Wavelet Transforms and Template Approaches to Face Recognition

SiNguyen Vo

A Thesis

In

The Department

Of

Computer Science

Presented in Partial Fulfillment of the Requirements
For the Degree of Master of / Magisteriate in Computer Science at
Concordia University
Montreal, Quebec, Canada

April 2002

Abstract

Wavelet Transforms and Template Approaches to Face Recognition

SiNguyen Vo

Face recognition is a very important task in many applications such as biometric authentication or for content-based indexing photo and video retrieval systems. In recent years, considerable progress has been made on the problems of face detection and recognition, using different methods divided into two groups of geometrical measures and template matching. However, as computation is very expensive and require a great amount of storage for the earlier methods based on correlation, several more recent methods have then been based on principal component analysis, neural network classification and deformable model of templates of features.

The first topic of the work reported in this thesis is the experimental evaluation of face recognition methods based on template approaches. Our aim is to test different approaches of template matching: using cross-correlation with Fast Fourier transform, using features obtained from filtering with Gabor wavelet transform or Daubechies wavelet transform, with both rigid grid matching and deformable graph matching. Then, in the second part of the thesis, we propose an implementation of face recognition based on the Daubechies wavelet transform with the matching of series of corresponding graphs while being both speed and storage friendly. The experiments performed on the entire image database of AT&T Laboratories Cambridge show that while the training phase

from our proposed face recognition system outperforms in terms of speed other previously described methods such as those based on Fast Fourier transform, Gabor wavelet transform, and even Eigenfaces, its recognition rate is 81% for the entire raw database and even reaches 91% when images are not distorted by strong facial expressions or accessories.

Thesis co-supervisor: Professor Ching Y. Suen
Thesis co-supervisor: Professor Tien D. Bui
Title: Wavelet Transforms and Template Approaches to Face Recognition

To my grandmother,
my mother,
my father,
and baby Thuan-Cat

Acknowledgements

Many persons contributed to the success of this thesis, and I wish to use this opportunity in order to express my personal consideration for their continuous support.

I am very grateful to my co-supervisors Professor Ching Y. Suen, Director of CENPARMI and Professor Tien D. Bui, Graduate Program Director, for their patience, timely advice and valuable guidance in the completion of this work. Without their permanent support and enthusiasm for research, this thesis would not have seen the light of day.

I wish to express my gratitude to my parents for providing encouragements and unconditional support, day after day, in all these years. I would like to address special thanks to my uncle Doan, for being the initiator of this thesis. Also thanks to my wife for her help, support and everyday encouragements.

Besides the persons mentioned above, there are many other people, family and friends that support me and encouraged me to realize my thesis. To them, I would like to address a big thank you without giving names in order for me not to miss anyone.

Table of Contents

List of Figures

List of Tables

Chapter 1

Fundamentals of Face Recognition

1.1. Introduction

The face recognition problem has attracted much attention from the public because of its many possible applications in automatic access control systems, human-computer interfaces and crowd surveillance.

Because a face is defined as a class of naturally structured but slightly deformable objects [37], face detection and face recognition are difficult but interesting problems as certain common but significant features, such as glasses, hair or moustache, can either be present or totally absent from a face. Furthermore, when these features are present, they may cloud out other basic facial features. Then with unpredictable imaging conditions in an unconstrained environment, a change in light source distribution can cast or remove significant shadows from a face. Thus these conditions represent challenging aspects of the face detection and recognition problem.

In the first chapter, we introduce the definitions of face detection and face recognition, as well as describe more in detail the challenging problems related to face detection and face recognition that almost every system has to deal with. Then, we will describe the different types of databases and present in detail the most frequently used ones. Chapter two presents the current state of the art in face recognition and face detection. The third and fourth chapters describe the implementations of different techniques of feature detection and matching, problems encountered and its corresponding workaround and are used as the basis to our proposition of a face recognition system in the fifth chapter. The fifth chapter contains the description and implementation of our system, as well as an analysis of its performance from results obtained from its execution on the database of ORL. The last chapter contains discussions about future works.

1.2. Overview of the Face Recognition

1.2.1. Anatomy of the Face Recognition

Every face recognition system must contain at least three main steps:

- Face detection and location of the key features
- Extraction of the relevant information, such as feature vectors
- Comparison of the extracted information

The first component of a face recognition system locates one or several face images from the input image and then their key features, for example the eyes, in a step called face detection. Depending on the conditions under which the input image is obtained, the face detection as well as the features finder algorithm may be simple or advanced and its invariance to view point, such as scale, in-plane and out-plane rotations, and lighting conditions, determines the system *robustness*.

The second component extracts the information from the face candidate locations. The relevant information is also often called feature vectors, which determine the *accuracy* of the face recognition system.

The third component compares the extracted and coded information with similar template from a particular database, using a matching algorithm. Approaches based on neural networks perform this step in a special way, as the templates are inherent to the nodes composing the neural networks architecture.

The percentages of false positives, which detect a face location in the image where there is no face or associates the face with wrong id, and false negatives, determine the

performance of the system. The robustness, accuracy, performance as well as speed are critical to the usefulness of a face recognition system.

1.2.2. Face Detection and Face Recognition

The face detection and face recognition are two different problems although they may make use of the same pool of algorithms. Given an arbitrary image in input, which can be a digitized video signal or scanned photograph, the human *face detection* problem can be stated as determining whether or not there are any human faces in the image, and if there are, return an encoding of the location and spatial extent of each human face in the image. An example of encoding may be the coordinates of the corners of the bounding box, which fits each face in the image. The face detection process is also referred as face location.

On the other hand, the *face recognition* may be seen as a special case of the face detection, and is usually performed after the face detection step is done. The face recognition problem is stated as comparing the input face against models in a library of known faces, called face database, and reporting if a match is found. Furthermore, the definition of face recognition may be expanded to making a distinction between *authentication* and *identification.*

Authentication sometimes is referred to as verification of a claimed identity. This is therefore a one to one (1:1) matching process, as the characteristic in question is matched against the reference associated with the claimed identity, according to a predefined threshold criteria.

Identification is different because it seeks to identify a user from within a population of possible users, according to a characteristic or multiple characteristics, which can be reliably associated with a particular individual, without an identity being explicitly claimed by the user. This is therefore a one to many (1:N) matching process involving a database of relevant data.

1.3. Overview of the Thesis

This work is inspired by the spectroface [67], motivating the use of wavelet, elastic graph matching [73] for the graph matching part, and the combination of Fourier and wavelet transform [70] for the detection part. Here, the Fourier transform of sinusoidal waves is used to correct positional differences between reference and test images, and the wavelet transform to extract differences. Although there are a number of papers presenting these techniques, the relations between them are not always clearly described. We hope to contribute to filling this gap with the current work. The approach is to present the current state of the art in face recognition as an introduction with an emphasis placed on the techniques that we are interested in, such as the techniques using the Gabor wavelet transform, using the Fourier transform from the spectroface, and using the discrete wavelet transform as in the papers related to stereo vision problem. Through implementation and experimentation, we hope to provide a more detailed comparison between these three popular transformation and matching techniques.

From the strength and weakness of each of the techniques in the comparison, we propose our own hybrid model, a face recognition system combining the interesting part of each technique with a different database to class representation scheme. We will justify our

face recognition system with the phases of implementation, experimentation and statistics on results through the performance measuring scheme.

1.4. Statement of the Face Recognition problem

1.4.1. Spatial transformation and pose differences

Spatial transformation of the input image, or at least sub-area of it, is required when the face image is not acquired under strictly controlled environment. Under these conditions, one face image may have a different scale, or even different pose with the ones used in the training session.

The scale of the face, when different in input, requires a normalization process, which is based on a stretching transform with respect to several feature points such as eyes, nose or mouth. These feature points may be located manually in an assisted environment, or automatically with a features template matching scheme, or low level processing based on facial geometry [59].

The pose represents a greater challenge because it requires larger modification to the facial features as well as the global face structure. The pose problem is the result of the change of the viewpoint or from the rotation of the head in three dimensions, also referred to as in-plane or out-plane rotation. In the case of an in-plane rotation, the subject is still under frontal view and the facial features still remain almost the same, with little changes. In the case of an out-plane rotation, the subject is no longer under frontal view, and the facial features may change substantially.

1.4.2. Illumination

Face illumination, as in the case of view and pose described previously, is a challenging problem under an uncontrolled or natural environment. The same person, with the same facial expression, and seen from the same viewpoint, can appear dramatically different when the lighting condition changes.

1.4.3. Facial Expression

Most face recognition systems require training data composed of faces under neutral expression. The reason is that the facial expression problem, such as a simple smile, is very different from the ones caused by the effect of scale, pose and illumination. While the latters can have the system classify the face as a non-face, the facial expression changes may have the system still recognize the face as a face, but of someone else. The facial expression analysis is in itself an entire field of study as it is a very important step in computer-human interaction.

1.4.4. Disguise

While disguise is a very common problem confronted by face detection and recognition, it is not often addressed in the research literature, beside work on glasses. Although additional artifacts, such as glasses, hairstyle, beard and moustache, or makeup, may help the face recognition process for human, it is a serious disturbing factor for machine recognition. Face recognition system, especially for authentication purpose, often consider small variations of the face image without having this one evolved, as in the

case of dealing with facial aging. In certain cases, the disguise problem may even be referred as the oclusion problem.

1.5. Databases

In the past, the performance of different face recognition and detection systems as reported in the literature has been measured on small databases, with experiments carried out on their own databases. The majority of these databases are collected under very controlled situations. However, depending on the desired operation, face detection or face recognition, the algorithms used to collect images, are quite different and the image data obtained varies from frontal, full profile or of free orientation.

1.5.1. Face Detection databases

The BioID face database is one of the most recently introduced databases. It is recorded and published to give all researchers working in the area of face detection the possibility to compare the quality of their face detection algorithms with others. During the recording special emphasis has been laid on "real world" conditions, therefore the test set features a large variety of illumination, background and face size. The data set consists of 1521 grey level images with a resolution of 384x286 pixel. Each one shows the frontal view of a face of one out of 23 different test persons. For comparison reasons, the set of manually located eyes positions, sometimes known as the ground truth, is given by the company BioID (see figure 1).

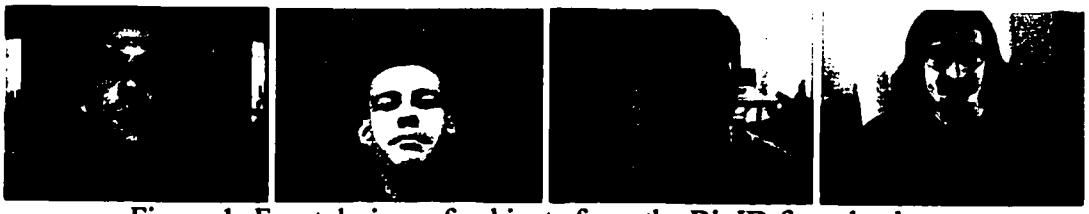


Figure 1: Frontal view of subjects from the BioID face database.
(Source The BioID Face Database, available free of charge online
http://www.bioid.com/technology/facedatabase.html)

The National Institute of Standards and Technology (NIST) offers a very interesting set of faces, called the Mugshot Identification Database. The NIST Special Database 18 is being distributed for use in development and testing of automated mugshot identification systems. It contains 3248 images of 1495 male and 78 female individuals and offers both front and side views when available (see figure 2).

Figure 2: Front and profile views of samples from the NIST Special Database 18.
(Source the NIST Special Database 18, available for order online
http://www.nist.gov/srd/nistsd18.htm)

One of the most famous face databases available on the Internet for face detection is without doubt the Carnegie Mellon Test Images (see figure 3). The image dataset is used by the Carnegie Mellon University (CMU) Face Detection Project and is provided for

evaluating algorithms for detecting frontal views of human faces. This particular test set was originally assembled as part of work in Neural Network Based face detection and combines images collected online at CMU and MIT. It also offers a ground truth such as the x and y coordinates of the eyes and mouth features in the image.

Figure 3: Carnegie Mellon Test Images.
(Source CMU, available free of charge online
http://vasc.ri.cmu.edu/idb/html/face/index.html)

Several more databases do exist and are available from the Internet. They mostly belong to academic research laboratories at different universities, but may come from private laboratories as well.

1.5.2. Face Recognition databases

The AT&T Laboratories offer a set of faces taken between april 1992 and april 1994 at the lab in Cambridge, UK (see figure 4). The database was used in the context of a face recognition project carried out in collaboration with the Speech, Vision and Robotics Group of the Cambridge University Engineering Department. Because the previous name of the lab is Olivetti Research Laboratory, the face database is often referred as the ORL database.

There are ten different images of each of 40 distinct subjects. For some subjects, the images were taken at different times, varying the lighting, facial expressions such as open and closed eyes, smiling and not smiling, and facial details such as with glasses or no glasses. All the images were taken against a dark homogeneous background with the

subjects in an upright, frontal position with tolerance for some small side movement such as some tilting and rotation of up to about 20 degrees, and some variation in scale of up to about 10%. The size of each image is 92x112 pixels, with 256 gray levels per pixel.

Figure 4: ORL face database with 10 images each of the 40 subjects.
(Source AT&T Laboratories Cambridge, available free of charge online http://www.cam-orl.co.uk/facedatabase.html)

In order to overcome the problem of meaningful comparisons between several databases, the Face Recognition Technology (FERET) facial database from the US Department of Defense Counterdrug Technology Development Program Office has often been designated as *standard* (see figure 5). Recent leading commercial face recognition systems are evaluated using the FERET facial database such as the FRVT (Face Recognition Vendor Tests) 2000 tests.

The FERET database comes from a program ran from 1993 through 1997 sponsored by the Department of Defense's Counterdrug Technology Development Program through the Defense Advanced Research Products Agency (DARPA). The FERET database contains two sets of a total of 14,051 images, of which 3,816 are frontal images but most previous work on FERET data has focused attention on the frontal images. Each probe set contains one or more images from a set of persons where each person will have exactly one match in the gallery. The gallery may contains images from other persons who are not in the probe population.

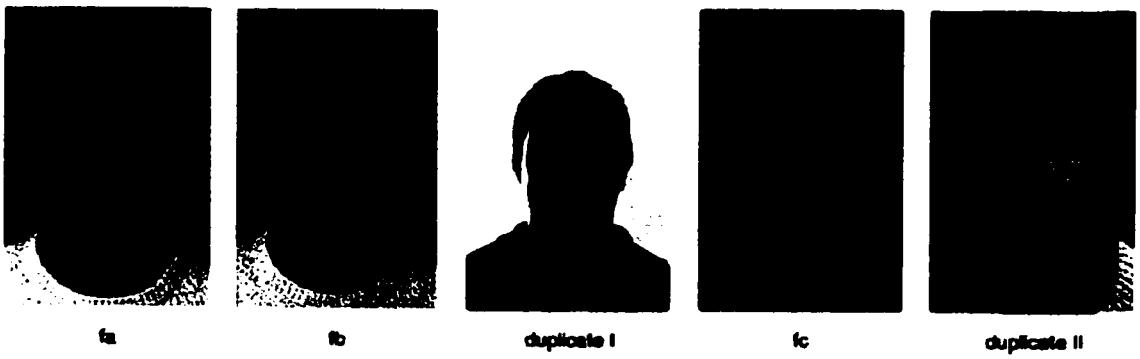


Figure 5: Examples of different categories of probes
(Source P. Jonathon Philips, IEEE Transactions of Pattern Analysis and Machine Intelligence, pp 1092, Vol 22, No 10, Oct 2000)

1.6. Applications of Face Recognition

1.6.1. Biometric authentication and identification

A definition of biometrics, according to the Biometric Consortium, is the automatic recognition of a person using distinguishing traits. It is the field dedicated to person identification and authentication based on the physiological or behavioural characteristics of the user, such as face images, facial thermogram, fingerprint, hand geometry, iris, retinal scan, handwritten signature and voice patterns.

A biometric system is a pattern recognition system that consists of two units, the enrolment module and the identification module. When the user enrols into the system, a biometric template, data describing their biometric, is created and stored either in a database or on a portable token such as a chip card, also known as smartcard. Upon verification, this template is retrieved and compared against the live sample within a predefined matching tolerance level. If the templates match, then a 'true' message is generated by the matching system.

Biometric systems based on face recognition have some advantages over more precise systems, such as those based on fingerprint or facial thermogram, as they are examples of non-intrusive techniques for person identification.

Figure 6: Crowd surveillance and scanning with FaceFinder.
(Source FaceFinder from Viisage online http://www.viisage.com/facefinder.htm)

Biometric implementations based on face detection and recognition range from simple desktop access control, such as TrueFace ID from eTrue, to a much larger scale implementation such crowd surveillance solution in train stations and airports, such as FaceFinder from Viisage (see figure 6).

Chapter 2

Approaches to Face Recognition

2.1. Introduction

Because the number of publications related to face detection and recognition is quite large, an exhaustive review may be difficult. The review of the state of the art, which follows, examines publications that represent many or most of the approaches explored in the literature and the most relevant to our work.

2.2. Simple template

2.2.1. Grayscale

One of the simplest face detection and then face recognition systems is based on geometrical features obtained from grayscale images. When the face database contains images, with a simple background, to be normalized, the grayscale approach is very efficient computationally.

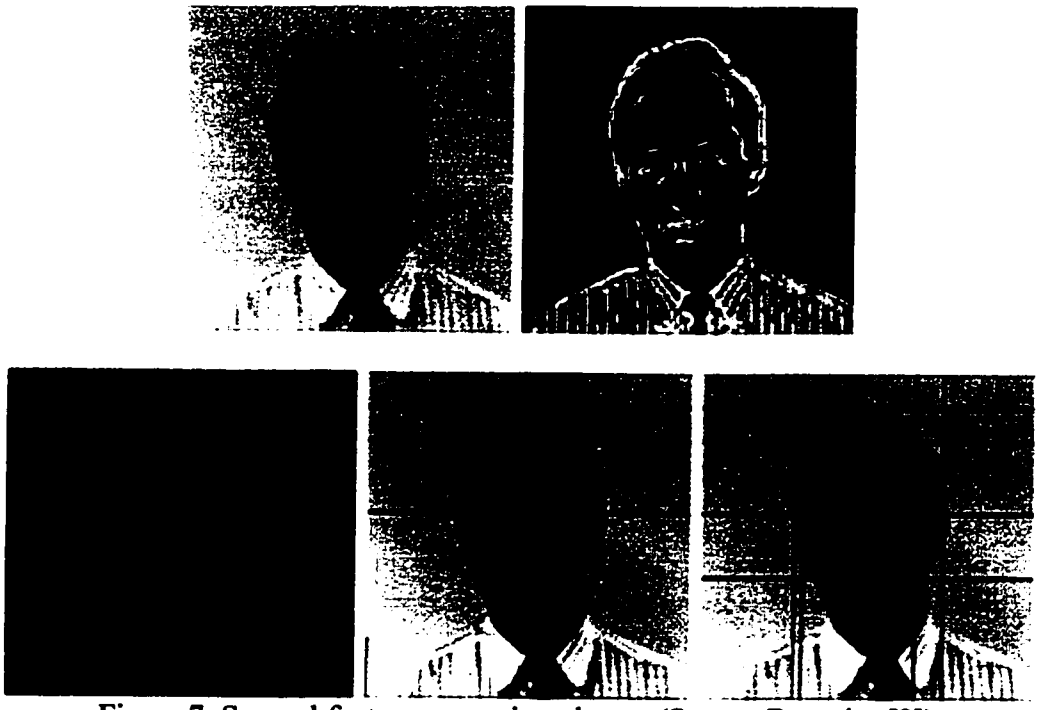


Figure 7: Several feature extraction phases. (Source Beumier [9])

The feature extraction phase, as proposed by Beumier [9], would compute the edges, extract the head and its contour, then compute the normalized distances between lines, vertical and horizontal references based on the grey level profile (see figure 7).

While being a very simple technique, the grayscale method is very sensitive to scale, lighting conditions, such as shading, and pose, such as in-plane and out-plane rotations.

2.2.2. Entropy

The face detection system as proposed by M. Lew and N. Huijsmans [2] is a view-based template matching system based on the Kullback measure of relative information [3]. As the Kullback relative information measures the class separation, the most informative pixels (MIP) may be defined as the ones which give the maximum class separation (see figure 8).

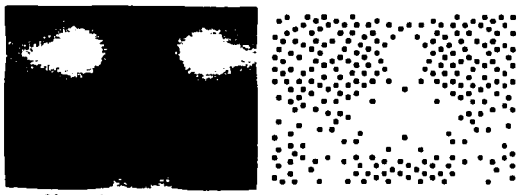


Figure 8: Histogram equalized Kullback image for the eye/nose template and its MIP. (Source M. Lew and N. Huijsmans [2])

The algorithm of the face detection system first tries to find the set of most informative pixels by using the Kullback relative information measures. Then, for every interesting image scale and every 23x32 window, when the distance from feature space metric (DFFS) to the face cluster is lower than the one to the non-face cluster, it assumes that a face is within the search window.

As stated by M. Lew and N. Huijsmans, the algorithm works quite well on the Massachusetts Institute of Technology (MIT) database, 23 images of group team photos

with a total of 149, on the Carnegie Mellon University (CMU) *subset* database, 42 scanned photographs with 169 faces, and especially on the Leiden 19th century portait database, 494 images containing 574 faces (see figure 9).

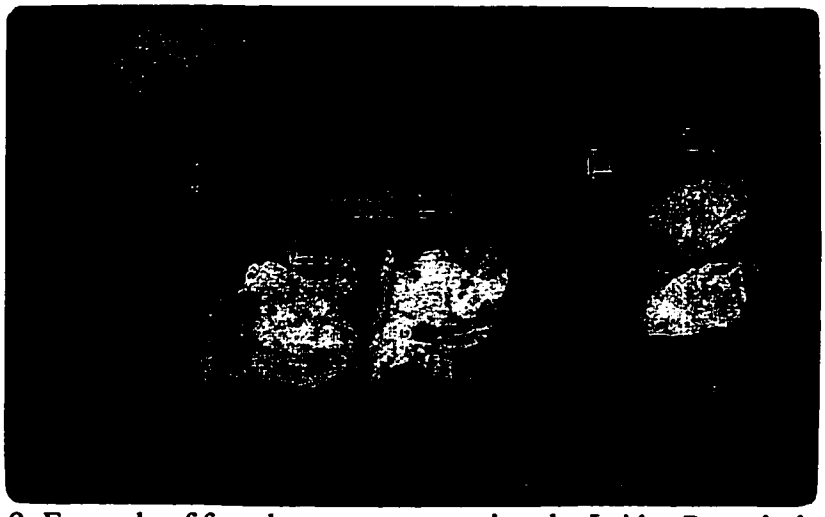


Figure 9: Example of face detector output using the Leiden Portrait database. (Source M. Lew and N. Huijsmans [2])

2.2.3. Colour correlogram

Query By Image Content (QBIC) systems often make wide use of colour histograms, because of its robustness property and speed over large digital libraries. However, as colour histograms only encode coarse characterization of an image, without any spatial information, it results in images with very different appearances having similar histograms. J. Huang and R. Zabih [8] suggest a method combining color information with spatial layout, while retaining the advantages of histograms, called color correlogram. This method computes the spatial correlation of pairs of colors as a function of the distance between pixels. It produces a table indexed by color pairs, where the d^{th} entry for row *(i, j)* specifies the probability of finding a pixel of color j at a distance d from a pixel of color i in this image.

