Escola Tècnica Superior d'Enginyeria de Telecomunicació de Barcelona

UNIVERSITAT POLITÈCNICA DE CATALUNYA

Department of Signal Theory and Communications

VIDEO-BASED FACE RECOGNITION USING MULTIPLE FACE ORIENTATIONS

AUTHOR: Edison Cristófani Calderó

ADVISOR: Prof. Josep Ramon Morros i Rubió

Barcelona, July 2009

Abstract

This work is focused on designing and implementing a real-time video-based face identification system with low memory and computational requirements and high recognition rates. Since profile features are stronger and, therefore, better when characterising faces than frontal faces, the system will detect and identify not only pure frontal but also profile faces. This property of profile faces will help to improve face recognition rates depending on the strategy for fusion of results used. Also, dimensionality reduction techniques will be studied and tested in order to find the fastest and most effective one. Modification in k Nearest Neighbor classifier will be carried out to add a penalisation factor in function of the distance, increasing classification results and strictness.

In order to find which are the best options for reducing computational requirements in a face identification system several simulations will be performed. Among many others, simulations will look for optimal values of the k parameter in k Nearest Neighbor, the number of transformed coefficients kept in a feature vector or the minimum size of face images and will test dimensionality reduction in images, variation of the number of models or fusion of results.

Finally, this work will show how a real-time system can be implemented in an ordinary computer obtaining successful results whether it be in real-time, adverse or controlled conditions environments.

Resumen

Este trabajo está centrado en el diseño e implementación de un sistema en tiempo real de identificación de caras basado en secuencias de vídeo que ofrezca un uso computacional y de memoria bajos y tasas de reconocimiento altas. Puesto que las características de las caras de perfil son más fuertes y, por tanto, mejores que las características de caras frontales, el sistema detectará e identificará no únicamente las caras totalmente frontales sino también las de perfil. Esta propiedad de las caras de perfil ayudará a mejorar las tasas de reconocimiento de caras dependiendo de la estrategia de fusión de resultados que se utilice. Se estudiarán también técnicas de reducción de dimensionalidad y se probarán con el fin de encontrar la que sea más rápida y efectiva. Se realizarán ciertas modificaciones en el classificador *k Nearest Neighbor* que penalizarán en función de la distancia, cosa que mejorará los resultados de las clasificaciones y la rigurosidad de las mismas.

Se llevarán a cabo diversas simulaciones para encontrar las mejores opciones que permitan reducir los requisitos computacionales de un sistema de identificación de caras. Entre otras muchas simulaciones, se buscarán los valores óptimos para el parámetro k del clasificador *k Nearest Neighbor*, el número de coeficientes transformados que se conservarán en los vectores de características o el tamaño mínimo en las imágenes de caras y se investigará la reducción de dimensionalidad en las imágenes, la variación del número de modelos utilizados o la fusión de resultados.

Por último, este trabajo mostrará cómo se puede implementar en un equipo de prestaciones limitadas un sistema en tiempo real que obtenga resultados satisfactorios ya sea en entornos en tiempo real, en condiciones adversas o controladas.

Resum

Aquest treball està centrat en el disseny i implementació d'un sistema en temps real d'identificació de cares basat en seqüències de vídeo que ofereixi un ús computacional i de memòria baixos i taxes de reconeixement altes. Donat que les característiques de les cares de perfil són més fortes i, per tant, millors que les característiques de cares frontals, el sistema detectarà i identificarà no únicament les cares totalment frontals sinó que també les de perfil. Aquesta propietat de les cares de perfil ajudarà a millorar les taxes de reconeixement de cares depenent de l'estratègia de fusió de resultats que s'utilitzi. S'estudiaran també tècniques de reducció de dimensionalitat i es provaran amb la finalitat de trobar la que sigui més ràpida i efectiva. Es realitzaran certes modificacions en el classificador *k Nearest Neighbor* que penalitzaran en funció de la distància, que millorarà els resultats i el rigor de les classificacions.

Es duran a terme diverses simulacions per a trobar les millors opcions que permetin reduir els requisits computacionals d'un sistema d'identificació de cares. Entre d'altres simulacions, es buscaran els valors òptims per al paràmetre k del classificador *k Nearest Neighbor*, el nombre de coeficients transformats que es conservaran en els vectors de característiques o la grandària mínima en les imatges de cares i s'investigarà la reducció de dimensionalitat en les imatges, la variació del nombre de models utilitzats o la fusió de resultats.

Finalment, aquest treball mostrarà com es pot implementar en un equip de prestacions limitades un sistema en temps real que obtingui resultats satisfactoris ja sigui en entorns en temps real, en condicions adverses o controlades.

Acknowledgements

I would like to thank **my family** for their true and unconditional support during all these years. **Prof. Ramon Morros**, thank you for the countless, interesting meetings and for guiding me through this master dissertation. Working with you was a pleasure. I thank **Prof. Ferran Marqués**, who arranged everything so I could work in the Image Processing Group. He also reviewed and approved my work. I can not forget to mention **Albert Gil**, who always helped me fixing all kinds of problems in my code. Thank you.

Agradecimientos

Me gustaría agradecer a **mi familia** su apoyo incondicional y sincero durante estos años. **Dr. Ramon Morros**, gracias por todas las incontables e interesantes reuniones y por guiarme durante este proyecto final de carrera. Ha sido un placer trabajar contigo. Mi agradecimiento al **Dr. Ferran Marqués**, que movió los hilos para que pudiera trabajar con el Grupo de Procesado de Imagen, revisó mi trabajo y dio su visto bueno. Y no puedo olvidarme de **Albert Gil**, que siempre estuvo dispuesto a ayudarme con todo tipo de problemas con mi código. Gracias también a ti.

Agraïments

M'agradaria agrair a **la meva família** el seu suport incondicional i sincer durant aquests anys. **Dr. Ramon Morros**, gràcies per totes les incomptables i interessants reunions i guiar-me durant aquest projecte fi de carrera. Ha estat un plaer treballar amb tu. El meu agraïment al **Dr. Ferran Marqués**, que va moure els fils perquè pogués treballar amb el Grup de Processament d'Imatge, va revisar el meu treball i va donar el seu vistiplau. I no puc oblidar-me de l'**Albert Gil**, que sempre va estar disposat a ajudar-me amb tota mena de problemes amb el meu codi. Gràcies a tu també.

Contents

Appendix:

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Context and motivation

Intelligent video security, automatic identification, tracking of individuals, indexation of faces from videorecordings or image bases, among many other applications, are now closer to be developed than ever. The social welfare is rapidly increasing its needs of this kind of applications and now is the time to take full advantage of it. A wide range of opportunities related to face identification or pattern recognition are yet to be discovered.

Technical improvements experienced during the last years allow implementing a face recognition application on current computers with any need of complex distributed processing. A new and very interesting horizon opens in biometrics, face recognition and face detection fields. In contrast to fingerprint and iris identifications, which are performed in controlled and almost ideal conditions, face detection and identification becomes a very exciting problem due to not only the quality of images needs to be taken into account but the many configuration parameters in an identification system.

Completely generic face identification applications in real time may require very demanding computational and memory resources not matching to nowadays standards. Strategies and techniques have to be created, implemented and tested, until a system is ready to work in its optimal point.

There are two types of face identification methods: feature-based and appearance-based. This work will focus on building an appearance-based face identification system. In these methods faces are considered to be groups of pixels that respond to a certain pattern which can be extracted and used for comparing between faces. Simulations involving several configuration parameters will be performed and identification systems will be discussed and tested, as well.

Feature-based methods only achieve high perfomance rates when having high-resolution face images. These methods look for relevant features in faces such as eyes, nose or mouth, being able to determine the identity of a given individual by comparing feature distances and positions with other candidates. It is not possible to find face features in environments where cameras are too far from the scene and faces are moving or have not enough resolution, therefore in these situations only appearance-based methods can be used. The process of extracting face features, comparing them with other faces and finding the closest one might seem apparently simple but it can be a very demanding and slow process if the system is not well configured. That is why it becomes necessary to restrict the size of images, the number of images per individual and other significant parameters.

Although it is possible to isolate or segment faces from an image by hand, this can be very unefficient when having 20 or 25 images per second and several cameras. A better option is having a face detector executing in a computer thanks to image processing techniques, which will be used in the final implementation of a face identification system, as detailed in this work. Face detector techniques can also be demanding if the whole image from the camera is scanned but these demands can be extremely reduced by avoiding certain zones where faces should not appear such as floors, ceilings, the sky, etc.

Most of the existing face identification systems only use frontal faces but in real-time situations a significant percentage of the detected faces can be profile faces, since individuals may move around the scenario or change the orientation of their bodies or heads. Also, profile faces are likely to be found in multicamera environments such as smart rooms. Moreover, detecting and classifying profile faces increases recognition rates by adding more temporal continuity as faces changing from frontal to profile orientations will not be lost neither discarded, but will greatly contribute in fusion of results by orientations.

The improvement obtained from fusion of results will depend on the number of faces detected in a given set of consecutive frames or time segment, their orientation and the fusion strategy chosen. These time segments may vary from one frame, providing instant fusion of results, to seconds. For each frame, it is possible to perform fusion of results if multiple versions of a same face, whether in a single or several orientations, can be detected in a multicamera environment. In time segments consisting of several frames, a fusion of instant fusion results for each frame can be performed in order to decide which individual was tracked and detected during the whole time segment. In case of not having instant fusion of results, a single frame-by-frame fusion of results is performed on the tested time segment.

A remarkable effort will be done in this work for successfully implementing a real-time face identification system, as a part of the "Proyecto Hesperia", funded by the Spanish Strategic Consortiums in Technical Research (CENIT in Spanish).

1.2 Structure of the dissertation

This work will be divided in three main parts:

- **Part I: State of the art**. As a review of some of the existing feature extraction and dimensionality reduction methods, Chapter 2 will introduce PCA, DCT, LDA and LPP methods. In Chapter 3, the problem of classification wil be introduced. Two types of classifiers will be introduced: kNN and Parzen. The k Nearest Neighbor (kNN) data classifier will be introduced and explained via examples. A Parzen density estimator used as a classifier is also presented. Chapter 4 explains the mechanisms to combine the results of multiple individual classifications into a more reliable final decision. Images with multiple face orientations will be used in this process, one of the main novelties of this project.

- **Part II: Setting up a face identification system**. Chapter 5 will describe in detail a feature-based face identification system and the stages and data resources needed, such as face databases and recordings. In Chapter 6 extensive simulations will be performed in order to obtain the optimal parameters that must be set in a face identification system. Experimental results will be given in Sections 6.2 to 6.10, as well as a conclusion.

- **Part III: Implementation of a real-time face identification system**. An implementation of a real-time face identification system is detailed in Chapter 7, including output images, computational reduction techniques applied and recognition rates obtained.

- Conclusions on this work will be commented in Chapter 8, as well as future work.

Part I

State of the art

Chapter 2

Feature extraction and dimensionality reduction

2.1 Introduction

Feature extraction is a process that consists on finding the set of quantitative characteristics that better define the input data in order to accomplish a goal, in this case, the classification of the faces into a set of predefined classes. In appearance-based face recognition, input data consists on face images and the features are the pixel values. Thus, face images can be represented by data vectors in a NxM dimensional subspace, being N and M the width and height of the face image. For practical values of N and M this leads to a large number of features. Classification of such high-dimensional data may be a cumbersome task for real-time applications. Fortunately, not all features are equally important for classification purposes. Dimensionality reduction refers to the process of data simplification that allows keeping a small subset of features or coefficients that, while preserving the discriminative power, alleviate the computational burden of the classification process. Therefore, the importance of choosing a dimensionality reduction technique and the right number of kept coefficients is not trivial, since performance and results depend on it.

Many feature extraction and dimensionality reduction techniques can be found in the literature. Perhaps the most popular is Principal Component Analysis [17]. Other popular techniques are Linear Discriminant Analysis [36], Discrete Cosine Transform [14] and Locality Preserving Projections [5]. Since the study of feature extraction is not the main goal of this thesis, PCA and DCT will be used in the following. Due to the modular structure of the system, any other feature extraction technique that outperforms these two may be used in the future with just minor modifications. As for LDA and LPP, these techniques are currently being tested by the Image and Video Processing Group (Department of Signal Theory and Communications, ETSETB-UPC) and will be introduced but not implemented due to extension restrictions.

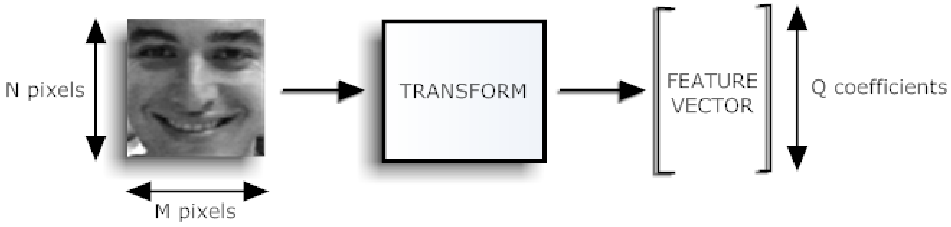

Figure 2.1: Main steps to obtain a vector of transformed coefficients from a face image.

All these techniques follow the same process:

1. Face images are resized, cropped or masked.

2. Images, treated as vectors or matrices, are projected using a transform matrix.

3. The first Q coefficients from the transformed images are kept in the so-called "feature vectors".

As an explanatory example on how dimensionality reduction works, Figure 2.2 shows the quality of a reconstructed image with PCA as the number of coefficients kept falls. Reconstructed images shown in Figure 2.2-d-e demonstrate that the less coefficients taken, the less face features are preserved. Note that when keeping as low as 10 coefficients the reconstructed image still preserves the most significant face features and could be used in face identification. This property of dimensionality reduction techniques will be very interesting when limiting the number of coefficients kept after a face image is transformed and reducing computational costs without recognition degradation.

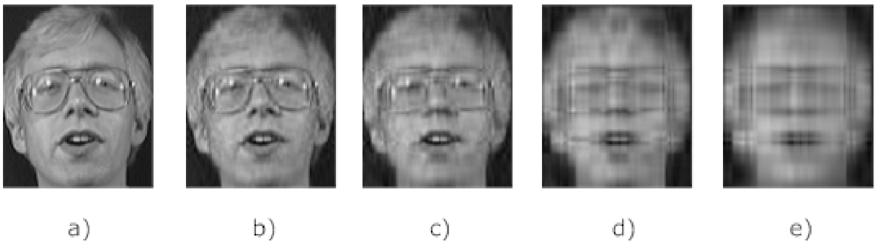

Figure 2.2: Example of dimensionality reduction. Number of coefficients: a) Original image, b) 20, c) 10, d) 5, e) 3.

LDA, PCA and LPP are data-dependent transformations. This means their projection matrices need to be computed from the data set to be transformed and any significant modification in the datased implies recalculating the matrix again. This is avoided in DCT since data is always transformed using an inmutable matrix as explained in Section 2.3. Given sufficient sample faces, LDA is superior to PCA, but in case of very limited number of samples, PCA can outperform LDA because LDA is sensitive to the training data set [19]. However, a difficulty in using the LPP method for image recognition is the very high-dimensional nature of the image space, that results in very big matrices that can also be singular [1].

Moreover, LDA and LPP techniques need a large number of training samples to compute an operational projection matrix and that is not always possible in face identification systems.

2.2 Principal Component Analysis

If a given set of images is represented by points in a multidimensional space, similar images are expected to fall in its neighborhood, forming a cluster in a delimited region. The shape or trend of the cluster can be easily determined using Principal Component Analysis (PCA) [17, 35]. This data-dependent technique is based on the Karhunen-Loève Transform (KLT), an orthogonal transform, which decorrelates every component contained in data and compacts energy distribution in few coefficients. This means stronger components characterising the data are identified with high energy levels while those weaker are superfluous and can be rejected, creating a sort of function described as $\mathbb{R}^N \to \mathbb{R}^d$, being d the number of coefficients kept. Note that rejecting weaker coefficients should not greatly alter the image properties nor its visual perception if parameter d is well chosen.

