

Unsupervised Video Indexing based on Audiovisual Characterization of Persons

Elie El Khoury

▶ **To cite this version:**

Elie El Khoury. Unsupervised Video Indexing based on Audiovisual Characterization of Persons. Human-Computer Interaction [cs.HC]. Université Paul Sabatier - Toulouse III, 2010. English. ⟨NNT : ⟩. ⟨tel-00515424v2⟩

HAL Id: tel-00515424

https://theses.hal.science/tel-00515424v2

Submitted on 6 Sep 2010 (v2), last revised 7 Sep 2010 (v3)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



HAL Authorization

University of Toulouse **Doctoral School MITT**

Unsupervised Video Indexing based on Audiovisual Characterization of Persons

PhD Thesis

presented and defended on June 03, 2010

to obtain the title of

PhD of Science of the University of Toulouse

(Speciality: Computer Science)

by

Elie El-Khoury

Jury :

President :	Régine André-Obrecht	University of Toulouse III, France
Reviewers :	Shih-Fu Chang	Columbia University, United States of America
	Bernard Merialdo	Eurecom, France
Examinators :	Sylvain Meignier	University of Le Maine, France
	Rémi Landais	Exalead, France
Advisors :	Philippe Joly	University of Toulouse III, France
	Christine Senac	University of Toulouse III, France

Institut de Recherche en Informatique de Toulouse — UMR5505

Acknowledgments

I would like to express my deep and sincere gratitude to my supervisor, Dr. Christine Sénac. her understanding, encouraging, personal guidance and kind support have provided a good basis for the present thesis.

I am deeply grateful to my supervisor, Professor Philippe Joly, head of the SAMOVA group at IRIT Laboratory for his detailed and constructive comments, for his original ideas, and for his important support throughout this work.

I wish to express my warm and sincere thanks to Professor Régine André-Obrecht, head of the department of Computer Science at Toulouse University III, who was my supervisor at the Master level and who introduced me the the field of speech processing. Her wide knowledge and her logical way of thinking have been of great value for me.

I owe my sincere gratitude to Professor Shih-Fu Chang, head of the department of electrical engineering at Columbia university - New York, who gave me the opportunity to work with his outstanding DVMM group. His ideas and concepts have had a remarkable influence on my entire career in the field of multimedia research.

I warmly thank Dr. Sylvain Meignier, member of the LIUM Laboratory at Le Maine University, with whom I worked through the EPAC project. His extensive discussions around my work and interesting explorations have been very helpful for this study.

My sincere thanks are due to the other official referees, Professor Bernard Merialdo, head of the Multimedia group at EURECOM, and Dr. Rémi Landais, research engineer at Exalead for their detailed review, constructive criticism and excellent advice during the preparation of this thesis. I wish to warmly thank Dr. Julien Pinquier. He gave me valuable advices and friendly helps during both research and teaching work. I owe my sincere gratitude to him.

I also wish to thank Dr. Jérôme Farinas, Dr. Isabelle Ferrané and Dr. Hervé Bredin for their technical assistance and their interesting discussions related to my work. During this work I have collaborated with many colleagues at SAMOVA group. I wish to extend my warmest thanks to Benjamin, Hélène, José, Giannis, Lionel, Philippe, Reda, Jérémy, Zein and Eduardo for their sympathetic help and friendly discussions. Many thanks to the Sport Branch of SAMOVA group that gave me the possibility to have some fun during work and to stay in a good mood.

I owe my sincere thanks to my lebanese friends without whom the living abroad has been difficult. I wish to especially thank Youssef (my cousin), Walid, Adèle, Rajaa, Fares, Jacques,

Layale, Sandy, Sarah, Fadi, Nadine, Julie, Youssef, Phélomène, Hervé, Elias, Pierre, Antoine, Rana, Serge, Wissam, Hikmat, Issam, William, Nemer, Mario, Michel, Joseph, Georges, Roland, Toni, Bendy, Rami, and many others. Their kind support and the time we spent together has been of great value for me.

Last but not least, I would like to warmly thank my parents who never stop believing in me. I also wish to sincerely thank my brother Roger and my sisters Marleine and Pauline for their permanent support.

to my parents,

"It's not that I'm so smart, it's just that I stay with problems longer".

Albert Einstein

Table of Contents

List of Figures **xiii**

General Introduction **1**

1 Context . . . 1
2 Characterization of persons . . . 2
3 Our Contribution . . . 3
4 Organization of this report . . . 3

Part I Audio speaker indexing **5**

Introduction **7**

Chapter 1 State-of-the-art of Speaker Diarization **11**

1.1 Acoustic Features . . . 12
1.2 Audio event segmentation . . . 13
1.3 Audio speaker segmentation . . . 14
1.3.1 Segmentation by silence detection . . . 15
1.3.2 Segmentation by speaker change detection . . . 15
1.3.2.1 Symmetric Kullbach-Leibler divergence . . . 16
1.3.2.2 Generalized Likelihood Ratio . . . 17
1.3.2.3 Bayesian Information Criterion . . . 18
1.3.2.4 Hotteling T^2-Statistics with BIC . . . 19
1.4 Audio speaker clustering . . . 21
1.4.1 BIC based approaches . . . 23
1.4.2 Eigen Vector Space Model approach . . . 23
1.4.3 Cross Likelihood Ratio clustering . . . 24
1.4.4 Hidden Markov Model approach . . . 25
1.4.5 Other clustering techniques . . . 26

1.5 Examples of state-of-the-art speaker diarization systems 26
1.5.1 The LIMSI speaker diarization system 26
1.5.2 The IBM speaker diarization system 27
1.5.3 The LIA speaker diarization system 28
1.6 Databases 30
1.6.1 ESTER-1 Corpus 30
1.6.2 ESTER-2 Corpus 30
1.6.3 EPAC-ESTER Corpus 30

Chapter 2 Proposed System for speaker diarization **33**
2.1 Proposed Generic GLR-BIC segmentation 34
2.1.1 Proposed Method 34
2.1.2 Bidirectional segmentation 36
2.1.3 Penalty coefficient decreasing technique 36
2.1.4 Other applications of the method 38
2.2 Proposed clustering 39
2.2.1 Improved EVSM clustering 39
2.2.2 BIC clustering 40
2.2.3 CLR Post-Clustering 42
A. Adaptative thresholding. 42
B. CLR+BIC fixed thresholding. 43
2.3 System architecture 44

Chapter 3 Experiments and Results **47**
3.1 Evaluation of the speaker segmentation 47
3.1.1 Experiments and Results 48
3.2 Evaluation of the acoustic events detection 52
3.2.1 Experiments and results 53
3.3 Evaluation of the speaker clustering 55
3.4 Evaluation of the speaker diarization system 56
3.4.1 Experiments and results 56

Conclusion **59**

Part II Visual people indexing **61**

Introduction **63**

Chapter 4 State-of-the-art **67**
4.1 Low-level visual features . . . 67
4.2 People detection . . . 72
4.2.1 Face detection . . . 72
4.2.2 Upper-body detection . . . 74
4.3 People tracking . . . 75
4.3.1 Existing methods for people tracking . . . 75
4.3.2 Face tracking . . . 76
4.3.3 Clothing tracking . . . 77
4.4 People clustering . . . 77
4.4.1 Drawback of people clustering methods . . . 77
4.4.2 The use of hair descriptors . . . 79
4.4.3 The use of SIFT features . . . 80

Chapter 5 Proposed Face-and-clothing based people indexing **85**
5.1 System Architecture . . . 86
5.2 Shot Boundary Detection . . . 86
5.3 Face based detection . . . 87
5.4 Clothing extraction . . . 88
5.5 People tracking . . . 88
5.5.1 Face-based people tracking . . . 89
5.5.2 Clothing-based people tracking . . . 93
5.6 Proposed methods for people clustering . . . 94
5.6.1 Face-based clustering . . . 94
5.6.1.1 Choice of the key-face . . . 95
5.6.1.2 SIFT matching . . . 96
5.6.2 Clothing based clustering . . . 99
Histograms Comparison . . . 99
Dominant Color . . . 100
Texture . . . 102
5.6.3 Hierarchical bottom-up clustering . . . 103

Chapter 6 Experiments and Results **107**
6.1 Evaluation tool . . . 107
6.2 Corpora . . . 108
6.2.1 Development corpus . . . 108
6.2.2 Test corpus . . . 108

6.3 Experiments on the development set 110
6.4 Results on the test set 112

Conclusion **127**

Part III Audiovisual fusion **129**

Introduction **131**

Chapter 7 State-of-the-art **135**
7.1 Fusion architectures 135
7.2 Mathematical aggregation operators 137
7.3 Existing works in audiovisual fusion 138
7.3.1 Audiovisual scene segmentation 139
7.3.2 Audiovisual video structuring 139
7.3.3 Audiovisual music video segmentation 140
7.3.4 Spatio-temporal detection of talking person 141
7.3.5 Audiovisual speaker recognition 141
7.3.6 Audiovisual synchronization 142
7.3.7 Audiovisual speaker diarization 143
7.3.8 Major casts list 145

