerators.

2.2.7 Others

Other biometric modalities include for example handwritten signature, which already has some application in e.g. mail delivery business. and gait (i.e. the way a person is walking). Since personal digital assistants are readily equipped with a tablet and pen, they seem naturally adequate for this implementation. However the consistent reproduction of a signature on these devices requires a non-negligible user training.

Another modality is given by gait (i.e. the way a person is walking), which is nonetheless in its early stage of research only.

A further possibility to enhance the accuracy of a recognition system is to combine several modalities, i.e. by creating a *multi modal* system [2-9][2-18][2-10]. By using a soft[3] combination of probabilities given by two or more modalities, a more robust decision is expected. For instance voice and face recognition could be used on a mobile phone already equipped with microphone and camera, or face and fingerprint could be used, while sharing part of the necessary dedicated image processing hardware.

2.3 Verification vs. identification

The distinction between identification and verification mainly resides in the number of comparison performed between templates, and in the presence or absence of a prior identity claim.

Verification means a one-to-one matching (often labelled “1:1” matching) for which the client presents a claim of identity. The system then extracts from its database a trained template corresponding to the claim and verifies that the newly extracted test template corresponds to the stored one. This operation is also often called *authentication*. The error rate is normally not dependent on the number of persons enrolled in the database, which means that the addition of new clients should not deteriorate significantly the verification rate.

3. By contrast to “hard” decisions where a logical “1” indicates an acceptance and a logical “0” a rejection, a soft decision provides intermediate probabilistic decisions.

Identification on the other hand implies a one-to-many search (often labelled “1:N” matching) in which no prior identity claim is made. The system will systematically compare the extracted test template to all the stored templates. Sometimes however only a subset of the enrolled templates is searched (e.g. based on additional “clues” such as the structure type in fingerprints). Moreover the test image might correspond to a person who was never seen by the system so far. Consequently it should be possible to use a rejection class (i.e. verification in an open-set by contrast to a closed-set).

Recognition is used in this thesis for matching one template against one or many references, i.e. without further distinguishing its use either in verification or in identification. Many authors apply it as a synonym of identification.

The execution time of a single verification is generally in the order of one second or below whereas to search large databases of several thousands of entries, a comparison in a much shorter time (typically 1 ms) is necessary. Apart from this speed difference the algorithms will be generally very similar in the way they handle a single matching.

Other types of classifications might be imagined, for instance the one answering the question “is this person known from the system?”, without necessity to retrieve the name of the person. In this thesis, we are however mainly interested in the identity verification problem, thus relaxing constraints on matching speed.

2.4 Applications of biometrics

Typical applications [2-1] of biometrics are (in increasing order of requested level of security):

- intelligent environments (automatic adaptation of settings);
- visitor management systems;
- time and attendance monitoring;
- device access control (ATM[4], personal computer, personal digital assistant, mobile phone, etc.);
- physical access control (for buildings or offices);
- national identity passports (border control).

4. Automatic telling machines.

These different applications require different security levels. For instance, access to an ATM should be more secured than the access to a home personal computer when the latter solely aims at restricting some access to children.

Similarly, some modalities are more convenient for certain applications. More intrusive systems can be used for physical access control on a door based on the required level of security. However it would not be practical to use iris scanning for time and attendance monitoring or for visitor management. In the latter, more natural and non-cooperative methods (e.g. facial recognition) are required.

Vastly deployed systems (e.g. national identity passports) should encompass several biometric modalities so that disabled persons failing to be enrolled with one modality for physiological reasons could use another modality without discrimination.

Based on these observations we can conclude that facial recognition is applicable and interesting in monitoring and lower-security domains, especially where ease of use is important. This is the principal motivation for using facial recognition in mobile environments. We will now consider the constraints involved by this choice.

2.5 Mobile applications of biometric authentication

Face recognition (i.e. identification or authentication) may be used on a mobile device (laptop, Personal Digital Assistants - PDA, mobile phone, etc.) in the following typical application scenarios:

1. Restricted use of the device and restricted access to the local data it is containing.
2. Access restriction to controlled products that are accompanied by the mobile device (from the production plant, throughout their transport and up to their distribution).
3. Use of the device to verify/recognize the identity of an individual in the field (e.g. police controls of driver licenses).
4. Automatic adaptation of the mobile device's user settings.

Based on the required level of security, facial verification might not be sufficiently effective for scenarios 2 and 3, while the fourth one involves identification rather than verification because no identity claim should be necessary to automatically adapt the settings. Consequently the

first category seems the most adapted to mobile devices when considering an authentication scenario.

Following the predictions of the International Biometric Industry Association (IBIA) reported by BMI [2-16], there will be a shift in the biometric market from the currently widely developed physical and logical access control (90% in 2001) to the telecom arena (59% of the market in 2006). Simultaneously, worldwide mobile phone sales totalled 520 million units in 2003, a 20.5 percent increase compared to 2002[5].

Furthermore the convergence of PDAs and mobile phones can be observed, the latter containing more and more features of the first. The trend is then naturally to have a single device with extended capabilities (wireless internet, phone, camera, GPS). This will result in an increased number of potentially sensitive and private data stored on the device and/or transferred to remote locations. Thus the need for more effective security is clearly evidenced. To achieve security and privacy of data during transmission, all communications will be encrypted in 3G devices. However, this clearly does not protect the terminal itself. Similarly, the subscriber account is protected by appropriate codes that are exchanged between the phone and the network but again, these codes protect the communications and not the device.

Currently available methods for the protection of mobile phones heavily rely on Personal Identification Numbers (PIN codes) that prove inefficient for the following reasons:

- A code (i.e. something that we know) is indeed identified, not the person actually knowing the code.
- A code can be easily stolen (e.g. many people write it down).
- A code can be easily guessed (e.g. people use very simple codes such as birth date or simple sequences like ‘1234’).
- Codes are sometimes not used at all due to their inconvenience or because the users don’t even know about their existence.

Most handsets do not provide PIN protection while left on, but rather ask for a PIN only when being switched on. This is not usable since the majority of the users leave their mobile on in order to be able to receive a call. However according to a survey carried out by Clarke et al. [2-4]

5. Source: http://www.cellular.co.za, 2004

on the users' opinion regarding the security of mobile phones, around 80% of the current users believe that an enhanced security of the mobile phones would be good or very good. The motivations most often cited by the subscribers to explain why they are not using PIN codes are the lack of convenience and the lack of confidence in PIN codes. Logically the authors of this study asked the respondents which biometric technology they would be ready to use on mobile phones. Not surprisingly the quoted leading technology was fingerprint (74% of positive responses), with voice, iris and face coming behind, and ear geometry and typing style being even less favoured. This can be explained by the fact that the vast majority of users already had some experience with the fingerprint technology whereas it is generally not the case for other biometric technologies.

In new generations of devices, face verification can be instead favoured due to the presence of digital image sensors, the cost of which being then shared between multiple applications. Consequently biometrics seem ready to be implemented in these devices either in order to increase the security *or* to augment the convenience at an equivalent security level.

Interestingly, and obviously for privacy reasons, 52% of the respondents to the survey by Clarke et al. also favoured storing templates on the handset instead of storing them on a remote network. This generally involves performing the verification on the terminal itself instead of acquiring a template and sending it to a base station to perform the verification remotely.

Incorporation of biometric technologies in 3G devices could be made seamlessly, i.e. without modifications of established 3G specifications (see *http://www.3gpp.org*), as it is not directly related to communication protocols but rather handled locally. Instead of storing a PIN inside the SIM (Subscriber Identity Modules), biometric verification would use the new USIM (Universal-SIM) to store the biometric template of the user. The encryption achieved on the module would further enhance the protection of the biometric features themselves.

As a conclusion, the mobile environments clearly requires security improvements and biometrics seem an adequate solution. Additionally, the face modality seems appropriate due to its low level of invasiveness and its minimum extra cost because of the growing presence of digital cameras on PDAs and mobile phones.

2.6 Privacy

A recurrent concern with the application of large-scale facial identification systems is the protection of civilian liberties. Organizations trying to protect the civilian liberties are becoming more present in all modern democracies. The ACLU (American Civil Liberties Union) in the US is certainly the most prominent in this area but other examples can be found all around the world.

In Switzerland, a pilot experiment organized at Zürich airport (Unique Airport) was the target of such organizations. The authorities actually accepted a test phase where passengers arriving from specific countries were enrolled in a face identification system. The so established database, maintained during 30 days only, would have been used to recover the origin and identity of persons who were later assigned an illegal residence status and lost their passport. With this information, illegal residents would be sent back to the originating country, according to the agreements of the International Civil Aviation Organisation. Due to the ethical concerns connected to this project, and especially due to the fact that the system would have been applied only to some specific incoming flights and to specific persons presenting the “profile of potential illegal immigrants”, it was decided to provisionally renounce to this application beyond the test phase, until a new federal migration policy will regulate the use of biometrics for this type of applications.

Clearly the application of biometrics in smaller scale scenarios, e.g. at company or building level, or further at personal level as it is the case for mobile phones, does not present the same controversy regarding ethical reasons. Anyway, proportionality remains an active concern in Switzerland.

Arguments formulated by the opponents to the biometric technology mainly concern the fact that:

- biometric templates cannot be changed: “once you are filed in a database you cannot change your identity”;
- compromised templates cannot be changed: “a publicly known template can be used by an unauthorized individual and cannot be changed to a new secured template”;
- security is not significantly increased at the cost of large invasiveness and liberty reduction;

the first two arguments referring to the user's loss of control upon her/ his private data.

Biometric face verification as proposed for personal mobile devices does not suffer from the above arguments if the templates are stored locally. In such case, the user template remains local to the terminal and is not delivered to remote (government or private service-based) databases, so there is no reason to dissimulate one's identity.

The risk of compromising one's template or the possibility to create fake templates might be increased by a mobile context, as compared to a centralized database. Indeed, the impostor attempts at extracting a reference template or creating a fake test template would be undetected on a stolen device, whereas repeated attempts on a central database would be logged, resulting in the attacker being discovered. In addition, the possibility of taking components of the stolen device apart might help the attacker finding covertly weaknesses in the device.

However current cryptography already provides a high level of security for the locally stored data and moreover the theft of the decrypted template would certainly not allow to reproduce a valid authentication or identification (i.e. to produce a fake fingerprint or a fake face mask). Indeed it is not directly the template that is used in the system but a live representation of the considered biometric from which a matching template is extracted. As long as the system performs liveliness tests and is secured against direct feeding of templates in the core of the verification engine this theft would not be dramatic. If direct feeding is possible, then it would be presumably more effective to directly feed an "accept" signal into the system by totally bypassing biometrics engine! This small example clearly shows that biometrics are only part of a whole verification system that must be secured at several levels, i.e. the security is only as high as its weakest part.

2.7 Performance evaluation of biometric systems

In Chapter 4 different implementations of the Elastic Graph Matching (EGM) algorithm will be compared. To be able to estimate their performance, it is necessary to define a technical performance evaluation. Other types of evaluations (e.g. vulnerability, user acceptance, privacy compliance [2-12]) go beyond the scope of this thesis.

Following Phillips [2-17] and Blackburn [2-3] it is necessary to divide a technical performance evaluation in three distinct steps, namely:

- Technological evaluation;
- Scenario evaluation;
- Operational evaluation.

The first category of tests is necessary to measure the state of the art and particular capabilities of a given algorithm. This is generally done using standard databases that also allow comparisons with similar results achieved by other research teams. Open competitions such as the Face Recognition Vendor Test (FRVT) also belong to this category.

The next category - i.e. scenario evaluation - is dependent on hardware (e.g. the camera), and on an application and its policy. The latter defines procedures to be applied during the enrolment, the training and the testing in the system, e.g. the number of enrolment attempts, the number of test attempts before a rejection, the required level of quality of the template acquisition during the enrolment, frequency of template update, etc. This evaluation allows to extract characteristics which may be different from those extracted with technological evaluation alone because they could not be imagined before the scenario was defined (e.g. incompatibilities between some hardware). Finally the operational evaluation is similar to the scenario evaluation but is performed at the actual site with real users (instead of involving engineers only !).

It is very important to understand that all performance evaluations are data dependent. Therefore in our opinion it does not make sense to announce error rates without a link to the data used, or to compare directly results that were obtained using different data sets. Even when using the same data set it should be a best practice to use the same protocol, i.e. dividing all the individuals in same groups of clients, evaluation clients and impostors, since the results may potentially be influenced by this distribution (this may not be the case for very large databases).

In this thesis most of the tests will fall in the technological evaluation category. For this purpose a sufficiently large database will be used to collect informations about modifications of the algorithms. However the evaluation of application configurations with such databases will be limited since they are fixed.

In the case of face verification on a mobile device it is however necessary to perform scenario evaluations that take into account variations in lighting and in view-point that are generally not present in a standard database. Scenario evaluation were used to assess the effectiveness of algorithmic improvements by using small data sets presenting particular characteristics. In addition a real-time software demonstrator was built in order to show the functionality of the system in a given range of operating conditions. Again these results can not be compared directly to other published results but they can serve to extract useful qualitative information about e.g. weakness of an implementation.

2.7.1 Definitions

We define the *training set* as a collection of images used to build a representation (template) of clients, called genuine clients, whereas the *testing set* is a collection of images of the same genuine clients (but images are different from the training set) and impostors, used as *probes* [2-12].

A basic way to report tests realized with these probes is to plot a histogram of the distances, or probabilities (i.e. confidence), returned by the algorithm. By placing a threshold at a certain distance, or probability, it is possible to decide whether the client shall be accepted or rejected. Clearly if there is an overlap between the two distributions related to the client and impostor probes, there will be some misclassified samples, i.e. genuine probes with distance above the threshold (respectively confidence probability below the threshold) will be falsely rejected. Similarly impostor probes with distances below the threshold (respectively confidence probability above the threshold) will be falsely accepted (see also Section 4.6 Classifiers and decision theory).

The *false acceptance rate* (FAR) is defined as the number of impostor probes that are falsely classified as genuine clients - given a certain threshold - over the total number of impostor probes. Similarly the *false rejection rate* (FRR) is the number of genuine probes that are falsely classified as impostors over the total number of genuine probes. The FRR is sometimes called *false non-match rate* (FNMR) whereas the FAR gets alternatively called *false match rate* (FMR).

By varying the threshold, it is possible to draw the FAR and FRR responses in function of the threshold, out of which it is possible to establish the so-called *Receiver Operating Characteristic* (ROC) which is a parametric function representing the FAR vs. FRR. Although the latter

definition of the ROC response will be adopted in this report, it happens sometimes that the ROC actually refers to the parametric plot of the genuine acceptance rate (i.e. 1-FRR) vs. the FAR, and that the so-called *Detection Error Trade-off* (DET) curve denotes instead the parametric response of FAR vs. FRR.

The *equal error rate* (EER) is defined as the point on the ROC curve (corresponding to a certain threshold) where the FAR equals the FRR. It should be noted that a notion of "total error rate" would be dependent on the probability of impostor and genuine occurrences. In a real system however it is difficult to measure the probability of an impostor attempt if it is not detected.

Additionally it is possible to use two other error criteria i.e. the *failure to enrol rate* and the *failure to acquire rate*. The first describes the proportion of volunteers who failed to obtain an enrolment in the system using the defined policy. The second describes the proportion of transactions (either genuine or impostor tests) that could not be realized due to the same policy. This policy could be for example an image quality evaluation. There are chances that rejected images will improve the FAR and FRR at the cost of increased enrolment and acquisition failures. This policy is however generally application dependent. Consequently in this thesis no special policy was developed and all transactions will be considered leading to increased FAR and FRR.

Following the UK CESG best practices [2-12] the genuine and impostor scores shall be used directly, i.e. no model should be substituted for the FAR and FRR calculations of the ROC (DET) curve. This also means that no a posteriori normalization shall be made based on the histograms of the probes. However when evaluation data are available normalization of probe distances using this a priori information is allowed (see also Section 4.6).

2.7.2 Technology evaluation

Several databases are available to the research community for evaluating face recognition implementations. Table 2-1 contains a list of the most widely used databases, which are generally applied to compare algorithms in publications as well as to make performance evaluations internally to the research teams.

Name	Num. of persons	Total num. of images	Image size in pixels	Color/ gray	Lighting variations[a]	Pose (P) or expression (E) variation[b]	
Yale [2-22]	15	165	320 x 243	Gray	Yes	E	c
Olivetti [2-23]	40	400	92 x 112	Gray	No		d
MIT [2-24]	16	432	120 x 128	Gray			c
Weizmann Institute [2-25]	28	840	352 x 512	Gray	Yes	P+E	c
UMIST [2-26]	20	1'013	~220 x 200	Gray			e
BioID [2-27]	23	1'521	384 x 286	Gray			c
University of Stirling [2-28]		1'591	~275 x 350	Gray	Yes	P+E	f
University of Oulu [2-29]	125	~2000	428 x 569	Color	Yes (color)	No	g
XM2VTS [2-30]	295	2'360	720 x 576	Color	No	No	h
Purdue / Aleix Martinez (AR face db) [2-31]	126	>4000	768 x 576	Color	Yes	P+E	i
Yale B [2-6][2-32]	10	5'850	640 x 480	Gray	Yes	P	c
Banca [2-33]	208	6'240	720 x 576	Color	No	No	j
Essex University [2-34]	395	7'900	~200 x 200	Color	No	P+E	c

Table 2-1: List of available databases (sorted by increasing number of images).

Name	Num. of persons	Total num. of images	Image size in pixels	Color/ gray	Lighting variations[a]	Pose (P) or expression (E) variation[b]	
Color Feret [2-35]	994	11’338	512 x 768	Color			c
Feret [2-36]	1’204	14’051	256 x 384	Gray			c
University of Notre-Dame [2-37]	334	33’247	1600 x 1200 2272 x 1704	Color	Yes	E	k
CMU (Pie) [2-38]	68	41’368		Color	Yes	P+E	c

Table 2-1: List of available databases (sorted by increasing number of images).

a. Indicates whether the database was collected under constant illumination condition or under different illumination conditions (artificial or natural light).
b. Indicates whether the database contains pose or expression variations; a “-” indicates that the database is taken under almost constant pose and with neutral expression.
c. Publicly and freely available for non-commercial use.
d. Once publicly and freely available, it seems no longer available on the original website at the time of writing this report.
e. Rotations.
f. This database is divided into a number of databases, the three largest being: nothingham scans, aberdeen, and stirling_faces. All data sets are publicly and freely available for non-commercial use.
g. Camera calibration available.
h. Non-free (£100 for academic; £200 for industry).
i. Occlusions.
j. Non-free (£100 for academic; £200 for industry). The database is divided into three sets representing the following acquisition conditions: controlled, degraded, and adverse conditions.
k. Freely available.

In this thesis the following databases were used:

- *the extended M2VTS database labelled XM2VTS [2-14];*
- *the Yale B database [2-32][2-6].*

While the XM2VTS database is very interesting due to its large number of clients, the Yale B database is interesting to specifically test the robustness against variations of the illuminant direction, of the light intensity and color, and also against variations of the clients' pose.

a) XM2VTS database

The XM2VTS database provides 4 recording sessions of 295 candidates that were acquired over a period of 5 months. Therefore, changes in physical condition, hairstyle and expressions are included in the database. For our experiments we used 8 still images per person, which where extracted from video sequences by the database providers, leading to a total of 2360 images. The original color images were transformed into grayscale (luma) images, and reduced to a size of 320x256 pixels using low-pass filtering and decimation in Adobe Photoshop.

A standard protocol for person verification - the so-called Lausanne protocol (Figure 2-1) - was developed for this database [2-11], according to which the 295 persons were subdivided into three sets, corresponding to clients, to evaluation impostors, and to test impostors, respectively.

The client set is itself composed of three sets of images (training, evaluation and test). The subdivision is made according to two different schemes that are called *configurations*. The main difference between the two configurations comes from the choice of client images used in the training, the evaluation and the testing of the system, among the eight available (two per session) as described in Figure 2-1. In Configuration I three training samples per client are available, which are taken from Sessions 1, 2 and 3 (1 image per session), whereas four training samples are available in Configuration II, which are taken from Sessions 1 and 2 only (two images per session). In both configurations the test samples come from Session 4. In this thesis we mainly used Configuration II since it seems more representative of real-world applications where the enrolment is done in a limited period of time (2 sessions in Configuration II vs. 3 in Configuration I).

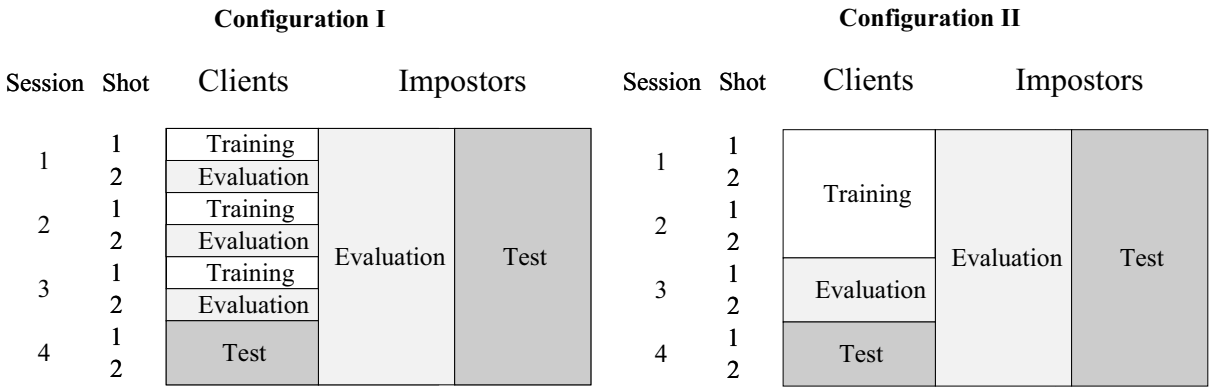


Figure 2-1: Configurations I and II of the Lausanne Protocol.

The evaluation set is mainly used for setting a class threshold or equivalently to normalize class distributions before using a uniform threshold (related details are furnished in Section 4.6).

The four training samples in Configuration II[6] can be used to build one or more templates for each client (this is indeed not defined in the protocol). It is possible to build a single template based on the 4 views, e.g. by averaging the four feature values extracted from the four training images, or better by using a weighted sum of these four feature values. However this does not seem to improve the performance of the system as 4 views are not sufficient for extracting the statistics necessary to the weighting.

On the other hand only one image can be used as a reference, e.g. shot 1 from session 1 (i.e. all the client images ending in "_1_1", e.g. image "003_1_1.png", "004_1_1.png", etc.). This image is then matched against all test images. This latter approach (see Figure 2-2b), which is used for most of the trials in this thesis, is called *single template* in the rest of this report.

If four different templates are extracted from the four training images, then each probe has to be matched against each template. In this case, the FRR will certainly be lower, because some variability of the reference class is obviously taken into account, but at the price of a possibly higher FAR.

Alternatively, only a subset of significantly different training samples could be used to extract additional reference templates (Figure 2-2c). A training template would thus be matched against the other existing cli-

6. The same training can be applied for the three training images available in Configuration I, but they are not explicitly discussed here because results for Configuration I are not reported.

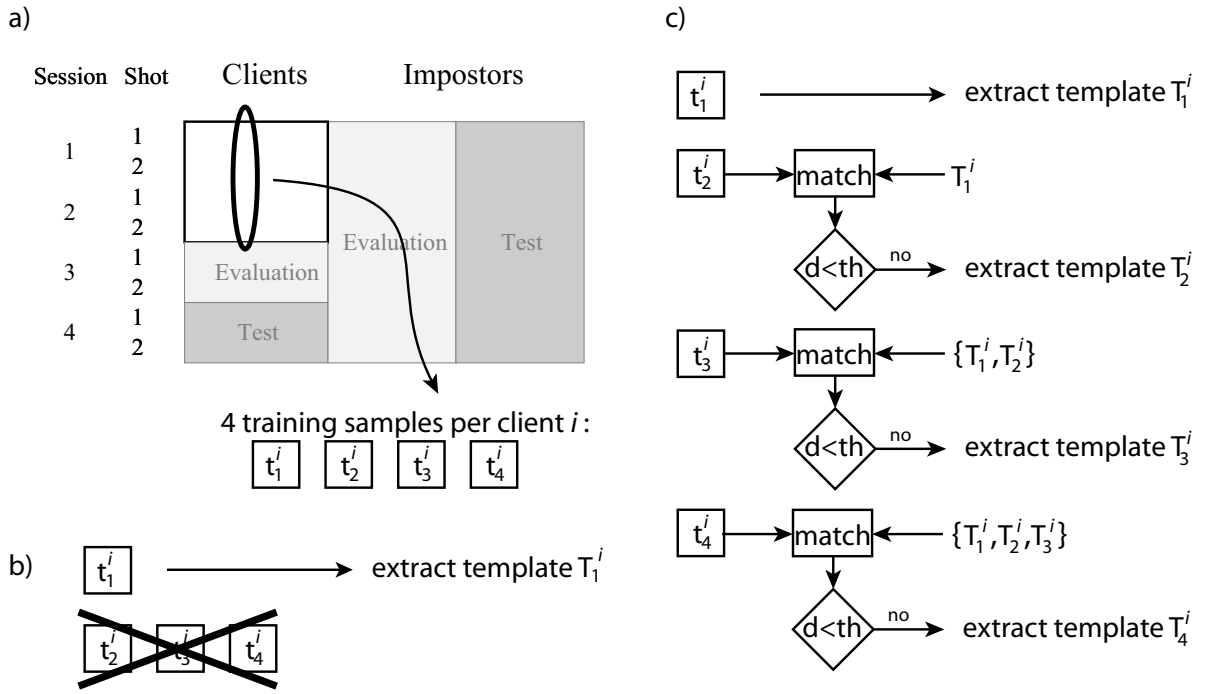


Figure 2-2: Single vs. multiple templates training.
a) Four training samples are extracted from the client training dataset; b) Single template approach: only the first sample is used to extract a reference template; c) Multiple templates approach: all the training samples are used to extract between 1 and 4 reference templates depending on matching distances (*d*: matching distance; *th*: pre-defined threshold depending on feature type).

ent templates and only those that lead to a large (i.e. larger than a predefined threshold) matching distance would be added to the reference template set for this client. This results in between 1 and 4 templates per client. In fact, a large distance between images in the training set might be due to variations introduced by e.g. glasses that shall be included in the template. This approach is called *multiple templates* in the rest of this report.

b) The Yale B database

This database contains 5760 images of 10 persons (i.e. 576 views for each person) corresponding to 9 poses and 45 illumination conditions. It is mainly used to report face identification results against variations in illuminant and pose rather than verification results due to the small number of enrolled clients. It shall be noted however that the variations of illumination concern only the direction (and so implicitly the intensity) of the illuminant, and not the color. Still, the illuminant color is an important issue as we will show it in Subection 4.2.1.

The Yale B database is divided in four subsets characterized by illuminant conditions getting increasingly severe with the subset index. Subset 1 contains 7 illumination conditions, Subset 2 contains 12, Subset 3 contains 12 and Subset 4 contains 14 illumination conditions. The last subset contains the worst illumination conditions, i.e. large shadows are covering the face region. In this thesis, comparisons with this database will be made by reusing the protocol defined by Georghiades [2-6]. The first experiment consists in enrolling the clients with Subset 1 in frontal pose (7 images per client) and in performing a closed-set identification of the Subsets 2, 3 and 4 (38 images per client).

Another protocol is defined by Short et al. [2-19] who perform the authentication by using the ten frontally illuminated images (i.e. one image per person) to build the templates and who use the remaining 5750 images as test claims (i.e. 575 genuine claims persons).

2.7.3 Scenario evaluation

The technology evaluation described above allows mainly performance comparisons between variants of algorithm implementations. However the databases used (XM2VTS and Yale B) do not present a sufficiently large number of different image acquisition conditions (e.g. illuminant color variations) as necessary to validate the functionality of a selected algorithm for the foreseen mobile application. Consequently a real-time demonstrator was built within this project, allowing to simultaneously:

- prove the functionality of the algorithm operating under given scenario conditions;
- identify particularly difficult conditions requiring an increased robustness.

No ROC was however calculated for results obtained with this demonstrator. The setup was constituted of an off-the-shelf webcam connected to a laptop equipment (see Appendix A for a description of devices and drivers used).

The camera was software controlled using the Logitech QuickCam Software Development Kit. This allowed to grab images from the camera and also to control some parameters such as exposure time, analog gain, etc. However it was not possible to fine tune parameters such as white-balance, which were rather handled after image acquisition using a self-made software intervening as a separate pre-processing.

Experiments allowed to test the robustness with respect to:

- Lighting direction, intensity, spectral distribution;
- Scale;
- Expression and other intrinsic variations (hairs, etc.).

For example it could be concluded that Logitech's proprietary schemes for compensating illumination intensity and white balancing as they are implemented in the QuickCam drivers used (see Appendix A) were insufficient to remove the illumination dependency problems. Namely, when switching from artificial to natural illuminants (or reciprocally), the drivers could not compensate for the new illumination conditions and the newly acquired image was very often falsely rejected.

2.8 Conclusion

In this chapter, we defined the term biometrics together with its utilization for verification and identification. Several biometric modalities were described and face verification was selected for low-power mobile applications due to its low intrusiveness and high ease of use. The rise in security level - though it is not as high as for other modalities - seems relevant considering the commodity added to the device.

However, the handling of the camera, and thus the image acquisition, is not very well constrained. Secondly, the mobile nature of the system will introduce large variations on illuminant and background. Consequently, the matching method requires an increased robustness as compared with a fixed system. In addition to robustness against facial features variations (intrinsic variations), the system will have to compensate for larger extrinsic variations:

- *Viewpoint:* the user holds the camera with a possible tilt (both in-plane and out-of-plane).
- *Scale:* the distance between user and camera will be variable (a factor of $\pm 50\%$ seems reasonable).
- *Illumination:* the user can employ the system under various illumination conditions such as indoor or outdoor, near a window or a lamp, etc.

A basic requirement is that the user should be capable of taking an image of the scene that contains his own face. Intrinsic variations must

be handled to some extent using statistical models of appearance variations.

The state-of-art regarding algorithm robustness will be presented in Chapter 3 and algorithmic improvements will be proposed in Chapter 4.

References

[2-1] J. Ashbourn, "Biometrics, Advanced Identity Verification: The Complete Guide," *Springer Verlag*, London, UK, 2000.

[2-2] Biometric Consortium, http://www.biometrics.org/html/introduction.html

[2-3] D. M. Blackburn, "Evaluating Technology Properly: Three Easy Steps to Success," *Corrections Today*, July 2001, pp. 56-60.

[2-4] N. L. Clarke, S. M. Furnell, P. M. Rodwell, and P. L. Reynolds, "Acceptance of subscriber authentication methods for mobile telephony devices," *Computers & Security*, Vol. 21, No.3, 2002, pp. 220-228.

[2-5] J. G. Daugman, "Biometric personal identification system based on iris analysis," *US Patent No. US 5291560*, 1994.

[2-6] A. S. Georghiades, P. N. Belhumeur, and D. J. Kriegman , "From Few to Many: Illumination Cone Models for Face Recognition under Variable Lighting and Pose," *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. 23, No. 6, June 2001, pp. 643-660.

[2-7] International Biometric Society, http://www.tibs.org/DefinitionofBiometrics.htm.

[2-8] A. K. Jain, A. Ross, and S. Prabhakar, "Fingerprint Matching Using Minutiae and Texture Features," in *Proc. Int'l Conf. on Image Processing (ICIP'01)*, Greece, Oct. 7-10, 2001, pp. 282-285.

[2-9] A. K. Jain and A. Ross, "Multibiometric Systems," *Communications of the ACM*, Vol. 47, No. 1, Jan. 2004, pp. 34-40.

[2-10] J. Kittler, M. Hatef, R. P. W. Duin, and J. Matas, "On Combining Classifiers," *Trans. Pattern Analysys and Machine Intelligence*, Vol. 20, No. 3, Mar. 1998, pp. 226-239.

[2-11] J. Luettin, and G. Maître, "Evaluation Protocol for the extended M2VTS Database (XM2VTSDB)," *IDIAP-Com 98-05*, 1998.

[2-12] A. J. Mansfield, and J. L. Wayman, "Best Practices in Testing and Reporting Performance of Biometric Devices," *NPL Report CMSC*, Version 2.01, Aug. 2002, available from: http://www.cesg.gov.uk.

[2-13] T. Matsumoto, "Gummy and Conductive Silicone Rubber Fingers," *8th Int'l Conf. on the Theory and Application of Cryptology and Information Security (ASIACRYPT'02)*, Queenstown, New Zealand, Dec. 1-5, 2002, pp. 574-576.

[2-14] K. Messer, J. Matas, J. Kittler, J. Luettin, and G. Maître, "XM2VTSDB: the Extended M2VTS database," in *Proc. 2nd Int'l Conf. on Audio- and Video-based Biometric Person Authentication (AVBPA'99)*, Washington DC, 1999, pp. 72-77.

[2-15] M. Most, "Metrics - Technology Assessment Guide," *Biometrics Market Intelligence*, Vol. 1, Issue 1, Feb. 2002, pp. 8-9.

[2-16] M. Most, Special Issue on Embedded Mobile Biometrics, *Biometrics Market Intelligence*, Vol. 1, Issue 8, Sep. 2002.

[2-17] P. J. Phillips, A. Martin, C. L. Wilson, and M. Przybocki, "An Introduction to Evaluating Biometric Systems," *IEEE Computer*, Vol. 33, No. 2, Feb. 2000, pp. 56-63.

[2-18] A. Ross and A. K. Jain, "Information Fusion in Biometrics," *Pattern Recognition Letters*, Vol. 24, No. 13, Sep. 2003, pp. 2115-2125.

[2-19] J. Short, J. Kittler, and K. Messer, "A Comparison of Photometric Normalization Algorithms for Face Verification", in *Proc. 6th IEEE Int'l Conf. on Automatic Face and Gesture Recognition (FGR'04)*, Seoul, Korea, May 17-19, 2004, pp. 254-259.

[2-20] X. Xia, and L. O'Gorman, "Innovations in fingerprint capture devices," *Pattern Recognition*, Vol. 36, No. 2, Feb. 2003, pp. 361-369.

[2-21] http://www.iris-recognition.org/commercial_systems.htm#patents

[2-22] http://cvc.yale.edu/projects/yalefaces/yalefaces.html

[2-23] http://mambo.ucsc.edu/psl/olivetti.html

[2-24] http://vismod.media.mit.edu/pub/images/

[2-25] ftp://ftp.wisdom.weizmann.ac.il/pub/FaceBase/

[2-26] http://images.ee.umist.ac.uk/danny/database.html

[2-27] http://www.humanscan.de/support/downloads/facedb.php

[2-28] http://pics.psych.stir.ac.uk/

[2-29] http://www.ee.oulu.fi/research/imag/color/pbfd.html

[2-30] http://www.ee.surrey.ac.uk/Research/VSSP/xm2vtsdb/

[2-31] http://rvl1.ecn.purdue.edu/~aleix/aleix_face_DB.html

[2-32] http://cvc.yale.edu/projects/yalefacesB/yalefacesB.html

[2-33] http://www.ee.surrey.ac.uk/banca/

[2-34] http://cswww.essex.ac.uk/mv/allfaces/faces94.html

[2-35] http://www.itl.nist.gov/iad/humanid/colorferet/home.html

[2-36] http://www.itl.nist.gov/iad/humanid/feret/feret_master.html

[2-37] http://www.nd.edu/~cvrl/UNDBiometricsDatabase.html

[2-38] http://www.ri.cmu.edu/projects/project_418.html

Chapter 3

State-of-art in Face Recognition

This chapter provides an overview of the state-of-art in face recognition, with the aim of selecting an appropriate category of methods fitting the specificities of mobile applications. For this purporse, a list of algorithm classification criteria will be established based on cognitive and computational considerations.

Firstly in Section 3.2, face recognition will be considered from a cognitive point of view, i.e. by analyzing how the human brain is able to recognize faces of familiar people. Some interesting properties of the human cognitive process will help to differentiating the possible approaches of the computational process. In Section 3.3, a list of criteria for classifying existing approaches will be given, while a large part of this chapter will be devoted to the non-exhaustive survey of face authentication algorithms (Section 3.4). Based on the discussed criteria, an algorithm will be selected in Section 3.5, which fulfils the requirements of mobile face verification. The optimization for mobile devices and environments of the selected algorithm will be then discussed in Chapter 4. Finally, a survey of consortia, available commercial products and patents will be given in Section 3.6 to complement the review of academic research. The whole information aims at sketching current trends in biometric development.

3.1 Introduction

Research in face verification truly started in the early 1970's but benefited from a regain of interest starting from 1995 and more recently due to the extended security demand following the events of September 11, 2001. It is therefore difficult to provide an exhaustive overview of all the current research undertaken in the field since the late 1980's. This chapter will however contribute in trying to sketch major trends and open issues in this domain.

In the 1990's as the major conferences explicitly dedicated to biometrics appeared[1] and the number of biometric-related articles in general purpose pattern recognition journals increased, several authors wrote surveys on face recognition from a computational point of view [3-18][3-32][3-33][3-86][3-99]. The most recent survey is the one made by Zhao et al. [3-106]. Additionally, several cognitive science publications sketched the recent advances in neuropsychology, where the face cognition process had always been of great interest.

At the same time, some authors comparatively implemented different methods and tested them on limited sets of galleries (e.g. [3-42][3-73][3-80][3-105], see also Colorado State University's project [3-109]). When published, these results allow the community to draw "some" conclusions but the lack of systematic tests on the same data set - i.e. each algorithm being optimized, instead of one optimized version vs. non-optimized ones - is still obvious. Recently, the National Institute of Standards (NIST) conducted several large tests of commercial and academic software[2] in order to evaluate the state-of-art in face recognition technology. This event was organized in 2000 and 2002. Results will be briefly discussed in Section 3.6. Databases used during the FRVT events were however not fully disclosed for academic testing.

3.2 From cognitive to computer science

In this section we will start by giving a short overview of latest advances in cognitive science aiming at understanding some basic concepts that will be useful for the design and classification of computational methods. Then, some classification criteria spanning the space of available algorithms will be defined. By listing major algorithms published from the late 70's and sketching their position in this space, we will be able to select a class of face verification algorithms based on the requirements of mobile applications.

3.2.1 Face recognition: a cognitive process

The process of face analysis by humans is very well developed and carries important social informations (e.g. it is used during communication, to evaluate the mood of a person or even to determine the gender or age of a person). Indeed humans are very effective in recognizing fac-

1. The pioneering conferences were definitely the Workshop on Face and Gesture Recognition (FG) and the Conference on Audio- and Video-based Biometric Person Authentication (AVBPA) launched in 1995 and 1997 respectively in Switzerland (Zürich and Crans-Montana).
2. Face Recognition Vendor Test (FRVT), http://www.frvt.org.

es from far, under varying viewpoints, in cluttered scenes, with varying expressions or even under varying illuminants [3-13]. These capabilities would be highly desirable in an automated system.

Cognitive science concepts involve both some *physiological* information (i.e. how the brain physically and chemically works) and *psychological* information (i.e. how the brain interprets these stimuli). These cognitive informations will possibly give us clues on how to improve the face recognition process, and they also help us in classifying face recognition algorithms. It is definitely not proven however that mimicking the brain behaviour leads to the best implementation of face recognition from a computer point of view.

a) Basics in neurophysiology

We will not extend too much on the retinal image formation e.g. involving photo-pigment receptors (rods and cones). For this information the reader is directed e.g. to Uttal et al. [3-98]. We will rather focus on the several levels of neural cells, which are responsible for extracting low-level characteristics from the image that can be later used by the brain to build higher-level models.

Ganglion neural cells existing in the retina extract both spatial patterns (differences of intensity across an image) and temporal information (differences occurring in time) due to their synaptic interconnections. These first patterns will be extracted through a process called *lateral inhibition* enhancing differences of contrast in the scene. Moreover depending on the size of the receptive field of these cells, finer or coarser spatial details will be selected. The retina consequently acts as a set of spatial filters effectively achieving a kind of subband processing of the image. These stimuli are then transmitted out of the eye to the brain through the optical nerve.

Neurons in the visual cortex (striate cortex) present the interesting feature of being selective to orientation [3-50]. The combined effect of ganglion and cortical neurons provide an encoding of the visual information which is highly optimal [3-107]. Indeed we could compare it to a kind of feature detector, which is involved in most pattern recognition processes. Gabor filter banks, which are largely used in pattern recognition (see Subsection 4.3.1), model quite adequately the effect of the cortical neurons [3-50][3-94].

This encoded information is then processed in different regions of the brain depending on the higher-level tasks associated to face analysis. This clearly means that certain regions of the brain are specialized in definite tasks. For example spatial perception is distributed in an extended region of the parietal and frontal lobes, while face perception is mainly involving occipital and temporal regions, especially near to the mid line of the brain [3-13].

Due to the presence of two eyes furnishing slightly different images, the brain is able to build a unified image and to extract information on the depth of the scene (this process is called stereopsis [3-13]). Motion information is also used to infer models of the 3D shape of an object since the motion will appear different for one eye than for the other. However, this twofold information on the depth and on the motion is not available in a 2D representation (e.g. photographs) of a face seen by the eyes, though the brain is still able to recognise the face. This could indicate that 3D information is not necessary for face recognition, but rather helps for compensating changes in lighting and view angle. Capability of reconstructing 3D information from shades in a 2D image is called *shape from shading* and was proposed by several authors (see Subsection 3.4.6 for some references).

Before considering the neuropsychology of the face interpretation process, we can summarize the adaptation of the human visual system to the task of facial recognition. Firstly, it is observed that multiresolution, coarse-to-fine information is present and probably necessary for face recognition. An indication supporting this assertion is that familiar faces can already be recognized from a coarse (i.e. “blurred”) image and that edges (in the extreme case: line drawings) are generally not sufficient to recognize someone although they help for finely separating different individuals. Additionally coarse-scale information gives information on the illumination context (shading) to apply “corrections”. Secondly, orientation information is present in the visual cortex and allows to extract patterns that are subsequently used by higher-level recognition mechanisms. It can be assumed that this combination of multiscale and orientation information leads to the robustness of human processed face recognition, even under a wide range of illuminants.

b) Basics in neuropsychology

All faces are essentially similar but are characterized by small distinctions that allow humans to discriminate them. How, from the inputs described before, is the brain able to perform this operation? Due to the complexity of the brain, current theories on this topic are still evolving. The first psychological evidences were studied from patients with brain injuries who lost some faculties. For example, prosopagnosia (the loss of the faculty to recognize faces but not to recognize other objects [3-13]) could be a good indicator that the face recognition process is using faculties different from those applied to recognize the make of a car for instance. Similarly, studies by Haselmo et al. reported by Farah [3-29] state that face-selective cells, in particular regions of the temporal cortex, are more sharply tuned to *emotional expression recognition* than to facial identity, whereas it is the opposite for inferior areas of the temporal cortex.

Starting from the hypothesis that different regions of the brain are used for the ability to recognize faces, while other regions are used for other objects, Farah concluded that facial recognition is somehow "special". Several experiments indeed showed the importance of the overall structure or *Gestalt* over explicit representation of the eyes, nose and mouth. Farah demonstrates that faces are recognized in their entirety concluding that "the whole face is more than the sum of its parts". This is indeed coherent with the early observations of Sir Galton [3-35] claiming that: "one small discordance overweights a multitude of similarities and suggests a general unlikeliness".

This property of a face being represented as a whole in the early cognition is called *holism*. Following Farah, although there could be a mix between holistic and feature (or *atomic*) representations, the latter is proportionally several orders of magnitude less important in face recognition. Also, it is generally believed that feature representations could be more useful in emotional expression recognition than in identity recognition, and that it would follow a top-down approach in which the part-based representation would follow the whole representation.

The holistic processing of the brain is also highlighted when considering the recognition of upside-down faces. Inverted faces are actually much harder to recognize than upright ones, even though the same features and configuration exist in both images [3-83]. If the configuration is slightly modified in both representations (e.g. the nose being higher)

the modification will be more visible in the upright than in the inverted face. In the extreme, if the whole face is inverted but facial parts (e.g. the eyes and the mouth) remain in an upright orientation the face appears “normal” when inverted but very strange in a normal orientation. This phenomenon is illustrated by the well-known “Thatcher illusion” published first by Thompson [3-95].

It is important to note that the holistic approach is however somehow different from other frameworks such as the *connectionist* model[3], although not incompatible with them. The distinction between connectionist [3-99] and non-connectionist models is mainly based on the parallel vs. serial processing of the image and on the memory storage (which is either distributed in the connectionnist approach or local/centralized in the non-connectionist approach). In the holistic model, no assumption is made about the processing or storage of templates. This means that algorithms processing sequentially a centrally represented face template can still be qualified as holistic but not as connectionist.

A second important parameter of the connectionist model is the dependence on *configurational* properties. Facial features (eye, nose, chin, etc.) can be labelled as *first order* features, while the configurational properties (spatial relations among first order features; face shape) represent *second order* features. Clearly, the first and second order features are necessary for face recognition and are in many ways holistic, but the holistic approach makes no explicit distinction between the two types of properties.

Bülthoff et al. [3-15] compared three different conjectures about the recognition of three-dimensional objects by the brain, namely:

- a 3D model is reconstructed from perceived scenes and this model is matched against a memorized 3D model;
- 3D models are used to synthesize (i.e. project) 2D images that are compared to the perceived images;
- several two-dimensional viewpoint-specific representations are stored and new 2D views are compared to the stored views using interpolations.

Similarly to the first and second order categories of above mentioned features, Bülthoff distinguishes *entry-level* from *subordinate-level* cat-

3. Additionally, there is sometimes a confusion made between holistic and connectionist models (e.g. in Gutta et al. [3-42]).

egories of objects, the first being used to group objects having a similar overall shape and spatial relationships (e.g. cars on the one hand, birds on the other hand, or faces, etc.) while the latter serves to discriminate two objects belonging to an entry-level category (e.g. two different faces). Bülthoff observes that entry-level classification is generally viewpoint-independent and is accordingly using a 3D model (first two conjectures), whereas subordinate classification (and thus face recognition) is generally viewpoint-dependent i.e. using multiple 2D views (third conjecture). Although it may be possible that entry-level classification of familiar objects is also viewpoint-dependent due to extensive experience [3-15], it remains that identity determination as such seems clearly viewpoint-dependent and thus 2D-based. Experiments show that recognition of familiar faces is equally efficient from frontal views as from three-quarter views, confirming the viewpoint dependence [3-13].

Finally, human face recognition is shown to be virtually unaffected by a transformation reversing hue and maintaining the same lightness, while it is completely disrupted if the lightness is inverted (this is known as “negative” effect). But different color perceptions can carry social information such as signs of illness, or embarrassment [3-13]. A consequence is that an absolute hue is not necessary, and that only a relative variation (e.g. a relative lightness) is sufficient for face identity recognition. In Subsection 4.2.1 we will illustrate that a robust relative measure of lightness is however often hard to obtain under varying types of illuminants.

3.2.2 Face recognition: a computational process

By definition, the ultimate goal of face verification is to extract physiological features, which depend on unobservable variables, through physical observations (i.e. patterns). Models will be built (deterministically and/or stochastically) from these patterns in order to infer an observed object to a given category. It should be noted that the dimensionality of the observable patterns, or variables, is usually several orders of magnitude smaller than the dimensionality of the original unobservable variables themselves, due to the limitations of the acquisition setup (e.g. image formation on a digital CMOS sensor). This may be a concern, because, although we can claim that some biometric features are unique among individuals (e.g. fingerprints [3-74]), their observable patterns or the feature vectors extracted from them might not be so! This observation is a fortiori also verified for face verification.

As an example, a 2D image of a face acquired with a digital image sensor is the result of an interaction involving the face surface and a light source, and the interaction between photons and electrons in the sensor diode. The final image is an enormous simplification of the situation itself. The underlying physiological characteristic could be for instance the 3D shape of the head (which is itself depending on the skeleton and on the disposition of muscles and fat under the skin), the texture and spectral reflectance of the skin. These unobservable variables are interacting with extrinsic parameters, such as the illuminant spectral distribution, and could be depending upon other unobservable instrinsic factors such as the temperature of the body and environment, humidness, recent exposure to the sun, etc. It should become clear that it is not possible to model all these factors from a single observation nor would it be possible to realize a biometric face verification system based on them. Nevertheless, understanding some of these phenomena will help us to eliminate as much as possible extrinsic variations. For example in Subsection 4.2.1 and Subsection 4.3.4 we will show possible normalization methods for images or features to get a measure being as much as possible independent from the illuminant spectrum.

The observables, which can be considered as random variables [3-34] (or random vectors) can then be used to train the system. This can consist for example in estimating the distribution of the classes in the space spanned by these random vectors. A classifier (see Section 4.6) is a function taking as inputs a single observation and resulting in a class inference, generally based on statistical class estimations. It has to be noted that estimating class distributions can consist either in estimating parameters of a model, or in making no assumption at all on the form of the distribution (non-parametric estimation). Generally the number of random variables is reduced so that the classifier can be simplified (see Section 4.4). A novel set of random variables previously unseen by the system can finally be inferred to one of the trained classes. However if the probability of the new sample belonging to the inferred class is too small – or otherwise stated if the distance to the inferred class is too large – the new sample might be rejected.

3.3 Classification of computational methods

3.3.1 Determination of algorithm selection criteria

In order to help us in the selection of a class of algorithms for face verification on mobile devices, selection criteria must first be set up. For instance, a complex list of criteria is given by Zhao et al. [3-106]. We will however classify the most popular algorithms using the following computational and/or cognitive criteria:

a) Holistic vs. feature-based methods;

b) Rigid vs. deformable templates;

c) Problem-specific vs. untuned image features;

d) Global vs. local image features;

e) Bottom-up vs. top-down methods;

f) Two- vs. three-dimensional methods;

g) Implementation specific characteristics.

A more precise definition of these criteria is given in subsections below, followed by a non-exhaustive review of published algorithms, which shall be examined in light of criteria a) to g). Finally, Section 3.5 will summarize the advantages and inconvenients of all methods with respect to criteria a) to c) – the most important criteria for mobile face authentication systems – in the form of a Vennes diagram (Figure 3-5).

a) Holistic vs. feature-based methods

The famous work of Brunelli and Poggio [3-14] distinguishes the use of individual facial features like eyes, eyebrows, mouth, nose or chin, from the use of global face templates. This corresponds to the cognitive distinction between feature-based and holistic representations, as already mentioned by Farah [3-29].

In feature-based methods the relative position, orientation and size of the facial features constitute a *feature vector* of random variables, which is then used as input to a general classifier (e.g. a Bayesian classifier or neural network). Consequently, feature-based methods consist mainly in precisely locating facial landmarks for extracting useful and possibly robust characteristics. By contrast, template-based methods extract a whole image of the face and feed it to a comparison mechanism (e.g. a simple correlation). Note that automatically extracting a

region of interest (i.e. the template) from an image containing a face requires the detection of the said face. This "template matching" operation is either done by scanning the image with the reference template and calculating the matching distance at all positions, or by locating special facial landmarks. The latter case is consequently a mix of template- and feature-based methods although it is often considered as template-based.

b) Rigid vs. deformable templates

This distinction has an important impact on the robustness to viewpoint variations. True deformable approaches are reported to perform better than rigid ones when faces slightly turn away from the camera [3-105]. In this case 3D objects still project on a 2D image the same patterns or first order properties with slightly different configurational (i.e. second order) properties. For larger out-of-plane rotations however, the projection of 3D objects produces completely different patterns, which can no longer be handled by simple deformations. This is especially true for salient facial features such as the nose or the ears.

c) Problem-specific vs. untuned image features

This distinction especially concerns the low-level pattern extraction rather than the higher level classification, which is anyway tuned to the application. Cottrell [3-23] distinguishes general pattern extraction techniques – i.e. techniques which are used for other pattern recognition problems up to a difference of parameters – from problem specific techniques that are learned based on a training data set. Generally the selection of parameters is not trained but is chosen based on the application (e.g. choosing parameters of Gabor filters).

The impact of one method or the other is not clear on the performance. However we believe that the generalization of trained patterns to a new data set could cause more problems than an untuned version depending on the quality of the training set. Consequently, untuned characteristics shall be preferred.

d) Global vs. local image features

Following Cottrell et al. [3-23] templates extracted from an image can be described as global when the spatial extent of the features covers the whole face and local when they cover only a small subregion. Some features are however lying in between e.g. when the information is extracted with a relatively spatially limited operator, e.g. Gabor filters[4]

sampled at discrete locations, but that it is possible to reconstruct or at least to approximate an entire image [3-82].

Local features have the advantage of being more robust to noise created for example by variations of expression or occlusion. Furthermore, they are definitely required in deformable models as it is more difficult to distort a global representation.

e) Bottom-up vs. top-down methods

Hallinan et al. [3-43] distinguish “bottom-up” from “top-down” approaches. The first is also called “feed-forward” while the second is commonly called a “feedback” or an “analysis by synthesis” approach. Whereas significant patterns are extracted and classified in bottom-up approaches, the input signal is analyzed and a new version is synthesized using the inferred model in top-down approaches. In the case of three-dimensional representations of objects methods falling into the top-down category consist in inferring parameters of a 3D objects from a 2D view, projecting a new (synthesized) 2D view, and then comparing the result of this projection to the original view to compare the exactitude of the parameters. In other words inverting the image formation allows to determine which object is most likely to have generated the image.

In the 2D case we will see that some methods allow to reconstruct an image from the pattern extracted in the original view. E.g. a projection of the 2D view on a subspace can be used to recreate an approximated view (cf. 3.4.3 Eigenface-based methods). This synthesized view can eventually be compared to the original view to measure how close the model represents the actual image. This reconstruction is generally possible with global image features (as defined in criterion d) above), but sometimes also with local features (e.g. when using a decomposition with local filters or wavelets).

This possibility is often helpful to overcome occlusion problems where only a small region of the face is different (e.g. due to the presence of sunglasses) from the reference: the global distance of the test template to the reference will lead to a rejection. But a synthesized version would show that only the eye region is actually disrupting the classification while the rest of the face is relatively close to the test template. It can be an advantage also to get rid of expression effects assuming

4. Gabor filters have theoretically an infinite impulse response but they are in practice windowed.

that only the identity is of interest by synthesizing an "expression free" version of the observed image.

However the fact that the image of ones face or fingerprint may be synthesized from a stored template may be a concern for privacy (see Section 2.6) because a stolen template may be used to recreate a useful representation, or for security.

f) Two- vs. three-dimensional methods

It seems quite immediate that 3D representations carry more information than 2D representations, and this could be potentially used by recognition algorithms. Before the apparition of 3D sensors, early attempts tried to generate 3D representations from 2D views (e.g. Kanade [3-52]) in order to build view-point invariant methods.

A list of available methods to infer a 3D model can be found in Uttal [3-98]. The 3D representation may possibly be used for a direct comparison of "meshes" (or wireframe representations) of 3D models, or for the synthesis of a novel 2D view for comparison purpose (i.e. the above mentioned synthesis/analysis method).

Profile view based methods should not be considered as a 3D method since they are not using a 3D representation anywhere in their dataflow.

Finally, as we saw in Subsection 3.2.1, the brain does not seem to infer 3D representations of faces, although depth information might be used for lighting compensations or other secondary tasks.

g) Implementation specific characteristics

This general category regroups several characteristics that are more related to a specific implementation than to a particular algorithm. It is therefore difficult to add these characteristics in the general comparison framework but they can however help in finely separating different implementations.

The first categorization is based on the spectral information contained in the image, i.e. whether the implementation is using grayscale, color or whether it is based on other multispectral representations of images. Special spectral bands, e.g. infra-red, could be used but they require dedicated equipment and do not allow comparing performance with other methods relying on standard databases (see also Section 2.7).

The use of a single view or of multiple views per client in the training phase is obviously linked to the 2D/3D and the rigid/deformable classification criteria. Firstly, because recognition of 3D objects may potentially need several simultaneously acquired 2D views [3-97] or because it will again need some special equipment (e.g. laser range scanner, stereo cameras) or special lighting conditions (e.g. for "shape from shading" methods). Secondly, because methods not handling specifically deformation can compensate this by keeping several views of a single face in their database. Even when compensating for deformations it may be necessary for certain clients to acquire several views to take into account varying situations (e.g. presence/absence of beard or glasses, etc.). Generally this improvement is not directly depending on the chosen method but can be applied to all of them.

Whereas almost all methods are based on static images some authors propose to use video streams, which can be either used as a special case of multi-view recognition with very high temporal correlation, or can benefit explicitly from spatio-temporal correlation [3-106]. In addition to face detection using motion clues, it is theoretically possible to enhance the resolution of a particular region using so-called super-resolution techniques [3-9]. We might assume that all the still-image methods described here could benefit from video-streams and as a result this characteristic does not constitute a major classification criterion in this discussion. Again no standard database seems to exist for testing such methods.

3.4 Non-exhaustive review of algorithms for face recognition

Bearing the above criteria in mind, we will now discuss several implementation variants based on five major classes of algorithms discussed in the next subsections, namely:

- Feature-based methods;
- Template-based methods;
- Eigenface-based methods;
- Local feature analysis methods;
- Graph-based methods.

A further class of algorithms to be discussed is grouping more exotic implementations that are not directly linked to the five above.

Each of these widely used approaches can be subdivided in a multitude of different flavours. We will now review several possible variants belonging to these categories in light of the previously defined criteria.

3.4.1 Feature-based methods

The early attempts of Alphonse Bertillon (1853-1914) to recognize criminals based on statistical evidences, like measurement of the head and face, hands and arms, probably inspired the first researchers on face recognition. Indeed, the use of facial features for recognition is rather obvious, still some facial features seem to bear more importance with respect to their discriminating capability. The work of Goldstein, Harmon and Lesk [3-36] is a representative example of this belief. In their work they started from a set of 34 features including properties (e.g. size, relative position, thickness, texture, opening, and so on) of the hair, eyebrows, eyes, nose, mouth, chin, ears, cheeks and forehead, and reduced it to a smaller set of 22 features showing that a large amount of correlation existed among several initial features. It is interesting to note that this study was carried out based on the subjective perceptual evaluation of human "jurors" that was then fed to a computer program for classification. There was clearly no discussion of the automatic extraction of the above-mentioned facial features. The survey of facial feature-based methods by Samal et al. [3-86] lists contributions of several approaches, but points out that generally, to let them perform reasonably well, strong assumptions on the image quality need to be made. One of these assumptions is that the image is sufficiently good (with respect to lighting, viewpoint, and occlusion by facial hair, glasses or scars) to easily detect the facial features, although they don't give clues on *how* to perform this operation. The fundamental article of Brunelli and Poggio [3-14] introduces some basic clues on the detection of facial features. In their description, prior normalization needs to be carried out to remove influence of illumination gradients. Then, the position of the eyes is located by template matching (robust cross-correlation) with an arbitrary set of templates. It has to be noted that further normalization is based on the distance between eyes, and that therefore a failure to detect correctly these features (e.g. due to glasses) will lead to a general failure of the algorithm. Other features such as location and size of nose, eyebrows and mouth are calculated by using integral projections.

Clearly, more robust methods are necessary. Starting from the fact that eye detection is a key issue in locating other facial features, several authors try to increase its robustness. E.g. Sirohey and Rosenfeld [3-88] compared two methods based on one hand on color information and anthropometric data, and relying on the other hand on linear and non-linear filtering. Without considering color-constancy problems and with a database of faces devoid of glasses, they achieved a performance of 90% of correct eye location on the Aberdeen subset of the Stirling database (see Subsection 2.7.2 for information on this database). A direct consequence is that feature-based identification methods would at most achieve 90% of correct identifications on this database when using this method assuming the classification is 100% correct once the fiducial points have been correctly detected. The expected performance of such methods on real-world images is thus relatively poor due to the variability of lighting conditions (see also Subsection 4.2.1, *Effects of illumination variations*).

Similarly, Han et al. [3-45] use the eye location as a preliminary step to face detection based on heuristic matching rules. Haro [3-46] uses a specific equipment, namely infra-red lighting, to benefit from the so-called “red-eyes” physiological phenomenon. Although effective, the application of this method is limited due to the special lighting required.

Using an eigenface approach (see Subsection 3.4.3 below), facial features can also be detected by means of special feature templates called eigenfeatures [3-79] and more specifically “eigeneye”, “eigenmouth” and “eigennose”. However the lack of holistic information generally results in unsatisfying detection rates.

An interesting category of methods originated from the work of Cootes et al. [3-20][3-21]. Active shape models (ASM) can be applied to several domains involving deformable objects, including faces and medical images. ASM are built on a principal component analysis of aligned model coordinates, and they represent thus the main motion modes of point models. When placing these points at precise facial features location (landmarks), it is possible to find a matching pattern by statistically allowing deformations. The degree of matching (or distance) can be measured by extracting information from the pixel underlying a point, e.g. grey-levels of the image [3-21]. By combining models under different viewpoints [3-22] it is also possible to predict novel views. This method can be used either to locate facial features in order to build “shape-free” representations [3-58] (see also Subsection 3.4.3 for an application in

the context of eigenfaces), or to recognize faces directly by using the matching distance [3-28]. In this sense this method is relatively similar to the graph matching approach that will be described in Subsection 3.4.5. One drawback of the ASM method is that for the iterative method to converge the model needs to be placed closely to the target matching position, so that this operation may need a prior detection or template matching phase (e.g. [3-3]).

Many other methods were proposed (e.g. [3-17][3-40][3-49][3-87][3-108]), also in relation with the problem of feature tracking in human-computer interfaces [3-72]. Still, the lack of robustness needs quite always to be compensated either by special equipment and/or by special guaranteed operating conditions. Finally, the detection of facial features is often related to the detection of the face itself. A review of face detection methods can be found in the paper by Yang et al. [3-102].

In conclusion, the detection of features is definitely very sensitive to the image acquisition. A false detection of an important facial feature generally leads to the rejection of the client unless a very complex heuristic process is used to detect such situations. Moreover even with comparative tests like those made by Goldstein et al. the selection of a starting set of facial features is rather subjective and the reduction to a smaller set is highly depending on the ultimate classification method. Based on these observations, we conclude that it is better to start from a holistic set of local features *not* related to facial features, and to select an appropriate reduced number of them based on statistical properties of face images, instead of achieving an arbitrary pre-selection by hand.

Even if the features are ultimately correctly detected, the pose variation will also have a high impact on the relative positions of the features due to the perspective, although their relative size would remain fairly stable. In order to compensate for this effect, an explicit pose estimation would thus be necessary.

Finally, occlusions (or even the presence/absence of glasses) are not well handled by this kind of methods, since they complicate the detection of possible key features. Regarding the classification introduced in Section 3.3, feature-based methods are typically using local image features, and are top-down and two-dimensional methods with a limited amount of deformation compensation. Feature-based methods may use either specific or untuned image features (i.e. the c) criterion) depend-

ing on whether trained filters are used to detect parts of the face (e.g. the trained “eigenfeatures” are specific).

3.4.2 Template-based methods

Template matching is a widely used technique basically consisting in the calculation of a correlation between a test image and a template of the reference. The template can represent the whole face [3-14] or parts of the face. But it is clear that a single template will not separate the global shape of the face from local texture [3-37]. For this reason, Fischler and Elschlager [3-31] proposed to consider objects as collections of components and to use an *embedding metric* to solve the problem of registration between two images. In their general framework, rigid templates (components) are connected by a number of springs, which serve both to constrain the movements of the templates and to measure a deformation penalty. Note that this spring model is devoid of any statistical model. It has to be stressed that this approach, although combining global and local information, is still linked to facial landmarks. It is otherwise very close to the Elastic Graph Matching approach (see Subsection 3.4.5).

Yuille [3-104] uses the same approach with templates that are less sensitive to illumination variations and that allow faster matching convergence. Moreover the templates themselves can be deformed to better fit the test image. Still these templates are related to facial features and thus, in one sense, this approach is also feature-based.

This direct form of image matching can be broadened to the matching of transformed images (considering the application of the edge map [3-104], or of the discrete cosine transform, the Fourier transform, the Gabor transform, etc.) or to vector-fields representing the image or part of the image (e.g. using Principal Component Analysis, Linear Discriminant Analysis, etc.), rather than directly the pixel values. The principal goal of these variants is generally to increase the robustness to variations of light conditions and of face expressions. Finally, multiple (possibly global) templates can be stored for each database entry representing different expressions or viewpoints (as discussed in Subsection 3.3.1 under item g).

Direct template extraction is conceptually simpler than facial features detection. Even if at first this method seems to require a higher computational power – because it requires computation of a transform of the whole image and of large correlations – it can appear advantageous on

some platforms due to its high regularity and intrinsic parallelism, when compared to complex and irregular methods for feature extraction and heuristic classification methods.

However, if template extraction requires prior detection of facial features, as it is sometimes proposed in the literature, the advantage of the robust holistic approach is clearly lost! For this reason, the scanning process of the correlation with the whole template shall be maintained, possibly in conjunction with more elaborated minimization functions such as multiscale (pyramidal) correlation, iterative methods, etc.

3.4.3 Eigenface-based methods

The now famous work of Turk and Pentland [3-96] – probably one of the most widely used methods and possibly counting the largest number of variants – is based on an earlier work of Kirby and Sirovich [3-89][3-51] who constructed a low-dimensional representation of images of human faces using the Karhunen-Loève transform.

In the eigenface approach, N×N pixel images are represented as vectors, where each of the N^2 vector element represents one picture element, or pixel. This can be achieved e.g. by concatenating all pixel rows, but other organizations can be imagined as well. Anyway, the eigenface approach does not take into account correlation between adjacent pixels so that the choice of a particular pixel organization in the vector representation does not play a crucial role. All possible image vectors represent the image space, which is consequently very large. It is expected however, that faces only span a small subspace of the image space. This subspace is later called the *face space*.

From a set of training images comprising faces from different persons, Turk and Pentland derive a subspace spanned by the so-called 'eigenfaces' – i.e. the principal components – that minimizes the reconstruction error of the training samples in the least-square sense. It can be showed (see e.g. [3-96]) that the basis of this subspace is obtained by calculating the eigenvectors $\vec{u}_k$ of the symmetric non-negative covariance matrix **C** (Eq. 3.1) established from the image training samples using Eq. 3.2. As the size of the images is N×N pixels, the resulting covariance matrix is N^2 by N^2.

$$\boldsymbol{C}\vec{u}_k = \lambda_k \vec{u}_k \qquad (3.1)$$

$$\boldsymbol{C} = \frac{1}{M}\sum_{n=1}^{M}(\vec{\Gamma}_n - \vec{\psi})(\vec{\Gamma}_n - \vec{\psi})^T \qquad (3.2)$$

where M is the number of images in the training set, $\vec{\Gamma}_n$ is the N^2-size vector representing the n-th image, $\vec{\psi}$ is the N^2-size vector representing the average face estimated from the same training set, λ_k and the superscript index T denotes matrix transposition. This procedure leads to M-1 eigenvectors $\vec{u}_k$ (eigenfaces) of size N^2, and M-1 eigenvalues, due to the singularity of the matrix **C** when M<<N (all other eigenvalues are 0). The eigenvectors are complemented by the average face $\vec{\psi}$ (cf. Figure 3-1).

Since obtaining the eigenvectors of a N^2 by N^2 matrix is a practically intractable problem, Turk and Pentland proposed a solution where eigenvectors of a M by M matrix are calculated first, and from which the M-1 eigenvectors of matrix **C** can be obtained.

A new face image $\vec{\Gamma}$ is projected on this subspace by first subtracting the mean face $\vec{\psi}$, and then by multiplying the result with the eigenvectors obtained from Eq. 3.1 (i.e. projecting it on the subspace spanned by the eigenvectors), cf. Eq. 3.3.

$$\omega_k = \vec{u}_k^T \cdot \left(\vec{\Gamma} - \vec{\psi}\right) \quad \text{for k=1,...,M-1} \qquad (3.3)$$

Projected face images of different persons should occupy different subspace regions of the generated face subspace, because the face images of different persons were taken in the training set and that finding the eigenvectors of the sample covariance matrix leads to a vector basis that maximizes variance across the different samples.

Equivalently it is possible to reconstruct a view of an enrolled person as a linear combination of the eigenfaces (Eq. 3.4).

$$\vec{\Gamma}' = \vec{\psi} + \sum_k \omega_k \vec{u}_k \qquad (3.4)$$

The possible face space dimensionality reduction (i.e. taking only the first K eigenfaces, K<M, corresponding to the highest eigenvalues instead of M vectors) thus allows a shortening of the feature vector, which is still sufficient for face recognition but not for reconstruction. Indeed, the reconstruction is error-free only for a face image present in

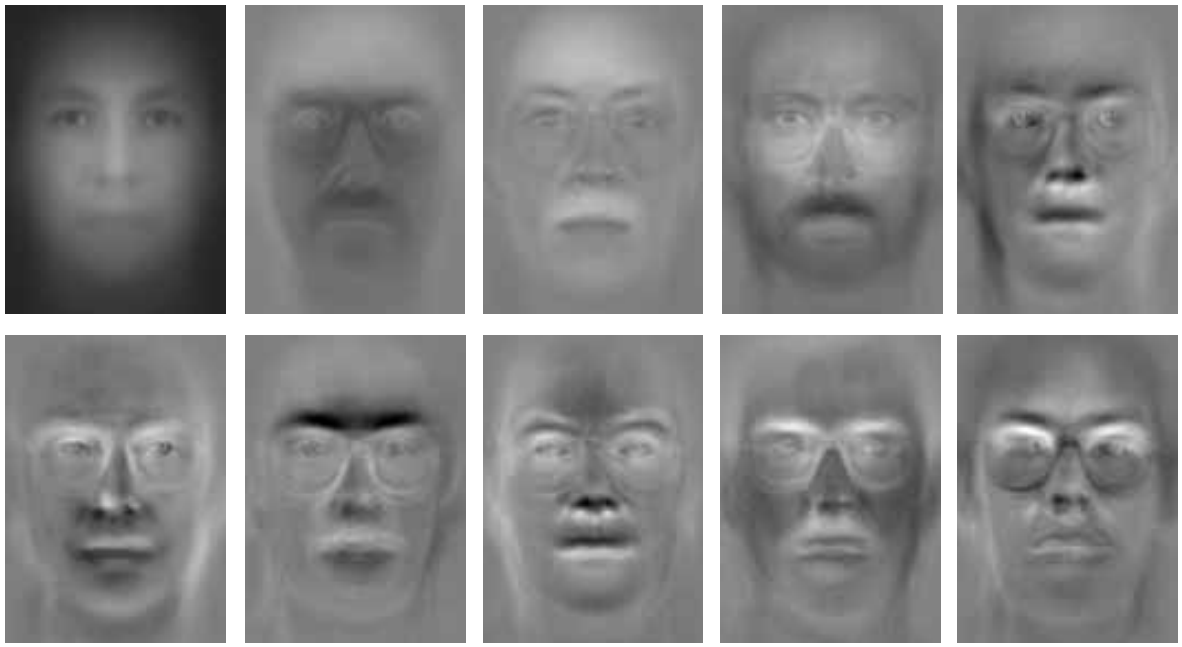


Figure 3-1: Example of eigenfaces.
Example obtained from the X2MVTS database (cf. Subsection 2.7.2), the data set comprising 200 images representing 100 different persons, with 2 images per person. Top-left image: Calculated mean face Ψ; Remaining images considered from left to right and from top to bottom: Calculated first 9 eigenfaces.

the training set and when using all the eigenfaces. The process is otherwise an approximation hopefully sufficiently close to the test image.

It is expected that images with slightly varying viewpoint, or illumination or facial expression, will still get projected onto a region of the subspace ω_k close to each other, i.e. that they will form a cluster in the face subspace. By measuring the Euclidean distance – or as a better choice a Bayesian estimation [3-68][3-69][3-70] – between the reference projection (or reference cluster) and test projections, it is possible to tell if the test view belongs to the same person as the reference one. Additionally, if the reconstructed image differs too much from the test face, it can be claimed that the image presented to the system does not correspond to a face (Figure 3-2).

Consequently, this method can also be used for face detection by calculating the so-called *distance from face-space* using different orientations and scales. Nevertheless, this would become computationally very intensive due to the interpolation process to produce scaled and rotated images. Alternatively, it would be possible to build several models (i.e. eigenfaces) at different scales and orientations.

Figure 3-2: Reconstruction of face and non-face type images using eigenfaces.
Example obtained from the XM2VTS database, the data set comprising 200 images representing 100 different persons, with 2 images per person. a) Out-of-sample image (i.e. image not included in the training phase) of an enrolled client, and b) Corresponding reconstructed face using 199 eigenfaces; c) Novel non-face image fed into the system, and d) Corresponding face image achieved after reconstruction. A simple Euclidean distance measured between the images c) and d) can indicate that c) does not represent a face, whereas image d) does indeed.

Because the Karhunen-Loève transform, also called Principal Component Analysis (PCA), produces a subspace that maximizes the variance but not the discriminability of the classes contained in the training set, some authors (e.g. Belhumeur et al. [3-10]) proposed to use the Linear Discriminant Analysis (LDA), also known as the Fisher Linear Discriminant (FLD) after the PCA step. The transformed version of the eigenvectors, which were called eigenfaces, are now logically called *fisherfaces* after the FLD has been applied. As discriminability among different classes, the fisherfaces are also called Most Discriminating Features, as compared to the Most Expressive Features that correspond to the eigenfaces. These fisherfaces were demonstrated to perform better than simple eigenfaces or than simple correlation.

Nevertheless, this transformation is class-dependent and is difficult to generalize when the number of samples per class is relatively small [3-65]. Liu and Wechsler proposed a slightly modified version of Fisherfaces [3-60] called “Enhanced Fisher linear discriminant”. Other methods such as Independent Component Analysis (ICA) can also be used advantageously instead of PCA, because they separate the high-order moments of the input additionally to the second-order moments, as is the case with PCA [3-8]. Wechsler et al. [3-59] proposed also an evolution pursuit algorithm implementing strategies for searching the optimal representation basis. The authors claim to have obtained improved performance as compared to the Eigenface approach, and better generalization than the Fisherface approach, which tends to overfit the

training set but give poor results with test images taken in different conditions.

In addition, since the direct pixel values are used as features, the method is strongly illuminant dependent. This can be already observed in the eigenfaces represented in Figure 3-1, since they clearly incorporate the information on the illuminant direction. This problem can be reduced if the training database contains enough images with different illuminants, i.e. if illuminants appearing in the test phase were present in the training phase. In this case, information about illuminants is taken into account in the eigenfaces, so that the illuminant is analyzed/ synthesized jointly to the facial features. Not only this does not generalize to non-trained illuminants but it also reduce the information allocated to facial features! Thus, a better solution is to preprocess the image prior to eigenface projection (similarly to the preprocessing performed with template-based methods). Chung et al. [3-19] proposed to use Gabor-filtered images before proceeding to PCA. Liu and Wechsler [3-61] proposed similarly the use of Gabor filters before their Enhanced FLD. Heseltine [3-48] implemented several pre-processing techniques (e.g. image smoothing, sharpening, edge-detection, contours, color normalization, etc.) showing that results can be significantly improved. However Adini [3-1] tested edge maps, image intensity derivatives and images convolved with 2D Gabor-like filters representations for recognition under changes in illumination direction, and concluded that: “all the image representations under study were insufficient by themselves to overcome variations because of changes in illumination direction”.

Another problem of the basic eigenface approach is its view-point and expression dependence. Indeed the system can only reconstruct images with view-points and expressions that were present in the training set! Consequently, the training phase is crucial for this system to work satisfyingly. Alternatively, it can be the goal of the system to remove some unwanted features (such as glasses or expressions): a system trained with facial features devoid of any image with glasses will be unable to reconstruct glasses in a novel view [3-43] (see Figure 3-3). This can be used e.g. to detect and remove the glasses prior to using another algorithm [3-75]. According to Belhumeur [3-10] the Fisherfaces are performing better than the Eigenfaces regarding the detection of glasses.

