# MULTIMODAL RECOGNITION OF EMOTIONS

# MULTIMODAL RECOGNITION OF EMOTIONS

Proefschrift

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus, prof. dr. ir. J. T. Fokkema,
voorzitter van het College voor Promoties,
in het openbaar te verdedigen op
27 oktober 2009 om 15.00 uur
door

Dragoş DATCU
Informatica Ingenieur
geboren te Slatina, Roemenië

Dit proefschrift is goedgekeurd door de promotoren:

Prof. dr. H. Koppelaar
Prof. dr. drs. L. J. M. Rothkrantz

Samenstelling promotiecommissie:

| | |
|---|---|
| Rector Magnificus, | voorzitter |
| Prof. dr. H. Koppelaar, | Technische Universiteit Delft, promotor |
| Prof. dr. drs. L. J. M. Rothkrantz, | Technische Universiteit Delft, promotor |
| Prof. dr. C. M. Jonker, | Technische Universiteit Delft |
| Prof. dr. M. A. Neerincx, | Technische Universiteit Delft |
| Prof. dr. I. Văduva, | University of Bucharest |
| Prof. dr. M. Novák, | Czech Technical University |
| Prof. Ing. V. Matoušek, | University of West Bohemia |

Typeset by the author with the LaTeX typesetting system.
On the cover: Face image sample from Cohn-Kanade database.
Printed in The Netherlands by: Wöhrmann Print Service.

# Contents

# Chapter 1

# Introduction

## 1.1 Emotions and every-day life

Emotions play an important role in every-day life of human beings. Our social behaviour is quintessentially based on communication, both in terms of language and non-verbal interaction. Information from the environment determines individuals to mutually interpret other persons' intentions, goals, thoughts, feelings and emotions and to change the behaviour accordingly. The term emotion stands for a concept that has been proved difficult to define. As claimed by the famous naturalist Charles Darwin, emotions emerged in the course of evolution as the means by which living creatures determine the significance of certain conditions to meet their urgent needs. The more developed and complex life organization is, the richer all sorts of emotion states developed by people are.

In real life, emotions detect changes in the environment and direct the attention to significant novel events. Positive and negative emotions reflect the readiness to act appropriately and to adapt to certain life contexts. Emotions represent a way to categorize events according to the emotion responses. A possible function of emotions is to express the extent to which the information processed will be of consequence to the individual.

Different approaches for analysing emotions focused on essentially biological reactions through interpretive conventions with little or no biological contributions. The emotions most commonly acknowledged as basic are sadness, anger, fear, happiness, surprise and disgust. These are considered to be primary emotions and are the most studied categories in the literature. The more complex emotion categories could be represented by cultural conditioning or association combined with the prototypic forms of emotion. Emotions are naturally conveyed by interpersonal communication channels set on all human interactive modalities (sound, sight and touch). In this process, individuals constantly produce and evaluate non-verbal signals which relate to facial and vocal expressions, body gestures and other physiological manifestations. Mehrabian [131] arguably stated that the facial expressions account with 55% to the overall message of the speaker, while the verbal and vocal components show contributions of only about 7% and 38% respectively. This supposition gives a rough indication over the fact that expressive contents in voice and face are two of the most important supports for transmitting emotions. The human face plays a significant role in verbal and non-verbal communication. As it was previously acknowledged in a scientific manner by Ekman [57] and others, facial expression represents one of the most important and immediate means for human beings to transmit feelings, intentions and persuasions. Knowledge about the activity of facial muscles allows for a complete characterization of one's facial display while facial expressions emerge. On the other hand, the syntactic and semantic meanings of utterances are not all speech conveys. The research in speech analysis classically set to find answers to the questions "What is being said?" and "Who says that?" which correspond to the tasks of speech recognition and speaker identification. More recently, the new question of "How that is being said?" has gained more attention from the research community.

## 1.2 Emotions and computers

Nowadays, computer systems already influence significant aspects of our lives, from audio-visual network communication to means of transportation, industry and production. Based on the scientific knowledge about life and environment, we have constantly attempted to improve the work conditions firstly by reducing the workload of the human factor with the functionality of operational automatic systems. The every-day life benefits now from the availability of systems to ease the access to information and facilitates the assistance of individuals in carrying out different activities.

What is the connection between emotions and computerized systems? In a paper from 1995 [149], Rosalind Picard introduced "affective computing" as a term to describe computing that relates to, arises from or influences emotions. According to this, machines should interpret the emotion state of humans and adapt their functional behaviour to them, giving the appropriate response for specific emotions. The process is not only considered to provide better performance in assisting humans but also to enhance computers' ability to make optimal decisions. Thoughts are directly correlated with the emotions felt by the user. Specific pattern recognition models that may facilitate the task of reading thoughts show very high complexity due to the potentially huge number of possibilities. As a consequence, it is extremely difficult for a computer to recognize the internal thoughts of the user. Estimating emotion states seems to be rather more tangible because of the relatively small number of simplified categories of emotions. Instead of measuring the emotion states or thoughts directly, computers may rather analyse observable functions of these states. Observations over functions of hidden emotion states may be used to infer the states themselves. The mechanism may be optimal during the generation of voluntary expressions and may also provide fundamental evidence on the true emotions during involuntary expressions. For processing emotions, the computers should first acquire ambient perceptual and contextual information related to the activity and state of the user. Secondly, the data should be automatically analysed for estimating the internal emotion state.

The adoption of such user emotion-centred technology is essential, with multiple social and economical implications. Computers that are capable of reading emotions may enhance the current human-computer interfaces through valuable affect interaction sessions by providing proper feedback to the user. Picard suggests that, by providing the users with information on their emotion state, such computer systems may boost the communication process. It could also help humans to improve communication skills. Consequently, emotion-enhanced human computer interfaces would also acquire a higher rate of acceptance and range of applicability as a result of the increased trust the users develop in such technology. Applications from the entertainment field may use emotion-based generated feedback towards increasing the involvement of the players. In a specific case, computer games may adapt playing conditions to the emotion level of the players. Systems for meetings may passively assist the audience, evaluate the dynamics of users' emotions and provide statistics concerning the level of implication and contribution of each individual. Similarly, computer-based therapy for treating various phobias may benefit from the automatic reading of patients' emotions to present the treatment conditions and to suggest doctors the possible ways to continue and to improve the process. Remote tutoring systems may improve the quality of teaching by enhancing the communication between the teacher and the students. Robotic systems specially designed for assisting elderly people or children in home environments, should integrate emotion reading systems to strengthen the communication with the subjects and to provide better help in case of emergency. Passive observation combined with automatic emotion assessment may be used also for determining the customers' affinity for specific products in the shop windows. "Smart buildings", "alive rooms" or "alive furniture" and "appliances" would effectively build more accurate internal representations of the user profile to adjust the settings in the home environment to better fit to the needs and preferences of people. Surveillance is another application domain in which the reading of emotions may lead to better performance in predicting the future actions of subjects. In this way, the emotion driven technology will further on enhance the existing systems for the identification and prevention of terrorist attacks in public places such as railway stations, airports, trains, buses or planes. Successfully achieving these goals by means of automatic systems or in combination with the classical human-based procedure, is equivalent to a tremendous decrease of the economical and social costs implied by the loss of human lives and property. In the same way, the biometrics field may lead to an increase of safety by restricting the access to private or specific items and infrastructures. Automotive industry will also incorporate emotion driven interfaces for increasing the safety on board. At the moment, commercial solutions with implemented face analysis, are already available as built-in features in cars for assisting drivers in traffic by warning and triggering their attention for obstacles on the road. Emotion analysis by computers may find interesting applications in the area of digital content handling. Indexing services may retrieve audio-visual data from large databases based on desired emotion clues. The process can take the form of finding pictures of faces showing similar facial expressions, looking for audio speech files that contain expressive utterances or detecting specific audio-visual segments in movies (e.g. automatically detecting passages of higher ranked violence). An adjacent application implies annotating and summarizing the multimodal content as a replacement for the tedious manual inspection by human experts. Computer systems and interfaces which run emotion recognition, may ultimately help artists in their creations by producing intermediate emotion-based representations to inspire and to evaluate the work of art. Computer-based reading of facial expressions may improve the performance of automatic systems for speech recognition and lip reading. Wearable computing is a long-time researched area that focuses on incorporating miniaturized component systems into personal items such as cloth, belts, wristwatches, mobile phones or eyeglasses. The functionality of these devices may be extended to support affect driven mechanisms to measure and communicate emotion information about the users. Interesting applications related to this field may be developed to help normal and impaired users get over the physical boundaries of the sensor human capabilities. Imagine the development of systems running on mobile phones or directly on eyeglasses that can automatically zoom in when the user feels uncomfortable trying to look at the departure timetable in the railway station.

These examples are only a few of the possible application areas of the automatic emotion reading technology. Ongoing projects have already spotted some of these domain areas and are set to find ways to apply the theoretical knowledge into working applications.

## 1.3 Automatic reading of emotions

The automatic recognition of emotions from audio-visual data by computers is inherently a difficult problem. The search for the best techniques for automating the measurement of human emotions, has challenged researchers to consider various scientific strategies.

Physiological methods make use of sensors to directly monitor specific parameters. Among these, electromyography - EMG use surface or needle electrodes applied on or right into the muscle tissue to quantify the amount of muscle contractions based on the electrical potentials. Electro-dermal activity - EDA measures the skin conductivity using sensors placed on fingertips. Blood volume pulse - BVP measures the volume of blood in the blood vessels under the skin. Electrocardiography - ECG uses electrodes applied on the subject's chest to measure the electrical activity of the heart in the form of heart rate and inter-beat intervals. Magnetoencephalography - MEG, functional magnetic resonance imaging - fMRI and electroencephalography - EEG are methods from the field of brain computing which measure the electrical and electromagnetic currents generated by the brain. Other methods involve the measurement of skin temperature using temperature sensors or infra-red imaging and the analysis of respiration from the contraction of rubber bands applied on the subject's chest.

The methods based on physiological measurements are mostly inefficient as they are invasive for the body or require special arrangements that are hard to obtain outside the lab environment. However, other non-obtrusive methods have been approached. From these, the ones that employ analysis on the audio and visual modalities by inspecting affect content in speech, body and face gestures, have gathered a lot of attention from the scientific community. The subject is more appealing now that new theoretical and technological findings allow for replacing the old measuring methods based on external sensors of direct body contact (e.g. electrodes on the face), with remote and totally non-invasive sensing devices. Eventually, these observation-centred techniques make possible the implementation of automatic systems for emotion understanding into real life applications.

## 1.4 Problem overview

Emotions typically present a high degree of variability which is generated by the inner individual characteristics and by the observation context. They vary in intensity, scale and representation. Except for the prototypic categories, the rest of emotions are culture dependent. Blended emotions show an increased complexity due to difficulty in identifying all the mixed feelings that trigger them. Emotion message received by either human observers or computer systems, may be the result of simulated/acted or of realistic/spontaneous emotions. Not with less importance, the hardware characteristics of the sound and visual sensors may influence the quality of processing emotions. A camera that has a low quality sensor or provides low-resolution image frames, as well as a low performing microphone device may provide noisy observations and introduce bias in processing the audio-video data.

In case of sound modality, the acoustic signal carries emotion content together with other information of the message such as characteristics related to the semantic content, gender, age or language. From this perspective, the recognition of emotions has to proceed with separating the emotion clues from the rest of information.

Assessing the facial expressions implies a close analysis of the face. The rather large variability of the face may be interpreted through the information the face reveals, as a complex multi-signal and multi-message system. As a multi-signal system, the face provides static, slow and rapid facial signals. The static signals represent several possibly permanent characteristics of the face, such as the skin pigmentation, the shape of the face, bone structure, fat deposits and the size, shape and location of the facial features (mouth, nose, eyes and brows). The slow signals reflect changes of the facial appearance that develop with time. Such changes relate to permanent wrinkles, muscle tone, skin texture or skin colouration that develop mainly with adults. The rapid signals are affected by the activation of facial muscles, may last for seconds or fractions of a second and may produce temporary changes in facial appearance, wrinkles and in the location and shape of facial features. Moreover, the face variability increases with cases when people alter the three types of face signals by using different hair styles, bangs and cosmetics or by intentionally or accidentally inhibiting muscle activations, masking one expression with another or by hiding the face with beards or sunglasses. As a multi-message system, the face produces messages related, at least to age, gender, race, emotion, mood, attitude, character, intelligence and attractiveness. Even by humans, none of these characteristics can be inferred without a certain degree of error. The error of emotion assessment mainly depends on weather the emotion is simulated or realistic, on the observer's ability to clearly perceive, understand and identify the emotion and on the context. The face observation conditions relate to the orientation of the face and to environment settings such as the lights and to the presence of objects occluding the face.

The observation context defines the type of media containing the emotion content. Assessing emotions from facial expressions in still pictures implies a static visual analysis, as opposed to video content that allows for dynamic time-based processing of emotions. In dynamic analysis, the face may present smooth or rigid head motions in the same time with expressing emotions. Handling sudden head movements involves additional procedures to correctly localize and track faces in sequences of video frames. For audio, affective bursts represent a separate type of acoustic manifestations carrying relevant emotion indicators.

While assessing emotions, humans do not rely on information from only one modality. Indeed, all modalities are simultaneously considered for accurately determining emotions. By analogy, the computer-based version of emotion recognition should extract and use mutually dependent emotion clues from multiple communication channels. These clues form a joint feature space on which some sort of data integration should be further on applied. At the moment, there is not much information on how this process develops by humans. Prior to building such a mechanism as part of an automatic system, specific techniques must be applied for finding the most relevant features and optimal data fusion models. The processing of multimodal content in which subjects express emotions in the same time with speaking is even more difficult because of the ambiguity in separating the influences of emotion and speech production on the facial appearance.

From the statistical point of view in the multimodal analysis of emotions, the semantics of what is being said, as well as the characteristics such as age, gender, skin colour, wearing eyeglasses, having beard and other indicators are considered noise and should be filtered out before the actual assessment of emotions. Choosing the best methods for modelling the perception of emotions implies the investigation of appropriate algorithms for audio and video data processing and for emotion pattern recognition. Secondly, determining the optimal values of the functional parameters of these algorithms is also very important for achieving higher performance. Finally, a particular aspect for the investigation is the implementation and the integration of all emotion data processing algorithms into working software systems. Although the hardware capabilities of current computer systems have greatly increased if compared to the previous generations, the rather high computational requirements of the models still make from the realization of real-time software prototypes of emotion driven systems, a task hard to fulfil.

## 1.5 Goals of the thesis

For several decades, automatic expression recognition has been scientifically considered a real challenging problem in the fields of pattern recognition or robotic vision. This thesis attempts to propose algorithms and techniques to be used for automatic recognition of six prototypic emotion categories by computer programs, based on the recognition of facial expressions and emotion patterns in voice. Considering the applicability in real-life conditions, the research is carried in the context of devising person independent methods that should be robust to various factors given the specificity of the considered modalities. An immediate focus represents the development of audio-visual algorithms and their implementation in form of software applications for automatic recognition of emotions. In this setup, the audio data is acquired from a microphone and the video input consists of sequences of frontal-view upright face images captured by a video camera. The previous works in the literature have lead to either only theoretical descriptions or both improvements and applications of existing models. Most of the unimodal system prototypes built in this way consist of off-line or semi-automatic emotion recognizers. Up to date there is barely any research publication supporting a fully developed and functional multimodal system for emotion recognition. With these aspects in mind, this thesis is intended to survey previous problems and solutions and the work on algorithms which involve audio and video processing for automatic emotion analysis and to test their implementation in the form of working prototypes.

The method to realize the survey consists of presenting the findings of theoretical and empirical studies on unimodal and bimodal recognition of human emotions and to test the models by means of automatic systems. We discuss about methods specifically applied on each modality, about audio and visual features and about the use of classification methods in the emotion recognition process.

In this study, we try to find answers to the following research questions:

- how to reliably detect and segment faces in single images; in addition to these, how to track faces in video sequences,

- what type of emotion indicators should be extracted from faces in single images and in sequences of images; what are the best models for representing faces showing prototypic facial expressions,

- which are the temporal correlations of visual features for recognition of basic emotions in video data,

- how to segment the audio signal and what types of acoustic features should be extracted from the resulting data segments,

- how to recognize emotions from single modalities and which binary and multi-class classification algorithms may be used to model separate and joint inner characteristics of vocal and facial expressions,

- how to achieve robustness for the recognition models, with regard to face and voice variability,

- how to determine facial expression labels from representations based on measurements of the facial muscle activity; in which way annotation and database structure may be used as bases for explaining the performance of the emotion computational models,

- how to fuse audio and visual clues for multimodal emotion recognition; how to solve the synchronization problem determined by the different sampling rates of unimodal signals,

- which development tools and methods can be used for implementing a fully functional software system for automatic reading of emotions,

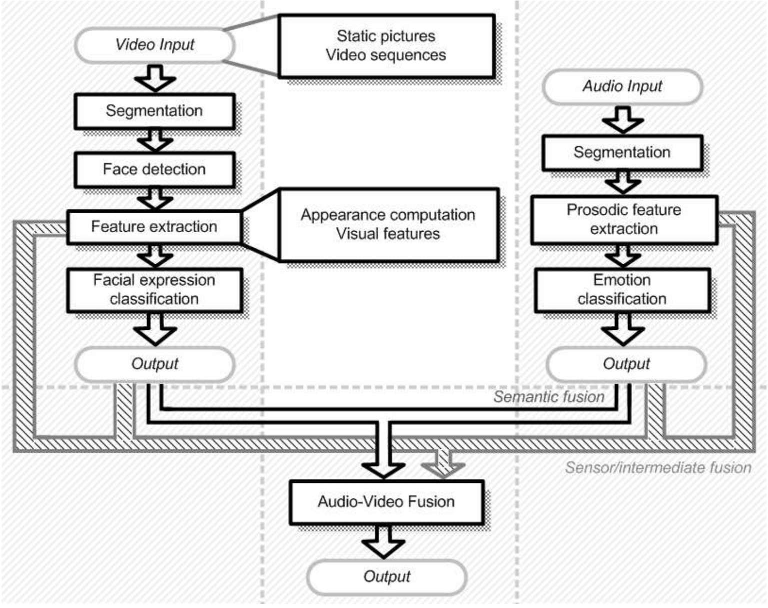


Figure 1.1: Diagram for bimodal emotion recognition

- which is the proper way to integrate separate data processing components into multimodal emotion-oriented applications.

Figure 1.1 schematically outlines the major components discussed in the thesis. In every chapter we emphasize, adapt, improve or propose methods to handle specific tasks of the emotion analysis.
We propose to conduct a series of exhaustively complete experiments with the goal of determining the most successful approach for each type of processing task. Additional contributions find roots in the specificity of the topic under research, in the form of practical improvements which help implementing and tuning the algorithms for real-time applications.

## 1.6 Outline of the thesis

The thesis contains nine chapters and covers the presentation of problems and solutions on different issues with regard to the unimodal as well as to the bimodal automatic recognition of human emotions.
The structure relates to a gradual introduction of theoretical and practical concepts in the increasing order of complexity, allowing in this way for an easy reading and understanding of the key elements.
**Chapter 2** presents an overview of relevant methodologies which have been previously used by researchers to work out specific problems of audio-visual data processing and pattern classification in the context of face detection, facial expression recognition and emotion extraction from speech signals.

The chapter also describes the theoretical bases of the classification models used in experiments reported in this thesis. Concretely, the focus is on support vector machines [185] and relevance vector machines [180] as kernel-based methods and on Adaboost [75], GentleBoost [76] and Adaboost.M2 [75] as boosting classifiers.

**Chapter 3** introduces the theoretical aspects as well as the practical use of active appearance models - AAMs [55]. This is a method we primarily use for processing and retrieving the information concerning the shape and texture of faces from both realistic single images and video sequences.

Combining motion and appearance represents a well-justified solution for tackling the face segmentation problems. The second part of the chapter deals with the description of the techniques involved in preprocessing the visual set of instances used for facial expression recognition. The procedure implies the selection of samples based on a given criterion, the normalization and filtering by using AAM models.

The methods described in this chapter are extensively used in the static and dynamic models for facial expression recognition presented in the following chapters.

**Chapter 4** presents a top-down approach for emotion recognition from single images. The experiments are run on a commonly used database in this field of research. The presented concepts are fundamental in the sense that they reflect the starting point for more complex algorithms for face analysis.

The static approaches are then extended to dynamic modelling in chapter **5**. In this setup, visual features are specifically adapted for the application on video sequences. Additionally, the computation of face motion vectors is employed for modelling the time-based facial variations.

**Chapter 6** illustrates two methods that cope with the problem of emotion recognition from speech data. Apart from using GentleBoost as a binary classifier, we investigate the use of hidden Markov models - HMMs for the temporal modelling of acoustic emotion patterns.

**Chapter 7** addresses several bimodal methods for facial expression recognition in video data. The goal is to obtain even more robust emotion recognition models by running sequential or parallel processing methods on face and utterance data. In this context, relevant information based on visual and acoustic clues is derived and fused together at different levels of abstraction.

**Chapter 8** describes the practical elements regarding the implementation of an automatic bimodal emotion recognition system. The system is based on fusing data of facial expressions and data of emotions extracted from speech signal. The working prototype being described in this chapter has been built based on the specifications of the best emotion recognition models presented throughout the thesis. So far, the software system has been introduced at various international conferences and national thematic presentations.

**Chapter 9** presents a multimodal framework proposed as a solution for the integration of all the data processing methods involved in the recognition of emotions. The typical complexity behind the emerging distributed multimodal systems is reduced in a transparent manner through the multimodal framework that handles data at different levels of abstraction and efficiently accommodates constituent technologies. The research is carried on for the application domain of systems for surveillance in train compartments.

The implemented algorithms presented in the previous chapters are integrated as modules that communicate with each other through tuple data spaces. These represent shared logical memories and allow for decoupling the communication in both time and space. The approach successfully complies with the requirements that arise in real-time application contexts.

**Chapter 10** summarizes the most important results of the approaches presented in this thesis and presents the perspectives in the area of automatic emotion recognition.

# Chapter 2

# Related work and definitions

## 2.1 Introduction

The recognition of emotions from audiovisual data is typically done in certain classification setups. Such a setup typically assumes collecting sensor data from microphones or video cameras, segmentation, feature extraction and classification. In this process, multiple emotion clues that regard facial expressions, speech and body gestures, may be extracted and integrated. This chapter presents the methodology with respect the data types, models and classification algorithms that may be used to build systems for recognizing emotions from facial expressions, speech and both modalities taken together.

For analysing facial expressions, the process implies the detection of faces and the computation of certain features based on the face image data.

The features may be extracted with holistic image-based approaches, from the whole face image or with local image-based approaches, from face regions that are informative with respect to facial expressions. Some approaches firstly model the facial features and then use the parameters as data for further analysis. In some cases, large feature sets are first extracted from the audio and visual data and then automatic methods are employed to selectively identify the most relevant features for classification.

In the literature, various types of audio and visual features are extracted and used for the classification. The classification maps feature data to certain emotion categories using distinct pattern recognition algorithms. Most of the research works on emotion analysis so far have used supervised pattern recognition methods to classify emotions. According to this, the models are first trained using annotated sets of data and then tested preferably using separate data. This chapter describes computational models that may be used as part of a multimodal emotion recognition framework. In the next chapter, we conduct a series of experiments using some of the algorithms presented here.

## 2.2 Models for human emotion recognition

In 2000, Pantic and Rothkrantz [145] presented the state of the art of models for automatic analysis of facial expressions. The paper [186], published in 2006, contains an overview of 64 emotion speech data collections, acoustic features and classification techniques for emotion analysis in speech. The most updated survey on data corpora and affect recognition methods for audio, visual and spontaneous expressions has been recently provided by Zeng et al. [211] in 2009.

### 2.2.1 Audio-visual databases

Assessing emotions is certainly a difficult task even for human observers. Because of this, the requirement to have fully labelled datasets prior to actually building emotion models seems to be quite hard to satisfy. The options in this case relate to using publicly available datasets and collecting new sets of data. In either case, the data sets may be described in several ways.

According to the authenticity of the emotions captured in the data, the data sets contain acted, elicited or real life emotions. The acted emotion recordings are obtained by asking the subjects to speak with or to show predefined emotions. Elicited emotion datasets are better than acted emotion datasets and result from provoking subjects to have certain emotion internal states. As opposed to these, the real life recordings provide the best datasets because they contain natural, unbiased emotions. Research studies have revealed that compared to genuine expressions, facial expressions of simulated emotions are characterized by irregular timing of facial muscle contraction, increased asymmetry and missing action components.

The Cohn-Kanade AU-Coded Facial Expression database [103] contains sequences of images from neutral to target display for 23 facial displays performed by 100 university students. In the past few years this database became the reference benchmark for testing models on facial expression recognition. The Karolinska Directed Emotional Faces - KDEF database [121] was originally developed to be used for psychological and medical studies of perception, attention, emotion, memory and backward masking experiments. The database contains 4.900 images of 7 different emotional expressions collected from 70 amateur actors aged between 20 and 30 years. The AR face database [127] includes over 4.000 frontal face images of 126 subjects under different facial expressions, illumination conditions and occlusions. The Japanese Female Facial Expression - JAFFE database [122] contains 213 images of 6 basic facial expressions and the neutral pose recorded from 10 different people. The Pose, Illumination, and Expression - PIE database [165] contains 41.368 images of 68 people in 13 different poses, 43 different illumination conditions, and with 4 different expressions. The Yale Face database [108] includes 165 grey-scale images of 15 individuals that show different facial expressions or configurations. The extended version Yale Face Database B [77] contains 5.760 single light source image samples of 10 subjects recorded under 9 poses and 64 illumination conditions.

The Danish Emotional Speech - DES corpus [60] includes speech recordings of five emotion categories expressed by four professional Danish actors. The Berlin Emotional Speech database - EMO-DB contains 495 samples of simulated emotional speech in German language by 10 actors. The AIBO database

[16] contains audio samples of spontaneous emotional reactions collected in a Wizard-of-Oz scenario in which German and English children had to instruct a Sony AIBO robot to fulfil specific tasks. The annotation of the database is word-centred and considers the neutral category, joyful, surprised, emphatic, helpless, touchy or irritated, angry, motherese, bored, reprimanding and other emotions. The 'Vera am mittag' database contains 12 hours of spontaneous emotional audio-visual recordings from unscripted, authentic discussions between the guests of a German TV talk show. The data is segmented into broadcasts, dialogue acts and utterances and is labelled on a continuous-valued scale on valence, activation and dominance emotion primitives. The Enterface05 multimodal database [126] contains 1166 simulated emotion recordings of 42 subjects of 14 different nationalities showing prototypic emotions.

### 2.2.2 Face detection models

The face detection problem implies the segmentation of image regions that are part of the face. This is done by using various visual features such as rectangular or elliptical parameters. Alternatively, the face location may be determined using the location of the eyes detected in video data from infra-red camera or by using pixel intensity methods that first search for skin-regions in the image. In several papers, the detection of faces in images have been approached by deformable models which represent the variations in either shape or texture of the face object. The active shape models - ASM [35] and active appearance models - AAM [55] are two deformable models that have been extensively researched and used with good results in the literature.
Point distribution models - PDMs relate to a class of methods used to represent flexible objects through sets of feature points that indicate deformable shapes. Marcel et al. [124] use ASMs and local binary patterns - LBPs to localize the faces within image samples. Tong et al. [182] propose a two-level hierarchical face shape model to simultaneously characterize the global shape of a human face and the local structural details of each facial component. The shape variations of facial components are handled using multi-state local shape models. Edwards et al. [55] introduced the AAM as a method to analyse the objects using both shape and grey-level appearances. The shapes are processed by point distribution models. The spatial relationships are determined using principal components analysis - PCA that build statistical models of shape variation. In a similar way, statistical models of grey-level appearance are derived by applying PCA on shape-free samples obtained by wrapping the face images using triangulation. The models of shape appearance and grey-level appearance are finally combined using PCA, to derive appearance vectors that control both grey-level and shape data.

### 2.2.3 Features derived from face images

Depending on the method used, the facial feature detection stage involves global or local analysis. Viola&Jones features relate to visual representations that are reminiscent of Haar basis functions. There are three types of features, each of them being computed based on the difference between the sums of pixel intensities in horizontally or vertically adjacent rectangular regions.
Although only two vertical and horizontal orientations are available, these fea-

tures provide a rich image representation to support effective learning. Their limited flexibility is compensated by extreme computational efficiency. The Gabor filters are orientation and scale tunable edge and line detectors, which are optimal in the sense of minimizing the joint two-dimensional uncertainty in space and frequency. Local binary patterns - LBPs have been proved to provide consistent representations of the face images. Several research papers [69] [67] [70] [71] [68] [111] [90] [160] [162] [81] have focused on using this type of features in combinations with different recognition algorithms. The few papers published so far [218] [219] [172] on the use of adapted LBP for the video analysis of facial expressions have indicated that models which rely on such features present very high performance. Optical flow-based features are natural representations to capture changes in the texture of face images. Such features reflect dense motion fields from the whole face image or from certain regions in the face area. The mapping of motion vectors to facial expressions implies computing motion templates by summing over the set of training motion fields. Motion vector data were employed for facial expression recognition in [129] [62] [203] [141] [142] [61] [205] [114] [92] [138] [54]. The shape and texture variation of faces provides relevant information to the classification. The AAM-oriented technique may be practically applied on face data to generate compact and parametrised description of the faces. Because of their capability to encode the face deformation, the appearance parameters may be further on used as input features for various classification methods. As an example, Saatci and Town [154] use AAM parameters for classifying gender and four basic facial expressions.

Some research works derive features based on the location of specific key points on the face image. These key points are also called face landmarks or facial characteristic points - FCPs. Different key point based face representations exist, like the FCP model of Kobayashi and Hara [106] or the face point model from the MPEG-4 standard.

The facial definition parameters - FDPs and the facial animation parameters - FAPs from the MPEG-4 framework provide definitions for facial shape and texture. They contain representations for the topology of the underlying face geometry and for face animations that reproduce expressions, emotions and speech pronunciations. FAPs make use of a spatial reference set of 84 face feature points on the neutral face, to derive high-level parameters, visemes and expressions. In the framework, viseme-oriented definitions are used to describe synchronized movements of the mouth and facial animation. Although FAPs may be successfully used to derive features for facial expression recognition [97], they have limited applicability due to the lack of a clear quantitative definition framework.

### 2.2.4 Acoustic features

Several acoustic features have been previously investigated [186] for emotion recognition from speech. They are determined based on specific measures that consider different characteristics of the audio signal.

The measures may be estimated from short time or long-time audio data segments. While long-time features are determined from the whole utterance, the short-time features are computed over smaller time windows, which typically range from 20 to 100 milliseconds.

The time-related measures can discern excitement and therefore emotions in

speech by referring to the rate and duration of speech and to silence measures such as the length of pauses between the words and sentences.
The intensity-related measures account for the amount of effort required to produce speech and so are linked to the emotion status of the speaker.
The frequency-related measures are traditionally used for the identification of phonemes given the pattern of their resonant frequencies.
The fundamental frequency of the phonation is also known as the pitch frequency and is defined as the vibration rate of the vocal folds. The pitch or fundamental frequency are good indicators for sensing the vocally-expressed excitement in speech signals.
The pitch of the glottal waveform is generated from the vibration of the vocal folds. The tension of vocal folds and the sub-glottal air pressure are essential factors for emotion production. The pitch period is another parameter that measures the time between two successive vocal fold openings.
Some other acoustic-based features commonly used for classifying emotions relate to formants, intensity of the speech signal, vocal-tract cross section areas, mel-frequency cepstral coefficients - MFCCs and linear predictive coefficients - LPCs.

### 2.2.5 Models for face representation

Several research papers approached the analysis of faces by using geometric features. For this, the authors represent the face in terms of sets of distance and angle features computed from the location of specific face landmarks. A precondition for this method is to first determine the location of these landmarks. The landmarks are also called key points or facial characteristic points and represent distinguishable points on the face area like the inner and outer corners of the eyes or the left and right-most points of the mouth and eyebrows. Several algorithms have been proposed for computing the location of these points.
Based on an initial estimation of the face location in the image, the coordinates of the face landmark points may be found by aligning a face model to the image. When using active shape models or active appearance models for aligning or detecting faces, the localization of face landmarks takes place simultaneously with fitting the face shape.
Several methods approach the recognition problem by using specialized classification models and features extracted from the face appearance [94][52]. McKenna et al. [130] use DPMs together with local image measurements in the form of Gabor wavelets, to model rigid and non-rigid facial motions and to track facial feature points.
To extract the face feature points, Zuo and de With [222] use active shape models which integrate global shape description and Haar wavelets which represent local texture attributes. Milborrow and Nicolls [133] propose simple methods like trimming covariance matrices by setting entries to zero or adding noise to the data, as extensions of AAM to locate face landmarks in frontal-view images. Martin et al. [125] create two active appearance models to track faces in input images. A Viola&Jones face detector is used to initialize the first AAM producing a coarse estimation of the model parameters. The second, more detailed AMM is used when the estimation error of the first AAM drops below a certain threshold. For the initialization, the process utilizes the result of the first AMM.

The authors perform the classification using the shape data and the combination of shape and appearance data extracted from gray-level images and from edge images of face samples.
The work [120] investigates the use of features derived from AAM which are based on 2D and 3D shape and 2D appearance for detecting face activity. Different normalization methods related to the face data are also studied and proved to be significantly beneficial for the recognition performance.
A representation based on topological labels is proposed by Yin et al. [204]. It assumes that the facial expressions are dependent on the change of facial texture and that their variation is reflected by the modification of the facial topographical deformation. The classification is done by comparing facial features with those of the neutral face in terms of the topographic facial surface and the expressive regions.
The system proposed by [135] is based on a 2D generative eye model that implements encoding of the motion and fine structures of the eye and is used for tracking the eye motion in a sequence.

#### Facial action units

The action units - AUs [56] reflect the distinguishable activation state of facial muscles or groups of facial muscles. The set of 44 unique action units together with the activation rules, are specified in the Facial Action Coding System - FACS [58]. Each action unit is assigned an arbitrary numeric code. One particular aspect of the FACS is that there is no 1:1 correspondence between groups of muscles and action units. Indeed, certain muscles exhibit different ways of contraction and so can produce a range of typical facial displays. The FACS coding also allow for the use of intensity of each facial display using a five point intensity scale.
Because of the fact that any human facial expression can be represented as a combination of Action Units, the research on detecting AUs has high relevance. In addition, the use of FACS labels offers a powerful mechanism to deal with the analysis of human facial activity in general and of human emotions in particular. Preparing AU-based data sets of facial expressions is a tedious process which implies visual inspection of every face image by specialized human observers.
Several papers have tackled the assessment of the activity of facial muscles through the analysis of Action Units [95] [215] [113] [34] [114] [177] [176][178] [179] [110] [58] [216] [14] [13] [120] [32] [200] [196] [183] [201] [171] [184] [182]. Other face codification procedures include FACS+, AFFEX and MAX [65].

### 2.2.6 Selection algorithms for relevant audio-video features

Studying the set of features extracted by various methods from the original data, one may conclude that not all the features present the same class discrimination power. For instance, in the case of using local binary patterns or Viola&Jones features, the methods first generate sets of visual features on the face image.
Imagine that such sets contain features that are located on face regions that do not change while showing expressions e.g. the nose region. As a consequence they do not offer sufficient information to aid the classification process. More-

over, keeping irrelevant features during classification may also attract a decrease of the classification performance.
Secondly, retaining large sets of features most frequently implies to have a large number of observation data to properly train the classifier models. To suffice this constraint, extensive additional work must be done to collect and annotate data. In many cases, this is probably almost impossible.
For the aforementioned reasons, the selection of features seems to be a practical solution for efficiently choosing the most relevant features for classification.
Feature selection basically represents an optimization problem which implies searching the space of possible subsets of features in order to identify the optimal or near-optimal feature subset, with respect to a specific criterion.
Several algorithms have been proposed for this task.

Sequential forward selection - SFS and sequential backward selection - SBS are two heuristic methods that provide suboptimal solution for the feature selection. SFS starts with an empty set of features and iteratively selects new features according to their relevance for the recognition problem. SBS starts with the set of all the possible features and progressively removes features with low discrimination power.

Sequential forward floating search - SFFS and sequential backward floating search - SBFS represent generalized schemes of the plus l-take away r method.

One of the most efficient methods to obtain low-dimensional representations of data is principal component analysis - PCA.
The algorithm implies finding a set of orthogonal basis vectors that describe the major variations under the constraint that the mean square reconstruction error is minimized.
For a set of data, the principal components are determined first and then expressed as eigenvectors. An orthogonal basis is created by ordering eigenvectors using the absolute magnitude of their eigenvalues and by selecting the ones that present the largest variance direction. Dimensionality reduction is achieved by keeping only a small number of eigenvectors. Because such a basis contains the largest eigenvectors, the approximation of data instances using weighted linear combination of basis components, yields to minimum reconstruction errors.
The process of using only a reduced set of the largest eigenvectors to project initial data instances, is equivalent to selecting the most representative features in terms of explaining the variation in data.
A notable drawback of PCA, as a linear feature extraction technique, is that it may loose important information for discriminating between different classes. This may be explained by the unsupervised process for generating the basis. As a result of this process, the basis components more likely represent information which is common to all data instances and not to class categories.
In [53], Dubuisson et al. apply PCA to obtain a set of eigenfaces from face images. To determine the optimal number of basis components, they apply FSS on the projected data by maximizing a general class separability measure. The measure used is the Fisher criterion and reflects the ratio of within-class scatter to between-class scatter of data.
PCA was used for selecting features for emotion recognition in speech in [109] and for multimodal emotion recognition in [194] and [123].

Linear discriminant analysis - LDA is a supervised feature selection method based on finding data projections that optimally separate the category clusters while minimizing the variance within each cluster.
One difference between PCA and LDA is that PCA change the location and form of the data while LDA does not change the location and achieves better class separability. A drawback of LDA is that it cannot be applied in some cases in which the number of samples is too small compared to the dimension of the original data. To solve this problem, Belhumeur et al. [19] propose that, prior to using LDA, to first reduce the dimensionality of the face images using PCA. In [53], the combination of PCA, LDA and SSF yields the best results for separating three and six facial expression categories. In the same setup, SFS and SBS show no major difference to the results of classification.
Another problem related to LDA regards the incapacity to achieve class separability if multiple clusters per class exist in the input feature space. Chen and Huang [195] considered this limitation and proposed clustering based discriminant analysis - CDA as a LDA variant which, more than LDA, minimizes the scatter in every cluster, for all the facial expression classes.

Independent component analysis - ICA is a unsupervised statistical technique from the class of blind source separation - BSS methods, for separating data into underlying informational components. The method applies a linear transformation of the observed random vectors into components which are minimally dependent on each other. In contrast with PCA that decomposes a set of signal mixtures into a set of uncorrelated signals, ICA decomposes a set of signal mixtures into a set of independent signals. In [32], the authors use ICA to extract and represent single and combined facial action units, as subtle changes of facial expressions. In a similar setup, Fasel and Luettin [64] showed that the use of ICA leads to slightly better results than using PCA, for recognizing single facial action units. Shin [170] uses PCA and ICA to recognize facial expressions based on a two-dimensional representation for 44 emotion words.

Genetic algorithms - GAs are optimization techniques based on the analogy with biological evolution. This class of methods involves searching a space of candidate hypotheses to identify the best hypothesis, using a predefined numerical measure called the fitness function. The algorithm iteratively generates new sets of hypotheses, by evaluating, probabilistically selecting and changing the most fit elements from the set. The elements are changed using genetic operators such as crossover and mutation.
Schuller et al. [156] proposed a speech oriented feature selection method which combines SFFS and GA methods. Applying SFS method on a speech related set of 58 prosodic, sub-band, MFCC and LPC features, Altun et al. [10] obtained a considerable reduction of the cross validation error for emotion analysis from the speech signal. Yu and Bhanu [209] use GAs to select Gabor features that are subsequently used for facial expression recognition.

### 2.2.7 Emotion recognition algorithms

The recognition of emotions has focused on the use of various algorithms. Depending on the context and on the type of data, these have been constantly

adapted and improved in due course of time [145].
First, we present the most used methods involved in various setups for detecting faces, recognizing face activity and facial expressions from video signals and for recognizing emotions from speech signals.

Decision tree learning is a practical method for inductive inference which approximate discrete-valued functions, using decision tree representations. A node in the tree represents a test for an attribute of the data and the branches reflect possible values of the attribute. Some decision tree algorithms replace the discrete valued attributes with continuous-valued decision attributes. The decision tree models that have only one attribute with two branches, are called decision stumps.
Some more complex models like boosting classifiers, commonly use and integrate several simple decision stumps to increase the classification performance. Conversely, decision stumps are practically used to combine several powerful binary classifiers such as support vector machines so to convert two-class classification models into a single multi-class classification setup.

Artificial neural networks - ANNs represent robust methods for learning real-valued, discrete-valued, and vector-valued target functions. Because of their efficiency, they are mostly applied for learning and interpreting complex real world and noisy sensor data, such as inputs from cameras and microphones.
The study of ANNs finds its roots partly in the observation that biological learning systems consists of very complex networks of interconnected neurons. Analogically, artificial neural networks have been defined as containing a densely interconnected set of simple units, where each unit has a number of real-valued inputs and a single real-valued output. In this setup, the output of one unit may be linked to the inputs of other units.
Different AAN algorithms do exist and range from single perceptrons that can only express linear decision surfaces, to complex multilayer networks. Among these, multilayer networks that use back propagation learning are capable to represent a rich variety of non-linear decision surfaces. They apply a gradient descent search through the space of possible network weights, iteratively reducing the error between the training instances and the network outputs.
The error surface of these models possibly contains several local minima. Moreover, the process is only guaranteed to converge toward some local minimum and not necessarily to the global minimum error.
AAN have been successfully used for face detection and facial expression recognition in [168] [48] [49] [106]. In [125], the authors classify six basic facial expressions plus the neutral state, using multi-layer perceptrons - MLPs with two hidden layers, tanh activation function and training based on back propagation algorithm. Another classification method the authors use consists of making separate AANs for each facial expression category and of identifying the model that give the minimum error between the image and model. Van Kuilenburg et al. [184] determine that a 3-layer feed-forward neural network with 94 input neurons and 15 hidden neurons leads to optimal results for classifying seven expression categories. AAN was used for emotion recognition in speech in [109] [137].

Partially observable Markov decision processes - POMDPs are similar to

HMMs in the sense that they describe observations as produced by hidden states, which are connected by Markovian chains. POMDP models implement actions and rewards and use decision theoretic planning. In this setup, agents are able to predict the effects of their actions based on the knowledge about the environment and further choose actions based on the predictions.
Hoey and Little [92] use POMDPs to build agents which learn relationships between unlabelled observations of a person's face and gestures, the context and their own actions and utility functions.

Bayesian reasoning is a probabilistic inference method based on the assumption that quantities of interest are governed by probability distributions. The probabilities and the observed data are used during a reasoning process to derive optimal decisions. The reasoning implies a quantitative approach to weighting the evidence supporting alternative hypothesis. In practice, the use of Bayesian models typically presents two major drawbacks that are related to the requirement for initial knowledge of many probabilities and to the significantly high computational costs.
A simple Bayesian model is the naive Bayes - NB learner, also called the naive Bayes classifier. Here, the training data is described in terms of tuples of attribute values and target functions that take values in some finite set. As opposite to other learning methods, the prediction process does not explicitly search through the space of possible hypotheses. Instead, the hypothesis is determined by counting the frequency of various data combinations within the training data. The naive Bayes classifier uses the assumption that the attribute values are conditionally independent given the target value. NB was used for facial expression recognition in [84]. In [159], the authors use adaptive GAs to reduce the prediction error for speech emotion recognition. The search is focused on features which perform poorly with respect to the probability of correct classification achieved by the Bayes classifier. These features are then removed from the SFFS procedure which use the same Bayes classifier-oriented criterion.
In contrast with naive Bayes classifier which assumes that all the variables are independent, Bayesian belief networks - BBNs apply the conditional independence assumption locally, to subsets of the variables. A BBN represents the joint probability distribution for a set of variables, by specifying a set of conditional independence assumptions together with sets of local conditional probabilities. Several methods for inference in BBNs have been proposed in the literature. They perform either exact or approximate inference. The approximate algorithms have lower precision but higher efficiency. For learning BBNs, different methods apply depending on whether all the variables are observable or not. A special category of algorithms relate to methods for learning the structure of the network, in the case that the information is not available beforehand.
To classify emotions with BBNs, audio or video measurements are collected and set as evidences in the model.
Standard BBNs are also called static BNs because they attempt the emotion classification based on only visual evidences and beliefs from a single time instant. Dynamic BNs - DBNs are extensions of BBNs which can express temporal relationships and dependencies in the audio-video sequences.
Datcu and Rothkrantz [39] use BBNs which integrate sensor level data and facial motion data to recognize facial expressions. Gu and Ji use DBNs to visually recognize the face muscle activity, the driver's vigilance level [83] and to detect

the fatigue [82]. Zhang and Ji [217] use BNs and DBNs to model the static and dynamic recognition of facial activity and facial expressions.

Support vector machines - SVMs are classification models that attempt to find the hyper-plane that maximizes the margin between positive and negative observations for a specified class. They have been proved to be extremely useful in a number of pattern recognition tasks, among them being facial expression recognition [14] [125] [209]. The mathematical formulation of this classifier is presented in the next section.
In [154], the authors investigate the use of SVM for gender and facial expression recognition. The optimal performance for recognizing three expression categories and the neutral state, is achieved by a model that first detects the gender and then uses cascaded binary SVMs for the classification.
Other works which develop SVM-based facial expression recognition models include [84][120][196]. You et al. use PCA, LDA and SVM for emotion recognition in speech in [206]. SVM was used for speech emotion classification also in [38] [117] [118] [107].

Boosting concerns a specific class of methods which generate a very accurate classification rule by integrating rough and moderately inaccurate weak hypotheses. These hypotheses represent weak or base learning algorithms.
Adaptive boosting - Adaboost is a binary boosting method proposed by Freund and Schapire [75] which has very high generalization performance. The algorithm works by assigning and iteratively updating the weights on training data instances in a dynamic manner, according to the errors at the previous learning step. Misclassified data get higher weights, leading the learning process to focus more on the hardest examples. The algorithm is a type of large margin classifiers which minimizes an exponential function of the margin over the training set.
An interesting aspect of Adaboost is the capacity to identify outliers which are defined as mislabelled, ambiguous or hard-to-classify training instances. These examples are found among the data which have the highest associated weights. By emphasizing the classification on such data, the algorithm frequently generates models that are less efficient. To solve this problem, Friedman et al. [76] proposed Gentle AdaBoost - GentleBoost, as a variant of Adaboost with less emphasis on outliers.
Adaboost.M2 [75] is a variant of Adaboost which performs multi-class classification by integrating several binary problems.

For recognizing emotions, several classification models have been employed as weak learners and combined in frameworks proposed by the boosting methods. In a comparative study, Littlewort et al. [116] use AdaBoost, SVMs and linear discriminant analysis with Gabor filters to build fully automatic recognition systems for six prototypic facial expressions and the neutral states. They report the best results for a model which uses SVM with features selected by AdaBoost. The research [13] presents an algorithm for action unit detection based on Viola&Jones face detection, Gabor filters, Adaboost used solely for feature selection and SVM for classification. The face samples were rescaled to 96x96 pixels. Adaboost performed the selection of 200 features from a set of 165.888 features obtained by applying Gabor filter on the face images. The authors of [201] use binary patterns derived from Viola&Jones features applied in video sequences. The features are used as input for standard Adaboost algorithm which classifies six basic facial expressions and eight action units. The work [201] applies Adaboost classification with Viola&Jones features on video sequences from the Cohn-Kanade database with the goal of recognizing facial expressions and AUs. He el al. [89] use Adaboost to boost ICA learners for facial expression recognition. Other works which involved boosting techniques for facial expression recognition were [90] [30][13][116] [84] [134].

Hidden Markov models - HMMs relate to optimal learning methods that deal with time sequential data and provide time scale invariability.
In a recent research, Tong et al. [183] present a novel technique based on dynamic BBNs for detection of facial action units. They present also a comparison between their probabilistic oriented approach and the classification based on standard Adaboost algorithm that indicates better performance of the first. The model uses data that relate to 3D face shape facial muscular movements, 2D global shape and 2D facial component shapes. In the paper [9], the authors use facial animation parameters (FAPs) and multi-stream hidden Markov models to do the recognition of the six basic facial expressions. In [156], the authors use a hybrid neural-net HMM and 2-fold stratified cross validation to dynamically classify speech emotion samples in Emo-db dataset. In a comparative approach, they obtain the best emotion recognition results by applying SVM-trees and feature selection methods. Other HMM-based approaches for emotion classification in speech are [193] [118] [190] [107] [147].

Other classification methods include k-nearest neighbours - kNNs, expert systems [146], associative memories [205], Gaussian mixture models - GMMs and transferable belief models - TBMs [86].

### 2.2.8 Approaches to multimodal emotion recognition

The multimodal approach for emotion recognition basically represents an extension of the previously described unimodal algorithms. For each modality, data is processed for extracting relevant features which serve as emotion patterns of some kind in a classification context.
Appart from the problem of recognizing emotions separately, the integration of clues from audio and video signals rises particular problems. In this setup, the emotion recognition models have to deal with the integration of two different types of sensor observations, where each of them has its own availability rate.
From the integration perspective and taking into account the levels of data abstraction that characterize the two types of data, the multimodal classification combines unimodal inputs in different ways. According to that, the combination takes the form of low level fusion or high level fusion.
Low level fusion is also called feature level fusion or signal level fusion. On the other hand, high level fusion is also called semantic fusion or fusion at the decision level.
In the case of low level fusion, the feature vectors extracted by specific audio and video data processing methods, are directly combined and used as inputs for the classification models. A first problem concerns the temporal aspect of the input data that has to be classified. The models have to employ mechanisms

to derive relevant spatial-temporal features and, based on this data, to proceed with learning schemes that are able to capture the dynamics of each emotion category.

This constraint can be satisfied by only a few classification models. Recurrent NNs are used by Caridakis et al. [26] to integrate MPEG-4 FAPs and acoustic features related to pitch and rhythm, for recognizing natural affective states in terms of activation and valence. This type of NNs assumes that past inputs affect the processing of future inputs, in this way providing a mechanism for dynamic modelling of multimodal data. NNs have been used as bimodal integration models for emotion recognition also in [144] and [194]. Other approaches for low level fusion used multi-stream HMMs [212], SVMs [197][144][123] and DBNs [158].

Because the two types of observations reflect different availability rates, special models are required here to properly synchronize the data.

The high-level fusion implies that the emotion recognition is done separately for each modality and that the final decision on the emotion labels is determined by running a classification model which takes the separate results as inputs. Although this approach shows simplicity, it suffers from the incapacity to model correlations that undoubtedly exist between audio and video events with respect to the emotion development.

Because the input features consist of emotion labels generated by separate emotion analysers, it is possible to increase the classification performance by introducing different weights for each emotion category. The approach may be motivated through the findings according to which certain emotions are better expressed in one modality than in the other. Some classification models used for high level fusion include SVM [123] and NNs [144][194].

## 2.3 Computational models and tools

### 2.3.1 Kernel-based methods

**Support vector machine classifier for recognition**

The support vector machine - SVM algorithm [185] has been successfully used in classification related problems since it was introduced by Vapnik in the late 1970's.

In a binary classification setup, the idea is that given the collection of $N$ input-target pairs with $\mathbf{x}_1$, $\mathbf{x}_2$, ..., $\mathbf{x}_N$ vectors in $R^m$ and $t_1$, $t_2$, ..., $t_N$, $t_k \in \{-1, +1\}$, the classification of a new sample $\mathbf{x}$ is done by checking the sign of $y(\mathbf{x})$, where $y$ represents a linear model of the form:

$$y(\mathbf{x}) = \mathbf{w}^T \phi(\mathbf{x}) + b,$$

where $\phi(\mathbf{x})$ is a well determined feature-space transformation. In the case the training data is linearly separable, there exists at least one solution for $\mathbf{w}$ and $b$ so as that $y(\mathbf{x}) > 0$ for all instances for which $t_n = +1$ and $y(\mathbf{x}) < 0$ for all instances for which $t_n = -1$. The two relations can be rewritten as $t_n y(\mathbf{x}) > 0$ for all the instances included in the training data set.

The condition for finding the optimal solution for the parameters $\mathbf{w}$ and $b$ is that it should provide the lowest generalization error. The support vector machine

tackles the problem by making use of the term of margin as the smallest distance between the decision boundary and any of the data samples.
The distance of a point $\mathbf{x}$ from the hyperplane $y(\mathbf{x}) = 0$, where $y(\mathbf{x}) = \mathbf{w}^T\phi(\mathbf{x})+b$, is $|y(\mathbf{x})|/||\mathbf{w}||$. The correct classification condition $t_n y(\mathbf{x}_n) > 0$ for all samples, leads to the following relation for the distance of the point $\mathbf{x}_n$ to the decision surface:

$$\frac{t_n y(\mathbf{x}_n)}{||\mathbf{w}||} = \frac{t_n\left(\mathbf{w}^T\phi(\mathbf{x}_n)+b\right)}{||\mathbf{w}||}.$$

The parameters $\mathbf{w}$ and $b$ are determined by maximizing the margin:

$$\arg\max_{\mathbf{w},b}\left\{\frac{1}{||\mathbf{w}||}\min_n\left[t_n\left(\mathbf{w}^T\phi(\mathbf{x}_n)+b\right)\right]\right\}.$$

For the data point that is the closest to the decision boundary, we arbitrarily set $t_n\left(\mathbf{w}^T\phi(\mathbf{x}_n+b\right) = 1$. It follows that for all the data points, the following holds:

$$t_n\left(\mathbf{w}^T\phi(\mathbf{x}_n)+b\right) \geq 1, n = 1, ...N.$$

One requirement of the optimization problem is the maximization of $||\mathbf{w}||^{-1}$. That is the same as minimizing $||\mathbf{w}||^2$ which, in turn, leads to the rewritten form of the optimization problem:

$$\arg\min_{\mathbf{w},b}\frac{1}{2}||\mathbf{w}||^2.$$

Solving for the model parameters is equivalent to minimizing a quadratic function subject to a set of linear inequality constraints. Each of the constraints is assigned a Lagrange multiplier $a_n \geq 0$ as follows:

$$L(\mathbf{w},b,\mathbf{a}) = \frac{1}{2}||\mathbf{w}||^2 - \sum_{n=1}^{N} a_n\{t_n\left(\mathbf{w}^T\phi(\mathbf{x}_n)+b\right)-1\}.$$

As suggested by the form of Lagrangian function, we try to minimize with respect to $\mathbf{w}$ and $b$ and to maximize with respect to $\mathbf{a}$, where $\mathbf{a} = (a_1, a_2, ..., a_N)^T$. Making the derivatives of $L(\mathbf{w},b,\mathbf{a})$ equal to zero, the following relations are obtained:

$$\begin{aligned}\mathbf{w} &= \sum_{n=1}^{N} a_n t_n \phi(\mathbf{x}_n)\\ 0 &= \sum_{n=1}^{N} a_n t_n.\end{aligned}$$

The dual representation of problem then gives the following:

$$\tilde{L}(\mathbf{a}) = \sum_{n=1}^{N} a_n - \frac{1}{2}\sum_{n=1}^{N}\sum_{n=1}^{N} a_n a_m t_n t_m k(\mathbf{x}_n,\mathbf{x}_m)$$

with constraints:

$$\begin{aligned} a_n &\geq 0, n = 1, 2, ..., N \\ \sum_{n=1}^{N} a_n t_n &= 0, \end{aligned}$$

where $k(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})^T \phi(\mathbf{x}')$ is the kernel function. The actual form of the optimization problem considers the optimization of the quadratic function of $\mathbf{a}$ subject to a set of inequality constraints. The condition that the Lagrangian function $\tilde{L}(\mathbf{a})$ is bounded below implies that the kernel function $k(\mathbf{x}, \mathbf{x}')$ must be positive definite.

The classification of unseen data can be done using the parameters $\{a_n\}$ and the kernel function:

$$y(\mathbf{x}) = \sum_{n=1}^{N} a_n t_n k(\mathbf{x}, \mathbf{x}_n) + b.$$

The optimization problem can be handled further more through the application of Karush-Kuhn-Tucker conditions, which implies that three more other properties hold:

$$\begin{aligned} a_n &\geq 0 \\ t_n y(\mathbf{x}_n) - 1 &\geq 0 \\ a_n\{t_n y(\mathbf{x}_n) - 1\} &= 0. \end{aligned}$$

According to the third condition, for every data instance $a_n = 0$ or $t_n y(\mathbf{x}_n) = 1$. The data instances for which $a_n \neq 0$, are so-called the *support vectors.* The values of $a_n$ are found by solving the quadratic programming problem. Eventually, the value of parameter $b$ is determined as follows:

$$b = \frac{1}{N_S} \sum_{n \in S} \left( t_n - \sum_{m \in S} a_m t_m k(\mathbf{x}_n, \mathbf{x}_m) \right).$$

The term $N_S$ denotes the number of support vectors of the model.