Chapter 8 Proposed audiovisual fusion methods **147**
8.1 Association between audio and video indexes 147
8.1.1 Automatic matching using weighted co-occurrence matrix 148
8.1.1.1 Index intersection 149
8.1.1.2 Index fusion 150
8.1.2 The use of the face size 152
8.1.3 Lips activity detector 154
8.2 Audiovisual system for people indexing 157

Chapter 9 Experiments and Results **163**
9.1 Database 164
9.2 Results of the speaker diarization 164
9.3 Results of the video people diarization 165
9.4 Results of the audiovisual association 166
9.4.1 The baseline system 167
9.4.2 The use of the face size 169

9.4.3 The use of the lips activity rate 170
9.4.4 The combined system 171
9.5 Analysis of the errors 173

Conclusion **175**

General Conclusions and Perspectives **177**

Appendix A Application of the GLR-BIC segmentation for Program Boundaries Detection **181**
A.1 Program boundaries detection using visual features 181
A.2 Program boundaries detection using acoustic features. 182
A.3 Program boundaries detection using audiovisual features. 183
A.4 Experiments 184

Appendix B Additional visual features **187**

Appendix C Output XML format of the audiovisual fusion **189**

Appendix D Publications **191**

Bibliography **193**

List of Figures

1 The general architecture of a speaker indexing system. 8

1.1 Creating the MFCC features of a signal x. 13

1.2 The multiple change detection algorithm used by [SFA01] and [CV03]. 20

1.3 The hierarchical bottom-up or top-down clustering. 21

1.4 The EVSM-based algorithm. 23

2.1 GLR-BIC segmentation. 37

2.2 Correction due to S1 and S2 (perfect correction). 38

2.3 Correction due to S1 (partial correction). 38

2.4 Correction due to S2 (partial correction). 38

2.5 EVSM-based hierarchical clustering. 40

2.6 Similarity matrix between segments and clusters. 42

2.7 The threshold Thr_{CLR} in respect to the duration time. 43

2.8 Standard vs Proposed Speaker Diarization systems. 46

3.1 The Argos precision and recall measures in respect to the ground truth segmentation and the system segmentation. 49

3.2 Comparison between the precision of BIC, GLR-BIC and Iterative system segmentation methods on the 20 files of ESTER-2 development set. 51

3.3 Comparison between the recall of BIC, GLR-BIC and Iterative system segmentation methods on the 20 files of ESTER-2 development set. 52

1 General steps for unsupervised visual people indexing. 64

4.1 Examples of rectangle features: (A) and (B) show *two-rectangle feature*, (C) shows a *three-rectangle feature*, (D) shows a *four-rectangle feature* [VJ01]. 74
4.2 A face image and its corresponding sift features. 82

5.1 General architecture of the people indexing system. 86
5.2 Multi-face frontal detection on a still image. 88
5.3 Extraction of clothing using frontal faces. 89
5.4 Extracting and modeling the color of the skin within the face. 90
5.5 Examples of skin color extraction within the face: For each face image, the corresponding extracted skin part image appears below it. 91
5.6 The backward-forward tracking scheme. 92
5.7 2D histogram for the H and S components computed on the clothing zone. 94
5.8 Choice of the key-face. 96
5.9 Example of good matches under some variation in lighting, orientation and scale. 97
5.10 Example of bad matches. 97
5.11 Example of 13 faces of the same person that were correctly matched using ANMPD distance: we can notice different facial expressions, lightning conditions, glasses and occlusions. This example is taken from the AR database [MB98]. 99
5.12 Two people with two different costume box: the noise is due to the background and to the foreground objects like hands and characters. 100
5.13 Extraction of the dominant color. 101
5.14 Examples of dominant color areas extracted. 102
5.15 First-level hierarchical clustering. 105

6.1 Comparison between the four features and the proposed clustering method. . . . 112
6.2 Comparison between applying the Histogram comparison directly on the costume box and applying it on the dominant color area. 113
6.3 Example 1 of cluster delivered at the end of the clustering process of “arret sur image” TV debates. 113
6.4 Example 1 of cluster delivered at the end of the clustering process of “arret sur image” TV debates. 114
6.5 Example of a cluster delivered at the end of the clustering process on the ABC news. 116

6.6 Example of a cluster delivered at the end of the clustering process on the CNN news. . . . 117
6.7 Example 1 of a cluster delivered at the end of the clustering process on France 2 news. . . . 117
6.8 Example 2 of a cluster delivered at the end of the clustering process on the France 2 news. . . . 118
6.9 Example of a cluster delivered at the end of the clustering process on the LBC news. . . . 118
6.10 Example of a cluster delivered at the end of the clustering process on the CCTV news. . . . 119
6.11 Example 1 on “le Grand Journal”. . . . 120
6.12 Example 2 on “Le Grand Journal”. . . . 121
6.13 Example 3 on “Le Grand Journal”. . . . 121
6.14 Example 4 on “Le Grand Journal”. . . . 122
6.15 Example 5 on “Le Grand Journal”. . . . 122
6.16 Example on “C’est dans l’air”. . . . 123
6.17 Example of the movie “Amelie”. . . . 123
6.18 Example 1 of the movie “Asterix et Obelix”. . . . 124
6.19 Example 2 of the movie “Asterix et Obelix”. . . . 124
6.20 Example 1 of the movie “Virgins suicide”. . . . 125
6.21 Example 2 of the movie “Virgins suicide”. . . . 125

7.1 Different types of fusion defined by Dasarthy [Das94]. . . . 137

8.1 Number of frames for each character appearance, on a TV talk show. . . . 148
8.2 A list containing the output of the association process. . . . 151
8.3 Talking person appearing with the audience in a TV debate. . . . 152
8.4 Three faces detected with their corresponding weights. . . . 153
8.5 Two faces with different sizes but similar weights. . . . 154
8.6 Mouth localization using geometrical constraint. . . . 155
8.7 Lips activity curve. . . . 156
8.8 Architecture of the audio-visual people indexing system. . . . 160
8.9 Output 2: list of persons with the most representative voices and faces. . . . 161

9.1 Example 1 of the movie “Les Choristes”. 168
9.2 Example 2 of the movie “Les Choristes”. 168

A.1 Variation of Feature F1 for 3 consecutive programs. 182
A.2 Distribution of Feature F1 for 3 consecutive programs. 183
A.3 Variation of Feature F3 for the same 3 consecutive programs. 184
A.4 Distribution of Feature F3 for the same 3 consecutive programs. 185

C.1 The output file in the XML format. 190

General Introduction

You're resting on your sofa, watching your favorite series on your HQ television. At the same time, you're reading your electronic version of the New York Times on your laptop, downloading a new song on your smart phone. Suddenly, you remember one of the "Charlie Chaplin" black and white movies. You open the Youtube web page, type few words, and instantly get it and start viewing it... While watching, memories take you back to the past, to the first time you watched color TV, the first time you used mobile phone, the first time you recorded a CD... What? like a zillion years ago?... That is how fast are the advances in data capturing, storage, indexing, and communication technologies!

1 Context

Since many decades, the multimedia technologies have facilitated the way delivering data to customers, and connecting with banks of audio, image, text and video information. Actually, billions of video files are viewed[1] and thousands of them are created every day. However, there are still limited tools to index, characterize, organize and manage these data. Thus, there are still limited applications that allow users interacting with them. Manually generating a description of audiovisual content of data is not only very expensive, but sometimes, time consuming, subjective and inaccurate.

It was obvious that many laboratories and scientist started giving the domain of multimedia indexing a special attention, and joining the efforts to propose new algorithms that enable fast and accurate access to the information the consumer is asking for. Since an audiovisual document is basically multimodal, and since the content of those media are generally correlated, recent research activities are focusing on finding ways to combine useful information coming from audio, video, image and text to enhance the content based multimedia indexing.

[1] Youtube hits one billion views per day in October 2009

The SAMOVA[2] team of the IRIT[3] laboratory was born in 2002 with the goal of exploiting the audio and video media, and to study the correlation between them. In the audio domain, we may include works carried on speech activity detection [PRAO03], language recognition [RFPAO05], singing detection and music characterization [LAOP09], etc. In the video domain, we may include works carried on human shape analysis [FJ06], person labeling using clothing [Jaf05], etc. In audiovisual domain, we may include works carried on dynamic organization of database using a user-defined similarity [PPJC08], similarity measures between audiovisual documents [HJC06], etc[4]. During this process, it has been found that person characterization was not paid enough attention except the orphan work on clothing.

In almost every audiovisual document, persons appear, interact and talk. Thus detecting, tracking, classifying and identifying them have very significant impacts on the knowledge of that document, and enable a huge amount of applications.

The work of this thesis is essentially focused on video indexing based on audiovisual characterization of persons. To be as generic and training-free as possible, we decide solving this task with an unsupervised manner.

2 Characterization of persons

The characterization of persons within an audiovisual document is one of the challenging problems in current research activities. Many of them have addressed this problem with only one modality.

From the audio point of view, the characterization of persons is generally known as speaker diarization: it aims to segment the audio stream into turns of speakers and then cluster all turns that belong to the same speaker. In other meanings, its goal is to answer the questions "who talk? and when?".

From the video point of view, the characterization of persons is generally known as people detection, tracking and recognition. In other words, it aims to answer the questions "who appear? and when?".