A solution to the view-point dependence consists in the removal of any “shape” information prior to the eigenface calculation, so as to keep only “texture” information. For example Craw et al. [3-25] propose to

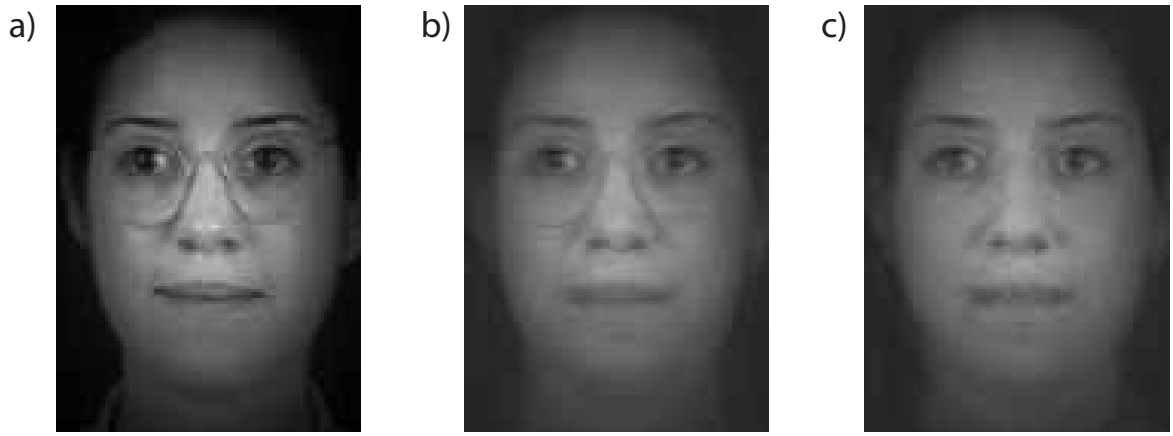


Figure 3-3: Reconstruction of a sample face with glasses using a set of eigenfaces.
a) Image containing glasses of a person included in the training set, though this particular image was not included in the training set (i.e. an out-of-sample image); b) Eigenface reconstruction of a) built on a training set of 190 images containing glasses and representing 190 different persons; c) Eigenfaces reconstruction of a) built on a training set of 192 images devoid of glasses and representing 189 different persons, with 1 image per person, and 3 additional images for the person illustrated in a).

use 34 control points in the face image (related to facial features) in order to *warp*[5] all the image faces to a standard face. Warping is achieved by bilinear interpolation between the control points. In their experiment Craw et al., *manually* selected and aligned the 34 points on all the images of the database. This approach is thus in turn sensitive to the precise detection (when performed automatically) of a high number of points and a complex bilinear interpolation is moreover necessary.

For Gong [3-37] only a sparse correspondence – i.e. a correspondence limited to a small set of features called salient features – is necessary, noticing also that dense correspondence based on a regular grid (i.e. similar to optical flow calculation) is expensive and is not interesting, except for taking into account parameters such as expression. Martinez [3-64] used only seven different facial features to normalize the shape, and ensured that the warping did not deform the important facial features. Further, he stated that: “It would be unrealistic to hope for a system that localizes all the facial features described above with a high degree of precision”. Martinez thus proposes to model the localization problem with a gaussian (or mixture of gaussians) model. However the method of Martinez requires an estimate of the error of the localization algorithm, which is almost as unrealistic as the assumption of precise localization of facial features.

5. Warping is the transformation used to register (i.e. align) two images by deforming a typical image template into alignment with a target image of interest. Warping is extensively used in pattern recognition, e.g. in biomedical image processing.

As a conclusion, we saw that the problem of warping the image for shape-free representation and normalizing it for illumination independence, largely enhances the quality of eigenface-based methods, but that they induce non-negligible extra manipulations. The proposed solutions for warping depend on the detection of facial features and they are prone to error and thus to reduced recognition rates. It is interesting to note that the proposed solution to achieve an illumination normalization seems more concerned by the direction of the illuminant rather than by the color of the illuminant, which is obviously a serious problem for mobile applications (see also Subsection 4.2.1).

Finally, although some authors consider the general eigenface method as holistic (Zhao for instance [3-106]), we regard this method as an hybrid between feature-based and holistic methods due to the alignment of faces which is often grounded on the prior detection of fiducial landmarks.

3.4.4 Local feature analysis methods

We use here the term *Local Feature Analysis* (LFA for short) for a method developed by Penev & Atick at Rockfeller University [3-76][3-77][3-78], which is actually an extension of the Eigenfaces developed by Kirby and Sirovich, and needing to be distinguished from the different LFA method developed by Arca et al. [3-4] (the latter being closer to Elastic Bunch Graph Matching and Active Shape Models, see Subsection 3.4.5).

LFA is used to extract local information from global PCA modes by using kernels at selected positions that have a local support instead of the global support of PCA kernels (i.e. the latter extends over the entire image). This solves two issues of the PCA approach. Firstly, the fact that the global representation of the PCA affects images presenting localized changes reduces the efficiency of PCA in case of pose, alignment, cropping and expression variations.

Indeed, different locations of features in the training set would lead to different eigenmodes. Consequently, if p parameters would be necessary to describe a well-localized feature, it would take $L \times p$ parameters to describe exactly the same feature appearing at L different locations. LFA introduces an explicit notion of locality so that the above situation can be encoded with p parameters for its description, plus one for its localization.

Secondly, PCA is non topographic, i.e. neighbouring vectors in the feature space (i.e. the face space) do not necessarily represent similar images. Conversely, two images differing only locally may not project onto neighbouring locations in the feature space. When using the same number of kernels as for PCA, a better perceptual reconstruction is achieved with LFA.

Starting from the PCA decomposition (Eq. 3.1 to Eq. 3.4), a topographical kernel K of the representation is calculated with Eq. 3.5[6]

$$\vec{K}_{x0} = \sum_{r=1}^{M} \vec{u}_r(x_0)\frac{1}{\sqrt{\lambda_r}}\vec{u}_r \qquad (3.5)$$

where x_0 represents a possible location of the sampling grid (i.e. one pixel index), $\vec{u}_r$ represents the r-th eigenvector ($\vec{u}_r(x_0)$ is the x_0-th element of the eigenvector $\vec{u}_r$ and is thus a scalar), M is the number of eigenfaces, and λ_r is the associated r-th eigenvalue.

K is thus a set of images (represented by vectors in the above notation), constructing a dense representation of the training ensemble.

The projection of an image on these kernels is given by the dot product:

$$O_{x0} = (\vec{K}_{x0})^T \cdot \vec{\Gamma} \qquad (3.6)$$

while the residual of the correlation of the outputs is obtained from Eq. 3.7.

$$\vec{P}_{x0} = \sum_{r=1}^{M} \vec{u}_r(x_0)\vec{u}_r \qquad (3.7)$$

This way, a dense representation K (for all x_0 on the sensor grid) is obtained, temporarily containing an important residual correlation (i.e. redundancy) as opposed to the previous low-dimensionality of the eigenfaces. Therefore, the authors propose as a second step to dynamically sparsify K, e.g. by using the so-called *lateral inhibition* principle. As a result, it is possible to predict the entire output from the knowledge of the output at a subset of points and from the correlation matrix P.

6. Note that the notation starting from Eq. 3.5 is adapted to match that of Turk and Pentland, and consequently does not reproduce the notation introduced by Penev et al.

Once a face is mapped onto the above representation, the final operation consists in matching it against the templates in the database. Several metrics can be used, e.g. the "normalized correlator between the two sets" [3-6]:

$$E = \sum_{m,\,n} O_m O_n \vec{P}_{m(n)} \qquad (3.8)$$

This method seems to present some advantages but curiously received less attention in the literature, possibly due to its inherent high mathematical complexity. This method is said to be used in Visionic's commercial system called FaceIt [3-6] (see also Section 3.6). Still, this method seems to suffer from the same limitation as the Eigenface approach, namely that a precise face detection is necessary for the alignment of faces.

3.4.5 Graph-based methods

The baseline eigenface approach benefits from an inherent simplicity that makes it interesting for low complexity applications and thus for low-power consumption. However it must be noted that the several *mandatory* improvements discussed in the case of eigenfaces or fisherfaces tend to increase their computational complexity by incorporating irregular processing. In the case of methods based on deformable templates, the computational complexity is present from the beginning but the process is more regular, i.e. applied uniformly over the whole image. Consequently this category of methods seems also interesting. Nonetheless, the deformable template based methods presented in the case of eigenfaces (see e.g. discussion of Craw et al. [3-25] in Subsection 3.4.3) consider only templates located on reference points (also called fiducial points) associated to facial features. Tuned filters (i.e. models of those fiducial points) are then used for the precise detection of those points, which is a clear drawback. A holistic method combining local templates not related to facial features and taking implicitly account of deformations would be more interesting.

A method falling into this category was introduced by Lades et al. under the name "Elastic (Bunch) Graph Matching" (EGM), which was inspired from the "Dynamic Link Architecture" (DLA) method. Because the DLA method is also based on fiducial points, some authors argue that this is a feature-based method [3-106]. However in our opinion, since the feature extraction is performed in the EGM algorithm with-

out association to particular facial features, but rather by using a regular grid, the EGM algorithm clearly fall in the holistic category.

The DLA algorithm is based on neuropsychologic considerations [3-62] but it appears to be indeed relatively close to the computational approach proposed by Fischler and Elschlager. In EGM, a face is holistically represented by labelled graphs, i.e. graphs formed of edges and nodes, containing simultaneously global (i.e. topographical) and local information. The local information associated to a node, i.e. a local template, is not associated to a particular fiducial point but is sampled on a regular grid in the reference image (Figure 3-4). In some contributions however (e.g. [3-73]), nodes are located on specific points of the face and the sampling is not regular.

At each node, a shape or texture information is extracted with untuned pattern recognition techniques such as Gabor filters. In the case of Gabor filters the information is pseudo-local because information from the whole image is used due to the infinite impulse response nature of the filters, even though pixels close to the node have more influence than pixels further away from the node. The Gabor filters will be discussed in Subsection 4.3.1.

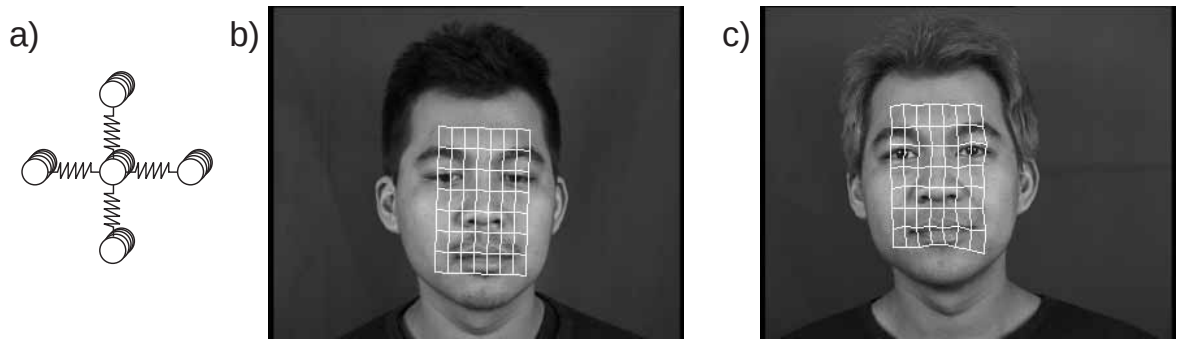


Figure 3-4: Labelled graphs and Elastic Graph Matching.
a) Model of five connected nodes, with their attributed features connected by edges (actually springs), as considered by Fischler and Elschlager [3-31]; b) Reference image with a regular orthogonal graph rotated to align with the line connecting the eye centres; c) Corresponding matched graph drawn on a test image.

The node information together with the geometric graph structure (i.e. distance between nodes) can be stored as a reference template that will be compared to a new view. When this new view is presented to the system a correspondence is searched between the two images, so that features of the test graph minimize a distance measure with respect to features of the reference graph. This distance measure also contains a penalty for deformations of the graph.

Finding such an isomorphism between two graph representations is known to be a complex task (noticing that conjectures about P or NP complexity are still pending [3-67]), and a simplification is necessary in order to find this correspondence in a reasonable amount of time. Lades [3-56] proposed first to perform a rigid scanning with an orthogonal graph (similar to the one used during the training phase) and, from the best orthogonal graph position, to deform iteratively the nodes using local random perturbations until no further improvement in the fit is observed. Kotropoulos [3-55] argued that this method is possibly not converging and proposed an optimization based on simulated annealing. As we will see in Subsection 4.5.1, convergence is strongly dependent on the type of features used. Nevertheless, to improve the robustness of the system with respect to scaling and rotations, Lades [3-57] devised a new optimal version of the rigid scanning, by searching for the best rotation and scaling using a Monte-Carlo approach.

The method proposed by the University of South California (USC) uses 40 values (called "jets") extracted from Gabor filters (5 resolutions and 8 orientations) at each node location, these jets being either complex [3-73] or real valued [3-56]. The proposed cost function C_{total} between features associated to a node is a normed dot product of jets C_{jets}, complemented by an additional term λC_{edges} for deformation penalty (Eq. 3.9).

$$C_{total} = C_{jets} + \lambda C_{edges} = \sum_{i \in V} \frac{J_i^I \cdot J_i^M}{\| J_i^I \| \| J_i^M \|} + \lambda \sum_{(i,j) \in E} \left(\overrightarrow{\Delta^M}_{ij} - \overrightarrow{\Delta^I}_{ij} \right) \qquad (3.9)$$

where λ is a factor of elasticity, E is the set of all edges that are connecting the node to its four neighbours, I represents the "image domain" (i.e. the test image), while M is the "model domain" (i.e. the reference), J denotes the jet, and V represents the set of vertices, and finally $\overrightarrow{\Delta_{ij}}$ is an edge between vertices $\vec{x}_i$ and $\vec{x}_j$:

$$\overrightarrow{\Delta_{ij}} = \vec{x}_j - \vec{x}_i \qquad (3.10)$$

The main computation consists then in the calculation of the 40 filtering operations per vertex and the iteration of the rigid and elastic matching. Lades performs this task on an MIMD architecture composed of 23 "transputers". The identification of a face in a database of 87 faces is reported to take about 25 seconds [3-56].

Another similar implementation by Duc et al. uses a Gabor filter bank composed of 18 filters (3 resolutions and 6 orientations) and an orthogonal 8 by 8 graph [3-27]. Wiskott claims that this method needs only small modifications to be applied to gender recognition, object recognition or other different tasks [3-100]. A proposed improvement consists in: incorporating phase information of the wavelet transforms and using a new metric; supporting large in depth rotations by using *bunch graphs* i.e. a collection of jets for each node instead of a single jet per node [3-100][3-101].

It must be noted however that bunch graphs are again related to fiducial landmarks. Attempts to increase the robustness with respect to depth rotated probes were made by modelling rotation properties and transforming features [3-66]. Some improvements to the bunch graph matching method were also proposed by Okada [3-73], namely that histogram equalization is performed after landmark detection (in this case nodes of the model graph are facial feature-based) to adjust for differences in lighting and camera settings. Precise measurement of the face size is achieved by computing the mean Euclidean distance of all landmarks. The method is believed to be robust to illumination variations because the Gabor kernels are DC free, and because similarity functions are normalized and robust to variations in pose, size and facial expression due to its deformable nature [3-33]. A complete commercial system based on the USC method exists [3-54] and a patent was filed (see Section 3.6).

As stated by Zhang et al., many choices exist for the features vectors [3-105]. Kotropoulos et al. use mathematical morphology operators (i.e. dilations and erosions [3-55][3-92], or morphological signal decomposition [3-93]) and use an Euclidean distance instead of a dot product. This solution is obviously less computationally intensive compared to filtering with multiple Gabor kernels. We will evaluate these classes of features in Section 4.2.

As a final note, interesting similarities between graph matching methods and earlier work are observed. For example Burr already proposed elastic matching of line drawings [3-16]. The formerly mentioned work by Fischler and Elschlager [3-31] is relatively close to graph matching despite of the fact that they consider facial features and components instead of more general patterns (e.g. texture, shape), an approach generally classified as *articulated* matching rather than elastic matching [3-2].

It has to be noted that known variants of the above methods are all model-free in the sense that no statistical model of deformations is built. The already mentioned work by Cootes (e.g. [3-20]) is however model-based although close to the graph matching approach. Nevertheless, Active Shape Models are clearly feature based, leaving the model-based but holistic area of our classification empty for the moment.[7] The multiresolution elastic matching of Bajcsy is also very similar, using partial differential equations as constraint equations during the matching [3-7].

Finally, warped images used in eigenfaces (i.e. shape-free eigenfaces) are also a form of deformable templates. However in this case, a modification of the test image (i.e. a bilinear interpolation) is necessary while EGM compensates directly for such deformations.

3.4.6 Other approaches

An important number of implementations differs notably from the above methods. For example several authors proposed the use of neural networks for classification (e.g. [3-84][3-85]), using possibly a preprocessing (e.g. a FFT [3-11][3-30]). Three-dimensional based methods based on stereo information [3-47], on shape from shading or profiles [3-5][3-38][3-39][3-103] are also quite common. Finally it is worth mentioning Hidden Markov Models (HMM) [3-63][3-71], error correcting codes [3-53], and saccadic movements (retinal sampling) [3-90][3-91]. However, the deeper study of these methods depart from the scope of this thesis.

3.5 Summary and algorithm selection

To get an overview of the listed implementations, we can use the classification criteria described in Subsection 3.3.1 and plot a Vennes graph representing different operating domains (Figure 3-5). Because a six dimensional diagram is difficult to represent, we will omit the following criteria in the diagram and in the discussion: d) Global vs. local image features, e) Bottom-up vs. top-down methods, f) Two- vs. three-dimensional methods, and g) Implementation specific characteristics [8]. In fact the three remaining criteria are the most important criteria for mobile applications: a) Holistic vs. feature-based methods, b) Rigid vs.

7. In Subsection 4.5.3 we will propose deformable models built on *principal warps*.
8. They were however used previously in the discussion.

deformable templates, and c) Problem-specific vs. untuned image features.

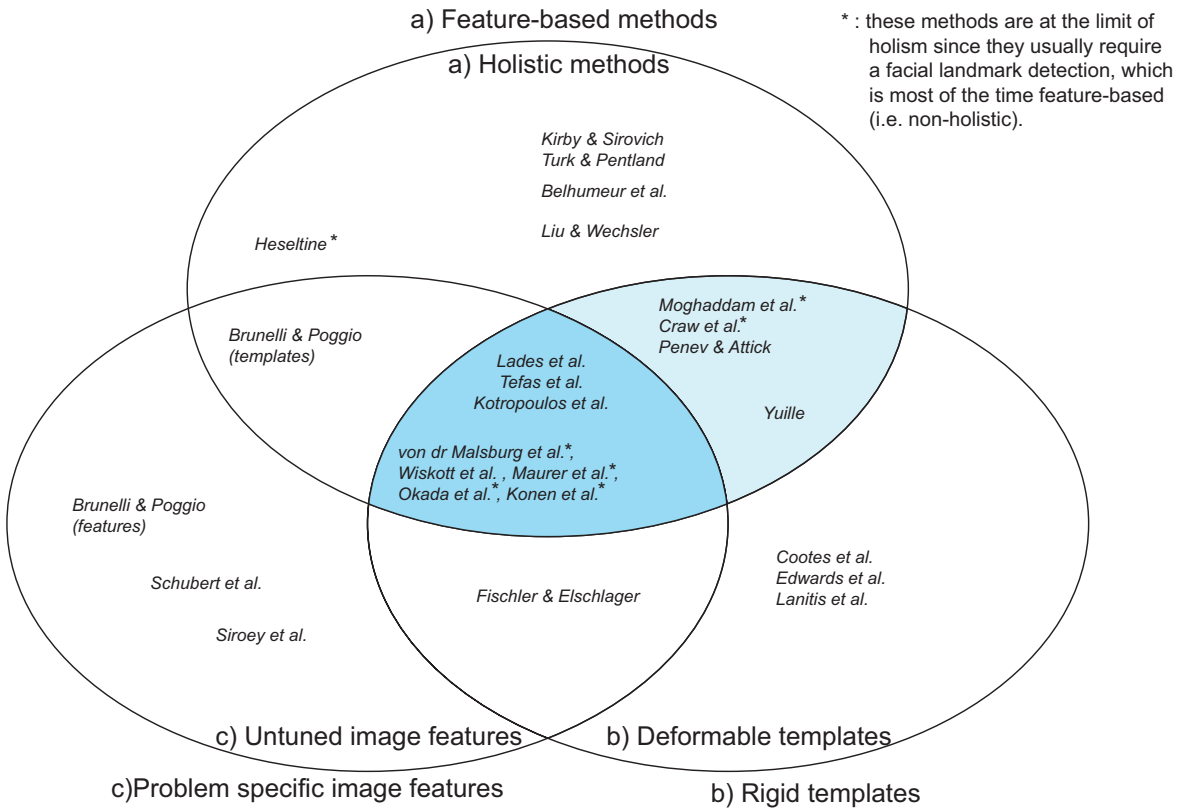


Figure 3-5: Classification along three criteria a), b) and c) of the surveyed algorithms. Vennes diagram representing the classification of several algorithms based on the major criteria defined in Subsection 3.3.1. Only the names of authors are given; see the full text for discussion and references.

Criterion a) Holistic vs. feature-based methods is important, because methods belonging to the latter are less robust due to possible imprecise localization of specific features (i.e. fiducial landmarks), or they are more prone to errors related to the occlusion of some essential features. Criterion b) Rigid vs. deformable templates is also important because the method shall still be robust enough even with an unconstrained view-point, which is essential for mobile device, and that rigid templates are less effective for this purpose. Note the importance of using deformable templates also for robustness to varying facial expressions and the fact that the global vs. local criterion is somehow anyway related to criterion b), because deformable templates are most of the time using local instead of global information. Finally, problem specific im-

age features can lead to generalization problems[9] that are not present with untuned features, thus the presence of criterion c).

The darker regions enclose the holistic methods that are using deformable templates (criteria a) and b)). A method for mobile face authentication should be preferably selected in this region. These regions are further split into “untuned image features” (darkest region) and “problem specific image features” regions, the former region being again preferred for mobile face authentication. Indeed, although methods belonging to the latter (and which are essentially derived from eigenfaces, e.g. LFA from Penev & Atick) can be trained more or less successfully to fit novel views, the efficiency of the training heavily depends on the quality of the training set. Consequently, it seems to us that untuned image feature extraction is preferable as it should generalize better to images other than those available in the training set, and as they should relax efforts in designing a “perfect” training set.

This selection consequently eliminates all the non-holistic methods (i.e. feature-based and local template-based methods), as well as the rigid methods such as the baseline eigenfaces of Turk and Pentland. The active shape model of Cootes et al. is interesting in the sense that deformations are modelled based on statistical observations. An extension of its use to non feature based methods would be however necessary in order to fit in the holistic category.

Finally inside the dark blue region, we can make a fine distinction between implementations of Lades, Kotropoulos and Tefas, from those of von der Malsburg, Maurer, Wiskott, Okada and Konen, because the prior use an orthogonal grid, whereas the latter use fiducial landmarks, falling close to feature-based (i.e. non-holistic methods). Since we have no a priori knowledge of the discriminatory power of the different parts of the face, it seems wiser to use an arbitrary grid and then to perform a statistical analysis to weight the importance of different nodes. One could argue however that adapting statistically the power of the nodes is a problem specific approach, thus falling back in the light blue category, but it remains clear that the approach (i.e. feature extraction) remains untuned.

To conclude this section, we will review some conclusions reported in the surveys mentioned at the beginning of this chapter. Chellappa [3-

9. I.e. the training set is overfitted, but new test images produce poor fitting.

18] makes essentially a discussion of the methods without comparing them explicitly. The same observation is true for Samal and Iyengar [3-86], who essentially listed issues in feature-based methods, and for Valentin and Abdi [3-99], who compared different connectionist models without comparing their performances on a given data set. The survey of Chellappa can be complemented by that of Fromherz [3-32] for the years 1995 to 1998. The conclusion of Gutta et al. [3-42] is also that holistic methods outperform feature-based approaches and simple correlation methods. However their definition of holism may not be precisely the one used in this report. Hancock et al. compared Eigenfaces and Graph Matching approaches from a human point of view [3-44]. Their conclusion is that both systems are able to capture human performance, provided that the shape-free PCA is used. They also state that PCA seems to account for how humans recognize previously unfamiliar faces, while graph matching may provide a representation of how novel views of familiar faces can be generalized.

Gross et al. systematically compared two algorithms (based on Moghaddam [3-68] and Penev [3-76]) with respect to robustness to expression, illumination, pose, occlusion and gender [3-41]. Both algorithms were provided with precise face location information, thus not testing directly their robustness to small or larger detection errors. Their conclusions are that the viewpoint can vary up to ±45° without largely reducing the recognition rates; and that illumination changes are handled well; and that expression does not cause too much problems except for large modifications; and that recognition of female subjects seems more robust (approx. 5%) than for male subjects. Zhang, Yan and Lades compared the Eigenface, Graph Matching and Neural Nets approaches [3-105], and concluded that eigenface performance deteriorates when the view-point and lighting conditions change. Consequently, they believe that Graph Matching is superior, although it may require a higher computational effort. However two important considerations should be made here: firstly, the simple version of eigenfaces was tested, and consequently shape-free eigenfaces would probably improve the performance; and secondly Lades, who is at the origin of Elastic Graph Matching, is co-author of this comparison, meaning that the conclusion could be potentially biased.

Zhao et al. [3-106] consider that the Elastic Bunch Graph Matching method has very good performance, although it is limited to relatively high resolution images.

Finally, two ultimate motivations favouring the use of Elastic Graph Matching are given by modularity and regularity. Modularity is indeed supported in the sense that it is possible to modify the type of extracted features (e.g. Gabor filtered images, or dilated and eroded images) with few or no modifications to the Graph Matching framework. The second argument referring to regularity is explained by the fact that this algorithm – although it seems at first glance very demanding regarding computational complexity – appears easier and more regular than eigenfaces with shape-free structures. For these reasons the rest of this report will describe algorithms based on Elastic Graph Matching with the addition of improvement for mobile applications (Chapter 4), as well as hardware architectures allowing their use on low-power devices (Chapter 6).

3.6 Projects consortia, commercial products and patents

First, a number of consortia, workgroups and directories related to Biometrics emerged recently. It is worth mentioning:

- *Biometrics consortium* (*http//www.biometrics.org*)
- *International Biometric Group* (IBG) (*http://www.biometricgroup.com*)
- *International Biometric Society* (*http://www.tibs.org*)
- *The Biometric Foundation* (*http://www.biometricfoundation.org*)
- *Association for Biometrics, UK* (*http://www.afb.org*)
- *SecureID news* (*http://www.secureidnews.com*)
- *InsideID* (*http://www.insideid.com*)
- *BioAPI consortium* (*http://www.bioapi.org*)
- *US Department of Defence Biometrics (http://www.biometrics.dod.mil)*
- *Biometrics Institute* (*http://www.biometricsinstitute.org*)
- *BioSecure*, European Network of Excellence (NoE) (*http://www.biosecure.org*)
- *Biometrics catalog* (*http://www.biometricscatalog.com*)

- *Biometric Information Directory* (*http://www.biometricinfodirectory.com*)
- *findBIOMETRICS* (*http://www.findbiometrics.com*)
- *Biometric Watch* (*http://www.biometricwatch.com/*)
- *Biometrics in Human Services User Group* (*http://www.dss.state.ct.us/digital/faq/dihsug.htm*)
- *COST 275 action* (*http://www.fub.it/cost275/*; see also *http://www.cost.esf.org*)
- *The Face Recognition Homepage* (*http://www.face-rec.org/*)

Specialized magazines also appeared, the leading ones being:

- *Biometric Technology Today* (*http://www.biometrics-today.com*)
- *Biometrics Digest* (*http://www.biodigest.com*)
- *Biometrics Market Intelligence* (*http://www.acuity-mi.com*)
- *Global ID magazine* (*http://www.global-id-magazine.com*)

As observed during the Face Recognition Vendor Test (FRVT) the state-of-art in facial recognition is constantly improving (1% false acceptances at 90% of correct recognition in 2003 on this particular database). This means that a description of commercially available products at the time of writing this report will inevitably become outdated soon after. Consequently the purpose of this section is not to build a precise image of state-of-art commercial solutions, but rather to give clues on – or trends in – this domain.

Participants to the FRVT 2002 event are a good representative subset of the key players in the field of facial recognition. These companies are briefly mentioned below (in alphabetical order), together with the type of algorithm used at time of this competition.

- *AcSys Biometrics Group (http://www.acsysbiometricscorp.com)*: their system seems to be based on an artificial neural network with a special “quantum mechanics inspired” back-propagation algorithm. No references to academic work were found.
- *Cognitech Systems GmbH (http://www.cognitech-systems.com):* different variants of their FaceVACS system are available. The system seems to be mainly based on a Fisherface variant.

- *C-VIS Computer Vision and Automation GmbH (http://www.c-vis.com):* they provide both hardware and software through their FaceSnap recorders (*http://www.facesnap.de*). The type of algorithms used is said to be "based on neural computing and combines the advantages of elastic and neural networks". It might thus be based on a method similar to Dynamic Link Architecture.
- *Dream Mirh Co. Ltd. (http://www.dreammirh.com):* this Korean company is active both in facial and fingerprint recognition. The type of algorithms used is not known. They provide with a solution for mobile phones, but the image is transmitted to a remote server without local processing on this mobile.
- *Neven Vision (formerly Eyematic Interfaces Inc.) (http://www.nevenvision.com):* their products mainly target mobile multimedia (e.g. real-time avatar chat, advanced user interfaces), but also provide real-time facial tracking and recognition. They do not mention explicitly if matching can be done on the mobile terminal. The type of algorithms used is not disclosed but seems to be based on Active Shape Models. In addition, 3D head reconstruction is mentioned.
- *Iconquest (http://www.iconquesttech.com):* Iconquest claims that their technology fractals based technology is especially effective when poor image quality is present.
- *Identix (http://www.identix.com):* their products are based on the FaceIt technology provided by Visionics (*http://www.FaceIt.com*), which uses the Local Feature Analysis (LFA) described in Subsection 3.4.4. IBIS mobile terminal allows to capture fingerprints and face image and transmits them wirelessly to a remote server and database for matching.
- *Imagis Technologies (http://www.imagistechnologies.com/):* ID2000 is mainly a law enforcement software. Their products are all based on their ID-2000 technology which is using spectral information in conjunction with 3D models. Re-rendered images are then wavelet transformed and compared.
- *Viisage Technology (http://www.viisage.com):* the underlying technology - namely a variant of eigenfaces - was mainly developed at the Massachusetts Institute of Technology (MIT). In addition, Viisage acquired ZN-Vision Technologies, a

spin-off from the Ruhr University in Germany, which was producing ZN-face based on Elastic Graph Matching of Gabor features.

- *VisionSphere Technologies Inc. (http://www.visionspheretech. com):* available products rely on the UnMask technology using so-called “holistic feature codes”. Further details on this method are however not disclosed, but the method requires precise face and eye location for normalization. The VisionSphere software was ported on a Texas Instrument TSM320DM640/642 DSP. Proposed products encompass FaceCam for physical access control, ItsMe for network access, and HawkEye for video surveillance. No mobile application of the DSP port is described.

Other producers of face recognition technology who did not participate to the FRVT 2002 event can however be found. Here is again a non-exhaustive list:

- *Gaga Security System* (*http://www.exim21.com/security/frm4000.php*) they promote a Facial Recognition Module and software development kit (SDK) called FRM4000. It is centred around a 32 bit DSP and a 230 or 430 thousand pixel CCD camera. The verification is said to be performed in about 0.5 second, however the algorithm used in the matching is not mentioned. Still, from the pictures displayed on the manufacturer’s website, it can be believed that a feature based approach is used. Finally, targeted applications seem not to be mobile, as the illustration given on the same website showcases a door access control.
- *FaceKey Corporation* (*http://www.facekey.com*) is an integrator of face and fingerprint biometric systems, mainly targeting access control. Detection and recognition are possible, the latter allowing the comparison of 20’000 faces per second.
- *Keyware* (*http://www.keywareusa.com*) is essentially an integrator of biometric technology. Example of products is the CAS SignOn logon software.
- *ID-arts* (*http://www.id-arts.com*) promote the “Passfaces Cognometric Authentication”, which is not really a biometric authentication. Rather, it replaces standard keywords with a sequence of faces that the user has to recognize.
- *ImageWare Systems Inc.* (*http://www.iwsinc.com*) claim to have developed a scalable “Biometric engine” that seems to integrate

third-party algorithms (such as Identix, Cognitech, Viisage, or Neven).

- *BioID* (*http://www.bioid.com*) proposes an SDK, which performs multimodal authentication relying on face, voice and "pass-phrase" (measurement of the lips movement). The technology used for the face modality is not mentioned.
- *Biometric Systems Inc.* (*http://www.biometrica.com*) targets mainly casinos. No technology information is available and this company seems to be an integrator rather than a developer of biometric technology. It is furthermore a subsidiary of Viisage Technology.
- *ITC-irst* (*http://spotit.itc.it*) delivers a system called SpotIt that is said to create "photographic quality face composites and to *automagically* find the most similar faces in a large database". Developed among others by R. Brunelli, this software can thus generate and transform faces. This technology is patented (US 5764790).

Note that this list is certainly not exhaustive, since hundreds of companies emerged in the five past years. For instance 546 companies in the biometric field are indexed by the "Biometric Information Directory" as of November 23, 2004. However 127 are related to the face modality i.e. less than one fourth, whereas 291 are related to fingerprint for instance. We believe however that the above subset is quite representative of the largest companies in years 2004-2005. Additionally, note that web links may change very rapidly and soon become obsolete.

A quick search on pending patents in the biometrics and face authentication domain was then performed. Firstly, while using the above mentioned companies as assignees, a number of patents (European, American, Japanese and Worldwide patent numbers are given when available) seemingly related to biometrics could be identified (see Table 3-1). It can be observed that large companies possess several patents, but from a preliminary review it is difficult to assess if these patents are strong and cover a large area or if they are more specific to a particular implementation used by the corresponding company.

In the US classification, the majority of patents regarding face authentication are within "image analysis" categories 382/115 ("personnel identification e.g. biometrics") and 382/118 ("using facial characteristics"). A search on the latter returns 231 patents, while the combination of both returns 35 patents only.

A second patent search was performed by seeking explicitly face recognition methods and apparatuses for face recognition. This search returned only 36 patens containing keywords “recognition” and “facial” or “face”. The most interesting results belonging to this category were between methods (Table 3-2) and applications and devices (Table 3-3).

Most recent patented methods cover the combination of 2D and 3D methods, while older methods focused mainly on the eigenface approach[10]. It can be readily observed that the number of devices and applications is much larger than the number of methods.

Company	Patent number (US/EP/WO)	Patent title
Neven Vision (formerly Eyematic interfaces)	*US 6'580'811*	*Wavelet-based facial motion capture for avatar animation*
	US 6'563'950 US 6'356'659 US 6'222'939	*Labeled bunch graphs for image analysis*
	US 6'466'695 /WO 0'111'551	*Procedure for automatic analysis of images and image sequences based on two-dimensional shape primitives*
	US 6'301'370 /WO 9'953'427 /EP 1'072'014	*Face recognition from video images*
	US 6'272'231	*Wavelet-based facial motion capture for avatar animation*
C-VIS	*US 5'864'363 /EP 0'735'757*	*Method and device for automatically taking a picture of a person's face*
Viisage	*US 6'681'032*	*Real-time facial recognition and verification system*
	20030059124	*Real-time facial recognition and verification system*
NextGen ID	*US 20040036574*	*Distributed biometric access control method and apparatus*

Table 3-1: Patents assigned to companies mentioned in Section 3.6.

10. Using “eigenfaces” as a keyword expected to appear in the title, the number of hits reaches 32 on the US patent database. Most of them correspond to applications or devices using the eigenface paradigm.

Company	Patent number (US/EP/WO)	Patent title
Secure ID-Net	*US 20010004231*	*Secure system using images of only part of a body as the key where the part has continuously-changing features*
Identix	*US 6'665'427*	*Apparatus and method for electronically acquiring fingerprint image with low cost removable platen and separate imaging device*
	US 6'484'260	*Personal identification system*
	US 6'175'407 /EP 1'157'351 /WO 0'036'548	*Apparatus and method for optically imaging features on the surface of a hand*
	US 5'825'474	*Heated optical platen cover for a fingerprint imaging system*
	US 5'748'766 /EP 0'976'088 /WO 9'741'528	*Method and device for reducing smear in a rolled fingerprint image*
	US 5'650'842	*Device and method for obtaining a plain image of multiple fingerprints*
	US 5'067'162 /EP 0'251'504 /JP 63'041'989	*Method and apparatus for verifying identity using image correlation*
	US 4'537'484 /WO 8'503'362	*Fingerprint imaging apparatus*

Table 3-1: Patents assigned to companies mentioned in Section 3.6.

Finally, a search with the keywords “mobile” and “face” was attempted. From the 34 results returned, only 2 seemed interesting, but they were not available at the time of writing the thesis report (Korean applications). None seemed to cover the full face verification on a mobile terminal.

Patent number (US/EP/WO)	Patent title	Short description
US2005008199	*Facial recognition system and method*	*Combination of a 2D image with a standard 3D representation*
GB 2'402'311	*Facial recognition using synthetic images*	*Synthesis of 2D images from a 3D model (issued by Canon)*
US2004151349 /WO2004066191	*Method and or system to perform automated facial recognition and comparison using multiple 2D facial images parsed from a captured 3D facial image*	*"Parsing" of 2D images from a 3D acquired representations*
US 5,164,992	*Face recognition system*	*The original Eigenface method*
US 6,044,168	*Model based faced coding and decoding using feature detection and eigenface coding*	*An Eigenface variant*

Table 3-2: Patents related to face recognition methods.

Patent number (US/EP/WO)	Patent title	Application domain
US2004239481	*Method and system for facial recognition biometrics on a fob*[a]	*Secured transactions (issued by American Express)*
US2004234109	*Facial-recognition vehicle security system and automatically starting vehicle*	*Automotive*
US2004164147	*Currency cassette access based on facial recognition*	*ATM*[b] */ currency*
US2004117638	*Method for incorporating facial recognition technology in a multimedia surveillance system*	*Video (multimedia) surveillance*
US2003161507	*Method and apparatus for performing facial recognition with a hand-held imaging device*	*Mobile device with database access to a remote server*

Table 3-3: Patents related to face recognition devices and applications.

a. Self-contained device which may be contained on any portable form factor.
b. Automatic Telling Machine.

As can be seen from the above examples, biometric face recognition is essentially distributed in the form of software development kits (SDK), and less often in the form of hardware platforms in conjunction with software. Most of the time however, this hardware is based on standard material (computers, operating systems). This is probably due to the fact that current applications target rather indoor access control applications than low-power or mobile applications where dedicated hardware is necessary. Except for the companies selling image acquisition devices, the market seems currently mainly software based, leaving open the door to fully integrated solutions. Moreover the trend already mentioned in Section 2.5 indicates that the mobile market should develop in the coming years. Although today's mobile verifications rely on a remote server, future applications may perform the full matching on the mobile itself. This step forward already happened with fingerprint technology and should be mainstream in a few years.

Finally the 2000 FRVT event allowed interesting conclusions to be drawn:

- performance was dramatically degraded with view angles larger than 25 degrees.
- performance was degraded when the test images were captured more than one year after the enrolment.
- handling of lighting changes (indoor vs. outdoor) needed to be thoroughly investigated.

And from the 2002 test:

- Although performing correctly with indoor lighting, the recognition rate decreased from 90% to 50% at a false acceptance rate of 1% when testing in outdoor conditions.
- Performance was largely improved when using 3D morphed images to compensate for geometry changes[11].
- Watch list applications (i.e. comparing a test image to a large list of persons, e.g. watch lists of suspected criminals) decreased in performance when the size of the list is rising (in numbers, the decrease amounts approximately to 2-3% when the size of the list doubles).

11. Feature points were set manually and then an automatic optimization was performed. Its semi-automatic nature and the involved complexity (4 minutes per image on a PC equipped with a Pentium 4 processor operating at a clock frequency of 2GHz) however hinder the use of this method on a mobile terminal.

Clearly the problem of view-pose and illuminant variations represents a hot topic of research for mobile applications.

3.7 Conclusion

In this chapter, we have first presented cognitive aspects of the face recognition process by humans, and we used them to characterize computational methods deployed to accomplish this task. A list of six criteria for classifying existing approaches was given in addition to an updated but nevertheless non-exhaustive list of implementations and the related classification. Finally a category of algorithm was selected that should potentially fulfil the requirements of mobile face verification.

The selected approach - Elastic Graph Matching - seems to be a powerful solution, although apparently computationally expensive compared to other very simple methods. However, the effort necessary to increase the robustness of these simple methods becomes more important than the complexity of graph matching. Having selected this method, we should keep in mind that large out-of-plane rotations are still a problem, although they can be partly solved by appropriate training, and that the robustness to illumination variations will be determined by the feature selection.

Finally we observed at the moment of writing this thesis that no product proposed on the market claims to perform face verification *entirely* on a mobile device.

References

[3-1] Y. Adini, Y. Moses and S. Ullman, "Face Recognition: The Problem of Compensating for Changes in Illumination Direction," *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. 19, No. 7, July 1997, pp. 721-732.

[3-2] J. K. Aggarwal, Q. Cai, W. Liao, and B. Sabata, "Articulated and elastic non-rigid motion: A review," in *Proc. IEEE Computer Society Workshop on Motion of Non-Rigid and Articulated Objects*, Austin, TX, 1994, pp. 16-22.

[3-3] J. Ahlberg, "An Active Model for Facial Feature Tracking," *EURASIP Applied Signal Processing*, Hindawi Publish. Corp., No. 6, 2002, pp. 566-571.

[3-4] S. Arca, P. Campadelli, and R. Lanzarotti, "A face recognition system based on local feature analysis", in *Proc. 4th Int'l Conf. on Audio- and Video-based Biometric Person Authentication (AVBPA'03)*, Guildford, UK, June 9-11, 2003, pp. 182-189.

[3-5] J. J. Atick, P. M. Griffin and A. N. Redlich, "Statistical Approach to Shape from Shading: Reconstruction of 3D Face Surfaces from Single 2D Images," *Neural Computation*, Vol. 8, No. 6, July 1995, pp. 1321-1340.

[3-6] J. J. Atick, P. M. Griffin, A. N. Redlich, "FaceIt: face recognition from static and live video for law enforcement", in *Proc. SPIE Human Detection and Positive Identification: Methods and Technologies*, Vol. 2932, Jan. 1997, pp. 176-187.

[3-7] R. Bajcsy, and S. Kovacic, "Multiresolution elastic matching," *Computer Vision, Graphics, and Image Proc.*, Vol. 46, 1989, pp. 1-21.

[3-8] M. S. Bartlett, H. M. Lades, and T. J. Sejnowski, "Independent component representations for face recognition", in *Proc. SPIE Symp. on Electr. Imaging: Science and Technology, Conf. on Human Vision and Electr. Imaging III*, Vol. 3299, San Jose, CA, 1998, pp. 528-539.

[3-9] B. Bascle, A. Blake, and A. Zisserman, "Motion deblurring and superresolution from an image sequence," in *Proc. 4th European Conf. on Computer Vision* (ECCV'96), Cambridge, England, Apr. 1996, pp. 573-581.

[3-10] P. N. Belhumeur, J. P. Hespanha, D. J. Kriegman, "Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection," *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. 19, No. 7, July 1997, pp. 711-720.

[3-11] S. Ben-Yacoub, B. Fasel and J. Lüttin, "Fast Face Detection using MLP and FFT," in *Proc. 2nd Int'l Conf. on Audio- and Video-based Biometric Person Authentication (AVBPA'99)*, Washington DC, Mar. 22-23, 1999, pp. 31-36.

[3-12] D. M. Blackburn, J. M. Bone, and P. J. Phillips, "FRVT 2000 Report," *Technical Report*, http://www.frvt.org.

[3-13] V. Bruce and A. Young, "In the Eye of the Beholder: The Science of Face Perception," *Oxford University Press*, New York, 1998.

[3-14] R. Brunelli and T. Poggio, "Face Recognition : Features versus Templates," *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. 15, No. 10, Oct. 1993, pp. 1042-1051.

[3-15] H. H. Bülthoff, S. Y. Edelman and M. J. Tarr, "How are three-dimensional objects represented in the brain?," *A.I. Memo No. 1479*, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA, 1994.

[3-16] D. J. Burr, "Elastic Matching of Line Drawings," *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. 3, No. 6, Nov. 1981, pp. 708-713.

[3-17] T. C. Chang, T. S. Huang and C. Novak, “Facial Feature Extraction from Color Images,” in *Proc. 12th IAPR Int’l Conf. Pattern Recognition (ICPR’94)*, Jerusalem, Israel, Oct. 1994, pp. 39-43.

[3-18] R. Chellappa, C. L. Wilson and S. Sirohey, “Human and Machine Recognition of Faces : A Survey,” *Proc. IEEE*, Vol. 83, No. 5, May 1995, pp. 705-740.

[3-19] K. Chung, S.C. Kee, S.R. Kim, “Face Recognition Using Principal Component Analysis of Gabor Filter Responses,” in *Proc. Int’l Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems*, Corfu, Greece, Sep. 26 - 27, 1999, pp. 53-57.

[3-20] T. F. Cootes and C. J. Taylor, “Active Shape Models - Smart Snakes,” in *Proc. British Machine Vision Conference*, Springer Verlag, 1992, pp. 9-18.

[3-21] T. F. Cootes, C. J. Taylor, A. Lanitis, D. H. Cooper and J. Graham, “Building and Using Flexible Models Incorporating Grey-Level Information,” in *Proc. 4th Int. Conf. on Computer Vision (ICCV’93)*, 1993, pp. 242-246.

[3-22] T. F. Cootes, G. V.Wheeler, K. N.Walker and C. J. Taylor, “View-Based Active Appearance Models,” *Image and Vision Computing*, Vol. 20, 2002, pp. 657-664.

[3-23] G. W. Cottrell, M. N Dailey, C. Padgett and R. Adolphs, “Is *All* Face Processing Holistic? The View from UCSD,” in M. Wenger and J. Townsend (Eds.), *Computational, Geometric, and Process Perspectives on Facial Cognition: Contexts and Challenges,"* Scientific Psychlogy Series, Lawrence Erlbaum Associates Publishers, Mahwah, NJ, 2001, pp. 347-395.

[3-24] I. Craw and P. Cameron, “Parameterising images for recognition and reconstruction,” in *Proc. British Machine Vision Conf. (BMV’91)*, Springer Verlag, London, 1991, pp. 367-370.

[3-25] I. Craw, N. Costen, T. Kato, G. Robertson and S. Akamatsu, “Automatic Face Recognition: Combining Configuration and Texture,” in *Proc. Int'l Workshop on Automatic Face- and Gesture-Recognition*, Zürich, Switzerland, June 26-28, 1995, pp. 53-58.

[3-26] I. Craw, N. Costen, T. Kato, D. Akamatsu, “How Should We Represent Faces for Automatic Recognition,” *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. 21, No. 8, Aug. 1999, pp. 725-736.

[3-27] B. Duc, “Feature design : applications to motion analysis and identity verification,” *Ph.D. Thesis No. 1741*, Ecole Polytechnique Fédérale de Lausanne, Switzerland, 1997.

[3-28] G. J. Edwards, T. F. Cootes, and C. J. Taylor, “Face Recognition using Active Appearance Models,” in *Proc. 5th European Conf. on Computer Vision (ECCV’98)*, Vol. 2, University of Freiburg, Germany, June 1998, pp. 581-695.

[3-29] M. J. Farah, M. Drain, and J. N. Tanaka, “What Is ‘Special’ About Face Perception,” *Psychological Review*, Vol. 105, No. 3, 1998, pp. 482-498.

[3-30] B. Fasel, “Fast Multi-Scale Face Detection,” *IDIAP Communication IDIAP-COM-98-04*, Martigny, Switzerland, 1998.

[3-31] M. A. Fischler and R. A. Elschlager, “The Representation and Matching of Pictorial Structures,” *IEEE Trans. Comput.*, Vol. 22, No. 1, Jan. 1973, pp. 67-92.

[3-32] T. Fromherz, P. Stucki and M. Bichsel, “A Survey of Face Recognition,” *MML Techical Report No 97.01*, Department of Computer Science, University of Zürich, Switzerland, 1997.

[3-33] T. Fromherz, "Face Recognition: a Summary of 1995-1997," *ICSI Technical Report TR-98-027*, International Computer Science Institute, Berkeley, CA, 1998.

[3-34] K. Fukunaga, "Introduction to Statistical Pattern Recognition," 2nd Ed., *Academic Press*, San Diego, USA, 1990.

[3-35] F. Galton, "Inquiries into human faculty and its development," *Macmillan*, London, UK, 1883.

[3-36] A. J. Goldstein, L.D. Harmon, and A.B. Lesk, "Identification of Human Faces," in *Proc. IEEE*, Vol. 59, No. 5, May 1971, pp. 748-759.

[3-37] S. Gong, S. J. McKenna, and A. Psarrou, "Dynamic Vision," *Imperial College Press*, London, UK, 2000.

[3-38] G. G. Gordon, "Face recognition based on depth maps and surface curvature," in *Proc. SPIE Geometric Methods in Computer Vision*, Vol. 1570, San Diego, CA, June 1991, pp. 234-246.

[3-39] G. G. Gordon, "Face Recognition from Frontal and Profile Views," in *Proc. Int'l Workshop on Automatic Face- and Gesture-Recognition*, Zürich, Switzerland, June 26-28, 1995, pp. 47-52.

[3-40] H. P. Graf, T. Chen, E. Petajan and E. Cosatto, "Locating Faces and Facial Parts," in *Proc. Int'l Workshop on Automatic Face- and Gesture-Recognition*, Zürich, Switzerland, June 26-28, 1995, pp. 41-46.

[3-41] R. Gross, J. Shi , and J. Cohn, "Where to go with Face Recognition," in *Proc. 3rd Workshop on Empirical Evaluation Methods in Computer Vision*, Kauai, Hawaii, Dec. 11-13, 2001.

[3-42] S. Gutta, J. Huang, D. Singh and I. Shah, "Benchmark Studies on Face Recognition," in *Proc. Int'l Workshop on Automatic Face- and Gesture-Recognition*, Zürich, Switzerland, June 26-28, 1995, pp. 227-231.

[3-43] P. L. Hallinan, G. G. Gordon, A. L. Yuille, P. Giblin and D. Mumford, "Two- and Three Dimensional Patterns of the Face," *A.K. Peters Eds*, Natick, MA, 1999.

[3-44] P. Hancock, V. Bruce and M. Burton, "A Comparison of Two Computer-based Face Identification Systems with Human Perception of Faces," *Vision Research*, Vol. 38, Aug. 1998, pp. 2277-2288.

[3-45] C. Han, H. M. Liao, G. Yu and L. Chen, "Fast Face Detection via Morphology-based Pre-processing," *Pattern Recognition*, Vol. 33, No. 10, Oct. 2000, pp. 1701-1712.

[3-46] A. Haro, I. Essa, and M. Flickner, "Detecting and Tracking Eyes by Using their Physiological Properties, Dynamics and Appearance," in *Proc. IEEE Computer Society Conf. on Computer Vision and Pattern Recognition (CVPR'00)*, June 2000, pp. 163-168.

[3-47] R. Herpers, G. Verghese, L. Chang, K. Darcourt, K. Derpanis, R.F. Enenkel, J. Kaufman, M. Jenkin, E. Milios, A. Jepson and J. K. Tsotsos, "An Active Stereo Vision System for Recognition of Faces and Related Hand Gestures," in *Proc. 2nd Int'l Conf. on Audio- and Video-based Biometric Person Authentication (AVBPA'99)*, Washington DC, Mar. 22-23, 1999, pp. 217-223.

[3-48] T. Heseltine, N. Pears and J. Austin, "Evaluation of Image Pre-processing Techniques for Eigenface Based Face Recognition," in *Proc. 2nd SPIE Int'l Conf. on Image and Graphics*, Vol. 4875, 2002, pp. 677-685.

[3-49] A. Jacquin and A. Eleftheriadis, "Automatic location tracking of faces and facial features in video sequences," in *Proc. Int'l Workshop on Automatic Face- and Gesture-Recognition*, Zürich, Switzerland, June 26-28, 1995, pp. 142-147.

[3-50] J. Jones and L. Palmer, "An evaluation of the two-dimensional gabor filter model of simple receptive fields in cat striate cortex," *Neurophysiology*, Vol. 58, No. 6, 1987, pp. 1233- 1258.

[3-51] M. Kirby and L. Sirovich, "Application of the Karhunen-Loeve Procedure for the Characterization of Human Faces," *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. 12, No. 1, 1990, pp. 103-108.

[3-52] T. Kanade, "Picture processing system by computer complex recognition," *Ph.D. Thesis*, Department of Information Science, Kyoto University, Japan, 1973.

[3-53] J. Kittler, R. Ghaderi, T. Windeatt, and J. Mats, "Face Identification and Verification via ECOC", in *Proc. 3rd Int'l Conf. on Audio- and Video-based Biometric Person Authentication (AVBPA'01)*, Halmstad, Sweden, June 6-8, 2001, pp. 1-13.

[3-54] W. Konen and E. Schulze-Krüger, "ZN-Face: A system for access control using automated face recognition," in *Proc. Int'l Workshop on Automatic Face- and Gesture-Recognition*, Zürich, Switzerland, June 26-28, 1995, pp. 18-23.

[3-55] C. Kotropoulos, A. Tefas and I. Pitas, "Morphological Elastic Graph Matching applied to frontal face authentication under well-controlled and real conditions," *Pattern Recognition*, Vol. 33, No. 12, 2000, pp. 1935-1947.

[3-56] M. Lades, J. C. Vorbrügen, J. Buhmann, J. Lange, C. von der Malsburg, R. P. Würtz and W. Konen, "Distortion Invariant Object Recognition in the Dynamic Link Architecture," *IEEE Trans. Comp.*, Vol. 42, No. 3, Mar. 1993, pp. 300-311.

[3-57] M. Lades, "Face Recognition Technology," *in "Handbook of Pattern Recognition and Computer Vision," 2nd Edition, C.H. Chen, L.F. Pau and P.S.P. Wang (Eds.)*, World Scientific Publishing Company, 1988, pp. 667-683.

[3-58] A. Lanitis, C. J. Taylor, and T. F. Cootes, "Automatic Interpretation and Coding of Face Images using Flexible Models," *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. 19, No. 7, 1997, pp. 743-756.

[3-59] C. Liu and H. Wechsler, "Evolutionary Pursuit and its Application to Face Recognition", *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. 22, No. 6, June 2000, pp. 570-582.

[3-60] C. Liu and H. Wechsler, "Robust Coding Schemes for Indexing and Retrieval from Large Face Databases," *IEEE Trans. Image Processing*, Vol. 9, No. 1, Jan. 2000, pp. 132-137.

[3-61] C. Liu and H. Wechsler, "Gabor Feature Based Classification Using the Enhanced Fisher Linear Discriminant Model for Face Recognition," *IEEE Trans. Image Processing*, Vol. 11, No. 4, Apr. 2002, pp. 467-476.

[3-62] C. von der Malsburg, "Pattern Recognition by Labeled Graph Matching," *Neural Networks*, Vol. 1, 1988, pp. 141-148.

[3-63] S. Marchand-Maillet and B. Mérialdo, "Pseudo two-dimensional Hidden Markov Models for face detection in colour images," in *Proc. 2nd Int'l Conf. on Audio- and Video-based Biometric Person Authentication (AVBPA'99)*, Washington DC, Mar. 22-23, 1999, pp. 13-18.

[3-64] A. M. Martínez, "Recognizing Imprecisely Localized, Partially Occluded and Expression Variant Faces from a Single Sample per Class," *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. 24, No. 6, June 2002, pp. 748-763.

[3-65] A. M. Martínez, and A. C. Kak, "PCA versus LDA", *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. 23, No. 2, Feb. 2001, pp. 228-233.

[3-66] T. Maurer and C. von der Malsburg, "Single View Based Recognition of Faces Rotated in Depth," in *Proc. Int'l Workshop on Automatic Face- and Gesture-Recognition*, Zürich, Switzerland, June 26-28, 1995, pp. 248-253.

[3-67] B. T. Messmer, "Efficient Graph Matching Algorithmus for Preprocessed Model Graphs," *Ph.D. Thesis*, University of Bern, Switzerland, 1995.

[3-68] B. Moghaddam and A. Pentland, "Maximum Likelihood Dectection of Faces and Hands," in *Proc. Int'l Workshop on Automatic Face- and Gesture-Recognition*, Zürich, Switzerland, June 26-28, 1995, pp. 122-128.

[3-69] B. Moghaddam, C. Nastar and A. Pentland, "Bayesian Face Recognition Using Deformable Intensity Surfaces," *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'96)*, San Francisco, CA, June 18-20, 1996, pp. 638-645.

[3-70] B. Moghaddam, W. Wahid and A. Pentland, "Beyond Eigenfaces : Probabilistic Matching for Face Recognition," *MIT Media Laboratory Perceptual Computing Section, Technical Report No. 443*, appears in *Proc. 3rd IEEE Int'l Conf. on Automatic Face and Gesture Recognition*, Nara, Japan, Apr. 1998, pp. 30-35.

[3-71] A.V. Nefian, and M. H. Hayes III, "Face Recognition using and Embedded HMM," in *Proc. 2nd Int'l Conf. on Audio- and Video-based Biometric Person Authentication (AVBPA'99)*, Washington DC, Mar. 22-23, 1999, pp. 19-24.

[3-72] R. Newman, Y. Matsumoto, S. Rougeaux and A. Zelinsky, "Real-Time Stereo Tracking for Head Pose and Gaze Estimation," in *Proc. 4th Int'l. Conf. on Automatic Face and Gesture Recognition (FG'2000)*, Grenoble, France, Mar. 28-30, 2000, pp. 122-128.

[3-73] K. Okada, J. Steffens, T. Maurer, H. Hong, E. Elagin, H. Neven and C. von der Malsburg, "The Bochum/USC Face Recognition System, and how it fared in the FERET phase III test," in H. Wechsler, P.J. Phillips, V. Bruce, F. Soulié, and T. Huang, editors, "*Face Recognition : From Theory to Applications*," Springer-Verlag, Heidelberg, Germany, 1998, pp. 168-205.

[3-74] S. Pankanti, S. Prabhakar, and A. K. Jain, "On the Individuality of Fingerprints," *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. 24, No. 8, Aug. 2002, pp. 1010-1025.

[3-75] J.-S. Park, Y. H. Oh, S. C. Ahn, and S.-W. Lee, "Glasses Removal from Facial Image Using Recursive PCA Reconstruction", in *Proc. 4th Int'l Conf. on Audio- and Video-based Biometric Person Authentication (AVBPA'03)*, Guildford, UK, June 9-11, 2003, pp. 369-376.

[3-76] P. S. Penev and J. J. Atick, "Local Feature Analysis: A general statistical theory for object representation," *Network: Computation in Neural Systems*, Vol.7, No.3, May 1996, pp. 477-500.

[3-77] P. S. Penev, "Local Feature Analysis: A Statistical Theory for Information Representation and Transmission", *PhD Thesis*, Rockefeller University, NY, May 1998.

[3-78] P. S. Penev, "Redundancy and Dimensionality Reduction in Sparse-Distributed Representations of Natural Objects in Terms of Their Local Features", in Proc. *Neural Information Processing Systems (NIPS)*, Nov. 28-30, 2000, Denver, CO, pp. 901-907.

[3-79] A. Pentland, B. Moghaddam, and T. Starner , “View-based and Modular Eigenspaces for Face Recognition,” in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'94)*, Seattle, WA, June 1994, pp. 84-91.

[3-80] P. J. Phillips, H. Moon, P. Rauss and S. A. Rizvi, “The FERET Sep. 1996 Database and Evaluation Procedure,” in *Proc. 1st Int'l Conf. on Audio- and Video-based Biometric Person Authentication (AVBPA'97)*, Crans-Montana, Switzerland, Mar. 12-14, 1997, pp. 395-402.

[3-81] P. J. Phillips, P. Grother, R. J. Micheals, D. M. Blackburn, E. Tabassi, and M. Bone, “FRVT 2002 Report,” *Technical Report*, available on http://www.frvt.org.

[3-82] M. Pötzsch, T. Maurer, L. Wiskott, and C. von der Malsburg, “Reconstruction from graphs labeled with responses of Gabor filters,” in *Proc. Int’l Conf. on Artificial Neur. Networks (ICANN’96)*, Bochum, Germany, July 16-19, 1996, pp. 845-850.

[3-83] S. S. Rakover and B. Cahlon, “Face Recognition: Cognitive and Computational Processes,” *Advances in Consciousness Research series,* John Benjamins Publishing Company, Amsterdam, The Netherlands, 2001.

[3-84] H. A. Rowley, S. Baluja, and T. Kanade, “Neural Network-Based Face Detection,” *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. 20, No. 1, Jan. 1998, pp. 23-38.

[3-85] H. A. Rowley, S. Baluja and T. Kanade, “Rotation Invariant Neural Network-Based Face Detection,” in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR’98)*, Santa Barbara, CA, June 23-25, 1998, pp. 38-44.

[3-86] A. Samal and P. A. Iyengar, “Automatic Recognition and Analysis of Human Faces and Facial Expressions: A Survey,” *Pattern Recognition*, Vol. 25, No. 1, 1992, pp. 65-77.

[3-87] A. Schubert, “Detection and Tracking of Facial Features in Real Time Using a Synergistic Approach of Spatio-Temporal Models and Generalized Hough-Transform Techniques,” in *Proc. 4th Int'l. Conf. on Automatic Face and Gesture Recognition (FG'2000)*, Grenoble, France, Mar. 28-30, 2000, pp. 116-121.

[3-88] S. A. Sirohey and A. Rosenfeld, “Eye detection in a face image using linear and nonlinear filters,” *Pattern Recognition*, Vol. 34, No. 7, 2001, pp. 1367-1391.

[3-89] L. Sirovich and M. Kirby, “Low-dimensional procedure for the characterization of human faces,” *J. Optical Society of America A*, Vol. 4, No. 3, Mar. 1987, pp. 519-524.

[3-90] F. Smeraldi, O. Carmona and J. Bigün, “Real-time Head Tracking by Saccadic Exploration and Gabor Decomposition,” in *Proc. Int'l Workshop on Advanced Motion Control (AMC'98)*, Coimbra, Portugal, June 29-July 1, 1998, pp. 684-687.

[3-91] B. Takacs and H. Wechsler, “Face Location Using a Dynamic Model of Retinal Feature Extraction,” in *Proc. Int'l Workshop on Automatic Face- and Gesture-Recognition*, Zürich, Switzerland, June 26-28, 1995, pp. 243-247.

[3-92] A. Tefas, C. Kotropoulos, and I. Pitas, “Using Support Vector Machines to Enhance the Performance of Elastic Graph Matching for Frontal Face Authentication,” *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. 23, No. 7, July 1997, pp. 735-746.

[3-93] A. Tefas, C. Kotropoulos, and I. Pitas, “Face Verification Using Elastic Graph Matching Base on Morphological Signal Decomposition,” *Signal Processing*, Elsevier, No. 82, 2002, pp. 833-851.

[3-94] J. Thiem, G. Hartmann, “Biology-inspired Design of Digital Gabor Filters upon a Hexagonal Sampling Scheme,” in *Proc. 15th Int’l Conf. on Pattern Recognition (ICPR2000)*, Vol. 3, Barcelona, Spain, Sep. 03-08, 2000, pp. 449-452.

[3-95] P. Thompson, "Margaret Thatcher: a new illusion," *Perception*, Vol. 9, No. 4, 1980, pp. 483-484.

[3-96] M. Turk and A. Pentland, "Eigenfaces for Recognition," *Cognitive Neuroscience*, Vol. 3, No. 1, 1991, pp. 71-86.

[3-97] S. Ullman and R. Basri, "Recognition by Linear Combinations of Models," *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. 13, No. 10, Oct. 1991, pp. 992-1006.

[3-98] W. R. Uttal, R. Kakarala, S. Dayanand, T. Sheperd, J. Kalki, C. F. Lunskis Jr. and N. Liu, "Computational Modeling of Vision: The Role of Combination," *Marcel Dekker* ed., New-York, 1999.

[3-99] D. Valentin, H. Abdi, A. J. O'Toole and G. W. Cottrell, "Connectionist Models of Face Processing: A Survey," *Pattern Recognition*, Vol. 27, No. 9, 1994, pp. 1209-1230.

[3-100] L. Wiskott, J.-M. Fellous, N. Krüger and C. von der Malsburg, "Face Recognition and Gender Determination," in *Proc. Int'l Workshop on Automatic Face- and Gesture-Recognition*, Zürich, Switzerland, June 26-28, 1995, pp. 92-97.

[3-101] L. Wiskott, J.-M. Fellous, N. Krüger and C. von der Malsburg, "Face Recognition by Elastic Bunch Graph Matching," *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. 19, No. 7, July 1997, pp. 775-779.

[3-102] M.-H. Yang, D.J. Kriegman, and N. Ahuja, "Detecting Faces in Images: A Survey," *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. 24, No. 1, Jan. 2002, pp. 34-58.

[3-103] K. Yu, X. Jiang, H. Bunke, "Face Recognition by Facial Profile Analysis," in *Proc. Int'l Workshop on Automatic Face- and Gesture-Recognition*, Zürich, Switzerland, June 26-28, 1995, pp. 208-213.

[3-104] A. L. Yuille, "Deformable Templates for Face Recognition," *Journal of Cognitive Neuroscience*, Vol. 3, No. 1, 1991, pp. 59-70.

[3-105] J. Zhang, Y. Yan and M. Lades, "Face Recognition: Eigenface, Elastic Matching, and Neural Nets," *Proc. IEEE*, Vol. 85, No. 9, Sep. 1997, pp. 1423-1435.

[3-106] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld, "Face Recognition: A Literature Survey," *ACM Computing Surveys*, Vol. 35, No. 4, Dec. 2003, pp. 399-458.

[3-107] L. Zhaoping, "Optimal Sensory Encoding," in M. A. Arbib (Ed.), "*The Handbook of Brain Theory and Neural Networks*," 2nd Edition, MIT Press, Cambridge, MA, Nov. 2002.

[3-108] M. Zobel, A. Gebhard, D. Paulus, J. Denzler and H. Niemann, "Robust Facial Feature Localization by Coupled Features," in *Proc. 4th Int'l. Conf. on Automatic Face and Gesture Recognition (FG'2000)*, Grenoble, France, Mar. 28-30, 2000, pp. 2-7.

[3-109] http://www.cs.colostate.edu/evalfacerec/index.html

Chapter 4

Elastic Graph Matching Algorithm Applied to Face Verification

This chapter presents the different building blocks involved in the Elastic Graph Matching (EGM) algorithm introduced in Subsection 3.4.5., and discusses solutions found in the literature and original contributions proposed in this thesis. The proposed algorithm improvements aim both at increasing the robustness in a mobile context and at reducing the computational complexity and thus the power-consumption of the corresponding hardware implementation.

Firstly, an overview of the dataflow of EGM applied to face authentication is given in Section 4.1 and enables the identification of key issues and building blocks on which the thesis is focused. Then an extended and original discussion of the influence of illumination variations on the face matching process will be carried out in Section 4.2. This contribution is accompanied by a review of several existing illumination normalization techniques, comparing their efficiency, reproducibility, and computational complexity in light of the previous analysis. Then, a detailed description of three types of pattern matching based feature extractors (namely Gabor filters, modified discrete cosine transform, and mathematical morphology operators) will be presented in Section 4.3. Their systematic comparison from both performance and computational complexity point of view represents also a distinct contribution. A description of feature-space reduction techniques will be discussed in the framework of EGM in Section 4.4, while Section 4.5 will concentrate on efficient matching strategies related to function minimization will also introduce several innovative contributions. Finally classifier design will be discussed in Section 4.6, and the performance of the different variants will be examined in Section 4.7. Conclusions drawn from the study of these building blocks will serve in Chapter 6 for the implementation of a low-power system-on-chip for face authentication.

4.1 Algorithm architecture

The general enrolment and testing procedures for the EGM algorithm are represented in Figure 4-1. During enrolment, a template is extracted from a single reference image. If several training samples are available, several templates can be added to the database or some statistics can be extracted to increase the robustness of the matching (e.g. Principal Component Analysis or Linear Discriminant Analysis, see Section 4.4). These templates and statistics are then used in the testing stage, where a new view containing a face is presented together with an identity claim (in the case of authentication).

As seen in Subsection 3.4.5, EGM consists in extracting local information at sparse discrete locations regularly sampled over the face, and in finding a correspondence (i.e. a graph isomorphism) between features of the test and reference graphs. Consequently, the usual pattern recognition process of first extracting features, and then using them for classification, is slightly modified here because of the iterative classification process. Indeed, based on the calculated distance, a new test template is extracted with a different graph configuration until the best match is found. At this stage the final distance is sent to a one dimensional classifier.

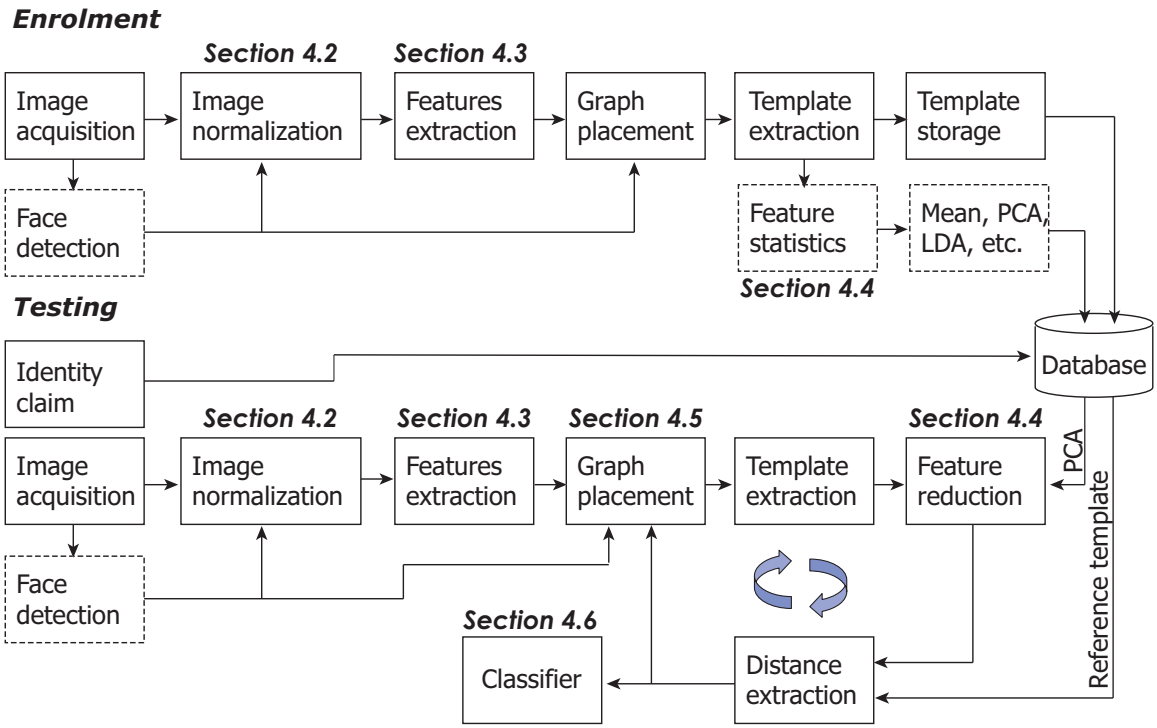


Figure 4-1: Enrolment and testing procedures in the case of face authentication.
Sections where each block is discussed are indicated. Major algorithmic contributions of this thesis can be found in Sections 4.2, 4.3 and 4.5.

Face detection during the enrolment is only necessary if this process requires to be fully automatic and unattended (i.e. without human intervention to place the reference graph). This would be the case in surveillance applications but obviously not in the context of mobile face authentication. This step will thus not be implemented here. During the testing phase, the face detection can be used as an a priori information to normalize the image or to initialize the graph placement so as to reduce the number of iterations. Similarly to the enrolment process, face detection can be however removed at the cost of a higher number of iterations. As a consequence, no face detection will be considered in this chapter, although some possible implementations will be described later in Section 5.1 with the aim of reusing algorithms and hardware for other applications (e.g. surveillance).

4.2 Image normalization

Image pre-processing or normalization may be necessary to compensate for variations of the illuminant (e.g. white balancing) or of the viewpoint (e.g. image warping). Because the Elastic Graph Matching algorithm intrinsically performs a kind of warping during the matching, this last normalization seems not necessary, while some photometric normalization will be required as demonstrated in following sections.

Illuminant variation is one of the most difficult problems for face recognition and probably one of the major limiting factors to high performance systems, which if for instance more critical than pose variation [4-1]. Accordingly, many different methods were proposed in the literature to compensate for this problem. Before looking at normalization methods and at the robustness of the different feature extraction techniques that are mostly used, the actual effect of illuminant variations on the acquired image will first be defined, taking into account several effects that are generally not discussed in the face recognition community (e.g. variation of light spectrum). Then, several techniques for modelling and removing the illuminant dependence will be reviewed and their performance evaluated from a theoretical point of view and in real situations.

4.2.1 Effects of illumination variations

In the following discussion, *variations of illuminant* encompass both:

- variations of the *perceived intensity*, which will make objects appear brighter or darker, and
- variations of the *illuminant spectrum*, which will alter the perceived colors of the objects *and* the contrasts between colored regions of an object (or between objects), even in *grayscale* images.

While the first category of variations is largely discussed in the face recognition literature, no extensive discussion of the second one could be found. It is however a largely known problem in machine vision (e.g. in color indexing [4-25]). This subsection will thus contribute to addressing the problem in the context of face recognition.

The following strong assumptions are generally made by authors proposing illumination normalization methods:

- *objects are illuminated directly by the light source*, i.e. no multiple reflections are involved. For instance, this assumption does not hold in a room with a colored wall which is reflecting part of the light (i.e. selected regions of the illuminant spectrum) on the face of the user. However it can be assumed that the perceived intensity created by this source is much smaller than that of the original light source.
- *the face is illuminated uniformly*, i.e. the light source geometry is an ideal plane-wave (light coming from infinity) originating from a far located point light source (approximation of a light bulb).
- *polarization and diffraction effects are not taken into account*.

The light reaching the surface of an object can be reflected in two different manners: either at the *interface* or in the *body* of the surface. The first is generally operating in a mirror-like manner, and is called *specular* reflection, while the second is highly diffused due to the interaction with the matter [4-64]. The specular reflection is usually only observed in some regions of the image, but with a very strong intensity and with a color that is close to the color of the illuminant instead of being close to the color of the object. Because the specular reflection is usually polarized, it is possible to remove it (or at least to detect it) by imaging the object through a polarizer [4-75].

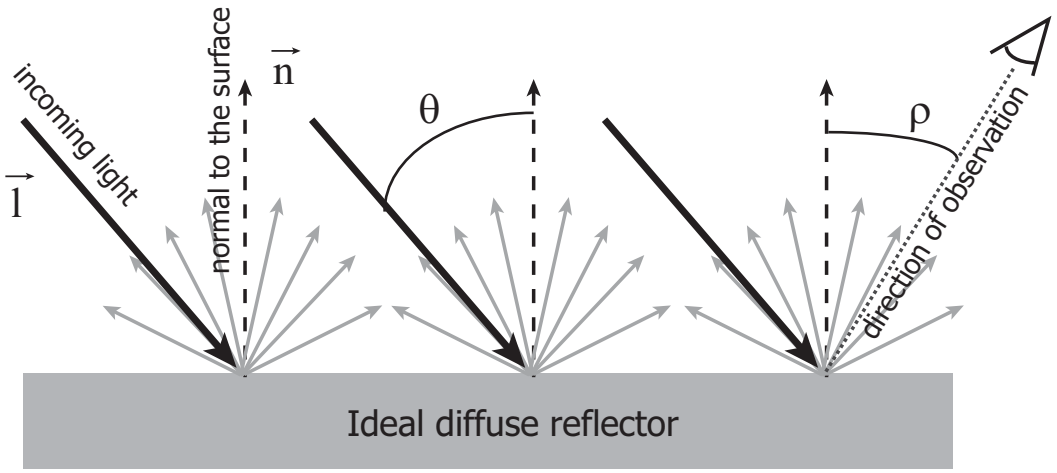


Figure 4-2: Diffuse reflection by a Lambertian reflector.
The light arriving on the surface is reflected equally in all directions (gray arrows). The intensity of the reflected light is only dependent on the angle between the incoming light direction and the normal to the surface of the reflector but not on the direction of observation.