The second approach is to develop support vector machine models which allow for training with data points that are not linearly separable in the feature space $\phi(\mathbf{x})$. More precisely, the model should tolerate the misclassification of certain data points. That can be done by introducing a linear function of the distance from the boundary, as a penalty in the model. Subsequently, every training data instance $\mathbf{x}_n$ is initially assigned a *slack variable* $\xi_n \geq 0$. Each $\xi_n$ is defined in such a way so that $\xi_n = 0$ for all data instances that are inside the correct margin boundary and $\xi_n = |t_n - y(\mathbf{x}_n)|$ for the rest of the instances.

As a consequence, the data points which are located on the decision boundary $y(\mathbf{x}_n) = 0$ imply that $\xi_n = 1$ and the misclassified instances imply that $\xi_n > 1$. From here, the following constraints result:

$$t_n y(\mathbf{x}_n) \geq 1 - \xi_n, n = 1, 2, ..., N.$$

The slack variables are constrained so as that $\xi_n \geq 0$. The goal is to maximize the margin and in the same time to penalize the points for which $\xi_n > 0$:

$$C \sum_{n=1}^{N} \xi_n + \frac{1}{2} ||\mathbf{w}||^2.$$

The term $C > 0$ controls the trade-off between the slack variable penalty and the margin.
The optimization problem can be rewritten using the Lagrangian:

$$L(\mathbf{w}, b, \mathbf{a}) = \frac{1}{2}||\mathbf{w}||^2 + C\sum_{n=1}^{N}\xi_n - \sum_{n=1}^{N} a_n\{t_n y(\mathbf{x}_n) - 1 + \xi_n\} - \sum_{n=1}^{N}\mu_n\xi_n.$$

The terms $a_n \geq 0$ and $\mu_n \geq 0$ are Lagrange multipliers. The Karush-Kuhn-Tucker conditions become:

$$\begin{aligned} a_n &\geq 0 \\ t_n y(\mathbf{x}_n) - 1 + \xi_n &\geq 0 \\ a_n\left(t_n y(\mathbf{x}_n) - 1 + \xi_n\right) &= 0 \\ \mu_n &\geq 0 \\ \xi_n &\geq 0 \\ \mu_n\xi_n &= 0, \end{aligned}$$

for $n = 1, 2, ..., N$. By equating the partial derivatives of $L$ with respect to $\mathbf{w}$, $b$ and $\xi_n$ to zero, it results that:

$$\begin{aligned} \frac{\partial L}{\partial \mathbf{w}} = 0 &\quad\Rightarrow\quad \mathbf{w} = \sum_{n=1}^{N} a_n t_n \phi(\mathbf{x}_n) \\ \frac{\partial L}{\partial b} = 0 &\quad\Rightarrow\quad \sum_{n=1}^{N} a_n t_n = 0 \\ \frac{\partial L}{\partial \xi_n} = 0 &\quad\Rightarrow\quad a_n = C - \mu_n. \end{aligned}$$

The Lagrangian can be written as follows:

$$\tilde{L}(\mathbf{a}) = \sum_{n=1}^{N} a_n - \frac{1}{2}\sum_{m=1}^{N}\sum_{n=1}^{N} a_n a_m t_n t_m k(\mathbf{x}_n, \mathbf{x}_m).$$

From the previous relations, it results that $a_n \geq 0$ and $a_n \leq C$. The problem is equivalent to the minimization of $\tilde{L}(\mathbf{a})$ with respect to the dual variables $\{a_n\}$ and with the following restrictions, for $n = 1, 2, ..., N$:

$$\begin{gathered} 0 \leq a_n \leq C \\ \sum_{n=1}^{N} a_n t_n = 0. \end{gathered}$$

Similar to the linearly separable data, there are data points for which $a_n = 0$ that do not contribute to the model classification. The other data points for which $a_n > 0$ constitute the support vectors and satisfy:

$$t_n y(\mathbf{x}_n) = 1 - \xi_n.$$

The support vectors have the property that they lie on the margin due to the fact that $a_n < C \Rightarrow \mu_n > 0$ and $\xi_n = 0$. The parameter $b$ can be determined using the relation:

$$b = \frac{1}{N_M} \sum_{n \in M} \left( t_n - \sum_{m \in S} a_m t_m k(\mathbf{x}_n, \mathbf{x}_m) \right).$$

The term $M$ is the set of indices of data instances for which $0 < a_n < C$.

### Relevance vector machine classifier for recognition

Tipping introduced the *relevance vector machine* - RVM [180] as a probabilistic sparse kernel model based on the support vector machine theory.

In the binary classification setting, the target variable is $t \in \{0, 1\}$. The model has the form of a logistic sigmoid function that is applied on a linear combination of basis functions:

$$y(\mathbf{x}, \mathbf{w}) = \sigma(\mathbf{w}^T \phi(\mathbf{x})),$$

where $\sigma$ is the logistic sigmoid function:

$$\sigma(x) = \frac{1}{1 + e^{-x}}.$$

The RVM model uses ARD prior so as that each weight parameter $w_i$ has assigned a separate precision hyper-parameter $\alpha_i$. The prior is considered to have the form of a zero-mean Gaussian distribution function. The weight prior takes the form:

$$p(\mathbf{w}|\alpha) = \prod_{i=1}^{M} \mathcal{N}(w_i|0, \alpha_i^{-1}),$$

where $\alpha = (\alpha_1, \alpha_2, ..., \alpha_M)$ and $M$ is the number of parameters. The key element of RVM is that, while maximizing the evidence with respect to the hyper-parameters $\alpha_i$, many of them go to infinity. That leads to the fact the corresponding weight parameters $w_i$ have posterior distributions which concentrate to zero. The effect is that these parameters do not contribute any more to the model classification and are pruned out. The model is then called to be a sparse model.

The computation of the model parameters implies the estimation of the hyper-parameters in an iterative manner. The process starts with an initial value of the hyper-parameters. Starting with an estimation of $\alpha$, each iteration implies building a Gaussian approximation to the posterior distribution. This also facilitates the construction of an approximation of the marginal likelihood. The re-estimation of the value of $\alpha$ is then obtained following the maximization of the approximate marginal likelihood. The process is eventually repeated until convergence is achieved.

The approximation of the posterior distribution is done using Laplace approximation.

The mode of the posterior distribution over $w$ is obtained by maximizing:

$$\ln p(\mathbf{w}|\mathbf{t}, \alpha) = \ln\{p(\mathbf{t}|\mathbf{w})p(\mathbf{w}|\alpha)\} - \ln p(\mathbf{t}|\alpha)$$
$$= \sum_{n=1}^{N} \{t_n \ln y_n + (1 - t_n) \ln(1 - y_n)\} - \frac{1}{2}\mathbf{w}^T \mathbf{A} \mathbf{w} + const,$$

where $\mathbf{A} = diag(\alpha_i)$. The solution is found by using iterative re-weighted least squares (IRLS).

The gradient vector and the Hessian matrix of the log posterior distribution are as follows:

$$\begin{aligned} \bigtriangledown \ln p(\mathbf{w}|\mathbf{t}, \alpha) &= \mathbf{\Phi}(\mathbf{t} - \mathbf{y}) - \mathbf{A}\mathbf{w} \\ \bigtriangledown \bigtriangledown \ln p(\mathbf{w}|\mathbf{t}, \alpha) &= -(\mathbf{\Phi}^T \mathbf{B} \mathbf{\Phi} + \mathbf{A}), \end{aligned}$$

where $\mathbf{B}$ is an $N \times N$ diagonal matrix with elements $b_n = y_n(1 - y_n)$, $\mathbf{y} = (y_1, ..., y_N)^T$ and $\Phi$ is the design matrix with elements $\Phi_{ni} = \phi(x_n)$.

At convergence, the negative Hessian corresponds to the inverse covariance matrix for the Gaussian approximation to the posterior distribution.

The mode of the approximation to the posterior distribution corresponds to the mean of the Gaussian approximation. The mean and covariance of the Laplace approximation is found by equating the previous equation of the gradient vector with zero. The solution is as follows:

$$\begin{aligned} \mathbf{w}^* &= \mathbf{A}^{-1} \mathbf{\Phi}^T (\mathbf{t} - \mathbf{y}) \\ \sum &= (\mathbf{\Phi}^T \mathbf{B} \mathbf{\Phi} + \mathbf{A})^{-1}. \end{aligned}$$

Given the Laplace approximation, the next step is to determine the marginal likelihood:

$$\begin{aligned} p(\mathbf{t}|\alpha) &= \int p(\mathbf{t}|\mathbf{w}) p(\mathbf{w}|\alpha) d\mathbf{w} \\ & \quad p(\mathbf{t}|\mathbf{w}^*) p(\mathbf{w}^*|\alpha) (2\pi)^{M/2} |\sum|^{1/2}. \end{aligned}$$

By substituting for $p(\mathbf{t}|\mathbf{w}^*)$ and $p(\mathbf{w}^*|\alpha)$ and by setting the derivative of the marginal likelihood with respect to $\alpha_i$ equal to zero the following is obtained:

$$-\frac{1}{2}(w_i^*)^2 + \frac{1}{2\alpha_i} - \frac{1}{2}\sum_{ii} = 0.$$

Setting $\gamma_i = 1 - \alpha_i \sum_{ii}$ further gives the following re-estimation formula for $\alpha$:

$$\alpha_i^{new} = \frac{\gamma_i}{(w_i^*)^2}.$$

If $\hat{\mathbf{t}} = \mathbf{\Phi} \mathbf{w}^* + \mathbf{B}^{-1}(\mathbf{t} - \mathbf{y})$, the approximate log marginal likelihood can be written as:

$$\ln p(\mathbf{t}|\alpha, \beta) = \frac{1}{2}\{N \ln(2\pi) + \ln|\mathbf{C}| + (\hat{\mathbf{t}})^T \mathbf{C}^{-1} \hat{\mathbf{t}}\},$$

where $\mathbf{C} = \mathbf{B} + \mathbf{\Phi} \mathbf{A} \mathbf{\Phi}^T$.

The use of ARD Gaussian prior over the weights $w_i$ is one factor which explains the high classification performance with regard to the over-fitting effect. One advantage over SVM is that for comparable generalization performance, it uses fewer kernel functions. This requires less memory and time for processing and so makes possible for the usage of RVM as a real-time classifier.

Another aspect of RVM is that the relevance vectors stand for representative training samples of the emotion classes, rather than data points close to the separation hyper-plane like in the case of SVM model.

### 2.3.2 Boosting methods

Boosting is a term that originates from the literature in the field of computational learning. Generically, the idea stays in combining the performance of many 'weak' classifiers with the goal of obtaining a powerful ensemble or 'committee'.

From the statistical point of view, AdaBoost related classifiers fit in the theory of additive logistic regression models. In that way, an additive function $\sum_m f_m(x)$ is determined so that to approximate $\log p_m(x)/(1-p_m(x))$, where $p_m(x)$ are the class probabilities associated to the classification problem.

#### Adaptive Boosting - Adaboost

Essentially, discrete Adaboost [75] is a method that relates to a two-class classification setup. A *strong* classifier is iteratively obtained by learning several *weak* classifiers that perform just better than random guessing.

If added to the ensemble, each classifier is trained on a data set which is specially generated by applying re-sampling or re-weighting on the instances of the original data set.

The training data set includes a set of instances $(x_1, y_1), ..., (x_N, y_N)$, where $x_i$ is a vector feature and $y_i = -1$ or 1.

By definition, $F(x) = \sum_1^M c_m f_m(x)$, where $c_m$ are constants, represents a linear combination of 'weak' classifiers $f_m(x) : \chi \to \{-1, 1\}$, where $\chi$ is the domain of the features $x_i$.

Each classifier is trained on weighted instances of the initial data set with the property that currently misclassified samples have assigned higher weights.

```
Discrete Adaboost
1. Start with weights $w_i = 1/N \ i = 1, 2, ..., N$.
2. Repeat for m=1,2,...,M:
   a. Estimate $f_m(x) \in \{-1, 1\}$ using the weights $w_i$
      on the training data $x_i$.
   b. Determine $err_m = E_w[1_{(y \neq f_m(x))}]$, $c_m = log((1 - err_m)/err_m)$.
   c. Update $w_i \leftarrow w_i exp[c_m 1_{(y \neq f_m(x))}]$, $i = 1, 2, ..., N$
      renormalise so that $\sum_i w_i = 1$.
3. Output the classifier $sign[\sum_{m=1}^M c_m f_m(x)]$.
```

#### Gentle Adaptive Boosting - GentleBoost

GentleBoost [76] classifier is a stage-wise estimation procedure based on adaptive Newton steps used to fit an additive logistic regression model. An important characteristic of this model is that it performs better than Adaboost in cases when the training data set contains outliers.

#### Adaboost.M2

Adaboost.M2 [75] represents an extension of the binary Adaboost model to multi-class classification setup.

The data set for classification consists of a sequence of N examples $< (x_1, y_1)$, ..., $(x_N, y_N) >$ with labels $y_i \in \{1, ..6\}$. During training, the algorithm makes use

```
Gentle Adaboost
1. Start with weights $w_i = 1/N$ $i = 1, 2, ..., N$. $F(x) = 0$.
2. Repeat for m=1,2,...,M:
    a. Estimate $f_m(x)$ by weighted a fit of $y$ to $x$.
    b. Update $F(x) \leftarrow F(x) + f_m(x)$.
    c. Update $w_i \leftarrow w_i e^{-y_i f_m(x_i)}$.
3. Output the classifier $sign[F(x)] = sign[\sum_{m=1}^{M} f_m(x)]$.
```

of a distribution $D$ over samples and a distribution $w$ over classification labels. These distributions are iteratively re-weighted based on the computation of a pseudo-loss function. Instead of the usual error-based measure, AdaBoost.M2 uses the pseudo-loss function for the selection of the best hypothesis.

```
Adaboost.M2
1. Initialize $D(i) = 1/N$ and $w_{i,y}^{1} = D(i)/(k-1)$ for $i = 1, 2, ..., N$, $y \in Y - \{y_i\}$
2. Repeat for m=1,2,...,M:
    a. Set $W_i^m = \sum_{y \neq y_i} w_{i,y}^m$ and $q_m(i,y) = \frac{w_{i,y}^t}{W_i^m}$ for $y \neq y_i$, $D_m(i) = \frac{W_i^m}{\sum_{i=1}^{N} W_i^m}$
    b. Generate hypothesis $h_m : X \times Y \rightarrow [0,1]$ using $D_m$ and $q_m$.
    c. Compute the pseudo-loss:
            $\varepsilon_m = \frac{1}{2} \sum_{i=1}^{N} D_m(i) \left(1 - h_t(x_i, y_i) + \sum_{y \neq y_i} q_m(i,y) h_t(x_i, y)\right)$
    d. Set $\beta = \frac{\varepsilon_m}{1 - \varepsilon_m}$
    e. Update the weights:
        $w_{i,y}^{m+1} \leftarrow w_{i,y}^{m} \beta_t^{(1/2)(1 + h_m(x_i, y_i) - h_m(x_i, y)}$ for $i = 1, 2, ..., N$, $y \in Y - \{y_i\}$
3. Output the classifier $h_f(x) = \arg\max_{y \in Y} \sum_{m=1}^{M} \left(log \frac{1}{\beta_t}\right) h_t(x, y)$.
```

At each step the pairs of samples and incorrect labels are penalized by increasing the associated weights. This implies the introduction of the pseudo-loss function in the weak classifier with direct implications on the algorithm's speed. The following passage depicts an optimal method for selecting the hypothesis. The pseudo-loss formula depicted in step 2.c. of the Adaboost.M2 algorithm can be rewritten as:

$$
\begin{aligned}
\varepsilon_t &= \frac{1}{2} \sum_{i=1}^{N} D_m(i) \left(1 - h_t(x_i, y_i) + \sum_{y \neq y_i} q_m(i,y) h_t(x_i, y)\right) \\
&= \frac{1}{2} \sum_{y=1}^{6} \left(\sum_{i=1}^{N} D_m(i) g(x_i, y)\right),
\end{aligned}
$$

where $g(x_i, y) = \begin{cases} 1 - h_t(x_i, y_i), & y = y_i \\ q_m(i,y) h_t(x_i, y), & y \neq y_i \end{cases}$.

The optimal pseudo-loss $\varepsilon_t$ value that determines the hypothesis $h_m$, is computed over all the facial expression classes. However, it is plausible to consider determining the contribution of each class separately to $\varepsilon_t$. In this way the minimum value of pseudo-loss is to be achieved by combining six optimal hypotheses each of which minimizes the term $\sum_{i=1}^{N} D_m(i) g(x_i, y)$, where $y \in \{1, .., 6\}$.

In the case the weak classifier is based on decision stumps, the search for best hypothesis per facial expression is firstly conducted for each feature and secondly per set of features.

$$\begin{array}{c|ccccc|}
 & D_m(1) & ... & D_m(c) & ... & D_m(N) \\
Em_1 & & & q_m(c,y)h_m(x_c,y_1) & & \\
\vdots & & & \vdots & & \\
Em_l & & ... & 1-h_m(x_c,y_c) & ... & \\
\vdots & & & \vdots & & \\
Em_6 & & & q_m(c,y)h_m(x_c,y_6) & &
\end{array}.$$

In the next chapters, Adaboost.M2 is used as a multi-class classifier for detecting facial activity and for recognizing facial expressions in single images and in video sequences.

In all these cases, we have used the hypotheses $h_m$ based on decision stumps classifiers. The formulae presented so far allow for a parallel implementation of Adaboost.M2. Multiple threads are able to run simultaneously and to determine the optimal signs and thresholds per emotion class. At iteration $m = 1, ..., M$, each thread is assigned to handle non-overlapping sets of features from the database. Equal sized feature sets imply approximately equal time for the threads to finish their processing. After all the threads have finished, Adaboost.M2 checks all the results and identifies $h_m : X \times Y \rightarrow [0, 1]$ which provides minimum pseudo-loss value $\varepsilon_m$.

The optimal threshold for one feature is determined following an evaluation on the sorted list of the values of the feature for all the samples. Four parameters are calculated: $T^+$ the total sum of positive example weights, $T^-$ the total sum of negative example weights, $S^+$ the sum of positive weights below the current example and $S^-$ the sum of negative weights below the current example.

The optimal threshold corresponds to the example that gives the minimum error:

$$err = min(S^+ + (T^- - S^-), S^- + (T^+ - S^+)).$$

Each term is adjusted according to the distributions $D_m$ and $q_m$. The first term of the minimum function represents the error of considering negative all the samples whose values are below the value of the current sample. Conversely, the second term relates to the error obtained following the assumption that all the samples with values above the value of the current sample are negative. At last, the optimal sign of the operation depends on the term for which the minimum was achieved. Our C++ multi-thread implementation of Adaboost.M2 runs on the Linux operating system.

## 2.4 Conclusion

This chapter has presented the most utilized algorithms for unimodal and multimodal emotion recognition. More precisely, we indicated which techniques are used for detecting faces, extracting features from audio-video data and for the classification of facial expressions and emotion from speech. Additionally, kernel-based SVM and RVM classifiers and Adaboost, GentleBoost and Adaboost.M2 boosting methods have been presented. On the practical side, we presented a usable method to optimize the Adaboost.M2 algorithm in a way that permits the development of a parallel C++ implementation.

Each classification approach has its own advantages and limitations. Anyway, the model performance also depends on other factors such as the quality of the data. In the next chapters, these methods are investigated practically in different classification setups for recognizing emotions.

# Chapter 3

# Active Appearance Models for face analysis

## 3.1 Introduction

Beside data acquisition, any attempt on studying facial expressions requires appropriate methods for preparing face images. The quality of the results at this step essentially influences the performance of subsequent recognition. Most of the previous research which used facial expression databases (including the data set we have focused on in this research), applied basic image-oriented procedures to localise and prepare the face data for the expression analysis. In several cases, the outcome consisted of face images that contain parts of background, subject's hair or cloth. In some other cases, the images especially those showing exaggerate expressions of surprise and fear, were clipped, producing face samples which did not contain the lower parts of the faces. Extracting and using visual features from such data is risky and should be normally avoided.

This chapter describes a technique that accurately extracts, aligns and scales face instances from images.
Active appearance model - AAM [55] is a statistical method that handles shape and texture variations of photo-realistic appearance. Seen as a top-down approach, the AAM makes use of prior knowledge on the grey-level appearance, shape structures as well as their relationships, in order to build generative models for the global analysis of a specific class of objects. In the context of facial expression recognition, AAM is used for automatically computing the appearance of faces and of facial features i.e. mouth, eyes, etc.
In the next sections, we first present the theoretical aspects of AAM and show how this method can be used as part of the more complex facial expression recognition process. Furthermore, we describe a procedure for selecting face samples from a commonly used facial expression database, based on an in-depth analysis of their action unit labels [56]. In this approach, the preparation of the face data set is then completed by applying AAM on the selected face images.

## 3.2 Extracting the face appearance through AAM search

The extraction of shape and texture of the face from an image is equivalent to an optimization problem that involves the criterion of minimizing the difference between the real face image and the one generated by the appearance model. The distance measure can be written as follows:

$$r(p) = g_{im} - g_m,$$

where $g_{im}$ is the vector of grey level values of the face patch in the input image and $g_m$ is the vector of grey level values for the face image as it is estimated by the current model parameters $p$.
The matching implies finding the optimal appearance model parameters which would lead to the minimization of a scalar measure on the image difference, such as the sum of squares of elements $E(p) = r^T r$.

## 3.3 Preparing the data set

Building a face appearance model implies a prior acquisition of the face data set. Each face sample has to be consistently annotated with a set of landmark points. The selection of these points is primarily based on the location of facial feature boundaries, at 'T' junctions between boundaries or other relevant face areas.
Each sample of face shape is described using $n$ 2D landmark points $(x_i, y_i)$ by a shape vector:

$$x = (x_1, ..., x_n, y_1, ..., y_n)^T.$$

An example of face shape annotation is illustrated in figure 3.1. The face shape data set is formed by concatenating the face shape vectors $x_i$ of all the face samples. Further statistical analysis is applied so as to obtain shape data samples represented in the same coordinate system.

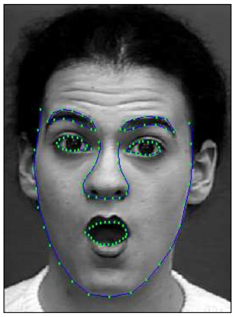


Figure 3.1: An annotation example of a face image that contains 122 landmark points.

### 3.3.1 Alignment of the face shape data set

A requirement of applying the statistical analysis is the alignment of shape data into a common co-ordinate frame. A common approach in obtaining aligned face shapes is to use Procrustes analysis - PA [80]. According to this method, the shapes are aligned in such way to minimize the sum of distances of each shape to the mean shape ($D = \sum |x_i - \bar{x}|^2$). A more precise definition introduces constraints regarding the fixed shape centre set to the origin, unit scale size $|x| = 1$ and specific orientations. The scaling constraint assumes that each aligned shape $x$ lies on a hyper-sphere.
A second method to achieve face shape alignment could allow to vary both scaling and orientation during the minimization of $D$. If $T_{s,\theta}(x)$ is an operator which applies scaling by $s$ and rotation by $\theta$ to the shape $x$, then the alignment of two face shapes, each centred on the origin ($x_1.1 = x_2.1 = 0$) implies the choice for a scale $s$ and rotation $\theta$ so that $|T_{s,\theta}(x_1 - x_2)|^2$ is minimized.
Another alignment method is to transform each face shape in the tangent space to the mean and to minimize the term $D$. The tangent space to $x_t$ is defined as the hyper-plane of vectors normal to $x_t$ and passing through $x_t$. The formulation leads to all the vectors $x$ for which the following relations hold: $(x_t - x).x_t = 0$ or $x.x_t = 1$ if $|x_t| = 1$.

### 3.3.2 Normalization of the face texture data set

The first step of building statistical models of the texture is the transformation of all the image samples so that their landmark points match the landmark points of the mean shape $\bar{x}$. Then, shape-normalized texture vectors $g$ are obtained by firstly using a triangulation algorithm on the shape data and secondly by warping the texture in the region between the set of landmark points in one face image to the correspondent region of the mean shape, for all the triangular regions. The texture data set is generated by sampling intensity information from the shape-normalized texture vectors in all the face images.
Further more, the texture data vectors are normalized by applying a scaling $\alpha$ and offset $\beta$ so as to reduce the effect of global lighting variation.

$$g = (g_{im} - \beta 1)/\alpha.$$

Finding the values of the parameters $\alpha$ and $\beta$ follows a recursive process in which one of the data set samples is used as a first estimate of the mean $\bar{g}$, then iteratively aligning the other samples to the estimated mean and re-estimating the mean. At each iteration, the values of $\alpha$ and $\beta$ are computed using the following formulae:

$$\begin{aligned} \alpha &= g_{im}\bar{g} \\ \beta &= (g_{im}.1)/m, \end{aligned}$$

where m is the number of elements in the vectors. An example of mean face shape and mean face texture is depicted in figure 3.2. At the end of the shape and texture data-preprocessing step, each face image sample has acquired an aligned shape vector and a texture vector of the shape-free face image patch. For the face image example in figure 3.1, the two types of data are illustrated in figure 3.3.

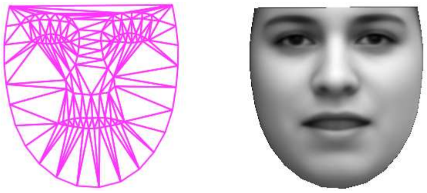


Figure 3.2: Average shape (left) and average shape-free texture (right).

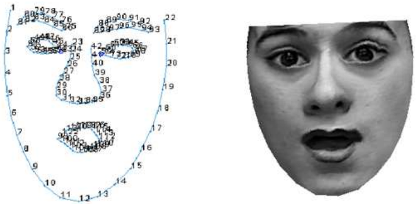


Figure 3.3: Shape data (left) and shape-free texture data (right) for the previously depicted example.

## 3.4 Learning the model parameters

### 3.4.1 Statistical models of shape variation

The goal of the analysis on shape variation is to model the distribution of shape vectors $x$ in the $2n$ dimensional space. The effect of applying principal component analysis - PCA on the shape data is the reduction of the shape dimensionality. This is achieved by identifying the main axes of the cloud of points formed by shape vectors in the original $2n$ space. Eventually, any initial shape can be approximated by using less than $2n$ parameters.

The procedure involves first, the computation of the mean $\bar{x}$ and the covariance $S$ of the data. Next, the set of eigenvectors $\{p_s^i\}$ and the associated set of eigenvalues $\{b_s^i\}$ are determined and sorted in the descendent order of the eigenvalues, so that $b_s^i \geq b_s^{i+1}$. Then, the shape data instances $x$ may be obtained using the formula:

$$x = \bar{x} + P_s b_s,$$

where $\bar{x}$ is the mean shape and $P_s = (p_s^1, p_s^2, ..., p_s^t)$. The $t$ dimensional vector $b_s$ defines a set of parameters of a deformable model and can be written as following: $b_s = P_s^T(x - \bar{x})$.

The number of parameters $t$ is chosen so that to obtain a desired proportion $f_v$ of explaining the total variation of the shape model: $\sum_{i=1}^{t} b_s^i \geq f_v V_T$. The term $V_T$ is the total variance in the data and is computed as the sum of all the eigenvalues $V_T = \sum b_s^i$.

An alternative method for determining $t$ is to include shape modes which represent models that approximate any initial shape sample with a given minimum accuracy. This can be done by building different models with increasing number of modes and by choosing the first model that can best approximate the shape data.

### 3.4.2 Statistical models of texture variation

Similarly, a PCA oriented statistical analysis on the texture data gives the following linear model:

$$g = \bar{g} + P_g b_g.$$

The term $\bar{g}$ is the mean normalized grey level vector, $P_g$ is a set of orthogonal modes of grey level variation and $b_g$ is the grey level model parameters.

### 3.4.3 Combined appearance models

The vectors $b_s$ and $b_g$ successfully represent the shape and appearance of the initial face samples. The two models of shape and texture can be combined into an appearance model of the 2D face object:

$$b = \left( \begin{array}{c} W_s b_s \\ b_g \end{array} \right) = \left( \begin{array}{c} W_s P_s^T (x - \bar{x}) \\ P_g^T (g - \bar{g}) \end{array} \right).$$

Because the vector $b_s$ has units of distance and $b_g$ has units of intensity, the term $W_s$ is introduced as the weight that compensates for the difference between the two data types. It may be computed through the ratio $r^2$ of total intensity variation to the total shape variation, where $W_s = rI$ or by displacing the elements of the shape vector $b_s$ from the optimum value and by computing the correspondent RMS difference in the grey level vectors.
Applying PCA on the concatenated vectors $b$ leads to the following formulation:

$$b = P_c c = \left( \begin{array}{c} P_{cs} \\ P_{cg} \end{array} \right) c,$$

where $c$ is the vector of appearance parameters. The appearance model may be written as:

$$\begin{array}{rcl} x & = & \bar{x} + P_s {W_s}^{-1} P_{cs} c \\ g & = & \bar{g} + P_g P_{cg} c, \end{array}$$

where $P_{cs}$ and $P_{cg}$ are matrices describing the modes of variation.

### 3.4.4 Synthesizing face samples with AAM

Given the shape model, a new shape $X$ can be obtained by using a transformation $S_t$: $X = S_t(x)$. The term $S_t$ depends on the scaling $s$, an in-plane rotation $\theta$ and a translation $(t_x, t_y)$. For a specific instance of the appearance parameter $c$, the shape and texture may be synthesized using the formulae:

$$\begin{array}{rcl} x & = & \bar{x} + Q_s c \\ g & = & \bar{g} + Q_g c \end{array}$$

and:

$$\begin{aligned} Q_s &= P_s {W_s}^{-1} P_{cs} \\ Q_g &= P_g P_{cg}. \end{aligned}$$

Subsequently, the transform $S_t$ is applied for adjusting the location of each model point according to the position, orientation and scale, for all points $i = 1, ..., n$:

$$S_t \begin{pmatrix} x_i \\ y_i \end{pmatrix} = \begin{pmatrix} t_x \\ t_y \end{pmatrix} + \begin{pmatrix} s \cdot cos\theta & -s \cdot sin\theta \\ s \cdot sin\theta & s \cdot cos\theta \end{pmatrix} \begin{pmatrix} x_i \\ y_i \end{pmatrix}.$$

A new texture can be obtained by applying a transform $T_u$ on the image intensities: $T_u(g) = \alpha \cdot g + \beta 1$, where $\alpha$ and $\beta$ are the scaling and offset parameters of the transformation.

### 3.4.5 AAM matching

The appearance model parameters $c$ and shape transformation parameters $t$ identify the position of the model points in the image. For matching, pixels are sampled around the image region $g_{im}$ and then the texture model is computed $g_s = T_u^{-1}(g_{im})$. The model texture can be written as: $g_m = \bar{g} + Q_g c$ and so the difference between the model and the image may be written as follows: $r(p) = g_s - g_m$ and $p^T = (c^T, t^T, u^T)$.
The Taylor expansion for the previous expression gives:

$$r(p + \delta p) = r(p) + \frac{\partial r}{\partial p} \delta p.$$

The term $\frac{\partial r}{\partial p}$ represents a matrix made up by $ijth$ elements $\frac{dr_i}{dp_j}$:

$$\frac{\partial r}{\partial p} = \begin{bmatrix} \frac{dr_1}{dp_1} & \cdots & \frac{dr_1}{dp_Q} \\ \vdots & & \vdots \\ \frac{dr_M}{dp_1} & \cdots & \frac{dr_M}{dp_Q} \end{bmatrix}.$$

The fitting procedure assumes that for a specific current residual $r$, a $p$ is chosen so that the value of $|r(p + \delta p)|^2$ is minimum. The choice has the form: $\delta p = -Rr(p)$, where:

$$R = (\frac{\partial r}{\partial p}^T \frac{\partial r}{\partial p})^{-1} \frac{\partial r}{\partial p}^T.$$

The term $\frac{\partial r}{\partial p}$ is computed once, during the training step, by altering the elements of $p$ with some amount (typically up to 0.5 standard deviation) and by computing the residuals for the face images included in the training data set. A Gaussian weighting function $w(x)$ is used for further smoothing the residuals:

$$\frac{dr_i}{dp_j} = \sum_k w(\delta c_{jk})(r_i(p + \delta c_{jk}) - r_i(p)).$$

The data set used for training can include both real face images and samples obtained by synthesizing new faces based on the current model parameters.

The model face is fit to the image face during an iterative procedure that alters the model parameters $c$, pose $t$ and texture transformation $u$ until the error value fits into an acceptable range:

**AAM fitting**

1. Project the texture sample into the texture model frame using $g_s = T_u^{-1}(g_{im})$.
2. Compute error vector $r = g_s - g_m$ and current error: $E = |r|^2$.
3. Determine the estimated displacements: $\delta p = -R \cdot r(p)$.
4. Update model parameters $p \longrightarrow p + k\delta p$; initially $k = 1$.
5. Compute new points $X'$ and model frame texture $g'_m$.
6. Sample image at new points and get $g'_{im}$.
7. Determine new error vector $r' = T_{u'}^{-1}(g'_{im}) - g'_m$.
8. Repeat the previous steps with $k = 0.5$, $k = 0.25$, etc., as long as $|r'| \geq E$.

## 3.5 Results of AAM on face data

The AAM training data set consists in 317 manually annotated frontal face images. Each face image sample contains 122 landmark points which are located along the shape contours of the face, the nose, the mouth, the eyes and the eyebrows. The face vectors are represented in the shape-free texture space using 45.117 pixel grey-level values.
Setting the proportion $f_v$ of explaining the total data variation $V_T$, to 95% for each of the shape, texture and appearance models, leads to a set of 30 modes corresponding to the shape model, a set of 137 modes corresponding to the texture model and a set of 68 modes corresponding to the combined model.
Varying the first six modes of variation of the appearance model with ±3 standard deviation, results to synthesized face images which are illustrated in figure 3.4.

Table 3.1 shows the amount of separate data variation as well as the accumulated data variation that is explained by the first ten modes of each face data model. The same type of results are depicted as overlapping contribution graphics for all the modes of each model, in figure 3.5.

The experiments involved also the study of the performance induced by setting different variance thresholds. Table 3.2 shows the results achieved by different appearance models obtained by training on data sets of various sizes and variance thresholds.

## 3.6 Case studies. Testing the face model

In this section, we run a series of experiments that aim at studying the robustness of the AAM model in conditions of translation, rotation and scaling.

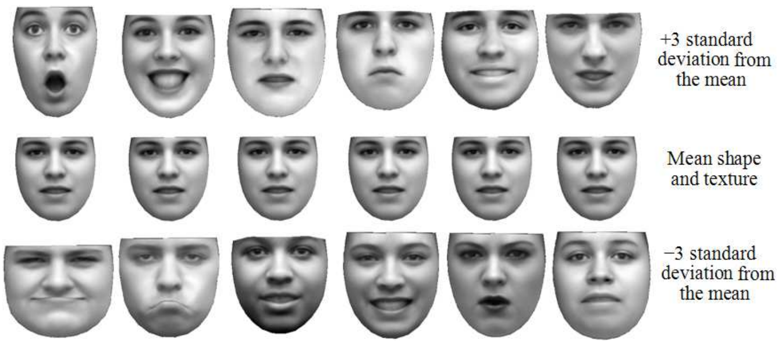

+3 standard deviation from the mean
Mean shape and texture
−3 standard deviation from the mean

Figure 3.4: The effect of changing the first six appearance modes with ±3 standard deviation from the mean shape and texture.

Table 3.1: Contribution of the largest eigenvalues of the shape, texture and combined models towards explaining the total variance in the face data.

| Mode | Variance | Acc.variance |
|---|---|---|
| 1 | 33.71% | 33.71% |
| 2 | 12.41% | 46.11% |
| 3 | 7.72% | 53.83% |
| 4 | 6.38% | 60.21% |
| 5 | 5.84% | 66.04% |
| 6 | 5.21% | 71.25% |
| 7 | 3.58% | 74.83% |
| 8 | 3.30% | 78.13% |
| 9 | 2.21% | 80.34% |
| 10 | 1.81% | 82.15% |

Shape model

| Mode | Variance | Acc.variance |
|---|---|---|
| 1 | 14.45% | 14.45% |
| 2 | 11.07% | 25.53% |
| 3 | 7.32% | 32.84% |
| 4 | 6.23% | 39.07% |
| 5 | 4.86% | 43.93% |
| 6 | 4.57% | 48.50% |
| 7 | 2.95% | 51.45% |
| 8 | 2.57% | 54.02% |
| 9 | 2.29% | 56.31% |
| 10 | 2.22% | 58.53% |

Texture model

| Mode | Variance | Acc.variance |
|---|---|---|
| 1 | 20.37% | 20.37% |
| 2 | 9.07% | 29.44% |
| 3 | 7.57% | 37.01% |
| 4 | 6.73% | 43.74% |
| 5 | 5.68% | 49.42% |
| 6 | 4.78% | 54.21% |
| 7 | 4.43% | 58.64% |
| 8 | 3.22% | 61.86% |
| 9 | 2.43% | 64.28% |
| 10 | 2.34% | 66.62% |

Combined model

For testing, a set of six previously annotated frontal face images were used. These samples were not included in the AAM training data set and relate to face postures of one subject showing the six prototypic facial expressions. Similarly, there are no other face samples of the same subject being considered as part of the training data set.

The testing consists of displacing characteristic shape parameters and in measuring the capability of the model to estimate these changes. All the displacements done for the experiments take as reference point the pixel located at the middle of the line segment delimited by the locations of the inner corners of the eyes. Figure 3.6 and 3.7 show the results of estimating the translation along the $X$

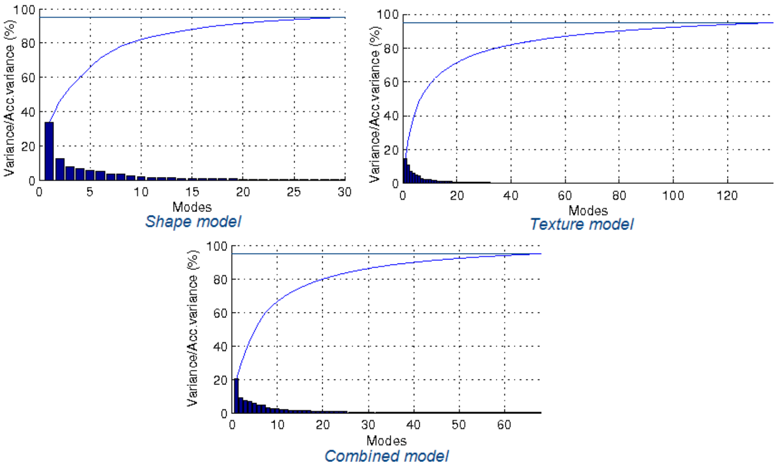


Figure 3.5: The amount of variance/accumulated explanation for each mode of variation for the shape, the texture and the combined models.

Table 3.2: The number of shape modes (SM), the number of texture modes (TM) and the number of combined model modes (CM) of AAM models trained with data sets of different sizes (NS:number of face samples) and for different thresholds of variance explanation (VAR).

| NS | variation **90%** | | | variation **95%** | | | variation **98%** | | |
|---|---|---|---|---|---|---|---|---|---|
| | sm | tm | cm | sm | tm | cm | sm | tm | cm |
| 50 | 13 | 27 | 17 | 20 | 36 | 28 | 28 | 43 | 38 |
| 75 | 15 | 36 | 21 | 22 | 49 | 35 | 35 | 61 | 52 |
| 100 | 15 | 44 | 23 | 23 | 62 | 41 | 39 | 79 | 65 |
| 125 | 16 | 50 | 25 | 25 | 73 | 46 | 43 | 96 | 77 |
| 150 | 16 | 56 | 26 | 25 | 84 | 50 | 45 | 113 | 88 |
| 175 | 16 | 60 | 27 | 26 | 93 | 53 | 48 | 128 | 97 |
| 200 | 17 | 65 | 28 | 27 | 102 | 57 | 50 | 143 | 107 |
| 225 | 17 | 68 | 28 | 28 | 110 | 59 | 52 | 158 | 115 |
| 250 | 17 | 72 | 29 | 28 | 118 | 62 | 53 | 171 | 123 |
| 275 | 18 | 75 | 30 | 29 | 125 | 64 | 55 | 185 | 131 |
| 300 | 18 | 79 | 30 | 29 | 133 | 67 | 56 | 199 | 139 |
| 317 | 18 | 80 | 30 | 30 | 137 | 68 | 56 | 207 | 143 |

and $Y$ axes. The displacement values are reported in percentage relative to the face width (which is about 229 pixels, as it is measured from the left to the right face shape boundaries, along the horizontal eyes' axis). The results relate to average measurements of pixel displacements per face shape landmark points. The model can handle translation on both sides from the true face shape position, up to 20% (about 45 pixels) along $X$ axis and up to 12% (about 27 pixels) along $Y$ axis.

In the same way, from figure 3.8 it can be concluded that changes in the face shape scale are correctable in the interval from 0.7 the original face width (about 160 pixels) up to 1.4 the original face widh (about 320 pixels).

Figure 3.9 illustrates the graphic of the shape rotation estimation results.

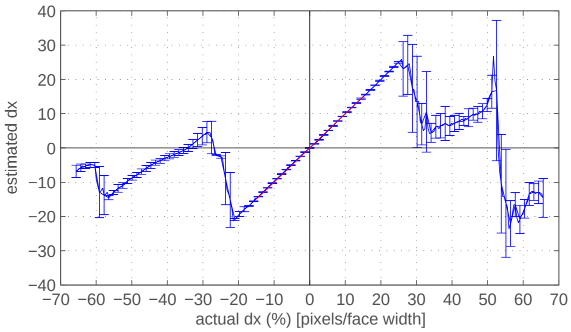


Figure 3.6: Performance of the model in predicting $dx$. Error bars are 1 standard deviation.

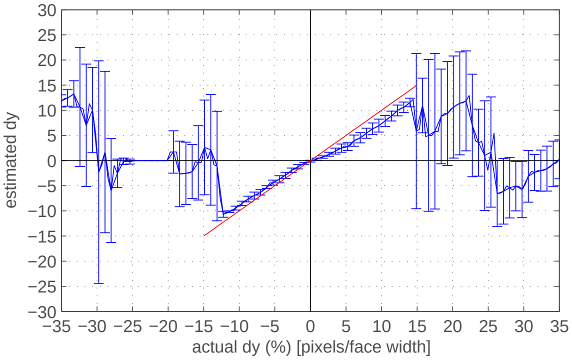


Figure 3.7: Performance of the model in predicting $dy$. Error bars are 1 standard deviation.

Each face shape has been systematically rotated with angles in the range [-90°,+90°] around the centre point of the line segment delimited by the eye positions. The results show that the model can adjust face shape rotations of 30° at both sides from the vertical line, around the eyes' centre point.

Another type of experiments is aimed at measuring the extent to which the previously developed appearance model can estimate concurrent changes of the shape parameters. The results are given as the root mean square (RMS) error per texture pixel, after comparing the image face with the model face in the reference texture space.
Figure 3.10 represents the RMS map in the case of translating the initial face shape on $X$ and $Y$ axes. In addition, figure 3.11 represents the RMS map for simultaneous translation on $X$ axis and scaling. Figure 3.12 shows the RMS error per face texture pixel in case of displacing the face shape along the $X$ axis and applying different degrees of rotation in the same time.
The dark spots on the map reflect areas regarding pairs of displacements with high performance in recovering the face texture.

These results are comparable to the results of the previous experiments that used the average displacement of the shape in pixel units. Translations can be

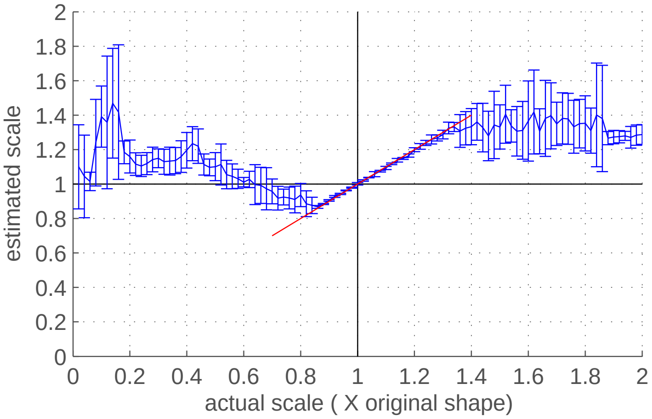


Figure 3.8: Performance of the model in predicting scale. Error bars are 1 standard deviation.

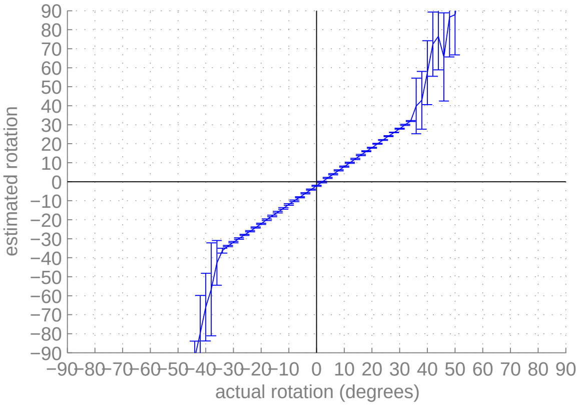


Figure 3.9: Performance of the model in predicting rotation. Error bars are 1 standard deviation.

handled with high accuracy up to a degree of 20% the original shape width. The same holds for rotations of up to approximately 35° on both sides from the vertical axis.

The range accounting for the recoverable amount of translation, rotation and scaling can be extended in two different ways that possibly can be used together.

The first solution consists in creating multi-resolution face appearance models. Separate AAM models are first built using training samples which correspond to levels of a Gaussian pyramid which is initially generated based on the initial face data set. During testing, the optimal AAM fit is given as the combined result of applying basic AAM search on each level of the pyramid.

The alternative method for improving the robustness of AAM search is to use a prior face detection model. In this case, the result obtained by applying the

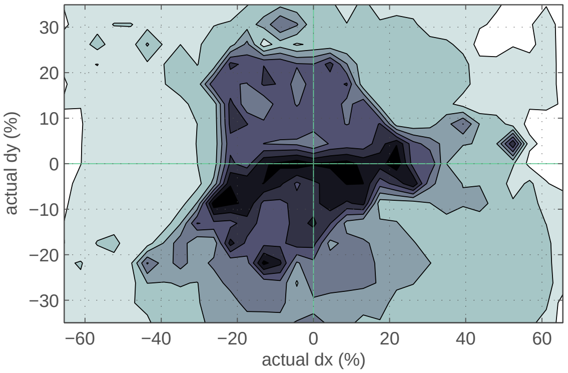


Figure 3.10: RMS intensity error/pixel between the modelled and the true appearances for predicting displacements of $dx$ and $dy$ simultaneously.

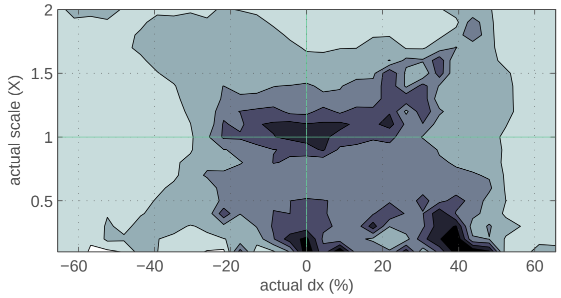


Figure 3.11: RMS intensity error/pixel between the modelled and the true appearances for predicting displacements of $dx$ and scale simultaneously.

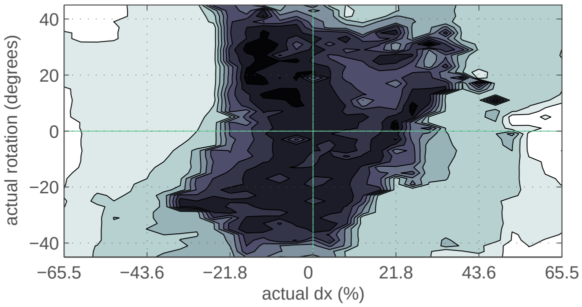


Figure 3.12: RMS intensity error/pixel between the modelled and the true appearances for predicting displacements of $dx$ and rotation together.

detection of faces in the image may be used for the AAM search as an initial estimate on the AAM pose parameters.

Furthermore, more robust AAM appearance models may be developed so as to handle wide ranges of head rotation. As an example, the so-called coupled-view appearance models attempt to model the relationship between the model parameters in different views (such as frontal and side views). Other derivations of AAM have the goal of modelling the 3D appearance of the face.

## 3.7 Database preparation

Prior to developing models for facial expression recognition, the video database of faces had to be prepared.

For training and testing the models, we have used data samples from the Cohn-Kanade database [103]. The database contains video samples of subjects ranging in age from 18 to 30 years. Sixty-five percent were female; 15 percent were African-American and three percent Asian or Latino. Image sequences from neutral to target display were digitized into 640 by 480 or 490 pixel arrays with 8-bit precision for grey-scale values. Subjects show various facial displays during different recording sessions. In total, the database contains 486 video samples. The last frame of each video sequence has attached a label of action units which characterizes the facial muscle activity of the subject at the moment. The labels contain single or combined action units chosen from a set of 37 action units. From the action unit set, 21.62% of the elements relate to upper facial action units, 45.95% relate to lower facial action units, 13.51% relate to action units which indicate the head and eye positions and 18.92% relate to miscellaneous actions and supplementary codes. Some of the labels include action units from a subset of 27 action units, which have intensity levels.

The first analysis of the Cohn-Kanade database leads to the conclusion that not all the video samples can be used for the modelling of facial expressions. The reasons regard the rather low quality of some of the frame images for some videos, the high amount of head rotation exhibited by some subjects during the recordings and the occlusion of parts of the face area in some frames. Mostly, the occlusion is due to textual information displayed on the low part of the image.

The data set we have obtained from the original Cohn-Kanade database contains 480 video samples which include 168 video samples of male subjects (35.00%) and 312 video samples of female subjects (65.00%). The selected recordings from the data set have been collected from 95 subjects out of which 34 are males (35.79%) and 61 are females (64.21%). Each video sample shows the generation of one facial expression as a transition from the neutral state (first frame) to the correspondent apex (the last frame of the sequence). The length of transition ranges from 3 frames to 65 frames. On average, expression fear is fully displayed after approximately 14 frames, expression surprise after 16 frames, expression sadness after 17 frames, expression anger after 21 frames, expression disgust after 18 frames and expression happiness after 19 frames.

The disadvantage of the Cohn-Kanade database is the lack of facial expression labels. In the next section, we propose a method for converting action unit labels to facial expressions.

Before beginning the analysis of facial expressions, we are interested to get an overview of the complexity of the data. First, we use the location of specific landmark points from the shape of the face, eyes, eyebrows and mouth as features. In total, the data set contains 94 landmark points for each face sample.

The face instances are normalized using the distance between the inner corners of the eyes. First, we want to get an idea about the complexity of the classification problem. More specifically, we try to obtain a visualization of the face samples, relative to the average face. Applying principal component analysis on the dataset and using only the first two most important eigenvectors (82.08% variation explanation), we obtain a 2D representation of all the faces (figure 3.13). The figure indicates a significant overlap of the point clouds, given the emotion categories. Table 3.3 represents the RMS for every shape pixel of the face and facial features between the face instances and the mean face by considering emotion categories and other displays. The values reflect the statistics in case of the dataset of faces scaled to the reference distance of 10 pixels between the inner corners of the eyes. Secondly, we employ the Shannon entropy to measure the distance of face instances from the mean face. Entropy uses the uncertainty as a measure to describe the information contained in a source. Maximum information is obtained in the case no a priori information is available. This is equivalent to saying that all samples are equally important and that the probability distribution describing the experiment is either uniformly distributed in continuous probability space or equally likely in discrete probability space.

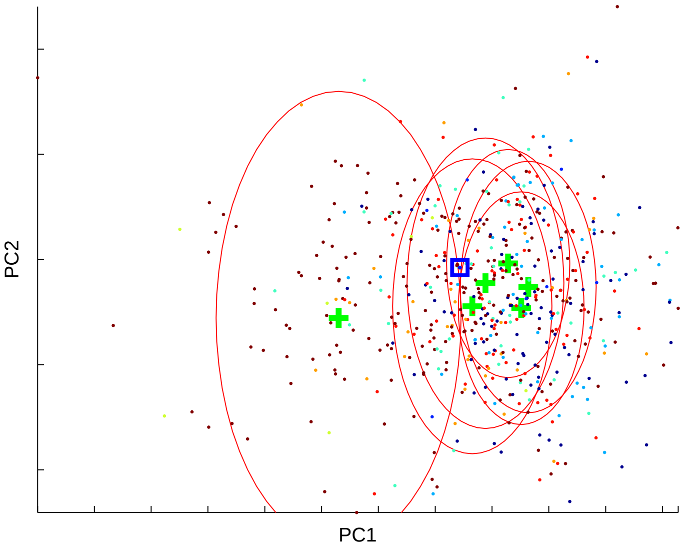


Figure 3.13: 2D projection of face samples from the Cohn-Kanade database using the first two principal components.

Histograms are computed from non-overlapping regions of each normalized face image, using the 3X3 local binary pattern - LBP operator introduced by Ojala et al. [139]. This operator can encode image micro-patterns and has successfully proved to be very efficient for texture analysis [140]. The approach considers the image LBP histogram as a probability distribution. The bin elements of the normalized concatenated histograms are regarded as probabilities $p_i$ of finding specific micro-patterns in the image. Then, the entropy of the face $I$ may be determined using the following formula:

$$H(I) = -\sum_{i=1}^{N} p_i \log p_i,$$

where $N$ is the number of bins. Table 3.3 shows the mean and standard deviation of the absolute difference between the entropies of the face instances and the mean face from Cohn-Kanade dataset.

| **category** | **min** | **max** | **mean** | **std** | **mean $\Delta H$** | **std $\Delta H$** |
|---|---|---|---|---|---|---|
| other displays | 12.39 | 23.64 | 16.70 | 1.76 | | |
| neutral | 13.17 | 18.73 | 15.90 | 1.10 | 1.20 | 0.35 |
| fear | 12.96 | 19.85 | 16.30 | 1.48 | 1.14 | 0.34 |
| surprise | 15.48 | 21.43 | 18.45 | 2.17 | 1.20 | 0.44 |
| sadness | 13.00 | 20.32 | 16.26 | 1.53 | 1.10 | 0.32 |
| anger | 12.93 | 18.75 | 15.49 | 1.28 | 0.99 | 0.40 |
| disgust | 14.32 | 17.45 | 15.86 | 1.29 | 0.82 | 0.56 |
| happy | 12.69 | 19.39 | 15.66 | 1.22 | 1.01 | 0.31 |
| all samples | 12.39 | 23.64 | 16.23 | 1.58 | | |

Table 3.3: Measures of the RMS between the face instances and the mean face.

### 3.7.1 Facial expression labelling

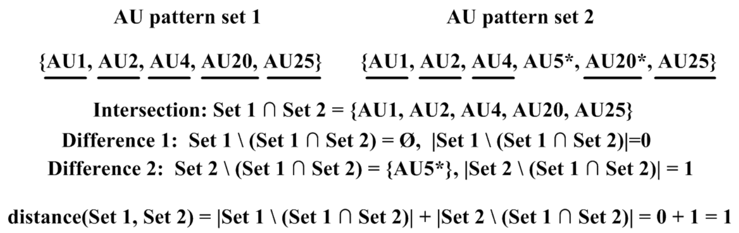
In order to assign labels of facial expressions to the face images from the Cohn-Kanade database, we use the original action unit labels in the database and the rules for converting patterns of action units to facial expression classes, as indicated in Facial Action Coding System Investigator's Guide [57] (see appendix B).

For each face image, the labelling task has to assign the facial expression label of the most appropriate action unit pattern. A distance measure is used to ease the matching between action unit patterns.

Given two pattern sets of action units, a convenient distance function takes into account the intersection set and two sets computed as the set theoretic differences between each of the pattern sets and the intersection set. The output of the function is the sum of the two numbers representing the cardinalities of the two difference sets. Figure 3.14 illustrates an example of distance function calculations. Finding the optimal action unit pattern that best matches a face sample action unit label, implies to first search for patterns that have maximum cardinality of the intersection set. For that, we consider only the AU codes and disregard the representations for the AU intensity. The procedure is detailed in the box below.

Applying this distance function for determining the facial expression class labels for the faces in the final image frames of the video samples in the Cohn-Kanade database, leads to the following observations.

Only a very small number of face images can be labelled as a result of a perfect match between the action unit code and a FACS action unit pattern. That means that the rest of the samples are labelled by analysing their distance functions.



