Filipe Daniel Lourenço Silva

Recognition of facial expressions and their temporal segments using LBP-TOP descriptors in Random Forests

Master Thesis in
Electrical and Computer Engineering

February 2015

· U C ·

UNIVERSIDADE DE COIMBRA

Recognition of facial expressions and their temporal segments using LBP-TOP descriptors in Random Forests

Author:
Filipe Silva

Supervisor:
Prof. Dr. Jorge Batista

Master Thesis in Electrical and Computer Engineering

Jury:
Prof. Dr. Helder Araújo
Prof. Dr. João Ferreira
Prof. Dr. Jorge Batista

February 2015

Acknowledgements

I cannot express enough gratitude for all of the individuals that helped me during the course of this work.

First and foremost I would like to thank my Mentor and Supervisor, Professor Jorge Batista, for his guidance, knowledge and continuous patience, all of which are seemingly endless. Without his help the success of this work would have never been possible.

To my colleagues in the lab, João Faro, Pedro Martins, João Filipe, and Pedro Marques, thank you for all of your support and help when I was in need of it most.

To my friends and family, thank you for putting up with me, even though it has not always been the easiest task. Also thank you for providing the occasional distraction when I needed a break, your company always makes for the best time. A special thanks to Rachel Ledig for all the precious help.

Finally, a big thanks to my parents, without you I would not be the person I am today. Thank you for always being there through good and bad, your support and encouragement has never gone unnoticed. You have given me the courage to reach for things I never thought imaginable, and for this I thank you so much.

Resumo

Este trabalho pretende desenvolver um sistema automático para reconhecimento de expressões faciais usando dados espaço-temporais 4D. Este tipo de dados tem a vantagem de ser mais robusto contra variações das condições ambientais, quando comparado com sistemas que usam dados 2D. Além disso permite o reconhecimento de expressões faciais a longo de todo o seu domínio temporal. Assim a correcta utilização deste tipo de dados é essencial para a trasmisão destes sistemas a aplicações da vida real.

Os dados são representados através dos histogramas de um descritor de texturas temporais, chamado Padrões Binários Locais em três planos ortogonais. A criação de uma nove abordagem para analisar estas representações dos dados por um algoritmo supervisionado de aprendizagem, chamado *Florestas de Hough*, permitiram a este algoritmo a aprendizagem das características discriminativas entre diferentes classes de expressões faciais.

Tanto para o reconhecimento da classe como para o estado temporal de uma expressão facial as arvores que compõem as *Florestas de Hough* votam num espaço de Hough com quatro dimensões, obtendo o estado da arte em reconhecimento de expressões faciais, com 93% de taxa de reconhecimento na base de dados disponível.

Keywords

Imagens APDI, LBP-TOP, Arvores de Decisão, Florestas Aleatórias, Florestas de Hough, Reconhecimento de Expressões Faciais, Aprendizagem Máquina

Abstract

This work aims to develop an Automatic Facial Expression Recognition system using 4D spatio-temporal data. This type data has the advantage of being more resilient against environmental variations, when compared with 2D approaches. Moreover it allows for a recognition of the facial expressions through out all of its temporal domain. Therefore the correct usage of this data is a necessary evolution for this systems to be used real-life application.

The data is represented through histograms of a temporal texture descriptor, the Local Binary Patters on Three Orthogonal Planes. The creation of a novel approach to analyse these representations of the data by a supervised machine learning algorithm, the *Hough Forests*, allowed for this algorithm to learn the discriminative features in between classes of Facial Expressions.

For the recognition of both the class and the temporal stage of a Facial Expression the trees that compose a *Hough Forest* vote in a 4D dimensional Hough Space, archiving a state of the art recognition rate of Facial Expressions in the available database, with 93% recognition rate of facial expressions.

Keywords

APDI Images, LBP-TOP, Decision Trees, Random Forests, Hough Forests, Facial Expression Recognition, Machine Learning

Contents

List of Tables

List of Figures

Acronyms

APDI Azimuthal Projection Distance Image

AEP Azimuthal Equidistant Projection

LBP Local Binary Patterns

LBP-TOP Local Binary Patterns on Three Orthogonal Planes

VLBP Volume Local Binary Patterns

2D Two dimensional spatial data

3D Three dimensional spatial data

4D Four dimensional spatio-temporal data

Chapter 1

Introduction

The perception of other's emotions, feelings and interactions has always been an essential ability for the human species. Facial Expression Recognition plays a major role in this skill, due to the fact that facial expressions are universal, unconscious and essential for communication [5]. The first studies on Facial Expressions and Physiognomy, the assessment of a person's character or personality from their outer appearance [6], date back to the 4th century BC. However, the first major advancements in this area only occurred during the 17th and 18th centuries, with the work of John Bulwer, Le Brun and Darwin. More recently, with the work of the psychologist Paul Ekman in the 1970s [7] and the appearance of cheap computational power, the interest of the scientific community in facial expression recognition and its technological applications has grown. In addition, large advancements occurring in the beginning of the 90's with the appearance of robust face detection and face tracking systems proved to be essential for this field. Throughout this period the scientific community has grown to realize that automatic facial expression and emotion recognition would completely change the way we interact with computers, making them more warm and receptive to our emotions, or perhaps in a distant future, even express their own. Other than direct human-technology interaction, fields such as psychology, artificial intelligence, biometric and security systems could all benefit from improved recognition techniques.

Despite all the advances of the last two decades, computers are not currently capable of archiving a high recognition rate of facial expressions when inserted into a non-laboratory environment. The systems with higher recognition rates use 2D pictures of posed expressions at its apex. These systems are still highly sensitive to the recording conditions such as occlusions, illumination and texture variations. Along with this issue, when using 2D facial intensity images, it is necessary to maintain a constant pose, usually a frontal one, to archive good recognition rates, which is not always possible in real life applications. Which means, a single 2D view is unable to exploit all the information displayed by the face, once out-of-

plane changes of the face are hard to detect. Recently, to address most of these problems, 3D spatial data is being used, due to its robustness against uncontrolled environmental conditions.

The analysis of facial expressions in Psychology is essential for the success of an automatic system [8]. Psychologists concluded that the temporal evolution of facial actions plays an important role in interpreting sophisticated emotional states and can help distinguish between posed and spontaneous affective behavior. So in order to maximize the number of applications where a facial expression recognition system can be used, the time domain should be encoded and taken into account.

1.1 Proposed Work

With these two issues in mind, both the need of 3D spatial and temporal data, a four dimensional approach is followed in this work. Instead of using static images, our method to Facial Expression Recognition uses a 4D video database [9], with three spatial dimensions, provided by depth images, plus a temporal one. Moreover, all the faces in the database are normalized, both in terms of the head-pose and face size. This eliminates most of the problems in 2D spatial approaches, such as pose, illumination or texture variations.

To encode the temporal information, a dynamic or temporal texture descriptor is required. Taking into account the success of *Local Binary Patterns* from Three Orthogonal Planes (LBP-TOP) in previous applications [10], such as feature extraction from the 2D videos, this was the descriptor adopted. This descriptor is very simple to implement and has one of the lowest computational costs, whilst maintaining a high discriminative power.

With all the expressions labeled, at this phase a supervised learning method is required to learn the facial expressions data. Recently, *Random Forests* methods have started to become a popular and successful method to solve this problem. These methods are of simple understanding and implementation, are able to handle large amounts of redundant data with a reasonable amount of noise and number of outliers, and have a low computational cost. A *Random Forest* based method, *Hough Forests*, proposed in [11] by Juergen Gall *et al.* and successfully applied in Facial Expression Recognition [4], which happens to be our case, was the chosen method for the supervised learning process. Once the LBP-TOP was never used in Hough Forests, the adaptation of this descriptor for this learning process was one of the main contributions of this work.

With the choices explained above we propose a system for facial expression recognition. This system attempts to classify eight facial expressions including classes of happiness, sadness, surprise, fear, anger, disgust and contempt. Another class with

the subjects saying the phrase *"Yes, we can!"* was used. The temporal location of these expressions is going to be predicted also, allowing for the classification of the temporal stage of the expression as onset, apex or offset.

1.2 Main Contributions

The main contributions to the field of facial Expression Recognition by this work is the development of a Facial Expression Recognition system that uses 4D spatio-temporal information, due to the lack of systems that take this type of information into account. The successful processing of this information is essential for the evolution of these systems to real-life applications, since its very resilient against environmental variation.

During this development, another contribution of our work was the adaption of a powerful low-level histogram texture descriptor, the LBP-TOP to a successful supervised machined learning method, the *Hough Forests*. This was possible by creating a novel approach for the examination of these histogram as the *Hough Forests* are created and used. To our knowledge there is no existence of systems that use this type of descriptor in a *Hough Forest*.

1.3 Thesis Overview

We start this thesis by giving a general view of the State of the Art of Facial Expression Recognition systems and all the different parts involved in the process applied here. In the Chapter 3 all the theoretical principles used in this work are going to be explained in detail. The Chapter 4 addresses the recognition of facial expressions and their temporal segments, with a detailed explanation of the methods used thought out this work. The complete implementation and the algorithms used are covered in the fifth chapter. In the Chapters 6 and 7 we present the experiments and analysis that confirm our approach. Finally, the conclusion and future directions are discussed in the last chapter.

Chapter 2

State of The Art

Before the 1970s, the majority of facial expressions would be classified by relying on human observers to give their personal analysis, although, these observations are generally not accurate or reliable science. In 1978, Paul Ekman and Wallace V. Friesen published the Facial Action Coding System (FACS) [7], which was the first investigation of its kind. In this approach they created a set of Action Units, quantified individual facial configurations (e.g. AU 1 – raising the inner brow), that when combined can represent a complete facial expression. Due to the fact that these Action Units cannot be interpreted individually, they are used in Affect Recognition Systems for high-level decision-making, which can include the recognition of basic emotions and various affective or complex psychological states (suicidal depression or pain) [12]. Besides the spatial configuration of facial expressions, temporal dynamics also play an important role in the interpretation of emotions. For instance, psychologists have found that spontaneous and deliberate expressions differ in temporal dynamics and smoothness [13], meaning that temporal data is essential for interpreting subtle facial expressions [14].

Ekman and his colleagues also claimed that the interpretation and execution of facial expressions is ingrained in our brains and is universal, being independent of the race, sex or age [15]. He developed an affect model for classification with six basic emotions: happiness, sadness, surprise, fear, anger and disgust with the emotion contempt later being added to this model [16]. However, researchers believe that this model is restrictive in its ability to classify the wide range of everyday emotions. Currently, a non-basic affect model is being considered. Meaning that instead of using a discrete approach, a continuous and multi-dimensional affect design would be applied [17]. Even though there still no consensus about the correct method to model affections, the basic emotion model is more widely accepted and therefore most commonly used to classify expressions today.

The information that facial expression recognition systems need to encode from facial expressions and its emotional significance is supplied by the cognitive sciences,

such as psychology. In addition, *Computer Vision* and *Machine Learning* focus on how to encode and use that information to classify facial expressions. For this classification and based in Ekman's theories, two main procedural approaches are followed: judgment-based approaches, that attempt to directly classify the expressions from a set of emotional categories; and sign-based approaches that describe facial expression by the detecting facial action units. Usually, in both of these approaches, three fundamental components can be found: Facial Registration, Facial Representation and Recognition [8]. During this chapter we will cover the most popular methods for each one of these states.

2.1 Facial Registration

For the creation a facial expression database, the pixel data of the face needs to be recorded. This is done in the first stage, the Facial Registration. Here, a system needs to detect and track the face within its environment. Certain registration errors due to changes in head pose and illumination are usually removed here [18]. This stage, depending on the output, can be classified as holistic, componential or configural. For most of the systems holistic registration is used, since the whole face is covered and more information is available. The componential classification, when compared with the holistic approach, is more robust against differences in configuration of the face, because it covers only some individual facial components (e.g. eyes and mouth) without considering the spatial relations between parts. In the configural representation fiducial points are detected and mapped to create a spatial facial representation, making it very robust against configural information of the face. In all of these three approaches Active Appearance Models can be used for model fitting, reducing the errors caused by configural information of the face.