Although the colour correlogram method is not a face detection or face recognition technique by itself, it can be a quite computational efficient technique during the pre-analysis phase.

2.2.4. Skin tone

Skin tone detection over a complex background offers many advantages such as speed, robustness under partial occlusion, out-plane and in-plane rotation, scale changes and resolution (see figure 10). However, skin tone usually helps the face localization phase to find face candidates rather than being used for the face recognition phase itself.

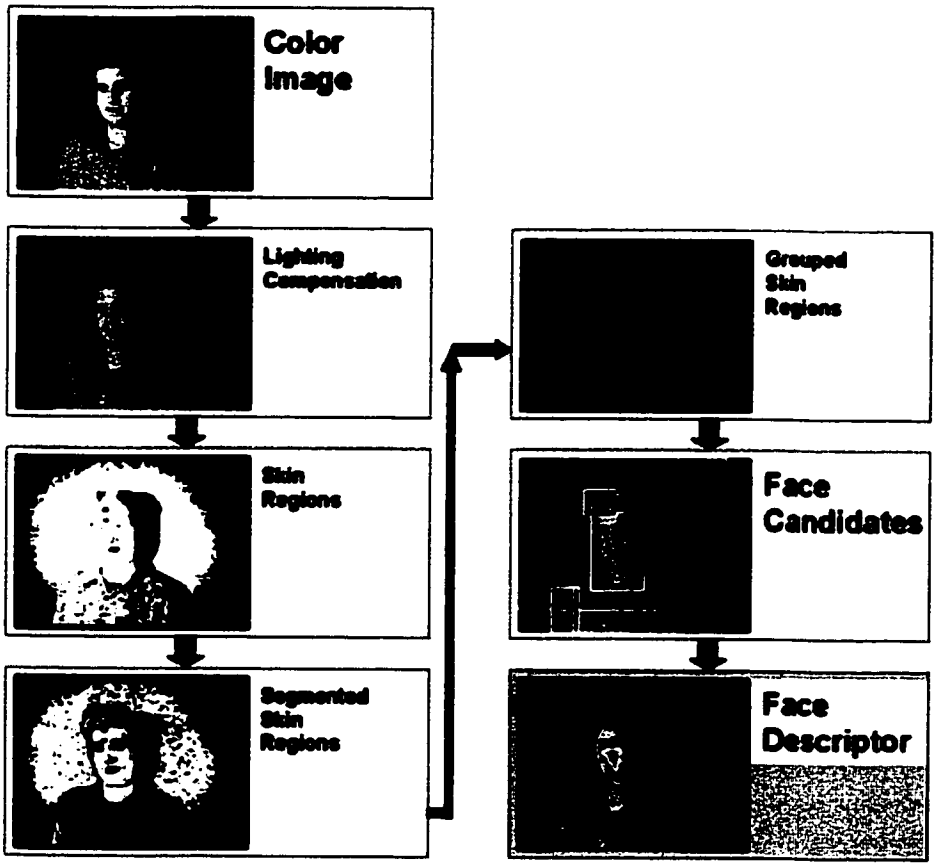


Figure 10: Face detection algorithm. (Source Hsu et al. [4])

Hsu et al. [4] propose a face detection system where skin tone method is used to detect skin regions over the entire image, and then generates face candidates based on the spatial arrangement of these skin patches. The localization of facial features is a template-based approach on several ellipses, ratio templates of YCbCr components of the image (see figure 11).

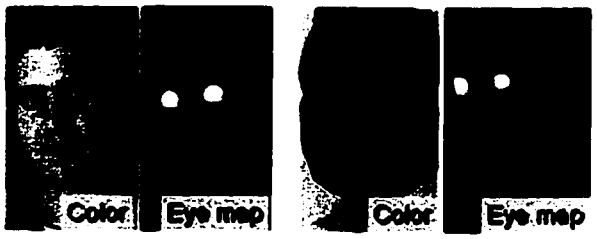


Figure 11: Construction of eye maps. (Source Hsu et al. [4])

Instead of using the YCbCr colour space, widely used in the video compression standards, several works have been using the Hue Saturation Luminosity (HSL) components to develop a skin-colour model in the chromatic colour space (see figure 12).

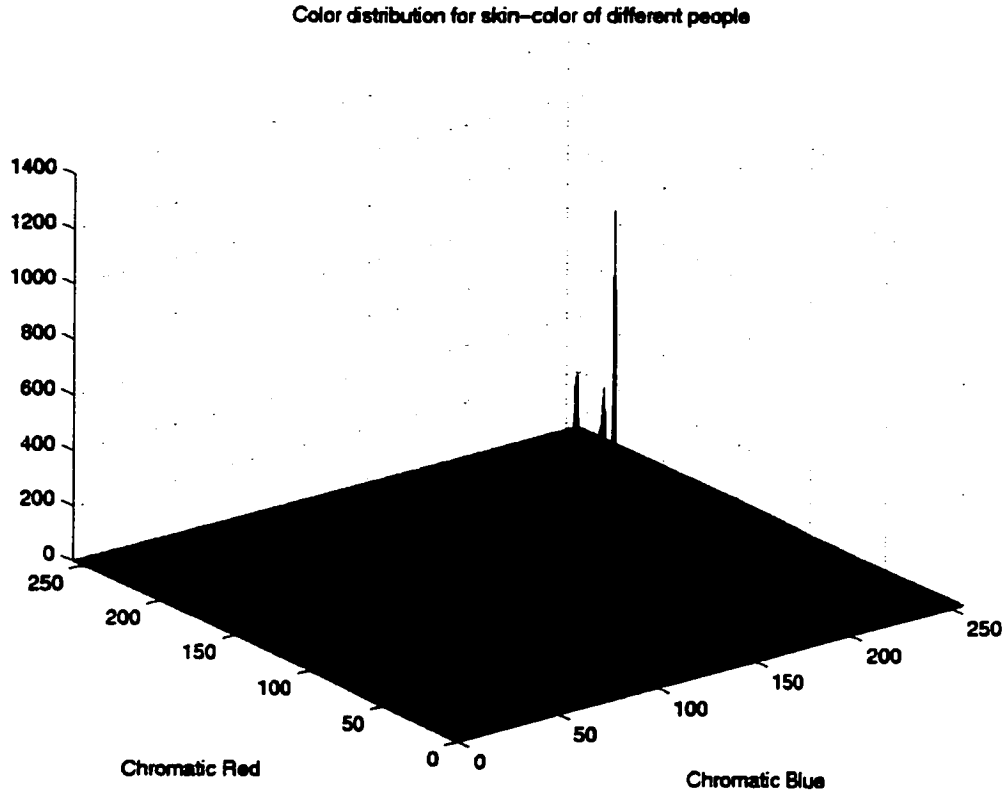


Figure 12: 32500 skin samples from 17 color images within different ethnicities: Asian, Caucasian and African. (Source H. Chang and U. Robles)

The colour histogram (see figure 12) reveals that the distribution of skin colour of different people are clustered in the chromatic colour space and a skin colour distribution can be represented by a Gaussian model N(mean, covariance). Under this fitting scheme, each pixel can be transformed into a likelihood of the pixel belonging to the skin. Then, with appropriate thresholding and combined with binary open-close operations, the result can be further be transformed into a binary image with skin regions and non-skin regions (see figure 13).

Figure 13: Colour image with skin like pixels (replaced by its HSL) and contour.

Often, when speed is traded for robustness, the method is quite sensitive to lighting condition, white balancing level used by the camera, noise, shading, variant reflection due to the complex structure of the human face. The skin tone approach is more efficient as a preliminary analysis for face detection than a method on its own.

2.2.5. Symmetry detection

A computational approach to face recognition is to use symmetry for face detection in images, such as proposed by C. Tyler and R. Miller [1]. The algorithm describes a symmetry axis extension procedure that finds local symmetry points by looking for lateral matches axes and then builds a symmetry axis by joined symmetry points with a line extension procedure (see figure 14). C. Tyler and R. Miller assumed that a face distinguishes itself from its surroundings by its bilateral symmetry.

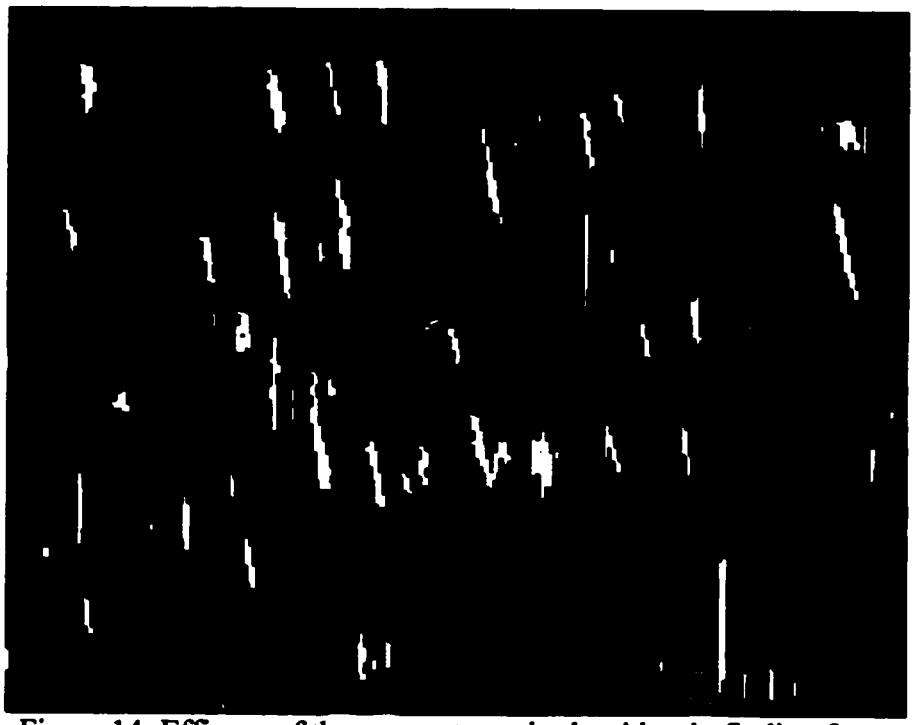


Figure 14: Efficacy of the symmetry axis algorithm in finding faces (Source C. Tyler and R. Miller [1])

However, the quality of the symmetry of the image is greatly affected by an out-plane rotation, or when illumination is asymmetric (see figure 15). In order to improve the algorithm, C. Tyler and R. Miller included some pre-filtering that enhanced edges relative to uniform areas in order to reduce the effects of asymmetric illumination (shadows ...) and the use of Eigentemplates approximated by Gabor filters in a preliminary analysis in a cluttered scene (see figure 16).

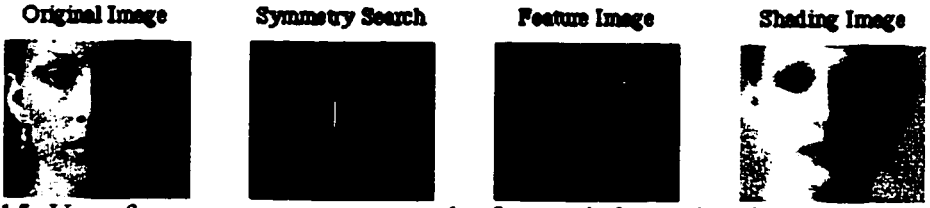


Figure 15: Use of symmetry to segregate the feature information from the illumination shading. (Source C. Tyler and R. Miller [1])

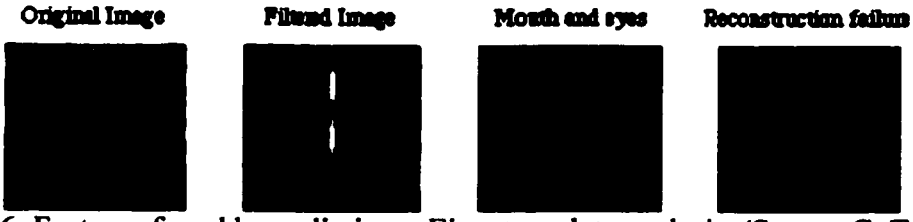


Figure 16: Features found by preliminary Eigentemplate analysis. (Source C. Tyler and R. Miller [1])

2.2.6. Receptive fields

Template-based face recognition systems often use several face images during the training phase. M. Lando and S. Edelman [21] take another approach by describing a computational model of generalization from a single view in face recognition, using receptive fields (RF). Largely based on the biological visual systems, this approach comes from the hypothesis that a natural basis for the definition of similarity is derived from the concept of processing units with localized receptive fields.

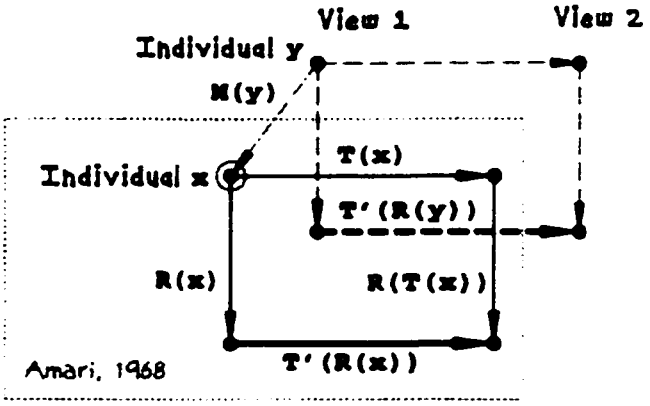


Figure 17: Same transformation to obtain similar changes in the internal RF-space. (Source M. Lando and S. Edelman [21])

The scheme is based on the assumption that such a generalization is made possible by previous experience of the visual system with similar views from other faces (see figure 17). From the computational point of view, M. Lando and S. Edelman assume that applying the same transformation (change of viewpoint or illumination) to images of

different faces results in similar changes in the internal RF-space, representation, of each face (see figures 18 and 19).

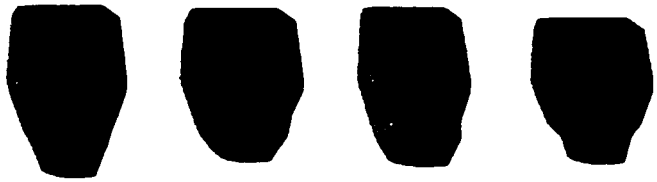


Figure 18: Faces obtained by random variation of the parameters in the face geometry model. (Source M. Lando and S. Edelman [21])

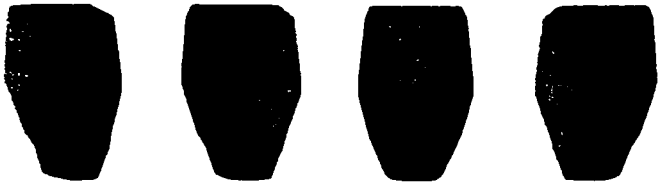


Figure 19: One of the face models rendered under different viewpoint and illumination conditions. (Source M. Lando and S. Edelman [21])

The mapping approximation is then obtained with a trained radial basis functions (RBF) neural network classifier. Under tightly controlled illumination and viewpoint, a database of twenty images of 18 different male faces is obtained, and is used during experiments to give a performance result of 59% to 66%. It is worth noting, that the model is more an approach to the theoretical understanding of the computational basis of class-based generalization in human vision than a practical computational approach.

2.2.7. View based template

Template-based face recognition systems have been quite successful on frontal views of face images, under controlled lighting conditions. Confronted with the problem of faces under varying pose, which includes scale, in-plane and out-plane rotation, D. J. Beymer [15] proposes a recognizer that tries to estimate an arbitrary pose before recognition. The

proposed recognizer combines a facial features finder, a pose estimation module, a geometrical alignment module and finally a template-matching module.

In the first stage, the features finder locates two eyes and at least one nose feature (see figure 20). The locations are used to bring the input face image into *rough* geometrical alignment with model faces, which includes position, scale and in-plane rotation.

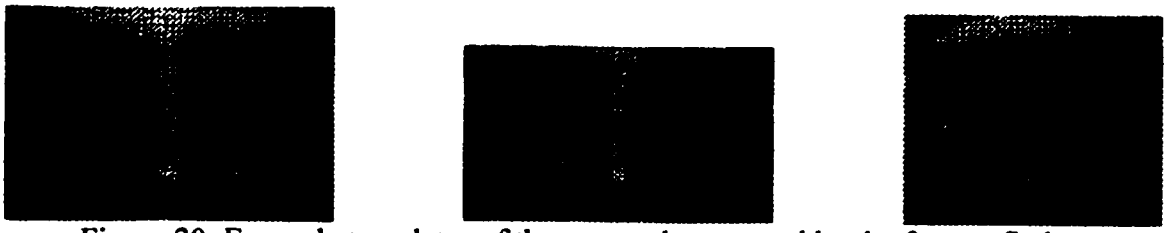


Figure 20: Example templates of the eyes and nose used by the feature finder.
(Source D. J. Beymer [15])

In order to estimate the out-plane rotation, templates generated from different poses are compared to the image from previous step, using a face similarity measure through correlation. The pose estimation module then selects only models whose pose is similar to the input's pose.

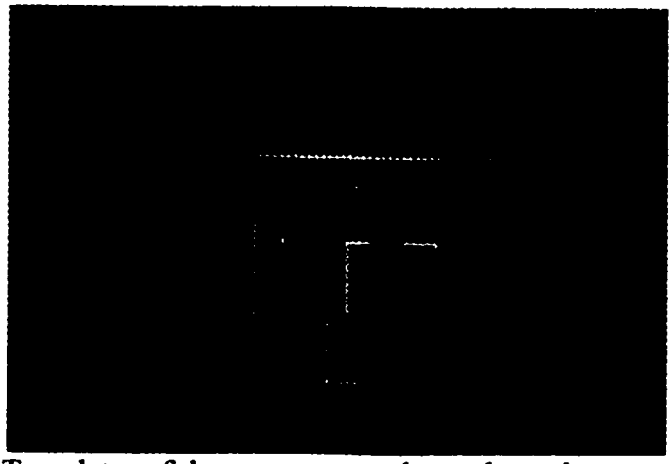


Figure 21: Templates of the eyes, nose and mouth used to represent faces.
(Source D. J. Beymer [15])

The final template-matching module has five pyramidal levels, from a general estimation of the full face to a sub-template centered around the iris center, or nose lobe feature (see

figure 21). Template-matching is performed by using normalized correlation on processed versions of the image and templates, and is defined as

$$r = \frac{\langle TI \rangle - \langle T \rangle \langle I \rangle}{\sigma(T)\sigma(I)}, \tag{5}$$

where T is the template, I is the sub-portion of image being matched against, < > is the mean operator and σ() measures the standard deviation.

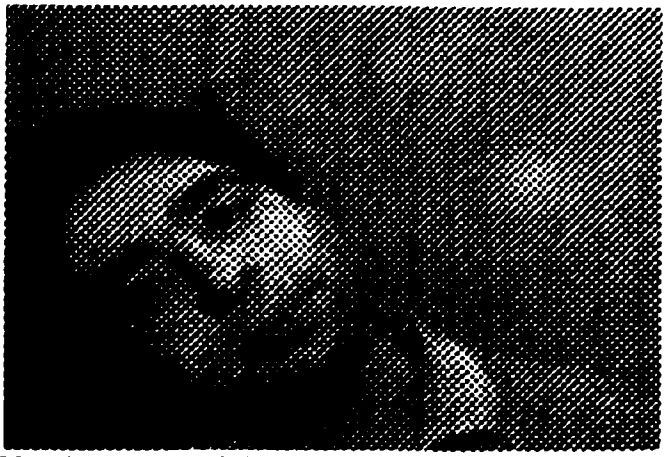


Figure 22: Iris and nose lobe features located by the feature finder.
(Source D. J. Beymer [15])

According to D. J. Beymer, the recognizer has achieved a recognition rate of 98% on a database of 62 people, with 930 modeling views and 620 testing views (see figure 22). However, more work needs to be done to handle expression and lighting conditions.

2.2.8. Geometrical face model

While most face detection and recognition systems based on morphological measures suffer from bad lighting condition, skew facial features and even facial expressions, S. Jeng, H. Y. M. Liao, C. C. Han, M. Y. Chern, Y. T. Liu [25] introduce a geometrical face model, constructed according to the relative geometrical relation among facial organs on a face to overcome problems experienced in Beymer's recognizer.

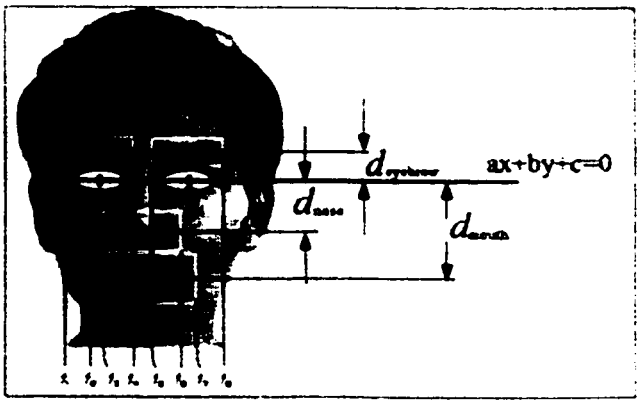


Figure 23: Geometrical face model.
(Source S. Jeng, H. Y. M. Liao, C. C. Han, M. Y. Chern, Y. T. Liu [25])

In the proposed system, the eyes and mouth are considered as primary features, while eyebrows and nostrils are considered as secondary features (see figure 23). The proportions between these facial features are used to group the detected features through a grouping algorithm (see figure 24). The feature blocks are then regarded as facial feature candidates.

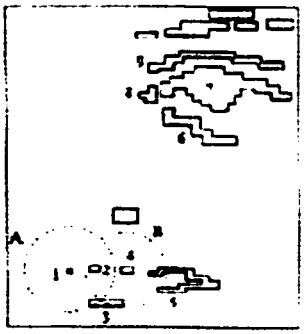


Figure 24: Grouping example.
(Source S. Jeng, H. Y. M. Liao, C. C. Han, M. Y. Chern, Y. T. Liu [25])

Because of illumination problems, common to this type of algorithms, the preprocessing is an important step in which an image face will be thresholded to produce a binary sketch-like image, including opening operation and high boost filtering (see figure 25).