**AU pattern set 1** **AU pattern set 2**

**{AU1, AU2, AU4, AU20, AU25}** **{AU1, AU2, AU4, AU5*, AU20*, AU25}**

**Intersection: Set 1 ∩ Set 2 = {AU1, AU2, AU4, AU20, AU25}**
**Difference 1: Set 1 \ (Set 1 ∩ Set 2) = Ø, |Set 1 \ (Set 1 ∩ Set 2)|=0**
**Difference 2: Set 2 \ (Set 1 ∩ Set 2) = {AU5*}, |Set 2 \ (Set 1 ∩ Set 2)| = 1**

**distance(Set 1, Set 2) = |Set 1 \ (Set 1 ∩ Set 2)| + |Set 2 \ (Set 1 ∩ Set 2)| = 0 + 1 = 1**

Figure 3.14: The distance function between two action unit patterns.

```
Action unit-based procedure for facial expression labelling
for each face image,
    - let $Set_1$ be the set of action units that correspond to the face image label
    - for each FACS action unit pattern
        - let $Set_2$ be the set of action units that correspond to the pattern
        - compute the intersection set $Set_1 \cap Set_2$
        - compute the $distance(Set_1, Set_2)$
    - find $Set_1$ and $Set_2$ with maximum $|Set_1 \cap Set_2|$ and minimum $distance(Set_1, Set_2)$
    - assign the face image with the facial expression class label associated to $Set_2$
```

Subsequently, there are image samples which present the same value of the distance function for different FACS patterns. Some of these FACS patterns relate to different facial expression classes. As an example, assume the label for a given face image contains action units 1, 2, 25 and 27. According to the FACS rules, there are two patterns of action units which show the same matching distance. The patterns are variants of different facial expression classes. While the first pattern contains action units 1, 2, 5Z, 25 and 27 and corresponds to class fear, the second pattern contains action units 1, 2 and 27 and corresponds to class surprise.

In such cases, we say that the facial expression labelling is ambiguous. Another interpretation may be that the two labels correspond to image samples of faces, each of them showing mixed facial expressions, in this example fear and surprise. In this research, we assume that these samples reflect mixed facial expressions. Table 3.4 and table 3.5 show the number of face image samples for each facial expression category, which were labelled using the AU code matching procedure. In table 3.5, the terms $M2$, $M3$ and $M4$ indicate mixtures of respectively two, three and four facial expression categories. It may be noticed that only 257 face image samples can be assigned to unique classes of basic facial expressions.

The non-ambiguous set accounts for 53.77% of the Cohn-Kanade database. The rest of 221 face image samples clearly represent hard to label facial expressions. They have the same distance value to FACS patterns of different facial expressions. Then, there are 180 samples showing mixtures of 2 facial expressions, 40 samples that show mixtures of 3 facial expressions and one sample showing a mixture of 4 facial expressions.

In table 3.6, the instances of each facial expression are structured according to the distance measure and to the number of mixed facial expressions. Table 3.7 shows the number of face samples for each type of mixture of two facial expressions. From the set of samples consisting of mixtures of 3 facial expressions, 62.50% of the face samples relate to mixtures of fear, surprise and disgust, 30.00% of the face samples relate to mixtures of sadness, anger and disgust and

Table 3.4: The number of samples that present non-ambiguous match between the AU label and FACS action unit prototypes.

| **Distance** | **Fear** | **Surprise** | **Sadness** | **Anger** | **Disgust** | **Happy** | **Total** |
|---|---|---|---|---|---|---|---|
| 0 | **2** | 0 | 0 | 0 | 0 | **18** | **20** |
| 1 | **9** | **6** | **4** | **4** | 0 | **56** | **79** |
| 2 | **14** | **1** | **24** | **13** | **6** | **22** | **80** |
| 3 | **6** | 0 | **13** | **12** | **1** | **4** | **36** |
| 4 | **6** | 0 | **7** | **18** | 0 | 0 | **31** |
| 5 | **3** | 0 | 0 | **8** | 0 | 0 | **11** |
| 6 | 0 | 0 | 0 | 0 | 0 | 0 | **0** |
| 7 | 0 | 0 | 0 | 0 | 0 | 0 | **0** |
| 8 | 0 | 0 | 0 | 0 | 0 | 0 | **0** |
| **Total** | **40** | **7** | **48** | **55** | **7** | **100** | **257** |

Table 3.5: The number of samples of mixed facial expressions. Mixed samples are counted for every constituent facial expression.

| **Dist.** | **Fear** | **Surprise** | **Sadness** | **Anger** | **Disgust** | **Happy** | M2 | M3 | M4 |
|---|---|---|---|---|---|---|---|---|---|
| 1 | **10** | **10** | **4** | 0 | **4** | 0 | 28 | 0 | 0 |
| 2 | **110** | **74** | **25** | **12** | **86** | 0 | 232 | 75 | 0 |
| 3 | **28** | **19** | **14** | **14** | **32** | **1** | 74 | 30 | 4 |
| 4 | **5** | **2** | **7** | **4** | **4** | **1** | 14 | 9 | 0 |
| 5 | **2** | 0 | **5** | **6** | **2** | **1** | 10 | 6 | 0 |
| 6 | 0 | 0 | **1** | **1** | 0 | 0 | 2 | 0 | 0 |
| 7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

7.50% of the face samples relate to mixtures of fear, surprise and sadness. The face sample showing a mixture of 4 facial expressions relates to fear, surprise, disgust and happiness.

The non-ambiguous facial expression images form a database of 257 samples. Checking for high quality images, we have selected a set of 251 samples out of the 257 samples. Table 3.8 shows the structure of this data set. A first observation on this table suggests the unbalanced character of the data set. Figure 3.15 shows the proportion of face samples for each gender. In-depth graphical representations of the number of frames associated to the transition from neutral to fully expressed facial expressions, are depicted in figure 3.16.

The conclusion of using the labelling procedure is that the Cohn-Kanade database contains face images of subjects showing pure, mixed or no facial expressions. By following FACS rules, such face image samples of mixed facial expression are hard to be labelled using unique classes in the range of the six

Table 3.6: The number of samples of mixed facial expressions. Mixed samples are counted for every constituent facial expression.

| **Dist.** | **Fear** | | **Surprise** | | **Sadness** | | **Anger** | | **Disgust** | | **Happy** | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | M2 | M3 | M2 | M3 | M2 | M3 | M2 | M3 | M2 | M3 | M2 | M3 |
| 1 | 10 | 0 | 10 | 0 | 4 | 0 | 0 | 0 | 4 | 0 | 0 | 0 |
| 2 | 90 | 20 | 54 | 20 | 20 | 5 | 7 | 5 | 61 | 25 | 0 | 0 |
| 3 | 20 | 7 | 11 | 7 | 9 | 5 | 11 | 3 | 23 | 8 | 0 | 0 |
| 4 | 4 | 1 | 1 | 1 | 4 | 3 | 2 | 2 | 2 | 2 | 1 | 0 |
| 5 | 2 | 0 | 0 | 0 | 3 | 2 | 4 | 2 | 0 | 2 | 1 | 0 |
| 6 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Table 3.7: The number of samples showing mixtures of two facial expressions.

| | **Fear** | **Surprise** | **Sadness** | **Anger** | **Disgust** | **Happy** |
|---|---|---|---|---|---|---|
| **Fear** | 0 | **76** | **4** | **4** | **41** | **1** |
| **Surprise** | **76** | 0 | 0 | 0 | 0 | 0 |
| **Sadness** | **4** | 0 | 0 | **4** | **32** | **1** |
| **Anger** | **4** | 0 | **4** | 0 | **17** | 0 |
| **Disgust** | **41** | 0 | **32** | **17** | 0 | 0 |
| **Happy** | **1** | 0 | **1** | 0 | 0 | 0 |

Table 3.8: The structure of the data set of non-ambiguous facial expression samples.

| Fear | Surprise | Sadness | Anger | Disgust | Happy | **Total** |
|---|---|---|---|---|---|---|
| 38 | 7 | 47 | 53 | 7 | 99 | **251** |

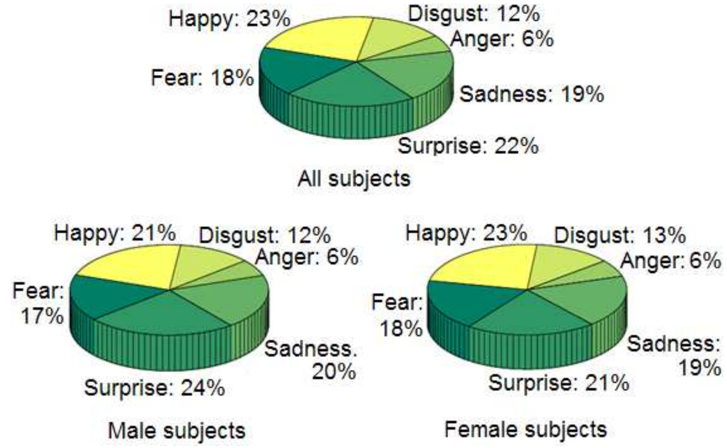


Figure 3.15: Gender-based analysis of the sample proportion for each emotion class.

prototypic facial expressions.

The mixed facial expression samples can still be used for the recognition models. An information aid is to find out the intensity of each facial expression for each face sample. Since this information is not available, in the next chapter we will describe a procedure for the automatic identification of ambiguous facial expressions in the face displays.

## 3.8 Conclusion

Applying statistical analysis on the face appearance provides a powerful mechanism for efficiently modelling the contextual data variance exhibited by illumination conditions, scaling and rotation, as well as the variance imposed by gender, age and different face postures.

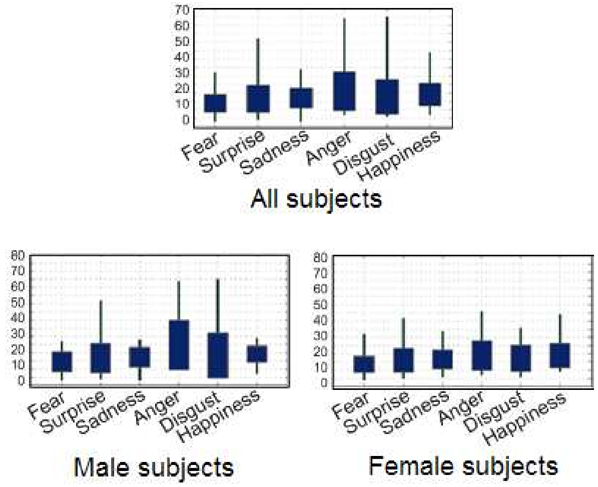


Figure 3.16: Gender-based analysis of the number of frames for the each emotion class. The vertical middle of each bar represents the mean. The bars extend upwards and downwards with a length equivalent to 1 standard deviation. The lines extend to the minimum and maximum values of the number of frames per video sample.

Furthermore, because of the fact the appearance parameters account for controlling the variation of both the shape and grey levels of the AAM model, it is presumable that they offer a solid base for being used as features for the classification of facial expressions.
Alternatively, AAM face models may be used together with more complex image processing techniques for features acquisition in the process of recognizing facial expressions.
The study of Action Unit representations for facial expression annotation provided a new method for selecting representative face samples. Following the mainstream research, in the next chapters we use Cohn-Kanade database for the face analysis experiments. Nonetheless, as result of applying a systematic study of the AU codes, the data subset we finally obtain contains more accurate face samples.

# Chapter 4

# Facial expression in static pictures[1]

## 4.1 Introduction

It is the human nature that we can estimate a person's psychological state following the observation on his face. Non-verbal communication channels are typically set during common interpersonal relations and visual messages are processed in a transparent manner.
Currently, the general tendency is to construct robotic systems that are able to understand the environmental world and to interact with the existent actors. Human-computer interfaces play an essential role in the perception and feedback of the system. In this context, the advantage of making machines to read human facial expressions is tremendous.
Facial expressions genuinely reveal emotion characteristics of the expresser. To address the problem of facial expression recognition from single images, in our approach we extract parametric information with high discrimination power from facial feature space and use it in a data-driven classification environment. The current chapter primarily focuses on the aspects related to classification methods for the recognition of universal expressions triggered by six basic emotion categories from single images. The methods include boosting and kernel oriented techniques for binary and multi-class classification.
To our knowledge, this is the first research that involves relevance vector machines for facial expression recognition. An important contribution relates to the preparation of the facial expression data set. The problem we try to solve rises from the unavailability of facial expression labels in the original database. The investigations presented here imply on a top-down approach for selecting samples based on their relevance to the classification setup.
Concretely, we begin with using the whole set of face instances and gradually exclude ambiguous samples as indicated by the recognition results and by the action unit-oriented analysis of their labels. The recognition process is mainly used to identify outliers in the initial data set. However, not all of these instances represent ambiguous facial expression samples. The selection process discards only the instances which relate to mixtures of facial expressions, where

[1] This chapter is an extended version of the paper Datcu et al.[41].

no facial expression category is dominant.
Another aspect of the research is the selection of visual features prior to classification. We show how to identify such relevant features with the use of boosting methods.

## 4.2 Related work

Geometric features have been used for facial expression recognition in [95] and [216]. The original local binary pattern - LBP operator has been introduced in the paper [139].
The works [71] and [70] apply the original LBP extraction from a 3X3 neighbourhood on 10X8 non-overlapping blocks of the face-normalized images. The classification of six facial expressions plus the neutral state is realized through a binary tree scheme and 21 binary classifiers based on a linear programming method. The accuracy of classification is then 93.8% on JAFFE dataset.
On a similar experimental setup, the papers [69] and [67] develop a coarse-to-fine classification scheme which consists of the combination of multi-template instance pairs, the weighted Chi-square and K-nearest neighbour classifier. The face normalization makes use of the location of the eyes. By applying the classification method for the recognition of six basic facial expressions and the neutral state on face images, the authors achieve an average recognition rate of 77% on JAFFE database. Further more, Feng et al. [68] continue the research with person dependent and person independent experiments.
He et al. [90] approach the recognition of facial expressions by extracting LBP features from face images which are firstly applied a wavelet decomposition into four frequency details. They extract specific LBP features which are called uniform patterns and which have assigned weights adaptively. The experiments are conducted in person dependent and person independent setups.
The work [160] reports the results on using simple uniform pattern LBP features for classifying seven facial expressions in the Cohn-Kanade database. The authors use classifiers based on template matching with weighted Chi square statistics and SVM with linear, polynomial and Radial Basis Function - RBF kernels. A second set of experiments demonstrate the reliability of LBP features for low-resolution analysis.
The work of Liao et al. [111] extends the previous works on local binary pattern operator by deriving ALBP as an rotation-invariant LBP. Beside extracting ALBP features in both intensity and gradient maps, the authors compute Tsallis entropy of Gabor filtered responses and null-space intensity (NLDAI) and energy variation patterns. These features are assumed to effectively capture the local intensity and energy variation patterns, the global low- and mid-frequency domain distributions and the discriminating appearance characteristics of the face images.
The features are derived from eight specific face regions which have assigned different weights according to the importance in generating clues for facial expressions. In an comparative experimental setup, rotation-invariant LBP has better performance (88.26%) than traditional LBP (85.57%) on the JAFFE database. The authors determine that the best result (94.59%) is obtained using ALBP features together with the two other types of features. The same

paper presents the outcome of a research on the relationship between the facial expression classification accuracy and the resolution of the face images.

The work of Shan et al. [163] contains an approach for learning discriminative LBP-Histogram (LBPH) bins for facial expression recognition. By using SVM classifier with multi-scale uniform pattern features, the authors report the recognition performance of 93.1% on the Cohn-Kanade database.

Gritti et al [81] present an ample comparison study on the performance of different facial descriptors for facial expression recognition. They derive and separately use features based on Local Binary Patterns (LBP), on Gabor wavelets and on the recently proposed local ternary patters - LTP. The advantage of using LTPs is that the generated codes are less sensitive to noise in near-uniform regions. In addition, they introduce the histograms of oriented gradients - HOG to the area of facial expression recognition. By checking the influence of varying the parameters during the generation of each type of feature, they determine that the best performance of a linear SVM classifier on the Cohn-Kanade database, is achieved with LBP features with overlapping. In the same research, the authors study the impact of face registration errors on the recognition using different facial representations. For this, the face samples in either the training data set or both training and testing data sets, are altered by displacing the face shapes from the true position of the eyes using Gaussian noise. This type of experiment indicated the superiority of using LBP features for the analysis.

## 4.3 Visual feature model

### 4.3.1 Viola&Jones features

Viola&Jones features have been initially proposed as a fast method for performing object detection [187] and later for face detection [188]. The facial expression recognition uses three types of simple features derived from pixel intensities of video frame images.

The first feature type is based on computing the difference between the sums of pixels within two horizontally or vertically adjacent rectangular regions (the first filters from the left in figure 4.1). The second type contains three rectangles and is determined by the difference between the sum of pixels in outside regions and the sum of pixels in the middle rectangle (the third and the fourth filters from the left in figure 4.1). The third feature type is computed by the difference of sums of pixels in diagonal pairs of regions (the first filter from the right side of figure 4.1). Each of the features is effectively computed by making use of the integral image as an intermediate representation of the original video frame image.

The integral image at location $(x, y)$ is computed as sum of pixels contained in the rectangle defined from the top-left most pixel location to the current location $(x, y)$. If $ii(x, y)$ is the integral image and $i(x, y)$ is the original image, then:

$$ii(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y').$$

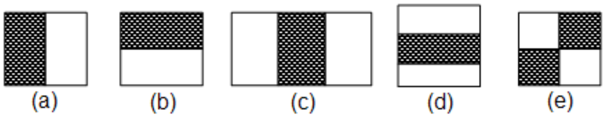


Figure 4.1: Basic types of 2D Viola&Jones features.

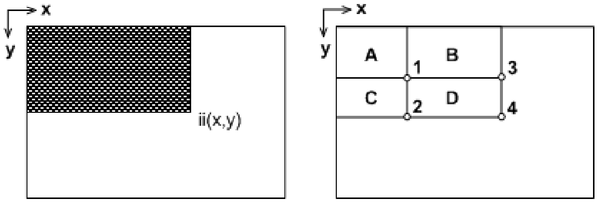


Figure 4.2: Integral Image representation.

The integral image can be computed in one pass over the original video frame image by using the recurrences:

$$\begin{aligned} s(x,y) &= s(x,y-1)+i(x,y) \\ ii(x,y) &= ii(x-1,y)+s(x,y), \end{aligned}$$

where $s(x,y)$ is the cumulative row sum, $s(x,-1)=0$ and $s(-1,y)=0$.
The sum of pixels in one rectangle on the image can be computed using four references to the integral image. As an example, the sum of region $A$ in figure 4.2 can be computed as following: $p_4+p_1-(p_2+p_3)$, where $p_k$ denotes the integral image computed at location of point $k$. By using the adjacency property, it is possible to use only six references to the integral image instead of eight to determine the value of each two-rectangle Viola&Jones feature, eight references instead of twelve for a three-rectangle feature and nine references instead of sixteen for the four-rectangle type of features.
Apart from the previously described features, other types of Viola&Jones filters have been recently proposed. An example is presented in the work [115] through an extended set of filters which present 45° rotation.

### 4.3.2 Local Binary Patterns

Local binary patterns represent micro-pattern structural features which can successfully facilitate a reliable representation of the face images. The original LBP operator was introduced by Ojala et al. [139] in the broad context of texture analysis. According to this, a binary pattern at a texture image pixel is generated by applying a threshold on the intensities or grey values of the surrounding

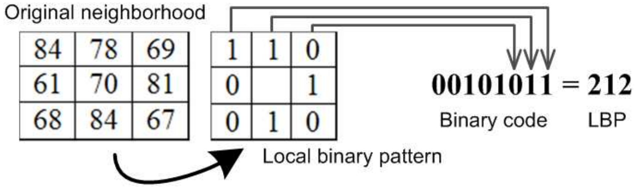


Figure 4.3: Basic LBP operator applied on a neighbourhood of 3X3 elements.

pixels in a 3 × 3 neighbourhood with the intensity of the pixel itself (see figure 4.3). Later, the work of [140] proposed the use of scaled LBP features which allow for sampling from neighbourhoods of different sizes. These features are computed from block neighbourhoods consisting of different number of circularly located pixels at any radius. Together with the location of the centre of the rectangular region, these two parameters completely identify an LBP. The same paper introduced the concept of uniform patterns as binary patterns which present at most two bitwise transitions from 0 to 1 or the other way round. For instance, 000000, 111111, 001110 and 100001 are uniform binary patterns.

The advantages of the uniform patterns are that they are very representative for the texture analysis and that they account for a rather small proportion in the total set of features. For a neighbourhood of (8,1), Ojala et al. [140] found that slightly less than 90 percent of the patterns are uniform patterns. Similarly, for a (16,2) neighbourhood, the uniform patterns account for around 70 percent of all the samples. That means that the data set for the analysis can be restricted to using only uniform patterns without considerable loss of performance.

Ahonen et al. [6][7] proposed a novel face representation for image-based face analysis in the context of using LBP features. An image area including the face sample is divided into small non-overlapping regions and LBP features are computed from each region. LBP codes are computed from each image region and concatenated into an enhanced feature vector. A common approach for the use of LBP codes is to compute data histograms and subsequently to use these for classification.

The LBP feature models described below follow the block-based technique for combining pixel-level and region-level data from face samples.

For an image plane, the basic 2D LBP is determined from $P + 1$ image pixels $\{g_c, g_0, .., g_{P-1}\}$, where $g_c$ is the grey value of the centre pixel and $\{g_0, .., g_{P-1}\}$ represent the gray values of $P$ equally spaced pixels on a circle of radius $R$.

The $P$ pixels are sampled at locations $(x_c + Rcos(2\pi p/P), y_c - Rsin(2\pi p/P))$. Figure 4.4 illustrates four different circular and symmetric local binary patterns with different number of neighbourhood points.

The grey level of pixel samples $(x, y)$ laying in-between image pixels is determined using bilinear interpolation in the following way: assuming that $x_0$ and $y_0$ are the integer parts of $x$ and $y$, and $\alpha_x$ and $\alpha_y$ are the reminders so that $x = x_0 + \alpha_x$ and $y = y_0 + \alpha_y$, then the interpolated grey level is:

$$\begin{aligned} g(x, y) &= (1 - \alpha_x)(1 - \alpha_y)g(x_0, y_0) + \alpha_x(1 - \alpha_y)g(x_0 + 1, y_0) + \\ &\quad (1 - \alpha_x)\alpha_y g(x_0, y_0 + 1) + \alpha_x\alpha_y g(x_0 + 1, y_0 + 1). \end{aligned}$$

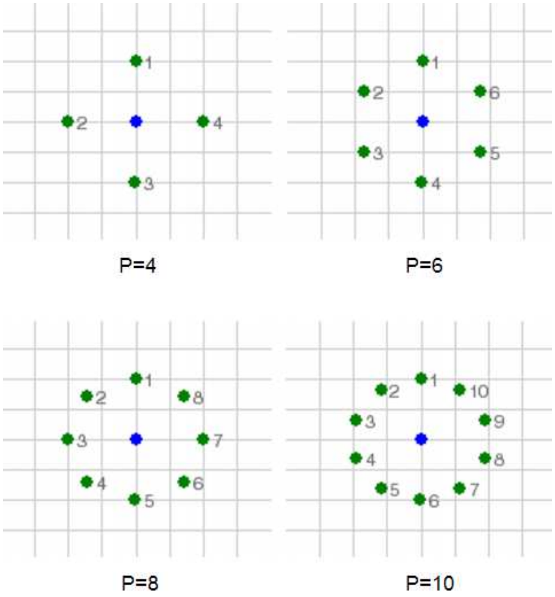


Figure 4.4: Circularly shaped local binary patterns with different number of points {4,6,8,10}.

The grey-scale invariance is achieved by subtracting the grey level of the centre pixel from the grey-levels of the neighbour pixels. Furthermore, the scale invariance is achieved by keeping the sign of each difference instead of the value previously computed. The formula for computing the LBP code is:

$$LBP_{P,R} = \sum_{q=0}^{P} s(g_q - g_c)2^q,$$

where the term $s(x)$ is:

$$s(x) = \begin{cases} 1 & if \quad x \geq 0 \\ 0 & if \quad x < 0. \end{cases}$$

Figure 4.5 shows an example of LBP patterns projected on a face image which was processed as described in chapter 3. Table 4.1 presents the LBP codes derived from the LBP blocks from this example.

The LBP codes computed from 4, 6 or 8 neighbouring points require only 1 byte to store while the LBP codes computed from 10 points or more require 2 bytes to store.

### 4.3.3 Geometric features

Following the use of active appearance models on the Cohn-Kanade database, as presented in section 3.7 of chapter 3, we build descriptive face models based

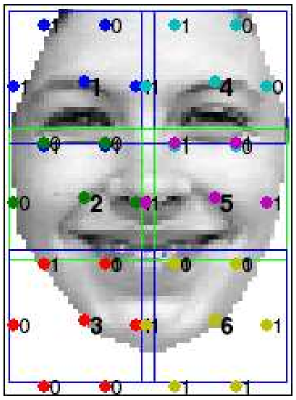


Figure 4.5: Example of LBP feature extraction from a 60×80 face image. The image segmentation has 2 horizontal × 3 vertical splits = 6 overlapping blocks (10% overlap). A distinct LBP code is generated from each block by using 6 points from a circular neighbourhood.

Table 4.1: The formation of LBP codes from an 60×80 image based on a 2×3 blocks configuration.

| | gc | $g_1$ | $g_2$ | $g_3$ | $g_4$ | $g_5$ | $g_6$ | LBP |
|---|---|---|---|---|---|---|---|---|
| 1 | 181 | 183.0 | 103.1 | 255.0 | 183.0 | 235.0 | 209.9 | 101111 |
| 2 | 171 | 191.0 | 116.0 | 150.6 | 168.0 | 198.0 | 149.7 | 100010 |
| 3 | 255 | 255.0 | 255.0 | 255.0 | 191.0 | 151.0 | 165.9 | 111000 |
| 4 | 224 | 130.0 | 222.4 | 237.0 | 230.0 | 255.0 | 186.5 | 001110 |
| 5 | 141 | 210.0 | 151.6 | 221.0 | 164.0 | 205.6 | 183.4 | 111111 |
| 6 | 183 | 255.0 | 161.5 | 153.5 | 210.0 | 255.0 | 255.0 | 100111 |

on geometric features. First, we select a set of face landmark points on the face area from the set of key points handled by AAM. The landmark points are also called facial characteristic points - FCPs and correspond to an extension of the FCP-set of Kobayashi & Hara model [106]. Figure 4.6 shows the FCPs used in the research.

The feature parameters are then computed as values of specific angles and/or Euclidean distances between FCPs. These features are described in table 4.2. Because they relate to the position of various facial features, the geometric features are assumed to reflect the onset of facial expression categories. The advantage of the model is that it can handle a certain degree of asymmetry by using parameters for both left and right sides of the face. The correlation with the settings of the camera is removed by normalizing the feature vectors with the distance between the inner corners of the eyes.

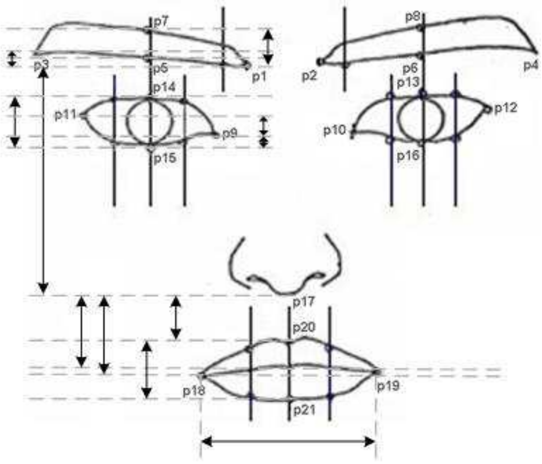


Figure 4.6: Facial Characteristic Point model.

Table 4.2: Facial Characteristic Points - based set of geometric features.

| $v_i$ | meaning | feature | $v_i$ | meaning | feature | $v_i$ | meaning | feature |
|---|---|---|---|---|---|---|---|---|
| $v_1$ | $(P_1, P_7)_y$ | l.eyebrow | $v_7$ | $(P_{14}, P_{15})_y$ | l.eye | $v_{13}$ | $(P_{17}, P_{20})_y$ | mouth |
| $v_2$ | $(P_1, P_3)_y$ | l.eyebrow | $v_8$ | $(P_9, P_{11})_y$ | l.eye | $v_{14}$ | $(P_{20}, P_{21})_y$ | mouth |
| $v_3$ | $(P_2, P_8)_y$ | r.eyebrow | $v_9$ | $(P_9, P_{15})_y$ | l.eye | $v_{15}$ | $(P_{18}, P_{19})_y$ | mouth |
| $v_4$ | $(P_2, P_4)_y$ | r.eyebrow | $v_{10}$ | $(P_{13}, P_{16})_y$ | r.eye | $v_{16}$ | $(P_{17}, P_{18})_y$ | mouth |
| $v_5$ | $(P_1, P_{17})_y$ | l.eyebrow | $v_{11}$ | $(P_{10}, P_{12})_y$ | r.eye | $v_{17}$ | $(P_{17}, P_{19})_y$ | mouth |
| $v_6$ | $(P_2, P_{17})_y$ | r.eyebrow | $v_{12}$ | $(P_{10}, P_{16})_y$ | r.eye | | | |

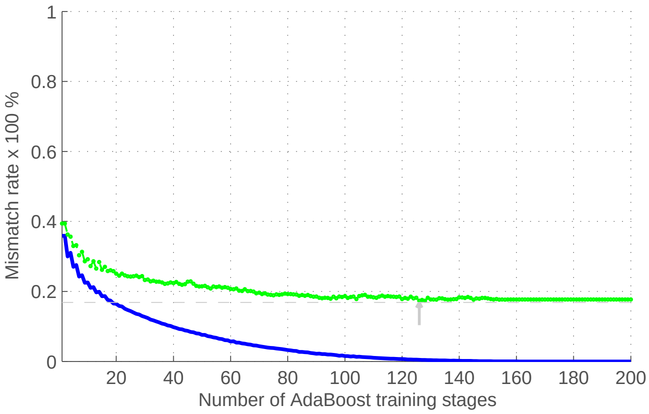


Figure 4.7: Train and test mismatch rate of facial expression detection using AdaBoost classifier with LBP features.

## 4.4 Detection of facial expressions

The detection of facial expressions is done in a binary classification setup. The classification model relies on AdaBoost method that is learned in 200 training stages. The two classes relate to instances of face images of the neutral emotion state and face images showing various basic facial expressions. The data set includes local binary pattern features extracted from 1.440 different face instances. A set of 179.520 LBP features is computed from each image instance. The neutral instances relate to the first frame of each video sequence from the Cohn-Kanade data set. The emotion instances have been selected from the last frame and the frame corresponding to the index 3/4 the number of video frames, for each video sequence of the database. In this way, the classification data set contains double the number of facial expression samples than the number of neutral instances. The evaluation has been done with n-folders cross validation methods, with n equals 50 folders.
The best AdaBoost facial expression detector has 126 base functions and shows an accuracy rate of 82.63% (figure 4.7). The training mismatch rate of this classifier is 0.45%.

## 4.5 Recognition of static facial expressions

The models we develop for the recognition of facial expressions in single pictures use the previously presented types of features namely Viola&Jones descriptors, local binary patterns and geometric parameters. The models are trained and tested on face data, each picture being rescaled so as to have the height of 80 pixels and the width of 60 pixels. We use both binary and multi-class classification algorithms. As binary classifiers, we investigate discrete Adaboost, support vector machines and relevance vector machines. For multi-class classification, we use Adaboost.M2 method.
Adaboost and Adaboost.M2 run several training steps, at each training step producing a new so-called weak classifier. All weak classifiers are binary trees

with two branches, also called decision stumps. A strong classifier is made at each step by linearly combining the current weak classifier with all the weak classifiers that are generated until that step.

The purpose of using multiple procedures relates to the intention of rigorously studying the characteristics of each approach. In addition, the attempt allows for further comparison with results reported in other research works that focused on facial expression recognition. The top-down approach starts with the full set of face instances from the Cohn-Kanade database and iteratively proceeds with the identification of outliers, removing samples of mixed facial expressions and training models on new data sets.

### 4.5.1 Local binary patterns

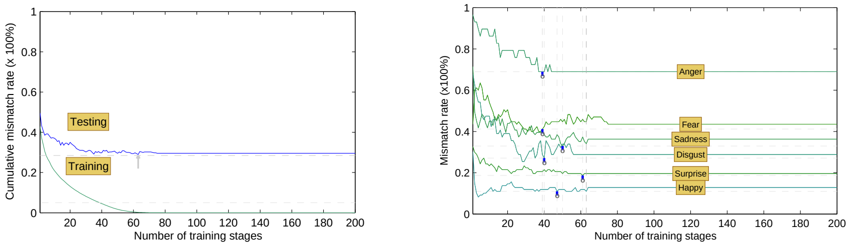
In the first experiment we extract and use sets of LBP features obtained using circular neighbourhoods of 4, 6, 8 and 20 points from local regions which show overlap from 0% to 80% with increment of 10%. Each overlapping block is assigned one LBP neighbourhood only. For each parameter which specifies the number of neighbouring points, we have created a separate data set.

The goal of the experiment is to show the descriptive power of LBP features given different sizes of circular neighbourhoods. Each data set contains the concatenation of all features extracted from adjacent regions by splitting the face sample image in 1, 2, 3, ..., 20 blocks along X and Y axes.

The minimum classification rate for six facial expressions is 66.46% and is obtained by LBPs from 4 points neighbourhoods. The maximum classification rate is 68.75% and is obtained by LBPs derived from 16 points neighbourhoods.

The second experiment is aimed to determine the classification rate using the concatenation of LBP features obtained in a way similar to the previous experiment. The splitting of the face sample image is done on 1, 2, 3, ..., 10 blocks along X and Y axes with overlap of 0% to 50% with the increment of 10%. From each block we extract features of 4, 6, 8 and 10 neighbouring points. In total, the LBP feature extraction produces a set of 72.600 codes for each face sample.

Figure 4.8 illustrates the training and testing errors achieved by multi-class Adaboost.M2 classifier at each step, for maximum 200 training steps. The graphic (b) in the right side of the figure, shows the test errors for each class of facial expressions. Each curve in the graphic contains a mark that indicates the number of training steps of one classifier that performs better on the recognition of that specific facial expression. The classifier which has the smallest misclassification rate for all facial expressions is called the optimal classifier and is obtained after 63 training steps. Because at each step one weak classifier is trained by using one feature only, the optimal classifier uses 63 features for each facial expression category. Figure 4.9 illustrates the receiver operating curves of 200 classifiers obtained at each step of the boosting classifier. The location of the points in the image is determined by the true positive rate and the false positive rate of the classifier associated. The north-west corner of the image indicates a classifier that has optimal facial expression recognition rate. Therefore, the best classification is generally achieved by the model which is represented by a point located closer to north-west corner. For each emotion category, the ROC graph shows a curve defined by all the points which are closer to the optimal classifier. Table 4.3 shows the confusion matrix of Adaboost.M2 classifier evaluated with leave-one-out cross validation. The overall test recognition rate of



(a) Cumulative error, optimal classifier at 63 training stages.

(b) The dependency of mismatch rate of each facial expression recognition given the number of training stages. LBP features.

Figure 4.8: Recognition results of Adaboost.M2 model trained with LBP features for 200 training steps.

Table 4.3: The confusion matrix of Adaboost.M2 classifier trained on LBP features and leave-one-out cross validation.

| | Fear | Surprise | Sadness | Anger | Disgust | Happy |
|---|---|---|---|---|---|---|
| Fear | **80.00** | 0.00 | 0.00 | 0.00 | 0.00 | 20.00 |
| Surprise | 0.00 | **100.00** | 0.00 | 0.00 | 0.00 | 0.00 |
| Sadness | 10.00 | 10.00 | **60.00** | 10.00 | 10.00 | 0.00 |
| Anger | 0.00 | 0.00 | 20.00 | **40.00** | 20.00 | 20.00 |
| Disgust | 0.00 | 0.00 | 8.33 | 8.33 | **75.00** | 8.33 |
| Happy | 11.11 | 0.00 | 0.00 | 0.00 | 0.00 | **88.88** |

the classifier is 71.04% and the training mismatch rate is 0.22%. Figure 4.10 shows the projection of the relevant LBP features on illustrative face samples from the Cohn-Kanade database. The graphics show only the features that form the weak classifiers of the optimal strong classifier. Each weak classifier uses only one LBP feature from the data set. The facial expression samples contain the projection of approximately 30.240 features. These LBP patterns originate from 63 features of the optimal classifier at each folder, for all 480 folders. For the data set of 480 samples, 82.81% of the binary LBP codes represent uniform patterns. Starting from the assumption that uniform patterns carry enough descriptive information for facial expression recognition, we have made a new analysis on the new data set. This data set contains all the previous LBP codes that account for uniform patterns. In the case of non-uniform patterns, we have replaced all the LBP values with an unique code.

Retraining the Adaboost.M2 classifier with uniform features and 20-folds cross validation leads to a test recognition rate of 70.00% for a classifier which achieves 0.15% error on the training data set. The optimal classification is obtained after 68 training stages. Table 4.4 shows the confusion matrix of this classifier. Based on the performance of Adaboost.M2 classifier with the set of 480 samples, it is possible to identify data instances that have a high degree of misclassification. These data points represent outliers for the classification context. By removing some of them, it is expected that the overall performance improves. Mostly we are interested to identify and remove outliers that correspond to face samples of mixed facial expressions. The misclassified samples of non-ambiguous facial expressions are not removed from the data set. To test whether a face instance

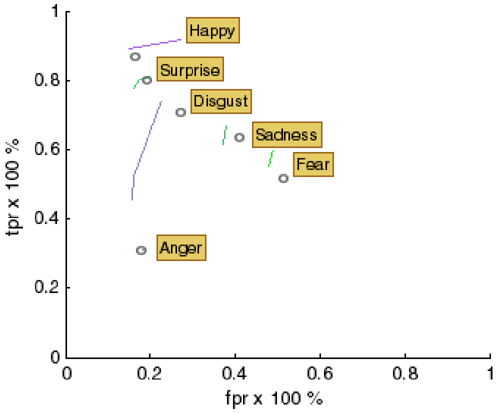


Figure 4.9: ROC graphic for Adaboost.M2 using LBP features. The ’o’ markers correspond to the performance of the classifier trained with 63 stages.

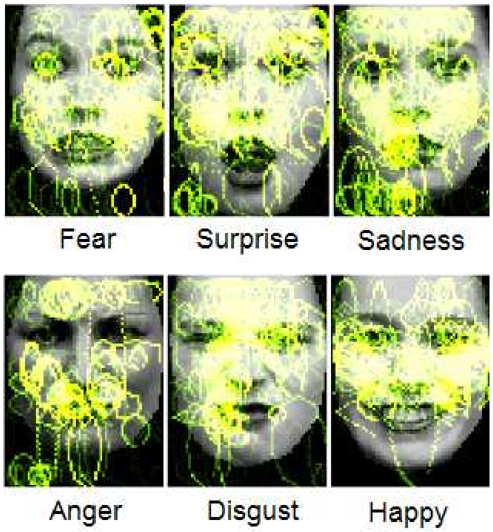


Figure 4.10: Graphical representation of LBP features selected by optimal Adaboost.M2 classifier on face samples representing basic facial expression. There are about 30.240 relevant features projected on each facial expression sample image, consisting of the reunion of the 63 LBP features on each of the 480 cross validation folders.

Table 4.4: The confusion matrix of Adaboost.M2 classifier trained on uniform LBP patterns and 20 folds cross validation.

| | Fear | Surprise | Sadness | Anger | Disgust | Happy |
|---|---|---|---|---|---|---|
| Fear | **49.41** | 17.64 | 21.17 | 1.17 | 1.17 | 9.41 |
| Surprise | 10.28 | **82.24** | 6.54 | 0.00 | 0.00 | 0.93 |
| Sadness | 14.28 | 2.19 | **73.62** | 0.00 | 2.19 | 7.69 |
| Anger | 20.68 | 10.34 | 34.48 | **27.58** | 3.44 | 3.44 |
| Disgust | 16.94 | 1.69 | 16.94 | 1.69 | **59.32** | 3.38 |
| Happy | 8.25 | 0.00 | 0.91 | 0.00 | 2.75 | **88.07** |

Table 4.5: The results of data pruning on binary and multi-class classifiers which use LBP features, 20 folds cross validation.

| | | **330** | | | **348** | | | **390** | |
|---|---|---|---|---|---|---|---|---|---|
| | ac | tpr | fpr | ac | tpr | fpr | ac | tpr | fpr |
| AM2 | 84.24 | 51.21 | 3.81 | 84.34 | 41.30 | 2.80 | 83.64 | 43.90 | 3.81 |
| | 84.24 | 96.29 | 3.21 | 84.34 | 97.64 | 4.05 | 83.64 | 97.53 | 3.61 |
| | 84.24 | 85.48 | 7.46 | 84.34 | 88.13 | 7.69 | 83.64 | 85.48 | 7.84 |
| | 84.24 | 00.00 | 0.31 | 84.34 | 00.00 | 0.00 | 83.64 | 00.00 | 0.00 |
| | 84.24 | 76.19 | 0.35 | 84.34 | 82.22 | 0.70 | 83.64 | 76.19 | 0.69 |
| | 84.24 | 96.90 | 4.72 | 84.34 | 98.88 | 4.55 | 83.64 | 96.90 | 4.72 |
| avg | **84.24** | | | **84.34** | | | **83.64** | | |
| SVM | 83.60 | 20.00 | 7.63 | 82.70 | 18.33 | 6.99 | 81.78 | 30.39 | 7.39 |
| | 93.90 | 82.56 | 2.02 | 93.38 | 80.87 | 2.22 | 94.36 | 84.84 | 2.06 |
| | 90.00 | 67.95 | 4.42 | 85.36 | 46.50 | 7.65 | 91.57 | 62.07 | 3.59 |
| | 98.22 | 2.50 | 0.00 | 98.56 | 12.50 | 0.00 | 98.21 | 10.00 | 0.00 |
| | 96.38 | 61.25 | 0.62 | 95.39 | 58.33 | 1.32 | 94.86 | 65.50 | 1.41 |
| | 92.67 | 88.36 | 4.99 | 91.99 | 84.44 | 5.11 | 92.33 | 86.06 | 4.18 |
| avg | **92.46** | | | **91.23** | | | **92.18** | | |
| Ada | 89.70 | 58.54 | 5.88 | 87.64 | 48.94 | 6.31 | 85.64 | 56.25 | 8.59 |
| | 95.45 | 91.36 | 3.21 | 94.25 | 86.36 | 3.08 | 92.56 | 81.05 | 3.73 |
| | 88.79 | 74.19 | 7.84 | 89.66 | 70.77 | 6.01 | 87.18 | 65.28 | 7.86 |
| | 97.88 | 28.57 | 0.62 | 97.99 | 62.50 | 1.18 | 98.21 | 66.67 | 1.05 |
| | 95.15 | 85.71 | 3.47 | 94.25 | 79.07 | 3.61 | 93.33 | 79.17 | 4.68 |
| | 93.64 | 87.63 | 3.86 | 93.68 | 88.66 | 4.38 | 93.85 | 84.31 | 2.78 |
| avg | **93.43** | | | **92.91** | | | **91.79** | | |

is non-ambiguous or contains mixed facial expressions, we use the methods described in section 3.7 of the previous chapter.

In order to inspect the influence of data pruning on the LBP-based facial expression recognition, we have adopted a data selection procedure. First, we identified 90 test samples which have been previously misclassified at all steps and for all folders of the cross validation procedure. Removing these samples lead to the first database which contains 390 samples. Similarly, pruning the samples which are misclassified in 90% of the test cases resulted to the second data set of 348 samples. The third data set of 330 samples is obtained by eliminating the test samples that are misclassified in 80% of the cross validation cases. According to the results from table 4.5 of the experiments using Adaboost.M2 classifier and the new data sets, the classification rate increases with 13.20% in case of the data set with 330 samples, with 13.30% for the data set of 348 samples and with 12.60% for the data set of 390 samples.

Beside multi-class classification models, we have also investigated the use of binary classifiers for the recognition of six facial expressions. The previous problem of training and testing with one multi-class classifier shifts to the context of training and testing six binary classifiers. One against the rest procedure is used to make the conversion between the six classes setup and the two classes

setup.
Each of these classifiers follows the sample evaluation procedure required by the 20-fold cross validation. The binary classifiers taken into account are the classic Adaboost and the support vector machines. The feature vectors used for the experiments with Adaboost are the same as in the case of the data set used for Adaboost.M2 classifier. Like Adaboost.M2, Adaboost produces strong committees that consist of sets of weak classifiers based on decision stumps. The approach is appropriate because both Adaboost.M2 and classic Adaboost carry out the classification together with the feature selection.
In the case of SVM, we use a specific feature selection method. Basically, we aim at using smaller vectors of relevant features instead of the initial vectors of 72.600 LBP features. A reasonable solution for the feature reduction is possible by evaluating the efficiency of LBP features in the Adaboost.M2 classification setup. More exactly, we check the features that have been automatically selected by Adaboost.M2 in the previous classification. For that, we choose the features which were used by weak classifiers in all the folders of the 20 fold cross validation. The size of the LBP feature vectors is then 1.047 for the data set of 390 samples, 687 for the data set of 348 samples and 626 for the data set of 330 samples. The databases for SVM-based analysis were built by removing the outliers and by selectively choosing the most relevant LBP features as indicated by the Adaboost.M2.
For SVM classifier we have used various kernel functions. We have found that the optimal kernel for classification is the polynomial function. In table 4.5, the results of SVM are based on the use of a polynomial of degree 4. Increasing the polynomial degree leads to slightly better results. However, given the structure and the rather small size of the database, polynomials of high degree are prone to overfit. The results show that SVM and Adaboost have comparable performance, with small advancement of Adaboost classifier. This can be explained from the point of view of the explanatory power of the features. SVM uses features that are proved to perform well in the context of the Adaboost classifier. Consequently, the same features may not show the same discriminatory characteristics in any other classification setups. The results shown represent a good basis for comparison with the results reported by previous works [162][49][218][111][22] and by future research.

### 4.5.2 Viola&Jones features

For the analysis, we have used Viola&Jones features of types (a),(b),(c) and (e). Based on an image size of 60X80 pixels we have originally generated a set of 617.525 basic features. The types (a),(b) and (c) each accounts with 27.65% to the total amount of features. We have prepared the final data set by uniformly sampling a third of the initial features. The procedure resulted to a set of 480 face samples, each being represented by a vector of 205.842 elements.
The graphic in figure 4.11 shows the cumulative training and testing errors obtained during training Adaboost.M2 classifier for 100 steps, by using 50-fold cross validation procedure. Figure 4.12 indicates the test mismatch rates achieved at each training step.

The optimal classification is obtained with a model trained in 45 stages. The overall test accuracy of this model is 72.29%, while the train mismatch rate is 0.40%. Table 4.6 shows the confusion matrix of this model. Figure 4.13

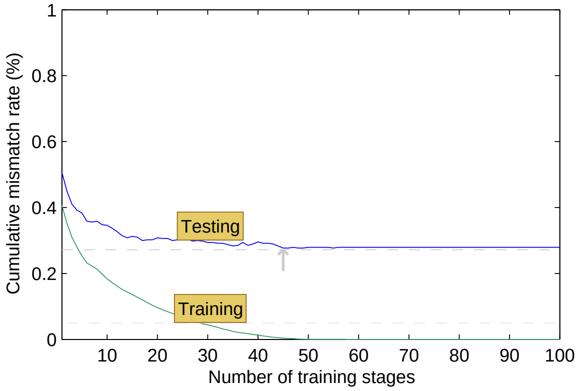


Figure 4.11: Cumulative error of optimal classifier Adaboost.M2 with Viola&Jones features has 45 training stages.

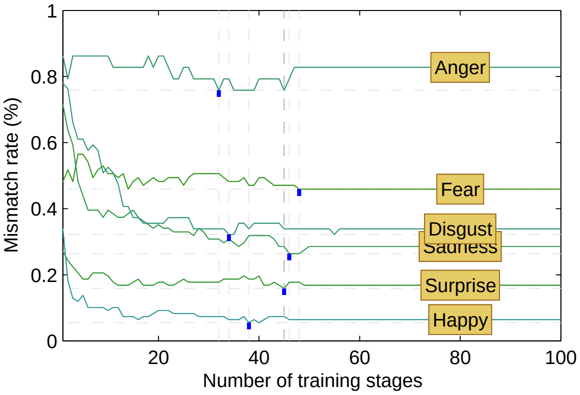


Figure 4.12: Dependency of Adaboost.M2 test mismatch rate of each facial expression recognition given the number of training stages.

Table 4.6: The confusion matrix of Adaboost.M2 trained in 64 stages.

| | Fear | Surprise | Sadness | Anger | Disgust | Happy |
|---|---|---|---|---|---|---|
| Fear | **52.94** | 16.47 | 15.29 | 1.17 | 5.88 | 8.23 |
| Surprise | 12.14 | **84.11** | 1.86 | 0.93 | 0.93 | 0 |
| Sadness | 10.98 | 4.39 | **71.42** | 4.39 | 3.29 | 5.49 |
| Anger | 27.58 | 3.44 | 24.13 | **24.13** | 13.79 | 6.89 |
| Disgust | 10.16 | 0 | 11.86 | 5.08 | **66.10** | 6.77 |
| Happy | 4.58 | 0.91 | 0.91 | 0 | 0.91 | **92.66** |

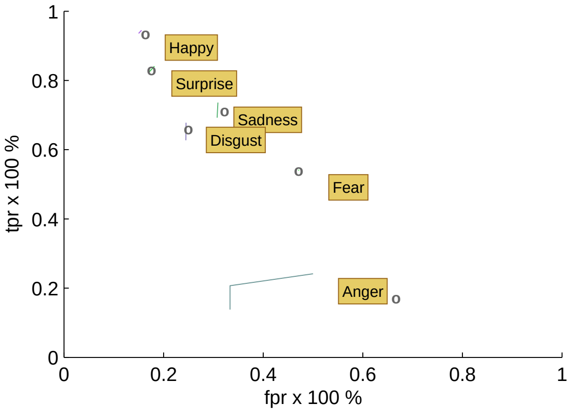


Figure 4.13: ROC for Adaboost.M2 using Viola&Jones features. The 'o' markers correspond to the performance of the classifier trained with 45 stages.

illustrates the receiver-operating curve which includes all classifiers generated during the Adaboost.M2 training. The ROC graph was generated based on 100 different classifiers obtained by training Adaboost.M2 with different number of hypotheses. Figure 4.14 shows the projection of Viola&Jones features on face samples, for every facial expression category. There are about 2.250 relevant features projected on each facial expression sample image, consisting of the reunion of the 45 LBP features on each of the 50 cross validation folders. Figure 4.15 illustrates the contribution of each type of relevant VJ features to the best performing classifier. It can be noticed that features which comprise of tree adjacent rectangular regions are the least significant for the classification of any of the six facial expressions.

Similarly, to the recognition of facial expressions using LBP patterns, next we conduct an analysis for identifying and removing outliers of mixed facial expressions and for building binary classification models on the resulting data sets. Adaboost and support vector machines are considered. The results of the experiment are presented in table 4.7.

### 4.5.3 Geometric features

The data used for training models on geometric feature, has been selected from the Cohn-Kanade database. Here, we follow the same top-down approach by starting with the full set of 485 face images and by removing ambiguous facial expression samples. The models include binary classification performed by support vector machines and relevance vector machines.
The classification results are analysed by checking the mismatch rate in a 5-fold cross validation setup. As it can be seen from table 4.8, the error rates in the case of RVM (9.16%) are comparable to those of SVM classifier (10.15%). One important aspect is that in case of RVM classifier, the number of relevance vectors (156) is smaller than the number of support vectors (276) of SVM. The effect is a decrease of the number of kernel functions and so of the model complexity.

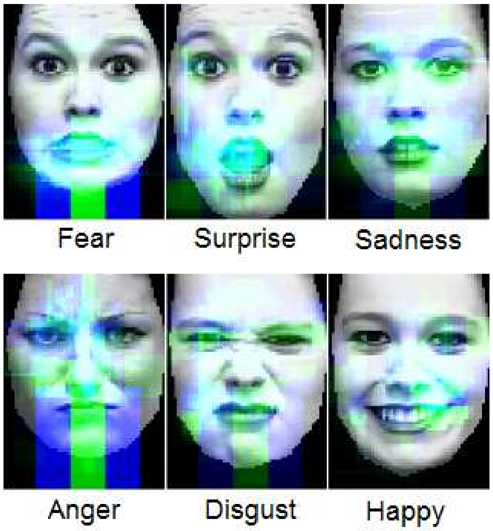


Figure 4.14: Graphical representation of the projection of Viola&Jones features selected by the optimal Adaboost.M2 classifier on face samples representing basic facial expression.

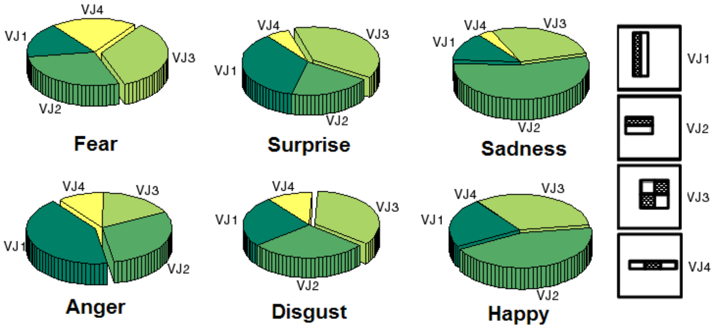


Figure 4.15: The proportion of each type of Viola&Jones features selected by the optimal Adaboost.M2 classifier.

Table 4.7: Results of data pruning on binary and multi-class classifiers using Viola&Jones features and 20 folds cross validation procedure.

| | **331** | | | **342** | | |
|---|---|---|---|---|---|---|
| | ac | tpr | fpr | ac | tpr | fpr |
| Ada.M2 | 89.12 | 67.56 | 2.72 | 89.41 | 63.63 | 1.63 |
| | 89.12 | 94.04 | 2.02 | 89.41 | 96.73 | 2.42 |
| | 89.12 | 89.06 | 4.87 | 89.41 | 90.76 | 3.64 |
| | 89.12 | 40.00 | 0.31 | 89.41 | 16.66 | 0.30 |
| | 89.12 | 80.48 | 1.03 | 89.41 | 86.36 | 2.36 |
| | 89.12 | 99.00 | 2.60 | 89.41 | 96.00 | 2.92 |
| avg | **89.12** | | | **89.41** | | |
| SVM | 91.14 | 53.03 | 5.76 | 91.50 | 51.66 | 5.76 |
| | 94.30 | 92.93 | 4.34 | 94.49 | 82.12 | 3.13 |
| | 93.35 | 80.23 | 3.66 | 95.31 | 85.73 | 3.54 |
| | 98.81 | 10.00 | 0.29 | 98.55 | 10.00 | 0.59 |
| | 95.46 | 77.50 | 3.03 | 93.86 | 75.00 | 4.24 |
| | 93.93 | 87.78 | 4.17 | 95.05 | 91.75 | 3.67 |
| avg | **94.49** | | | **94.79** | | |
| Ada | 91.24 | 51.35 | 3.74 | 89.77 | 43.59 | 4.29 |
| | 96.37 | 90.48 | 1.62 | 94.15 | 85.39 | 2.77 |
| | 93.35 | 81.25 | 3.75 | 90.35 | 77.61 | 6.55 |
| | 97.28 | 40.00 | 1.84 | 97.95 | 20.00 | 0.89 |
| | 96.98 | 90.24 | 2.07 | 95.91 | 87.80 | 2.99 |
| | 96.07 | 95.00 | 3.46 | 95.03 | 94.06 | 4.56 |
| avg | **95.21** | | | **93.85** | | |

Table 4.8: Mismatch rates of SVM and RVM binary classifiers. The terms SVs and RVs represent the number of support vectors and relevance vectors used in the models.

| **Expression** | **Mismatch rate** | **SVs** | **Mismatch rate** | **RVs** |
|---|---|---|---|---|
| *Surprise* | 14.32±1.80% | 63 | 6.25±1.51% | 21 |
| *Sadness* | 3.06±2.78% | 23 | 12.29±2.57% | 34 |
| *Anger* | 5.91±1.55% | 46 | 5.00±1.87% | 15 |
| *Happy* | 3.16±2.47% | 38 | 7.92±1.71% | 23 |
| *Disgust* | 9.54±1.97% | 34 | 8.54±1.91% | 25 |
| *Fear* | 24.97±2.07% | 72 | 15.00±2.38% | 38 |

Practically speaking, that means less processing time and also less memory for using this type of classifier.
Nevertheless, the analysis of facial expressions in static images has its own limitations. That can be mainly explained by the lack of dynamic characteristics of salient features involved in the structure of facial expressions. An important improvement for the recognition system may include also the encoding and use of the knowledge over these elements, as indicated by Datcu and Rothkrantz [39][202].