Another concern related with the creation of a database is the type of data that is being recorded. Data can be temporally static or dynamic, using two (2D) or three spatial dimensions (3D), with or without fiducial facial points, and containing spontaneous or posed expressions. The majority of the databases use 2D static images of the expression's apex or 2D videos. The technological progress made in the last decade, has allowed for the acquisition of 3D data accurate enough to record all the necessary details of facial expressions [19]. However, when 3D spatial data plus a temporal dimension (4D) is considered there is only a few databases with this type of data, since the 3D spatial data must be acquired at a considerable framerate. Therefore, systems that attempt Facial Expressions Recognition using 4D spatio-temporal data are not common [20]. Another alternative approach to the problem of the computational cost of 3D spatial data is to map this data into 2D representations, with the most popular method being a depth map of the 3D facial

meshes, originated from z values at each x and y position [21][22]. As an alternative 2D representation with APDI images were used in some systems [23].

Initially it may seems that the more data available in a database the better. However, all this information is complex to model and it is specially hard to obtain spontaneous facial expression, since the presence of sensor technologies necessary in the place make it impossible [8].

2.2 Facial Representation

The second component of facial expression recognition systems is Facial Representation. Here, features are extracted by converting pixel data into higher-level representations of the same data. The goal is to minimize variations of facial details within the class while maximizing it between classes and, typically eliminating registration errors caused by environmental conditions (e.g. illumination and color) [12] or configural facial information. This component can be classified based on the information that it encodes. When we encode frame-by-frame, time is not taken into consideration and, therefore we classify it as spatial. However, in a spatio-temporal representation, a neighborhood of frames is encoded and therefore time is considered [8]. Besides temporal classification, other methods use different types of features that are encoded in space, depending if appearance or geometric features are used. While appearance representation uses textural information directly from the pixel data and usually encodes low-level information, a geometric representation ignores the texture and explicitly describes the location of fiducial facial points as a shape, encoding more high-level information [18]. Currently, in static approaches, appearance representation is the most common, because it can pick up the necessary details of facial expressions. However, identity biases still a large issue for these low level representations. As spatial and appearance representations, descriptors such as *Local Binary Patters* (LBP [24], *Local Phase Quantization* (LPQ)[25] and *Histogram of Gradients* [26] are typically used as Low-Level Histogram representations. *Gabor Filters* [27] are also a popular approach for this component.

The majority of systems developed so far have attempted recognition of expressions from static data. However, more recent works employ dynamic data for this purpose. The features extracted for static and dynamic systems can differ greatly, due to the temporal nature of data [19]. Considering 2D spatio-temporal representations, the technique of extracting features from Three Orthogonal Planes (TOP) has become a very popular approach to adapt spatial representations to the spatio-temporal domain. Here appearance descriptors are preferred, such as LBP-TOP [10] and LPQ-TOP [18] in which experiments proved that these provide a better performance, when compared with their spatial version counterparts [28].

Although in most of the 2D approaches appearance representation is preferred, in 3D feature extraction the most common representations are shape based. Spatial representations are obtained by detecting fiducial facial points [29] and subsequently mapping the shape of the face. When spatio-temporal approaches are considered, these points also need to be tracked in time. Then the spatial changes of the face are used to learn different deformations [30][31]. The fiducial facial points can also be used in methods based in the fitting of *Morphable Models* [32][20], facial models that attempt to fit in certain emotional models.

In some facial expression recognition systems various descriptors and two or more facial representations are combined [33], obtaining very good results.

2.3 Recognition

In many approaches the Recognition component starts with a pre-stage called Dimensionality Reduction. Its main purpose is to increase the discriminant features between classes and reduce the redundancy of the data. This component can address numerous facial expression recognition problems, like illumination variations, identity bias and registration errors. Dimensionality reduction can be divided in to Feature Selection and Feature Extraction [34]. The purpose Feature Selection is to select a relevant subset from the facial representations and optionally weigh the selected subset [8]. Methods such as *AdaBoost* and *GentleBoost* [35] are the most commonly used. Feature Extraction is responsible to transform a subset into a lower dimensional space by selecting the regions or features of interest. If training data is used the transformation is classified as adaptive, or supervised. *Linear Discriminant Analysis* (LDA) [27] is the most popular technique. In comparison, as a non-adaptive, or unsupervised approach, *Principal Component Analysis* (PCA) [36] is the most used one.

For the Recognition process itself, a statistical model is applied to the transformed features or, in some cases, to the facial representation directly. The goal is to learn discriminant features that will allow for a high-level classification of the non-learned cases. Again the same challenges arise, such as illumination variations, registrations errors, head-pose variations, occlusions and identity bias. The statistical model used needs to address these problems. Usually methods used for 3D facial expression analysis are the same as in 2D. For instance models such as *Hidden Markov Models* [18], *Support Vector Machine* (SVM) [37], *Random Forests* [4], *Sparse Coding* [29] or *Dynamic Bayesian Networks* [38] are used. Boost technics and *Neural Networks* can also be used [8]. To improve prediction, a combination of these statistical models is possible [30]. To our knowledge the recognition of facial expressions with a *Hough Forest* approach, a *Random Forest* based method, was

only applied by G. Fanelli *et al.* in [4]. In their approach 2D facial videos were used for training and classification, both the facial expressions and their temporal segments were classified.

Chapter 3

Background Theory

The theory behind the methods we used to predict both the temporal moment and the class of an expression is going to be covered throughout this chapter. Starting with a brief psychological base for a better understanding of emotional classifications of a Facial Expression Recognition system. Following with an explanation of the 2D Facial representations available in our database, that are the *Depth Map* and the *Azimuthal Projection Distance Image* (APDI), along with the chosen method for the Facial Representation, *Local Binary Patters on Three Orthogonal Planes* (LBP-TOP), a dynamic or temporal texture descriptor. Finally, the Recognition stage is implemented with *Hough Forests* voting in a *4D Hough Accumulator*.

3.1 Psychological Basis

As stated in the previous chapters, Paul Ekman is considered to be one of the main contributors for the psychological theories used in Facial Expression Recognition. In order to attempt to classify expressions it is necessary to be very familiar with his work on the basic affect model, in which he grouped the facial expressions into seven basic emotions: happiness, sadness, surprise, fear, anger, disgust and contempt [15][16]. These are the facial expression we try to distinguish in our work.

Ekman also classified the temporal evolution of an expression, by using with four temporal segments: neutral, onset, apex and offset. During the Neutral segment the subject is expressionless, without noticeable signs of muscular movement. The Onset segment starts with the beginning of muscular activity and continues while this activity increases in intensity. Apex is the plateau of a facial expression, where the intensity usually reaches its maximum. This is the temporal phase that allows for a better discrimination in between different expressions. Lastly, the offset is the stage of muscular relaxation, where the facial activity decreases. In our work, we try to identify these temporal segments of expressions, by predicting its temporal moment, in addition to classifying the expression.

3.2 2D Representations of the 3D Facial Geometry

The most common 2D representation of the 3D facial geometry are *Depth Maps*. This is a very simple representation which makes it common for all kinds of Computer Vision applications. In this approach the image contains information that is linear and directly related with distance of the scene, from a certain viewpoint. This way, each pixel is originated from the Z coordinate depth value of the scene at each X and Y position, when the viewpoint referential is taken into account.

As an alternative representation, *Azimuthal Projection Distance Image* (APDI) can be used [1] and was chosen for this work. These images used the *Azimuthal Equidistant Projection* (AEP), used in Geography and Earth Sciences, that when adapted to 3D image processing are based on the normal vector to the 3D surface we want to represent. If we consider this normal, for each pixel, and its projection onto a *Euclidean* plane, a pixel is then represented by the absolute length of the projected line in this plane. This line starts in the center of the projection, the intersection of the normal vector with the plane, and ends in the projected terminal point of the vector, the AEP point.

Formally, considering that $n(i,j)$, with $0 \leq j \leq H$ and $0 \leq i \leq W$, denotes a normal-map (H the height and W the width of the map), where for the pixel (i,j) we have $n(i,j) = (n_x, n_y, n_z)$. This way the normal $n(i,j)$ is projected onto a *Euclidean* plane and the AEP point $n'(i,j) = (x', y')$ in this plane is defined as:

$$x' = k \cos\Theta(i,j) \sin\big(\phi(i,j) - \phi_0(i,j)\big) \tag{3.1}$$

$$y' = k(\cos\Theta_0(i,j) \sin\phi(i,j) - \sin\Theta_0(i,j) \cos\Theta(i,j) \cos\big(\phi(i,j) - \phi_0(i,j)\big) \tag{3.2}$$

with $k = \frac{c}{sin(c)}$, where c is defined as:

$$\cos(c) = \sin\Theta_0(i,j) \sin\Theta(i,j) + \cos\Theta_0(i,j) \cos\Theta(i,j) \cos\big(\phi(i,j) - \phi_0(i,j)\big) \tag{3.3}$$

Where $\Theta = \frac{\pi}{2} - \arccos(n_z)$ and $\phi = \arctan(\frac{n_y}{n_x})$ representing the elevation angle measured from the z-axis, and $\phi = \arctan(\frac{n_y}{n_x})$, the azimuth angle. Θ_0 and ϕ_0 are the elevation and azimuth angles of the reference normal, the normal to the *Euclidean* plane of the projections. In our case it is necessary to be able to directly compare the projection coordinates of neighbouring points. So at every point the reference

normal is set to $(\hat{n}) = (1, 0, 0)$. This way the distance calculated is always compared with this normal. Therefore $\Theta_0 = \frac{\pi}{2}$ and $\phi_0 = 0$. So 3.1 and 3.2 become:

$$x' = k\cos\big(\Theta(i,j)\big)\sin\big(\phi(i,j)\big) \tag{3.4}$$

$$y' = k\cos\big(\Theta(i,j)\big)\sin\big(\phi(i,j)\big) \tag{3.5}$$

With this the final image is created with the same dimensions as the normal map and the value for each pixel is given by the length of the project line by the following formula:

$$I(i,j) = \sqrt{x'^2 + y'2} \tag{3.6}$$

The length of the projected line and the elevation and azimuth angles can be seen in the Figure 3.1

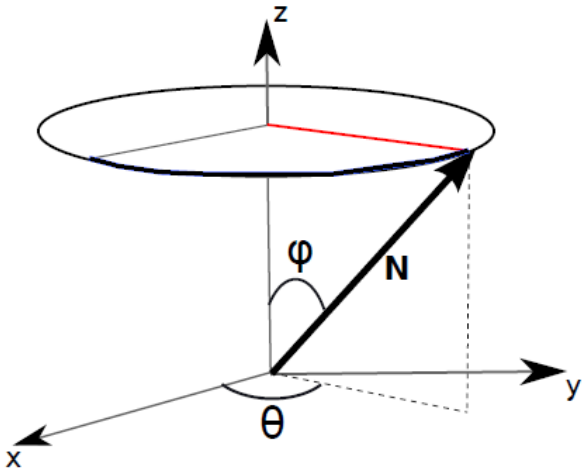

Figure 3.1: The projected line (red) and the elevation (Θ) and azimuth angles (ϕ)[1].

As example one example of each of these representation can be seen in the Figure 3.2.