Figure 25: Results after performing the preprocessing step.
(Source S. Jeng, H. Y. M. Liao, C. C. Han, M. Y. Chern, Y. T. Liu [25])

While S. Jeng, H. Y. M. Liao, C. C. Han, M. Y. Chern and Y. T. Liu claim a detection rate of 86%, 98 successfully out of 114 face images, the detection scheme from a cluttered scene, as well as the partial covering problem, is still to be improved.

2.3. Deformable template

2.3.1. Shape models

Gabor feature jets have previously been used for static face analysis with elastic graph matching and coarse to fine correspondance matching, from the work of R. P. Würtz [71]. However, each feature point is treated independently with no global shape model as constraints. S. J. McKenna, S. Gong, R. P. Würtz, J. Tanner, D. Banin [11] propose an approach, called shape models, that gives the ability of tracking facial feature points. The proposed approach incorporates the Gabor feature jets with the point distribution models (PDMs). The PDMs provide a method for representing flexible objects by means of a set of feature points describing a deformable shape.

Figure 26: Example of training set before alignment (top) and after (bottom).

(Source S. J. McKenna, S. Gong, R. P. Würtz, J. Tanner, D. Banin [11])

As described by S. J. McKenna, S. Gong, R. P. Würtz, J. Tanner and D. Banin, an object's shape is represented as a 2n-dimensional vector of n image coordinates, $s=(x_0, y_0, x_1, y_1, \ldots, x_{n-1}, y_{n-1})$ which corresponds directly to the location of Gabor feature jets (see figure 26). It is worth noting that these locations need not lie on any meaningful contours. The translation, scaling and rotation operations are normalized using an iterative object alignment algorithm, which is achieved by rotating, translating and scaling shapes in order to minimize the weighted sum of squares distance between them. And while the shape models have been initially proposed to track facial feature points in video sequences (see figure 27), it may be useful for face detection and recognition because of its robustness property.

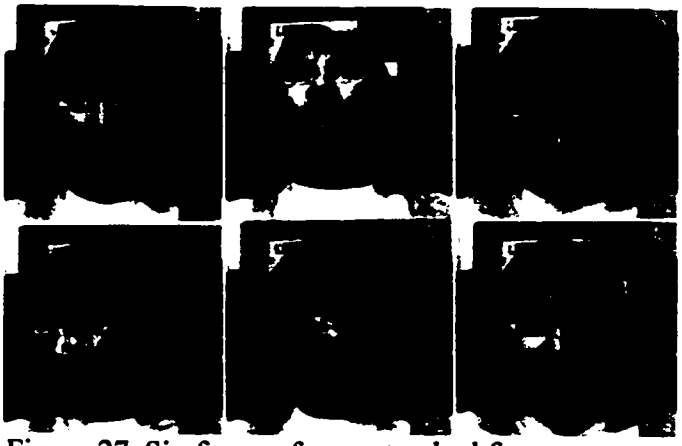


Figure 27: Six frames from a tracked face sequence.
(Source S. J. McKenna, S. Gong, R. P. Würtz, J. Tanner, D. Banin [11])

However, the fact that PDMs are linear makes them inappropriate for modeling non linear effects such as bending or rotation.

2.3.2. Model based

Similar to the shape models described earlier, A. Lanitis, C. J. Taylor and T. F. Cootes propose a statistical approach using point distribution models (PDM) to provide a compact and parametrized description of shape for an instance of a face [16]. The approach is divided into two main phases: the modeling, which generates the flexible models of facial appearance and the interpretation, which uses the models to code and interpret face images. In the modeling phase, the shapes of facial features and their spatial relationships are combined in a single flexible shape model. The models derived from a set of training images are generated by a statistical analysis of the positions of the landmark points (see figure 28).

Figure 28: Locations of model points on a training image (left) and the mean shape (right). (Source A. Lanitis, C. J. Taylor and T. F. Cootes [16])

In the interpretation phase, when a new image is presented to the system, its facial features are located, based on the flexible shape model obtained during training. The new face is then deformed to the mean face shape and the resulting set of appearance parameters can be used for face recognition. By integrating the gray level information into the shape model, A. Lanitis, C. J. Taylor and T. F. Cootes obtain a peak result of 94% of classification on a database of 20 individuals for training and 100 test images from 10 individuals not in the database (see figure 29).

Figure 29: Examples of detected faces with different out-plane rotations.
(Source A. Lanitis, C. J. Taylor and T. F. Cootes [16])

However, as the authors of the proposed algorithm suggest, a larger training and test sets are needed to better characterize its performance.

2.4. Principal component analysis approach

Template based face recognition systems often perform poorly on large face databases. In order to reduce the dimension of the database without key information lost, T. E. de Campos, R. S. Feris and R. M. Cesar Junior [36] use the principal component analysis (PCA) technique, based on the Karhunen-Loeve (KL) transformation of images, as the main part of the proposed face recognition system. The principal component analysis technique is often combined with other classification techniques, such as neural networks.

An image database of faces, also referred as face space, is often defined as N individuals of M images each. The principal component analysis technique allows reducing the face space by projecting it into a low dimensional linear subspace, which corresponds to the first L principal components, called eigenvectors, of the N by M images. The principal components are the eigenvectors of the covariance matrix W, which is defined as

$$W = X \times X^{t}, \tag{7}$$

where X is the array constructed from the database, with each column of X is an one dimension vector transformed from the two dimensional image by appending line after

line, and X^t is the transposition of the matrix X. In this context, the first eigenvector of W is oriented in the direction of the largest variance among the faces, and corresponds to an average face as it has features shared to all the pictures (see figure 30). By definition, each eigenvector, called Eigenface, is orthogonal to the others.

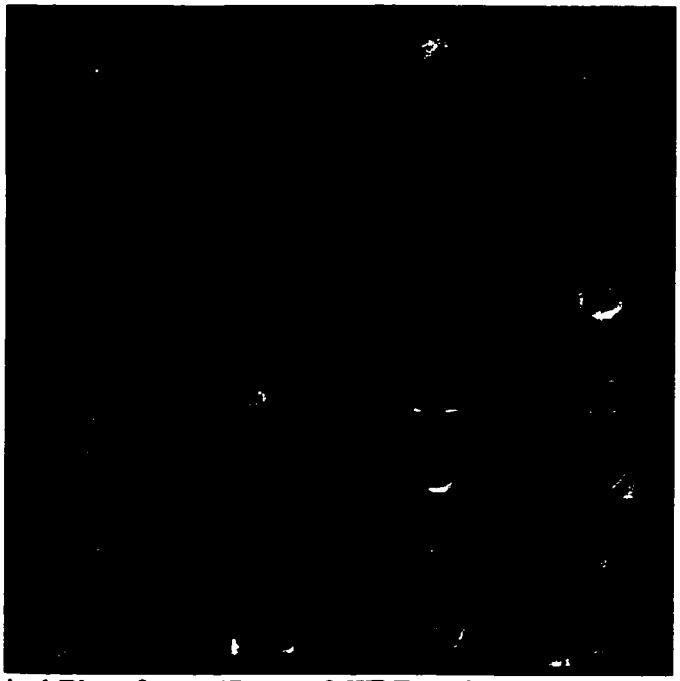


Figure 30: Typical Eigenfaces. (Source MIT Face Recognition online home page)

Although the principal component analysis technique often yields nice performance, it is quite expensive computationally because all pixels in the image are necessary to obtain the representation used to match an image with all others in the database. T. E. de Campos, R. S. Feris and R. M. Cesar Junior propose the use of Eigeneyes, referring to the application of the principal component analysis technique in restricted areas of the image, such as the eyes region (see figure 31).

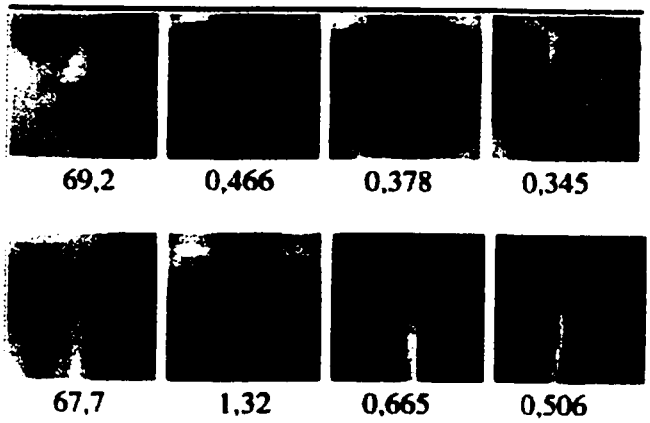


Figure 31: Images of the first four eigenvectors with their respective eigenvalues of the faces database and eyes database.
(Source T. E. de Campos, R. S. Feris and R. M. Cesar Junior [36])

From the database of 96 images, with sixteen subjects and six different images for each subject, the recognition rate reaches 62.5% with 48 Eigeneyes, while getting only 43.75% with the 48 Eigenfaces. The performance difference is explained because with the inclusion of the nose and mouth region, face expressions imply strong distortions and reduce the recognition performance.

While presenting attractive advantages such as direct recognition without any significant low-level processing, low-dimensional subspace representation and especially simple and efficiency against other methods, the principal component analysis technique suffers from some limitations: it is sensitive to scale, and needs an accurate scale normalization process, it is sensitive to the pose problem and illumination because of its use of the raw intensity data of the image, and it is also sensitive the problem of cluttered scene as experiments are usually done with an uniform plain background. The learning process is also one of its disadvantages, as it is very time consuming and requires the number of face classes to be larger than the dimension of the face space. The recognition process is expensive as well because it is correlation-based, as the test image has to be correlated with each of the Eigenfaces.

2.5. Filter-based approach

2.5.1. Belief network

Various face detection and recognition systems use a filter approach to detect feature points that are unique to the structure of the human face. However, too many unrelated feature points may lead to false candidates, and even false rejections. K. C. Yow and R. Cipolla propose an approach of grouping feature points into different belief networks [18] (see figure 32).

Figure 32: From left to right: input image, filter output G, features detected using G only, filters used, energy output E, features detected using G and E.
(Source K. C. Yow and R. Cipolla [18])

The Gaussian derivative filters are quite robust and can even detect facial features when the face is being viewed from a large angle from the front. However, they often produce a weaker response in comparison with responses from the filters on the background. A high threshold applied to the filter response to eliminate background features may lose facial features, while a low threshold may let too many background points pass through to the grouping stage, a combination of Gaussian derivative filters and its Hilbert transform, at different scales and orientations, is used to produce an oriented energy output

$$E_\theta = G_\theta^2 + H_\theta^2, \tag{6}$$

where θ is the orientation of the filter, G_θ is the convolution output of the filter at orientation θ and H_θ the convolution output of the Hilbert transform.

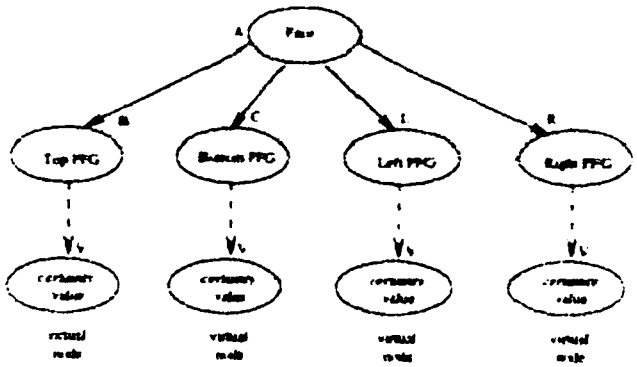


Figure 33: Face modeled as a belief network. Each of the child node will propagate evidence to the parent node depending on whether the associated partial face group (PFG) is present. (Source K. C. Yow and R. Cipolla [18])

A belief network is a way of representing the conditional independence relationship between a set of variables. It is a directed acyclic graph (DAG) where the nodes represent a set of random variables and the links represent the influence of a parent node over a child node (see figure 33). In the approach proposed by K. C. Yow and R. Cipolla, it is used to propagate evidence in order to reject false face candidates.

Figure 34: Results of face detection.
(Source K. C. Yow and R. Cipolla [19])

While the algorithm offers interesting results, with 90% detection rate on a 60 test images, and illumination invariance due to the Gaussian derivative filters (see figure 34),

it is still sensitive to extreme illumination conditions as well as occlusions and facial expressions.

2.5.2. Random labeled graph matching

Face detection and recognition systems often introduce computational difficulty because of the extracted features matching process, which are not perfectly reliable. Since the spatial arrangement of the features may be used to localize the face and further match it, T. K. Leung, M. C. Burl and P. Perona propose a statistical model based on the mutual distances between facial features [26]. The arrangement of facial features is modeled as a random graph in which the nodes correspond to the features and the arc lengths correspond to the distances between these features, which are different with different people. By assuming that anthropometric data are jointly Gaussian if the specimens belong to the same population, T. K. Leung, M. C. Burl and P. Perona suggest that the arc lengths are jointly Gaussian distributed with some mean and covariance.

They use in the experiments a multi-orientation, multi-scale Gaussian derivative filters as local detectors to detect the candidate feature points. After the convolution of the image with a set of Gaussian derivative filters at different orientations and scales, the obtained vector of filter responses at a particular spatial location gives a description of the local image brightness, and is matched against a template database. The feature point is then considered as part of the pool candidates if the match exceeds a certain threshold.

Because finding the best constellation of the feature points can be viewed as a problem of random graph matching, the complexity of matching a full graph is M^N, where M is the number of candidate points in the image and N is the number of nodes in the template

graph. In order to reduce the complexity, T. K. Leung, M. C. Burl and P. Perona exploit the statistical structure of the graph by estimating the locations of the other features given the positions of certain features. By knowing where the other features should be found and how much variance exists in their expected locations, the number of constellations formed is greatly decreased. The constellations are formed only from candidates that lie inside the appropriate search ellipses (see figure 35).

Figure 35: Location of missing features estimated from two points. The ellipses show the areas which include the missing features with high probability.
(Source T. K. Leung, M. C. Burl and P. Perona [26])

Figure 36: Best correct match and best incorrect match.
(Source T. K. Leung, M. C. Burl and P. Perona [26])

The performance rate of the proposed system reach 89% over quasi-frontal views of faces, collected from 150 frames in laboratory, and 95% on a database of 180 images of 18 subjects under well-controlled situations, such as quasi-frontal views with plain white background and same lighting conditions (see figure 36). However, the authors acknowledge several extensions to the algorithm as it is quite dependent of the quasi-frontal view and accepts only in-plane rotation.

2.5.3. Monte-Carlo

According to J. Matas, K. Jonsson, J. Kittler [12], approaches using principal component analysis (PCA) and dynamic link architecture (DLA), while obtaining high recognition rates, are too sensitive to changes in scale, rotation and illumination condition. The normalization and segmentation processes typically depend on the reliability of facial detectors. By proposing an integrated approach, where localization and normalization are achieved simultaneously, J. Matas, K. Jonsson and J. Kittler hoped to decrease this dependence.

When a test image is compared to reference images, the search is made in the space of all affine transformations of the test image, augmented by the space of all linear mappings of the reference images. The direct approach must then evaluate many correlations per verification, which is extremely computationally inefficient. In the proposed algorithm, J. Matas, K. Jonsson and J. Kittler evaluate the correlation in a Monte-Carlo fashion, a more efficient statistical method, by estimating the correlation from a small sample of suitably chosen points (see figure 37).

Figure 37: Example of matching score: short 1 and 2, combination and response.
(Source J. Matas, K. Jonsson and J. Kittler [12])

As stated by J. Matas, K. Jonsson and J. Kittler, the Monte-Carlo technique is twenty five times faster than regular correlation and is expected to outperform techniques based on

DLA while experimenting on 280x350 image resolution from the M2VTS database of 5 shots of 3 sequences of 37 persons.

2.5.4. Direct convexity estimation

While edge detection is so far the core of most techniques based on attentional mechanisms, edge based approaches suffer severe flaws from illumination and surrounding objects in a cluttered scene. As the face is considered as a three-dimensional objects with convex and concave regions, A. Tankus, Y. Yeshurun and N. Intrator suggest an attentional operator Y-phase [14], which detects smooth three-dimensional convex or concave objects in the image by processing directly the intensity values of the image (see figure 38).

Figure 38: Face detection using attentional Y-phase operator.
(Source A. Tankus, Y. Yeshurun and N. Intrator [14])

The attentional mechanism Y-phase is defined as the derivative of the *argument*, denoted as phase, with respect to the y-direction.

$$\frac{\partial}{\partial y}\theta(x,y) \approx \left[G_\sigma(x) D_\sigma(y)\right] * \theta(x,y), \qquad (3)$$

where $G_\sigma(x)$ is the one-dimensional Gaussian with zero mean and standard deviation σ, $D_\sigma(y)$ is the derivative of that Gaussian, and $\theta(x,y)$ the polar representation of the intensity gradient which is defined as

$$\theta(x,y) = \arctan\left(\frac{\partial}{\partial y} I(x,y), \frac{\partial}{\partial x} I(x,y)\right) \quad (4)$$

The face detection algorithm is a series of mirrored auto-correlations from the Y-phase operator output, which can also be described as the best cross-correlation between left and right halves, by mirroring one of them, among all possible window positions. According to A. Tankus, Y. Yeshurun and N. Intrator, the Y-phase operator is robust to variations in illumination, scale as well as in-plane and out-plane rotations.

2.5.5. Optical flow

As many methods are derived from and motivated by the function of simple cells in the primary visual cortex of mammals, P. Kruizinga and N. Petkov introduced the so-called *cortical images* [7] to perform the multi-scale matching of two images: Under the proposed algorithm, two input images are used to compute a set of cortical images, from which an image pyramid is constructed. Then the matching of these two sets of cortical image pyramids are determined by an optimal mapping of one image to the other one. The cortical image is computed from the response of a simple visual cortical cell modelled by a function *g*, to an input image *s(x, y)* and at a certain orientation and scale (figure 39).

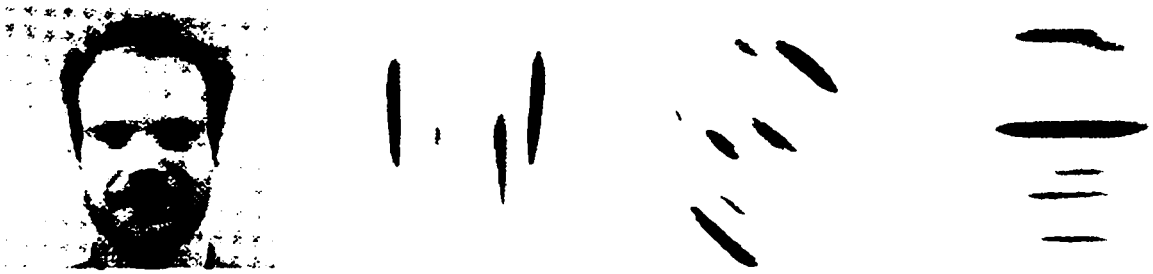


Figure 39: Input face image and three cortical images computed at different orientation. (Source P. Kruizinga and N. Petkov [7])

Optical flow algorithm is often used for motion analysis. Using two or more consecutive frames of an image sequence, it is defined as a 2n-dimensional vector field, which specifies the actual or most likely displacement of image points from frame to frame. An interesting point in this algorithm is that by its definition, the optical flow leads to implementation on parallel machine. The result, when applied on a database of 50 images of 25 persons, reached 92% of recognition rate (see figure 40).

Figure 40: Image A, B and C. Image C is obtained by replacing the finest 8x8 blocks of A by the corresponding best matching blocks of B. (Source P. Kruizinga and N. Petkov [7])

However, current implementation is quite computational time unfriendly as it would last for over 100 hours to find the best match of an input image on a 1000 faces database. Another disadvantage is that P. Kruizinga and N. Petkov noted that this method is quite sensitive to luminance and contrast differences in the images to be matched.

2.5.6. Support vector machines

Most face recognition systems assume that the geometry of the image formation process is frontal. When confronted to additional images from the same person, the pose estimation quickly becomes a problem. By introducing the use of the support vector machines (SVM), J. Huang, D. Li, X. Shao and H. Wechsler [10] hope to address the problem of pose estimation. The support vector machines implementation, in the case of two class pattern recognition, tries to seek separating *hyperplanes* (see figure 41), using

either polynomials of degree 3 or radial basis functions (RBF) as kernel approximation functions.

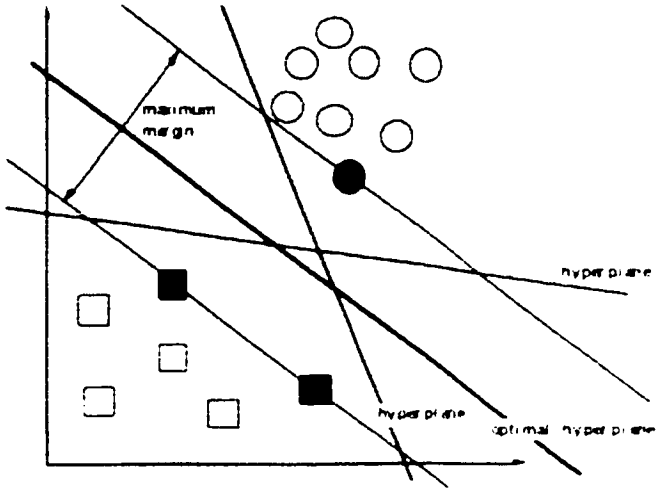


Figure 41: Separation of hyperplanes in a two-dimensional space.
(Source J. Huang et al. [10])

The system starts with the detection of a pattern as a face using a decision tree classifier, using normalized RGB color information. Then, by using as constant the distance between the two eyes, the image covering facial region is cropped and normalized to the size of 32x32 to account for geometry and illumination changes. And finally, it discriminates the face pose using the SVM. From a 600 facial images of 200 subjects from the FERET database, the SVM algorithm appears to provide robust discrimination and classification results, and offers 99.33% to 99.78% using polynomials kernels, and 100% using RBF kernels.

2.5.7. Wavelet transform

The wavelet transform approach is usually often associated with the problem of matching images, especially in the stereo vision matching problem [55]. Because wavelet transform based systems is quite sensitive to data from the high pass frequencies, its performance

suffers when compared to those obtained from principal component analysis approach or neural networks based systems. However, under roughly constant illumination, wavelet transform approach is very attractive due to its computational complexity of O(N). C. Garcia, G. Zikos and G. Tziritas [59] propose a face recognition system combining a wavelet packet analysis technique, for detection of feature vectors, with a probabilistic distance as a classifier.

In the proposed system, the facial features are extracted using simple image processing techniques, such as the integral projection performed on the binary picture obtained with a Laplacian operator. The integral projection, horizontal and vertical, technique was first proposed by Kanade [66], and is defined as

$$\forall\{x,y\} \in A,\ H(y) = \sum_{x=x_1}^{x_2} I(x,y) \text{ and } V(x) = \sum_{y=y_1}^{y_2} I(x,y), \tag{8}$$

where *I(x,y)* is the input image, *H(y)* and *V(x)* are the horizontal and vertical projection over the image area $A = [x_1,y_1] \times [x_2, y_2]$.