## 4.6 Conclusion

The current chapter highlighted the potential of boosting methods and kernel methods with LBP, Viola&Jones and geometric features, applied for recognizing expressions in single images. Although the models have comparable performance results, each presents specific advantages and limitations. This research work was one of the first that use the relatively new relevance vector machine method for the classification of basic facial expressions. The results of applying RVM indicates that this method has high results for facial expression classification in static images and that it leads to a decrease of complexity, compared to SVM. Further research aims at including more emotion classes for analysis. The still image analysis is very restrictive with respect to the subtle dynamics of the facial features. Additional research has been initiated to handle the temporal behaviour in the classification models so as to make possible the use of the recognition systems to run on image sequences. Another idea for increasing the capabilities and efficiency is to make use of fusion techniques to handle multiple video modalities.

# Chapter 5

# Facial expression in video sequences

## 5.1 Introduction

Automatic recognition of facial expressions from videos has an important role for the field of Human Computer Interaction - HCI. The research in this area has gathered a lot of attention from the community of scientists working in various disciplines such as behavioural sciences, social sciences and engineering. Factors such as rotation, occlusion, poor illumination, age, skin colour or presence of beard or glasses lead to an increase in the complexity for accurately determining the facial expressions in video sequences. During time, scientists have constantly tried to solve these specific problems using various methods.

In this chapter we focus on the dynamic recognition of six prototypic facial expressions of emotions as defined by Ekman [56]. We try to find out how rapid facial signals can be used for obtaining accurate automatic judgements of emotions and what is the estimation error in this case. Secondarily we run algorithms for the detection of action units - AUs which represent key elements of the FACS emotion labelling method. The AUs are capable to exhaustively describe the facial movements and so to provide encodings to any facial expression. The research on algorithms for detecting AUs may further lead to the automation of this process and so to the replacement of the classic approach of very time-consuming manual annotation procedure. In addition to that, the models may be extended for recognizing a much wider range of facial expressions.

Some research works by Datcu and Rothkrantz [45][46] and others [101][210][197] have attempted to use extra cues for the expression analysis. Such additional data is extracted from other informational channels that normally exist during a regular interaction between humans or between humans and computers.

This chapter presents several methods that have been previously used for the recognition of facial expressions. Namely we use features such as volume local binary patterns [218], Viola&Jones [188] and features based on optical flow together with boosting algorithms and support vector machines to run the classification process. A novel aspect of this work is the use of Viola&Jones features for facial expression recognition in video data. Another contribution is the use of Adaboost.M2 [76] algorithm as a multi-class classifier to operate on all the previously mentioned features. A new aspect on the practical side of the research is the possibility to work with a rather uncommonly large number of features (hundreds of thousands of features). In the same context, special arrangements are made for being able to run the recognition algorithms that make use of databases that range in size from several hundreds of megabytes to several gigabytes of memory space. Eventually, we present comparisons of the methods taken into consideration.

## 5.2 Related work

The role of optical flow for modelling the facial muscle movements in the context of showing different facial expression displays, was studied in [129]. The work of [62] describes a method for real-time tracking of facial expressions. The visual observation consists of the estimation of the optical flow that is coupled with a geometric model and with a facial muscle model. The result is a topologically invariant anatomically based model describing the force-based deformations of the tissue and skin and the control parameters of the muscle actuators. The statistical analysis of motion patterns in specific regions of the face was approached in [203] through a correlation-based optical flow model applied on points lying in rectangles which enclose various face features.

In [141], the velocity vector computed with a gradient-based optical flow constitutes the primary parameters for deriving features that are used to classify the prototypic facial expressions by means of a HMM model. The authors present an improvement of the optical-flow based HMM model [142] which was capable of handling time-sequential images that contain multiple facial expressions that could abruptly change from one expression to another expression. The work of [61] uses optical flow together with geometric, physical and motion-based dynamic models, in order to obtain a parametric representation of the independent muscle action groups and to estimate the face motion. The same methods are used also for the interpretation of facial expressions and as a tool for being used in an attempt to extend the FACS model specifications. Lien et al. [112] describe a computer vision system to automatically recognize facial expressions based on FACS action units. In the approach, the facial motion is estimated based on three methods of facial feature point tracking using a coarse-to-fine pyramid method, dense flow tracking together with principal component analysis and gradient component analysis in the spatio-temporal domain. In [114], the detection of action units is done using dense flow estimation by a wavelet motion model and second order 3-state and third order 4-state left-to-right discrete HMMs. More recently, the work [92] uses optical flow fields in the context of learning decision theoretic models of facial expressions and gestures using partially observable Markov decision processes. In [9], Aleksic and Katsaggelos use FAPs describing the movement of the outer-lip contours and eyebrows observations into a multi-stream hidden Markov model for automatic facial expression recognition. The work [138] proceeds with the classification of six basic facial expressions by using Lukas-Kanade optical flow to estimate the motion in a set of 12 rectangular so-called flow regions which are related to a subset of twelve facial muscles. The preliminary work of Duthoit et al. [54] presents a compass-based visualization of the optical flow estimates as a method to deter-

mine the dominant motion vectors of facial expressions.
The original Local Binary Pattern visual operator [139] has been recently extended to video analysis in the form of volume LBP transforms. The few papers that use this approach report extremely high classification performance for facial expression recognition. Choosing for volume LBP as features from video data and for SVM with second degree polynomial kernel function, Zhao et al. [218][219] successfully obtained 96.27% overall classification rate of facial expressions from video on the Cohn-Kanade database. The authors extend the static LBP by carrying the analysis at three parallel (VLBP) or orthogonal (LBP-TOP) planes according to image and temporal axis. The paper of Taini et al. [172] presents an approach for the recognition of six basic facial expressions from near-infra-red video sequences by using LBP-TOP features. The features are extracted from 9X8 volume blocks from each video sequence. By converting the six-class classification into 15 binary SVM-based classifiers, the authors achieve different results for each context of analysis: 79.40% in strong, 73.03% in weak and 76.03% in dark illumination. The paper also describes comparative experiments on visual-light video samples. The dataset consisted of 1602 video samples and the evaluation method was 10-fold cross validation.
Several papers [215][113][34][114][177][176][179][110][216][14][32][200][201] [171] have tackled the recognition of facial expressions by firstly assessing the activity of facial muscles through the analysis of action units.
Bartlett et al. [15] approach the automatic detection of face actions in sequences of images with holistic spatial analysis, explicit measurement of face features and with estimation of motion flow fields. The holistic method implies the detection of action units using feed forward neural network on coefficients obtained by applying PCA on difference image data. The second approach implies the use of neural networks on measurements of face wrinkles and eye opening. The third method uses a correlation similarity measure on face flow vectors. In [95], the authors use 10 action parameters computed by the difference between the geometric features of face in the first frame and the last frame. Principal component analysis and 2D Gaussian models are then used to reduce the dimension of these features. The facial expressions are finally determined from videos, using distance-based classification in 2D emotion space and feature profile correlation.

## 5.3 Data set

The models for facial expression recognition in video data we present in this chapter use video samples from the Cohn-Kanade database [103]. The database originally contains video recordings of subjects showing the prototypic six facial expressions. Each recording starts with the subject in neutral state and gradually goes through a sequence of intermediate frames to the frame in which the subject shows fully developed facial expressions. The recorded expressions are acted, are affected by subject's knowledge on the appearance of each expression and follow the instructions given to each subject.
Based on the appropriateness and quality of the frame images, we have first made a selection of video sequences. Some samples were rejected because of occlusion (e.g. hair covering the face of the subject in some cases or the layer containing video information-oriented text overlapping the face area) or exagger-

ate rotation of the subject head. Our selection of samples resulted in a dataset of 480 video samples (Fear: 85, Surprise: 107, Sadness: 91, Anger: 29, Disgust: 59, Happy: 109).

The procedure of generating samples by preprocessing the frames from each video sequence had the following steps: the detection of faces in each frame image of video sequences; the computation of face shape of each detected face; the alignment of faces in each frame of each video sequence; normalization and rescaling to width:60xheight:80 pixels image. The face proportions were preserved by forcing the middle point on the line segment between the eyes be positioned at a specific location and by setting the distance from the left most to the right most points along the horizontal eyes' line to be constant. The detection of faces was realized with a standard Viola&Jones face detector [188]. Active Appearance Model - AAM [55] has been used for extracting the face shape in each image frame. Compared to many of the previous works, the face image samples we obtained do not contain any background information nor include partially cut face regions.

We obtained sharp samples of undistorted faces lying on a pure black background by using face shape information provided by AAM. Figure 5.1 illustrates a sample obtained during the data-preprocessing step. As it can be seen, we used the visual data only from the face area. Eventually we have kept only three image frames for each video recording by considered the first, the middle and the last frames of the sequence. For the facial expression analysis based on optical flow we have selected four samples per video sequence at equal temporal distance.

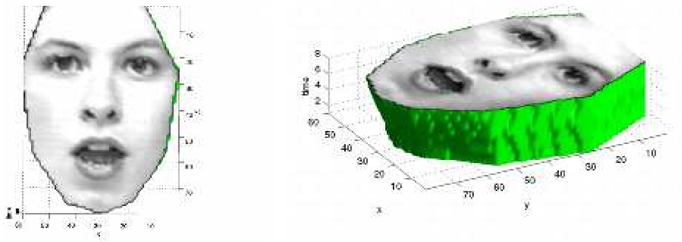


Figure 5.1: Example of a video sequence containing only the face area.

## 5.4 Models

### 5.4.1 Volume Local Binary Patterns

The changes in motion and appearance of facial expressions may be modelled in the context of dynamic texture analysis, by considering volume local binary patterns - VLBPs [219] as highly discriminative features. An advantage of using VLBP features is the good robustness to translation, rotation and illumination.

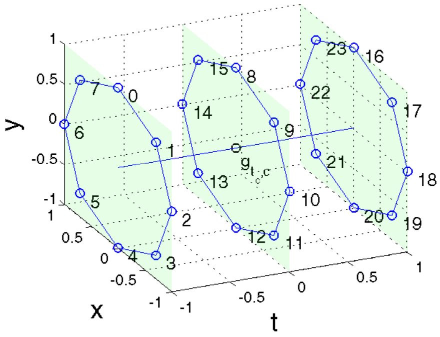


Figure 5.2: Basic VLBP with radius $R = 1$ and number of pixels in plane $P = 8$.

The concept represents an extension of original local binary patterns - LBPs [139] to spatial-temporal domain. The LBP-TOP block-based approach takes into account information at the pixel-level, region-level and volume-level, by performing elliptical sampling in spatial and temporal orthogonal planes. The basic 3D LBP (or VLBP) operator samples neighbour points around the centre pixel in the local volume associated with three parallel planes (figure 5.2) which correspond to the three selected frames of each video sequence. The value of each VLBP feature is computed using the formula:

$$VLBP_{P,R} = \sum_{q=0}^{3P+1} v_q 2^q,$$

where $v_q$ are the elements of vector $V$:

$$\begin{aligned} V = & v(s(g_{Fr_1,c} - g_{Fr_2,c}), \\ & s(g_{Fr_1,0} - g_{Fr_2,c}), s(g_{Fr_1,1} - g_{Fr_2,c}), .., s(g_{Fr_1,P-1} - g_{Fr_2,c}), \\ & s(g_{Fr_2,0} - g_{Fr_2,c}), s(g_{Fr_2,1} - g_{Fr_2,c}), .., s(g_{Fr_2,P-1} - g_{Fr_2,c}), \\ & s(g_{Fr_3,0} - g_{Fr_2,c}), s(g_{Fr_3,1} - g_{Fr_2,c}), .., s(g_{Fr_3,P-1} - g_{Fr_2,c}), \\ & s(g_{Fr_3,c} - g_{Fr_2,c})). \end{aligned}$$

The terms $\{g_{Fr_k,c}\}$ represent the grey values of the centre points and $\{g_{Fr_k,0}$, ..., $g_{Fr_k,P-1}\}$ are gray levels of $P$ neighbouring points sampled from the local region in frame $k$ of the video sequence. VLBP-TOP takes the concatenation of binary codes $XY - LBP$, $XT - LBP$ and $YT - LBP$ of basic 2D LBP codes in $XY$, $XT$ and $YT$ orthogonal planes (figure 5.3). The neighbour pixels are selected using elliptical sampling. The notation of LBP-TOP feature is $LBP - TOP_{P_{XY},P_{XT},P_{YT},R_X,R_Y,R_T}$, where the terms $R_X$, $R_Y$, $R_T$ are the radii and $P_{XY}$, $P_{XT}$, $P_{YT}$ are the numbers representing the sample points in $XY$, $XT$ and $YT$ planes.

For experiments, we have generated LBP-TOP features with $P_{XY} = \{2, 4, 6, 8\}$ and $P_{XT} = P_{YT} = 4$. For every video sequence, the complete set of LBP-TOP features was obtained by splitting the 60x80 pixels image frames into different number of regions (figures 5.4 and 5.5) on $X$ and $Y$ axes and by computing the LBP-TOP features from each region. The radii were scaled up so as to fit

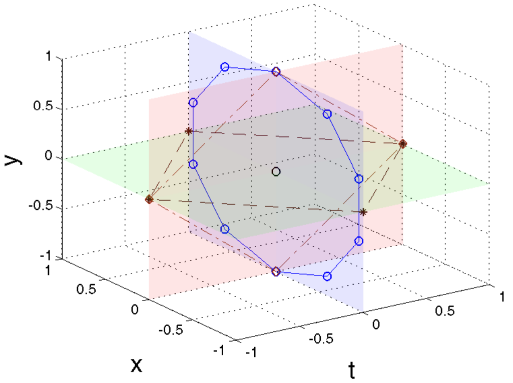


Figure 5.3: LBP-TOP sampled with radii $R_X = R_Y = R_Z = 1$ and different number of pixels in plane ($XY$:10, $XT$:4, $YT$:4).

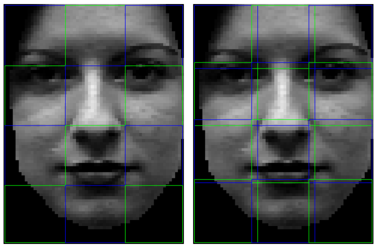


Figure 5.4: 3X4 split configuration, non-overlapping and 20% overlapping blocks.

each region size. In this research, we used both categories of non-overlapping and overlapping (10%, 20%, 30%, 40%, 50%) block regions. We started by considering the whole image region and further split in $2, 3, ..10$ block regions on both $X$ and $Y$ axis (100 split configurations in total). Based on the Cohn-Kanade database, in total we have derived 4 configurations regarding the number of points on $XY$ plane, 6 block overlapping rates and 100 split configurations (1 to 10 blocks on horizontal axis and 1 to 10 blocks on vertical axis). That makes 2.400 data sets of LBP-TOP features to train and test.

### Gentle Adaptive Boosting - GentleBoost

Table 5.1 shows performance of the best one-against-the-rest GentleBoost classifiers, from the set of 2400 data sets of combined VLBP and VLBP-TOP features. The term $NP$ is the number of points per plane, $NX$ and $NY$ are the number of splits on $X$ respectively $Y$ axis and $OR$ is the overlap ratio.
The terms fpr, tpr and ac represent the false positive rate, the true positive rate and the accuracy. They represent measures on the performance of the classification process. For evaluation, we have used leave-one-out cross validation

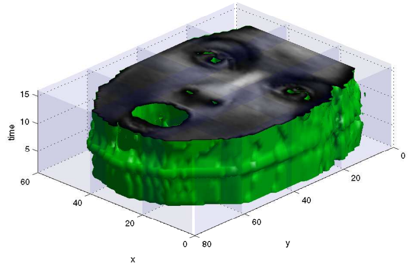


Figure 5.5: Example of 3X4 non-overlapping blocks applied on a video sequence.

method. These performance indicators will be used also in the rest of the chapter. GentleBoost classifier committees were trained with 200 stages. As it can be seen from the table, the performance of the detection for the six facial expressions is very high and very close to the state-of-the art performance achieved by other researchers. Still, our intention was to study the use of a multi-class classifier for the problem of recognizing the facial expressions. Therefore, we use Adaboost.M2 as a multi-class dynamic classifier.

Table 5.1: Test results of GentleBoost 1-R ensembles trained for recognizing basic facial expressions.

| Clsf.1-R | NP | NX | NY | OR | fpr(%) | tpr(%) | ac(%) |
|---|---|---|---|---|---|---|---|
| Fear | 10 | 2 | 1 | 50% | 0.00 | 82.35 | 96.88 |
| Surprise | 6 | 2 | 1 | 20% | 0.54 | 87.85 | 96.88 |
| Sadness | 10 | 1 | 2 | 10% | 0.00 | 92.31 | 98.54 |
| Anger | 8 | 2 | 1 | 30% | 0.00 | 93.10 | 99.58 |
| Disgust | 8 | 2 | 1 | 40% | 0.00 | 76.27 | 97.08 |
| Happy | 8 | 2 | 1 | 40% | 0.00 | 92.66 | 98.33 |

**Adaboost.M2**

To our best knowledge, to date no research has been published on the use of Adaboost.M2 classifier for the recognition of facial expressions in video data. For classification, we have used a set of 145.200 input parameters consisting of all numbers of points per plane, image block splitting and overlapping combinations of VLBP and VLBP-TOP features. The accuracy of the Adaboost.M2 classifier trained with the set of VLBP and VLBP-TOP features is 73.33% (see table 5.2). For evaluation, we have used 20-fold cross validation procedure.

### 5.4.2 Viola&Jones features

The Viola&Jones features have been originally introduced for being used with 2D images. More recent attempts have addressed the application of these features on video image sequences. The method implied a volumetric approach based on the computation of features from each of the constituent frame images of the video sequence. The 3D Viola&Jones features used for the recognition of

Table 5.2: The confusion matrix of the Adaboost.M2 classifier (20-fold cross validation) that uses VLBP and VLBP-TOP features (left) and classification measures (right).

| % | Fear | Surprise | Sadness | Anger | Disgust | Happy |
|---|---|---|---|---|---|---|
| Fear | **64.70** | 12.94 | 7.05 | 1.17 | 3.52 | 10.58 |
| Surprise | 11.21 | **79.43** | 2.80 | 0.93 | 2.80 | 2.80 |
| Sadness | 14.28 | 6.59 | **74.72** | 0 | 1.09 | 3.29 |
| Anger | 13.79 | 17.24 | 24.13 | **20.68** | 10.34 | 13.79 |
| Disgust | 6.77 | 3.38 | 16.94 | 1.69 | **64.40** | 6.77 |
| Happy | 4.58 | 0.91 | 1.83 | 0 | 0.91 | **91.74** |

| fpr(%) | tpr(%) |
|---|---|
| 9.62 | 64.71 |
| 6.70 | 79.44 |
| 7.20 | 74.73 |
| 0.67 | 20.69 |
| 2.61 | 64.41 |
| 6.20 | 91.74 |

facial expressions are represented graphically in figure 5.6. Each of the features is effectively computed by making use of the integral video as an intermediate representation of the original image sequence. At spatial location $(x, y)$ and time $t$, the integral video $iv(x, y, t)$ is computed as the sum of intensities of all pixels within the volumetric region delimited by $(x, y, t)$:

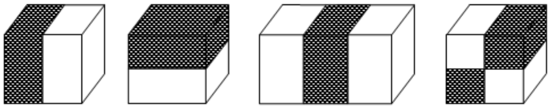


Figure 5.6: Basic types of 3D Viola&Jones features.

$$iv(x,y,t) = \sum_{x' \le x, y' \le y, t' \le t} i(x', y', t'),$$

where $i(x, y, t)$ is the original video. The integral video can be computed in one pass over the image sequence by using the following recurrences:

$$\begin{aligned} s_1(x,y,t) &= s_1(x,y-1,t) + i(x,y,t) \\ s_2(x,y,t) &= s_2(x-1,y,t) + s_1(x,y,t) \\ iv(x,y,t) &= iv(x,y,t-1) + s_2(x,y,t), \end{aligned}$$

where $s_1(x, -1, t) = s_2(-1, y, t) = iv(x, y, -1) = 0$. The sum of intensities of all the pixels in volume $A$ in figure 5.7 may be determined using eight references to the integral video, as follows: $S_A = p_8 + p_1 + p_6 + p_3 - (p_5 + p_4 + p_7 + p_2)$. For the case of using only three frames from each video sequence, the computation of VJ features is done by firstly determining the 2D features from each of the selected frame images and secondly by summing up the results for each feature (figure 5.8). Based on an image size of width:60xheight:80 pixels, we have generated a set of 617.525 basic features (170.775 features for each of the first three V&J feature types and 105.200 features of the last type).

### Adaboost.M2

For recognizing facial expressions with 3D Viola&Jones features, we used AdaBoost.M2 classifier on a data set of 308.763 features (half of the initial fea-

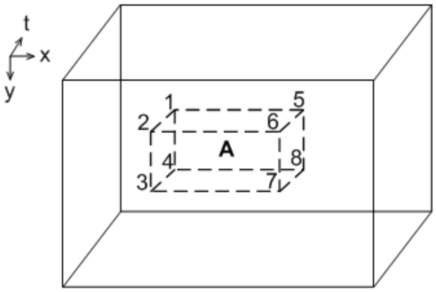


Figure 5.7: Integral video representation.

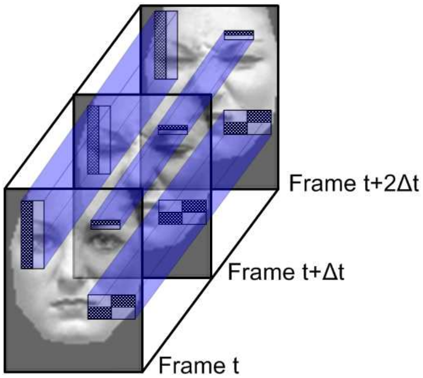


Figure 5.8: 3D Viola&Jones features applied on a video which shows the onset of facial expression ’disgust’.

ture set) determined for the selected video sequences from the Cohn-Kanade database.
We have used leave-one-out cross validation and Adaboost.M2 classifier with maximum 100 stages. The accuracy for the recognition of prototypic facial expressions is 74.17%. As can be seen from table 5.3, the classifier misses all the 29 samples of the facial expression anger.

### 5.4.3 Optical Flow

The production of facial expressions is naturally associated with the activation of various facial muscles. In turn, this leads to specific movement patterns in the underlying face tissue. Under the assumption that the face image intensity is locally constant across short periods of time, optical flow measures the magnitude and the direction of the movement by computing the displacement of pixels in the face area. For that, we have used an algorithm which computes

Table 5.3: Left: confusion matrix of the model using 3D VJ features; right: classification measures of the same model.

| % | Fear | Surprise | Sadness | Anger | Disgust | Happy |
|---|---|---|---|---|---|---|
| Fear | **82.35** | 3.52 | 4.70 | 0 | 1.17 | 8.23 |
| Surprise | 5.60 | **86.91** | 3.73 | 0.93 | 0.93 | 1.86 |
| Sadness | 12.08 | 5.49 | **72.52** | 1.09 | 5.49 | 3.29 |
| Anger | 13.79 | 0 | 48.27 | **0** | 27.58 | 10.34 |
| Disgust | 16.94 | 3.38 | 20.33 | 5.08 | **49.15** | 5.08 |
| Happy | 5.55 | 0.92 | 1.85 | 0 | 1.85 | **89.81** |

| fpr(%) | tpr(%) |
|---|---|
| 9.39 | 82.35 |
| 2.96 | 86.92 |
| 9.28 | 72.53 |
| 1.11 | 0.00 |
| 4.05 | 49.15 |
| 4.85 | 89.81 |

the pyramidal Lucas-Kanade optical flow [119]. The method has considerable high speed for processing the data. The following part presents this algorithm in detail. Given two consecutive frames $Fr_i$ and $Fr_{i+1}$ having the same width $n_x$ and height $n_y$, that reflect different stages of the onset or offset of some possibly mixed facial expressions, the image pyramid representations are created at first. If $Fr_i^0$ and $Fr_{i+1}^0$ represent the original images with resolution $n_x^0 = n_x$ and $n_y^0 = n_y$, the following layers of the pyramid are constructed recursively $Fr_i^1$, $Fr_i^2$, ..., $Fr_i^L$ and $Fr_{i+1}^1$, $Fr_{i+1}^2$, ..., $Fr_{i+1}^L$. The images $Fr_i^l$ and $Fr_{i+1}^l$ have the width $n_x^l$ and height $n_y^l$ determined as the largest integers that satisfy:

$$\begin{aligned} n_x^l &\leq \frac{n_x^{l-1}+1}{2} \\ n_y^l &\leq \frac{n_y^{l-1}+1}{2}. \end{aligned}$$

According to the previous rule for sampling images at different layers of the pyramid, the coordinates of one pixel $u$ in image $Fr_i^0$ or image $Fr_{i+1}^0$, translate, in image $Fr_i^l$ or image $Fr_{i+1}^l$ at layer l, to the equivalent coordinates $u^l$: $u^l = u/2^l$. The optical flow is determined starting with the deepest level (that corresponds to the original image) and by propagating the result at each step as initial guess on the pixel displacement for the next level. At one level, the displacements of pixels are computed iteratively, in a sequence of steps 1,2,...,$k$. The region in which the displacement of pixel $(p_x, p_y)$ is determined, is confined to small neighbourhood areas $A$ and $B_k$ from images $Fr_i^l$ and $Fr_{i+1}^l$ at layer $l$, as follows:

$$\begin{aligned} \forall(x,y) \in [p_x - w_x - 1, p_x + w_x + 1] \times [p_y - w_y - 1, p_y + w_y + 1] \\ A(x,y) = Fr_i^l(x,y) \\ \forall(x,y) \in [p_x - w_x, p_x + w_x] \times [p_y - w_y, p_y + w_y] \\ B_k(x,y) = Fr_{i+1}^l(x + \nu_x^{k-1}, y + \nu_y^{k-1}), \end{aligned}$$

where $\bar{\nu}^{k-1} = [\nu_x^{k-1}, \nu_y^{k-1}]$ is the initial estimation of the pixel displacement, as computed at step $k-1$. At step $k=1$, the initial guess is initialized with zero: $\bar{\nu}^0 = [0,0]^T$. The process is carried in such a way so as to minimize an image matching error function. More specifically, the goal is to determine the residual displacement of pixel $(p_x, p_y)$, as the vector $\bar{\eta}^k = [\eta_x^k, \eta_y^k]$ which minimizes the error function:

$$\varepsilon^k(\bar{\eta}^k) = \varepsilon(\eta_x^k, \eta_y^k) = \sum_{x=p_x-w_x}^{p_x+w_x} \sum_{y=p_y-w_y}^{p_y+w_y} (A(x,y) - B_k(x+\eta_x^k, y+\eta_y^k))^2.$$

The solution is given by Lucas-Kanade optical flow and has the following form: $\bar{\eta}^k = G^{-1}\bar{b}_k$, where the term $\bar{b}_k$ relates to a $2 \times 1$ vector of the form:

$$\bar{b}_k = \sum_{x=p_x-w_x}^{p_x+w_x} \sum_{y=p_y-w_y}^{p_y+w_y} \begin{bmatrix} \delta I_k(x,y) & I_x(x,y) \\ \delta I_k(x,y) & I_y(x,y) \end{bmatrix}.$$

The image difference $\delta I_k$ is given by:

$$\begin{aligned} &\forall (x,y) \in [p_x - w_x, p_x + w_x] \times [p_y - w_y, p_y + w_y] \\ &\qquad\qquad \delta I_k(x,y) = A(x,y) - B_k(x,y). \end{aligned}$$

Additionally, the spatial derivatives $I_x$ and $I_y$ have the following expressions:

$$\begin{aligned} &\forall (x,y) \in [p_x - w_x, p_x + w_x] \times [p_y - w_y, p_y + w_y] \\ &I_x(x,y) = \frac{\partial A(x,y)}{\partial x} = \frac{A(x+1,y) - A(x-1,y)}{2} \\ &I_y(x,y) = \frac{\partial A(x,y)}{\partial y} = \frac{A(x,y+1) - A(x,y-1)}{2}. \end{aligned}$$

The term $G$ from the solution of the Lucas-Kanade optical flow, denotes a $2 \times 2$ matrix which has the form:

$$G = \sum_{x=p_x-w_x}^{p_x+w_x} \sum_{y=p_y-w_y}^{p_y+w_y} \begin{bmatrix} I_x^2(x,y) & I_x(x,y)I_y(x,y) \\ I_x(x,y)I_y(x,y) & I_y^2(x,y) \end{bmatrix}.$$

The updated pixel displacement $\bar{\nu}^k$ at step $k$ is computed using the residual displacement $\bar{\eta}^k$, as follows: $\bar{\nu}^k = \bar{\nu}^{k-1} + \bar{\eta}^k$. At the next step $k+1$, the value of $\bar{\nu}^k$ becomes the initial guess for determining the updated Lucas-Kanade optical flow. The solution for optical flow is obtained after a fixed number of steps $K$ or when the computed value of the residual $\bar{\eta}^k$ is smaller than a threshold.
The total displacement of pixel $(p_x, p_y)$ at pyramid layer $l$ then becomes:

$$\bar{\nu} = d^l = \bar{\nu}^K = \sum_{k=1}^{K} \bar{\eta}^k.$$

The final solution for the total pixel displacement is determined by repeating the iterative optical flow for all the pyramid layers $l = L, L-1, ..., 0$. Eventually, the total displacement of pixel $(p_x, p_y)$ from frame $Fr_i$ to frame $Fr_{i+1}$ has the form: $d = \sum_{l=0}^{L} 2^l d^l$. The optical flow for face images can be computed using a grid of control points that are uniformly located on the face area.
An example of a grid of control points mapped on the face area is illustrated in figure 5.9. Along with the size of the pyramid $L$ and the size of integration

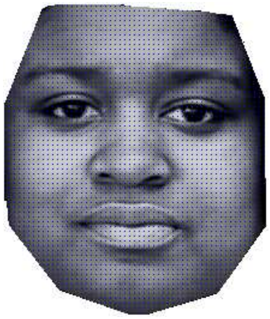


Figure 5.9: Grid of 3.019 control points for estimating optical flow from face images.

window $\{w_x, w_y\}$, the density of the control points along both $X$ and $Y$ axes represent parameters for the application of optical flow for face image sequences. Given the Cohn-Kanade database, 24 data sets were generated according to variation in the pixel sampling density (control points located every 2 respectively 3 pixels along $X$ and $Y$ axes), size of integration window ($w_x$, $w_y$={3,4,5,6} pixels) and pyramid size ($L$={1,2,3} layers).

The pixel displacements associated to the control points were computed for every two consecutive image frames, for all video sequences. The optical flow resulted for one video sequence is shown, as an example, in figure 5.10. Two data sets were eventually made by selecting three respectively four frames from each video sequence in the database. Applying optical flow method lead to two respectively three images, each containing the reunion of all the pixel displacements on the face area. By combining the images, we have obtained image samples that contain facial motion vectors representing the onset of different facial expressions. Both procedures are illustrated in figure 5.11. For recognizing facial expressions, we have experimented with different classification methods on the two data sets.

Beside considering optical flow map, we have also tested the VLBP and Viola&Jones features (figure 5.12) as inputs for the classifiers. Learning Adaboost.M2 classifier with 100 training stages on optical flow data obtained by setting pixel sampling density to 3, size of integration window to 3 and pyramid size to 1, has lead to the test recall rate of 85.41% (figure 5.13). The classifier has been selected as the best from 24 Adaboost.M2 classifiers (with ROC graph depicted in figure 5.14) trained on optical flow data resulted by changing the specific optical flow parameters. The confusion matrix of the best classifier obtained based on optical flow parameters $< 3.3.1 >$ is presented in table 5.4.

### 5.4.4 Dynamic Facial Characteristic Point model

While generating facial expressions, the face presents significant changes of shape as well as of texture. As it is illustrated in image 5.15 for the basic facial expressions, the onsets potentially determine specific patterns in both

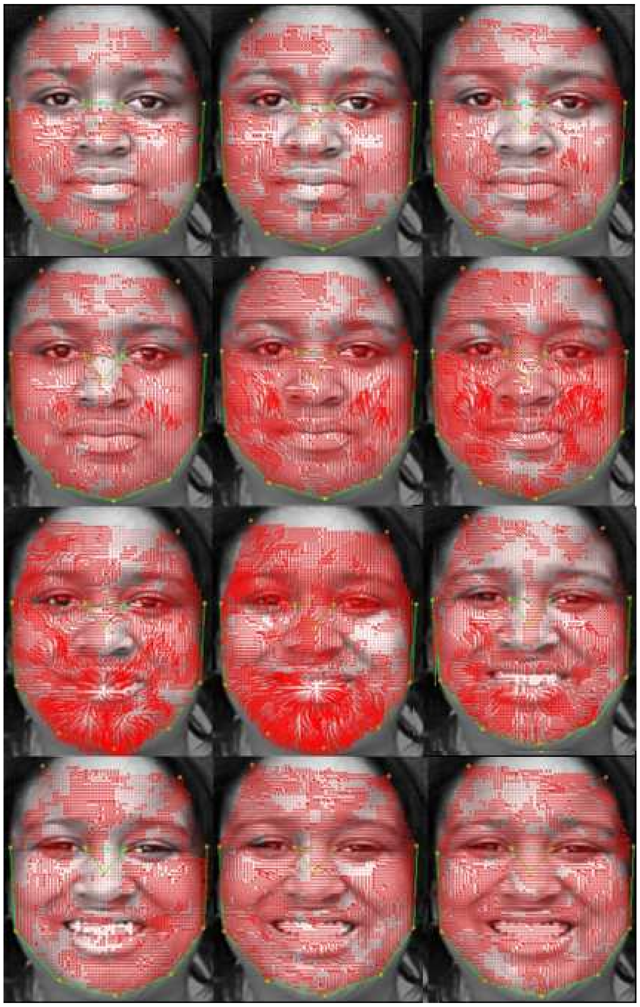


Figure 5.10: Example of optical flow estimation in a video sample from the Cohn-Kanade database.

Table 5.4: Confusion matrix (left) and classification measures (right) of the best classifier trained with optical flow features.

| % | Fear | Surprise | Sadness | Anger | Disgust | Happy |
|---|---|---|---|---|---|---|
| Fear | **85.88** | 4.70 | 3.52 | 1.17 | 0 | 4.70 |
| Surprise | 6.54 | **85.98** | 4.67 | 0 | 0.93 | 1.86 |
| Sadness | 6.59 | 3.29 | **85.71** | 2.19 | 1.09 | 1.09 |
| Anger | 3.44 | 0 | 27.58 | **48.27** | 20.68 | 0 |
| Disgust | 1.69 | 3.38 | 3.38 | 1.69 | **88.13** | 1.69 |
| Happy | 4.62 | 0.92 | 1.85 | 0 | 0 | **92.59** |

| fpr(%) | tpr(%) |
|---|---|
| 5.07 | 85.88 |
| 2.68 | 85.98 |
| 5.15 | 85.71 |
| 0.88 | 48.27 |
| 1.90 | 88.13 |
| 2.15 | 92.59 |

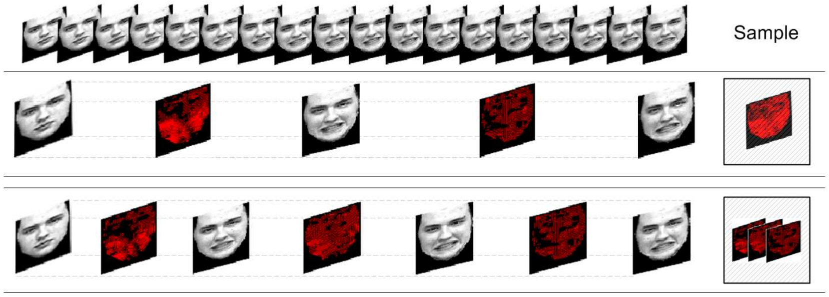


Figure 5.11: Two frame selection methods based on optical flow.

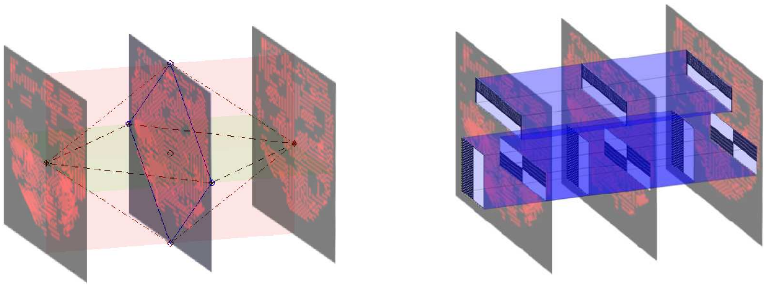


Figure 5.12: The use of optical flow maps together with VLBP features (left) and with Viola&Jones features (right).

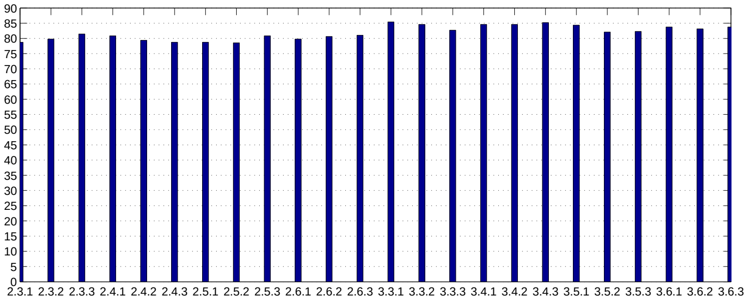


Figure 5.13: The accuracy (%) achieved by Adaboost.M2 facial expression classifier for each combination of optical flow parameters: <ofpctdense.ofsizecell.ofnrlevpyr>.The highest recall rate (85.41%) is obtained for configuration: $< 3.3.1 >$.

texture and shape. The facial expressions may be modelled by considering the dynamics that normally occur in the face appearance. In this way, we computed the variation of size of line segments delimited by pairs of characteristic points

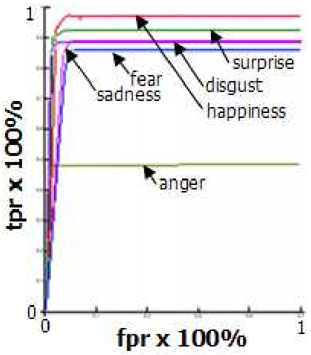


Figure 5.14: ROC graph of facial expression recognition using features computed from optical flow maps (Adaboost.M2 classifier and leave-one-out cross validation).

on the face area. Such characteristic points relate to key points of facial features like the inner and outer corners of the eyes and eyebrows or the upper and lower points of the mouth and eyes.

In order to determine the location of the characteristic points on the face area, we have used active appearance model, as described in chapter 3. Prior to using AAM for extracting the face shape, we have used a Viola&Jones face detector [187]. The results of the face detector are used as initial estimation of the location of the model face in the image face. Given the appearance model, the locations of facial characteristic points were computed in each frame of the video sequence.

We have used an adapted set of Facial Characteristic Point model (see figure 4.6 in chapter 4) based on the Kobayashi&Hara FCP model [106]. The location information was used to track the characteristic points and to determine the variation of the size of line segments between different pairs of characteristic points. Table 4.2 shows all the geometric features used for the analysis. Figure 5.16 shows how the variation of geometrical features correlates with the onset of each prototypic emotion. We have used the geometrical features and support vector machine classifier [185] to model the patterns of face shape deformation. For the analysis, we have used a C++ implementation of multi-class SVM [28]. The results of applying SVM to facial expression data using the variation of geometric feature is presented in table 5.5.

## 5.5 Action unit detection

Apart from directly recognizing facial expressions, we also considered FACS labels from the Cohn-Kanade database for detecting facial action units in video sequences. The detection was done by applying binary Adaboost classifier with maximum 100 training steps. In total, we have used 72.600 VLBP and VLBP-

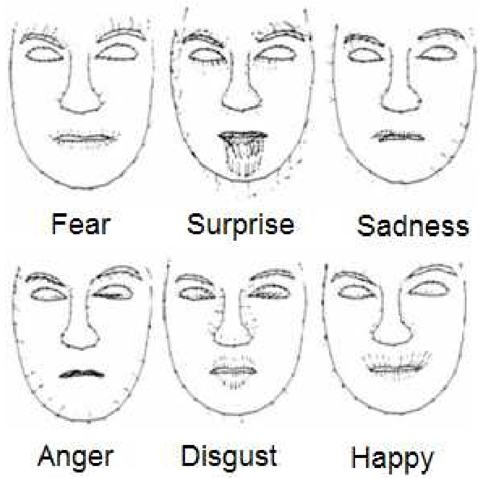


Figure 5.15: Example of face shape deformation for each facial expression.

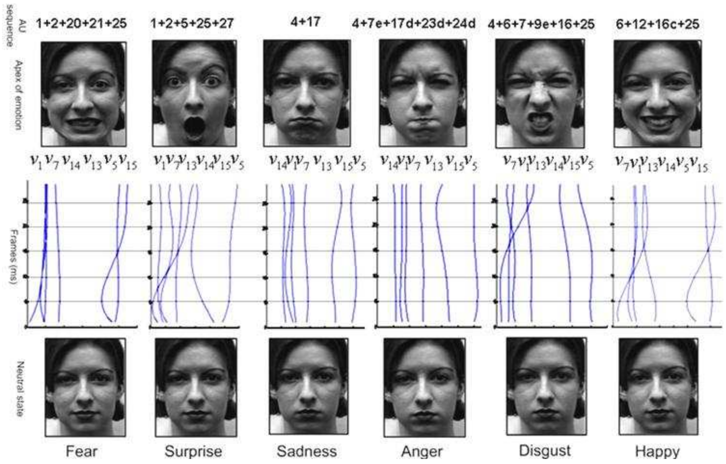


Figure 5.16: The variation of FCP-based parameters for each facial expression.

TOP combined features extracted from image sequences, where the size of a frame image was 60x80 pixels. The results of this analysis for 13 AUs, are shown in table 5.6.

Table 5.5: The confusion matrix (%) for the facial expression recognition using SVM (polynomial kernel of degree 3).

| | Fear | Surprise | Sadness | Anger | Disgust | Happy |
|---|---|---|---|---|---|---|
| Fear | **88.09** | 2.38 | 4.76 | 3.57 | 1.19 | 0 |
| Surprise | 0 | **88.67** | 2.83 | 8.49 | 0 | 0 |
| Sadness | 5.43 | 2.17 | **85.86** | 2.17 | 1.08 | 3.26 |
| Anger | 10.71 | 0 | 3.57 | **85.71** | 0 | 0 |
| Disgust | 5.35 | 5.35 | 3.57 | 1.78 | **82.14** | 1.78 |
| Happy | 4.62 | 0 | 7.40 | 2.77 | 5.55 | **79.62** |

Table 5.6: Results for the detection of 13 AUs. The analysis was done using the Adaboost classifier with VLBP and VLBP-TOP features.

| **AU** | **AU1** | **AU2** | **AU4** | **AU5** | **AU6** | **AU7** | **AU9** |
|---|---|---|---|---|---|---|---|
| *fpr(%)* | 1.18 | 0.00 | 2.50 | 1.23 | 2.16 | 2.25 | 0.45 |
| *tpr(%)* | 94.05 | 95.31 | 92.06 | 90.48 | 94.23 | 96.61 | 88.24 |
| *ac(%)* | 97.24 | 98.82 | 95.10 | 97.06 | 96.86 | 97.47 | 98.03 |
| | **AU12** | **AU15** | **AU17** | **AU20** | **AU25** | **AU27** | |
| *fpr(%)* | 1.52 | 1.61 | 1.22 | 0.92 | 2.30 | 0.00 | |
| *tpr(%)* | 96.43 | 91.43 | 97.78 | 94.44 | 97.79 | 93.62 | |
| *ac(%)* | 98.03 | 97.29 | 98.43 | 98.43 | 97.76 | 98.82 | |

## 5.6 Results

In this chapter, we have presented various methods for the recognition of facial expressions in video data. The supervised models have been learned based on instances from the Cohn-Kanade database. The methods with good performance in classification have been based on the use of support vector machine and geometric features computed from facial characteristic points and on the use of optical flow based features and Adaboost.M2 classifier.

In addition to the face analysis of visual features with direct recognition of facial expressions, we have also studied the detection of action units from FACS. In that context, we have trained and tested Adaboost classifiers with a data set of samples consisting of 3 frames per video sample. Each classifier was able to perform the detection of one action unit. In total, we have built 13 classifiers. The training implied a number of 100 training stages. As shown in table 5.6, the performance is very high with a minimum accuracy of 95.10% and maximum of 98.82%. There is a rather big difference between these results and the results of recognizing facial expressions (85.41% in case of Adaboost.M2). One straightforward explanation relates to the nature of the two classification algorithms namely the classic Adaboost is a binary classifier while Adaboost.M2 is a multi-class classification method. The training of a classifier for the detection of one action unit involved a one against the rest data preprocessing and classification procedure. The existence of multiple classes to be handled simultaneously implies that the errors made during the classification of one emotion class evidently affect the classification of the other emotion classes. Indeed, the instances that account for the false negative rate in the case of recognizing samples of one class of emotions accounted for the false positive rate in the case of other class or possibly of other classes of emotions.

Another observation is that the more unbalanced the data set is with respect to the number of samples per class, the better is to use more validation folders for the cross validation procedure. Of course, it is a trade off between the time required by running the algorithms and the effectiveness of the training and measuring processes. Secondly, another explanation is in the manner of classification of the two algorithms. Both methods consist of linearly combining sets of so-called weak learners that, in our case, stand for basic decision tree classifiers. For Adaboost, the construction of weak learners follows the measurement of a prediction error computed given distributions over sample weights. The Adaboost training mechanism assumes that the samples that are currently misclassified will be overemphasized so as to achieve good classification at the next step.

For the classification of facial expressions, we dealt with noisy data and that has negative implication towards the tendency for overfitting. That is also one reason for which we have used k-fold cross validation procedure to measure the performance of the recognizers. In turn, in the case of Adaboost.M2 the weak classifiers are influenced by the pseudo-loss function that takes into account distributions on samples and weighting functions over the sample labels. The weak learner is trained so as to account for both hard to classify samples and incorrect class labels. The influence of incorrect class labels on the pseudo-loss function and on the final classification performance increases with the consistency of class separability and with the number of classes.

At the emotion class level, the performance of recognition is primarily affected by the number of samples per class used for training. As shown in tables 5.2, 5.3 and 5.4, the true positive rate for facial expression classes anger and disgust are strongly correlated with the low number of samples in the initial data set.

One problem we had to confront with during the experiments was the computationally demanding models. On one hand, we had data sets that range up to several GBytes of RAM memory and on the other was the extensive amount of time needed to run the algorithms. The solution for the first problem was to acquire superior hardware platforms. For the second, we have made an improvement to the Adaboost.M2 classification algorithm that allowed for a parallel implementation. As consequence, for instance the amount of time needed for completing the training task for recognition of facial expressions using Viola&Jones features was approximately two weeks on an Intel Xeon Dual Core, Quad 2,66 GHz with 16 Gb RAM memory Linux machine given that all eight CPU cores were used in parallel.

## 5.7 Conclusion

An essential contribution of the research presented in this chapter comes from a practical issue regarding the preprocessing of data. We use face shape information to clip the face area and so to filter out the background. By using appearance models, we have achieved a better way to generate clear, uncut and noise-free image samples of faces.

Some of the previous researches have worked with data sets of face samples which were obtained by simply clipping the face area from the original image. The process produced face image samples with either missing face parts (i.e. ears, top of the head) or extra visual data commonly being part of the background. In this study, we have used classification methods such as support vector machines and a variety of boosting methods like Adaboost and Adaboost.M2. While the first boosting method is suitable for binary classification, the use of Adaboost.M2 represented a proper option for the simultaneous analysis of multiple emotion classes. The results we obtained for facial expression recognition as well as for action units detection in case of binary classifiers were very high and comparable with the state-of-the art approaches in the area. Conversely, the results of applying Adaboost.M2 as a multi-class classifier were lower.

The advantage of using Adaboost.M2 is two fold. First, the time required for running one multi-class classifier is less that the time needed to run multiple binary classifiers. Secondly, having only one classifier allows for working with more elegant models in terms of capturing and understanding the specific characteristics of the occurrence of each facial expression.

Future work aims partly at increasing the emotion classes. Multimodal approaches imply fusion of audio and visual clues for recognition of emotions in video sequences. Such methods would valuably provide higher confidence on the recognition system's outcome based on the additional amount of data at different levels of abstraction.

# Chapter 6

# Emotion extraction from speech[1]

## 6.1 Introduction

The quality of the interaction between human beings and computers greatly improves by using methods to automatically perceive and generate the user feedback based on human non-verbal communication. As the recent developments on speech driven technologies have led to even more reliable human computer systems, there has been an increasing interest in studying more sophisticated techniques for estimating the emotion state of the speaker.

Although the primary support of the voice is to communicate, voice can be also seen as an indicator of the psychological and physiological state of the speaker. Prosodic elements transmit essential information with regard to the speaker's attitude, emotion, intention, context, gender, age and physical condition.

One question the researchers have tried to answer in their way to study the expressive speech is how does speech endowed with a particular emotion compare with neutral speech. The approach is valid under the assumption that the neutral speech does exist and is detectable. Collecting such neutral speech data is formally established by using listening panels, by the researchers deciding on the neutral property of the signal or by taking data samples from well-defined neutral contexts. As an example, newscasts are assumed to provide no obvious additional emotion content.

The emotion may be defined in terms of acoustic correlates. According to Murray and Arnott [136], the emotion in voice may be represented in terms of voice quality, utterance timing and utterance pitch contour. The pitch and the intensity of the signal are typically used as basic features for studying the emotions in speech. Other distinctive parameters generally used for emotion analysis are: the speech rate, formants, vocal-tract cross section areas, mel-frequency cepstral coefficients and features based on the Teager energy operator.

The emotion classification task is not trivial as, on one hand, an indicator may exhibit the same behaviour for more than one class of emotions and so may not offer sufficient power of discrimination and, one the other hand too many indicators may negatively influence the classification. In this context, the attempt

[1] This chapter is an extended version of the paper Datcu et al.[42].

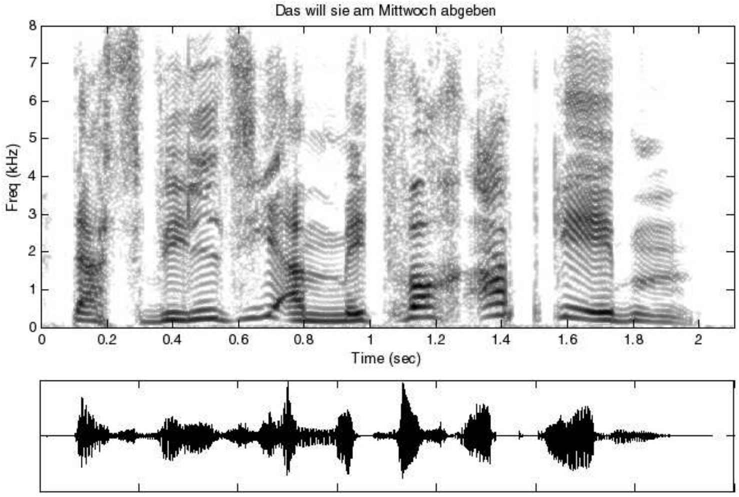


Figure 6.1: The spectrogram of one speech sample from Berlin database, corresponding to the pronunciation of the sentence ”Das will sie am Mittwoch abgeben” (”She will hand it in on Wednesday”) by a male with angry voice. Below the spectrogram is the time-domain speech signal.

to model the user’s emotions by examining sets of pertinent acoustic cues in the process of reliably discriminating between emotion classes, has become a relevant research topic. Changes in the emotional state of the speaker propagate as variations in the quality of voice due to the change in the number and size of resonant peaks i.e. the amplitude and bandwidth of formants.

By using a spectrogram, it is possible to analyse how the speech spectrum changes over time. Figure 6.1 shows an example of spectrogram computed from the audio signal. One important issue for the recognition of emotions in speech represents the segmentation of the speech signal. This process dramatically affects the subsequent results of the recognition of emotions.

The data for research on emotion analysis characteristically contains spontaneous, acted or elicited speech. From these, one would preferably choose corpora related to spontaneous speech because these optimally exhibit natural and authentic emotions of the speakers. A problem for most of the studies on emotion from speech is the impossibility to get spontaneous emotion speech data in a controlled manner. In absence of spontaneous speech, acted speech from amateur or professional actors is assumed to provide a consistent substitute to natural emotion. Samples based on elicited speech are considered to have properties that are closer to the properties exhibited by real-life emotion-enriched speech. One argument for justifying the acceptance of elicited emotion speech data is that even in real life, people produce emotion expressions with a certain level of voluntary control. The researchers interested in emotion speech analysis have constantly created collections of emotion speech data. An up-to-date record of available emotion speech corpora has been published by Ververidis et

al. [186] in 2006.
The recognition of emotions from speech involves applying a set of data processing methods on the input speech signal.
The input requires a specific form, depending on the type of emotion analysis system. The most important aspect of the input is the unit data of processing. For instance, for real-time continuous systems, the input typically consists of sliding analysis windows that show a certain degree of overlap. The sliding window can be of a fixed or variable size and is used for determining the audio data points that are further used for extracting useful emotion indicators. The choice of the type and size of the analysis window turns to be a very important aspect with implications to the performance of the emotion recognition algorithm. The size of the analysis window must be carefully adjusted so as to cope with the influence of factors such as intonation, rhythm or degree of stress.
According to linguistics, the long-time features are referred to as supra-segmental or prosodic in contrast to phonetic which are speech features at the phoneme level. The emotion is encoded as a combination of features at the supra-segmental, segmental and intra-segmental level of abstraction. Apart from a few attempts to classify emotions dynamically, current approaches mostly use static feature vectors derived on a turn, word and chunk level. A standard choice is the whole turn, which is the full section of speech by one person in a conversation. This is motivated through the fact that during a turn a change of emotion seems to occur commonly seldom enough. Alternatively, specific sub-turn entities are also known to offer high effectiveness for providing relevant features, depending on the type of application. For instance, the work [100] shows that syllable-based feature vector sequences extracted at the chunk level perform much better than the frame-based feature vector sequences.
In the current research, a different approach is taken. The analysis window relates to speaker's full turn. In addition, the relevant features are identified by segmenting the full utterance into non-overlapping frames and by testing the power of the features from each frame individually. Figure 6.2 shows the diagram of components we use in our emotion recognition from speech system. Given the set of prosodic features, we determine the segmentation type and the utterance frame structure that leads to good recognition of emotions. We model the emotion characteristics of speech using Gentle AdaBoost and hidden Markov Model - HMM methods. The optimal classifiers are determined by employing receiver-operating-curves - ROCs graphs which show the trade-off between the hit rate and the false positive rate. Finally, the overall recognition results are analysed for different types of audio signal segmentation on two different data sets.

## 6.2 Related work

Recently, the recognition of emotions in speech has been extensively researched and various methods have been used. In 1993, Murray and Arnott [136] presented an up to date review of notable research works in the area of emotion analysis in speech. It was known that the vocal expression of emotion is influenced by numerous factors but there was a considerable argument over the proportion on the contribution of each of them to the perceived emotion. Following the aggregation of various research findings, it turned that the pitch envelope i.e. the level, range, shape and timing of the pitch contour, is the most important parameter in differentiating between the basic emotions. Secondly, the voice quality seemed to be the most important factor to distinguish between the secondary emotions. It has been also brought into discussion the cross-cultural aspect of basic emotions and the difficulties pertaining to the semantics of emotion terms.

Dellaert et al. [50] present a new method for extracting prosodic features from speech, based on a smoothing spline approximation of the pitch contour. According to the results, it seems that this type of features contains sufficient information to classify the utterances according to their emotion content with results that are comparable with human performance on the same task. The selection of significant features is realized by studying the quality of each feature individually and in combinations. In the second set of speaker dependent experiments and four basic emotions, the authors demonstrate that majority voting of subspace specialists which are KNN classifiers with the value of k selected using cross validation, proves to have a definite and large performance benefit over ordinary feature selection methods.

The work [36] investigates the hypothesis that variations of features in the prosodic domain when compared to a reference point which corresponds to a well-controlled, neutral state, do reflect clear emotive patterns. Based on modelling four prototypic emotion classes together with the neutral state, the findings validate the proposed hypothesis. Additionally, some evidence is found that the prosodic domain under study contains also some variation patterns of emotions which are commonly associated with stylistic and dialect aspect of speech.

Beside rhythm and intonation, voice quality is identified as being a fundamental factor for inflicting expression of emotions. Zetterholm [213] carries a study which involves six different voice qualities such as model, breathy, creaky, harsh, tense and the compound breathy and tense voice. According to the author, anxiety seems to be correlated with a breathy voice quality. Similarly, indifference and sadness are associated with a creaky voice quality.

The method presented in [11] attempts the classification of five emotion classes by computing a fuzzy membership index given a reference point for each emotion class. Compared to the previous works in the area which relate mostly to qualitative emotion analysis, the algorithm allows for measuring the degree of each emotion as an index on some arbitrary scale.