Let $\underline{\underline{X}}$ be a PxQ matrix containing Q images, one image in every row. Image pixels are scanned left-to-right or top-to-bottom so an image represented by an NxM matrix forms an $1x(NxM = P)$ vector. Let $\underline{\underline{C}}$ be a QxQ square matrix being the covariance matrix of $\underline{\underline{X}}$

$$\underline{\underline{C}} = \sum_{m=1}^{M} \left(\underline{X}_m - \underline{\mu}\right) \left(\underline{X}_m - \underline{\mu}\right)^T$$

and let $\underline{\mu}$ be the mean of all face samples

$$\mu\left[m\right] = \frac{1}{N} \sum_{n=1}^{N} X(i,j)$$

In order to determine the stronger components of the data set, a Singular Value Decomposition (SVD) is performed on $\underline{\underline{C}}$.

$$\underline{\underline{C}} = \underline{\underline{U}} \cdot \underline{\underline{\Sigma}} \cdot \underline{\underline{V}}^T$$

$\underline{\underline{\Sigma}}$ is defined as a positive diagonal QxQ matrix where elements in its diagonal are known as singular values or eigenvalues. Each eigenvalue has a related orthonormal vector in $\underline{\underline{U}}$, the so-called eigenvector. It can be shown that $\underline{\underline{U}}^T \underline{\underline{U}} = \underline{\underline{I}}$, meaning that PCA is a bijective or unitary transformation. This statement is not true when dimensionality reduction is applied, since a reconstructed version of the data is obtained when anti-transforming.

Once eigenvalues and eigenvectors (from now on referred as $\underline{\underline{W}}$ or projection matrix) are computed, data in $\underline{\underline{X}}$ can easily be transformed and anti-transformed as follows, respectively:

$$\underline{\underline{Y}} = \underline{\underline{W}}^T \underline{\underline{X}}$$

$$\underline{\underline{X}} = \underline{\underline{W}} \cdot \underline{\underline{Y}} = \underline{\underline{W W}}^T \underline{\underline{X}} = \underline{\underline{X}}$$

Note that when only d dimensions are kept, the anti-transformation definition varies slightly from the previous one.

$$\widetilde{\underline{\underline{X}}} = \widetilde{\underline{\underline{W}}} \cdot \underline{\underline{Y}} = \widetilde{\underline{\underline{W}}} \cdot \widetilde{\underline{\underline{W}}}^T \cdot \underline{\underline{X}} \approx \underline{\underline{X}}$$

where $\widetilde{\underline{\underline{W}}}$ denotes a matrix containing only the first d bottom eigenvectors in $\underline{\underline{W}}$. It is interesting to remark that PCA effectively decorrelates data since the correlation matrix of $\underline{\underline{Y}}$ is a diagonal matrix.

A projection matrix trained with a significant number of faces does not need to be recomputed when adding or deleting feature vectors since face characteristics are already well defined and changes do not greatly affect the results of a projection. Any projection matrix computed under these conditions can be successfully used for projecting faces from any other databases.

2.3 Discrete Cosine Transform

In order to reduce the computational cost of extracting face features [14, 6, 24], this dissertation will test a technique based on the Discrete Cosine Transform or DCT. This technique is a particular case of the Fourier Transform. There are several formal definitions of the DCT. The one known as DCT-II is widely used in image processing and will be used in this work.

Given an NxN greyscale image, the two-dimensional DCT-II transform is define as

$$X(i,j) = \frac{2}{N} k_i k_j \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} I(x,y) cos\left(\frac{\pi(2x+1)i}{2N}\right) cos\left(\frac{\pi(2y+1)j}{2N}\right)$$

$$k_i, k_j = \begin{cases} 1/\sqrt{2} & i,j=0 \\ 1 & otherwise \end{cases}$$

The resulting matrix contains NxN DCT coefficients which are related to one of the non-dependent NxN DCT basis. Figure 2.3 shows the $8x8$ DCT invariant bases for an $8x8$ image.

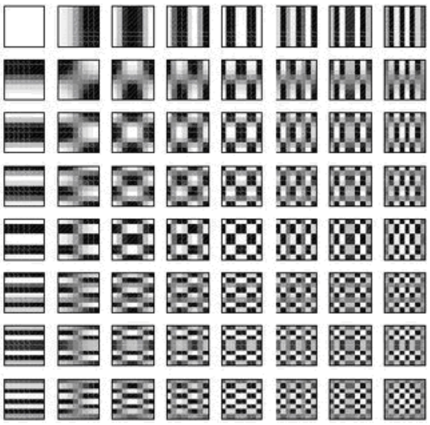

Figure 2.3: Invariant bases used for 2D DCT transformation.

As in PCA, the DCT transform has the property of compacting the energy of the image, that is, the most relevant coefficients for describing an image are located in the upper left of $\underline{\underline{X}}$. This phenomenon is shown in Figure 2.4.

An NxN image has a total of NxN transformed coefficients but a face identification system can be designed using only a small subset, as it will be explained in detail in Part III. This will tremendously reduce computational costs and memory usage in face identification, classification and face model creation stages. Face model files stored in disk will be much smaller, as well.

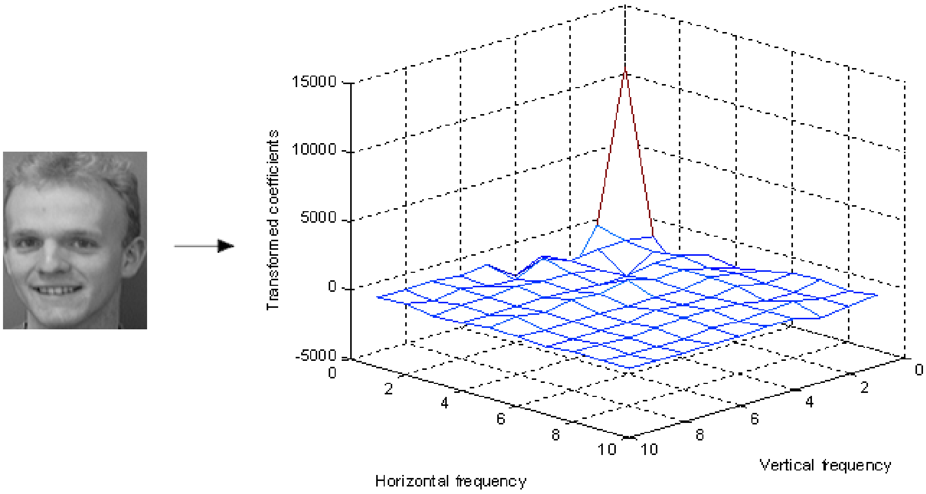

Figure 2.4: Example on how DCT compacts energy distribution around the first transformed coefficients.

Once DCT coefficients are computed, they are usually scanned in a zigzag order starting from $X(0,0)$, as shown in Figure 2.5.

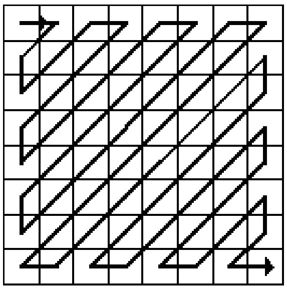

Figure 2.5: Typical zigzag scan order for an 8x8-block matrix as used in DCT.

2.4 Linear Discriminant Analysis

Linear Discriminant Analysis (based on Fisher's Discriminant Analysis, FDA[11]) is another dimensionality reduction technique in high-dimensional data [36, 15]. Two scatter matrices $\underline{\underline{S}}_B$ and $\underline{\underline{S}}_W$ are defined as between-class and within-class variance matrices, respectively. While between-class variance is a typical data set variance calculation, within-class variance determines the variance between each one of the existing classes represented in the whole data set. These matrices can be found as follow

$$\underline{\underline{S}}_B = \sum_{j=1}^{c} N_j \cdot (\underline{\mu}_j - \underline{\mu})(\underline{\mu}_j - \underline{\mu})^T$$

$$\underline{\underline{S}}_W = \sum_{i=1}^{c} \sum_{j=1}^{N_j} (\underline{x}_j^i - \underline{\mu}_j)(\underline{x}_j^i - \underline{\mu}_j)^T$$

where:

- c is the number of classes
- $N_j$is the number of data vectors in class j
- $\underline{\mu}$ is the mean of all data vectors
- $\underline{\mu}_j$ is the mean of the *j-th* class
- $\underline{x}_j^i$ is the *i-th* data vector of the *j-th* class

Unlike PCA, this technique maximises the between-class covariance to within-class covariance ratio in order to find an optimal projection that best discriminates data vectors

$$\underline{\underline{W}}_{opt} = \arg\max_{\underline{\underline{W}}} \frac{\underline{\underline{W}}^T \underline{\underline{S}}_B \underline{\underline{W}}}{\underline{\underline{W}}^T \underline{\underline{S}}_W \underline{\underline{W}}}$$

where matrix $\underline{\underline{W}}$ is the eigenvectors matrix of $\underline{\underline{S}}_W^{-1}\underline{\underline{S}}_B$.

2.5 Locality Preserving Projection

LPP [5] is introduced as another linear dimensionality reduction technique. LPP is used to build a transformation matrix that maps vectors to a subspace. The maps or affinity matrix $\underline{\underline{A}}$ can be built as

$$\underline{\underline{A}}_{i,j} = \exp\left(-\frac{\|x_i - x_j\|^2}{\sigma^2}\right)$$

Another possibility to construct the affinity matrix is using kNN. If x_i is a k-neaterst neighbor of x_j or x_j is a k-neaterst neighbor of x_i, the (i, j) element of the affinity matrix $\underline{\underline{A}}_{i,j}$ is set to 1. Otherwise, it is set to 0.

Finally, Laplacian eigenmaps are the optimal solution of the following expression

$$\underline{\underline{W}}_{opt} = \arg\min_{\underline{\underline{W}}} \sum_{i=1}^{N}\sum_{j=1}^{N} \left\|\underline{\underline{W}}^T \underline{x}_i - \underline{\underline{W}}^T \underline{x}_j\right\|^2 \underline{\underline{A}}_{i,j}$$

Chapter 3

Classification

3.1 Introduction

For a given set of data samples belonging to different data classes, each one representing a particular combination of usual features found in data, a classification technique should be able to successfully assign a class to each data sample. The process of assigning samples to classes is known as classification, and it is performed by minimising the distance in a *d-dimensional* space between the sample and every sample in the set of data classes. Different classifiers may offer different performances and error rates, as will be described in Chapter 6.

This chapter will introduce two classifiers: k Nearest Neighbor, including a modification, and Parzen. Due to the possibility of choosing the most favourable distance metric when classifying, Section 3.4 will describe the most common metrics.

3.2 k Nearest Neighbor

The main idea of k Nearest Neighbor or kNN [16, 9] classifier is based on a simple concept. Let $\underline{P}$ be a feature vector of length N and let $\underline{\underline{Q}}^i = \left\{\underline{q}_0^i, \ldots, \underline{q}_{N-1}^i\right\}$ be a matrix with N_i training vectors of the i-th model on each row, distances between $\underline{P}$ and the available feature vectors for every one of the N_q models are calculated and stored in a distance vector $\underline{\underline{D}}$.

The k smallest distances in $\underline{\underline{D}}$ are selected, that is the nearest feature vectors to $\underline{P}$.

Every selected neighbor will correspond to a face model or class and its identity will be registered as a *vote*. The winning class will be decided simply by taking the most voted rule.

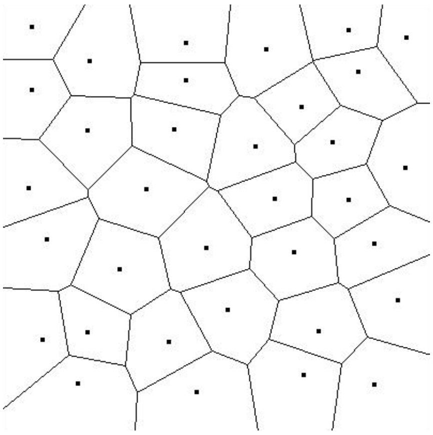

Figure 3.1: Voronoi tessellation of a given distribution of sample vectors.

Given a set of points or vectors in a hyperplane, kNN classificates the regions around them creating a Voronoi tessellation as in Figure 3.1.

kNN shows to be robust enough in two critical situations, despite its simplicity. Let us assume that a test vector belonging to a given class B must be classified and that it is far enough from the other existing class, named B, as shown in Figure 3.2.

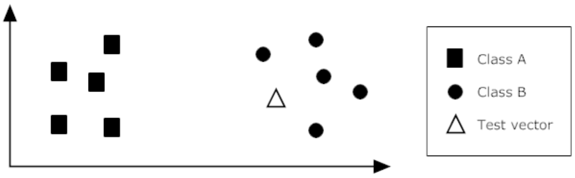

Figure 3.2: First example on kNN. Hypothetical scenario for two distant classes and a test vector close to class B.

Step by step results of the classification process and numerical values are shown in Figure 3.3 and Tables 3.1-3.2, respectively, where *Accumul.* stands for *Accumulated votes.*

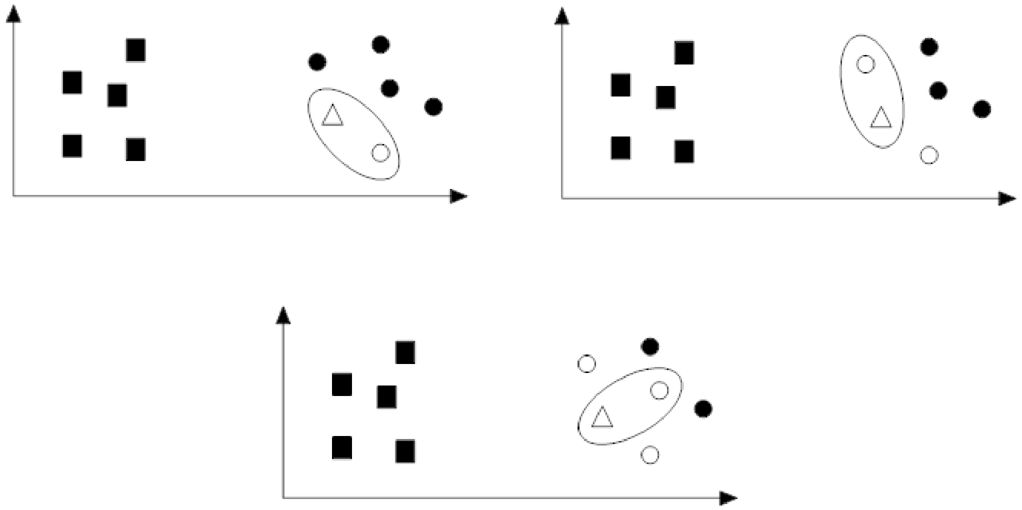

Figure 3.3: First example on kNN. Selected neighbors. Top-to-bottom, left-to-right: k=1, k=2, k=3.

		Class A		**Class B**			
k-th NN	**Decision**	**Winner**	**Accumul.**	**Winner**	**Accumul**	**Match**	**Partial**
1	B	No	0	Yes	1	Yes	B
2	B	No	0	Yes	2	Yes	B
3	B	No	0	Yes	3	Yes	B

Table 3.1: First example. Instant classification results and votes given for each class.

Class A	**Class B**	
Accumul.	**Accumul.**	**Winner class**
0	3	B

Table 3.2: First example. Final decision by majority voting.

As k increases, the threshold distance to a near neighbor also increases or remains the same as in the previous k. In this scenario, class B will continue to be selected up to $k = 5$ due to the proximity of the test vector to the class B cluster. Note that even choosing $k = 6$ as a working parameter for kNN the classified class will be class B, winning by 5 votes to 1.

Let us assume now that a second test vector belonging to a new class B must be classified and that class A is very close to class B. In this particular case an *a priori* guess of the right identity of the test vector can be dangerous as none of the classes fit best the test vector.