[2]Structuration, Analyse et Modelisation des documents Video et Audio.
[3]Institut de Recherche en Informatique de Toulouse: http://www.irit.fr
[4]http://www.irit.fr/recherches/SAMOVA/

A few other research activities have addressed the problem of persons characterization from a multimodal point of view. However their applications are generally limited and constrained. Thus, we may define this task by trying to answer the following questions:

- "Who talk and appear? and when?"
- "Who talk without appearing? and when?"
- "Who appear without talking? and when?"

3 Our Contribution

Our main contribution in this Ph.D, financed by the French Ministry of Education, can be divided into three parts:

- Propose an efficient **audio indexing system** that aims to split the audio channel into homogeneous segments, discard the non-speech segments, and group the segments into clusters, that each corresponds ideally to one speaker. This system must process without *a priori* knowledge (unsupervised learning) and must be suitable to any kind of data: TV/radio broadcast news, TV/radio debates, movies, etc.
- Propose an efficient **video indexing system** that aims to split the video channel into shots, detect and track people in every shot, and group all faces into clusters, that each corresponds ideally to one person. This video system must process without *a priori* knowledge and may be suitable to any kind of data.
- Propose an efficient **audiovisual indexing system** that aims to combine audio and video indexing systems in order to deliver an audiovisual characterization of each person talking and/or appearing in the audiovisual document, and a robustified audio indexing output (respectively video indexing output) using the help of video (respectively the aid of audio).

4 Organization of this report

This report is composed of three main parts:

1. Part I considers the audio channel: state-of-the-art methods for speaker diarization are reviewed in chapter 1, our proposed audio indexing system is described in chapter 2, and the experiments and the results are detailed in chapter 3.

2. Part II considers the video channel: existing methods for people detection, tracking and recognition are reviewed in chapter 4, our proposed face-and-clothing based people indexing system is presented in chapter 5, and the experiments and the results are described in chapter 6.

3. Part III considers the fusion between audio and video descriptors: existing works on audiovisual fusion are detailed in chapter 7, our proposed audiovisual association system is described in chapter 8, and the experiments and the results are shown in chapter 9.

Part I

Audio speaker indexing

Introduction

Most of works in audio indexing are leading to annotate an input audio signal with information that attributes temporal regions of signal to their specific sources/classes and then to give a special ID to each class. These IDs may identify particular speakers, music, background noise or other signal sources like animal voices, applause, etc. Even though we are only interested in the issue of speaker indexing, it is very important to have a good processing way to get rid of the non speech segments.

The audio speaker indexing aims to detect speaker identity changes in a multi-speaker audio recording and classifies each detected segment according to the identity of the speaker. It is sometimes confused with speaker diarization that consists in answering the question "Who spoke when?". In another meaning, its purpose is to locate each speaker turn and to assign it to the appropriate speaker cluster. The output of the corresponding system is a set of segments with a unique ID assigned to each person. Another definition of *speaker diarization* is *speaker segmentation and clustering*. On one hand, the speaker segmentation aims to detect speaker changes in an audio recording. On the second hand, the speaker clustering aims to group segments corresponding to the same speaker into homogeneous clusters.
The general architecture of the speaker indexing system is illustrated in Fig.1.

Domains that receive special research attention are *telephone speech*, *broadcast news* (radio, TV) and *meetings* (lectures, conferences and debates). The corresponding speaker diarization systems have been evaluated by organizations like NIST (National Institute for Standards and Technology) and AFCP (Association Francophone de la Communication Parlée).

Hypotheses

In this work, many hypotheses were taken in order to make the problem of *speaker diarization* the most general and the most useful.

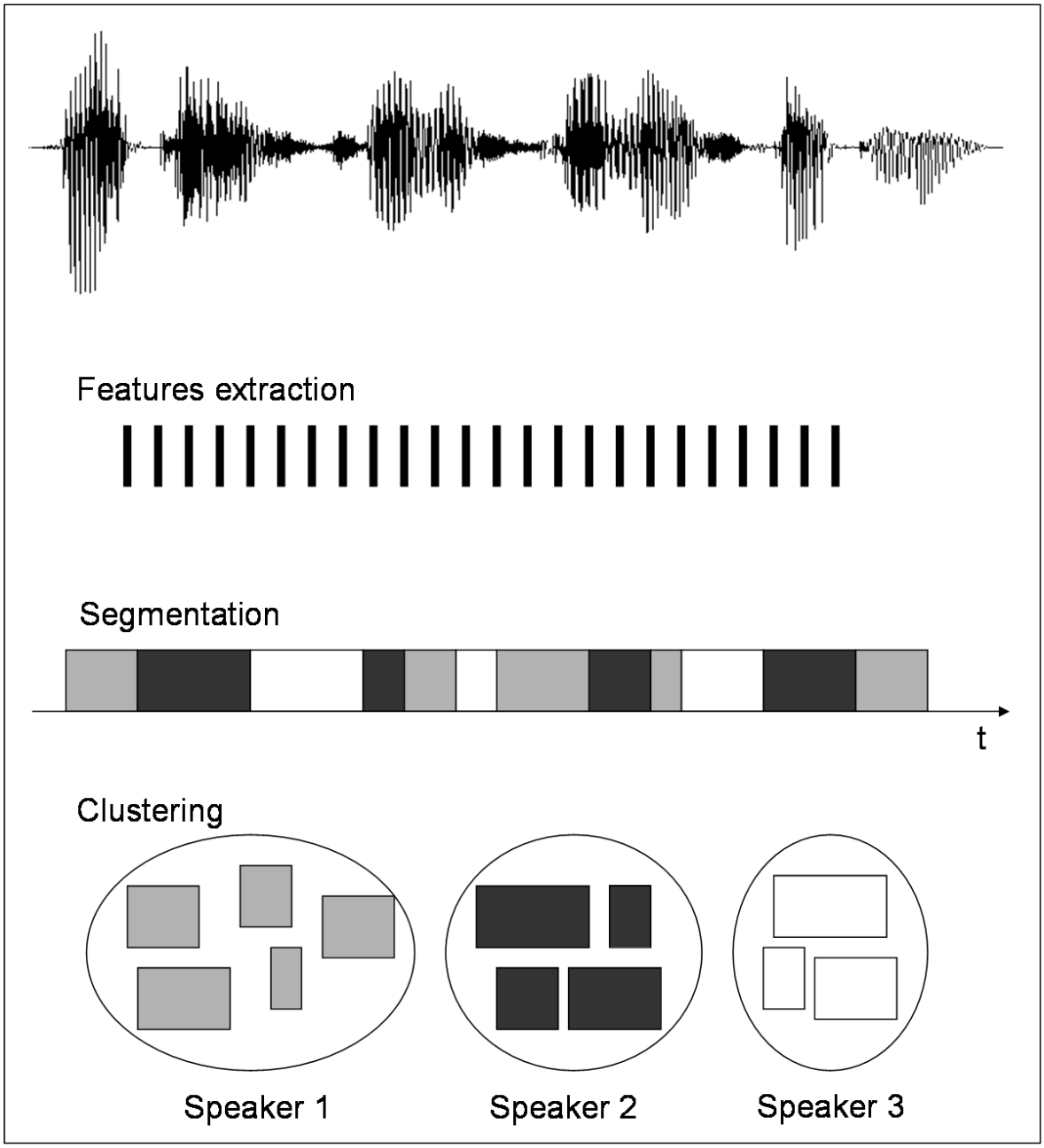


Figure 1: The general architecture of a speaker indexing system.

- **Unknown number of speakers.** Unlike telephone conversations where almost two people talk, a more realistic case considers that the number of speakers is unknown and one of the final goals is to determine this number.

- **No *a priori* knowledge about speakers and language.** We consider that the identity of the speakers in the documents is unknown and that there are no trained models for each

speaker and each language. However, a knowledge about the background is allowed i.e. a universal model separating "studio clean" recording from "outdoor noisy" recording can be trained.

- **Not only speech.** Recording speech data contain generally in addition to speech, music and other non-speech sources. Thus, the realistic choice is to build a system that first detects the speech and non speech regions in order to enable processing on the speech regions on later stages.

- **People may talk simultaneously.** In many existing systems, this hypothesis was neglected or at least not paid a special attention. Effectively, this is not very important if the data processed are broadcast news: in this case, the speech is even prepared previously and then read, or at least "*speakers are polite*". But in some meeting or translation conditions, it is obvious that we should take care of this assumption.

Applications

Audio speaker indexing is very useful in many types of applications because it provides extra information according to the speakers. By adding this knowledge to speech transcripts, it becomes easier for humans to localize relevant information and for speech translation systems to process it. Some of those applications may be:

- **Indexing audio recording databases.** Effectively, this is its first goal because it may be used as a preliminary step in every task of *Information Retrieval*. Typical automatic uses of such system output might be speech summarization and translation. Coupled with the speaker identification process, it allows, for example, retrieving all speeches of a certain political leader. It may be useful to know the speech duration of each candidate during a presidential campaign. Also, it may be used to retrieve the speech of a journalist in order to identify the topics addressed in a broadcast news recording.