In [109], the authors use principal component analysis to study the importance of individual features in representing emotion categories. Three methods for extracting features from short-time analysis frames as well as from entire utterances separately and taken together, are combined with three classification methods. The authors show that, when compared with vector quantization and artificial neural networks, the Gaussian mixture density mapping with both short-term and long-term features gives the best results for emotion recognition.

Huber et al. [96] focus on the detection of anger or frustration and on the differentiation between anger and neutral emotion, in the context of realistic dialogue scenario. In order to get a more realistic data, the recordings have been done according to the Wizard of Oz scenario (WoZ). The result of the experiments which involved multi layer perceptrons (MLPs) as classifiers and models at the word level and at the sentence level, indicated the possibility to create systems for automatic classification of anger versus non-anger. Later, the authors improve the models using methods which are specific to the dialogue management, such as the detection of repetitions and reformulations, the value of swear words and dialogue acts as well as mimic.

Phonetic feature and prosodic features can be optimally combined so as to serve as input for an emotion recognition system. The work of Nicholson et al. [137] uses the pitch and the speech power as prosodic features and Linear Prediction Coding (LPC) parameters as phonetic features for a speaker and context independent system for emotion recognition in speech using neural networks. An architecture implies that the network is composed of eight back propagation sub-neural networks, with one network for each of the eight emotions that are examined. The output value of each sub-network represents the likelihood of the utterance corresponding to the associated emotion category. The model is compared with two other models consisting of a network that models all the emotion categories simultaneously and of a single-layer neural network that use Learning Vector Quantization (LVQ). The authors show that, as result to open and closed experiments and using gender-based and speaker independent models, the neural network architecture that models all the emotion classes, show the best generalization performance.

Chateau et al. [29] present a study of perception, analysis and modelling of styles or the 'emotional quality' of speech. The speech emotion quality is evaluated in terms of the emotion content that describes the listener's global impression as elicited by the audition. Specific subjective criteria for evaluating the emotion quality are used to generate perceptive portraits of speech. The evaluation is carried by using linear models to connect the perceptive portraits to physical data derived from signal analysis.

The work [155] shows an ample study on the recognition of emotions from affect bursts in German speech data. The term affect bursts has been introduced by Scherer in 1994 and is defined as brief and clearly identifiable events of nonverbal expressions of affect. The results suggest that affect bursts, presented without context, can convey a clearly discernible emotional meaning. Moreover, ten distinct emotion categories can be reliably distinguished. The investigation extends to the influence of the segmental structure on emotion recognition, as opposed to prosody and voice quality. The study then takes into account the position of affect bursts that are relatively universal and show strong inter-individual differences, to the affect emblems that are culture-dependent and show only small individual differences.

The work by Aina et al. [8] takes into account joy, anger and sorrow as fundamental emotions for the recognition models. The models are evaluated using 10-fold cross validation and classifiers such as support vector machines, neural networks and ensemble of neural networks and decision trees, at the phoneme level, voiced segment and at word level. Per-phone experiments indicated that features extracted from vowels and diphthongs, reflect emotions better. The word-based models outperform phoneme-based models. In the same way, features computed at the word level seem to have higher relevance than features computed over voiced utterances, in the context of recognizing emotions. Secondly, support vector machines show the best emotion recognition results followed by neural networks that outperform decision trees in case of features extracted from word segments. Another outcome of the study shows that the performance of the best automatic model is comparable to the performance of the human listeners in the case that there is no access to linguistic content of the utterances.

Kwon et al. [107] provide a comparison on the emotion recognition performance of various classifiers. They obtained SVM and HMM based classifiers with significantly better results on SUSAS database from the previous approaches.

While statistic features, such as the mean and range of F0, seem to be associated with the arousal dimension of emotion, temporal features prove to be more relevant to the transmission of valence, attitude and intention [100]. The work presents a scheme for speech emotion classification that allows for the combination of statistic features and temporal features. According to the scheme, GMM and HMM are initially used for modelling the two types of features respectively. Then, the GMM likelihoods and HMM likelihoods of the speech signals of each class are used as features for a classifier based on weighted Bayesian Classifier and on one-hidden layer MLP. The model is trained and tested on a dataset which contains 200 samples for each of five basic emotions and the neutral emotion state.

For example, Yu et al. [208] applied a multilevel structure based on coupled hidden Markov models to estimate engagement levels in continuous natural speech. The continuous speech signal is segmented into spoken utterances and the acoustic features are computed from each utterance segment. The extracted non-linguistic information is used for predicting the emotional states such as discrete emotion types or arousal/valence levels by employing SVM-based classifiers. The HMM uses the previous information to model the user's emotional state and engagement in conversation as a dynamic, continuous process.

Recent researches of Rothkrantz et al. [152] [151] focus on studying the effect of the workload on speech production by making use of a psychological experimental setup. A full analysis on each acoustic feature is conducted in order to create efficient models for stress detection.

Some works have been focused on using additional information regarding speech. The paper of Lee et al. [108] uses three sources of information - acoustic, lexical and discourse - for recognizing emotions from speech. Linear discriminant and k-nearest neighbourhood classifiers are used to classify acoustical information to anger and frustration - as negative emotions and to neutral or positive emotions. The different features are extracted by using certain portions of the signal.

Features at acoustic and linguistic levels each offers different types of clues for the emotion classification process. Batliner et al. [17] show the results of emotion recognition from speech models on realistic, spontaneous speech data. The approach considers several classification models and several types of features extracted from the turn level and from the word level. It proves that fusing input features of different types gives better results than combining partial results of the models using the features separately.

Schuller et al. [157] present methods for emotion recognition from speech at turn-level and from sub-turn entities. Beside the annotation-based syllable chunking, the authors introduce an automatic chunking algorithm based on a time-synchronous one-pass Viterbi-beam search and on a token passing process with direct context free grammar. In both cases, the recognition is realized using support vector machines with linear kernel and one-against-one oriented multi-class discrimination. The classification results at the chunk and syllable levels are mapped on the speech turn levels using different voting algorithms. The models are evaluated on the EmoDB database with leave-one-speaker-out scheme in the case of speaker independent tests and with stratified-cross-validation in case of speaker dependent tests.

Vlasenko et al. [191] conduct experiments with hidden Markov models using one-from-all and one-against-other strategies. They also employ the search for the optimal values for the number of HMM states and the number of Gaussian mixture components. The results relate to 83% recognition rate for seven emotion classes on Berlin Emotional Speech database.

Using two different types of models, at the turn level and at the word level, Vlasenko et al. [189] demonstrate the dependence of emotion recognition on the spoken phonetic content for both, acted and spontaneous emotions. The experiments are realized using data samples from two databases, EmoDB and Speech Under Simulated and Actual Stress (SUSAS).

Danisman et al. [38] propose an approach for emotion classification of speech utterances based on ensemble of support vector machine classifiers. The input consists of the fusion at the feature level of MFCC, total energy and F0 parameters. Binary support vector machine models are trained on balanced datasets obtained by equally selecting negative samples from each emotion category. The model output corresponds to the cumulative addition of the predictions of all support vector machine models. For testing the models, the authors create Emotional Finding Nemo (EFN) database for which they use four basic emotion and neutral labels. In addition, the models are tested on publicly available DES and Berlin Emotional Speech data sets.

## 6.3 Model

Figure 6.2 shows the components used for the recognition of emotions in a speech signal. The process basically implies looking for specific indicators in the speech signal and using these elements as input for a classification process. Before being actually used for analysis, the speech signal undergoes a set of operations.

The speech signal is first segmented based on speaker's turn. Subsequently, each audio data segment is split into several data frames. As we intend to identify which parts of utterances carry the most informative clues for recognizing emotions, we consider all the data frame combinations. The frame configuration indicates which frames are selected in each utterance for extracting the acoustic features. Figure 6.3 illustrates the frame-oriented feature selection process. For each combination, we extract acoustic features from the selected frames. The result consists in sets of prosodic feature values that represent the original data. These data sets are further used in parametric classification of emotions from speech. The best models are then found from the best-performing emotion classifiers.

### 6.3.1 Data sets

The first data set used for emotion analysis from speech is Berlin [23], a database of German emotion speech. The database contains utterances of both male and female speakers, two sentences. The emotions were simulated by ten native German actors (five female and five male subjects). The result consists of ten utterances which correspond to five short and five long sentences. The length

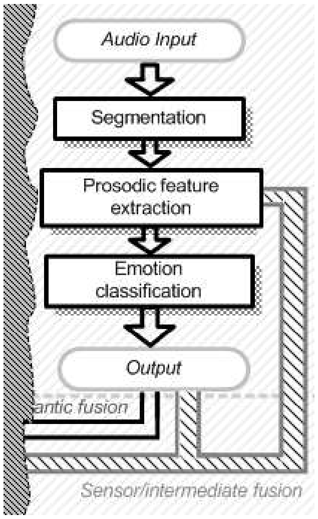


Figure 6.2: Diagram with the principal components of a emotion recognition from speech system.

of the utterance samples ranges from 1.2255 seconds to 8.9782 seconds. The recording sampling frequency is 16kHz. The final speech data set contains the utterances for which the associated emotional class was recognized by at least 80% of the listeners. Following a speech sample selection, an initial data set was generated comprising 456 samples and six basic emotions (anger: 127 samples, boredom: 81 samples, disgust: 46 samples, anxiety/fear: 69 samples, happiness: 71 samples and sadness: 62 samples).

Subsequently, the Danish emotional speech database - DES [60] is used for comparison. This database contains recordings of four basic emotion categories (surprise, happiness, sadness, anger) plus the neutral state. In order to get emotion enriched speech signals, two male and two female actors were recorded. During the recording session, each of the four actors had to speak several utterances once for each of the five emotions. The utterances involved 2 single words, 9 sentences and 2 passages of fluent speech. In addition, there are 8 passages and 10 sentences for target voices. The data set for analysis has 279 samples out of which 46 are for neutral, 46 samples represent anger, 48 represent happy, 46 for sadness, 48 for surprise and 45 represent targeted. The results from a listening test done on 20 subjects with the average age of 38 years, ranging from 18 to 59 years, indicated that the emotions were correctly identified in 67% of the cases, ranging from 55% to 80%. The emotions were correctly estimated from the audio signal by female listeners in 69% of the cases and by male listeners in 66% of cases.

In the case of dynamic modelling of emotions, we use the Enterface05 multimodal emotion database [126]. For the experiments, we extract and process

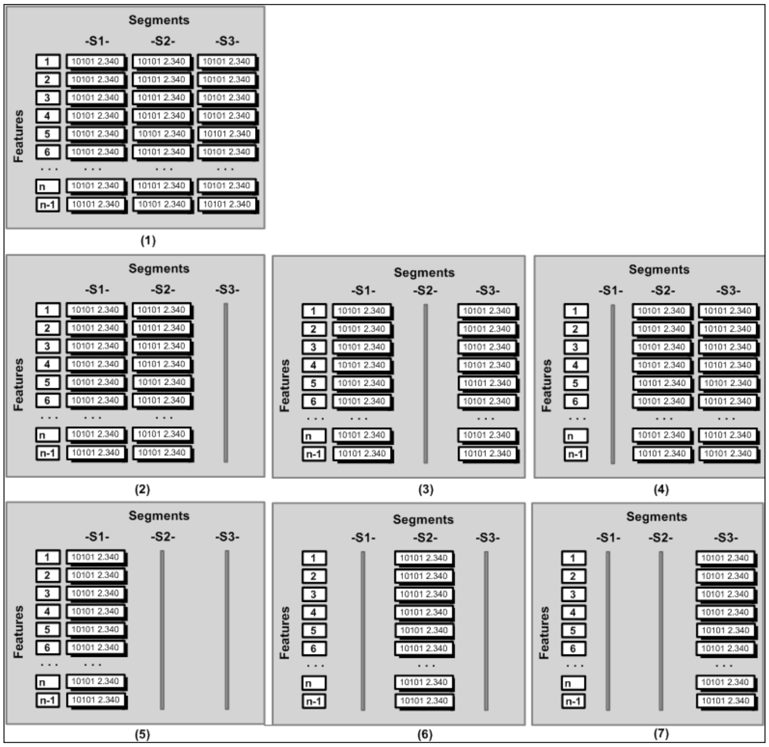


Figure 6.3: Databases obtained by using three frames per utterance segmentation.

Table 6.1: The utterance segmentation and the number of resulting data sets.

| **Nr.of frames per utterance** | **1** | **2** | **3** | **5** | **10** | **Total** |
|---|---|---|---|---|---|---|
| Nr. of data sets | 1 | 3 | 7 | 31 | 1023 | 1065 |

only the audio part of the data set. In total, the database consists of 1293 video samples out of which 216 instances are of class anger, 216 are of class disgust, 213 are of class happiness, 216 are of class surprise, 216 are of class sadness and 216 are of class fear. The database includes audio recordings in English from 44 subjects.

### 6.3.2 Multi-frame analysis

The analysis is handled separately for a different number of frames per utterance. In the current approach there are five types of frame segmentations performed on the non-silence parts of the initial audio data. The segmentation implies splitting the utterance into audio frames of equal sizes. Each type of splitting produces a number of data sets, according to all the frame combinations in one utterance. In total, there are about 1065 data sets to be considered (table 6.1).

### 6.3.3 Feature extraction

The selection of prosodic features for multi-class emotion recognition is handled in the form of a binary classification problem. Firstly, the problem is approached using 'one-against-the rest' paradigm. According to this, the classification problem implies the discrimination of one class of emotion from the rest of classes by using all the class-dependent features.

Praat tool [21] was used for extracting features from all speech samples. According to each segmentation type, the parameters: mean, standard deviation, minimum and maximum of the following acoustic features were computed: fundamental frequency (pitch), intensity, F1, F2, F3, F4 and bandwidth.

In the current case, we use GentleBoost method to iteratively identify class-dependent features and to discriminate between emotion classes. Figure 6.4 shows the feature selection process applied on frame-based segmented data. In the end, the process leads to the same number of feature subsets as the number of classes of emotion states to be classified.

## 6.4 Results

### 6.4.1 Static modelling of emotion recognition

The first investigation we conduct uses GentleBoost classifiers with frame-based segmentation on two different databases. The models based on GentleBoost are learned in maximum 200 training stages.

We evaluate the performance of each classifier using 5-fold cross validation (for Berlin data set) and with 2-fold cross validation (for DES data set) methods. Depending on the number of sub-frames per speech frame, the different data sets are used to generate sets of classifiers.

The ROC graphic in figure 6.5 shows the trade-off between the hit and the

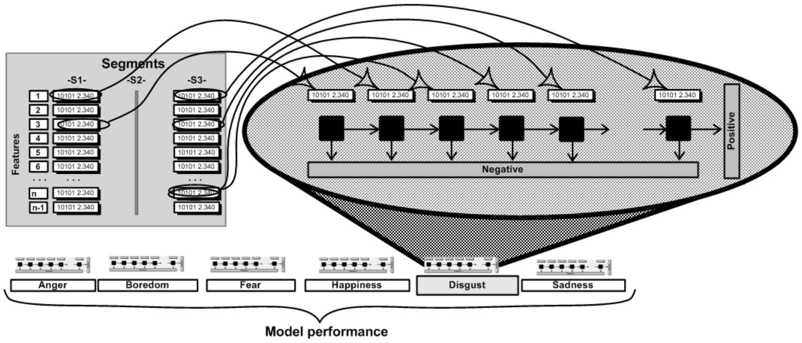
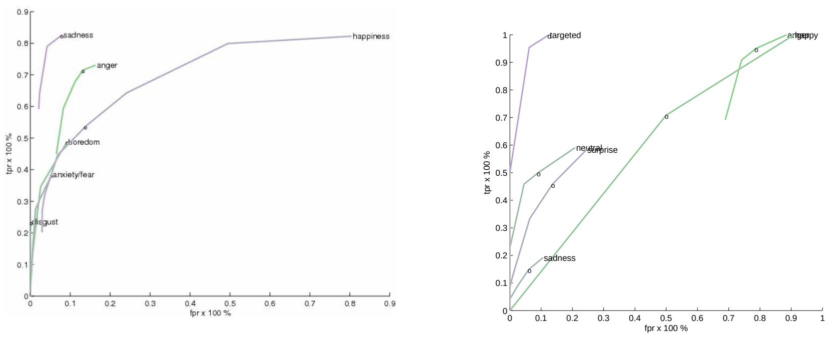


Figure 6.4: Example of six emotions classification model based on GentleBoost one-against-the rest binary classifiers.

false-positive rates for all the GentleBoost classifiers generated from Berlin and DES data sets. One curve on the graph stands for the set of representative GentleBoost classifiers generated by using the specific data set, associated with a certain frame configuration. Each point on the figure stands for the performance of a GentleBoost classifier that has been selected using the highest true-positive rate criterion. For each emotion class, a total number of 200 points is taken into account and only the ones with the highest scores are displayed on the same emotion curve.
Analysing each emotion curve separately, the final classifier to be chosen, also called the strong committee, is the one that is the closest to the north-west corner of the figure. In other words, the classifier in question is the one that has the highest true positive rate - tpr while the false positive rate - fpr is the lowest in the set of classifiers on the same curve. Table 6.2 (for Berlin data set) and table 6.3 (for DES data set) depict the characteristics of each classifier that is selected for each emotion curve separately. The column nr.stages shows the number of stages required to train the associated strong committee. An additional field - ac in each table, shows the accuracy rate achieved by the classifiers. Each classifier is identified by the structure of the frames into the utterance sample (column frames). A digit from one binary sequence specifies that the correspondent frame contributes ('1') or not ('0') with features at the classification process. An observation on the tables proves that the majority of the strong classifiers lying on the emotion curves in the ROC graph clearly express the efficiency of using a ten frames per utterance configuration for segmentation.

Due to differences on the emotion classes for Berlin and DES data sets, it is rather hard to compare on the performances achieved in the analysis. However, there are three common emotion classes: anger, happiness and sadness. The overall results indicate the higher performance of classifiers trained on Berlin data set over the classifiers trained on DES data set. This can be mainly explained by the bigger size of the training set in the case of Berlin data set.
Although the true positive rate is the same for emotion class anger, the accuracy of the best committee trained on Berlin data set is considerably higher (83%) compared to 44% for the best classifier trained on DES data set. For the same



(a) Berlin data set.

(b) DES data set.

Figure 6.5: ROC graphs showing the classifiers with the highest true positive rates for each emotion class.

Table 6.2: The optimal committees for each emotion class, Berlin data set.

| **emotion** | **nf** | **frames** | **nr.stages** | **ac(%)** | **tpr(%)** | **fpr(%)** |
|---|---|---|---|---|---|---|
| anger | 10 | 1101000001 | 5 | 0.83±0.03 | 0.72±0.16 | 0.13±0.06 |
| boredom | 2 | 10 | 58 | 0.84±0.07 | 0.49±0.18 | 0.09±0.09 |
| disgust | 10 | 0100001000 | 21 | 0.92±0.05 | 0.24±0.43 | 0.00±0.00 |
| anxiety/fear | 10 | 1110000011 | 86 | 0.87±0.03 | 0.38±0.15 | 0.05±0.04 |
| happiness | 10 | 1111010100 | 40 | 0.81±0.06 | 0.54±0.41 | 0.14±0.13 |
| sadness | 10 | 1011111101 | 13 | 0.91±0.05 | 0.83±0.06 | 0.08±0.06 |

emotion class, the training size is almost three times bigger (127 samples) in the case of Berlin data set than for DES data set (46 samples). The classifiers selected for emotion class happiness have similar performance with 71 training samples in the case of Berlin data set and 48 training samples for DES data set.

Table 6.4 (for Berlin data set) and table 6.5 (for DES data set) show the influence of the number of frames per utterance on the general recognition results. For each choice of number of frames, the best classifier is determined against the highest true-positive rate criterion. The results presented are independent on the emotion class and so represent a good criterion for comparison.

Although the true positive rate tend to be higher for classifiers trained on DES data set, the accuracy rate is still low compared to the accuracy of classifiers trained on Berlin data set. This is associated with the higher false positive rate in the case of DES data and also to the higher classification stability in the case of Berlin data set.

Table 6.3: The optimal committees for each emotion class, DES data set.

| **emotion** | **nf** | **frames** | **nr.stages** | **ac(%)** | **tpr(%)** | **fpr(%)** |
|---|---|---|---|---|---|---|
| anger | 10 | 0000100000 | 11 | 0.44±0.08 | 0.72±0.05 | 0.62±0.08 |
| happy | 5 | 01000 | 24 | 0.80±0.01 | 0.48±0.15 | 0.13±0.04 |
| neutral | 10 | 0001111000 | 19 | 0.83±0.01 | 0.46±0.28 | 0.09±0.05 |
| sadness | 10 | 0000011000 | 6 | 0.78±0.05 | 0.35±0.18 | 0.13±0.12 |
| surprise | 10 | 0011011100 | 5 | 0.75±0.11 | 0.40±0.56 | 0.18±0.25 |
| targeted | 10 | 1110100110 | 129 | 0.97±0.01 | 0.95±0.07 | 0.03±0.02 |

Table 6.4: The dependency of emotion recognition on the number of frames per utterance for Berlin data set.

| nf | ac(%) | tpr(%) | fpr(%) |
|---|---|---|---|
| 1 | 0.85±0.11 | 0.36±0.63 | 0.07±0.17 |
| 2 | 0.83±0.31 | 0.44±0.67 | 0.10±0.41 |
| 3 | 0.84±0.17 | 0.46±0.62 | 0.09±0.23 |
| 5 | 0.84±0.13 | 0.50±0.63 | 0.10±0.22 |
| 10 | 0.77±0.33 | 0.58±0.64 | 0.20±0.45 |

Table 6.5: The dependency of emotion recognition on the number of frames per utterance for DES data set.

| nf | ac(%) | tpr(%) | fpr(%) |
|---|---|---|---|
| 1 | 0.61±0.67 | 0.56±1.12 | 0.37±1.01 |
| 2 | 0.61±0.67 | 0.56±1.12 | 0.37±1.01 |
| 3 | 0.63±0.56 | 0.56±1.05 | 0.35±0.87 |
| 5 | 0.63±0.49 | 0.67±0.90 | 0.37±0.74 |
| 10 | 0.60±0.55 | 0.70±1.10 | 0.42±0.86 |

One difference should be noted on the analysis methods used for choosing the best classifiers from the tables. While for the first the criterion was to choose the classifiers with the best trade-off between hit rate and false positive rate, the last involved the choice for the classifiers with the highest true positive rate. The observation that 10 frames per utterance configuration is optimal, as obtained from the tables 6.2 and 6.3, can be traced in the true positive rates from tables 6.4 and 6.5.

### 6.4.2 Dynamic modelling with hidden Markov models

Apart from using boosting methods, we have also investigated dynamic models for emotion recognition. A six-emotion classes classifier is constructed by building a set of six different hidden Markov models - HMMs, one for each emotion category. We extract audio features in the form of mel-frequency cepstral coefficients - MFCCs using HTK toolkit [207]. Each HMM model is trained separately, using instances from Enterface05 database. The data features correspond to observations that are modelled with Gaussian mixtures associated to HMM states. Figure 6.6 shows three HMM models used for analysis. They differ in their number of states. The first HMM model includes two states of onset and offset. The second model extends the first HMM type with the apex state. The last model adds one more state that corresponds to the neutral emo-

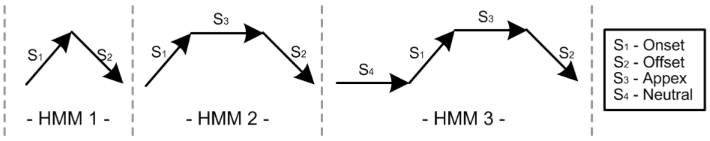


Figure 6.6: Three types of Hidden Markov Models. The number of states indicates the number of emotion transitions which are modelled.

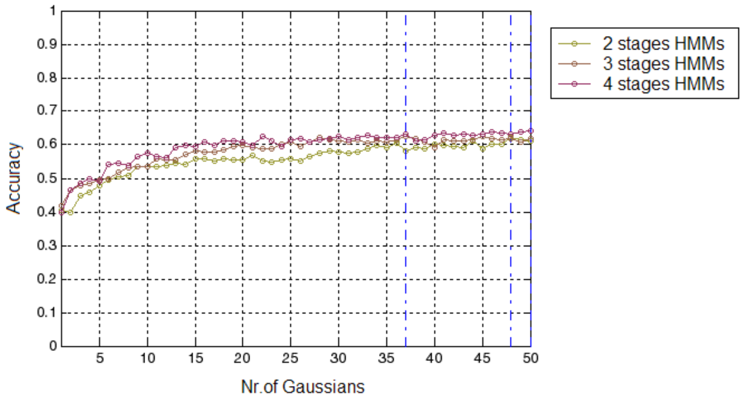


Figure 6.7: The accuracy of HMM-based classifiers with 2, 3 and 4 states and Gaussian mixtures of different number of components. The best performing HMM classifier has accuracy of 64.27%. The results are obtained using leave-one-speaker-out cross validation.

Table 6.6: The confusion matrix of the HMM that has 4 states and 50 Gaussian mixtures; the accuracy of the six emotion classes classifier is 64.27%.

| | Anger | Disgust | Fear | Happy | Surprise | Sadness |
|---|---|---|---|---|---|---|
| Anger | **85.19** | 3.24 | 0.93 | 2.31 | 5.56 | 2.78 |
| Disgust | 10.65 | **54.63** | 4.17 | 9.26 | 7.87 | 13.43 |
| Fear | 18.52 | 12.96 | **37.50** | 6.02 | 10.65 | 14.35 |
| Happy | 8.45 | 7.98 | 3.76 | **67.61** | 5.63 | 6.57 |
| Surprise | 11.11 | 5.09 | 6.02 | 5.09 | **61.11** | 11.57 |
| Sadness | 3.70 | 5.56 | 3.70 | 0.93 | 6.48 | **79.63** |

tion category. Figure 6.7 shows the accuracy results in the case of HMM-based classifiers which have two, three and four states and Gaussian mixtures of 1, 2, 3, ..., 50 components. The evaluation method was leave-one-speaker-out cross validation. The best 2-states HMM classifier has 48 Gaussian components and achieves the accuracy 62.26%. Similarly, the best 3-states HMM classifier includes 37 Gaussian components and has 62.34% accuracy. Finally, the most reliable recognizer for six emotion classes, is represented by the 4-states HMM classifier which contains mixtures of 50 Gaussian components and has the accuracy 64.27%. The confusion matrix of this classifier is presented in table 6.6. Figure 6.8 shows examples of the best HMM models which use 2, 3 respectively 4 HMM states. The result we obtained with 4-states HMM is better than the result achieved by Paleari and Huet [143] (less than 35%) and the results of Mansoorizadeh and Charkari [123] (53%) for emotion recognition from speech, using static modelling on Enterface05 database.

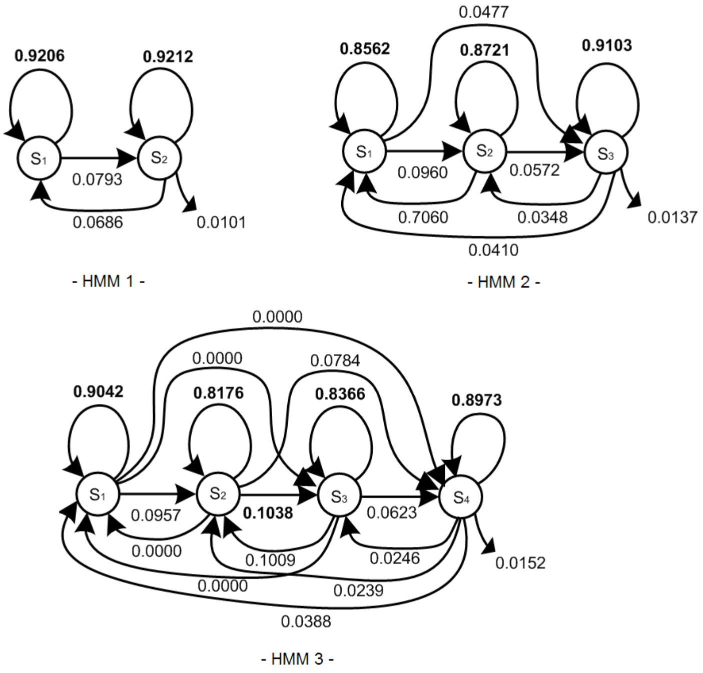


Figure 6.8: Graphical representation of HMM examples with 2, 3 and 4 states. Each example corresponds to one classifier in the set of 44 classifiers of the cross validation method.

## 6.5 Conclusion

In the current chapter, we have conducted a set of analysis on different types of utterance segmentation. As a base technique, we used GentleBoost classifiers with maximum 200 training stages. The optimal strong classifier has been selected by making use of ROC graphs. Secondly, we investigated dynamic modelling of emotion recognition with hidden Markov models. The results have been eventually commented for better understanding the underlying characteristics regarding each emotion category. We advocate for studying the effect of multi-frame speech segmentation as a primary step for the recognition of emotions.

# Chapter 7

# Bimodal data fusion for emotion recognition

## 7.1 Introduction

Within the last couple of years, automatic multimodal recognition of human emotions has gained a considerable interest from the research community. From the technical point of view, the challenge is, in part, supported also by the successes that have been noticed in the development of methods for automatic recognition of emotion from separate modalities. By taking into account more sources of information, the multimodal approaches allow for more reliable estimation of the human emotions. They increase the confidence of the results and decrease the level of ambiguity with respect to the emotions among the separate communication channels.

This chapter provides a thorough description of a bimodal emotion recognition system that uses face and speech analysis. Basically, we use hidden Markov models - HMMs to learn and to describe the temporal dynamics of the emotion clues in the visual and acoustic channels. This approach provides a powerful method enabling to fuse the data we extract from separate modalities.

The complexity of the emotion recognition using multiple modalities is higher than the complexity of the unimodal methods. Some causes for that relate to the asynchronous character of the emotion patterns and the ambiguity and the correlation which possibly occur in the different informational channels. For instance, speaking while expressing emotions implies that the mouth shape is influenced by the pronounced phoneme and by the emotion state. In this case, the use of the regular algorithms for facial expression recognition we have presented in the previous chapters, shows limited performance and reliability. In order to apply fusion, the model differentiates the silence video segments and the segments that show the subject speaking. Figure 7.1 depicts an example of the speech-silence based segmentation. The models we build in this chapter run emotion analysis on the data segments which embody activity in both visual and audio channels. In the following section, we present algorithms and results achieved in some recent and relevant research works in the field of multimodal emotion recognition. Then, we describe our new system. We present the details of all steps involved in the analysis, from the preparation of the multimodal

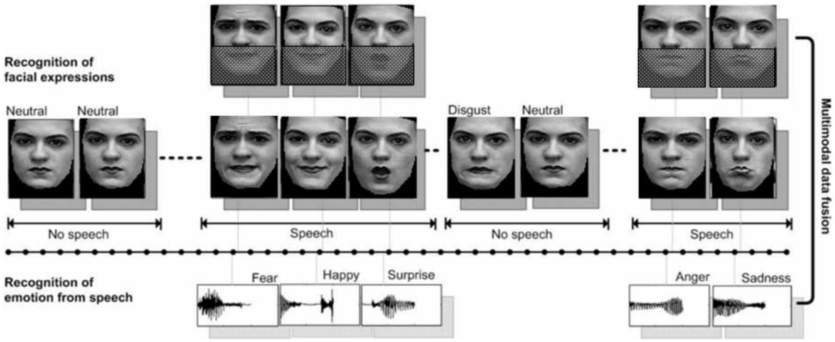


Figure 7.1: Fusion model using speech and silence audio segments.

database and the feature extraction to the classification of six prototypic emotions. Apart from working with unimodal recognizers, we conduct experiments on both early fusion and decision level fusion of visual and audio features.
The novelty of our approach consists of the dynamic modelling of emotions using hidden Markov models in combination with local binary patterns (LBPs) [140] as visual features and mel-frequency cepstral coefficients (MFCCs) as audio features. In the same time, we propose a new method for visual feature selection based on the multi-class Adaboost.M2 classifier. A cross database method is employed to identify the set of most relevant features from a unimodal database and to proceed with applying it in the context of the multimodal setup.
We report on the results we have achieved so far for the discussed models. The last part of the chapter relates to conclusions and discussions on the possible ways to continue the research on the topic of multimodal emotion recognition.

## 7.2 Related work

A noticeable approach for approaching the recognition of emotion represents the multimodal analysis. The multimodal integration of speech and face analysis can be done by taking into account features at different levels of abstraction. Depending on that, the integration takes the form of fusion at the low, intermediate or high levels. The low-level fusion is also called early fusion or fusion at the signal level and the high-level fusion is also called semantic, late fusion or fusion at the decision level. Several researchers have recently tackled these types of integration.
Han et al. [87] propose a method for bimodal recognition of four emotion categories plus the neutral state, based on hierarchical SVM classifiers. Binary SVM classifiers make use of fusion of low-level features to determine the dominant modality that, in turn, leads to the estimation of emotion labels. The video processing implies the use of skin colour segmentation for face detection and optical density and edge detection for face feature localization. The algorithm extracts twelve geometrical features based on the location of specific key points on the face area. In case of speech analysis, twelve feature values are

computed using the contours of pitch and the energy from the audio signal. On a database of 140 video instances, the authors report an improvement of 5% compared to the performance of the facial expression recognition and an improvement of 13%, compared to the result of the emotion recognition from speech.

Wimmer et al. [197] study early feature fusion models based on statistically analysing multivariate time-series for combining the processing of video based and audio based low-level descriptors - LLDs. Paleari and Huet [143] research the multimodal recognition of emotions on Enterface 2005 database. They use mel-frequency cepstral coefficients - MFCC and linear predictive coding - LPC for emotion recognition and optical flow for facial expression recognition together with support vector machines and neural network classifiers. The recognition rate of emotion classification is less than 35% for speech-oriented analysis and less than 30% in case of face-oriented analysis. Though, combining the two modalities leads to an improvement of 5% in case of fusion at the decision level and to almost 40% recognition rate in case of early fusion.

Another study on the Enterface 2005 database is presented by Mansoorizadeh and Charkari [123]. They apply principal components analysis - PCA to reduce the size of the audio and visual feature vectors and binary support vector machines - SVM for the bimodal person-dependent classification of basic emotions. Depending on the type of fusion, the inputs of the SVM models contain either separate or combined audio-visual feature vectors. For speech analysis, the features relate to the energy, the pitch contour, the first 4 formants, their bandwidth, and 12 MFCC components of the audio signal. For face analysis, the features represent geometric features that are computed based on a set of specific key points on the face area. The likelihood results of the binary SVM classifiers are used in a rule based system to determine the emotion labels of the video instances. The authors report 53% the classification rate for emotion recognition from speech, 36.00% for facial expression recognition, 52.00% for feature level fusion and 57.00% for decision level fusion.

The work of Hoch et al. [91] presents an algorithm for bimodal emotion recognition in automotive environment. The fusion of results from unimodal acoustic and visual emotion recognizers is realized at abstract decision level. For the analysis, the authors used a database of 840 audiovisual samples that contain recordings from seven different speakers showing three emotions. By using a fusion model based on a weighted linear combination, the performance gain becomes nearly 4% compared to the results in the case of unimodal emotion recognition.

Song et al. [166] present emotion recognition based on active appearance models - AAM for facial feature tracking. The facial animation parameters - FAPs are extracted from video data and are used together with low level audio features as input for a HMM to classify the human emotions.

Paleari and Lisetti [144] present a multimodal fusion framework for emotion recognition that relies on MAUI - Multimodal Affective User Interface. The approach is based on the Scherer's theory Component Process Theory (CPT) for the definition of the user model and to simulate the agent emotion generation.

Sebe et al. [158] propose a Bayesian network topology for recognizing emotions from audio and facial expressions. The database they used includes recordings of 38 subjects who show 11 classes of affects. According to the authors, the achieved performance results pointed to around 90% for bimodal classification of emotions from speech and facial expressions compared to 56% for the face-only classifier and about 45% for the prosody-only classifier. Zeng et al. [210] conducted a series of experiments related to the multimodal recognition of spontaneous emotions in a realistic setup for adult attachment interview. They use Facial Action Coding System - FACS [57] to label the emotion samples. Their bimodal fusion model combines facial texture and prosody in a framework of Adaboost multi-stream hidden Markov model - AdaMHMM.

Joo et al. [101] investigate the use of S-type membership functions for creating bimodal fusion models for the recognition of five emotions from speech signal and facial expressions. The achieved recognition rate of the fusion model was 70.4% whereas the performance of the audio-based analysis was 63% and the performance of the face-based analysis was 53.4%.

Caridakis et al. [26] describe a multi-cue, dynamic approach in naturalistic video sequences using recurrent neural networks. The approach differs from the existing works at the time, in the way that the expressibility of the user is modelled using a dimensional representation of activation and valence instead of the prototypic emotions. The facial expressions are modelled in terms of geometric features from MPEG-4 facial animation parameters - FAPs, and are computed using the location of 19 key points on the face image. Combining FAPs and audio features related to pitch and rhythm leads to the multimodal recognition rate of 79%, as opposed to facial expression recognition rate of 67% and emotion from speech detection rate of 73%.

The work of Meng et al. [132] presents a speech-emotion recognizer that works in combination with an automatic speech recognition system. The algorithm uses hidden Markov model as a classifier. The features considered for the experiments consisted of 39 MFCCs plus pitch, intensity, 3 formants and some of their statistical derivatives.

An emotion recognition study on a language independent database has been done in [194]. The authors extract MFCC and formant frequency features from the speech signal and Gabor wavelet features from the face images. The classification of six emotions uses neural networks and Fisher's linear discriminant analysis - FLDA. The results indicate the higher efficiency of using the audio signal with 66.43% recognition rate over the visual processing with 49.29% recognition rate. The audio-visual fusion has classification rate of 70%.

Busso et al. [24] explore the properties of both unimodal and multimodal systems for emotion recognition in case of four emotion classes. In this study, the multimodal fusion is realized separately at the semantic level and at the feature level. The overall performance of the classifier based on feature level fusion is 89.1% which is close to the performance of the semantic fusion based classifier when the product-combining criterion is used. Go et al. [78] use Z-type membership functions to compute the membership degree of each of the six emotions based on the facial expression and the speech data. The facial expression recognition algorithm uses multi-resolution analysis based on discrete wavelets. An initial gender classification is done by the pitch of the speech signal criterion. The authors report final emotion recognition results of 95% in case of male and 98.3% for female subjects. Fellenz et al. [66] use a hybrid classification procedure organized in a two-stages architecture to select and fuse the features extracted from face and speech to perform the recognition of emotions. In the first stage, a multi-layered perceptron - MLP is trained with the back propagation of error procedure. The second symbolic stage involves the use of PAC learning paradigm for Boolean functions.

## 7.3 Data set preparation

The models we are going to build for multimodal emotion recognition are based on the use of hidden Markov model classifier. In the current research context, HMM is used as a supervised machine learning technique. Based on that, the HMM training and testing processes rely on the use of fully labelled samples of audio-visual data instances. At the moment of starting this research, finding a fully annotated database turned to be difficult task to fulfil. At first, this was because of the lack of multimodal databases. Some databases had no emotion labels and were not proper for audio-visual processing. We specifically avoided using multimodal data sets that have recordings with noise and utterance overlapping in the audio signal, or with occlusion and too much rotation of the faces. The database we have eventually decided to use for our research is Enterface 2005 [126]. This database contains audio-visual recordings of 42 subjects who represent 14 different nationalities. A percentage of 81% are men, while the remaining 19% are women. At the recording time, 31% of the subjects wore glasses and 17% had beard.
The recording procedure first consisted of listening to six successive short stories, each of them eliciting a particular emotion. The emotions relate to the prototypic emotions which are: happiness, sadness, surprise, anger, disgust and fear, as identified by Ekman [57]. Then, the subjects had to read, memorize and finally utter five different reactions to each story, all by using English language. For each story, the subjects were asked to produce messages that contain only the emotion to be elicited and to show as much expressiveness as possible. The recording setup implied the use of a monochromatic dark grey panel for the image background and constant illumination. The audio-visual data was encoded using Microsoft AVI format. The image frames were stored using the image resolution of 720x576 pixels, at the frame rate of 25 frames per second. The audio samples were stored using uncompressed stereo 16-bit format at the sample rate of 48000 Hz. Figure 7.2 illustrates an example from Enterface 2005 database. We have started the data pre-processing step from the set of 1293 samples from Enterface 2005 database. In a previous chapter, we have used this data set and we have extracted the whole set of audio instances for building models for emotion extraction from the speech signal. In the context of multimodal processing, we had to first verify the appropriateness of each video sample. As a result, we have removed a subset of 463 instances. From the set of

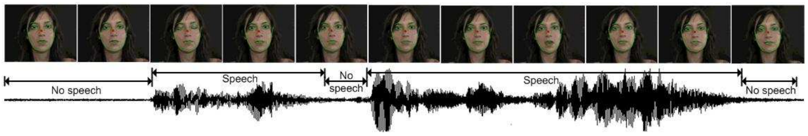


Figure 7.2: An example from Enterface 2005 database.

830 remaining samples, 135 accounted for emotion class fear, 143 for surprise, 137 for sadness, 145 for anger, 141 for disgust and 129 for happiness. This subset represents a well-balanced multimodal database of simulated emotion recordings from 30 subjects. Figure 7.3 illustrates the duration in seconds, of the utterances from the final multimodal database. Like in the case of unimodal emotion recognition, the data pre-processing step involved using specific vision-oriented methods for extracting and normalizing the actual face images from each video frame. At first, we used Viola&Jones face detection algorithm [188] and active appearance models [55] to obtain the location and the shape of the faces. Then, we have removed the unnecessary image patches and scaled down the face images to 60 pixels width by 80 pixels height. Here, unnecessary image patches relate to the visible parts of background, subject's hair and cloth. The data preparation procedure is explained in chapter 3.

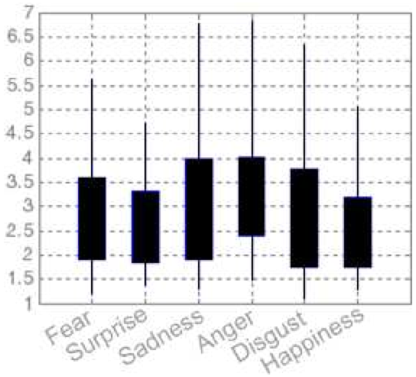


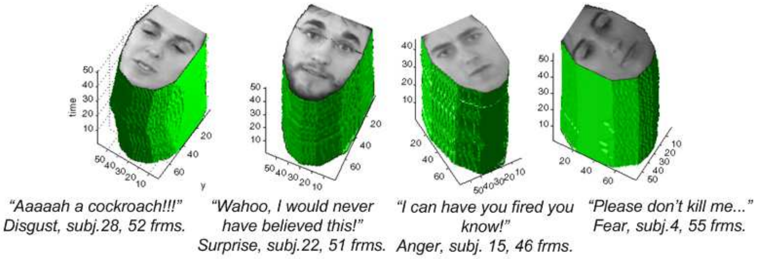
Figure 7.3: Utterance duration (in seconds) for each emotion class.

For aligning the faces, we used the reference key point located at the middle of the line segment delimited by the inner corners of the eyes. Figure 7.4 illustrates the result of applying the previously described methods on four video samples containing faces.

## 7.4 Architecture

### 7.4.1 Classification model

HMM represents a statistical method for modelling data that can be characterized in terms of an underlying process which generates measurable and observable sequences. The causality of the observation sequence $O = (o_1, o_2, ..., o_T)$ is interpreted through the so-called hidden states $Q = (q_1, q_2, ..., q_T)$. The observations and the states are part of a state alphabet set $S = (s_1, s_2, ..., s_N)$ and of a observation alphabet set $V = (v_1, v_2, ..., v_M)$. By definition, the HMM model has the form: $\lambda = (A, B, \pi)$. The term $A$ refers to the state transition table



"Aaaaah a cockroach!!!" Disgust, subj.28, 52 frms.

"Wahoo, I would never have believed this!" Surprise, subj.22, 51 frms.

"I can have you fired you know!" Anger, subj. 15, 46 frms.

"Please don't kill me..." Fear, subj.4, 55 frms.

Figure 7.4: 60×80 video samples containing the face area only.

that gives the time-independent probability of state $j$ following state $i$:

$$A = [a_{ij}], a_{ij} = P(q_t = s_j | q_{t-1} = s_i).$$

The term $B$ is the observation model which gives the time-independent probability of observation $k$ being generated from state $j$:

$$B = [b_i(k)], b_i(k) = P(x_t = v_k | q_t = s_i).$$

The last element of the HMM model, $\pi$ refers to the initial state probability: $\pi = [\pi_i], \pi_i = P(q_1 = s_i)$. There are two assumptions that hold for HMM. According to the first assumption, the current state only depends on the previous state. This is also called the Markov assumption: $P(q_t | q_1^{t-1}) = P(q_t | q_{t-1})$. The relation is also interpreted as the base for the memory of the HMM model. The second assumption states that the output observation at time $t$ depends only on the current state and so, it is independent on the previous states and observations: $P(o_t | o_1^{t-1}, q_1^t) = P(o_t | q_t)$.

Given the definition of HMM, it is possible to identify the parameters $A$, $B$ and $\lambda$ that best explain the observed data. It also is possible to compute the probability that the model produced the observed data. Eventually, the most probable sequence of steps for some observation data can be determined, given the model.

### Evaluation

In a multi-class classification setup, the decision is taken according to the model that gives the best prediction for an observation sequence. The probability of the observation sequence $O$ for the state sequence $Q$ is:

$$P(O | Q, \lambda) = \prod_{t=1}^{T} P(o_t | q_t, \lambda) = b_{q1}(o_1) \times b_{q2}(o_2) ... b_{qT}(o_T).$$

The probability of the state sequence is $P(Q|\lambda) = \pi_{q_1} a_{q_1 q_2} a_{q_2 q_3} ... a_{q_{T-1} q_T}$. Then, the probability of the observations can be formulated as follows:

$$\begin{aligned} P(O|\lambda) &= \sum_{Q} P(O|Q, \lambda) P(Q|\lambda) \\ &= \sum_{q_1 \dots q_T} \pi_{q_1} b_{q_1}(o_1) a_{q_1 q_2} b_{q_2}(o_2) ... a_{q_{T-1} q_T} b_{q_T}(o_T). \end{aligned}$$

The complexity of computing the probability of $O$ can be decreased by caching calculations for each state as a trellis of states at each time step. The cached value $\alpha$ is regarded as the probability of the partial observation sequence $o_1$, $o_2$, ..., $o_t$ and state $s_i$ at time $t$ and is determined as the sum over all states at the previous time step. By definition, the forward probability is: $\alpha_t(i) = P(o_1 o_2 ... o_t, q_t = s_i|\lambda)$. The probability of an observation sequence is then equal to the sum of the elements of the last column of the trellis. The trellis is computed using the forward algorithm:

**Forward algorithm**
1. Initialization: $\alpha_1(i) = \pi_i b_i(o_1)$, $1 \leq i \leq N$;
2. Induction: $\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i) a_{ij}\right] b_j(o_{t+1})$, $1 \leq t \leq T-1$, $1 \leq j \leq N$;
3. Termination: $P(O|\lambda) = \sum_{i=1}^{N} \alpha_T(i)$.

For each state $s_j$, the term $\alpha_j(t)$ gives the probability of getting to that state having observed the sequence until time $t$. With the use of cache variable $\alpha$, the complexity of the computations becomes $N^2 T$ instead of $2TN^T$. A similar approach aims at introducing a backwards variable $\beta_t(i)$ as the probability of the partial observation sequence from time $t+1$ to $T$ with initial state $s_i$: $\beta_t(i) = P(o_{t+1} o_{t+2} ... o_T | q_t = s_i|\lambda)$.

### Decoding

The sequence of hidden states which is most likely to have produced an observation sequence can be decoded using Viterbi algorithm. By definition, the probability of the most probable state sequence given the partial observation is:

$$\delta_t(i) = \max_{q_1, q_2, \dots, q_{t-1}} P(q_1 q_2 ... q_t = s_i, o_1, o_2, ..., o_t|\lambda).$$

Viterbi algorithm has the following steps:

### Learning

For the given face data set, the goal of training HMM is to determine the parameters $\lambda = (A, B, \pi)$ of the model. One solution for that is to first divide the training observation vectors equally amongst the states of the model and to compute the mean and variance of each state. Then, finding the maximum likelihood state sequence with Viterbi algorithm allows us to reassign the observation vectors to states and to recompute the mean and variance of each state. The process can be repeated until the mean and variance estimates do not change. The re-estimation procedure is done using Baum-Welch re-estimation formulae.

```
**Viterbi algorithm**
1. Initialization: $\delta_1(i) = \pi_i b_i(o_1)$, $1 \leq i \leq N$, $\psi_1(i) = 0$;
2. Recursion:
$\delta_t(j) = \max_{1\leq i\leq N} [\delta_{t-1}(i)a_{ij}]\, b_j(o_t)$
$\psi_t(j) = arg \max_{1\leq i\leq N} [\delta_{t-1}(i)a_{ij}]$
,where $2 \leq t \leq T, 1 \leq j \leq N$
3. Termination:
$P^* = \max_{1\leq i\leq N} [\delta_T(i)]$
$q_T^* = arg \max_{1\leq i\leq N} [\delta_T(i)]$
4. Backtracking the most probable state sequence:
$q_t^* = \psi_{t+1}(q_{t+1}^*)$, $t = T-1, T-2, ..., 1$.
```

For the means and covariances of a HMM, the re-estimation formulae have the following form:

$$\begin{aligned}
\hat{\mu}_j &= \frac{\sum_{t=1}^{T} L_j(t)o_t}{\sum_{t=1}^{T} L_j(t)} \\
\hat{\Sigma}_j &= \frac{\sum_{t=1}^{T} L_j(t)(o_t - \mu_j)(o_t - \mu_j)'}{\sum_{t=1}^{T} L_j(t)}.
\end{aligned}$$

The term $L_j(t)$ represents the probability of being in state $j$ at time $t$. The use of this term in the re-estimation formulae comes from the fact that the full likelihood of an observation sequence includes the summation of all possible state sequences. In this way, each vector $o_t$ contributes to the computation of the maximum likelihood parameters for each state $j$. By using $L_j(t)$, each observation is assigned to every state as opposed to assigning each observation vector to a specific state. The values for $L_j(t)$ can be computed by using the so-called Forward-Backward algorithm, as follows:

$$\begin{aligned}
L_j(t) &= P(q_t = j|O, \lambda) \\
&= \frac{P(O, q_t = j|\lambda)}{P(O|\lambda)} \\
&= \frac{1}{P(O|\lambda)}\alpha_j(t)\beta_j(t).
\end{aligned}$$

The estimation of the transition probabilities is done in the same way, using slightly more complex formulae.

### Testing

Testing the classifiers involves the use of distinct HMM models for every class of emotions. Let these models be designated as $l_i$ and $i = 1, ..., C$, where $C$ equals the number of emotion categories. A given observation sequence $O$ is evaluated by computing the probabilities of individual models $P(l_i|O)$. Applying the Bayes rule and assuming equal a priori probability of each model $P(l_i)$, the pattern classification can be performed by maximizing the likelihood function $P(O|l_i)$ as discussed above.

All the unimodal and multimodal experiments with hidden Markov models we have conducted in this research and which are presented in the following parts of the chapter, are based on the use of HTK toolkit for HMMs [207] developed by Microsoft Corporation and Cambridge University Engineering Department.

### 7.4.2 Emotion estimation from speech

The assessment of the emotion levels from speech can be naturally done by identifying patterns in the audio data and by using them in a classification setup. The features we extract are the energy component and 12 mel-frequency cepstral coefficients together with their delta and the acceleration terms. These features are computed from 25 ms audio frames, with 10 ms frame periodicity from a filter bank of 26 channels. A Hamming window is used on every audio frame during the application of Fourier transform. The feature extraction procedure determines the conversion of the original audio sampling rate of 48kHz to the MFCC frame rate of 100Hz. Each MFCC frame contains 39 terms, as indicated previously.

The recognition of emotions is realized using the HMM algorithm. Each emotion has associated one distinct HMM and the set of HMMs forms a multi-class classifier. For evaluation, we use 3-fold cross validation. The samples from the same subject are part of either the training set or the test set. This restriction assumes that the testing is done on instances of subjects other than those of the subjects included in the training data set. The method is supposed to give a better estimation of the performance of the classifiers. For finding the best HMM model, we conduct experiments in which we investigate the optimal values for the HMM parameters. In this way, we build and test models which use 2,3 and 4 HMM states (figure 6.6). The 2 state HMMs encode the emotion onset and offset. The 3 state HMMs encode the emotion onset, apex and offset. The models with 4 states encode the neutral state and emotion onset, apex and offset states.

For each state configuration, we build distinct models of HMMs with Gaussian mixtures with different number of components (1..50 components). The results of testing all the models, are illustrated in figure 7.5. Following the evaluation, it results that the most efficient configuration is to use 4 states and 40 Gaussians per mixture and that the accuracy of this classifier is 55.90%. Table 7.1 presents the confusion matrix of this classification model.

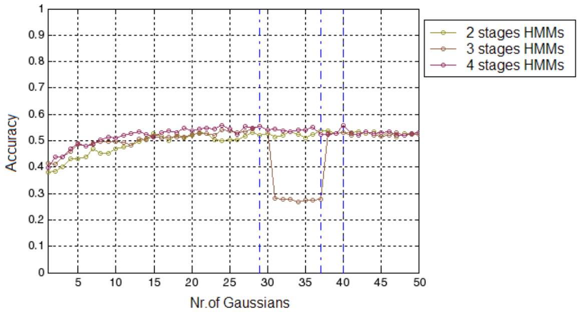


Figure 7.5: The accuracy of HMM-based classifiers of emotions in speech signal. The number of states is 2, 3, 4 and the number of Gaussians varies from 1 to 50. Three fold cross validation method is used for performance estimation.

Table 7.1: The confusion matrix of the HMM that has 4 states and 40 Gaussian components; the accuracy of the emotion recognition from speech model is 55.90% for six basic emotion categories.

| | Fear | Surprise | Sadness | Anger | Disgust | Happy |
|---|---|---|---|---|---|---|
| Fear | **91.72** | 2.07 | 0.69 | 1.38 | 2.76 | 1.38 |
| Surprise | 24.11 | **44.68** | 9.22 | 11.35 | 4.26 | 6.38 |
| Sadness | 25.19 | 14.81 | **41.48** | 6.67 | 5.19 | 6.67 |
| Anger | 19.38 | 18.60 | 3.88 | **48.06** | 6.98 | 3.10 |
| Disgust | 23.78 | 9.09 | 8.39 | 8.39 | **38.46** | 11.89 |
| Happy | 5.84 | 5.84 | 10.22 | 2.19 | 6.57 | **69.34** |

### 7.4.3 Video analysis

The goal of the video analysis is to build models to dynamically process the video data and to generate labels according to the six basic emotion classes. The input data is represented by video sequences that can have different number of frames, as determined by the utterance-based segmentation method. The limits of each video data segment are identified by using the information obtained during the analysis of the audio signal. Based on the set of frames, a feature extraction is applied for preparing the input to the actual classifier. HMM models are then employed to classify the input sequences in terms of the emotion classes.

One problem that has to be taken into account while developing the facial expression recognizers is that both the input set of features and the classifier models should be chosen in such a way so as to handle the time-dependent variability of the face appearance. More specifically, some of the inner dynamics of the face are generated due to the effect of the speech process that is present in the data. Taking into account the aforementioned issues, the focus of the research is to study the selection of most relevant visual features and to use the values of these features as data observations for the HMM-based classifiers.

#### Database preparation

The first problem of feature selection is tackled by conducting a separate research using a second database namely the Cohn-Kanade [103]. In this context, the multi-class classification method Adaboost.M2 is used as a feature selection algorithm. The procedure is based on the primary property of the Adaboost.M2 to identify the most important features while running the training phase of the classification process. We use the same set of prototypic emotions as for the main study on the Enterface05 dataset.

The first problem is to make a proper data set of representative face image samples. For this, we use the results of the study on the Cohn-Kanade action units based facial expression labelling, which are presented in chapter 3. The basic set of non-ambiguous facial expression samples from the Cohn-Kanade database includes 251 instances. Each instance corresponds to the last frame of the video sequence and represents the face at the apex of one facial expression. As it can be seen in table 3.8 and table 7.2, the structure of this set indicates a rather unbalanced data.

While emotion class happiness accounts for 99 samples, there are only 7 instances for each of the emotion classes surprise and disgust. To obtain a balanced dataset, we adopt the solution of adding new instances to the undersized classes and to remove data instances from the oversized classes.