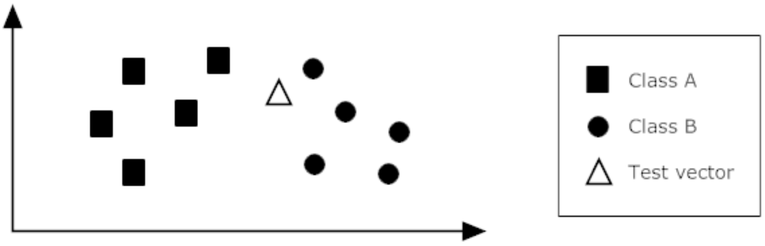

Figure 3.4: Second example on kNN. Hypothetical scenario for two close classes. A test vector is situated amid the two classes but closer to B.

Step by step results of the classification process and numerical values are shown in Figure 3.5 and Tables 3.3-3.4, respectively.

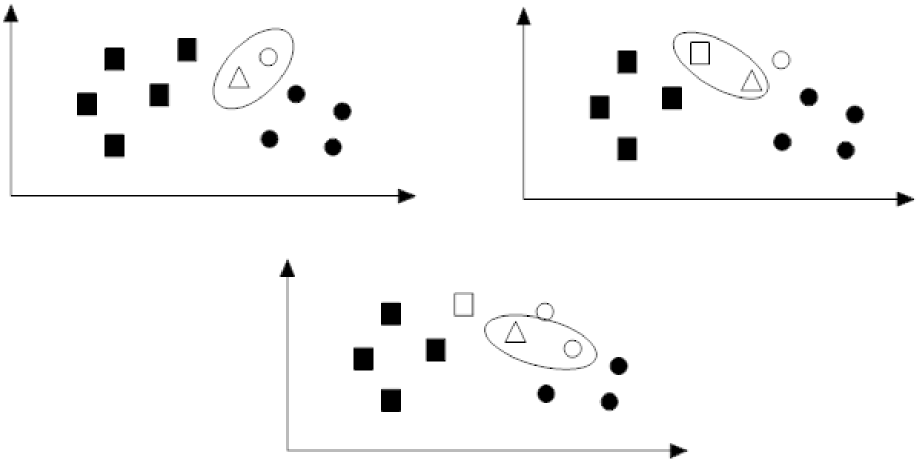

Figure 3.5: Second example on kNN. Selected neighbors. Top-to-bottom, left-to-right: k=1, k=2, k=3.

		Class A		**Class B**			
k-th NN	**Decision**	**Winner**	**Accumul.**	**Winner**	**Accumul.**	**Match**	**Partial**
1	B	No	0	Yes	1	Yes	B
2	A	Yes	1	No	1	No	A/B
3	B	Yes	1	No	2	Yes	B

Table 3.3: First example. Instant classification results and votes given for each class.

Class A	Class B	
Accumul.	**Accumul.**	**Winner class**
1	2	B

Table 3.4: First example. Final decision by majority voting.

The fact of a test vector not being situated in a space region under the clear influence of any of the existing classes makes more difficult its classification. In $k = 2$ (see Figure 3.5) each of the two classes has been selected once and the decision is randomly taken. This kind of situations were random decisions needs to be taken should be avoided by selecting only odd values of k.

In both situations the feature vector has been correctly classified but when having overlapped regions it is risky to classify using majority voting.

3.3 k Nearest Neighbor with probabilities

In both examples in Section 3.3 class B has been correctly decided although using kNN as is could imply risks. Despite kNN can be thought to be a good option for classifying data because of its simplicity, the majority rule voting may seem unfair as all votes score the same regardless of the distance. In this dissertation kNN is modified so that a vote is penalised by the inverse of the distance between the test vector $\underline{P}$ and the selected neighbor, in such a way that the bigger the distance, the more penalisation. The k_i inverse distances in $d_{i,\underline{Q}_i}$ of a same class Q^i are also normalised by the overall sum of the k nearest neighbor inverse distances of all Q classes leading to the *a posteriori* probability for each class [9]:

$$p(\underline{P}|Q_i) = \frac{\displaystyle\sum_{i=1}^{k_i} \frac{1}{d_{i,\underline{Q}_i}}}{\displaystyle\sum_{j}^{k} \frac{1}{d_{j,\underline{Q}}}}$$

where

$$k = \sum_i k_i$$

This modification on kNN is named kNN with probabilities. These probabilities for each class allow the classifier to give an estimation on how confident the classification of the test vector is. If for a given class the classifier outputs a probability near to 1 one can guess that the other existing classes were far enough. On the contrary, a probability of $1/k$ would show that neighbors are equidistant to $\underline{P}$.

In kNN classifiers with probabilities a minimum probability threshold can be set to accept or reject a classification result. This threshold should be empirically adjusted depending on the quality of the images or the number of coefficients kept in transformations.

3.4 Distance metrics

When classifying vectors a distance metric needs to be chosen. The election of the proper metric should depend on the computational costs and the recognition rates achieved. This trade-off will be discussed in Chapter 6 and the metric offering higher recognition rates will be used as the default metric in this work. Next, a brief introduction to the metrics taken into account.

Minkowski distance

Minkowski distance is a distance generalisation in a p-dimensional Euclidean space. For two p-dimensional vectors, the distance between them is obtained as follows

$$d(\underline{x}, \underline{y}) = \left(\sum_{i=1}^{n} |x_i - y_i|^p \right)^{1/p}$$

Let $p \to \infty$, this case is known as maximum norm or Chebyshev distance, which takes the maximum distance of all coordinates as the shortest way from $\underline{x}$ to $\underline{y}$.

$$d(\underline{x}, \underline{y}) = \lim_{p \to \infty} \left(\sum_{i=1}^{n} |x_i - y_i|^p \right)^{1/p} = \max_i |x_i - y_i|$$

Note that the larger p is, the smaller the distance due to the inverse power of p. A proper value of p has to be selected in feature vector classifications.

Euclidean distance

As a particular case of the Minkowski distance, Euclidean distance is defined for Euclidean spaces based on Pythagoras' theorem.

$$d(\underline{x}, \underline{y}) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

This distance is widely used in 2D metrics but can also be highly useful in n-dimensional vectors such as feature vectors due to its low computational costs. This metric is a candidate to be chosen as a metric in face classifiers and will be tested in detail.

Manhattan distance

Another particular case of the Minkowski distance is defined in this section. Manhattan distance uses $p = 1$ for giving distances.

$$d(\underline{x}, \underline{y}) = \sum_{i=1}^{n} |x_i - y_i|$$

This name is given by a simple geometrical interpretation: the distance between two points in a Manhattan-like structure is the minimum number of blocks that a person should cross.

This metric is even simpler than Euclidean distance and will also be tested as a candidate metric.

Mahalanobis distance

Given two n-dimensional vectors, being $\underline{x}$ the test vector and $\underline{y}$ the vector to compare to, $\underline{\underline{S}}_y$ the covariance matrix of $\underline{y}$ and $\underline{\mu}$ the mean vector of $\underline{x}$, Mahalanobis distance is defined as

$$d(\underline{x}, \underline{y}) = \sqrt{\left(\underline{x} - \underline{\mu}\right)^T \underline{\underline{S}}_y^{-1} \left(\underline{x} - \underline{\mu}\right)}$$

In high-dimensional data calculations, this metric may require highly demanding. Computing the inverse of a high-dimensional covariance matrix as in $\underline{\underline{S}}_y^{-1}$ increases the computational complexity greatly. Due to this complexity this metric is discarded in kNN feature classification.

Cosine similarity

Given the previous n-dimensional vectors, cosine similarity gives an idea of how similar are the two vectors as a function of the cosine.

$$d(\underline{x}, \underline{y}) = \cos(\underline{x}, \underline{y}) = \frac{\underline{x}^T \underline{y}}{\|\underline{x}\| \, \|\underline{y}\|}$$

The possible results are:

- $-1 < \cos(\underline{x}, \underline{y}) < 0$ for opposite vectors
- zero for orthogonal vectors
- $0 < \cos(\underline{x}, \underline{y}) < 1$ for similar vectors

3.5 Parzen classifier

Emanuel Parzen [25] published in 1962 a kernel density estimator for d-dimensional sample vectors delimited by a hypercube of side lengths $h(n)$ and volume $V_n = \prod_n h(n)$. For simplicity, all sides are defined equal in length

$$V_n = h_n^d$$

where h_n is known as the Parzen window width or bandwidth.

Let x be a sample vector and $X = \{x_1, x_2, ..., x_n\}$ be a group of n sample vectors described by an unknown probability density function, the number of vectors k_n in X that are contained inside the d-dimensional hypercube centered in x is

$$k_n = \sum_{k=1}^{n} K\left(\frac{x - X_k}{h_n}\right)$$

where $K(\cdot)$ is known as Parzen window function or kernel function. It is zero-mean, unit-variance defined and satisfies $\int_{\mathbb{R}^d} K(y)dy = 1$ and $\int_{\mathbb{R}^d} K^2(y)dy < \infty$. The final expression of the probability density function $f_x(x)$ is

$$f_n(x) = \frac{k_n}{N \cdot V_n} = \frac{1}{N \cdot h_n^d} \sum_{k=1}^{n} K\left(\frac{x - X_k}{h_n}\right)$$

In case that $h_n \to 0$ the kernel function approximates to a Dirac delta function. On the contrary, as h_n increases $f_n(x)$ gets smoother. In Figure 3.6 it is shown how the density distribution of a given set of samples changes in function of h and n.

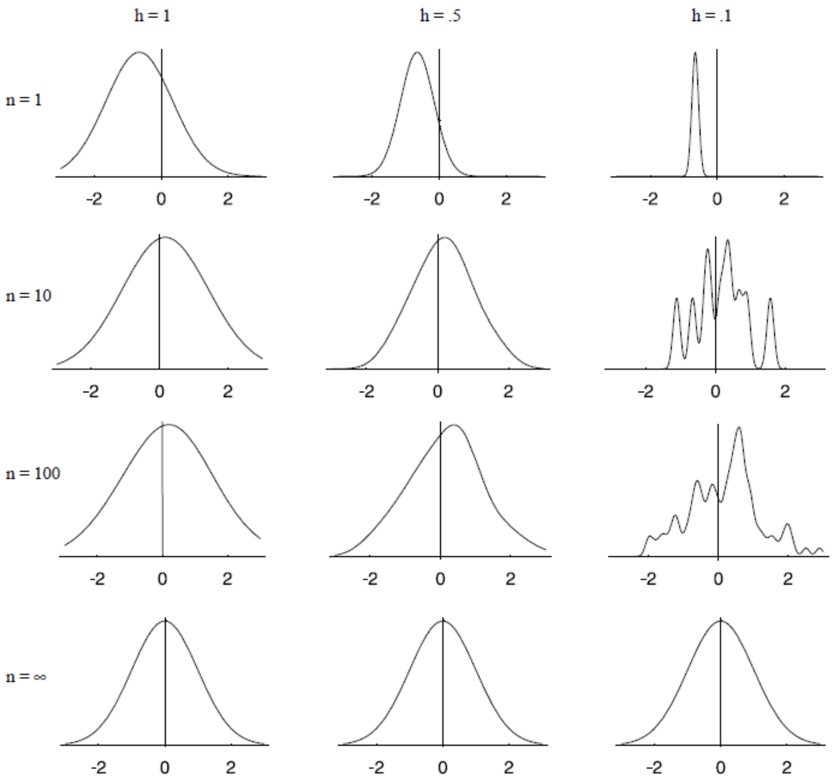

Figure 3.6: Evolution of a density distribution in function of parameters n and h when using Parzen estimation [9].

In this dissertation Gaussian kernel functions will be used for Parzen density estimation [33]. This function computes distances by Mahalanobis distance as shown next:

$$K\left(\frac{x-X_k}{h_n}\right)=\frac{1}{(2\pi)^{n/2}\,|C_X|^{1/2}}\exp\left(-\frac{(x-X_k)^T C_X^{-1}(x-X_k)}{2h_n^2}\right)$$

where C_X is the variance matrix of the X.

Prior to this work, studies and experiments done by the author showed that recognition rates using Parzen density estimator as a classifier were very poor compared to kNN. Parzen also offered much higher computational burdens due to Mahalanobis distance calculations of high-dimensional data such as images. Moreover, an optimal and general working point could not be determined since window width was highly data-dependant and suggestions in the literature were contradictory. Thus, Parzen classifier will not be tested as significant modifications are needed and it is not the main purpose of this work.

Chapter 4

Fusion of results

4.1 Introduction

As it has been previously stated, a face identification system may detect frontal, semi-frontal or profile faces. Video sequences usually contain several images every second and, because of the nature of a still scenario, faces tend to change very slowly between frames. Fusion of results will take advantage of detecting several consecutive faces of a same individual in a time segment with no or slow changes in orientation. This face redundancy means lower chances to fail when classifying. Moreover, the process of a face changing its orientation from frontal to profile or viceversa can last for enough frames so the system is able to track it and merge the results.

Detected faces will be weighted by their own normalised orientation recognition rates, which have to be previously computed, and the confidence given by the face detector. Recognition rates for each orientation may differ one from another and assigning an equal weighting would not improve results but worsen them. Avoiding such a trivial solution requires accurate system training using well-known, labeled video sequences. For a correct training process, an identification of every individual appearing in the scene must be provided to the process so it could compare the classification results with the true identification.

Fusion of results is thought to be performed in the next situations:

- When collecting several faces from a same camera, giving results after a time interval.
- When collecting faces from several cameras in different face orientations, giving

fusion of results on every frame.

- A combination of the two above, in a multi-camera environment detecting faces in different orientations and giving final fusion of results after each time interval.

Note that the more frames collected, the better the results should be. On the contrary, too long time intervals reduce the possibility of giving real time results.

4.2 Fusion strategies

Once trainings are performed, an approximate recognition rate for every orientation is ready to be used in fusion of results. In [18] a fusion strategy named Matcher Weighting is proposed and used in biometrics for multimodal fusion of monomodal scorings. Based on this idea, identification error rates will be used as penalisation factors to create a factor w^m which is the weighting coefficient for the m-th monomodal scoring, x^m.

$$w^m = \frac{\frac{1}{e^m}}{\sum_{m=1}^{M} \frac{1}{e^m}}$$

The final fusion of results is defined as any of the existing weighted means. The following expressions shows the most common ones.

$$u = \sum_{m=1}^{M} x_m w_m$$

$$u = \left(\prod_{m=1}^{M} x_m^{w_m} \right)^{1/\sum\limits_{m=1}^{M} w_m}$$

$$u = \frac{\sum\limits_{m=1}^{M} w_m}{\sum\limits_{m=1}^{M} \frac{w_m}{x_m}}$$

The expressions above are for weighted arithmetic, geometric and harmonic means, respectively.

Another fusion strategy is the one based on a kNN classifier with probabilities which is introduced in Section 3.3. For every possible model in a scenario this classifier outputs an identity guess of the classified face and a sort of probability of being the true identity.

In normal conditions, a focused and clear face should obtain a high probability for its true identity and small or null probabilities for the rest. Moreover, this fusion strategy takes into account that recognition rates can be different depending on the orientation of a face and that when using face detectors a confidence parameter can be given.

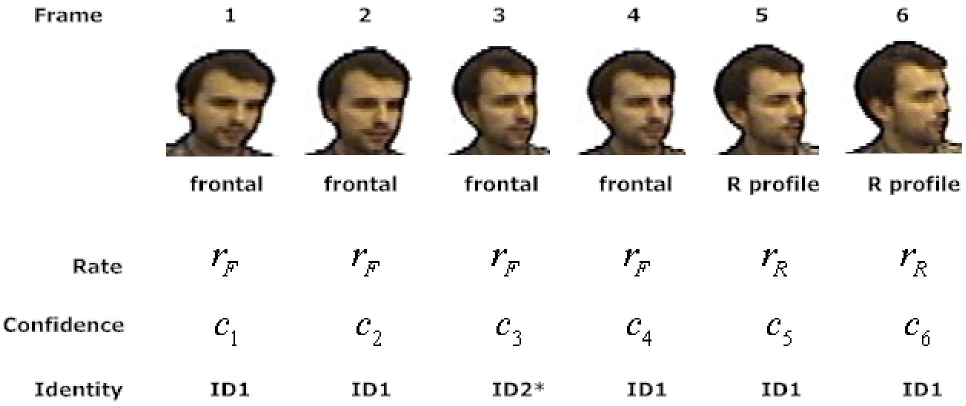

Figure 4.1: Example of fusion of results for a sequence including frontal and right profile faces.