- **Automatic Speech Recognition.** Speaker segmentation algorithms are used to split the audio recording into small homogeneous segments. Speaker clustering algorithms are also used to cluster the input data into speakers towards model adaptation that is successfully used to improve ASR systems performance.

- **Speaker tracking, speaker recognition.** Speaker diarization can be used as a pre-processing low-cost module for speaker-based algorithms by splitting the whole data into individual speakers. Thus, the decision is more reliable because it is taken on relatively long segments and huge clusters instead of only some tens of milliseconds.

This part is organized as follows: Chapter 1 presents the state-of-the-art works on *speaker diarization*. In chapter 2, we detail our proposed methods for *speaker segmentation* and *speaker clustering*. Chapter 3 describes the experiments and the results.

Chapter 1

State-of-the-art of Speaker Diarization

Contents

1.1 Acoustic Features 12

1.2 Audio event segmentation 13

1.3 Audio speaker segmentation 14

1.3.1 Segmentation by silence detection 15

1.3.2 Segmentation by speaker change detection 15

1.4 Audio speaker clustering 21

1.4.1 BIC based approaches 23

1.4.2 Eigen Vector Space Model approach 23

1.4.3 Cross Likelihood Ratio clustering 24

1.4.4 Hidden Markov Model approach 25

1.4.5 Other clustering techniques 26

1.5 Examples of state-of-the-art speaker diarization systems 26

1.5.1 The LIMSI speaker diarization system 26

1.5.2 The IBM speaker diarization system 27

1.5.3 The LIA speaker diarization system 28

1.6 Databases 30

1.6.1 ESTER-1 Corpus 30

1.6.2 ESTER-2 Corpus 30

1.6.3 EPAC-ESTER Corpus 30

In this chapter, the main existing techniques for speaker diarization are reviewed. First, the acoustic features that have been found useful for speaker diarization are listed in section 1.1. In section 1.2, a brief look on the audio event segmentation is presented. Then, the different approaches used for speaker segmentation and speaker clustering are respectively described in sections 1.3 and 1.4. Some of the famous existing systems are presented in section 1.5. The databases used in our work are described in section 1.6.

1.1 Acoustic Features

Acoustic features extracted from the audio recording provide information on the speakers during their conversation. This information allow the system to separate them correctly.

As for many speaker-based processing techniques, the cepstral features are the mostly used in speaker diarization systems. These parametrization features are: the Mel Frequency Cepstrum Coefficients (MFCC), the Linear Frequency Cepstrum Coefficients (LFCC), the Linear Predictive Coding (LPC), etc.

Moreover, in the area of audio event segmentation (speech, music, noise and silence), features like the energy or the 4 Hertz modulation energy were shown to be useful for speech detection. Other features like the number and the duration of the stationary segments obtained from a forward/backward segmentation [AO88] are used for example for music detection.

In addition, some frequential information like the pitch frequency and the harmonical frequencies are used to separate for example males from females in the speech part.
In the following subsections, the acoustic features used in our work are detailed:

- **Mel Frequency Cepstrum Coefficients.** The ceptral information of an audio signal allows to separate the glottal excitation and the resonance of the vocal tract. By filtering the signal, only the contribution of the vocal tract is kept. MFCCs were introduced in [Mer76]. They are generally derived as seen in Fig.1.1. After windowing the signal using Hamming approximation, the Fourier transform is computed on every window, then the powers of the spectrum are mapped onto the MEL scale using triangular overlapping windows. After that, the logs of the powers of each of the MEL frequencies are taken. Finally the inverse of the fast Fourier transform of the list of Mel log powers are computed.

Thus, the MFCCs are the amplitudes of the resulting spectrum. Practically, a MFCC vector is extracted every 10 milliseconds on a shifted Hamming window of 20 milliseconds.

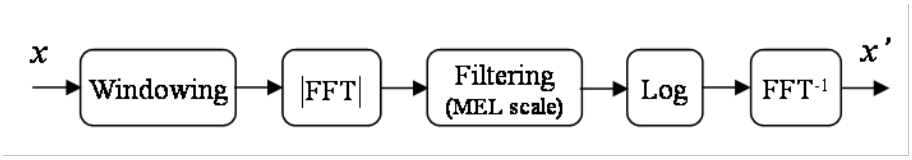


Figure 1.1: Creating the MFCC features of a signal x.

- **4 hertz modulation energy.** Unlike the music signal, the speech signal has an energy modulation peak around the 4 Hz syllabic rate (4 syllables per second). This property was used in [PRAO03] to separate speech from music, but also can be used to distinguish clean speech from noisy speech, or mono-speaker speech from interaction zones where two or more people talk simultaneously. Typically, a value of the 4 Hz modulation energy is computed every 16 milliseconds.

- **Pitch frequency.** This feature characterizes the gender of the speaker. The pitch frequency of the voice is generally around 150 Hertz for a man. In opposite, it is around 250 hertz for a woman and around 350 hertz for children. This property can be used to help the clustering process. Moreover, algorithms used to estimate this pitch can help the speech detection and music detection because unlike instrumental voices, a normal human voice cannot be less than 60 Hz and higher than 400 Hz. In this work, we used the pitch estimators of *The Snack Sound Toolkit*[5].

- **Number and duration of segments** provided by the forward/backward segmentation method [AO88]. This segmentation method estimates the boundaries of every phonetic unit present in the acoustic signal. Unlike speech signal, music signal is characterized by a relative lower number of those units and a higher value of their duration.

1.2 Audio event segmentation

Known as "Segmentation en Evénements Sonores"(SES) by the french community, the output of such a segmentation is a list containing the starting and the ending times of all the audio

[5]http://www.speech.kth.se/snack/

events that occur in the audio recording. Those events are: *speech, non-speech, music, non-music and (speech + music).* Typically, a SES system is used as a preprocessing step for the speaker diarization system and, to the best of our knowledge, all existing methods build those two systems completely separatly. That is why the results of the second system are directly related to the output of the first one: if the turns of speaker X were not detected as speech by the SES system, X will be missed and there is no possibility to find it again. As seen in section 2.3, we describe a framework to handle this weakness by proposing an iterative system that enables both *audio event segmentation* and *speaker diarization.*

The task of audio event segmentation can be divided into two main issues:

On one hand, algorithms used for speech activity detection are often based on Gaussian Mixtures Models (GMMs) for both Speech and Non-Speech components [GL94] using the MFCC vectors. Those models need learning and depend on the training data. However, unsupervised methods use robust features like the 4Hz modulation energy described in the previous section that practically is affected by the database variation.

On the other hand, algorithms used for music detection are also based on both supervised methods using GMMs on MFCCs and unsupervised methods using the number and the duration of segments as explained previously.

The fusion of supervised and unsupervised methods was developed at the IRIT Laboratory and gave results among the best on ESTER-1 database (cf. section 1.6.1 [GGM$^+$05]. For more details about those methods, please refer to [PRAO03]. Recently, methods bases on Support Vector Machine (SVM) are shown to provide a slightly better performance [TMN07].

1.3 Audio speaker segmentation

Speaker segmentation consists in segmenting the audio recording into homogeneous segments. Each segment must be as long as possible and must contain the speech of one speaker. This segmentation is closely related to acoustic change detection as it will be pointed out later on (cf. section 2.3).

Two main categories of speaker segmentation can be found in the literature: the segmentation by silence detection and the segmentation by speaker change detection. Those two techniques are explained in the following subsections.

1.3.1 Segmentation by silence detection

It is the intuitive and trivial solution to separate turns of speaker in the audio recording. It assumes that changes between speakers happen through a silence segment. A silence is characterized by a low energy level. For some types of data like telephone conversations where the noise is strongly present, this hypothesis is not realistic. Most existing methods for silence detection use:

- **the mean power of the signal.** This is the simplest way to detect silence [NA99]. This method encounters two main problems. First, the choice of the threshold used to isolate silence is not very stable because it depends on the processed data. Then, the boundaries are not well detected because the mean average of the power is computed every 0.5 or 1 second.

- **the histogram of the energy.** This method [MC98] splits the audio recording into segments of 15 seconds. The histogram of each segment is approximated by a Gaussian distribution. If the segment is shown to be homogeneous in terms of the probability density function, it is indexed as silence or non-silence. If it is not the case, the segment is splitted by using the k-means algorithms that computes the average mean and the standard deviation of both silence and non-silence parts.

- **the variability of the energy.** This method [GSR91] consists in computing the variability of the energy for a signal portion. If the variability is low, then this portion is considered as silence. If this variability is high, it is considered as speech.

- **the zero-crossing rate.** The silence, besides being characterized by a low-level energy, has a high zero-crossing rate [TP99]. This rate represents the number of times the signal has zero amplitude by temporal unit.

All approaches for speaker segmentation by silence detection need a threshold that depends on the audio document. Furthermore, there is no efficient method to determine optimally this threshold.

1.3.2 Segmentation by speaker change detection

The speaker change detection (SCD) is the most common method used for speaker segmentation. It aims to detect boundaries for each speaker turn within the audio recording even if there is

no silence between two consecutive speakers. That explains the numerous existing methods for SCD and why speaker segmentation has sometimes been referred to as SCD.