3.3 Local Binary Patterns

The Local Binary Pattern (LBP) is a powerful operator for texture description. It was first presented by Ojala *et al.* in 1996 [39] and since then it has been successfully used in a variety of Computer Vision applications, due to the fact that it is highly discriminative, computationally efficient, and invariant to monotonic gray-level changes and rotations [2]. The LBP operator is obtained by comparing the value of a central pixel, a point, with a neighborhood of sampling points and assigning them a binary value. The value 1 is attributed to a neighbor pixel if its value is

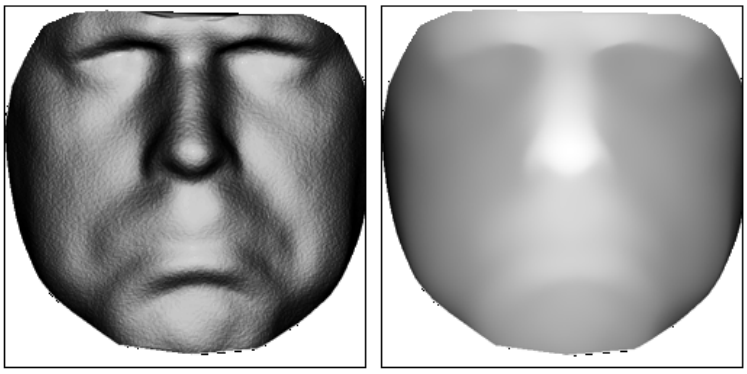

Figure 3.2: *Left*: An APDI image; *Right*: A Depth Map image.

higher than the central pixel, and 0 otherwise. The central pixel is then labeled with a binary number created by starting at one neighbor and then following a circular direction to the next ones. Formally, for a pixel the LBP operator is defined by the following equation:

$$LBP(x_c, y_c)_{N,R} = \sum_{p=0}^{N-1} S\big(I(x_p, y_p) - I(x_c, y_c)\big)2^p \tag{3.7}$$

with c referencing a central pixel and p its neighbours; while x and y are the pixel coordinates. Both N, the number of neighbors, and R, the distance to the central pixel, are the parameters of this operator. The function $I(x, y)$ returns the value of the pixel (x, y) and S is defined as follows:

$$S(x) = \begin{cases} 1 & x \leq 0 \\ 0 & x > 0 \end{cases} \tag{3.8}$$

As seen in Figure 3.3, depending on these parameters a subpixel point can be obtained, and its sampling is done by bilinear interpolation [2].

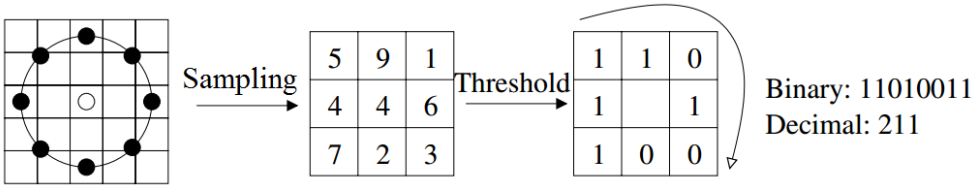

Figure 3.3: The sampling, threshold and creation of the central pixel decimal value [2].

This operator is applied to all the pixels in the image, with exception for those closer than R pixels to the image border. Finally, one or more histograms with 2^N bins are created to describe a texture.

3.3.1 Three Dimensional LBP

When the available texture has motion in the temporal domain, it is a better approach to use a dynamic texture descriptor that encodes this information. G. Zhao and M. Pietikïainen [3] considered all the advantages of the LBP and adapted it to a video, a volume of images, calling this new dynamic texture descriptor *Volume Local Binary Patterns* (VLBP). For this, the LBP operator was applied to three sequential frames, spaced by L number of frames. In this case the binary number of the central pixel in the central frame is obtained by comparing its value with not only the neighboring pixels in each of the three frames, but also the value of the two central pixels from the first and last frames.

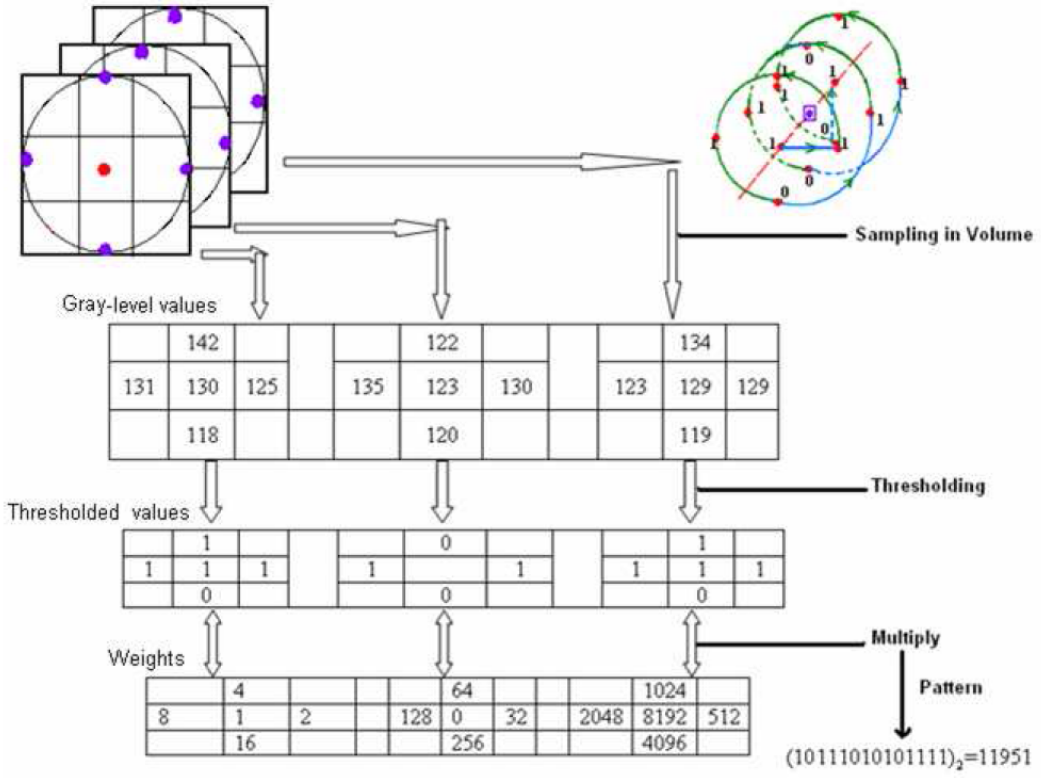

Figure 3.4: VLBP - Top: order of the sampled frames; Bottom: procedure and its weights [3].

Due to the fact that the weights of each neighbor point increases as we progress through the frames, as seen in the Figure 3.4, the number of possible patterns will be 2^{3N+2}, which is also the number of histogram bins, with N being the number of neighbors around a central pixel in one single frame. This rapid increase in the size of the histogram bins makes it hard to extend the VLBP to a large number of neighboring points, restraining its applicability [3].

To address this issue Zhao and Pietikïainen created a Local Binary Pattern on Three Orthogonal Planes to describe the temporal texture. Consider a video with

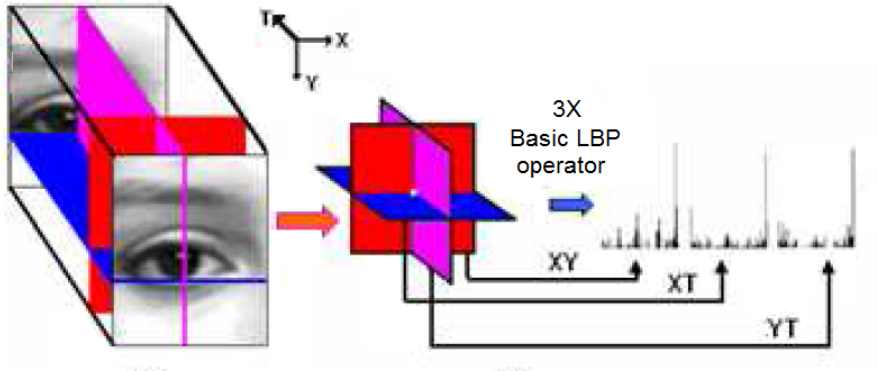

Figure 3.5: The procedure for the LBP-TOP [3].

three dimensions; where X and Y are the spatial dimensions and T is the temporal one. They then applied the LBP operator to the XY, XT and YT planes. In this case the two parameters of the LBP, R and N, operator are extended for these three planes and can be different for each one of them. After creating an independent histogram for each of these planes, they are concatenated into one histogram that describes a three dimensional texture, as seen in the figure 3.5. Since each one of the histograms are independent, and therefore the number of bins is going to be $3 * 2^N$, a much smaller number than the VLBP approach, which is a definite advantage. This can be seen in the Figure 3.6

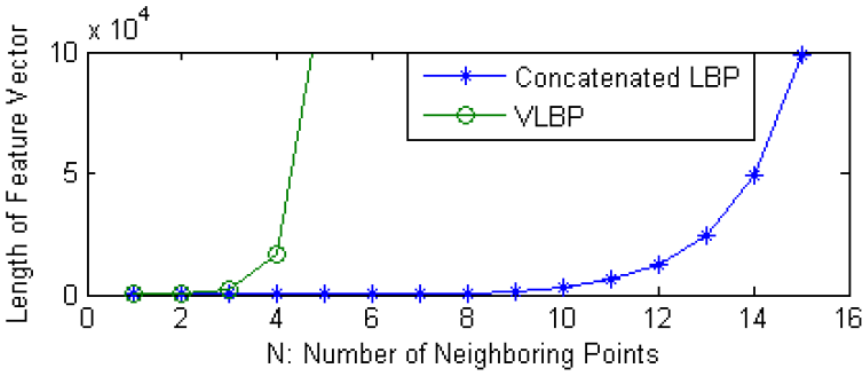

Figure 3.6: The size of the histogram versus the number of neighbours pixels [3].

Another advantage of this descriptor is that since the motion direction of the textures is unknown, the neighboring points in a circle on the three orthogonal planes encode the motion in all planes of the video and not only in the spatial plane (XY) [3]. This differs from the VLBP operator.

3.4 Random Forests

A *Random Forest* is a supervised ensemble learning method composed of a set of decision trees. It was first referred by Breiman [40] in 2001 and defined as "a classifier consisting of a collection of tree structured classifiers $\{h(x, \Theta_k), k = 1, ...\}$ where the $\{\Theta_k\}$ are independent identically distributed random vectors and each tree casts a unit vote for the most popular class at input x". The construction starts by presenting the tree with a set of categorized samples relevant to a classification task in an effort to create a decision map, with a top-down approach [41]. Usually each tree is constructed recursively starting from the root, by feeding a sub-set of samples to the first node. For each node a set of random binary tests, that are applicable to any sample, is assigned along with a group of samples. Each specific test, depending on its result, allows the processed samples to move to one of two child nodes, by the end splitting all the given samples between the two of them. From the set of tests applied, the one that splits the samples in the best way is selected for that specific node. The criterion to grade a test is usually application-specific. The process is repeated creating more subsequent child nodes until the depth (number of previous nodes) reaches the maximum value or the number of samples in the node is small enough. Then the terminal node is considered a leaf-node and stores the statistical information needed in the voting stage, based on the frequency of the samples that reached that leaf-node. After the construction, when a sample is unknown, all the trees cast a probabilistic vote, by testing that sample through series of nodes, until a leaf is reached, as seen in the Figure 3.7.

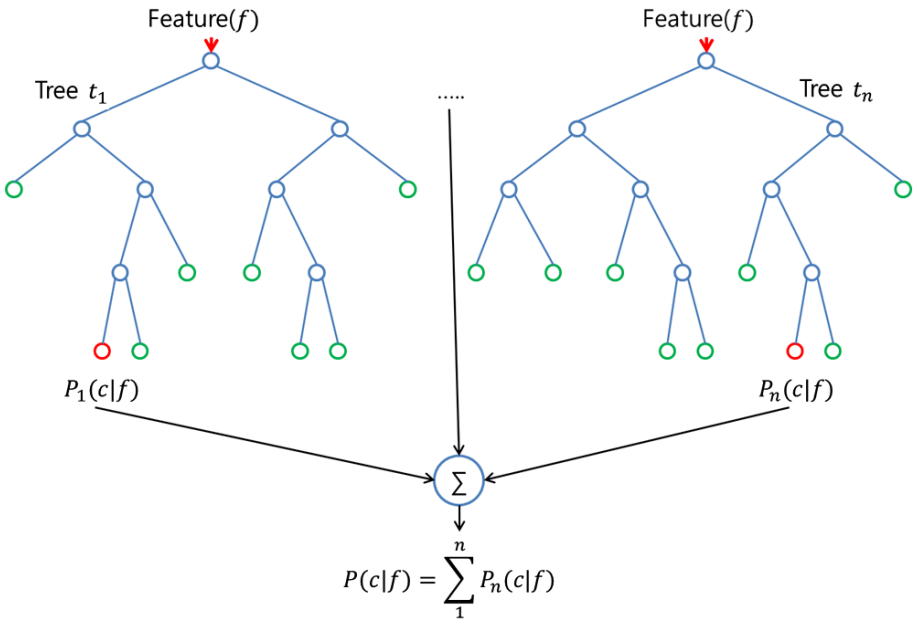

Figure 3.7: The voting process of a *Random Forest*, with n the number of trees.