Table 7.2: The structure of the non-ambiguous set of 251 samples in the Cohn-Kanade database.

| | Nr. Non-ambiguous samples | | | | | | | Nr. Mixed emotion samples | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Distance** | **0** | **1** | **2** | **3** | **4** | **5** | total | **0** | **1** | **2** | **3** | **4** | **5** | total |
| Fear | 2 | 8 | 13 | 6 | 6 | 3 | **38** | 0 | 0 | 0 | 0 | 0 | 0 | **0** |
| Surprise | 0 | 6 | 1 | 0 | 0 | 0 | **7** | 0 | 0 | 0 | 0 | 0 | 0 | **0** |
| Sadness | 0 | 4 | 23 | 13 | 7 | 0 | **47** | 0 | 0 | 0 | 0 | 0 | 0 | **0** |
| Anger | 0 | 4 | 12 | 12 | 18 | 7 | **53** | 0 | 0 | 0 | 0 | 0 | 0 | **0** |
| Disgust | 0 | 0 | 6 | 1 | 0 | 0 | **7** | 0 | 0 | 0 | 0 | 0 | 0 | **0** |
| Happiness | 18 | 55 | 22 | 4 | 0 | 0 | **99** | 0 | 0 | 0 | 0 | 0 | 0 | **0** |

Table 7.3: The structure of the balanced set of 303 samples selected from the Cohn-Kanade database.

| | Nr. Non-ambiguous samples | | | | | | | Nr. Mixed emotion samples | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Distance** | **0** | **1** | **2** | **3** | **4** | **5** | total | **0** | **1** | **2** | **3** | **4** | **5** | total |
| Fear | 2 | 10 | 17 | 11 | 7 | 3 | **50** | 0 | 0 | 0 | 0 | 0 | 0 | **0** |
| Surprise | 0 | 24 | 4 | 0 | 0 | 0 | **28** | 0 | 10 | 12 | 0 | 0 | 0 | **22** |
| Sadness | 0 | 5 | 24 | 14 | 7 | 0 | **50** | 0 | 0 | 0 | 0 | 0 | 0 | **0** |
| Anger | 0 | 4 | 12 | 12 | 18 | 7 | **53** | 0 | 0 | 0 | 0 | 0 | 0 | **0** |
| Disgust | 0 | 0 | 24 | 4 | 0 | 0 | **28** | 0 | 4 | 18 | 0 | 0 | 0 | **22** |
| Happiness | 17 | 21 | 10 | 2 | 0 | 0 | **50** | 0 | 0 | 0 | 0 | 0 | 0 | **0** |

In case of emotion class fear, we have added 12 image samples by selecting one frame which precedes the apex in 12 video sequences of emotion fear, from the non-ambiguous set of 251 samples. The action unit based patterns of 2 selected instances have distance 1, 4 instances have distance 2, 5 instances have distance 3 and one has distance 4 to the action unit pattern of class fear.

For class surprise we add 21 non-ambiguous new face images by selecting three frames preceding the frame showing the facial expression apex in 7 different videos. The selected frames are not consecutive and show considerable degree of facial expression onset. Furthermore, 22 new instances are added to the set of surprise class from the sample subset of mixed emotions. The action unit based patterns of 10 of these instances have distance 1 and the rest have distance 2 to the action unit pattern of class surprise. The size of emotion class sadness is increased by adding 3 instances as preceding frames to the apex of 3 video sequences. They have AU pattern distances 1, 2 and 3 respectively.

The emotion class anger has neither received new samples nor it has undergone sample removal.

For the emotion class disgust, we add 21 non-ambiguous instances from 7 videos and 22 mixed emotion face instances each being related to the apex in the sequence. Similarly to the case of emotion class surprise, the non-ambiguous samples precede the sample associated to the apex frame and are not consecutive. From each video sequence, we have selected 3 such instances. For the samples of mixed emotions, we selected 4 instances with action unit patterns at distance 1 and 18 samples with action unit patterns at distance 2 to the action unit pattern of emotion disgust. In the case of the last emotion class, happiness, we have removed 49 image samples from the initial non-ambiguous set. The structure of the resulting balanced database is depicted in table 7.3.

### Visual feature selection

Because the sets of the visual features we derive from the face samples, are too large to be used directly as observations in the HMM classification setup, we have to decrease their size. This can be done by transforming the original visual features to other set of more representative features. Such a method is used by Mansoorizadeh and Charkari [123] by applying PCA technique to obtain a reduced set of visual features. Another approach is to select only a limited number of relevant visual features. Wang and Guan [194] use a stepwise method based on Mahalanobis distance. Genetic algorithms also can be employed at this step [209]. Shan et al. [161] propose matrix-based canonical correlation analysis - MCCA as a method to reduce the feature set by identifying the factor pairs from mouth and eye regions, that best represent the facial expressions. The boosting methods represent a specific class of algorithms that can be successfully used to select representative features. We do the feature selection by following the same steps we have made for unimodal facial expression recognition. We use the local binary patterns - LBPs and Adaboost.M2 classifiers. The result of this part of research will be later applied for the facial expression recognition in video data of speaking subjects. Previous studies on facial expression recognition in single images, like the one of Dubuisson et al. [53], showed that different face regions produce features with different informative power for classification.
In our dynamic recognition setup, we want to also investigate the contribution of speaking mouth region and other face regions to the classification results. In addition to using the whole face image, we define two symmetric models of face regions around the face features. Figure 7.6 illustrates the face regions taken into account. Regions $R_8$, $R_9$, $R_{10}$ and $R_{11}$ are located on the mouth area and therefore are considered to be essentially influenced during the production of speech and while expressing emotions. The first face region model consists of using regions $R_1$ ... $R_7$ and the second model consists of using regions $R_1$ ... $R_{11}$.

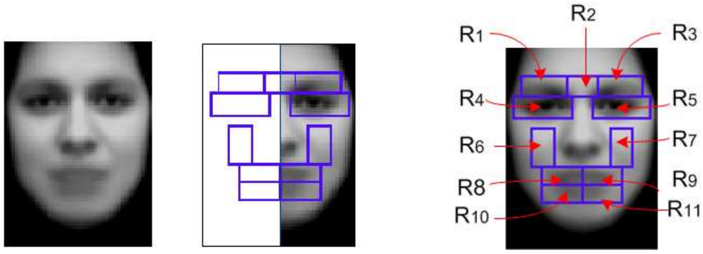


Figure 7.6: Average face sample from the balanced Cohn-Kanade database. A symmetric facial feature model is used to delimit rectangular face regions from which specific visual features are extracted.

We generated 27.226 LBP features located on the whole face image, 22.276 LBP features based on the second face region model and 14.176 LBP features

Table 7.4: Confusion matrix of the Adaboost.M2 facial expression classifier using LBP features extracted from the whole face image.

| | Fear | Surprise | Sadness | Anger | Disgust | Happy |
|---|---|---|---|---|---|---|
| Fear | **74.00** | 4.00 | 4.00 | 2.00 | 6.00 | 10.00 |
| Surprise | 2.00 | **94.00** | 2.00 | 0.00 | 2.00 | 0.00 |
| Sadness | 10.00 | 4.00 | **56.00** | 8.00 | 14.00 | 8.00 |
| Anger | 5.66 | 0.00 | 18.86 | **64.15** | 7.54 | 3.77 |
| Disgust | 8.00 | 0.00 | 24.00 | 10.00 | **56.00** | 2.00 |
| Happy | 6.00 | 0.00 | 6.00 | 6.00 | 10.00 | **72.00** |

based on the first face region model. The Adaboost.M2 classifier was then used to identify the features that provided the best facial expression recognition results. For evaluation we used 20-folds cross validation method.

Figures 7.7, 7.9 and 7.11 show the train and test mismatch rates achieved by Adaboost.M2 for each LBP-based data set. The right side of these figures illustrates the test mismatch rate for each emotion category. The minimum test mismatch rate in the case of using the whole face LBP feature set, is 30.69%. The result has been achieved for the optimal number of 30 stages of training, with the train mismatch rate of 0.42%. Using LBP features according to the first face model of 7 regions, generates a classifier with the minimum test mismatch rate of 46.20%, the train mismatch rate of 1.82% for 51 stages of training. Finally, using LBP features according to the third face model of 11 regions generates a classifier with the minimum test mismatch rate of 35.31% and the train mismatch rate of 0.26% for 40 stages of training. Tables 7.4, 7.5 and 7.6 show the confusion matrices of the three LBP-based Adaboost.M2 classifiers.

Based on the previous feature selection procedures, we find a relevant set of features for each facial expression category. Figures 7.8, 7.10 and 7.12 show graphical representations for the projection of the optimal LBP features on average faces for each facial expression category. The graphical representations account for the LBP features which are selected by Adaboost.M2 during the training of all the 20 folders of the cross validation method. For instance, for each facial expression class in figure 7.8, the projection contains 600 LBP features (30 features x 20 folders).

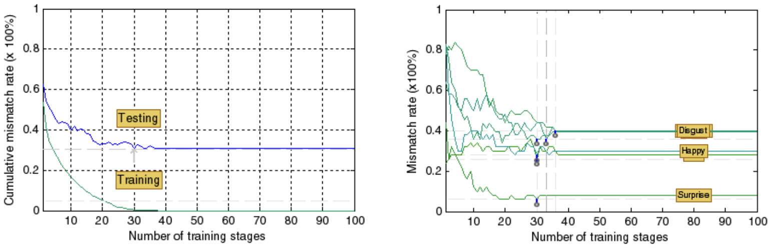


Figure 7.7: Train and test mismatch rate of Adaboost.M2 using LBPs from the whole face image.

Tables 7.7 and 7.8 show the proportion of optimal LBP features from each

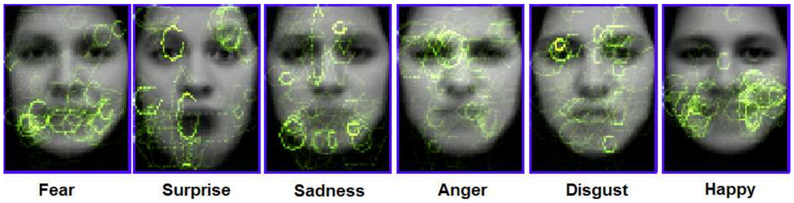


Figure 7.8: Projection of the set of 30 LBPs of the whole face model, on average face images showing the six basic emotions.

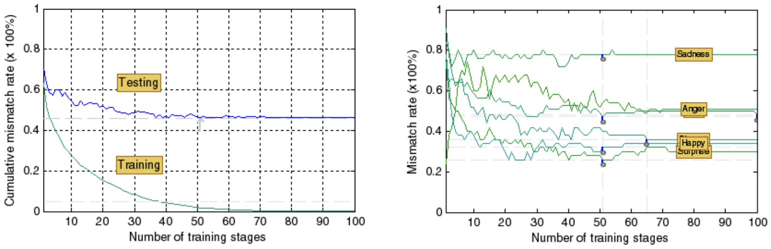


Figure 7.9: Train and test mismatch rate of Adaboost.M2 using LBPs from 7 face regions.

face region, for the model of 7 face regions and for the model of 11 face regions.

### HMM-based facial expression recognition

Making facial expression recognizers with hidden Markov models implies the identification of the optimal model parameters. Finding the best number of states, the best number of Gaussian mixture components and the best set of local binary patterns, represent a non-trivial task. We start from the results of the Adaboost.M2 classifiers. In the case of facial expression recognition using LBP features extracted from 7 face regions, we have found that the optimal number of training stages is 51. At each training stage, Adaboost.M2 selects a subset of six LBP features, one for each facial expression category. As a consequence,

Table 7.5: Confusion matrix of the Adaboost.M2 facial expression classifier using LBP features extracted from 7 face regions.

| | Fear | Surprise | Sadness | Anger | Disgust | Happy |
|---|---|---|---|---|---|---|
| Fear | **46.00** | 4.00 | 26.00 | 8.00 | 2.00 | 14.00 |
| Surprise | 12.00 | **74.00** | 6.00 | 2.00 | 6.00 | 0.00 |
| Sadness | 28.00 | 12.00 | **22.00** | 16.00 | 16.00 | 6.00 |
| Anger | 18.86 | 1.88 | 16.98 | **52.83** | 5.66 | 3.77 |
| Disgust | 8.00 | 8.00 | 10.00 | 14.00 | **60.00** | 4.00 |
| Happy | 6.00 | 0.00 | 8.00 | 4.00 | 14.00 | **68.00** |

Table 7.6: Confusion matrix of the Adaboost.M2 facial expression classifier using LBP features extracted from 11 face regions.

| | Fear | Surprise | Sadness | Anger | Disgust | Happy |
|---|---|---|---|---|---|---|
| Fear | **66.00** | 4.00 | 18.00 | 4.00 | 0.00 | 8.00 |
| Surprise | 6.00 | **80.00** | 10.00 | 0.00 | 4.00 | 0.00 |
| Sadness | 6.00 | 12.00 | **40.00** | 14.00 | 18.00 | 10.00 |
| Anger | 13.20 | 0 | 18.86 | **54.71** | 7.54 | 5.66 |
| Disgust | 2.00 | 4.00 | 20.00 | 4.00 | **68.00** | 2.00 |
| Happy | 2.00 | 0.00 | 8.00 | 6.00 | 4.00 | **80.00** |

Table 7.7: Proportion of LBP features in 7 regions selected by Adaboost.M2 classifier.

| (%) | R1 | R2 | R3 | R4 | R5 | R6 | R7 |
|---|---|---|---|---|---|---|---|
| Fear | 11.56 | 10.67 | 7.24 | 21.35 | 29.48 | 8.01 | 11.69 |
| Surprise | 9.80 | 1.22 | 23.95 | 14.69 | 35.10 | 0.95 | 14.29 |
| Sadness | 20.03 | 8.53 | 16.02 | 15.37 | 6.20 | 11.11 | 22.74 |
| Anger | 33.12 | 25.22 | 8.28 | 9.03 | 19.07 | 2.26 | 3.01 |
| Disgust | 14.32 | 6.66 | 22.74 | 19.35 | 20.85 | 7.04 | 9.05 |
| Happy | 4.92 | 4.67 | 2.90 | 3.41 | 6.06 | 30.18 | 47.85 |

Table 7.8: Proportion of LBP features in 11 regions selected by Adaboost.M2 classifier.

| (%) | R1 | R2 | R3 | R4 | R5 | R6 | R7 | R8 | R9 | R10 | R11 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Fear | 1.2 | 0.6 | 4.5 | 4.6 | 8.5 | 6.1 | 2.1 | 5.6 | 18.8 | 30.6 | 16.8 |
| Surprise | 10.4 | 1.6 | 13.6 | 7.6 | 27.2 | 0.5 | 8.6 | 11.9 | 7.8 | 6.1 | 4.2 |
| Sadness | 10.4 | 4.3 | 7.2 | 8.7 | 2.8 | 6.5 | 5.7 | 13.6 | 10.6 | 8.3 | 21.4 |
| Anger | 23.7 | 19.4 | 5.3 | 6.9 | 17.1 | 0.9 | 0.7 | 3.0 | 6.4 | 10.4 | 5.7 |
| Disgust | 8.1 | 4.8 | 19.2 | 16.8 | 10.6 | 5.3 | 2.8 | 11.8 | 11.8 | 4.6 | 3.8 |
| Happy | 2.1 | 1.7 | 2.7 | 2.3 | 4.0 | 21.0 | 41.3 | 15.4 | 6.0 | 2.4 | 0.6 |

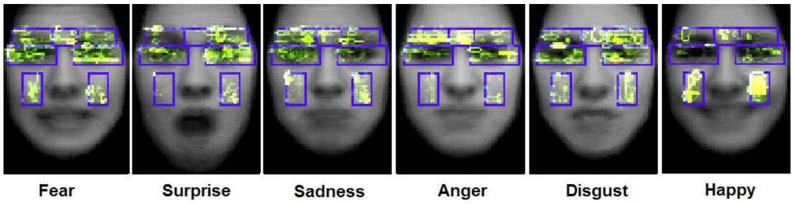


Figure 7.10: Projection of the set of 51 LBPs of the model of 7 face regions, on average face images showing the six basic emotions.

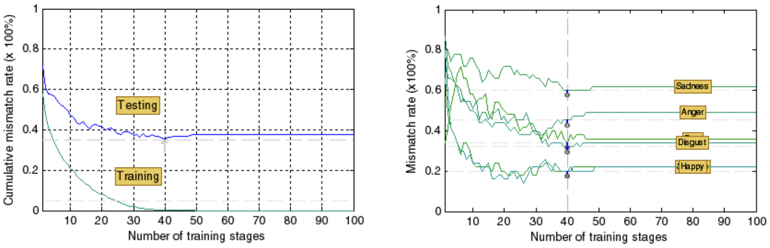


Figure 7.11: Train and test mismatch rate of Adaboost.M2 using LBPs from the 11 face regions.

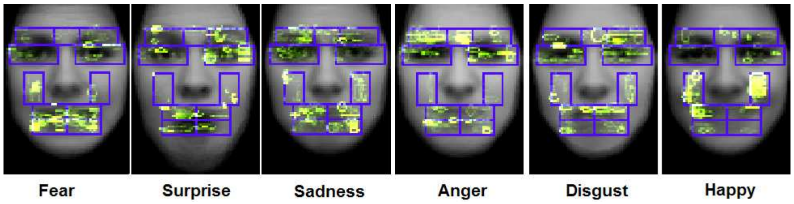
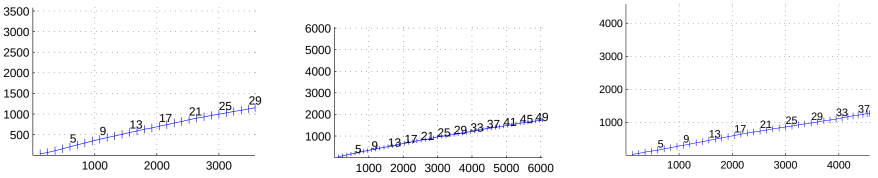


Figure 7.12: Projection of the set of 40 LBPs of the model of 11 face regions, on average face images showing the six basic emotions.

there would be 306 features which account for the six facial expressions of the final optimal classifier. Taking into account the fact that for evaluation we have used cross validation with 20 folders, it results that the final LBP feature set contains 6120 LBP features. However, the set obtained by concatenating all the subsets of the 20 folders of the cross validation method, does not include only distinct features. In fact, an important part of the subset relates to features that are commonly selected by Adaboost.M2 during training multiple folders. In addition, Adaboost.M2 may select the same feature multiple times during the training at the same the cross validation folder. This is depicted graphically in figure 7.13(b). For example, the set of features collected by taking the first 45 most important LBPs from 7 face regions, for all emotion categories, includes 5400 features, though the same set contains only 1599 distinct LBP features. We define the importance of LBP features based on the number of times an LBP is selected by the optimal Adaboost.M2 classifier during training. Figure 7.14 illustrates the accumulated percentage of importance for the set of LBPs for the whole face and for the two face region models. In the figure, the LBPs are presented in the descending order. Using the feature importance measure, we make separate data sets by gradually choosing the first most important features for all emotion classes, from the feature sets of the 20 cross validation folders.

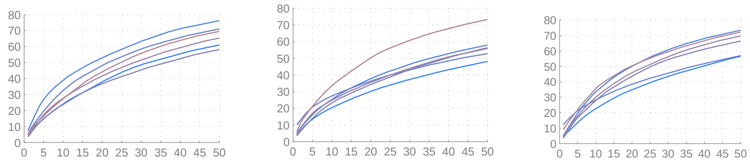


(a) Whole face.

(b) 7reg.

(c) 11reg.

Figure 7.13: The concatenated set of LBP features extracted from the whole face and two types of face regions. The x axis represents the size of the feature set; the y axis represents the number of distinct LBP features selected by Adaboost.M2.



(a) Whole face.

(b) 7reg.

(c) 11reg.

Figure 7.14: Importance of LBP features extracted from the three face region models. The features are sorted in the descending order of the selections(%) by the Adaboost.M2 classifier, for each basic emotion category.

For evaluation of facial expression recognition, we have generated HMM models for each emotion category. The training data sets have been created by taking into account the emotion label of each video sample. At the testing stage, each video instance is analysed using six HMM models, one for each emotion. Figure 7.15 shows the performance of different HMM classifiers on the data set of LBPs extracted from the whole face region and on the data sets of LBPs extracted from 7 and 11 face regions. The best facial expression recognition model uses 268 distinct features that correspond to the selection of 45 features from each facial expression category. The accuracy of this classifier is 37.71%. Table 7.9 shows the confusion matrix of the HMM classifier.

Apart from using LBP visual features, we also used features derived from optical flow estimation method. For that, we have applied the pyramidal Lucas-

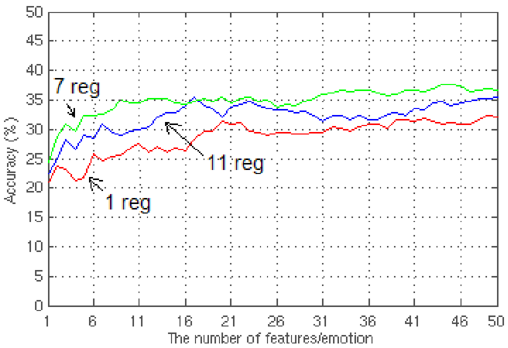


Figure 7.15: Facial expression recognition recognition results by using HMM models with different number of LBP features. The best HMM uses 45 LBP features for each facial expression category, from 7 different face regions.

Table 7.9: Confusion matrix of the best HMM facial expression classifier using LBP features.

| (%) | Anger | Disgust | Happy | Surprise | Sadness | Fear |
|---|---|---|---|---|---|---|
| Anger | **18.62** | 15.86 | 11.03 | 24.13 | 17.24 | 13.10 |
| Disgust | 9.92 | **60.28** | 10.63 | 63.82 | 6.38 | 6.38 |
| Happy | 6.20 | 17.05 | **48.06** | 18.60 | 5.42 | 4.65 |
| Surprise | 10.48 | 2.79 | 9.09 | **53.14** | 16.08 | 8.39 |
| Sadness | 17.51 | 10.94 | 10.94 | 25.54 | **19.70** | 15.32 |
| Fear | 9.62 | 16.29 | 6.66 | 26.66 | 14.07 | **26.66** |

Kanade optical flow algorithm [119] to obtain the displacement of texture pixels between consecutive video frames from all the video instances from the Enterface 2005 database. Based on the dense optical flow estimation, we used the previously described models of 7 and 11 face regions to determine the magnitude and the orientation of the aggregated motion vectors in each face region. This results to a set of 14 features in the case of using the model of 7 face regions and to a set of 22 features in the case of using the model of 11 face regions. The two feature sets represent the time-dependent characteristics of the emotions. Building 2-states HMM-based facial expression classifiers on the databases obtained by using the two sets of optical flow features, we obtained 19.51% accuracy for the model of 7 face regions and 19.63% for the model of 11 face regions. The low results are comparable to random guessing.

### 7.4.4 Fusion model

Considering the previous results of unimodal emotion estimation, it turns out that the use of audio data leads to better recognition rate (55.9%), when compared to the use of facial expression oriented models (37.71%). The next step in the attempt to get higher performance for the emotion recognition, is to combine the information from the two unimodal approaches. Depending on the type of information taken into consideration, we can define separate categories of integration. Using combined sets of audio and video features as input for the classification models is considered to fall in the category of low-level data fusion. This approach is also called early fusion or signal level fusion. Conversely, the use of final emotion estimates from unimodal face and speech analysis is defined as high level fusion. This alternative is also called late fusion or fusion at the decision level.

Prior to building models which integrate audio and video data, the first problem that regards the video segmentation must be solved. We identify the beginning and end points of audio-video data chunks based on the turn-based segmentation. The long pauses in conversation are used as indicators for identifying the edges of a segment. After we obtain audio-video segments, we proceed by removing the sub-segments that denote the lack of speech. Based on the resulting data segments, the distinct sets of audio and video features are further extracted following the same procedures as in the case of unimodal emotion recognition.

In case of the audio signal, we extract sequences of MFCC frames at the rate of 100 frames/second, each frame being sized to 39 acoustic features. As result to video processing, we extract visual feature sets at the rate of 25 feature sets/second. Extracting LBP features from the whole face image leads to sets of 331 features/set. Similarly, using LBP features from 7 face regions generates sets of 307 features/set and using LBP features from 11 face regions generates sets of 335 features/set.

From the point of view of using HMM for modelling the emotions, each feature set represents one observation. Unlike in the case of high-level fusion which runs separate classification on each modality, in the case of low-level fusion the integration implies the concatenation of the visual and acoustic feature observation vectors. Because of the difference between the 100Hz rate of MFCC frames and the 25Hz rate of video frames, a special feature formatting procedure has to be done to first synchronize the unimodal sets of features. This additional step can be done by up-scaling the observation rate of the visual feature sets to the observation rate of the audio feature sets. The second solution is to proceed with downscaling the observation rate of the audio feature sets to the observation rate of the visual feature sets. In the current approach we opt for the first solution. The process is illustrated in figure 7.16.

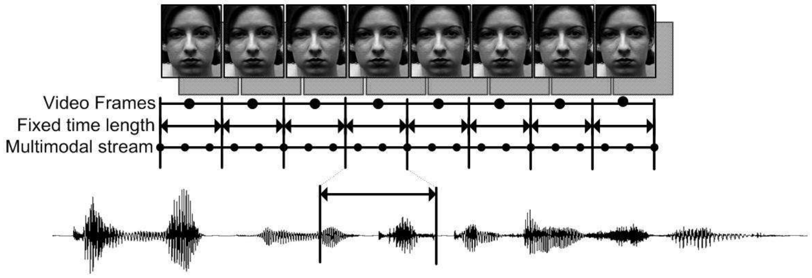


Figure 7.16: Data synchronization process that takes into account the sampling rate of different data channels.

The recognition of emotions based on low-level fusion of audio-visual data is done by using the synchronized bimodal observation vectors with HMMs. For each emotion category, we create a separate HMM and combine all the models to obtain a multi-class emotion classifier. For evaluation, we have used 3 fold-cross validation method with the additional restriction that the train and test data sets do not contain samples on the same subject, for all subjects.
The simplest model consists of HMMs with one Gaussian for each state. Combining the acoustic features and LBP features extracted from 7 face regions leads to the final model accuracy of 38.55%. Using acoustic features and LBP features extracted from 11 face regions leads to the accuracy of 39.15%. Both results are superior to the results of the recognition of emotions using visual data. Still, they are worse than the results from the emotion extraction from speech. Setting the number of HMM Gaussian components to 40 and the number of HMM states to 4 like in the case of the best speech-oriented emotion classifier and combining with LBP features from the 7 regions leads to a classifier which shows 22.18% accuracy. The recognition of emotions based on decision level fusion implies the combination of the final classification results obtained by each modality separately. For this, we take into consideration four sets of unimodal classification results namely from the speech-oriented analysis and from the separate LBP-oriented analysis which use visual features from the whole face image, from 7 face regions and from 11 face regions. We use these sets together with a weight function that allows for setting different importance levels for each set of unimodal results. This weight-based semantic fusion models the asynchronous character of the emotion in visual and auditory channels. The best model obtained in this way has the accuracy of 56.27%. Although this result reflects an improvement when compared to the emotion recognition from the unimodal approaches considered, it represents only a slight increase of performance.

## 7.5 Results

Studying the unimodal recognition of emotions on Enterface 2005 shows that the speech-oriented analysis proves to be more reliable than the facial expression analysis. The best classifier we obtained in case of using HMM models with MFCC features has the accuracy of 55.90%. Conversely, the best HMM-based facial expression recognition model we got uses LBP features and has the accuracy of 37.71%. The difference of 18.19% between the classification rates achieved on separate modalities is close to the same difference between the unimodal performances reported in [143] and [123]. However, we obtained better results than the results from these two research papers, for the emotion analysis on separate modalities. Moreover, as opposed to the work [123] which attempts the person dependent recognition of emotions, our models are completely independent of the identity of the subjects. To support this approach, we use n-fold cross validation and separate the samples of each subject in the train set from the test set.
The best facial expression recognition classifier we have obtained is based on the use of local binary patterns. The rather low results of the models based on optical flow estimation, can be explained by the limited visual representation of the feature set. Extracting feature observations from consecutive frames of the 25Hz video sequences, does not offer enough information to describe the dynam-

ics of the emotion generation process. A solution is to calculate and to derive features from the face motion flow applied over large integration windows. The fusion of audio and video features leads to results that are at best, close to the best unimodal classification result. In order to improve the fusion results, more investigations are needed.

## 7.6 Conclusion

The current chapter has proposed a method for bimodal emotion recognition using face and speech data. The advantage of such a method is that the resulting models increase the efficiency of single modality emotion analysis. We focus on the person-independent recognition of prototypic emotions from audio-visual sequences.

The novelty of our approach is in the use of hidden Markov models for the classification process. Furthermore, we introduced a new technique to select the most relevant visual features, by running a separate modelling study on a separate database of facial expressions. The HMM and Adaboost.M2 algorithms we have used for the recognition relate to multi-class classification methods. Finally, we showed that the fusion at the semantic level provides the best performance for the multimodal emotion analysis.

# Chapter 8

# Implementing an automatic emotion recognition system

## 8.1 Introduction

In the current chapter, we present the details for the implementation of a bimodal emotion recognition system that incorporates facial expression recognition and emotion extraction from speech. The multimodal emotion data fusion model works at the high, semantic level. The system performs the classification of emotions from the set of six prototypic emotions.
The implementation of the single modality data processing methods is based on the research of Datcu and Rothkrantz [44][42], which studied the recognition of emotions from faces and from speech signal.
For facial emotion expressions, the system uses Viola&Jones features and boosting techniques for face detection [188], active appearance model - AAM [55] for extracting the shape of the face and support vector machines - SVM [185] for the classification of feature patterns to the prototypic expression classes [57]. For training and testing the system, we have used the Cohn-Kanade database [103]. The system makes use of a connection to the video camera that constantly provides digitized image sequences. Additionally, a microphone is used to capture the audio signal.
The implemented models of facial expression analysis regard the processing of frontal face image samples. The process does not require manual calibration. In order to speed up the recognition of facial expressions, we adopt a method that allows to temporarily suspend the activity of the face detection module. The method uses the result of AAM from processing one video frame as an initial estimate for AAM processing the next video frame (figure 8.1). The procedure has the advantage of increasing the speed of face analysis algorithms. Conversely, the efficiency of the approach has limited efficiency as the accuracy is highly affected by sudden changes of face position and orientation. The emotion recognition from speech employs different types of segmentation of audio signal. The features used by the classifier are: the mean, standard deviation, minimum and maximum of the following acoustic features: fundamental frequency (pitch), intensity, F1, F2, F3, F4 and bandwidth. The fusion model aims at determining the most probable emotion of the subject, given the emotions determined by

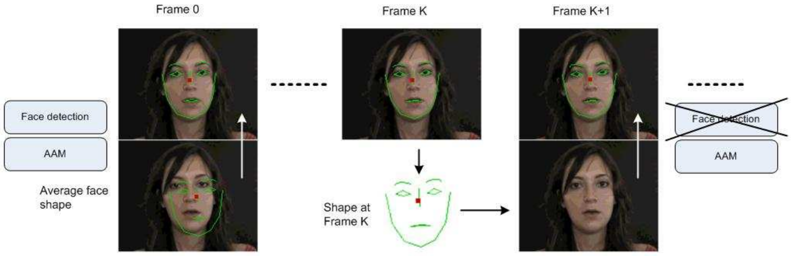


Figure 8.1: Replacement of frame-based face detection by optical flow-based tracking.

Figure 8.2: SNC-RZ25N/P Sony IP camera.

processing the previous audio-visual frames. For the latter, the system uses an integration window which contains the current and the previous $n$ frames.
Our initial system implementation was designed to use a specialized surveillance camera. The hardware characteristics of SNC-RZ25 IP camera (figure 8.2) optimally comply with the requirement for high quality video data.
The camera device has an accurate vision sensor and a built-in microphone. Among other options, it has optimized levels of sensitivity and MPEG-4 as well as JPEG compression mechanisms. The implemented system was presented at several international conferences [43][45] during demo sessions.

## 8.2 Interface to the camera

In this section, we describe the role and the integration of the components of the emotion recognition system. The boxed figure 8.3 shows the UML Use Case Diagram for SONYcam application.
The application has a simple graphical interface that consists of several control and graphical components. These interface components facilitate the access to system parameters and display results of the emotion recognition. During runtime, the user may change the value of several system parameters such as the video frame-processing rate, the frame image resolution and parameters that relate to the graphical representation of the recognition results.

In the beginning, the application identifies the optimal values of most of the system parameters, by taking into account the characteristics of the working context. For instance, if the application runs on hardware devices with limited processing capacity, the video processing is automatically adapted to lower video frame rates.
Similarly, for devices with small screen size, such as smart phones or personal digital assistants - PDAs, the interface is adapted so as to show the user only the most relevant information.
An additional control panel of SONYcam interface (figure 8.8) allows for handling the motor capabilities of SNC-RZ25N/P camera. The user can control camera's pan/tilt/zoom functions by clicking on specific interface buttons. The generated interface events are translated into commands that are later sent to the camera. Apart from making the emotion recognition system, we have also implemented special interface modules that run on different hardware devices. An example is the interface package that was developed using Java programming language and runs on Zaurus PDA. In the original system setup, the communication between the system components was made using the iROS framework [150]. Figure 8.4 shows the UML sequence diagram of the system. In that setup, all components transmit and receive messages containing data and commands, by making use of the iROS events. Every time the processing components request audio-video data from the camera, they initiate communication sessions with the main thread of the camera interface driver. A representative component is then created to subsequently handle all the command and data transfers between the two parts (figure 8.5). The data-streaming module that handles the retrieval of video frames from the camera uses the video frame rate setting indicated by the user or automatically determined by the system. In addition to the implementation of the audio-video emotion-processing component, the system interfaces and the camera driver, we have built additional software tools to monitor, control and analyze the work state of the emotion recognition system.
The role of the camera driver interface is to facilitate access of other software components to the functions of the camera device.
A distinct case is when the connection with the Sony camera is done via a wireless network. Such a situation demands for hardware devices which have two network interfaces available. The components of the software package may be

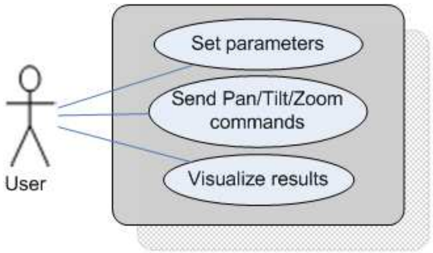


Figure 8.3: UML Use Case Diagram for SONYcam application.

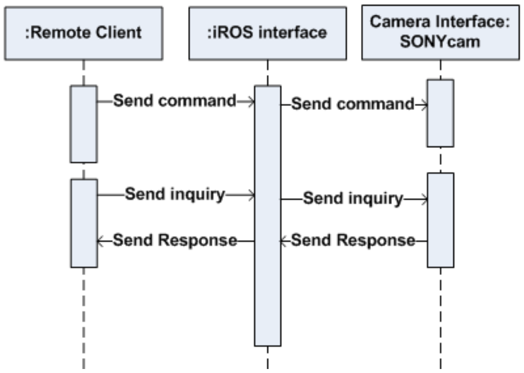


Figure 8.4: UML Sequence Diagram for SONYcam application and remote clients.

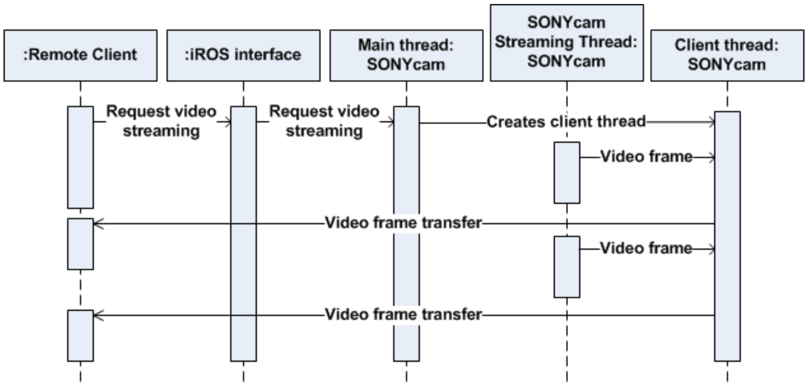


Figure 8.5: UML Sequence Diagram for SONYcam application.

installed on hardware platforms that are part of separate networks, using one or more iROS communication frameworks. Figure 8.6 shows the UML deployment diagram of the emotion recognition system. In this way, we implemented special software components. These components connect to two or more iROS data heaps that work as bridges between these networks and transfer the data packages between the existent software components. For instance, if the interface is running on Zaurus PDA device, this software interface connects the wireless network of the PDA and the Ethernet.

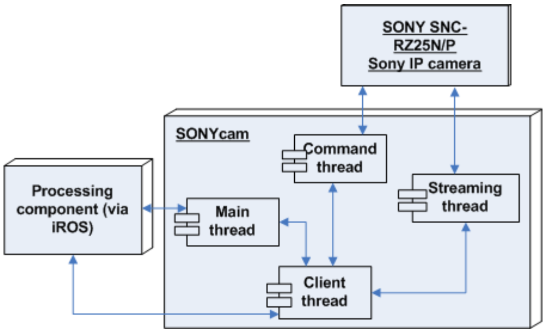


Figure 8.6: UML deployment diagram of the emotion recognition system.

Figure 8.7 shows an example of an architecture that includes both wired and wireless software components. The proprietary camera functions are made accessible through the HTTP server of the IP camera as software modules organized based on Common Gateway Interface - CGI protocol. In the next section,

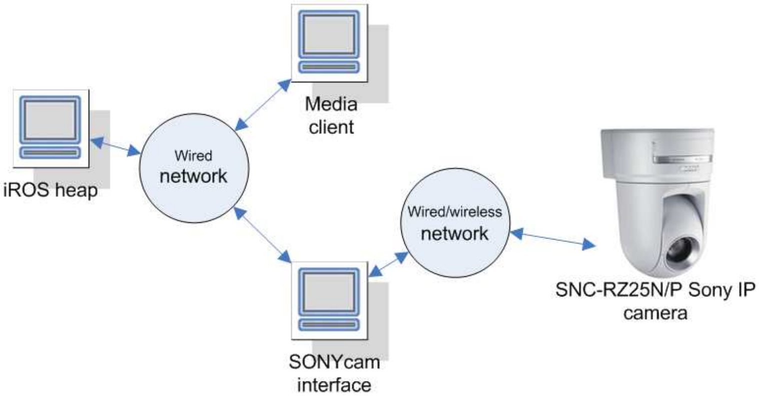


Figure 8.7: Setup for the SONYcam application running on different networks.

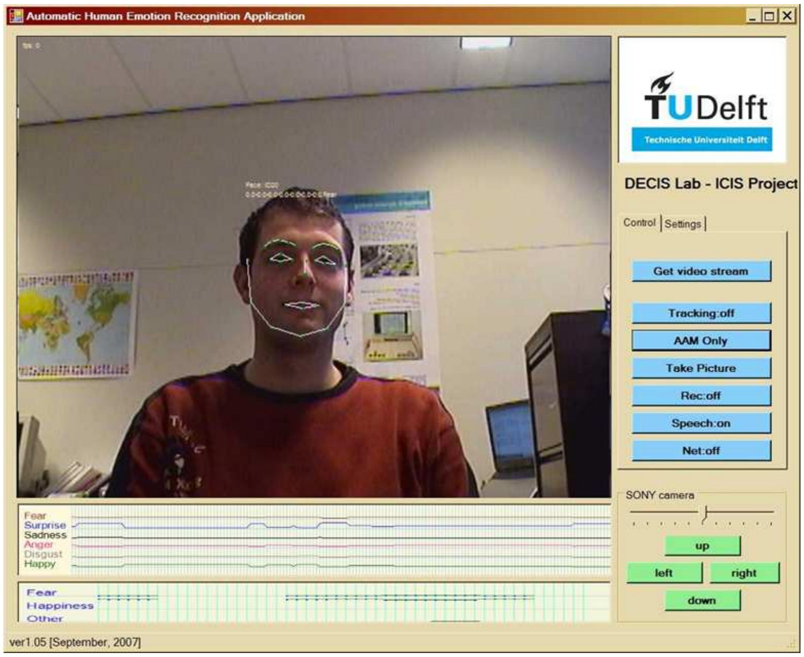


Figure 8.8: Screen view of our software implementation of the bimodal emotion recognition algorithm.

we present the software libraries and the routines for audio and video data processing.

## 8.3 Implementation

Figure 8.8 shows the screen view with a snap shot of our multimodal emotion recognition application. The system is based on the algorithms described by Datcu and Rothkrantz [45]. The unimodal data processing modules and the emotion classifier are developed to work on Windows operating system.
The face detection module is based on the implementation of Viola&Jones method from Open Source Computer Vision Library OpenCV [2] of Intel. As for the active appearance model, we have used and updated the algorithm implementation from the AAM-API library [169]. The component for speech processing uses Tcl/Tk scripts in combination with Snack Sound Toolkit [3], a public domain toolkit developed at KTH, the Royal Institute of Technology, Sweden.
From the face data, a separate component extracts geometric features and prepares them for the classification. The routines for facial feature extraction, facial expression recognition and the emotion extraction from speech are implemented using C++ programming language. For the classification component, we have used LIBSVM library [28]. The AAM module of our initial facial expression

recognition system proceeds with the detection of faces in each frame of the video sequence.

We have obtained a considerable improvement in terms of speed by nearly doubling the frame rate with an algorithm that uses information regarding the face shape of one subject at the current frame, as initial location for the AAM fitting procedure applied for processing the next frame of the video sequence.

The proposed procedure works well for a limited degree of face translation or rotation. The disadvantage of this method is the higher probability of generating faulty results for the extraction of face shape.

Figures 3.6, 3.7 and 3.9 from chapter 3 indicate that AAM model can handle face rotation up to 30° and face translation up to 15%, relative to the width of the face in the initial frame. Rotations and translations above these limits attract definitive erroneous results of the facial expression classification in the current frame and of the face analysis in the next video frames. In order to make the system robust in such cases, the system runs the detection of faces at specific time intervals.

The initial implementation was a stand-alone application for emotion recognition. Then, we integrated the software application in a human-computer interaction system for crisis management [72]. The communication with the other software modules was based on iROS middleware. Eventually, at the last phase we replaced iROS with our implementation of the multimodal framework [47] as reported in chapter 9 of this thesis.

The speed of our software implementation is about 5 fps, on a computer with Intel Core 2 CPU @2.00 GHz and 2.00 GB of RAM. At the moment when we completed the first prototype implementation of the system, only a few research groups worldwide had working systems for emotion recognition. Some of them were not entirely automatic and required calibration either before or at running time.

## 8.4 Conclusion

The development of automatic systems for emotion analysis represents long-time efforts on several related fields. The first step of such an endeavour typically consists of researching features, algorithms and models that may be used for assessing the human emotion states. As for the implementation of these research findings into working systems, different software tools may be used.

The first straightforward result represents computer-based systems that show limited robustness and functionality. In most of the cases, such implementations relate to off-line systems or systems that require the intervention of the operator to calibrate and adjust the settings to each work context.

Furthermore, the majority of the systems are able to run the analysis of emotions based only on single images. This, however, tends to gradually change, as researchers propose even better and more efficient solutions to the problem.

Based on the audio-visual processing and classification algorithms that have been described in the previous chapters, we have made an implementation of a fully automatic system, which recognizes six prototypic emotions in video sequences. The system uses audio and video data and determines the emotion by integrating unimodal emotion recognition results in the form of high-level, semantic fusion.

Finally, studying the efficiency of the implemented algorithms, we increased the speed of the system with a technique that reuses the estimation on the geometrical characteristics of the face, for the analysis of next frames in the video sequence. The software system presented in this chapter has been already successfully used as a major component of a complex human-computer interaction system. Future work will study the applicability of the result in other user-centred application domains.

# Chapter 9

# Multimodal Workbench for Automatic Surveillance Applications[1]

## 9.1 Introduction

The challenge to build reliable, robust and scalable automated surveillance systems has interested security people ever since the first human operated surveillance facilities came into operation. Moreover, since the bombings in London and Madrid in 2005, research in methods to detect potentially unsafe situations in public places has taken flight. Given the size and complexity of the sensing environment surveillance systems have to cope with, including the unpredictable behaviour of people interacting in this environment, current automated surveillance systems typically employ diverse algorithms (each focusing on specific features of the sensor data), many sensors (with overlapping sensing area and able to communicate with each other through a network), and different types of sensors (to take advantage of information only available in other modalities).

It is our belief that in modern surveillance applications, satisfactory performance will not be achieved by a single algorithm, but rather by a combination of interconnected algorithms. Noticeable developments have lately been achieved on designing automated multimodal smart processes to increase security in everyday life of people. As these developments continue, proper infrastructures and methodologies for the aggregation of various demands that inevitably arise, such as the huge amount of data and computation, become more important.

In this chapter, we present a framework for automated surveillance designed to facilitate the communication between different algorithms. This framework is centred on the shared memory paradigm, the use of which allows for loosely coupled asynchronous communication between multiple processing components. This decoupling is realized both in time and space. The shared memory in the current design of the framework takes the form of XML data spaces. This suggests a more human-modelled alternative to store, retrieve and process data. The framework enhances the data handling by using a document-centred ap-

[1] This chapter is based on the book chapter Datcu et al.[47].

proach to tuple spaces. All the data is stored in XML documents and these are subsequently received by the data consumers following specific XML queries. In addition, the framework also consists of a set of software tools to monitor the state of registered processing components, to log different types of events and to debug the flow of data given any running application context.
The remainder of this chapter is structured as follows. Section 9.2 starts with an overview of related work, examining the existing approaches and motivates the key ideas of each approach. Then we give an overview of our framework and discuss its main building blocks. Section 9.4 shows the framework in action with an example of a surveillance application built on top of the framework. The application uses several cameras and microphones to detect unusual behaviour in train compartments. Finally, section 9.5 summarizes our contribution.

## 9.2 Related work

### 9.2.1 Video in automated surveillance research

The bulk of existing work on automated surveillance, has largely concentrated on using video only. Video based surveillance related algorithms have been extensively investigated [74]. The underlying algorithms consist of methods ranging from simple background extraction algorithms to more complex methods such as dynamic layer representations [174] and optical flow methods [164].
Depending on the characteristics of the sensing environment, more specific methods have been employed. The performance of face detection algorithms [105], for example, depends on the size of the surveillance area, whereas the different methods for people or (more generally) object detection and tracking depend on the amount of occlusion and the motion characteristics of the objects involved. Even in visual event detection, different methods (e.g. video event graphs [85] and HMMs [214]) have been developed.
Some researchers approach the surveillance problem with special kinds of visual sensors and their specific algorithms. For example, in [220] the authors integrate normal camera, infra-red - IR and Doppler vibrometers - LDVs in their multimodal surveillance system for human signature detection.

### 9.2.2 Audio in automated surveillance research

As sound/speech is not always present, it is impractical to do continuous sound based tracking. Therefore, sound based surveillance algorithms are usually sound event detectors or recognizers. These algorithms usually consist of a training phase in which various features are extracted from training sound signals to obtain characteristic acoustic signatures of the different types of events. During operation, a classification system matches sound signals based on acoustic signatures to detect events [33] [79] [88]. The work of Atrey et al. [12] adapts a multi level approach in which audio frames are first classified into vocal and non-vocal events. Then, a further classification into normal and excitement events is performed. The events are modelled using Gaussian mixture models - GMMs with optimized parameters using four different audio features.

### 9.2.3 Multimodal Audio-video based approaches

Sound based algorithms are most effective when used in combination with sensors from different (in particular the visual) modalities. Audio can be used complementary to help resolve situations where video based trackers loose track of people due to occlusion by other objects or other people. The combined approach yields better and more robust results as demonstrated by various researchers using decentralized Kalman filters [167], particle filters [221] and importance particle filters [148].
In general, multimodal approaches of surveillance applications consist of unimodal, modality specific low-level feature extraction algorithms and higher-level multimodal fusion algorithms. For example, in [173] the authors use Kalman filtering to combine standalone audio and video trackers. Other high-level fusion techniques rely on Bayesian networks [93] [99] [153], rule based systems [37] and agent-based systems. A different approach is adopted in [18], where a system for tracking of moving objects was developed. In the research, the authors use a probabilistic model to describe the joint statistical characteristics of the audio-video data. Typical of this approach is that fusion of the data is done at a low level, exploiting the correlations between the two modalities. The model eventually uses audio and video variables to describe the observed data in terms of the process that generates them.

### 9.2.4 High level interpretation

At a higher level, automated surveillance research is focused on semantic interpretation of the events in their spatial and temporal relationships. In [98], an automatic surveillance system is discussed, that performs labelling of events and interactions in an outdoor environment. It consists of three components - an adaptive tracker, an event generator which maps object tracks onto a set of pre-determined discrete events, and a stochastic parser. The system performs segmentation and labelling of surveillance video of a parking lot and identifies person-vehicle interactions.
In [25], human behaviour models are used to obtain interpretation. Generally, high-level approaches are heavily dependent on the context in which the system is applied. There are at least two projects where extensive research has been conducted in applying video and audio processing algorithms for improving passenger safety and security in public transport systems.
In the PRISMATICA project [153], a surveillance system for railway stations is designed to integrate different intelligent detection devices. The detection of events is done using different algorithms that work on data collected from geographically distributed visual, audio and other types of devices. Bayesian networks are used for presenting and fusing data captured by these devices. In the ADVISOR project [37], a surveillance system was developed for detecting specific behaviour (such as fighting or vandalism) in metro stations. Different methods have been defined to compute specific types of behaviours in various environments. All these methods have been integrated in a coherent framework.

Table 9.1: Comparison of existing frameworks. The comparison is based on: modality dependency (1), applied for surveillance applications (2), scalability (3), performance (4), transparency (5), modularity (6), repository services (7), coordination services (8) and security services (9). The qualifications +, 0, -, and ? indicate good, neutral, bad and unknown respectively.

| **Framework** | **Ref.** | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| XMIDDLE | [128] | 0 | - | + | 0 | + | + | + | + | + |
| JMF/Jini | [1] | 0 | - | + | 0 | + | + | + | + | + |
| Vinci | [5] | 0 | - | + | + | + | + | + | + | - |
| E-Speak | | 0 | - | + | 0 | + | + | + | + | + |
| iROS | [150] | + | ? | ? | - | + | + | + | + | - |
| ADVISOR | [93][37] | + | + | + | + | + | 0 | - | - | - |
| PRISMATICA | [153] | + | + | + | + | + | + | - | - | - |
| KNIGHT | [99] | - | + | ? | + | ? | 0 | - | - | - |

### 9.2.5 Frameworks

Published architectures of frameworks that handle the management of data for given application contexts vary. The recent literature includes a few approaches of systems specifically designed for automatic surveillance. The focus of these has been mainly set on defining audio or video based algorithms for surveillance. Although these systems have architectures typically designed and optimized to make the modules communicate with each other, they offer very limited support in terms of extensibility and reliability. Moreover, from the system engineering perspective these approaches are far from optimal, as very important functions such as communication, resolution needs, resource management (e.g. handling distributed data sources, data sharing) and exception handling (in case sensors fail), rely on good overall frameworks.

In recent years, application specific frameworks based on emerging technologies such as peer-to-peer environments, message based middlewares and service-oriented designs have been developed. Most of them use XML or XML based technologies to support data manipulation and to facilitate component interaction. XMIDDLE [128], for example, is a mobile computing middleware using XML and Document Type Definition - DTD, Schema to enhance data with typed structures and Document Object Model - DOM [199] to support data manipulation. Further on, the system uses XPath [51] syntax to address the data stored in hierarchical tree-oriented representations. Other examples are Jini and Java Media Framework of Sun Microsystems, Vinci [5], HP Web Services platform and e-Speak of Hewlett Packard. A few studies like the work of Datcu and Rothkrantz [40] and the work of Ponnekanti et al. [150], have conducted researches for the development of frameworks for multimodal fusion applications.

Table 9.1 gives a comparison of existing frameworks. We have tried to make an overview of the frameworks used by different researches, and briefly indicate their characteristics.

## 9.3 The multimodal framework in general

The multimodal framework in this chapter centres around the shared memory paradigm. It introduces a novel technique in the way data is handled by different purpose data consumers. Comparing with the traditional way implying direct connections between the system components each connection having its own data format, the new approach suggests a more human-modelled alternative to store, retrieve and process the data.

The data is conferred an underlying structure that complies with eXtended Markup Language - XML. The shared memory in the current design of the multimodal framework takes the form of XML data spaces. The use of shared memories allows for loosely coupled asynchronous communication between multiple senders and receivers. The communication decoupling is realized in both time and space. The specification fully complies with the requirements of data manipulation in a multi data producer/consumer context where the availability of data is time-dependent and some connections might be temporarily interrupted. The information is organised in documents that are structured using XML standard. Prior to extracting the meaningful information, XML Schema [63] is used to validate each existing XML document. Furthermore, the binary data can be easily interchanged via XML documents after converting it using XML MIME protocol.

The multimodal framework also consists of a set of software tools to monitor the state of registered processing components, to log different types of events and to debug the flow of data given any running application context. Depending on the type of application to be built on top of the multimodal framework, specific routing algorithms may be used to manage the data transfer among existing shared memories on different physical networks. This capability is highly required commonly for multimodal applications that involve distinct wireless devices. Considering the study case of an automatic surveillance application, this capability allows wireless devices such as PDAs or mobile phones equipped with video camera to communicate with the system's core and to send useful video data.

The framework specifications below solely emphasize the presence and role of all its components through existing technologies and standards and do not discuss implementation details. Yet several proposed technologies present a certain degree of freedom in some functional aspects for the implementation phase. Although the multimodal framework has been designed by taking into consideration the further development of an automatic surveillance oriented application, it can be adopted as basis for any kind of complex multimodal system involving many components and heavy data exchange.

By applying the framework specifications, a common description of possible processing components along with their interconnections for a surveillance application is provided as base for eventual specific examples that may be given throughout the chapter. Because the workbench implementation itself relies on the philosophy of shared XML data spaces, special attention is given to examples on how to integrate the two underlying technologies for modules and XML data management. The examples aim at studying the usability and extensibility of concepts in formulating proper data and command requests to ensure a logic and transparent data flow through the network of data processing nodes.

The algorithms that support the illustrated processing components of the

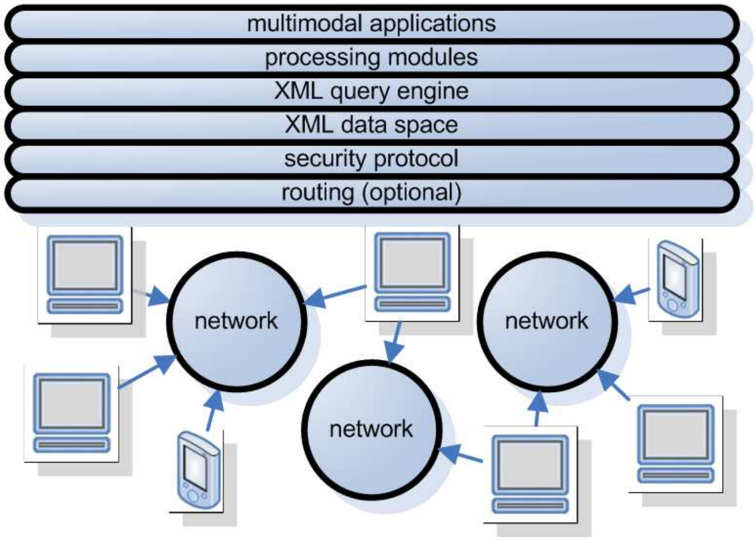


Figure 9.1: Functional layers of the multimodal framework.

automatic surveillance application are not described in full detail due to space and topic considerations. Indeed, these audio-visual data processing components cover a broad range of research areas. Besides, any standard hardly exists to favour one method over the other. In some restricted cases, algorithms are employed for exemplification purposes, though. One distinct remark concerning the multimodal framework is that it should support platform independent interconnection of the processing modules. This requirement is essential for applications working in heterogeneous environments. An overview of the current multimodal framework is shown in the diagram in figure 9.1. If the processing modules are envisaged as services of the multimodal workbench, then a multimodal application specifies, along with the service set, a working plan on how the services are used to get the data through different abstraction levels to obtain the desired semantic information. The multimodal framework defines a way to represent the work plan through the Monitor application and uses that to get a detailed state analysis over its fulfilment.

Each service registered in the framework publishes its input and output XML formatted specifications using Web Services Description Language - WSDL [31]. The service oriented overview is given in figure 9.2. Similar approaches to integrate distributed services in distributed object environments - DOEs, include: Jini [1] proposed by Sun Microsystems in the late 90's, the service-oriented architecture - SOA, Common Object Request Broker Architecture - CORBA from OMG and Distributed Component Object Model - DCOM from Microsoft.