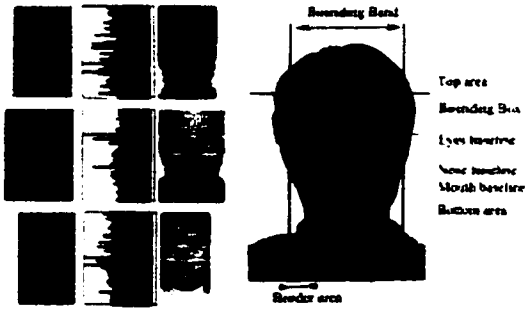


Figure 42: Bounding box and selected facial features baselines.
(Source C. Garcia, G. Zikos and G. Tziritas [59])

The integral projection is used to determine the bounding box of the face and its corresponding facial feature baselines (see figure 42). The wavelet coefficients corresponding to the facial feature baselines, define a set of features to be compared with the *Bhattacharyya* probabilistic distance. The wavelet packet decomposition, as

performed in the proposed approach, is a generalization of the classical wavelet decomposition with an exception that in this case, the details as well as the approximations can be split. This results in a wavelet decomposition tree as in figure 43.

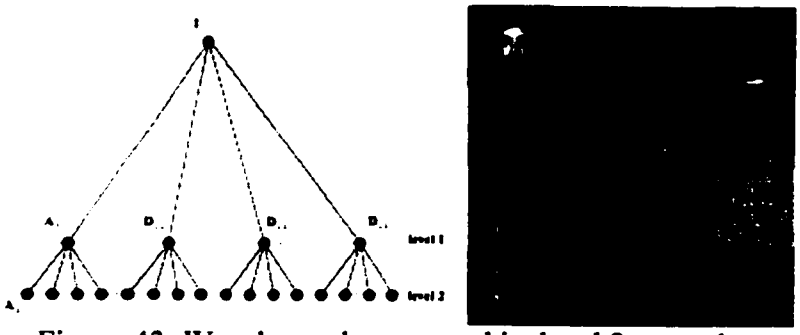


Figure 43: Wavelet packet tree and its level 2 example.
(Source C. Garcia, G. Zikos and G. Tziritas [59])

Because images from the FERET database belonging to the same individual usually present variations in expression and illumination, a subset of 310 images of 155 individuals has been selected for the evaluation of the proposed algorithm's performance, which vary from 97% to 96.12% for different subsets.

2.6. Connectionist approach

2.6.1. Iconic filter banks

As the scale problem is often handled by forming multi-resolution representations of the input images and then performing the same detection procedure at all available resolutions, B. Takács and H. Wechsler [6] propose an approach using the iconic filter banks, which encode multi-resolution face and facial landmarks characteristics. According to B. Takács and H. Wechsler, the face detection and recognition using the connectionist self-organization feature map (SOFM) classifiers on prototypes obtained

with the filter banks may achieve robustness to scale and illumination. On a subset of the FERET database, 415 images of 200 subjects, a performance of 95% to 97% is obtained.

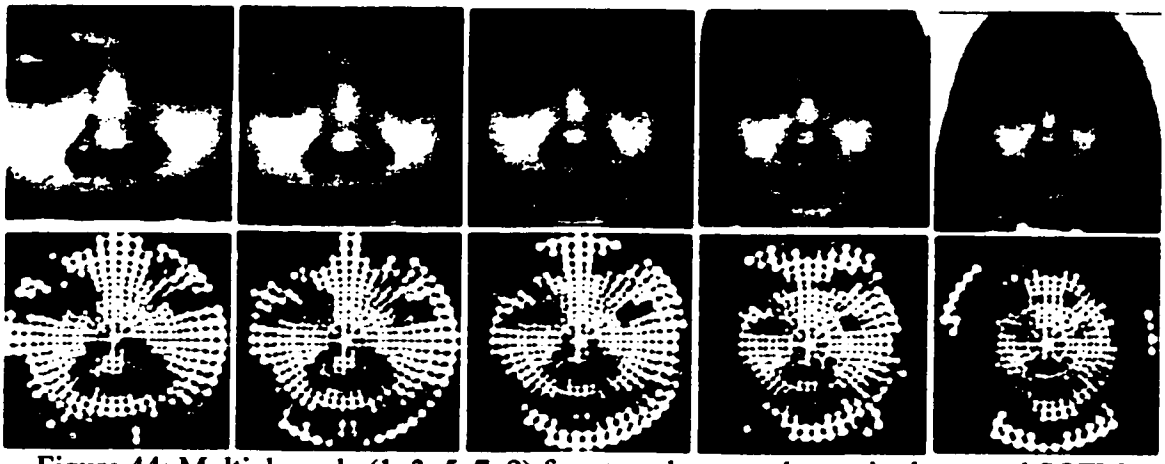


Figure 44: Multiple scale (1, 3, 5, 7, 9) face templates used to train the neural SOFM filters. (Source B. Takács and H. Wechsler [6])

The filter banks concept is closely related to connectionist SOFM: they are *adaptive visual filters* [6] forming non-orthogonal vector basis corresponding to the self-organized facial landmarks, according to the work of B. Takács and H. Wechsler. The iconic representation of feature vectors is obtained from the responses of ensembles of Gaussain spatial filters at a number of orientation and scales, from the filter banks outputs (see figure 44).

The connectionist model used by B. Takács and H. Wechsler is derived from Kohonen's self organizing features maps (SOFM), which basic algorithm can be formulated as [6]:

- Select the size $N=(m,n)$ of the 2D topological map M and initialize weights to small random values.
- For given feature pattern F, find the neuron k with weight vector W_k on the topological map, that minimizes the distance

$$d_k = \frac{\|W_k - F\|}{N}, \qquad (1)$$

- Update weights to neuron k and its neighbors l using

$$\Delta W_l = \mu(t).h_{l,t}.(W_l - F), \quad (2)$$

where $\mu(t)$t and $h_{l,t}$t are adaptation gains and neighbourhood window, respectively, that decrease with time *(t)*.

The activation of a neuron *k* on the topological map corresponds to the template vector which is closest in some distance measure to a given feature pattern. It is then considered as a measure of similarity. However, despite interesting results on facial databases, B. Takács and H. Wechsler specify that the pose problem still exists for future research.

2.6.2. Multi-layered perceptron classifier

Among the most influential works, which address the problem of face detection and recognition, is the application of neural networks. The system proposed by K. K. Sung and T. Poggio [37] detects faces by exhaustively scanning an image for the face-like window patterns at all possible scales (see figure 45). The input image is divided into multiple, possibly overlapping sub-images of the current window size. Then, at each window, the system attempts to classify the enclosed image pattern, a 19x19 pixels window pattern, as being either a face, or not a face.

Figure 45: A canonical face pattern, a 19x19 mask for eliminating near boundary pixels and the resulting canonical face pattern after applying the mask.
(Source K. K. Sung and T. Poggio [37])

Going one step further than just applying the window pattern into the multi-layered perceptron, K. K. Sung and T. Poggio compute a vector of distances from the new

window pattern to the window pattern prototypes in the image window vector space, using a Mahalanobis-like distance metric. The multi-layered perceptron is then trained from these vectors of distance measurements (see figure 46).

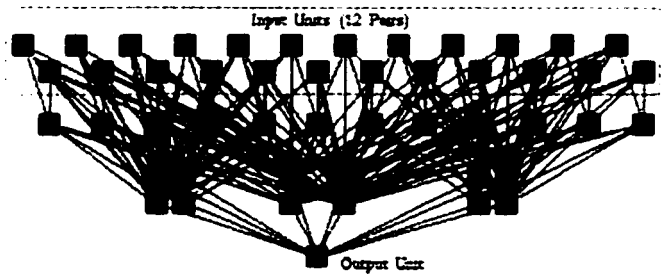


Figure 46: One of the multi-layered perceptron (MLP) net architecture trained to identify face patterns from vector of distance measurements.
(Source K. K. Sung and T. Poggio [37])

The performance of the system reachs 96.3% on a database of 301 frontal and near frontal face mugshots of 71 different people from digitized images taken by a CCD camera in a laboratory environment. With a second database of 23 images with a total of 149 face patterns, with variation in quality ranging from high quality CCD to low quality newspaper scans, the system still reach 79% of detection rate with 5 false positives. However, because the current system is based on near frontal views, the authors acknowledge that different poses will be of future works.

2.7. Evolutionary computation

The eyes, among the facial landmarks used for face detection and recognition, play an important role in face normalization. Because their position and the inter-ocular distance are relatively constant for most people, the eyes detection provides a good framework for image normalization and then facilitates the localization of other facial landmarks. According to J. Bala, K. DeJong, J. Huang, H. Vafaie and H. Wechsler [23], crafting of visual routines is of optimization problems and should lead to non deterministic methods,

such as evolutionary computation (EC) in general and genetic algorithms (GA) in particular.

The process of natural selection leads to evolution as a result of adaptive strategies, which are continuously tested for their *fitness* in a closed loop control. By analogy, the evolutionary computation attempts then to emulate computationally the survival of the fittest in the complex and difficult problems. Using similar principles, genetic algorithms, initially introduced by Holland as adaptive search techniques, maintain typically a constant-sized population of individuals, also called *chromosomes*, which represent samples from the space to be searched. Each individual is then evaluated on the basis of its overall fitness with respect to some pre-specified constraints. New individuals are produced by selecting high performing individuals while retaining many of the features of their parents. Two main genetic operators, crossover and mutation, are used to form the new generation of individuals. The crossover operates by randomly selecting a point in the two selected parents gene structures and exchanging the remaining segments of the parents to create new individuals. The mutation operates by randomly changing one or more components of a selected individual and it acts as a perturbation operator to bring new information, also called genetic material, into the populations. The mutation operator prevents any stagnation, or premature convergence, that might occur during the search process.

The several steps in the face recognition algorithm proposed by J. Bala, K. DeJong, J. Huang, H. Vafaie and H. Wechsler [24], include a dimensionality reduction by principal component analysis (PCA) module, an eyes detection module by genetic algorithms and finally a face recognition module by optimal projection axes (OPA).

The feature list consists of one hundred and forty seven features { x_1, x_2, ... x_{147} } measured over 6x4 windows of two pixels overlap (see figure 47).

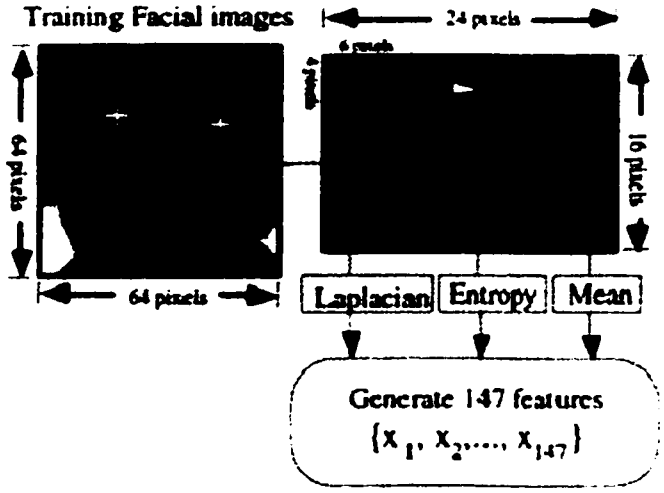


Figure 47: Features extraction where x_1 to x_{49} contain the mean for each window, x_{50} to x_{98} the entropies for each window and x_{99} to x_{149} the means for each window after applying the Laplacian over 24x16 frames.
(Source J. Bala, K. DeJong, J. Huang, H. Vafaie and H. Wechsler [24])

The optimal projection axes method works by searching through all the rotations defined over whitened principal component analysis subspaces. The whitening transformation is an operator defined by J. Bala, K. DeJong, J. Huang, H. Vafaie and H. Wechsler to counteracts the fact that the mean square error (MSE), a principle underlying PCA, preferentially weights low frequencies [23]. As the search space may be too large for systematic search, it is driven by genetic algorithms. Experiments on a 30 dimensional PCA space, from a subset of the FERET database, yield a result of 97.02% recognition rate.

2.8. Commercial implementations

To some people, face recognition is still a component of science fiction movies. This way of thinking is partially due to the fact that although various techniques do exist, they are

still purely academic research and barely make their way out of technical reports from university research laboratories. As the needs for more security increased recently, people start to explore into biometric techniques, with proven efficacy such as fingerprint recognition and iris recognition, and implementations in face recognition are becoming more available and financially more accessible to the general public (see table 1).

Vendor	Product Availability	Technology	Current Implementations
BioID BiometricStore	BioID Client/Server	Eigenface / neural network: images compiled into single reference face	• limited IT security rollouts in SE Asia • software bundled with PC cameras
Biometrica	Casino Information Network Casino Information Database Visual Casino App. Suite	Eigenface	• 1:many gaming applications (Foxwoods, Trump, Stratosphere)
eTrue (formerly Miros) BiometricStore	TrueFace for Web, Network, PC, NT, Database	Neural network	• distributed in facial-recognition ATM's • airport baggage identification • component of NT-based software suites
Viisage	Viisage Gallery (including C++ DLL)	Eigenface	• 1:many driver's license applications • 1:many gaming surveillance applications
Visionics BiometricStore	FaceIt DB FaceIt NT, C++ SDK Identification and Verification SDK	Local feature analysis	• 1:many surveillance applications • 1:many driver's license applications • component of NT-based software suites

Imagis	ID 2000	Automatic face processing	• 1:many criminal justice applications • limited gaming implementations
AcSys BiometricStore	HNeT Facial Recognition System	Holographic/Quantum Neural Technology	none/limited
Keyware BiometricStore	FaceGuardian	Local feature analysis	limited
ZN Vision Technologies AG	Phantomas, ZN-Face	Neural network	• card-based 1:1 physical security • 1:many 'mugshot' applications
Berninger Software	Visec-FIRE	Automatic face processing	none/limited
Digitech	digitech::face SDK	"Computer vision technology", no further details	none/limited
IVS (Intelligent Verification Systems)	FaceKey	Unknown	limited
Neurodynamics	Nvisage	Neural network	limited
Cognitec/ Plettac Electronics	FaceVACS	Feature analysis	limited safe-deposit applications
SSK-Virtual Image	Imager	"Face vectors", no further details	none/limited
VisionSphere	UnMask	"Holistic Feature Coding"	limited driver's license applications

Table 1: Facial scan vendors.
(Source Facial Scan http://facial-scan.com online home page)

Techniques and algorithms used in the Current-Off-The-Shelf (COTS) face recognition products use private and confidential information. The COTS products usually fall into three categories of implementations, authentication for network logon in information

technology (IT) uses, authentication at access checkpoint and identification of people in a crowd.

Chapter 3

Implementations of wavelet transforms

3.1. Introduction

A wavelet transform is defined as a signal decomposition approach to overcome the shortcomings of the window Fourier transform. The wavelet transform represents a function as a superposition of a family of basis functions called wavelets. A set of basis functions can be generated by translating and dilating the mother wavelet corresponding to a particular basis. While this definition represents the discrete wavelet transform with the Daubechies coefficients, the Gabor wavelet transform approach, as in the research literature, does not observe all requirements to be called wavelet.

In order to conduct experiments on the techniques described in the following sections, we implement a series of modules with each module corresponding to a technique. We will begin to discuss, in the following sections, about the theoretical aspects of image processing and feature detection used in this work, using Fourier transform, Gabor wavelet transform and discrete wavelet transform. Then, we will propose and describe workaround solutions to all problems encountered during the implementation of the techniques used in the experiments, such as the Numerical Recipes problem for computing a discrete wavelet transform.

3.2. Basic image preprocessing

3.2.1. Color – grayscale transformation

While the color information may provide certain type of features to the face detection and recognition systems, most of these are associated grayscale images, or on color images reduced to grayscale, for better storage and to be more computationally friendly.

A color image may be viewed as a two dimensional array, usually represented with the RGB format for computer image. There are several ways, or standards, of transforming a natural color image to a grayscale image. One of these is the standard proposed for NTSC and PAL, in the video field, and is defined for each pixel in the image as

$$\text{Gray} = 0.299 * \text{Red} + 0.587 * \text{Green} + 0.114 * \text{Blue} \quad (9)$$

3.2.2. Histogram equalization

Histogram modeling techniques, commonly referred as histogram equalization, provide a sophisticated method for modifying the dynamic range and contrast of an image by altering that image such that its intensity histogram has a desired shape. Unlike contrast stretching, histogram modeling operators may employ non-linear and non-monotonic transfer functions to map between pixel intensity values in the input and output images. Histogram equalization employs a monotonic, non-linear mapping which re-assigns the intensity values of pixels in the input image such that the output image contains a uniform distribution of intensities, or in other words, a flat histogram. Because it is effective in detail enhancement, it may improve image processing operators. This technique is often used in the correction of non-linear effects introduced by digitizers.

3.3. Fast Fourier Transform

The Fourier transform translates a function in the time domain into a function in the frequency domain. It breaks the function into a series of sine waves of different frequencies, and its basis functions are sines and cosines.

3.3.1. Convolution

By the definition of the filter approach, the result from a convolution of a filter and an image is used to match one face image to the other. Suppose more generally that f and h are two arbitrary finite extent images of dimensions M x N and P x Q, respectively, the usual time integral convolution, in its linear and discrete form in a spatial domain, is defined as

$$\begin{aligned} g(m,n) &= f(m,n) * h(m,n) \\ &= \sum_{p=0}^{M-1} \sum_{q=0}^{N-1} f(p,q) h(m-p, n-q) \end{aligned} \tag{20}$$

In order to save processing cycles, especially in the case of convolution of an image with several Gabor wavelet transform at different scales and different orientations, we use a very interesting property of the fast Fourier transform (FFT), which is performing the convolution in the frequency domain. The linear convolution, by discrete Fourier transform, is then defined as

$$\hat{g} = \hat{f} \otimes \hat{h} \tag{21}$$

In equation (21), the fast Fourier transform is applied first into f, and h, then a pointwise complex multiplication will produce $\hat{g}$, which is going through the inverse Fourier transform to finally give the convoluted result g.

In the general case, the images f and h do not have necessarily the same dimensions, since in most applications an image is convolved with a filter function of different, and usually much smaller, extent. Because the function $\hat{g}$ is a periodic function, the periodic replicas will overlap if horizontal or vertical periods are too small. To cancel the wraparound error, also called the spatial aliasing, the functions f and h are modified by

increasing their size by zero padding them. The correct period lengths are equal to the lengths of the correct linear convolution result, which for two arbitrary M x N and P x Q image functions will be (M + P – 1) x (N + Q – 1). And as required by certain FFT implementation, such as the one from Numerical Recipes, the period length is padded to the closest upper power of 2. The FFT offers a computational complexity of order not exceeding $MN\log_2(MN)$, which represents a considerable speedup.

3.3.2. Normalized cross-correlation

The correlation between two signals is a standard approach to feature detection. As one of the simplest face detection, and may be face recognition, it identifies a pattern within an image using cross correlation of the image with a suitable mask (see figure 48).

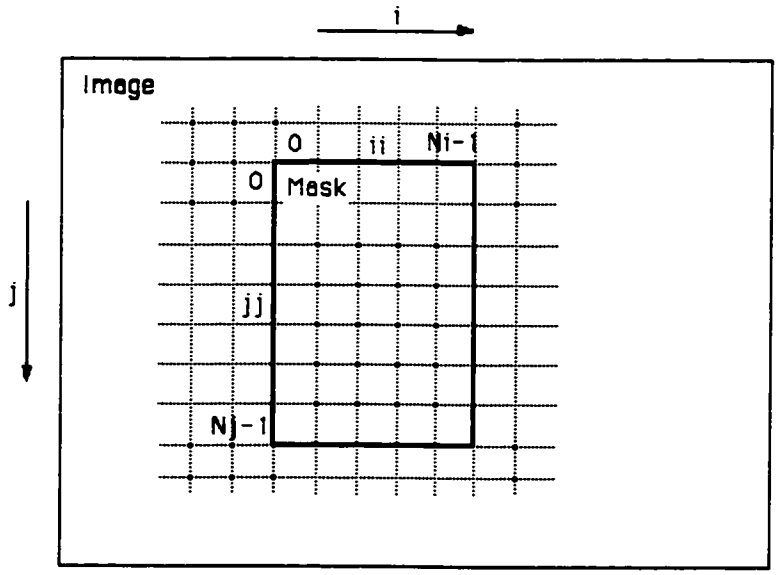


Figure 48: Search for the pattern in the image with the mask.
(Source Cross Correlation by P. Bourke online
http://www.swin.edu.au/astronomy/pbourke/analysis/correlate)

The cross correlation gives a very high response when the mask and the pattern being sought are similar. The mask is itself an image which needs to have the same functional

appearance as the pattern to be found. The peaks in this cross correlation "surface" are the positions of the best matches in the image of the mask. The cross correlation in our experiments (see figure 49), as in the case for the convolution by FFT, is performed in a more computationally efficient way by using the property of the FFT.

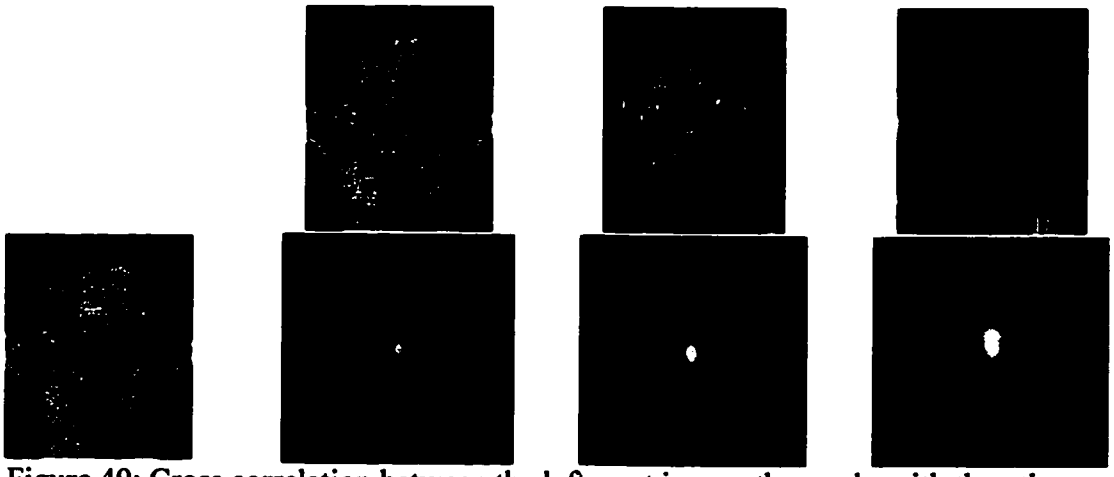


Figure 49: Cross correlation between the left most image, the mask, with the others images. The brightest spot is where the mask corresponds most with the pattern, which is this case, is equal to the whole image.