Technically, two main types of SCD systems can be found in the bibliography. The first kind are systems that perform a single processing pass of the audio recording. The second kind are systems that perform two-pass algorithms: in the first pass, many points of change are suggested with a high false alarm rate. Then, in the second pass, those points are re-evaluated and some are discarded in order to converge into an optimum speaker segmentation output.

In the following sections are presented some existing methods that were successfully used for SCD. Those methods were applied for either a single processing pass or multiple processing passes. Moreover, they can be classified into three categories: metric-based approaches like the symmetric Kullbach-Leibler (KL2) distance, model-based approaches like the Generalized Likelihood Ratio (GLR) and the Bayesian Information Criterion (BIC), and mixed approaches like the Hotelling T^2-Statistics and BIC.

1.3.2.1 Symmetric Kullbach-Leibler divergence

The Kullbach-Leibler [KL51] measures the difference between the probability distributions of two continuous random variables. It is given by:

$$D(p_1, p_2) = \int_{-\infty}^{+\infty} p_1(x) ln(\frac{p_1(x)}{p_2(x)}) dx \tag{1.1}$$

Because this expression is not symmetric in respect to the two variables, the symmetric KL (KL2) is proposed:

$$\Delta = \frac{D(p_1, p_2) + D(p_2, p_1)}{2} \tag{1.2}$$

When the distributions are Gaussian N1(μ_1,σ_1) and N2(μ_2,σ_2), it becomes:

$$\Delta = \frac{1}{2}[\frac{\sigma_1^2}{\sigma_2^2} + \frac{\sigma_2^2}{\sigma_1^2} + (\mu_1 - \mu_2)^2(\frac{1}{\sigma_1^2} + \frac{1}{\sigma_2^2})] \tag{1.3}$$

where μ_i represent the mean average and σ_i the covariance of a Gaussian distribution N_i.

In [SJRS97], the KL2 is used as follows: for every point of the audio recording, the two adjacent windows from both sides are considered. The duration of each window is fixed to 2 seconds. The mean and the covariance are estimated on each window. Thus, the KL2 distance can be easily computed. This process is repeated for every point so a distance curve is drawn and the local maxima are detected. Those local maxima correspond ideally to points of speaker changes.

1.3.2.2 Generalized Likelihood Ratio

For genericity reasons that will be addressed later in this chapter, we will describe this method using an unknown signal that may be an acoustic signal, a video signal or an audiovisual signal.

Let $X = x_1, \ldots, x_{N_x}$ be the sequence of observation vectors of dimension d to be modeled and M the estimated parametrical model and $L(X, M)$ the likelihood function. The GLR introduced by Gish et al. [GSR91] considers the two following Hypotheses:

- H_0: This hypothesis assumes that the sequence X corresponds to only one homogeneous segment (in the case of audio signal, it corresponds to only one audio source). Thus, the sequence is modeled by only one multi-Gaussian distribution.

$$(x_1, \ldots, x_{N_x}) \backsim N(\mu_X, \sigma_X) \tag{1.4}$$

- H_1: This hypothesis assumes that the sequence X corresponds to two different homogeneous segments $X_1 = x_1, \ldots, x_i$ and $X_2 = x_{i+1}, \ldots, x_{N_x}$ (in the case of audio signal, it corresponds to two different audio sources or more particularly to two different speakers). Thus, the sequence is modelled by two multi-Gaussian distributions.

$$(x_1, \ldots, x_i) \backsim N(\mu_{X_1}, \sigma_{X_1}) \tag{1.5}$$

and

$$(x_{i+1}, \ldots, x_N) \backsim N(\mu_{X_2}, \sigma_{X_2}) \tag{1.6}$$

The generalized likelihood ratio between the hypothesis H_0 and the hypothesis H_1 is given by:

$$GLR = \frac{P(H_0)}{P(H_1)} \tag{1.7}$$

In terms of likelihood, this expression becomes:

$$GLR = \frac{L(X, M)}{L(X_1, M_1)L(X_2, M_2)} \tag{1.8}$$

If this ratio is lower than a certain threshold Thr, we can say that H_1 is more probable, so a point of change in the signal is detected.

By passing through the log:

$$R(i) = -logGLR \tag{1.9}$$

and by considering that the models are Gaussian, we obtain:

$$R(i) = \frac{N_X}{2} \log |\Sigma_X| - \frac{N_{X_1}}{2} \log |\Sigma_{X_1}| - \frac{N_{X_2}}{2} \log |\Sigma_{X_2}| \tag{1.10}$$

where Σ_X, Σ_{X_1} and Σ_{X_2} are the covariance matrices of X, X_1 and X_2 and N_X, N_{X_1} and N_{X_2}, are respectively the number of the acoustic vectors of X, X_1 and X_2.

Thus, the estimated value of the point of change by maximum likelihood is given by:

$$\widehat{i} = arg \max_i R(i) \tag{1.11}$$

If $\widehat{i}$ is higher than the threshold $T = -logThr$, a point of speaker change is detected. The major disadvantage resides in the presence of the threshold T that depends on the data. That is why, Rissanen [Ris89] introduced the Bayesian Information Criterion (BIC).

1.3.2.3 Bayesian Information Criterion

For a given model M, the BIC is expressed by:

$$BIC(M) = \log L(X, M) - \frac{\lambda}{2} n \log N_X \tag{1.12}$$

where n denotes the number of the observation vectors of the model. The first term reflects the adjustment of the model to the data, and the second term corresponds to the complexity of the data. λ is a penalty coefficient theoretically equal to 1. [Ris89].

The hypotheses test of Equ.1.7 can be viewed as the comparison between two models: a model of data with two Gaussian distributions (H_1) and a model of data with only one Gaussian distribution (H_0). The subtraction of BIC expressions related to those two models is:

$$\Delta BIC(i) = R(i) - \lambda P \tag{1.13}$$

where the log-likelihood ratio $R(i)$ is already defined in Equ.1.10, and the complexity term P is given by:

$$P = \frac{1}{2}(d + \frac{1}{2} d(d+1)) \log N_X \tag{1.14}$$

d being the dimension of the feature vectors.

The BIC can be also viewed as the thresholding of the log-likelihood distance with an automatic threshold equal to λP.

Thus if $\Delta BIC(i)$ is positif, the hypothesis H_1 is privileged (two different speakers). There is a change if:

$$\{\max_i \Delta BIC(i) \geq 0\} \tag{1.15}$$

The estimated value of the point of change can also be expressed by:

$$\widehat{i} = arg\max_i \Delta BIC(i) \tag{1.16}$$

Chen et al [CG98] used this criterion to segment the data of the DARPA evaluation campaign. They said that the BIC procedure has the advantage of not using a threshold, because at that time, the existing methods used thresholds and the retrieving of the optimal thresholds was so complicated. But they forgot the penalty coefficient λ that is practically not necessarily equal to 1.

Many multi-points detection algorithms based on BIC were then developed. In [TG99], Tritschler used a shifted variable size window to detect the points of speaker change in a broadcast news audio recording. Then, the authors of [Cet00], [DW00], [SFA01] and [CV03] proposed improvements in order to obtain either more accurate detection or faster computational time. Fig.1.2 illustrates the segmentation process used in [SFA01] and [CV03]: it shows that there are 8 parameters that should be carefully tuned and that depend from the processed data. This weakness motivated us to propose a parameters-free method as seen later in section 2.1.

1.3.2.4 Hotteling T^2-Statistics with BIC

It can be easily shown that the segmentation algorithms based only on BIC have a quadratic complexity. Even if we can improve the time machine by sampling the audio signal, the computational cost stays relatively high because we should compute two full covariance matrices in each shifted variable size window.

Moreover, when estimating the mean and the covariance, the segmentation error is relatively high if the acoustic events have short durations. That is why, Zhou et al. [ZH05] used an approach for SCD using the T^2-statistics and BIC.