Decision Trees are able to map a complex input space into a simpler one by splitting the original problem into smaller ones that are solvable with simple tests. Consequently it achieves a complex and highly non-linear mapping that selects the most discriminant information for a specific classification. Usually, for large amounts of learning data, this eliminates the need of an extra dimensionality reduction step. [42].

It was proved [40] that assembling several trees together and training them in a random manner accomplishes a higher generalization and stability when compared to a single decision tree. This randomization is introduced by training a tree with a random subset of learning data and a random subset of binary tests, from all possible, for each non-leaf node. To keep the training of a tree efficient, the tests should be simple and selected in a way that clusters the training data. This allows the creation of simple models that make good predictions [42, 11]. As stated in [40] by Breiman, Random Forests are relatively robust to outliers and noise; give useful internal estimates of error, strength, correlation and the importance of the variables and are simple and easily parallelized. These characteristics make them a successful learning method in various Computer Vision applications [4, 42, 11].

3.5 Hough Forests

In the last years, Random Forests has become a very popular method in Computer Vision for applications such as action recognition, object detection, etc. Usually they are trained with image patches to be used as discriminative codebooks for image categorization or semantic segmentation. Therefore there is no geometric information stored at the leaves, but only class labels. Some other implementations do take this geometric information into account, but address the problem as pure classification one, which is insufficient for our case.

To improve the Random Forest approach Juergen Gall *et. al* [11] proposed a method based in the generalized *Hough Transform*. Instead of learning an explicit codebook of patch appearances, a direct mapping between the appearance of an image patch and the probabilistic Hough vote of a tree is used. Basically, the outputs of the trees are votes in a continuous space with several dimensions. Since these dimensions can be either discrete or continuous, this approach cannot be classified as a standard supervised learning classification or regression problem since both of these can be addressed.

When compared with other supervised learning approaches, there are some advantages to the *Hough Forest* method. It is able to handle very large and high-dimensional training datasets without a substantial overfitting and is very efficient in the testing phase, because the number of possible outcome leaves for the sample

decreases logarithmically it runs through the tree. Moreover this method can even tolerate a reasonable amount of errors in the training data, discarding the need of pixel-accurate segmentation of this data. Lastly, trees are built in effort to decrease the entropy of the set of image patches as we move towards the leaves by choosing the right criterion to grade a node's test. This causes the tree to produce probabilistic votes with small uncertainty.

Chapter 4

Recognition of Facial Expressions and their Temporal Segments

As we have seen, both the 3D spatial and the temporal information of facial expressions are very important for the application of Automatic Expression Recognition systems in real-world applications. The 3D spatial shape of the face is very robust again illumination and texture variations, and eliminates the need of a constant head pose, when compared with 2D spatial data. The temporal information of facial actions plays a key role when interpreting sophisticated or subtle emotional states and can help distinguish between posed and spontaneous affective behavior. Also, this temporal domain is less affected by identity bias problems, when compared with static representations of the face, since the changes in the face are tracked over time. These static representations are not suitable for real-life situations, because generally they only represent the apex of the expression, and therefore, do not allow for the detection of subtle spontaneous expressions or the classification of an expression at an early or late stage.

With the problems stated above in mind, we propose a judgment-based approach for Facial Expression Recognition system, using a 4D spatio-temporal facial expression registration represented by a temporal texture descritor. The goal of this system is the classification of facial expressions into one of seven basic emotion labels or a phrase and their temporal segments (onset, apex or offset stages). These temporal segments are referenced to the apex moment of a certain expression by its offset vector. Considering a temporal moment of an expression, its offset vector is given by that temporal value minus the value of the expression's apex.

To construct a Facial Expression Recognition system we follow the same procedure that was used in [4] and [43]. Therefore, assuming that a set of training and testing facial expressions have the respective labels and all the faces cropped and aligned. Each of these expressions is represented by a video of a 2D representation of their 3D surface. These videos are sampled in smaller spatio-temporal patches, a

video block, that is represented in the figure 4.1.

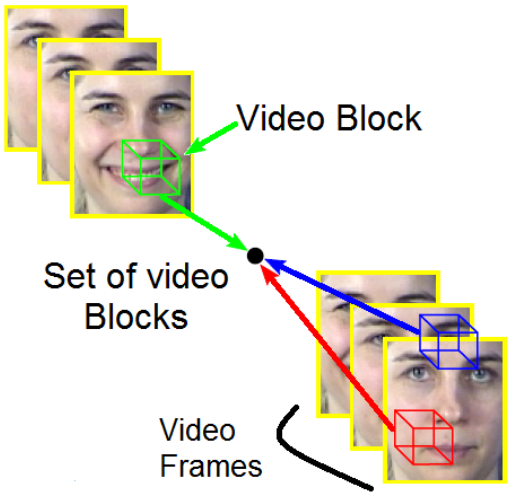

Figure 4.1: Representation of a video block and a set of blocks from various expressions [4].

Each one of these blocks is labeled with the expression and the offset vector where it came from. If a set of facial expressions is sampled, obtaining a set of video blocks, a supervised machine learning method, in our case a *Hough Forest*, can map the visual appearance of each one of these blocks into its class and offset vector. Contrary to [4] the visual appearance of a block is represented by the three LBP-TOP accumulative histograms of that block. To characterize an unknown block in relation to its class and offset vector a tree can vote in a four dimensional *Hough* space. One dimension is for the class of this block and the three others for the offset vectors, since a video has three dimensions, as seen in the Figure 4.2. Considering a set of blocks from a specific temporal moment of an unknown facial expression video, this *Hough* space is going to accumulate the votes from all blocks given by the trees of the *Hough Forest* (*4D Hough Accumulator*). This way this Hough space is a probabilistic map for the possible class and offset vector of the unknown expression. The absolute maximum of this Hough space is considered to be the final prediction for the expression's class and offset.

As stated above the visual appearance of a block is represented by the three accumulative histograms of the texture descriptor LBP-TOP applied to that block. Since this feature descriptor is different from the ones used in [4], it is necessary to create a new method to test a block once it arrives at a node of a tree. The description of this test is done is the next section of this chapter (4.1). In the second section of this chapter the construction of *Hough Forests* and the voting process of the *4D Hough Accumulator* are going to be addressed with more detail.

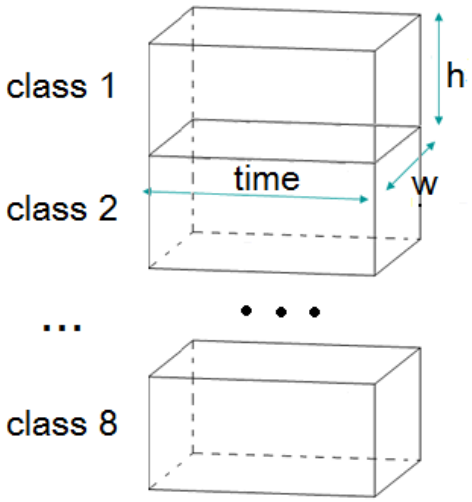

Figure 4.2: The four dimensions of the *Hough* space: Time, height and width of the video plus the classes.

4.1 Testing a Video Block

Since the LBP-TOP features presented in a block differ from the approaches that guide us [4], this descriptor need to be adapted to be used in the Hough Forests. So a new method for testing a block in a node needed to be implemented. This new test needed to be simple and with a low computational cost to ensure a computationally efficient tree construction. Initially, considering that the appearance of a block is represented by three normalized histograms, one for each plane of the video. These histograms are then transformed into a function of cumulative probability. After that, we randomly select one of these histograms, represented by the function $y = H_i(x)$, with $i = \{1, 2, 3\}$. Two random values, t_X and t_Y, that will refer to the X and Y coordinate of this histogram are generated. These values vary from zero to one, once the histogram is normalized and if the output of $H(t_X)$ is smaller than t_Y ($H(t_X) < t_Y$) the test's output is zero, otherwise it is one. Formally, as represented below.

$$T(H_i) = \begin{cases} 1 & H(t_X) \geq t_Y \\ 0 & H(t_X) < t_Y \end{cases} \tag{4.1}$$

This way three random values are generated for each test: the selected histogram consisting of a discrete value from one to three, and the two values referring the histogram's coordinates, t_X and t_Y, two continuous values from zero to one. An example of this test can be seen in the Figure 4.3.

The test created allows for the use of a low-level histogram based texture descrip-

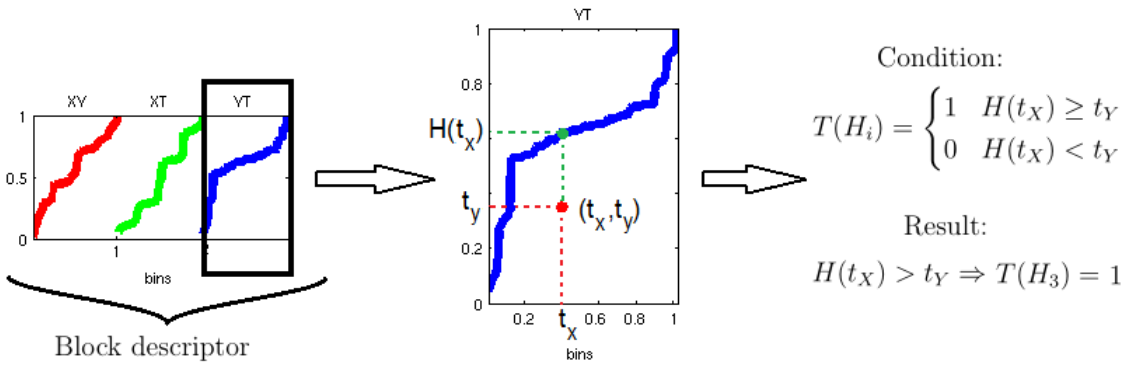

Figure 4.3: An example of the designed test, with the result 1.

tor, the LBP-TOP, in the supervised machine learning method proposed by Juergen Gall in [11], the *Hough Forests*, explained in detail through the next section.

4.2 Constructing a Hough Forest

The construction of a *Hough Forest* follows the common *Random Forest* structure. Consider a set of image patches, from the training data, as samples to start a decision tree represented by $\{Pi = (A_i, c_i, d_i)\}$, where c_i is the class label of the patch, d_i is the offset of a patch from a reference point and A_i is the appearance of the patch. This appearance is characterized by one or more channels of extracted features from a patch and is represented by $Ai = (I_{1i}, I_{2i}, ...I_{Ci})$, where each I_{ji} is a single feature channel and C is the number of channels. During the training of a tree, each non-leaf node is assigned with a subset of patches and large pool of random binary tests, each binary test represent by $t_{(Ai)}$. Depending on the test that is chosen, this subset is then split into two smaller subsets that are passed to two new child nodes. The best test should be picked so that both the class label and the offset vector uncertainties decrease towards the leaves. For this, quantifying these uncertainties in a set of patches, S, is essential. Considering S, the class label uncertainty is measured through the entropy of the classes within a set defined by the formula:

$$U_1(S) = |S| \cdot \sum_c Entropy(\{c_i\}) \tag{4.2}$$

To define the offset vector uncertainties, a simple mean of the quadratic error is enough and is defined by the formula:

$$U_2(S) = \sum_{i=1}^{N} (d_i - d_S)^2 \tag{4.3}$$

where d_S is the mean offset vector over all object patches in the set and N the number of patches. The binary test that minimizes the sum of the two children's subset uncertainty is considered the best one. The uncertainty that is minimized in each non-leaf node is randomly decided, alternating the nodes that decrease the class-label uncertainty with the nodes that decrease the offset uncertainty. This allows for the sets that reach the leaves to have low variations in both class labels and offsets. Each one of these leaves, L, stores p_c^L, the proportion of patches per class that reached the leaf, i.e. $\sum_c p_c^L = 1$, and $D_c^L = \{d_i\}_{c_i=c'}$, the offset vectors for each class c.