However, the template matching using cross correlation, even normalized cross correlation, suffers from several disadvantages. If the image energy varies with position, matching will fail. For example, the correlation between the feature and an exactly matching region in the image may be less than the correlation between the feature and a bright spot. The lighting condition across the image sequence as well as the pose problem can be serious disadvantages in using this technique.

3.3.3. Phase correlation

By determining the location of the peak of the inverse Fourier transform of the crosspower spectrum phase, the FFT technique determines the displacement of one image with respect to the other. Since the phase difference for every frequency contributes

equally, the location of the peak will not change if there is noise, which is limited to a narrow bandwidth, in other words, a small range of frequencies. Rotational movement can be deduced in a similar manner as translation using phase correlation, by representing the rotation as a translation displacement of polar coordinates. In fact, rotation is invariant to the Fourier transform. Rotating an image rotates the Fourier transform of that image by the same angle. Phase correlation is a frequency-domain motion measurement method that makes use of the shift property of the Fourier transform to determine translation, scale and orientation changes (see figure 50).

Figure 50: Retrieving the scale, translation and orientation angle for the same image and its transformed (scale 1.3, translation (5, 10) and rotation 30 degrees).

However, the phase correlation is very sensitive to the image structure. The phase correlation on images with the same person under a small change of pose would be unable to give correct result (see figure 51).

Figure 51: Incorrect result with different poses.

3.4. Gabor wavelet transform

3.4.1. Definition

Gabor filter is a simple model, of similar shape, for the responses of simple cells in the primary visual cortex [72]. The Gabor filter, also known as a short time Fourier transform (see figure 52), has been introduced to cover the shape problem of the data window.

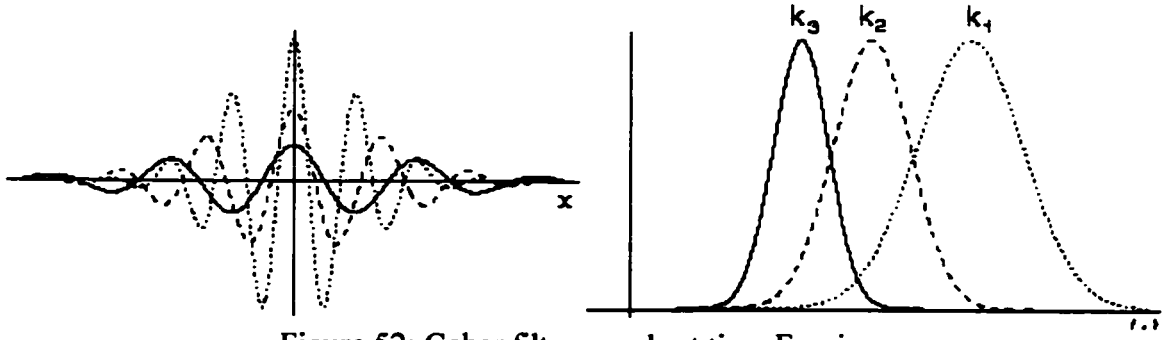


Figure 52: Gabor filter as a short time Fourier.
(Source M. Pötzsch and M. Rinne [72])

In most proposed face recognition systems using the Gabor wavelet transform, the faces are represented as the convolution results of a face image with a bank of multi-scale and multi-orientation kernels at different locations on the image. At each of these locations, the images are convolved with 40 Gabor filters (8 orientations and 5 scales). The convolution of the image $I(\vec{x})$ with a bank of Gabor filters is expressed as

$$(WI)(\vec{k},\vec{x_0}) = \int \psi_{\vec{k}}(\vec{x_0}-\vec{x}) I(\vec{x}) d^2x = \psi_{\vec{k}} I \tag{16}$$

The filters form a self-similar family of complex Gabor kernels

$$\psi_j(\vec{x}) = \frac{k_j^2}{\sigma^2}\exp\left(-\frac{k_j^2 x^2}{\sigma^2}\right)\left[\exp(i\vec{k}_j\vec{x}) - \exp\left(-\frac{\sigma^2}{2}\right)\right] \tag{17}$$

Because all kernels are similar in the sense that they can be generated from one kernel by dilatation and rotation, it is known as wavelet transform [78]. *Gabor wavelets* are

localized in both space and frequency domain and have the shape of plane waves of a wave vector $\bar{k}$, restricted by a Gaussian envelope function of width σ (see figure 53). In addition, the last part $\exp\left(-\frac{\sigma^2}{2}\right)$ of the equation (17) makes the kernels DC free, such that their Fourier transform vanishes at zero. Each kernel responds best at the frequency given by the characteristic wave vector

$$\bar{k}(\nu,\mu)=\frac{\pi}{2}2^{-\frac{\nu}{2}}\left(\cos\left(\frac{\pi}{8}\mu\right),\sin\left(\frac{\pi}{8}\mu\right)\right), \tag{18}$$

where scale $\nu \in \{0,...,4\}$; orientation $\mu \in \{0,...,7\}$

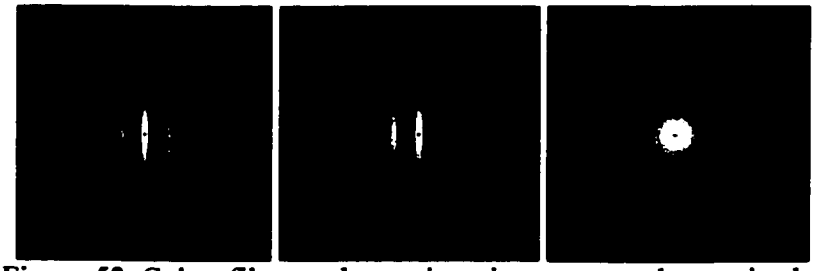


Figure 53: Gabor filter real part, imaginary part and magnitude.

Since the wavelets have a limited localization in space and frequency, they are robust to shift, scaling and rotation.

3.4.2. Set of features

As a grid of 5 x 7 is laid over the image, the set of features is a 35 (5 x 7) feature vectors set, one feature vector computed at each node of the grid. The feature vector, also called Jet [78], is a set $\{J_j\}$ of 40 complex Gabor wavelet coefficients obtained for one image point. It can be written as

$$J_j = a_j \exp(i\phi_j), \tag{19}$$

with magnitudes a, which vary slowly with position.

The matching process uses the grid, composed of a set of feature vectors from one image to another with the similarity distance.

3.4.3. Similarity distance

A Gabor wavelet transform responds strongly to edges of the direction if perpendicular to its wave vector [72]. But when hitting an edge, the real and the imaginary parts oscillate with the characteristic frequency instead of providing a smooth peak. Thus, the rapid phase variations cause problems as feature vectors jets taken from an image few pixels apart from each other have very different coefficients, although they represent almost the same local feature.

The problem can be solved by ignoring completely the phase, as proposed by Lades et al [72], or approximately compensated for its variations explicitly, as proposed by Wiskott et al [75]. When the phase is ignored, the magnitude varies slowly with position, and S_m, the similarity with magnitude, is defined as

$$S_m\left(\vec{J},\vec{J'}\right)=\frac{\sum_i a_i a_i'}{\sqrt{\sum_i a_i^2 \sum_i a_i'^2}} \tag{14}$$

The displacement $\vec{d}$ may be estimated by maximizing S_ϕ, the similarity in phase, in its Taylor expansion [75] around $\vec{d}=0$.

$$S_\phi\left(\vec{J},\vec{J'}\right)=\frac{\sum_i a_i a_i' \cos\left(\phi_i-\phi_i'-\vec{d}\vec{k}_i\right)}{\sqrt{\sum_i a_i^2 \sum_i a_i'^2}} \tag{15}$$

3.4.4. Graph composition

The Gabor wavelet transform approach is usually associated with graph matching, labelled graphs [76] or bunch graphs [73]. A labelled graph representing a face image consists of N nodes connected by E edges. The nodes are located at facial landmarks in the case of bunch graph. In the simplest form of a labelled graph, the nodes are simply located on a uniform grid, then they are moved individually to form a deformed grid where feature vectors respond best (see figure 54).

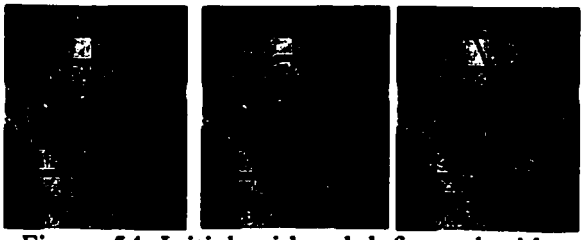


Figure 54: Initial grid and deformed grids.

3.5. Daubechies wavelet transform

3.5.1. Definition

Recently, the discrete wavelet transform (DWT) has become popular in image coding applications, especially in the field of image compression or indexing. Like the fast Fourier transform (FFT), the discret wavelet transform (DWT) is also a fast, linear operation that operates on a data vector whose length is an integer power of two, transforming it into a numerically different vector of the same length. Also like the FFT, the wavelet transform is *invertible* and in fact orthogonal. In fact, the inverse transform, when viewed as a big matrix, is simply the transpose of the transform.

However, though the sines and cosines functions define a unique set of basis functions of the Fourier transform, there is not one single unique set of wavelets. A particular set of wavelets is specified by a particular set of numbers, called wavelet filter coefficients. In our work, we will largely restrict ourselves to wavelet filters in a class discovered by Daubechies [68]. This class includes members ranging from highly localized to very smooth. As an example, the simplest and most localized member of this class, often referred as DAUB4 with a compact support of 4 and which is orthogonal, has only four coefficients, $c_0 \ldots c_3$.

The advantage of the orthogonal wavelet transform is that all of the information of the original data set is retained and can be used to reconstruct the original data set. And as it has compact support, its value is zero outside a bounded interval.

The discrete wavelet transform is defined of applying a wavelet coefficient matrix-like, hierarchically, to the data vector. The DWT decomposes an image into a pyramid structure of subimages with various resolutions corresponding to the different scales. The inverse wavelet transform is calculated in the reverse manner, by starting from the lowest resolution subimages, the higher resolution images are calculated recursively. Each stage of the decomposition will create one low pass subimage and three highpass directional subimages. These highpass directional subimages provide the information about the changes in the horizontal, vertical and diagonal directions, respectively. The low pass subimage is a low resolution version of the original image.

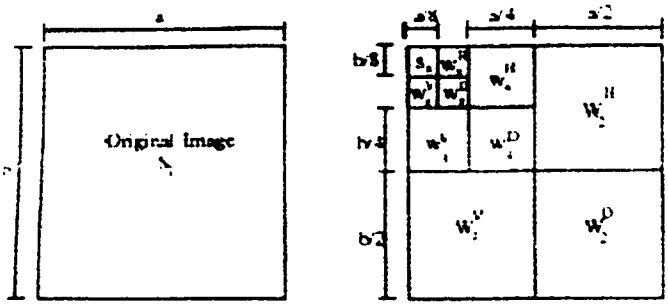


Figure 55: Wavelet transformed image.
(Source M. K. Mandal and T. Aboulnasr [69])

Figure 55 shows a three-level wavelet deomposition of an image S_1 of size *a* x *b* pixels. In the first level of decomposition, one low pass subimage S2 and three orientation selective high pass subimages W2H, W2V, W2D are created. The process is repeated on the low pass subimage to form higher level wavelet decomposition, recursively.

The low pass subimage is very important. In the case of the frequency-based face recognition method, called spectroface, the Fourier transform is applied to the low frequency subband of the decomposed, which is an optimal approximate image of the mother image in a lower dimension.

3.5.2. Set of features

Given an input image, we apply the wavelet decomposition using the coefficients from the Daubechies wavelet transform. Unlike the spectroface, where the low pass subimage is used as input to the Fourier transform into the frequency domain, the low pass subimage used in this work is only used for normalization.

As in figure 51, a three level wavelet decomposition is applied to the input image using the numerical recipes (NR) implementation. The resulting pyramidal decomposition is then modified backward to correct the NR effect, which will be described in the

following chapter. A series of 5 x 7 grids are laid on each subimage but the low pass image, which represents the low resolution version of the original image (see figure 56).

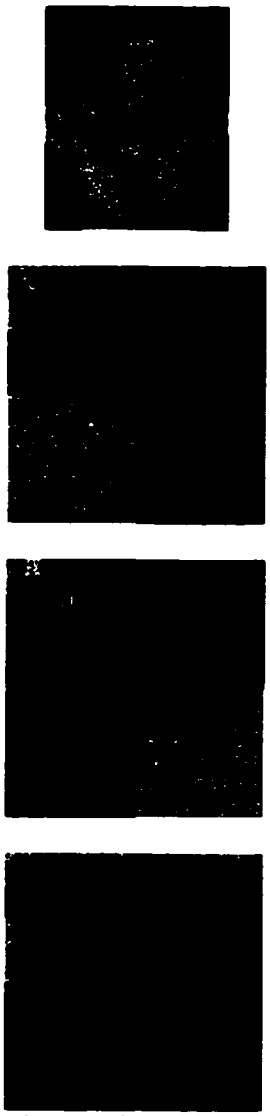


Figure 56: Example of face image, its standard and non-standard wavelet decompositions, and the reference grid over the non-standard wavelet decomposition.

The feature points are located on the nine grids, under the nodes of each grid. Each node gives a numerical value V which represents the response of the discrete wavelet transform of the image at a certain scale and position. In order for the image matching to be invariant to local image intensity, the value V is then normalized with its

corresponding value from the low pass subimage at this scale. The feature vector is then composed with the previously normalized computed 9 x 5 x 7 = 315 values.

3.5.3. Similarity distance

Under the previously defined multiresolution wavelet analysis scheme, the single point to point similarity distance may simply be an Euclidian distance. If we consider single point to point match at the j-th level, the similarity distance can be defined using only 3 differential components ($D_{j,p}(x,y)$, $p = 1, 2, 3$). The feature vector component $B_j(x,y)$ at each position (x, y) is defined as

$$B_j(x,y) = \left(B_{j,1}(x,y)\ \ B_{j,2}(x,y)\ \ B_{j,3}(x,y)\right), \tag{10}$$

with

$$B_{j,p}(x,y) = \frac{D_{j,p}(x,y)}{\left|A_j(x,y)\right|},\ \ p = 1,2,3, \tag{11}$$

where |.| denotes the magnitude.

In fact, the normalized difference subband can be defined as

$$SB_j\left((x,y),(x',y')\right) = \left|B_j(x,y) - B_j'(x',y')\right|, \tag{12}$$

where $SB_j((x,y),(x',y'))$ is the similarity distance for feature vector components B.

3.5.4. Graph composition

In the case of the Gabor wavelet transform based graph matching, the single graph has, at each one of its nodes, a feature vector composed of several values coming from several orientations and scales. In the case of the discrete wavelet transform with Daubechies coefficients, the same type of graph can be deducted. A grid, simple form of graph, has

several values of each of its nodes. These values come from the three differential components at each level of decomposition.

3.5.5. Modification of Numerical Recipes implementation

In the current work, we are using the implementation of the discrete wavelet transformation from the famous Numerical Recipes (NR) [75]. Under the NR definition, a standard wavelet decomposition of a d-dimensional array is most easily obtained by transforming the array sequentially on its first index, for all values of its other indices, then on its second index, and so on. Each transformation corresponds to multiplication by an orthogonal matrix. This leads to the situation found on figure 57, which is not really suitable for composing the multi level grids. As

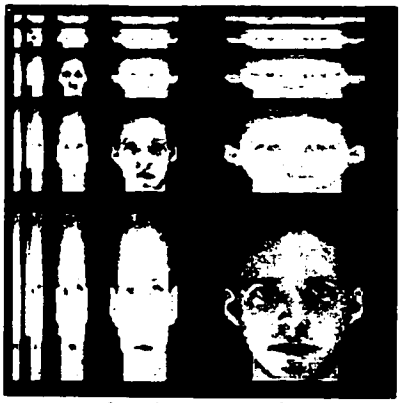


Figure 57: A standard discrete wavelet decomposition with DAUB1 coefficients in the NR implementation.
(Source Numerical Recipes in C [75])

In order to lay grids on each of the subimages, we have had to modify the standard wavelet decomposition result by computing the reverse path, taken from the lowest level (see figures 58 and 59). The resulting two-dimensional array contains the low pass subimage, and three differential components at each j-th level.

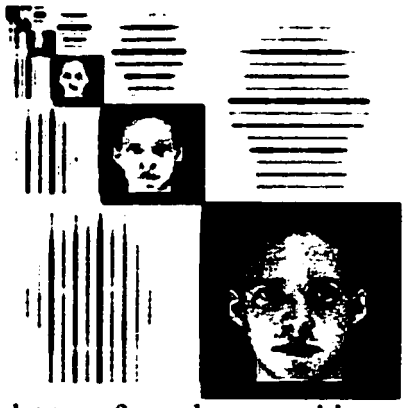


Figure 58: Modified wavelet transform decomposition, with DAUB1 coefficients.

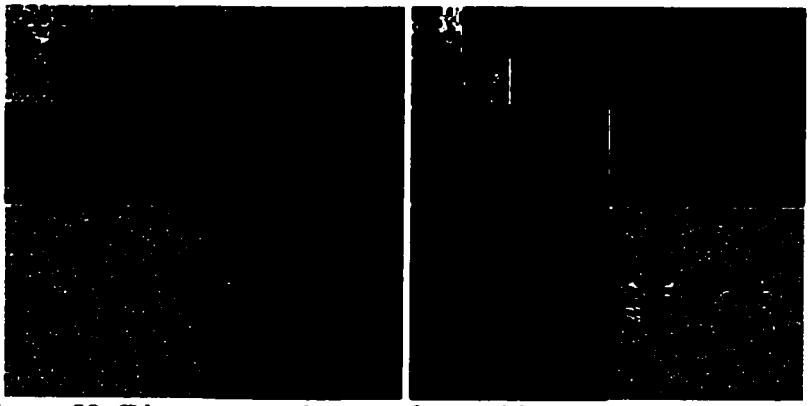


Figure 59: Discrete wavelet transform with DAUB12 coefficients.

Chapter 4

Implementations of matching techniques

4.1. Introduction

Because each face is represented by a set of feature vectors positioned on the nodes of a coarse and rectangular grid placed on the image, the comparison of two face images is accomplished by matching and adapting a grid taken from one image to the features of the other image.

As in the case for experiments on the wavelet transforms, we perform the experiments on graph matching by implementing the different techniques of graph matching. These techniques include the case of rigid graph matching and deformable graph matching. Again, we will propose improvements, such as the border displacement coefficient, as well as workaround solutions to all the problems encountered.

4.2. Rigid graph matching

General attributed graphs, grid in a special case, describe objects on sparse locations by attaching to each node a feature vector that contains information on the local neighborhood of the node location. Matching with a rigid graph would be moving this rigid graph over a search area in the image, to a position where a similarity function is at its highest value (see figure 60).

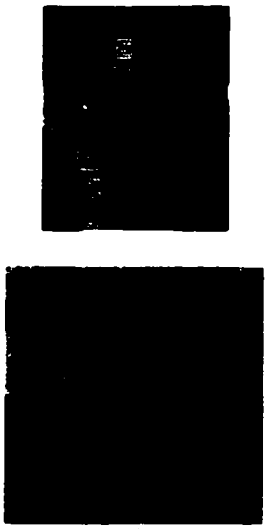


Figure 60: Rigid grids for matching with Gabor wavelet transform (top) and Daubechies wavelet transform (bottom).

4.3. Deformable graph matching

Deformable graph matching is also sometimes called elastic graph matching (EGM) [73]. It consists in locating an attributed graph on the image that is as close as possible to the reference graph. The distance between two graphs is evaluated by a dissimilarity function, that considers both the feature vectors of each node and the deformation information attached to the edges [76]. The dissimilarity measures, where the contribution from nodes and edges are independent, are defined as

$$d(G,R)=\sum_{i=1}^{N_n} d_n\left(G_{ni},R_{ni}\right)+\lambda\sum_{j=1}^{N_e} d_e\left(G_{ej},R_{ej}\right), \qquad (13)$$

where G_{ni} represents the *i*th node of grid G, R_{ej} is the *j*th node of grid R, and N_n N_e are the number of nodes and edges, and λ is the weighting factor which characterizes the *stiffness* of the graph. A plastic graph which opposes no reaction to deformation corresponds to $\lambda = 0$, while a totally rigid graph has very large λ.

The deformation is achieved by displacing each node around its current location, and by placing it at the location where the minimum value of d(G, R) is obtained. This operation is applied on each node successively, and the whole process is repeated until no further decrease is obtained (see figure 61).

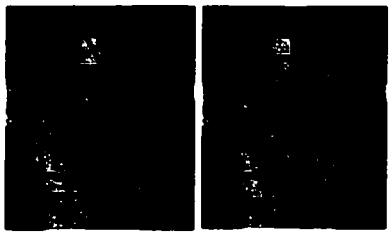


Figure 61: Deformed grid for matching with Gabor wavelet transform with different displacement coefficient values, relaxed and enforced.

However, in order to lower the effect of the face border background, the graph should be more reticent in being moved near the edges of the image. We introduce a parameter called border displacement coefficient (BDC), dependent of the distance to the closest border, which restraints the nodes near the border to move toward it. The BDC is only applied to the nodes located on the external layer of the grids.

Also, in order to prevent each node of the graph to move freely to a too large distance, a general displacement coefficient (GDC) is used, thus allows the graph to be deformed to a certain extent only.

Chapter 5

Proposed Face Recognition system

5.1. Description

Based on the knowledge acquired from the experiments from the last two chapters, we propose a face recognition system, based on the Daubechies wavelet transform for computing features and rigid graph matching for comparison. For each face image in the ORL database, we compute its corresponding list of feature points and values described in section 3.5.2, or by using the following algorithm given in pseudo code:

- Center face image in a square image, pad if necessary (Width = Height) giving I
- Apply a three (3) scales Daubechies wavelet transform on the image I to obtain a series of decomposition.
- For each of the three scales, such as in figure 60, lay a centered proportional grid on the three high pass subimages.
- For each node of the grids, normalize its value with the low pass subimage at the corresponding scale.
- Get the nodes of all the grids to form a rigid graph.

Our approach of measuring the difference between two face images is a little bit different.

- Each grid is compared to the equivalent grid on the other image, using the Euclidian distance at each node.
- The values obtained, for each grid, form a vector.
- The norm of the previous vector is computed and represents each grid.
- At each of the three scales, we have a norm of a vector, composed of three values with the norm computed at the previous step for each grid.

- And finally, the norm of the final three components vector (for the three scales) determines the difference between two face images.

5.1.1. Training process

During the training process, for each class of person, every image is matched against others of the same class. The average on the difference as well as the standard deviation are computed and the image with the smallest average of the difference is the winner, representing the class.