A simple yet efficient model describing diffuse reflection is the so-called Lambertian model (see Figure 4-2), where the intensity of the reflected light follows the following geometrical law:

$$I_{diffuse} = (\vec{n} \cdot \vec{l}) k_d I_{incoming} = k_d I_{incoming} \cos\theta \qquad (4.1)$$

where k_d is the reflectance (also called albedo) of the surface and is dependent on the wavelength λ, θ is the angle between the normal to the surface $\vec{n}$ and the direction of the incident plane wave $\vec{l}$, and $I_{incoming}$ is the light intensity. Eq. 4.1 means also that the quantity of light perceived from a reflection on a lambertian reflector is independent of the viewpoint (represented by angle ρ). However it is clearly dependent of the angle between the normal to the surface and the direction of the illuminant (either plane wave direction or vector from a point light source to the surface).

Since the lambertian model only accounts for the diffuse part of the reflected light, another model is generally used to model both the diffuse and specular components of the reflection. In the *dichromatic reflection model* [4-64] the reflected radiance is composed of two contributions, one due to the light reflected at the interface and one due to the light reflected from the surface body. Moreover each contribution can be separated into a part depending on geometry and a part depending on the wavelength of the illuminant (Eq. 4.2).

$$I_{diffuse}(\lambda, \theta, \rho) = I_i(\lambda, \theta, \rho) + I_b(\lambda, \theta, \rho)$$
$$I_{diffuse}(\lambda, \theta, \rho) = m_i(\theta, \rho)c_i(\lambda) + m_b(\theta, \rho)c_b(\lambda) \quad (4.2)$$

where λ is the wavelength, θ is the angle between the normal to the surface and the incoming light, ρ is the angle between the normal to the surface and the viewer, c_i and c_b are spectral reflectances (of the interface and body respectively), which are independent of the geometry, and m_i and m_b depend only on geometry and not on wavelength.

The Phong model [4-56] largely used in computer graphics for 3D rendering is a particular case of the Dichromatic Reflection Model [4-64] but has no real physical foundation apart from being "pleasant" to the eye. Other more complicated models were proposed, especially to express the reflection as a physical process [4-52].

The assumption that light reaching the sensor came from a single reflection means that *ambient* light is not taken into account. For indoor environments without direct lighting, or outdoor during hazy days, it is however possible that the illumination is solely due to this neglected ambient light. The main characteristic of a perfectly diffuse light[1] is that the reflected light is uniform with respect to geometry but is however dependent on the wavelength, i.e. a third term depending only on the wavelength should be added (Eq. 4.3).

$$I_{diffuse}(\lambda, \theta, \rho) = m_i(\theta, \rho)c_i(\lambda) + m_b(\theta, \rho)c_b(\lambda) + L(\lambda) \quad (4.3)$$

Additionally, shadows are not taken into account in the lambertian and dichromatic models described above. Shadows can be of two types [4-66]: an *attached shadow* occurs when the angle θ between the normal to the surface and the incoming light direction is obtuse (i.e. lighting coming from the other side of a face); a *cast shadow* occurs when the object point is obstructed from the light source by another object (typically the nose tip). In the first case, only ambient light remains[2], which means that the perceived intensity is almost constant on the attached shadow if the object reflectance in this region is constant. In the case of cast shadow however, it is not possible to model the effect correctly, and chances that the performance degrades with increasing number of cast shadows is high. But if this type of shadow is sufficiently large, it is perhaps possible to minimize its influence by locally compensating for it.

1. "Perfectly diffuse" lights are rarely encountered in practice.
2. In the Dichromatic Reflection Model, this means that $m_i = m_b = 0$.

Following Martinkauppi [4-48] it is possible to consider the skin as diffusing - i.e. matt - because the surface of the skin does not allow for large specular reflections and is optically inactive (i.e. no fluorescence). The glossiness of the skin could be due mainly to sweat, skin oil or chemical products covering the surface. Therefore it can be assumed that the contribution of m_i in Eq. 4.2 is relatively small as compared to the contribution of m_b. To simplify the rest of the discussion, we will therefore consider only this latter contribution, i.e. an "enhanced Lambertian model" taking into account spectral distributions. It should be noted also that m_b can vary - smoothly if the geometry does not present pronounced edges - from pixel to pixel due to the object geometry where as c_b should be approximately the same on surfaces of the same colour and change e.g. near the mouth or the eyes. Therefore Eq. 4.3 becomes:

$$I_{diffuse}(x, y;\lambda, \theta, \rho) = m_b(x, y;\theta, \rho)c_b(\lambda) + L(\lambda) \qquad (4.4)$$

Spectral components $c_b(\lambda)$ and $L(\lambda)$ will now be discussed. The intensity perceived on a photo-diode of the image sensor depends on the spectral power distribution *I(l)* of the illuminant, the spectral reflectance *R(l)* of the object on which light is reflected, and the spectral sensitivity $\eta(\lambda)$ of the photo-diode (we do not distinguish this sensitivity from the spectral transmission of the color filters that are deposited on the surface of the photo-diode and that actually lead to its particular spectral sensitivity $\eta(\lambda)$).

$$P_i(x, y) = \int \eta_i(\lambda)I(\lambda, \theta)R(x, y;\lambda, \theta)d\lambda \qquad (4.5)$$

where *i* represents one of the tri-stimuli *red* (R), *green* (G), and *blue* (B). Note that *x* and *y* correspond to the photo-diode coordinates on the image sensor. Light received on this photo-diode depends actually on the world coordinates of the corresponding object point, and also on θ, although it is not mentioned explicitly in Eq. 4.5 because it is independent of the viewing angle.

Using the modified Dichromatic Reflection Model, the spectral reflectance *R* can be replaced in Eq. 4.5 with the expression for the diffused reflection $I_{diffuse}$ (i.e. Eq. 4.4):

$$\begin{aligned} P_i(x, y) &= \int \eta_i(\lambda)I(\lambda)[c_b(\lambda)m_b(x, y;\theta, \rho) + L(\lambda)]d\lambda \\ P_i(x, y) &= m_b(x, y;\theta, \rho)\int \eta_i(\lambda)I(\lambda)c_b(x, y, \lambda)d\lambda + \int \eta_i(\lambda)I(\lambda)L(\lambda)d\lambda \end{aligned} \qquad (4.6)$$

Features extraction from grayscale images such as edges, textures or shapes will thus be influenced by the following variations:

Issue 1: modification of the light position relatively to the geometry of the object i.e. of $m_b(\theta)$ but not of the viewer.

Issue 2: modification of the sensitivity of the sensor or of the light intensity i.e. $\eta_i(\lambda)$ or $\int I(\lambda)d\lambda$.

Issue 3: modification of the spectral density of the illuminant i.e. $I(\lambda)$.

Nevertheless, only the information contained in c_b and in m_b is of interest for pattern recognition, while the rest can be considered as noise. Solutions to circumvent issues 1 to 3 are now discussed.

1) Modification of the light position w.r.t. the geometry of the object

We assume here that the m_b parameter follows the Lambertian model with attached shadows:

$$m_b(x, y;\theta, \rho) = max\{\cos\theta(x, y), 0\} \qquad (4.7)$$

With no ambient light (i.e. L=0), an angle between two plane surfaces will produce a visible edge (see Figure 4-3), depending on the angle between the surfaces and the light direction. For instance, if we apply a 3x3 Sobel filter [4-33] on the luma image, the detected edge will be more distinct if the angles θ for the two planes are more different. Thus, for the same object geometry, the contrast of the angle will be different depending on the orientation of the light. Now with L>0, and with m_b=0 (e.g. light coming from the bottom), the illuminant will be constant, i.e. all colors will appear the same. Therefore the edge illustrated in Figure 4-3 will not appear at all if both planes are of the same color. The perceived brightness will however be different if the colors are different.

The face does not contain sharp edges as illustrated in this example, but contains significant curvature variations, and planes with various orientations. Consequently, edge detection (e.g. used in eye or mouth detection) performed under a direct light or under a diffused (or ambient) one will produce significantly different contrasts (see Figure 4-4).

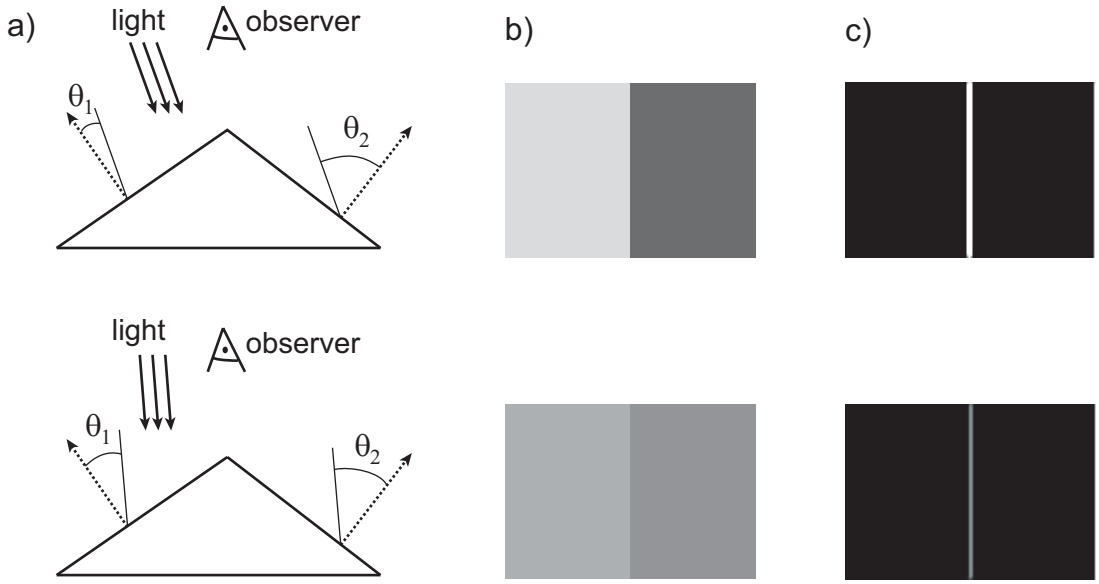


Figure 4-3: Effect of geometry and light direction on contrast of edges.
a) Light arriving on two oriented planes, with two different incident light angles (the observation angle remains unchanged); b) observed reflected intensity on the two planes; c) result of edge detection applied on b) (using a 3x3 Sobel operator).
Note that the contrast between the two planes is higher when θ_1 and θ_2 are more distinct for the two planes (upper situation)

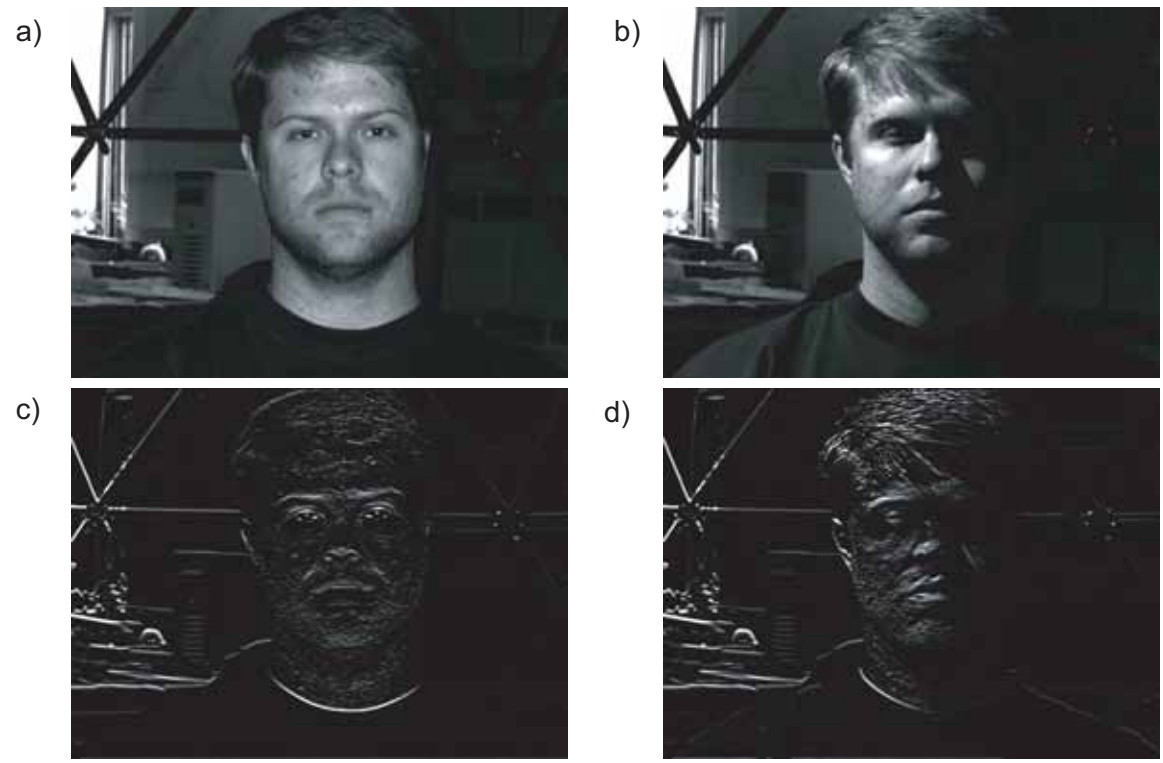


Figure 4-4: Edge detection on faces under different illuminants.
a) face image with frontal light; b) face image with light coming from the side (azimut: 70°; elevation: 45°); c) horizontal edge detection applied on image a); d) horizontal edge detection applied on image b) (images were taken from the Yale B database [4-27]).

2) Modification of the sensitivity of the sensor or of the light intensity

Let us consider the case of CMOS APS image sensors with photo-diodes collecting the energy of the photons [4-11] operating in global-shutter exposure mode.

The exposure time, i.e. the time during which the photo-diode will collect photons energy, will have an influence on the value read for a given tri-stimulus. Taking two images with two different exposure times will lead to different pixel values for the same object. In this thesis, we make the assumption that the relation between sensitivity and exposure time is linear, leading to a normalized spectral sensitivity (i.e. not depending on the exposure time or converter gain), multiplied by the exposure time T_{exp}, which is constant over the whole image (Eq. 4.8).

$$ɔ_i(x, y) = m_b(x, y;\theta, \rho)T_{\exp}\int \eta_i^0(\lambda)I(\lambda)c_b(x, y, \lambda)d\lambda \qquad (4.8)$$

This "illuminant" variation (i.e. two different T_{exp} in two different images) is usually easy to compensate for as the gain T_{exp} is constant over the entire image. This assumption is however not always true, e.g. if there are gain mismatches between pixels of the sensor or more evidently when saturation occurs on the photo-diodes. On the other hand variations of the spectral distribution of η could be handled as variations of the spectral density of the illuminant, i.e. by white-balancing. However, since it is sensor dependent and not illumination dependent, this calibration can be performed once for all.

3) Modification of the spectral density of the illuminant

The EGM algorithm is essentially working on *grayscale* images. A grayscale representation proportional to perceived brightness can be obtained from color image sensors by either computing its *CIE luminance Y* from (R,G,B) components, or alternatively its *luma Y'* from the gamma corrected (R,G,B) components, denoted R', G' and B' [4-59].

$$Y = 0.2125 \cdot R + 0.7154 \cdot G + 0.0721 \cdot B \qquad (4.9)$$

$$Y = 0.299 \cdot R' + 0.587 \cdot G' + 0.114 \cdot B' \qquad (4.10)$$

From Eq. 4.9 and Eq. 4.10 it can be observed that if the spectral power distribution of the illuminant changes not only the perceived color will be different but the perceived brightness too (both luminance and luma). Furthermore, the transform is not univocal, as some colored objects might give a very different luminance under a light containing more blue and less red than the canonical light, while a differently

colored object might give the same perceived brightness under those light conditions. Consequently the contrast between two color regions can largely change with the color (i.e. the power spectral distribution) of the illuminant (see for instance Figure 4-5).

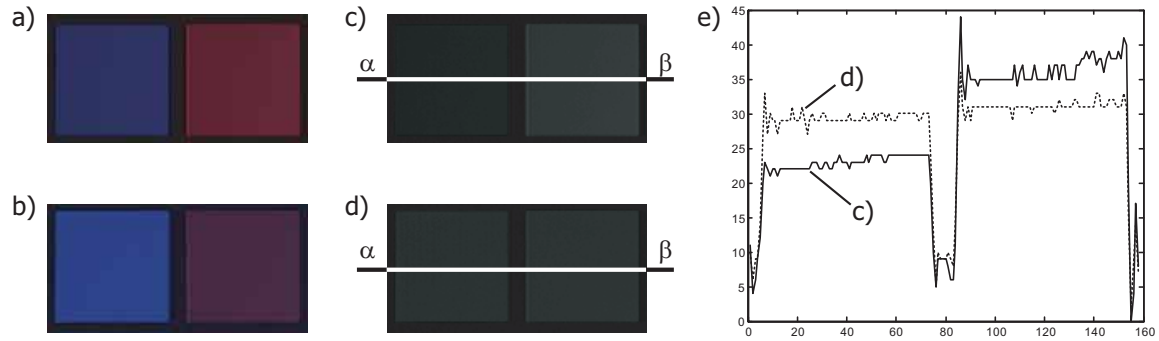


Figure 4-5: Two color patches under two different illuminants and their respective luma.
a) Two patches (blue and red, separated by a black margin) under a Sylvania warm white fluorescent light; b) the same patches under a Solux 4700 K daylight-like light; c) luma of a); d) luma of b); e) contrast along the line α-β for b) (plain line) and c) (dashed line).
It can be observed that the two colors give the same luma under the Solux 4700 K but different luma under the Sylvania. Without taking into account the black region present between the two color patches, but considering a juxtaposition of the two colors, an edge would be detected in one case and not in the other similarly to Figure 4-3. These color patches are details from MacBeth™ images taken from the database of Barnard et al. [4-4].

Methods that propose to recover correct colors under an unknown illuminant from an arbitrary image are called *color constancy* methods. This operation is also sometimes called *white-balancing* for image sensors, because a white reference was traditionally used to calibrate cameras under a new unknown illuminant. In the general form, this operation can be expressed as a linear transform:

$$\begin{bmatrix} \tilde{R} \\ \tilde{G} \\ \tilde{B} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \qquad (4.11)$$

The diagonal matrix is a special simplification known as the von Kries model (or von Kries adaptation):

$$\begin{bmatrix} \tilde{R} \\ \tilde{G} \\ \tilde{B} \end{bmatrix} = \begin{bmatrix} a_{11} & 0 & 0 \\ 0 & a_{22} & 0 \\ 0 & 0 & a_{33} \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \qquad (4.12)$$

This model is valid if the spectral sensitivities of the color filters are narrow-band and non-overlapping (i.e. independent). This strong assumption generally does not hold, but the resulting simplification is still sufficient for most applications. If this is not the case, it is necessary to apply a so-called spectral sharpening [4-22][4-23] before the von Kries approximation to increase the independence of the tri-stimuli.

These three coefficients can be for instance recovered by one of the following very popular white-balancing methods [4-61]:

- white-patch (inspired from Land's Retinex theory);
- grey world;
- gamut mapping.

The first two approaches are very easily implemented but are generally not sufficient to compensate for illuminant color variations. The third method, while being notably more robust, is also seriously more computationally complex and involves a large memory footprint. In a mobile environment, only the two first approaches seem feasible. Qualitative results obtained with a webcam demonstrator showed that none performed sufficiently well all of the time. Nevertheless, with a more constrained environment (e.g. containing assuredly a white region), these methods can prove successful enough.

If the face recognition is based on luma or luminance images, then color constancy can be combined with luma calculation (Eq. 4.10), leading to Eq. 4.13. More information on color constancy can be found in Barnard et al. [4-5][4-6] or Tuceryan and Jain [4-74].

$$\begin{aligned}\tilde{Y} &= \begin{bmatrix}0.2125 & 0.7154 & 0.0721\end{bmatrix}\begin{bmatrix}a_{11} & 0 & 0\\ 0 & a_{22} & 0\\ 0 & 0 & a_{33}\end{bmatrix}\begin{bmatrix}R\\ G\\ B\end{bmatrix}\\ &= \begin{bmatrix}0.2125\cdot a_{11} & 0.7154\cdot a_{22} & 0.0721\cdot a_{33}\end{bmatrix}\begin{bmatrix}R\\ G\\ B\end{bmatrix}\end{aligned} \qquad (4.13)$$

We must stress again that performing poorly color constancy will result in a poor "brightness constancy" as well, as this was already illustrated in Figure 4-5! That is, the perceived brightness will be different under different illuminant colors (i.e. spectral distributions).

Even in the case of a single spectral sensitivity, common to all pixels (i.e. a "true" grayscale image sensor, with no color filter deposited on top of the photo-diodes, or with the same filter deposited on all the diodes, e.g. for infra-red radiation filtering), the problem of variable perceived brightness exists, but in this case it will be impossible to perform "brightness constancy" because at least three independent basis functions (i.e. a tristimulus) are necessary to approximate accurately naturally occurring spectral reflectances [4-31].

4.2.2 Illuminant normalization techniques

This subsection reviews existing solutions proposed in the literature to solve the illumination variation problem, i.e. the three issues mentioned on page 90 and detailed above. Numerous methods have been indeed proposed but we will show that they only partially solve the problem or that they involve a very important computational complexity. We will then identify the remaining problem and propose solutions towards more robust yet less complex face authentication solutions.

Two approaches are possible to remove the influence of illumination variations:

- predict the variability of the features;
- derive new insensitive features (i.e. invariant or nearly invariant).

These two approaches will now be discussed and a solution for low-power mobile applications will be proposed.

a) Variability prediction

Obviously the most intuitive method to achieve perfect illumination prediction with a bottom-up approach (see Subsection 3.3.1) would be to generate a 3D model (from a 3D view, or from a single or multiple 2D views, i.e. what is sometimes called a 2½ model, using e.g. shape-from-shading approaches), then synthesizing a 2D view from this 3D model while modifying lighting (and all spectral distributions of reflectances and lighting, i.e. a kind of *ray tracing* approach) including attached shadows until the obtained 2D view resembles the original one. A novel "re-lighted" (i.e. synthesized) view would then be used for recognition. This approach was proposed e.g. by Blanz and Vetter and coined "morphable models" [4-12]. The FRVT 2002 reported significant improvements on almost all algorithms when using this method. Nevertheless,

this task remains very difficult due to the large number of parameters involved in real scenes (multiple lights, multiple reflections, occlusions) making the use of simplified methods compulsory.

A nice theoretical framework was proposed by Belhumeur et al.: the *illumination cone* [4-8][4-26][4-27]. The underlying idea is that images of convex objects illuminated by an arbitrary number of point light sources located at infinity under a Lambertian hypothesis form a convex cone in $\mathbb{R}^n$ (where n is the number of pixels in the image), and the dimension of this cone equals the number of distinct normals of the object surface. Since all kinds of lighting conditions can be described as a sum of elementary point light sources, it is possible to synthesize a novel view with corrected illuminant from a set of primitives (related to eigenimages). The illumination cone method requires three or more images of the very same scene under different illumination conditions (covering plausible portions of the solid angle) in order to build a basis of the 3D illumination subspace. Although this method does not require an a priori knowledge of the *position* of light sources with respect to the camera viewpoint (as it is the case with some shape from shading approaches), the condition that the three illumination conditions are different (i.e. *mutually independent*) is already a strong assumption. This means for instance that a special setup, or at least a special scenario, is necessary during the enrolment of a new client. The authors claim however that “waving a light source around the scene” is sufficient to generate independent images in the illumination cone and to recover the basis functions.

Basri [4-7] extends this concept of illumination cone by showing that it can be quite accurately approximated by a 9-dimensional linear subspace, by expressing the effects of Lambertian reflectance as being equivalent to a low-pass filter on lighting. Based on the results of Basri, Lee [4-45] shows that there exists a set of nine single light source directions that can be used to derive a subspace that is sufficient to recognize faces under different illumination conditions, without resorting to 3D reconstruction of the model.

While these methods compensate for issues 1 and 2 listed on page 90, they do not directly solve the problem of illuminant spectrum variations (issue 3). For instance, Georghiades reports the excellent performance of illumination cones on the Yale B face database [4-27], which is very constrained in terms of face pose and illuminant spectrum, i.e. only the illuminant direction largely varies, thanks to a geo-

desic alignment of xenon strobes. Belhumeur claims that color can be incorporated into the model at the price of some more degrees of freedom (the cone in $\mathbb{R}^n$ becomes a cartesian product of three cones in $\mathbb{R}^{3n}$) [4-8]. However, no quantitative results are given to support this claim. Under variable pose the illumination cone can be transformed into a *family* of illumination cones. Georghiades et al. synthesized new poses by using a Generalized Bas Relief (GBR) surface of the face and by casting methods using ray-tracing approaches. It is however questionable how the synthesis decreases the separability of two-faces, as it is certainly always possible to find another pair of GBR and lighting conditions that would have produced the same image.

The Quotient Image method [4-67] is another class-based, image-based re-rendering method. From a bootstrap set of images taken under three independent illuminants, the authors create synthetic images under different lighting conditions. Additionally, this method relies on an illumination invariant "signature" image per new face from a single input image alone. However, a strong assumption is that all faces share the same shape but differ in albedo only.

Again, issue 3 listed on page 90 is not solved by this method, since the authors claim "we will make the assumption that varying illumination does not affect the saturation and hue composition, only the gray-value distribution (shades of color) of the image" [4-67]. In addition, these methods always assume that the face detection was successful and precise (see also Section 3.5) - in order to apply this eigenimage approach - whereas this assumption will almost certainly reveal being false due to the lighting conditions that are especially bad. Consequently, the client will be definitely rejected if the detection fails. This last argument represents a strong motivation for using photometric invariant features, instead of statistically-based normalization techniques.

b) Derivation of insensitive features

Finding functions of an image that are *insensitive* to illumination changes indeed looks like a Quest for the Holy Grail. Chen and Belhumeur [4-15] showed however that under the Lambertian hypothesis no discriminative function can be used to distinguish any two images that are truly *invariant* to illumination changes. Indeed, it is always possible to find a family of surface and light sources that could have produced them. The Quotient Image mentioned in a) can be classified

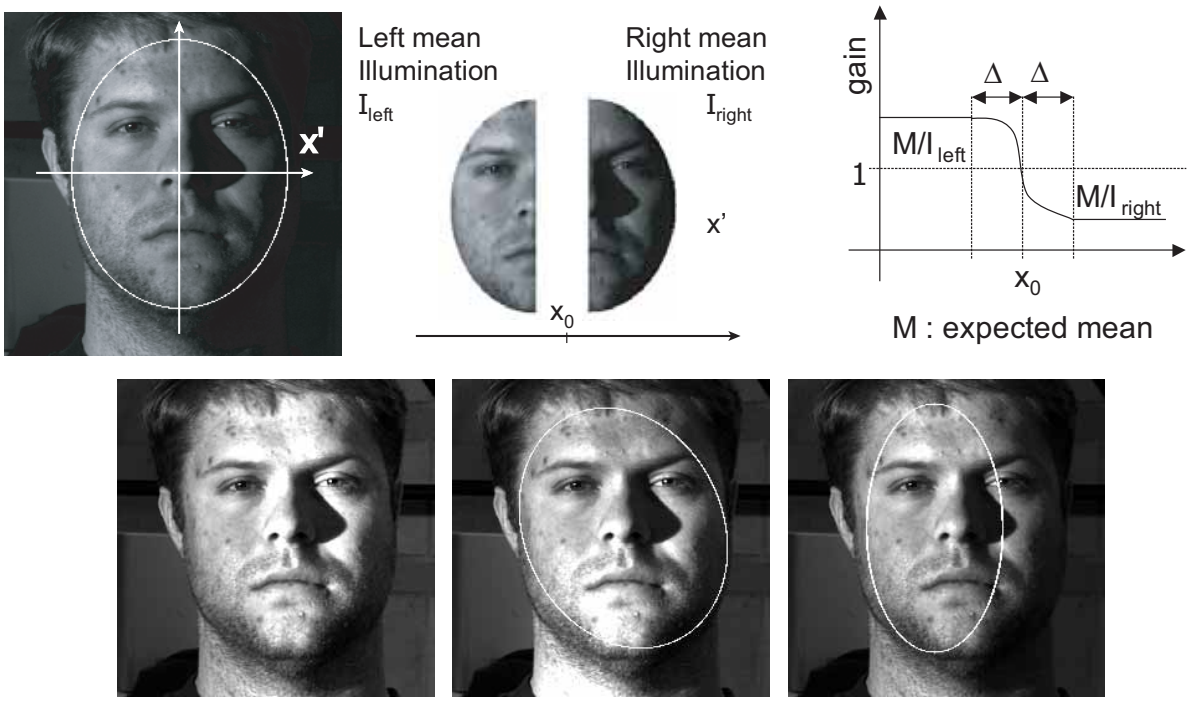


Figure 4-6: Illumination normalization with a sigmoid.
Top: from a detected ellipse the mean left and right illuminants are calculated. A gain is applied over the entire image along axis x' with values following a sigmoid. Bottom: three examples of normalization: correct detection (left), incorrect angle (middle), and incorrect position and width (right).

as well in this category, since it can be used as an "invariant signature" as stated by the authors.

No particular model of the face is used with feature derivation and as compared to predictive methods. Nevertheless, the face location is used in some methods. For example, if the face is correctly located (including rotation and scale) and if that the face is considered as having mainly two flat regions (the cheeks), then it is possible to measure the average illumination on both "sides" of the face and to remove the illumination dependence.

Kotropoulos et al. used this approach [4-39] while compensating the illumination with a sigmoid function. This normalization works quite effectively only if the face is precisely located, if there are no attached shadows, and if the light direction is not too much lateral. From the previous discussions it should be clear that the first assumption, namely the face can be precisely located, does not hold, especially when the lighting comes from one side! This is illustrated in Figure 4-6 where the sigmoid gain is applied to an image of the Yale B database with both correct and incorrect detection positions[3]. From this example it can be seen that a misplaced axis can lead to worse results than without nor-

malization. Additionally, large attached shadows are often present in the image and the simplified “two surfaces” model is no longer valid.

Having a model-independent method (i.e. without using a-priori knowledge of the face location) would be definitely more interesting. A simple method is the histogram transformation e.g. equalization or stretching, sometimes with a logarithm shape. However this is only poorly effective even when considering a region-based histogram equalization [4-65].

Another category of methods that includes homomorphic and isotropic filtering tries to recover the face reflectance by eliminating the illuminant contribution from Eq. 4.4, which leads to Eq. 4.14.

$$P(x,y)\frac{1}{I(\lambda;x,y)} = m_b(x,y;\theta,\rho)c_b(\lambda) \qquad (4.14)$$

Homomorphic filtering uses the assumption that the illuminant I varies slowly while the reflectance might change more sharply. Consequently, the reflectance can be extracted by high-pass filtering the logarithm of the image.

Land’s Retinex algorithm proposes to derive an invariant representation as the quotient of a point-wise division of the image with a low-pass filtered version of the image. However this methods creates halos where discontinuities are present in the image (e.g. on the border of the face).

Isotropic and anisotropic smoothing [4-30] estimate the illumination as a blurred version of the original image. The blurring operation contains a smoothing constraint, and the problem leads to a function minimization (which can be either isotropic or anisotropic).

However Short and Kittler [4-68] showed that none of these methods worked well in every situation. Adini [4-1] compared several different image representations such as edge maps, image intensity derivatives, or Gabor filtered images (see also Subsection 4.3.1). However no algorithm appeared as really effective in compensating for variations of illuminant.

We draw the following conclusions from the above observations:

3. These detection errors are extremely plausible under this illumination conditions !

- Reconstructionist approaches (i.e. variability prediction) seem to perform better than using illumination insensitive features;
- The complexity of reconstructionist approaches is much higher;
- Performance is dependent on the quality of training set;
- Cast shadows can only be predicted using 3D models.

Consequently, for mobile low-power applications we will consider only insensitive feature extraction techniques, although it shall be clear that they will be limited in performance. No specific preliminary image normalization will be performed as it was described in Figure 4-1, but the extracted features will be normalized (i.e. post-processing instead of pre-processing).

The next section compares three different feature extraction categories which are often used in pattern recognition. After presenting parameter choices, we compare their performance in terms of computational complexity and their robustness as photometric invariants.

4.3 Feature extraction

The original measurement (or *pattern*) at our disposal for the face authentication task is a collection of quantized discrete image values (picture elements, or pixels), representing either a single luminance value (i.e. grayscale) or multiple chrominance values (generally three, e.g. red, green and blue, by similarity with the human vision system).

A feature extraction step transforms the original feature into a new representation by means of a linear or non-linear conversion. The mapping of original measurements into more effective ones is also sometimes called feature *selection*. In this thesis however, this term is reserved for statistical transformations like Principal Component Analysis (PCA) and will be discussed later in Section 4.4. The feature extraction techniques mentioned here on the other hand are untuned techniques (i.e. which are not depending on a training stage) that are believed to generalize more easily to novel views and clients.

The goal of the feature extraction in this work is to:

- reduce the size of the original space;
- reduce the correlation between original features;
- derive new features that are invariant to illuminant variations.

The first objective may not be achieved directly, as the feature extraction techniques that will be described below temporarily lead to larger representations than the original image. For example, calculating multiple eroded version of an image (see Subsection 4.3.3) enlarges the feature set. However this novel feature space may allow a coarser sampling of the information used in the classification process (e.g. by taking only the features associated to the position of graph nodes instead of all the pixels belonging to the face surface), or it can provide a more effective use of feature selection techniques like PCA, allowing later on a more effective feature reduction. Similarly, the correlation reduction is usually also carried out by feature selection techniques like PCA.

Numerous techniques were employed for feature extraction in pattern recognition problems, and correspondingly, many can be possibly used for facial recognition. Without making distinctions between first-order and second-order features (as defined in Subsection 3.2.1, page 38) we could cite the following:

- statistics of a region, spatial distributions (e.g. histogram);
- transform features (e.g. Fourier transform, cosine transform);
- edges, contours, boundaries;
- moments;
- texture;
- shape.

This list is far from being exhaustive. In addition, features belonging to one category can be extracted by using a wealth of different techniques. For instance, surveys on texture[4] classification techniques can be found for instance in Randen and Husoy [4-60], and Tuceryan and Jain [4-74]. Techniques belonging to this category can be for instance based on Gabor filter banks as will be discussed below in Subsection 4.3.1.

The last feature category, i.e. shape, is used here to describe local information characterizing a macroscopic region of the face rather than the global shape of the face. Consequently, shape can be considered as an extension of texture feature extraction (although there is clearly no repetitive pattern in the face) and shape information can be extracted

4. It should be however kept in mind that the definition of texture is not the same for all authors.

with methods similar to those employed in texture analysis. Shape information is efficiently extracted with mathematical morphology operators, as it will be described in Subsection 4.3.3, but also with other approaches such as Hough transform or moments.

The other categories mentioned in the above list, or other methods not even mentioned here, although probably deserving some interest as well, were not investigated in this thesis work. In the rest of this section, three feature extraction techniques will be analysed and compared when used in conjunction with EGM, namely:

- Gabor transform-based features.
- Discrete cosine transform-based features;
- Mathematical morphology based features.

Note that using them with other paradigms may lead to different performance results.

4.3.1 Gabor transform-based features

Different categories of feature extraction for texture analysis were proposed. Let us mention signal processing feature extraction [4-60] from Randen and Husoy. This analysis consists in first filtering the image and then applying a local energy estimation function (e.g. squaring of the filtered image with a logarithmic normalizing nonlinearity [4-60]). The idea is that the energy distribution in the frequency domain can be used to classify textures, so that sub-band filtering an image and calculating the energy remaining in a single band can be used as a signature of the texture. This key idea is at the base of Gabor transform-based features.

Originally, the Gabor transform (i.e. the decomposition of a signal with a set of Gabor elementary functions) was proposed for 1D signals. The generalized 1D Gabor *inverse transform* can be expressed as [4-78]:

$$f(t) = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} a_{mr} \cdot g_0(t - mT) \cdot \exp\left(i2\pi \frac{nt}{T'}\right) \qquad (4.15)$$

where:

$$i = \sqrt{-1} \qquad (4.16)$$

and where g_0 is the Gabor elementary *function* (generally a gaussian), the time parameters T and T' describe, respectively, the shift of elementary functions in the time domain (T) and in the frequency domain (T'), the latter being obtained by a complex phase term in the time domain. Note that the original contribution of D. Gabor dealt with the special case $T=T'$.

The *inverse transform* of Eq. 4.15 signifies that any original signal $f(t)$, $t \in R$, can be reconstructed as a weighted sum of shifted versions of an elementary window function g_0, which modulates plane waves of different frequencies. The Gabor *transform*, which is sometimes called the direct or forward transform, then consists in finding the weights a_{mr}, associated to the time parameters T and T', representing the signal $f(t)$.

This transform is also a special case of the *discrete wavelet transform*, the wavelet function being the Gabor elementary function in this case, and the coefficients a_{mr} being the wavelet coefficients [4-14].

Furthermore, a transform is said to provide a *complete representation* if we can reconstruct f in a numerically stable way from the transform of f, or alternatively, if any function f can be written as a superposition of the transform's elementary functions.

When elementary functions form an orthonormal base, the function f can be reconstructed by a linear superposition of the bases weighed by the wavelet coefficients:

$$f(t) = \sum_{m,n} \langle f, \psi_{m,n} \rangle \psi_{m,r} \qquad (4.17)$$

where the operator <,> denotes the inner product, and the set of functions $\psi_{m,n}$ forms the basis.

Nevertheless, since 1D (and also 2D) Gabor functions (arranged in the form of Gabor wavelets) do *not* form orthogonal bases, a dual frame $\tilde{\psi}$ should be found:

$$f(t) = \sum_{m,r} \langle f, \psi_{m,r} \rangle \tilde{\psi}_{m,r} \qquad (4.18)$$

A comprehensive description of Gabor analysis and its applications can be found in Feichtinger and Strohmer [4-20][4-21]. In the following discussion however, the basis of Gabor functions used for analysis does

not provide a complete representation, but are rather used in the sense of spectral analysis. For this reason, the Gabor elementary functions will be called *Gabor filters* from now on. Several Gabor filters form a *Gabor filter bank*, which is used for spectral analysis of a 2D image.

Gabor filters corresponding to gaussian windows are interesting for time-frequency signal analysis because they present the best compromise between spatial (or time) and frequency extension.

The 1D Gabor filter can be extended to the 2D case [4-46][4-13]: :

$$g(x,y) = \frac{1}{2\pi\sigma\beta}\exp\left(-\pi\left[\frac{(x-x_0)^2}{\sigma^2}+\frac{(y-y_0)^2}{\beta^2}\right]\right)\exp(i\pi(u_0x+v_0y)) \qquad (4.19)$$

where (x_0, y_0) is the center of the receptive field in the image space, σ and β are the standard deviations of the Gaussian along x and y, and (u_0, v_0) describe the maximum frequency response of the function.

Basically a 2D Gabor filter represented in the original space domain corresponds to a sinusoidal plane wave of given frequency and orientation multiplied by an elliptic gaussian envelope. Hence, the frequency representation of a 2D Gabor filter corresponds to the convolution of an elliptic 2D gaussian, different from the one in the original space domain, with a dirac impulse centred at the plane wave spatial frequency.

A special case of modulating Gaussian is the one that has the major axis having the same orientation as the plane wave. Eq. 4.19 becomes:

$$g(x,y) = 2\pi\sigma_x\sigma_y\exp(-2\pi^2[\sigma_x^2\tilde{x}^2+\sigma_y^2\tilde{y}^2])\exp(i2\pi\omega_0\tilde{x}) \qquad (4.20)$$

$$\begin{bmatrix}\tilde{x}\\ \tilde{y}\end{bmatrix} = \begin{bmatrix}\cos\phi & \sin\phi\\ -\sin\phi & \cos\phi\end{bmatrix}\begin{bmatrix}x\\ y\end{bmatrix} \qquad (4.21)$$

Note that within Eq. 4.20, σ_x and σ_y correspond to the standard deviation of the Gaussian in the *frequency* domain, and that they account for the bandpass characteristics of the filter, unlike the previous definition in Eq. 4.19. The constant in front of the expression was adapted accordingly. The parameter ω_0 corresponds to the center passband frequency of the bandpass along the frequency axis[5]. Note, finally, that this form is not separable in x and y, unless $\sigma_x = \sigma_y = \sigma$.

5. Since the Fourier transform of a rotated 2D function results in the rotation of the Fourier transform of this 2D function.

Equivalently in the frequency space:

$$G(\omega_x, \omega_y) = \exp\left(-\frac{1}{2}\left[\frac{(\tilde{\omega}_x - \omega_0)^2}{\sigma_x^2} + \frac{\tilde{\omega}_y^2}{\sigma_y^2}\right]\right) \qquad (4.22)$$

$$\begin{bmatrix}\tilde{\omega}_x \\ \tilde{\omega}_y\end{bmatrix} = \begin{bmatrix}\cos\phi & \sin\phi \\ -\sin\phi & \cos\phi\end{bmatrix}\begin{bmatrix}\omega_x \\ \omega_y\end{bmatrix} \qquad (4.23)$$

Gabor filter banks seem well suited for the task of image filtering in vision applications. Indeed Gabor filter banks share strong similarities with visual cortical neurons [4-58][4-37]. But although many authors indeed agree that Gabor filter banks are performing admirably for image analysis, and particularly for face recognition, they don't all agree on the parameters – such as the number and characteristics of the different passband frequencies $\tilde{\omega}_x$, the number and characteristics of the different orientations ϕ, the width σ_x and σ_y of the passband characteristics, etc. – and how to handle complex images resulting from the filtering operation, i.e. how to extract *Gabor features*.

Jain and Farrokhnia [4-34] use a real (i.e. even) version of the basic Gabor filter. Using our notation, Eq. 4.20 gets reformulated as follows:

$$g(x, y) = 2\pi\sigma_x\sigma_y \exp(-2\pi^2[\sigma_x^2\tilde{x}^2 + \sigma_y^2\tilde{y}^2])\cos(2\pi\omega_0\tilde{x}) \qquad (4.24)$$

This version is very similar to the previous one in the frequency domain, but instead of having one gaussian centred at (ω_x,ω_y) it will have one additional gaussian at (-ω_x, -ω_y), both with half their peak value.

The complex form in Eq. 4.20 is indeed the unilateral form of the real version in Eq. 4.24. This is due to the fact that the imaginary part of the filter considered in the original space domain, is the Hilbert transform of the real part, and therefore the complex form of the filter is an analytical signal. Similarly, a real image filtered with an analytical signal is analytical, and consequently the magnitude of the signal filtered with an analytical filter gives the instantaneous envelope of the signal filtered with the real function, whereas the magnitude of the real (sine or cosine) filter alone will produce a rectified signal that is only a poor approximation of the envelope (see Figure 4-9). For these reasons, results are often reported with the magnitude of the complex response, or, otherwise, the magnitude of the real response is subsequently averaged over a small neighborhood in order to further smooth the approx-

imated envelope. Note that the smoothing the rectified signal tends to reduce the selectivity of the filters.

As mentioned previously, sever Gabor filters with different passband characteristics are combined to form a Gabor filter bank. The repartition of this group of filters in the frequency domain is called the lattice. Examples of lattices used in the case of Gabor filter banks are the rectangular checker-board in the log-polar frequency coordinate system with an octave-band structure [4-10] or the rosette-like representation in cartesian coordinates. Jain et al. [4-34][4-74] suggest the use of 5 radial frequencies and 4 orientations. Bigün et al. [4-10] use 5 frequency bands and 6 orientations. This choice is clearly application-dependent and needs to be finely tuned. Bovik et al. [4-13] describe a method for optimally selecting the parameters of the filter bank.

After Gabor filtering (with real filters), Tuceryan and Jain [4-74] suggest the application of a sigmoidal nonlinearity and finally the local computation of the energy using an absolute average deviation of the transformed values from the mean, calculated within a window. Bovik et al. [4-13] propose to calculate the magnitude of the complex response of the filters and then to apply a smoothing filter in the form of a Gaussian filter related to the shape of the corresponding subband filter[6]. They further describe the use of the phase information to enhance the classification. Bigün et al. [4-10] propose the use as Gabor features, the real or complex moments of the complex response. These Gabor features enable to code the power spectrum with less parameters than with the squared modulus of the complex filtered images. They note that complex moments are best for textures with N-folded symmetries, which is obviously not the case for facial patterns that are more stochastic.

Gabor filtering being seemingly successful with texture analysis, it was of course also applied to face and fingerprint recognition. For the latter Jain et al. [4-35] used the same feature extraction technique as described in their texture analysis technique [4-34]. For face recognition, Lades et al. [4-41] propose the use of complex Gabor filters together with a graph matching approach. In their work, they use a modified version of the Gabor filters (Eq. 4.25) which is said to be “DC-free”.

6. Yet this seems unnecessary, since the magnitude of the complex response corresponds already to the envelope.

$$g(\vec{x}) = \frac{(k_x{}^2 + k_y{}^2)}{\sigma^2}\exp\left(-\frac{1}{2}\left[\frac{k_x^2x^2 + k_y^2y^2}{\sigma^2}\right]\right)\left[\exp(i[k_x^2x^2 + k_y^2y^2]) - \exp\left(-\frac{\sigma^2}{2}\right)\right] \quad (4.25)$$

Note that the definition of k and σ proposed by Lades et al. differs from the convention used in Eq. 4.20, but that the same structure remains, except for the final real exponential that is subtracted to the complex one. Finally, it can be observed that for a large σ (that is for filters with large spatial extent), the DC compensation term tends to vanish.

For face recognition applications, Lades suggests to use five frequencies and eight orientations, and to use the magnitude of the complex filter response as the local energy measure. The Gabor feature vector (called *jet*) composed of several filter responses at one spatial coordinate location, is compared to a reference feature vector using the normalized dot product. Wiskott et al. propose to enhance the graph matching of Gabor features by considering phase information [4-76].

Similarly Duc et al. [4-17] use a combination of 18 filters (6 orientations and 3 resolutions, similarly to what is depicted in Figure 4-7), and again calculate the magnitude of the complex Gabor responses. Duc et al. further compare the feature vectors by using the square of a weighted Euclidean distance between the test feature vector and the reference feature vector. This weighting factor is determined statistically using the Fisher Linear Discriminant (FLD).

Finally, other authors used Gabor features in conjunction with face recognition techniques different from Elastic Graph Matching. For example correlation of Gabor filtered images [4-2], or PCA/LDA of Gabor filtered images [4-16][4-47].

From the above, it shall be clear that no author systematically reported results about these propositions in the frame of graph matching.

In the rest of this subsection, the parameters σ_x, σ_y, and ϕ (defined in Eq. 4.20 and Eq. 4.22) corresponding to the configuration of the p-th filter orientation and q-th octave band, will be chosen, following the proposition of Duc et al. According to Eq. 4.26, the resulting in a filter structure is now given by Eq. 4.27 and Eq. 4.28.

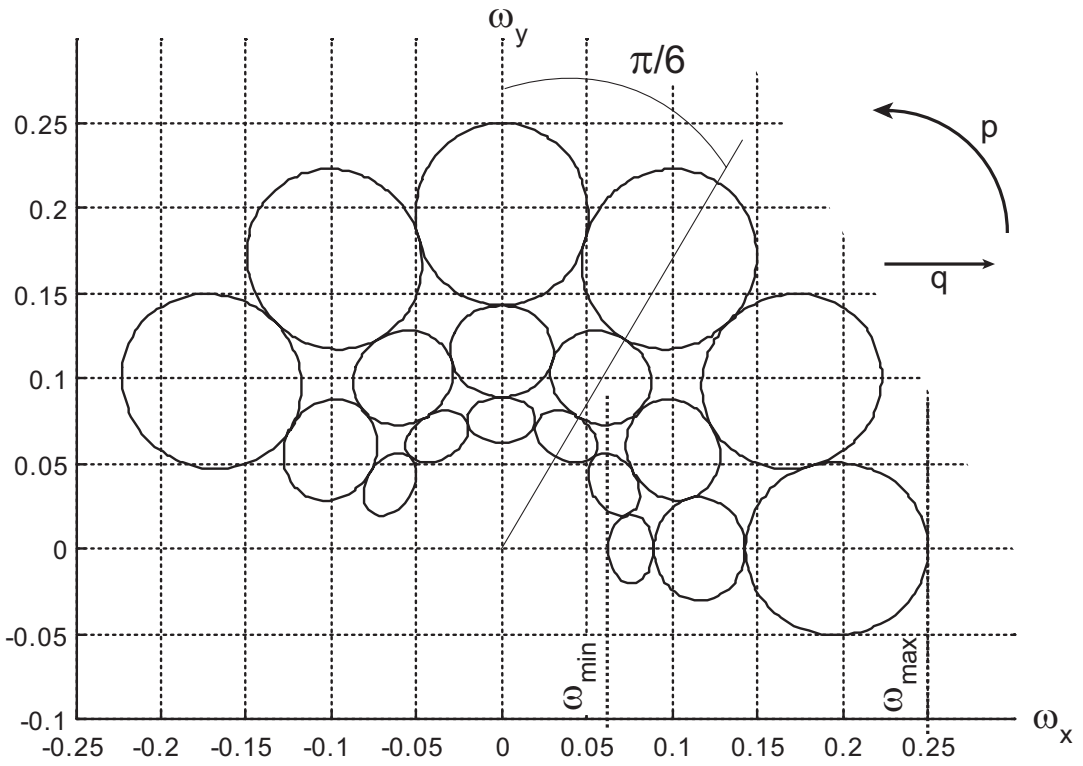


Figure 4-7: Example of Gabor filter bank represented in the 2D frequency domain. The filter bank counts 18 elementary filters distributed along 3 increasing frequency bands, and along 6 orientations. Each elementary filter is indicated by a contour on which the magnitude takes the value 1/sqrt(e). Further parameters are described in the text.

$$\begin{aligned} \sigma_{x,q} &= (\omega_{max} - \omega_{min})/(2 \cdot 2^{q-1}) \\ \sigma_{y,q} &= \sqrt{(\sigma_{x,q})^2 - (\omega_{0,q})^2} \cdot \tan\left(\frac{\pi}{2n}\right) \\ \phi_p &= p\frac{\pi}{n} \\ \omega_{0,q} &= \omega_{min} + \sigma_{x,q}\lfloor(1 + 3(2^{q-1} - 1))\rfloor \end{aligned} \qquad (4.26)$$

$$g_{p,q}(x,y) = 2\pi\sigma_{x,q}\sigma_{y,q}\exp(-2\pi^2[\sigma_{x,q}^2\tilde{x}^2 + \sigma_{y,q}^2\tilde{y}^2])\exp(i2\pi(\omega_{0,q}\tilde{x})) \qquad (4.27)$$

$$\begin{pmatrix} \tilde{x} \\ \tilde{y} \end{pmatrix} = \begin{pmatrix} \cos\phi_p & \sin\phi_p \\ -\sin\phi_p & \cos\phi_p \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix} \qquad (4.28)$$

with p={1..m} and q={1..n), ω_{max} and ω_{min} corresponding to the lowest and highest cut-off frequencies[7] of the filters along the definition depicted in Figure 4-7, whereas the operator $\lfloor\ \rfloor$ indicates the floor operation. The following numerical values where also experimentally established by Duc et al.: m=3, n=6, ω_{min}=1/16, ω_{max}=1/4.

7. Normalized frequencies, where the value *1* represents the Shannon-Nyquist frequency.

The space domain response of each filter $g_{p,q}(x,y)$ is then applied by convolution (denoted by * in Eq. 4.29) on the original (reference or test) image $f(x,y)$, or equivalently by the point-wise multiplication (denoted by $\otimes$ in Eq. 4.30) of the 2D Fourier transform F of the image f and of the Fourier transform G of the filter g:

$$h_{p,q}(x,y) = f(x,y) * g_{p,q}(x,y) \qquad (4.29)$$

$$H_{p,q}(\omega_x,\omega_y) = F(\omega_x,\omega_y) \otimes G_{p,q}(\omega_x,\omega_y) \qquad (4.30)$$

The latter solution was used in the algorithm implementation. Namely, the Fast Fourier transform F of the real image f was calculated using the 2D FFT function provided in the Intel Image Processing Library (IPL). In this work, the filters $G_{p,q}$ were designed in the frequency domain, using Eq. 4.22 and Eq. 4.26. Then, pixel-wise multiplication is performed with each filter frequency response, and the inverse FFT (IFFT) is again calculated with Intel IPL. This library implements a floating point version of the FFT/IFFT, with a precision of 24 bits. Note that a more in-depth study of the Gabor filter bank implementation should for instance consider the effects of truncating the infinite gaussian responses in the spatial and the frequency domains.

The Gabor features[8] J are then obtained for each pixel coordinates (x,y), from the magnitude of the complex response (see also Figure 4-8):

$$J_{p,q}(x,y) = \left\| h_{p,q}(x,y) \right\| \qquad (4.31)$$

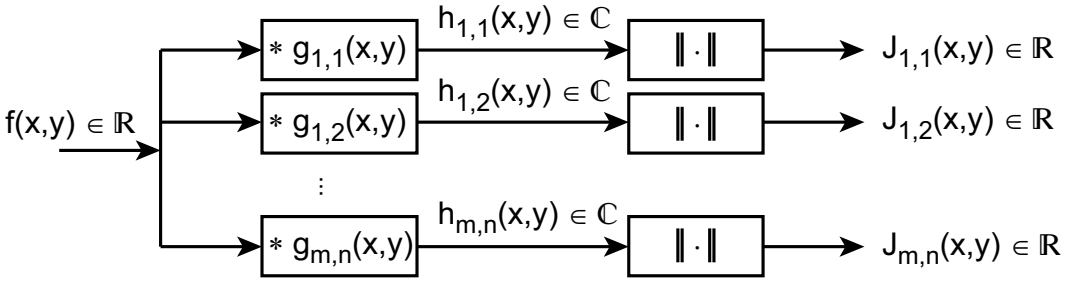


Figure 4-8: Gabor feature extraction.
The input image f is individually convolved with every filter in the bank, resulting in m×n complex responses h. The magnitude J of each filter response is then taken, so that the final Gabor features J are real. The filtered image h and the magnitude response J have the same size as the original image f

8. An upper case symbol is used historically for J, the magnitude of h, although it applies in the original image domain and not in the frequency domain as for all other upper case symbols.

Additionally, it can be observed that Gabor filters applied using Eq. 4.29 or Eq. 4.30 have a finite size, whereas, by definition, they span the entire image and frequency domains. Consequently, depending on the image size and on the parameters of the filters, the truncation will more or less affect their response. The effects of this truncation are not discussed here.

The graph distance is obtained from the magnitude of: the filtered reference image, $J_{ref,p,q}(x,y)$, and the filtered test image $J_{tst,p,q}(x,y)$; either with an Euclidean feature distance (Eq. 4.32) or with a sum of normed dot products of features (Eq. 4.33).

$$d_{eucl} = \sum_{i \in nodes} \sqrt{\sum_{p=1}^{m} \sum_{q=1}^{n} [J_{ref,p,q}(x_i, y_i) - J_{tst,p,q}(\hat{x}_i, \hat{y}_i)]^2} \qquad (4.32)$$

$$d_{dot} = \sum_{i \in nodes} \frac{\sum_{p=1}^{m} \sum_{q=1}^{n} J_{ref,p,q}(x_i, y_i) \cdot J_{tst,p,q}(\hat{x}_i, \hat{y}_i)}{\sqrt{\sum_{p=1}^{m} \sum_{q=1}^{n} J_{ref,p,q}^2(x_i, y_i)} \sqrt{\sum_{p=1}^{m} \sum_{q=1}^{n} J_{tst,p,q}^2(\hat{x}_i, \hat{y}_i)}} \qquad (4.33)$$

where (x_i, y_i) are the pixel coordinates associated to the *i*-th node in the reference image, which are different from $(\hat{x}_i, \hat{y}_i)$, the pixel coordinates associated to the *i*-th node in the test image.

To show the effect of using these Gabor filters, the eighteen complex filters and the magnitude of the filtered images are depicted in Figure 4-9 for a single input image: a) represents the original image; b) represents the axes (x,y) over the filter response in the spatial domain; c) represents the axes (ω_x,ω_y) over the filter response in the frequency domain; d) represents the real part of the filter responses in the spatial domain, for the set of 18 filters (p=1..6, q=1..3) describe in Eq. 4.26–Eq. 4.28; e) represents the result of the filtering of a) with the set of filters described in d); f) represents the magnitude of complex results obtained from the filtering of a) with the complex versions of the filters that correspond to d).

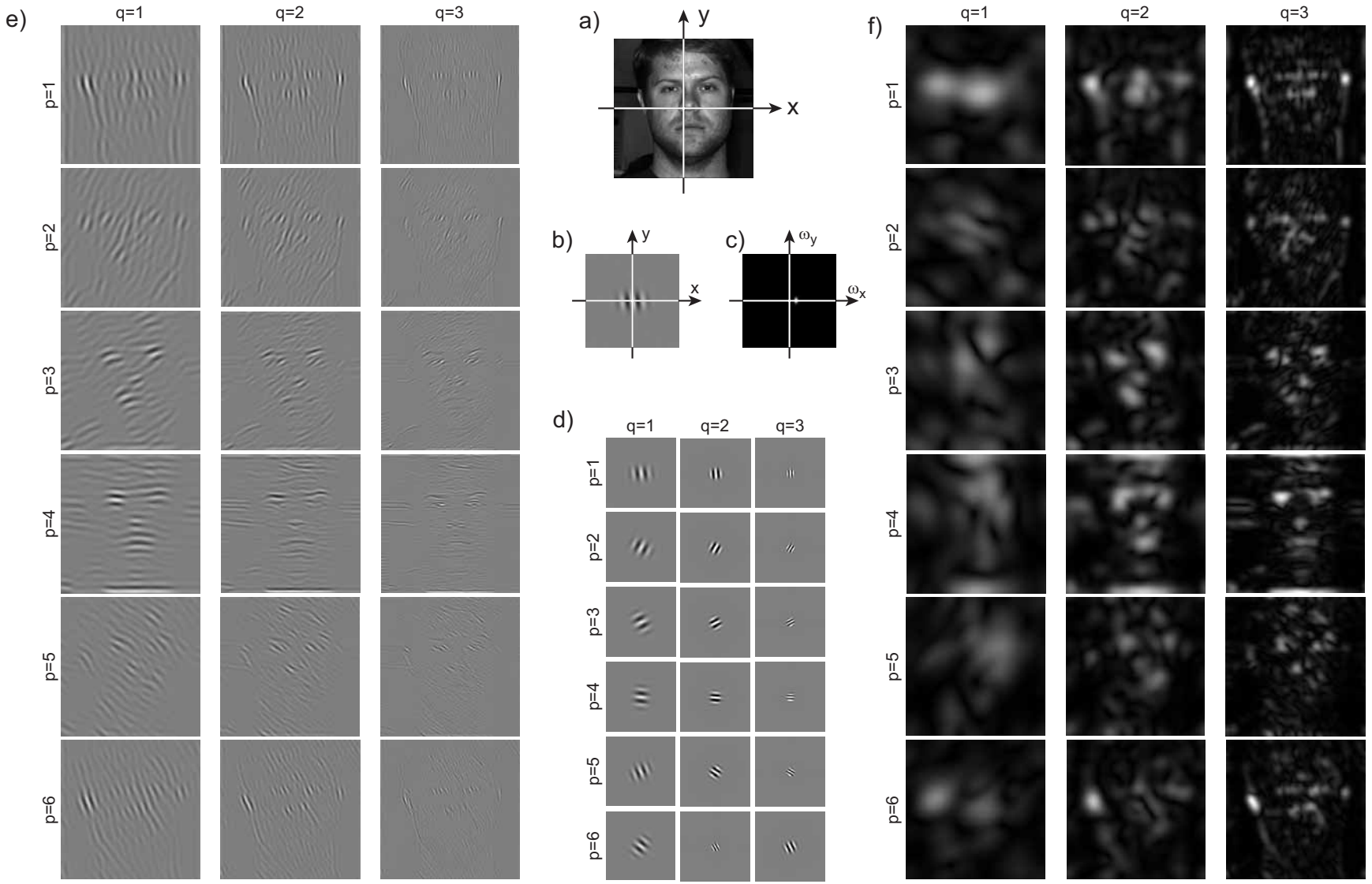


Figure 4-9: Example of an image processed with real and complex Gabor filter banks.
The description of this figure in given within the text, on page 110.

As the elementary Gabor filters are band-pass filters, they will remove[9] the DC term that is actually gathering in the frequency domain the contribution of the ambient illuminant, i.e. the $L(\lambda)$ term in Eq. 4.4 and Eq. 4.6. However, they will not compensate for typical Lambertian variations of the illumination, i.e. variations that are due to a varying angle θ between the face surface and the light direction, which will cause variations of the image spectrum within the passband region of the Gabor filters.

Moreover, it can be showed that only the normed dot product will be robust to variations of θ in Eq. 4.7, assuming that no attached shadows appear ($-\pi < \theta < \pi$). A corollary is that a normalization of the Euclidean distance should be used as well instead of the above metrics proposed by Duc [4-17]. Still, it is clear that Gabor features cannot solve the shadow and illuminant color variation problems.

Recalling Adini's observation that "all image representations under study [including Gabor features] were insufficient by themselves to overcome variations because of changes in illumination direction" [4-1], we are interested to evaluate the performance of these features on several databases. These results will then serve as a reference point to compare other less complex implementations.

First, the Yale B database [4-27] was used in identification mode as defined on page 26 in Subsection 2.7.2. Figure 4-10 illustrates the results obtained for two face images processed by the 2D complex Gabor filter bank comprising $m \times n = 18$ elementary filters, with $m=3$, $n=6$. The upper original image was obtained with a frontal illuminant, whereas the lower original image was achieved with an illuminant oriented at 60° with respect to the normal. The magnitude of the images resulting from the application of the complex Gabor filters with $q=1$, and $p=1,2,3$ are depicted on the right hand-side of the original images in Figure 4-10. It can be clearly observed that, although the spatial distributions of peak responses remains globally the same, their amplitudes are notoriously different between the left and right parts of the face in the top and bottom rows due to Lambertian effect. It can be expected that normalized metrics such as the normed dot product of Lades will partly compensate for this - at the expense of higher computational complexity - but that the Euclidean distance for instance will be less efficient.

9. Notice that the DC component removal is performed up to a certain level only, since Gaussian functions expand to infinity.

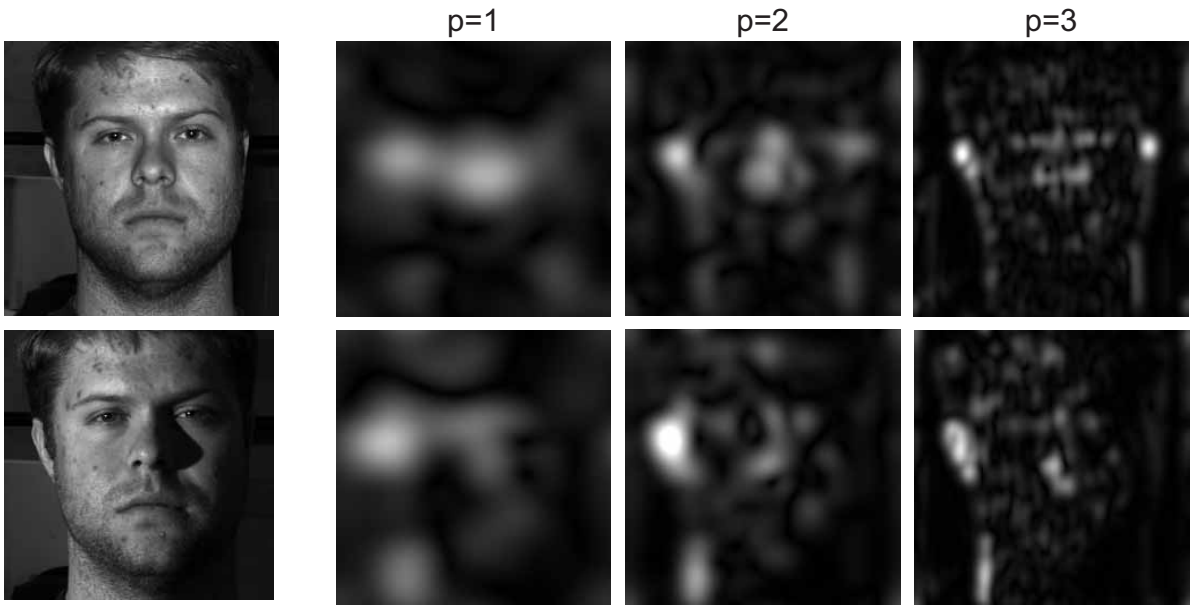


Figure 4-10: Effect of illumination variations on Gabor features.
The upper original image corresponds to a frontal illuminant, while the lower original image corresponds to an illuminant oriented at 60° with respect to the normal. The right hand-side images correspond to the magnitude of the images processed with three different filters (q=1, p=1,2,3)

Method	Error rate (%) vs. illuminant change		
	Subset 2	Subset 3	Subset 4
Gabor EGM Eucl. distance	0.8	6.4	63.6
Gabor EGM Dot product dist.	0.0	0.0	10.7

Table 4-1: Identification results on the Yale B database.

Identification was performed on this database (see results in Table 4-1), using Subset 1 as the reference set, and matching Subsets 2, 3 and 4 against it. Lighting conditions become worse, i.e. always further from a frontal illumination, with increasing subset index. It can be observed that, as expected, the normed dot product distance performs much better than the Euclidean distance. Furthermore, the error rate increases in Subset 4 because non-linear effects (i.e. shadows) start to become significant. An error rate of 10% on this subset can be considered as a good result as compared to published results (see for example [4-27]), which confirms the strength of this feature extraction technique.

It is insufficient to perform experiments only using the XM2VTS database in order to demonstrate the superiority of one distance metric over the other with respect to the illuminant direction, because this database contains only frontally illuminated images. However overall re-

sults on this database will be used for further comparisons, when a large number of clients and test impostors is required.

Figure 4-11 shows the ROC using Lausanne protocol configuration 2 with single and multiple templates (see Subsection 2.7.2 for details on this protocol). It can be observed that the EER is slightly higher than 6% when using a single template (i.e. images suffixed _1_1 as reference). For a 2% of false acceptance the corresponding false rejection rate is around 10%. For the multiple templates experiment, the EER goes as low as 2.5% with the dot product metrics and 2.25% with the Euclidean metrics. For FAR=2% the FRR is now only 3.5% (respectively 2.25% with the Euclidean metrics). Although the dot product metrics performs slightly better than the Euclidean metrics for single templates, the opposite can be observed for the multiple templates case.

Nevertheless the difference become very small and a larger database should be used to make more precise observations in this domain (especially where the FRR is low). For instance, two tests are performed for each of the 200 clients of the XM2VTS database according to the Lausanne protocol, and consequently, it is not possible at all to measure error rates that are lower than 1/(200×2)=0.25%.

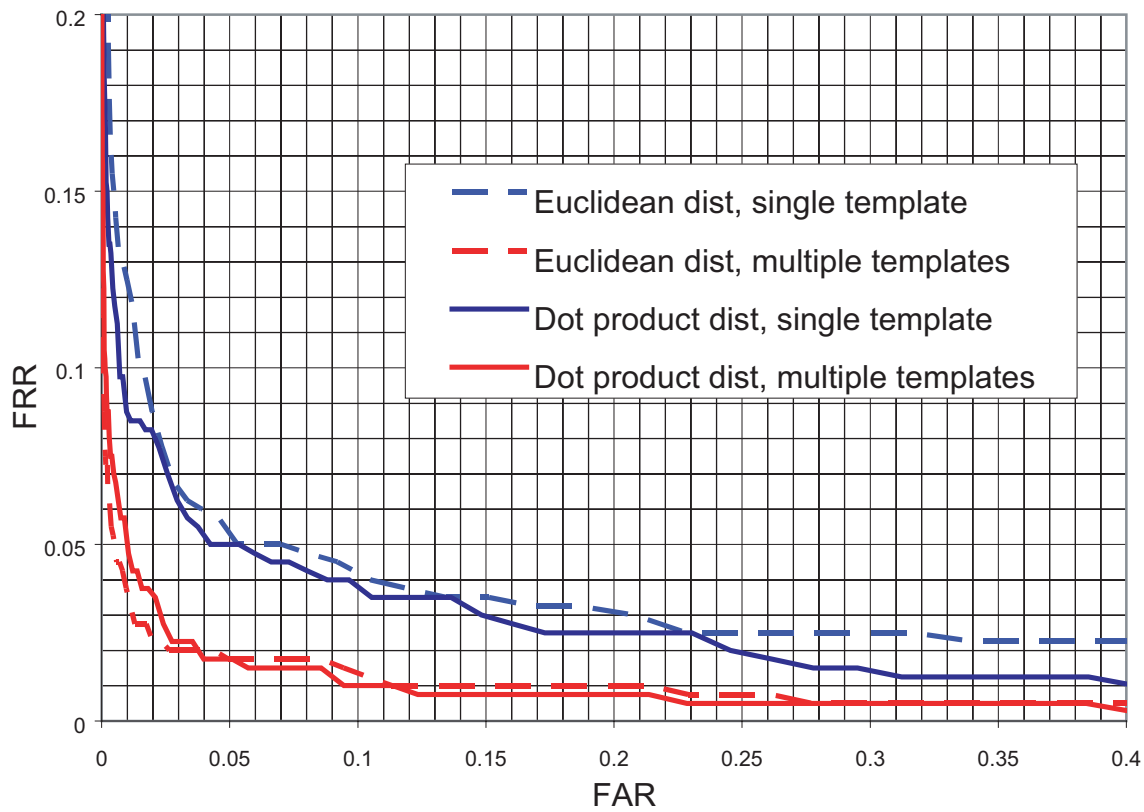


Figure 4-11: ROC of EGM with Gabor features on XM2VTS database.
Euclidean vs. dot product metrics and single vs. multiple templates.

The overall performance of EGM with Gabor features on this database is nevertheless much better than when using the Yale B[10] database. This is again an evidence that the reported results are strongly database dependent and should be considered with due caution.

An issue in graph matching already raised by Lades [4-41] and later by Duc [4-17] is the convergence of the rigid graph matching (see also Subsection 4.5.1). This convergence may be hindered when local high-frequency information is present, i.e. if features are “too local”. Indeed, the high frequency information is very useful to finely discriminate otherwise similar features, but it changes rapidly in the image thus creating local graph distance minima. Lades thus proposes to use only the lowest frequency bands in a first pass to determine coarsely the location of the matched graph, and then to refine the distance measure by using all frequency bands and orientations. Duc used multiresolution techniques, i.e. filtering images with a low-pass gaussian filter, and then decimating the result. This is indeed similar to the proposal of Bovik to use Gaussian filters [4-13]. With this respect, the “average absolute deviation from the mean” feature extraction technique used by Jain et al., probably also increases the convergence ([4-35], and see also Section 5.2).

To illustrate this phenomenon we plot the *objective function*, also sometimes called the *global field potential*, i.e. the distance returned by the graph matching operation (or matching error) vs. the two-dimensional (x,y) displacement around the matched location[11]. This is represented as a surface plot in Figure 4-12 for the two distance functions described in Eq. 4.32 and Eq. 4.33. Since the objective functions are expressed in function of the 2D displacement around the matched location, one is using the variables dx and dy instead of x and y. The smoothest result seems to be obtained with the Euclidean distance measure, but both objective functions are anyway sufficiently smooth for iterative methods to converge rapidly. Some local minima exist anyway and the graph matching algorithms (see Section 4.5) will have to handle them appropriately, e.g. by using different initial conditions.

Another issue with Gabor features comes from their orientation: because the filtered images are rotation dependent, a rotated face will appear differently after filtering. Considering a discrete set of rotations

10. However the scenarios were different: identification in a closed-set was considered with the Yale B database, and authentication with the XM2VTS database.
11. Only orthogonal graphs with correct orientation and scale are considered here.

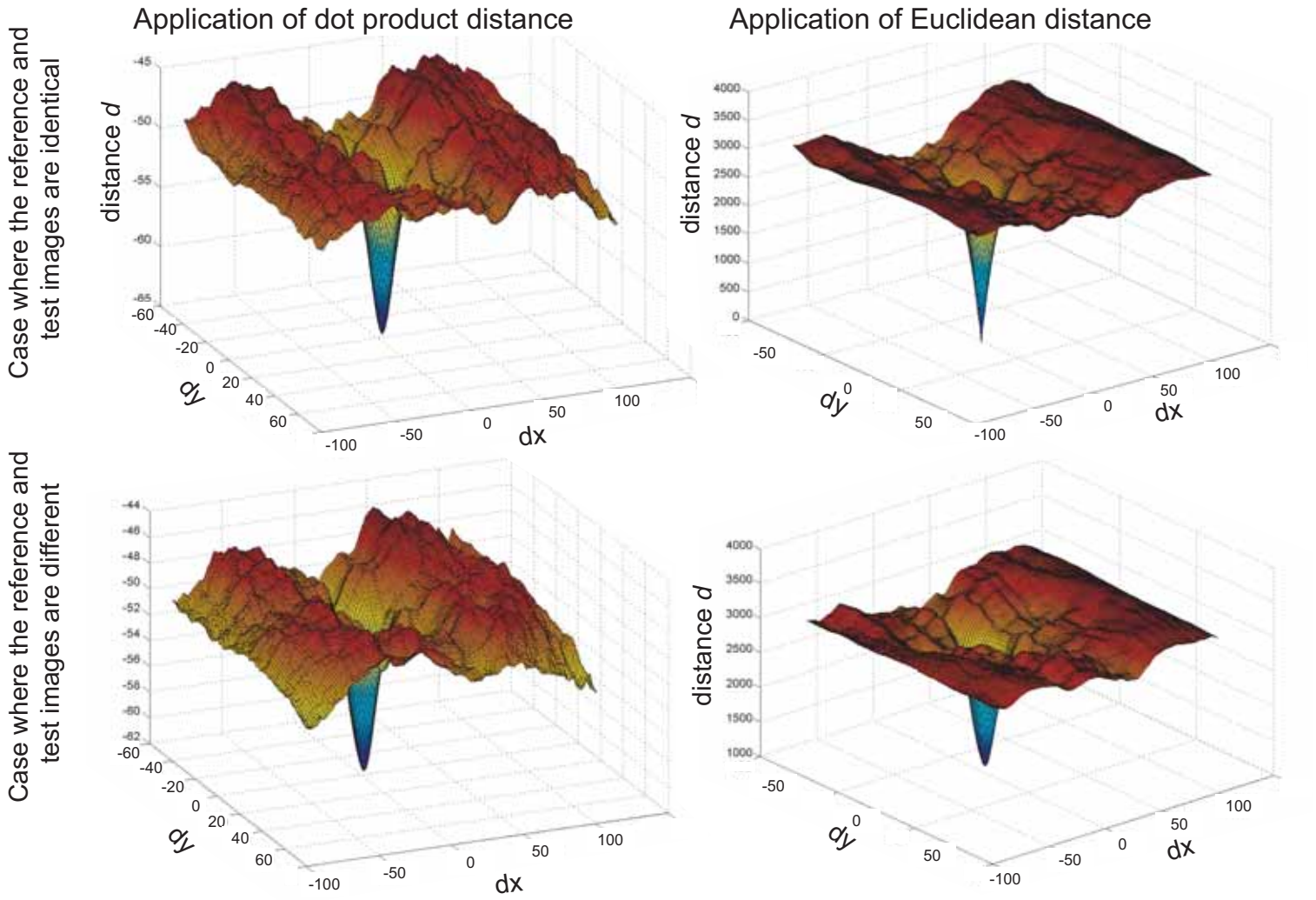


Figure 4-12: Objective functions of Gabor features using two distance metrics.
The objective functions are expressed in function of the 2D displacement (*dx*,*dy*) around the matched location.

corresponding to the n=6 orientations of the Gabor filter bank, cf. Figure 4-7, it is possible to elegantly replace the rotation of the original image and the costly computation of an interpolation, by merely rearranging the terms $J_{tst,p,q}$ appearing in Eq. 4.32 and Eq. 4.33 along q while keeping p constant, to emulate the same rotation. This solution was already proposed by Jain for fingerprint matching (using the so-called fingercodes; see Section 5.2). Its precision is limited by the number of m different orientations present in the filter bank: the more orientations, the more effective this rotation will be.