As an example, Figure 4.1 shows a hypothetical situation where an individual is detected during 6 frames. The first four frames are considered to be frontal while the latter two are right profiles. r_F and r_R are frontal and right profile recognition rates and c_k is the confidence given by the face detector for the k-th frame. Frame 3 is wrongly classified as *'identity 2'* and the other frames are classified as *'identity 1'*.

Given three possible identities in this example, the fusion of results for the i-th identity would be as follows

$$v_i = \sum_{n=1}^{N_{frames}} c_n p_{i,n} r$$

being

$$r = \begin{cases} r_F & \text{frontal} \\ r_L & \text{left profile} \\ r_R & \text{right profile} \end{cases}$$

The final results for the time interval in Figure 4.1 are

$$v_1 = c_1 p_{1,1} r_F + c_2 p_{1,2} r_F + c_4 p_{1,4} r_F + c_5 p_{1,5} r_R + c_6 p_{1,6} r_R$$

$$v_2 = c_3 p_{2,3} r_F$$

$$v_3 = 0$$

Note that by using this fusion strategy it is possible to merge different types of results as long as they are normalised. For further details, several simulations are described in Section 6.8. Other data fusion strategies widely used in multimodal identification such as the Minimax rule [7] can be found in the literature but will not be investigated in this work due to extension restrictions.

Part II

Setting up a face identification system

Chapter 5

Appearance-based face identification systems

5.1 A functional scheme

A face identification system can be easily described using a functional scheme made up of modules. This simplification is useful in the design and testing stages, as well as in making the reader to better understand how the system works.

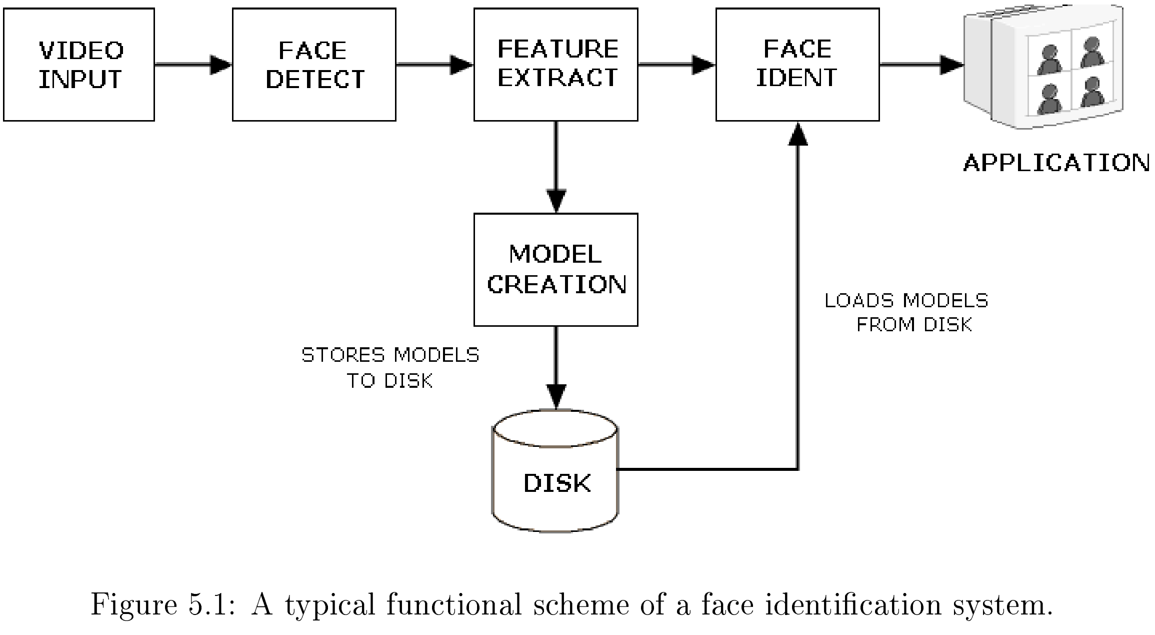

Figure 5.1: A typical functional scheme of a face identification system.

Figure 5.1 shows a possible scheme which counts with five main stages, which are:

- **Video input**: Faces can be found in audiovisual data such as video streams or separated frames stored in a disk unit or directly acquired from video cameras.

Ideally, cameras should be placed where both frontal and profile faces could be successfully obtained with an acceptable image quality.

- **Face detection**: Once images are acquired, faces can be detected by hand or using image processing techniques. Both techniques imply severe trade-offs. By hand face segmentation and labeling is a tedious work and can take very long to complete but, on the other hand, human perception of a face is almost always correct and no mistakes are expected to happen. Image processing techniques such as Viola-Jones[34] are extremely fast and achieve high detection rates but a certain number of false positive images will be inevitably given along with positives.

- **Feature extraction**: Face images need to be dimensionally reduced since, as explained in Chapter 2, only some of the transformed coefficients need to be kept to successfully characterise a human face. Depending on the dimensionality reduction technique used, configuration parameters and other restrictions, recognition rates may vary. Regions of interest of the acquired images may be reduced in order to avoid unnecessary use of the system.

- **Model creation**: This module is an interface between the application and the computer, so data vectors and models can be created, modified, updated, deleted, or loaded. Each one of these model files represents a single individual in multiple poses and face orientations, so the face identification module can look for the closest data vectors when a new face is detected. Note that the proposed system is a closed system with a predefined number of individuals and no new models will be created or deleted during its execution.

- **Face identification**: This module will accept several configuration parameters that need to be tested by performing simulations in order to best adapt the identification system to its environment. The individual's identification number and likelihood of the classification for a given feature vector will outputted after time-based or orienation-based fusion. Results can also be given instantly, that is, without fusion of results.

5.2 Training stage

First of all, the detection and recognition system must be trained in order to create face models out of individuals, who will be recorded in a normal situation. The presence of frontal and profile faces can be ensured by asking individuals to look towards the camera and to both sides.

Faces are automatically detected, segmented and transformed into dimensionally reduced data vectors, which are added to a face model file. Moreover, training sets for each individual are given an identity and a name by hand. Face models accept feature vectors and store them based on their face orientations.

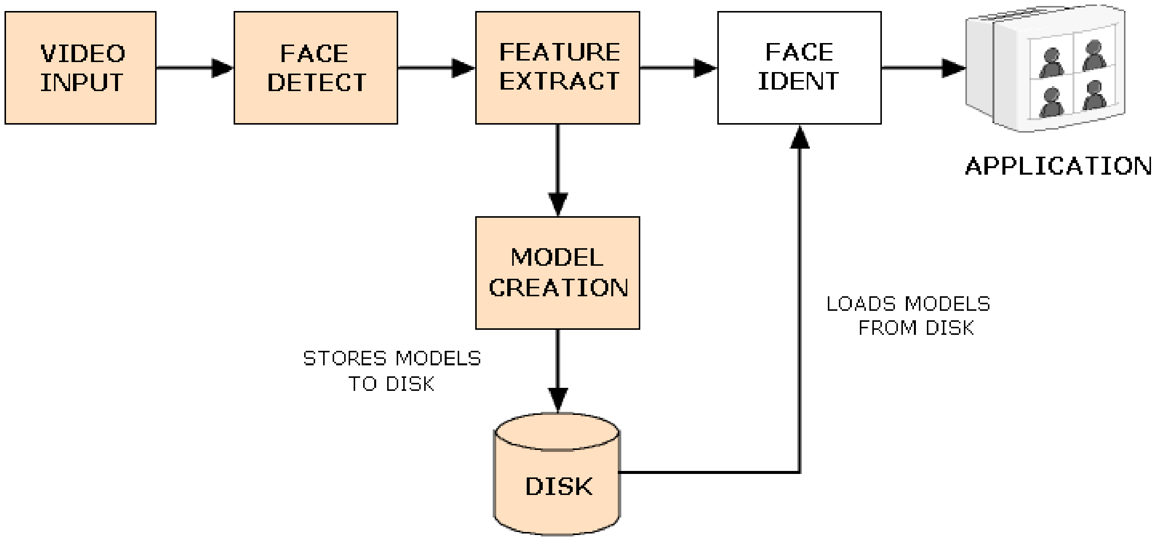

Figure 5.2: A functional scheme of the training stage in a face identification system.

5.3 Testing stage

After creating face models for every individual in the closed set of identities the system is ready to start identifying individuals. With no *a priori* information given, it must be able to detect and segment faces so they can be classified by comparing their transformed coefficients with those stored in the existing face models. When the classifier finds a feature vector in a face model that minimises distance to the tested feature vector, the identity of the chosen face model is displayed.

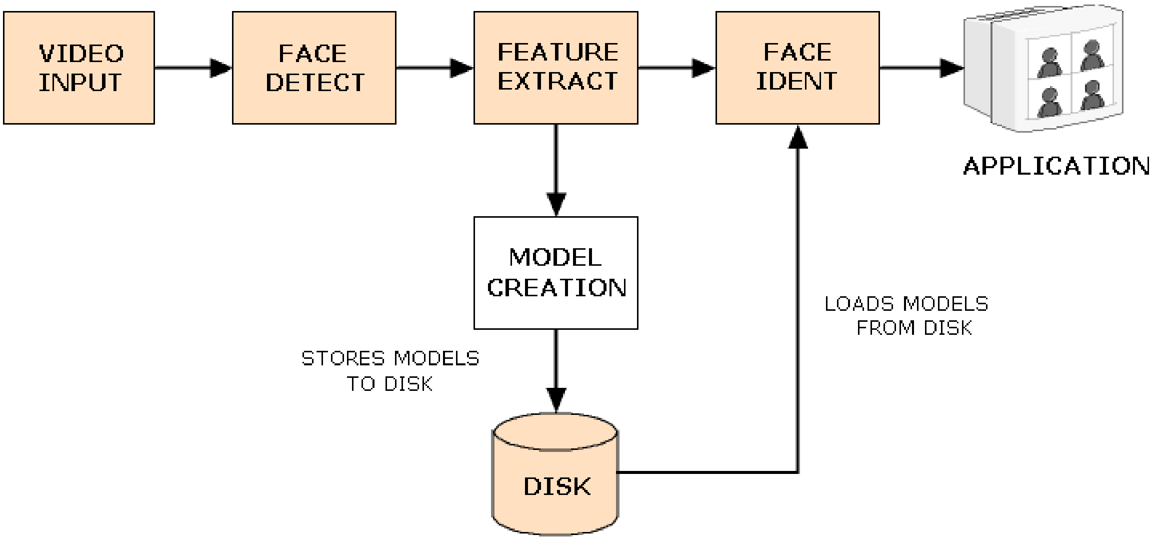

Figure 5.3: A functional scheme of the testing stage in a face identification system.

5.4 Face databases

In a face identification system there can be several parameters to be adjusted. A first step towards a real-time face identification system is to use databases providing well-known, high resolution faces. The number of coefficients kept, minimum image sizes, and other relevant parameters will be set up using an ideal working scenario and then results will be validated in more adverse conditions.The final implemented system will surely underperform in adverse conditions and it is critical to previously find a working point showing the most consistent and robust results.

In this section, five databases and their main characteristics are introduced. ORL and the Extended Yale Face B databases can be considered as best case scenarios, while the other three describe a more realistic and challenging situation.

ORL

This database [3, 28], also known as AT&T, was recorded between 1992 and 1994 for research purposes in face recognition and biometrics. The Cambridge University Engineering Department took 10 different frontal images of 40 individuals in a black background which included lighting variations, several facial expressions and facial details such as glasses. The size of images is 92-112 pixels and 8-bit greyscale.

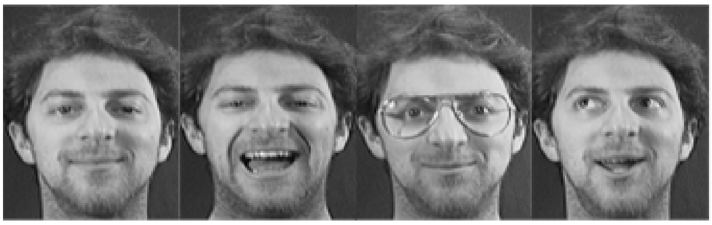

Figure 5.4: Example faces from the ORL face database.

Yale

The Yale Face Database [32] was created in Yale University. The database provides 11 frontal GIF images from 15 individuals in several facial expressions and lighting variations. The size of images is 320x243 pixels, 8-bit greyscale.

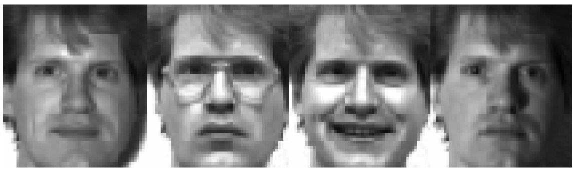

Figure 5.5: Example faces from the Yale face database.

The Extended Yale Face Database B (cropped version)

This database [13, 31] is the extended version of Yale Face Database B and contains 16128 manually cropped faces of 28 individuals. Each individual undergo a set of 9 different poses and 64 illumination variations. Faces are 8-bit, greyscale, PGM raw format and sized 168x192. A Sony XC-75 camera was used to acquire the images.

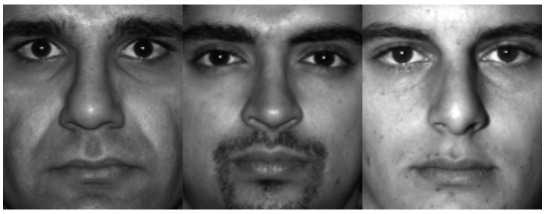

Figure 5.6: Example faces from the ExtendedYale Face Database in its cropped version.

CLEAR Evaluation recordings

The Spring 2007 CLEAR evaluation and workshop [30, 22] is an international effort to evaluate systems that are designed to recognise events, activities, and their relationships in interaction scenarios. The CLEAR 2007 evaluation is supported by the European Integrated project CHIL and the US National Institute of Standards and Technology (NIST). In CHIL (Computers in the Human Interaction Loop) project, meetings in smart rooms were recorded by institutions such as UKA, IBM, UPC, ITC or AIT. The aim of the project was to create an environment where computers would serve humans in an interaction loop.

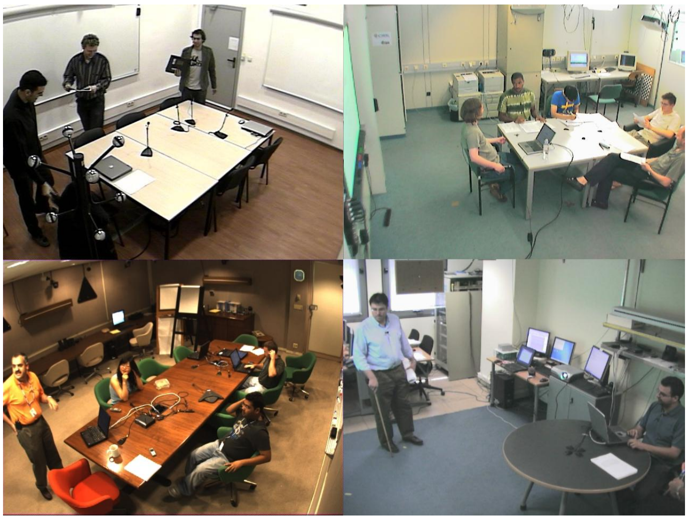

Figure 5.7: Smart rooms of four participating institutions in CHIL project. Ordered top-to-bottom, left-to-right: UPC, UKA, IBM and AIT.

This database contains 20-minute recordings using four cameras plus one overhead camera in different scenarios. Sequences were recorded at 25 fps. Images are RGB, 28-bit, JPEG format with a size of 640x480 pixels. A total of 28 individuals are included and hand-labeled in this database. A significant proportion of hand labelings are defective and this may be considered as a worst case scenario face database.