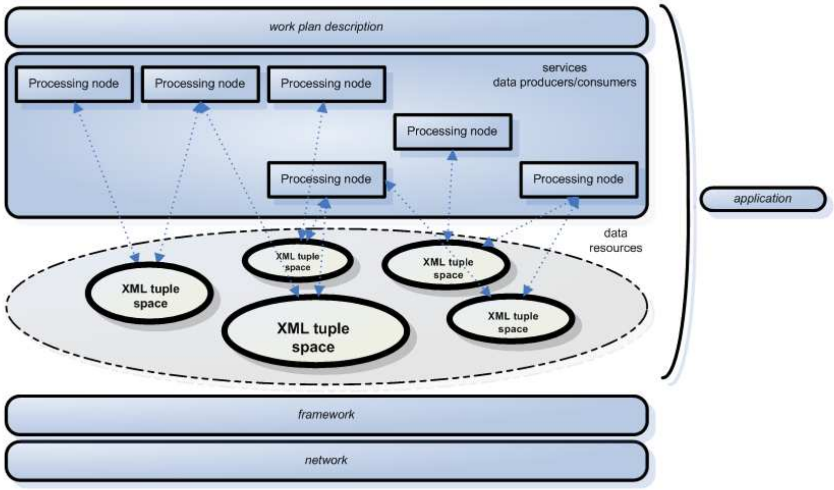


Figure 9.2: Overview of the service-oriented framework.

### 9.3.1 XML data spaces

An essential requirement for choosing the architecture of a multimodal framework is to define the support for data integration. Several technologies like Remote Procedure Call - RPC, Remote Method Invocation - RMI or Common Object Request Broker Architecture - CORBA, have been previously proposed to overcome the problems which arise in the distributed computing environment. Still, such methods limited robustness in specific working contexts.

Due to its capability to solve the typical problems regarding synchronization, persistence and data communications, the extensively researched algorithms based on tuple spaces have been used ever since the introduction by Yale University in the mid 80's. The goal of the original Linda system was to achieve coordination for various parallel applications. The greatest advantage of the tuple spaces is the decoupling of sender and receiver components in both time - by removing the existence simultaneity restriction, and space - by removing the address information requirement. One property of shared memory spaces is that they remove the requirement for prior information about the availability and the address of the destination components.

Later systems which reused some characteristics of Linda were: Melinda which introduced the idea of multiple data spaces, Limbo for adaptive mobile applications, Lime and eLinda which proposed fully distributed tuple spaces and additional forms for the data output operations. Another features of eLinda regard Programmable Matching Engine - PME, as a more flexible mechanism to address tuples and the multimedia support based on Java Media Framework - JMF. More recent extensions of original Linda system include JavaSpaces from Sun Microsystems and TSpaces [73] from IBM. These systems store and exchange data in form of objects. JavaSpaces uses transaction terms to address tuple space operation sets which can be rolled back in faulty cases. An additional feature is the introduction of leases to the context of tuple spaces. This is basically used to automatically remove tuples after their time expiration. The architecture of TSpaces was defined in such a way so as to integrate services which are developed using different programming languages. This could be done by using either WSDL or proprietary description languages - IDLs. The same framework included a rule-based mechanism to associate tuple space events to particular actions.

The work of Khushraj et al. [104] proposed sTuples as a framework which integrates semantic tuple spaces. This framework extended a secure communication layer that supports a description-logic reasoning engine and a web ontology language. More recent versions of tuple space models propose to encode the data using the XML format. Such systems include xSpace [20], Rogue-Ruple [175] and XMLSpaces.NET [181].

xSpace system introduces a query language derived from xPath to enhance the access to both the content and structure of the data. The model offers support for both XML Schema and DTD. XMLSpaces.NET implements the tuple spaces on the .NET platform.

The architecture of our multimodal framework represents an adapted version of Ruple XML spaces. Ruple Alpha V2 release supports XML Spaces, HTTP and Simple Object Access Protocol - SOAP for data accessibility, MIME attachments and XQL, a subset of XML query language for document retrieval.

Our multimodal framework system is adapted for fast and reliable storage and retrieval of the data. To achieve this, four basic operations (figure 9.3) on XML data spaces are provided, as follows: write - for sending XML documents to the XML spaces, read - for retrieving XML documents from the XML spaces based on XML querys, readMultiple - for retrieving a set of XML documents based on XQuery XML queries, respectively take - for retrieving and removing XML documents based on XML queries. In the current framework architecture, the query can be made to retrieve data from multiple XML spaces. An example of such a query procedure is illustrated in figure 9.4. The operation demands that separate connections have to be created first to each XML tuple space involved in the query. After retrieving the XML data using the basic tuple operations, the result is generated using the specifications of the query. The specifications relate to certain elements and attributes of the XML data. In case that the query includes elements of audio/video data, these are finally extracted from the previously generated XML document. The encoding and decoding of audio/video data follow the specifications of base64 protocol. The multimodal framework integrates XML spaces as well as services which work simultaneously as data producers and consumers. From the connectivity point of view, these entities may be located at different nodes of the interconnected physical networks. The number of XML data spaces that can be created at a certain moment is limited only by the characteristics of the hardware equipments that support the logical infrastructure of the framework. High-speed transfers of the XML data among the data producers and consumers, are achieved by permanently optimizing the inter and intra data flows of the XML data spaces.

Figure 9.5 illustrates an example for the integration of XML spaces. Every two logical data spaces are asynchronously connected through at least one common node in the network. Such nodes act as bridges between XML tuple spaces located in distinct physical networks. In the figure, Physical Network Layer I depicts the interconnection in the case of three different physical networks

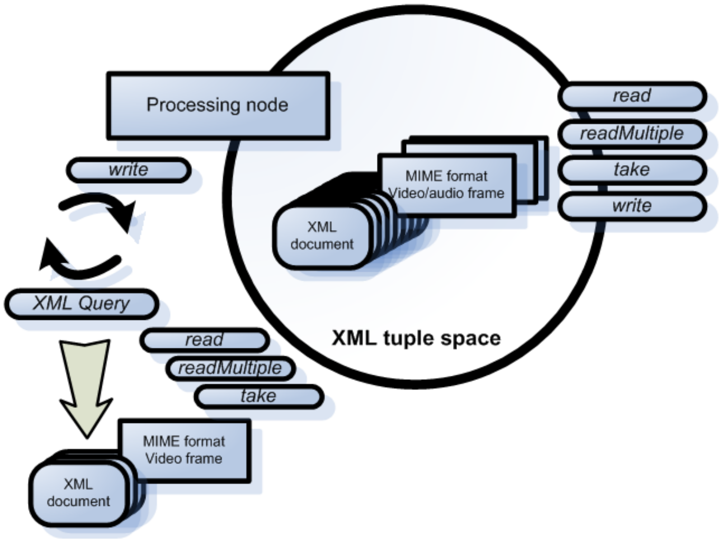


Figure 9.3: The basic operations used to handle the XML documents in XML tuple spaces.

(Net1, Net2 and Net3). The physical network Net1 consists of the following logical processing nodes: PN1, PN2, PN3, PN4, PN5, PN6. The same physical network includes two hardware sensor nodes SN1 and SN2. The logical processing nodes PN1, PN2 run on the same hardware node that is connected to physical network Net1. Similarly, logical processing nodes PN8 and PN9 run on the same hardware node which is connected to physical network Net2. Net2 also includes processing nodes PN6, PN7 and PN10. Net3 contains processing nodes PN10, PN11, PN12 and PN13.
The logical processing nodes from each physical network make one XML data space. The three data spaces are logically connected through logical processing nodes that run on hardware nodes that link different hardware networks. In the example, PN6 connects data spaces from physical networks Net1 and Net2 and PN10 connects the data spaces from physical networks Net2 and Net3. The interconnection mechanism generates a logical network that supports a larger XML data space. This data space spans across three different physical networks. The effect is that nodes that have no direct visibility with each other, can still exchange data through a sequence of interconnected XML spaces. In the second case, Physical Network Layer II illustrates an example of a unique physical network Net. This network includes all physical notes that make the three XML spaces of the multimodal framework. Our implementation of XML data spaces follows some specifications made by other research papers.
Beside the basic XML space operators, our framework model specifies certain facilities to the processing modules. The processing modules may create new data spaces, subscribe or un-subscribe to existent data spaces, send and retrieve

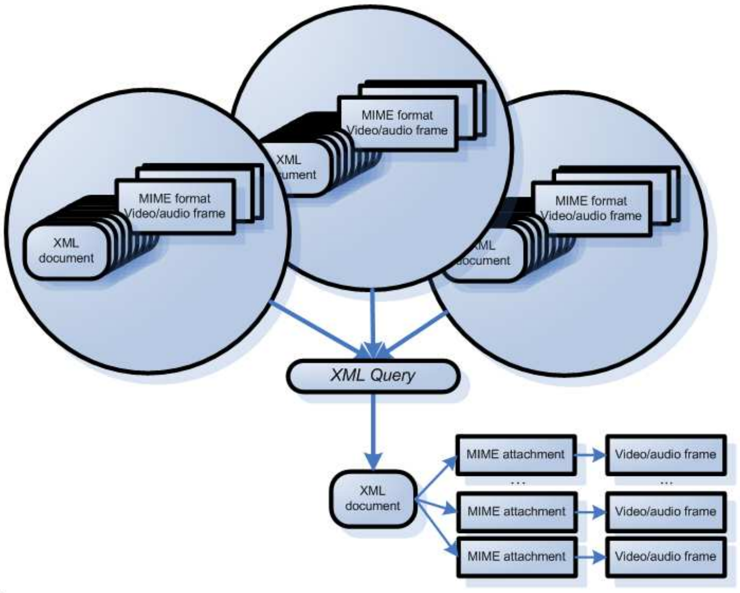


Figure 9.4: Audio/video data retrieval based on XML query from multiple XML tuple spaces.

XML-formatted data by calling the specific functions. In order to do this, the processing nodes have to specify the identification code of the data space. The Monitor tool provides effective mechanisms that report on the workload and status of the current multimodal application. In special cases like breakdowns, this framework component may keep the integrity of the data by restarting the processing modules and by recovering the affected data spaces.

### XML document lease

The framework automatically checks for the availability of the documents as specified by the XML document lease. Before sending XML data to data spaces, the XML documents are assigned leases which specify the data expiration time. The parameter determines for how long the XML documents are available in the data space. Regular checks update the document list of every data space to identify the documents that have the expiration time up. In such cases, these documents are removed from the set of XML documents associated to the XML data space.

By using the information from the document's lease, the system restricts the access of the multimodal applications that run on top of the multimodal framework, to data that have low level of integrity. Occasionally connected devices can provide data to the multimodal framework and also receive processing results. Eventually, the method leads to an efficient and flexible exchange of data

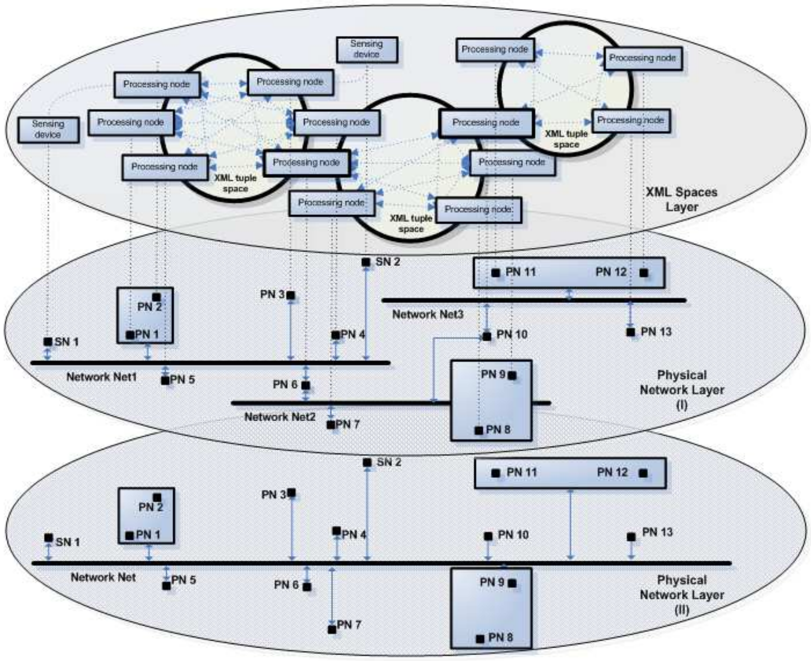


Figure 9.5: Connections between multiple data spaces and the distribution of nodes on the physical network.

between the sender and destination nodes.

### Multimedia data support

Automatic systems for face analysis that incorporate several data processing components transfer face images among components using intermediate representations using the raw format. Baayens [192] provides general theoretical aspects for developing systems that generate and exchange parametrized representations based on 3D surface analysis. This approach generates smaller amount of data to be transferred and possibly leads to faster facial expression recognition.
The second approach is to transfer the face images in the raw format. Any XML documents may include audio/video data in form of attachments. As illustrated in figure 9.6, the manipulation of multimedia data is supported through certain system functions. The base64 [102] binary-to-text encoding scheme is used to convert audio or video frame binary sequences to sequences of printable ASCII characters. This operation allows for the integration of binary data into text-based XML files.
The reverse of the encoding is the text-to-binary format conversion. This is requested at the time of retrieving XML documents from the XML data spaces,

prior to the actual audio/video data processing.

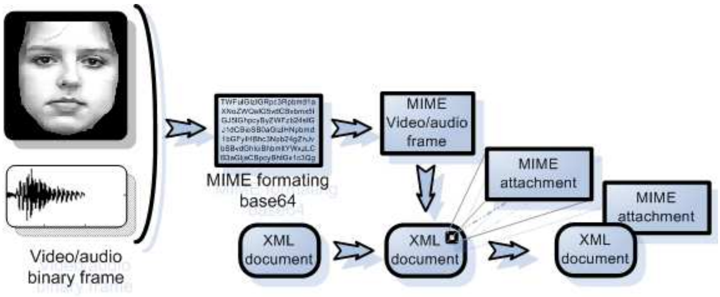


Figure 9.6: Transformation of binary data to base64 formatted attachments.

To support the target multimodal applications, the current framework provides special functions to fuse data of different types. The fusion procedure depends on the level of abstraction that characterizes each data type. The types relate to raw, intermediate and semantic data. The raw data is represented by unchanged signal-level data that is received from the input-sensing devices. Examples of this type are the image pixel intensities, other visual features extracted from the video frames acquired from the video cameras, parts of the audio signal or acoustic features computed from the audio signal acquired from the microphone devices.

The intermediate data represent all kinds of features that are used as input to other processing modules, yet do not have a self-contained semantic meaning to the application-context. For example, facial features such as geometric features determined based on the location of particular facial characteristic points or features related to the phonemes detected from the speech signal, are considered to belong to the class of intermediate data. It is a thin line between the meaning of the low-level data and the meaning of the intermediate data. Therefore, in most of the cases researchers consider both types of data as being part of only one category, the low-level data. The third type is the high-level, semantic data. This category contains highly aggregated data extracted following extensive audio-visual processing on the raw, the intermediate or other high-level data. Examples from this category are: the number of persons or the number of human faces detected in the video frames, the description of the gestures or the description of the emotion status for each subject or the meaning of what the subjects are saying.

In our multimodal framework, we define separate data spaces according to the category of data they store. As shown in figure 9.7, the XML data spaces include raw data spaces, intermediate data spaces and semantic data spaces. The integration of data extracted from different data spaces is done in a similar manner. Accordingly, there are three types of data fusion methods: low-level fusion, intermediate fusion and high-level fusion. The low-level fusion is also called early fusion or fusion at the signal level. High-level fusion is also called semantic fusion, late fusion or decision level fusion. The low-level data fusion

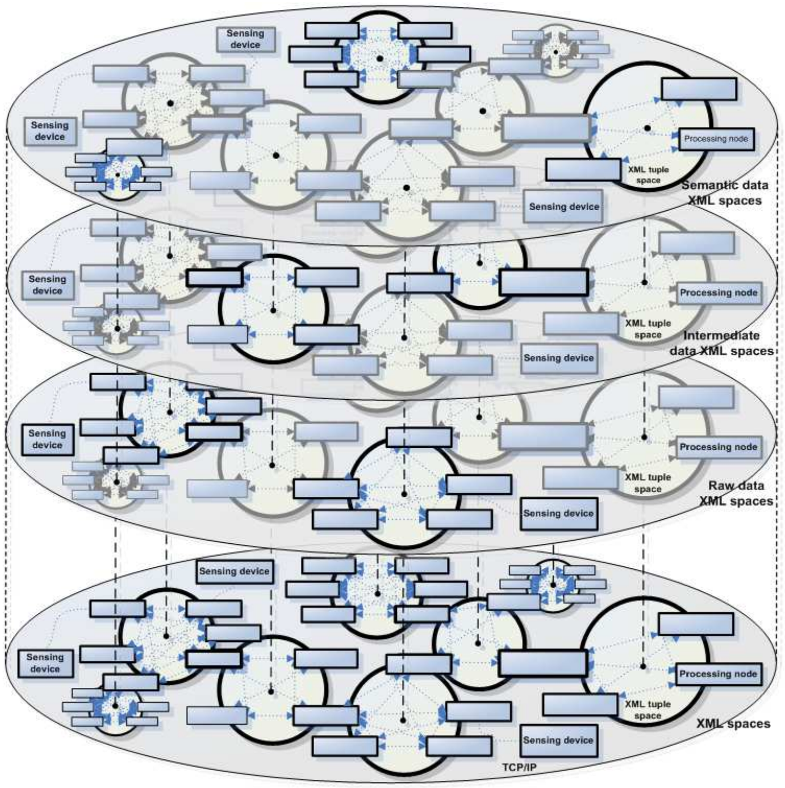


Figure 9.7: Transformation of data to higher levels of abstraction through different XML Spaces.

involves operations on raw data received directly from the sensing devices. The output represents intermediate or data at the high abstraction level. The intermediate data fusion operates on either intermediate-level or both low-level and intermediate-level data. Analogically, the high-level data fusion integrates high level data or combinations of high-level data with intermediate and low level data.
In figure 9.7, the types of fusion are represented by links between logical processing nodes that are part of data spaces at different levels of abstraction.

### 9.3.2 Querying data from XML data spaces

Query languages present an efficient way to support content-based information extraction and updating. In the current framework architecture, the data is stored using XML specifications. Prior to processing the data, the registered framework modules perform query requests on XML shared spaces. For this, a XML query language is adopted for accessing data from well-defined XML data spaces. Our multimodal framework supports queries in the XQuery format [27].

XQuery is a typed, functional language for querying XML documents. It has been developed by XML Query Working Group of the World-Wide Web Consortium (W3C). The XQuery specification set defines high-level descriptions of the XML syntax as well as of the underlying semantics. The query expressions consist of sets of basic building blocks defined as Unicode strings that contain keywords, symbols and operands.

There are several XML-related standards that XQuery shares common interoperability with. XPath describes how to manipulate the content of XML documents which are generated in correspondence with XML Schema. The XML Schema contains essential information about the structure of XML documents. Alternatives to XML Schema are Document Type Definition - DTD which is included in the original specification of XML, or Document Structure Description - DSD. According to W3C, XML Schema does not represent a requirement for using XQuery. However, in the current multimodal framework, we define the use of XML Schema as condition for getting access to the data. For that, the XML Schema data is retrieved from the data space containing the XML documents. In case the structure of an XML document does not comply with the data from the document's XML schema, an error is raised and the access to the content of the document is not granted any longer.

Another use of XML schema is to verify if attributes are missing. In this case, attributes are created and initialized using the default values.

A major advantage of XQuery is the support for expressions like **f**or, **l**et, **w**here, **o**rder by and **r**eturn - **FLWOR**. These language expressions allow for using variable iteration and binding to obtain intermediate query representations. In this way, XML documents are first parsed by an XML parser to generate XML Information Set.

In our implementation, we use Xerces [4] for parsing XML documents using DOM/SAX parsers. After parsing, the documents are validated using XML Schema. The result represents an abstract structure called Post-Schema Validation Infoset - PSVI.

To make the connection between the framework's variables and functions and the XQuery expressions, we implemented an interface based on XQuery Expression Context. The building blocks of the process are represented graphically in figure 9.8. After using the XML Schema, the system creates and labels an internal representation of the data. The resulting XML Data Model - XDM is derived from the Information Set or PSVI. Eventually, the values of the query expressions are determined during several static and dynamic evaluations.

In figure 9.8, the component Interface connects framework routines with the XQuery XML query processor. As part of this process, the input and output parameters of the routines and the parameters used in XQuery expressions are optimally matched and translated. Specific data types such as the face and gesture types, are implemented as part of the extended type set of XQuery/XPath Data Model - XDM.

For the face detection routine, our implementation [198] uses Viola&Jones visual features as input and Adaboost and relevance vector machine - RVM classifiers.

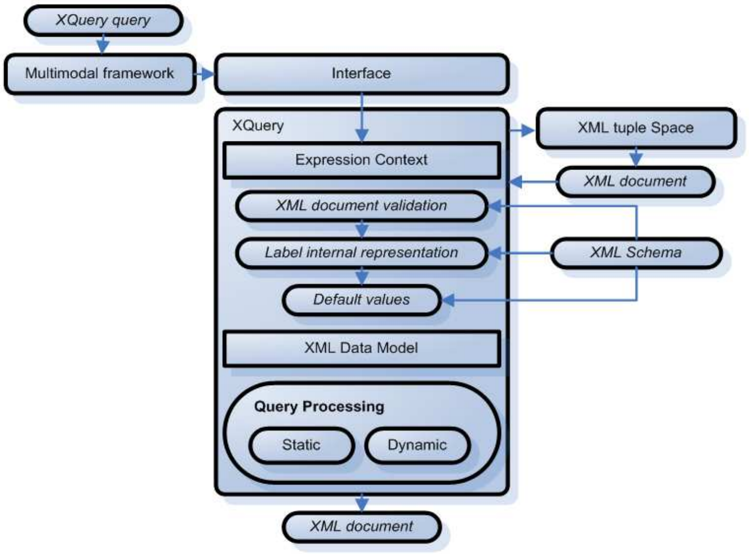


Figure 9.8: Interface between the multimodal framework and the XQuery.

### 9.3.3 Comparison of the multimodal framework with iROS framework

In our research on multimodal frameworks, we have extensively used the iROS Event Heap [150] made by Stanford University. iROS represents an adaptive middleware infrastructure that provides coordination of data and interactions in ubiquitous computing environments. Among other features, it supports transparent communication, content based addressing, limited data persistence and logical/physical centralization.

Table 9.2 presents the list of functional characteristics for both iROS and our multimodal framework. Our multimodal framework provides support for both wired and wireless adhoc networks. The access to our multimodal framework is done in a transparent way, without being necessary to know the address of any network components. Conversely, in case of using iROS, the processing nodes have to acquire the physical address of iROS DataHeap. This is because iROS is based on the centralized component approach. In our framework model, we propose the integration of routing support as solution for multi-hop networks. Using that, the system functionality may seamlessly cover different physical networks. In this view, the framework nodes act as bridges that extend the accessibility range of the system. On the other hand, the functional accessibility range of iROS is restricted to the network where the centralized component resides.

Our multimodal framework is up and running as long as there is at least one node to support it. iROS functionality is conditioned by the state of the centralized component (iROS DataHeap). If a problem occurs at that layer, then

Table 9.2: Comparison between TUDelft multimodal framework and iROS

| | **iROS** | **TUDelft framework** |
|---|---|---|
| *Approach* | centralized | decentralized |
| *Network type* | fixed | (multi-hop)adhoc |
| *Network protocol* | TCP | UDP |
| *Routing* | standard | advanced |
| *Binary audio/video data support* | - | ok |
| *XML support* | - | ok |
| *Semantic query* | - | ok |
| *Data structure validation* | - | ok |
| *Physical networks connectivity* | - | ok |
| *Fault tolerance* | - | ok |
| *Service support* | - | ok |
| *Monitoring tools* | - | ok |
| *Security* | - | ok |
| *Decoupling(time,space)* | ok | ok |
| *Event handling* | ok | ok |
| *Limited data persistence* | ok | ok |
| *Log support/activity tracking* | - | ok |
| *Update operation support* | - | ok |
| *Replication* | - | ok |

the functionality of the entire framework is compromised. In our framework, the transfer of data and control packages is realized using UDP protocol. This feature naturally fits with the work conditions in wireless networks that imply that sudden connection breakdowns may occur.

Additionally, we propose a connection recovery strategy to handle such cases. For that, the framework keeps track of the network disconnections and makes sure the data packages are temporary stored until the nodes can be reached again.

iROS employs the TCP protocol to connect the client nodes to the iROS Data-Heap. To ensure for the integrity of the data in this setup, additional work must be done.

One important characteristic of our multimodal framework is the support for XML documents. Besides, we propose a query-oriented method to access and validate the data based on XML content and XML Schema. Together with the support for binary audio-video data, these functions provide multimodal applications with proper handling of the multimodal content.

### 9.3.4 The general model for an automatic surveillance application

Based on the description of the multimodal framework, we have made an implementation of the system using Java programming language. In this part of the chapter, we present an application for automatic surveillance that is built on top of the previously described multimodal framework.

The surveillance application employs the context awareness produced by applying data processing methods on the input generated by the sensing devices. The methods for audio-visual data processing are implemented and provided at the level of general processing nodes which are part of the multimodal framework or at the level of specific processing nodes which are part of the surveillance application.

The diagram in figure 9.9 presents the typical processing modules of the automatic surveillance application. The processing nodes include the detection of type and location of acoustic patterns such as glass breaking or shouting, speech recognition, emotion extraction from speech, face and gaze detection, gesture and facial expression recognition. The goal of the modules is to process data and to extract useful information from audio and video signals.

Depending on the type of inputs and on the algorithms involved, the results of these modules represent data at different levels of abstraction. For instance, the role of the Behaviour recognition module is to detect certain types of aggression that take place in the scene being analysed. The analysis focuses on single persons as well as on groups of persons. The data which regard the aggression is stored in the XML data space *scene.person.semantic.aggression*. The application uses the information from this shared space to generate the system feedback in the form of two types of actions.

Based on which action types are employed, we say the application runs active or passive surveillance. The first type includes the set of actions taken when aggression is detected. It may imply the recording of the events or the generation of notifications such as SMS, email or phone messages. The decision on the type of action is made using the data stored in the data space *scene.systemFeedback*. The second type of actions concerns motor commands for the pan-tilt-zoom camera. The camera motor commands are determined by tracking subjects based on visual and audio signals and are stored in the data space *scene.action.focus*. For tracking, the module *Focus* integrates clues which regard the location of the faces, from the data spaces *video.face.features* and *video.frames*, the location of the source of sound, from the data space *audio.semantic.sound.location* and aggression information, from the data space *scene.person.semantic.aggression*. The result is stored in the data space *scene.action.focus* and is further taken by a camera interface module.

Listing 9.1 shows typical examples for querying data from the data space *video. frames* containing video frames captured by the video cameras that are connected to the framework. The XML Schema of the documents containing the video frames, is presented in listing 9.2. The query *Q1* generates XML formatted results of video frame data. The second query *Q2* makes a selection of jpeg video frames received from camera *camera1*. The XML result of this query contains only the XML element *data*. The query *Q3* represents an alternative way of writing the query *Q2*. The last query example *Q4* iterates through all the video frames of type jpeg received in the last 60 seconds by camera *camera1*. Then, a locally declared function is called to detect the faces in each video frame.

In our multimodal workbench, the face detection routine is managed as a service that has the WSDL specification as exemplified in listing 9.3. The query for retrieving data generated by the face detection routine, has XML Schema presented in listing 9.4. In order to correctly estimate the emotion of the subjects, the system integrates audio and visual features from the low-level or semantic data. For that, the features are first retrieved using query expressions. The emotion information extracted from speech signal is stored in the data space *audio.speech.semantic.emotion* and has the XML Schema as showed in listing 9.5.

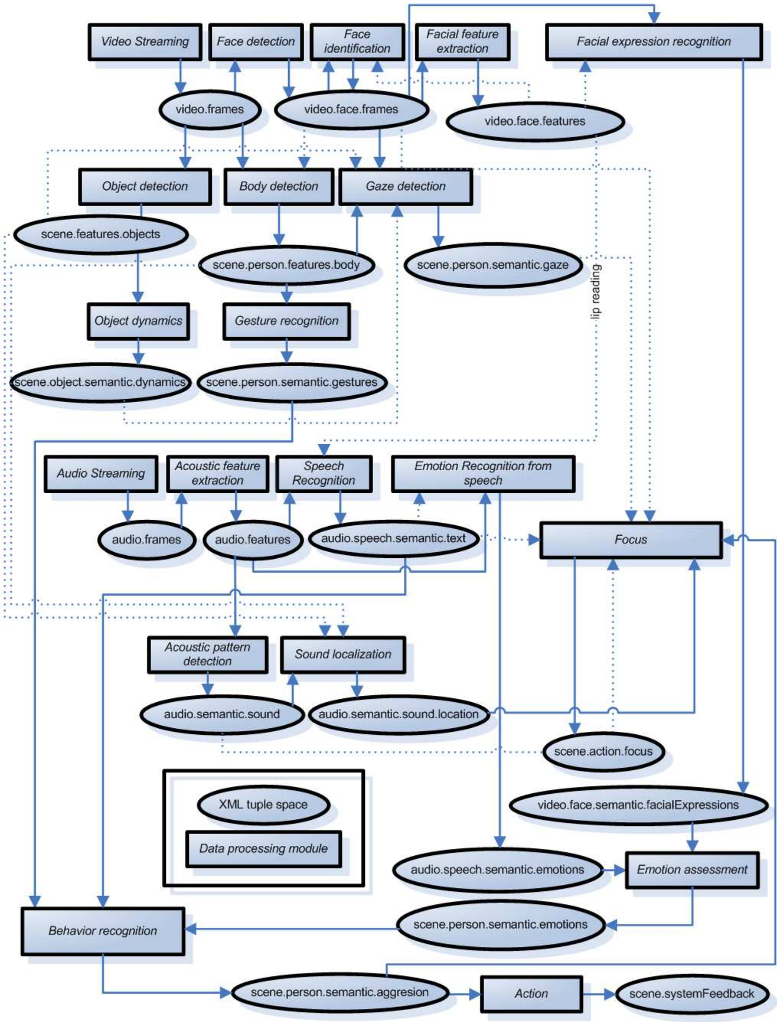

Video Streaming
Face detection
Face identification
Facial feature extraction
Facial expression recognition
video.frames
video.face.frames
video.face.features
Object detection
Body detection
Gaze detection
scene.features.objects
scene.person.features.body
scene.person.semantic.gaze
lip reading
Object dynamics
Gesture recognition
scene.object.semantic.dynamics
scene.person.semantic.gestures
Audio Streaming
Acoustic feature extraction
Speech Recognition
Emotion Recognition from speech
audio.frames
audio.features
audio.speech.semantic.text
Focus
Acoustic pattern detection
Sound localization
audio.semantic.sound
audio.semantic.sound.location
scene.action.focus
XML tuple space
Data processing module
video.face.semantic.facialExpressions
audio.speech.semantic.emotions
Emotion assessment
Behavior recognition
scene.person.semantic.emotions
scene.person.semantic.aggresion
Action
scene.systemFeedback

Figure 9.9: Typical diagram of the shared data spaces and of the processing modules for the automatic surveillance application.

The facial expression recognition results are stores in XML documents from the data space *video.face.semantic.facialExpressions*. These documents have XML Schema as illustrated in listing 9.6. The surveillance application includes the emotion recognition from speech routine based on GentleBoost classifier as described by Datcu and Rothkrantz [42]. A distinct type of emotion recognition is the assessment of the stress level. This is implemented as part of the module *Emotion recognition from speech* and follows the specifications by Rothkrantz et al. [152]. The result constitutes partial input for the module *Emotion assessment*. The facial expression recognition routine is represented by module *Facial expression recognition* which implements models based on RVM, SVM [41][202], and Bayesian belief networks [39], as proposed by Datcu and Rothkrantz. The classification uses geometric features that represent distances and angles between specific key points on the face image. Finally, the fusion of audio-visual emotion data uses dynamic Bayesian belief networks described by Datcu and Rothkrantz [46] or hidden Markov models (see chapter 7).
Listing 9.7 shows an example of a query that prepares the unimodal emotion recognition, in case of the decision level fusion. The result of the fusion contains time-ordered emotion labels structured using tags which identify the human subjects. In terms of XQuery specifications, the operation is a join on partial XML data from two XML documents. The same procedure of feature retrieval is used in the case of low-level fusion. However, in this case an initial synchronization handles the difference between the sampling rates of the audio and the video features. The synchronization is done taking into account the capturing timestamps of each feature (figure 9.10). The multimodal fusion is represented by the processing module *Emotion assessment*. The output of this module is stored in the data space *scene.person.semantic.emotions*.

## 9.4 Automatic surveillance application

In this section, we present a multimodal surveillance experiment based on the framework, to illustrate the feasibility of our approach and to show how the framework can be used. The experimental setting is a Dutch international train (BeNeLux train) compartment. The scenarios for the recordings involved a number of hired actors and train conductors playing specific (normal as well as unusual) scenarios. The data was used as input to a surveillance application built on top of the framework.

Listing 9.1: Examples of using XQuery language to retrieve data from tuple <video.frames>.

```
Q1:
doc(video.frames.xml)/videodata/frame

Q2:
doc(video.frames.xml)/videodata/frame[cameraID=camera1 and
        dataType=jpg]/data

Q3:
for \$video_frame in doc(video.frames.xml)/videodata/frame
where \$video_frame/cameraID=camera1 and dataType=jpg
return \$video_frame/data

Q4:
<faces>
{

let \$current_time:= current-time()
for \$vf in doc(video.frames.xml)/videodata/frame
where xs:time(\$vf/time)>$current_time-60
order by xs:time(\$vf/@time)
return
<face faceID= {local:getFaceID(\$vf)} >
      <personID>unidentified</personID>
      \$vf/cameraID
      \$vf/time
      local:detectFaces(\$vf/width, \$vf/height, \$vf/dataType,
                  \$vf/data)
</face>
}
</faces>
```

Listing 9.2: XML Schema for tuple <video.frames>

```
<xs:schema xmlns=”http://www.w3.org/2001/XMLSchema”>
<xs:element name= videodata type=videodata-type >
<xs:complexType name= videodata-type >
  <xs:sequence>
     <xs:element name= frame type=frame-type minOccurs= 1
                 maxOccurs= unbounded />
  </xs:sequence>
</xs:complexType>
<xs:complexType name= frame-type >
  <xs:sequence>
     <xs:element name= cameraID type= xs:string minOccurs= 1
                 maxOccurs= 1 />
     <xs:element name= time type= xs:time minOccurs= 1
                 maxOccurs= 1 />
     <xs:element name= width type= xs:integer minOccurs= 1
                 maxOccurs= 1 />
     <xs:element name= height type= xs: integer minOccurs= 1
                 maxOccurs= 1 />
     <xs:element name= dataType type= xs:string minOccurs= 1
                 maxOccurs= 1 />
     <xs:element name= data type= xs:string minOccurs= 1
                 maxOccurs= 1 />
  </xs:sequence>
  <xs:attribute name= frameID type= xs:string use= required />
</xs:complexType>
</xs:schema>
```

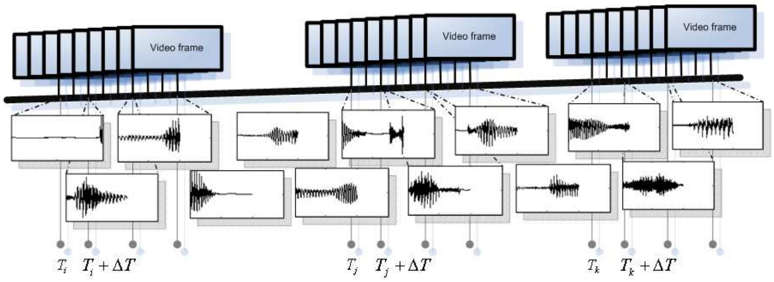


Figure 9.10: Synchronization of low level visual and audio features.

### 9.4.1 Goal

BeNeLux train compartments are equipped with several microphones and cameras. Currently, the audio and video data captured by these sensors is transmitted to a central location, where operators monitor the data manually and take appropriate action when unusual events occur. Figure 9.11 shows the interface an operator is confronted with. Our goal is to use the framework presented in this chapter to build a system to automate the manual inspection process that is currently performed by the human operators. More specifically, the system should detect unusual behaviour in the train compartment and notify the op-

Listing 9.3: Example of WSDL statement for Face Detection processing module

```
< wsdl:definitions>
<xsd:element name=”faceDetection”>
   <xsd:complexType>
      <xsd:sequence>
          <xsd:element name=”width” type=”string”/>
          <xsd:element name=”height” type=”string”/>
          <xsd:element name=”dataType” type=”string”/>
          <xsd:element name=”data type=”string”/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>

< wsdl:message name=”getWidthRequest”>
<wsdl:part name=”width ” type=”xs:unsignedShort”/>
</wsdl:message>

< wsdl:message name=”getHeightRequest”>
<wsdl:part name=”height” type=”xs:unsignedShort”/>
</wsdl:message>

< wsdl:message name=”getDataTypeRequest”>
<wsdl:part name=”dataType” type=”xs:string”/>
</wsdl:message>

< wsdl:message name=”getDataRequest”>
<wsdl:part name=”data” type=”xs:string”/>
</wsdl:message>

<wsdl:message name=”getFaceResponse”>
<wsdl:part name=”face” type=”faceDetection”/>
</wsdl:message>

<wsdl:portType name=”FaceDetectionLib”>
<wsdl:operation name=”detectFaces”>
<wsdl:input message=”getWidthRequest”/>
<wsdl:input message=”getHeightRequest”/>
<wsdl:input message=”getDataTypeRequest”/>
<wsdl:input message=”getDataRequest”/>
<wsdl:output message=”getFaceResponse”/>
</wsdl:operation>
</wsdl:portType>
.
.
.
</ wsdl:definitions>
```

Listing 9.4: XML Schema for tuple <video.face.frames>

```
<xs:schema xmlns=”http://www.w3.org/2001/XMLSchema”>
<xs:element name= faces type= faces-type >
<xs:complexType name= faces-type >
  <xs:sequence>
      <xs:element name= face type= face-type minOccurs= 1
                 maxOccurs= unbounded />
  </xs:sequence>
</xs:complexType>
<xs:complexType name= face-type >
  <xs:sequence>
      <xs:element name= personID type= xs:string minOccurs= 1
                 maxOccurs= 1 />
      <xs:element name= cameraID type= xs:string minOccurs= 1
                 maxOccurs= 1 />
      <xs:element name= time type= xs:time minOccurs= 1
                 maxOccurs= 1 />
      <xs:element name= width type= xs:integer minOccurs= 1
                 maxOccurs= 1 />
      <xs:element name= height type= xs:integer minOccurs= 1
                 maxOccurs= 1 />
      <xs:element name= dataType type= xs:string minOccurs= 1
                 maxOccurs= 1 />
      <xs:element name= data type= xs:string minOccurs= 1
                 maxOccurs= 1 />
  </xs:sequence>
  <xs:attribute name= faceID type= xs:string use= required />
</xs:complexType>
</xs:schema>
```

erators. This approach implies that the operator is still in charge of taking the appropriate actions.

### 9.4.2 Experimental setup

In order to capture realistic data, several professional actors and a train conductor were asked to play specific scenarios in the train compartment. We used four cameras and four microphones in the compartment to capture these scenarios. As BeNeLux train compartments are already equipped with cameras, we have used these pre-installed cameras to capture video. The microphones however, where not located at the positions we preferred. Therefore, we installed four microphones to capture audio data. As can be seen from figure 9.12, the microphones do not cover the entire compartment. So, most of the unusual scenarios where played near the microphones. The unusual scenarios we asked the actors to play fell in the category of behaviours we want our system to detect, namely: fighting (including aggression towards the conductor and disturbance of peace), graffiti and vandalism, begging and sickness. The framework modules used to detect the behaviour include face recognition component, gesture recognition component, face detection component, facial expression recognition component, and emotion recognition from speech component. The mapping between the behaviour types and the processing modules, is given in table 9.3. The resulting surveillance application that was built on top of the framework consists of several interconnected detection modules made available by the framework.

Listing 9.5: XML Schema for audio.speech.semantic.emotions.xml.

```
<xs:schema xmlns=”http://www.w3.org/2001/XMLSchema”>
<xs:element name= audio  type=audio-type >

<xs:complexType name=audio-type >
  <xs:sequence>
      <xs:element name= frame  type=frame-type  minOccurs= 1
                maxOccurs=unbounded />
  </xs:sequence>
</xs:complexType>

<xs:complexType name=frame-type >
  <xs:sequence>
      <xs:element name= personID  type= xs:string  minOccurs= 1
                maxOccurs= 1 />
      <xs:element name= time  type=interval-type  minOccurs= 1
                maxOccurs= 1 />
      <xs:element name=emotion  type=  emotion-type  minOccurs= 1
                maxOccurs= 1 />
  </xs:sequence>
  <xs:attribute name=frameID  type= xs:string  use=required />
</xs:complexType>

<xs:complexType name=  interval-type >
  <xs:sequence>
      <xs:element name= start  type= xs:time  minOccurs= 1
                maxOccurs= 1 />
      <xs:element name= stop  type= xs:time  minOccurs= 1
                maxOccurs= 1 />
  </xs:sequence>
</xs:complexType>

<xs:complexType name=emotion-type >
  <xs:sequence>
      <xs:element name= happiness  type= xs:integer  minOccurs= 1
                maxOccurs= 1 />
      <xs:element name= sadness  type= xs:string  minOccurs= 1
                maxOccurs= 1 />
      <xs:element name= fear  type= xs:string  minOccurs= 1
                maxOccurs= 1 />
      <xs:element name= disgust  type= xs:string  minOccurs= 1
                maxOccurs= 1 />
      <xs:element name= surprise  type= xs:string  minOccurs= 1
                maxOccurs= 1 />
      <xs:element name= anger  type= xs:string  minOccurs= 1
                maxOccurs= 1 />
  </xs:sequence>
</xs:complexType>
</xs:schema>
```

Listing 9.6: XML Schema for video.face.semantic.facialExpressions.xml.

```
<xs:schema xmlns="http://www.w3.org/2001/XMLSchema">
<xs:element name= video type= video-type >

<xs:complexType name= video-type >
  <xs:sequence>
     <xs:element name= frame type=frame-type minOccurs= 1
                 maxOccurs= unbounded />
  </xs:sequence>
</xs:complexType>

<xs:complexType name= frame-type >
  <xs:sequence>
     <xs:element name= personID type= xs:string minOccurs= 1
                 maxOccurs= 1 />
     <xs:element name= time type= xs:time minOccurs= 1
                 maxOccurs= 1 />
     <xs:element name= emotion type=emotion-type minOccurs= 1
                 maxOccurs= 1 />
  </xs:sequence>
  <xs:attribute name= frameID type= xs:string use= required />
</xs:complexType>

<xs:complexType name= emotion-type >
  <xs:sequence>
     <xs:element name= happiness type= xs:integer minOccurs= 1
                 maxOccurs= 1 />
     <xs:element name= sadness type= xs:string minOccurs= 1
                 maxOccurs= 1 />
     <xs:element name= fear type= xs:string minOccurs= 1
                 maxOccurs= 1 />
     <xs:element name= disgust type= xs:string minOccurs= 1
                 maxOccurs= 1 />
     <xs:element name= surprise type= xs:string minOccurs= 1
                 maxOccurs= 1 />
     <xs:element name= anger type= xs:string minOccurs= 1
                 maxOccurs= 1 />
  </xs:sequence>
</xs:complexType>

</xs:schema>
```

Table 9.3: The mapping between the behaviour to recognize and the framework modules involved.

| *Behaviour* | *Services* |
|---|---|
| **Fighting** | gesture recognition<br>emotion recognition in speech |
| **Graffiti and vandalism** | Sound event recognition<br>gesture recognition |
| **Begging** | gesture recognition<br>motion tracking<br>speech recognition |
| **Sickness** | gesture recognition<br>emotion recognition from speech<br>speech recognition |

Listing 9.7: Data preparation for semantic level data fusion for emotion recognition based on audio-video data.

```
<emotion>
{
for \$es in doc( audio.speech.semantic.emotions.xml)//audio/frame
let \$ev in doc( video.face.semantic.facialExpressions.xml)//video/
frame [
\$es/personID=personID and xs:time(\$ev/time) >= xs:time(\$es/time/
start) and
        xs:time(\$ev/time) <= xs:time(\$es/time/stop) and
        xs:time(\$es/time/start) >= xs:time(\$time_start) and
        xs:time(\$es/time/stop) <= xs:time(\$time_stop)]
        order by \$ev/time
return
<person personID= {\$ev/personID} >
                <video id={$ev/@frameID}>
                        \$ev/time
                        \$ev/emotion
                </video>
                <audio id={\$es/@frameID}>
                        \$es/time
                        \$es/emotion
                </audio >
</person >
}
</emotion>
```

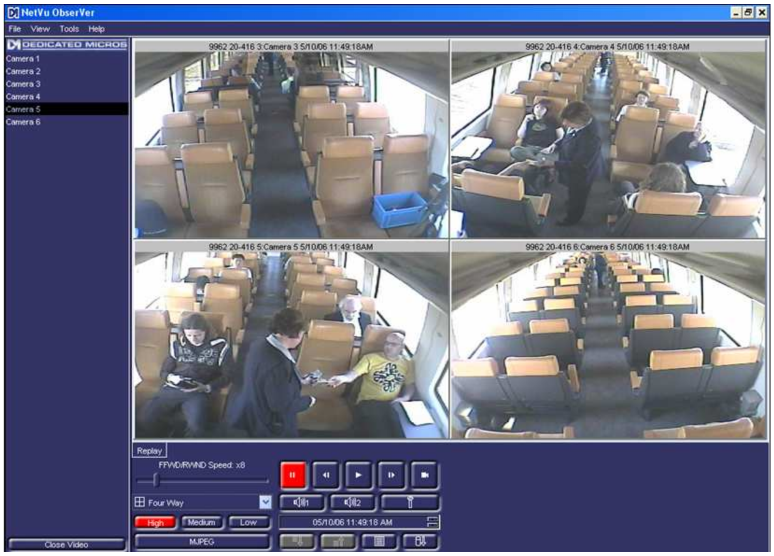


Figure 9.11: User interface for train compartment surveillance operator.

Each module is specialized in handling a specific task. For example, the sound event detection module detects whether there is one shouting. The final module,

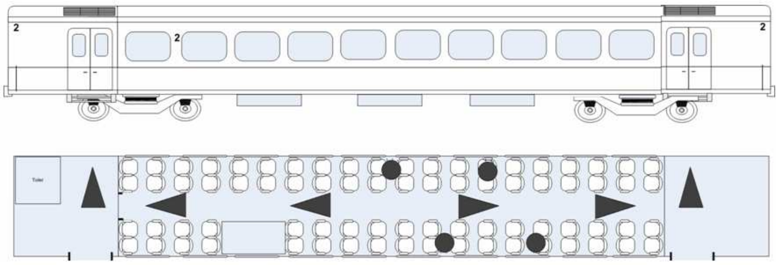


Figure 9.12: Side view of the train (top) and top view of the interior of a BeNeLux train compartment and the location of the sensors (bottom). The circles indicate the position of the omnidirectional microphones and the triangles indicate the cameras and their direction.

called the *aggression detection* module, detects unusual behaviour by fusing the results of different detection modules (figure 9.13). In the application, we have made a distinction between primary and secondary modules. Primary modules collect and process real time data from the hardware sensors. Primary modules are typically feature extraction or object detection algorithms, processing data in the raw data XML and intermediate data XML spaces. These are active as long as the surveillance system is running. The secondary modules operate on the semantic data spaces. They do not necessarily involve real time processing and are typically activated by triggers generated by primary modules.

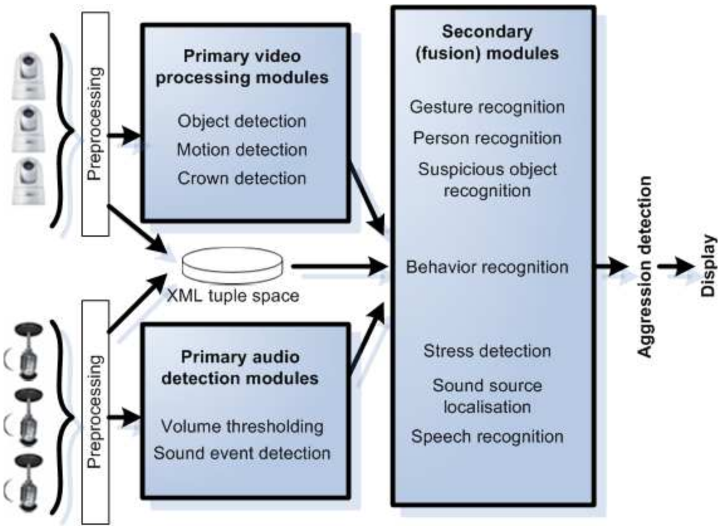


Figure 9.13: Overview of the aggression detection system.

## 9.5 Conclusion

In this chapter, we proposed a framework for automated multimodal surveillance designed to facilitate the communication between different processing components. Based on the description of the framework, we present preliminary results of an ongoing project for building multimodal surveillance applications.

The framework supports shared XML data spaces. This allows for decoupling the communication between processing components in both time and space. A set of software tools monitor the state of processing components, to log different type of events and to check the flow of data given any application context. Although the framework has been designed by taking into consideration the further development of an automatic surveillance application, the architecture may be used as basis for any complex multimodal system which involves many processing components and high data transfer rates.

The surveillance system we have implemented fully complies with the data manipulation requirements of multiple data producers/consumers, where the availability of data is time-dependent and connections might be temporarily interrupted. At the moment, we have running prototypes of the multimodal framework and of the surveillance application implemented in the Java programming language.

# Chapter 10

# Conclusions and future work

Starting with the initial assumption that humans can naturally tell the feelings or attitude simply by listening or by looking at someone's face, the thesis aimed at developing algorithms to automatically process the audio and video data, to extract the essential acoustic structures and to recognize the underlying emotions.

In due course of time, a wide range of algorithms, methods and solutions have been proposed for multimodal emotion recognition. Even more, as a consequence of the recent developments on the topic as well as of the popular adoption of specific algorithms as standard in the field, several research groups have already implemented automatic or semi-automatic system prototypes. In parallel, a few commercial systems have also made their way on still under-developed specialized markets. However, even though each system comes with the promise of achieving rather high performance in well-defined working contexts, at this moment there is still no generally acknowledged system to stand for a solution of all the problems raised in the context. Completely solving for the variability, the subtlety and the complexity associated to the problem of automatic facial expression recognition remain as unsurpassed milestones.

In spite of some tendencies for adopting models, serious issues currently draw from the lack of reference theoretical frameworks. For instance, the missing common methodology for properly describing the states of emotion in speech leads to many contradicting results among the researchers. Another example is the lack of knowledge regarding the dynamics and interdependencies between clues on different audio-visual communication channels. These drawbacks eventually decrease the verifiability and increase the difficulty of defining models in different approaches. Because of the difficulty to obtain recordings of spontaneous emotions, most of the studies on facial expressions, emotion extraction from speech and multimodal recognition of emotions so far, use data sets of simulated emotions. Of course, the purpose of the research is to develop systems that should work in real life settings. According to that theory, even in real life people need to emotionally enact with a certain amount of voluntary control. Still, the question of whether it is feasible or not to study and to develop systems based on acted emotions while thinking at that as at the true complexity of the problem, persists.

In the thesis, we presented the algorithms and methods we used for the development of the fully functional software prototype.

The results obtained from the models that process data from audio and visual channels separately or simultaneously, provide interpretation in terms of clear evidence for the possibility to automate the whole process of multimodal human emotion decoding. As off now, much attention and sustained efforts have taken place on the way of overcoming the essential difficulties concerning the recognition of human facial expressions by automatic systems.

We have shown how to handle the variability regarding face and voice and which visual and acoustic features are necessary to efficiently recognize emotions from separate modalities. We described several methods for segmenting audio-visual data and determined optimal emotion indicators by rigorous validation based on models' performance. Several supervised learning methods have been employed to support the classification of six basic emotions. The results of experiments clearly indicated high performance for facial expression recognition models that consider temporal aspects to the analysis. For multimodal emotion recognition, we presented techniques for synchronizing the audio and video feature streams and we investigated several fusion methods to integrate data at different levels of abstraction. To integrate the functional audio-video processing components, we proposed a multimodal framework and we discussed how to use it in a specific case of multimodal surveillance applications. These tasks constitute intermediate research steps that are essential for the initial goal of developing automatic systems for multimodal emotion recognition. In this way, we used the findings of each of these research tasks for implementing a computer-based system for reading emotions. On the practical side, the algorithmic implementations will benefit from the even increasing capability of the hardware platforms. At this moment, the high demand of computer power shown by the few prototypes of emotion analysis systems can be sustained by only some of the commonly available personal computers. This aspect will gradually change by allowing more and more devices to comply with the data processing requirements.

The automation of human emotion assessment will also make space for the identification of new application domains. Among these, human-computer interaction, psychological research, biometrics, safety and telecommunications are the first to make use of the new technology. Probably some of the most profitable areas will relate to systems aimed to facilitate and even to boost the inter-human communication abilities. A concluding example which is worth mentioning is in the field of support systems for disabled persons.

Eventually, each single research work should be seen as contribution part of the global endeavour to reach our long-term goal according to which we all should benefit from the existence and accessibility of human affect-oriented systems. Such systems will enhance the functions of every day apparatus and will ease to some extent the communication among humans and between humans and machines. The future improvements on the topic will come along with advancements on interdisciplinary research fields. From the point of view of the computational field, the findings in human behaviour-oriented areas will most likely have equal if not higher contribution to the matter. Future work on computer-based reading of facial expressions will possibly make the transition from 2D face processing to 3D appearance analysis and from single-view data streams to the use of multiple cameras. Essential contributions are required in the field of increasing the robustness in terms of orientation, occlusion and illumination. Additionally, more efficient algorithms are mandatory for extending the range of facial expression categories to non-prototypic emotions. In case of emotion extraction from speech data, more robust models should be able to handle continuous speech, speech with overlap produced by several users and speech in different languages. The multimodal data fusion for emotion recognition remains an open challenge as several problems still persist, related to finding optimal features, integration and recognition. At the moment, the implementation of single modality emotion reading systems for real life applications, is feasible and such models have rather high accuracy and robustness. Conversely, reading emotions by computer-based applications is still at the preliminary phase, shows very limited performance and is mostly restricted to the lab environment.

# Bibliography

[1] Jini. *http://jini.org.*

[2] OpenCV: Open Source Computer Vision Library. *http://www.intel.com/technology/computing/opencv.*

[3] Snack Sound Toolkit. *http://www.speech.kth.se/snack.*

[4] Xerces parser. *http://xml.apache.org/xerces-c/pdf.html.*

[5] Rakesh Agrawal, Roberto J. Bayardo, Daniel Gruhl, and Spiros Papadimitriou. Vinci: A service-oriented architecture for rapid development of web applications. In *In Proc. of the 10th International Conference on World Wide Web, Hongkong*, pages 355–365, 2001.

[6] Timo Ahonen, Abdenour Hadid, and Matti Pietikainen. Face recognition with local binary patterns. *Lecture Notes in Computer Science : Computer Vision - ECCV 2004*, 3021:469–481, 2004.

[7] Timo Ahonen, Abdenour Hadid, and Matti Pietikainen. Face description with local binary patterns: application to face recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(12):2037–2041, 2006.

[8] Olusola Olumide Aina, Knut Hartmann, and Thomas Strothotte. Extracting emotion from speech: Towards emotional speech-driven facial animations. In *Smart Graphics, LNCS 2733*, pages 162–171, 2003.

[9] Petar S. Aleksic and Aggelos K. Katsaggelos. Automatic facial expression recognition using facial animation parameters and multi-stream HMMs. *IEEE Transactions on Information Forensics and Security*, pages 3–11, 2006.

[10] Halis Altun, John Shawe-Taylor, and Gokhan Polat. New feature selection frameworks in emotion recognition to evaluate the informative power of speech related features. In *Signal Processing and Its Applications, 2007. ISSPA 2007. 9th International Symposium on*, pages 1–4, 2007.

[11] Noam Amir and Samuel Ron. Towards an automatic classification of emotions in speech. In *In Proceedings of the 5th International Conference of Spoken Language Processing*, volume 3, pages 555–558, 1998.

[12] Pradeep K. Atrey, Namunu C. Maddage, and Mohan S. Kankanhalli. Audio based event detection for multimedia surveillance. In *In IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 813–816, 2006.

[13] Marian Stewart Bartlett, Gwen Littlewort, Mark Frank, Claudia Lainscsek, Ian Fasel, and Javier Movellan. Fully automatic facial action recognition in spontaneous behavior. In *FGR06*, pages 223–230, 2006.

[14] Marian Stewart Bartlett, Gwen Littlewort, Claudia Lainscsek, Ian Fasel, and Javier Movellan. Machine learning methods for fully automatic recognition of facial expressions and facial actions. In *Proc. IEEE Int'l Conf. Systems, Man and Cybernetics*, pages 592–597, October 2004.