Considering that a particular filter with scale p is tuned to certain spatial frequencies, and considering that a scaling of the face image will modify these spatial frequencies, scaling is thus also an issue. Indeed, the ratio between the filter size in the spatial domain (as defined similarly to the frequency domain size depicted in Figure 4-7) and the face size in the spatial domain (e.g. the distance between the eye centres) shall be used, e.g. to rearrange the terms $J_{tst,p,q}$ along p while keeping q constant. This solution is limited by the even smaller number of different scales in the filter bank. For large scale variations, it will be thus necessary to resort to image interpolation and to filter this new image.

One can observe that using 18 complex Gabor filters requires a very high computational power. If only a small portion of the filtered image positions was necessary for matching, it would be advantageous to use the convolution (Eq. 4.29) form only at those locations. However, since the graph will be scanned over a large part of the image, it appeared beneficial to compute the Gabor responses over the entire domain (Eq. 4.30). Therefore, filtering is more efficiently done in the frequency domain due to the large kernels involved. This translates into using one direct Fast Fourier Transform (FFT) of the original image to apply the filtering according to Eq. 4.30, and in computing 18 spectral responses by performing a complex multiplication for every duplet (ω_x,ω_y), and 18 inverse FFT (IFFT).

This huge complexity inevitably disqualifies this feature extraction technique from those usable for mobile applications. It remains however interesting for applications where power-consumption is not a concern and where dedicated hardware for performing FFT can be used. In the rest of this thesis, we will use the Gabor feature extraction technique as a reference implementation, which performs pretty well from the signal processing viewpoint, for comparison with further methods which are more economic regarding power saving.

4.3.2 Discrete cosine transform-based features

Another signal processing based technique for texture analysis uses the 2D Discrete Cosine Transform (DCT). There are several motivations for using this analysis tool. Firstly, it is relatively popular in image compression (e.g. JPEG) and consequently fast implementations exist together with dedicated hardware accelerators. Secondly, this image transform is equivalent to a critically sampled filter bank and thus can also act as a bandpass filter, similarly to the Gabor filters. Finally matching of JPEG compressed images would be potentially possible without fully uncompressing them, i.e. comparing directly two JPEG coded images after Huffman decoding but without performing the inverse DCT [4-19]. The only limitation would thus be the 8x8 block decomposition. This is especially interesting for search engines operating on large databases but less interesting for mobile applications, where only a small number of matching operations is performed.

Application of DCT to texture analysis was proposed e.g. by Ng et al. [4-53] (including a comparison with the Gabor filter bank approach) and to face recognition e.g. by Pan and Bolouri [4-54]. Ng et al. used a 3x3 DCT and a moving local average on a window of 15x15 pixel as the local energy measure. They excluded the DC component for obvious reasons. Their conclusion for texture analysis experiments was that local variance DCT performed relatively well as compared to Gabor. However, in terms of computational complexity the local variance estimation was the most critical part to optimize.

One could think of using larger kernels for the DCT e.g. an 8x8 as used in JPEG compression. Again higher frequencies might cause some problems to the convergence of the method, and as in Gabor filtering it might be wise to use only the lowest frequencies in a first pass and then to refine the convergence using higher frequency information.

Similarly to Gabor features, the different DCT features mentioned above are sensitive to the direction of lighting (Lambertian model), even with the suppression of the DC component. This can be explained by the fact that although the features are no longer dependent on the mean value of the pixels, they are still depending on the contrast of the image. Consequently, it would be necessary to compensate for these variations by normalizing the DCT features similarly to the normed dot product used for Gabor features.

Sanderson and Bengio [4-63] (following Sanderson and Paliwal [4-62]) propose another alternative – called *DCT-mod2* – to increase the robustness to variations of lighting direction: they replace the first three DCT coefficients by their respective horizontal and vertical "deltas". They propose using 8x8 blocks with a 50% horizontal and vertical overlap, and to keep only 12 baseline DCT coefficients (according to Figure 4-13b) plus 6 "delta coefficients" (3 horizontal and 3 vertical deltas [4-63]), which can be interpreted as "transitional spatial information". Features obtained with this method are represented in Figure 4-13.

Results on the Yale B database (Table 4-2) are not as good as those obtained with Gabor features (dot product metrics), although DCT-mod2 features seem to achieve a certain robustness against variations of lighting (for an overall comparison see Subsection 4.7.2). Again, it is not possible to compensate for non-linear illumination effects with such features.

Results on the XM2VTS database are much less satisfying. The achieved EER is larger than 20% for the single template experiment, and around 10% with the multiple templates method. Although Bengio

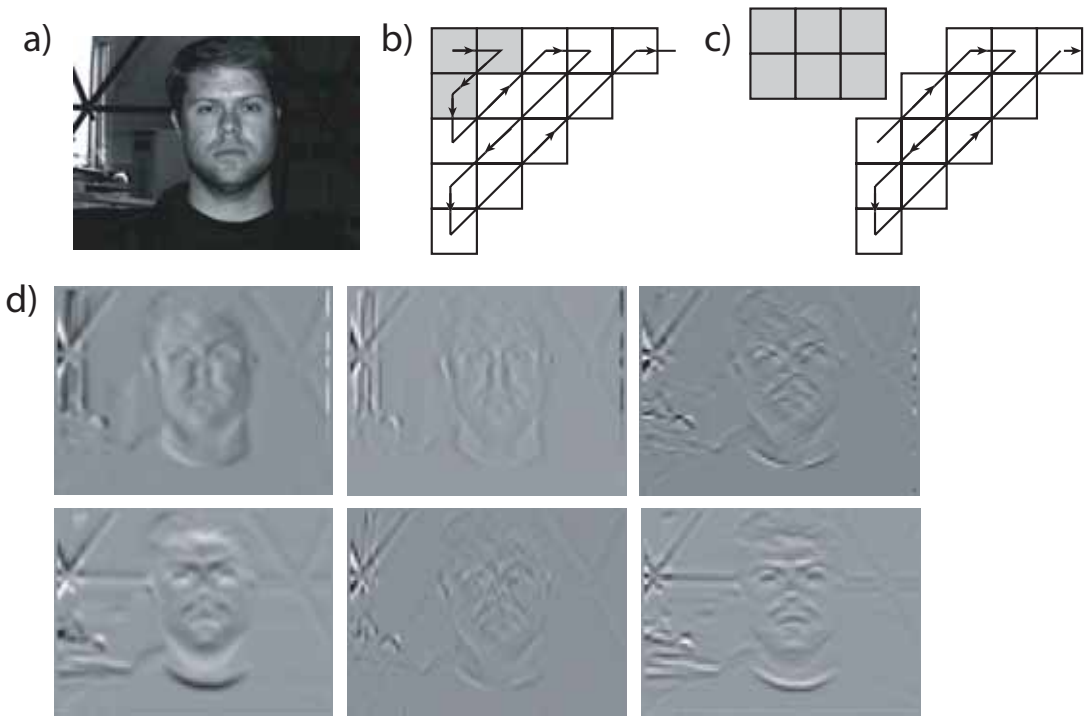


Figure 4-13: DCT-mod2 features.
a) Original image; b) Zig-zag pattern applied to an elementary 8x8 block of DCT coefficients, the coefficients marked in gray being dropped; c) Set of 6 delta coefficients [4-63] marked on gray, complemented by the 12 DCT coefficients from b); d) Set of 6 delta images achieved from the delta coefficients.

Method	Error rate (%) vs. illuminant change		
	Subset 2	Subset 3	Subset 4
DCT-mod2 EGM	0.0	5.0	66.4

Table 4-2: Identification results on the Yale B database.

et al. reported very good results with these features in combination with Hidden Markow Models (HMM) on the same database, we observe that this seems to be a poor combination with EGM. Although no definitive explanation on the large difference of results between [4-63] and our results can be put forward, it is probable that DCT-mod2 features lead to a poor convergence of the EGM algorithm. Nevertheless, it is not possible to directly conclude that HMMs perform better than EGM.

In conclusion, although the DCT-mod2 features are relatively robust to variations of illuminant direction, they seem to be difficult to use in conjunction with EGM.

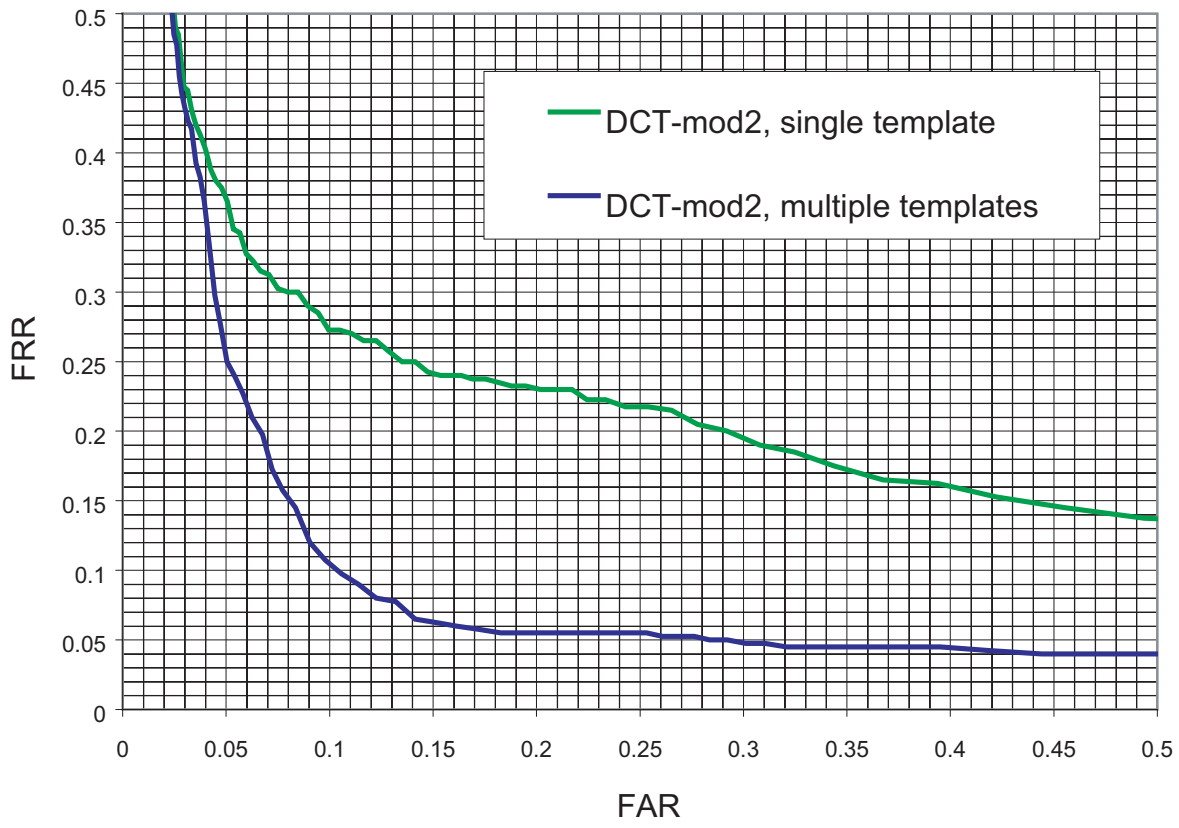


Figure 4-14: ROC of EGM with DCT-mod2 features on XM2VTS.

4.3.3 Mathematical morphology based features

Instead of analyzing the image as a texture - i.e. searching the image for specific spatial distributions - it is possible to concentrate on shape information. For example Yuille proposed to extract valley, edge and peak information [4-79].

Mathematical morphology is a technique widely used in image processing for different purposes: segmentation, shape analysis, filtering, texture analysis, etc. [4-29][4-33]. The two basic operations called erosion and dilation are used to define many other morphological operations (e.g. opening and closing). Given a binary image X (i.e. a 2D matrix of binary pixels) and a 2D binary structuring element B, the erosion is defined as:

$$E_B(X) = \{\vec{x} \,|\, (B_{\vec{x}} \subset X)\} \qquad (4.34)$$

where $B_{\vec{x}}$ denotes the structuring element translated to position $\vec{x} = (x, y)$. This means that the binary eroded image takes the value 1 at all points $\vec{x}$ where the support of $B_{\vec{x}}$ is *included* in X, and the value 0 elsewhere. Similarly the dilation is defined as:

$$D_B(X) = \{\vec{x} \,|\, ((B_{\vec{x}} \cap X) \neq \varnothing)\} \qquad (4.35)$$

This means that the dilated image is assigned the value 1 at all points $\vec{x}$ such that the support of $B_{\vec{x}}$ *hits* X i.e. X and B have a non-empty intersection[12], and 0 elsewhere. Examples of binary dilated and eroded objects are illustrated in Figure 4-15.

These operations can be extended to grayscale images dilated (or eroded) with binary structuring elements, and to grayscale images dilated or eroded with grayscale structuring elements. The general equation is given in Eq. 4.36.

$$\begin{aligned} D_G(X) &= max\{X(\vec{x} - \vec{z}) + G(\vec{z}), \vec{z} \in G, (\vec{x} - \vec{z}) \in X\} \\ E_G(X) &= min\{X(\vec{x} + \vec{z}) - G(\vec{z}), \vec{z} \in G, (\vec{x} + \vec{z}) \in X\} \end{aligned} \qquad (4.36)$$

where G is the support of the grayscale structuring element (i.e. the region where $B_{\vec{x}}$ is non null, see also Figure 4-15). Note that using a flat binary structuring element on a grayscale image simplifies the above

12. The overlap must be at least of one pixel.

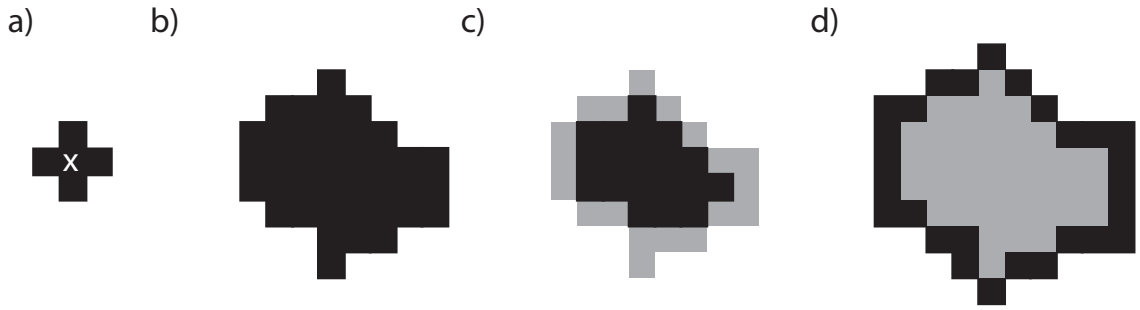


Figure 4-15: Binary dilation and erosion.
a) 3x3 "cross" structuring element (SE) with its origin denoted by an 'x'; the black region is called the *support* of the SE; b) original image; c) eroded image (the gray zone showing the contour of the original pixels); d) dilated image (the gray zone showing the original pixels).

equations to Eq. 4.37. This corresponds to the operation of finding the maximum/minimum of X on the support of the SE.

$$\begin{aligned} D_G(X) &= max\{X(\vec{x}-\vec{z}), \vec{z} \in G, (\vec{x}-\vec{z}) \in X\} \\ E_G(X) &= min\{X(\vec{x}+\vec{z}), \vec{z} \in G, (\vec{x}+\vec{z}) \in X\} \end{aligned} \qquad (4.37)$$

Despite its potential, only few biometric applications use mathematical morphology. Soille et al. [4-69] report that directional linear structuring elements could be used for ridge orientation detection in fingerprint matching (see also Section 5.2). Kotropoulos and Tefas proposed the use of multiscale dilations and erosions [4-39][4-71][4-72] of a luminance image in order to locally extract shape information. Multiscale structuring elements [4-32] are formed by revolution and are thus isotropic[13], providing good invariance to rotation. For instance spherical, paraboloid or disc shapes are used. These shapes are flat (i.e. all the values on the support have the same value, leading to a binary structuring element) allowing further simplification of the calculation of the gray level dilation (respectively erosion) by reducing it to a simple calculation of the maximum (respectively minimum) value appearing on the support as formulated in Eq. 4.37. The flat disc structuring element is defined as follows:

$$G_n(\vec{z}) = \begin{cases} 1, \|\vec{z}\| \le \sigma_n \\ 0, elsewhere \end{cases} \qquad (4.38)$$

13. Due to quantization, structuring elements are however not exactly isotropic, especially the smallest.

where σ_n is the radius of the *n*-th multiscale structuring element. The set of all structuring elements thus allows a multiscale analysis of the image, i.e. the multiscale flat structuring elements are being successively used by incrementing the index or scale *n*, i.e. the radius σ_n.

The use of multiscale flat elements allows to optimize the computations by reusing results calculated for the previous lower radius index *n* – which is possible because the latter is entirely included in the new one – and are thus well suited for parallel hardware implementations [4-70]. Although it would be possible to use decomposition algorithms [4-43], or a running min-max [4-57] for the calculation of each scale, the reuse of results from the previous scale saves more computations.

Following Kotropoulos et al. we chose to use 9 circular flat structuring elements of odd size ranging from 3×3 to 19×19 pixels (see Figure 4-16). The extracted features - that will be called *morphological features* $J_n(x,y)$ from now on - are thus formed of 19 values (i.e. 9 dilated pixels D_n, the original pixel X, and 9 eroded pixels E_n), as expressed in Eq. 4.39. The index *n* of $J_n(x,y)$ will be further called the *level* of this morphology decomposition.

$$\begin{aligned} J(x,y) &= \{D_9(x,y);\ldots;D_1(x,y);X(x,y);E_1(x,y);\ldots;E_9(x,y)\} \\ J_1(x,y) &= D_9(x,y);\ldots;J_9(x,y) = D_1(x,y);J_{10}(x,y) = X(x,y);\ldots \end{aligned} \qquad (4.39)$$

$D_n(x,y)$ means the *n*-th scale of dilation and $E_n(x,y)$ the n-th scale of erosion, keeping in mind that the $S\times S$ size of the structuring element is related to the scale index *n* by:

$$S = 2\sigma_n + 1 = 2n + 1 \text{ with } n=\{1,2,3,...,9\} \qquad (4.40)$$

Authentication results achieved with these morphological features on the XM2VTS database are given in Figure 4-17 and compared to results obtained with Gabor features. It can be observed that the results obtained with morphology features are much less satisfying than those achieved with Gabor features. Indeed a 14% EER is now achieved with single template, respectively 7% with multiple templates, as compared to the previous 6%, respectively 2.25%, for Gabor features with single and multiple templates.

4.3.4 Normalized mathematical morphology based features

Compared to Gabor filtered or DCT transformed images, those obtained using mathematical morphology are strongly illumination de-

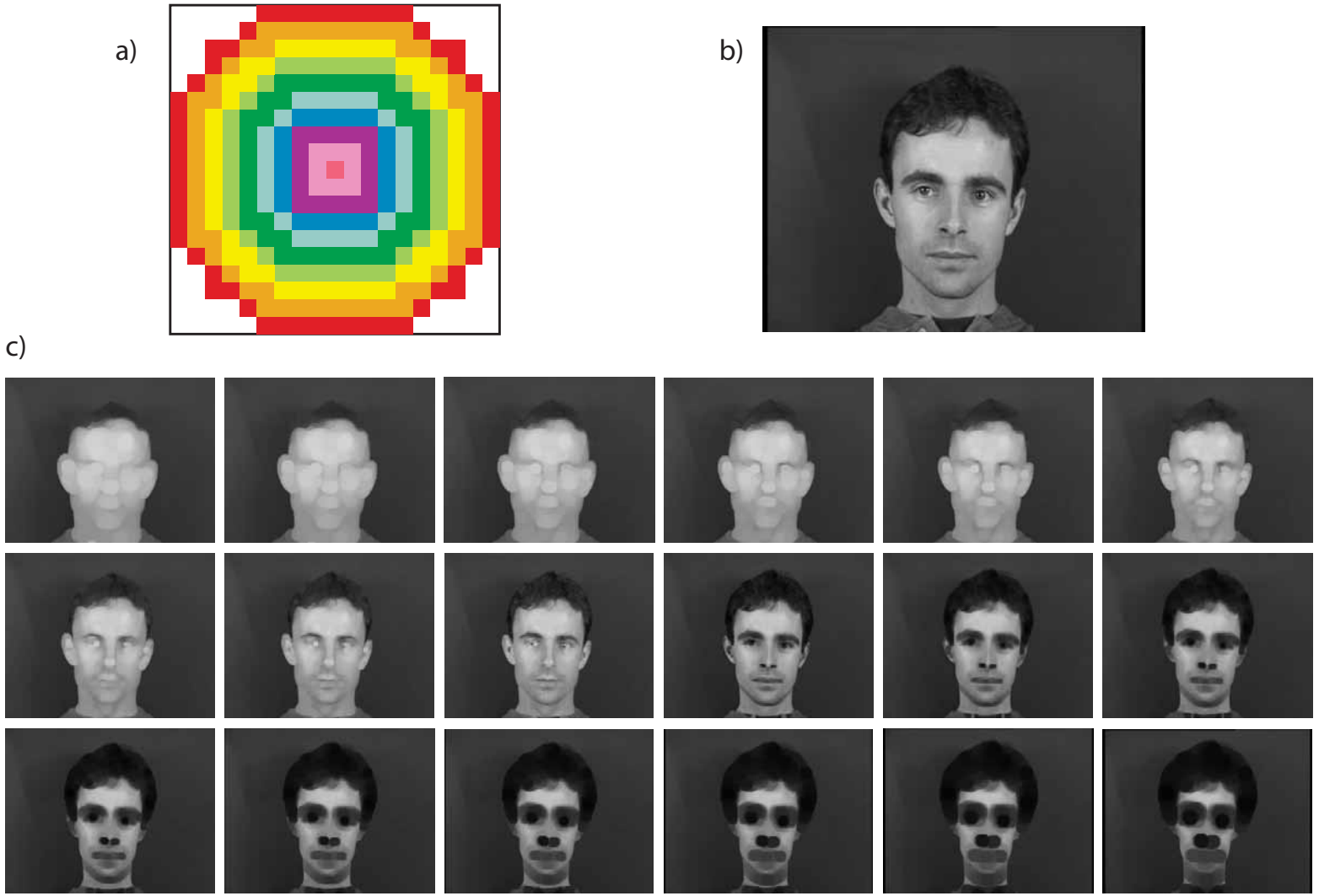


Figure 4-16: Multiscale morphological erosions and dilations.
a) Multiscale structuring elements from the central pixel to the 19x19 element, the colors emphasizing the shape difference with passing from one scale to the next; b) Original image; c) Set of dilated images (from SE 19x19 down to 3x3) and eroded images (from SE 3x3 up to 19x19), according to Eq. 4.39 but without the original image.

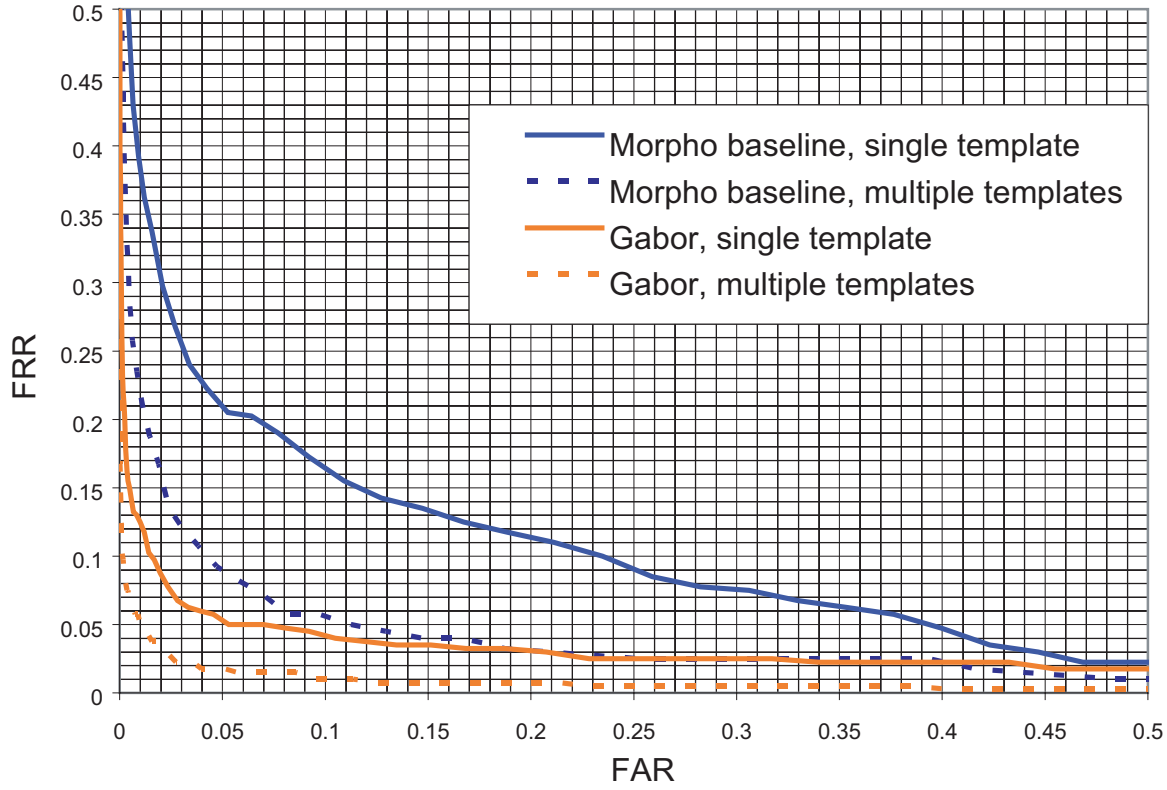


Figure 4-17: ROC of EGM with morphological features (XM2VTS database).

pendent due to the non-removal of the DC component i.e. ambient lighting (not to mention the sensitivity to the direction of lighting). Thus a normalization shall be applied either to the original image prior to performing the morphology computation (see Subsection 4.2.1), or to the morphological features themselves to generate new features that are illumination invariant. These categories can be further divided in model-dependent or model-independent categories, the first depending on the prior detection of the face or of facial features.

Applying a model-independent per-pixel normalization on the original image by considering a local neighborhood of N×N pixels around each original pixel to normalize means using more data (N^2) than when performing the normalization on a subset of the morphological values (<19) for equivalent size of windows (typically N=19). For these reasons, it was chosen to normalize the morphological features independently at each pixel coordinates keeping in mind that the operation with the largest structuring elements involves a neighborhood of 19x19 pixels.

A first normalization attempt was to divide each morphological value by the average of the 19 values (Eq. 4.42 and Eq. 4.41), while a second was to consider only the two extremes for the normalization (Eq. 4.43).

$$M(x,y) = \left(\sum_{n=1}^{19} J_n(x,y) \right) / 19 \qquad (4.41)$$

$$\tilde{J}_n(x,y) = \frac{J_n(x,y)}{M(x,y)} \qquad (4.42)$$

$$\hat{J}_n(x,y) = \frac{J_n(x,y) - J_{19}(x,y)}{J_1(x,y) - J_{19}(x,y)} \qquad (4.43)$$

Figure 4-18: Examples of baseline and normalized morphological features.
Top: from left to right, dilated image with SE of size 17x17, original image, and image eroded with same SE, i.e. scales n={2, 10, 17}; Middle: first normalization (Eq. 4.42) with scales n={2, 10, 17}; Bottom: second normalization (Eq. 4.43) with scales n={2, 10, 17}.

It should be noted that the second normalization (Eq. 4.43) brings the two extreme images $\hat{J}_1$ and $\hat{J}_{19}$ to respectively 1 and 0. Correspondingly, this results in only 17 levels of morphology as compared to the 19 original features.

Figure 4-19: Influence of illumination variation on morphological features.
Same as Figure 4-18 but for an original image with a lighting direction from the left.

Results applied to an image of the Yale face database B are given in Figure 4-18 and Figure 4-19 (they can be compared with Gabor features in Figure 4-10). The global normalization seems successful in both cases, except for the cast shadows caused by the nose. The effect of these shadows is a *local* perturbation not found in XM2VTS images. However, it may be assumed that since the rest of the nodes will cover robust features, the graph will converge to the correct position and hopefully the total graph distance will still be low enough in order to lead to an acceptance. Furthermore it would be possible to attribute less importance to nodes around the nose region using for example statistical methods (see Subsection 4.4.3).

We must stress here that these normalizations are *independent* of any face localization and are thus robust to false detections. Note also that

these normalizations are non-linear (due to the nature of mathematical morphology) and correspondingly not equivalent to dividing the original pixels by a local neighborhood. Indeed, some shape information is already included in the averaged morphological features (Eq. 4.41) or in the extremes used in the denominator of Eq. 4.43.

When introducing Eq. 4.43 in the dichromatic reflection model (Eq. 4.6) it can be observed that the m_b variation (due to the light position relatively to the object) can be considered as a scaling factor and eliminated under some constraints:

- no saturation should occur in the image, as it would lead to a non-linear scaling of the pixel values.
- the geometrical parameter m_b should vary smoothly and should be considered as almost constant over the region spanned by the largest SE.

The size of the structuring element used for the normalization is important, since a too small surface is more sensitive to noise, but in the case of a too large surface, the condition 2 above is generally dismissed by the fact that the surface curvature is not constant, and that the scaling factor is therefore varying. However, empirical results showed that the normalization using the 19x19 element (i.e. $\hat{J}_1$–$\hat{J}_{19}$) was efficient.

Results on the XM2VTS database (see Figure 4-20) show that the normalized morphological features perform much better than the original morphological features, even though this database does not present large variations in illuminant! This can be explained by the fact that the *average* illumination is still slightly varying from image to image although the illuminant direction is always frontal. In addition, we can observe that the normalization of Eq. 4.43 seems to be more effective than the normalization of Eq. 4.42, especially with multiple templates.

In terms of convergence the objective functions of the three methods (see Figure 4-21) show different behaviours. The baseline morphological features seems to have the smoothest objective function. However in this example the matched image (from the XM2VTS database) had a dark uniform background contrasting with the average illumination of the face. Because the morphological features are sensitive to the average intensity of the illuminant, the distance increases continuously when the test graph is displaced by (dx,dy) from the correct location, until it is entirely on the dark background and the objective function

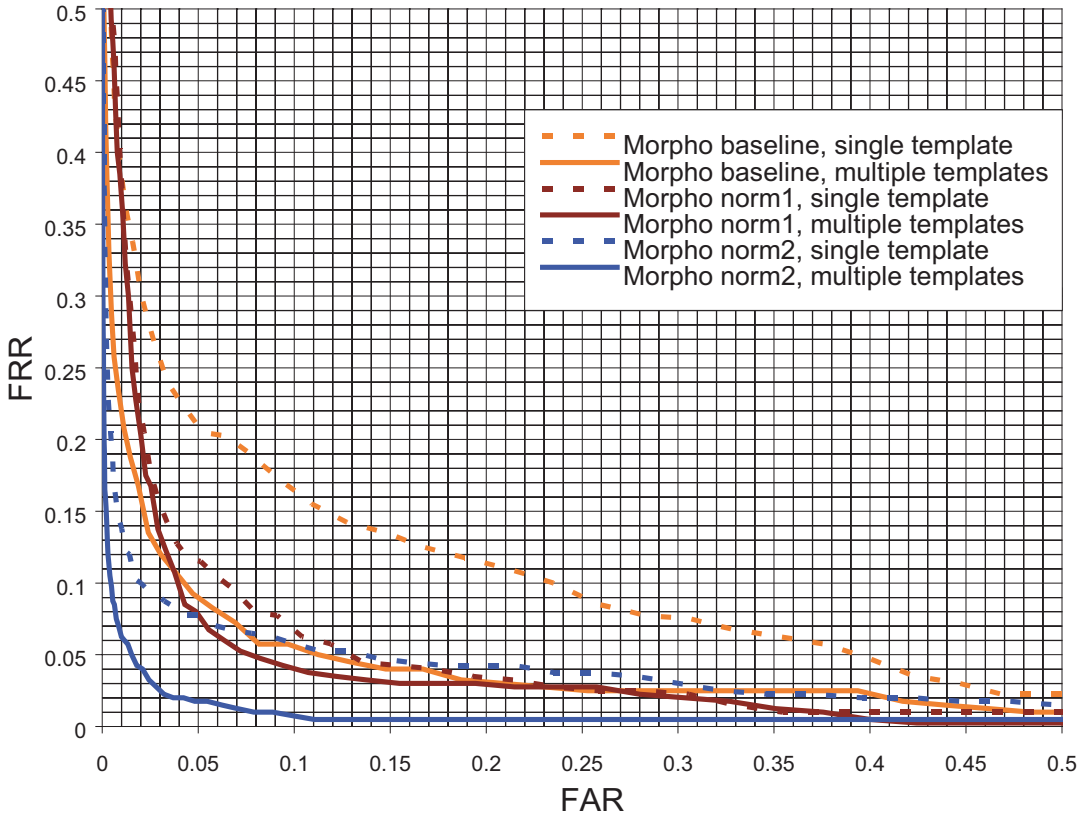


Figure 4-20: ROC of EGM using normalized morpho features (XM2VTS database).

becomes flat. This behaviour would *not* be present with complex backgrounds where the average illuminant of the background might be the same as of the face.

The objective functions of the two normalized feature versions are relatively flat in regions surrounding the correct face location. This is natural since the dilated and eroded values of all scales will tend to similar values around the face location due to the uniform background. This flat surface signifies that the convergence of iterative algorithms will be possible only within a small region around the correct face location. Also, the surface is not completely flat but presents some small undulations leading to local minima that can trap iterative algorithms. The iterative optimization method should be robust to local minima (e.g. simplex downhill or simulated annealing, see Subsection 4.5.1). Problems of convergence can be observed in Figure 4-20 since at low FRR the slope of the curves gets flatter. This means that a number of genuine client test images are not matched correctly and correspondingly that they will be accepted only with a large increase in the threshold[14].

14. For this reason it is sometimes propose to distinguish between False Rejections and Failures to Acquire (FTA). Assuming that misplaced graphs are due to FTA, the FRR would be improved. However this invariably leads to the rejection of the client !

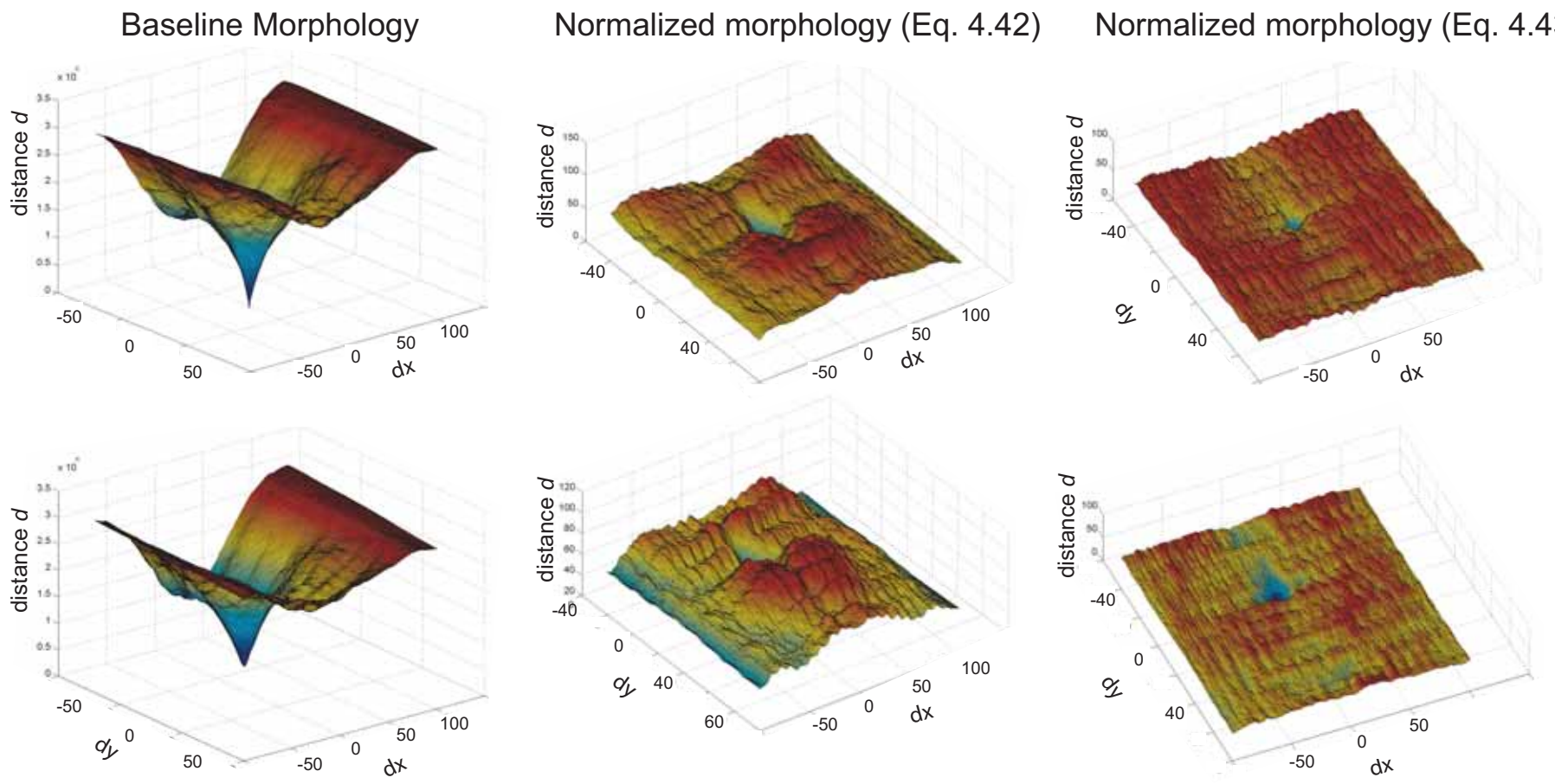


Figure 4-21: Objective functions for baseline and normalized morphological features.
Top row: matching the same image; bottom row: matching a different image.

This number of false rejections seems higher than with Gabor filters. Interestingly the ROC of the second normalization method seems to follow the one from the Gabor features at low false acceptances. Nevertheless, when considering the region corresponding to higher false acceptances, the ROC of the second normalized morphology is higher than Gabor features one, because it is limited by a more important number of false rejections, i.e. of misplaced matched graphs. This possibly implies that Gabor feature performances could be reached with an improved matching strategy. On the other hand the first normalization method seems rather to follow the original morphological features.

Large variations in illuminant are however not present in the XM2VTS database and, as a consequence it is not possible to categorically compare the robustness of the morphological feature to that of Gabor features under “real” lighting conditions. The above results show anyhow that the performance of normalized morphological features in some situations is close to that of Gabor features at a much reduced complexity!

The YaleB database is clearly a more significant test of robustness to variations of illumination. The results (see Table 4-3) are close to those obtained with Gabor features without normalization (Euclidean distance) and with modified DCT. Nevertheless they are clearly worse than the results obtained with normalized Gabor features. Consequently, although their computational complexity is drastically reduced, their distinctiveness and robustness are lower in extreme lighting conditions. Still, these features remain very interesting for applications that are more controlled than images present in the YaleB database, and which require a low-level of complexity.

Besides, the scenario evaluation experiments clearly showed the inefficiency of standard morphological features and the interesting behavior of normalized ones. Yet, in order to be independent on the illuminant spectral distribution, other normalization techniques are necessary (e.g. white balancing), even when working on luminance images (as discussed in Subsection 4.2.1).

Method	Error rate (%) vs. illuminant change		
	Subset 2	Subset 3	Subset 4
EGM norm. morpho (Eq. 4.43)	0.0	15.8	61.4

Table 4-3: Identification results on the Yale B database.

4.4 Feature space reduction

So far the morphological features were extracted and compared directly using an Euclidean distance. However, it can be noted that features from two adjacent levels (i.e. dilation or erosion with two successive structuring element scales) share some common information. It is hoped that this correlation can be reduced (and consequently, that the feature size can be reduced) by using principal component analysis (PCA). After PCA the variation present in the original data set should still be retained in the reduced data set.

The benefit of this reduction is two-fold. Firstly, the storage necessary for a template is reduced and secondly, it is hoped that the use of PCA will improve the recognition results since features that show few variations can be considered as noise and will contribute less to the distance calculation.

The following subsections described briefly PCA as well as Linear Discriminant Analysis (LDA) and report results achieved on the XM2VTS database when applying them to the normalized morphological features and Gabor features.

4.4.1 Principal component analysis

Principal Component Analysis (PCA) is a coordinate transformation method. An observation $\vec{x}$ is projected on the orthonormal basis formed by the so-called Principal Components [4-36]:

$$\vec{y} = \boldsymbol{A}^T \cdot \vec{x} \qquad (4.44)$$

where $\boldsymbol{A}$ is the orthogonal matrix whose k^{th} column is the eigenvector of Σ, the covariance matrix of the distribution of $\vec{x}$, which is assumed to be known. The inverse of this operation - i.e. expressing $\vec{x}$ as a linear combination of linearly independent vectors - is often called the *Karhunen-Loève* expansion (K-L expansion).

The eigenvectors $\boldsymbol{A}$ and eigenvalues Λ are consequently related by:

$$\Sigma \boldsymbol{A} = \boldsymbol{A}\Lambda \qquad (4.45)$$

Multiplying each term of Eq. 4.45 on the right with $\boldsymbol{A}^T$ (i.e. $\boldsymbol{A}$ transposed), and keeping in mind that $\boldsymbol{A}$ is orthogonal (and thus that $\boldsymbol{A}\boldsymbol{A}^T=\boldsymbol{I}$, where $\boldsymbol{I}$ is the identity matrix) leads to:

$$\Sigma = \boldsymbol{A}\Lambda\boldsymbol{A}^T \tag{4.46}$$

Consequently, the eigenvectors $\boldsymbol{A}$ and eigenvalues Λ can be found using the Singular Value Decomposition (SVD) method, that in its general form states that any arbitrary matrix $\boldsymbol{X}$ can be decomposed into a multiplication of two orthogonal matrices $\boldsymbol{U}$ and $\boldsymbol{A}$ and a diagonal matrix $\boldsymbol{L}$:

$$\boldsymbol{X} = \boldsymbol{U}\boldsymbol{L}\boldsymbol{A}^T \tag{4.47}$$

In our case $\boldsymbol{U}=\boldsymbol{A}$ by the fact that the covariance matrix is symmetrical. Efficient numerical methods for calculating the SVD of special matrices exist [4-3] and were used during the simulation.

However, the hypothesis that the true covariance matrix of the distribution is known is generally false. Alternatively, we have at our disposal the sample covariance matrix $\boldsymbol{S}$ built from n independent observations of a p-element random vector $\vec{x}$. The elements S_{jk} of $\boldsymbol{S}$ can be expressed as:

$$S_{jk} = \frac{1}{n-1}\sum_{i=1}^{n}(x_{ij}-\bar{x}_j)(x_{ik}-\bar{x}_k)^T \tag{4.48}$$

$$\bar{x}_j = \frac{1}{n}\sum_{i=1}^{n}x_{ij} \tag{4.49}$$

where x_{ij} indicates the j-th element of the i-th observation $\vec{x}$, and j=1..p, k=1..p.

In Eq. 4.44 the projection matrix $\boldsymbol{A}$ shall consequently be replaced by the matrix containing the eigenvectors of $\boldsymbol{S}$ instead of the eigenvectors of the true covariance matrix. Note that for problems where n is much smaller than the dimensionality of the feature vectors, the rank of $\boldsymbol{S}$ might be smaller than (n-1).

A measure of how well a subset of eigenvectors captures the variance contained in the sample scatter matrix is the so-called *cumulative percentage of total variation* (CPTV) accounted for the k first principal components:

$$t_k = 100\left(\sum_{j=1}^{k} l_j\right) \Big/ \left(\sum_{j=1}^{p} l_j\right) \tag{4.50}$$

where l_j are the p elements of Λ (Eq. 4.45), i.e. the eigenvalues of the system, sorted by decreasing value. The features projected on the k first principal components, which are associated to the largest eigenvalues, are further called the *most expressive features*, or MEF.

The purpose of PCA is then to reduce the number of eigenvectors in $\boldsymbol{A}$ and still retaining as much as possible of the variation t_k present in the p original variables. Additionally, it is hoped that - since the eigenvectors corresponding to the smallest eigenvalues marginally contribute to the variance - their suppression will enhance the classification because they can be considered as noise. However, as it will be illustrated now, this is obviously not always the case.

Let us consider two normally distributed classes for which the direction of the largest within-group variation belonging to each group, is the same as the direction of the between-groups variation (see the left hand-sided part of Figure 4-22, depicting the distribution of two features belonging to two distinct groups, marked in blue and green, respectively).

Obviously the largest (i.e. first) principal component is also oriented in this direction, and retaining only this principal component will enable an excellent separation of the two groups. Now let us consider the case where the largest within-group variation is orthogonal to the between-groups orientation (cf. the right hand-sided part of Figure 4-22).

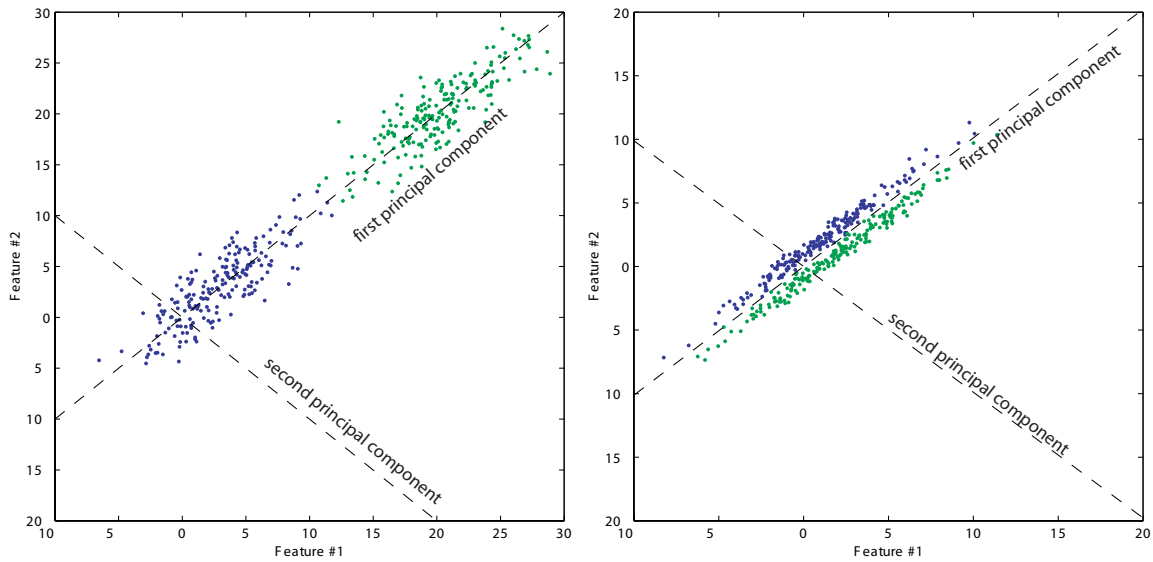


Figure 4-22: Principal components of two classes distributions.
In the left part of the figure, the first principal component is sufficient for class separation, while it is not the case in the right part (although the two classes are still separable).

In this case the largest (i.e. first) principal component contains no useful information for classification and the discriminating information was thrown away with the eigenvector associated to the smallest eigenvalue. Interestingly, the cumulative percentage of variation was of 98% in the first case, and 96% in the second. It shall now be clear that this measure does not give an idea about the most discriminating principal components! The automatic selection of "good" principal components therefore translates in heuristic attempts. It is sometimes possible to select manually a subset of principal components by visually inspecting their effect.

PCA was applied to Gabor features and to normalized mathematical morphology features (but not to DCT features). The CPTV of these features obtained on the XM2VTS database while using 200 client images and considering a node independent approach (see below) are reported in Table 4-4.

Feature type	k=1	k=2	k=3	k=5	k=10
Gabor	62%	74%	82%	91%	97%
Norm. morpho. (Eq. 4.43)	54%	74%	81%	89%	97%

Table 4-4: Cumulative percentage of total variation (t_k) of extracted features.

As it can be observed from Table 4-4, the three first principal components account for more than 4/5 of the variation. Therefore, keeping only the first 3 principal components should realize an effective compression of the templates, while maintaining performances. The coprocessor architecture to be described in Chapter 6 was designed especially for peforming the PCA feature reduction from 17 normalized morphological features down to 3 features.

Application of PCA to the normalized morphological features is done by calculating the sample covariance matrix using all the node features from all the reference images in the database leading to 64×200 samples for the single reference case. The PCA matrix calculated based on the training set is then applied during evaluation and testing. Results are reported in Figure 4-23 by varying from 3 to 11 the number of principal components retained. It can be observed that increasing the number of principal components seems to increase the performance but that this improvement is limited and that adding further components

will not improve results. Consequently, even if 5 principal components contain the major part of variance in the samples (cumulative percentage of the five first eigenvalues is around 90%) the performance of the recognition seems not to be improved. This behavior was predictable because as stated earlier, the largest eigenvalues do not necessarily contribute to the discriminance. Correspondingly, it might be necessary to remove certain principal components but heuristic methods were not experimented.

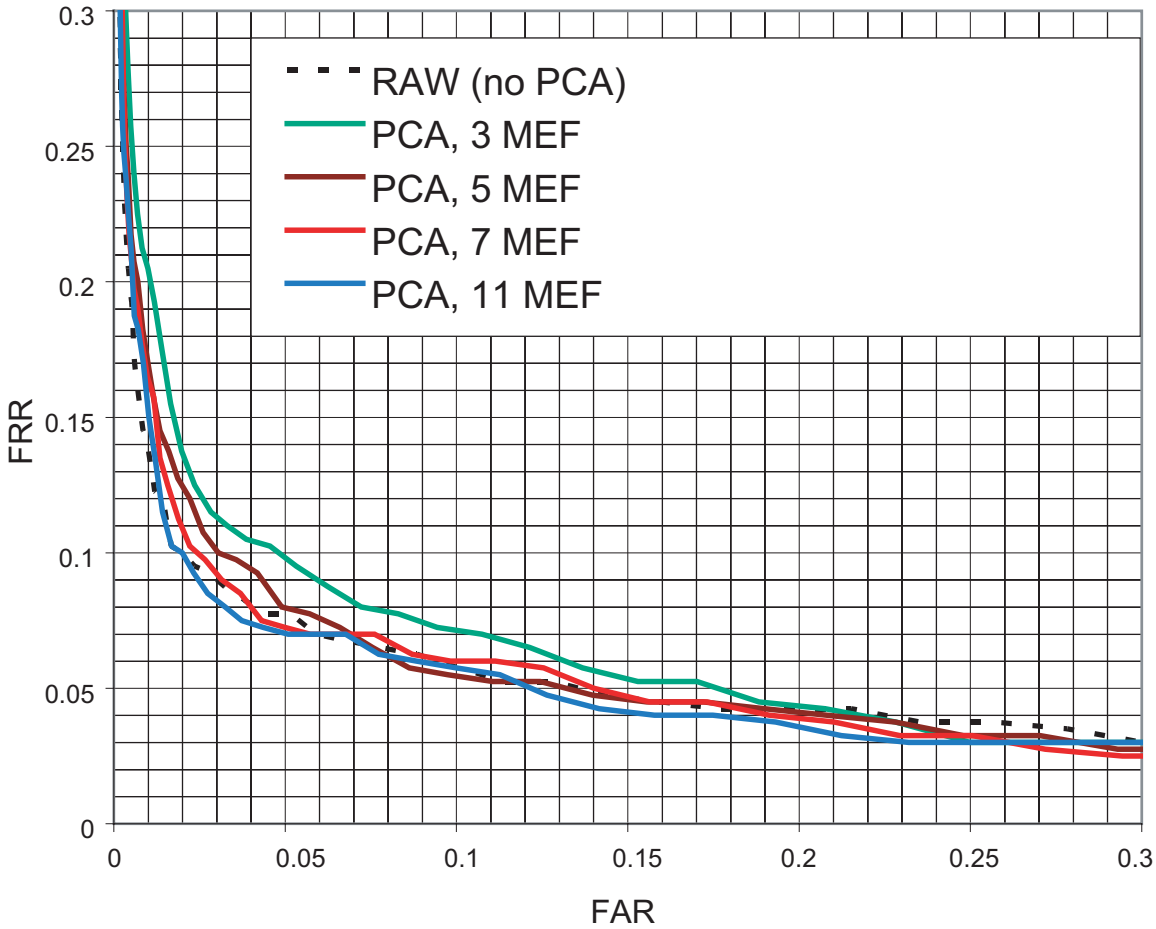


Figure 4-23: ROC of EGM using normalized morphology and PCA.
Only the *k* largest principal components are retained (*k*=3, 5 , 7) and used to produce the *k* most expressive features (MEF). The PCA was applied indepently of the node index. The XM2VTS database was used for this example.

In the above application of PCA all the features were equally considered i.e. independently of the node indexes. Kotropoulos et al. [4-39] use a node-dependent version, i.e. using one PCA matrix for each node. However, if each node feature vector can be considered as an independent random process, then the first version should give results similar to a node dependent version. A comparison of the performance of both approaches is given in Figure 4-24.

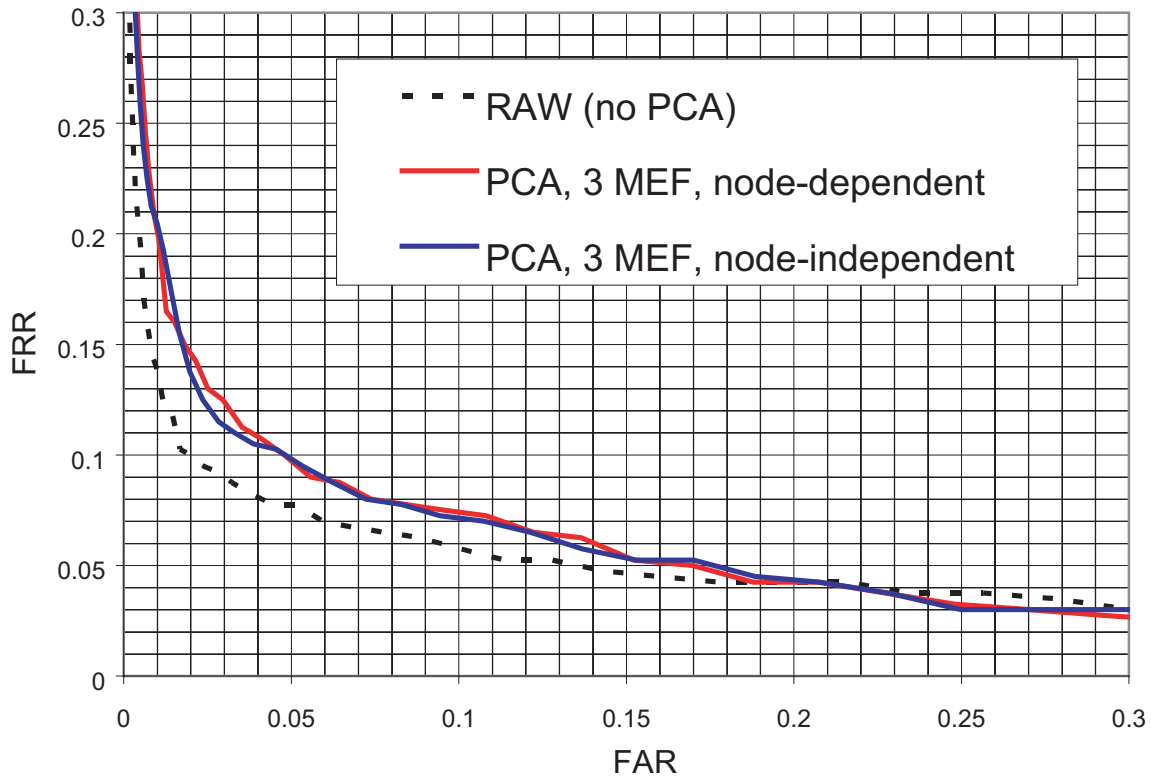


Figure 4-24: ROC of node-independent vs. node-dependent PCA (XM2VTS database).

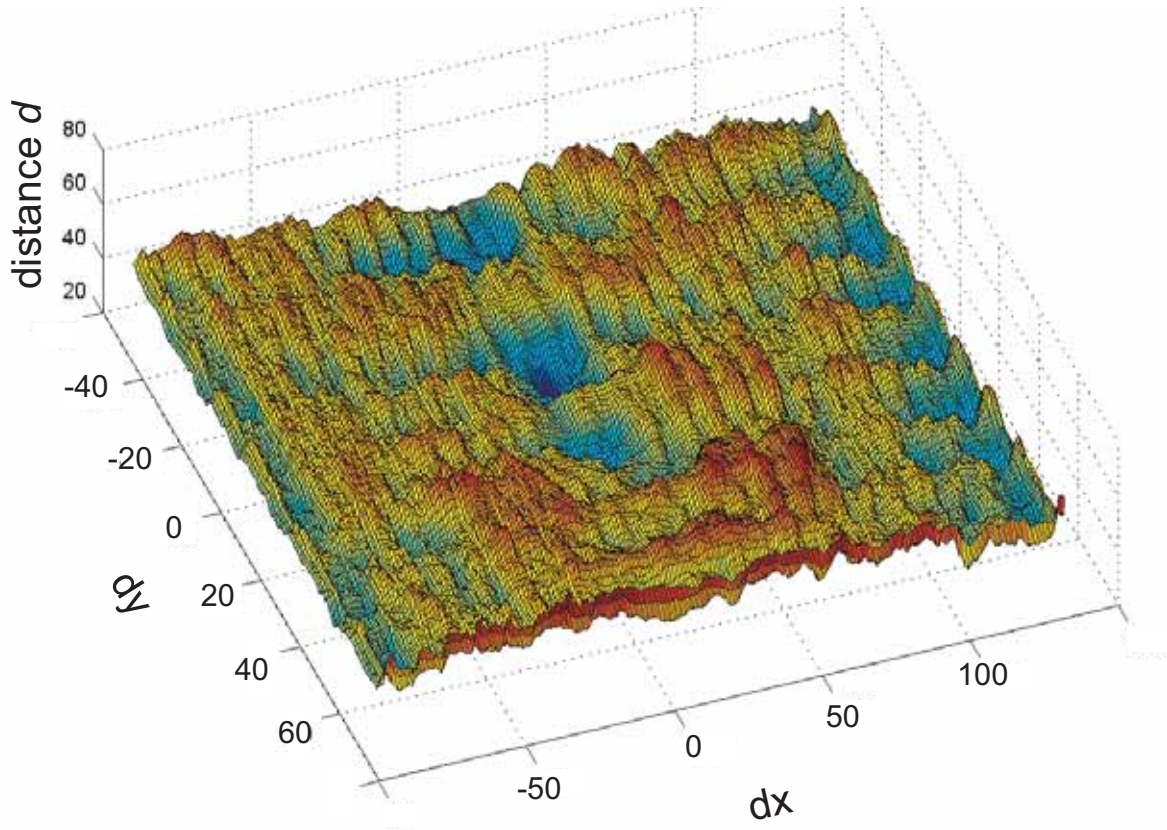


Figure 4-25: Objective function for normalized morphological features with PCA.
Only the three most expressive features are retained, independently of the node positions.

No clear difference is observed if we remember that the error margins for such measurements account for around 0.5% on false rejections.

Similarly to the situation observed for the RAW case (i.e. without applying PCA), the convergence of the matching process may be weakened because the reduced features are more locally discriminating. The objective function obtained with PCA features is depicted in Figure 4-25. It may be observed that although the minimum can still be found the number of local minima and the amplitude of the oscillations are larger.

Finally, the Gabor features were tested with PCA using the dot product metrics only, and keeping 10 features out of 18. Results are reported in Figure 4-26. Similarly to the case of mathematical morphology, the use of PCA slightly degrades the performance of the algorithm rather than to improve it.

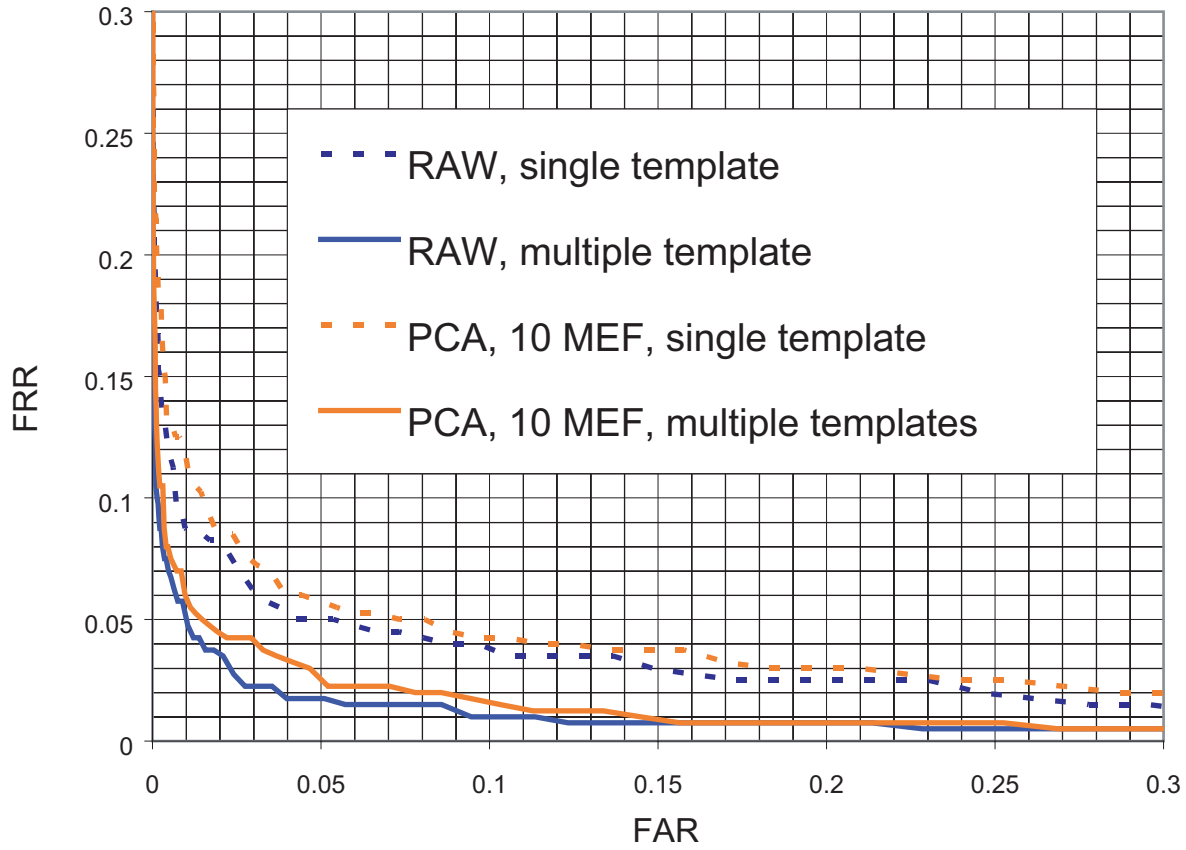


Figure 4-26: ROC of EGM using Gabor features (dot product metrics) and PCA.
10 features out of 18 are used. Single and multiple template scenarios are reported.

4.4.2 Linear discriminant analysis

Using the principal components does not guarantee that the class separation will be optimal as already illustrated in Figure 4-22. A better statistical analysis - the so-called linear discriminant analysis - consists in performing a feature space transformation that maximizes the between-groups variance while simultaneously minimizing the within-group variance. The within-class and between-classes scatter matrices[15] of L classes for a random vector $\vec{x}$ are defined in Eq. 4.51 and Eq. 4.52, respectively [4-24].

$$\boldsymbol{S_w} = \sum_{i=1}^{L} P_i E\left\{(\vec{x}-\vec{m}_i)(\vec{x}-\vec{m}_i)^T \middle| \omega_i\right\} = \sum_{i=1}^{L} P_i \Sigma_i \qquad (4.51)$$

$$\boldsymbol{S_b} = \sum_{i=1}^{L} P_i (\vec{m}_i-\vec{m}_0)(\vec{m}_i-\vec{m}_0)^T \qquad (4.52)$$

P_i is the probability that class i appears; $\vec{m}_i$ is the expected vector of class i and $\vec{m}_0$ is the expected vector of the mixture distribution:

$$\vec{m}_0 = E\{\vec{x}\} = \sum_{i=1}^{L} P_i \vec{m}_i \qquad (4.53)$$

The matrices S_w and S_b can be combined in a mixture scatter matrix:

$$\boldsymbol{S_m} = \boldsymbol{S_w} + \boldsymbol{S_b} = E\{(\vec{x}-\vec{m}_0)(\vec{x}-\vec{m}_0)^T\} \qquad (4.54)$$

The assumption of equally probable classes allows to extract $P_i = 1/L$ from the equations, and consequently, to simplify the problem.

A commonly used optimization criterion is the maximization of the *Fisher Linear Discriminant*:

$$J = tr(\boldsymbol{S_w^{-1}} \boldsymbol{S_b}) \qquad (4.55)$$

where *tr()* refers to the trace function, i.e. the sum of the diagonal elements of $\boldsymbol{S_w^{-1}} \boldsymbol{S_b}$.

The matrix operating the corresponding transformation is formed of the eigenvectors of $\boldsymbol{S_w^{-1}} \boldsymbol{S_b}$ that correspond to the largest eigenvalues. Fi-

15. Note that the terms within-group and within-class, respectively between-groups and between-classes, are used as synonyms. The second denomination clearly establishes the relation of a group with a class.

nally, this is equivalent to taking the complementary set of eigenvectors of $\boldsymbol{S}_{\boldsymbol{m}}^{-1}\boldsymbol{S}_{\boldsymbol{w}}$ corresponding to the smallest eigenvalues (see e.g. Fukunaga [4-24] for a demonstration).

So far we considered the problem of a single transformation simultaneously maximizing the class separability of L classes. It is however possible to consider a simplified two-class problem in which we consider one of the L classes and the mixture of the remaining (L-1) classes. For this special case, it can be shown that only one reduced feature (i.e. the eigenvector corresponding to the largest eigenvalue) is necessary to preserve all classification information in the sense of J in Eq. 4.55 [4-24]. This is the approach followed by Kotropoulos [4-39] as well as by Duc [4-17], although they don't recommend the use of a single reduced feature. Similarly to Yang [4-77], Kotropoulos proposes to apply PCA first in order to reduce first the dimensionality of the feature space to the rank of the total scatter matrix. This is judicious since the training set comprises less images than the number of features per node, thus leading to ill-conditioned problems (scatter matrices are singular).

Each node was considered individually and the number of features obtained after PCA (the prior most expressive features) was conserved for the LDA features (the so-called *most discriminating features*, or MDF), i.e. no further dimension reduction was operated. In this work, LDA was applied on three principal components as a two-classes problem as well as an L-classes problem.

The first observation made after experimentation is that the L-classes LDA shows no improvement over standard PCA, which was seen to perform already worse than with RAW features. This is possibly due to the fact that it is not easily possible to simultaneously maximize the separation of 200 classes. For two-classes PCA (i.e. client dependent LDA), we tried to keep 3 MDFs as proposed in Tefas and Kotropoulos, and alternatively to keep only a single MDF as suggested by Fukunaga. Unfortunately, two problems arose while using these versions of the LDA: firstly the convergence is largely reduced due to the fact that the features are now extremely discriminant. Consequently, it is necessary to perform a first matching step using RAW or PCA features before applying LDA. Secondly, the number of training samples in the XM2VTS face database is by far too small for an efficient generalization of the LDA. Results obtained with this database were consequently not sufficiently significant to assess whether LDA can improve verification results or not.

4.4.3 Node weighting

In the previous subsections we applied PCA and LDA to morphological features, i.e. we reduced the number of features per node, then calculated the Euclidean distance between test and reference reduced features, and finally added all the node scores to form a global graph score (for deformation penalty, see Subsection 4.5.2).

$$d = \sum_{n \in nodes} \left\| A^T \vec{x}_{n,ref} - A^T \vec{x}_{n,test} \right\| \qquad (4.56)$$

In this scheme all nodes contribute equally to the graph score, whereas there are certainly some node locations that contain more discriminant information than others (e.g. some regions are more subject to cast shadows and are thus less reliable). Consequently, it should be possible to apply LDA to the node scores and to build the graph score as a weighted sum of node scores, the weights being deduced statistically from the training set. This is indeed done by Kotropoulos [4-38] and called *discriminant grids*. The distance becomes:

$$d = \sum_{n \in nodes} DP_n \cdot d_n \qquad (4.57)$$

The node distances d_n might be obtained by using the Euclidean norm of morphological features directly, or by using their PCA/LDA reduced version. The weights DP_n (discriminating power) are calculated using the Fisher's Linear Discriminant ratio:

$$DP_n = \frac{(m_{n,inter} - m_{n,intra})^2}{\sigma^2_{n,inter} + \sigma^2_{n,intra}} \qquad (4.58)$$

where $m_{n,inter}$ and $m_{n,intra}$ represent the mean intra-class and inter-classes matching error at node n, respectively. Similarly, and $\sigma^2_{n,inter}$ and $\sigma^2_{n,intra}$ are the corresponding variances. Again, the number of samples used for estimating these parameters is generally small (e.g. 4 samples in the XM2VTS database with configuration II of the Lausanne Protocol). It is consequently not sure that a generalization will work for novel views. Moreover, this might be very sensitive to occlusion, because some nodes could get an important weight and contribute entirely to a rejection based only on a single node discrepancy. Results for node weighting using Gabor features with dot product metrics on XM2VTS database are given in Figure 4-27. The 4 images of the

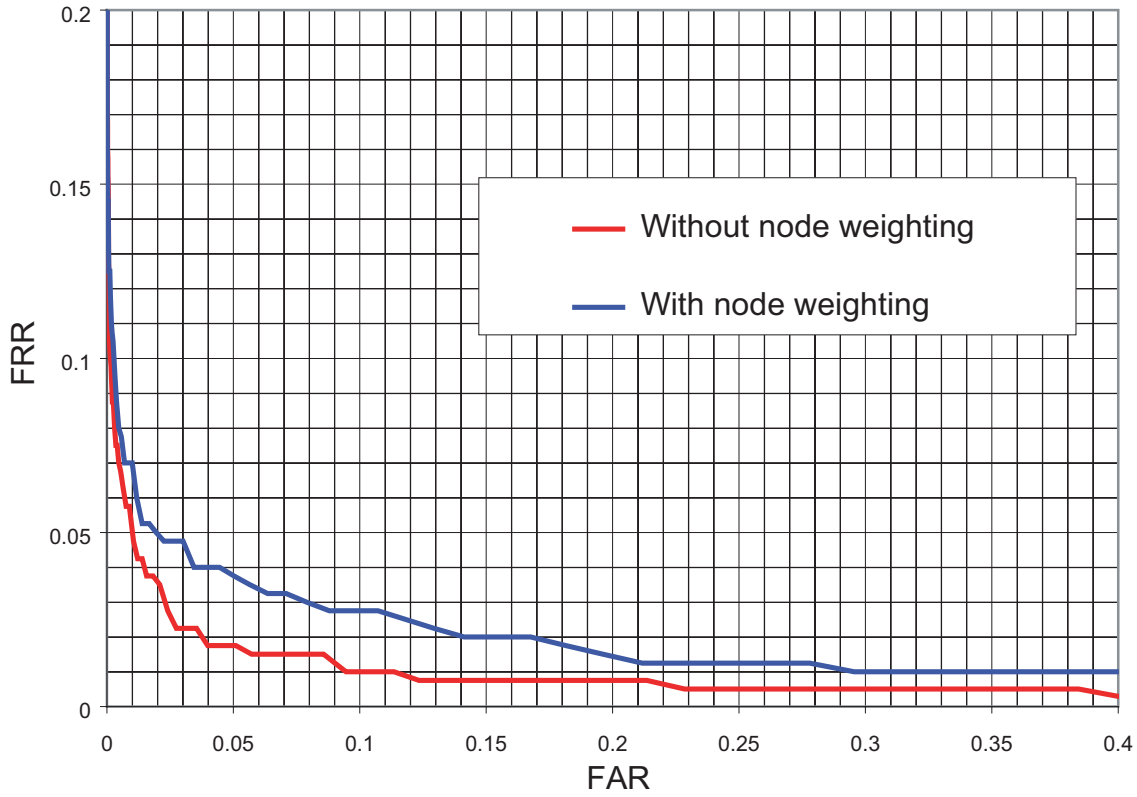


Figure 4-27: Node weighting (i.e. discriminant grids) vs. standard method.

training set were used to calculate 600 m_{intra} and 119400 m_{inter} errors, and the multiple template scheme was used. Surprisingly, the performance of the node weighting method is lower than the one achieved with the "standard" multiple template. This could be again due to the too small number of training templates to estimate the m_{intra} parameter (only 3 images in this case).

In conclusion, node weighting can be useful depending on the application, if a large training set is available and/or if the templates are updated relatively often. This can however be added as a post-processing step, since the largest part of the algorithm remains the same.

4.5 Graph matching

The operation of finding a correspondence between the orthogonal reference graph and a distorted graph in a test image is called graph matching. The general matching problem is definitely a highly complex operation (P or NP-complexity [4-41]). However Lades [4-40] proposed a simplified version of graph matching consisting in the exploration of only a limited number of displacements. Recalling Subsection 3.4.5, there are two main approaches to describe how facial features can move (e.g. lip contours) or can be modified (e.g. opened/closed eyes): ei-

ther by using physical models only, which are strong simplifications not relying on the underlying physiological models, or by using statistical models of deformations. Since most authors used the first approach in connection with graph matching, we will start describing this approach first. At the end of this section we will give directions for using probabilistic deformation models in the form of “eigengraphs” or “eigenwarps” related to the theory of eigenshape/shapespace.

Independently of the approach followed (i.e. either physical or statistical), we will first try to find the best position of our rigid (i.e. orthogonal) graph before considering elastic deformations. Kotropoulos et al. [4-39] claim that this solution is not satisfactory due to the fact that the rigid graph may not converge to the correct position, and that the successive elastic matching operations will not be able to recover the correct final position of the test graph. We observed that this claim is not true as long as the next points are respected:

- the features need to be slowly varying, that is the objective function needs to be smooth (see section Subsection 4.3.1 and Subsection 4.3.3 for examples of objective functions).
- the scale of the graph and its orientation need to be adapted already in the rigid search phase.

The rest of this section describes a robust method for finding the best position of the rigid graph, and two methods for finding the best correspondence between two graphs using elastic deformations.

4.5.1 Rigid matching using iterative methods

Lades et al. [4-40] originally proposed to perform graph matching in two phases, a first phase consisting in a rigid matching – i.e. scanning the image with an orthogonal version of the test graph – and a second phase where nodes are displaced elastically around the position resulting from rigid matching. In a more recent contribution, Lades [4-41] proposes to use a “modified Global Move” scheme, which consists in transforming the test graph in size and orientation as well as translating it, instead of the simple rigid scanning approach. Kotropoulos et al. [4-39] tried to optimize the translation parameters simultaneously to the deformations for the above-mentioned reasons. They make no assumption on the size of the graph and on its orientation. Duc et al. [4-17] use the same approach as the first attempt by Lades, i.e. scanning with a rigid graph, however using coarse-to-fine strategies with gaussian pyramids.

Clearly, the second method proposed by Lades [4-41] is more robust for images containing rotations and scaling variations. Already on database images, the effect was shown to be important, and it can be expected that the rotation and scaling variations will be even larger in "real-world" images taken with an image acquisition device held by the user.