HESPERIA recordings

This recording was performed in 2009 within the framework of the HESPERIA project [2]. A total of 12 individuals were recorded while performing a model training sequence for the implementation of a real-time face identification system as explained in Chapter 7. Each individual had to stand and speak to the front, look to the right, left and front, and finally leave the smart room passing by the camera. For testing purposes, every individual had to enter and cross the smart room. In addition, individuals were asked to enter in pairs for extra testing.

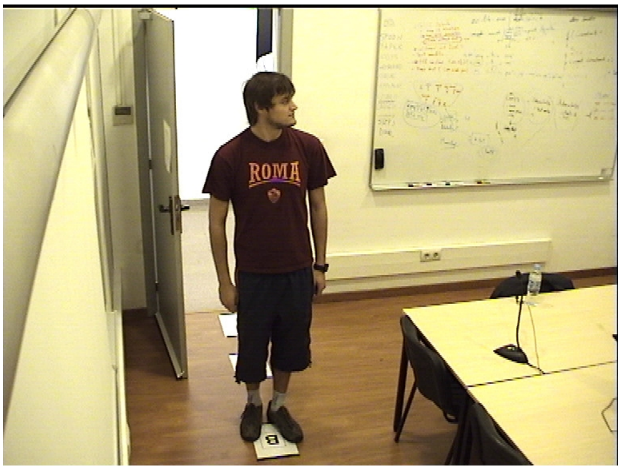

Figure 5.8: An individual performing a recording for the HESPERIA project. This image was used for model training purposes.

Sequences were recorded at 25 fps. Images are 24-bit RGB, JPEG format with a size of 768 x 576 pixels.

Chapter 6

Simulations and results

6.1 Introduction

This work requires simulations to evaluate the parameters needed to implement a face identification system and to test their performance under different scenarios and conditions. These parameters are supposed to be independent and joint parameter simulations will not be performed for simplicity. In this chapter, simulations achieving conclusive results will be explained in detail so that the reader can evaluate the performance of the system, as well as the possible applications.

Simulations will be put forward according to the work time line and a definite structure as follows:

- Introduction and data used
- Experimental results
- Conclusions on simulations, performance, or any other remarkable issues

Simulations will be initially coded in Matlab and then implemented in C++ using ImagePlus (proprietary image processing libraries developed by the Department of Signal Theory and Communications, ETSETB-UPC), OpenCV [23] and Boost [4] libraries.

6.2 Size of images and recognition rates

None of the reviewed papers suggested how the resolution of face images is related to recognition rates. This experiment will find if there is any relation between size and recognition rates and will discuss the opportunities it may offer.

High resolution of face images can contain basic face characteristics such as shape, eyes and nose distribution, and other important factors to face identification. Besides, these high resolution images may also contain weaker features adding noise which, if not removed, could worsen the performance of a classification system.

As images are reduced in size, only the strongest principal face characteristics are kept and appearance-based face identification is still possible while visual recognition may not. Principal face characteristics in small images can be as specific as in bigger images, with the exception of clipping some of the possible negative effects of lower energy components.

Computational complexity and execution times are an important trade-off in massive image collection such as face identification in smart rooms or controlled environments. Oversized images can lead an identification system to underperform in real-time applications. The time spent in training and testing stages will be crucial and decisive for image size selection, specially if using PCA transformation.

Experimental results

Since this experiment requires images to be enlarged or reduced from their original size, ORL database will be used so that 112x92-pixel images may be reduced to very small sizes. Resized images will be transformed, trained and classified so that recognition rates can be compared and used as another selection parameter, along with time spent in training and the size of the projection matrix in memory. No dimensionality reduction will be performed during this experiment.

Data used	
Database	ORL
Image size	112x92 pixels
Number of models	40
Training set	5 faces per model
Testing set	5 faces per model

Table 6.1: Details for this simulation.

For a given set of N models which are trained with M images each, being PxQ their size, the correlation matrix will be NxM in size and $(NxM)\,x\,(PxQ)$ for the transformation basis.

			Max. recog. rate (%)	
Image size (pixels)	**PCA basis size**	**Basis processing time**	**PCA**	**DCT**
112x92 (original)	8Mb	1.9e+07	98.33	98.13
60x60	2.8M (-65%)	3.7e+06 (-81%)	93.75	98.75
32x32	805k (-90%)	1.1e+06 (-94%)	93.75	98.13
16x16	202k (-97%)	3.2e+05 (-98%)	92.50	98.13
8x8	61.2k (-99.23)	8e+04 (-99.6%)	80.63	88.13

Table 6.2: PCA and DCT maximum recognition rates are compared when resizing ORL faces. The size and processing time of each PCA basis are also included along with the reduction compared to the original data set.

The smaller the image is, the lowest the maximum recognition rate within the whole range from 1 to 80 transformed coefficients. On the other hand, even 8x8-pixel images give acceptable results when comparing to bigger images, as well as significantly lower basis size and basis processing time. Note that processing times are given in CPU clocks, being 10e+6 clocks about one second on a 3 GHz CPU.

Conclusions

Image reduction does not strongly affects performance in a PCA or DCT identification system but remarkably decreases execution times and memory usage, which can be fundamental improvements in real-time applications. Following these criteria, image sizes between 16x16 and 32x32 should provide balanced performances. 32x32-pixel face images will be used in the following simulations and DCT will be selected as the best feature extraction technique since it is less size-dependent and proves to be faster than PCA.

6.3 Blurry faces in identification systems

Images gathered from cameras usually are not well focused or are taken while the lens is adapting to a new focal distance or illumination condition. This may cause faces to be blurry and unrecognizable to the human eye. However, this undesirable phenomenon and its effects on face identification have been barely documented. This experiment aims to check whether blurry images of individuals lower identification rates or not.

Experimental results

This simulation will embrace three minor tests for a later comparison. Since ORL face database offers focused faces, a modified version original ORL face database is created applying a uniform smoothing filter of dimension $N = 3$ (see figure 6.1):

$$h_{smooth} = \frac{1}{3^2} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

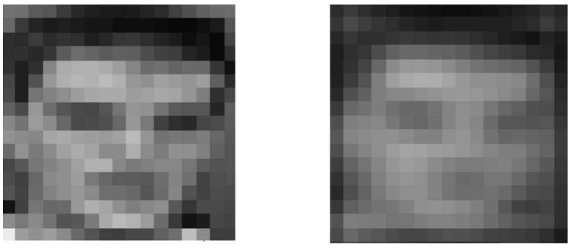

Figure 6.1: Left: focused ORL 16-16 pixel face; right: filtered or blurred ORL face.

- Test A: Both testing and training sets use clear faces.
- Test B: Blurry faces classified using using a clear training set.
- Test C: Blurry faces classified using a blurry training set.

The whole ORL database is used in this experiment (see Table 6.3).

Data used for Tests A-B-C	
Database	ORL
Image size	16x16 pixels
Number of models	40
Training set	6 faces per model

Table 6.3: Details for this simulation.

The three simulations (A, B and C) will be run using kNN ($k = 1$) and DCT transformation.

	kNN		fusion-kNN	
	Recog. rate (%)	**Min. coeff**	**Recog. rate (%)**	**Min. coeff**
Test A	95.63	53	100	7
Test B	94.37	31	100	7
Test C	96.87	19	100	11

Table 6.4: Recognition rate and minimum number of coefficients to achieve it for the three introduced tests. kNN is used with and without fusion of results. *Min. coeff* is referred to the minimum number of coefficients kept to obtain maximum recognition rates.

Conclusions

Given the Table 6.4, the result of Test C is remarkable. The recognition rate is higher than in the other tests and the minimum number of coefficients needed is much lower, too. This might be caused by the dimensionality reduction when blurring images, that is to say, filtering higher frequencies or noise removal. In some way, classifying with blurred training and testing faces using the first N coefficients of the feature vectors does not affect the identification system performance, but it increases it.

Although blurring images in a pre-processing stage is not suitable, it is proven that a slightly unfocused camera during both training and testing stages should not alter the recognition rates. Test B reveals that classifying blurry faces with clear training models is not a significant impairment.

6.4 Number of transformed coefficients kept

As far as it has been simulated, selecting the right number of transformed coefficients to be kept during face classifications is a very variable decision. This section will study the evolution of the recognition rates using both DCT and PCA. The final goal is to set a standardised number of coefficients that provides good performance results in order to implement the face identification system in Chapter 7.

Experimental results

The simulation will run the same hypothetical scenario (see Table 6.5) 100 times, from 1 to 100 coefficients.

Data used	
Database	CLEAR
Image size	variable
Number of models	10
Training set	10 faces per model
Testing set	8 faces per model

Table 6.5: Details for this simulation.

DCT and PCA results show a similar behaviour as the number of coefficients kept increases. At about 50 and 60 coefficients the trends of the plots in Figure 6.2 seem to converge to a certain recognition rate and results for higher values do not improve performace.

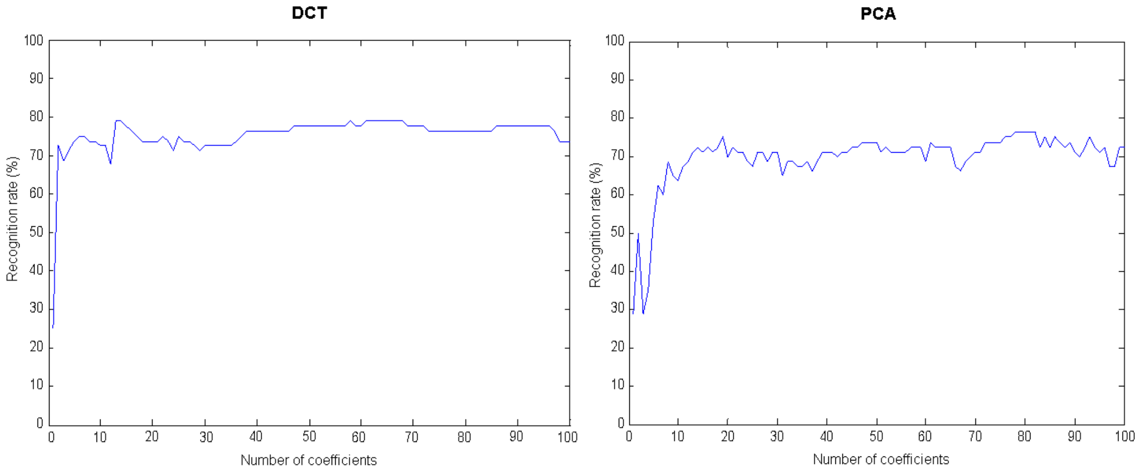

Figure 6.2: Recognition rates for DCT and PCA in function of the number of transformed coefficients kept.

Conclusions

Thanks to this experiment, the number of transformed coefficients kept is set to 60. This helps avoiding too high computational costs since keeping fewer coefficients unloads the system and that is a very interesting point for any real-time system needing the lowest possible computational costs.

6.5 Choosing the best distance metric

In Section 3.4 Manhattan, Euclidean, Minkowsky and cosine similarity distances were introduced as possible choices for face classification. Choosing the distance metric that best fits the needs of a classifier could sensibly improve classification and identification results. This will be proven in this experiment and the result will be used as a general rule in future experiments.

Experimental results

In order to choose a distance measurement function with results as independent from a face database as possible, three different databases are separately used:

Database	Focused	Different poses	BG	Std BG	Models
ORL	Yes	Yes	Little	Yes	40
Yale	Yes	Yes	Over 50%	No	15
Cropped Yale B	Yes	Yes	None	N/A	38

Table 6.6: Significant features about ORL, Yale and cropped Extended Yale B databases, where *BG* stands for *Background appearance* and *Std BG* tells whether the background is uniform or not.

Table 6.6 shows if the evaluated database presents clear and sharp images, different illumination conditions or poses, the level of appearance of the background and if a standard one was used. For each model, a total of 10 faces were selected, six for training and 4 for testing purposes. A total of 60 transformed coefficients are kept in these simulations. Results using the DCT-based feature extraction technique are shown in Table 6.7.

	Maximum recognition rate (%)		
	Manhattan	**Euclidean**	**Cosine Sim.**
ORL	93.37	91.25	80
Yale	53.75	55	55
Cropped Yale	87.5	85,62	78.75

Table 6.7: Results obtained using Manhattan, Euclidean and Cosine similarity metrics.

Conclusions

Given the results for the three distance metrics, Manhattan is chosen to be the one used throughout this research work. Furthermore, as said in Section 3.4 its lower computational cost in front of the other distance metrics reinforce the final selection.

6.6 *k* parameter in kNN

As explained in Section 3.2, k denotes the k-th of data vectors closer to the test vector. This Section will perform simulations of a face identification system with different values of k and the optimal one, that is the one providing higher recognition rates, will be taken as a reference in the rest of the work.

Experimental results

Simulations are performed using the ORL database and 30-coefficient DCT as feature extraction technique. kNN and kNN with probabilities will be tested. Further details of the simulation are described in Table 6.8.

Data used	
Database	ORL
Image size	112x92 pixels
Number of models	40
Training set	5 faces per model
Testing set	5 faces per model

Table 6.8: Details for this simulation.

Figure 6.3 shows the evolution of the recognition rate in kNN as the k parameter increases. It can be observed the recognition rate decreases dramatically for any value other than $k = 1$. This suggests that $k = 1$ should be chosen for standard kNN.

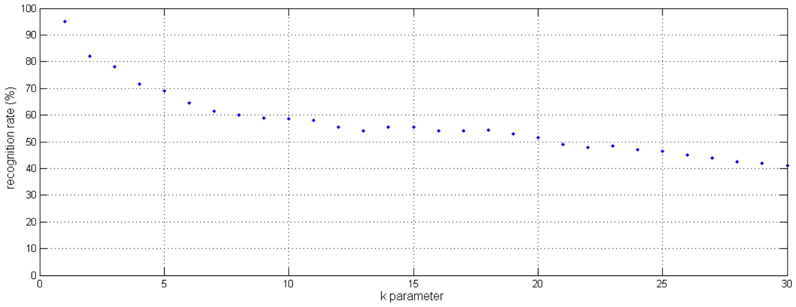

Figure 6.3: Recognition rates in function of the k parameter using ORL.

In contrast to kNN, for kNN with probabilities recognition rates do not decrease so abruptly, as shown in Figure 6.4. For $k = 2$ the recognition rate is likely to be equal as in $k = 1$ and higher values of k tend to stabilise at 80%. This Figure also suggests that $k = 1$ should be chosen for kNN with probabilities.

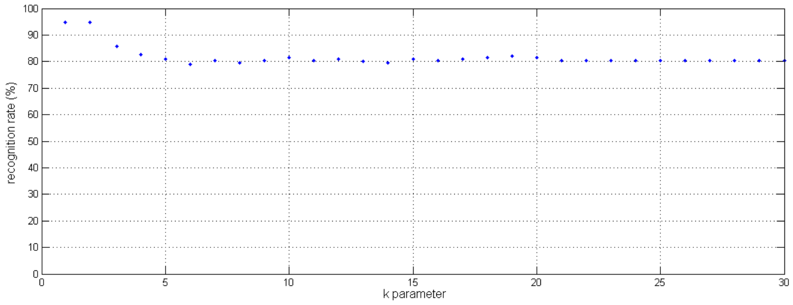

Figure 6.4: Recognition rates in function of the k parameter using ORL. Probabilities are given by kNN.

In order to reinforce the results found in Figure 6.4, another simulation is performed using a worst-case scenario face database such as CLEAR. More details in Table 6.9.

Data used	
Database	CLEAR (frontal faces)
Image size	variable
Number of models	10
Training set	10 faces per model
Testing set	8 faces per model

Table 6.9: Details for this simulation.

The trend of the recognition rate in Figure 6.5 reinforces the statement that the lower the k parameter, the better results.