[15] Marian Stewart Bartlett, Paul A. Viola, Terrence J. Sejnowski, Beatrice A. Golomb, Joseph C. Hager, and Paul Ekman. Classifying facial actions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(10):974–989, October 1999.

[16] Anton Batliner, Christian Hacker, Stefan Steidl, Elmar Nöth, S. D'Arcy, M. Russell, and M. Wong. "You stupid tin box" - children interacting with the AIBO robot: A cross-linguistic emotional speech corpus. In ELRA, editor, *Proceedings of the 4th International Conference of Language Resources and Evaluation LREC 2004*, 2004.

[17] Anton Batliner, Stefan Steidl, Björn Schuller, Dino Seppi, Thurid Vogt, Laurence Devillers, Laurence Vidrascu, Noam Amir, Loic Kessous, and Vered Aharonson. Combining efforts for improving automatic classification of emotional user states. In *5th Slovenian and 1st international Language Technologies Conference*, 2006.

[18] Matthew J. Beal, Nebojsa Jojic, Ieee Computer Society, and Hagai Attias. A graphical model for audiovisual object tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25:828–836, 2003.

[19] Peter N. Belhumeur, Joao P. Hespanha, and David J. Kriegman. Eigenfaces vs. fisherfaces: recognition using class specific linear projection. In *IEEE Trans. Pattern Analysis and Machine Intelligence*, volume 19, pages 711–720, 1997.

[20] Umesh Bellur and Siddharth Bondre. xSpace: a tuple space for XML & its application in orchestration of web services. In *SAC '06: Proceedings of the 2006 ACM symposium on Applied computing*, pages 766–772, New York, NY, USA, 2006. ACM.

[21] Paul Boersma and David Weenink. Praat: doing phonetics by computer (version 4.3.14) [computer program]. 2005.

[22] Ioan Buciu and Ioannis Pitas. Application of non-negative and local non negative matrix factorization to facial expression recognition. In *ICPR04*, pages I: 288–291, 2004.

[23] Felix Burkhardt, Astrid Paeschke, Miriam Rolfes, Walter Sendlmeier, and Benjamin Weiss. A database of german emotional speech. In *Proceedings Interspeech*, pages 1517–1520, 2005.

[24] Carlos Busso, Zhigang Deng, Serdar Yildirim, Murtaza Bulut, Chul Min Lee, Abe Kazemzadeh, Sungbok Lee, Ulrich Neumann, and Shrikanth Narayanan. Analysis of emotion recognition using facial expressions,

speech and multimodal information. In *ICMI '04: Proceedings of the 6th international conference on Multimodal interfaces*, pages 205–211, New York, NY, USA, 2004. ACM.

[25] Hilary Buxton and Shaogang Gong. Visual surveillance in a dynamic and uncertain world. *Artificial Intelligence*, 78(1-2):431–459, 1995.

[26] George Caridakis, Lori Malatesta, Loic Kessous, Noam Amir, Amaryllis Raouzaiou, and Kostas Karpouzis. Modeling naturalistic affective states via facial and vocal expressions recognition. In *ICMI '06: Proceedings of the 8th international conference on Multimodal interfaces*, pages 146–154, New York, NY, USA, 2006.

[27] Don Chamberlin. XQuery: An XML query language. *IBM Systems Journal*, 41(4):597–615, 2002.

[28] Chih-Chung Chang and Chih-Jen Lin. *LIBSVM: a library for support vector machines*, 2001.

[29] Noël Chateau, Valérie Maffiolo, Thibaut Ehrette, and Christophe d'Alessandro. Modelling the emotional quality of speech in a telecommunication context. Kyoto, Japan, 2002. Advanced Telecommunications Research Institute (ATR), Kyoto, Japan.

[30] Hsiuao-Ying Chen, Chung-Lin Huang, and Chih-Ming Fu. Hybrid-boost learning for multi-pose face detection and facial expression recognition. *Pattern Recognition*, 41(3):1173–1185, 2008.

[31] Roberto Chinnici, Jean-Jacques Moreau, Arthur Ryman, and Sanjiva Weerawarana. Web services description language (WSDL) version 2.0 part 1: Core language. W3C recommendation, World Wide Web Consortium (W3C), 26 Juni 2007.

[32] Chao-Fa Chuanga and Frank Y. Shih. Recognizing facial action units using independent component analysis and support vector machine. *PR*, 39(9):1795–1798, September 2006.

[33] Chloé Clavel, Thibaut Ehrette, and Gaël Richard. Events detection for an audio-based surveillance system. In *Multimedia and Expo, IEEE International Conference on*, pages 1306–1309. IEEE Computer Society, 2005.

[34] Jeffrey Cohn, Takeo Kanade, and Ying-Li Tian. Recognizing action units for facial expression analysis. In *CMU-RI-TR*, 1999.

[35] Timothy F. Cootes, Christopher J. Taylor, David H. Cooper, and Jim Graham. Active shape models: their training and application. *Computer Vision and Image Understanding*, 61(1):38–59, 1995.

[36] Roddy Cowie and Ellen Douglas-Cowie. Automatic statistical analysis of the signal and prosodic signs ofemotion in speech. In *Spoken Language, 1996. ICSLP 96. Proceedings., Fourth International Conference on*, volume 3, pages 1989–199, 1996.

[37] Frédéric Cupillard. Video understanding for metro surveillance. In *IEEE International Conference on Networking, Sensing and Control*, 2004.

[38] Taner Danisman and Adil Alpkocak. Emotion classification of audio signals using ensemble of support vector machines. In *Perception in Multimodal Dialogue Systems*, pages 205–216. Springer Berlin / Heidelberg, 2008.

[39] Dragoş Datcu and Léon J.M. Rothkrantz. Automatic recognition of facial expressions using Bayesian belief networks. In *Proceedings of IEEE SMC 2004*, pages 2209–2214, 2004.

[40] Dragoş Datcu and Léon J.M. Rothkrantz. A multimodal workbench for automatic surveillance. In *Proceedings of Euromedia 2004*, pages 108–112, 2004.

[41] Dragoş Datcu and Léon J.M. Rothkrantz. Facial expression recognition with relevance vector machines. In *Proceedings of IEEE International Conference on Multimedia & Expo*, pages 193–196, 2005.

[42] Dragoş Datcu and Léon J.M. Rothkrantz. The recognition of emotions from speech using GentleBoost classifier. In *Proceedings of CompSysTech 2006*, june 2006.

[43] Dragoş Datcu and Léon J.M. Rothkrantz. Multimodal workbench for human emotion recognition. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR'07*, Minneapolis, Minnesota, USA, 2007. Software Demo.

[44] Dragoş Datcu and Léon J.M. Rothkrantz. The use of active appearance model for facial expression recognition in crisis environments. In *Proceedings of ISCRAM 2007*, pages 515–524, 2007.

[45] Dragoş Datcu and Léon J.M. Rothkrantz. Automatic bi-modal emotion recognition system based on fusion of facial expressions and emotion extraction from speech. In *IEEE Face and Gesture Conference FG2008*, 2008.

[46] Dragoş Datcu and Léon J.M. Rothkrantz. Semantic audio-visual data fusion for automatic emotion recognition. In *Proceedings of Euromedia 2008*, pages 58–65, 2008.

[47] Dragoş Datcu, Zhenke Yang, and Léon J.M. Rothkrantz. Multimodal workbench for automatic surveillance applications. In *Multimodal surveillance: Sensors, Algorithms, and Systems*, chapter 14, pages 311–338. 2007.

[48] Edwin J. de Jong and Léon J.M. Rothkrantz. FED - an online facial expression dictionary. In *Proceedings of Euromedia 2004*, April 2004.

[49] Chathura R. de Silva, Surendra Ranganath, and Liyanage C. de Silva. Cloud basis function neural network: A modified RBF network architecture for holistic facial expression recognition. *PR*, 41(4):1241–1253, April 2008.

[50] Frank Dellaert, Thomas Polzin, and Alex Waibel. Recognizing emotion in speech. In *Proceedings of the Conference on Spoken Language - ICSLP*, pages 1970–1973, 1996.

[51] Steve Derose, Eve Maler, and David Orchard. XML path language (XPath). Technical report, 2001.

[52] Fadi Dornaika and Franck Davoine. Simultaneous facial action tracking and expression recognition in the presence of head motion. *International Journal of Computer Vision*, 76(3):257–281, March 2008.

[53] Séverine Dubuisson, Franck Davoine, and Mylène Masson. A solution for facial expression representation and recognition. *Signal Processing: Image Communication*, 17(9):657–673, October 2002.

[54] Carmen J. Duthoit, Tamara Sztynda, Sara K. L. Lal, Budi T. Jap, and Johnson I. Agbinya. Optical flow image analysis of facial expressions of human emotion: forensic applications. In *e-Forensics '08*, pages 1–6, ICST, Brussels, Belgium, Belgium, 2008. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).

[55] Gareth J. Edwards, Christopher J. Taylor, and Timothy F. Cootes. Interpreting face images using active appearance models. In *FG '98: Proceedings of the 3rd. International Conference on Face & Gesture Recognition*, page 300, Washington, DC, USA, 1998. IEEE Computer Society.

[56] Paul Ekman and Wallace V. Friesen. Constants across cultures in the face and emotion. *Journal of Personality and Social Psychology*, 17(2):124–129, 1971.

[57] Paul Ekman and Wallace V. Friesen. *Facial action coding system: investigator's guide*. Consulting Psychologists Press, Palo Alto, 1978.

[58] Paul Ekman, Wallace V. Friesen, and Joseph C. Hager. *Facial Action Coding System*. A Human Face, Salt Lake City, 2002.

[59] Paul Ekman, Wallace V. Friesen, and Joseph C. Hager. *Facial action coding system. The manual (on CD)*. Research Nexus division of Network Information Research Corporation, 2002.

[60] I. S. Engberg and A. V. Hansen. Documentation of the Danish Emotional Speech Database (DES). Technical report, Center for Person Kommunikation, Denmark, 1996.

[61] Irfan A. Essa. Coding, analysis, interpretation, and recognition of facial expressions. *PAMI*, 19(7):757–763, July 1997.

[62] Irfan A. Essa, Trevor Darrell, and Alex Pentland. Tracking facial motion. In *In Proceedings of the Workshop on Motion of Nonrigid and Articulated Objects*, pages 36–42. IEEE Computer Society, 1994.

[63] David C. Fallside. XML Schema. In *Technical report*. World Wide Web Consortium, 2000.

[64] Beat Fasel and Juergen Luettin. Recognition of asymmetric facial action unit activities and intensities. In *Proceedings of International Conference on Pattern Recognition (ICPR 2000), Barcelona, Spain*, 2000.

[65] Beat Fasel and Juergen Luettin. Automatic facial expression analysis: a survey. *PR*, 36(1):259–275, January 2003.

[66] Winfried A. Fellenz, John G. Taylor, Roddy Cowie, Ellen Douglas-Cowie, Frédéric Piat, Stefanos D. Kollias, Christos Orovas, and Bruno Apolloni. On emotion recognition of faces and of speech using neuralnetworks, fuzzy logic and the assess system. In *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN'00*, 2000.

[67] Xiaoyi Feng. Facial expression recognition based on local binary patterns and coarse-to-fine classification. In *Computer and Information Technology, 2004. The Fourth International Conference on*, pages 178–183, 2004.

[68] Xiaoyi Feng, Jie Cui, Matti Pietikinen, and Abdenour Hadid. Real time facial expression recognition using local binary patterns and linear programming. 3789:328–336, 2005.

[69] Xiaoyi Feng, Abdenour Hadid, and Matti Pietikinen. A coarse-to-fine classification scheme for facial expression recognition. *Image Analysis and Recognition*, pages 668–675, 2004.

[70] Xiaoyi Feng, Matti Pietikäinen, and Abdenour Hadid. Facial expression recognition with local binary patterns and linear programming. 2005. Pattern Recognition and Image Analysis 15(2): 546-548.

[71] Xiaoyi Feng, Matti Pietikinen, and Abdenour Hadid. Facial expression recognition with local binary patterns and linear programming. 2004. Proc. of the 7th International Conference on Pattern Recognition and Image Analysis: New Information Technologies (PRIA-7-2004), St. Petersburg, Russia, 666-669.

[72] Siska Fitrianie, Ronald Poppe, Trung Bui, Alin G. Chitu, Dragos Datcu, Ramon Dor, D.H.W. Hofs, Pascal Wiggers, Don J.M. Willems, Mannes Poel, Lèon J.M. Rothkrantz, Louis G. Vuurpijl, and Job Zwiers. A multimodal human-computer interaction framework for research into crisis management. In *Proceedings of the fourth international conference on information systems for crisis management ISCRAM'07*, pages 149–158. Academic and Scientific Publishers NV., 2007.

[73] Marcus Fontoura, Tobin J. Lehman, Dwayne Nelson, Thomas Truong, and Yuhong Xiong. TSpaces services suite: Automating the development and management of Web services. In *WWW (Alternate Paper Tracks)*, 2003.

[74] Gian Luca Foresti, Christian Micheloni, Lauro Snidaro, Paolo Remagnino, and Tim Ellis. Active video-based surveillance system: the low-level image and video processing techniques needed for implementation. *Signal Processing Magazine, IEEE*, 22(2):25–37, March 2005.

[75] Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of online learning and an application to boosting. *Journal of Computer and System Science*, 55:119–139, 1997.

[76] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Additive logistic regression: a statistical view of boosting. *Annals of Statistics*, 28(2), 2000.

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

[78] Hyoun-Joo Go, Keun-Chang Kwak, Dae-Jong Lee, and Myung-Geun Chun. Emotion recognition from the facial image and speech signal. In *SICE 2003 Annual Conference*, volume 3, pages 2890–2895, August 2003.

[79] Richard S. Goldhor. Recognition of environmental sounds. In *Proc. of IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP '93)*, volume 1, pages 149–152, 1993.

[80] Colin R. Goodall. Procrustes methods in the statistical analysis of shape. *Journal of the Royal Statistical Society*, pages 285–339, 1991.

[81] Tommaso Gritti, Caifeng Shan, Vincent Jeanne, and Ralph Braspenning. Local features based facial expression recognition with face registration errors. In *Proceedings of the 8th IEEE Int'l Conference on Automatic Face and Gesture Recognition - FG2008*, 2008.

[82] Hai-Song Gu and Qi-Ang Ji. An automated face reader for fatigue detection. In *AFGR04*, pages 111–116, 2004.

[83] Hai-Song Gu and Qi-Ang Ji. Facial event classification with task oriented dynamic Bayesian network. In *CVPR04*, pages II: 870–875, 2004.

[84] Guodong Guo and Charles R. Dyer. Learning from examples in the small sample case: Face expression recognition. *SMC-B*, 35(3):477–488, June 2005.

[85] Asaad Hakeem and Mubarak Shah. Multiple agent event detection and representation in videos. In *Proc. of the 20th National Conference on Artificial Intelligence*, pages 89–94, Pittsburgh, Pennsylvania, July 2005.

[86] Zakia Hammal, Laurent Couvreur, Alice Caplier, and Michèle Rombaut. An approach based on the fusion of facial deformations using the transferable belief model. *International Journal of Approximate Reasoning*, 46(3):542–567, December 2007.

[87] Meng-Ju Han, Jing-Huai Hsu, Kai-Tai Song, and Fuh-Yu Chang. A new information fusion method for bimodal robotic emotion recognition. *JCP*, 3(7):39–47, 2008.

[88] Aki Harma, Martin F. McKinney, and Janto Skowronek. Automatic surveillance of the acoustic activity in our living environment. *Multimedia and Expo, 2005. ICME 2005. IEEE International Conference on*, July 2005.

[89] Lianghua He, Jianzhong Zhou, Die Hu, Cairong Zou, and Li Zhao. Boosted independent features for facial expression recognition. In *Advances in neural networks - ISNN 2005*, pages 137–146. Springer-Verlag Berlin Heidelberg, 2005.

[90] Lianghua He, Cairong Zou, Li Zhao, and Die Hu. An enhanced LBP feature based on facial expression recognition. In *Engineering in Medicine and Biology IEEE-EMBS 2005*, pages 3300–3303, 2005.

[91] Stefan Hoch, Frank Althoff, Gregor McGlaun, and Gerhard Rigoll. Bimodal fusion of emotional data in an automotive environment. volume 2, pages ii/1085–ii/1088 Vol. 2, March 2005.

[92] Jesse Hoey and James J. Little. Value directed learning of gestures and facial displays. In *CVPR04*, pages II: 1026–1033, 2004.

[93] Somboon Hongeng, Francois Brmond, and Ramakant Nevatia. Bayesian framework for video surveillance application. In *Proceedings of the 15th International Conference on Pattern Recognition*, pages 164–170, 2000.

[94] Changbo Hu, Ya Chang, Rogerio Feris, and Matthew Turk. Manifold based analysis of facial expression. *IVC*, 24(6):605–614, June 2006.

[95] Chung-Lin Huang and Yu-Ming Huang. Facial expression recognition using model-based feature extraction and action parameter(s) classification. *JVCIR*, 8:278–290, 1997.

[96] Richard Huber, Anton Batliner, Jan Buckow, Elmar Nöth, Volker Warnke, and Heinrich Niemann. Recognition of emotion in a realistic dialogue scenario. In *in Proc. Int. Conf. on Spoken Language Processing*, pages 665–668, 2000.

[97] Spiros Ioannou, Amaryllis Raouzaiou, Kostas Karpouzis, and Stefanos Kollias. Adaptation of facial feature extraction and rule generation in emotion-analysis systems. International Joint Conference on Neural Networks (IJCNN 2004), Budapest,Hungary, July 2004., 2004.

[98] Yuri Ivanov, Chris Stauffer, Aaron Bobick, and W. E. L. Grimson. Video surveillance of interactions. In *CVPR Workshop on Visual Surveillance, Fort Collins*, pages 82–89. IEEE, 1999.

[99] Omar Javed, Zeeshan Rasheed, Orkun Alatas, and Mubarak Shah. Knightm: A real time surveillance system for multiple overlapping and non-overlapping cameras. In *In Proceedings of ICME*, pages 649–652, 2003.

[100] Dan-Ning Jiang and Lian-Hong Cai. Speech emotion classification with the combination of statistic features and temporal features. *Multimedia and Expo, 2004. ICME '04. 2004 IEEE International Conference on*, 3:1967–1970 Vol.3, June 2004.

[101] Jong-Tae Joo, Sang-Wook Seo, Kwang-Eun Ko, and Kwee-Bo Sim. Emotion recognition method based on multimodal sensor fusion algorithm. In *ISIS'07*, 2007.

[102] Simon Josefsson. Rfc3548 - the base16, base32, and base64 data encodings, july 2003.

[103] Takeo Kanade, Jeffrey F. Cohn, and Yingli Tian. Comprehensive database for facial expression analysis. In *Proceedings of the Fourth IEEE International Conference on Automatic Face and Gesture Recognition (FG'00)*, pages 46–53, 2000.

[104] Deepali Khushraj, Ora Lassila, and Tim Finin. sTuples: Semantic tuple spaces. In *First Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services (MobiQuitous04)"*, pages 267–277, August 2004.

[105] Tae-Kyun Kim, Sung-Uk Lee, Jong Ha Lee, Seok-Cheol Kee, and Sang Ryong Kim. Integrated approach of multiple face detection for video surveillance. In *Proc. of the International Conference of Pattern Recognition (ICPR2002)*, pages 394–397, 2002.

[106] H. Kobayashi and F. Hara. Recognition of six basic facial expressions and their strength by neural network. In *IEEE International Workshop on Robot and Human Communication*, pages 381–386, 1992.

[107] Oh-Wook Kwon, Kwokleung Chan, Jiucang Hao, and Te-Won Lee. Emotion recognition by speech signals. In *Interspeech 2003*, pages 125–128, 2003.

[108] Kuang-Chih Lee, Jeffrey Ho, and David J. Kriegman. Acquiring linear subspaces for face recognition under variable lighting. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(5):684–698, 2005.

[109] Yang Li and Yunxin Zhao. Recognizing emotions in speech using short-term and long-term features. In *Proc. Int. Conf. on Spoken Language Processing*, volume 6, pages 2255–2258, 1998.

[110] Ying li Tian, Takeo Kanade, and Jeffrey F. Cohn. Evaluation of Gabor-wavelet-based facial action unit recognition in image sequences of increasing complexity. In *AFGR02*, pages 218–223, 2002.

[111] Shu Liao, Wei Fan, A.C.S. Chung, and Dit-Yan Yeung. Facial expression recognition using advanced local binary patterns, Tsallis entropies and global appearance features. pages 665–668, 2006.

[112] Jenn-Jier J. Lien, Takeo Kanade, Jeffrey Cohn, C.C. Li, and Adena J. Zlochower. Subtly different facial expression recognition and expression intensity estimation. In *CVPR98*, pages 853–859, 1998.

[113] Jenn-Jier James Lien, Takeo Kanade, Jeffrey F. Cohn, and Ching-Chung Li. Automated facial expression recognition based on FACS action units. In *AFGR98*, pages 390–395, 1998.

[114] Jenn-Jier James Lien, Takeo Kanade, Jeffrey F. Cohn, and Ching-Chung Li. Detection, tracking, and classification of action units in facial expression. *RAS*, 31:131–146, 2000.

[115] Rainer Lienhart and Jochen Maydt. An extended set of Haar-like features for rapid object detection. volume 1, pages I–900–I–903 vol.1, 2002.

[116] Gwen Littlewort, Marian Stewart Bartlett, Ian Fasel, Joshua Susskind, and Javier Movellan. Dynamics of facial expression extracted automatically from video. *IVC*, 24(6):615–625, June 2006.

[117] Jia Liu, Chun Chen, Jiajun Bu, Mingyu You, and Jianhua Tao. Speech emotion recognition based on a fusion of all-class and pairwise-class feature selection. In *ICCS07*.

[118] Jia Liu, Chun Chen, Jiajun Bu, Mingyu You, and Jianhua Tao. Speech emotion recognition using an enhanced co-training algorithm. In *IEEE ICME07*.

[119] Bruce D. Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision (DARPA). In *Proceedings of the 1981 DARPA Image Understanding Workshop*, pages 121–130, April 1981.

[120] Simon Lucey, Iain Matthews, Changbo Hu, Zara Ambadar, Fernando de la Torre, and Jeffrey Cohn. Aam derived face representations for robust facial action recognition. In *FGR06*, pages 155–162, 2006.

[121] Daniel Lundqvist, Anders Flykt, and Arne Öhman. The Karolinska directed emotional faces - KDEF. In *CD ROM from Department of Clinical Neuroscience, Psychology section*. Karolinska Institutet, 1998.

[122] Michael Lyons, Shigeru Akamatsu, Miyuki Kamachi, and Jiro Gyoba. Coding facial expressions with Gabor wavelets. In *Third IEEE International Conference on Automatic Face and Gesture Recognition*, pages 200–205. IEEE Computer Society, 1998.

[123] Muharram Mansoorizadeh and Nasrollah M. Charkari. Bimodal person-dependent emotion recognition comparison of feature level and decision level information fusion. In *PETRA '08: Proceedings of the 1st international conference on PErvasive Technologies Related to Assistive Environments*, pages 1–4, New York, NY, USA, 2008. ACM.

[124] Sebastien Marcel, Jean Keomany, and Yann Rodriguez. Robust-to-illumination face localisation using shape models and local binary patterns. Technical report, IDIAP, 2006.

[125] Christian Martin, Uwe Werner, and Horst-Michael Gross. A real-time facial expression recognition system based on active appearance models using gray images and edge images. In *Proceedings of the 8th IEEE Int'l Conference on Automatic Face and Gesture Recognition - FG2008*, 2008.

[126] Olivier Martin, Irene Kotsia, Benoit Macq, and Ioannis Pita. The eNTERFACE'05 Audio-Visual Emotion Database. In *22nd International Conference on Data Engineering Workshops. ICDEW'06*, page 8, 2006.

[127] Aleix Martinez and Robert Benavente. The AR face database. In *CVC Technical Report (24)*, June 1998.

[128] Cecilia Mascolo, Licia Capra, Stefanos Zachariadis, and Wolfgang Emmerich. XMIDDLE: A data-sharing middleware for mobile computing. *Wireless Personal Communication Journal*, 21(1):77–103, 2002.

[129] Kenji Mase. An application of optical flow - extraction of facial expression. In *MVA*, pages 195–198, 1990.

[130] Stephen J. McKenna, Shaogang Gong, Rolf P. Würtz, Jonathan Tanner, and Daniel Banin. Tracking facial feature points with Gabor wavelets and shape models. In *AVBPA97*, pages 35–42, London, UK, 1997. Springer-Verlag.

[131] Albert Mehrabian. Communication without words. *Psychology today*, 2(4):53–56, 1968.

[132] Hong Meng, Johannes Pittermann, Angela Pittermann, and Wolfgang Minker. Combined speech-emotion recognition for spoken human-computer interfaces. In *Signal Processing and Communications, 2007. ICSPC 2007. IEEE International Conference on*, pages 1179–1182, Nov. 2007.

[133] Stephen Milborrow and Fred Nicolls. Locating facial features with an extended active shape model. In *ECCV '08: Proceedings of the 10th European Conference on Computer Vision*, pages 504–513, Berlin, Heidelberg, 2008. Springer-Verlag.

[134] Stephen Moore and Richard Bowden. Automatic facial expression recognition using boosted discriminatory classifiers. In *AMFG07*, pages 71–83. Springer-Verlag Berlin Heidelberg, 2007.

[135] Tsuyoshi Moriyama, Jing Xiao, and Jeffrey F. Cohn. Meticulously detailed eye model and its application to analysis of facial image. In *Systems, Man and Cybernetics, 2004 IEEE International Conference on*, volume 1, pages 629–634, October 2004.

[136] Iain R. Murray and John L. Arnott. Toward the simulation of emotion in synthetic speech: A review of the literature on human vocal emotion. *The Journal of the Acoustical Society of America*, 93(2):1097–1108, 1993.

[137] J. Nicholson, K. Takahashi, and R. Nakatsu. Emotion recognition in speech using neural networks. *Neural Computing & Applications*, 9:290–296, 1999.

[138] Robert Niese, Ayoub Al-Hamadi, Axel Panning, and Bernd Michaelis. Real-time capable method for facial expression recognition in color and stereo vision. In *ICCSA07*, pages 397–408, 2007.

[139] Timo Ojala, Matti Pietikäinen, and David Harwood. A comparative study of texture measures with classification based on feature distributions. *PR*, 29(1):51–59, January 1996.

[140] Timo Ojala, Matti Pietikäinen, and Topi Mäenpää. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):971–987, 2002.

[141] Takahiro Otsuka and Jun Ohya. Recognizing multiple persons' facial expressions using HMM based on automatic extraction of significant frames from image sequences. In *ICIP97*, pages II: 546–549, 1997.

[142] Takahiro Otsuka and Jun Ohya. Recognizing abruptly changing facial expressions from time-sequential face images. In *CVPR98*, pages 808–813, 1998.

[143] Marco Paleari and Benoit Huet. Toward emotion indexing of multimedia excerpts. In *Content-Based Multimedia Indexing, 2008. CBMI 2008. International Workshop on*, pages 425–432, June 2008.

[144] Marco Paleari and Christine L. Lisetti. Toward multimodal fusion of affective cues. In *Proceedings of the 1st ACM international workshop on Human-centered multimedia*, pages 99–108, 2006.

[145] Maja Pantic and Léon J.M. Rothkrantz. Automatic analysis of facial expressions: The state of the art. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22:1424–1445, 2000.

[146] Maja Pantic and Léon J.M. Rothkrantz. Self-adaptive expert system for facial expression analysis. *Systems, Man, and Cybernetics, 2000 IEEE International Conference on*, pages 73–79, 2000.

[147] Montse Pardàs and Antonio Bonafonte. Facial animation parameters extraction and expression recognition using hidden Markov models. *SP:IC*, 17(9):675–688, October 2002.

[148] Daniel Gatica Perez, G. Lathoud, L. McCowan, J.M. Odobez, and D. Moore. Audio-visual speaker tracking with importance particle filters. In *Proc. of the International Conference on Image Processing (ICIP 2003)*, volume 3, pages 25–28, 2003.

[149] Rosalind Picard. Affective computing. Technical Report 321, MIT Media Laboratory, Perceptual Computing Section, Cambridge, Massachusetts, November 1995.

[150] Shankar R. Ponnekanti, Brad Johanson, Emre Kiciman, and Armando Fox. Portability, extensibility and robustness in iROS. In *PERCOM '03: Proceedings of the First IEEE International Conference on Pervasive Computing and Communications*, page 11, Washington, DC, USA, 2003. IEEE Computer Society.

[151] Léon Rothkrantz, Dragoş Datcu, and Neil Absil. Multimodal affect detection of car drivers. *Neural Network World*, 19(3):293–305, 2009.

[152] Léon J.M. Rothkrantz, Pascal Wiggers, Jan-Willem A. van Wees, and Robert J. van Vark. Voice stress analysis. In *TSD 2004 : text, speech and dialogue*, pages 449–456, 2004.

[153] Benny Lo S.A. Velastin, Maria Alicia Vicencio-Silva and Louhadi Khoudour. A distributed surveillance system for improving security in public transport networks. *Special Issue on Remote Surveillance Measurement and Control*, 35(8):209–213, 2002.

[154] Yunus Saatci and Christopher Town. Cascaded classification of gender and facial expression using active appearance models. In *FG'06*, pages 393–400, Los Alamitos, CA, USA, 2006. IEEE Computer Society.

[155] Marc Schröder. Experimental study of affect bursts. *Speech Communication*, 40(1-2):99–116, 2003.

[156] Björn Schuller, Stephan Reiter, and Gerhard Rigoll. Evolutionary feature generation in speech emotion recognition. In *ICME06*.

[157] Björn Schuller, Bogdan Vlasenko, Ricardo Minguez, Gerhard Rigoll, and Andreas Wendemuth. Comparing one and two-stage acoustic modeling in the recognition of emotion in speech. In *Automatic Speech Recognition & Understanding, 2007. ASRU. IEEE Workshop on*, pages 596–600, 2007.

[158] Nicu Sebe, Ira Cohen, Theo Gevers, and Thomas S. Huang. Emotion recognition based on joint visual and audio cues. *Pattern Recognition, International Conference on*, 1:1136–1139, 2006.

[159] Mohammad H. Sedaaghi, Constantine Kotropoulos, and Dimitrios Ververidis. Using adaptive genetic algorithms to improve speech emotion recognition. In *Multimedia Signal Processing, 2007. MMSP 2007. IEEE 9th Workshop on*, pages 461–464, 2007.

[160] Caifeng Shan, Shaogang Gong, and Peter W. McOwan. Robust facial expression recognition using local binary patterns. *Image Processing, 2005. ICIP 2005. IEEE International Conference on*, 2:II–370–3, Sept. 2005.

[161] Caifeng Shan, Shaogang Gong, and Peter W. McOwan. Capturing correlations among facial parts for facial expression analysis. In *BMVC07*, 2007.

[162] Caifeng Shan, Shaogang Gong, and Peter W. McOwan. Facial expression recognition based on local binary patterns: A comprehensive study. *Image and Vision Computing*, 2008.

[163] Caifeng Shan and Tommaso Gritti. Learning discriminative LBP-histogram bins for facial expression recognition. In *BMVC08*, 2008.

[164] Jeongho Shin, Sangjin Kim, SangKyu Kang, Seongwon Lee, Joon Ki Paik, Besma R. Abidi, and Mongi A. Abidi. Optical flow-based real-time object tracking using non-prior training active feature model. *Real-Time Imaging*, 11(3):204–218, 2005.

[165] Terence Sim, Simon Baker, and Maan Bsat. The cmu pose, illumination, and expression (pie) database. In *Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition*, May 2002.

[166] Mingli Song, Jiajun Bu, Chun Chen, and Nan Li. Audio-visual based emotion recognition, a new approach. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 2:1020–1025, 2004.

[167] Sascha Spors, Rudolf Rabenstein, and Norbert Strobel. Joint audio-video object tracking. In *IEEE International Conference on Image Processing (ICIP)*, pages 393–396, 2001.

[168] Ioanna-Ourania Stathopoulou and George A. Tsihrintzis. An improved neural-network-based face detection and facial expression classification system. In *Systems, Man and Cybernetics, 2004 IEEE International Conference on*, volume 1, pages 666–671, 2004.

[169] Mikkel B. Stegmann. The AAM-API: An open source active appearance model implementation. In *Proceedings of Medical Image Computing and Computer-Assisted Intervention - MICCAI*, pages 951–952. Springer, 2003.

[170] Young suk Shin. Facial expression recognition in various internal states using independent component analysis. In *AMDO06*, pages 291–299, 2006.

[171] Yi Sun, Michael Reale, and Lijun Yin. Recognizing partial facial action units based on 3D dynamic range data for facial expression recognition. In *Proceedings of the 8th IEEE Int'l Conference on Automatic Face and Gesture Recognition - FG2008*, pages 1–8, 2008.

[172] Matti Taini, Guoying Zhao, Stan Z. Li, and Matti Pietikinen. Facial expression recognition from near-infrared video sequences. In *Proc. 19th International Conference on Pattern Recognition (ICPR 2008), Tampa, FL, accepted*, 2008.

[173] Fotios Talantzis, Aristodemos Pnevmatikakis, and Lazaros C. Polymenakos. Real time audio-visual person tracking. *Multimedia Signal Processing, 2006 IEEE 8th Workshop on*, pages 243–247, October 2006.

[174] Hai Tao, Harpreet S. Sawhney, and Rakesh Kumar. Object tracking with Bayesian estimation of dynamic layer representations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(1):75–89, 2002.

[175] Patrick Thompson. Ruple: an XML space implementation, 2002.

[176] Ying-Li Tian, Takeo Kanade, and Jeffrey Cohn. Recognizing lower face action units in facial expression. In *AFGR00*, pages 484–490, 2000.

[177] Ying-Li Tian, Takeo Kanade, and Jeffrey Cohn. Recognizing upper face action units for facial expression analysis. In *CVPR00*, pages I: 294–301, 2000.

[178] Ying-Li Tian, Takeo Kanade, and Jeffrey Cohn. Recognizing action units for facial expression analysis. *PAMI*, 23(2):97–115, February 2001.

[179] Ying-Li Tian, Takeo Kanade, and Jeffrey Cohn. Recognizing facial actions by combining geometric features and regional appearance patterns. In *CMU-RI-TR*, 2001.

[180] Michael E. Tipping. The relevance vector machine. In Sara A. Solla, T. K. Leen, and K. R. Müller, editors, *Advances in Neural Information Processing Systems*, volume 12. MIT Press, 2000.

[181] Robert Tolksdorf, Franziska Liebsch, and Duc Minh Nguyen. XMLSpaces.NET: An extensible tuplespace as XML middleware. Technical report, 2003.

[182] Yan Tong, Wenhui Liao, and Qiang Ji. Facial action unit recognition by exploiting their dynamic and semantic relationships. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(10):1683–1699, October 2007.

[183] Yan Tong, Wenhui Liao, Zheng Xue, and Qiang Ji. A unified probabilistic framework for facial activity modeling and understanding. In *CVPR*, 2007.

[184] Hans van Kuilenburg, Marco Wiering, and Marten den Uyl. A model base method for automatic facial expression recognition. In *Lecture Notes in Computer Science*, volume 3720, pages 194–205. Publisher Springer Berlin / Heidelberg, 2005.

[185] Vladimir N. Vapnik. *The nature of statistical learning theory.* Springer-Verlag New York, Inc., New York, NY, USA, 1995.

[186] Dimitrios Ververidis and Constantine Kotropoulos. Emotional speech recognition: Resources, features, and methods. *Speech Communication*, 48(9):1162–1181, September 2006.

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

[188] Paul Viola and Michael J. Jones. Robust real-time face detection. *IJCV: International Journal of Computer Vision*, 57(2):137–154, May 2004.

[189] Bogdan Vlasenko, Björn Schuller, Andreas Wendemuth, and Gerhard Rigoll. On the influence of phonetic content variation for acoustic emotion recognition. In *PIT '08: Proceedings of the 4th IEEE tutorial and research workshop on Perception and Interactive Technologies for Speech-Based Systems*, pages 217–220, Berlin, Heidelberg, 2008. Springer-Verlag.

[190] Bogdan Vlasenko and Andreas Wendemuth. Tuning hidden Markov model for speech emotion recognition. In *Fortschritte der akustik*, pages 317–318, 2007.

[191] Bogdan Vlasenko and Andreas Wendemuth. Tuning hidden Markov model for speech emotion recognition. In *DAGA 2007. 33rd German Annual Conference on Acoustics*, 2007.

[192] Annelieke Vries-Baayens. *CAD product data exchange: conversions for curves and surfaces, PhD. thesis.* Delft University Press, The Netherlands, 1991.

[193] Johannes Wagner, Thurid Vogt, and Elisabeth André. A systematic comparison of different HMM designs for emotion recognition from acted and spontaneous speech. In *ACII07.*

[194] Yongjin Wang and Ling Guan. Recognizing human emotion from audiovisual information. *Acoustics, Speech, and Signal Processing, 2005. Proceedings. ICASSP'05. IEEE International Conference on*, 2:ii/1125–ii/1128 Vol. 2, March 2005.

[195] Xue wen Chen and Thomas Huang. Facial expression recognition: A clustering-based approach. *Pattern Recognition Letters*, 24(9-10):1295–1302, June 2003.

[196] Jacob Whitehill and Christian W. Omlin. Haar features for FACS AU recognition. In *FGR06*, pages 97–101, 2006.

[197] Matthias Wimmer, Björn Schuller, Dejan Arsic, Bernd Radig, and Gerhard Rigoll. Low-level fusion of audio and video feature for multi-modal emotion recognition. In *3rd International Conference on Computer Vision Theory and Applications. VISAPP*, volume 2, pages 145–151, Madeira, Portugal, January 2008.

[198] Wai Shung Wong, William Chan, Dragoş Datcu, and Léon J.M. Rothkrantz. Using a sparse learning relevance vector machine in facial expression recognition. In *Proceedings of Euromedia 2006*, pages 33–37, 2006.

[199] Lauren Wood, Arnaud Le Hors, Vidur Apparao, Steven Byrne, Mike Champion, Scott Isaacs, Ian Jacobs, Gavin Thomas Nicol, Jonathan Robie, Robert Sutor, and Chris Wilson. Document object model (DOM) level 1 specification (second edition). World Wide Web Consortium, Working Draft, September 2000.

[200] Qiang Ji Yan Tong, Wenhui Liao. Inferring facial action units with causal relations. In *CVPR06*, pages II: 1623–1630, 2006.

[201] Peng Yang, Qingshan Liu, and Dimitris N. Metaxas. Boosting coded dynamic features for facial action units and facial expression recognition. In *CVPR07*, pages 1–6, 2007.

[202] Zhenke Yang, Siska Fitrianie, Dragoş Datcu, and Léon J.M. Rothkrantz. An aggression detection system for the train compartment. In *Advances in artificial intelligence for privacy protection and security*, volume 1, chapter 11, pages 249–286. World Scientific Publishing Co. Pte. Ltd., 2009.

[203] Larry S. Davis Yaser Yacoob. Recognizing human facial expressions from long image sequences using optical flow. *PAMI*, 18(6):636–642, June 1996.

[204] Lijun Yin, Johnny Loi, and Wei Xiong. Facial expression analysis based on enhanced texture and topographical structure. In *Systems, Man and Cybernetics, 2004 IEEE International Conference on*, volume 1, pages 586–591, 2004.

[205] M. Yoneyama, Y. Iwano, A. Ohtake, and K. Shirai. Facial expressions recognition using discrete Hopfield neural networks. In *ICIP97*, pages I: 117–120, 1997.

[206] Mingyu You, Chun Chen, Jiajun Bu, Jia Liu, and Jianhua Tao. Emotion recognition from noisy speech. In *IEEE ICME06*, pages 1653–1656, 2006.

[207] S. Young, G. Evermann, M. Gales, T. Hain, D. Kershaw, X. (A.) Liu, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, and P.Woodland. *The HTK book (for HTK Version 3.4)*. 2006.

[208] Chen Yu. Detecting user engagement in everyday conversations. In *In Proc. 8th Int'l Conf. on Spoken Language Processing (ICSLP '04)*, pages 1–6, 2004.

[209] Jiangang Yu and Bir Bhanu. Evolutionary feature synthesis for facial expression recognition. *PRL*, 27(11):1289–1298, August 2006.

[210] Zhihong Zeng, Yuxiao Hu, Glenn I. Roisman, Zhen Wen, Yun Fu, and Thomas S. Huang. Audio-visual spontaneous emotion recognition. In *Artifical Intelligence for Human Computing*, volume 4451 of *Lecture Notes in Computer Science*, pages 72–90. Springer, 2007.

[211] Zhihong Zeng, Maja Pantic, Glenn I. Roisman, and Thomas S. Huang. A survey of affect recognition methods: adio, visual and spontaneous expressions. *IEEE Transactions on pattern analysis and machine intelligence*, 31(1), 2009.

[212] Zhihong Zeng, Jilin Tu, Brian Pianfetti, Ming Liu, Tong Zhang, Zhenqiu Zhang, Thomas S. Huang, and Stephen Levinson. Audio-visual affect recognition through multi-stream fused HMM for HCI. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 2:967–972, 2005.

[213] Elisabeth Zetterholm. Prosody and voice quality in the expression of emotions. In *SST Proceedings of the Seventh Australian International Conference on Speech Science and Technology*, pages 109–113, 1998.

[214] Dong Zhang, Daniel Gatica-Perez, Samy Bengio, and Iain McCowan. Semi-supervised adapted HMMs for unusual event detection. In *CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1*, pages 611–618, Washington, DC, USA, 2005. IEEE Computer Society.

[215] Liang Zhang. Automatic adaptation of a face model using action units for semantic coding of videophone sequences. *CirSysVideo*, 8(6):781, October 1998.

[216] Yongmian Zhang and Qi-Ang Ji. Facial expression understanding in image sequences using dynamic and active visual information fusion. In *ICCV03*, pages 1297–1304, 2003.

[217] Yongmian Zhang and Qiang Ji. Active and dynamic information fusion for facial expression understanding from image sequences. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(5):699–714, May 2005.

[218] Guoying Zhao. Dynamic texture recognition using local binary patterns with an application to facial expressions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6):915–928, 2007.

[219] Guoying Zhao and Matti Pietikäinen. Experiments with facial expression recognition using spatiotemporal local binary patterns. In *Multimedia and Expo, 2007 IEEE International Conference on*, pages 1091–1094, 2007.

[220] Zhigang Zhu, Weihong Li, and George Wolberg. Integrating LDV audio and IR video for remote multimodal surveillance. In *CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Workshops*, Washington, DC, USA, 2005. IEEE Computer Society.

[221] Dmitry N. Zotkin, Ramani Duraiswami, and Larry S. Davis. Joint audio-visual tracking using particle filters. *EURASIP J. Appl. Signal Process.*, 2002(1):1154–1164, 2002.

[222] Fei Zuo and Peter H. N. de With. Fast facial feature extraction using a deformable shape model with Haar-wavelet based local texture attributes. In *International Conference on Image Processing ICIP'04*, 2004.

# Appendix A

# List of action units, codes and descriptors in FACS

| AU | Description |
|---|---|
| AU 1 | Inner brow raiser |
| AU 2 | Outer brow raiser |
| AU 4 | Brow lowerer |
| AU 5 | Upper lid raiser |
| AU 6 | Cheek raiser and lid compressor |
| AU 7 | Lid tightener |
| AU 43 | Eye closure |
| AU 45 | Blink |
| AU 46 | Wink |

| AU | Description | action |
|---|---|---|
| AU 9 | Nose wrinkler | up/down |
| AU 10 | Upper lip raiser | up/down |
| AU 11 | Nasolabial furrow deepener | oblique |
| AU 12 | Lip corner puller | oblique |
| AU 13 | Sharp lip puller | oblique |
| AU 14 | Dimpler | horizontal |
| AU 15 | Lip corner depressor | up/down |
| AU 16 | Lower lip depressor | up/down |
| AU 17 | Chin raiser | up/down |
| AU 18 | Lip pucker | oblique |
| AU 20 | Lip stretcher | horizontal |
| AU 22 | Lip funneler | oblique |
| AU 23 | Lip tightener | oblique |
| AU 24 | Lip presser | oblique |
| AU 25 | Lips parts | up/down |
| AU 26 | Jaw drop | up/down |
| AU 27 | Mouth stretch | up/down |
| AU 28 | Lip suck | oblique |

Table A.1: Upper (left) and lower (right) face action units

| Code | Description |
|---|---|
| 51 | Head turn left |
| 52 | Head turn right |
| 53 | Head up |
| 54 | Head down |
| 55 | Head tilt left |
| 56 | Head tilt right |
| 57 | Head foreward |
| 58 | Head back |
| 61 | Eyes turn left |
| 62 | Eyes turn right |
| 63 | Eyes up |
| 64 | Eyes down |
| 65 | Walleye |
| 66 | Cross-eye |

| | Description |
|---|---|
| Action Descriptor 19 | Tongue show |
| Action Unit 21 | Neck tightener |
| Action Descriptor 29 | Jaw thrust |
| Action Descriptor 30 | Jaw Sideways |
| Action Unit 31 | Jaw clencher |
| Action Descriptor 32 | Bite |
| Action Descriptor 33 | Blow |
| Action Descriptor 34 | Puff |
| Action Descriptor 35 | Suck |
| Action Descriptor 36 | Bulge |
| Action Descriptor 37 | Lip wipe |
| Action Unit 38 | Nostril dilator |
| Action Unit 39 | Nostril compressor |

Table A.2: Head and eye possitions (left) and miscellaneous actions (right)

| **Code** | **Description** |
|---|---|
| 70 | Brows and forehead not visible |
| 71 | Eyes not visible |
| 72 | Lower face not visible |
| 73 | Entire face not visible |
| 74 | Unscorable |

Table A.3: Supplementary codes of visibility

| **Code** | **Description** |
|---|---|
| 40 | Sniff |
| 50 | Speech |
| 80 | Swallow |
| 81 | Chewing |
| 82 | Shoulder shrug |
| 84 | Head shake back and forth |
| 85 | Head nod up and down |
| 91 | Flash |
| 92 | Partial flash |

Table A.4: Supplementary codes of gross behavior

# Appendix B

# AU to facial expression conversion table

| Emotion | Prototypes | Major variants |
|---|---|---|
| **Surprise** | 1+2+5B+26<br>1+2+5B+27 | 1+2+5B<br>1+2+26<br>1+2+27<br>5B+26<br>5B+27 |
| **Fear** | 1+2+4+5*+20*+25,26 or 27<br>1+2+4+5*+25,26 or 27 | 1+2+4+5*+L or R20*+25, 26 or 27<br>1+2+4+5*<br>1+2+5Z with or without 25,26,27<br>5*+20* with or without 25,26,27 |
| **Happy** | 6+12*<br>12C/D | |
| **Sadness** | 1+4+11+15B with or without 54+64<br>1+4+15* with or without 54+64<br>6+15* with or without 54+64<br>**Note**: 25 or 26 may occur with all prototypes | 1+4+11 with or without 54+64<br>1+4+15B with or without 54+64<br>1+4+15B+17 with or without 54+64<br>11+15B with or without 54+64<br>11+17<br>**Note:** 25 or 26 may occur with all major variants |
| **Disgust** | 9<br>9+16+15,26<br>9+17<br>10*<br>10*+16+25,26<br>10+17 | |
| **Anger** | 4+5*+7+10*+22+23+25,26<br>4+5*+7+10*+23+25,26<br>4+5*+7+23+25,26<br>4+5*+7+17+23<br>4+5*+7+17+24<br>4+5*+7+23<br>4+5*+7+24 | Any prototype without any of AUs: 4, 5, 7 or 10 |

Table B.1: Action unit patterns to facial expression conversion table (Facial Action Coding System Investigator's Guide [57] [58] [59])

# Summary

Multimodal recognition of emotions has been lately extensively studied by several research groups worldwide. The race for making computer programs to be able to read emotions has been motivated by the potentially wide range of applications which involve human-machine interfaces. The emotion recognition mechanisms which now represent true engineering milestones for research area, will definitely represent standards to empower useful devices in every-day life of people in the near future. This thesis proposes different approaches for automatic recognition of emotions by considering vision and speech data. The major characteristic of the recognition is the robustness to various working contexts like face orientation, illumination, subject's gender, age and skin colour.

The modalities have been investigated separately and together in order to identify which models lead to better recognition results. The recognition process implies the classification of six categories, according to the prototypic emotions as defined by Ekman.

The facial expression classification models have been adapted and tested on samples from the unimodal Cohn-Kanade database and on the bimodal Enterface05 database.

By using the action unit annotations and by studying the data set outliers, we obtained well-balanced databases of facial expressions. In addition to building models for facial expression analysis, we have focused also on the detection of action units. The work is significant because it allows for more flexibility in decoding facial expressions.

Eventually, the facial expressions are identified using static and dynamic models. In contrast to static models which attempt the classification from separate images, dynamic models use extra clues regarding dynamic shape and texture changes from video data. In either case, we have used parameter facial representations from adapted local binary patterns, Viola&Jones features, geometric and optical flow features. For dynamic analysis of data, we separated the influence of speech on the face appearance. In the research, we proposed a version of the Adaboost.M2 multi-class classifier and used this implementation in the context of analysing large sets of visual features for facial expression recognition.

The extraction of specific indicators of emotion from speech audio data has been realized using prosodic features and classification models like GentleBoost and hidden Markov models.

In order to process real-life data, we have done research on various multimodal data fusion techniques. The fusion is firstly assumed to increase the confidence of the recognition results compared to the results of unimodal processing of emotions. Subsequently, applying fusion on face and voice data aims at decreasing the ambiguity of the classification when there is no emotion consistency among

modalities. We used the emotion recognition models which were presented in this thesis for the implementation of an on-line software prototype. The system has been described and showed in several conference papers and during the demo sessions of international conferences.
The set of audio and video emotion recognizers are eventually integrated using a common multimodal framework. We have developed a working prototype which allows for loosely coupled asynchronous communication between multiple processing components. Based on the multimodal framework, we further developed an application for detecting aggression in train compartments.

# Samenvatting

Multi-modale herkenning van emoties is een onderwerp dat de laatste tijd heel veel aandacht heeft gekregen in verschillende onderzoeksgroepen wereldwijd. Het streven, om als eerste een computer programma te creren dat in staat is om emoties te herkennen, wordt gemotiveerd door de vele toepassingen mogelijkheden op het gebied van mens-machine interactie. Emotie herkennings mechanismen die momenteel nog als belangrijke technische mijlpalen op onderzoeksgebied worden beschouwd, zullen in de nabije toekomst hun weg vinden in bruikbare apparaten voor dagelijks gebruik.

In deze thesis worden een aantal methoden voor emotie herkenning met behulp van beeld en geluid gepresenteerd. Een belangrijke eis voor de gepresenteerde methoden is robuustheid in verschillende contexten zoals gezichts orintatie, belichting, geslacht, leeftijd en huidskleur. De modaliteiten zijn zowel afzonderlijk als in combinatie onderzocht om de modellen te ontdekken die de beste resultaten opleveren. Het resultaat van het herkennings proces is een classificatie van de data in n van de zes prototype emotie klassen gedefinieerd door Ekman. De gezichtsexpressie classificatie modellen zijn getest met gezichten uit zowel het uni-modale Cohn-Kanade database als het bi-modale Enterface05 database. Door gebruik te maken van action unit annotaties en door buitenliggers in de data verzameling te bestuderen, hebben we een goed gebalanceerde database van gezichtsexpressie verkregen. Naast het construeren van modellen voor gezichtsexpressie analyse, hebben we ons ook bezig gehouden met het herkennen van action units. Action units stellen ons in staat om gezichtsexpressies op een flexibele manier te decoderen.

In het uiteindelijk resultaat worden gezichtsexpressies herkend door gebruik te maken van zowel statische als dynamische modellen. In tegenstelling tot statische modellen, die proberen te classificeren met behulp van afzonderlijke beelden, maken dynamische modellen gebruik van de extra informatie in de dynamische vorm en textuur van veranderingen in de video data. In beide gevallen gebruiken we geparametriceerde gezicht representaties van aangepaste local binary patterns, Viola&Jones kenmerken, geometrische en optical flow kenmerken. Voor de dynamische analyse van de data, hebben we de invloed van spraak op de vorm van het gezicht gescheiden. In dit onderzoek gebruiken we een versie van de Adaboost.M2 multi-class klassificator voor het verwerken van de grote verzameling van gezicht kenmerken ten behoeve van gezichtsexpressie herkenning.

Voor het extraheren van specifieke indicatoren voor emotie uit spraak data gebruiken we prosodische kenmerken en klassificators als GentleBoost en hidden Markov models.

Om echte multi-modale data te verwerken, hebben we verschillende multi-modale data fusie technieken onderzocht. De fusie van gezicht en spraak informatie heeft primair als doel om het vertrouwen van de resultaten ten opzichte van de uni-modale verwerking te verhogen. Daarnaast kan fusie de ambiguteit bij de classificatie, wanneer de emotie tussen de modaliteiten niet consistent, verlagen. De emotie herkennings modellen, gepresenteerd in deze thesis, hebben we gemplementeerd in een online software prototype. Het prototype is beschreven in verschillende conferentie papers en gedemonstreerd op verschillende internationale conferenties. De audio en video emotie herkenners zijn uiteindelijk gentegreerd met behulp van een multi-modale raamwerk. Binnen dit raamwerk bestaat het prototype uit een losse koppeling van a-synchronisch communicerende verwerkingscomponenten. Op basis van dit raamwerk is er ook een applicatie gebouwd voor het detecteren van agressie in trein coupes.

# Acknowledgements

First of all, I am grateful to my supervisor Prof. Leon Rothkrantz for his enthusiasm, trust, encouragement, understanding and for his endless ways of making difficult problems of profession and not only, more manageable. I consider myself lucky for working with such a person. I also would like to express my esteem to his wife, Fien Rothkrantz for her amiability and friendliness.
I thank my promoter Prof. Henk Koppelaar for his generosity, his exceptional ideas and support in making this thesis come to an end. Additionally, I thank Prof. Ion Văduva, Prof. Catholijn Jonker, Prof. Mark Neerincx, Prof. Mirko Novák and Prof. Václav Matoušek for their support and involvement in the reviewing process of this PhD. thesis.
I am indebted to Prof. Radu Mârşanu, Prof. Ion Ivan, Prof. Constanţa-Nicoleta Bodea, Prof. Csaba Fabian and Prof. Gheorghe Dodescu for their implication and guidance in my early stages of preparing for an academic career abroad.
I thank Ruud and Bart for their earnest in handling all kinds of technical problems and to Toos and Bianca for their readiness and prompt arrangements of administrative tasks.
Many thanks go to the group of friends and colleagues from Delft, specially to Remus and Dana, Alin and Dana, Florin and Nora, Cristi and Vali Coman, Ciprian, Bogdan Tatomir, Cătălin, Bogdan, Iulia, Anca, Zhenke, Pascal, Martijn de Jongh and Ramon.
Siska, thank you for your pleasant discussions and for making the long trips for the project meetings to become shorter and engaging to us. I wish you get the strength to succeed into your life endeavour.
Special thanks go to Mirela for the time spent together and for the abounding range of emotions we have shared in our pursuit.
I also thank some of my friends from Romania chiefly Teddy, Valeriu, Ioana, Cătălin Stăvaru, Cătălin Pau, Ovidiu and Cristi.
The list of persons would not be complete without mentioning the persons to whom I owe my current professional and life achievements, motivations, ambitions and visions.
I thank Nicu Gugiu and Florin Stoicescu for sharing their ideas with me from the very beginning. My recognition goes to Prof. Dr. Ilie Georgescu for his excellent advices which helped me stay on the track during the time. Last but not least, I would like to acknowledge the most important persons in my life, my parents Ileana and Virgil. They have offered me true examples of perseverance, integrity and moderation and they gave me unconditioned support during the difficult times I had to cope with so far.

Dragoş Datcu Delft, October 2009

# About the author

Dragos Datcu was born in Slatina, Romania, on March 12, 1980. From 1994 he attended secondary school at ”Radu Greceanu” high school in Slatina. Between 1998 and 2003 he followed the bachelor studies at the Faculty of Economic Cybernetics, Statistics and Informatics from the Academy of Economic Studies in Bucharest.

Thanks to a three month scholarship awarded by the Romanian Ministry of Education and Research, in November 2002 he joined the Knowledge Based Systems Group of the Department of Information Technology and Systems at the Delft University of Technology, the Netherlands. In the summer of 2003 he returned to the Netherlands to continue his research work in the same group. In September 2004 he started a special MSc. programme in Media and Knowledge Engineering at the same university, supported through a scholarship awarded by the Netherlands organization for international cooperation in higher education - Nuffic.

From September 2004 to October 2009 he worked as a PhD. student in the Man-Machine Interaction Group of the Department of Electrical Engineering, Mathematics and Computer Science of Delft University of Technology. After the defence of his PhD. thesis, he will continue the work at the Netherlands Defence Academy.