This phase of rigid matching can thus be considered as a function minimization operation with position, scale, and orientation parameters. Lades [4-41] uses a modified Monte-Carlo method for finding this minimum, which is subsequently called the best "Global Move". In this work we used a Simplex Downhill [4-42] method (sometimes called Nelder-Mead) since it seems simpler (i.e. not using gradients) and sufficiently robust (e.g. relatively robust to local minima) for our purpose. Two implementations of this matching method were carried out, with dimensions of the search space being either:

- four: two dimensions for translation, one for scaling and one for orientation, leading to a simplex containing five four-dimensional vertices.
- five: two dimensions for translation, two for scaling and one for orientation, leading to a simplex containing six five-dimensional vertices.

Although all results presented here were using four-dimensional vertices, the second implementation has the advantage of accounting for out-of-plane rotations of the head that might deform the graph. For instance, turning the head left or right will tend to shrink the graph width, whereas the graph height will remain unchanged. Of course the features should be modified accordingly: for instance the structuring elements in morphology should be no longer circular but become ellipsoid.

The five (respectively six) initial vertices can be chosen randomly or can be based on the results of a face detection process (see Section 5.1). If no preliminary face detection is available, a sparse rigid scanning of the image can be performed as in [4-40] with a few scales (e.g. 3 different scales; no orientation seems necessary) accounting for the expected variations in the acquisition scenario. This latter solution was used in all the experiments reported here unless stated otherwise.

Even though the graph scale (i.e. the distance between nodes) is adapted using the Simplex Downhill matching, we still extract the same fea-

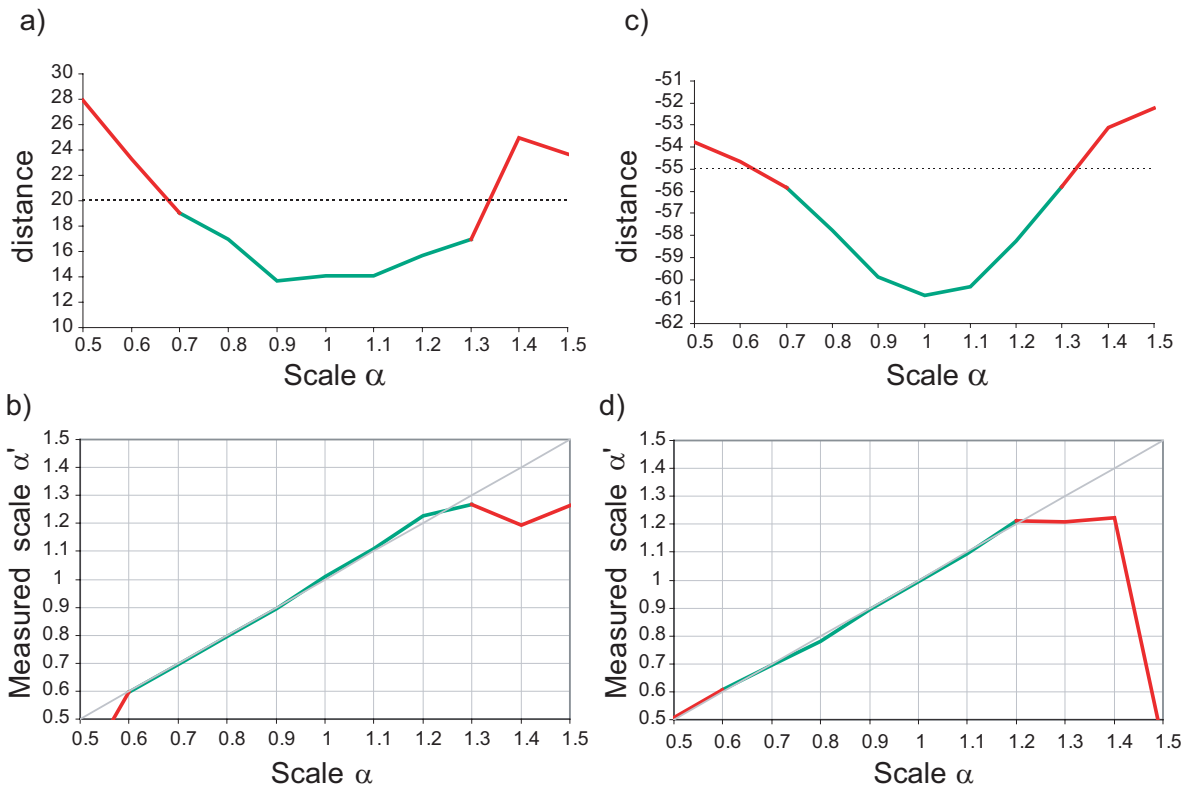
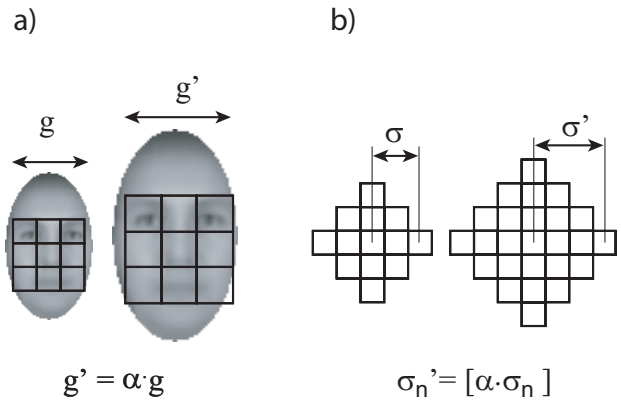


Figure 4-28: Effect of face scaling on the matching distance and measured scale.
In this experiment, a test image from the XM2VTS database is scaled (i.e. interpolated), with different scale factors α, compared to the reference image size (α=1), which was used to extract the reference template. The scaled versions are then matched individually against the reference template of the same client extracted from a different image, resulting in several graph distances and measured scales α'. Each measured scale α' corresponds to the ratio of the matched test graph width over the reference graph width.
a) Matching distance obtained for the scaled test images against the reference template (α=1); distances in red would lead to a rejection, while distances in green would lead to an acceptance; b) Measured scale factors vs. applied scale factors; the diagonal line α'=α corresponds to correct scale detections (in green; incorrect scale detection are reported in red); c) and d) Same as a) and b) but for Gabor features and dot product metrics instead of normalized morphology.

ture information at each node location, i.e. the structuring elements are not modified according to this scale ratio. Yet, with large scale differences (typically, a factor of 2 in the face width is possible by stretching or bending the arm) the patterns extracted from the face with the morphological operators will be scaled too. Consequently, when using mathematical morphology, a shape feature extracted with a 9x9 structuring element in a 50 pixel-wide face would definitely be best extracted with a 13x13 structuring element in a 70 pixel-wide face[16].

16. Applying a simple scaling: 9×70/50=12.6≅13



α	σ								
0.8	1	2	2	3	4	5	6	6	7
0.85	1	2	3	3	4	5	6	7	8
0.9	1	2	3	4	4	5	6	7	8
0.95	1	2	3	4	5	6	7	8	9
1	1	2	3	4	5	6	7	8	9
1.05	1	2	3	4	5	6	7	8	9
1.1	1	2	3	4	6	7	8	9	10
1.15	1	2	3	5	6	7	8	9	10
1.2	1	2	4	5	6	7	8	10	11

Figure 4-29: Adaptation of the size of structuring elements.
a) *g* being the face width in the reference image, the graph size will be scaled by a factor α in the test image, so that it corresponds to the face width *g'* in the test image; b) each SE radius σ_n must be scaled accordingly, resulting in σ_n'; c) table delivering the index *n* of σ_n' in function of the radius index σ_n, and of the scale factors α corresponding to ±20% scalings; the indexes *n* are rounded to the closest integer radiuses.

Experiments show that the graph registration is generally correctly carried-out for scale factors ranging from 0.7 to 1.4 although the resulting distance (without e.g. adapting the structuring elements) is increased as can be observed on Figure 4-28. The distance is significantly larger for scale factors smaller than 0.8 or larger than 1.2.

The modification of the graph scale during the iterative rigid matching, but without scaling the morphological features, will therefore only be able to compensate for scale factors in the order of ±0.1 or 0.2, but will be ineffective otherwise.

For example in case of mathematical morphology, a difference α of scale in the image shall indeed correspond to a difference α of scale in the radius of the structuring elements (the radius σ_n of the SE was introduced in Subsection 4.3.3, Eq. 4.40). Adaptation of the structuring element radius based on the measured scale factor is illustrated in Figure 4-29.

We therefore propose an elegant solution to solve this problem, namely to switching (i.e. by reindexation) between SE radiuses dynamically, according to the measured scale that was estimated in the frame of the matching process. For example, with a detected scaling of 85%, features extracted with an SE radius of 9 in an α=1 image, would be replaced by features extracted with an SE radius of 8 (i.e. an SE of 17x17 pixels instead of 19x19 pixels), features extracted with a σ_8 SE would

be replaced by features extracted with a σ_7 SE, and so on. Note that SEs with radiuses larger than 9 are normally not available in the feature extraction and that the radius of the SE must be at least 1.

Unfortunately, the normalization along Eq. 4.43 depends on the values obtained with the smallest and largest structuring elements. That is, in the normalization of Eq. 4.43 we should also replace the terms $J_1(x,y)$ and $J_{19}(x,y)$ by the adapted extrema (i.e. for instance $J_1(x,y)$ and $J_{15}(x,y)$ for $\alpha = 0.8$). Consequently, in order to adapt the scaling during the rigid matching, we would need to perform the normalization on the test image several times (i.e. whenever the scale factor α' changes significantly), leading to a more expensive solution.

In terms of robustness, this solution is somewhat better for $\alpha < 1$ as can be observed in Figure 4-30. Globally, as reduced images also correspond to reduced information, the distance matching distance is anyway increasing with a smaller α. It is also logical that for $\alpha > 1$ the distance will hardly be improved by the SE radius switching. Surprisingly though, the matching distance for $\alpha > 1$ was not improved by the SE radius switching as compared to the situation where no switching was performed.

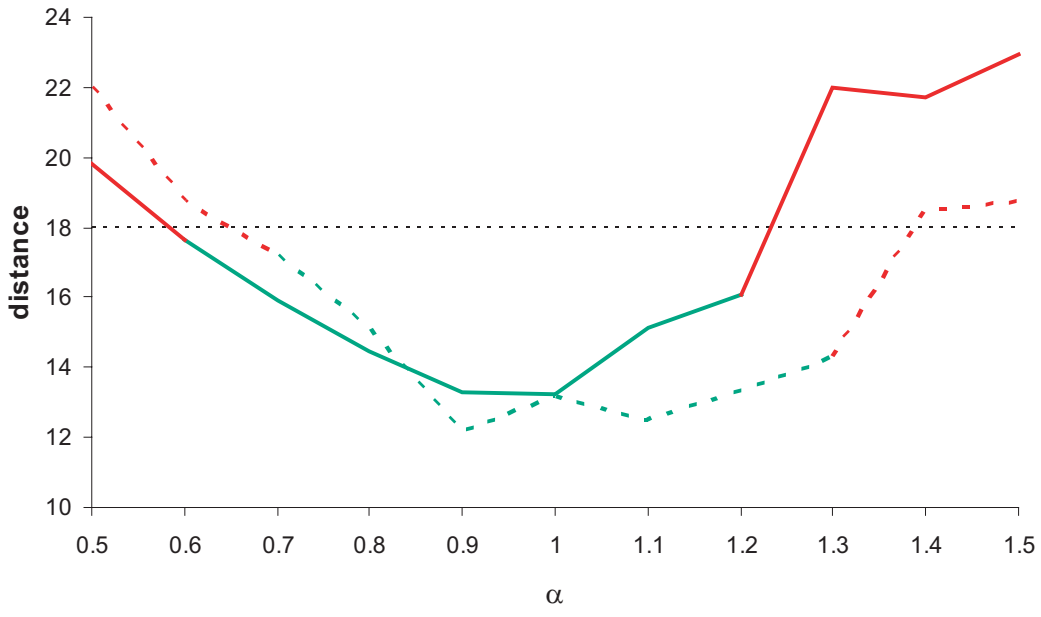


Figure 4-30: Comparison of matching distances with and without SE radius switching. The dashed line represents the matching distance obtained with scaled test images against a reference template (α=1), when the SE radius switching is not used. The plain line corresponds to the same experiment, when the SE radius switching is actually used. For $\alpha < 1$ the matching distance is somewhat smaller when applying the SE radius switching scheme (with re-normalization of the morphological features using the new extremum features J_1 and J_{19}, according to the measured scale α'). SE radius switching seems however not efficient for α>1.

Finally, a more effective solution is to store several reference templates in multiple scales and to switch between reference features based on the measured scale, instead of switching the SE radiuses, when the graph scale is more than 20% away of the reference size. Unlike SE switching, this last solution has the advantage of being applicable to all features (i.e. also to Gabor features). This will be applied for instance in face detection using EGM (Subsection 5.1.2).

The Gabor features seem also to suffer from scale variations, but little can be done to compensate for this, except to filter the image with new filter bank coefficients, or to resample the original image before applying the same filter bank. Indeed, the only possibility to adapt the already calculated features would be to switch one filtered image with another, i.e. to switch the features $J_{p,q}$ (as defined in Eq. 4.31) along q.

4.5.2 Elastic matching using physical models

Once the best position of the rigid graph has been located, it is necessary to move individually the nodes around the position resulting from rigid matching to compensate for:

- face expression variations;
- pose / viewpoint variations;
- lighting direction variation that may displace features like edges.

In Figure 4-31, the node positions of an orthogonal graph (actually only a 4-connected neighborhood around node i) are depicted as black squares, whereas displaced nodes are depicted as blue squares. The distance between connected nodes m and n in the rigid graph is noted $\vec{d}_{r,m,n}$, whereas $\vec{d}_{t,m,n}$ represents the distance between two connected nodes n and i in the deformed case.

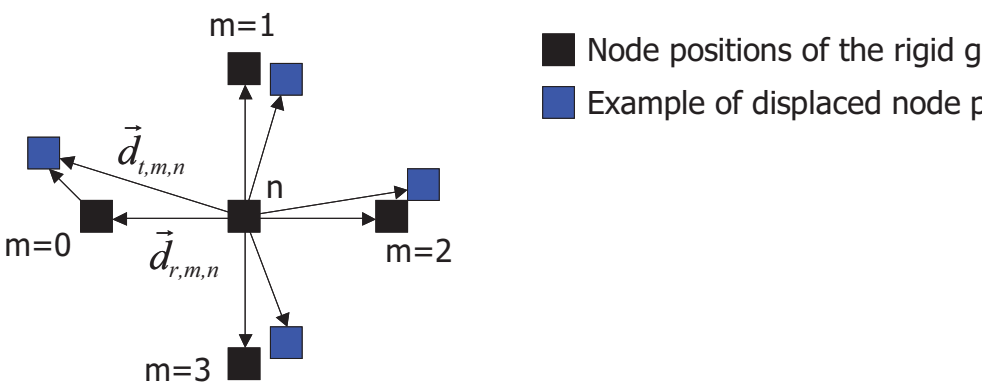


Figure 4-31: Elastic deformation penalty.

Physical deformation models will penalize node displacements off the position they reached during rigid matching. The penalty $d_{e,n}$ is added to the between-features distance calculated for each node *n*, and the resulting total distance is evaluated for several possible node displacements in a given neighborhood, until the best elastic correspondence between the reference and test graphs is found. It has to be noted that the graph matching is thus limited by the penalty function *and* the amount of displacement allowed during each iteration. This process can be described by the algorithm below:

```
iterate:
{
   for all nodes in the graph
   {
      for all displacements in the neighborhood
      {
         calculate the feature distance for this node and displacement
         calculate the distance penalty for this node
         get the total node score
         keep the best node displacement achieved so far
      }
   }
   calculate the feature distance for the whole graph
} until stop condition
```

Kotropoulos [4-39] proposed for $d_{e,n}$ the use of the norm of the difference vector between corresponding edges (Eq. 4.59), whereas Lades [4-40] is using for $d_{e,n}$ the square of the former distance (Eq. 4.60).

$$d_{e,n} = \sum_{m \in N} \left\| \vec{d}_{t,m,n} - \vec{d}_{r,m,n} \right\| \qquad (4.59)$$

$$d_{e,n} = \sum_{m \in N} \left\| \vec{d}_{t,m,n} - \vec{d}_{r,m,n} \right\|^2 \qquad (4.60)$$

These two penalties are independent of translations. In addition, the deformation penalty should be also independent of rotations and scaling, since we clearly searched for them in the iterative (Simplex downhill) matching step. This means that rotation and scaling of a rigid graph should *not* contribute to the deformation penalty. This is not the case of the metrics proposed by Lades and Kotropoulos.

Consequently, we define a new penalty, which is close to the potential energy of a contracted or elongated spring, and which depends only on the elongation (or contraction) of the distance between connected nodes during elastic matching, with respect to this distance at the end of rigid matching. Namely, the difference of the norms is used instead of the norm of the difference:

$$d_{e,m} = \sum_{m \in N} \left(\left\| \vec{d}_{t,m,n} \right\| - \left\| \vec{d}_{r,m,n} \right\| \right)^2 \qquad (4.61)$$

N is the ensemble comprised of the four-connected neighbors.

Note that although the distance between nodes in an orthonormal graph (i.e. $\left\| \vec{d}_r \right\|$) is constant over the entire graph, it is dependent on the current estimated scale parameter found during iterative rigid matching (Simplex downhill). An important result is therefore that homothetic scalings of the entire graph do not contribute to the deformation penalty.

It should be remembered that this arbitrary model is not at all related to the actual possible movements of face parts, which cannot be simplified as springs. Complex finite-element models would be necessary to closer model the elasticity of the skin and/or articulations of facial parts.

The deformation penalty is added to the squared Euclidean feature distance with an elasticity coefficient λ:

$$d_{total} = d_{features} + \lambda \cdot d_{elast} =$$
$$\sum_{n \in nodes} \left\| \vec{J}_{ref,n} - \vec{J}_{tst,n} \right\|^2 + \lambda \cdot \sum_{n \in nodes} \left(\sum_{m \in N} \left(\left\| \vec{d}_{t,m,n} \right\| - \left\| \vec{d}_{r,m,n} \right\| \right)^2 \right) \qquad (4.62)$$

where $J_{ref,n}$ and $J_{tst,n}$ are the feature vectors associated to node n, e.g. obtained from Eq. 4.31, Eq. 4.39, Eq. 4.42 or Eq. 4.43, in the reference and test images, respectively. The summation on n indicates that the squared Euclidean feature distance contributions are summed together, as it is done for the deformation penalties.

The total metrics d_{total} incorporating the deformation penalty is solely used to find the best correspondence between reference graphs, but it is not utilized for the classification (i.e. acceptation or rejection decision), which is based only on the between-features distance $d_{features}$.

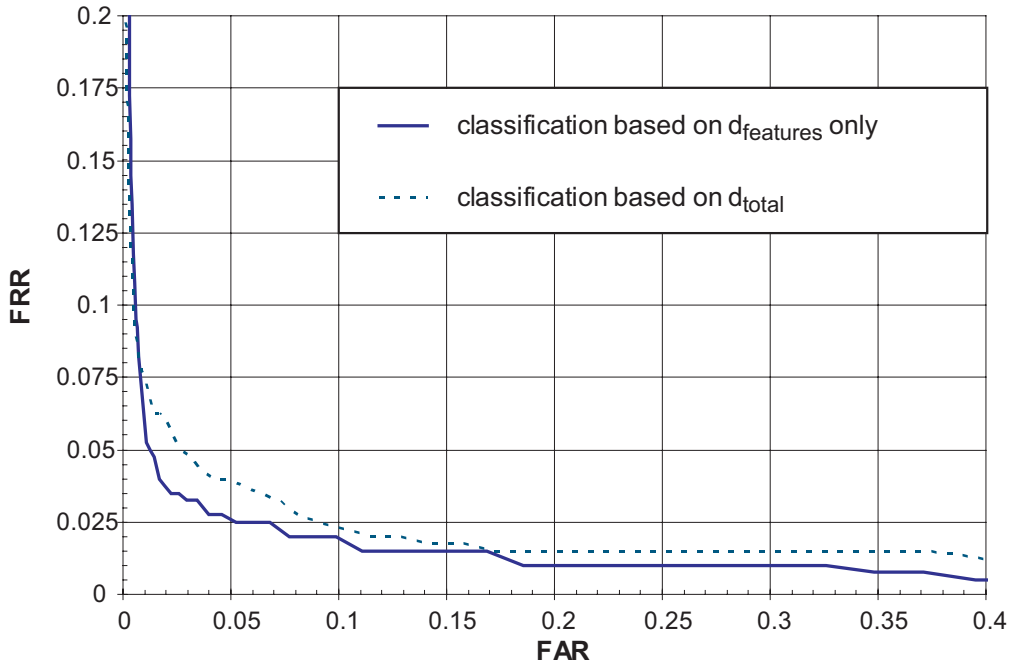


Figure 4-32: Effect of using the deformation for classification.
ROC of normalized morphological features on the XM2VTS database. Using d_{total} leads to worse results than using $d_{features}$ only.

Indeed, when comparing both approaches, $d_{features}$ gives better results, at least in case of normalized mathematical morphology (see Figure 4-32).

To simplify the calculation and to avoid square roots that are difficult to implement in hardware, we can stay with the squared versions of the Euclidean distance of features and of the deformation penalty, and take the absolute value of the differences instead of the square. The square of the distances values being monotonously increasing the result is not significantly modified (except for the contribution of the node penalty over the deformation penalty).

$$d_{total} = \sum_{n \in nodes} \left\| \vec{J}_{ref,n} - \vec{J}_{tst,n} \right\|^2 + \lambda \cdot \sum_{n \in nodes} \left(\sum_{m \in N} abs(\left\| \vec{d}_{t,m,n} \right\|^2 - \left\| \vec{d}_{r,m,n} \right\|^2) \right) \quad (4.63)$$

This simplification incurs a small loss in performance as can be observed on Figure 4-33, namely the EER is increased from around 3% to around 5%.

The iterative process of displacing nodes can be achieved in several ways as well. We can perform a fixed number of displacement or displace the nodes until the graph distance no longer decreases. The order

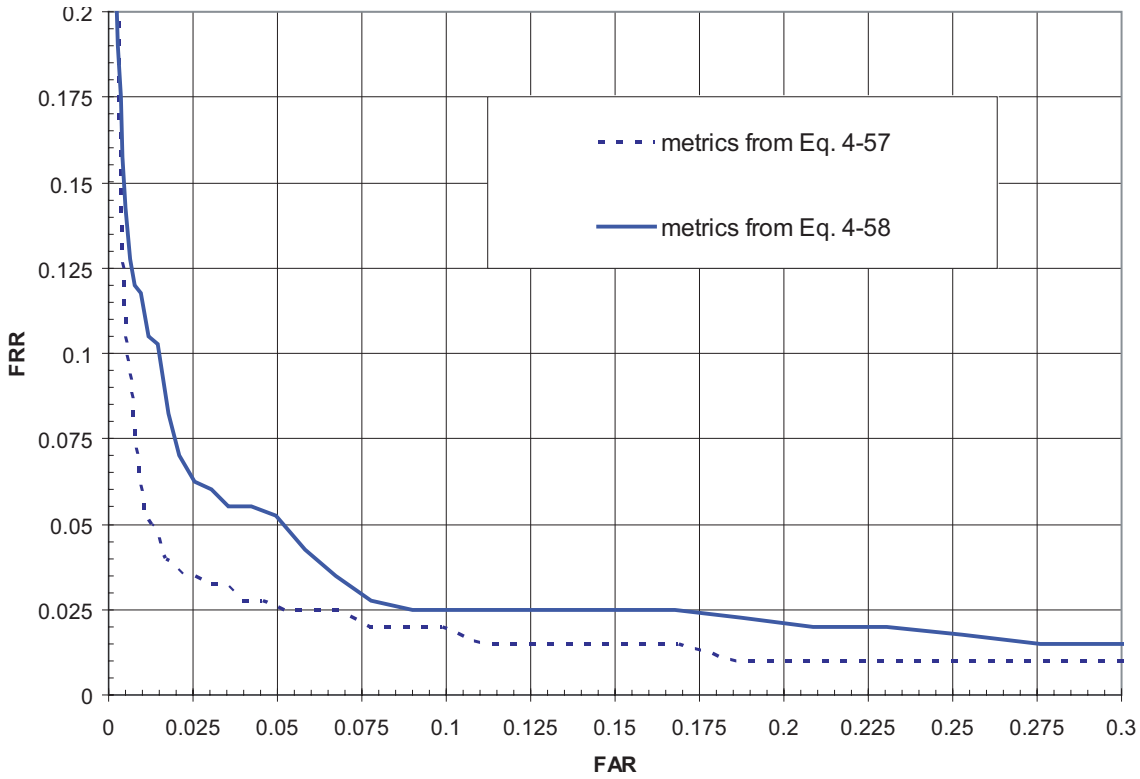


Figure 4-33: Effect of simplified metrics (Eq. 4.63) vs. metrics proposed in Eq. 4.62.

in which nodes are taken can be random or sequential (i.e. nodes taken in order). If we consider the displacement of all nodes in the graph in multiple steps, it can be shown that using the same node order in all steps will rapidly lead to a final distance that is clearly not the best match we can find. Since efficient random number generation is complex in terms of requested hardware resources (although generally easily available in software libraries, e.g. GNU libc), we will use a very simple "pseudo-random" number (PRN) generator (see Subsection 6.5.2). This PRN generator is very limited and could definitely not be used in cryptography applications for instance! However, it is sufficient for our purpose while not causing a large increase in hardware resources.

4.5.3 Elastic matching using statistical models

The elastic matching procedure described in the previous subsection is a largely simplified version of the general graph matching problem [4-44]. However the above method lacks the use of statistical information telling which deformations are more common than others. An example of statistically derived deformations is described by Metaxas [4-50] in conjunction with finite element descriptions.

Therefore we propose to express (or to approximate) any possible deformation with a basis of deformations. Deformations of a graph comprising N×N nodes would basically require 2×N^2 parameters (e.g. 128 for N=8). But, by using Karhunen-Loève techniques, it is possible to find the principal modes of deformations based on a manually or automatically labelled training set, and to express all deformations with the most expressive deformations, i.e. with a limited number of parameters. Logically, these deformation modes could be called “principal warps” or something similar (e.g. “eigenwarps”...).

From a set of parameters a new deformed graph would be created, features extracted, and a new graph distance calculated. Again, a Simplex Downhill method could be used to find what are the best parameters, and consequently, find a deformed graph. Moreover, it would be possible to implement the whole rigid and elastic process in one step by using M parameters including scale, orientation and translation. This original method was however not tested on a full database.

The EGM coprocessor architecture in Chapter 6 is based on rigid matching (i.e. simplex downhill succeeding coarse rigid matching), followed by elastic matching, but it would need minor modifications to perform this statistical matching procedure discussed above.

4.6 Classifiers and decision theory

The features obtained from a matched graph (i.e. morphological features or Gabor features after convergence of the iterative algorithm) form an *observation vector* that will be used by a classifier to infer the sample to a given trained class or to the rejection class. So far we reduced the observation vector to a single valued observation using either an Euclidean distance (Eq. 4.32) or a normed dot product (Eq. 4.33) between reference and test features. Correspondingly, the classification was reduced to a unidimensional two-classes (genuine vs. impostor) problem i.e. to a threshold selection.

An optimal decision theory based on a posteriori probabilities is the so-called Bayesian decision that classifies a test pattern in a class *c* if the a posteriori probability of this pattern belonging to class *c* is higher than the a posteriori probabilities of this pattern belonging to all other classes. This minimizes the classification error probability by assigning the same importance to all errors.

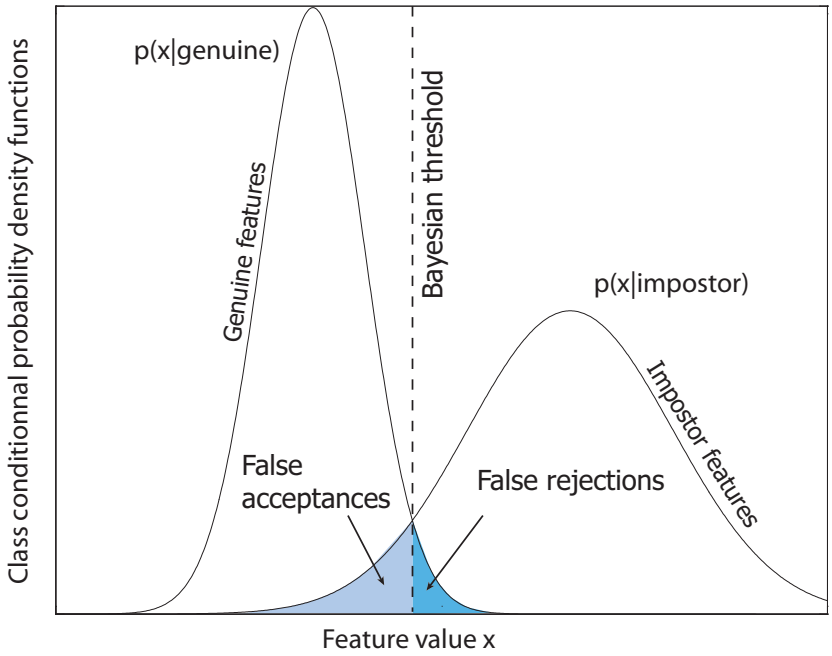


Figure 4-34: Two mono-dimensional distributions and the Bayesian decision threshold. The two distributions correspond to the probability density function (pdf) of an observed *x* to belong to the genuine or impostor class. The Bayesian threshold minimizes the total error (shaded areas) when the two pdfs are crossing. Note that although pdfs are gaussian in this example it is rarely the case with real features!

Starting from a mono-dimensional feature space (e.g. our matching distance), the attribution of a probe to one of two classes following the Bayesian decision corresponds to comparing the value to a simple threshold. It can be showed that the optimal threshold in the Bayesian sense is the one where the a posteriori probabilities are crossing, or equivalently, where the conditional probability density functions (pdf) are crossing if the a priori probabilities of the classes are equal. An example with a single feature is depicted in Figure 4-34. For all other thresholds the total error (i.e. the shaded area) is larger. The underlying assumption is that the form of both the genuine and the impostors pdfs are known (e.g. gaussian in Figure 4-34). This not the case for the matching distances: models for the genuine and impostor probability are not known!

Instead of resorting to a simple distance metrics, it would be possible to make a Bayesian decision in the full $M \times N$ dimensional space, where M is the number of features per node and N is the number of nodes. Using an Euclidean metrics is then optimal in the Bayesian sense only under a certain number of strong assumptions that most often do not hold as it will now be demonstrated.

As a starting point, a model is necessary for the likelihood function of the class c in the $M \times N$-dimensional feature space. The first strong assumption is that this likelihood function has a general multivariate normal distribution, so that the Bayesian classifier has a quadratic form [4-73]. This is obviously not the case here. A second assumption is that the classes are equiprobable, so that this quadratic form can be expressed in the form of a distance classifier. Finally, if and only if the covariance matrix is diagonal (i.e. features are independent), then the optimal distance reduces to an Euclidean distance. Based on this distance the probe would be attributed to the class with minimum distance or labelled as impostor when being closer to a general impostor class or further than a predefined threshold for the class with minimum distance. If and only if these restrictive hypotheses hold true, could an optimal decision be based on an Euclidean distance.

As a preliminary conclusion:

- the client features distributions are clearly neither normal, nor independent, and the Euclidean distance is then sub-optimal;
- the genuine matching distance distribution is impossible to estimate reliably due to the too small number of samples available;
- as a corollary, it is not possible to select an optimal threshold (in the Bayes sense).

Instead of finding parameters of a model, an approach using non-parametric clustering might prove useful, however the number of samples at our disposal is generally too low for these kinds of methods.

Alternatively, other distances (e.g. Manhattan norm) may be more adapted to the features distribution, but it was not possible to observe a consistent improvement. For all these reasons the Euclidean distance was maintained with normalized morphological features and dot product with Gabor features.

Since the class distribution of genuine matching distances cannot be estimated with only a few training samples, we might try to estimate the distribution of the impostor distances instead as it is larger. Then we could try to normalize both the genuine and impostor distributions based on this parameter estimation. Even if the class conditional impostor distributions cannot be estimated reliably, it is possible to normalize them by applying an arbitrary normalization. We can for instance use the mean value and variance of the impostor distance dis-

tribution d_i (obtained e.g. on the evaluation set defined in the Lausanne protocol, see Figure 2-1) of a class and apply the following normalization:

$$\tilde{d}_i = \frac{d_i - \overline{d}_i}{\sigma_i} \qquad (4.64)$$

where $\overline{d}_i$ is the impostor average for client i, and σ_i is the standard deviation of this impostor distribution. This normalization is called *z-norm* and often applied in speaker verification problems [4-9]. Another normalization called *tanh* (for its use of the hyperbolic tangent) gives the same results when using the impostor score estimates.

After this normalization, the distribution of evaluation impostor distances will have zero mean and unit variance as shown in Figure 4-35. The normalized genuine distance distributions corresponding to different classes do not necessarily share the same mean but nonetheless the selection of a global threshold (i.e. the same for all classes) is possible. This global threshold can then be varied to produce the ROC curves pictured so far. The decision is not optimal in the Bayes sense but at least is normalized for all classes (i.e. the threshold of normalized distances is almost class-independent).

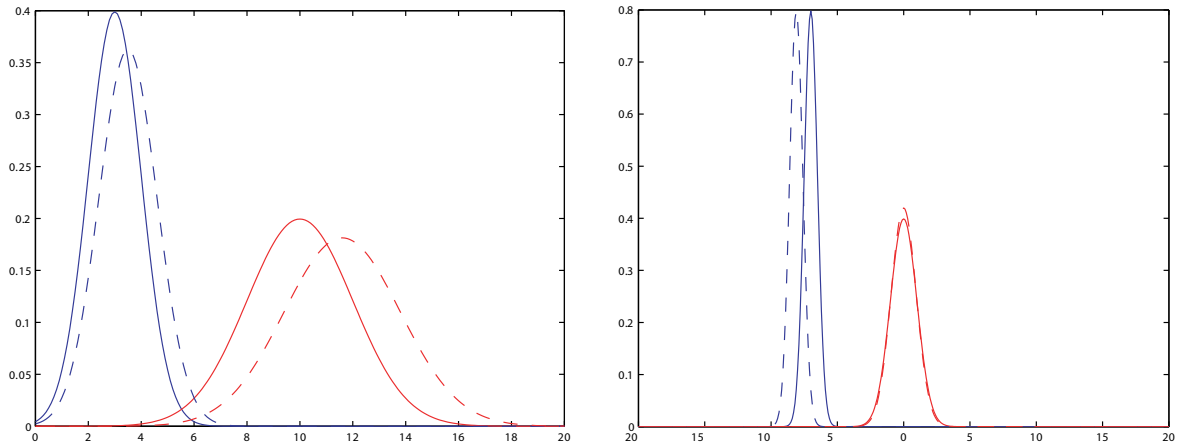


Figure 4-35: Genuine and impostor distances of two clients.
The two classes of genuine and impostor distances are assumed to have a gaussian distribution as represented here (a property that is not verified in practice) with different mean and variance parameters.The blue curves represents the genuine distance distribution and the red curves the impostor genuine distance distribution. The plain and dashed lines represent the distance distributions obtained when comparing test clients to two different genuine references. Curves on the left plot represent the situation before normalization, while the right plot represent the situation after normalization.

4.7 Performance evaluation

Although results of the different variants were already presented in the previous sections, they will be now recalled and compared to state-of-art results found in the literature.

4.7.1 XM2VTS database

Two competitions were organised based on this database in 2000 (at ICPR [4-49]) and 2003 (at AVBPA [4-51]). The database was made available to the competitors in advance so they could tune their system for this particular database.

In both cases, results were reported with the two Lausanne protocol configurations, and two face registration modes: fully automatic registration, and semi-automatic registration by manually locating the faces. Only the results of the fully automatic registration are considered here. The performance of a semi-automatic method applied to real applications is anyway always lower than the correct face detection performance! Additionally, only configuration II results were gathered in this thesis.

In the ICPR and AVBPA experiments, three thresholds were determined based on the *evaluation* set:

- T1 at FAR = 0;
- T2 at FAR = FRR;
- T3 at FRR = 0.

Then, these thresholds were used on the *test* set to determine the FAR and FRR at these points. Since results for genuine evaluation tests were not gathered in this thesis[17] it is difficult to directly compare performance of the ICPR and AVBPA algorithms to that of the algorithms described in this thesis. Nonetheless, it is possible to report the three points T1, T2 and T3 on our Receiver Operating Curves. Should the published algorithms be better, the three points would lie below our ROC.

In 2000, four institutions participated in the competition, namely the Aristotle University of Thessaloniki (AUT), the Dalle Molle Institute for Perceptual Artificial Intelligence (IDIAP), the University of Sydney,

17. Only the evaluation impostors were used to determine thresholds as explained in Section 4.6.

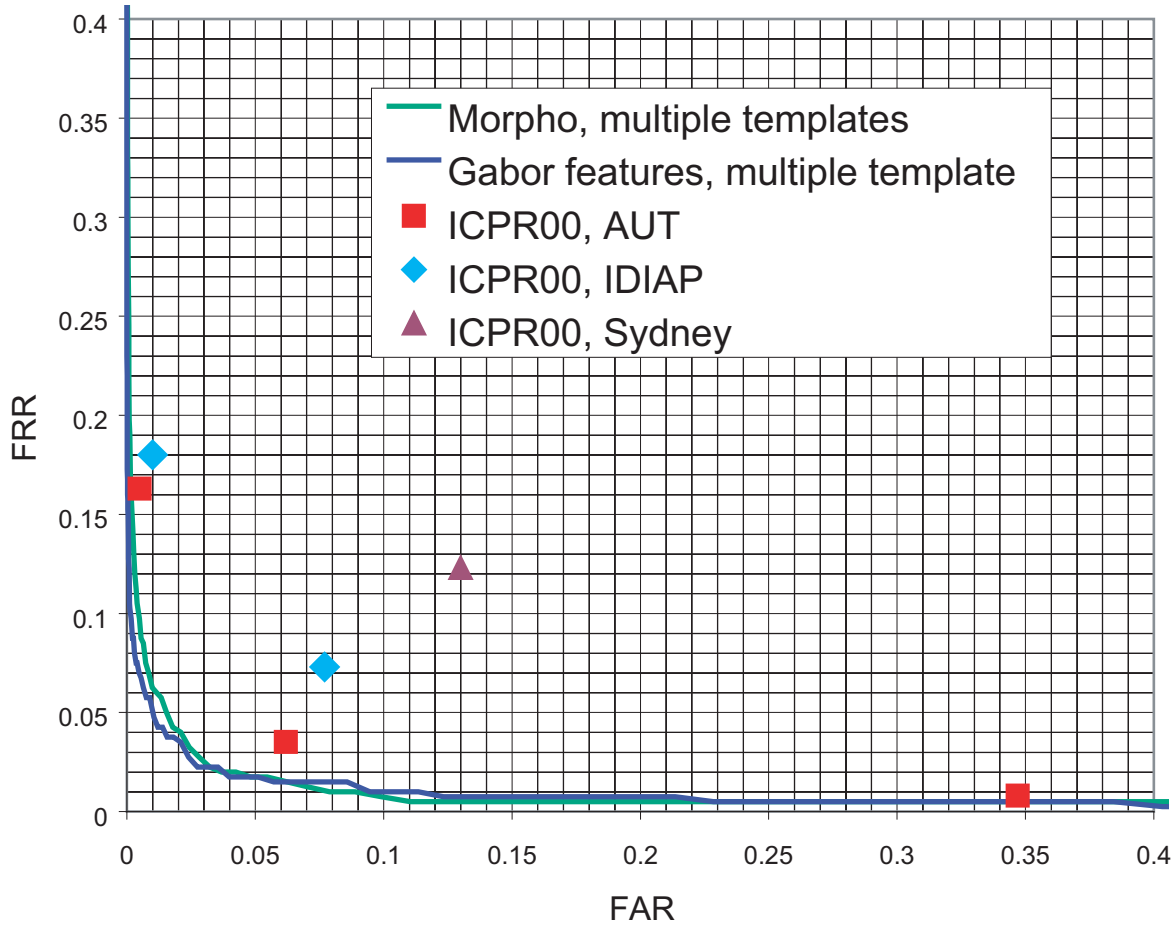


Figure 4-36: Performance comparison of our results (2004) against ICPR'00 results.

and the University of Surrey (UniS). Based on configuration II and on the fully automatic registration, the algorithms of only 3 institutions were tested (see [4-49] for details). Results are represented in Figure 4-36. It can be observed that both EGM with Gabor features and EGM with normalized morphological features as applied in this thesis outperform the three externally reported algorithms.

In 2003, six participants (5 academic and 1 commercial) entered into the competition for a total of 9 tested algorithms. Most of the competitors present in 2000 also participated to the 2003 test [4-51][18]. However, only four results are available for testing on Configuration II with fully automatic registration. This time only the results corresponding to T2 are available (although the text describes the three types of thresholds). Results are compared in Figure 4-37. It can be observed that three algorithms out of four outperform ours. Interestingly, the best performing algorithm (from IDIAP) uses DCTmod2 features in

18. The participation of the Universidad Politcnica de Valencia (UPV) is however new.

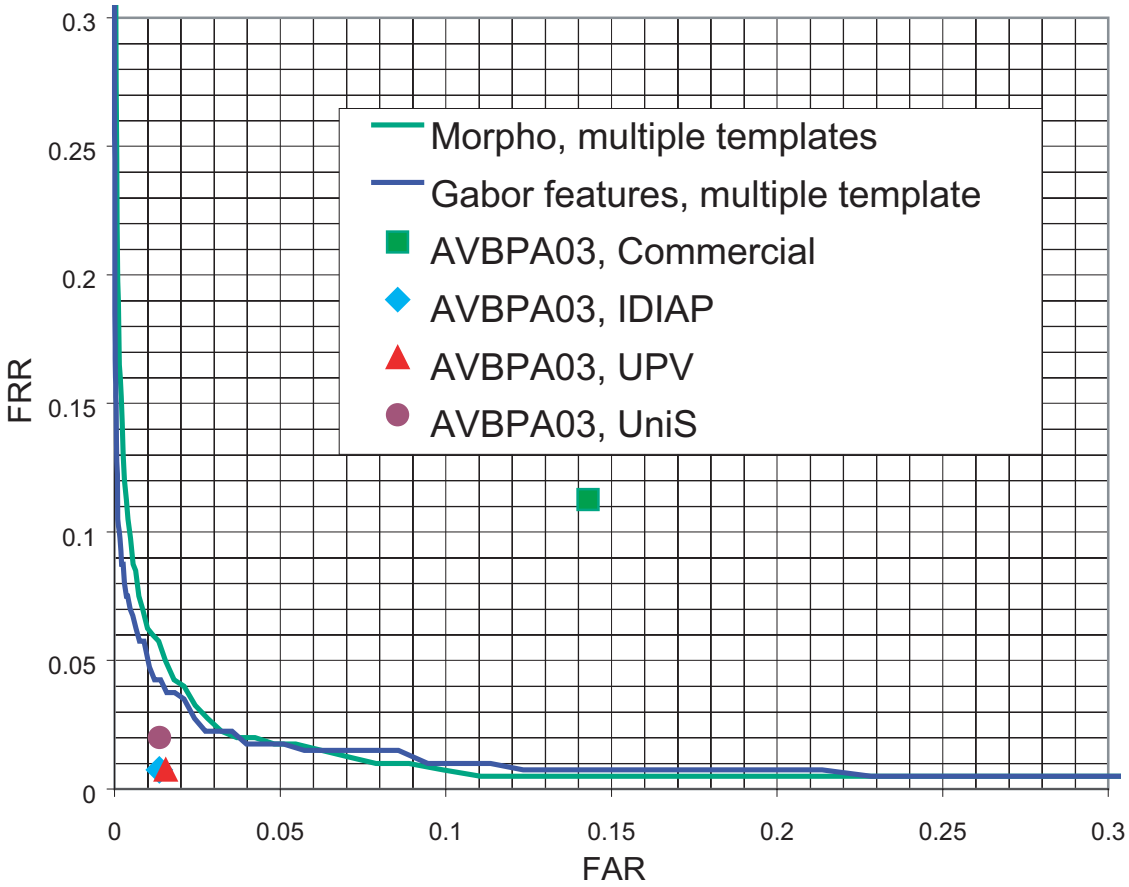


Figure 4-37: Performance comparison of our results (2004) against AVBPA'03 results.

combination with Hidden Markow Models (HMMs), although DCTmod2 seemed to perform poorly with our EGM algorithm (see Subsection 4.3.2).

4.7.2 Yale B database

This database was described in Subsection 2.7.2 on page 26. The tests reported throughout this thesis were performed on this database according to the setup defined in [4-27], i.e. by performing an identification in a closed-set and using manual registration. Results are recalled in Table 4-5.

It can be observed that the very computationnally-intensive methods based on illumination cones reported in [4-26] are the most effective as they achieve the lower error rates[19]. For instance the cones-cast method [4-27] which performs an outstanding 0.0% error rate on subset 4

19. Results are reported from [4-27], not from our own experiments.

Method	Error rate (%) vs. illuminant change		
	Subset 2	Subset 3	Subset 4
Cones-attached [4-27]	0.0	0.0	8.6
Cones-cast [4-27]	0.0	0.0	0.0
EGM Gabor (dot dist.)	0.0	0.0	10.7
EGM Gabor (eucl. dist.)	0.8	6.4	63.6
EGM DCTmod2	0.0	5.0	66.4
EGM norm. morpho	0.0	15.8	61.4

Table 4-5: Results on Yale B database (error rates in %).

when using a costly ray-tracing approach. Nevertheless, the EGM with Gabor features and dot-product distance are very close to the cones-attached method. Although the DCTmod2 features seem to perform well on this database according to [4-51], we observed from our experiments that their behaviour was not satisfying on the XM2VTS database.

4.8 Conclusion

Starting from the conclusion of the previous chapter – where it was observed that Elastic Graph Matching (EGM) seemed an interesting category of algorithms for face verification on mobile devices – this chapter essentially aimed at proposing and designing new computational methods, comparing them with existing ones, and selecting adequate parameters to implement EGM algorithm, including its hardware realization to be described in Chapter 6.

The first issue was the extraction of features robust to variations of illuminant (so-called photometric invariants). For this purpose, several candidates were evaluated. Gabor features showed to perform extremely well but at a very high computational cost. Alternatively, morphological features represented a reduce amount of computations but were extremely sensitive to variations of lighting. Model-free normalization of morphological features was shown to be a very interesting compromise with only a small computational overhead as compared to baseline morphology.

Regarding feature space reduction, it was shown that principal component analysis can be applied without node distinction but that it does

not guarantee an improvement in terms of recognition rates. Application of linear discriminant analysis seems interesting when considering the two-classes problem, i.e. a transform per client. However, experiments on the XM2VTS showed no improvement, probably due to the fact that the training set of this database is too small.

Iterative methods based on the Simplex-Downhill were proposed in order to find the best rigid graph matching including rotation and scale. Finally, an elastic metric not penalizing rotations and scaling was also introduced.

References

[4-1] Y. Adini, Y. Moses, and S. Ullman, "Face Recognition: The Problem of Compensating for Changes in Illumination Direction," *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. 19, No. 7, July 1997, pp. 721-732.

[4-2] O. Ayinde, and Y.-H. Yang, "Face Recognition Approach Based on Rank Correlation of Gabor-filtered Images," *Pattern Recognition*, Vol. 35, 2002, pp. 1275-1289.

[4-3] Z. Bai, J. Demmel, J. Dongarra, A. Ruhe, and H. van der Vorst, "Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide," *Society for Industrial and Applied Mathematics (SIAM)*, Philadelphia, NJ, 2000.

[4-4] K. Barnard, L. Martin, B. Funt, and A. Coath , "Data for Colour Research," *Color Research and Application*, Vol. 27, No. 3, 2000, pp. 148-152.

[4-5] K. Barnard, V. Cardei, and B. Funt, "A Comparison of Computational Color Constancy Algorithms – Part I: Methodology and Experiments with Synthesized Data," *IEEE Trans. Image Processing*, Vol. 11, No. 9, Sep. 2002, pp. 972-984.

[4-6] K. Barnard, L. Martin, A. Coath, and B. Funt, "A Comparison of Computational Color Constancy Algorithms – Part II: Experiments with Image Data," *IEEE Trans. Image Processing*, Vol. 11, No. 9, Sep. 2002, pp. 985-996.

[4-7] R. Basri, and D. Jacobs , "Lambertian Reflectance and Linear Subspaces," in *Proc. 8th IEEE Int'l Conf. on Computer Vision (ICCV'01)*, Vancouver, Canada, July 7-14, 2001, Vol. 2, pp. 383-390.

[4-8] P. N. Belhumeur, D. J. Kriegman , "What is the Set of Images of an Object Under All Possible Lighting Conditions ?," in *Proc. Conf. Computer Vision and Pattern Recognition (CVPR'96)*, San Francisco, CA, June 18-20, 1996, pp. 270-277.

[4-9] S. Bengio, J. Mariéthoz, and S. Marcel, "Evaluation of biometric technology on XM2VTS," *Technical Report IDIAP-RR 01-21*, IDIAP, Martigny, Switzerland, 2001.

[4-10] J. Bigün, and J. M. H. du Buf, "N-folded Symmetries by Complex Moments in Gabor Space and Their Application to Unsupervised Texture Segmentation," *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. 16, No. 1, Jan. 1994, pp. 80-87.

[4-11] N. Blanc, S. Lauxtermann, P. Seitz, S. Tanner, A. Heubi, M. Ansorge, F. Pellandini, E. Doering, "Digital Low-Power Camera on a Chip," in *Proc. COST 254 Int.l Workshop on Intelligent Communication Technologies and Applications, with Emphasis on Mobile Communication*, Neuchâtel, Switzerland, May 5-7, 1999, pp. 80-83.

[4-12] V. Blanz, and T. Vetter, "Face Recognition Based on Fitting a 3D Morphable Model", *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. 25, No. 9, Sep. 2003, pp. 1063-1074.

[4-13] A. C. Bovik, M. Clark, and W. S. Geisler, "Multichannel Texture Analysis Using Localized Spatial Filters," *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. 12, No. 1, Jan. 1990, pp. 55-61.

[4-14] C. S. Burrus, R. A. Gopinath, and H. Guo, "Introduction to Wavelets and Wavelet Transforms: A Primer", *Prentice Hall*, Upper Saddle River, NJ, 1998.

[4-15] H. F. Chen, P. N. Belhumeur, and D. W. Jacobs , "In Search of Illumination Invariants," in *Proc. Conf. Computer Vision and Pattern Recognition (CVPR'00)*, Hilton Head, SC, June 13-15, 2000, Vol. 1, pp. 1254-1261.

[4-16] K. Chung, S. C. Kee, S. R. Kim, "Face Recognition Using Principal Component Analysis of Gabor Filter Responses," in *Proc. Int'l Workshop on Recognition, Analysis, and*

Tracking of Faces and Gestures in Real-Time Systems, Corfu, Greece, Sep. 26 - 27, 1999, pp. 53-57.

[4-17] B. Duc, S. Fischer, and J. Bigün, "Face Authentication with Gabor Information on Deformable Graphs," *IEEE Image Processing*, Vol. 8, No. 4, Apr. 1999, pp. 504-516.

[4-18] J. Serra, "Introduction to Mathematical Morphology," *Computer Vision, Graphics, and Image Processing*, Vol. 35, No. 3, Sept. 1986, pp. 283-305.

[4-19] S. Eickeler, S. Muller, and G. Rigoll , "Recognition of JPEG Compressed Face Images Based on Statistical Methods," *Image and Vision Computing*, Vol. 18, No. 4, 2000, pp. 279-287.

[4-20] H. G. Feichtinger, and T. Strohmer, Editors, "Gabor Analysis and Algorithms: Theory and Applications," *Birkhäuser*, Boston, MA, 1998.

[4-21] H. G. Feichtinger and T. Strohmer, Editors , "Advances in Gabor Analysis," *Birkhäuser*, Boston, MA, 2003.

[4-22] G. D. Finlayson, M. S. Drew, and B. V. Funt , "Color Constancy: Generalized Diagonal Transforms Suffice," *J. Optical Society of America A*, Vol. 10, No. 11, Nov. 1994, pp. 3011-3019.

[4-23] G. D. Finlayson, M. S. Drew, and B. V. Funt , "Spectral sharpening: sensor transformations for improved color constancy," *J. Optical Society of America A*, Vol. 11, No. 5, May 1994, pp. 1553-1563.

[4-24] K. Fukunaga, "Introduction to Statistical Pattern Recognition," 2nd Ed., *Academic Press*, San Diego, CA, 1990.

[4-25] B. V. Funt and G. D. Finlayson , "Color constant color indexing," *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. 17, No. 5, May 1995, pp. 522-529.

[4-26] A. S. Georghiades, D. J. Kriegman, and P. N. Belhumeur , "Illumination Cones for Recognition Under Variable Lighting: Faces," in *Proc. Conf. Computer Vision and Pattern Recognition (CVPR'98)*, Santa Barbara, CA, June 23-25, 1998, pp. 52-58.

[4-27] A.S. Georghiades, P.N. Belhumeur, and D.J. Kriegman, "From Few to Many: Illumination Cone Models for Face Recognition under Variable Lighting and Pose," *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. 23, No. 6, 2001, pp. 643-660.

[4-28] T. Gevers, "Adaptive Image Segmentation by Combining Photometric Invariant Region and Edge Information," *IEEE Trans. Pattern Anal. Machine Intell.*, Vol 24, No. 6, June 2002, pp. 848-852.

[4-29] J. Goutsias, L. Vincent, and D. S. Bloomberg, "Mathematical Morphology and its Applications to Signal Processing," *Series "Computational Imaging and Vision,"* Kluwer, Dordrecht, The Netherlands, 2000.

[4-30] R. Gross, and V. Brajovic , "An Image Preprocessing Algorithm for Illuminant Invariant Face Recognition," in *Proc. 4th Int'l Conf. on Audio- and Video-based Biometric Person Authentication (AVBPA'03)*, Guildford, UK, June 9-11, 2003, pp. 10-18.

[4-31] G. Healey, and Q.-T. Luong, "Color in Computer Vision: Recent Progress," *in "Handbook of Pattern Recognition and Computer Vision,"* 2nd edition, C. H. Chen, L. F. Pau and P. S. P. Wang Eds., World Scientific Publishing Company, 1998, pp. 283-312.

[4-32] P. T. Jackway, and M. Deriche, "Scale-Space Properties of the Multiscale Morphological Dilation-Erosion," *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. 18, No. 1, Jan. 1996, pp. 38-51.

[4-33] A. K. Jain, "Fundamentals of Digital Image Processing," *Prentice & Hall*, Information and System Science Series, Englewood Cliffs, NJ, 1989.

[4-34] A. K. Jain, and F. Farrokhnia, "Unsupervised Texture Segmentation Using Gabor Filters," *Pattern Recognition*, Vol. 24, No. 12, 1991, pp. 1167-1186.

[4-35] A. K. Jain, S. Prabhakar, L. Hong, and S. Pankanti, "Filterbank-based Fingerprint Matching," *IEEE Trans. Image Processing*, Vol. 9, No. 5, 2000, pp. 846-859.

[4-36] I. T. Jolliffe, "Principal Component Analysis," *Springer Verlag*, Springer series in Statistics, New York, 1986.

[4-37] J. P. Jones, and L. A. Palmer, "An Evaluation of the Two-Dimensional Gabor Filter Model of Simple Receptive Fields in Cat Striate Cortex," *Journal of Neurophyisiology*, Vol. 58, No. 6, Dec. 1987, pp. 1233-1258.

[4-38] C. Kotropoulos, A. Tefas, and I. Pitas , "Frontal Face Authentication Using Discriminating Grids with Morphological Feature Vectors," *IEEE Trans. Multimedia*, Vol. 2, No. 1, Mar. 2000, pp. 14-26.

[4-39] C. Kotropoulos, A. Tefas and I. Pitas, "Morphological Elastic Graph Matching applied to frontal face authentication under well-controlled and real conditions," *Pattern Recognition*, Vol. 33, No. 12, 2000, pp. 1935-1947.

[4-40] M. Lades, J. C. Vorbrügen, J. Buhmann, J. Lange, C. von der Malsburg, R. P. Würtz and W. Konen, "Distortion Invariant Object Recognition in the Dynamic Link Architecture," *IEEE Trans. Comp.*, Vol. 42, No. 3, Mar. 1993.

[4-41] M. Lades, "Face Recognition Technology," *in "Handbook of Pattern Recognition and Computer Vision,"* 2nd edition, Eds. C. H. Chen, L. F. Pau and P. S. P. Wang, World Scientific Publishing Company, 1998, pp. 667-683.

[4-42] J. C. Lagarias, J. A. Reeds, M. H. Wright, and P. E. Wright, "Convergence Properties of the Nelder-Mead Simplex Method in Low Dimensions," *Journal of Optimization*, Society for Industrial and Applied Mathematics (SIAM), Vol. 9, No. 1, 1998, pp. 112-147.

[4-43] R.W.-K. Lam, and C. K. Li, "A Fast Algorithm to Morphological Operations With Flat Structuring Element," *IEEE Circuits and Systems - II: Analog and Digital Signal Processing*, Vol. 45, No. 3, Mar. 1998, pp. 387-391.

[4-44] R.S.T. Lee, and J.N.K. Liu, "Invariant Object Recognition Based on Elastic Graph Matching", *IOS Press*, Amsterdam, The Netherlands, 2003.

[4-45] K.-C. Lee, J. Ho, and D. Kriegman , "Nine Points of Light: Acquiring Subspaces for Face Recognition under Variable Lighting," in *Proc. Computer Vision and Patter Recognition (CVPR'01)*, Kauai, Hawaii, Dec. 8-14, 2001, pp. 519-526.

[4-46] T. S. Lee , "Image Representation Using 2D Gabor Wavelets," *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. 18, No. 10, Oct. 1996, pp. 959-971.

[4-47] C. Liu and H. Wechsler, "Gabor Feature Based Classification Using the Enhanced Fisher Linear Discriminant Model for Face Recognition," *IEEE Trans. Image Processing*, Vol. 11, No. 4, Apr. 2002, pp. 467-476.

[4-48] B. Martinkauppi, "Face Colour Under Varying Illumination - Analysis and Applications," *Academic Dissertation*, University of Oulu, Finland, 2002.

[4-49] J. Matas, M. Hamouz, K. Jonsson, J. Kittler, Y. Li, C. Kotropoulos, A. Tefas, I. Pitas, T. Tan, H. Yan, F. Smeraldi, N. Capdevielle, W. Gerstner, Y. Abdeljaoued, J. Bigün, S. Ben-Yacoub, and E. Mayoraz , "Comparison of Face Verification Results on the

XM2VTS Database," in *Proc. Int'l Conf. Pattern Recognition (ICPR'00)*, Barcelona, Spain, Sept. 3-8, 2000, Vol. 4, pp. 4858-4863.

[4-50] D. N. Metaxas, "Physics-based deformable models: Applications to ComputerVision, Graphics and Medical Imaging," *Kluwer Academic Publishers*, Dordrecht, The Netherlands, 1997.

[4-51] K. Messer, J. Kittler, M. Sadeghi, S. Marcel, C. Marcel, S. Bengio, F. Cardinaux, C. Sanderson, J. Czyz, L. Vandendorpe, S. Srisuk, M. Petrou, W. Kurutach, A. Kadyrov, R. Paredes, B. Kepenekci, F. B. Tek, G. B. Akar, F. Deravi, and N. Mavity , "Face Verification Competition on the XM2VTS Database," in *Proc. 4th Int'l Conf. on Audio- and Video-based Biometric Person Authentication (AVBPA'03)*, Guildford, UK, June 9-11, 2003, pp. 964-974.

[4-52] S. K. Nayar, K. Ikeuchi, and T. Kanade, "Surface Reflection: Physical and Geometrical Perspectives," *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. 13, No. 7, July 1991, pp. 611-634.

[4-53] I. Ng, T. Tan, and J. Kittler, "On Local Linear Transform and Gabor Filter Representation of Texture," in *Proc. Int'l Conf. Pattern Recognition (ICPR'92)*, The Hague, Aug. 30-Sept. 3, 1992, pp. 627–631.

[4-54] Z. Pan, and H. Bolouri, "High Speed Face Recognition Based on Discrete Cosine Transforms and Neural Networks," *Technical Report*, University of Hertfordshire, UK, 1999.

[4-55] A. Papoulis, and S. U. Pillai, "Probability, Random Variables and Stochastic Processes," *Series in Electrical and Computer Engineering*, 4th Edition, McGraw-Hill, New York 2002.

[4-56] B. Phong , "Illumination for computer generated pictures," *Commun. ACM*, Vol. 18, 1975, pp. 311-317.

[4-57] I. Pitas, "Fast Algorithm for Running Ordering and Max/Min Calculation," *IEEE Circuits and Systems*, Vol. 36, No. 6, June 1989, pp. 795-804.

[4-58] D. A. Pollen, and S. F. Ronner, "Visual Cortical Neurons as Localized Spatial Frequency Filters," *IEEE Systems, Man, and Cybernetics*, Vol. 13, No. 5, Sep.-Oct. 1983, pp. 907-915.

[4-59] C. A. Poynton, "Poynton's Colour FAQ," http://www.inforamp.net/~poynton/Poyntoncolor.html.

[4-60] T. Randen, and J.H. Husøy, "Filtering for Texture Classification: A Comparative Study," *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. 21, No. 4, Apr. 1999, pp. 291-310.

[4-61] A. Rizzi, C. Gatta, and D. Marini, "Color correction between gray world and white patch", in *Proc. SPIE, Human Vision and Electronic Imaging VII*, Vol. 4662, , pp. 367-375.

[4-62] C. Sanderson, and K. Paliwal , "Polynomial Features for Robust Face Authentication," in *Proc. IEEE Int'l Conf. on Image Proc. (ICIP'02)*, Rochester, NY, 2002, Vol. 3, pp. 997-1000.

[4-63] C. Sanderson, and S. Bengio , "Robust Features for Frontal Face Authentication in Difficult Image Conditions," in *Proc. 4th Int'l Conf. on Audio- and Video-based Biometric Person Authentication (AVBPA'03)*, Guildford, UK, June 9-11, 2003, pp. 495-504.

[4-64] S. A. Shafer, "Using Color to Separate Reflection Components," *COLOR: research and application*, John Wiley & Sons Inc., Hoboken, NJ, Vol. 10, No. 4, 1985, pp. 210-218.

[4-65] S. Shan, W. Gao, B. Cao, and D. Zhao , "Illumination Normalization for Robust Face Recognition Against Varying Lighting Conditions," in *Proc. IEEE Int'l Workshop on Analysis and Modelling of Faces and Gestures (AMFG'03)*, Nice, France, Oct. 17, 2003, pp. 157-164.

[4-66] A. Shashua, "On Photometric Issues in 3D Visual Recognition from a Single 2D Image," *Int'l Journal of Computer Vision (IJCV)*, Vol. 21, No. 1/2, 1997, pp. 99-122.

[4-67] A. Shashua, and T. Riklin-Raviv , "The Quotient Image: Class-Based Re-Rendering and Recognition with Varying Illuminations," *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. 23, No. 2, Feb. 2001, pp. 129-139.

[4-68] J. Short, J. Kittler, and K. Messer , "A Comparison of Photometric Normalization Algorithms for Face Verification," in *Proc. 6th IEEE Int'l Conf. on Automatic Face and Gesture Recognition (FGR'04)*, Seoul, Korea, May 17-19, 2004, pp. 254-260.

[4-69] P. Soille, and H. Talbot, "Directional Morphological Filtering," *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. 23, No. 11, Nov. 2001, pp. 1313-1329.

[4-70] P. Stadelmann, J.-L. Nagel, M. Ansorge, F. Pellandini, "A Multiscale Morphological Coprocessor for Low-Power Face Authentication," in *Proc. XIth European Signal Processing Conference (EUSIPCO 2002)*, Toulouse, France, Sept. 03-06, 2002, Vol. 3, pp. 581-584.

[4-71] A. Tefas, C. Kotropoulos, and I. Pitas, "Using Support Vector Machines to Enhance the Performance of Elastic Graph Matching for Frontal Face Authentication," *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. 23, No. 7, July 1997, pp. 735-746.

[4-72] A. Tefas, C. Kotropoulos, and I. Pitas, "Face Verification Using Elastic Graph Matching Base on Morphological Signal Decomposition," *Signal Processing*, Elsevier, No. 82, 2002, pp. 833-851.

[4-73] S. Theodoris, and K. Koutroumbas, "Pattern Recognition," *Academic Press*, London, 1998.

[4-74] M. Tuceryan and A. K. Jain, "Texture Analysis," *in "Handbook Pattern Recognition and Computer Vision,"* C.H. Chen, L.F. Pau, and P.S.P. Wang, eds., Singapore, 1993, pp. 235–276.

[4-75] S. Umeyama, and G. Godin , "Separation of Diffuse and Specular Components of Surface Reflection by Use of Polarization and Statistical Analysis of Images," *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. 26, No. 5, May 2004, pp. 639-647.

[4-76] L. Wiskott, J.-M. Fellous, N. Krüger, and C. von der Malsburg, "Face Recognition and Gender Determination," in *Proc. Int'l Workshop on Automatic Face- and Gesture-Recognition (FG'95)*, Zürich, Switzerland, 1995, pp. 92-97.

[4-77] J. Yang, and J.-Y. Yang, "Why can LDA be performed in PCA transformed space?," *Pattern Recognition*, Vol. 36, 2003, pp. 563-566.

[4-78] J. Yao, P. Krolak, and C. Steele , "The Generalized Gabor Transform," *Image Processing*, Vol. 4, No. 7, July 1995, pp. 978-988.

[4-79] A. L. Yuille, "Deformable Templates for Face Recognition," *Journal of Cognitive Neuroscience*, Vol. 3, No. 1, 1991, pp. 59-70.

Chapter 5

Further Use of Elastic Graph Matching

Elastic Graph Matching is a general framework that can be applied to face authentication but also to other pattern registration and recognition problems. In this chapter, the EGM framework will first be applied to the face detection problem, which is in fact very similar to the face recognition paradigm (Section 5.1). Then, the same EGM framework will be applied to fingerprint matching, which is a very interesting category of applications asking for elastic deformations. New and original concepts will be introduced to allow matching of partial graphs (Section 5.2).

5.1 Face detection

The application of face detection in still images or video sequences is multifold. For instance, it can help:

- detecting if a face is at all present in the image, in order to launch more complex operations when required;
- accelerating face recognition techniques by allowing them to focus only on most interesting regions of a scene;
- automatically pointing a camera in the direction of the face of a person using pan and tilt, e.g. in surveillance applications (possibly in conjunction with automatic identification);
- counting the number of faces appearing in the scene.

Since it is a key function needed in many face-related applications, it was logically the subject of numerous publications. A thorough review of state-of-art face detection algorithms can be found e.g. in [5-14].

It shall be noted that the face authentication algorithm described in Chapter 4 does not rely on the precise detection of the face position, rotation and scale. Indeed, the face detection process is somewhat embed-

ded in the recognition process, since the image is scanned in order to find the best matching position. This is of course a key to its robustness, but it has also a severe impact on its computational complexity. Using a preliminary fast face detection before the iterative matching methods get applied (see Subsection 4.5.1), would allow to reduce the number of coarse rigid matching steps.

Of course, if a rejection decision occurs, it might be equally well due to a real impostor attempt or to a false face detection (i.e. increasing the FRR but not the FAR), or to other aspects like inappropriate scale. Consequently, all rejections should be processed subsequently by a full EGM step to keep the FRR at a tolerable level (Figure 5-1). Since more genuine attempts are expected than impostor attempts, overall computation savings can be expected.

Moreover, it shall be noted that progressing along the upper horizontal path in Figure 5-1 (i.e. converging rapidly from the input image to a correctly placed graph leading to a genuine client acceptance) does not influence the genuine acceptance decision, as long as the score threshold separating a genuine client from an impostor is not modified. Indeed, if the lower branch, labelled as “rejection”, was removed, using a reduced EGM method would only increase the false rejection rate, without modifying the false acceptance rate. The aim of this lower branch is thus to reduce the false rejection rate, as it does not influence the false acceptance rate, as compared to the standard EGM matching described in Chapter 4.

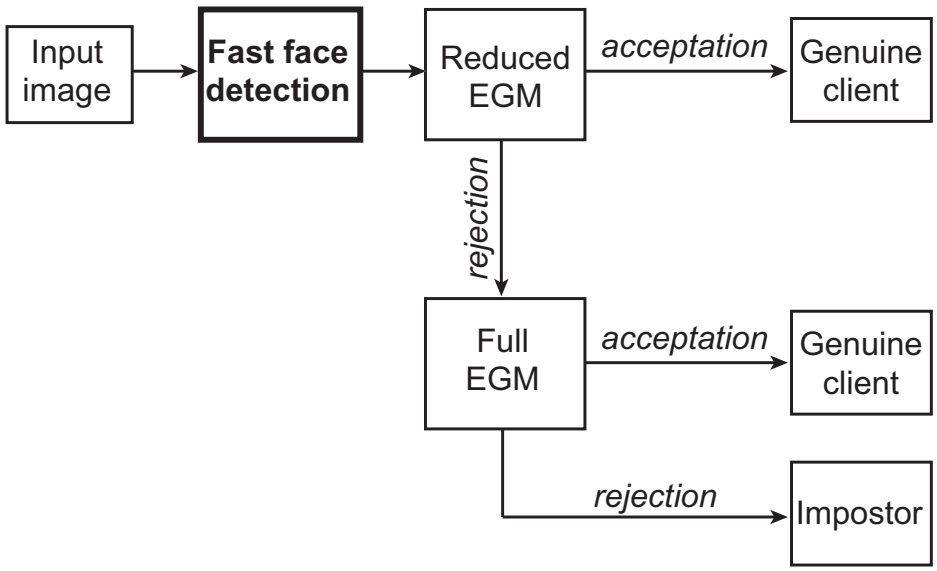


Figure 5-1: Insertion of a fast face detection.

This face detection should not necessarily be very precise in terms of face position, orientation and scale, but should be significantly faster than the face registration based on full graph matching. A typical algorithm category corresponding to this criterion is color-based detection. In Subsection 5.1.1, a very simple implementation of this algorithm will be described together with a possible hardware implementation.

Nevertheless, a precise face registration is sometimes required for instance when the enrolment process of a face authentication system is done automatically and in an unattended manner (i.e. without the presence and control of a human operator). In this case, precise location, rotation and scale are necessary, e.g. to efficiently place a reference grid on the face to extract a client template. As it was discussed in Chapter 3, a detection method relying on face features only will generally fail in complex situations (e.g. eyes hidden behind sun glasses). Consequently holistic approaches are more robust for this purpose. In Subsection 5.1.2, a face detection method based on graph matching that can be combined to the simple color-based method will be described, allowing to reuse the same hardware as in face authentication.

5.1.1 Color-based method

Color-based face detection methods consist in assigning a per-pixel probability of this pixel belonging to a skin region using color information, which is subsequently segmented. The resulting face region can be further analysed to detect facial features.

The only model involved in this method is that of the skin color. It is therefore a very simple method, furthermore much less computationally intensive than model-based methods relying on neural-networks, hidden-markov models, etc. A strong limitation comes however from the influence of the illuminant spectral density as was already discussed in Subsection 4.2.1. Indeed, under certain illuminant color, it may be difficult to distinguish skin regions from otherwise different color patches.

Before proceeding to the description of a color-base face detection method, it is important to recall that the face authentication method described in Chapter 4 is based on grayscale images (luma or luminance). This means that to integrate the face detection and authentication tasks on the same device, a color sensor would be necessary, with a subsequent conversion from color to grayscale images for the authentication performed after the face detection[1].

a) A color-based face detection algorithm

A very simple yet effective method of assigning a per-pixel probability of a pixel to belong to a skin patch was proposed by Soriano et al. [5-13]. The red, green and blue components of a pixel $\{R,G,B\}$ are transformed into normalized $\{r,g,b\}$ components using the following relationships:

$$r = \frac{R}{R+G+B},\ g = \frac{G}{R+G+B},\ b = \frac{B}{R+G+B} \qquad (5.1)$$

The purpose of the division by the sum of the components is to be as much as possible independent of the illumination *intensity*. Note that the $R+G+B$ sum does however represent neither the luminance nor the luma, but a weak approximation of them. Furthermore variations of illuminant color will still result in variations of r, g and b. Finally note that in practice only r and g are used, while b is discarded. These two components are subsequently used to index a two-dimensional lookup table (LUT) containing the probability of this color to belong to a skin patch, which is called the *skin locus*.

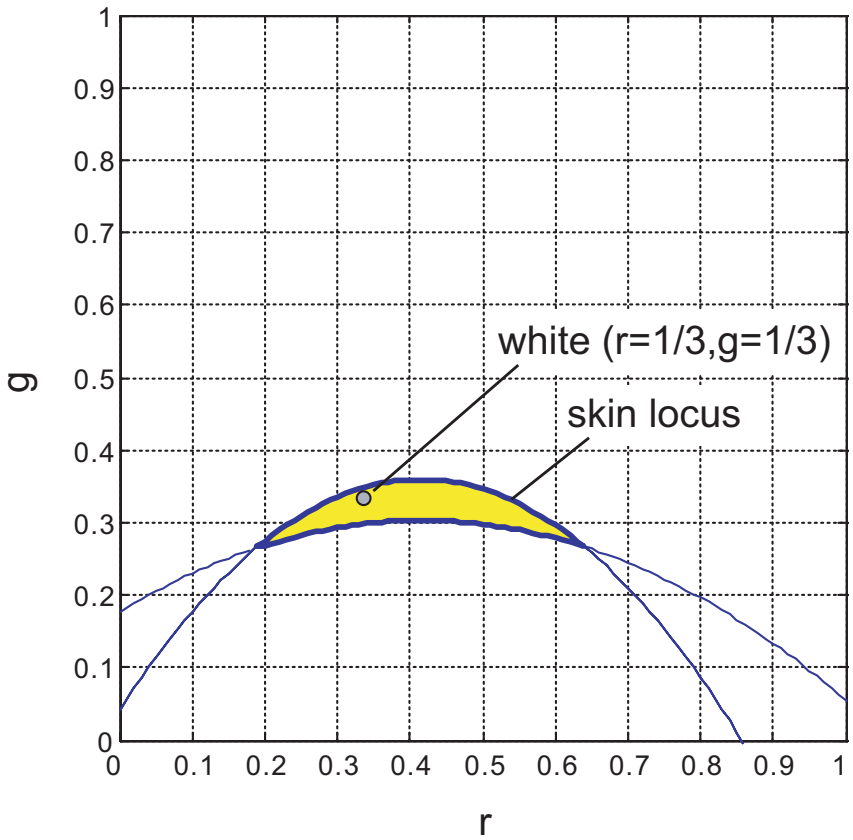


Figure 5-2: Skin color locus in the *r*-*g* space.
See [5-13] for parameters of the parabolas defining the skin locus.

1. Alternatively, the green component can be used as the “grayscale” information.

Soriano et al. showed that this skin locus can be modelled as the intersection of two parabolas (see Figure 5-2). In the following discussion, we will thus refer to the "Soriano look-up table" (or Soriano LUT). Additionally, Soriano et al. propose to dynamically modify the locus using a so-called *back-propagation* algorithm, to incorporate dynamic variations of the illuminant.

The information coming from a low-cost color CMOS or CCD image sensor is usually organized according to the Bayer pattern (Figure 5-3a). Before calculating the *r* and *g* components, it is thus necessary to perform a *demosaic* operation on this array of values. It was shown that a very simple down-sampling[2] leading to a 1/4 image (Figure 5-3b) is sufficient for the color-based face detection. Even though the visual quality and resolution of the obtained *R*,*G*,*B* image might not be satisfactory for applications where a visualization is necessary, it is largely sufficient for the purpose of face detection [5-3].

Using the parameters described by Soriano et al., a correct detection of around 99% is obtained on the XM2VTS database [5-3]. Of course this database is rather optimal for color-based face detection, since it provides with a uniform blue background.