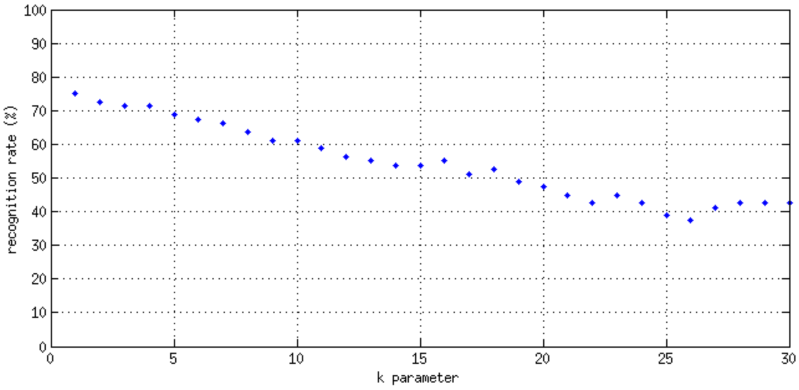

Figure 6.5: Recognition rates in function of the k parameter using CLEAR, a very low resolution database.

Conclusions

It has been proven that $k = 1$ offers the highest recognition rates in both standard kNN and kNN with probabilities. High and low resolution face databases were used with similar results. From now on in this work, this will be the chosen value for the k parameter.

6.7 Number of models and overall performance

As one could think, true identification rates should increase inversely proportional to number of models taken into account in a classification system, as the training region of the N-dimensional space decreases its feature vector density. In order to prove this statement this simulation is performed using DCT and PCA, although DCT was selected in Section 6.2 as the preferred feature extraction technique.

Experimental results

ORL face database is used in this experiment to simulate a best-case scenario due to its high resolution images, homogeneous illumination and good focusing. In non-ideal conditions, the system should underperform compared to the results found in this section.

As the number of coefficients taken into account grows, the recognition rates in PCA and DCT from each test rapidly converge to their upper bounds. Recognition rates for 20 models are mostly above of the 40-model trace but follow a very similar pattern as the number of coefficients increases. In case of having only 5 models on the database, results are outstanding and converge to 100% with only 7 coefficients. Moreover, Tables 6.11 and 6.12 show the maximum recognition rates for PCA and DCT (using kNN), including a 5-image time-based fusion of results (fusion-kNN).

Data used	
Database	ORL
Image size	112x92 pixels
Number of models	40
Training set	5 faces per model
Testing set	5 faces per model

Table 6.10: Details for this simulation.

PCA results	**kNN**		**fusion-kNN**	
Number of models	**Recog. rate (%)**	**Min. coeff**	**Recog. rate (%)**	**Min. coeff**
40	94.38	27	100	8
30	95	19	100	7
20	93.75	19	100	6
10	97.5	8	100	4
5	100	7	100	5

Table 6.11: Effect of the number of models on the recognition rate. Results for PCA.

DCT results	**kNN**		**fusion-kNN**	
Number of models	**Recog. rate (%)**	**MinCoeff**	**Recog. rate (%)**	**MinCoeff**
40	98.13	19	100	8
30	98.33	19	100	7
20	97.5	12	100	4
10	100	9	100	4
5	100	7	100	4

Table 6.12: Effect of the number of models on the recognition rate. Results for DCT.

Conclusions

An important conclusion can be drawn by observing Tables 6.11 and 6.12: there is no significant difference between having 20 or 40 models in terms of recognition rates, which is a very important sign of a robust performance. This conclusion allows face identification systems to classify among large numbers of training models without noticing an abrupt downperformance in the tested range of face models.

6.8 Fusion of face orientations using weighted scores

This experiment expects to take advantage of the possibilities offered by multiple camera environments, as introduced in Chapter 4. Having cameras on every corner oriented towards the center of the room allows the recognition system to detect faces in at least one orientation. For simplicity only three orientations are considered: frontal (which covers approximately $[-45^o, +45^o]$), right profile ($[-45^o, -90^o]$) and left profile ($[+45^o, +90^o]$).

Both training and testing processes will be time-segmented, that is, training will take place for a time interval of T_{TR} seconds and testing for T_{TE} seconds. T_{TR} will restrict the number of trained faces. Every face detected within a T_{TE} seconds segment will be classified and subsequently be treated as a unique candidate, as error-free face tracking is assumed.

Experimental results

This simulation will use the following data:

Data used	
Database	CLEAR
Recording	UKA, IBM, UPC, ITC, AIT
Image size	30x30 pixels (DCT)
Number of models	28
Training set	segments of 10 and 20 seconds
Testing set	segments of 1, 2, 3 and 5 seconds

Table 6.13: Details for this simulation.

Every available model on CLEAR databases and the four existing cameras (see Section 5.4) will be used in this experiment since depending on the recordings used, recognition rates vary greatly.

Train segments will be 10 and 20 seconds of duration while test segments will adapt their duration to real-time applications, as shown above. Test segments more than 5 seconds of duration are unlikely to be used in situations suitable for real-time face recognition such as meetings, security video footage and any other dynamic situation.

Let $\{\psi_{R,E}\} = \left\{\psi^F_{R,E}, \psi^L_{R,E}, \psi^R_{R,E}\right\}$ be the set of weights obtained for T_{TR} and T_{TE} so that every weight in $\{\psi_{R,E}\}$ corresponds to the recognition rate performed on validation tests for frontal, right and left faces (F, L and R superscripts, respectively). Every time a test face is assigned to a certain model the scoring for that model increases by $\{\psi_{R,E}\}$ depending on the orientation of the face until the test segment is over. Majority voting is performed to finally select the winner model. Next, the results for training process:

Training segment duration	**10**	**10**	**10**	**10**	**20**	**20**	**20**	**20**
Testing segment duration	**1**	**2**	**3**	**5**	**1**	**2**	**3**	**5**
F+R+L fusion recog. rate (%)	78.57	82.14	89.29	92.86	82.14	92.86	96.43	96.43
Frontal faces recog. rate (%)	53.66	51.70	54.24	53.60	80.49	76.1905	75.47	72.53
Left faces recog. rate (%)	48.61	48.39	50.86	54.85	58.33	56.4516	57.14	61.54
Right faces recog. rate (%)	46.15	49.15	51.48	54.96	55.38	59.322	63.90	65.25

Table 6.14: Results for fusion, frontal, right and left profile recognition rates using 28 CLEAR individuals and several training and testing segment durations.

Fusion recognition rates for 20 seconds of training are remarkable better than the case for 10 seconds and prove that even with small and fuzzy images, high recognition rates can be achieved when using fusion of validation scores, time-segmented data and several face orientations, thanks to multiple cameras available.

Results obtained in CLEAR 2007 Evaluations [10] are shown compared to those from this section in Table 6.15. The authors note in [10] that due to imprecise hand-labeling, additional frames were generated by changing the bounding box around the face so that better results could be obtained. This modification is not performed in this work and results contain defective faces as in a worst-case scenario. Moreover, short and long training segment durations are different in these experiments, being 15 and 30 seconds for CLEAR, 10 and 20 for this work, respectively.

Training segment duration	**Short**		**Long**	
Testing segment duration	**1**	**5**	**1**	**5**
CLEAR results	**84.6**	90.8	**89.3**	94.4
This work results(%)	78.57	**92.86**	82.14	**96.43**

Table 6.15: Fusion results from this work compared to those from winner of the CLEAR 2007 Evaluation.

A second experiment is performed to evaluate the classification system in environments with fewer participating models. Only 10 models out of the best performers are chosen in this experiment in order to obtain high recognition rates.

Data used	
Database	CLEAR
Recordings	UPC, ITC, AIT
Image size	30x30 pixels (DCT)
Number of models	10
Training set	segments of 10 and 20 seconds
Testing set	segments of 1, 2, 3 and 5 seconds

Table 6.16: Details for this simulation.

Training segment (s)	**10**	**10**	**10**	**10**	**20**	**20**	**20**	**20**
Testing segment (s)	**1**	**2**	**3**	**5**	**1**	**2**	**3**	**5**
Fusion recog. rate (%)	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
F-faces recog. rate (%)	94.74	92.31	89.66	79.28	100.00	100.00	94.83	90.09
L-faces recog. rate (%)	47.06	60.71	66.67	71.05	52.94	64.29	69.05	76.32
R-faces recog. rate (%)	88.24	90.32	86.96	81.58	88.24	90.32	91.30	89.47

Table 6.17: Results for fusion, frontal, right and left profile recognition rates using 10 CLEAR individuals and several training and testing segment durations.

The classification system achieves high performance levels for 10 models.

Conclusions

The results obtained in these simulations are above expected. Fusion of results significantly increases the recognition rate when using several face orientations in a multi-camera environment. Results for 10 CLEAR models or less are remarkable given that only 10-second training and 1-second testing segments are needed to achieve a recognition rate of 100%. Moreover, having 28 CLEAR models under the same conditions report a recognition rate of 78.57%. Therefore, using higher quality face databases or videorecordings should noticeably improve performance.

Despite the mentioned differences, simulation results in Table 6.15 are quite similar and, in testing durations of 5 seconds, those from this work outperform the winning results of the CLEAR 2007 Evaluation contest.

6.9 Number of cameras in a smart room

This experiment will test the effect of incrementing the number of operating cameras inside a smart room. The more cameras, the higher possibilities to acquire frontal and profile faces in order to merge them. As it was shown in Section 6.8, the fusion of face orientations improves the overall face identification results and this leads to answering the question of which is the number of cameras in a multi-camera environment to be suitable for a face identification system.

Experimental results

Since real-time applications require real-time results the maximum duration of the testing segments will be 2 seconds. Table 6.18 shows the configuration of the simulation.

Database	CLEAR
Recording	UKA, IBM, UPC, ITC, AIT
Image size	30x30 pixels (DCT)
Number of models	28
Training set	segments of 10 and 20 seconds
Testing set	segments of 1 and 2 seconds

Table 6.18: Details for this simulation.

In Table 6.19 results for one and two cameras are shown. It is easy to observe that having only one camera and using fusion of orientations achieves very low recognition rates in all cases while two operative cameras helps taking some advantage of fusion of results. These results are not acceptable enough for a face identification system in an adverse environment.

Number of cameras	**1**				**2**			
Training segment duration	**10**	**10**	**20**	**20**	**10**	**10**	**20**	**20**
Testing segment duration	**1**	**2**	**1**	**2**	**1**	**2**	**1**	**2**
F+R+L fusion recog. rate (%)	39.28	39.28	53.57	53.57	64.29	67.86	67.86	75
Frontal faces recog. rate (%)	50	42.55	83.33	81.4	54.55	55.67	87.27	85.57
Left faces recog. rate (%)	33.33	39.53	60	65.71	26.67	32.08	40	43.4
Right faces recog. rate (%)	41.17	44.82	84.61	81.82	53.49	56.41	65.12	67.95

Table 6.19: Results for fusion, frontal, right and left profile recognition rates using one and two cameras, 28 CLEAR individuals, several training and testing segment durations.

In contrast to the previous results, Table 6.20 shows the results for three and four cameras. Performance is greatly improved when using more than three cameras and training segments of 10 seconds while using 20 seconds for training overloads the system in vain and does not improve recognition rates after fusion of results. Adding a fourth camera has no effect niether on the results by orientation nor in fusion of results.

Number of cameras	**3**				**4**			
Training segment duration	**10**	**10**	**20**	**20**	**10**	**10**	**20**	**20**
Testing segment duration	**1**	**2**	**1**	**2**	**1**	**2**	**1**	**2**
F+R+L fusion recog. rate (%)	78.57	82.14	82.14	92.86	78.57	82.14	82.14	92.86
Frontal faces recog. rate (%)	58.44	55.88	84.42	81.62	53.66	51.7	80.49	76.19
Left faces recog. rate (%)	50	48.96	60.71	58.33	48.61	48.39	58.33	56.45
Right faces recog. rate (%)	50.94	53.54	64.15	66.67	46.15	49.15	55.38	59.32

Table 6.20: Results for fusion, frontal, right and left profile recognition rates using three and four cameras, 28 CLEAR individuals, several training and testing segment durations.

Conclusions

Multi-camera environments enable multiple face orientations systems. It has been stated that the effectivity of adding cameras to a smart room is not a trivial matter, since results show that for more than three cameras there is no significant variation in the performance of the system.

6.10 Auto-updating classification system based on model variance

During the testing stage in a face recognition system new faces may be included in a certain model, whether it is as a new training feature vector or it substitutes an existing one, or a new model may be created. Reducing a training data set variance might incur in two situations: an excessively spread training data set becoming more compact or a training data set shrinking its volume. The first situation could separate overlapped model regions by reducing their span and deleting aberrant cases. The second one can leave incoming faces out of their true model and be assigned to the closest one, decreasing recognition rates.

This experiment will prove if an auto-updating classification system can be implemented only by reducing the model variance toward a defined limit.

Experimental results

The variances from the existing models need to be calculated and, for each model, the feature vector which contributes the most to increasing the variance of its own model is be marked. When a test feature vector is assigned to a particular model the total variance of the model is calculated after replacing the previously marked feature vector. If the resulting variance is less than the previous and more than a stated lower limit, the feature vector is replaced.

Database	CLEAR
Recording	UKA, IBM, UPC
Image size	30x30 pixels (DCT)
Number of models	10 (IBM 2, UKA 4, UPC 4)
Training set	10 faces per model (7 profile / 3 frontal)
Testing set	8 faces per model (6 profile / 2 frontal)

Table 6.21: Details for this simulation.

Three CLEAR databases are used for a total of 10 models and, as in the previous experiment, frontal and profile faces are used.

	Recognition rate (%)							
	UPC		**IBM**		**UKA**		**UPC+IBM+UKA**	
	Orig	*Upd*	*Orig*	*Upd*	*Orig*	*Upd*	*Orig*	*Upd*
Frontal	70.83	66.66	93.75	75	81.25	81.25	63.89	65.28
Profile	78.12	78.12	100	91.67	96.87	93.75	72.50	71.25
Frontal+Profile	56.25	56.25	81.25	87.50	93.75	96.87	60	65

Table 6.22: Recognition rates obtained after updating is performed.

In table 6.22 are shown the results for both non-updated and updated training data sets while performing classification. On average, recognition rates are worse after using the auto-updating classification system excepting when processing frontal and profile at a time.

Conclusions

Besides the acceptable results for frontal+profile faces, increasing recognition rates as much as 6.25%, this system cannot be safely used in other unknown circumstances at this time. More simulations should be performed using face databases that include individuals in time-lapse of months or even years so that the feasibility of an auto-updating system could be properly tested.

Part III

Implementation

Chapter 7

A real-time face identification system

7.1 Introduction

To demonstrate the previous results found in Chapter 6, a face identification system is built and run in real-time conditions. Tests will take place in the smart room described in Section A.1 but the system should be able to operate in other environments. This means software cannot be environment-dependent at all. The system will be coded in C++ in the Smart Flow framework [21], which allows the possibility of running multiple processes known as clients. Every client can receive or send data flows through connection ports, along with timestamps to synchronise the execution of clients.

The main purpose of the system is to track and identify every person entering a room. Video streams will be obtained from a camera in front of the door and at height not much higher than a person, since face detectors usually underperform when detecting overhead faces.

As a first approach to this challenge, the most significant features of the system are stated:

- Viola-Jones method [34] is chosen as face detector, using frontal and profile filters cascades.
- Due to limitations in profile cascades, images will be vertically flipped when having right profile faces.
- DCT is chosen as feature extraction technique. The number of coefficients kept is set to 60.

- kNN with probabilities will classify feature vectors, whether testing or training mode is active.

- The area the video stream shows is reduced to a region where faces can appear, decreasing computational costs.

- The application will also output XML strings with information to fit the client's requirements.

7.2 Design of the system

The system will be made up of five clients. These five clients and their connections are shown in Figure 7.1. Ports on the left of the modules are data inputs while on the right are outputs.

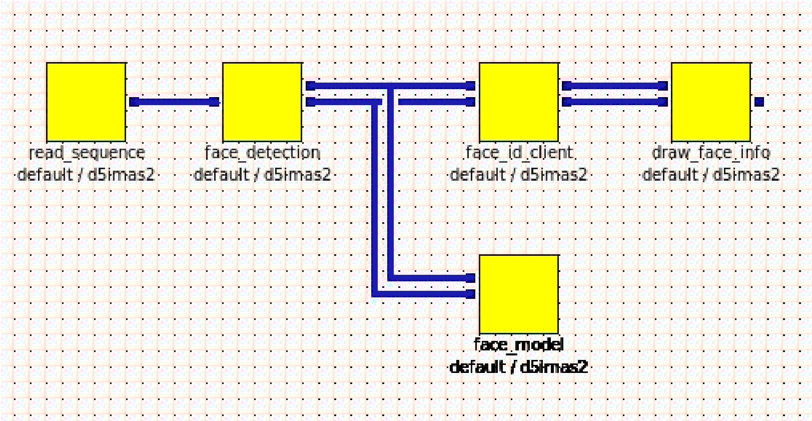

Figure 7.1: Snapshot of the structure used in Smart Flow for implementing a face identification system.