4.2.1 The Voting Process

To predict the class and offset vector of an unknown case, patches of video are densely sampled from the test case. Each one of these patches is passed through the trees and the leaves that they arrive in are used to cast votes, both for the class and offset vector. Now, by assuming that the bounding box of the patches is fixed during both training and testing, the only parameter that defines these patches is its centroid. So considering a patch, $P(y) = (A(y), c, d(y))$, centered in y, with c and $d(y)$ the class and offset vector to predict, respectively. $Q_c(x)$ is a random event denoting the existence of a patch that belongs to the class label c and has the offset vector x. With this, in the voting process, we are interested in finding the conditional probability $p(Qc(x)|A(y))$, the probability of a patch having a certain class and offset vector knowing its appearance, $A(y)$. This probability can be decomposed as:

$$\begin{aligned} p\big(Q_c(x)|A(y)\big) &= \sum_{l \in C} p\big(Q_c(x)|c(y)=l, A(y)\big) \times p\big(c(y)=l|A(y)\big) \\ &= p\big(Q_c(x)|c(y)=c, A(y)\big) \times p\big(c(y)=c|A(y)\big) \\ &= p\big(d(c,y)|c(y)=c, A(y)\big) \times p\big(c(y)=c|A(y)\big) \end{aligned} \tag{4.4}$$

Both factors of this equation can be estimated by passing the patch $P(y)$ through the trees. If a patch ends up in a leaf L of the tree T, the first factor of the equation can be approximated as the Parzen-Window estimate of D_c^L, the offset vector for a class c. The second factor is approximated by p_c^L, the probability of the patch belonging to class c. Therefore, by replacing these factors, the last equation applied

for a single tree T can be written as:

$$p\big(Q_c(x)|A(y),T\big) = \left(\frac{1}{D_c^L}\sum_{d\in D_c^L} G\big((y-x)-d\big)\right)\cdot p_c^L \tag{4.5}$$

Where G is a 3D *Gaussian Parzen Window* function. With this we can apply this equation to all the trees, calculating the average over all of them. The equation for the forest, F, is then defined as:

$$p\big(Q_c(x)|A(y),F\big) = \frac{1}{|F|}\sum_t p\big(Q_c(x)|A(y),T_t\big) \tag{4.6}$$

This equation defines the probabilistic vote for a single patch. Considering all the patches, a *Hough Accumulator* integrates all of their votes and is defined as:

$$V(x,c) = \sum_{y\in S(x)} p\big(Q_c(x)|A(y),F\big) \tag{4.7}$$

The local maximum of the Hough accumulator is calculated to predict the class and the offset vector of the unknown case, as seen in the figure 4.4. A whiter pixel, represents a higher probability.

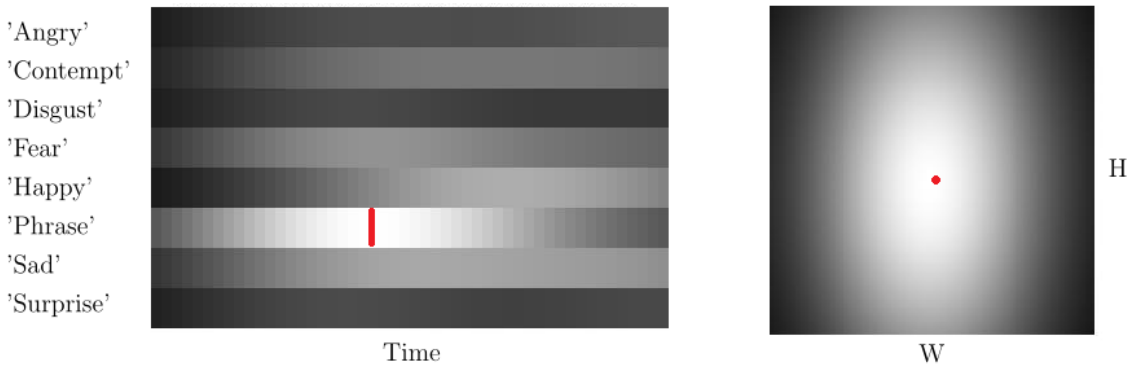

Figure 4.4: *Left:* Example of a Hough voting space reduced to the two dimensions, expression class and time; *Right:* A frame from the temporal maximum. The absolute maximum is marked in red.

To finish, some considerations should be taken into account. First, note that through out the voting process, the classes are treated independently, while the offset vectors are smoothed by the Gaussian window. Therefore this supervised learning method cannot be categorized as either a classification or regression one, but a combination of both. Secondly, when the learning data has a time dimension, time-scale invariance can be achieved by using different sampling densities, or rates of sampling, for the data in that dimension. By using different sample densities for the same case, different frame rates can be simulated. These samples are then

learned by the same Hough Forest, archiving time invariance. Although, the *Hough Forests* have some tolerance to variances built in which was enough for the facial expression we used, so we did not consider multiple time scales.

Chapter 5

Implementation

In this chapter, the details of our Facial Expression Recognition approach are going to be discussed. Our implementation can be divided into three parts: Data Reading from the database, Features Extraction using LBP-TOP and Learning and Classification with Hough Forests.

5.1 Data Reading from the Database

To construct our Facial Expression Recognition system, the format of the data available in the database needs to be taken into account. The two videos with 50 frames of both Depth Maps and APDI images available allows for a selection of the best fitting format, which in this work is the APDI video. This type representation allows for a better discrimination from the dynamic texture descriptor used, the LBP-TOP, due to the fact that facial textures are more apparent in APDI images, as seen in the figure 5.1.

For each one of the videos in the database, the face is cropped, aligned (without a tilted head-pose) and all the surrounding information other than the facial area is removed, being replaced by white pixels. The feature extractor cannot encode these pixels, since they are not facial information. Ultimately, a mask for each frame was made, creating a 2D video mask that identifies the facial area. To create this mask, the border between the face and the blank pixels is clearly identified due to the distinct contrast. Each frame was then converted to a binary image using a high threshold, but one that is lower than the white pixels of the surroundings. Then the pixels in the face that were higher than this threshold value were filled so that the mask should only be false (0) when out of the face. An image of a mask from a frame can be seen in the Figure **??**.

The next step of the data reading is the resizing of each frame, from 500x500 pixels to 100x100 pixels, for a lower computational cost, without losing significant facial information. After reading the video, the apex frame of each expression was set

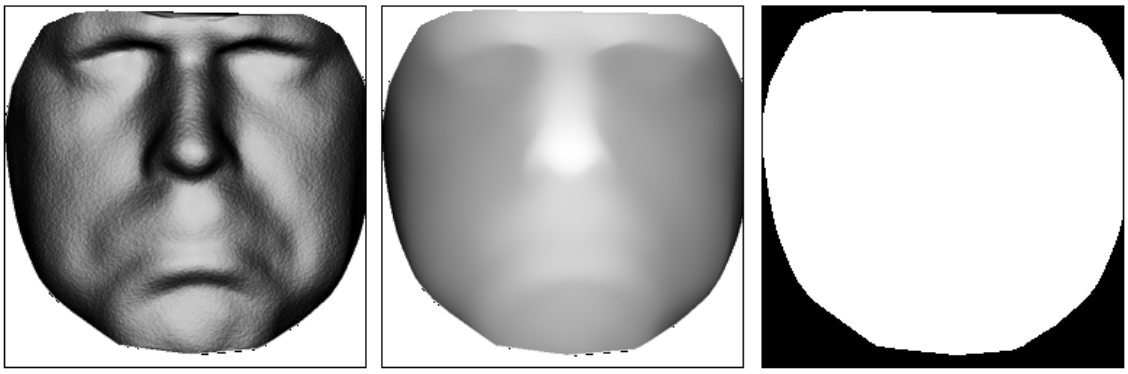

Figure 5.1: *Left:* APDI image; *Center:* Depth Map image; *Right:* a frame mask.

manually for each expression. The pixel with the coordinates $X = 50$ and $Y = 50$ of the apex frame is the reference coordinate for the offset vector we want to predict.

5.2 Feature Extraction using LBP-TOP

The Feature Extraction from the APDI videos was achieved using a dynamic texture descriptor, Local Binary Patterns on Tree Orthogonal Planes of the video. To use this descriptor the implementation of Zhao and Pietikäinen [35][47] was adapted to our system. Their implementation was created to output three LBP histograms from a block of video, one for each plane. In our approach, the video is not divided in a grid of pre-defined blocks; instead, all the pixels of a video could be chosen randomly to serve the central pixel of a block. Once the size of these blocks is fixed, this pixel is enough to reference a block. It is too computationally expensive and unnecessary to compute the LBP-TOP feature histogram for each possible block before the creation of the trees. As an alternative, the LBP-TOP value of each pixel is calculated for the whole video and stored beforehand in an LBP-TOP video with three channels per pixel, one for each video plane, as seen in the Figure 5.2. Then, the histograms of the randomly selected blocks for a specific tree are calculated online, during the creation of this tree. This means that only the histograms of the blocks used by a tree are calculated. In comparison with the implementation of Zhao and Pietikäinen the output of our implementation is a LBP-TOP video, rather than a set of three histograms.

Another issue of Zhao and Pietikäinen's implementation was that their method is not ready to receive a 2D video mask identifying the facial area, while ignoring the white pixels in the surroundings of the face, during the calculation of the LBP's. Consequently we modified the technique so that if any of the pixels used in a LBP operation from any video plane are out of the mask, this operation is ignored and those pixel channels are set to -1. This way, all the pixels in a block that do not

belong to the facial area are ignored during the creation of the histograms.

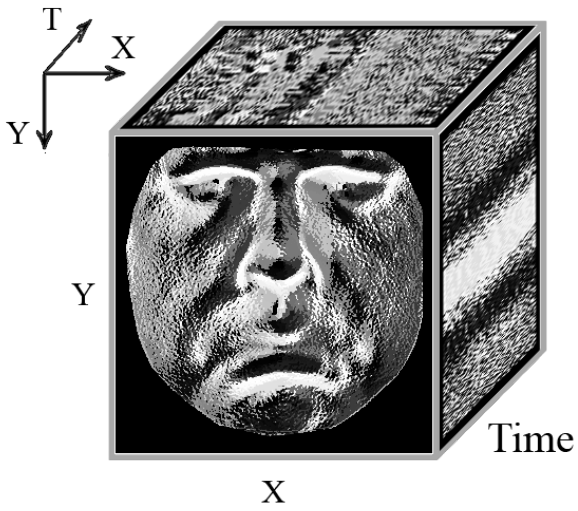

Figure 5.2: A grey-scale representation of the LBP values in the XY, XT and YT planes of the LBP-TOP video.

5.3 Learning and Classification with Hough Forests

The learning and classification stage is carried out by the Hough Forest method, and implemented throughout this work based on [32] and [40]. The first step is to randomly select a percentage of all the cases that are going to be learned by the trees, while leaving the other cases to be used in the test phase. For a certain tree, each of the learning cases is randomly sampled to obtain a certain number of blocks. The LBP-TOP cumulative histogram is then calculated in all the blocks from all the cases, using the LBP-TOP video. For a better comparison in between blocks, these histograms are normalized to one. The tree is then initialized, passing all the sampled blocks to the first node. For this node and each of the subsequent ones, a set of random tests is generated and the subset of blocks is split accordingly to each one of these tests. A test is rejected if the split subsets are smaller than a certain specified number, in order to obtain a good statistical power in each leaf node. The accepted tests are graded using the method in [40] which allows a test to be weighed with respect to the entropy of its classes or offset vectors, depending on a random decision at each node. The test's grade is the sum of the two grades of the subsets that were split by that test. For each one of these subsets, the entropy is calculated by one of the formulas below, depending if the class' (Eq. 5.1) or the offset vector's (Eq. 5.2) entropy is being evaluated.

$$U_1(S) = -|S| \cdot \sum_c p_c \ln(p_c) \tag{5.1}$$

$$U_2(S) = \sum_i \|d_i - d_S\|^2 \tag{5.2}$$

Where S is the set of patches that is being evaluated, d_S the mean distance of the set, c a certain class, p_c the proportion of a certain class in the set and i the i^{th} sample of the set.

The best (or most fit) test is the one with the smaller sum. This test is then used and the subsets created by it are passed to the child nodes. When none of the tests in a node can split its set into subsets that are large enough to obtain a good statistical power, that node is considered a leaf node. In our case, a maximum depth was not established since better prediction was our aim, despite the slightly higher computational power required.