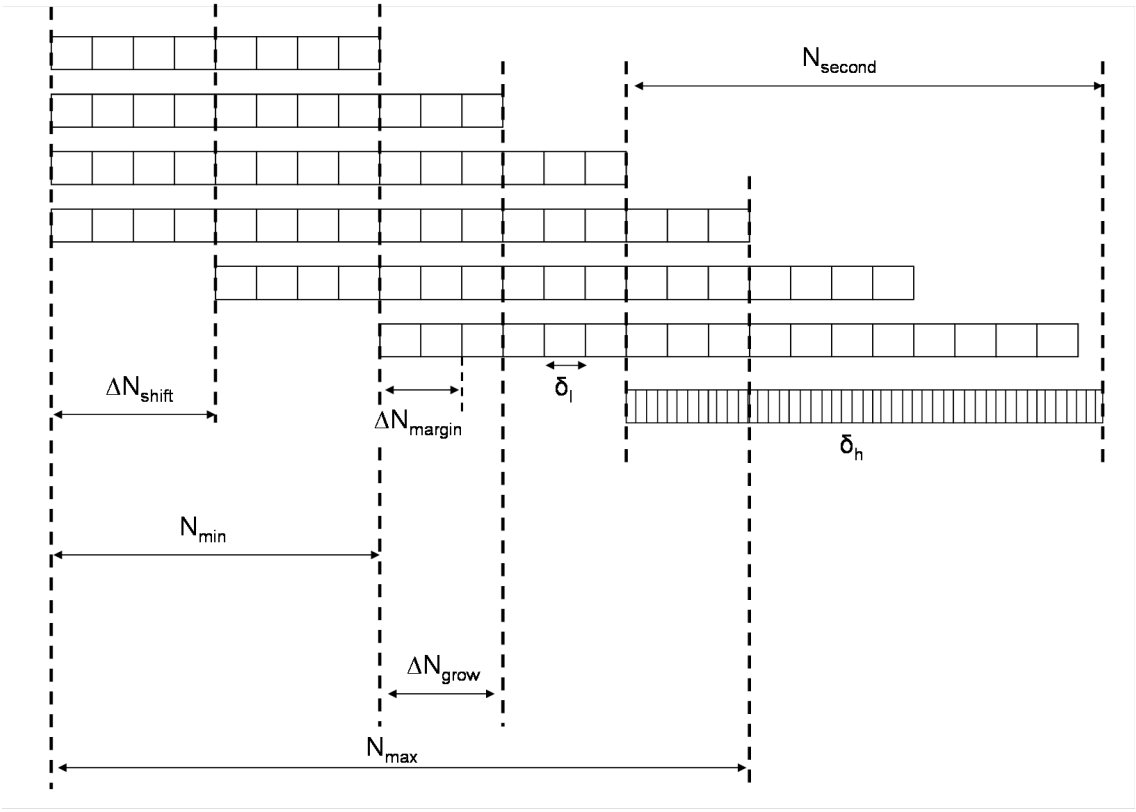


Figure 1.2: The multiple change detection algorithm used by [SFA01] and [CV03].

The T^2-statistics expression is given by:

$$T^2 = \frac{i(N_X - i)}{N_X}(\mu_{X_1} + \mu_{X_2})'\Sigma^{-1}(\mu_{X_1} - \mu_{X_2}) \tag{1.17}$$

where i corresponds to the point of change, Σ the common covariance matrix, μ_{X_1} and μ_{X_2} the estimated mean average of the Gaussian models of the two sub-windows separated by the point i. For more details about the combination between the T^2-statistics and the BIC, please refer to [ZH05].

In addition to the above techniques, there are few works that use the dynamic programming to find the speaker change points [VCR03], the *Maximum Likelihood* (ML) coupled with BIC [ZN05], or genetic algorithm [SSGALMBC06] where the number of segments is estimated via the Walsh basis function and the location of change points is found using a multi-population genetic procedure.

1.4 Audio speaker clustering

At the end of the speaker segmentation process, segments that contain the speech of only one speaker are provided. The next step aims to agglutinate together all segments that belong to the same speaker. This step of clustering can be used in many applications: for example, Automatic Speech Recognition (ASR) systems use homogeneous clusters to adapt the acoustic models using MAP (*Maximum A Posteriori*) to the speaker and so increase recognition performance.

This blind speaker clustering with no *a priori* information about the number of people and their identities, can be viewed as an **unsupervised classification** problem. In general, unsupervised classification methods use a **hierarchical clustering**.

Hierarchical clustering

The goal of the hierarchical clustering is to gather iteratively a set of elements. There are two approaches illustrated in Fig.1.3: the **bottom-up clustering** and the **top-down clustering**. The first one considers at the beginning every element as a cluster and merge after each iteration the two most similar clusters in terms of a **merging criterion**. This process is repeated until a defined **stopping criterion** is verified. Contrarily, the top-down clustering considers at the beginning the whole set of elements as only one cluster and then, after each iteration, splits the cluster in terms of a **splitting criterion**. This process is repeated until the stopping criterion is verified.

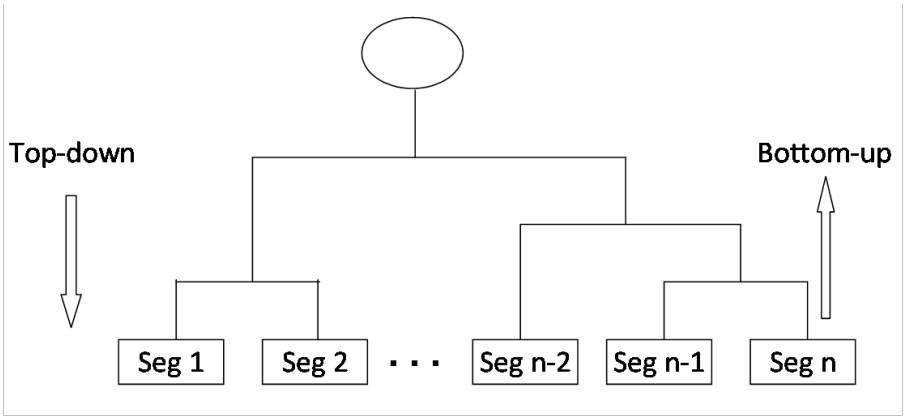


Figure 1.3: The hierarchical bottom-up or top-down clustering.

The Bottom-up clustering known also as *agglomerative clustering* is by far the mostly used in the literature because it uses the output speaker segmentation techniques to define a clustering starting point.

In the design of such systems for speaker clustering, the merging/splitting criterion C corresponds to the distance/similarity between clusters. And sometimes, instead of defining an individual value pair, a distance/similarity matrix is described, which is created with the distance/similarity from any possible pair.

More precisely, the criterion C between two clusters of elements G_1 and G_2 can be expressed in different possibilities:

- **single linkage**: also known as minimum pair, the clustering criterion C is defined as the minimum criterion separating two elements, each belonging to one cluster.

$$C(G_1, G_2) = \min_{i \in G_1, j \in G_2} C(i, j) \tag{1.18}$$

- **complete linkage**: also known as maximum pair, the clustering criterion C is defined as the maximum criterion separating two elements, each belonging to one cluster.

$$C(G_1, G_2) = \max_{i \in G_1, j \in G_2} C(i, j) \tag{1.19}$$

- **average linkage**: also known as average pair, the clustering criterion C is defined as the mean average criterion of all pairs of elements, each belonging to one cluster. N_1 and N_2 denote the number of elements respectively of G_1 and G_2 in the following formula.

$$C(G_1, G_2) = \frac{\Sigma_{i \in G_1, j \in G_2} C(i, j)}{N_1 N_2} \tag{1.20}$$

- **full linkage**: unlike the above linkage methods, this method considers a class of elements as only one element (in our case, a cluster of segments is considered as only one segment obtained by the concatenation of all the segments in the cluster). The characteristics of each class are re-computed at the end of every iteration. That involves a huge computational cost of the clustering process unlike previous methods.

The following subsections review the main systems existing in the literature for speaker clustering. Even though some of them may be suitable for online configuration where no information on the complete recording is available, the systems listed below were initially developed for an offline configuration where they have access to the whole recording file before processing it.

1.4.1 BIC based approaches

The Bayesian Information Criterion that was well explained for speaker segmentation is by far the most commonly used distance and merging criterion for speaker clustering. It was initially proposed for clustering by Chen et al. in [CG98]. The pair-wise distance matrix is computed for each iteration and the pair with the lowest ΔBIC value is merged. The process finishes when all pairs have a $\Delta BIC > 0$. Considering two clusters G_1 and G_2, each of those clusters is modeled by a multi-Gaussian distribution. The ΔBIC distance is given by:

$$\Delta BIC(G_1, G_2) = (n_1+n_2)\log|\Sigma| - n_1\log|\Sigma_1| - n_2\log|\Sigma_2| - \frac{\lambda}{2}(d+\frac{d(d+1)}{2})log(n_1+n_2) \quad (1.21)$$

where n_1, n_2 are the sizes of G_1 and G_2. Σ_1, Σ_2 and Σ are respectively the covariance matrices of G_1, G_2 and $G_1 \bigcup G_2$. d is the dimension of the feature vectors.

1.4.2 Eigen Vector Space Model approach

This method proposed by Tsai [TCCW05] uses the vector space model, which was originally developed in document-retrieval research, to characterize each utterance as a *tf-idf*-based vector of acoustic terms, thereby deriving a reliable measurement of similarity between utterances. Fig.1.4 describes the EVSM algorithm.

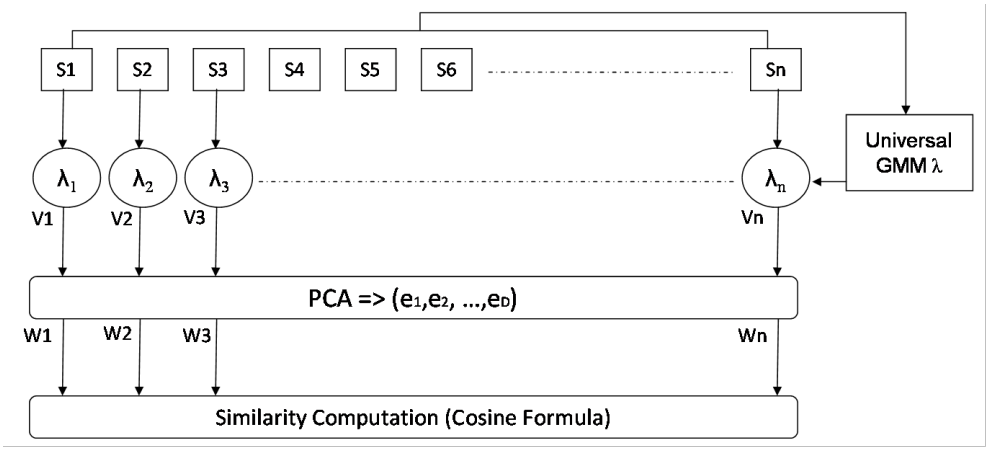


Figure 1.4: The EVSM-based algorithm.