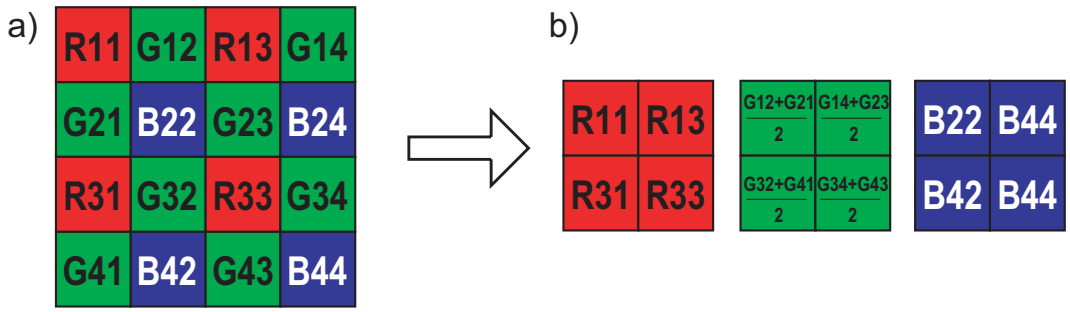


Figure 5-3: Demosaicing of a Bayer pattern.
a) Bayer pattern; b) Demosaicing by simple sub-sampling.

b) A color-based face detection coprocessor

A dedicated hardware architecture (Figure 5-4) is proposed to allow a fast and low-power implementation of such a detection system. The pixel flow directly goes from the sensor to the detection unit and a limited number of memory entries (typically amounting to one line of pixels) is necessary to perform the demosaic operation.

2. Two green components are averaged according to Figure 5-3b) in order to produce the same number of green pixels as there are red and blue pixels.

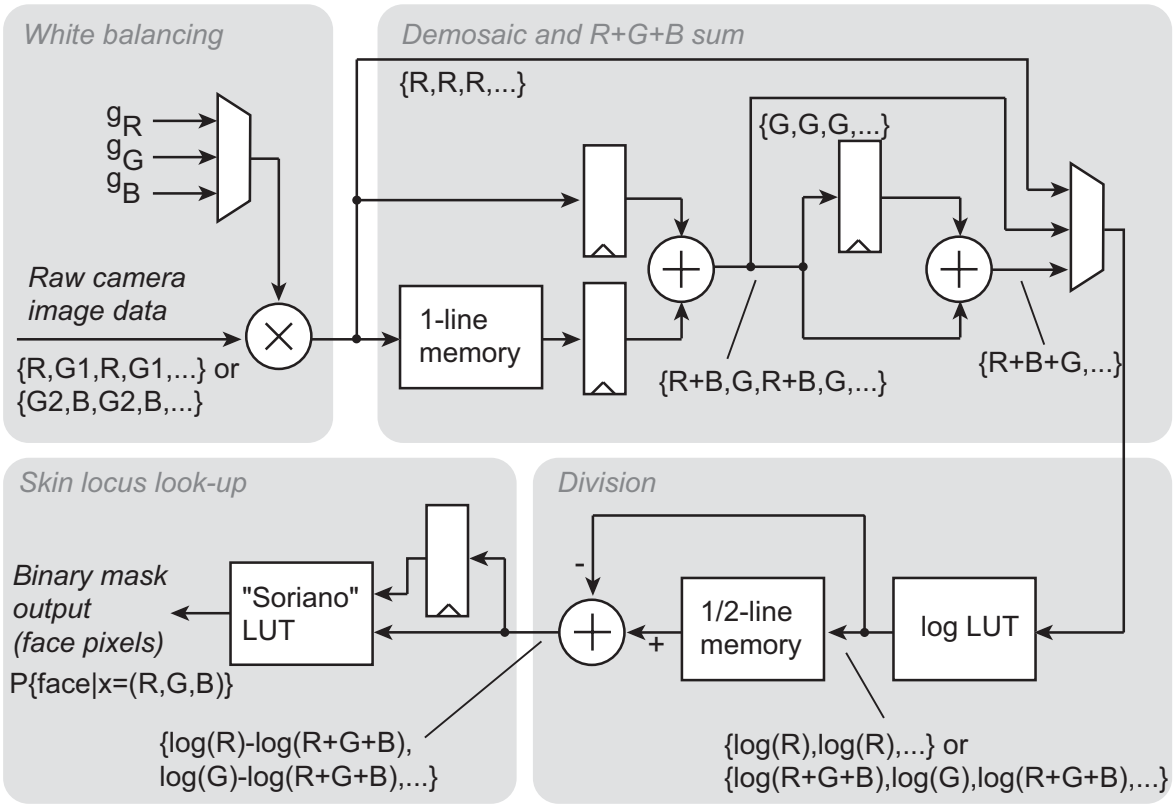


Figure 5-4: Dataflow of the color-based face detection coprocessor.

The raw camera image data arrive as a sequence of *R*, *G*, and *B* values, or more precisely as a sequence of *R* and *G* values in odd-indexed lines, and of *G* and *B* values in even-indexed lines. Firstly, these color values can be corrected using a white-balancing technique, i.e. by multiplying them with constant weights g_R, g_G, and g_B. Note that the determination of these weights needs to be handled outside of this architecture (e.g. using a white-patch, grey world or gamut mapping algorithm, see Subsection 4.2.1), and that the three weights g_R, g_G, and g_B, correspond to a von Kries model simplification.

The architecture then handles two consecutive pixel lines in order to produce half a line of binary mask values to indicate whether the given pixel belongs to the face color space face or not. This corresponds to a vertical and horizontal decimation by a factor of two (Figure 5-3).

The processing starts with an odd line, the incoming R and G components being stored in a 1-line memory. Then, when the pixels of the even line arrive, i.e. *G* and *B* components, the first adder will compute either the sum *R+B*, or the sum *V1+V2* (which will be shifted right by one position), in order to produce the down-sampled *V* component (see Figure 5-3b). This last result will be stored in a register, and the second adder will produce the *R+B+V* result.

The division operation appearing in Eq. 5.1 is replaced with the subtraction of the logarithms of the arguments. The inverse operation is not necessary as the entries of the skin locus look-up table are replaced by their logarithm counterpart.

Therefore, once the *R*, *G*, and *R+G+B* values are available, their logarithms will be searched in a look-up table ("log LUT"). Since the processing started with an odd line, the first logarithms to be computed will be *log(R)*, available every second pixel clock cycle. The *log(R)* results need to be stored in a memory containing half a line as they will be used one line later. Then, when pixels of the even line are available, *log(G)* will be computed, as well as *log(R+G+B)*. At the same time, a subtractor will produce the result of the divisions in Eq. 5.1 in the log domain. Once the two normalized coordinates *r* and *g* are available, they can be used to address a second look-up table delivering the binary pixel classification, indicate whether the given pixel belongs to the color space of a face or not.

The system was implemented in a Field Programmable Gate Array (FPGA) connected to a VGACAM image sensor (see Subsection 6.8.2 for a detailed description of this test platform). The VGA image (640x480 pixels) flow entering the coprocessor results in a 320x240 binary mask output that can be stored in a memory or further used "online" for segmentation. The last mask bit outputs arriving two clock cycles after the end of a VGACAM frame, a theoretical frame rate of around 7 Hz is reachable, assuming that the clock frequency of the system is 2.5 MHz. In practice for the developed demonstration set-up, this frame rate is much lower because the binary mask is read from the memory and sent to the PC via a slow USB connection.

Each output bit requires only one multiplication, three additions, two subtractions and four look-up operations[3]. The control part of the circuit is however rather significant, since it represents more than 30% of the total resources. Still, the circuit size remains remarkably small (only 431 logical cells, i.e. 1.8% of an Altera 20K600 APEX device).

5.1.2 Model-based method

Because color-based face detection alone is not sufficiently robust for most applications, a model-based approach is described below, reusing the EGM algorithm described in Chapter 4. The principal advantage of

3. Datapath only, i.e. without taking into account the counters required by the control state-machines.

using the same algorithm to perform both the detection and the authentication resides in the hardware reuse.

We suggest building several representative templates of an average face and to match them against the test image supposedly containing a face. The matching distance and best graph location will then give the location of the face present in the database. As it was exposed in Subsection 4.5.1 the elastic graph matching algorithm can tolerate variations of scaling from roughly 0.7 to 1.4, and robustness to in-plane rotations up to angles of 45°. For factors outside these ranges, it is possible to build additional face templates.

The features used for the face detection could be again Gabor features or normalized morphological features. Since the first seem more robust to changes in illumination and since they provide with a smoother objective function, they will be used in the experiments reported in this section.

The average face template can be constructed by either extracting a reference template from a single image of an average face, or by averaging templates extracted from several face images contained in a database. We arbitrarily chose the first solution, but extracted seven templates from scaled images of a single average face (Figure 5-5). The scale factor ranges from 1/4 to 4/3.

The average face was obtained from 200 images representing a population of 200 persons (1 image per person) present in the XM2VTS database. Each face image was preliminarily manually aligned (i.e. translated, scaled and rotated) and cropped, so that all images share the same number of pixels and so that the centres of the eyes are aligned over the different images. Note that other facial features (e.g. the mouth or the eyebrows) are logically less precisely aligned than the eyes, and consequently significantly more blurred as compared to the latter. Finally, the average face is obtained by a per-pixel average:

$$\psi(x,y) = \frac{1}{200}\sum_{n=1}^{200} f_n(x,y) \qquad (5.2)$$

where $\{f_1, \ldots, f_{200}\}$ represents the set of training faces, and $\psi(x,y)$ is the resulting average face.

Template level	Graph horizontal size (*sx*)	Graph vertical size (*sy*)	*sx* lower limit (0.7×*sx*)	*sx* upper limit (1.4×*sx*)	Scale factor	Size of the scaled average face
0	21	34	14.7	29.4	1/4	80x60
1	28	44	19.6	39.2	1/3	107x80
2	35	56	24.5	49.0	5/12	133x100
3	49	78	34.3	68.6	7/12	187x140
4	63	101	44.1	88.2	3/4	240x180
5	84	134	58.8	117.6	1/1	320x240
6	112	179	78.4	156.8	4/3	427x320

Table 5-1: Parameters of the seven templates used for EGM face detection.
Each template level corresponds to a scale factor. Level 5 corresponds to the size of the images that will be used during the detection process. For other levels, the average face image is scaled according to this ratio, resulting in the image size given in the last column, and indicated in [pixel × pixel]. A reference template was extracted from each scaled average face using a graph of size *sx* pixels by *sy* pixels. The lower and upper limits of *sx* for a template level delimit the actual (i.e. calculated from the matched graph) *sx* range that will be tolerated for graphs matched with this level.

The face detection is performed in a 320×240 pixel image, with the aim of finding face sizes ranging approximately from 14×22 pixels to 150×240 pixels.

The detection thus solely consists in matching the test image against these seven reference templates using the rigid EGM algorithm (i.e. the rigid matching using iterative methods, as defined in Subsection 4.5.1, but not the elastic matching), as illustrated in Figure 5-5. For each template, the rigid graph size is initialized to the one that was used to extract the template from the scaled image (see Table 5-1, columns 1 to 3).

At the end of the matching, the position of the matched graph having obtained the smallest distance will be used as the detected face location, assuming this smallest distance is below a pre-defined threshold. Moreover, the graph horizontal width *sx* of the resulting graph will be calculated and compared to the graph horizontal size *sx* that was used to extract the template that gave this minimum distance. If the calculated *sx* is outside a given range (defined by the "*sx* lower limit" and "*sx* upper limit" in columns 4 and 5 of Table 5-1), this graph will be rejected

as obviously not giving realistic results. In this case, the next smallest distance will be selected, and *sx* will be again compared to the plausible range defined in Table 5-1. For instance, if the minimum distance was obtained with template level 3, but that the horizontal graph size *sx* is 18, the corresponding matched graph will be rejected in favour of the graph having obtained the second minimum distance.

Graph placement in this model-based method can then be initialized with fast color-based detection (e.g. initial simplex placement) or, alternatively, the white-balance in the color-based method (parameters g_R, g_G and g_B in Figure 5-4) can be calibrated based on the face region provided by the model-based face detection. In the latter option, it is possible to measure the average values $\bar{r}$, $\bar{g}$ and $\bar{b}$ over the model-based detected face and to update the look-up table to be used in the color-based method (e.g. using back propagation). Alternatively, it is possible to recover a color corrected image before applying the color-based method described in Subsection 5.1.1. This is an interesting alternative instead of applying a model-free color constancy algorithm on the whole image (e.g. white-patch or gamut mapping).

The von Kries model (as initially defined in Eq. 4.12) in this space is:

$$\begin{bmatrix} R_{can} \\ G_{can} \\ B_{can} \end{bmatrix} = \begin{bmatrix} \bar{r}_{can}/\bar{r} & 0 & 0 \\ 0 & \bar{g}_{can}/\bar{g} & 0 \\ 0 & 0 & \bar{b}_{can}/\bar{b} \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \qquad (5.3)$$

with the index "*can*" designating typical values under a canonical illuminant, the bar "-" marked over parameter symbols meaning the average value of this parameter, and *r*, *g* and *b* follow the definition of Eq. 5.1. Compared to Eq. 4.12, we thus make the further assumption that the "intensity" R+G+B is preserved (i.e. $R_{can} + G_{can} + B_{can} = R + G + B$).

However, using both approaches simultaneously (i.e. fast detection *and* calibration) is dangerous as this could lead to diverging solutions. Indeed, a false detection would lead to a false white-balancing (e.g. correcting image colors so as the color of a non-face region corresponds to skin-like colors). This in turn would completely bias the color face detection and the entire face detector would need to be reset to reasonable values in order to be again able to detect faces.

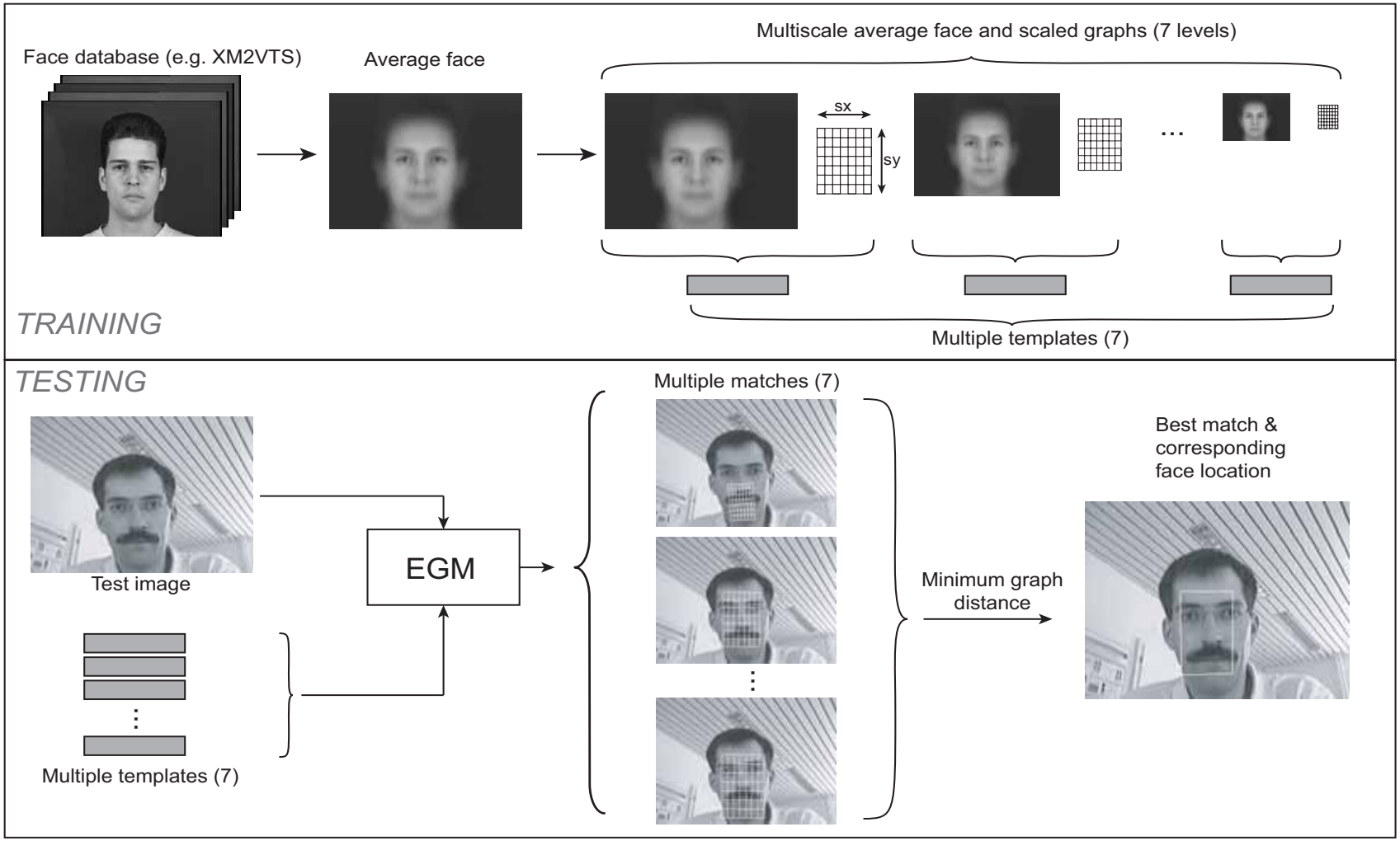


Figure 5-5: Training and testing of the model-based face detection.
In the training phase, the average face is constructed from a set of face images, and scaled to 7 sizes. A graph of corresponding size is used to extract a reference template from each image. During the testing, the EGM algorithm is applied to the test image using successively the 7 reference templates. The resulting 7 matched graphs are sorted according to their matching distance, and the graph with minimum distance is used as face location.

5.2 Fingerprint verification

The algorithmic framework based on elastic graph matching will now be extended to the fingerprint modality. Similarly to face detection, re-using the same algorithm potentially allows to reuse part of the previously developed hardware and to obtain so for instance an efficient mobile multimodal biometric verification system.

Automatic fingerprint identification systems (AFIS) were among the first applications of machine pattern recognition, although they still represent a complex challenge especially with poor quality fingerprint images. According to Maltoni et al. [5-9], fingerprint matching methods can be classified in:

- minutiae-based,
- correlation-based, and
- ridge feature-based techniques.

The first category attempts to detect and match bifurcations or endings of the ridges, while the second and third methods are performing a correlation of the image or of ridge patterns extracted from the image.

While minutiae extraction is the most widely used technique, it is known to be problematic with low-quality fingerprint images because of the large number of falsely detected minutiae. On the other hand, ridge feature-based methods are less sensitive to image quality[4], but are generally characterized by a lower distinctiveness. Finally, the computational requirements of correlation- and ridge feature-based are high, although they are far more regular than minutiae-based methods and are thus good candidates for DSP or ASIC implementations. In particular, correlation and filtering are efficiently performed in the frequency domain.

5.2.1 Requirements for mobile applications

Although the degrees of freedom involved in the acquisition of fingerprint images seem reduced as compared to face images, a large number of perturbations may decrease the matching rate of the algorithm. Primarily, some perturbations not specific to mobile applications are caused by the degradation of the finger skin. For instance, a very low

4. They are somehow holistic, similarly to elastic graph matching for face authentication, because they don’t separate first-order features (minutiae) from configuration.

Figure 5-6: Example of reduced overlap between two fingerprint images.

Figure 5-7: Drier and greasier fingerprint impressions of the same finger.

ridge prominence, too moist or too dry fingers, or cuttings can significantly reduce the matching performance.

Due to the reduced available space and the required low cost of fingerprint sensors embedded on mobile devices, the imaging area is generally smaller than for desktop fingerprint sensors. Additionally, whereas finger placement guides may exist on the latter to guarantee a correct alignment of the finger on the device, the finger is placed free-

ly on the sensing surface of mobile devices. The possible in- and out-of-plane rotations lead to large variations of the imaged finger area and to a large range of elastic deformations. The reference template used to match a new fingerprint shall clearly contain as much information as possible so that the overlap of two prints is as large as possible. For instance, fingerprints illustrated in Figure 5-6 have only a small overlapping area and will therefore be difficult to match reliably.

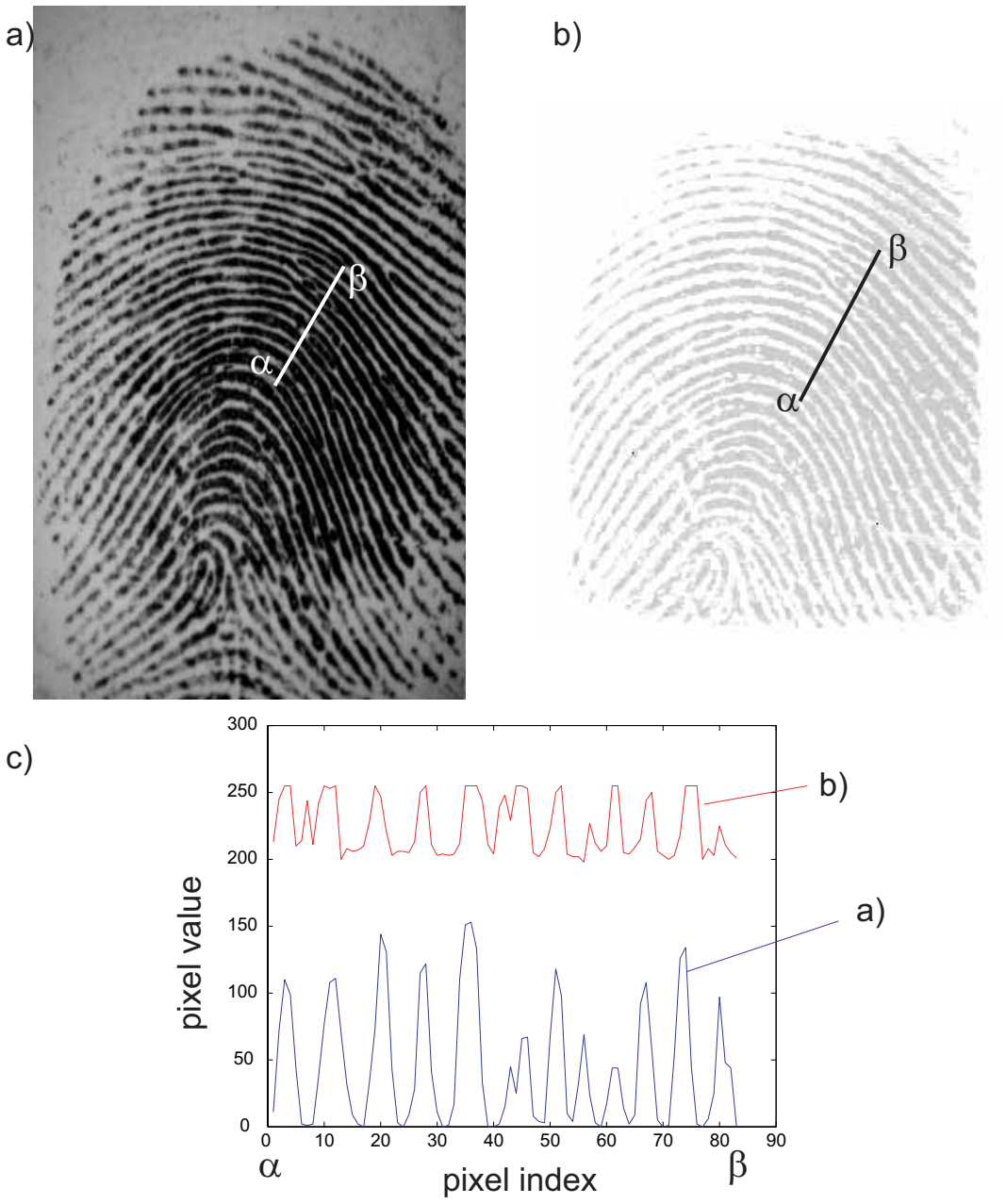


Figure 5-8: Two fingerprint images of the same finger captured with different sensors.
a) capture with an optical sensor (Biometrika FX2000); b) capture with a capacitive sensor (Veridicom FPS200); c) ridge profiles along the line α–β in the images a) and b); a higher value corresponds to a lighter pixel representation.

In addition, the pressure applied on the sensor may vary from one session to the other, leading to “dryer” or to “greasier” prints (Figure 5-7), or equally problematically may introduce elastic distortions in the print. These distortions of the ridge pattern will have an influence on the local ridge spectrum, thus affecting the performance of ridge-feature based methods. While the ridge prominence will have a linear effect on the spectrum energy, distortion of the ridges will produce non-linear variations of the spectrum that will be more difficult to compensate.

Finally, images acquired with different types of sensors (e.g. capacitive, optical, thermal, or even scanning of an ink rolled fingerprint) lead to quite different image quality. For instance, Figure 5-8 illustrates a fingerprint acquired with two types of sensors (optical and capacitive). It can be observed that the ridge profile depicted in Figure 5-8c) is different, and consequently the feature extracted by filtering will be different as well. Performing a cross-match with a template enrolled with a different sensor is thus a complex task.

Some of these problems can be partly solved by a good teaching and practicing of the enrollees. E.g. the users should learn how to place their finger consistently on the device, the sensor should be cleaned regularly, etc.

Nevertheless, robust matching techniques are clearly desirable. Robust feature extraction and matching should therefore:

- enrol most of the fingerprint area using techniques such as fingerprint mosaicking (e.g. [5-7]) or be able to match two fingerprints based only on a small overlapping region;
- be holistic, i.e. the matching process shall not be solely based on the successful detection of a single singular point;
- tolerate rotated and translated fingerprint images;
- tolerate variations of the average ridge value (i.e. remove the DC component of the ridge profile);
- tolerate variations of the finger pressure, i.e. the contrast present in the image (i.e. normalize the ridge amplitude) and the saturations of the ridge.

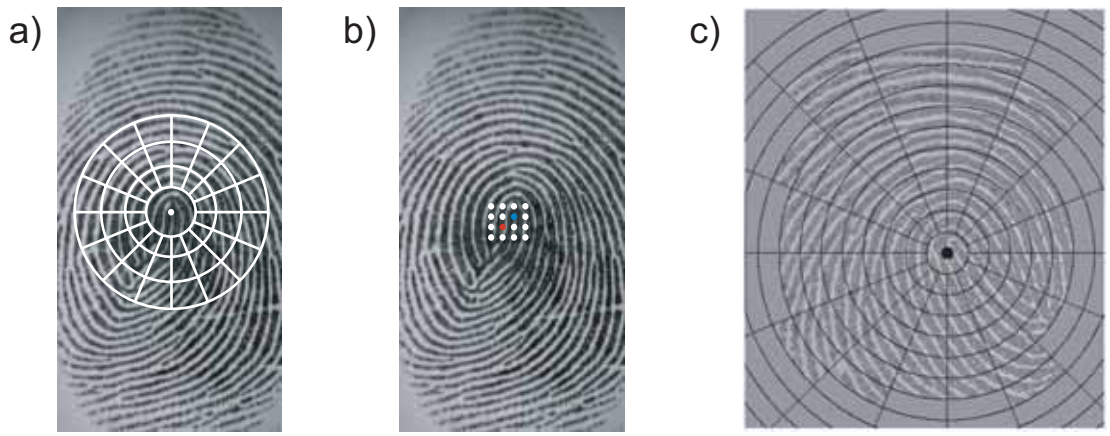


Figure 5-9: Circular tessellation of a fingerprint.
a) Standard circular tessellation around the core; b) multiple core positions to increase the robustness; c) sector rejection based on foreground-background information.

5.2.2 Brief review of existing algorithms

A technique belonging to the ridge feature category is the so-called *FingerCode* [5-5]. It is especially appealing due to its regularity, simplicity and performance. However, FingerCode relies on the precise detection of a singular point, called *core*, around which a circular tessellation is laid, which divides the fingerprint in sectors (Figure 5-9a). A single feature is extracted from each sector using Gabor filters, resulting in a feature vector, which is called the FingerCode.

However the false detection of the core will unsurprisingly lead to a reduced score if the core is erroneously displaced from several pixels, and to a false rejection of the code if the core is detected completely elsewhere. Solutions for probing multiple positions of the core were proposed [5-2] (Figure 5-9b) but involve calculating multiple times the sector-based absolute average deviation or storing multiple templates. The latter solution simultaneously increases the false acceptance rate. Additionally, the algorithm will fail with images that do not even contain the core singularity, although an overlap exists between the test finger impression and the reference template. This last problem will definitely occur with small area sensors (e.g. solid-state sensors) available in mobile applications.

Still, even when the core is present in the image some sectors might fall outside the fingerprint region (i.e. the background). Consequently, only sectors lying on the foreground of the reference and test images shall be used for verification (see Figure 5-9c).

A square tessellation whose position is not related to any landmarks was proposed by Jain et al. [5-6] and Ross et al. [5-10]. Still the translation and rotation parameters are obtained from a minutiae matching step. This registration thus depends on multiple singular points instead of just one, but the misdetection of these minutiae (e.g. when only very few minutiae are present) will lead to a false registration that will ultimately lead to a rejection of the sample. Therefore, we propose to increase the robustness of the matching method by using a template-based approach.

5.2.3 Proposed algorithm based on EGM

An original method based on Gabor filtering not relying at all on the detection of singular points is now proposed, based on a modified Elastic Graph Matching algorithm. This method uses ridge features extracted from Gabor filtered images and aims at finding the largest graph correspondence between reference and test images. Deformations can be taken into account by the EGM algorithm to compensate for pressure variations on the sensor plate.

The image is firstly normalized using the method proposed by Hong et al. [5-4] and filtered with several Gabor filters. Gabor filter banks were introduced in Subsection 4.3.1. For this application the filter bank is composed of N=8 complex filters with following parameters: p=1, q=8, ω_{min}=0.09, ω_{max}=0.11 (see the definition of these parameters provided starting from page 108). The filter bank response in the frequency domain is represented in Figure 5-10 for p=1 and q=8, and filtered images are depicted in Figure 5-11.

In order to reduce the computational complexity, the convolution is performed in the Fourier space by multiplying point-wise the discrete Fourier transform (DFT) of the image and of the filters. After filtering, the magnitudes of the N complex filter responses are calculated in the image space and used as N bidimensional feature images. A N-element feature vector $J_q(x,y)$ (similarly to Eq. 4.31) corresponds thus to the magnitude of a pixel filtered with the q-th filter in the bank and located at spatial coordinates (x,y).

The feature images are subsequently used to separate the foreground from the background of the image. A binary matrix FG representing the foreground mask is obtained using the decision criterion defined in Eq. 5.4, with a typical threshold of $T=10$.

$$FG(x,y) = \begin{cases} 0, S(x,y) \le T \\ 1, S(x,y) > T \end{cases} \text{ where } S(x,y) = \sum_{k=1}^{N} J_k(x,y) \qquad (5.4)$$

Alternatively, it is possible to use a higher threshold and then to use a mathematical morphology closure operation to fill "holes" that may appear in the image. Note that the detected borders of the fingerprint foreground extend slightly in the background, due to the large delay of the filter. Again, for a more precise detection it might be necessary to use a mathematical morphology erosion operation with a structuring element approximately equal in size to the Gabor filter size.

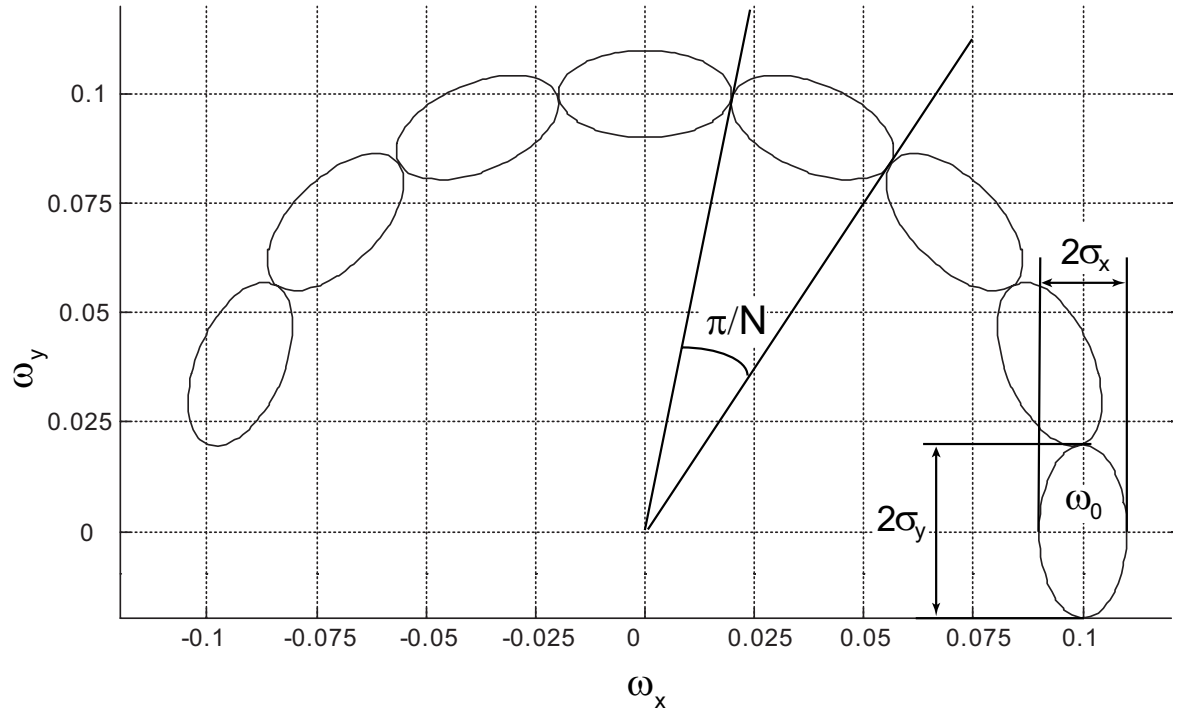


Figure 5-10: Gabor filter bank used for fingerprint authentication (p=1,q=8).

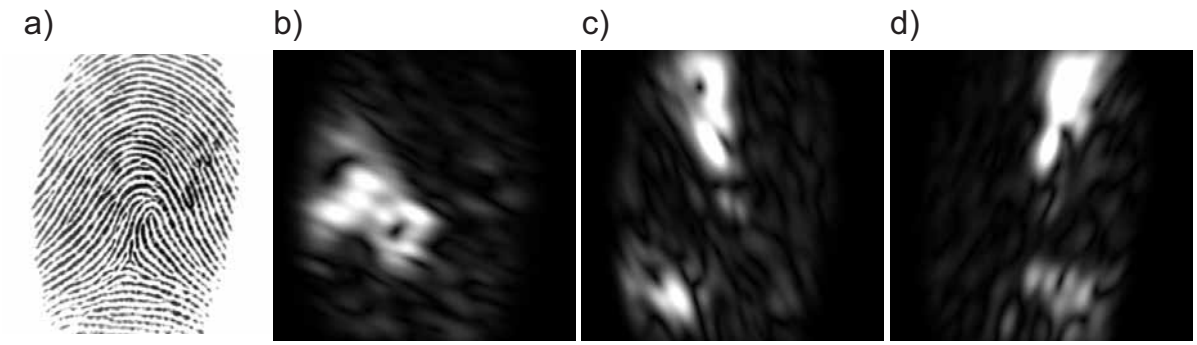


Figure 5-11: Example of feature images.
a) Original image; b)-d) Images filtered with orientations π/8, 3π/8 and 5π/8 respectively.

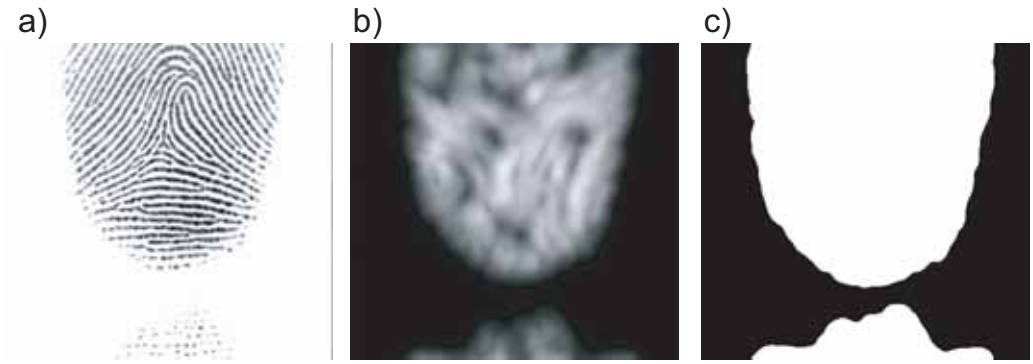


Figure 5-12: Fingerprint segmentation.
a) Original image; b) S(x,y); c) FG(x,y) (background in black vs. foreground in white).

A typical foreground vs. background segmentation is illustrated in Figure 5-12. This information is later used to ignore graph nodes lying in the background during graph matching.

The EGM process aims at finding a graph isomorphism between two representations. This means features extracted from a reference graph in the reference image are compared to features extracted from a test graph, which is iteratively placed in the test image until distance measure between the features is minimized. Although the overlap between reference and test graphs is entire in face-based applications (i.e. all reference nodes exist in the test image), this is usually not the case in fingerprint matching, where only a small region of the reference and test graphs overlap.

Consequently it is necessary to find a common subgraph to both test and reference images in order to measure a distance (similarly to Subsection 4.3.1). This subgraph will be called the *Greatest Common Graph* (*GCG*) as defined in Eq. 5.5.

$$GCG = FG^{ref} \cap FG^{test} \qquad (5.5)$$

Reference features are regularly extracted on an orthogonal grid (e.g. 32x32 nodes) laid on the entire image during the enrolment. These features are annotated with a foreground vs. background tag as some of the nodes do not contain any information (cf. Figure 5-13a). The matching of a new test image is performed with several applications of the iterative Simplex Downhill method, which was described in Subsection 4.5.1. More precisely several randomly chosen sets of vertices (typically 8) containing orientation and translation parameters (x and y position of the grid center) are used for the initialization of the algorithm, giving rise to eight local minima, the absolute minimum being used later on.

During each iteration the graph corresponding to the 3 parameters of a Simplex vertex (translation + orientation) is placed on the test image and the only nodes that are taken into account are those lying simultaneously on the foreground of the reference *and* test images. An additional step is used to remove nodes that are not connected to at least two neighbouring foreground nodes. The Simplex orientation parameter is constrained to stay between -90° and +90° which we consider as the maximum tolerable orientation of a finger on the sensor.

Similarly to face authentication (see Eq. 4.32 and Eq. 4.33), the distance can be obtained either by summing the node Euclidean distances (Eq. 5.6) or by summing the dot products of the corresponding feature vectors (Eq. 5.7).

$$d_{eucl} = \sum_{i \subset GCG} \sqrt{\sum_{q=1}^{n} [J_{tst,q}(x_i, y_i) - J_{ref,q}(\hat{x}_i, \hat{y}_i)]^2} \qquad (5.6)$$

$$d_{dot} = \sum_{i \subset GCG} \frac{\sum_{q=1}^{n} J_{tst,q}(x_i, y_i) \cdot J_{ref,q}(\hat{x}_i, \hat{y}_i)}{\left\| J_{tst,q}(x_i, y_i) \right\| \left\| J_{ref,q}(\hat{x}_i, \hat{y}_i) \right\|} \qquad (5.7)$$

The normed dot products performs usually better as it compensates for the thickness of fingerprint ridges, whereas the Euclidean distance would require an additional normalization step. The following normalization appeared to be relatively appropriate:

$$\widehat{J}_{tst,q}(x_i, y_i) = \frac{J_{tst,q}(x_i, y_i)}{\frac{1}{n}\sqrt{\sum_{q=1}^{n} [J_{tst,q}(x_i, y_i)]^2}} \qquad (5.8)$$

Since Gabor features are not rotation invariant (as it was the case with circular structuring elements used in the mathematical morphology), a rotated fingerprint will have a larger distance. To alleviate this problem, we propose to “rotate” the Gabor features depending on the orientation of the graph as originally proposed by Jain et al.

Finally, some iterations of elastic matching are used to displace the nodes in the foreground. An example of genuine matched graphs is illustrated in Figure 5-13 and of impostor matched graphs in Figure 5-14. It can be observed in the latter that the matching algorithm effi-

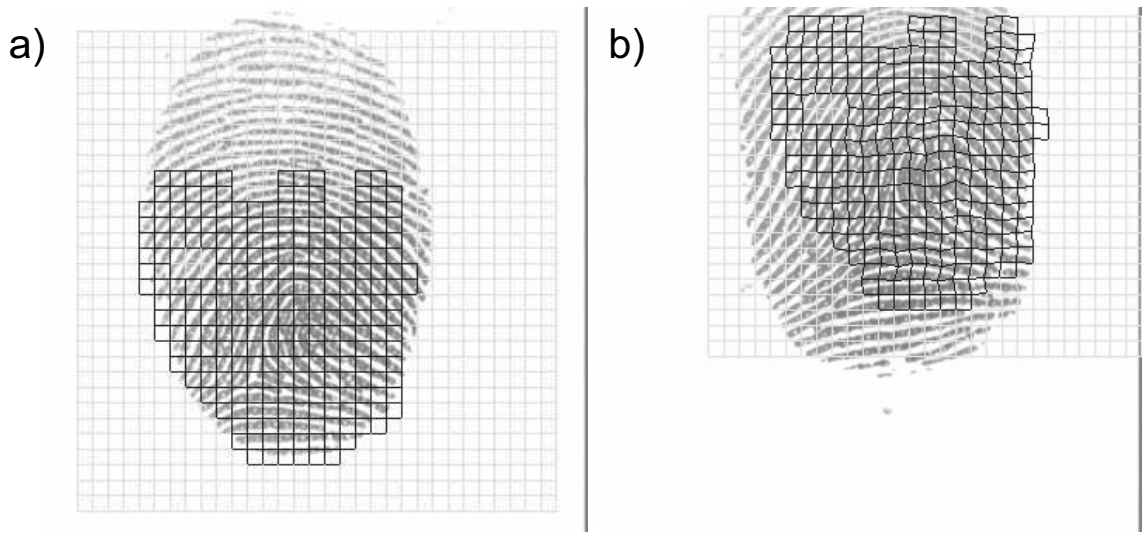


Figure 5-13: Example of a genuine graph match.
a) Reference image; b) Test image. The dark edges link nodes that belong to the foreground of both images (i.e. those nodes that are used for distance calculation).

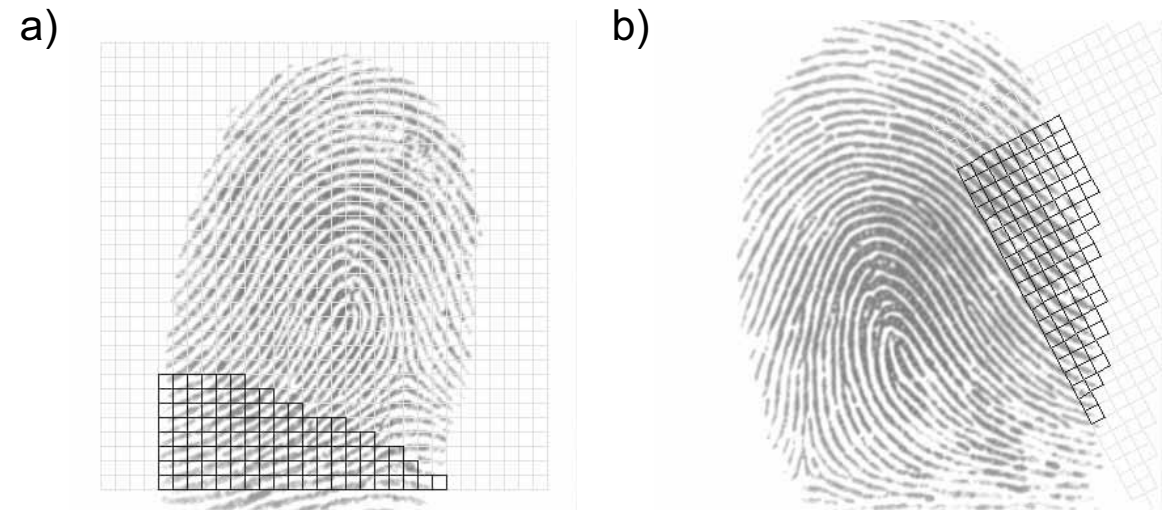


Figure 5-14: Example of an impostor graph match.
a) Reference image; b) Test image (colors having the same meaning as in Figure 5-13).

ciently finds small similar regions between the two non-matching prints, eventually leading to a small matching distance.

However this match should be clearly rejected as, for instance, the core of the fingerprint is respectively above the matched graph in the reference image and below the matched graph in the test image. The simple number of matching nodes in the graph cannot be used alone, as some genuine matches have only a very small number of nodes in common. Consequently, additional heuristics can be used to discard matches as a post-processing step. Note that this post-processing step will not influence the correct placement of the matched graph, i.e. the *registration* rate, but only the final decision.

Bazen et al. [5-1] proposed an interesting extension of the well-known Poincaré index method, to detect the presence of the core[5]. We propose a modification of their method, in order to detect the core and its relative position to the graphs. The principal modification is clearly indicated in the following description of the method.

First, the gradient vector [*Gx*,*Gy*] is computed at each pixel location, using two 3×3 Sobel operators. Then the squared gradient is averaged on a small neighborhood (typically 19×19 pixels) using Eq. 5-9, which still results in two values per pixel, $\overline{G_{sx}}$ and $\overline{G_{sy}}$.

$$\begin{bmatrix} \overline{G_{sx}} \\ \overline{G_{sy}} \end{bmatrix} = \begin{bmatrix} \frac{1}{size(W)}\left(\sum_W G_x^2 - \sum_W G_y^2\right) \\ \frac{2}{size(W)}\sum_W G_x G_y \end{bmatrix} \qquad (5.9)$$

W is the window on which the averaging is performed, and *size(W)* corresponds to the number of pixels contained in the window.

Now, instead of calculating a new gradient vector of the squared directional field (called [*Jx Jy*] in Bazen's contribution) and calculating its rotational, we propose to directly calculate the index with Eq. 5.10:

$$Index = \sum_A rot\left[\overline{G_{sx}}\ \overline{G_{sy}}\right] = \sum_A \left(\frac{\partial}{\partial x}\overline{G_{sy}} - \frac{\partial}{\partial y}\overline{G_{sx}}\right) \qquad (5.10)$$

where *A* is again a window on which the index is calculated (typically 19×19 pixels), which supposedly contains a singular point. Finally, the minimum of the above index is calculated to locate the core (as depicted in Figure 5-15).

Only when there is a strong evidence of a core presence in both reference and test images, complemented by the fact that they do not match, is the matching distance increased (or the template is definitely rejected). Whenever a core is not detected in one of the two images, the score is left unchanged since it is not possible to know if this is due to a real absence of core or to a detection error.

5. The authors seem however to use a side effect ot the numerical computation, when they calculate the singular point from the rotational of a gradient, whereas it is well-known that *rot(grad X)=0* for all *X*.

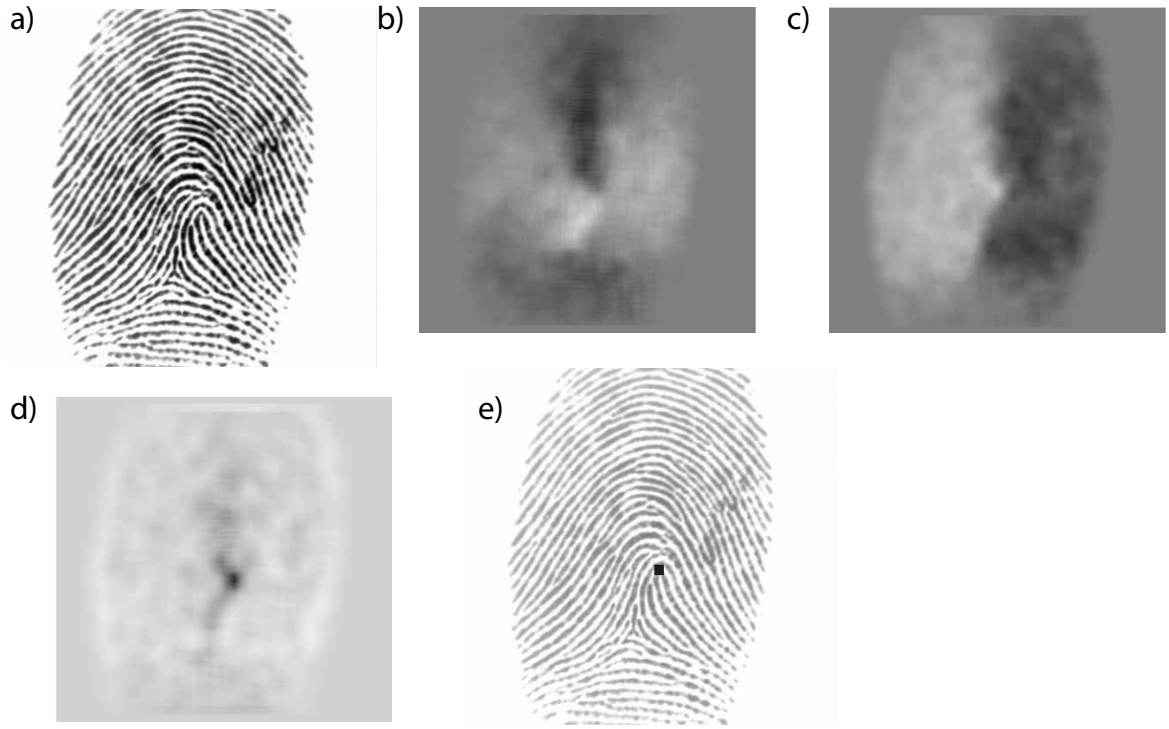


Figure 5-15: Detection of the fingerprint core.
a) Original image (from the FVC 2002 database); b) Averaged gradient component $\overline{G_{sx}}$; c) Averaged gradient component $\overline{G_{sy}}$; d) Index, as defined in Eq. 5.10, logarithmically plotted in order to precisely depict the minimum, i.e. the darker area; e) Dot depicting the location of the minimum of c).

On the DB1a set of the FVC2002 database (see below), only one occurrence of a falsely detected core with high probability was observed (i.e. a 1‰ error), whereas a large number of existing cores were not detected (around 26%). As this is substantially lower than the problem of small graph registration a significant reduction of the FAR shall be observed due to the 71% of correct core detection.

5.2.4 Performance evaluation

The Fingerprint Verification Competitions (FVC2000 and FVC2002 [5-8]) were organized to evaluate the performance of state-of-the-art fingerprint matching algorithms. Four databases were used for FVC2002 and are available in Maltoni et al. [5-9], allowing the comparison of algorithms on a fair basis.

In this report, only the results for the first database are reported (labelled as Db1a). This database comprises 8 impressions of 100 fingers scanned with a 500 dpi optical sensor (388x374 pixels). The performance evaluation scenario was suggested by the FVC2002 organizers, i.e. each sample is matched against the remaining samples of the same

finger to compute the false rejection rate (FRR), and the first sample of each finger is matched against the first sample of the remaining fingers to compute the false acceptance rate (FAR). Compared to the official competition, the results reported here are different with respect to the following points: no time limit was imposed in our case, since this is not a fully optimized solution, and the database was used to somehow tune the parameters, whereas only a limited number of images were available to the competitors for this tuning.

The receiver operating curve (Figure 5-16) shows an equal-error-rate of around 10% for the elastic graph matching method. It can be observed that introduction of elasticity improves the results of approximately 1.5% in EER. According to the FVC'2002 results, Bioscrypt's algorithm obtains the best performance with an EER of 0.1% on this database. The average EER for all participants after removal of the two best and two worst results reaches 4%. Although original, the algorithm described in this section clearly requires further investigations in order to be competitive with other state-of-art algorithms.

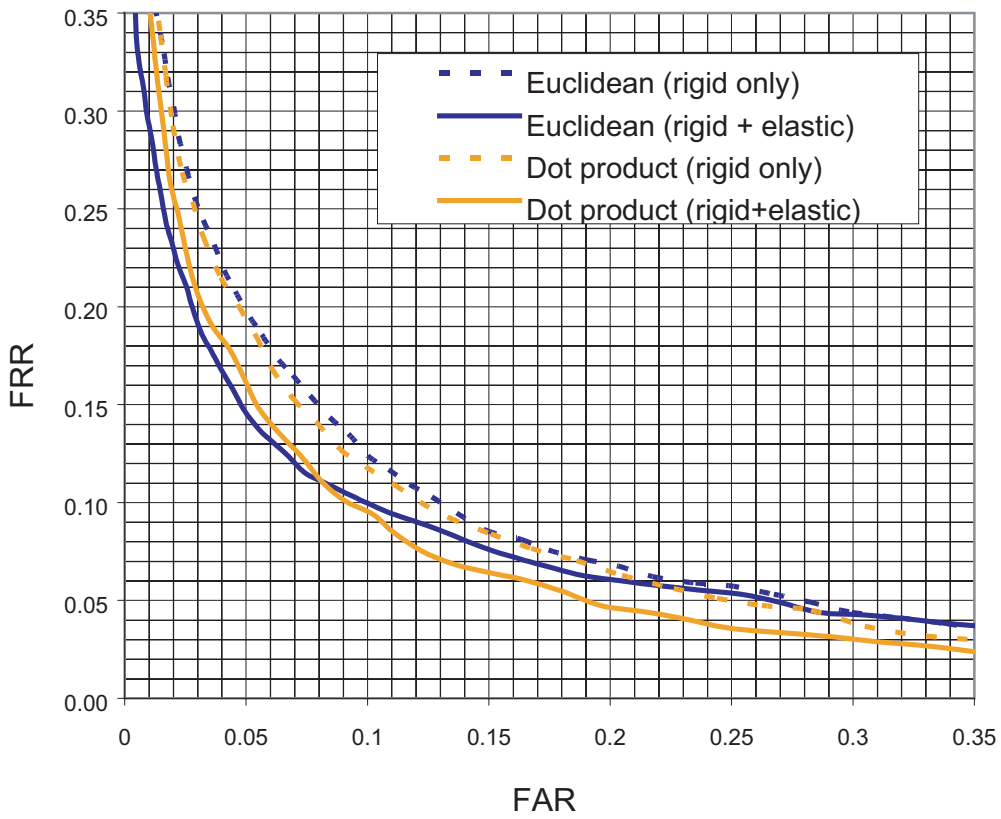


Figure 5-16: ROC of EGM fingerprint verification on the FVC'2002 DB1_a database. EGM with Gabor features using Euclidean metrics and dot product metrics

5.2.5 Further possible improvements

This method inspired from template matching is relatively computationally complex due to the use of Gabor features and of iterative matching. However it is fairly robust to bad acquisition conditions especially when false minutiae detection could occur. As a side effect, its distinctiveness is sometimes also lower than that of minutiae-based methods.

False match of small areas (as in Figure 5-14) needs to be reduced as it is the principal cause of false acceptances (and accordingly of the high EER). Adding a constraint that the two fingerprints must overlap with a larger area would indeed reduce the number of false acceptances, but since the FVC'2002 database contains a large number of fingerprint pairs that have only a small common area the number of false rejections would be in turn increased. Alternatively, using a rejection of matches based on the core detection was shown to be a possible solution, but the core detection should be however improved with respect to its current implementation.

Matching speed could be increased either by using different types of features (e.g. mathematical morphology) or by reducing the number of matching steps used in the iterative methods by using multiresolution approaches, less nodes per graph or less features per node.

Preliminary experimentation shows that mathematical morphology features could be used instead of the costly Gabor features. Following the results published by Soille and Talbot [5-12], we propose to use the mathematical morphology opening and closing with line segments of varying length and of varying orientation. The opening O_L and closing C_L of an image X with a line segment L are defined as:

$$O_L(X) = D_L(E_L(X)) \qquad (5.11)$$

$$C_L(X) = E_L(D_L(X)) \qquad (5.12)$$

where D_L and E_L represent the dilation and erosion with the line segment, respectively (these operators were defined in Subsection 4.3.3).

In the case of fingerprints, a set of line segments L is used, which all have the same length but different orientations. The example depicted in Figure 5-17 uses 8 orientations, with a line segment of 11 pixels.

The *Min* quantity introduced by Soille and Talbot, which corresponds to a pixel-wise minimum of the openings performed with all the structuring elements, is further used to segment the image into foreground (*FG=1*) and background (*FG=0*).

$$Min(X) = \{O_{L_i}(X) \mid (O_{L_i}(X) \leq O_{L_j}(X)), \forall \alpha_i \neq \alpha_j\} \qquad (5.13)$$

$$FG(X) = \begin{cases} 1, Min(X) \leq T \\ 0, Min(X) > T \end{cases} \qquad (5.14)$$

where α_i represents the angle of the *i*-th line segment L_i, and T is the threshold. This threshold can usually be determined with a simple algorithm (e.g. by segmenting histogram peaks).

We propose to use the set of morphological features $J_L(X)$ defined in Eq. 5.15 for fingerprint matching.

$$J(X) = [255 \cdot I_X - abs(C_L(X) - O_L(X))] \cdot FG(X) \qquad (5.15)$$

where I_X is the unit matrix of the same size as X, L represent a linear structuring element, and the *abs* function and the multiplication act as point-wise operators.

Images resulting from those operations are represented in Figure 5-17. These features clearly extract information about the local orientation, similarly to the behaviour of Gabor features. The establishment of the adequate number of orientations, the number and the length of the line segments, need further experimentation. It has to be noted however that this approach is especially appealing due the possible recursive implementation of erosions and dilations along discrete lines, which allow constant time computation with respect to the length of the structuring element [5-11].

Finally, EGM could be combined with a minutiae-based method to improve its distinctiveness. Contrarily to the method of Ross et al. [5-10] which uses minutiae registration first to then perform ridge-based matching, we propose to use ridge-based registration first, and then to extract minutiae in well defined regions of the images. This holistic approach should indeed be more robust to fingerprint quality degradation, assuming that the minutiae extraction during the enrolment is supervised and that the quality of the minutiae can be controlled.

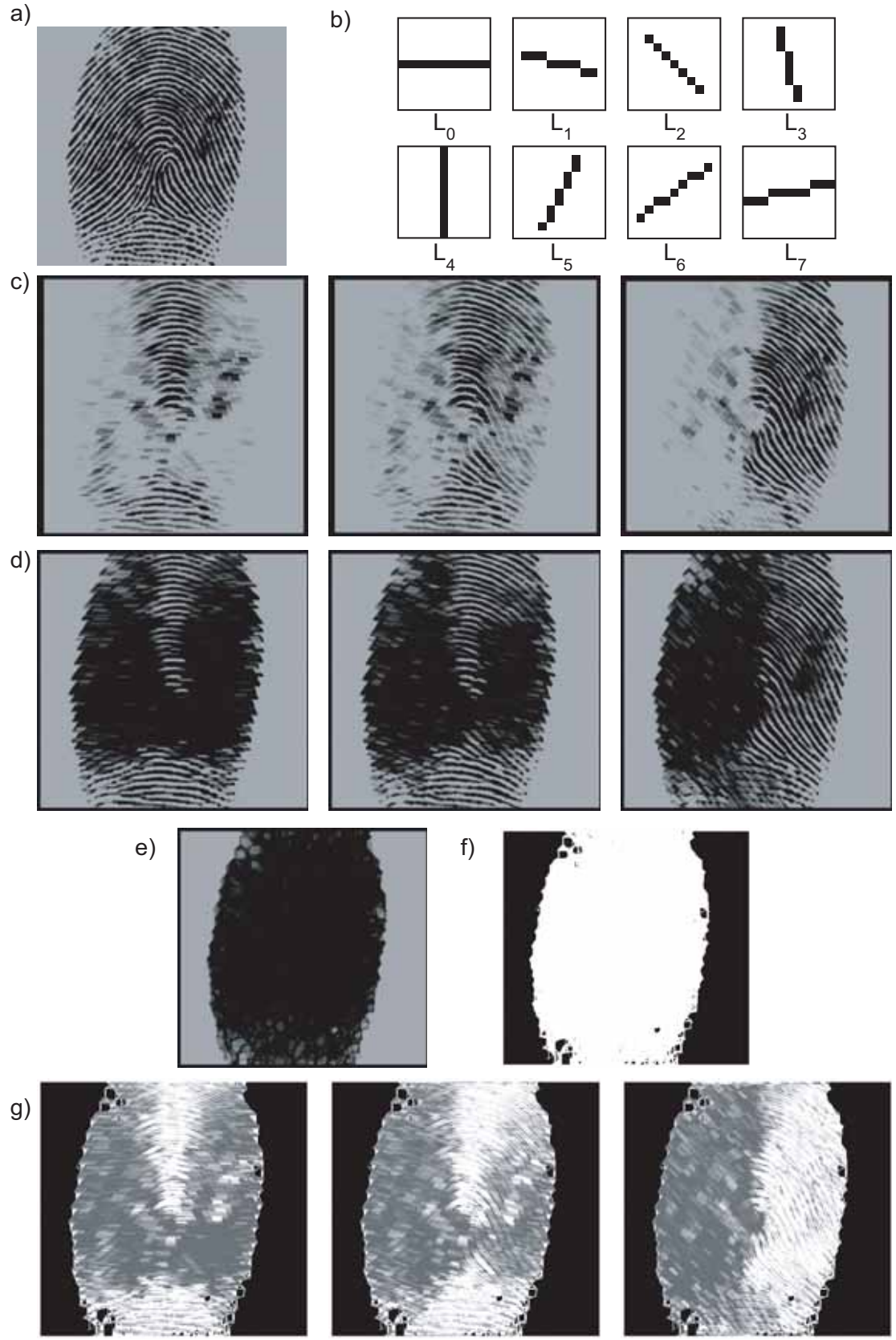


Figure 5-17: Morphological opening features of fingerprint images.
a) Original image; b) Structuring elements L_i; c) Closing $C_L(X)$, for L_0, L_1, and L_2; d) Opening $O_L(X)$, for L_0, L_1, and L_2; e) *Min(X)*; f) Thresholded version of e); g) Morphological features *M(x)*, for L_0, L_1, and L_2.

5.3 Conclusion

Many different categories of applications can benefit from the robustness of Elastic Graph Matching. This chapter introduced two applications, namely face detection and fingerprint matching.

Face detection based on EGM combined to e.g. color-based face detection allows the robust detection of the face location that may be necessary in unattended automatic enrolment applications.

Fingerprint matching using EGM of Gabor features, yet not achieving the performance of world leading companies, delivers interesting results that could be used to develop robust fingerprint matching on mobile devices. Additionally, a novel feature extraction technique based on morphological opening was proposed showing promising properties.

The final benefit of reusing EGM for these two applications would be the possibility to implement face detection on the same hardware (e.g. simplified enrolment where the user just controls the graph placement but does not require to manually select the eye locations) and to implement a multimodal mobile biometric system (face and fingerprint).

References

[5-1] A. M. Bazen, and S. H. Gerez , "Systematic Methods for the Computation of the Directional Fields and Singular Points of Fingerprints," *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. 24, No. 7, July 2002, pp. 905-919.

[5-2] E. Bezhani, D. Sun, J.-L. Nagel, and S. Carrato, "Optimized filterbank fingerprint recognition," in *Proc. 15th SPIE Annual Symposium on Electronic Imaging, Image Processing: Algorithms and Systems II*, Conf. 5014, Jan. 21-23, 2003.

[5-3] P. Geiser, "Réalisation VLSI d'un système de détection de visage," *M.Sc. thesis report*, IMT, University of Neuchâtel, Feb. 2003.

[5-4] L. Hong, Y. Wan, and A. Jain, "Fingerprint Image Enhancement: Algorithm and Performance Evaluation," *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. 20, No. 8, Aug. 1998, pp. 777-789.

[5-5] A. K. Jain and S. Pankanti , "Fingerprint Classification and Recognition," Image and Video Processing Handbook, A. Bovik (Ed.), *Academic Press*, New York, 2000, pp. 821-836.

[5-6] A. K. Jain, A. Ross, and S. Prabhakar, "Fingerprint Matching using Minutiae and Texture Features," in *Proc. Int'l Conf. on Image Processing (ICIP'01)*, Thessaloniki, Greece, Oct. 7-10, 2001, pp. 282-285.

[5-7] A. K. Jain, and A. Ross , "Fingerprint Mosaicking," in *Proc. IEEE Int'l Conference on Acoustics, Speech, and Signal Processing (ICASSP'02)*, Vol. IV, Orlando, Florida, May 13-17, 2002, pp. 4064-4067.

[5-8] D. Maio, D. Maltoni, R. Cappelli, J. L. Wayman, and A. K. Jain , "FVC2002: Second Fingerprint Verification Competition," in *Proc. Int'l Conf. on Pattern Recognition (ICPR'02)*, Vol. 3, Quebec City, Canada, Aug. 11-15, 2002, pp. 811-814.

[5-9] D. Maltoni et al., Handbook of Fingerprint Recognition, *Springer Verlag*, Heidelberg, Germany, 2003.

[5-10] A. Ross, A.K. Jain, and J. Reisman , "A Hybrid Fingerprint Matcher," in *Proc. Int'l Conf. on Pattern Recognition (ICPR'02)*, Vol. 3, Quebec City, Canada, Aug. 11-15, 2002, pp. 795-798.

[5-11] P. Soille, E. Breen, and R. Jones , "Recursive implementation of erosions and dilations along discrete lines at arbitrary angles," *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. 18, No. 5, May 1996, pp. 562-567.

[5-12] P. Soille and H. Talbot , "Image structure orientation using mathematical morphology," in *Proc. 14th Int'l Conf. on Pattern Recognition (ICPR'98)*, Vol. 2, Brisbane, Aug. 16-20, 1998, pp. 1467-1469.

[5-13] M. Soriano, B. Martinkauppi, S. Huovinen, and M. Laaksonen , "Skin detection in video under changing illumination conditions," in *Proc. 15th Int'l Conf. on Pattern Recognition (ICPR'00)*, Barcelona, Spain, Sept. 3-8, 2000, pp. 1839-1842.

[5-14] M.-H. Yang, D. J. Kriegman, N. Ahuja , "Detecting Faces in Images: A Survey," *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. 24, No. 1, Jan. 2002, pp. 34-58.

Chapter 6

A Low-Power VLSI Architecture for Face Verification

This chapter describes a VLSI realization performing face verification, which can be embedded in a mobile device, e.g. a PDA or mobile phone.

Some low-power design techniques are recalled after having analyzed the sources of power dissipation. The design-space exploration methodology that was used to select a particular architecture is exposed. Then, the overall face verification system is described and especially one of its components, namely the Elastic Graph Matching coprocessor. Finally, the area, timing and power consumption figures are discussed.

6.1 Introduction

As a reminder, the major requirements for a mobile face verification system additionally to the robustness to intrinsic and extrinsic variations of the face appearance are the low-power operation and a response time remaining below one second for a single verification.

In Section 3.6, the review of existing commercial applications showed that no true solution for mobile devices existed at the realization time of this Ph.D. project. The only known embedded application was running on a 32 bit digital signal processor (DSP), though no indication on the brand and model was given (*http://www.exim21.com/security/frm4000.php*). It is therefore difficult to compare the performance of the system described in this chapter to other systems.

Comparison with an off-the-shelf DSP against the processor core described in this chapter should take into account e.g. the power consumption of the IO circuitry and the clock generation with a PLL, which are not included in the power consumption results that will be given in Subsection 6.8.3.

Additionally, the power figures of DSPs are strongly dependent on activity, and precise information is rarely available in the documentation and/or publications furnished by the manufacturer.

6.2 Low-power design

Portable devices probably represent the key market for low-power electronics. Indeed, battery technology is not evolving as fast as integration capabilities of VLSI circuits. Thus the power requirements of such large circuits cannot be satisfied with the available power furnished by the battery. This increasing gap needs to be compensated by reducing drastically the power-consumption of electronic devices.

However, further motivations for low power devices such as reducing battery cost or reliability are now emerging even in non-mobile devices. Cooling of the circuit requires enhanced packaging that can represent a large part of the final price of a device. A circuit that dissipates more heat will also tend to have a shorter life duration, and consequently its reliability might be decreased. Higher operating frequencies might also cause problems to other system parts due to increased proportion of energy dissipated in electro-magnetic radiation. Finally, the ecological impact is non-negligible.

In this thesis, since we are mainly targeting the mobile category of devices that is usually battery-limited, we will be however confronted to the first (classical) motivation.

It is possible to further distinguish low-*energy* from low-*power* designs, as batteries are generally limited both in energy (dependent on the time the device is switched on) and power (i.e. the maximum instantaneous power that may be delivered). In the case of mobile face verification, power will be consumed intermittently only during a short time when the matching is performed. Power seems thus more critical than energy.

In a top-down approach, low-power optimization can be applied at several levels, e.g. at:

i) system;

ii) architecture of components;

iii) logic;

iv) circuit;

v) and physical.

It is generally accepted that the higher-level optimization leads to higher power savings. In this thesis, we will be mainly concerned with the levels i) to iii) only, while the circuit and physical design optimization will depend on the library used in the synthesis stage.

The system organization will be described in Section 6.3. At this level, it is possible to save power e.g. by reducing inter-circuit communications, or by organizing memories (in order to reduce the number of accesses).

The architecture optimization is tightly linked to the algorithm chosen. The architecture of an Elastic Graph Matching Coprocessor will be described in Section 6.4. All aspects, including the pipeline structure, the number of arithmetic units and of embedded memories, and the definition of an instruction set, have a strong influence on the performance of the architecture.

Finally, the logic implementation of a single function (e.g. a multiplier) has also an impact on the power consumption. At this level standard libraries are generally used and the best compromise in terms of speed, area and power consumption is simply selected from the available logic structures.

6.2.1 Power estimation

Low-power design cannot be achieved without accurate power estimation tools. Power estimation allows to evaluate the impact of design choices and can be performed at the three optimization levels i) to iii) above. Such estimations are naturally less accurate at higher levels of abstraction (e.g. system level), while they are much more precise at the transistor level after synthesis in standard cells, placing and routing. In this work no power estimation was made at the higher (behavioral) level of the design, but power estimation was carried out on the netlist, i.e. after synthesis, prior to placing and routing.

Following Pedram's nomenclature [6-17], a *direct simulation* approach was used by applying a representative set of input vectors. Namely a full execution of the coprocessor's rigid matching program (see Subsection 6.6.2 for a description of this program) was simulated representing several millions of clock cycles. Each node in the netlist was annotated

accordingly with its activity, and power estimation was carried out with Synopsys Power Compiler. Results will be described in Section 6.8.

6.2.2 Sources of power dissipation

Understanding the different causes of power dissipation in CMOS circuits will give an insight into the optimization of the design at higher levels. Firstly, the total power dissipation can be decomposed in dynamic, short-circuit and leakage power [6-20]:

$$P_{total} = P_{dynamic} + P_{short-circuit} + P_{static} \qquad (6.1)$$

Although the major issue with CMOS technologies larger than 0.35 µm was clearly the dynamic power consumption related to capacitance switching, an important issue comes from static power-consumption due to leakage current in nowadays aggressive technologies [6-21][6-22]. Indeed, the face verification system will not be operating continuously but only at given time intervals. Thus, the leakage current occurring in the idle mode might represent a large power consumption reducing the life of batteries.

Techniques such as minimum input vector selection, gated VDD, variable/multi threshold voltage, or weak inversion can be used to reduce the power consumption induced by leakage current although these techniques we will not be described in details here. In running mode, the effect of static power consumption will be important in 0.18 µm technologies and below, only if a very low supply voltage is used. The system described in this chapter will work at nominal supply voltage (1.8 V) in a 0.18 µm technology. Thus, only dynamic power consumption will be considered first, assuming that techniques such as gated supply voltage could be added later independently of architectural choices.

Considering, the clock frequency f, the equivalent switched capacitance $C_{switched}$, the supply voltage V_{dd} (which is supposed to correspond as well to the voltage swing), the number of transistors N and the average switching activity a (i.e. the number of gate transitions during a period of the clock signal divided by the total number of gates), the dynamic power consumption can be computed by:

$$P_{dynamic} = C_{switched} \cdot V_{dd}^2 \cdot a \cdot N \cdot f \qquad (6.2)$$

Consequently, it is possible to reduce the dynamic power consumption by lowering each of the parameters involved. Unfortunately these are closely related and cannot be reduced simultaneously. For instance lowering the supply-voltage - which is obviously the most aggressive approach due to the squared term - will lead to larger propagation delays. This might be compensated by larger gates or parallel calculation, i.e. increased switched capacitance, which in turn rises again the dynamic power consumption.

6.2.3 Low-power design techniques

Some general design techniques [6-17] aiming at decreasing the dynamic power consumption are reviewed below, and their application in the coprocessor architectures described later on are mentioned explicitly.

a) Technology scaling

Generally the switched capacitance can be reduced proportionally to the technology scaling (i.e. moving to a smaller technology node, e.g. from 0.18 to 0.13 µm). When keeping the supply voltage constant while scaling the technology, the gate delay gets shorter than if the supply voltage was scaled down accordingly. When reducing the supply voltage a squared effect can be observed on the power consumption together with only a linear influence on the gate delay. Therefore scaling down the technology always leads to lower power consumption, but fabrication costs of state-of-art technologies are usually much higher than for more mature technologies. For low-voltage, the threshold voltage becomes a major issue both in terms of speed and of leakage. Designing architectures for technologies below 0.1 µm and supply voltages below 1 V should hence deserve another discussion [6-23].

The coprocessors realized for the face verification system described in this Ph.D. work were synthesized in a 0.18 µm technology with a working frequency of 10 MHz. The library is thus substantially faster than needed for the circuit, which means that it could readily work at higher frequencies (see Subsection 6.8.3), but the power consumption is reduced anyway compared to a circuit synthesized in a larger technology (see Subsection 6.8.3).

b) Reduced clock frequency

Reduction of clock frequency requires further techniques to compensate for the consecutively lowered throughput e.g. using pipelining

and/or parallel processing (a general overview of architectural choices is provided e.g. by Hennessy and Patterson [6-8]). The number of cells is consequently increased, but the capacitance of each cell may be lower due to looser timing constraints. Additionally a reduction of the supply voltage could be used.

Pipelining consists in reducing the depth of the critical path by partitioning it using additional flip-flops or latches. The latency of a pipeline structure is increased but the throughput is augmented by the number of pipeline stages, since new data can enter the pipeline at each clock cycle. However, long pipelines may generate hazards related to data dependency and branching. Mechanisms used in general purpose architectures to circumvent these hazards are generally irregular and complicated and consequently do not fit well to low-power consumption needs. A trade-off between parallelism and regularity is thus necessary to find the correct pipeline length. For instance, a three-stage pipeline was selected in the face authentication coprocessor (see Subsection 6.5.1).

In parallel processing multiple results are computed in parallel during a single clock period. For programmable processor architectures one can further distinguish the way this parallelism is handled, namely between Multiple Instruction Multiple Data (MIMD) architectures and Single Instruction Multiple Data (SIMD) architectures. Some inherent parallelism must be present in the algorithm either at the instruction level (ILP, Instruction level parallelism) or at the data level. While the MIMD structure is generally optimal for ILP extraction, SIMD architectures benefit from important data parallelism.

Very Long Instruction Word (VLIW) architectures can combine both MIMD and SIMD approaches by allowing several instructions to occur simultaneously, and one or many of these instructions may be executed on different data. The sequencing of this parallelism is however done off-line with a special compiler, and not at run-time as it is the case in superscalar architectures. SIMD and VLIW architectures fit very well low-level regular signal processing, while superscalar MIMD architectures are generally reserved for general purpose processors.

Finally, the clock frequency of the face verification system that was realized is only 10 MHz and the throughput is still largely sufficient with the parallelization of certain units.

c) Reduction of switching activity

The internal switching activity of cells is generally difficult to estimate, as it is strongly data dependent. This dependency can be due to spatial correlation or to temporal correlation. Moreover spurious transitions (called *glitches*) and circuit style (i.e. static CMOS vs. pass-transistor logic or differential cascode voltage swing styles) also affect the switching activity.

The larger power savings are usually achieved at the system and architectural levels: by reducing the computational complexity of the algorithms - i.e. the number of operations to be performed - the number of gates and their switching activity will be drastically reduced together with the required maximal frequency.

At lower levels it is possible to statistically estimate the correlation of signals and to reorganize gates or possibly to encode the signal to minimize the number of transitions. This is especially useful for large shared buses that have an important capacitance and that present a large correlation (e.g. an address bus with sequential addressing). In this case recoding of the bus values - e.g. using bus inversion or gray encoding - can lead to important power savings.