Once again, the depth of a node is the number of previous nodes in the tree. Our algorithm processes the nodes by increasing depth order. Therefore the nodes with a certain depth are all processed before their child nodes. When none of the nodes produce children, the tree is over, advancing to the next one. The whole algorithm can be seen in the Algorithm 1.

The classification is then accomplished with a 4D *Hough Accumulator*. Here the cases that were not used to train the trees are classified. For that we start by dense sampling an unknown case that was chosen for classification. The LBP-TOP cumulative histogram is calculated for all the sampled blocks. Then, each one of these blocks, with unknown class, c_i, and offset vector, d_i, is passed through all of the trees, being split according to the binary tests previously established in the non-leaf nodes and, depending on the leaf reached, L, a vote per tree is obtained. A vote is composed of eight groups of offset vectors, one per class of emotion, $D_c^L = \{d_i\}_{\in c}$, and the proportion of each class that reached the leaf, p_c. In the voting process, a single vote is processed by accumulating the votes of each offset vector though the tree dimensional space of its class dimension. The votes in this three dimensional space are smoothed by a 3D *Gaussian Parzen Window*, as seen in the equation below:

$$V(x,c) = \left(\frac{1}{D_c^L} \sum_{d \in D_c^L} G\big((y-x)-d\big) \right) \cdot p_c^L \tag{5.3}$$

Where y is the position of the central pixel of each vote's block, x and c are the spatio-temporal and class dimensions of the Hough Space, repectivly, with $x \in \mathbb{R}^3$.

The process is repeated for all the votes of all the sampled blocks and the absolute maximum of the Hough Accumulator is the final prediction for the all four dimension, one to classify the class and three to the offset vector of the unknown case.

Data: LBP-TOP Videos and Expression Labels
Result: Hough Forest

```
for Tree = 1 to Number Of Trees do
    Randomly select the Blocks from the Learning Cases;
    Calculate the Blocks' Histograms;
    Initiation of the first node;
    for Depth = 1 to Tree's Maximum Depth do
        for Node = 1 to Number of Nodes do
            Load the node's Blocks;
            Randomly create all of the node's tests;
            Do all the tests for all of Blocks;
            Divide the set of Blocks depending on the Tests;
            Validate and grade all the Tests;
            if No valid tests then
                The node is a leaf-node;
                Save the proportion of blocks per class;
                Save the offset vector per class;
            else
                The node is a non-leaf node;
                Save the best test;
                Create the two child-nodes with Depth = Depth + 1;
            end
        end
        if No child-nodes at Depth + 1 then
            Break;
        end
    end
end
```

Algorithm 1: Creation of a *Hough Forest*

Data: LBP-TOP Videos and *Hough Forest*
Result: Hough Forest

```
Randomly select the Blocks from the unknown case;
Calculate the Blocks' Histograms;
for Each Block do
    for Each Tree do
        Assign the first node to the Block;
        for Depth = 1 to Tree's Maximum Depth do
            if The assigned node is a Leaf then
                Save the Leaf's vote (D_c^L, p_c^L);
            else
                Test the block with the test of the assigned node;
                if Test's result is 0 then
                    Assign the first node's child to the block;
                else
                    Assign the second node's child to the block;
                end
            end
        end
        for Each offset vector (x) do
            for Each class (c) do
                Add the vote's proportion to the coordinate (x,c) of the voting
                Hough Space;
            end
        end
    end
end
Check the absolute maximum of the voting Hough Space;
```

Algorithm 2: Voting in a Hough Space for an unknown Facial Expression.

Chapter 6

Experimental Results

After the implementation of our Facial Expression Recognition system, it is necessary to experimentally verify its results. The procedures used for this verification and its results are covered thought this chapter.

The database we used, [9], has 192 expressions available, divided equally between six people. Each one of individuals performs four repetitions for each expression, with the documented emotions being happiness, sadness, surprise, fear, anger, disgust, contempt and the phrase "Yes, we can!". A single case is represented by a 2.5 seconds APDI video with 100x100 pixels, at 20 frames per second. In each expression, the subject starts with a neutral face, flowing with the onset, apex and offset stage of each expression, as seen in the Figure 6.1.

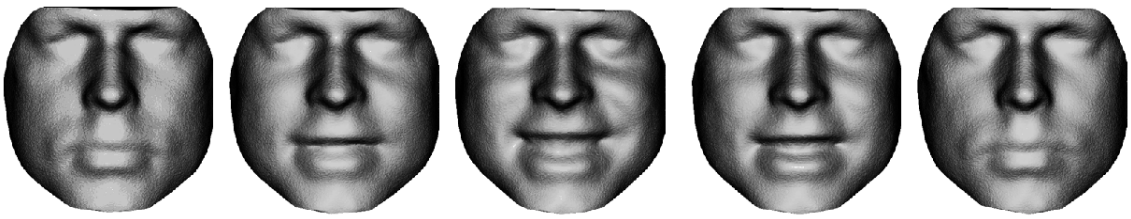

Figure 6.1: Five subsequent APDI Images from a Happy Facial Expression.

6.1 Tuning of the Parameters

After the reading of all the videos and their respective classes, the next step is the tuning of the parameters within the whole algorithm. In our case, this was one of the harder tasks, due to the fact that a full run of the algorithm is a long process and there are many parameters to tune. Meaning that the performance effects of changing a certain parameter are difficult to identify and modify in a short period of time making the whole process very time consuming.

The first parameters that are tuned belonged to the descriptor we used, the

LBP-TOP. The possible parameters are R, the radius, and N, the number of neighbors. The authors of this descriptor already used these in facial expression recognition applications in [10] and tested the descriptor with various configurations, obtaining $R = 3$ and $N = 8$ as the best parameters for the three orthogonal planes of the video. These values were then used in our experiments.

For the construction of the *Hough Forests*, four parameters could be set. Hypothetically, the number trees, when increased, should make the success of the classification better until a certain point, and then stabilize. The number of tests per node and the number of samples per expression should also both increase the success of the experiments until a certain value, where the success stops having major improvements. As for the size of a block, that has three values (X,Y and Z), the only way to decide the best dimensions is by trial and error, since this value is very case specific and depends, for example, on the size of the facial features or the framerate of an expression. Therefore, these four parameters should be decided by gradually changing one of them, while keeping the other ones constant. When the increase of a parameter does not majorly improve the success of the classification, this value is considered to be the best. This adjustment guarantees a good performance with a low computational cost. Besides tuning our algorithm, these experiments are also useful to investigate the influence of each of the parameters in the success of our approach.

The parameters for the voting stage are the number of samples per tested expression and the variance of the 3D *Gaussian Parzen Window*. The latter is referred to be $var = 9$ by the authors [11] of the Hough Forests and used in our approach. For the number of samples per tested expression a much higher value that the learning stage should be considered, since dense sampling is necessary, so we considerer double the number of samples, when compared with the learning stage.

This way, the experiments to tune the algorithm are preformed by randomly selecting five different training and testing sets of expressions that were maintained through the tuning experiments, allowing for more stable and comparable results. During this phase 85% of the expressions are used to train the trees while the remaining are used for testing purposes. We run each of these training and testing sets with all the proposed values for all four parameters involved in the construction of the Hough Forest. Then, 100 tests are performed, their accuracy is registered and the mean of the five sets is shown from 0% to 100%. Therefore, based in the pre-experiments during the development of our approach, the following four base parameters were determined to be: 200 for the number of samples per expression; 100 for the number of tests per node; 5 for the number of trees and $(X, Y, T) = (200, 40, 7)$ for the size of the video block. The first parameter to be varied, while keeping the other three constant, is the size of the blocks and as stated above

this parameter was chosen by trial and error. The influence of the T coordinate was very small and considered always seven. The Figure 6.2 shows the success of our experiments based in different block sizes. The complete table can be seen in Appendix A.1.

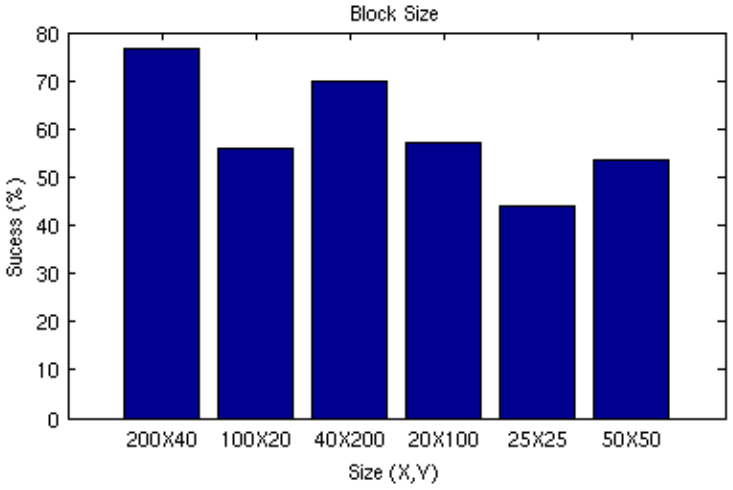

Figure 6.2: The obtained Success versus the Block Size.

For the other three parameters the same five training and testing sets are used, with different increasing values for the respective parameter. The relation between these values and the mean of the five sets of expressions accuracy is shown in the Figures 6.3, 6.4 and 6.5.

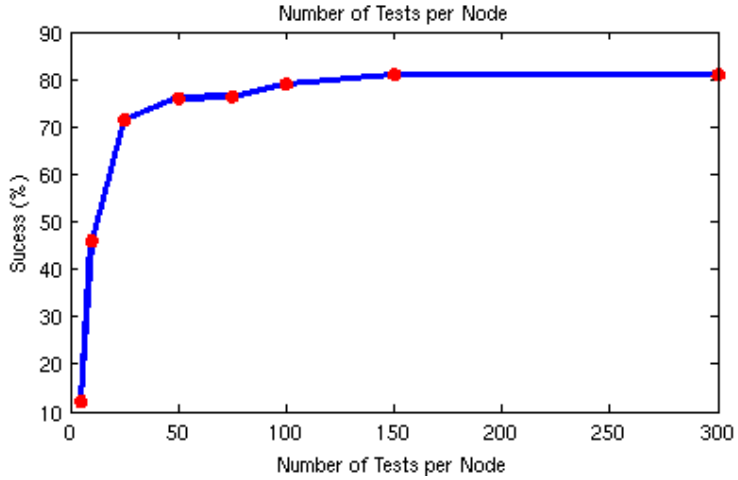

Figure 6.3: The obtained Success versus Number of tests per node.

Taking into account the values which do majorally improve the success of the classification, we select the flowing parameters: 300 for the number of samples per expression; 150 for the number of tests per node; 10 for the number of trees and $(X, Y, T) = (200, 40, 7)$ for the size of the video block. These values are the result of

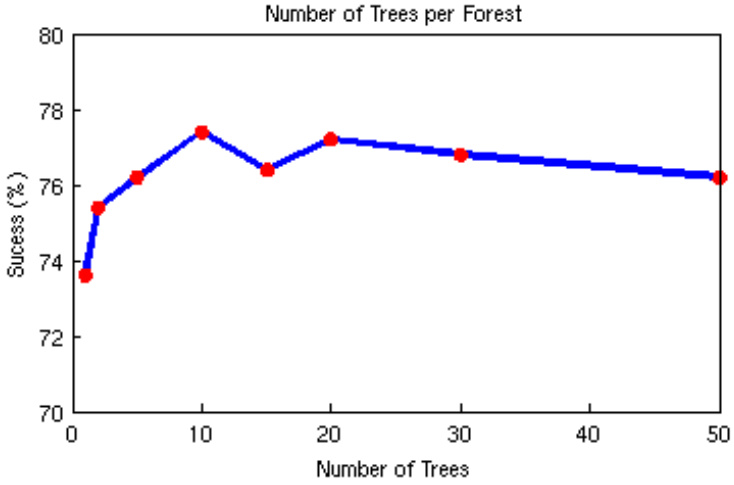

Figure 6.4: The obtained Success versus Number of Trees in a Forest.