To begin, a "universal GMM" is created using all the segments to be clustered. The training method is based on the k-means clustering initialization followed by Expectation-Maximization (EM). An adaptation of universal GMM is then performed for each of the utterances using *maximum a posteriori* (MAP) estimation. This gives N *utterance-dependent* GMMs $\lambda_1, \lambda_2, ..., \lambda_N$.

The use of such a model adaptation instead of a direct EM-based training of GMM has two-fold advantages. One is to produce a more reliable estimate of the GMM parameters for short utterances than it can be done with direct EM-based training. The other is to force the mixtures of all the utterance-dependent GMMs to be of the same order.

Next, all the mean vectors of each utterance-dependent GMM are concatenated in the order of mixture index to form a super-vector, with dimension of D. Then, Principal Component Analysis (PCA) is applied to the set of N super-vectors, $V_1, V_2, ..., V_N$, obtained from N utterance-dependent GMMs. This yields D eigenvectors, $e_1, e_2, ..., e_D$, ordered by the magnitude of their contribution to the between-utterance covariance matrix:

$$B = \frac{1}{N} \sum_{i=1}^{N} (V_i - \overline{V})(V_i - \overline{V})' \tag{1.22}$$

where $\overline{V}$ is the mean vector of all V_i for $1 < i < N$. The D eigenvectors constitute an eigenspace, and each of the supervectors can be represented by a point on the eigenspace:

$$V_i = \overline{V} + \sum_{d=1}^{D} \phi_{i,d} e_d \tag{1.23}$$

where $\phi_{i,d}$, $1 \leq d \leq D$, is the coordinate of V_i on the eigenspace. Then, the authors use the cosine formula for each pair of vectors in order to quantify the similarity between the corresponding pair of segments/clusters.

$$S_{i,j}(V_i, V_j) = \frac{W_i \cdot W_j}{\|W_i\| \, \|W_j\|} \tag{1.24}$$

where W_i and W_j are the vector V_i and V_j obtained after the reduction of the dimension.

1.4.3 Cross Likelihood Ratio clustering

The Cross Likelihood Ratio (CLR) clustering was used as a final step of *a posteriori* clustering in many speaker diarization systems as in [RSC+98] and [ZBMG05]. After a first step of clustering (using for example the BIC clustering), the background environment contribution in the cluster models must be reduced and normalized in order to allow additional clustering for speakers whose environmental conditions change during their speech. Moreover, the size of the clusters is good enough to allow building a more complex and robust speaker model, such as Gaussian Mixture Model (GMM), for each cluster. Thus, a universal background model (UBM) should

be learned and then adapted for each cluster, providing the initial speaker model. After each iteration, the clusters that maximize the Cross Likelihood Ratio (CLR) are merged:

$$CLR(G_1, G_2) = \frac{L(G_1/M_2)}{L(G_1/UBM)} \times \frac{L(G_2/M_1)}{L(G_2/UBM)} \tag{1.25}$$

Where M_1, M_2 are the models associated to the clusters G_1 and G_2, and UBM is the universal background model and L(.) is the likelihood. When the $CLR(G_1, G_2)$ is greater than an *a priori* threshold thr, the clustering step stops to merge.

The UBM results from the fusion of four models which are gender (male/female) and bandwidth (narrow/wide bands) dependent models with 128 diagonal covariance components. Then, the cluster model is derived from the UBM by MAP adaptation (means only).
Although the UBM model could be learned once, the evaluation of CLR clustering done in [BZMG06] shows that the threshold may depend on the corpus.

1.4.4 Hidden Markov Model approach

The Hidden Markov Model (HMM) was also used for speaker clustering. Every state in the model represents a cluster and the transitions between states characterize the changes between speakers.

In [ALM02], the clustering is composed of several sub-states to impose the minimum duration constraints considering that the HMM is ergodic in nature. The probability density function (PDF) of each state is represented by a GMM. The process starts by over-clustering the data (larger number of clusters than the expected number of speakers). The parameters of the HMM are then trained using the EM algorithm. The merging between two clusters is done using log likelihood ratio (LLR) distance.

In [MBI01], the clustering does not belong to the hierarchical category as all the methods described above. The speaker diarization is done with only one path unlike most of the existing systems that generally separate the segmentation step from the clustering step. The HMM is generated using an iterative process, which detects and adds a new state (i.e. a new speaker) at each iteration. The speaker detection process is composed of four steps:

1. *Initialization:* a first speaker model S0 is trained on the whole speech recording.

2. *Adding a new speaker:* a new speaker model is trained using 3 seconds of the recording that maximizes the sum of likelihood ratios of model S0. A new state S1 related to that new model is added to the HMM configuration.

3. *Adaptation speaker model:* the models are adapted after each iteration where new state are created.

4. *Assessing the stopping criterion:* This criterion is based on the comparison of the probability along the Viterbi path between two iterations of the process.

1.4.5 Other clustering techniques

Unlike the systems described above, there are some clustering methods that define metrics to determine the optimum number of clusters and then to find the optimum clustering given that number.

In [TW07], the optimum amount of speakers is computed using BIC and the optimum clustering that optimizes the overall model likelihood is obtained by a genetic algorithm. In [Roy97], a speaker indexing algorithm is proposed to dynamically generate and train a neural network to model each postulated speaker found within a recording. Each neural network is trained to differentiate the vowel spectra of one specific speaker from all other speakers. In [Lap03], self-organizing maps are proposed for speaker clustering using a Vector Quantization (VQ) algorithm for training the code-books representing each of the speakers.

1.5 Examples of state-of-the-art speaker diarization systems

Many systems exist in the literature and were evaluated in international/national competitions. In the following subsections, we choose to review three of those famous systems.

1.5.1 The LIMSI speaker diarization system

The Speaker Diarization (SD) system described here was used by the LIMSI[6] in the Rich Transcription (RT) evaluation conducted by NIST in 2007 on meeting and lecture recordings. This system [ZBLG08] is built upon the baseline diarization system designed for broadcast news data and it combines an agglomerative clustering based on BIC with a second clustering using

[6] http://www.limsi.fr/

state-of-the-art speaker identification (SID) techniques. This system is similar to the LIUM[7] system that gives the best results in the last ESTER competition.

Speech is extracted from the signal by using a Log-Likelihood Ratio (LLR) based speech activity detector (SAD). The LLR of each frame is computed between the speech and non-speech models with some predefined prior probabilities. To smooth LLR values, two adjacent windows with a same duration are located at the left and right sides of each frame and the average LLR is computed over each window. Thus, a frame is considered as a possible change point when a sign change is found between the left and right average LLR values.

Initial segmentation of the signal is performed by a local divergence measure between two adjacent sliding windows. A Viterbi re-segmentation is applied to adjust segments boundaries. A first agglomerative clustering is processed using BIC. Then, speaker recognition methods are used: feature warping normalization is performed on each segment in order to map the cepstral feature distribution to a normal distribution and reduce the non-stationary effects of the acoustic environment. The GMM of each remaining cluster is obtained by maximum a posteriori (MAP) adaptation of the means of a universal background model (UBM) composed of 128 diagonal Gaussians. The second stage of agglomerative clustering is carried out on the segments according to the cross log-likelihood ratio.

1.5.2 The IBM speaker diarization system

The SD system described here was used by the IBM[8] in the RT07. In summary, the system [HMVP08] has the 3 following characteristics:

- the use of an SAD algorithm based on a speech/non-speech HMM decoder, set to an optimal operating point for "missed speech / false alarm speech" on development data.
- the use of word information generated from Speech to Text (STT) decoding by means of a speaker-independent acoustic model. Such information is useful for two reasons: it filters out short silence, background noise, and vocal noise that do not discriminate speakers and it provides more accurate speech segments to the speaker clustering step.
- the use of the GMM-based speaker models that are built from the SAD segmentation. The labels of each frame are refined using these GMM models, followed by smoothing

[7] http://www-lium.univ-lemans.fr/
[8] http://www.research.ibm.com/

the labeling decision with its neighbors. This method was used to detect change points accurately within the speech segments.

1.5.3 The LIA speaker diarization system

The SD system described here was also used by the LIA[9] in the RT07. This system [FE08] is structured of 4 main steps:

1. a speech/non-speech detection using the Linear Frequency Cepstrum Coefficients (LFCC) is based on a two state HMM. Those two states represent speech and non-speech events. Each of those states is initialized with a 32-component GMM model trained using Expectation-Maximization (EM) and Maximum Likelihood (ML) algorithms.

2. a pre-segmentation based on the GLR criterion is used in order to initialize and speed-up the later segmentation and clustering stages.

3. a unique algorithm for both speaker segmentation and clustering is performed using an evolutive hidden Markov model (E-HMM) where each E-HMM state characterizes a single speaker and the transitions represent the speaker turns.