Techniques for minimizing glitching activity consist mainly in balancing path delays in combinatorial blocks. If all gates switch nearly simultaneously then the average number of glitches at the output can be reduced. This is already a low-level optimization (e.g. delay insertion after technology mapping) but it can have an impact on the selection of arithmetic structures for instance (see for example Subsection 6.5.3).

The standard cell library used to implement the co-processors is based on a static CMOS logic style. Busses that are internal to the coprocessor were not optimized using recoding, as their capacitance is not significant. This technique could have been applied to the system structure (e.g. bus interconnection with large shared memories), but the system power optimization was not carried out in the frame of this Ph.D. work. Therefore, the reduction of switching activity was mainly tackled algorithmically, by reducing the complexity of the processing and by optimizing the architectural design.

d) Reduction of physical capacitance

The real physical capacitance depends on the gates used in a cell library, on the fan-out of each cell, and on the length of the interconnects.

The total capacitance is consequently very hard to estimate in the early stages of the design and a precise measure can be obtained only after technology mapping or even after the full placing and routing of the circuit (for the clock distribution network for instance).

Again the trade-off with speed should be mentioned since smaller gates have a smaller capacitance but imply also slower speed. Known techniques for reducing the capacitance are e.g. resource sharing, gate sizing, logic minimization, subfunction extraction, or register sharing. Some of these optimizations are possible at architectural level while others such as gate-sizing are done during technology mapping. These techniques were not applied in the design made for this Ph.D. thesis.

e) Clocking

In synchronous circuits an important part of the power consumption comes from the clock generation and distribution. Indeed the clock signal is generally the only continuously switching signal. Additionally it drives a very large capacitance distributed over the whole circuit. Finally the timing constraints on this signal are usually very tight meaning that large transistors are required on the clock path.

Gated-clock [6-18] is a widely used technique that turns off parts of the clock tree when the corresponding blocks are idle. Indeed even though no *data* activity might be present (i.e. no switching of the data signals at the output of flip-flops) in idle blocks, the *clock* tree continuously switches the capacitance of the flip-flop gates connected to the clock. If instead the clock is disabled by an AND gate that is then fed to many flip-flops, an important gain in power-consumption is observed. Note that altering the clock path was a major concern for CAD tools in the past, but modern tools now embed this functionality throughout their flow. Clock-gating gains will be discussed in Subsection 6.8.3.

With deep sub-micron technologies the global interconnect delays have become a major issue as compared to gate delay or local interconnect delays [6-21]. Therefore bringing the clock signal with minimum skew in all parts of the circuit is extremely power consuming. Consequently it is also proposed to partition the circuit in small blocks and to generate local clocks based on a master clock. This technique is known as GALS (globally asynchronous locally synchronous) and can relax the constraints on the clock tree synthesis. The ultimate solution would be to use fully asynchronous circuits using handshake signals instead of clocks for synchronization [6-24]. Assuming that handshake signals

are local (and thus fast) would as well relax timing constraints. Nevertheless, clock frequency and area are sufficiently small in our case so that these techniques are not yet necessary.

The present system will use only fully synchronous communications, with gated-clock applied where necessary (see Subsection 6.8.3).

f) Memories

Memories usually represent an important if not the major part of the total power consumption. Memory accesses also often represent the bottleneck of an architecture in terms of throughput, and thus systems requiring very fast data access also imply large power consumption [6-10][6-5]. Again by extracting data parallelism at the algorithmic and system level, it is generally possible to build a hierarchy of smaller and slower memories to save large amounts of power consumption. Similarly local memories consume less power than large global memories.

Cache memories are small and fast memories used to increase the data throughput rather than to decrease power consumption. But so-called *scratch pad* memories i.e. small temporary memories, which are explicitly addressed, present a much lower power consumption than the access to larger memories [6-3]. By carefully designing the software it is thus possible to reduce memory access related power consumption.

g) System-on-Chip (SoC)

At the system level power consumption is dominated by external interconnects, i.e. taking a signal off-chip and feeding it to another chip for further processing. For example, Ansorge et al. illustrate how the reduction of power consumption in image sensor chips is limited by the power consumption of the pads [6-1].

It is thus interesting to build System-on-Chips [6-19][6-11] where many different building blocks are assembled on the same silicon surface thus eliminating the need to output a large number of high-frequency signals. Hopefully only low-frequency signals need to be taken off-chip, which enables the use of slower low-power pads. For example in a face verification system, it might be possible to do the verification on-chip and to output only a decision signal (i.e. 1 bit) - or an encrypted form of this decision - instead of taking out an entire video stream (e.g. 8 bits pixels at a pixel frequency of 4 MHz).

h) Software and firmware

Application software and firmware will also have an influence on the power consumption of programmable processors. The most evident one is the effective use of the available resources by using first the lowest power consuming memories or registers for storing data.

The instruction selection and encoding can also be optimized so that the hardware performing the instruction decoding will lead to a minimum of transitions for repetitive patterns in the instruction code. Similarly, statistical analysis of data can help to benefit from spatial and temporal correlations: operations could be reordered in order to minimize the number of transitions. Hardware-software optimization is consequently an important topic of research.

6.2.4 Design methodology

The design methodology is critical to achieve a good adequation between architectures and algorithms in order to meet the power consumption requirements. Thus the modularity of the electronic devices (e.g. their programmability) and the associated tools are important factors in hardware-software co-design.

The traditional way to realize electronic devices is either to use programmable general purpose processors or to use Application Specific Integrated Circuits (ASIC). The major advantages of the first solution are its modularity (the system can be modified at low-cost by software, i.e. by modifying the firmware) and its price because it can be produced in very large quantities. An ASIC on the other hand can be tightly targeted to a specific application, and consequently it can use the minimum amount of die area (and thus cost), and is optimal with respect to performance and power consumption. However it involves larger development time and fabrication costs that are only interesting in large scale products.

With the recent advent of embedded systems, processor cores can be found on the market [6-29][6-30]. The possibility to combine new cores with a multitude of readily available intellectual property (IP) blocks (e.g. arithmetic, communication, processors, memories, etc.) allows the rapid development of numerous embedded systems.

Another interesting category of processors is the so-called Application Specific Instruction Processor (ASIP), sometimes called Customizable Processor Cores [6-28]. ASIP can be defined as processor cores that can

be modified depending on application needs, i.e. the instruction set can be modified without large changes in the processor architecture, which is thus specialized for a small domain of important applications. Example of changes are: number of storage elements (e.g. registers, memories), functional units (e.g. arithmetic and logical units, multiply-accumulate units) and data communication resources (e.g. buses, ports). The flexibility of the ASIP is higher than the ASIC because the basis of the processor and most of the tools used to program it can be reused. The efficiency is increased by suppressing instructions that are rarely used and adding dedicated instructions.

Industrial examples of customizable cores commercialized with complete software design environments are available (e.g. [6-31]).

Our interest in this thesis was to develop a new processor core that can be optimized based on the application requirements, namely:

- speed;
- fixed-point precision;
- power-consumption.

6.2.5 Design space exploration

Hardware-software codesign is very often involved in the realization of an ASIP for signal processing applications. Hardware-software codesign consists in evaluating design alternatives affecting simultaneously the frontend design space (i.e. software parameters, including code transformations, instruction set coding) and the backend design space (i.e. architectural parameters).

In a classical scheme the hardware and software design would be separated at the specification level and handled in parallel until both are achieved. Finally, a system verification tool would analyze the performance of the whole system. In the codesign approach however, these two design phases are closely inter-related. While the architecture is fixed for general purpose and domain specific processors, so that only software optimization needs to be performed, the partitioning between software and hardware becomes fuzzier in the case of ASIP, because the addition or removal of certain hardware structures has a direct impact on the software.

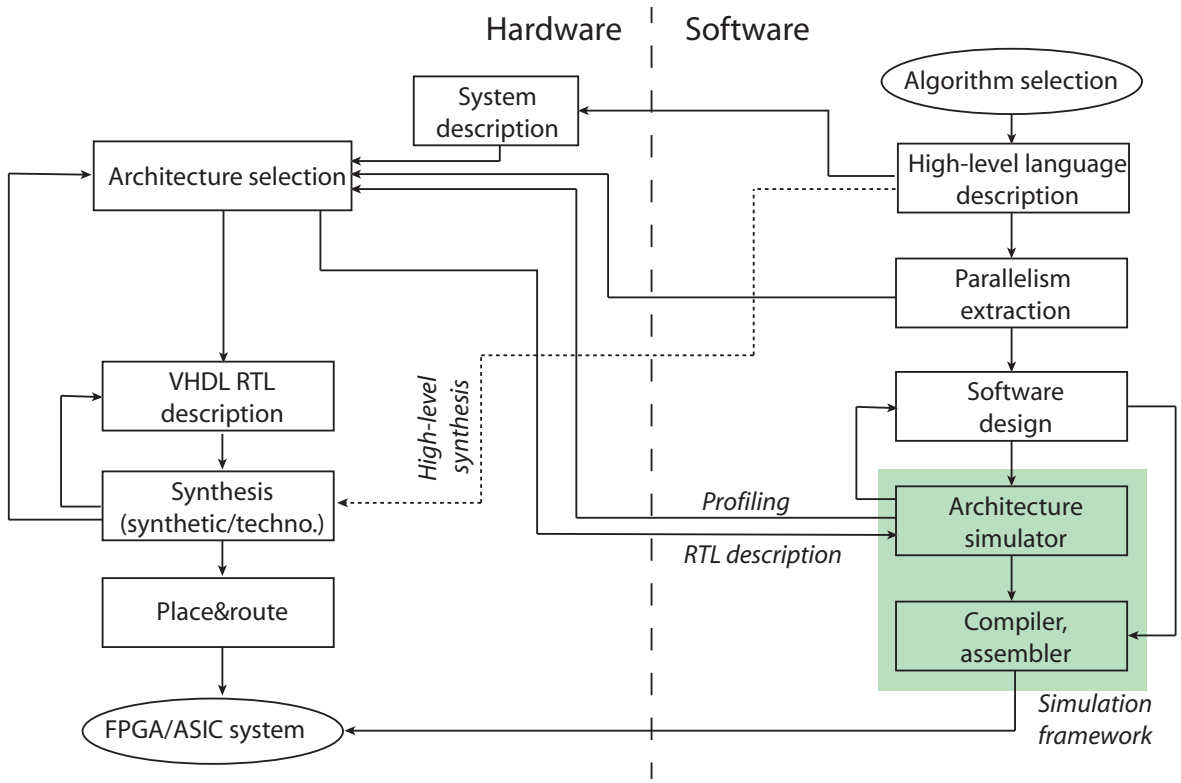


Figure 6-1: Hardware-software codesign methodology.

The design methodology from algorithm to the FPGA/ASIP applied in this work is illustrated in Figure 6-1. The first step is generally the translation of the algorithm or specification into a high-level description, which can be used for both the demonstration of the algorithm and as a model for verification. C++ was used throughout this work.

A dashed line representing high-level synthesis allows to move directly from high-level language description (e.g. C++) to synthesizable RTL descriptions. SystemC from Synopsys (http://www.systemc.org) and Handel-C from Celoxica (http://www.celoxica.com) are two leading products in this domain. C or C++ are generally not optimal for automatic parallelism extraction thus high-level synthesis might fail to extract sufficient parallelism unless explicitly indicated beforehand by the developer. Other languages which embed a description of the parallelism might be better suited for higher level descriptions and for subsequent parallelism extraction (e.g. parallel-Haskell [6-14]). However this methodology was not used in this work.

Instead, an estimation of the available parallelism in the designed system and sub-blocks was performed from the high-level description. This parallelism extraction requires manual hardware-software partitioning, i.e. to manually devise an architecture and the software running on it (sequencing of operations).

In the case of an ASIP the architecture selection consists in choosing the backend (i.e. number and type of ALUs, registers and memories) and the instruction set available to exploit these resources. The description of this ASIP can be done either in a synthesizable language (e.g. VHDL) or at register-transfer-level (RTL) with an architecture simulator. This latter option was used in this work and a framework of tools comprising an architecture simulator, debugger, profiler and assembler was developed in C++. This additional effort prior to the description of the architecture in VHDL is retrieved at later stages of the design when assembler or debugger tools are necessary, since most of the software infrastructure can be reused. Moreover, it should be theoretically possible to automatically generate a synthesizable RTL description in VHDL from the corresponding C++ RTL architectural description used in the architecture simulator, this task being much simpler than high-level synthesis. This was however not attempted in this work. Although all these steps (i.e. architecture selection, software design, VHDL description) were done manually, it should be possible to automate at least part of them. For example automatic compiler generation is a hot research subject [6-7][6-13] that would allow faster semi-automatic exploration of the design space.

Once stable, the simulated architecture is translated in VHDL and synthesized using synthetic[1] and technology (standard cells) libraries. Additionally, an optimized synthetic library such as the Synopsys DesignWare Foundation library allows to optimally select structures such as arithmetic units (especially multipliers in our case) based on constraints specified by the developer, allowing to use only a behavioral description of the operation in the VHDL code.

The design-space exploration consists in trying different architectural configurations and to associate to each of them a cost metric. This metric representation is then used to estimate so-called Pareto points that represent an optimal solution with respect to a single criterion (e.g. the architecture having the minimum power consumption given a circuit area and throughput). Criterions for design-space exploration might be speed/throughput, circuit area, or power-consumption (or a combination of power and throughput such as MIPS/W, i.e. millions of instructions per second and per Watt).

1. The synthetic library contains the information that enables the synthesis tools to perform high-level optimizations, including implementation selection among those available for a component.

At the synthesis level it is possible to give constraints that will have an influence on the selected structure (e.g. a Carry-Select vs. a Carry-Skip adder) but not on the number of arithmetic units, register banks, memories, etc. Consequently the design space exploration is usually done at a higher level of the design. Nevertheless as mentioned earlier, it is generally not possible to estimate reliably the power consumption or the area at higher levels. In this work, the estimation of area and power-consumption were done only after synthesis of the circuit, prior to placing and routing of the standard cells. In the future, it would be interesting to perform a switching activity analysis already in the architecture simulator so as to be able to estimate power consumption at higher abstraction levels.

6.3 System description

The Morphological Elastic Graph Matching (MEGM) algorithm described in Chapter 4 will now be realized in the form of a SoC containing ASIPs for certain processing tasks. The next two sub-sections describe the choices available for partitioning the whole system and a brief description of the system's components.

6.3.1 Partitioning of the system

The whole system should be realized as a SoC containing different blocks for each task. The tasks necessary for face verification are:

- Image acquisition;
- Feature extraction;
- Graph matching;
- Control (decision, sequencing of the system).

Other tasks that might be added but are not compulsory are:

- Image pre-processing, e.g. face detection (see Subsection 5.1.1), white-balancing (see Subsection 4.2.2), etc.;
- Display (color format conversion, etc.), user interface;
- Networking.

The image acquisition and control will be implemented using independent blocks. The control is carried out by a micro-controller unit (MCU) or a digital signal processor (DSP) that will simultaneously

handle the other tasks of the mobile device (e.g. voice compression / decompression or channel coding / decoding on a mobile phone, user interfacing, etc.).

The feature extraction and graph matching tasks can be realized in the same block or in two separate blocks. Parameters influencing this decomposition are:

- modularity;
- sequencing and inter-dependence of the feature extraction and graph matching tasks;
- structure and resources needed by the feature extraction and graph matching tasks.

If the two blocks are realized as two independent IP blocks, one would gain in modularity in the sense that it is then possible to use the graph matching co-processor in conjunction with another feature extraction co-processor. Other types of features might indeed be used in different applications (e.g. fingerprint matching, see Section 5.2) or in applications requiring a different trade-off between the level of security and the hardware complexity (for instance, in Subsection 4.3.1, Gabor features proved to achieve slightly better verification performance and increased robustness to illumination variations, yet at a much higher computational cost).

Two alternatives for sequencing the feature extraction and graph matching tasks are possible:

- Extract features only at given node positions and on demand;
- Extract features for the whole image, store them back in a global memory, and subsequently read results from the memory for given node positions.

The advantages of the first alternative are:

i) reduced memory requirements, because only 17 features are calculated and stored at each iteration, instead of calculating and storing 128×128×17 features once (the image being 128×128 pixel wide);

ii) a reduced computational complexity, because some features may never be required and thus calculated, whereas all the possible features are calculated in the second alternative.

This last advantage will hold if and only if certain locations are never reached by a node. Nevertheless, during rigid matching, a large portion of the image is scanned, which implies that a large number of positions (possibly all the positions, if a rigid scan with a small step is used, see Subsection 4.5.1) will indeed be explored. Potentially, features at certain positions might be reached, and thus calculated, even more than once (associated to different node indexes) unless a costly caching mechanism is used. In addition, optimal methods are generally found for extracting features on the whole image. For example running min-max methods can be used for morphology [6-12], or by reusing data from the calculation of erosion and dilation with smaller structuring elements, and with the same structuring element but at different neighbouring locations [6-25]. In consequence, the first alternative (i.e. a serial computation) seems more effective, although it involves a larger latency and a larger memory requirement.

Whereas graph matching mostly involves complex addressing and multiplication-accumulations, the computation of morphological dilation and erosion involves solely comparisons and table look-ups. As a result, resources would not be used continuously in a single block implementation, assuming that the two tasks are performed subsequently for the above mentioned reasons.

Accordingly, the system should optimally contain two independent blocks used in sequence and switched-off when not used (i.e. in idle mode). SIMD structures will be used in these two blocks, benefiting from the large data parallelism available, and instructions containing multiple operations will be issued thanks to the simplified sequencing. This can be considered as a kind of VLIW architecture, although the length of the instruction word remains small, because at least two operations (one operating furthermore on multiple data) are contained in a single instruction, and the sequencing of these instructions is established during code development (assembly/compilation) rather than at run time. Indeed, following the ASIP paradigm, it is possible to keep a reduced instruction width with a carefully optimized instruction set.

The resulting global structure of the face verification system is illustrated in Figure 6-2 and the sequencing of this system operations is represented in Figure 6-3.

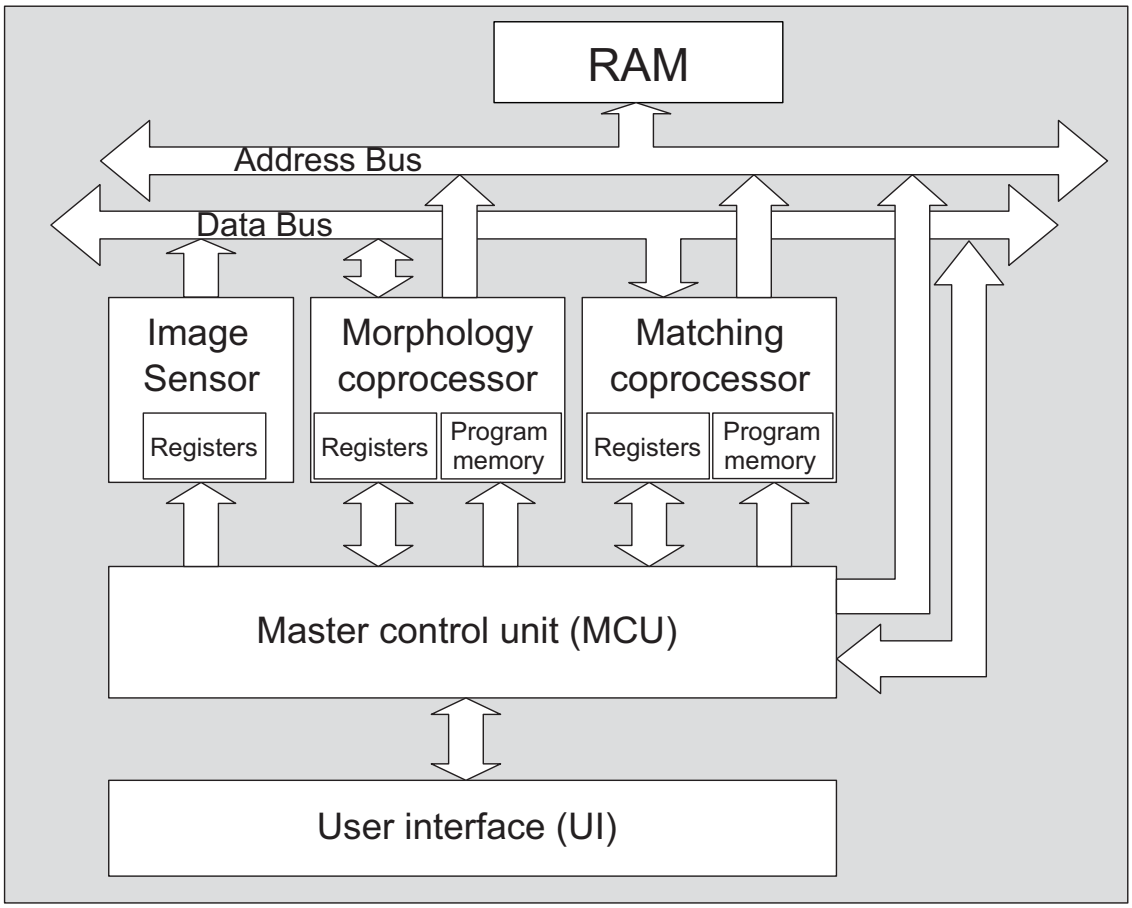


Figure 6-2: Face verification system-on-chip structure.

6.3.2 Components.

The task of the MCU is to manage the user interface and high level tasks. When a face verification is to be performed - meaning that the user activated this function by providing a claim of identity - the master microcontroller will request an image capture from the image sensor and store it to the shared RAM memory using direct memory access (DMA). Before the acquisition, it will be sometimes necessary to calibrate some image sensor parameters (e.g. exposure time and analog gain for a grey level image sensor, white balancing for a color image sensor).

The face verification system described in this report is based on the characteristics of a 256×256 pixel APS CMOS image sensor [6-4], but it could be adapted to other image formats. Color is not necessary from a strict biometric verification point of view. However, color might be wanted for fast face detection (as it was discussed in Section 5.1) and therefore the system was also tested with a VGACAM CMOS image sensor [6-27] implemented using the Bayer color pattern [6-2].

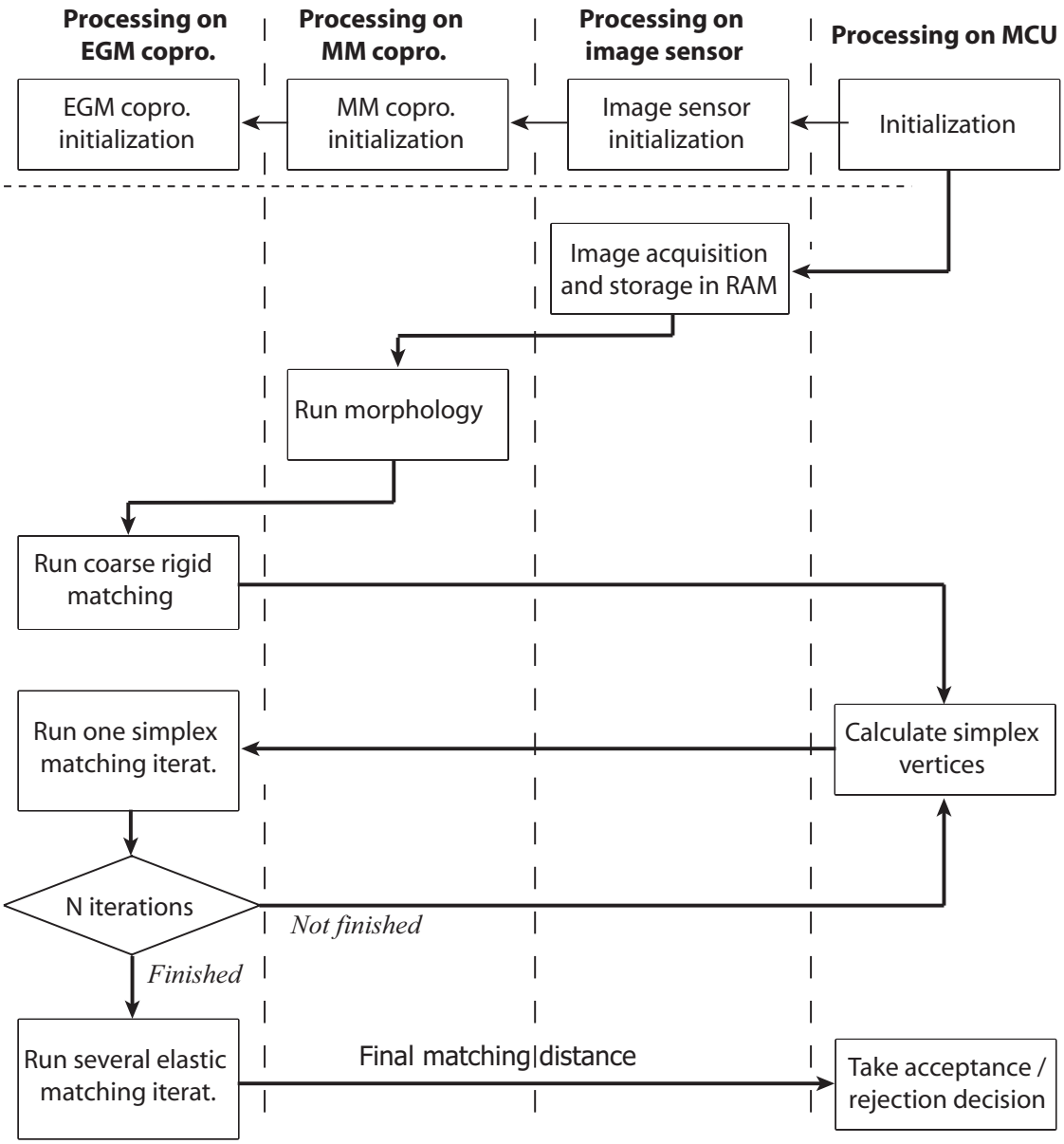


Figure 6-3: Sequencing of operations on the Face Verification architecture.
The execution of programs on the Elastic Graph Matching (EGM) coprocessor, Morphology (MM) coprocessor, and image sensor is controlled by the Master Control Unit (MCU) which ultimately takes an acceptance or rejection decision based on the final matching distance.

These sensors were tested as discrete components connected to the FPGA development platform, but they could have been integrated in the case of a SoC.

Control of the image sensor is done from IO ports of the master control unit (MCU). A 146 by 146 pixel region-of-interest (ROI) is extracted from the 256 by 256 pixel image, respectively from the 640 by 480 pixel image. In the latter case, only green pixels (i.e. one pixel out of four because of the Bayer pattern) are used as they are believed to be less sen-

sitive to variations of the illuminant spectrum. Indeed, the width of the spectral sensitivity of the green filters seems to be the smallest. The ROI selection might be based on some face detection, or it might be simply fixed as the central pixels of the full 256 by 256 pixels. Note that a 146 width is necessary in order to obtain a 128x128 feature image, due to the fact that dilations and erosions cannot be calculated in a band of 9 pixels along the border of the image [6-25][6-26]. These pixels are stored in the shared memory with a special scheme, i.e. at a pre-defined page address and with a particular alignment (see Figure 6-4).

All the SoC blocks have access to the shared memory through a common bus, either in write-only, read-only or read-write mode. The bus is arbitrated by the MCU since it has the knowledge of who is using the bus at any time. The arbitration is done in software avoiding the need of a complex hardware arbiter and of performance degradations. Therefore all blocks are bus master at a time, while other units cannot access the shared memory at the same time. This removes also the need for multiplexing the bus in time for parallel simultaneous access, which would otherwise lead to a bus and memory working at very high frequencies.

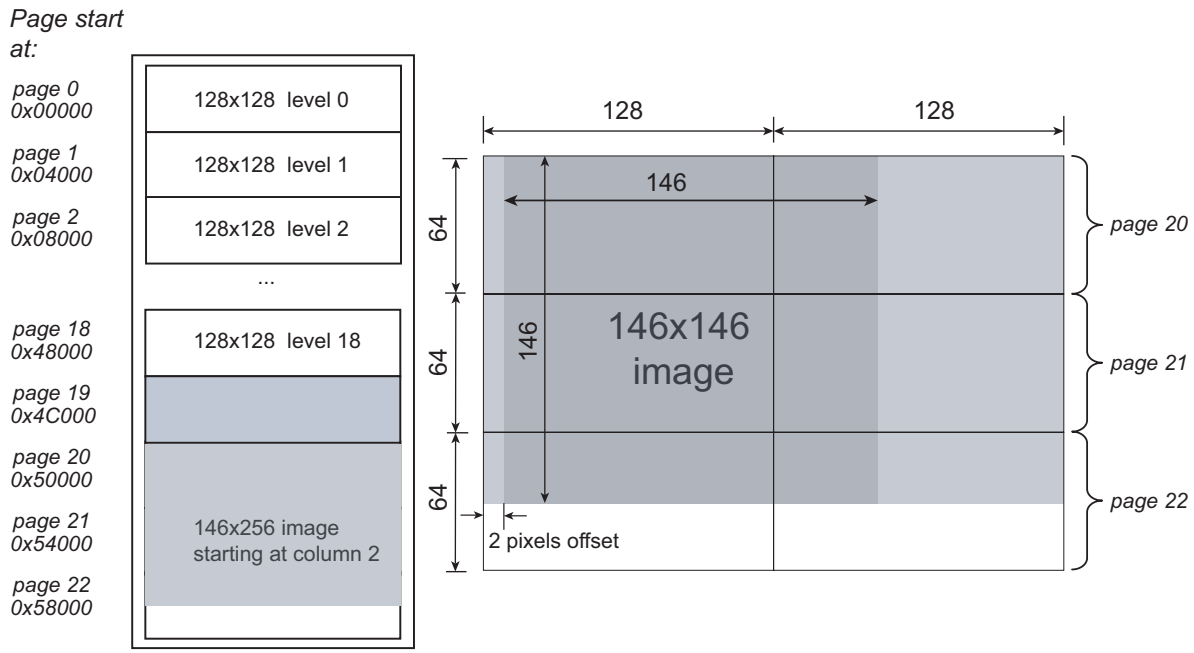


Figure 6-4: Page organization of the shared memory.
Each page contains 128x128 pixels (or equivalently 256x64 pixels as illustrated on the right) and thus needs 5 bits for the page index and 14 bits for the address within one page.

During the face verification process, the shared memory (which should be at least 360 KB to store the subsequent morphological features) is organized in pages (see Figure 6-4). The 146 by 146 probe image will be arranged in 146 lines of 256 pixels, with the effective pixels starting at offset 2 (see Stadelmann [6-26] for motivation behind this choice).

The feature extraction co-processor (i.e. the morphology co-processor) then calculates 19 (or 17 in the case of normalized features) images and stores them starting at the first page, with the storing address being the concatenation of the feature level (most significant bits) and of the *x* and *y* coordinates (less significant bits). This is the main reason for using power-of-two image sizes: the address of a feature can be easily built by concatenation of coordinates without requiring arithmetic operations (see also Figure 6-5).

Finally the graph matching coprocessor accesses the feature vectors from the shared memory and performs the matching operation. The score of the best match together with its corresponding node coordinates can be read by the master control unit (MCU) from accessible internal registers of the graph matching coprocessor. The MCU can similarly write configuration registers and modify the program memory and tables of both coprocessors. The graph matching coprocessor will now be described more in detail.

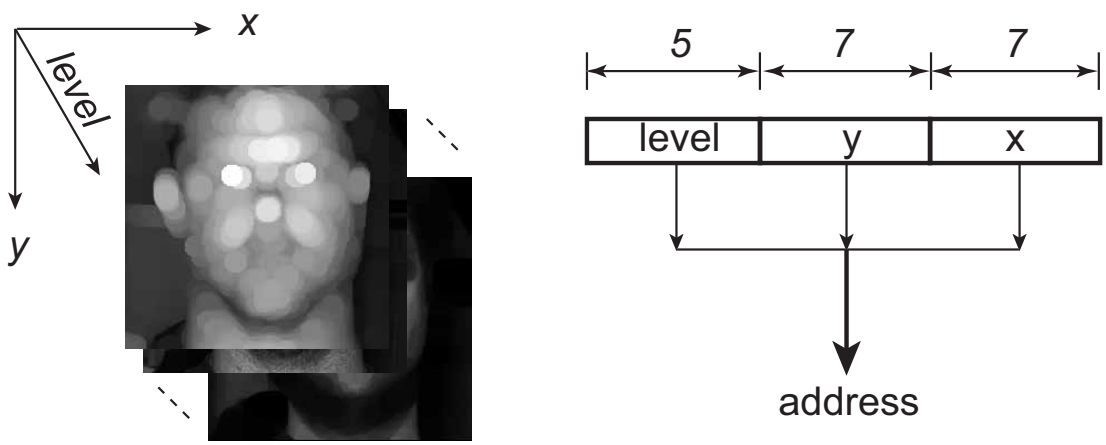


Figure 6-5: Feature address construction.
The shared memory address is established from the concatenation of the feature vector index (or "level") and the (x,y) coordinates. The level thus corresponds to the page index in Figure 6-4, while the concatenation of *y* and *x* corresponds to a 14 bit address within the given page.

6.4 Elastic Graph Matching Coprocessor

The Elastic Graph Matching (EGM) coprocessor is a programmable pipelined architecture with an instruction set optimally designed to perform matching operations between graphs. Starting from pre-calculated features (e.g. morphological features [6-25]), it finds the optimum correspondence between reference features - associated to a reference graph - and features of a test image. The result of the matching is a score that will be used by a higher level unit (the so-called master control unit, or MCU) to decide upon an acceptance or a rejection. Additionally the MCU can access the graph coordinates of the corresponding best match for further processing (e.g. display or connection with face detection algorithms).

This section describes more in detail the type of architecture used for this coprocessor (the description of the general structure of the verification system was already given in Section 6.3). Subsection 6.4.1 recalls the algorithmic choices leading to this architecture, whose components are described more in detail in Section 6.5, while Section 6.6 demonstrates how the graph matching tasks are performed on it.

The execution of programs on the EGM coprocessor is entirely managed by the MCU (although the first possesses its own control, see Subsection 6.5.1) and therefore cannot be used independently of the MCU. The MCU can read or write coprocessor registers and embedded memories to exchange data, or alternatively it can use the shared RAM (see Section 6.3) to exchange large amounts of data. Especially, commands (e.g. reset or run) are sent by writing special registers of the coprocessor.

The EGM coprocessor can be considered as a Reduced Instruction Set Computer (RISC) architecture due to the following characteristics:

- fixed-size instruction length;
- register-register transfer architecture (load-store architecture).

Moreover, the coprocessor executes several operations during each cycle, their sequence being scheduled at compile time, unlike superscalar architectures, for which scheduling occurs at runtime. The complexity shift from hardware to the software represents a highly efficient trade-off when dealing with low-power architectures. This type of software sequencing of instruction parallelism results in Very Large Instruction

Words (VLIW) architectures in which each program instruction (to be executed during one clock cycle) contains several fields describing the multiple operations, and these instructions are thus usually large. In the case of the EGM coprocessor, the actual width of the instructions is reduced (13 bits, see Section 6.7) although they contain two fields, which encode from 1 to 5 operations. This compaction of the instruction is possible thanks to the following characteristics:

- use of dedicated purpose registers;
- single instruction multiple data (SIMD) structure;
- indirect addressing.

Compared to general purpose register (GPR) architectures, where each register can be used in each instruction, dedicated registers permit the reduction of the instruction width due to their implicit usage[2]. Additionally, it is possible to reduce the bus size and thus the power consumption, since a reduced capacitance is achieved when connecting a register only to selected processing units of the architecture. However, the asymmetry that is introduced with dedicated registers quite often implies manual optimizations of the source code, whereas GPR architectures can benefit from efficient compilers. Nevertheless, current compilers should more and more cope with asymmetric architectures.

The use of indirect addressing (e.g. register indirect, indexed [6-8]) – combined with auto-increment – instead of immediate addressing also reduce the instruction set width. Note that some of the index registers can be simultaneously implicit!

Digital Signal Processing (DSP) processor architectures are often specialised to deal with streams of data. An example of such specialization is clearly the use of circular buffers. In this case, a pointer is checked to verify if it reached the end of the buffer. If this is not the case, the pointer is incremented whereas it is otherwise reset to the beginning of the buffer. In the EGM architecture, virtual “free” circular buffers are obtained with an implicit buffer length being a power of two. Thus the pointer is automatically reset when the incrementation overflows its capacity.

2. The instruction thus only contains the opcode since the involved registers are implicitly declared. In GPR architectures, the instruction might require the declaration of up to three indexes (i.e. first operand register, second operand register, and result register), which implies the use of a large instruction word.

Media processors dealing with streams of data in real-time generally manage one input and one output buffer to fetch and store data as they process them. On the other hand, other architectures use a global memory where they fetch the data to process, and to store the result. This last type of architecture was used for the EGM coprocessor to simplify the synchronization of the incoming data, i.e. the morphological features. Indeed, although the pixel stream would be regular, the morphological values are produced irregularly[3] requiring a large input buffer.

6.4.1 Algorithmic choices

Some simplification needs to be performed on the algorithms as they were described in Chapter 4, so that they can be implemented in the form of a low-power embedded system.

Firstly, the normalized mathematical morphology features extracted from a 146 by 146 pixel image result in 128×128×17 features that are quantized to 8 bits. The structuring element sizes range from 3×3 pixel to 19×19 pixel. The extracted morphology features could be replaced by other features, but the maximum number of features per node is limited to 32 due to the width of the indexes (and due to the size of the shared RAM).

The number of graph nodes was maintained at 64, although the images are now smaller than those of for instance the XM2VTS database. The 19 quantized features per node are reduced to 3 features with PCA and/or LDA using fixed point arithmetic operations with saturation (see Subsection 6.5.3). The matching strategy will be described more in detail in Section 6.6.

Although it was observed in Subsection 4.4.1 that the Principal Component Analysis (PCA) or the Linear Discriminant Analysis (LDA) seemed not to improve results when using the XM2VTS database, this hypothesis was not tested on a smaller dataset, e.g. with 5 clients as described in the application scenario, and with more training samples. Nevertheless, as 3 principal components seemed to contain more than 80% of the cumulative percentage of the total variation, this selection of 3 reduced features seems reasonable.

Both the reference template and the PCA-LDA matrix transformation are stored in embedded SRAMs and can thus be updated whenever nec-

3. Even though their sequence is predictable.

essary (e.g. when new users are added to the database). Should the PCA reduction be bypassed, only the program memory would need to be changed. Finally, for other related applications (e.g. fingerprint matching as seen in Section 5.2), changing the instruction set might be easily achieved by following the ASIP implementation strategy described in Subsection 6.2.4, thus generating a new version of coprocessor.

6.5 Description of the coprocessor structure

It is not uncommon to report only the number of arithmetic operations as a measure of an algorithm complexity. Still, with multimedia applications where a large quantity of data is involved, the number of load and store operations together with the addressing scheme are of equal - if not of greater - importance.

In the EGM algorithm, the computational complexity being attributed to arithmetic operations comes mainly from the feature reduction and from the Euclidean distance calculation (see Subsection 6.5.3). During the elastic phase, the deformation penalty calculation involves also a relatively high number of multiplications, and the computation is moreover irregular as compared to feature reduction. Still, the graph addressing can be very time consuming on general purpose architectures, so that the arithmetic units may not be fed with feature vector elements at each clock cycle.

The addressing scheme is highly dependent on the system structure. By using a power-of-two image size, it is possible to construct addresses by simple index concatenation without resorting to costly multiplications and additions. While this could be realized on every general purpose architecture, the concatenation would require at least two OR operations to concatenate the x,y and level values, but probably also two shift and some move operations. On the other hand this concatenation can be done “for free” if the hardware is designed carefully.

Similarly, the addressing of all the graph nodes requires the memorization and the manipulation of the graph coordinates. Having these coordinates in a large single-port shared memory that is simultaneously containing the morphological features would clearly limit the data throughput. Moreover, frequent load-store operations to a global memory are power consuming, especially if the memory bandwidth must be increased to fetch enough data to feed the arithmetic units. Keeping

the coordinates in a small and local dedicated memory (or register bank) thus allows to reduce the power consumption. Additionally, the scanning process involved in the rigid and elastic matching can efficiently use the auto incrementation addressing schemes when the coordinates are stored in registers.

Finally, the efficiency of the EGM coprocessor addressing scheme is further enhanced by sharing the same adder among different operations, namely for:

- incrementing the node index;
- incrementing the level index;
- incrementing the node coordinates.

Clearly, an implementation on a general purpose processor would not be as optimal. On the other hand, this structure does not necessarily suit to applications that are not using graph structures, and the interest of realizing ASIP architectures is thus high.

For all the above-mentioned reasons, the coprocessor was divided explicitly into an addressing unit, a processing unit and a control unit, using dedicated hardware resources for each unit (Figure 6-6). Programming of these three units is achieved by using different fields of the same instruction, similarly to a VLIW processor.

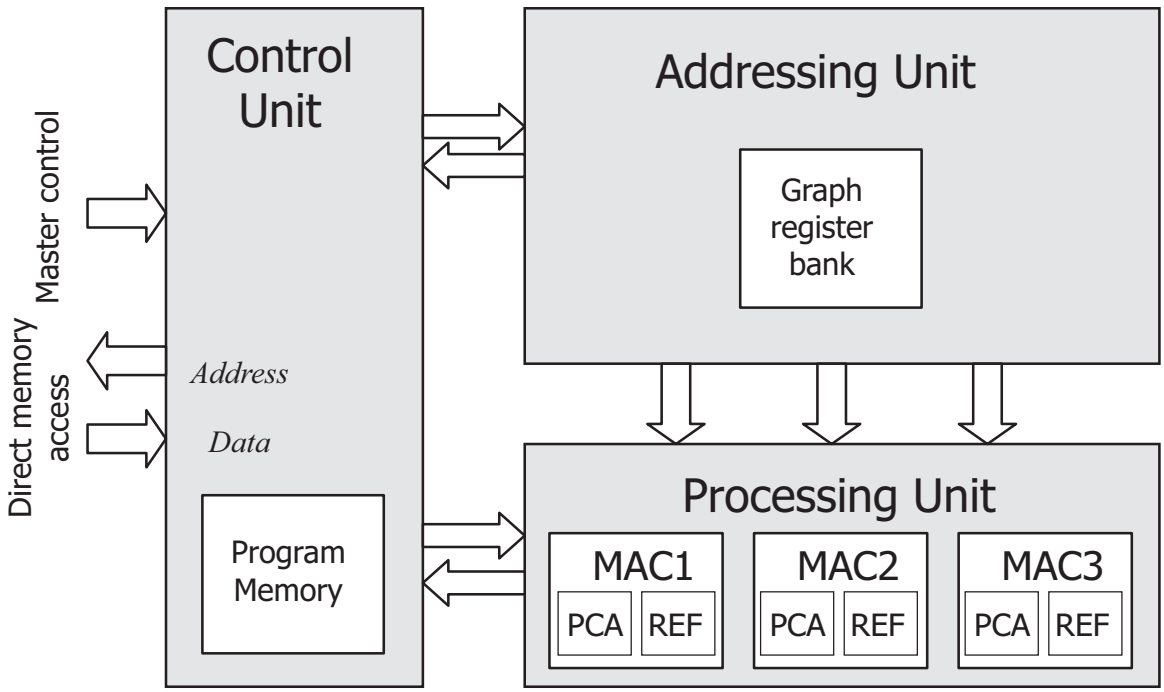


Figure 6-6: Structure of the EGM coprocessor.

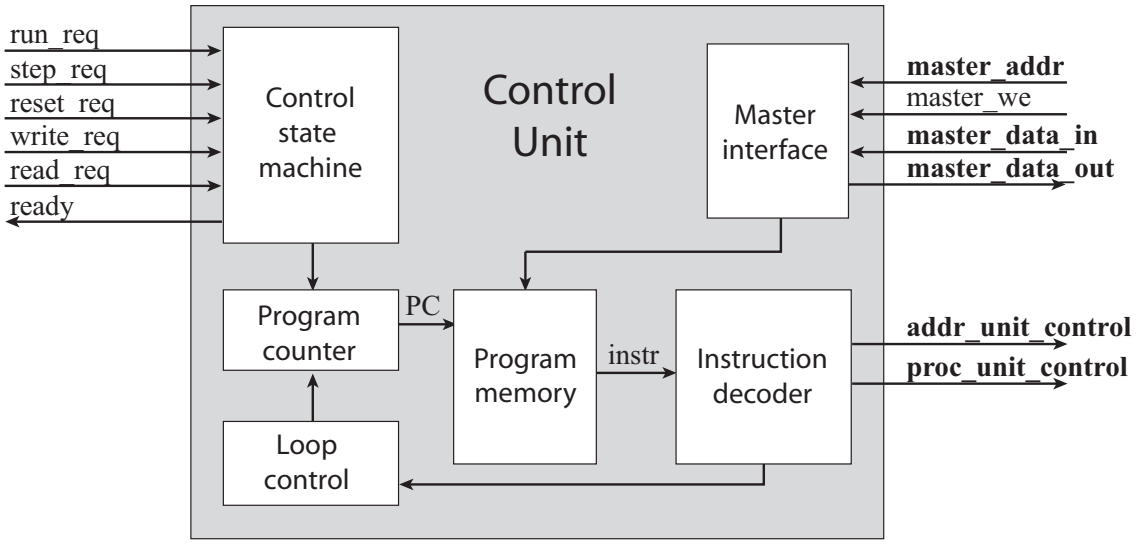


Figure 6-7: Control unit structure.

6.5.1 Control unit

The control unit is composed of a state machine (activated by master control unit queries), a master interface for decoding master read and write requests, and a program unit composed of a program counter (PC), memory and instruction decoder. A simple hardware loop support is also provided (Figure 6-7).

The main operational modes of the state machine are (Figure 6-8):

- read;
- write;
- reset;
- run (until a return instruction is reached);
- run one cycle[4].

The machine is in idle mode the rest of the time. The former states, or operational modes, are activated by five associated signals set by the master control unit. The ready signal of the EGM coprocessor (Figure 6-7) informs the master on its current state. The read and write operations are used to initialize internal registers and memories of the coprocessor and to read results such as the matching score of node positions of the best graph match.

4. This command is used in a “step-by-step” mode (i.e. in a debug mode), so as to trace the execution of the coprocessor.

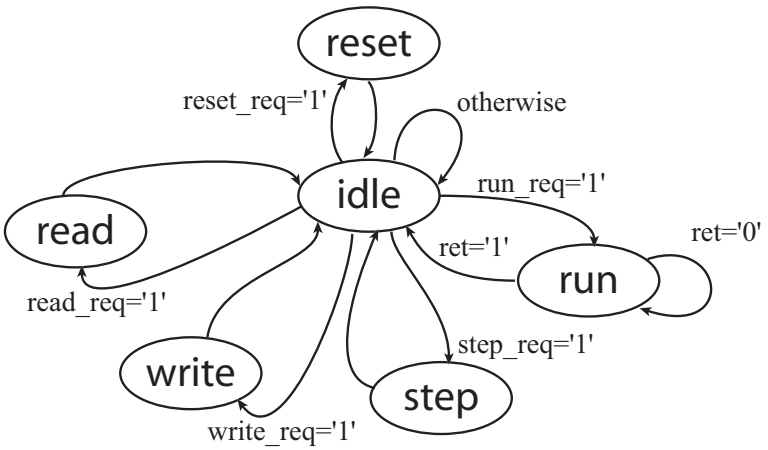


Figure 6-8: Coprocessor control state machine.

The programs (i.e. the sequences of instructions executed by the coprocessor) corresponding to the various operations to be performed are stored in the program memory at different locations. To launch one operation the MCU sets a correct starting address in the program counter (PC) and issues a run request. The coprocessor will go out of its idle state, execute the program until it reaches a return instruction, and goes back to the idle mode. The program memory can store up to 256 instructions (see Section 6.7 for details on the instruction set), although the whole EGM software uses currently only 143 instructions.

Before running one of the subprograms, the MCU will possibly require uploading data to the EGM coprocessor with write commands. For instance, the reference memories may need to be updated if the client is changed. In Simplex Matching, the new graph coordinates will also have to be uploaded, etc. After having run a program, the master will download useful information, i.e. the match score and sometimes the coordinates of the matched graph.

A two-stage pipeline allows to fetch and decode one instruction per cycle while all the computations are performed in the succeeding clock cycle (see Figure 6-9). The results are stored at the end of the cycle (rising edge at the beginning of the third cycle).

A hardware loop mechanism (see Figure 6-10) is provided that handles deterministic loops, i.e. loops executed a pre-defined number of times, unlike loops containing conditional branching. These loops are typical of the C++ *for* or *while* loops with a constant expression, e.g. *for(int i=0;i<64;i++)*.

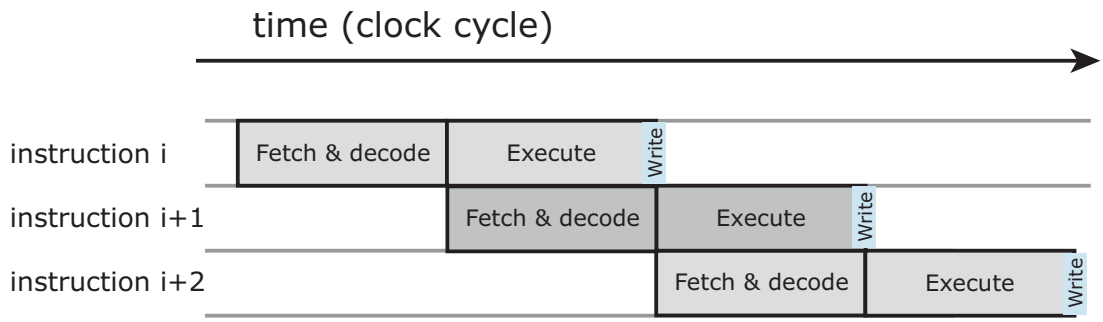


Figure 6-9: Two-stage pipeline.

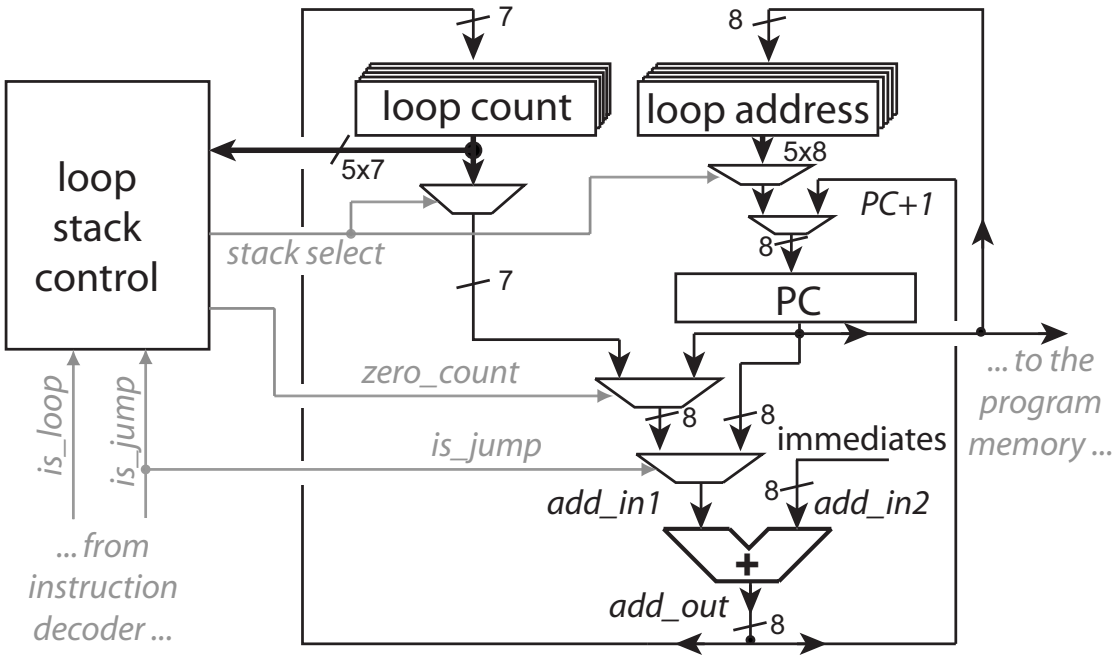


Figure 6-10: Loop stacks and program counter unit.

This unit is composed of a program counter (PC) register, containing the current address to fetch instructions from memory, and five loop count and loop address registers implementing the loop and jump assembly instructions. An adder is available and shared among two tasks: it serves both to decrement the currently selected loop count and to increment the PC.

The PC can take either the incremented value (PC+1), or one of the loop addresses contained in the registers (i.e. when the jump contained in the instruction is taken). The current PC can be used to store a loop address when a loop instruction is encountered (i.e. *is_loop* is raised).

The loop stack control selects the first pair of (loop count, loop address) registers that correspond to a non-zero loop count (the zero values indicates an empty stack register). The same control allows to store new count and address values to the next free register when a "loop" instruction is encountered. When the loop count reaches zero and a jump instruction is issued, the control selects the next non-empty pair of loop registers (i.e. the loop address is "unstacked"). Note that in the latter case, the next PC is taken instead of the stored loop address, although the *is_jump* control is raised.

A stack of 5 return address and loop count registers allows a maximum of 5 nested loops, which is largely sufficient for elastic graph matching subprograms. No hardware mechanism for stack overflow detection is provided so that the assembler (or at least the programmer) shall ensure that no more than 5 nested loops are indeed used. Loop counters are 7 bits wide, allowing a maximum loop count of 128, and the loop address is 8 bits wide.

A loop expression is composed of both a *loop* and a *jump* assembly instructions (see Figure 6-11). The *loop* instruction is a special instruction using 7 bits to store the loop count (see Section 6.7 for a description of the entire instruction set). The next PC value, i.e. amounting to (PC+1), is stored in the next available loop address register, and it will be used as the return address when a corresponding jump instruction is reached, if the loop counter is still larger than zero (otherwise PC+1 will be used as the next address).

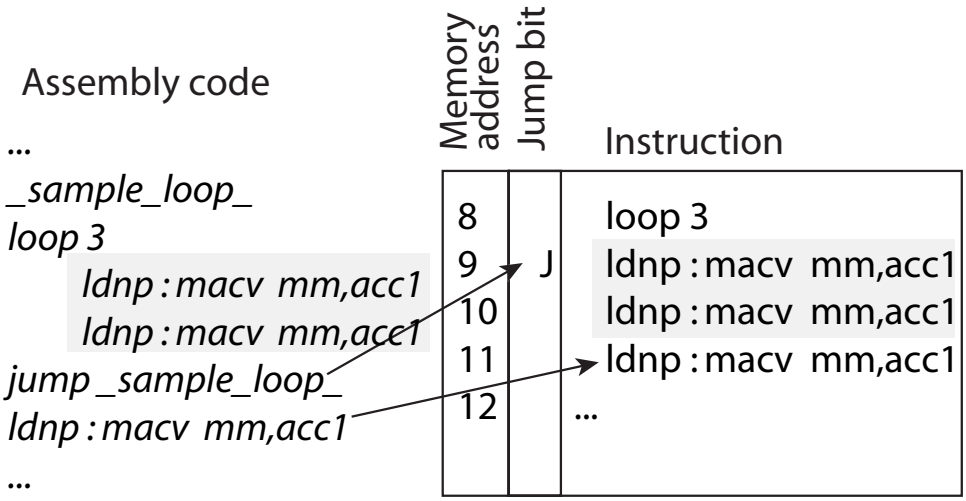


Figure 6-11: Loop and jump assembly instructions.
In this example, the gray-marked instructions will be iterated three times (note that the meaning of the *ldnp* and *macv* instructions is not important for this example; these instructions will be described in Section 6.7). The "loop" assembly code translates into a loop instruction located at the program memory location 8. The jump instruction does not translate into an opcode, but into a jump tag (J) in the second instruction before the actual (conditional) jump. In this example, this corresponds to the instruction located at address 9, because the actual (conditional) jump will occur once the execution of the instruction 10 is terminated, so that instruction 9 will then be executed instead of instruction 11 if the loop is selected. Instruction 11 corresponds to the instruction appearing after the jump in the assembly code. This loop construction is often used in order to avoid the insertion of a no-operation instruction: the single instruction between the loop and jump mnemonics is duplicated, and the loop count is divided by two (see also Figure 6-13).

A jump assembly instruction does not correspond to a real jump opcode in the program memory, but it is indicated by a single prefix bit present in all instructions (see also Section 6.7). This 1-bit extension of the instruction width induces an area penalty on the program memory, but it is largely compensated by the fact that no cycle penalty is incurred by a jump instruction or that no complex prediction mechanism is necessary.

A simple constraint on loop instructions is that the jump bit must be present in the instruction preceding the last instruction before the jump gets executed (see example in Figure 6-11), in order to use a single adder/subtracter for simultaneously decrementing the loop count and incrementing the program counter.

This also means that at least two instructions must be present between the loop and jump instructions otherwise no-operation (NOP) instructions must be inserted. However loop unrolling (i.e. duplicating the instruction or sequence of instructions, and dividing the loop count by 2) can be efficiently used so that this requirement is not too limiting (see Figure 6-13).

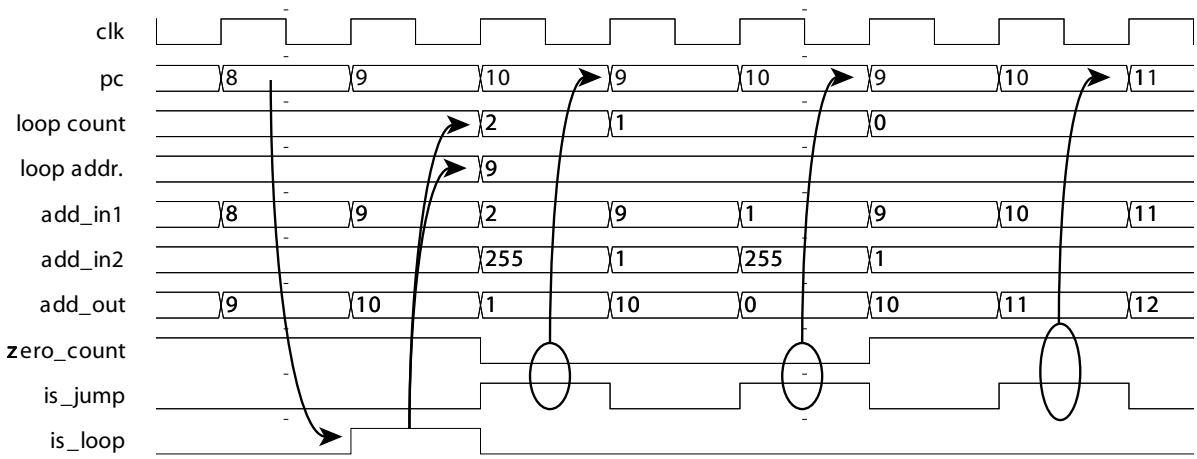


Figure 6-12: Typical waveform of the loop stack and PC unit.
The waveform corresponds to the assembly code and program instructions described in Figure 6-11. The control signals coming from the instruction decoder (i.e. *is_jump* and *is_loop*) arrive on this unit in the clock cycle following its decoding (*pc*+1) as they come out of a register (remember that the architecture uses a two-stage pipeline).
When *is_loop* is asserted, the loop count and loop address are stored. Then, each time *is_jump* is raised and *zero_count* is false, the address stored in *loop_addr* is taken as next *pc* value. Otherwise, PC+1 is taken.
Notes: the actual value stored in the loop count register is N-1, where N is the loop value specified in the assembly code (here N=3, thus *loop_count* initially receives the value 2 when *is_loop* is raised). This is because the loop count is stored during the first iteration and that decrementation will not occur for this iteration. This also indicates that each loop is at least executed once !

a)
```
...
_nop_loop_
loop 7
      ldnp : macv  mm,acc1
      nop
jump _nop_loop_
...
```

b)
```
...
_unrolled_loop_
loop 3
      ldnp : macv  mm,acc1
      ldnp : macv  mm,acc1
jump _unrolled_loop_
ldnp : macv  mm,acc1
...
```

Figure 6-13: Loop unrolling.
The two codes are equivalent from a functional point of view. In b), the loop unrolling permits to avoid the insertion of a no-operation (nop) instruction, therefore saving 7 cycles. Due to the odd number of iterations however, the unrolled code requires one more instruction after the jump instruction.

Note that the loop count and address registers are very rarely updated, which indicates that they are very good candidates for clock gating.

In conclusion, a one-cycle penalty exists only for the first iteration (i.e. storing the loop count and the return address in the stack), but not for each jump iteration being executed afterwards for the given loop. This mechanism showed to be very efficient for simple and very regular loops involved in the scanning process.

6.5.2 Addressing unit

A graph is composed of 64 nodes for which the horizontal (*x*) and vertical (*y*) pixel coordinates need to be stored. Seven bits are required to represent *x*, respectively *y*, since the features size is 128 by 128.

Two copies of the node coordinates need to be kept: one for the graph currently in use and one for the best graph found so far. When a new "best score" is found the "current" graph is copied in the "best" graph. This may be accomplished either for a single node (during elastic matching) or for all nodes (e.g. during rigid matching). Alternatively, the best graph may be copied into the current graph to re-initialize the current values (e.g. during elastic matching). The graph coordinates are used to build an address for fetching features. Interestingly, only the current graph is used for this purpose, while the best graph is used for storage only. Consequently, and referring to Figure 6-14, only one mux-tree is necessary on the "current" graph register bank (*cx* standing for the register bank containing *x*-oriented graph coordinates, and *cy* for the register bank storing the *y*-oriented graph coordinates), while the registers of the "best" graph bank (*bx* for bank containing *x* graph coordinates, respectively *by* for the bank containing *y* graph coordinates) are in a one-to-one correspondence linked to the related regis-

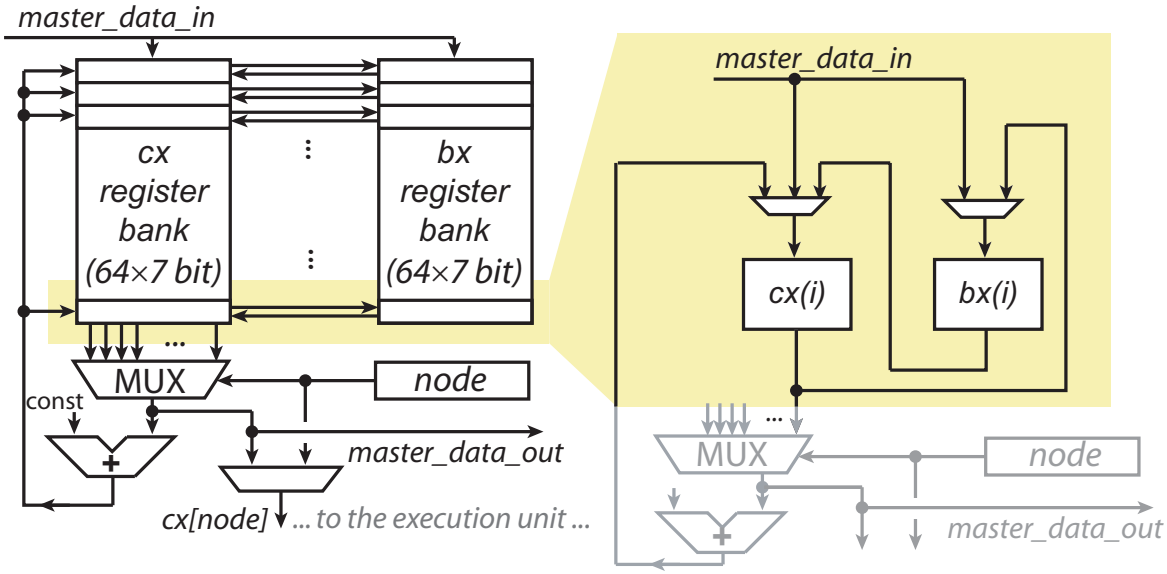


Figure 6-14: Structure of the node position register banks.
Left: structure of the "best" (*bx*) and "current" (*cx*) register banks allocated to *x*-oriented graph coordinates; right: internal structure of the *i*-th slice of the *bx* and *cx* banks (i taking values from 1 to 64). Note: the same structure is duplicated for cy and by, but the adder is shared among the two banks.

ters of the current graph bank. It should be noted that there exists additionally a write access from the MCU to the "best bank" and a read-write access to the "current bank". Consequently, reading the "best bank" involves first copying its values to the "current bank". This operation is executed in only one cycle.

An alternative to copying the registers would have been to use a pointer to select the current register bank between the two register banks. This would have indeed implied the modification of a single register (a pointer) instead of 128. However this would have required a second large mux-tree (or equivalently 128 additional 7 bit multiplexers) to allow an indexed access to the registers contained in *bx* or *by*, although node positions are copied from one bank to the other only a limited number of times (i.e. when a new minimum is found). The "best" coordinate registers have actually only a very small decoding tree since a register can be copied only to its corresponding "current" register, while the "current" registers involve a very large mux-tree for selecting the current node coordinates indexed by the node register (see Figure 6-14, right). Furthermore, as only one node can be saved during elastic matching, a single flag would not have been possible, but a stack of 64 flags would have been necessary.

Finally, an adder exists in this addressing unit, and it is being time-multiplexed to increment both *cx*, *cy*, and the node and level indexes.

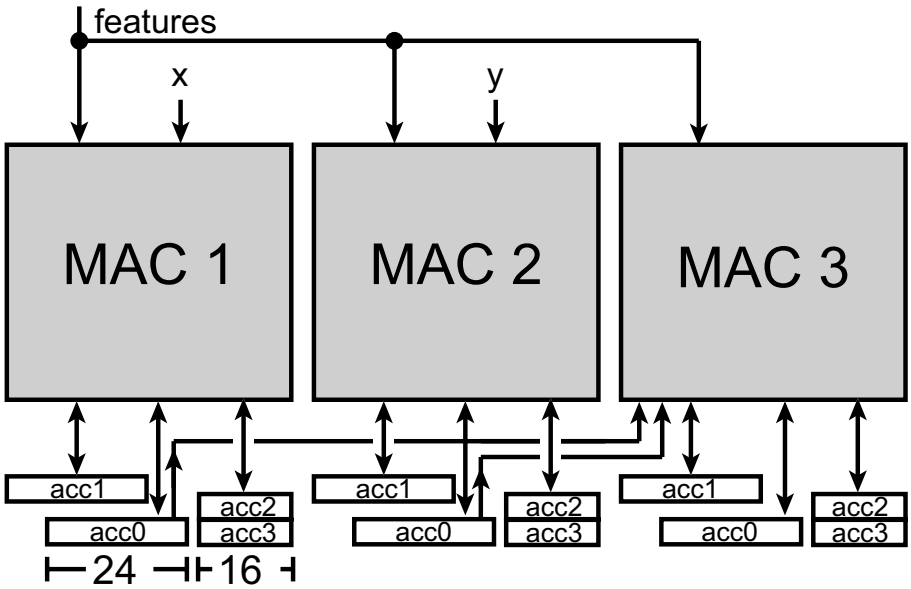


Figure 6-15: Structure of the processing unit.
Each multiply-accumulate (MAC) element possesses four registers, i.e. two 16-bit registers and two 24-bit registers, one of the latter being read-accessible by MAC3. The features are passed by the addressing unit to all the three MAC elements, but MAC3 can not read the *x* and *y* node positions received from the addressing unit.

6.5.3 Processing unit

The operations performed by the three MAC units embedded in the processing unit are:

- multiplication (signed *and* unsigned);
- squaring (special case of multiplication);
- multiplication-accumulation;
- addition;
- subtraction (incl. comparisons);
- absolute value.

The MAC unit is composed of 3 multiplier-accumulators (MAC) in parallel acting as a SIMD structure (Figure 6-15). The same operation can indeed be performed with different data on the three MACs, only on two of them, or only on one. These MACs are connected to two dedicated local memories and to four dedicated registers (Figure 6-15), called *acc0*, *acc1*, *acc2* and *acc3*. The output of *acc0* in MAC unit 1 and 2 is connected to the third MAC for sharing data so that data can be transferred from MAC1 to MAC3, and from MAC2 to MAC3. Note however that it is not possible to transfer data from MAC3 to MAC1, neither it is from MAC3 to MAC2.

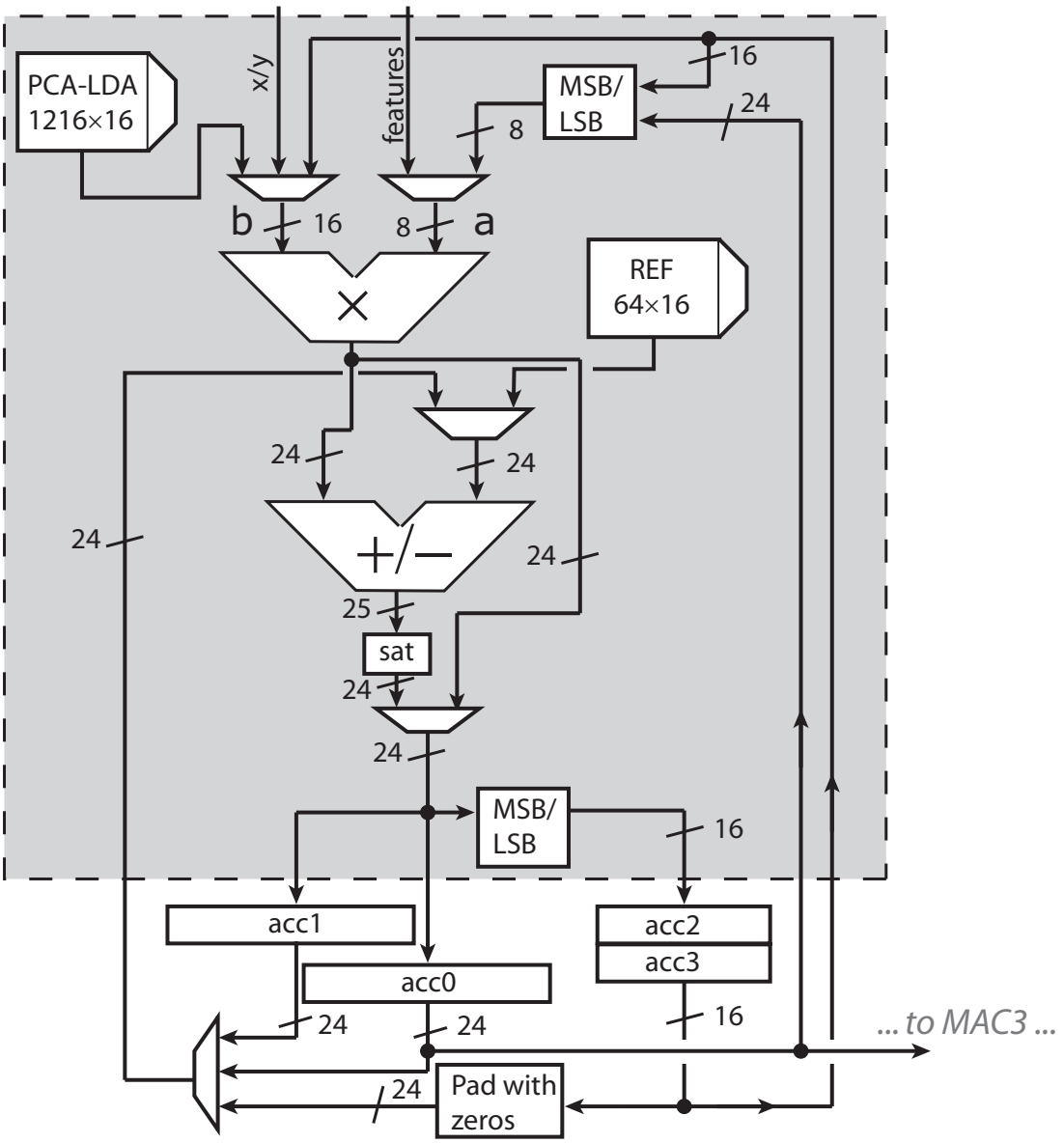


Figure 6-16: Internal structure of the multiply-accumulate element.
Structure of a MAC element comprising a multiplier, an adder-subtracter, two local memories (for storing the PCA-LDA coefficients and reference features) and a register bank. MSB/LSB: selection of the most significant bits (respectively less significant bits) from the input word. SAT: saturation of the input word if the most significant bit is 1 (unsigned case). The accumulators *acc2* and *acc3* are zero padded before being sent to the adder, and these zeros can be added either in front or at the end of the 16-bit word in order to form the 24-bit word.

Furthermore *acc0* and *acc1* are 24 bits wide, while *acc2* and *acc3* are 16 bits wide. Additional dedicated purpose registers (e.g. grid size, elasticity, score) are present but cannot be called explicitly.

The fixed-point multiplication is performing a 16×8 bit operation producing a result represented either over 24 bits or over 16 bits (including possible saturation), and with support of both signed and unsigned operands. To accommodate with this, some registers are 24 bit wide in or-

der to hold partial accumulations, while others are only 16 bit to store final results.

The motivation for using 16×8 multipliers comes from a careful investigation of the precision required in the most critical part of the processing (i.e. the feature reduction), while the precision is less critical in deformation calculation and Euclidean distance calculation (i.e. squaring), because saturation may occur there without prejudice[5].

The feature vectors are quantized on 8 bits in the morphological coprocessor. Clearly, varying the word length of the fixed-point PCA coefficients will lead to various degrees of precision. The PCA coefficients are signed and two bits are used to code its integer part. The fixed point format could thus be of the form 2.Q where Q is the number of bits of the fractional part. The precision obtained with Q={6,14,22} is reported in Table 6-1. Similarly, modifying the world length of the accumulator also influences the error as compared to a floating point implementation. For instance it is possible to accumulate the result on 24 bits (i.e. 2.22 format) and then to truncate this result to 16 bits at the end of the accumulation. This corresponds to the last but one line in Table 6-1. It can be observed that using 16 bit coefficients is sufficient, and that the truncation after accumulation does not incur a large additional penalty. In addition, depending on the type of operations and their size, some shift operations are sometimes performed to align 16 bit data with 24 bits.

Format of PCA coeffs	Size of result	Error variance
2.6 (8 bits)	16 bits	2.72
2.14 (16 bits)	24 bits	$4.1 \cdot 10^{-5}$
2.14 (16 bits)	16 bits[a]	$4.9 \cdot 10^{-5}$
2.22 (24 bits)	32 bits	$6.5 \cdot 10^{-10}$

Table 6-1: Error variance vs. size of PCA coefficient and of accumulator.
Reduction (i.e. multiplication-accumulation) of 17 feature elements of 8 bits each. Variance of the absolute difference between the floating point feature elements and the corresponding fixed point feature elements after accumulation. 100’000 random feature vectors were used.

a. A truncation from 24 to 16 bits is performed at the end of the accumulation.

5. As score equivalent to the saturated value would anyway be larger than the decision threshold, implying that the client will be anyway rejected. Therefore, a score larger than the saturation value would not modify the classification result. This would be a problem however, if a non-saturated overflow happened during the score accumulation, because it might lead to a falsely low score.

Considering the large number of operations potentially realizable with the structure described in Figure 6-18, a general instruction set would result in very large opcodes. Nonetheless, in the framework of ASIP and by making a careful use of implicit operands, the number of different operations can be drastically reduced.

a) Multiplier-accumulator structure

The critical path of the architecture is clearly located in the multiplier-accumulator datapath. The structure shall handle both signed and unsigned operands and shall perform the operation in one cycle. Sequential multipliers are thus rejected in favour of a combinational (sometimes called parallel-parallel) structure.

Modern synthetic libraries have at their disposal a wide range of structures (e.g. adder, adder trees, adder-subtractors, multipliers) that can be moreover automatically fine-tuned to the specific constraints of the design. It is either possible to infer an optimal multiply-accumulate structure from its behavioral description in VHDL or to instantiate it directly in the VHDL code. However, the MAC used here is special with the following respect:

- Possibility to perform a multiplication, a multiplication-accumulation, squaring, addition, subtraction, or absolute value;
- Usage of signed and unsigned operands.

Therefore some existing and specific solutions for multiplication-accumulation will be briefly discussed here and used to justify the choice of a MAC structure that was made. In the end, basic building blocks from a synthetic library will be used.