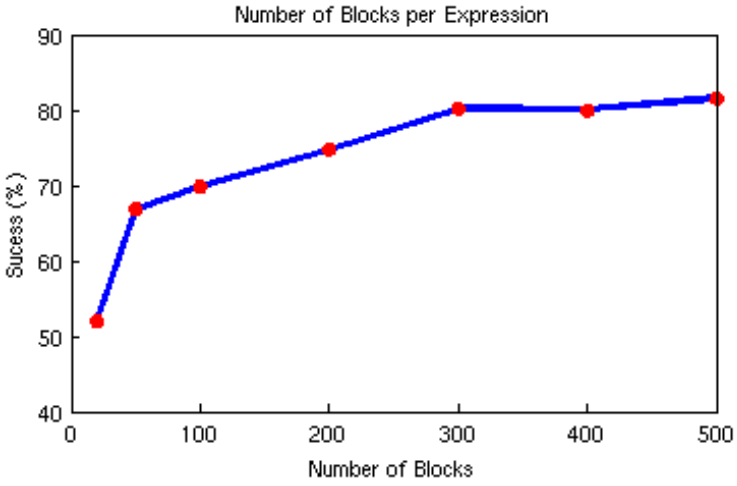

Figure 6.5: The obtained Success versus Number of Blocks sampled per Expression.

our tuning and are the ones that obtain the best relation between success rate and computational cost.

6.2 Final Results

After the tuning, the definitive experiments of our approach can be tested. For this we need to test the success of both the expression classification and the prediction of the spatio-temporal offset vector. The experiments of the later one are addressed in the next section. In the Section 6.2.2 the emotional classification is presented.

6.2.1 Temporal Classification

The experiments used to test the success of the spatio-temporal offset vector, start by randomly selecting five training and testing sets, with 90% training cases. During the testing phase the only samples that are fed to the trees are selected from a specific temporal area by sampling blocks from a specific frame. This way we associated the stages initial neutral, onset, offset and final neutral with the frames -20, -10, 10 and 20 respectively, and in relation to the apex frame, the zero frame. The apex was considered a stage itself as well. Then, the success of the temporal prediction is measured by the proportion of predicted frames that fall in a certain interval. With these intervals being -25 to -16, -15 to -5, -4 to 4, 5 to 15 and 16 to 25 for the same respective stages. Also, the absolute error of the difference between the frame that is predicted and the sampled frame is registered. This is done for 50 testing expressions and the mean and standard deviation of this error is measured. The success of the emotional classification is measured for each one of these states too. The average of the five testing sets are showed below:

Stage Name	Stage Int.	Avg. Emotion Class. Succ.	Avg. Temp. Success	Avg. Pred. Error	Avg. Std. of Pred. Error
Ini. Neutral	[-25 -14]	14.0%	12.4%	5.24	3.19
Onset	[-15 -5]	56.0%	32.8%	5.27	2.65
Apex	[-4 4]	91.2%	81.2%	2.51	2.15
Offset	[5 15]	67.0%	31.8%	5.69	3.49
Fin. Neutral	[16 25]	33%	13.2%	8.04	3.77

Table 6.1: Results of the temporal experiments. The errors are measured in frames.

6.2.2 Emotional Classification

To ensure success of the expression classification, the goal is to obtain the highest possible classification rate for the eight classes present in the database. For this the

parameter selected to be the most fit during the training phase is used. Once again, five training and testing sets, with 90% training percentage, are randomly selected. The average classification of the five sets was 87%.

To increase the success of our approach while considering the results of the temporal tests we only sample the testing expressions between -10 and 10 frames in reference to the apex. This temporal interval has more discriminative features than the beginning or the end of an expression. This frame interval proved to have the best success rate for expression classification.

	Avg. Success	Set 1	Set 2	Set 3	Set 4	Set 5
No Temporal Selection	0.87	0.78	0.96	0.79	0.9	0.94
With Temporal Selection	0.93	0.92	0.98	0.9	0.93	0.94

Table 6.2: Final results, with and without temporal selection.

Next we present the confusion matrix for both cases, with and without temporal selection. This matrix represents the relations between the ground true emotions, in the first column, and the classified emotions, in the first line.

	An	Co	Di	Fe	Ha	Ph	Sa	Su
An	61.1	0	15.3	0	0	0	16.7	6.9
Co	0	100	0	0	0	0	0	0
Di	0	0	97.1	0	0	0	0	2.9
Fe	0	2.9	2.9	79.1	0	0	0	14.9
Ha	0	0	0	0	90	10	0	0
Ph	0	0	0	0	0	94.1	0	5.8
Sa	0	7.3	0	0	0	0	92.7	0
Su	0	0	0	11.1	0	0	0	88.8

Table 6.3: Confusion matrix of the run with temporal selection.

	An	Co	Di	Fe	Ha	Ph	Sa	Su
An	71.4	0	9.5	0	0	0	19.0	0
Co	0	100	0	0	0	0	0	0
Di	0	0	89.4	0	5.2	5.2	0	0
Fe	7.6	0	11.5	65.4	3.8	0	7.7	3.8
Ha	0	0	0	0	100	0	0	0
Ph	0	0	0	0	0	95.5	0	4.5
Sa	7.1	7.1	0	0	0	0	85.7	0
Su	0	0	0	0	0	0	0	100

Table 6.4: Confusion matrix of the run without temporal selection.

Chapter 7

Experimental Analysis

In this chapter we present a experimental analysis of the results that were shown in the previous chapter. The variations of the *Hough Forest*'s parameters, both the temporal and the emotional classification are analysed through the next two sections.

7.1 The Hough Forest's Parameters

The tuning process addressed in the previous chapter, other than the optimization of our system, allowed us to study the influence of the Hough Forest parameters in the success rate of our system.

As we can see in Section 6.2, the block size has a big effect on the classification success rate, as expected, since the shape and size of the block directly influences the histograms of our texture descriptor (LBP-TOP). Contrary to [4] where the best block was a vertical rectangle that covered almost half of the face, in our approach the best block had horizontal orientation, also covering half almost of the face.

As we increase the number of executed tests per node, the success rate archives a satisfactory value (72%) with only 25 tests per node, a surprisingly low value. We decided to select 150 tests per node for the final results, which increased the success 10 percentual points, and was the value at which the success rate stops increasing despite the added number of tests, as seen in the Figure 6.3.

The number of trees per *Hough Forest* surprisingly does not heavily influence the success rate of our algorithm, archiving the highest rate with 10 trees and then actually slightly decreasing the success rate as the number of trees was increased. This can be due to the statistical variance through the algorithm, since a multitude of random variables are involved. This variance can be as seen in the Figure 6.4 when the success decreased from 10 to 15 trees and then increased again for 20 trees. This is why 10 trees per *Hough Forest* were selected as the value for the

final results. When compared with [4], where only 5 trees per forest were used, our approach needs more trees to archive maximum success rate.

As we increase the number of blocks that were sampled in each one of the Facial Expressions used in the learning stage, contrary to [4], the success rate of our method increases, continuing until 300 Blocks per expression, the value we selected for the number of blocks during the final results.

7.2 Final Results

The temporal tests performed in the last chapter proved that the success rate of our approach increases immense as we classify an expression with blocks sampled from a temporal area closer to the apex. This agrees with the psychological theory of the face, since the most discriminative phase of a facial expression is at the apex. When the blocks are sampled from a temporal area further from of the apex, the appearance of these blocks are more similar to different classes, decreasing the success of our system, in both the temporal stage and the emotional classification. For the same reason both the error and the standard deviation of this error decrease as we sample the expression closer to the apex.

Due to the lack of systems that used Facial Expressions with 3D spatial and temporal information and the absence of Facial Expression databases with this type of information in which the faces were normalized (centered, cropped and aligned) is was not possible to directly compare our system with other approaches. Although when compared to [4], which achieved a maximum of 87.1%, our approach obtained a similar success rate of the expression class, with 87% of the expressions being correctly recognised. Other approaches that did not use *Hough Forests*, such as [20] and [44], that applied *Deformable Models* and *Expressive Maps* respectively, both obtained 90.4% success rate with other databases. Since the facial expressiveness is higher closer to the apex, when we consider Facial expression with only 10 frames before and after the apex our success rate increased to 93%, confirming that the beginning and the end of an expression are less descriptive in regards to their emotional class.

When we look into the confusion matrix its clear that both Angry and Fear are the hardest class to identify. The Expression Angry is often confused with Disgust or Sadness and the expression Fear was mostly confused with Surprise and vice-versa. Both of these results are acceptable since these expressions have similar appearances, even for humans. The confusion in between these facial expressions was encountered by other systems as well [8].

Chapter 8

Conclusions and Future Work

Automatic Facial Expressions Recognition systems are quite far from operating in real-world applications. During this work we proposed a Facial Expression Recognition system to approximate these systems to the real-world by considering 3D spatial and temporal facial Expressions. The 3D spatial information is more robust to environmental conditions, such as illumination, texture and pose variations, while the temporal information takes into account the temporal changes that happen in the face, allowing for an early and more realistic detection of emotions. The success of our method is comparable with the State of the Art for systems that take temporal information into account, and allows for the detection of an expression before or after its apex.

8.1 Future Work

Facial Expression Recognition systems are still in an infant stage, if we consider its final goal the applicability to real-word applications. They face a multitude of computer vision problems, most specifically relating to the variation of the environmental conditions. Therefore, as a future approach, high-level shape representations seem to be the trend, since they play an important role in the human vision, as argued by the cognitive sciences. They are also less sensitive to identity bias and environmental conditions, such as variations of illumination and texture, but are more sensitive to other problems such as registration errors. Recently, 3D registration tries to address most of these problems, but the heavy computational cost and the elevated price of an adequate sensor technology with enough speed to take the temporal modeling into account, essential for real world applications [19], make this method unapplicable to the real world. However, instead of using only shape representations, another alternative would be the use of various representations and weighing them based in their reliability [8].

Spontaneous expressions are of vital importance for correct human emotion mod-

eling through facial expressions. In an ideal system both posed and spontaneous expressions should be learn so that this system can distinguish them. The gathering of these spontaneous expressions is another problem of Facial Expression databases, since the actual sensor systems with the necessary precision are to bulky and distract the subjects. A possible approach would be to provoke a certain emotion and record the facial expressions without the knowledge of these subjects. Although its not an easy process to ensure that all the subjects feel the desired emotions or even provoke these feelings, especially in a laboratorial setting. This means that it is necessary to wait for technological developments that will allow for the extraction of facial expressions from the real world.

Appendix A

Tables of Results

In this section we present the complete results for all the learning and testing sets of facial expressions presented during the Chapter 6.

(X, Y)	Mean	Set 1	Set 2	Set 3	Set 4	Set 5
(200, 40)	0.768	0.89	0.71	0.74	0.7	0.8
(100, 20)	0.56	0.78	0.38	0.58	0.64	0.42
(39, 199)	0.698	0.81	0.64	0.68	0.67	0.69
(21, 99)	0.572	0.67	0.48	0.58	0.66	0.47
(27, 27)	0.44	0.62	0.32	0.38	0.57	0.31
(49, 49)	0.534	0.7	0.35	0.53	0.61	0.48

Table A.1: The results of the Block Size variations versus the Success Rate for the 5 testing sets.

No. of Tests	Mean	Set 1	Set 2	Set 3	Set 4	Set 5
5	0.12	0.2	0.11	0.05	0.45	0.19
10	0.46	0.69	0.26	0.43	0.59	0.35
25	0.713	0.85	0.58	0.71	0.7	0.62
50	0.76	0.91	0.6	0.77	0.68	0.78
75	0.763	0.91	0.63	0.75	0.71	0.79
100	0.79	0.9	0.71	0.76	0.72	0.79
150	0.81	0.92	0.75	0.76	0.76	0.83
300	0.81	0.92	0.75	0.76	0.73	0.82

Table A.2: The results of the Number of Tests per Node versus the Success Rate for the 5 sets.