	0	8.956526	1.307652	1.903013	1.350065	1.333296	4.433446	1.572674	1.550774	1.365042	2.377249	2.561141
	8.956526	0	8.591523	9.125095	8.597456	8.667041	11.302814	8.879164	8.795525	8.644679	8.155982	2.978055
	1.307652	8.591523	0	1.647116	0.758271	0.753382	3.940478	1.156257	1.421057	0.703581	2.027932	2.530556
	1.903013	9.125095	1.647116	0	1.375344	1.748041	4.747345	1.723898	2.129271	1.681302	2.608042	2.569043
	1.350065	8.597456	0.758271	1.375344	0	0.849607	4.09081	1.079718	1.526663	0.774125	2.040206	2.541509
	1.333296	8.667041	0.753382	1.748041	0.849607	0	3.957883	1.28748	1.536508	0.774651	2.090789	2.534424
	4.433446	11.302814	3.940478	4.747345	4.09081	3.957883	0	3.941922	4.403935	3.944453	4.476308	2.748576
	1.572674	8.879164	1.156257	1.723898	1.079718	1.28748	3.941922	0	1.820177	1.179496	2.264079	2.526031
	1.550774	8.795525	1.421057	2.129271	1.526663	1.536508	4.403935	1.820177	0	1.501462	2.468537	2.47288
	1.365042	8.644679	0.703581	1.681302	0.774125	0.774651	3.944453	1.179496	1.501462	0	2.05688	2.538848
smallest average=2.027932 index=3/10 2.dwt												

Table 2 : Compute average of matching coefficients and its standard deviation for the class S1. In this example, the image "2" is the winner because of its smallest average.

5.1.2. Recognition of a face

When a face image is presented to the face recognition system, a series of grids are laid over its decomposition by the Daubechies wavelet transform (coefficient 12). The nodes of these grids, at every scale of the decomposition, form a template to be compared to all internal templates. In the case of the ORL database, every attempt of face recognition will match the incoming face image to the 40 internal templates. As the normalized difference is defined as the difference between the incoming and the internal template, divided by the internal template average value, one class is preferred when the normalized difference

between its template and the one from the incoming image is the smallest. The pseudo code to choose one class instead of another one is defined as

```
valMatchOne = ComputeNormalizedDifference(pCandidate, pTemplateOne,
        pGrid)
diffToOne = | valMatchOne – pTemplateOne.fAverage |
valMatchOther = ComputeNormalizedDifference (pCandidate, pTemplateOther,
        pGrid)
diffToOther = | valMatchOther – pTemplateOther.fAverage |
if      (valMatchOther / pTemplateOther.fAverage) < (valMatchOne /
        pTemplateOne. fAverage) and
        (diffToOther / pTemplateOther.fAverage) < (diffToOne / pTemplateOne.
        fAverage) and
        (diffToOther < pOther->fStdev)
then
        select pTemplateOther as the class of the candidate
endif
```

5.2. Data structures

Because the computational time for Gabor wavelet transform is quite long, even with the speedup through FFT convolution, the data is computed in advance, before the matching process, and then saved into disk under a file with extension .gwt for Gabor and .dwt for Daubechies. For example, when the result of a convolution of a Gabor wavelet transform

is to be saved to a disk, the matrices containing floating point data is saved one by one, for each frequency and each orientation, in its binary form.

When the matching process is invoked, a new test image is computed for the case of discrete wavelet transform with Daubechies coefficients, and convolved with the filters for the case of Gabor wavelet transform. The matching process then makes use of this computed data with the corresponding loaded from disk for the other parts.

5.3. Database to class partitioning

In this work, we do take another approach by selecting one image per class to represent this particular class. This particular image will have its transformed data closest to the mean of differences, when compared to the other images in the same class.

For example, in the case of a discrete wavelet transform, all images from a class are decomposed with the discrete wavelet transform to form grids with feature vectors. Each image is then compared to the other ones, always from the same class, to give out a dissimilarity measure. From these dissimilarity measures, a mean is computed, along with its variance. The image, which contains the closest dissimilarity measure to this mean value will be chosen as the reference for this class. The variance value determines the threshold value for this particular class.

When a test image is presented to the system for identification, it is decomposed, under the discrete wavelet transform scheme, or convolved with the Gabor wavelet transform filters under the Gabor wavelet transform scheme. The resulting transformed data is then compared to the corresponding data from the reference image in each class. The smallest

dissimilarity measure, under the threshold for a class, will determine whether this test image belongs to this class.

5.4. Implementation

The implementation does not only contain our proposed face recognition system, but also several modules used in a comparison of different techniques. The programs developed are running on a personal computer in the Microsoft Windows 2000 environment, so they can take advantage of the graphical interface offered by this environment and the integrated development environment (IDE) of type rapid application development (RAD) for this platform. The programs implemented in this work were developed using Borland C++ Builder 5 for the Microsoft Windows 2000 environment.

5.4.1. Graphical user interface

A face recognition system can run completely in a console type application, asking for an input image and giving out its identity, 0 or 1 as an answer. However, the applications developed in a Windows may provide a fast visual feedback (see figure 62), especially when working with filters and filtered images.

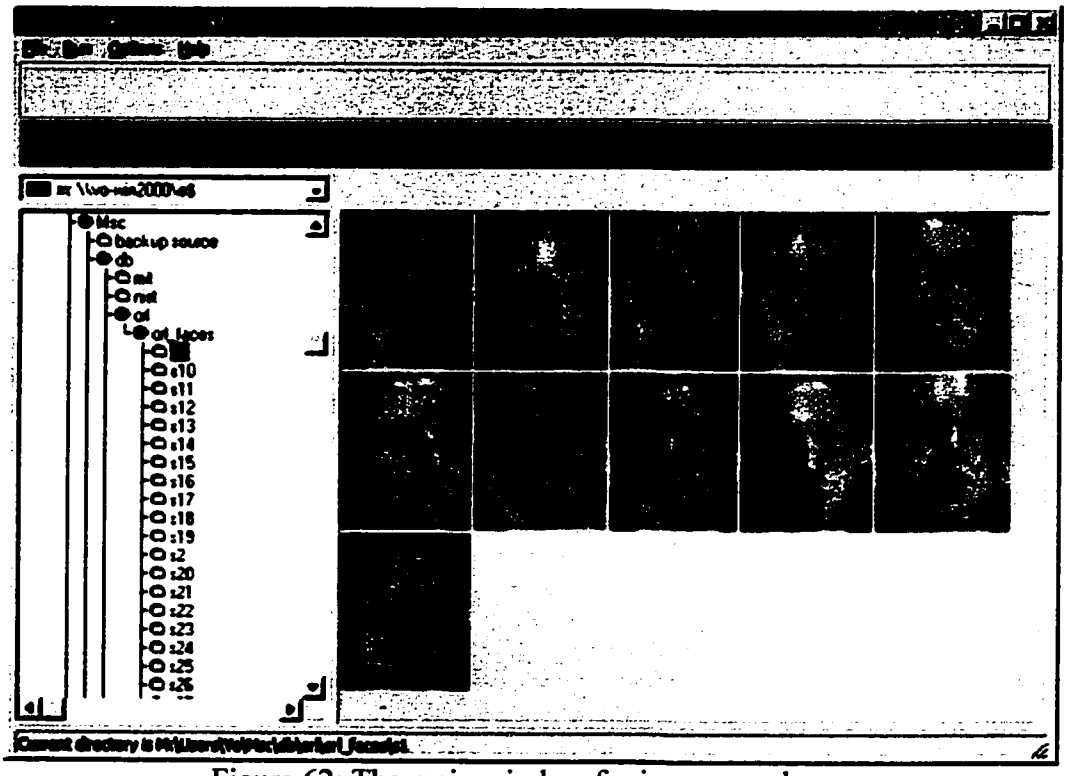


Figure 62: The main window for images explorer.

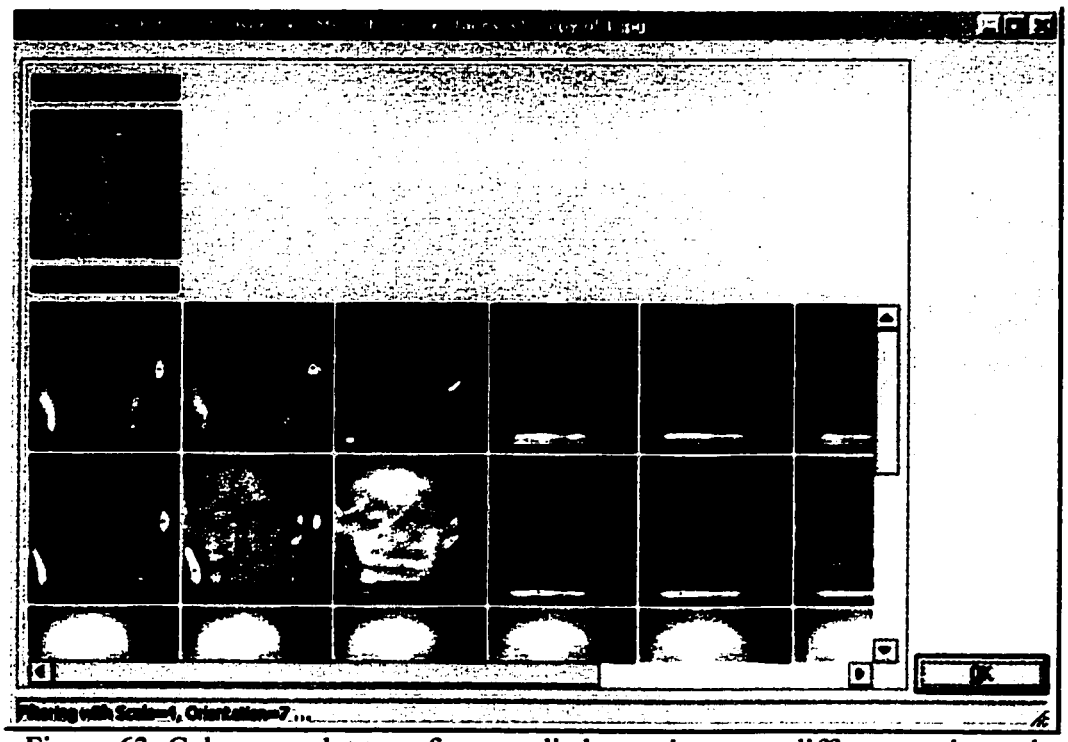


Figure 63: Gabor wavelet transform applied to an image at different scales and orientations.

In figure 63, one can select from the Run menu, an option to compute the Gabor wavelet transform for all images on disk inside the current folder, or simply select an image in the image explorer window to have it computed.

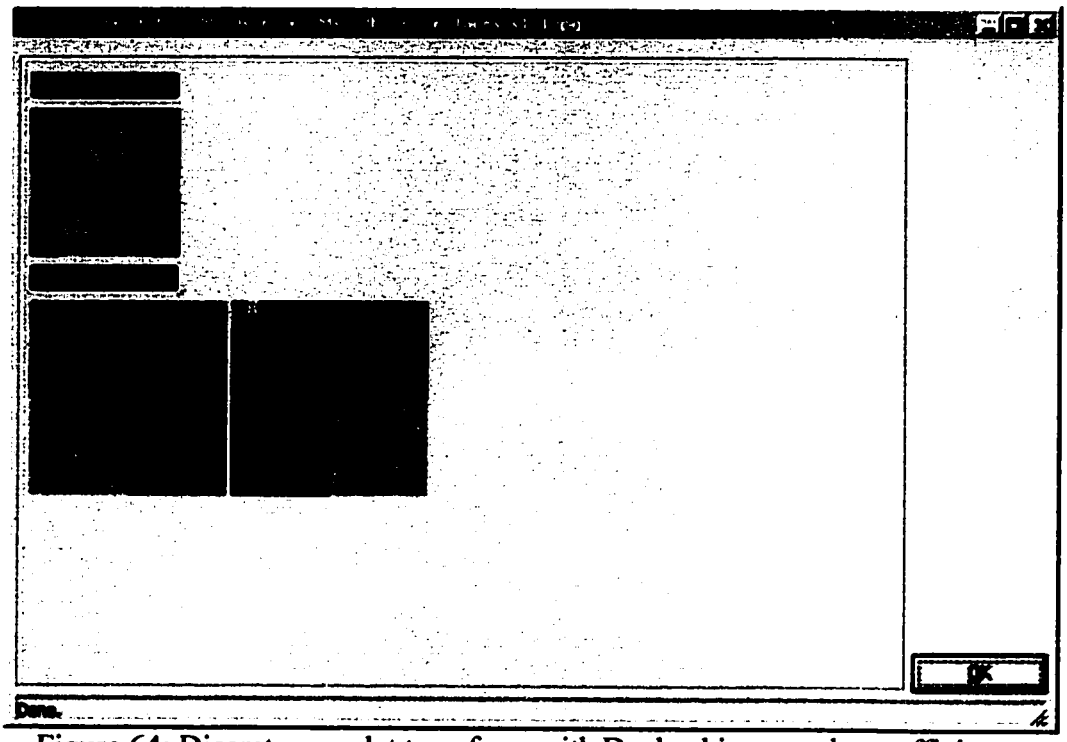


Figure 64: Discrete wavelet transform with Daubechies wavelet coefficients.

In figure 64. the Run menu also offers the option of computing the discrete wavelet transform, using Daubechies wavelet coefficients chosen from the Options.

The Compute Correlation submenu offers the option of testing correlation directly on grayscale images, on each scale and orientation for Gabor wavelet transform data or on each subimage on each scale in the case of discrete wavelet transform.

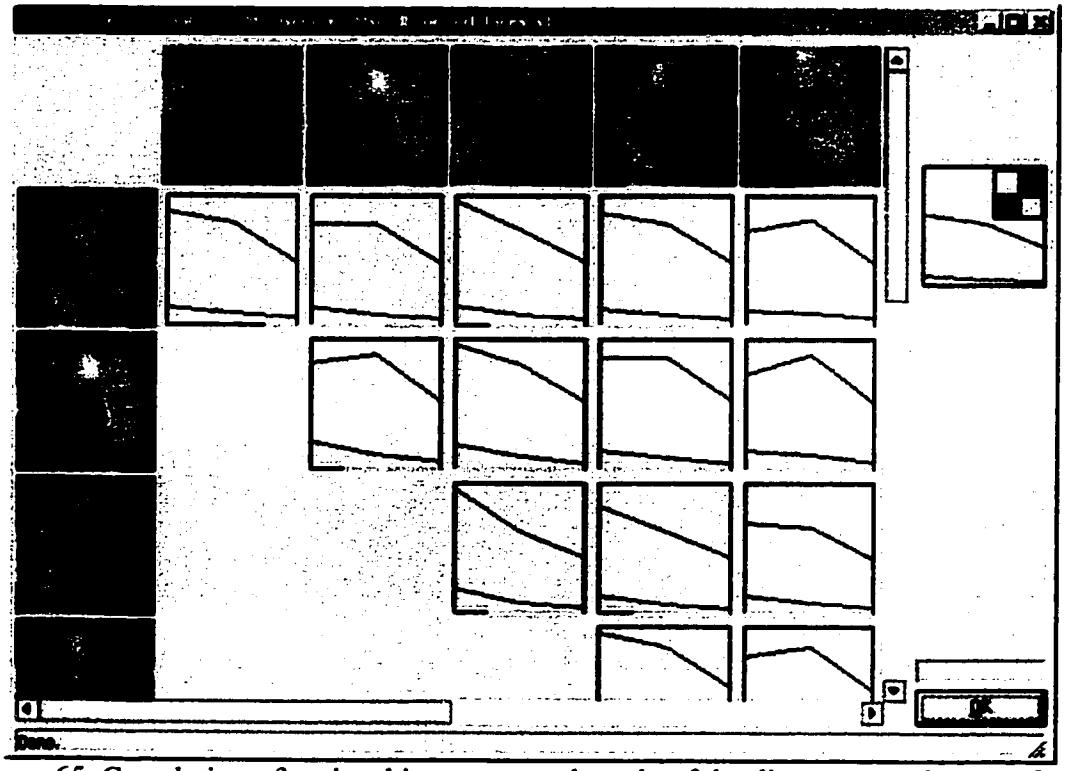


Figure 65: Correlation of each subimage on each scale of the discrete wavelet transform.

In figure 65, each correlation of two persons gives out a graph with three lines joining the three scale corresponding to the three level wavelet decomposition. These three lines correspond to the vertical differential component, the horizontal and the diagonal differential component at each scale.

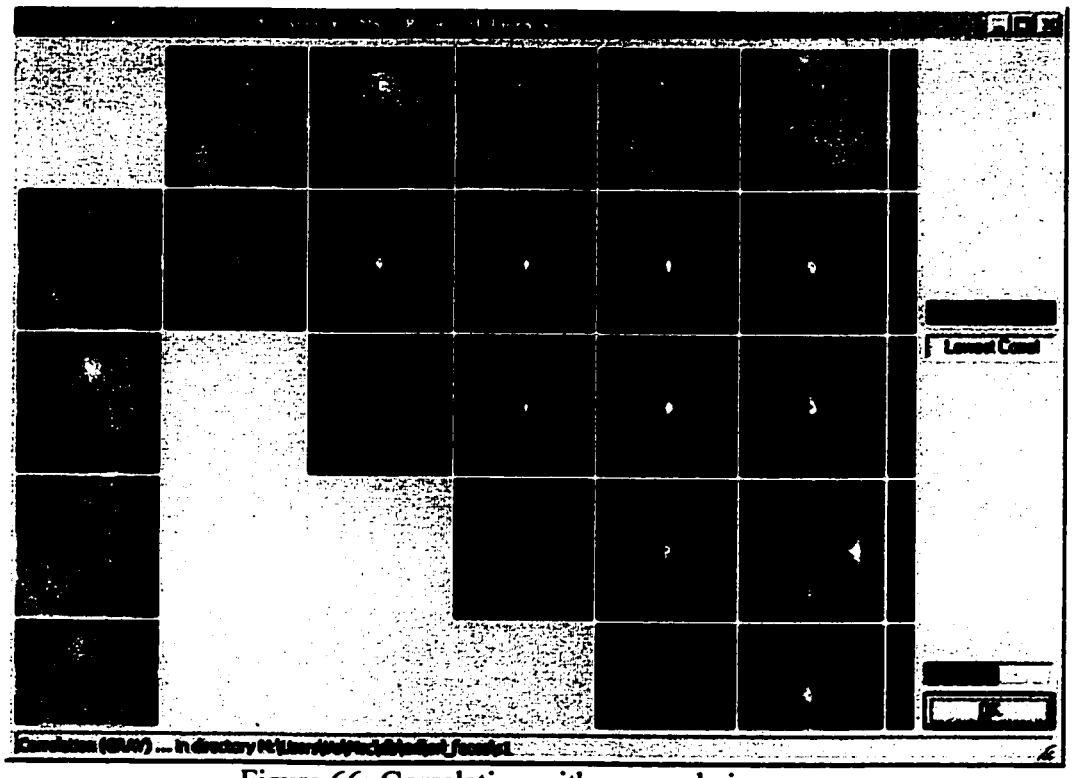


Figure 66: Correlation with grayscale images.

In figure 66, the correlation, computed by Fast Fourier transform, of the same face image presents a centered bright spot while the correlation of two face images with different poses presents a larger area.

The Run menu also offer matching options, using either Gabor wavelet transform data or discrete wavelet transform. The matching process can be with deformable grid matching, or simply a rigid grid matching.

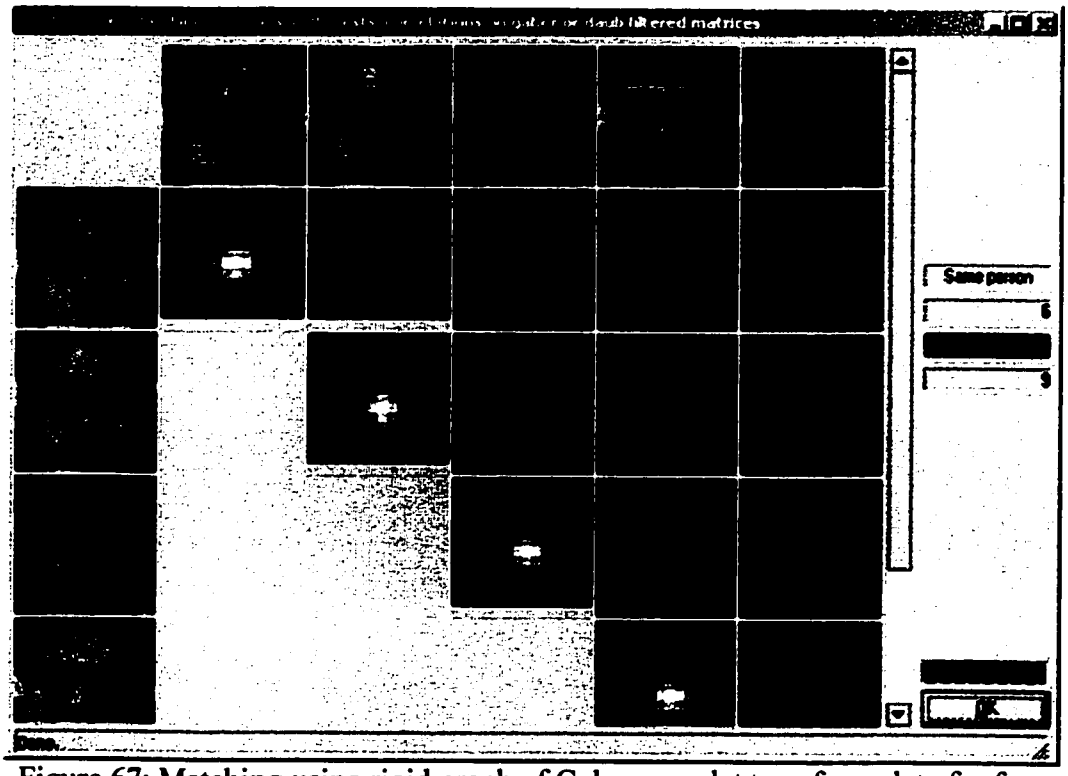


Figure 67: Matching using rigid graph of Gabor wavelet transform data for four individuals.

In figure 67, a grid is laid over the data obtained from the Gabor wavelet transform convolved with the image for one person is matched with the grid from another one. Under a threshold, the matching of one grid to another one determines if one image corresponds to the other, i.e. the same person. The brightest spot in the matching image corresponds to the best match location when one grid is moved around to match the reference one.