The multiplication of two integers basically consists in adding partial products, i.e. performing a logical *and* operation on the shifted versions of a multiplicand *a* and on the corresponding bits of the multiplier *b* (Figure 6-17). The adder-tree implements the summation of N partial products, where N is the width of the multiplier. This operation is performed efficiently by using *redundant recoding* so that the carry propagation of each full-adder can be calculated independently from the input carry [6-6]. Consequently, all the partial products of one column of the adder tree can be added in parallel and independently of other columns. So-called 3:2 compressors are used to perform the operation described in Eq. 6.3 (i.e. the sum of the three partial products *x*, *y* and *z* results in two intermediate values, ν_c and ν_s, and not directly in ν).

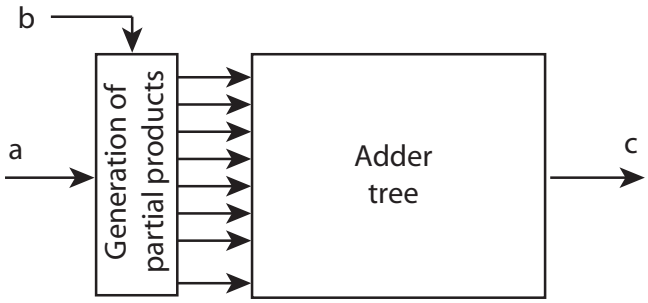


Figure 6-17: Basic multiplier structure.

$$x + y + z = v_c + v_s = v \qquad (6.3)$$

A final carry-save stage is thus necessary to get the final result. The resulting structure is called *Wallace tree* [6-9][6-6]. Additionally to the gain obtained from the lack of carry propagation it is possible to equalize the different data path lengths by taking into account the difference of propagation delay between the sum and carry outputs of the 3:2 compressor [6-16]. The Wallace tree structure contains approximately the same number of full-adders as the carry-propagate tree array but it is always faster although much more irregular. The reduced delay could be advantageously used to further reduce the supply voltage to achieve a given throughput. The gain in power consumption is thus important for VLSI implementation but may not be worthy for an FPGA implementation where dedicated hardware is present to build more regular carry propagation structures.

The partial products may be generated by performing a bit-wise logical *and* operation between the multiplier word and a multiplicand bit, and by shifting the result accordingly, or they can be recoded in a higher-radix. The so-called Booth encoding is a Radix-4 recoding that reduces the number of partial products by a factor of two [6-6]. Additionally, a modified Booth encoding may be used for signed numbers represented in two's-complement [6-6]. It is generally observed that Booth encoding improves the computation speed for word sizes larger than a certain threshold. In this work, we let the synthesizer decide upon the selection of Booth encoding since both the Booth encoded and the non-encoded partial product building blocks are present in the Synopsys DesignWare library. As it could be expected, the tool did not select Booth encoding for a 8x16 multiplier under our relatively loose timing constraints (i.e. a clock frequency of 10 MHz), but the synthesizer nevertheless implemented a Wallace tree. As a further optimization, the accumulation can be incorporated in the Wallace tree as another carry-save step.

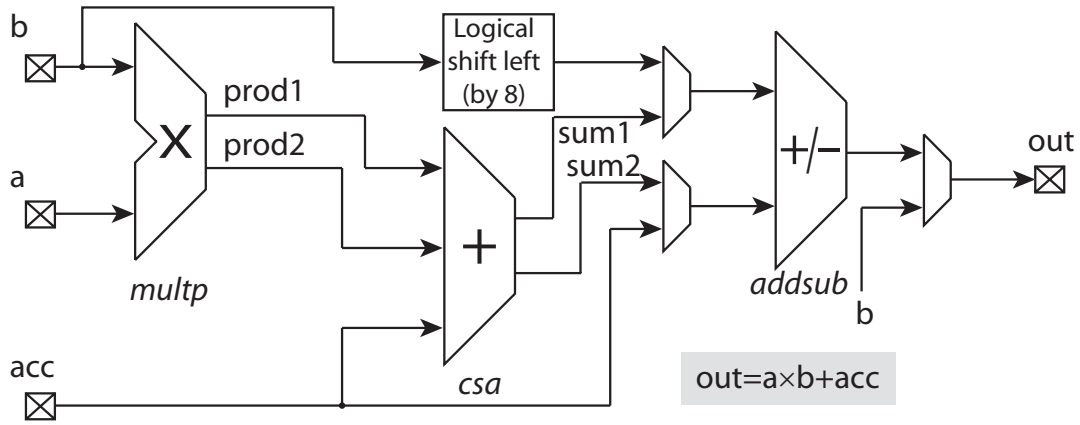


Figure 6-18: Final multiplier-accumulator structure.
Structure composed of instantiated version of Synopsys DesignWare partial product multiplier (multp), carry-save adder (csa) and adder-subtractor (addsub). The multp was replaced during synthesis by a non-Booth encoded Wallace tree and the addsub by a ripple-carry adder.

A final addition-subtraction stage is used to perform either an addition of operands *a* and *b* directly, or to add the two carry save results of the tree. This final addition is again performed by a slow ripple-carry adder, and not a fast carry look-ahead adder for instance due to the relatively loose timing constraints.

The multiplication of two numbers represented in two's-complement signed notation can be done with two different methods:

- Sign-extend both the multiplier and the multiplicand, i.e. replicate the most significant bit;
- Sign-extend the multiplier only (or the partial multiples) and perform a subtraction on the last partial multiple instead of an addition.

The second solution clearly requires less adders in the tree that sums the multiplicands and is thus more interesting.

All these considerations were used in the design to build the structure described in Figure 6-18 using DesignWare building blocks.

6.6 Coprocessor tasks

The three main tasks of the EGM coprocessor are:

- coarse rigid matching (scanning) at multiple scales;
- simplex downhill matching (see Subsection 4.5.1);
- elastic matching (see Subsection 4.5.2).

Each task corresponds to a distinct program, i.e. a list of instructions terminated by a special command signifying the end of the execution. To perform a full matching operation, the MCU will launch these three tasks in sequence as already illustrated in Figure 6-3. These tasks involve elementary tasks that are described in the next subsection. The specificities of the tasks are then discussed from Subsection 6.6.2 to Subsection 6.6.4.

6.6.1 Elementary tasks

Some basic operations are performed in all the programs mentioned above. Namely:

- fetching a feature element associated to a node (i.e. based on the node with image coordinates (x,y)) and feature element index;
- performing PCA/LDA transformation;
- calculating the Euclidean distance between a reduced reference feature vector and the reduced test feature vector.

These tasks are illustrated in Figure 6-19. Assuming that one feature vector element is fetched at each clock cycle, many calculations can be performed in parallel. Since we are calculating 3 reduced features out of the 17 original features (see Subsection 4.4.1) the 3-way data parallelism is exploited. Nevertheless it would be easy to fetch more than one feature per cycle, e.g. by fetching multiple levels corresponding to a single (x,y) coordinates pair. In the SoC described in Section 6.3 only one shared memory was used. This memory delivers eight bits of data at each clock cycle at an operating frequency of 10 MHz. For faster memory accesses it could be possible to build a bank of memories and to fetch simultaneously multiple data - one from each memory - from the same address. The requirement for this would be that the features can be stored the same way, which is indeed possible. This scheme was not necessary here since fetching a single data and sending it to all the computational units guarantees a sufficient throughput.

6.6.2 Rigid matching program

The rigid matching processing consists in scanning the image with a rigid version of the graph. This program is generally executed before the Simplex Downhill matching in order to find a starting point for the iterative matching, when no face detection is available. In the rigid matching program, a graph distance is calculated using the procedure

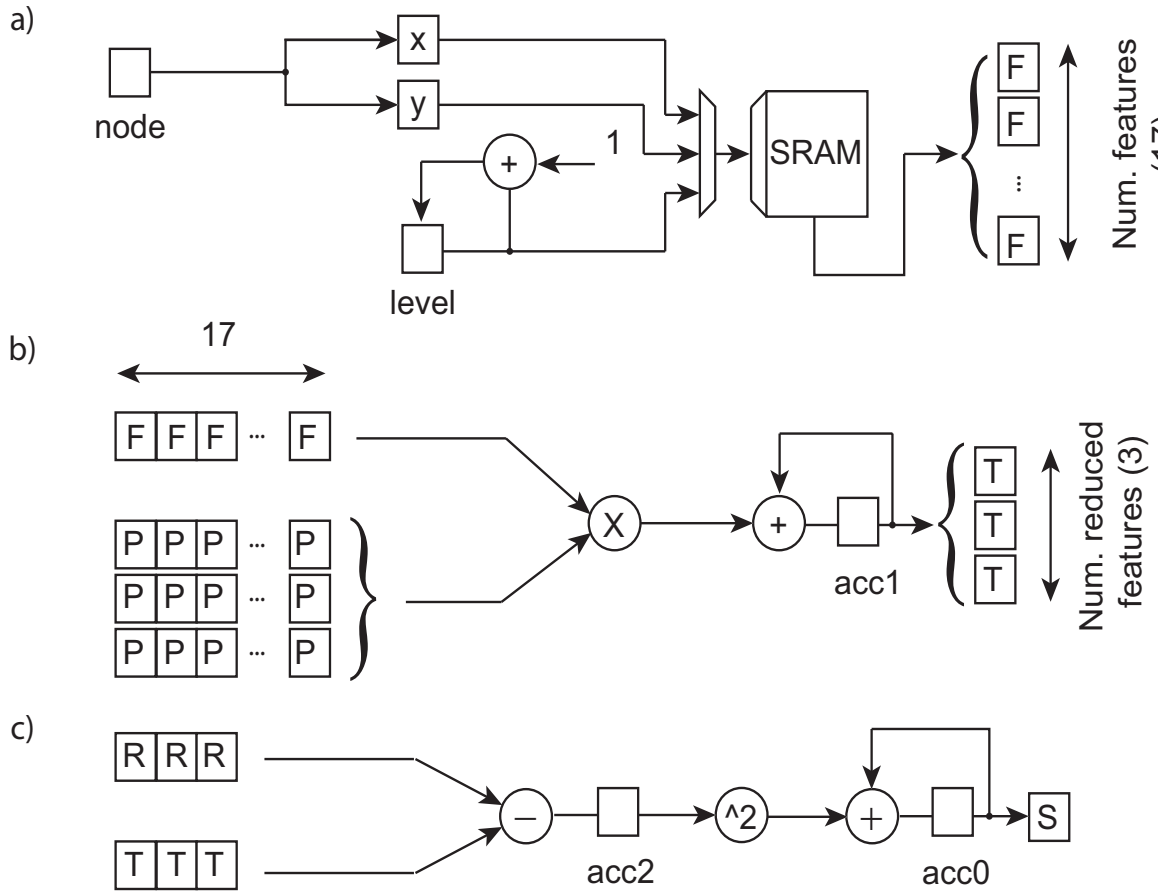


Figure 6-19: Elementary feature processing operations.
a) feature vector elements *F* are fetched from the external memory (SRAM) based on *level* and *x* and *y* coordinates, which are fetched from the *cx* and *cy* register banks using the *node* index; the *level* index is incremented 17 times, while the *x* and *y* coordinates remain constant;
b) 17 feature vector elements *F* associated to a single *node* index are now reduced to 3 feature elements *T* using a 3×17 PCA-LDA matrix *P*;
c) 3 test feature elements *T* are compared to 3 reference features *R* using an Euclidean metric (i.e. a subtraction, a squaring, and an accumulation), the result being the node score *S*.

described in the preceding subsection, and a copy of the current node position registers is placed in the best node position registers, whenever the new distance is smaller than the previous best distance (the best distance is of course also updated). Once the Euclidean distance of a single node is calculated (i.e. the whole feature vector associated to a node has been fetched from the shared RAM, reduced and the distance has been calculated and stored) the current node position register is automatically incremented from a pre-defined value (e.g. 1, 2, 3, etc.; depending on the granularity of the rigid matching). End of lines are handled differently (i.e. a different constant is added to the node position so that it is re-positioned at the beginning of the next line). Four nested loops are necessary to scan the image horizontally and vertically, for all nodes and for all feature elements in the vector.

6.6.3 Simplex downhill matching program

Once a starting position has been selected with the coarse rigid matching, the EGM coprocessor executes a series of simplex iterations. The MCU writes a new graph configuration in the current node position registers and launches the corresponding EGM coprocessor program, which returns the associated graph distance. Based on this distance, the master defines a new graph by altering the translation, scale and rotation parameters of the current graph. In this case, the post incrementation of the graph node positions by the EGM coprocessor is not used. The processing is otherwise exactly the same as for rigid matching.

The MCU has the possibility to calculate the next graph configuration parallelly to the calculation of the current distance by the EGM coprocessor. Essentially, the selection of the next graph position can be represented by a binary tree, where the decision between two branches is dictated by the current distance, which is either better or worse than the previous distance. The two branches (i.e. the next graph configurations) corresponding to these two possible outcomes can be either both pre-calculated or a single outcome can be pre-calculated. In the former case, one outcome is used, while the other is *always* trashed. In the latter case, the calculated outcome is sometimes used but *sometimes* trashed, if the wrong outcome was pre-calculated. It would be possible to increase the probability of pre-calculating the correct outcome by using a prediction mechanism. This prediction is however costly and it is not guaranteed that an actual gain can be achieved.

Theoretically it would be even possible to write new graph configurations to the “best” graph registers concurrently to running EGM coprocessor, since the latter is not using these registers in this case. However this solution is currently not yet implemented, because the slow communication speed between the two units results in an overall data communication time that is longer than the processing time.

6.6.4 Elastic matching program

The elastic phase starts with the calculation of the Euclidean distance between reference and test feature vectors, but using a node-by-node approach. An additional deformation penalty is then calculated and added to the features distance, while displacing the considered node. This displacement is limited to a window around its best current position. Firstly, the node index is displaced to the top-left position of the

window. After having calculated the sum of the feature distance and deformation penalty for this position (and possibly updated the corresponding best node score and best node position), the graph node is post-incremented by one position to the right. Again, the last position on the right is handled differently so as to return to the beginning of next line of the window. The deformation penalty is obtained according to Eq. 4.63.

The squared horizontal and vertical rigid distance between node positions are stored in the MAC unit registers and are updated after simplex matching depending on the size of the rigid graph.

As already mentioned, nodes are displaced one after the other and the best displacement is always stored in the best node position register bank. Obviously, taking all the 64 nodes in a sequential order is not optimal. In Chapter 4, the nodes were selected using a random order (e.g. generated from the libc “rand” command). The generation of high quality random numbers is however not necessary, but a small degree of randomness is still desirable. Therefore, a very simple Pseudo-Random Number Generator (PRNG) was embedded in the coprocessor. This PRNG is simply a shift-register with exclusive-OR (XOR) gates. Each time the value is shifted, the output of the register is modified and delivers a new value (see Figure 6-20). Two combinations are possible for the connection of the XORs, and the initialization of the shift-registers can be modified. This leads to a small number of different sequences of node indexes, but is still sufficient for this purpose. By shifting the register 63 times, all the node indexes appear exactly once, except the value 0 which for obvious reasons cannot occur. However, the node index can be set to 0 with the `resn` instruction to move this particular node.

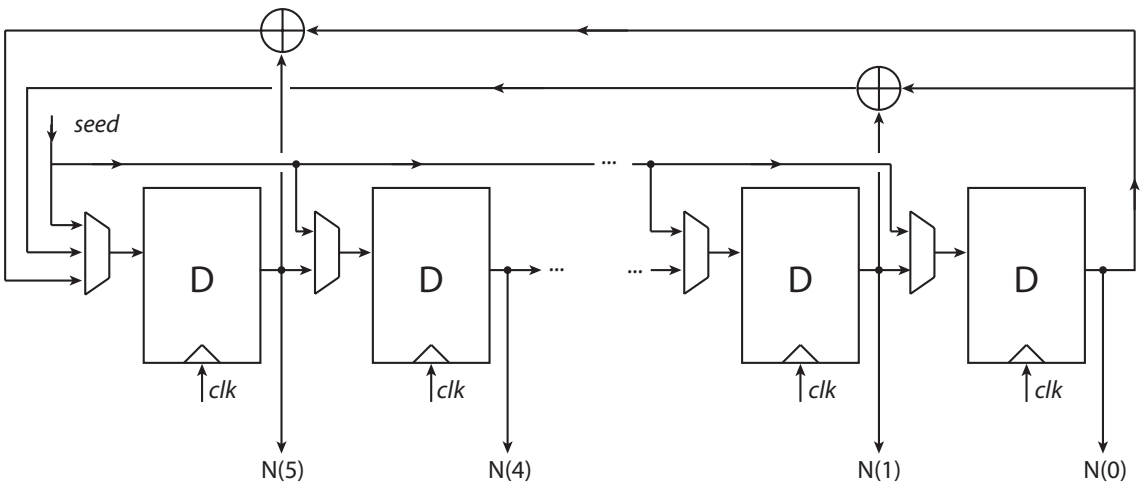


Figure 6-20: Simple pseudo-random number generator.

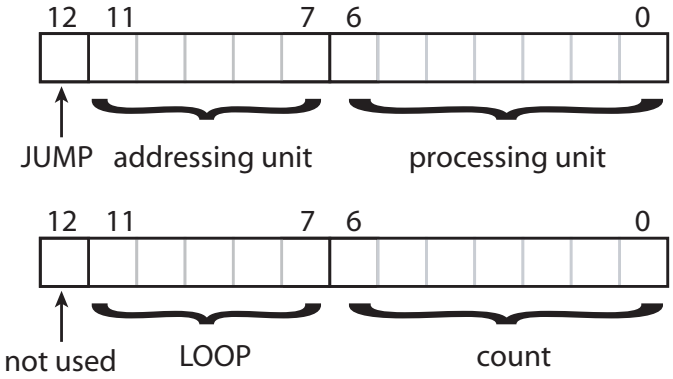


Figure 6-21: EGM coprocessor instructions.
The 13 bit instruction word contains a jump bit and two fields, which can be used to encode the operations of the addressing and processing units, or to encode a LOOP operation. In the latter case, the second field encodes the number of times the loop must be taken (count).

6.7 Instruction set

Each EGM coprocessor instruction is 13 bit wide (see Figure 6-21). Bit 12 corresponds to a *jump bit* (Subsection 6.5.1), bits 11 down to 7 encode a load-store instruction and bits 6 down to 0 encode a MAC instruction, which contains up to three SIMD operations.

Moreover two special instructions exist:

- LOOP: encoded as "`011101LLLLLLL`," where L represents the loop count. Since this instruction is 12 bits wide, it corresponds implicitly to a NOP for the LSU and MAC units;
- RET: encoded as "`011100`∅∅∅∅∅∅," where ∅ represents a "don't care" value (i.e. it can be either 0 or 1).

Associated to a loop instruction, a label is embraced with "_" characters (e.g. _mylabel_) in the assembly code. The corresponding name should be used in the matching jump instruction (for an example see the code excerpt listed in Table 6-2). This is useful to help reading the assembly code, although the assembler does not currently check for matching labels.

In addition, comments are present in the assembly source code but are not translated into opcodes. Comments start with a ";" symbol and end up at the end of the line. Further available instructions are described in Table 6-3 (load-store unit instructions) and Table 6-4 (multiply-accumulate unit instructions).

Table 6-2 is a sample of assembly code that performs the calculation of the Euclidean distance between reference and test feature vectors associated to the 64 nodes of a graph i.e. the operations described in Figure 6-19.

```
resn : resa acc0  ; reset the accumulator
resl : ld_ref     ; load the first reference
                  ; and reset the level index

_nodes_1_to_64_   ; example of label
loop 64

   ; begin PCA transform
   ldnp : ld_pca
   ldnp : multv mm,acc1

   loop 7
      ldnp : macv mm,acc1
      ldnp : macv mm,acc1
   jump _7_

   ldnp    : macv mm,acc1
   inccx 0 : macv mm,acc1  ; displace node position

   ; subtract reference feature to test feature
   ; and square and add the result
   incn    : subvref acc1, acc2
   nop     : ld_ref
   resl    : asquarv_hi acc2,acc0

jump _nodes_1_to_64_

resn : add3 1,acc0
nop  : add3 2,acc0        ; the score is finally in acc0

nop  : compscv acc0,score ; compare and update
```

Table 6-2: Sample of assembly code.

Note how the loop is unrolled: two `ldnp` instructions inside the seven iteration loop, plus three instructions outside achieve the 17 feature vector fetch from the external RAM. This prevents from inserting a `NOP` inside the loop to respect the two-instruction loop constraint induced by the pipelined execution of the instructions.

Mnemonic	Operation(s)	Binary instruction [11:7]	Argument
nop	None	**00**∅∅∅	-
ldnp	mm←EXT[L | CY[N] | CX[N]]; L++; P←PCA[L | N]	**0100**∅	-
incn	N++	**0101**∅	-
inccx *i*	CX[N]+=IMM[*i*]	**100**ii	*i*={0,1,2,3}
inccy *i*	CY[N]+=IMM[*i*]	**101**ii	*i*={0,1,2,3}
movc	X=CX[N]; Y=CY[N]	**11110**	-
movcp *i*	X=CX[N+*imm*]; Y=CY[N+*imm*]	**0110**i	*i*={1,8}
movcn *i*	X=CX[N-*imm*]; Y=CY[N-*imm*]	**0111**i	*i*={1,8}
resn	N=0	**11001**	-
resl	L=0	**11000**	-
movbc	CX=BX; CY=BY	**11010**	-
randn	RAND=RAND>>1; N=RAND(0)	11011	-

Table 6-3: Addressing unit instructions.
The opcode is represented as bold font weight in the binary instruction (operands are in regular font weight). The symbol "∅"denotes a don't care bit; "←" indicates a load operation from the memory specified on its right; "[]" denotes an indirect addressing, where the value within brackets is the index and the value on the left of the brackets is the indexed array (register bank or memory); "|" denotes the concatenation of the values on the left and on the right; "++" means a post-increment by one of the value on the left of the symbol and "+=" a post-increment with the value on the right; ";" separates several operations performed in the same instruction; EXT means the external shared memory; ACC*i* means the accumulator bank of the i-th MAC unit; BX (resp. BY) means the best node horizontal (resp. vertical) position register bank; CX (resp. CY) means the current node horizontal (resp. vertical) position register bank; The following are special registers: N is the node index; L is the feature level index; P*i* (resp. R*i*) is the registered output of the PCA (resp. REF) memory in the *i*-th MAC unit. IMM is a bank of four immediates; X is a register input to MAC1; Y is a register input to MAC2; and RAND is the pseudo-random shift register.

Mnemonic	Operation(s)	Binary instruction [6:0]	MAC used	Argument 1 (*i*)	Argument 2 (*j*)
nop	None	**000000**∅	-	-	-
multv *i*	ACC[*i*]=MM×PCA$^{(+/-)}$	**1001**∅ii	all	{0,1,2,3}	-
macv *i*	ACC[*i*]=ACC[*i*]+MM×PCA$^{(+/-)}$	**1000**∅ii	all	{0,1,2,3}	-
asquarv_hi *i*, *j*	ACC[*j*]=msb(ACC[*i*]$^{(+/-)}$)×ACC[*i*]$^{(+/-)}$	**101**∅ijj	all	{2,3}	{0,1,2,3}[a]
squar12_lo *i*, *j*	ACC[*j*]=lsb(ACC[*i*]$^{(+/-)}$)×ACC[*i*]$^{(+/-)}$	**1100**ijj	{1,2}	{2,3}	{0,1,2,3}[a]
addabs3	ACC[*j*]=ACC[*j*]+abs(ACC[*i*]$^{(+/-)}$)	**1101**ijj	{3}	{2,3}	{0,1,2,3}
mac3 *i*, *j*	ACC[*j*]=ACC[*j*]+ACC[*i*]×ELAST	**111**iijj	{3}	{0,1,2,3}	{0,1,2,3}[a]
subvref *i*, *j*	ACC[*j*]=ACC[*i*]-R; R←REF[N]	**010**iijj	all	{0,1,2,3}	{0,1,2,3}[b]
subvpos *i*, *j*	ACC^{1}[*j*]=ACC[*i*]-X; ACC^{2}[*j*]=ACC[*i*]-Y	**011**iijj	{1,2}	{0,1,2,3}	{0,1,2,3}
add12 *i*	ACC^{3}[*i*]=ACC^{1}[0]+ACC^{2}[0]	**00110**ii	{3}	{0,1,2,3}	-

Table 6-4: MAC instructions (part 1).
See also symbols defined in Table 6-3. Additional symbols: "×" denotes the multiplication; a "(+/-)" superscript denotes a signed operand (operands are unsigned by default); "msb" means the 16 most significant bits of the 24 bit register enclosed in the parentheses, while "lsb" means the 16 least significant bits of the register; "abs" means the absolute value of the number in parentheses; ACC^{n} means that the operation is performed only on the *n*-th MAC unit; the symbol "<,>$_{n}$" means the selection of one parameter in the list, based on the value of *n* (i.e. the first parameter when *n*=0, the second when *n*=1, etc.); ELAST is a dedicated register containing the elasticity coefficient; GX and GY are dedicated registers containing the horizontal and vertical distance between graph nodes in the rigid graph; SCORE is a dedicated register containing the current best score.

a. If ACC[*j*] is 16 bit wide, only the MSB part of the result is stored.
b. If ACC[*i*] is 24 bit wide, only the MSB are used.

Mnemonic	Operation(s)	Binary instruction [6:0]	MAC used	Argument 1 (*i*)	Argument 2 (*j*)
sub3 *i*, *j*	$ACC^3[i]=ACC^3[i] - <GX,GY>_j$	**00111**ij	{3}	{2,3}	{0,1}
add3 *i*, *j*	$ACC^3[j]=ACC^3[i]+ACC^j[0]$	**0010**iij	{3}	{0,1}	{1,2,3}
compsc	If $ACC^3[0]$<SCORE Then BX[N]=CX[N]; BY[N]=CY[N]; SCORE=$ACC^3[0]$	**000010**∅	{3}	-	-
compscv	If $ACC^3[0]$<SCORE Then BX=CX; BY=CY; SCORE=$ACC^3[0]$	**000011**∅	{3}	-	-
set_sc	SCORE=0xFFFFFF	**000001**∅	{3}	-	-
movxy *i*	$ACC^1[i]$=CX[N]; $ACC^2[i]$=CY[N]	**000100**i	{1,2}	{2,3}	-
ld_pca	P←PCA[L | N]	**000110**∅	all	-	-
ld_ref	R←REF[N]	**000111**∅	all	-	-
resa *i*	ACC[*i*]=0	**000101**i	all	{0,1}	-

Table 6-5: MAC instructions (part 2).
See symbol definitions in Table 6-3 and Table 6-4.

6.8 Design analysis and performance figures

The EGM coprocessor was described and simulated at register-transfer-level (RTL) using the VHDL language. The same description was then used to synthesize an FPGA version and an ASIC version. The embedded memories are however target dependent, and consequently slight modifications of the description were necessary to interface these memories.

6.8.1 VHDL description

The RTL description was hierarchically organized so that leaf blocks could be tested individually. The FPGA version contains a USB and/or Ethernet interface, which is not present in the ASIC version. The only difference between the VHDL description to be synthesized in the FPGA flow from that synthesized in the ASIC flow is due to the utilization of custom SRAM memories. Indeed, the FPGA memories are provided by the “genmem” tool of Altera, while the memories for the ASIC flow were generated with a custom tool provided by UMC (see also Subsection 6.8.3).

For the FPGA version a top test bench was developed to simulate USB commands, while for the ASIC version the top test bench was developed without these interfaces. This allowed however in both cases to simulate an entire matching of a test image against a reference template, and to extract switching activities for power estimation.

6.8.2 FPGA implementation

The VHDL description was firstly synthesized into a netlist of standard FPGA cells using Synopsys Design Compiler (dc_shell) as a “front end” to other FPGA specific Synopsys tools (e.g. FPGA Compiler). The resulting EDIF (Electronic Design Interchange Format) netlist was then transferred to the Altera Quartus II tool, which performed the final place and route, and a timing analysis of the resulting design. Another solution could have been to synthesize the design from VHDL directly in Quartus II. However, as an ASIC version was foreseen, reusing the same front end and synthesis tool simplified the work.

The targeted FPGA device is an Altera APEX 20K600E-3 whose characteristics are reported in Table 6-6. The synthesis was constrained for a clock signal operating at 10 MHz[6] and for minimum area. The relative area occupation of the device is reported in Table 6-7.

Typical # gates	# Logical Elements	# Embedded System Blocks	Maximum # RAM bits	Maximum # user I/O pins
600'000	24'320	152	311'296	588

Table 6-6: Altera APEX 20K600E-3 FPGA characteristics.

Component	Area (in logical elements)	Percentage of available area	Memory use (in RAM bits)
Clock division + glue logic	5	<0.1	
USB / Ethernet interface	653	2.7	
VGACAM interface	280	1.1	
External SRAM interface	148	0.6	
Morphology coprocessor	3743	15.4	12'816
EGM coprocessor	7162	29.4	64'768
- glue logic	*30*		
- control	*852*		*3'328*
- load-store unit	*2965*		
- mac unit	*3300*		*61'440*
- pseudo-random unit	*15*		
Total	**11'989**	**49.3**	**77'584**

Table 6-7: Relative area occupation of the components on the APEX 20K600E-3.

With a critical path of 72.5 ns in the morphology coprocessor, the slack time is 27.5 ns. The operating frequency could thus be slightly increased. However the clock division is much simpler when using integer division factors.

6. The on board FPGA clock signal operates at 40 MHz and is divided inside the FPGA device. USB communication is performed at this maximum frequency of 40 MHz and converted to synchronous signals operating at 10 MHz that are delivered to the coprocessor.

Program	Simulated time[a] [ms]	Measured time[b] [ms]	Overhead [%]
Morphology	200	208	5.0
EGM	460	1'128	
- Rigid matching 5x9[c]	*120*	*128*	*6.7*
- Rigid matching 6x10[c]	*90*	*115*	*27.8*
- Rigid matching 7x11[c]	*80*	*102*	*27.5*
- Simplex matching[d]	*23*	*543*	*>2000*
- Elastic matching[e]	*160*	*240*	*50.0*
Total	**660**	***~1'400***	

Table 6-8: Simulated and measured execution times.

a. Execution time on the respective coprocessor, *without* the communication with the MCU.
b. Execution time including communication with the MCU via the Ethernet interface; averaged over ten measurements using the Win32 *QueryPerformanceCounter* function. This value is subject to variations of the Ethernet network load.
c. Using a stride of 3 pixels horizontally and vertically.
d. Using 150 simplex iterations.
e. Four full iterations (64 nodes displaced in a 11 by 11 window each time).

The simulated and measured execution times are reported in Table 6-8. The discrepancy between the two values is due to the communication overhead between the master control unit (MCU) and the morphology and/or EGM coprocessor. This communication consists at minimum in sending run commands to the coprocessors, but it can also take place to initialize registers and memories or to read back results calculated on the coprocessor.

As can be seen in Table 6-8 most of the time is spent in the EGM coprocessor, which in addition requires a greater number of transactions between itself and the MCU. Consequently the average overhead incurred by the Ethernet communication is much higher for this coprocessor than for the morphology coprocessor.

For instance, running the morphology coprocessor is a task requiring very few control transactions: namely a run request, and one or two read requests to poll the coprocessor for the completion of the operation[7]. The overhead is thus only 5% in this case. However for most of

7. This could be suppressed if the MCU would support interrupts.

the EGM programs this overhead is larger. For instance in rigid matching, beside the run command the graph must be initialized and read at the end of the processing. The largest overhead is found in the Simplex matching program, where for each iteration a graph is sent and a score is read. The large number of iterations and the small execution time of a single iteration lead to a very large overhead.

Still, the communications between the master control unit (MCU) and coprocessor (through Ethernet) are asynchronous and must be synchronized on the FPGA. This requires additional wait states which are of course more costly at frequencies as low as 10 MHz. In addition, although the bandwidth of Ethernet is rather high, its latency is non negligible. As a conclusion, not only the size of the transactions but also their number is important to minimize this overhead.

Assuming synchronous communications between the MCU and the coprocessors (with possible wait states of one or two cycles) would dramatically reduce the overhead reported in Table 6-8 down to around 20%, i.e. meaning a total execution time below one second. This corresponds indeed to the requirements specified in Section 6.1.

6.8.3 ASIC implementation

Similarly to the FPGA implementation, the VHDL description was first synthesized into a netlist of standard cells using Synopsys Design Compiler (dc_shell). The target VLSI process is a 0.18 μm with 6 metal layers from UMC [6-32] and the *umcl18u250t2* standard cell library was supplied by Virtual Silicon Technologies [6-33].

The synthesis constraints were as following:

- clock signal frequency of 10 MHz, with clock skew of 2 ns;
- operating conditions: typical (25°C, Vdd=1.8 V);
- wire load model: default;
- minimum area.

The synthesis resulted in a total area of around 1.28 mm^2 (see Table 6-9 for the detail per component). Interconnect area was not estimated because routing was not realized. Seemingly, the major area occupation is due to the memories (around 65% of the total area) and thus, logically, the MAC units containing the more memories use most of the area. Beside memories, registers take also around 19% of the circuit area[8] and will consequently also have an impact on the power figures.

Component	Non-combinatorial area[a] [μm^2]	Combinatorial area [μm^2]	Total area [μm^2]
Control	103'384	12'006	115'390
- interface	*5'618*	*3'781*	*9'399*
- instruction decoder	*86'570*	*2'143*	*88'713*
- *ram (PM)*			*79'610*
- *decoder*			*8'891*
- *others*			*212*
- pc and loop stack	*11'196*	*6'082*	*17'278*
Addressing unit	192'556	86'336	278'892
- interface	*10'416*	*15'400*	*25'816*
- decoder	*0*	*1'545*	*1'545*
- mux (each)	*0*	*10'756*	*10'756*
- regbanks (each)	*91'070*	*23'935*	*115'005*
- others	*0*	*9*	*9*
Mac unit	782'466	101'043	883'509
- interface	*0*	*3'455*	*3'455*
- unit 1 or 2 (each)	*259'044*	*30'551*	*289'595*
- *regs*			*25'408*
- *ram (PCA)*			*202'206*
- *ram (REF)*			*47'016*
- *mac*			*14'940*
- *others*			*25*
- unit 3	*264'378*	*36'486*	*300'864*
- *regs*			*36'686*
- *ram (PCA)*			*202'206*
- *ram (REF)*			*47'016*
- *mac*			*14'924*
- *others*			*32*
Pseudo-random unit	606	606	1'212
Total	**1'079'012**	**199'991**	**1'279'003**

Table 6-9: Area of synthesized ASIC cells.
Indication of cell area only, without interconnection network.

a. This value includes the area for register *and* memory.

Compared to the 100 ns cycle length, the critical path lasts only 11.6 ns resulting in a low 11.6% duty cycle. It would be thus possible to run the processor with a clock frequency of 80 MHz without re-synthesizing the

8. This number can be obtained by subtracting the memory area to the total non-combinatorial area.

coprocessor unit, or possibly even at higher pace by applying tighter synthesis constraints. While the power consumption would be increased by roughly a factor of 8, the necessary energy for a single face verification would not change.

Power-estimation with Synopsys Power Compiler was performed with the “typical” library after back-annotating the activity of the netlist in ModelSim using a test bench reproducing the stimuli of the rigid matching phase (see Subsection 6.6.2). The dynamic power consumption reaches 5.5 mW, while the static power consumption (leakage) is only 1.3 µW. The detail of dynamic power consumption per block is described in Table 6-10.

Component	Switching power [mW]	Internal power [mW]	Total power [mW]
Control	0.04	0.32	0.36
- interface	*0*	*0.12*	*0.12*
- instruction decoder[a]	*0.02*	*0.06*	*0.08*
- pc and loop stack	*0.02*	*0.14*	*0.16*
Addressing unit	0.08	2.31	2.39
- interface	*0.07*	*0.14*	*0.21*
- decoder	*<0.01*	*<0.01*	*<0.01*
- mux (all)	*<0.01*	*0.02*	*0.02*
- regbanks (all)	*0.01*	*2.15*	*2.16*
Mac unit	0.55	1.52	2.07
- interface	*<0.01*	*<0.01*	*0.01*
- units 1 & 2 (together)[a]	*0.35*	*0.98*	*1.33*
- regs	*0.12*	*0.42*	*0.54*
- mac	*0.23*	*0.56*	*0.79*
- unit 3 [a]	*0.19*	*0.54*	*0.73*
- regs	*0.07*	*0.27*	*0.34*
- mac	*0.12*	*0.27*	*0.39*
Pseudo-random unit	0	<0.01	<0.01
Clock tree	0.72	0	0.72
Total	1.39	4.15	5.54

Table 6-10: Dynamic power consumption of the EGM coprocessor[b].

a. Without embedded RAM
b. Estimated by simulation

Component	Size (words×bits)	Total power [mW]
Program memory	256×13	2.06
PCA memory[a]	1216×16	2.65
REF memory[a]	64×16	1.61
Total		14.84

Table 6-11: Maximum power consumption of the embedded RAMs at 10 MHz.

a. Per memory. One memory present in each of the three MAC unit.

The clock tree accounts for around 13% of the consumption of the core and the addressing unit containing the large register banks consumes slightly more power than the three MAC units. The consumption of the registers in the addressing and MAC units, together with the switching of the clock net accounts for nearly 70% of the total power consumption! The overhead of the control comprising the instruction decoding, state-machines, etc. remains small (around 6.5%).

Note that the embedded memories are not taken into account. Only the worst case power consumption is given in the device datasheet (Table 6-11). Although the power consumption of these memories in read mode is probably less than the reported value, a subtotal of 14.8 mW for these seven memories leads to a total of 20.3 mW for the entire coprocessor at 10 MHz. Considering this worst case situation, the power consumption of the sole memories account for 73% of the dynamic power consumption!

Since most of the power consumption of the core comes from the registers and clock tree net, it is clear that inserting gated-clock should dramatically reduce the power consumption. The gated-clock insertion was performed automatically in Synopsys using the following commands:

```
set_clock_gating_style /* before elaborate */
elaborate -library MYLIBRARY -gate_clock gm_top
propagate_constraints -gate_clock /* before compile */
compile
report_clock_gating
```

Only registers containing at least three flip-flops and an enable input were replaced with gated clock versions, but this represents already 80% of the total number flip-flops.

More importantly, the register banks in the addressing unit (which have a very low activity because they change their values only when a *compsc* instruction is true or when the graph is displaced) will greatly benefit from gated-clock.

Performance figures with and without gated clock are reported in Figure 6-22. Additionally, the area after clock-gating was reduced to 1.24 mm^2 (vs. the previous 1.28 mm^2). While this might seem strange at first sight, it can be explained by the fact that flip-flop with complex enables or resets were initially curiously implemented with *scan-flip-flops*, which involve a very large area penalty as compared to simple flip-flops. By using combinatorial logic and simple flip-flops instead the area was reduced by approximately 2%.

Component	Switching power [mW]	Internal power [mW]	Total dynamic power [mW]
Control	0.05	0.25	0.30
- interface	*0.01*	*0.12*	*0.13*
- instruction decoder[a]	*0.02*	*0.06*	*0.08*
- pc and loop stack	*0.02*	*0.07*	*0.09*
Addressing unit	0.12	0.46	0.58
- interface	*0.12*	*0.06*	*0.18*
- decoder	*<0.01*	*<0.01*	*<0.01*
- mux (all)	*<0.01*	*0.02*	*0.02*
- regbanks (all)	*<0.01*	*0.38*	*0.38*
Mac unit	0.56	1.44	2.00
- interface	*<0.01*	*0.01*	*0.01*
- units 1 & 2 (together)[a]	*0.37*	*0.94*	*1.31*
- *regs*	*0.13*	*0.38*	*0.50*
- *mac*	*0.24*	*0.56*	*0.80*
- unit 3 [a]	*0.19*	*0.49*	*0.68*
- *regs*	*0.07*	*0.22*	*0.29*
- *mac*	*0.12*	*0.27*	*0.39*
Pseudo-random unit	0	<0.01	<0.01
Clock tree	0.29	0	0.29
Total	1.02	2.15	3.17

Table 6-12: Dynamic power consumption of the EGM coprocessor with gated clock. Switching power represents the power consumption of the nets interconnecting the cells, while the internal power represents the dynamic power of the internal nets of the cells.

a. Without embedded RAM

The power consumption after clock gating is reported in details in Table 6-12. The two major reductions in power consumption can be found in the clock tree (factor 2.5) and in the addressing unit register banks (factor 5.5) as can be seen on Figure 6-22. On the other hand, the registers of the MAC unit do not benefit significantly from clock gating. This can be explained by the fact that the latter are very often modified, whereas the node position registers in the LSU unit are only rarely modified and consequently benefit from gating.

Note that the capacity of the interconnects was only approximated based on the fanout of the cells, but that Synopsys PowerCompiler has no exact information on the route length connecting those cells, and thus no exact information on the real net capacitance. A better estimation could be achieved after place and route, when the capacitance of the interconnects is precisely determined. The same applies to cross-talk, bonding pads, and other such sources of dynamic and static power dissipation.

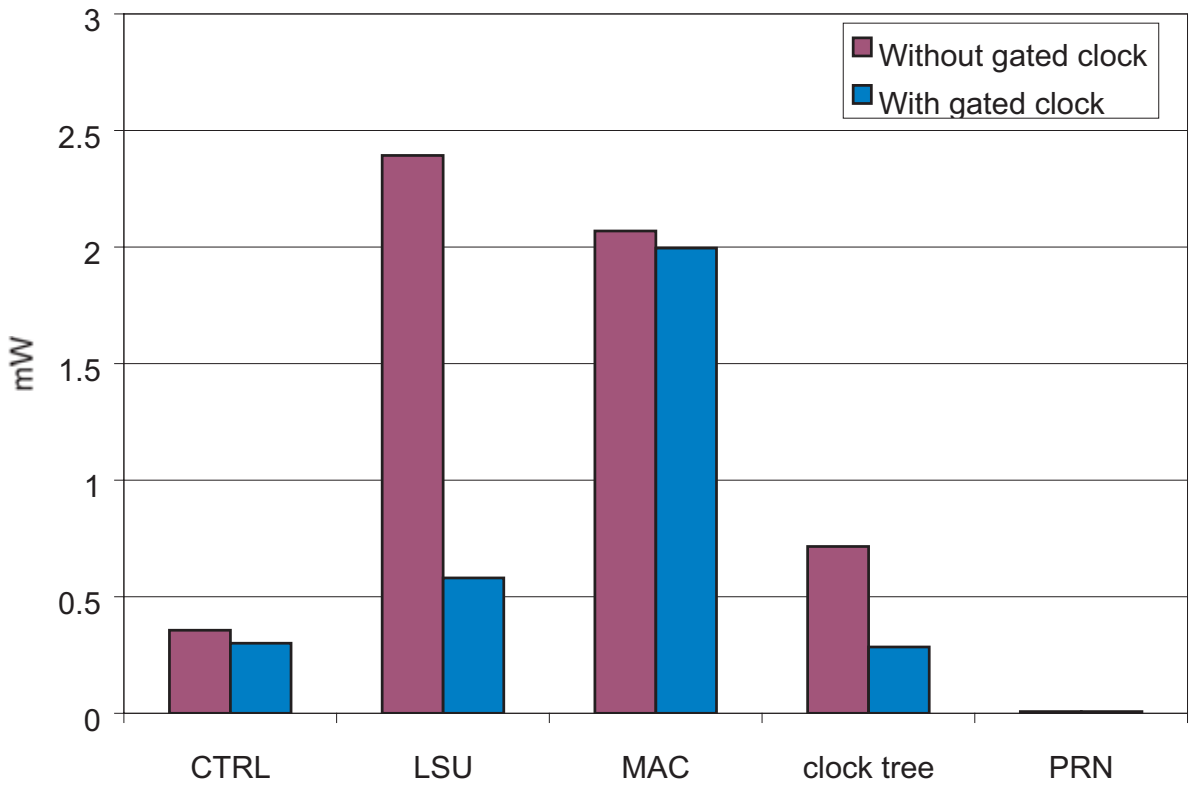


Figure 6-22: Effect of automatic clock-gating on power consumption.

Finally, due to the low duty-cycle, it is theoretically possible to reduce the supply voltage V_{dd} so as to reach a duty cycle of 100%. The power consumption (static and dynamic) would be reduced as well. Starting from the delay equation derived from the *alpha-power law* [6-15]:

$$\tau = k \cdot \frac{V_{dd}}{(V_{dd} - V_{th})^{\alpha}} \qquad (6.4)$$

where k is a constant for the particular design and technology, and has to be determined experimentally.

One can find k by replacing the values for the UMC process ($V_{dd} = 1.8V$, $\alpha \cong 1.5$, $V_{th} = 0.5V$) and by using $\tau = 12$ ns since the critical path of the co-processor has an associated delay of around 11.6 ns. The constant k is thus approximately $k \approx 9.88$. Now the supply voltage that would achieve a 100% duty cycle at 10 MHz, i.e. $\tau = 100$, can be solved numerically. In this case, the supply voltage would be as low as 0.66 V, i.e. very close to the threshold voltage $V_{th} = 0.5V$.

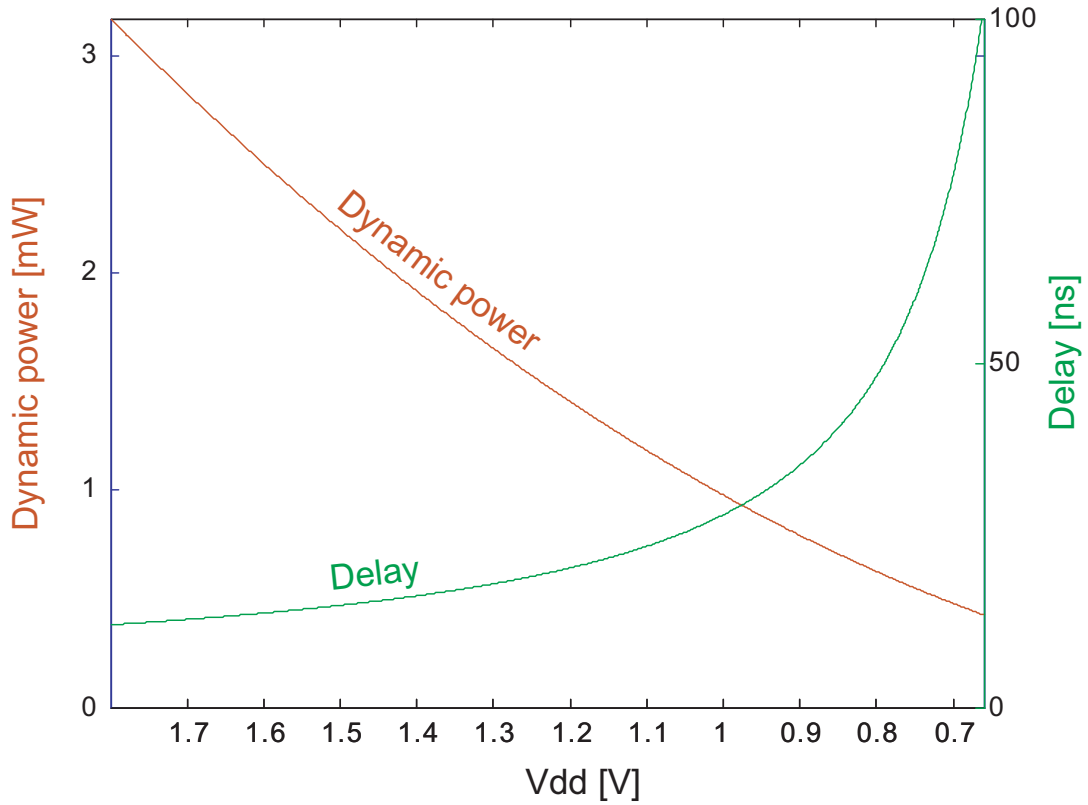


Figure 6-23: Effect of supply voltage scaling on dynamic power and delay.
A constant threshold voltage of 0.5 V is assumed in the UMC 0.18 µm process. The maximum delay is 100 ns corresponding to a supply voltage of 0.66 V. The dynamic power at the nominal VDD=1.8 V is 3.97 mW.

Since the dynamic power is proportional to the square of the supply voltage, this power would be theoretically reduced to 0.43 mW at $V_{dd} = 0.66V$, while still fulfilling the timing constraints. The static power is proportional to V_{dd} but would remain as negligible as in the nominal case. Instead of using this very low supply voltage, a continuous scaling is possible leading to the power and delays in Figure 6-23.

For instance, at $V_{dd} = 1V$ the dynamic power is approximately 0.98 mW, i.e. 70% less than at $V_{dd} = 1.8V$, while the delay is not dramatically changed (i.e. still far from 100 ns). It should be kept in mind in all cases that this voltage scaling does not apply to the embedded memories that clearly consume the largest amount of power. An alternative would be to use low-power memories, but these were not available for the selected UMC technology.

6.9 Conclusion

In this chapter we first defined basic rules for reducing the power consumption:

- use the lowest possible supply voltage, and the lowest possible operation frequency;
- use parallelism and pipelining when data or instruction parallelism is available, in order to reduce the required frequency;
- use power management (e.g. gated-clock) whenever possible.

Then we described a design flow for the realization of application specific processor (ASIP) architectures together with the necessary tools. Finally, the system-on-chip was discussed with a description of its different components, furnishing a more detailed view of the Graph Matching coprocessor.

The two implementation requirements for this face verification system were that a single face verification gets processed within less than one second, and that the realization complies with the needs of low-power consumption. The first requirement is clearly met, as the full matching operation on the processor takes less than 0.7 second. However communication speed can be a limiting factor and should be designed carefully so that the total execution time does not exceed the 1 second barrier. The second requirement seems also reached, although it is harder to evaluate since no other dedicated architecture for face verification of elastic graph matching was characterized in terms of power consump-

tion. Nevertheless with a total power consumption in the order of tens of milliwatts, this consumption is lower than the power consumption of general purpose or digital signal architectures, which are more often in the order of hundreds of milliwatts, when not more. Still, we considered the power consumption of a single component, i.e. the EGM coprocessor, to which the consumption of the morphology coprocessor, master control unit (MCU), CMOS camera and shared memories should be added. To extract a more precise figure of the power consumption, the design of this System-on-Chip should be continued and completed after closure of this Ph.D. work.

Possible improvements of the EGM architecture should target a reduction of the number and organization of embedded memories, as they seem to consume the major part of the power. Replacement with special low-power memories shall also be considered.

References

[6-1] M. Ansorge, S. Tanner, X. Shi, J. Bracamonte, J.-L. Nagel, P. Stadelmann, F. Pellandini, P. Seitz, and N.Blanc, "SmartLow-Power CMOS Cameras for 3G Mobile Communicators," *Invited Paper,* in *Proc 1st IEEE Int'l. Conf. on Circuits and Systems for Communications (ICCSC'02)*, June 26-28, 2002, St-Petersburg, Russia, pp. 216-225.

[6-2] B. E. Bayer, "Color imaging array", *US Patent No. 3971065*.

[6-3] L. Benini, and G. De Micheli, "System-Level Power Optimization: Techniques and Tools," *IEEE Symposium on Low Power Electronics and Design* (*ISLPED'99)*, San Diego, California, Aug. 16-17, 1999, pp. 288-293.

[6-4] N. Blanc, S. Lauxtermann, P. Seitz, S. Tanner, A. Heubi, M. Ansorge, F. Pellandini, E. Doering, "Digital Low-Power Camera on a Chip," in *Proc. COST 254 Int.l Workshop on Intelligent Communication Technologies and Applications, with Emphasis on Mobile Communication*, Neuchâtel, Switzerland, May 5-7, 1999, pp. 80-83.

[6-5] T.J. Van Eijndhoven, K.A. Vissers, E.J.D. Pol, P. Struik, R.H.J. Bloks, P. van der Wolf, H.P.E. Vranken, F.W. Sijstermans, M.J.A. Tromp, A.D. Pimentel, "TriMedia CPU64 Architecture," *IEEE Int'l. Conf. on Computer Design (ICCD'99)*, Austin, Texas, Oct. 10-13, 1999, pp. 586-592.

[6-6] M.D. Ercegovac, and T. Lang, "Digital Arithmetic," *Morgan Kaufmann Publishers*, San Francisco, CA, 2004.

[6-7] D. Fischer, J. Teich, M. Thies, and R. Weper, "Efficient Architecture/Compiler Co-Exploration for ASIPs," in *Proc. ACM Int'l Conf. on Compilers, Architectures and Synthesis for Embedded Systems (CASES'02)*, Grenoble, France, Oct. 8-11, 2002, pp. 27-34.

[6-8] J.L. Hennessy, and D.A. Patterson, "Computer Architecture: A Quantitative Approach," *Third Edition*, *Morgan Kaufmann Publishers*, San Francisco, CA, 2003.

[6-9] K. Hwang, "Computer arithmetic : principles, architecture, and design", J. Wiley, New York, NY, 1979.

[6-10] S. Ishiwata, and T. Sakurai, "Future Directions of Media Processors," *IEICE Electronics, Special Issue on Multimedia, Network, and DRAM LSIs*, Vol. E81-C, No. 5, May 1998, pp. 629-635.

[6-11] M. Keating, and P. Bricaud, "Reuse Methodology Manual for System-on-a-Chip Designs," 2nd Edition, *Kluwer Academic Publishers*, Boston, MA, 1999.

[6-12] C. Kotropoulos, A. Fazekas, and I. Pitas, "Fast multiscale mathematical morphology for frontal face authentication," in *Proc. IEEE-EURASIP Workshop Nonlinear Signal and Image Processing (NSIP'03)*, Grado, Italy, June 8-11, 2003 (CD-ROM).

[6-13] R. Leupers, "LANCE: A C Compiler Platform for Embedded Processors," *Embedded Systems/Embedded Intelligence*, Nürnberg, Germany, Feb. 2001.

[6-14] R.S. Nikhil,and Arvind, "Implicit Parallel Programming in pH," *Academic Press*, San Diego, CA, 2001.

[6-15] K. Nose. and T. Sakurai, "Optimization of VDD and VTH for low-power and high speed applications", in *Proc. 2000 Conf. on Asia South Pacific Design Automation*, Yokohama, Japan, 2000, pp. 469 - 474

[6-16] V.G. Oklobdzija, D. Villeger, and S.S. Liu, "A Method for Speed Optimized Partial Product Reduction and Generation of Fast Parallel Mulitpliers Using an Algorithmic Approach," *IEEE Trans. Comp.*, Vol. 45, No. 3, Mar. 1996, pp. 294-306.

[6-17] M. Pedram, “Power minimization in IC design: Principles and applications,” *ACM Trans. Design Automation*, Vol. 1, No. 1, Jan.1996, pp. 3–56.

[6-18] W. Qing, M. Pedram, W. Xunwei, “Clock-gating and its application to low power design of sequential circuits,” *IEEE Trans. Circuits and Systems I: Fundamental Theory and Applications*, Vol. 47, No. 3, Mar. 2000, pp. 415-420

[6-19] R. Rajsuman, “System-on-a-Chip: Design and Test,” *Artech House*, Boston, MA, 2000.

[6-20] K. Roy, and S.C. Prasad, “Low-Power CMOS VLSI Circuit Design,” *Wiley Interscience*, New York, 2000.

[6-21] Semiconductor Industry Association, “SIA Roadmap, 2002 Update,” http://www.sia-online.org/

[6-22] D. Soudris, C. Piguet, and C. Goutis, “Designing CMOS Circuits for Low Power,” *Kluwer Academic Publishers*, Dordrecht, The Netherlands, 2002.

[6-23] Ch. Schuster, J.-L. Nagel, Ch .Piguet, P.-A. Farine, “Leakage reduction at the architectural level and its application to 16 bit multiplier architectures,” in *Proc.14th IEEE Int’l. Workshop on Power and Timing Modeling, Optimization and Simulation (PATMOS’04)*, Sep. 15-17, 2004, Santorini, Greece, pp. 169-178.

[6-24] K. Slimani, J. Fragoso, L. Fesquet, Marc Renaudin, “Low Power Asynchronous Processors”, Chapter 22 in *“Low-Power Electronics Design”*, C. Piguet ed., CRC Press, Boca Raton, FL, July 2004.

[6-25] P. Stadelmann, J.-L. Nagel, M. Ansorge, F. Pellandini, “A Multiscale Morphological Coprocessor for Low-Power Face Authentication,” in *Proc. XIth European Signal Processing Conference (EUSIPCO 2002)*, Toulouse, France, Sept. 03-06, 2002, Vol. 3, pp. 581-584.

[6-26] P. Stadelmann, “Features Extraction for Low-Power Face Verification”, *Ph.D. thesis*, to appear.

[6-27] S. Tanner, S. Lauxtermann, N. Blanc, M. Waeny, M. Willemin, J. Grupp, R. Dinger, E. Doering, M. Ansorge, F. Pellandini, and P. Seitz, “Low-Power Digital Image Sensor for Still Picture Image Acquisition,” in *Proc. SPIE Photonics West Conf. 2001*, Vol. 4306, San José, CA, Jan. 21-26, 2001, pp. 358-365.

[6-28] C. Weaver, R. Krishna, L. Wu, and T. Austin, “Application Specific Architectures: A Recipe for Fast, Flexible and Power Efficient Designs,” in *Proc. ACM Int’l Conf. on Compilers, Architectures and Synthesis for Embedded Systems (CASES’01)*, Nov. 16-17, 2001, Atlanta, GE, pp. 181-185.

[6-29] http://www.design-reuse.com

[6-30] http://www.opencores.org

[6-31] http://www.tensilica.com/

[6-32] http://www.umc.com/English/pdf/0.18-DM.pdf

[6-33] http://virtual-silicon.com/

Chapter 7

Conclusions

7.1 Summary

The research performed in biometrics in the industry and in universities is presently considerably expanding. Indeed, the demand is increasing in security areas: passports and country border crossing, plane boarding, surveillance in public places, etc. Nevertheless, the consumer area is also starting to expand and selected applications will progressively appear on the market: access control to devices that can be easily stolen or that are vulnerable to attacks, cars, doors, etc. On the other hand, computer and software providers have more difficulties in selling biometric systems for domestic computers. Indeed, the risk of impostors trying to physically access the workstation is so low that it does not seem to justify the acquisition of a biometric device. Still, this could change if the authentication gets used for online transactions, in which case the impostor might not be sitting behind the computer, but would try cracking the transaction from a remote location.

Even so, our research focused mainly on applications targeted at simplifying the everyday life of consumers, i.e. increasing the security without compromising ease of use, rather than targeting highly secure but invasive applications.

The principal achievement of this Ph.D. work is the development of: 1) a set of algorithms for face authentication that take into account the robustness requirements needed for mobile applications, that are modular, and of reduced complexity; 2) a dedicated low-power coprocessor architecture and related software; 3) the realization of a prototype demonstrating the feasibility of implementing the face authentication entirely on a low-power embedded system, with a response time lying below one second.

7.2 Recall and discussion of contributions

The contributions of this thesis are now recalled and discussed chapter by chapter.

The main contribution of Chapter 2 resides in the identification of mobile applications of biometrics. It was demonstrated that mobile phones or mobile PDAs clearly require a higher degree of security that can be offered by biometrics. It was also established that face authentication is an interesting modality for this purpose, due to the low invasiveness and to the growing presence of digital cameras on such devices. An extension of this contribution lies in the definition of the performance requirements of face authentication on mobile devices, such as reduced power consumption, and a response time being lower than or equal to one second. The tolerances with respect to varying viewpoints and scale factors were discussed, and the illumination variation was identified as the key problem for face authentication on mobile devices, thereby motivating its detailed study in the following chapters.

In Chapter 3, a list of six criteria for classifying existing approaches was given, which were subsequently used to select an algorithm fulfilling the performance requirements cited in Chapter 2. An important contribution consists in the review of the state-of-art, by discussing an important yet non-exhaustive list of possible implementations. The major research directions were classified based on the subset of the three most important criteria from the above list, and an algorithm category was selected. Finally, a specific contribution was also provided by reviewing the commercial products and patents existing in the field.

The first contribution of Chapter 4 is a detailed analysis of the effects of illumination variation (intensity *and* spectral distribution) on face authentication and the review of several normalization techniques based on this analysis, successively considering methods that predict the effect of luminance (reconstructionist methods) and those that derive new, less sensitive, features. A number of conclusions were drawn and robust features were selected for the targeted application, although it is known that reconstructionist methods perform usually better but at a higher cost which is currently not affordable on mobile devices.

A second contribution of Chapter 4 is brought by the systematic comparison of three feature extraction techniques (Gabor filtered images,

Discrete Cosine Transform, and mathematical morphology) combined to the Elastic Graph Matching (EGM) algorithm, which was selected in Chapter 3. Follows the introduction of a novel low-complexity feature extraction technique based on mathematical morphology. The performance comparison takes of course into account the authentication error rates obtained on standard databases, but also the computational complexity and the impact on the algorithm convergence. Again, although computationally complex techniques such as Gabor features globally perform best, in some situations (e.g. under relatively well controlled lighting conditions), simple methods exhibit performances that are very close to those of Gabor features, though at a much reduced cost (approximately 30 times less additions for normalized morphology). Depending on the application conditions, feature extraction techniques based on mathematical morphology are thus of great interest.

Following the propositions found in external publications, a reduction of the feature space was experimented on the mathematical morphology features by means of Principal Component Analysis or Linear Discriminant Analysis. The fact that no improvement was however observed seems in contradiction with results of an external team [4-39]. While this is most certainly related to a too small number of training samples, it might be due to a problem in the setup or performance measure of one of the two teams.

A third distinct contribution of this chapter consisted in the use of a Simplex Downhill method for the iterative optimization of the graph correspondence. Compared to other methods found in the literature (e.g. Monte Carlo or simulated annealing approaches), this method is extremely simple and does not require complex gradient calculations or storage of a large number of data. This method was combined to a new scale adaptation and a new graph deformation penalty, which permitted the simultaneous optimization of rotation and scaling. In addition, an original morphological level switching strategy was presented to compensate scale factors lying outside the tolerance range of 0.7 to 1.4.

Finally, a discussion of the classification was performed, pointing out that the commonly used Euclidean metrics is not optimal, except under strong assumption that actually never holds true. Nevertheless, an a posteriori normalization (z-norm) of the distances was used, inspired from speaker verification problems, showing a significant improvement in the Receiver Operating Curves.

Chapter 5 presented a VLSI architecture for online color-based face detection and strategies to combine it with model-based face detection using the Elastic Graph Matching algorithm. The combination of a low-complexity solution presenting an important false detection rate with a more precise but more time consuming solution is very interesting in that it allows to remove quickly the regions which are not interesting and to focus only on selected regions.

A second contribution was given in Chapter 5 by extending the use of EGM to the fingerprint modality in combination with Gabor filtering. Original experiments using mathematical morphology to replace the computationally expensive Gabor filters were also described. These preliminary results illustrate the interest of such methods when only small portions of fingerprints are available for matching. However, this field of research remains open to further improvements.

Finally, the originality elaborated in Chapter 6 resides in the description of an embedded low-power face authentication system, including a dedicated coprocessor executing most of the low-level and regular EGM operations. Indeed, it is recalled that no other (academic or commercial) system was found by the author at time of the preparation and closure of this Ph.D. work, that enables the full verification of a face on a mobile device.

7.3 Perspectives

One of the most challenging subjects remaining open to future research in the area of mobile face authentication concerns the robustness to varying illumination conditions. The influence of further aspects, like for instance pose and expression variations, can be relatively efficiently compensated. Since the illumination problem is a recurrent problem in many machine vision applications, there is no doubt that innovative solutions will be proposed in a near future.

Although in Chapter 3, we abandoned 3D representations in favour of 2D ones for matching faces, due to the complexity of the former and the fact that 3D representations alone do not seem to contain more information than 2D ones, the *combination* of the 3D with the 2D information can hopefully describe more precisely the spatial organization of a scene, helping to predict and compensate for non-linear illumination effects such as cast shadows and highlights (see Subsection 4.2.1). With the apparition of low-cost 3D acquisition devices, the 3D reconstruc-

tionist approaches (i.e. methods predicting the variability of a 2D representation based simultaneously on 2D and 3D representations) might become more accessible, as compared to the current 2D-based reconstructionist methods mentioned in Chapter 4, which can definitely not operate in real-time.

Finally, the possible incorporation of the results achieved in this Ph.D. thesis in a commercial product would of course ask for complementary applied research activities. In this perspective, and starting from the existing prototype, more detailed specifications regarding the operating conditions, the definition of the possible lighting conditions, the type and resolution of the camera to be selected, the number of users who shall be authenticated per mobile device, etc., should be established based on a precise application scenario. Then the integration of the whole system comprising the camera and the two dedicated coprocessors should preferably be realized in form of a System-on-Chip, enabling the realization of a new prototype generation.

Appendix A

Hardware and software environment

A.1 Real-time software demonstrator

The webcam-based real-time demonstrator described in Subection 2.7.3 was made of the following components:

- Acer Laptop, 700 MHz Mobile Pentium III processor, 256 MB of memory, USB 1 connectivity;
- Logitech webcams: compatibility and tests performed with QuickCam Express, QuickCam Pro 3000, QuickCam Pro 4000;
- Logitech drivers: compatibility and tests performed with QuickCam software, versions 6.01 up to 9.02;
- Logitech Software Development Kit (SDK): version 1.0 (File version: 2.11.15.0). No update was available and no support is provided;
- Compiler: Microsoft Visual C++ 6.0;
- Additional libraries: Intel Image Processing Library, version 2.5.

WARNING: the Logitech SDK is not compatible with the latest versions of the QuickCam software! More precisely, the interface defined in the “include” files shipped by Logitech in the SDK is not the same as the interface exposed by the actual driver in the latest versions. A trick that was proposed by a German team (Rainer and Ralf Gerlich, BSSE, Germany) is working with all the drivers we tested. However, the software needs to be recompiled for each new driver version! The solution was exposed in 2004 in the following message board:

http://www.cedhart.ch/cgi-bin/messageboard/data/board1/23.shtml

The procedure they describe is as follows:

1. Find the HPortal 1.0 Type Library in the OLE/COM Inspector from Visual C++ 6.0;
2. Save the IDL description;
3. Compile this IDL to vportal2.h and vportal2_i.c using the following MIDL compiler command: midl /h vportal2.h /iid vportal2_i.c vportal2.idl
4. Replace the header and c files provided in the SDK with those compiled with MIDL.

A “cleaner” solution would probably consist in using more standard interfaces, such as Twain (*http://www.twain.org*).

Acknowledgments

I would like to thank the experts who dedicated part of their valuable time to proof-read this manuscript. Firstly, I am deeply grateful to Professor Pellandini for welcoming me in his team, back in 1998, and for giving me the opportunity to work on very stimulating problems, one of which led to this thesis. Prof. Pellandini succeeded in creating a very cheerful and motivating working atmosphere at the Electronics and Signal Processing Laboratory (ESPLAB).

Then, I am also thankful to Prof. Farine who is the successor to Prof. Pellandini at the head of ESPLAB. I hope he will be as successful as Prof. Pellandini, for the next 30 years !

Dr. Michael Ansorge deserves very special thanks. Firstly, because he spent an tremendous amount of time to very carefully read my manuscript, making comments and suggesting modifications that significantly improved the text. Secondly, for the phenomenal time and efforts he spent developing the activities of the laboratory in the field of low-power multimedia processing, and because I could learn a lot under his guidance, during the almost 8 years I worked at ESPLAB.

Further, I would like also to express my gratitude to Dr. Christian Piguet for all the interesting and vivacious discussions, about electronics, politics or environment, we had so far, and that we will most certainly continue to have in the future.

Also, I would like to thank the CSEM teams, from Neuchâtel and from Zürich, for the fruitful collaboration maintained during more than five years, and for their financial support.

Then, of course, I am thankful to all my present and former colleagues who indeed participated in maintaining this cheerful atmosphere. Mentioning them all here would be certainly too long and omitting someone would be an offence... nevertheless, I am specially thankful to Giusy, Patrick, Christian R., and Alain.

Last but not least, I want to thank Steve and Christian S. who shared with me the usual up and downs of researchers: I believe we always succeeded maintaining a nice climate in our R.132 office.

Remerciements

Et en français... je tiens à remercier tout spécialement mes parents de m'avoir toujours laissé le libre choix, entre autres dans mes études, et de m'avoir témoigné leur confiance. J'ai pu grâce à eux bénéficier d'un climat idéal au cours de mes études, sans devoir travailler pour financer ma formation, et pouvant dès lors consacrer une grande partie de mon temps à celle-ci. Je leur en suis profondément reconnaissant et je leur dédie cette thèse.

Enfin, je remercie Sylviane qui partage ma vie depuis 6 ans. Elle m'a à maintes reprises fortement "encouragé" à finir cette thèse alors que, parfois, je me laissais un peu déborder par d'autres projets dans le cadre mon travail à l'IMT. Peut-être désormais aurai-je d'avantage de temps pour m'occuper de notre maison ! :-)

Curriculum vitae

Jean-Luc NAGEL
Born in Neuchâtel, on March 29th, 1975.
Swiss citizen.

Contact

IMT - ESPLAB
Rue Breguet 2
2000 Neuchâtel
Switzerland

Les Biolles 8
2019 Chambrelien
Switzerland

E-mail: jean-luc.nagel@unine.ch

Education

Degree of Physics - Electronics from the University of Neuchâtel.	1998
Diploma project on “Asynchronous CoolRisc ™”.	1998
Semester project on “Image compression using block truncation coding”.	1997

Experience

Scientific collaborator at the Electronics and Signal Processing Laboratory of the Insitute of Microtechnology of the University of Neuchâtel	since 1998
Coaching of 6 semester projects, 4 diploma projects, and 4 traineeships	1998-2006
Teaching assistant for exercises of electronics (introductory level)	1997-1998
Trainee at the CSEM (Centre Suisse d’Electronique et Microtechnique) dealing with low-level arithmetic routines for the low-power CoolRisc micro-controller family.	1997 (1 month)

Scientific projects at IMT

CTI 3921.1 with SysNova S.A. on composite video signal decoding on a digital signal processor	1998
CTI JOWICOD with ST Microelectronics on joint source-channel coding for 3G devices	1999
Cooperation projects with CSEM S.A. on biometrics	
- SmartPix	2000-2001
- SmartCam	2002-2003
- BioMetrics	2004-2005
Digital design at very low supply voltage (cooperation project with CSEM S.A. on low-power design)	2003-2005

Skills

Software development languages:	C/C++, Matlab
Hardware description languages:	VHDL
VLSI design tools:	synthesis tools from Synopsys, front end and simulation from Mentor Graphics (Renoir, ModelSim), back end from Cadence (Silicon Ensemble), FPGA design environment from Altera (Quartus).

Domains of interest

All aspects of digital image processing, including pattern recognition, image compression (lossy and lossless), image correction (e.g. white-balancing, denoising), for various applications ranging from consumer market (e.g. digital cameras) to medical applications, and, of course, biometrics.

Low-power VLSI digital and system-on-chip design is another important domain of interest, especially architecture level design (behavioral, RTL), and hardware-software co-design.

Publications involving the author

Publications related to biometrics

- M. Ansorge, J.-L. Nagel, P. Stadelmann, and P.-A. Farine, “Biometrics for Mobile Communicators," Invited Talk, *2nd IEEE Int’l. Conf. on Circuits and Systems for Communications (ICCSC’04)*, June 30-July 2, 2004, Moscow, Russia.
- J.-L. Nagel, P. Stadelmann, M. Ansorge, and F. Pellandini, "Comparison of feature extraction techniques for face verification using elastic graph matching on low-power mobile devices," in *Proc. IEEE Region 8 Int'l. Conf. on Computer as a Tool (EUROCON’03)*, Ljubljana, Slovenia, Sept. 22-24, 2003, Vol. II, pp.365-369.
- T.D. Teodorescu, V.A. Maiorescu, J.-L. Nagel, M. Ansorge, "Two color-based face detection algorithms: a comparative approach," in *Proc. EEE Int'l Symp. on Signals, Circuits, and Systems, SCS 2003*, Iasi, Romania, July 10-11, 2003, pp. 289-292.
- E. Bezhani, D. Sun, J.-L. Nagel, S. Carrato. “Optimized filterbank fingerprint recognition," in *Proc. 15th Annual Symposium on Electronic Imaging, Image Processing: Algorithms and Systems II*, Vol. 5014, Santa Clara, CA, USA, Jan. 21-23, 2003, pp. 399-406.
- J.-L. Nagel, P. Stadelmann, M. Ansorge, F. Pellandini, “A Low-Power VLSI Architecture for Face Verification using Elastic Graph Matching," in *Proc. XIth European Signal Processing Conf., EUSIPCO 2002*, Toulouse, France, Sept. 03-06, 2002, Vol. 3, pp. 577-580.
- P. Stadelmann, J.-L. Nagel, M. Ansorge, F. Pellandini, “A Multiscale Morphological Coprocessor for Low-Power Face Authentication," in *Proc. XIth European Signal Processing Conf., EUSIPCO 2002*, Toulouse, France, Sept. 03-06, 2002, Vol. 3, pp. 581-584.
- M. Ansorge, S. Tanner, X. Shi, J. Bracamonte, J.-L. Nagel, P. Stadelmann, F. Pellandini, P. Seitz, N.Blanc, “SmartLow-Power CMOS Cameras for 3G Mobile Communicators," Invited Paper, in *Proc 1st IEEE Int'l Conf. on Circuits and Systems for Communications, ICCSC'02*, June 26-28, 2002, St-Petersburg, Russia, pp. 216-225.
- M. Ansorge, F. Pellandini, S. Tanner, J. Bracamonte, P. Stadelmann, J.-L. Nagel, P. Seitz, N. Blanc, C. Piguet, “Very Low Power Image Acquisition and Processing for Mobile Communication Devices," Keynote paper, in *Proc.of IEEE Int’l. Symposium on Signals, Circuits & Systems, SCS 2001*, Iasi, Romania, July 10-11, 2001, pp. 289-296.

Other publications

- C. Schuster, J.-L. Nagel, C. Piguet, P.-A. Farine, “An architectural Design Methodology for Minimal Total Power Consumption at Fiexed Vdd and Vth”, *Journal of Low Power Electronics (JOLPE)*, Vol.1, 2005, pp. 1-8.
- C. Schuster, J.-L. Nagel, C. Piguet, P.-A. Farine, “Leakage reduction at the architectural level and its application to 16 bit multiplier architectures," in *Proc. 14th IEEE Int’l. Workshop on Power and Timing Modeling, Optimization and Simulation*, Sept. 15-17, 2004, Santorini, Greece, pp. 169-178.
- C. Piguet, C. Schuster, J-L. Nagel, “Optimizing Architecture Activity and Logic Depth for Static and Dynamic Power Reduction," in *Proc. 2nd Northeast Workshop on Circuits and Systems, NewCAS’04*, Montréal, Canada, June 20-23, 2004.
- R. Costantini, J. Bracamonte, G. Ramponi, J.-L. Nagel, M. Ansorge, F. Pellandini, “A Low-Complexity Video Coder Based on the Discrete Walsh Hadamard Transform," in *Proc.of*

EUSIPCO 2000, European Signal Processing Conference 2000, Tampere, Finland, Sept. 4-8, 2000, pp. 1217-1220.

- J.-L. Nagel, M. Ansorge, F. Pellandini, “Low-cost Composite Video Signal Acquisition and Digital Decoding for Portable Applications," in *Proc. COST 254 Int’l Workshop on Intelligent Communication Technologies and Applications, with Emphasis on Mobile Communication*, Neuchâtel, Switzerland, May 5-7, 1999, pp. 71-76
- J.-L. Nagel and C. Piguet, "Low-power Design of Asynchronous Microprocessors," PATMOS'98, Oct. 6-8, Copenhagen, Denmark, 1998, pp. 419-428.

Short courses / seminars

- J.-L. Nagel, “Face Authentication” presented in the frame of the Summer Course of the LEA (Laboratoire Européen Associé), *Highlights in Microtechnology*, July 2004 and July 2005, Neuchâtel, Switzerland.
- J.-L. Nagel, “White balancing” presented in the frame of the Summer Course of the LEA (Laboratoire Européen Associé), *Highlights in Microtechnology*, July 2004 and July 2005, Neuchâtel, Switzerland.
- J.-L. Nagel, “Biometrics and face Authentication” presented in the frame of the Advanced Signal Processing course organized by PD Dr. M. Ansorge as part of the MSc degree in Micro and Nanotechnology, May 2005, Neuchâtel, Switzerland.