No. of Trees	Mean	Set 1	Set 2	Set 3	Set 4	Set 5
1	0.736	0.87	0.62	0.73	0.73	0.73
2	0.754	0.9	0.63	0.77	0.72	0.75
5	0.762	0.89	0.66	0.78	0.71	0.77
10	0.774	0.9	0.68	0.77	0.7	0.82
15	0.764	0.88	0.68	0.75	0.7	0.81
20	0.772	0.92	0.67	0.75	0.72	0.8
30	0.768	0.89	0.72	0.75	0.72	0.76
50	0.762	0.89	0.67	0.75	0.7	0.8

Table A.3: The results for the number of Trees in a Forest versus the Success Rate for the 5 sets.

No. of Blocks	Mean	Set 1	Set 2	Set 3	Set 4	Set 5
20	0.52	0.72	0.42	0.48	0.59	0.4
50	0.668	0.84	0.56	0.75	0.64	0.55
100	0.698	0.88	0.63	0.75	0.67	0.56
200	0.748	0.9	0.69	0.77	0.69	0.69
300	0.802	0.92	0.8	0.77	0.75	0.77
400	0.8	0.92	0.81	0.75	0.73	0.79
500	0.816	0.92	0.81	0.77	0.76	0.82

Table A.4: The results of the Number of Sampled Blocks per Expressions versus the success for the 5 sets.

Set	Stage Name	Stage Int.	Avg. Emotion Class. Succ.	Avg. Temp. Success	Avg. Pred. Succ.	Avg. Std. of Pred. Error
Set 1	Ini. Neutral	[-25 -14]	0.1	0.08	4.36	2.593
	Onset	[-15 -5]	0.64	0.46	4.74	2.028
	Apex	[-4 4]	0.9	0.76	1.9	1.298
	Offset	[5 15]	0.82	0.22	5.3	3.215
	Fin. Neutral	[16 25]	0.3	0.2	8.68	3.216
Set 2	Ini. Neutral	[-25 -14]	0.04	0.1	7.66	4.663
	Onset	[-15 -5]	0.56	0.3	5.7	2.866
	Apex	[-4 4]	0.84	0.8	1.94	1.659
	Offset	[5 15]	0.6	0.32	6.98	4.187
	Fin. Neutral	[16 25]	0.34	0.2	7.8	4.262
Set 3	Ini. Neutral	[-25 -14]	0.24	0.12	5.38	2.955
	Onset	[-15 -5]	0.4	0.2	6.4	3.534
	Apex	[-4 4]	0.9	0.84	2.72	2.373
	Offset	[5 15]	0.64	0.3	5.78	3.333
	Fin. Neutral	[16 25]	0.44	0.08	7.76	4.023
Set 4	Ini. Neutral	[-25 -14]	0.08	0.22	4.38	2.702
	Onset	[-15 -5]	0.66	0.28	5.94	2.780
	Apex	[-4 4]	0.96	0.82	2.667	2.503
	Offset	[5 15]	0.8	0.50	5.62	3.811
	Fin. Neutral	[16 25]	0.34	0.12	8.74	3.890
Set 5	Ini. Neutral	[-25 -14]	0.24	0.1	4.42	3.038
	Onset	[-15 -5]	0.56	0.4	3.6	2.090
	Apex	[-4 4]	0.96	0.84	3.333	2.944
	Offset	[5 15]	0.52	0.25	4.76	2.911
	Fin. Neutral	[16 25]	0.24	0.06	7.24	3.497

Table A.5: The results of the Temporal Error, its Standard Deviation and the Success Rate for the 5 sets.

Bibliography

[1] A. Seck, B. Tiddeman, and H. M. Dee, "Multi-scale azimuthal projection distance image of normal maps for 3d facial skin texture analysis," in *BMVC Student workshop*, 2013.

[2] T. Ahonen, A. Hadid, and M. Pietikainen, "Face description with local binary patterns: Application to face recognition," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 28, pp. 2037–2041, Dec 2006.

[3] G. Zhao and M. Pietikainen, "Dynamic texture recognition using local binary patterns with an application to facial expressions," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 29, pp. 915–928, June 2007.

[4] G. Fanelli, A. Yao, P. Noel, J. Gall, and L. J. V. Gool, "Hough forest-based facial expression recognition from video sequences," in *Trends and Topics in Computer Vision - ECCV 2010 Workshops, Heraklion, Crete, Greece, September 10-11, 2010, Revised Selected Papers, Part I*, pp. 195–206, 2010.

[5] A. Mehrabian, "Communication without words," *Psychology Today*, vol. 2, no. 4, pp. 53-56, 1968.

[6] R. Highfield, Wiseman, and R. Highfield, "How your looks betray personality," *New Scientist*, vol. 2695, 2009.

[7] V. Bettadapura, "Face expression recognition and analysis: The state of the art," *CoRR*, vol. abs/1203.6722, 2012.

[8] E. Sariyanidi, H. Gunes, and A. Cavallaro, "Automatic analysis of facial affect: A survey of registration, representation and recognition," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. PP, no. 99, pp. 1–1, 2014.

[9] ISR, "4d facial dynamics database," 2012. [Online; accessed 4-February-2015].

[10] G. Zhao and M. Pietikainen, "Experiments with facial expression recognition using spatiotemporal local binary patterns," in *Multimedia and Expo, 2007 IEEE International Conference on*, pp. 1091–1094, July 2007.

[11] J. Gall and V. Lempitsky, "Class-specific hough forests for object detection," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pp. 1022–1029, June 2009.

[12] M. Pantic, "Machine analysis of facial behaviour: Naturalistic and dynamic behaviour," *Philosophical Transactions of Royal Society B*, vol. 364, pp. 3505–3513, 2009.

[13] J. Cohn and K. Schmidt, "The timing of facial motion in posed and spontaneous smiles," *International Journal of Wavelets, Multiresolution and Information Processing*, vol. 2, pp. 1 – 12, March 2004.

[14] Z. Ambadar, J. Schooler, and J. Cohn, "Deciphering the enigmatic face: The importance of facial dynamics to interpreting subtle facial expressions," *Psychological Science*, 2005.

[15] P. Ekman, *Emotions inside out: 130 years after Darwin's the expression of the emotions in man and animals*. Annals of the New York Academy of Sciences, New York Academy of Sciences, 2003.

[16] A. Ortony and T. J. Turner, "What's basic about basic emotions?," *Psychological review*, vol. 97, no. 3, p. 315, 1990.

[17] H. Gunes and B. Schuller, "Categorical and dimensional affect analysis in continuous input: Current trends and future directions," *Image Vision Comput.*, vol. 31, pp. 120–136, Feb. 2013.

[18] B. Jiang, M. Valstar, B. Martinez, and M. Pantic, "A dynamic appearance descriptor approach to facial actions temporal modeling," *Cybernetics, IEEE Transactions on*, vol. 44, pp. 161–174, Feb 2014.

[19] G. Sandbach, S. Zafeiriou, M. Pantic, and L. Yin, "Static and dynamic 3d facial expression recognition: A comprehensive survey," *Image and Vision Computing*, vol. 30, no. 10, pp. 683–697, 2012. 3D Facial Behaviour Analysis and Understanding.

[20] Y. Sun and L. Yin, "Facial expression recognition based on 3d dynamic range model sequences.," in *ECCV (2)* (D. A. Forsyth, P. H. S. Torr, and A. Zisserman, eds.), vol. 5303 of *Lecture Notes in Computer Science*, pp. 58–71, Springer, 2008.

[21] S. Berretti, B. B. Amor, M. Daoudi, and A. D. Bimbo, "3d facial expression recognition using SIFT descriptors of automatically detected keypoints," *The Visual Computer*, vol. 27, no. 11, pp. 1021–1036, 2011.

[22] N. Vretos, N. Nikolaidis, and I. Pitas, "3d facial expression recognition using zernike moments on depth images," in *Image Processing (ICIP), 2011 18th IEEE International Conference on*, pp. 773–776, Sept 2011.

[23] G. Sandbach, S. Zafeiriou, and M. Pantic, "Binary pattern analysis for 3d facial action unit detection," in *Proceedings of the British Machine Vision Conference (BMVC 2012)*, (Guildford, UK), September 2012.

[24] T. Ahonen, A. Hadid, and M. Pietikainen, "Face description with local binary patterns: Application to face recognition," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 28, pp. 2037–2041, Dec 2006.

[25] T. Ahonen, E. Rahtu, V. Ojansivu, and J. Heikkila, "Recognition of blurred faces using local phase quantization," in *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, pp. 1–4, Dec 2008.

[26] T. Gritti, C. Shan, V. Jeanne, and R. Braspenning, "Local features based facial expression recognition with face registration errors," in *Automatic Face Gesture Recognition, 2008. FG '08. 8th IEEE International Conference on*, pp. 1–8, Sept 2008.

[27] H. bo Deng, L. wen Jin, L. xin Zhen, and J. cheng Huang, "Hong-bo deng, lian-wen jin, li-xin zhen, jian-cheng huang a new facial expression recognition method based on local gabor filter bank and pca plus lda a new facial expression recognition method based on * local gabor filter bank and pca plus lda," 2005.

[28] B. Jiang, M. Valstar, B. Martinez, and M. Pantic, "A dynamic appearance descriptor approach to facial actions temporal modeling," *Cybernetics, IEEE Transactions on*, vol. 44, pp. 161–174, Feb 2014.

[29] Y. Lin, M. Song, D. T. P. Quynh, Y. He, and C. Chen, "Sparse coding for flexible, robust 3d facial-expression synthesis," *Computer Graphics and Applications, IEEE*, vol. 32, pp. 76–88, March 2012.

[30] M. Valstar and M. Pantic, "Fully automatic recognition of the temporal phases of facial actions," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 42, pp. 28–43, Feb 2012.

[31] F. Tsalakanidou and S. Malassiotis, "Robust facial action recognition from real-time 3d streams," in *Computer Vision and Pattern Recognition Workshops, 2009. CVPR Workshops 2009. IEEE Computer Society Conference on*, pp. 4–11, June 2009.

[32] L. Yin, X. Wei, P. Longo, and A. Bhuvanesh, "Analyzing facial expressions using intensity-variant 3d data for human computer interaction," in *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, vol. 1, pp. 1248–1251, 2006.

[33] T. Senechal, V. Rapp, H. Salam, R. Seguier, K. Bailly, and L. Prevost, "Facial action recognition combining heterogeneous features via multikernel learning," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 42, pp. 993–1005, Aug 2012.

[34] P. Pudil and J. Novovicova, "Novel Methods for Subset Selection with Respect to Problem Knowledge," *IEEE Intelligent Systems*, vol. 13, pp. 66–74, Mar. 1998.

[35] E. Owusu, Y. Zhan, and Q. Mao, "A neural-adaboost based facial expression recognition system.," *Expert Syst. Appl.*, vol. 41, no. 7, pp. 3383–3390, 2014.

[36] J. Nicolle, V. Rapp, K. Bailly, L. Prevost, and M. Chetouani, "Robust continuous prediction of human emotions using multiscale dynamic cues," in *Proceedings of the 14th ACM International Conference on Multimodal Interaction*, ICMI '12, (New York, NY, USA), pp. 501–508, ACM, 2012.

[37] T. Wu, M. Bartlett, and J. R. Movellan, "Facial expression recognition using gabor motion energy filters," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*, pp. 42–47, June 2010.

[38] Y. Tong, J. Chen, and Q. Ji, "A unified probabilistic framework for spontaneous facial action modeling and understanding," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 32, pp. 258–273, Feb 2010.

[39] T. Ojala, M. Pietikäinen, and D. Harwood, "A comparative study of texture measures with classification based on featured distributions," *Pattern Recognition*, vol. 29, pp. 51–59, Jan. 1996.

[40] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, pp. 5–32, Oct. 2001.

[41] J. R. Quinlan, "Induction of decision trees," *MACH. LEARN*, vol. 1, pp. 81–106, 1986.

[42] G. Fanelli, M. Dantone, J. Gall, A. Fossati, and L. V. Gool, "Random forests for real time 3d face analysis," *International Journal of Computer Vision*, August 2012.

[43] A. Yao, J. Gall, and L. Van Gool, "A hough transform-based voting framework for action recognition," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pp. 2061–2068, June 2010.

[44] O. Ocegueda, T. Fang, S. Shah, and I. Kakadiaris, "Expressive maps for 3d facial expression recognition," in *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pp. 1270–1275, Nov 2011.