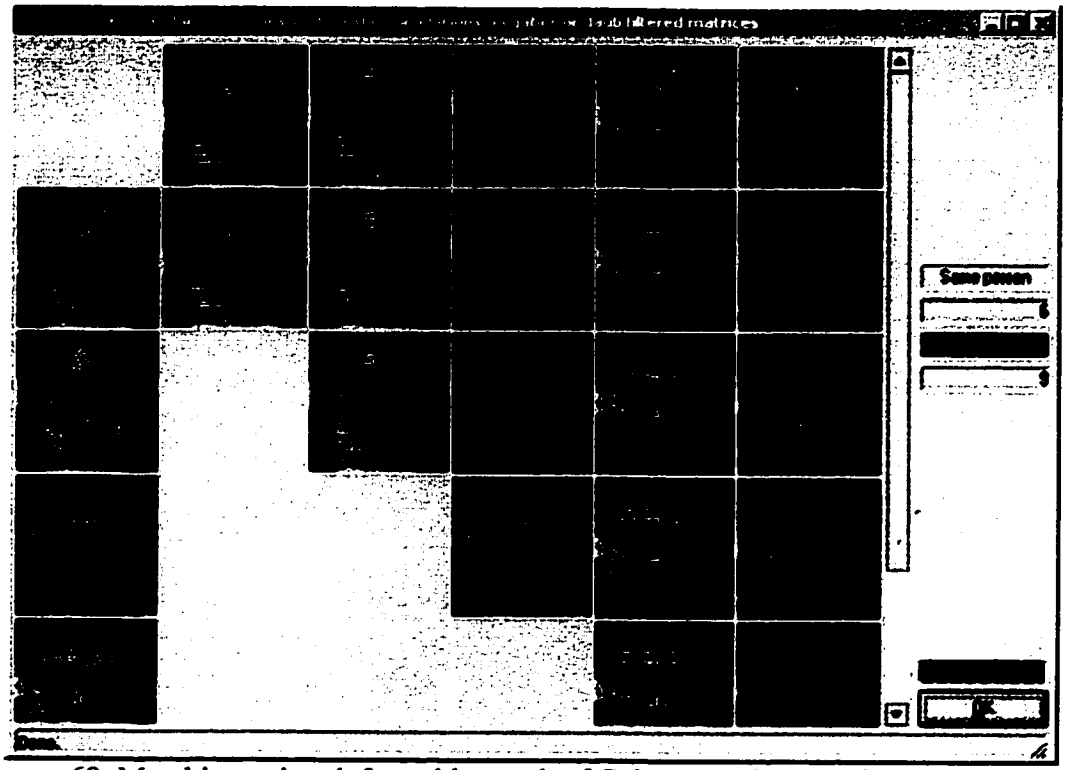


Figure 68: Matching using deformable graph of Gabor wavelet transform data for four individuals.

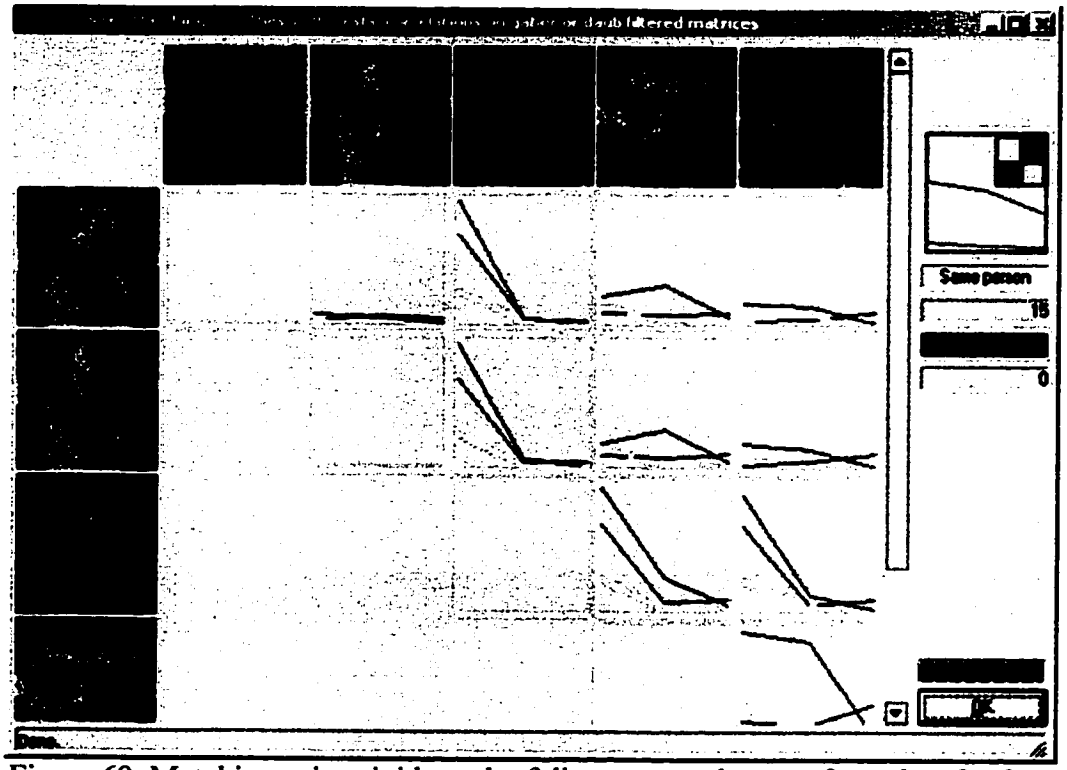


Figure 69: Matching using rigid graph of discrete wavelet transform data for four individuals.

In figures 68 and 69, as the first two images are coming from the same person, under different pose, the matching of their series of grid shows a small dissimilarity. Under a class dependent threshold, the matching succeeds for either authentication or identification scheme.

5.4.2. Runtime options

The program offers several runtime options, such as multithreading options, Gabor wavelet transform, Daubechies coefficients, and matching options.

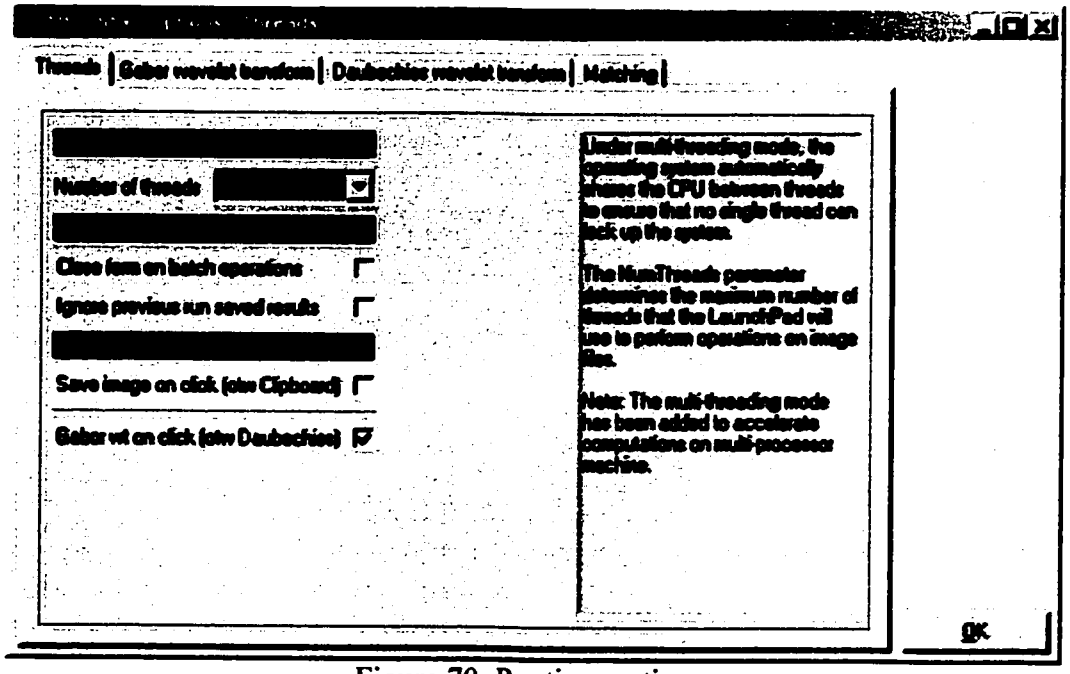


Figure 70: Runtime options.

In figure 70, we have the options of choosing the number of threads when running under a machine with multiprocessors, in order to accelerate computation such as for the Gabor wavelet transform.

The Gabor wavelet transform screen allows a choice of the number of frequencies, the number of orientations as well as its width. The Daubechies wavelet transform screen

allows choosing the desired coefficient scheme, such as DAUB1, DAUB2 ... until DAUB20.

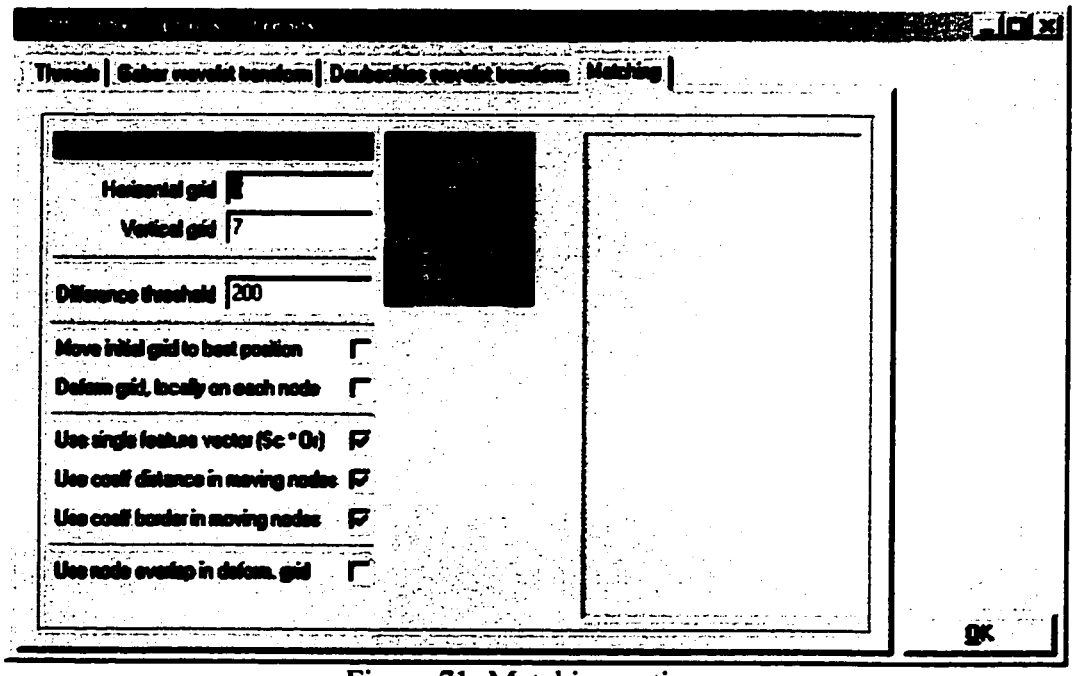


Figure 71: Matching options.

The Matching submenu allows experimentations on different ways of graph matching, especially the border coefficients and displacement coefficients activation (see figure 71).

5.5. Experimental results

As the ORL database contains sometimes strong different face expressions, in-plane and out-plane rotations, different scales, and accessories such as glasses, the recognition algorithm may introduces some errors and reduces its recognition rate when a performance test is run on the whole database.

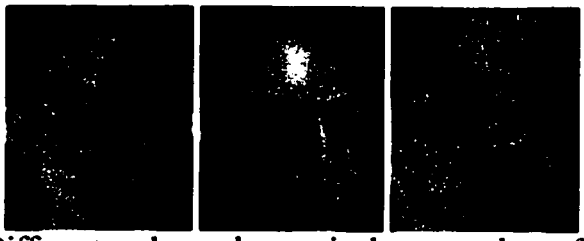


Figure 72: Different scales and poses in the same class of individuals.

In order to study the performance of the proposed face recognition system, we perform a test run on the whole ORL database, including faces with small changes in scale, in-plane and out-plane rotations (see figure 72), displacement as well as faces with facial expression changes and accessories such as glasses (see figure 73). The face images from the ORL are used without any low-level image processing and re-alignment or de-rotation applied. The recognition rate in table 3, as the inverse of false match, reaches 81% on a complete 400 raw images database (see appendix A).

Another indicator, such as the false rejection rate, is used to discriminate face images with large changes, i.e. when the difference to the template of the class is larger than its corresponding threshold, computed as the standard deviation. It has no effect, however, in the recognition rate when an incoming face image is to be matched against the internal database of a fixed number of subjects.

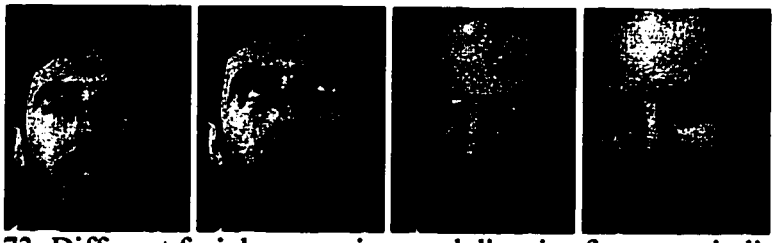


Figure 73: Different facial expressions and disguise from two individuals.

		2
		1
	3	2
	1	1
	6	
	1	1
	5	
	8	1
	2	2
		2
	9	
		2
	1	1
	1	2
	4	1
	3	2
		1
		3
		2
	8	1
	8	
	1	3
	4	2
	1	3
	1	3
	2	3
	1	2
	2	3
	4	3
	1	3
		3
	1	4
	5	1
	1	2
	2	3
	1	4
	5	2
	1	1
	5	2
	5	2
	25.75%	19.00%

Table 3: Performance of recognition on the complete ORL database.

Our face recognition system reaches 91.3% of recognition rate on a pre-selected subset of the ORL database, when the images with large facial expressions are manually removed

from each class. However, the subset of 46 images even retains the images with changes in scale and rotation, in the results from table 4 (see appendix B).

	1	1
	1	1
	1	
	3	
	1	2
	15.22%	8.70%

Table 4: Performance of recognition on a subset of 46 images.

Chapter 6

Conclusions

6.1. Conclusion on experimentations

In this work, we present a comparison of three techniques of face recognition on a view-based approach for recognizing faces under frontal, and near frontal poses. With the experimental results, the FFT is proved to be very efficient in giving a rough location of the face on the other image while being insufficient in acting as a classifier for face recognition, either for identification or authentication. The Daubechies wavelet transform approach in the literature is very computational and storage friendy, but difficult to implement for face related problems due to its dependencies on highpass data. While the Gabor wavelet transform is invariant to small changes, its dependency on the lowpass data makes it a weak classifier.

However, motivated by the success of the Gabor wavelet transform for its invariance to small scale change, small pose change (in-plane and out-plane rotation) and small translation, by the success of the discrete wavelet transform with Daubechies coefficients for its speed and storage friendly, we have presented a hybrid approach, using the grid and graph matching from the Gabor wavelet transform approach, applied to the three-level decomposition with the discrete wavelet transform with a normalization grid matching at each level.

We have also presented another approach in representing the face database with different classes into one image in each class, along with its transformed data and computed mean and class dependent threshold. The implementation has helped to verify the theoretical aspects, from the literature, and of this work.

6.2. Remarks and future works

The global approach, i.e. entire face transformation, is more suitable for face recognition than for face detection in cluttered scenes. In the future, we plan to apply this normalized feature vector from discrete wavelet transform combined deformable graph matching scheme into localized area to detect smaller features, such as eyes, and experience the affine invariant wavelet transform to overcome the problem of scale and pose dependencies. We would also like to improve the preprocessing part of the system for normalization processes.

While a face recognition system has been described, using different transformation and matching techniques, more work still need to be done to create a truly general face recognition system.

Bibliography

[1] C. Tyler, R. Miller. Computational approaches to face recognition, *implementation for ARVO (http://www.ski.org/CWTyler_lab/CWTyler/PrePublications/ARVO/1998 /FaceRecog/index.html)*, 1998.

[2] M. S. Lew, N. Huijsmans. Information Theory and Face Detection, *technical report from the Department of Computer Science, Leiden University*, October 1997.

[3] S. Kullback. Information Theory and Statistics, *Wiley*, New York, 1959.

[4] R. L. Hsu, M. Abdel-Mottaleb, A. K. Jain. Face Detection in Color Images, *technical report from Department of Computer Science & Engineering, Michigan State University*, January 2001.

[5] A. Hilti, I. Nourbakhash, B. Jensen, R. Siegwart. Narrative-level visual interpretation of human motion for human-robot interaction, *proceedings of IROS 2001*, Hawaii, October 29 – November 3, 2001.

[6] B. Takács, H. Wechsler. Detection of faces and facial landmarks using iconic filter banks, *Pergamon, pp. 1623-1635*, 1997.

[7] P. Kruizinga, N. Petkov. Person identification based on multiscale matching of cortical images, *Proceedings of the International Conference and Exhibition on High-Performance Computing and Networking, HPCN Europe '95, Milan, Italy, May 3-5 1995, Vol. 919 of Lecture Notes in Computer Science, Berlin: Springer-Verlag, pp. 420-427*, 1995.

[8] J. Huang, R. Zabih. Combining Color and Spatial Information for Content-based Image Retrieval, *European Conference on Digital Libraries*, September 1998.

[9] Beumier. Face identification, *implementation from the Royal Military Academy of Belgium*, 1997.

[10] J. Huang, D. Li, X. Shao, H. Wechsler. Pose Discrimination and Eye Detection Using Support Vector Machines, *proceedings of NATO-ASI on Face Recognition : From Theory to Applications, Springer Verlag, pp. 528-536*, 1998.

[11] S. J. McKenna, S. Gong, R. P. Würtz, J. Tanner, D. Banin. Tracking Facial Feature Points with Gabor Wavelets and Shape Models, *lecture notes in Computer Science, Springer Verlag, to appear in Proceedings 1st International Conference on Audio and Video based Biometric Person Authentication*, 1997.

[12] J. Matas, K. Jonsson, J. Kittler. Fast Face Localisation and Verification, *technical report from the Centre for Vision, Speech and Signal Processing, University of Surrey, submitted to Elsevier Preprint*, April 1999.

[13] J. Kittler, J. Matas, K. Jonsson, M. U. R. Sánchez. Combining Evidence in Personal Identity Verification Systems, *Pattern Recognition Letters, pp. 845-852*, September 1997.

[14] A. Tankus, Y. Yeshurun, N. Intrator. Face detection by direct convexity estimation, *Pattern Recognition Letters 18, pp. 913-922*, 1997.

[15] D. J. Beymer. Face Recognition Under Varying Pose, *technical report from the Massachusetts Institute of Technology, Artificial Intelligence Laboratory, C. B. C. L. Paper No. 89, A. I. Memo No. 1461*, December 1993.

[16] A. Lanitis, C. J. Taylor, T. F. Cootes. Automatic Interpretation and Coding of Face Images Using Flexible Models, *IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 19, No. 7*, July 1997.

[17] S. McKenna, S. Gong. Tracking Faces, *Second International Conference on Automated Face and Gesture Recognition, Vermont*, October 1996.

[18] K. C. Yow, R. Cipolla. Finding Initial Estimates of Human Face Location, *technical report CUED/F-INFENG/TR 239 from the Department of Engineering, University of Cambridge*, October 1995.

[19] K. C. Yow, R. Cipolla. Feature-Based Human Face Detection, *technical report CUED/F-INFENG/TR 249 from the Department of Engineering, University of Cambridge*, August 1996.

[20] K. C. Yow, R. Cipolla. Scale and Orientation Invariance in Human Face Detection, *technical report from the Department of Engineering, University of Cambridge*, 1996.

[21] M. Lando, S. Edelman. Receptive field spaces and class-based generalization from a single view in face recognition, *technical report from the Department of Applied Mathematics and Computer Science, The Weizmann Institude of Science*, August 1995.

[22] J. Bala, K. DeJong, J. Huang, H. Vafaie, H. Wechsler. Visual Routine for Eye Detection using Hybrid Genetic Architectures, *13th International Conference on Pattern Recognition (ICPR), Austria*, 1996.

[23] J. Huang, C. Liu, H. Wechsler. Eye Detection and Face Recognition Using Evolutionary Computation, *technical report from the Department of Computer Science, George Mason University*, 1998.

[24] J. Bala, K. DeJong, J. Huang, H. Vafaie, H. Wechsler. Using Learning to Facilitate the Evolution of Features for Recognizing Visual Concepts, *technical report from the Datamat System Research, Inc, appeared in the Special Issue of Evoluationary Computation*, 1996.

[25] S. Jeng, H. Y. M. Liao, C. C. Han, M. Y. Chern, Y. T. Liu. Facial feature detection using geometrical face model: An efficient approach. *Pattern Recognition, Vol. 31, No. 3, pp. 273-282*, 1998.

[26] T. K. Leung, M. C. Burl, P. Perona. Finding Faces in Cluttered Scences using Random Labeled Graph Matching, *technical report from the California Institute of Technology, the University of California at Berkeley and Università di Padova, appeared in the Fifth International Conference on Computer Vision*, June 1995.

[27] S. Romdhani. Face Recognition using Principal Components Analysis, *MSc Thesis, Department of Electronics and Electrical Engineering, University of Glasgow*, March 1996.

[28] H. Abdi, D. Valentin. De la reconnaissance des Objets à la Reconnaissance des Visages, *F. Cordier Edition, pp. 13-41.*

[29] A. Pentland, B. Moghaddam, T. Starner. View-Based and Modular Eigenspaces for Face Recognition, *technical report of the M.I.T Media Laboratory Perceptual Computing Section, No. 245, appeared in the IEEE Conference on Computer Vision and Pattern Recognition*, 1994.

[30] B. Moghaddam, A. Pentland. An Automatic System for Model-Based Coding of Faces, *technical report of the M.I.T. Media Laboratory Perceptual Computing Section, No. 317, appeared in the IEEE Data Compression Conference*, March 1995.

[31] P. J. B. Hancock, V. Bruce. A comparaison of two computer-based face identification systems with human perceptions of faces, *technical report from the Department of Psychology, University of Stirling, to appear in Vision Research*, November 1997.

[32] R. Feraud. PCA, Neural Networks and Estimation for Face Detection, *proceedings of the NATO Advanced Study Institute to Applications, Stirling, Scotland, UK*, 1997.

[33] P. J. B. Hancock, A. M. Burton, V. Bruce. Face processing : human perception and principal components analysis, *Memory and Cognition 24, pp. 26-40*, 1996.

[34] P. J. B. Hancock, A. M. Burton, V. Bruce. Preprocessing images of faces : Correlations with human perceptions of distinctiveness and familiarity, *to appear in proceedings of IEEE fifth international conference on image processing and its application, Edinburgh*, July 1995.

[35] P. J. B. Hancock, A. M. Burton, V. Bruce. Testing Principal Component Representations for Faces, *proceedings of 4th Neural Computation and Psychology Workshop, Springer-Verlag, pp. 84-97*, 1997.