- **read_sequence**: loads and reads video frames stored in disk or captures video directly from a video camera. Sends images every $1/fps$ seconds, where fps stands for 'frames per second'.

- **face_detection**: performs Viola-Jones scanning for frontal faces. In case any frontal face is found, left-profile faces and right-profile faces cascades are used. Profile faces cascades are only suitable for left-profile faces and the image must be flipped. In order to save computational time the client can restrict the effective working area of the video stream. The right area must define the size and position of the rectangle that will mask the original frame. Moreover, a restriction on the maximum position change of a face is set up. Whether an individual is still or not, his face will remain in a limited area for, at least, the next frame. When Viola-Jones detects a face too far from its previous position, it is rejected by face_detection client as it could be a false positive since faces do not suddenly appear in continuous recorded video sequences. This client outputs the input flow and basic information about the region of interest where the face was detected, which will be used by other clients to crop faces from a frame.

- **face_model**: checks if a face model file exists, creating it if not. Receives video flows along with regions of interest to perform DCT transformation on cropped faces.

- **face_id_client**: loads face models from disk. It also performs DCT transformation as in face_model client and classifies faces.

- **draw_face_info**: this client is able to display and save the original video stream and highlighting the detected faces. The name of the individual displayed appears under the highlighted face.

7.3 Training and testing recordings

In order to create models from individuals a training recording is required. Once individuals enter the smart-room or scenario it is necessary to wisely choose where they have to look. During the recordings performed for this real-time implementation, individuals are asked to look right, left and front for at least 2 seconds. Next, they must exit the scenario at a normal pace. These guided recordings give the face detector good frontal and profile faces, as well as faces in motion. Initially, detected faces from a same individual are labeled with a unique identification number and a default name such as 'unknown' is given (see Figure 7.2). It is up to the user to give a short name for every individual in the training recordings, although face classification will only operate with identification numbers.

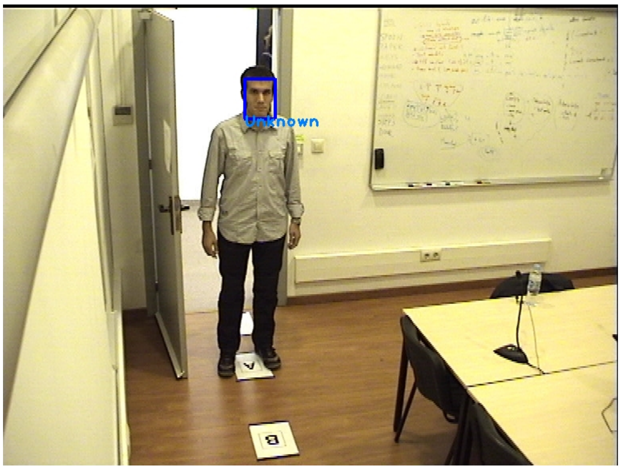

Figure 7.2: Example frame of a training sequence where faces and their identities are still unknown. After detecting the face, a bounding box is created and draw_face_info draws a rectangle around it.

In testing recordings individuals are asked to normally walk around the scenario without looking at the camera, since it would distort the final results.

7.4 Face detection

Viola and Jones introduced in [34] a face detector based on cascades of very simple and aggressive classifiers which look for very specific face features, discarding backgrounds and other objects in an image. This technique uses Adaptive Boosting (also known as *AdaBoost*) and inherits the idea of weak Haar classifiers from Freund and Schapire's first general approach in [12]. The main idea is that, if well chosen, the union of weak classifiers leads to a very powerful classifier. For example, 32 classifiers would perform up to 80,000 operations in 384x288-pixel images. The first classifiers discard very strong features as backgrounds and shape of a face while the other classifiers scan the image for eyes and nose position, auto-scaling their scanning area from a minimum to a maximum face size. Scaling and strictness are adaptable parameters of a face detector, and should be carefully used because the overall face identification system could underperform.

Cascades of classifiers are obtained after a very broad 'face/no-face' machine learning process. Thousands of examples in different scales are needed in order to obtain cascades with low false positive rates [29, 26]. These face detectors also output a joint confidence parameter for all the involved weak classifiers that could prevent the recognition system from classifying wrong faces [8].

This implementation will work with a face detector in OpenCV, using frontal and left profile faces cascades since no cascade for right profile faces is available yet. The face detection module will perform as follows:

- The detector scans the image for frontal faces from minimum to maximum sizes. The implemented can bear one individual at a time.

- If no frontal face is found, the detector scans for left profile faces from minimum to maximum sizes.

- If no left profile face is found, the detector vertically flips the input image and scans for right profile faces from minimum to maximum sizes.

- Finally, if the detector has found a face its coordinates will be outputted as a bounding box. In other case, an empty bounding box is sent.

7.5 Reducing computational costs

Computational costs in real-time face identification systems can be significantly reduced by applying very simple strategies that will be explained next.

Reduced constant region of interest

Video sequences from cameras might not be initially set up for face identification. Given a frame from a video sequence, as in Figure 7.3, individuals' faces may be located only in certain regions of the frame. Not detecting in regions in margins or close to the camera is an effective way to reduce computational costs. Note that in Figure 7.3 there is an individual in the bottom-left quadrant that will not be detected as the face is unfocused.

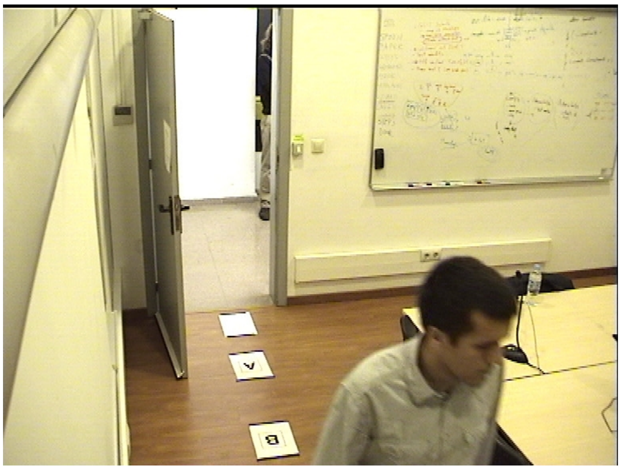

Figure 7.3: The individual is situated in a region of the frame where faces cannot be detected.

The implemented face identification system will allow restricting the detection region as shown in Figure 7.4. A constant reduced region of interest can be set up by giving its (x, y) top left coordinates, width and height. Results in Section 7.6 use a constant reduced region of interest of $215x340$ which implies a scanning area almost 5 times smaller than in images of size $720x480$ pixels.

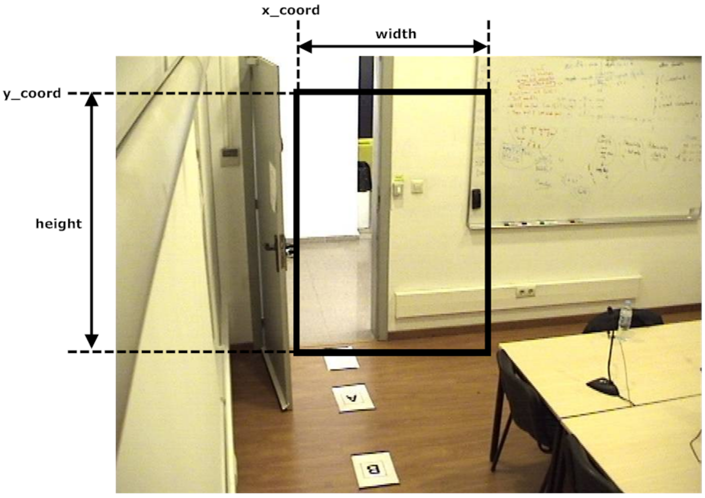

Figure 7.4: Definition of the coordinates and sizes of a rectangle that will mask and restrict the region of interest.

Minimum and maximum size of faces

Frontal or profile cascades of Haar classifiers accept a minimum face size when detecting faces. This reduces computational costs since the face detector does not need to scan for the whole range of face sizes. Figure 7.5 shows an example frame with two faces: the one on the top is 20x30 pixels and the one on the bottom is 40x80 pixels, being minimum and maximum face sizes respectively. The first one is located outside the smart room and thanks to the minimum size limitation will not be detected or classified. Maximum size limitation avoids the face detector to find false faces around in the room, specially in walls and tissues.

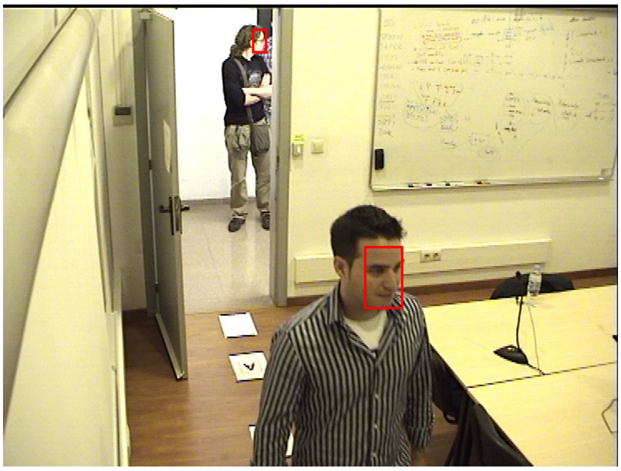

Figure 7.5: Examples of faces out of the expected size range.

Dynamic Region of Interest detection

When an individual enters a room, his path follows a trace similar to that in Figure 7.6. Note that the trace creates the illusion of very close regions of interest around faces, even when sampling every five frames as in the image.

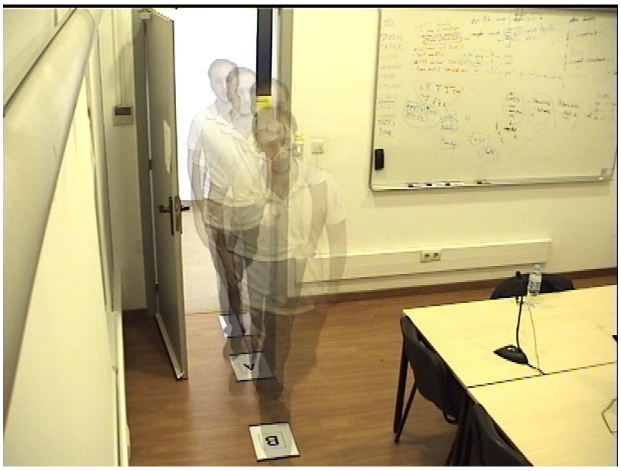

Figure 7.6: Trace of an individual entering a smart room. Images are overlapped every 5 frames for the sake of simplicity.

In fact, in Figure 7.7 can be clearly seen that two consecutive faces will be certainly separated by very few pixels in vertical or horizontal dimensions. The implemented face detector will be able to only scan around an even more restricted region of interest. Everytime a face is detected, the region of interest is dynamically centered to the individual's face. Note that the first time a face is detected or a face trace is lost the face detector should have the Dynamic Region of Interest detection mode disabled. Not disabling it could make the system scan permanently in the last region of interest stored.

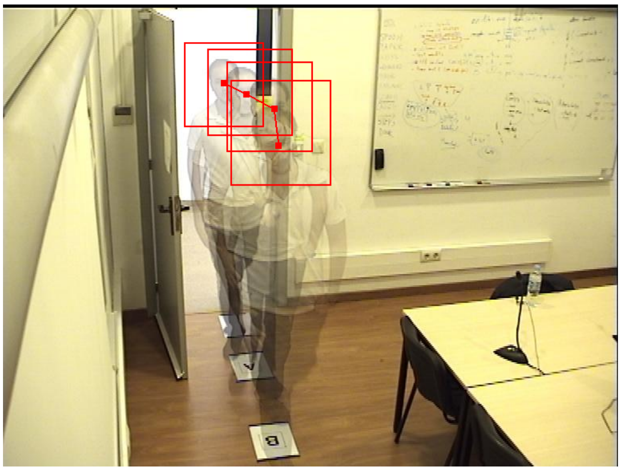

Figure 7.7: Dynamic regions of interest centered on the detected faces along the trace.

In this particular case, an aggressive region of interest of size 120x120 is set up. That means having a scanning region 5 times smaller than in the constant reduced region of interest and 24 times smaller than the original frame. Computational cost is extremely reduced using this strategy.

7.6 Results

The implementation of a real-time face identification system is finally tested. The system works as expected and results are outstanding due to the good quality of the video sequences. As an example, Figure 7.8 shows an individual correctly detected and classified with his name appearing in the image (see Figure 7.9 for a closer view). Note that the individual standing outside the smart room does not interfere in the performance of the system.

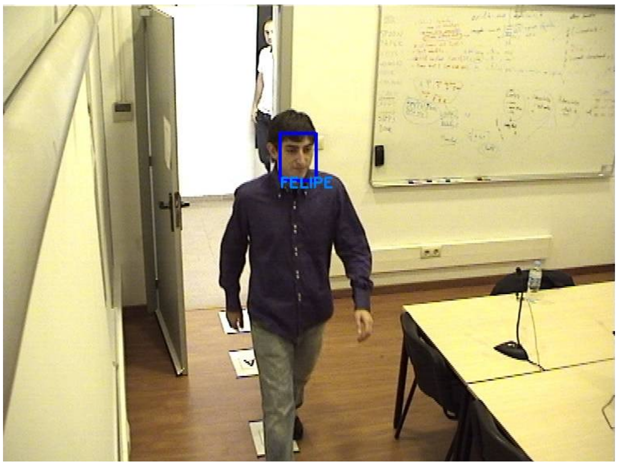

Figure 7.8: The individual has been classified and identified, and his name is drawn by the draw_face_info module.

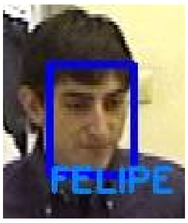

Figure 7.9: A closer look of the resulting identification.

The testing sequence has a duration of 1 minute and 48 seconds in which 372 frontal and 78 profile faces are detected. Profile faces are defined as those faces not detected by Viola-Jones method using a frontal but a profile cascade of filters. A total of 12 individuals enter the smart room and their traces are followed for time segments of variable length. That means after a certain number of detected faces the system performs a fusion of results. Table 7.1 shows the evolution of instant recognition rates for frontal and profile faces and fusion of results when changing the time segment length of the implemented system.

	Recognition rate			
segment length (sec)	**fusion**	**frontal**	**profile**	**number of fusions**
5	86.02%	82.53%	97.44%	93
25	91.67%	82.53%	97.44%	24
30	91.30%	82.53%	97.44%	23

Table 7.1: Results for fusion and instant frontal and profile recognition rates for time segments of lengths 5, 25 and 30 seconds.

As segment length decreases, the number of fusions increases achieving lower instant and fusion recognition rates since availabe information on identities is limited. On the other hand, too long time segments do not improve fusion recognition rates as there is too much redundant information. A permanent value for the time segment length is set to 25 frames. Using this configuration, Table 7.2 explains the effects of varying the k parameter in kNN.

	Recognition rate		
***k* param**	**fusion**	**frontal**	**profile**
1	91.97%	82.53%	97.44%
3	91.67%	80.38%	96.15%
5	91.67%	77.42%	94.87%
7	91.67%	76.34%	93.59%

Table 7.2: Results for fusion and instant frontal and profile recognition rates in function of the k parameter.

Note that, as proven in Section 6.6, the higher the k, the lower the instant performance despite fusion of results gives the same recognition rates for $k = \{1, 3, 5, 7\}$. Therefore, better results are expected when using $k = 1$ and will be set as a default parameter of the system.