[36] T. E. de Campos, R. S. Feris, R. M. Cesar Junior. Eigenfaces versus Eigeneyes : First Steps Toward Performance Assessment of Representations for Face Recognition, *lecture notes in Artificial Intelligence, vol. 1793, pp. 197-206, Springer-Verlag press*, April 2000.

[37] K. K. Sung, T. Poggio. Example-based Learning for View-based Human Face Detection, *technical report from the Center for Biological and Computational Learning and the Massachusetts Institute of Technology Artificial Intelligence Laboratory, A.I. memo no. 1512, C.B.C.L. paper no. 112*, December 1994.

[38] H. A. Rowley, S. Baluja, Takeo Kanade. Human Face Detection in Visual Scenes, *technical report CMU-CS-95-158R from the School of Computer Science, Carnegie Mellon University*, November 1995.

[39] Steve Lawrence, C. Lee Giles, Ah Chung Tsoi, Andrew D. Back. Face Recognition: A Convolutional Neural Network Approach, *IEEE Transactions on Neural Networks, Special Issue on Neural Networks and Pattern Recognition, Volume 8, Number 1, pp. 98-113*, 1997.

[40] S. Gong, A. Psarrou, I. Katsoulis, P. Palavouzis. Tracking and Recognition of Face Sequences, *European Workshop on Combined Real and Synthetic Image Processing for Broadcast and Video Production, Hamburg, Germany*, November 1994.

[41] S. Edelman, D. Reisfeld, Y. Yeshurun. Learning to recognize faces from examples, *technical report from the Department of Applied Mathematics and Computer Science, The Weizmann Institute of Science*, October 1991.

[42] L. Wiskott, C. von der Malsburg. Face Recognition by Dynamic Link Matching, *internal report from the Institut für Neuroinformatik*, May 1996.

[43] H. A. Rowley, S. Baluja, Takeo Kanade. Rotation Invariant Neural Network-Based Face Recognition, *technical report CMU-CS-97-201 from the School of Computer Science, Carnegie Mellon University*, December 1997.

[44] S. Baluja. Face Detection with In-Plane Rotation: Early Concepts and Preliminary Results, *technical report JPRC-TR-97-001 from the School of Computer Science, Carnegie Mellon University*, 1997.

[45] S. Hung Lin, Y. Chan, S. Y. Kung. A probabilistic decision-based neural network for locating deformable objects and its applications to surveillance system and video browsing, *International Conference on Acoustics, Speech and Signal Processing, vol. 6, pp. 3553-3556*, 1996.

[46] N. Intrator, D. Reisfeld, Y. Yeshurun. Face Recognition using a Hybrid Supervised / Unsupervised Neural Network, *technical report from the Department of Computer Science, Tel-Aviv University*, June 1995.

[47] H. Abdi, D. Valentin. Modèles Neuronaux, Connexionistes et Numériques pour la Mémoire des Visages, *technical report from the School of Human Development, University of Texas and Université de Bourgogne à Dijon, Psychologie Française, 39(4), pp. 357-392*, 1994.

[48] R. Feraud, O. Bernier. Ensemble and Modular Approaches for Face Detection : a Comparaison, *Neural Information Processing System 10, pp. 472-478*, 1997.

[49] P. Juell, R. Marsil. A hierarchical neural network for human face detection, *Pergamon, pp. 781-787*, August 1995.

[50] H. A. Rowley, S. Baluja, Takeo Kanade. Neural Network-Based Face Detection, *technical report submitted to Computer Vision and Pattern Recognition*, 1996.

[51] C. Wong, D. Kortenkamp, M. Speich. A Mobile Robot That Recognizes People, technical report submitted to the IEEE International Conference on Tools with Artificial Intelligence, 1995.

[52] A. Psarrou, S. Gong, H. Buxton. Modelling Spatio-Temporal Trajectories and Face Signatures on Partially Recurrent Neural Networks, *IEEE International Conference on Neural Networks, Australia*, 1995.

[53] S. Ranganath, K. Arun. Face recognition using transform features and neural networks, *Pergamon, pp. 1615-1622*, November 1996.

[54] K. S. Yoon, Y. K. Ham, R. H. Park. Hybrid approaches to frontal view face recognition using the hidden Markov model and neural network, *Pergamon, pp. 283-292*, April 1997.

[55] J. Magarey, A. Dick. Multiresolution Stereo Image Matching Using Complex Wavelets, *proceedings of 14th International Conference on Pattern Recognition (ICPR), vol. I, pp. 4-7*, August 1998.

[56] V. H. S. Ha, J. M. F. Moura. Affine Invariant Wavelet Transform, *technical report from the Department of Electrical and Computer Engineering, Carnegie Mellon University*, 2001.

[57] R. Piché. The Discrete Wavelet Transform, *supplementary notes for the course 73110 Numerical Analysis, University of Technology*, February 1999.

[58] B. Jawerth, W. Sweldens. An Overview of Wavelet Based Multiresolution Analyses, February 1993.

[59] C. Garcia, G. Zikos, G. Tziritas. Wavelet Packet Analysis for Face Recognition, *technical report from the Institute of Computer Science, Foundation for Research and Technology Hellas, submitted to Elsevier Preprint*, December 1999.

[60] N. Kingsbury, J. Magarey. Wavelet transforms in Image Processing, *Signal Processing and Prediction I, EURASIP, ICT Press, pp. 23-34, Prague*, 1997.

[61] E. Loupias, N. Sebe. Wavelet-based Salient Points for Image Retrieval, *research report RR 99.11 from the Laboratoire Reconnaissance de Formes et Vision, INSA Lyon*, November 1999.

[62] C. Kuglin, D. Hines. The Phase Correlation Image Alignment Method, *in Proceedings of International Conference Cybernetics and Society, pp. 163-165*, 1975.

[63] H. P. Pan. Uniform full-information image matching using complex conjugate wavelet-pyramid, *Wavelet Applications in Signal and Image Processing IV, vol. 2825 of Proceedings of SPIE, pp. 697-721*, August 1996.

[64] J. Walder. Using 2-D wavelet analysis for matching two images, *tutorial from the Technical University of Ostrava (http://www.cg.tuwien.ac.at/cgi-bin/toISO-8859-2.en/studentwork/CESCG-2000/JWalder/index.html#kap1).*

[65] S. Krüger, A. Calway. Image Registration using Multiresolution Frequency Domain Correlation, *technical report from the Department of Computer Science, appeared in British Machine Vision Conference, pp. 316-325*, September 1998.

[66] T. Kanade. Picture processing by computer complex and recognition of human faces, *technical report from the Department of Inform. Sci., Kyoto University*, 1973.

[67] J. H. Lai, P. C. Yuen, G. C. Feng. Spectroface: A Fourier-based Approach for Human Face Recognition, *technical report from the Department of Computer Science, Hong Kong Baptist University, Hong Kong*, 1999.

[68] I. Daubechies. Orthogonal Bases of Compactly Supported Wavelets, *Communications on Pure Applied Mathematics, vol. 41, pp. 909-996*, 1988.

[69] M. K. Mandal, T. Aboulnasr. Image Indexing Using Moments and Wavelets, *IEEE Transactions on Consumer Electronics, vol. 42, no. 3, pp. 557-565*, August 1996.

[70] C. Kato, Y. Hayashi, H. Tsuuchiya, Y. Saito, K. Horii. The Fourier – Wavelet Transform Method for the Extraction of Facial Expression, *Proceedings of PSFVIP – 3*, March 18-21, 2001.

[71] M. Lades, J. C. Vorbrüggen, J. Buhmann, J. Lange, C. von der Malsburg, R. P. Würtz, W. Konen. Distortion invariant object recognition in the dynamic link architecture, *IEEE Transactions on Computers, vol. 42, pp. 300-311*, 1993.

[72] M. Pötzsch, M. Rinne. Gabor Wavelet Transform, *web tutorial (http://www.neuroinformatik.ruhr-uni-bochum.de/ini/VDM/research/computerVision/imageProcessing/wavelets/gabor/contents.html)*, April 1996.

[73] L. Wiskott, J.M. Fellous, N. Krüger, C. von der Malsburg. Face Recognition by Elastic Bunch Graph Matching, *in Intelligent Biometric Techniques in Fingerprint and Face Recognition, eds. L.C. Jain et al., publ. CRC Press, ISBN 0-8493-2055-0, Chapter 11, pp. 355-396*, 1999.

[74] G. M. Haley, B. S. Manjunath. Rotation-Invariant Texture Classification Using a Complete Space-Frequency Model, *IEEE Transactions on Image Processing, Vol. 8, No. 2*, February 1999.

[75] W. H. Press, S. A. Teukolsky, W. T. Vetterling, B. P. Flannery. Numerical Recipes in C, The Art of Scientific Computing, Second Edition, in Cambridge University Press, 1992.

[76] B. Duc, S. Fischer, J. Bigün. Face Authentication with Gabor Information on Deformable Graphs, *report submitted for publication in IEEE Transactions on Image Processing,* June 1998.

[77] R. P. Würtz. Object Recognition Robust Under Translations, Deformations, and Changes in Background, *IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 19, no. 7, pp. 769-779,* July 1997.

[78] J. P. Lewis. Fast Template Matching, *in Vision Interface, pp. 120-123,* 1995.

Appendices

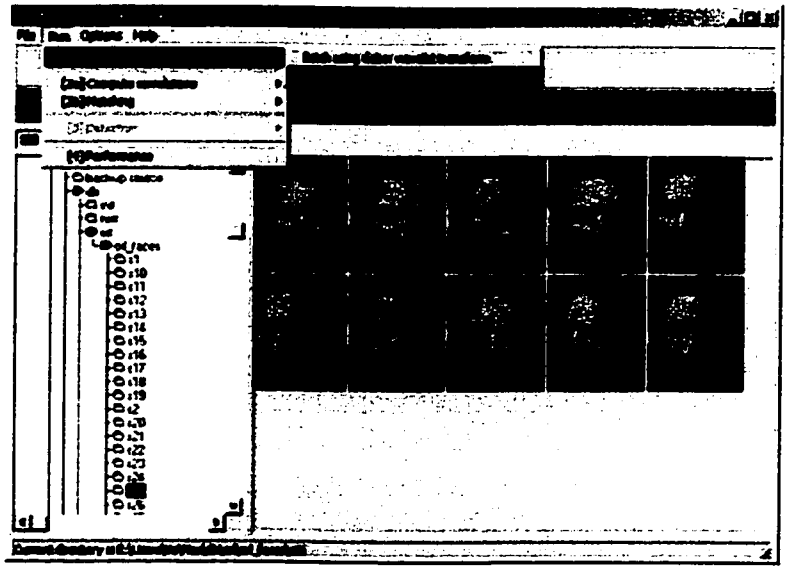
Appendix A

Example of run on the complete ORL database

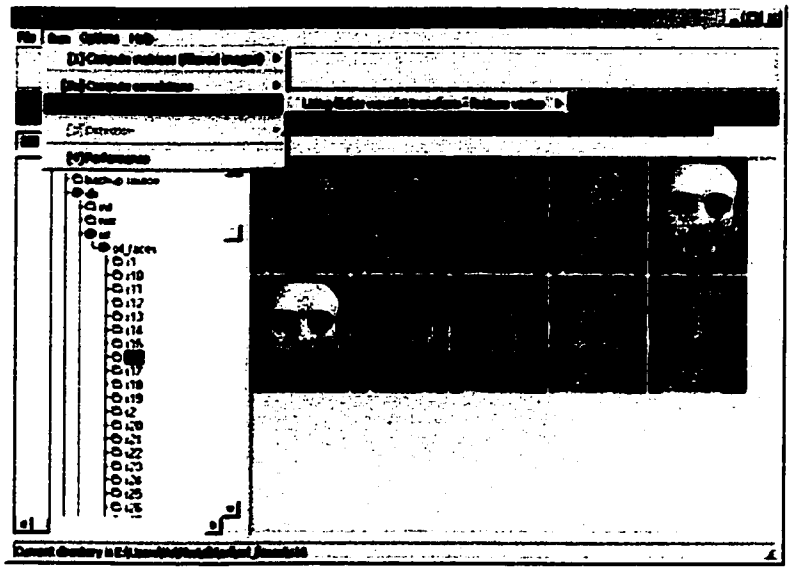
Three main steps are to be executed:

- Compute the Daubechies wavelet transform on each of the image
- Compute the template average and standard deviation for each class
- Match each image of the 400 images database to the list of templates (40 in the ORL database)

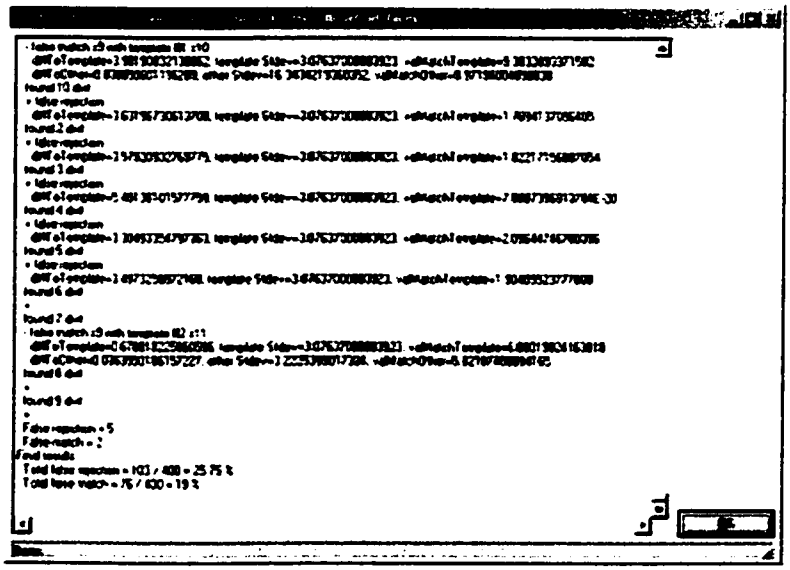
For each directory of the ORL database, we compute the decomposition by Daubechies wavelet transform by executing the menu Run – Compute Matrices – Batch using Daubechies wavelet transform. A binary file containing the decomposition result is created for each face image in the selected directory.



For each class of the ORL database, we compute the average and standard deviation by executing the menu Run – Matching – Using Daubechies wavelet transform – Rigid template / grid. A text file, called template.txt, is created in the selected directory, and contains statistics on matching one image to the other.



Going back to the parent directory "orl_faces", we launch the massive performance test from the menu Run - Performance, which produces statistics on matching each image of the 400 images database to the list of 40 templates computed in the previous steps.



Appendix B

Example of log produced by the performance run on a small subset

```
Current directory is E:\Msc\db\tests\good performance
Searching subdirectory s1
. found template.txt
. found 9 .dwt files
. 1.dwt average = 1.64621794223785 stdev = 1.16931903362274
. 2.dwt average = 1.29864394664764 stdev = 1.10487294197083
. 3.dwt average = 1.88392603397369 stdev = 1.23535299301147
. 4.dwt average = 1.31162297725677 stdev = 1.13788998126984
. 5.dwt average = 1.36009395122528 stdev = 1.10439002513885
. 6.dwt average = 3.71780800819397 stdev = 1.4235919713974
. 7.dwt average = 1.52906894683838 stdev = 1.04916596412659
. 8.dwt average = 1.76553797721863 stdev = 1.1487820148468
. 9.dwt average = 1.3249009847641 stdev = 1.10626196861267
. smallest is 2.dwt average = 1.29864394664764 stdev = 1.10487294197083
. add template.txt for s1
Searching subdirectory s12
. found template.txt
. found 10 .dwt files
. 1.dwt average = 8.1622428894043 stdev = 12.9501075744629
. 10.dwt average = 7.96559381484985 stdev = 12.9626970291138
. 2.dwt average = 7.98812198638916 stdev = 12.8396873474121
. 3.dwt average = 38.8473129272461 stdev = 13.7214021682739
. 4.dwt average = 8.66503810882568 stdev = 12.9038181304932
. 5.dwt average = 9.46107292175293 stdev = 12.4512434005737
. 6.dwt average = 18.1906814575195 stdev = 11.4430637359619
. 7.dwt average = 8.89774322509766 stdev = 12.8059463500977
. 8.dwt average = 8.38361167907715 stdev = 12.7328176498413
. 9.dwt average = 8.26640129089355 stdev = 13.0590333938599
. smallest is 10.dwt average = 7.96559381484985 stdev = 12.9626970291138
. add template.txt for s12
Searching subdirectory s18
. found template.txt
. found 10 .dwt files
. 1.dwt average = 18.7130355834961 stdev = 25.2009620666504
. 10.dwt average = 15.0863618850708 stdev = 25.1631202697754
. 2.dwt average = 16.0980491638184 stdev = 25.1913185119629
. 3.dwt average = 33.2934112548828 stdev = 14.2402276992798
. 4.dwt average = 14.5267581939697 stdev = 24.9830074310303
. 5.dwt average = 14.5631542205811 stdev = 25.2409286499023
. 6.dwt average = 14.2394161224365 stdev = 25.0980930328369
. 7.dwt average = 15.2227802276611 stdev = 25.1555728912354
. 8.dwt average = 71.0721588134766 stdev = 25.9522857666016
. 9.dwt average = 14.0094089508057 stdev = 25.0823974609375
. smallest is 9.dwt average = 14.0094089508057 stdev = 25.0823974609375
. add template.txt for s18
Searching subdirectory s23
. found template.txt
. found 8 .dwt files
. 1.dwt average = 2.64517211914063 stdev = 1.12796795368195
. 10.dwt average = 2.29645299911499 stdev = 1.04872798919678
. 3.dwt average = 1.65020799636841 stdev = 0.933712005615234
. 4.dwt average = 1.6369069814682 stdev = 0.899653971195221
. 5.dwt average = 1.87860405445099 stdev = 0.915775001049042
. 6.dwt average = 1.99842894077301 stdev = 1.05639803409576
. 7.dwt average = 1.65045499801636 stdev = 0.875358998775482
. 9.dwt average = 2.60217094421387 stdev = 1.14798998832703
. smallest is 4.dwt average = 1.6369069814682 stdev = 0.899653971195221
. add template.txt for s23
Searching subdirectory s30
. found template.txt
. found 9 .dwt files
. 1.dwt average = 5.17206478118896 stdev = 2.68811893463135
. 2.dwt average = 2.63228011131287 stdev = 2.38771796226501
. 3.dwt average = 2.6546471118927 stdev = 2.32255911827087
. 4.dwt average = 3.40459108352661 stdev = 2.27189803123474
. 5.dwt average = 7.56848907470703 stdev = 2.96109104156494
. 6.dwt average = 2.98351192474365 stdev = 2.4781539440155
. 7.dwt average = 3.09061098098755 stdev = 2.44068193435669
. 8.dwt average = 2.82783389091492 stdev = 2.45716190338135
```



```
. 9.dwt average = 3.03655791282654 stdev = 2.34117889404297
. smallest is 2.dwt average = 2.63228011131287 stdev = 2.38771796226501
. add template.txt for s30
Added 5 templates.
Computing grid.
Computing one single grid.
Computing match.
Working on subdirectory s1
. found 1.dwt
. +
. found 2.dwt
. + false rejection
    diffToTemplate=1.3249009847641, template Stdev=1.10626196861267, valMatchTem
plate=7.77868175712194E-20
. found 3.dwt
. +
. found 4.dwt
. +
. found 5.dwt
. +
. found 6.dwt
. - false match s1 with template #1 s12
    diffToTemplate=2.61557722091675, template Stdev=1.10626196861267, valMatchTe
mplate=3.940478324890l4
    diffToOther=4.11431884765625, other Stdev=13.0590333938599, valMatchOther=4.
1520824432373
. found 7.dwt
. +
. found 8.dwt
. +
. found 9.dwt
. +
. False rejection = 1
. False match = 1
Working on subdirectory s12
. found 1.dwt
. +
. found 10.dwt
. +
. found 2.dwt
. +
. found 3.dwt
. + false rejection
    diffToTemplate=34.1407318115234, template Stdev=13.0590333938599, valMatchTe
mplate=42.4071350097656
. found 4.dwt
. +
. found 5.dwt
. +
. found 6.dwt
. - false match s12 with template #2 s18
    diffToTemplate=8.54342460632324, template Stdev=13.0590333938599, valMatchTe
mplate=16.8098258972168
    diffToOther=2.75772857666016, other Stdev=25.0823974609375, valMatchOther=16
.7671375274658
. found 7.dwt
. +
. found 8.dwt
. +
. found 9.dwt
. +
. False rejection = 1
. False match = 1
Working on subdirectory s18
. found 1.dwt
. +
. found 10.dwt
. +
. found 2.dwt
```

```
. +
. found 3.dwt
. +
. found 4.dwt
. +
. found 5.dwt
. +
. found 6.dwt
. +
. found 7.dwt
. +
. found 8.dwt
. + false rejection
    diffToTemplate=66.2276840209961, template Stdev=25.0823974609375, valMatchTe
mplate=80.2370910644531
. found 9.dwt
. +
. False rejection = 1
. False match = 0
Working on subdirectory s23
. found 1.dwt
. +
. found 10.dwt
. +
. found 3.dwt
. + false rejection
    diffToTemplate=1.53470718860626, template Stdev=1.14798998832703, valMatchTe
mplate=1.0674637556076
. found 4.dwt
. + false rejection
    diffToTemplate=2.60217094421387, template Stdev=1.14798998832703, valMatchTe
mplate=7.77868175712194E-20
. found 5.dwt
. +
. found 6.dwt
. +
. found 7.dwt
. + false rejection
    diffToTemplate=1.28393292427063, template Stdev=1.14798998832703, valMatchTe
mplate=1.31823801994324
. found 9.dwt
. +
. False rejection = 3
. False match = 0
Working on subdirectory s30
. found 1.dwt
. - false match s30 with template #1 s12
    diffToTemplate=1.75456643104553, template Stdev=2.34117889404297, valMatchTe
mplate=4.79112434387207
    diffToOther=2.87883377075195, other Stdev=13.0590333938599, valMatchOther=5.
3875675201416
. found 2.dwt
. + false rejection
    diffToTemplate=3.03655791282654, template Stdev=2.34117889404297, valMatchTe
mplate=7.77868175712194E-20
. found 3.dwt
. +
. found 4.dwt
. +
. found 5.dwt
. - false match s30 with template #1 s12
    diffToTemplate=4.98537254333496, template Stdev=2.34117889404297, valMatchTe
mplate=8.02193069458008
    diffToOther=0.0213785171508789, other Stdev=13.0590333938599, valMatchOther=
8.28777980804443
. found 6.dwt
. +
. found 7.dwt
. +
```



```
. found 8.dwt
. +
. found 9.dwt
. +
. False rejection = 1
. False match = 2
Final results
. Total false rejection = 7 / 46 = 0.152173913043478 %
. Total false match = 4 / 46 = 0.0869565217391304 %
```