Outstanding instant recognition results and fusion of results close to 92% are achieved with this implementation of a real-time face identification system, working on an ordinary dual-core laptop.

Chapter 8

Conclusions and future work

8.1 Conclusions

This work has successfully achieved its final goal: implementing a real-time face identification system. Hundreds of simulations have led to the configuration parameters that were pursued from the very beginning, achieving final recognition rates of 90% and above.

DCT transformation has proved to be more versatile and faster in characterising faces than PCA, as well as better results were obtained. In cooperation with DCT, k Nearest Neighbor is the best classifier in terms of conceptual simplicity, lower computational costs and ease of implementation and modification. Moreover, a reviewed version of kNN was implemented, making possible to obtain probabilities from a face classification.

The main conclusions obtained from this work are detailed next:

- The size of a face does not significantly affect the recognition rate of PCA or DCT-based identification systems. A standard size of 32x32 pixels is set to reduce computational costs.

- Manhattan distance is chosen after proving better classification results and less computational cost.

- The k parameter in kNN is set to $k = 1$ after showing better results thant other options. This also reduces computational costs since only one neighbor has to be taken into account.

- Increasing the number of face models within the tested range does not strongly affect recognition rates but necessarily increases computational costs.

- Slightly blurry face imagesare not a significant inconvenience to a face identification system.

- Under the multi-camera environment described there is a maximum in the number of cameras that can provide optimal recognition rates.

- An auto-updating face identification system based on face color variance is not an option since its advantages are not conclusive

- False detections can be reduced by using an entropy-based discriminator, improving recognition rates.

8.2 Future work

Due to limitations in time, other feature extraction techniques such as LDA and LPP or classificators such as SVM were not considered in this research work but might be interesting alternatives to PCA, DCT and kNN. A work exploring the possibilities offered by these techniques could help improve the implemented identification system even more.

This work is open to multimodal fusion and could be complemented by speaker identification or gesture detection by testing every described weighted mean and other fusion strategies such as the Minimax rule.

Multicamera environments allow the system to create 3D face models from several faces taken from different relative angles. The implemented system could be adapted to fit the needs of a 3D face identification system.

Appendix A

Smart rooms and image acquisition

A.1 A Smart room

Designing a face identification system requires video recordings in controlled environments in order to arrange all the necessary simulations and tests. The Technical University of Catalonia, (UPC, located in Barcelona, Spain) is in possession of an advanced smart room where video and audio data can be extracted from different types of sensors.

This smart room is meant to be a test field for image or sound processing researchers since it recreates a small meeting room with quite a large quantity of sensors. Some of them are [20]:

- A 68-microphone array.

- Three clusters with four Hammer microphones each.

- Six JVC TK-C1481BEG cameras focused to the centre of the room, one on each corner and two in the middle.

- An overhead camera in the centre of the room.

- Two PTZ cameras in the bottom of the room focused to the door.

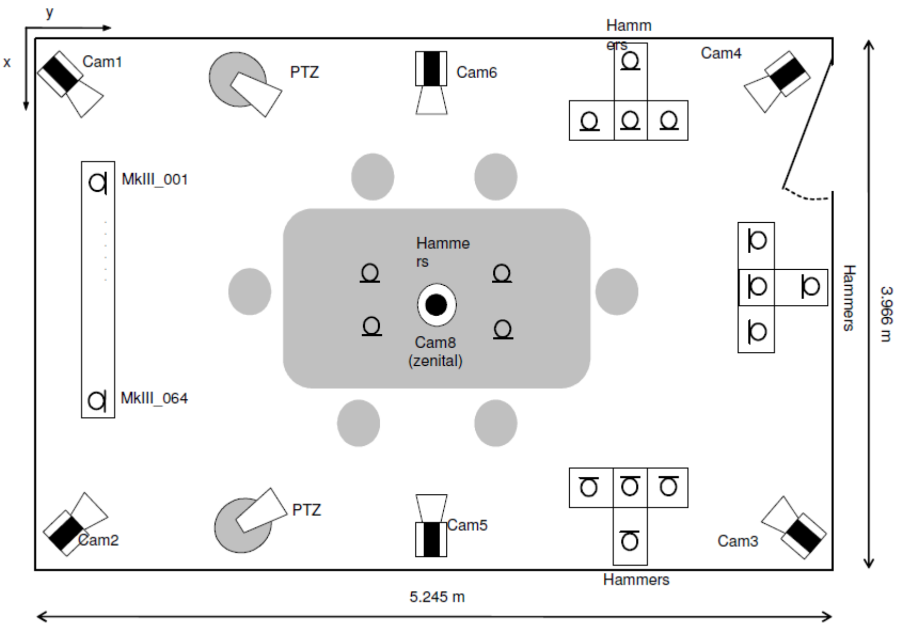

Figure A.1: Sensors that may be found in a smart room. Reproduced with permission of the owner [20].

Along with all the devices detailed above, the smart room is equipped with control and storage servers among many others, that allow the operators to easily synchronise and acquire data from sensors. This work will only use the data directly from cameras since no multimodal processing is expected.

The quality of the images acquired from a camera may vary the performance of a face recognition system. The effect of different image sizes in performance rates will be discussed in Part II, as well as blurry or unfocused images. As long as faces be big enough and not distorted, AdaBoost face detectors can successfully perform under severe conditions.

PTZ cameras have a large focal length, that implies that the region of the room that is focused is smaller than using a JVC camera. Faces from PTZ's will be bigger than those from JVC's, and this situation could lead the recognition system to underperform if using both cameras at the same time.

A.2 Acquiring images

Images can be defined as sets of light points creating visual perceptions and can be easily represented in color or luminance matrices or vectors. RGB or YUV color models are widely used and allow images to be represented as the addition of several light components.

RGB images are represented into red, green and blue component matrices while YUV images contain luminance and chrominance matrices.

Black and white images are mostly used in face identification systems since color levels do not contribute in improving recognition rates. On the contrary, having multiple different color channels increases computation costs unnecessarily.

Black and white images can be extracted applying a linear combination of the three RGB matrices, which are also represented in the so-called Y or luminance component matrix in YUV models

$$Y = W_R \cdot R + W_G \cdot G + W_B \cdot B$$

where weighting coefficients are usually set to

$$W_R = 0.299$$

$$W_G = 0.114$$

$$W_B = 1 - W_R - W_G$$

A.3 Deinterlacing images

The illusion of motion in video sequences for the human eye starts at about 10 frames/seconds[27] but current video devices use higher frame rates. This is due to the persistence of vision in the retina. Moreover, frame rates of 40 Hz or lower cause disturbing flickering phenomena and since video is usually recorded at 24-30 frames/second, video frame rates must be doubled whether by repeating frames or by interlacing.

This interlace technique merges images of a 50 Hz video sequence into 25 Hz. Odd and even lines of the interlaced image are drawn from odd and even lines of the original frames. After interlacing, the human eye is not able to detect the flickering phenomena and images seem to be constant in illumination but fast horizontal movements create a saw teeth effect as seen in Figure A.2.

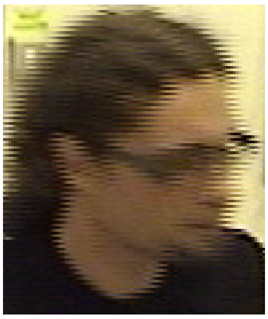

Figure A.2: Interlaced image of a face.

This situation is not desirable in face detection systems as face detectors could not detect a face presenting such a high saw teeth effect. Image doubling is performed in order to avoid this effect. One of the two interlaced images is extracted by taking either odd or even lines, which are doubled and stored in the deinterlaced image as in Figure A.3.

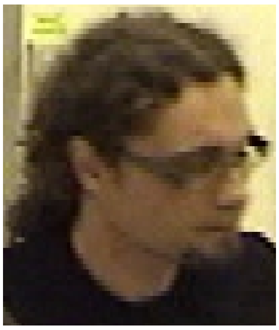

Figure A.3: Deinterlaced image of the same face as in Figure A.2.

If necessary, images databases in Section 5.4 will be doubled using an simple application created for this purpose.

Appendix B

Entropy-based invalid image discriminator

Introduction

As an optional but useful part of a face recognition system, image discriminators can increase recognition rates by avoiding the classifiers to train or test images not containing faces. Bad croppings or segmentations are usual problems in face detection and their contribution can be very negative in face classification, since invalid images may contain plain backgrounds or partial faces.

A heuristic mathematical method based on information entropy will be proven, as plain or semi-plain images contain significantly less information than those framing a full face.

Experimental results

Let $F(p,q)$ be a function that returns

$$F(p,q) = \begin{cases} 1 & if\ p = q \\ 0 & if\ p \neq q \end{cases}$$

and let $T(x,y)$ be a given image of size NxM. A density estimation must be performed over $T(x,y)$ as shown in

$$p(k,T) = \frac{1}{NM} \sum_{n=1}^{N} \sum_{m=1}^{M} F(T(n,m),k)$$

The entropy is obtained as

$$H(T) = -\sum_{k=1}^{255} p(k,T) log_2 p(k,T)$$

which satisfies

$$\lim_{p(k,T)\rightarrow\infty} p(k,T) log_2 p(k,T) = 0$$

For 30x30-pixel images and a color depth of 8 bits, a threshold of $H(T) < 150$ is set to reject invalid images. Next are shown some of the rejected CLEAR database images with this discriminator.

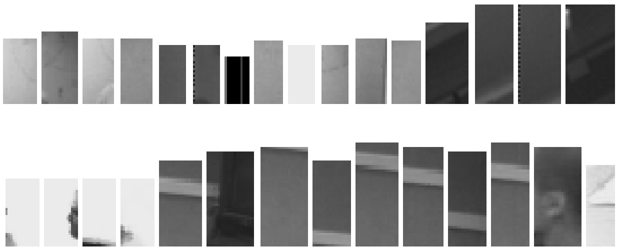

Figure B.1: Defective faces that are rejected by the discriminator.

To numerically illustrate the effects of the proposed invalid image discriminator, three data sets from IBM, ITC and UPC are simulated with a conservative threshold (see table B.1).

Recording	**Total images**	**Rejected images**	**False negatives**
IBM	1155	198 (17%)	20 (10.1%)
ITC	1793	157 (8.7%)	32 (20.4%)
UPC	3473	346 (10%)	13 (3.7%)

Table B.1: Rejection and false negative rates for CLEAR database.

Conclusions

Using an entropy-based discriminator can substantially reduce the number of invalid images from a training or testing data set. Although false negatives reduce the number of valid faces, in most cases these images do not appropriately represent a face fully and clearly enough and avoids the problem of processing and classifying invalid faces.

Bibliography

[1] Face recognition using laplacianfaces. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(3):328–340, 2005.

[2] Proyecto hesperia. https://www.proyecto-hesperia.org, July 2009.

[3] AT&T. The database of faces. http://www.cl.cam.ac.uk/research/dtg/attarchive/, July 2009.

[4] Boost. Boost c++ libraries. http://www.boost.org/, July 2009.

[5] Deng Cai, Xiaofei He, and Jiawei Han. Document clustering using locality preserving indexing. *IEEE Transactions on Knowledge and Data Engineering*, 17(12):1624–1637, 2005.

[6] W. Chen, Meng J. Er, and Shiqian Wu. Illumination compensation and normalization for robust face recognition using discrete cosine transform in logarithm domain. *Systems, Man and Cybernetics, Part B, IEEE Transactions on*, 36(2):458–466, 2006.

[7] Jacek Czyz and Pierre Dupont (ucl/ingi. Decision fusion in identity verification using facial images, 2003.

[8] Cem Demirkir and Bülent Sankur. Face detection using look-up table based gentle adaboost. In *AVBPA*, pages 339–345, 2005.

[9] R.O. Duda, P.E. Hart, and D.G. Stork. *Pattern Classification.* Wiley, 2001.

[10] Hazim Kemal Ekenel, Qin Jin, Mika Fischer, and Rainer Stiefelhagen. Isl person identification systems in the clear 2007 evaluations. pages 256–265, 2008.

[11] R.A. Fisher. The statistical utilization of multiple measurements. *Annals of Eugenics*, 8:376–386, 1938.

[12] Yoav Freund and Robert E. Schapire. A short introduction to boosting. In *In Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, pages 1401–1406. Morgan Kaufmann, 1999.

[13] A.S. Georghiades, P.N. Belhumeur, and D.J. Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Trans. Pattern Anal. Mach. Intelligence*, 23(6):643–660, 2001.

[14] Ziad M. Hafed and Martin D. Levine. Face recognition using the discrete cosine transform. *International Journal of Computer Vision*, 43(3):167–188, 2001.

[15] Tae-Kyun Kim and Josef Kittler. Locally linear discriminant analysis for multimodally distributed classes for face recognition with a single model image. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(3):318–327, 2005.

[16] Ludmila I. Kuncheva. *Combining Pattern Classifiers: Methods and Algorithms.* Wiley-Interscience, 2004.

[17] Haiping Lu, K. N. Plataniotis, and A. N. Venetsanopoulos. Multilinear principal component analysis of tensor objects for recognition. In *in Proc. Int. Conf. on Pattern Recognition*, pages 776–779, 2006.

[18] Jordi Luque, Ramon Morros, Jan Anguita, Mireia Farrus, A. Garde, D. Macho, Ferran Marqués, C. Martínez, Verónica Vilaplana, and Javier Hernando. Audio, video and multimodal person identification in a smart room. In *Lecture Notes in Computer Science LNCS, CLEAR 2006*, pages pp. 258–269, Southampton, UK, April 2006.

[19] Aleix M. Martínez and Avinash C. Kak. Pca versus lda. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23(2):228–233, February 2001.

[20] Albert Gil Moreno. Sistema de gestió de vídeo off-line per una smart-room. Barcelona, Spain, 2007. Universitat Politècnica de Catalunya.

[21] NIST. The nist smart space project. http://www.nist.gov/smartspace/, July 2009.

[22] US National Institute of Standards and Technology (NIST). Clear 2007. http://www.clear-evaluation.org/, July 2009.

[23] OpenCV. Opencv wiki. http://opencv.willowgarage.com/wiki/, July 2009.

[24] Zhengjun Pan and Hamid Bolouri. High speed face recognition based on discrete cosine transforms and neural networks. Technical report, 1999.

[25] Emanuel Parzen. On estimation of a probability density function and mode. *The Annals of Mathematical Statistics*, 33(3):1065–1076, 1962.

[26] J. Ruiz del Solar R. Verschae. A hybrid face detector based on an asymmetrical adaboost cascade detector and a wavelet-bayesian-detector.

[27] M. Robin and M. Poulin. *Digital Television Fundamentals: Design and Installation of Video and Audio Systems*. McGraw-Hill Professional, New York, 2nd edition, 2000.

[28] F. S. Samaria, F. S. Samaria, A.C. Harter, and Old Addenbrookes Site. Parameterisation of a stochastic model for human face identification, 1994.

[29] Moshe Butman Shai Avidan. Efficient methods for privacy preserving face detection.

[30] Rainer Stiefelhagen, Keni Bernardin, Rachel Bowers, R. Travis Rose, Martial Michel, and John S. Garofolo. The clear 2007 evaluation. In *CLEAR*, pages 3–34, 2007.

[31] Yale University. The extended yale face database b. http://vision.ucsd.edu/ leekc/ExtYaleDatabase/ExtYaleB.html, July 2009.

[32] Yale University. Yale faces. http://cvc.yale.edu/projects/yalefaces/yalefaces.html, July 2009.

[33] Pascal Vincent and Yoshua Bengio. Manifold parzen windows. In *Advances in Neural Information Processing Systems 15*, pages 825–832. MIT Press, 2002.

[34] Paul Viola and Michael Jones. Robust real-time object detection. In *International Journal of Computer Vision*, 2001.

[35] Ming-Hsuan Yang. Kernel eigenfaces vs. kernel fisherfaces: Face recognition using kernel methods. *Automatic Face and Gesture Recognition, IEEE International Conference on*, 0:0215, 2002.

[36] W. Zhao, R. Chellappa, and P.J. Phillips. Subspace linear discriminant analysis for face recognition. Technical report, 1999.