4. a post-normalization and re-segmentation is applied to facilitate the estimation of the mean and variance on speaker-homogeneous segments.

Table 1.1 illustrates the main difference between the three above systems.

Even though the robustness of those system shown by their performance, there are still some points that can be improved:

- preprocessing by removing all non-speech parts using thresholding methods. First, it will be better to split the stream into homogenous zones, and then make the decision on those zones: this decision will be more confident. Second, the use of diarization information (i.e. audio clusters) is helpful to make decision on regions of doubt where the values are on borders (i.e. close to the threshold).

- there is generally reverse connections between the different steps of the system: it will be better to use, for example, not only the segments in order to create the clusters, but also the clusters in order to rectify the segments (by splitting or changing borders).

[9] http://www.lia.univ-avignon.fr/

Table 1.1: Comparison between three state-of-the-art systems.

	LIMSI	**IBM**	**LIA**
Acoustic features	MFCC + (their 1st and 2nd derivatives) + (1st and 2nd derivatives of the energy)	MFCC	LFCC
Speech Acoustic Detection	LLR using 256-components GMM for speech and non-speech	HMM decoder using 3-components GMM for speech and non-speech	HMM decoder using 32-components GMM for speech and non-speech
Speaker segmentation	Gaussian divergence measure + Viterbi re-segmentation using 8-components GMM	Speech to Text decoding for segments purification	GLR with fixed threshold
Speaker clustering	BIC clustering + SID clustering using 128-components GMM	Estimation of the initial number of clusters + BIC clustering + refinement using 10-components GMM	E-HMM where every state characterizes a speaker (128-components GMM)
Performance at RT'07 on lecture sessions	26%	31%	31%

We will propose in chapter 2 some solutions to overcome those weaknesses without any adaptation on any kind of data (e.g. news, debates and movies).

1.6 Databases

In order to test our proposed methods and compare them to the state-of-the-art ones, we use three audio recording databases.

1.6.1 ESTER-1 Corpus

ESTER[10] which is the French acronym for *"Evaluation des Systèmes de Transcription Enrichie d'émissions Radiophonique* is an evaluation campaign of French broadcast news transcription systems". The ESTER-1 Corpus (years 2003-2005) includes 100 hours of manually annotated recordings and 1,677 hours of non transcribed data. The manual annotations include the detailed verbatim orthographic transcription, the speaker turns and identities, information about acoustic conditions, and name entities.

The acoustic resources come from six different radio sources, namely: France Inter, France Info, Radio France International (RFI), Radio Télévision Marocaine (RTM), France Culture and Radio Classique. For more details on that corpus please refer to [GGM+05].

1.6.2 ESTER-2 Corpus

The ESTER-2 Corpus (years 2008-2009) was recorded with the same conditions than the ESTER-1 corpus. It includes 100 hours of manually annotated recordings that come from 5 different radio sources, namely: France Inter, France Info, TVME, Radio Africa 1 and 45 of EPAC-ESTER corpus. Table 1.2 indicates the amount in hours of the annotated data for training, development and test tasks.

1.6.3 EPAC-ESTER Corpus

EPAC[11] is the French acronym of *"Exploration de masse de documents audio pour l'extraction et le traitement de la parole conversationnelle"*. It is a french ANR-MDCA project (years 2006 - 2009) that gathers four laboratories (IRIT[12] , LIA[13], LIUM[14] , LI[15]) in order to investigate new techniques for automatic speech processing specially in the context of mono-channel meeting

[10] http://www.afcp-parole.org/ester
[11] http://epac.univ-lemans.fr/
[12] http://www.irit.fr
[13] http://www.lia.univ-avignon.fr/
[14] http://www-lium.univ-lemans.fr/
[15] http://www.li.univ-tours.fr/

Table 1.2: Amount of transcribed and non transcribed recordings of ESTER-2.

Source	**Training Set**	**Development Set**	**Test Set**
France Inter	26 h	2 h	3h40
RFI	69 h	40 min	1h10
Africa 1	10 h	2h20	1h
TVME	-	1 h	1h30
EPAC-ESTER	45 h	-	-
Total	**150 h**	**6 h**	**7h20**

recordings. The EPAC corpus includes 100 hours of manually annotated data. These conversational data were selected from the 1,677 hours of the non-transcribed broadcast recording of ESTER-1 Corpus.

Chapter 2

Proposed System for speaker diarization

Contents

2.1 Proposed Generic GLR-BIC segmentation 34

2.1.1 Proposed Method 34

2.1.2 Bidirectional segmentation 36

2.1.3 Penalty coefficient decreasing technique 36

2.1.4 Other applications of the method 38

2.2 Proposed clustering 39

2.2.1 Improved EVSM clustering 39

2.2.2 BIC clustering 40

2.2.3 CLR Post-Clustering 42

2.3 System architecture 44

In this thesis, we investigate new techniques and propose improvements for speaker segmentation and speaker clustering. We try to make those techniques the most robust and the most portable for all kind of database (broadcast News, meetings, films, series, TV games).

On one hand, we present a new technique for speaker segmentation. This technique combines the generalized likelihood ratio (GLR) and the Bayesian information criterion (BIC). Although this method is firstly proposed for speaker segmentation within the audio recording, we show during our work, that it can be used for other modalities (video, images). That is why, this method is described in a generic segmentation framework.

On the other hand, we investigate the Eigen-Vector-Space-Model based technique for speaker clustering and propose some improvements by adding the "pitch constraint". Then, we review the existing BIC clustering and we propose some modifications by applying a "local-and-global clustering" and by enhancing the cluster purity.

Finally, we propose an iterative scheme to improve not only the speaker segmentation and the speaker clustering, but also the speech activity detection and the music activity detection. That is what makes our speaker diarization system original.

2.1 Proposed Generic GLR-BIC segmentation

A well-known BIC speaker segmentation was proposed by Sivakumaran et al. to detect multi-points changes in audio recordings [SFA01]. It was then improved by Cettolo et al. in [CV03]. In our work we applied this method and we figure out some limitations. Although the amount of parameters to be tuned is important, the penalty coefficient is not as stable as expected and there is a possible cumulative error due to the sequential segmentation process: if a point is erroneously detected, the next point might be affected by this error and might not be detected correctly. All those limitations encouraged us to propose a new segmentation based on GLR and BIC.

2.1.1 Proposed Method

Fig.2.1(a) illustrates a signal where the theoretic segmentation is represented by the points $R_1, R_2, ..., R_n$. The proposed segmentation method follows four main steps.

1. **Splitting step.** It consists in splitting the signal into equal size windows. Then, we detect the most probable point of change in each window. This step is shown in Fig.2.1(b). Mathematically, this point corresponds to the maximum of the GLR expression (or to the maximum of ΔBIC). The advantage of this step is that there is no need to fix a threshold as for the standard use of GLR described in section 1.3.2.2.

2. **Most probable point detection step.** In the first step, we have obtained points of change $P_1, P_2, ..., P_m$ which separate the best way the two mono-Gaussian models existing in every window. However, those models are not very representative because they are influenced only by local data in a window with a fixed size and fixed boundaries. Thus,

we repeat the first step using windows that are chosen as seen in Fig.2.1(c): to detect a point of change P'_i, we use the window $[P_{i-1}, P_{i+1}]$. The resulting models will be quite close to Gaussian distributions. If two consecutive windows vote for the same point, or for two close points (difference $< 0.2sec$), we decide to merge those two points by considering their mean and consequently, the number of detected points will decrease.

3. **Re-Adjustment step.** This step illustrated in Fig.2.1(d) consists in repeating the second step several times until that the repartition of change points is stabilized i.e. the convergence to Gaussian distributions is accomplished. The obtained points are annotated $q_1, q_2, ..., q_t$ where $t < m$.

4. **Definitive change detection step.** At this stage, the points $q_1, q_2, ..., q_t$ represent the most probable locations of change. Thus the BIC criterion is applied to select only the points that are effectively points of source changes. This step is illustrated in Fig.2.1(e).

The algorithm applied in step 4 is implemented as follows:

```
Let m = number of points q_i,
   i = 1,
   initialize W = [q_0, q_2],
   while ( i ⩽ m)
       in W, search ΔBIC_max ,
       if ΔBIC_max ⩾ 0 then
           q_i = argmax ΔBIC_max , S = q_i
           increment i,
       else
           increment i,
       End if
       E= qi+1 , W=[S, E]
   End while
```

When this algorithm is applied for speaker segmentation, the choice of the analysis window size should be given a special attention. On the first hand, if this window is too large, it may contain more than two sources, and consequently yield a high number of missed detections. On the other hand, if the window is too short, the lack of data will cause poor Gaussian estimation

and accordingly, poor segmentation accuracy. That is why we choose a typical value of 2 seconds for the initial window size in the splitting step.

Tests done on broadcast news show the efficiency of this method compared to the one used by [SFA01] and [CV03] where only BIC is applied on a shift variable size window. Please refer to the results obtained in section 3.

However, when applying this method on meeting data, some errors occurred in regions containing multiple speakers. For example, in the scenario illustrated in the ground truth of Fig.2.2 where "*Spkr1 continues speaking even when Spkr2 starts his turn*", the GLR-BIC segmentation may fall in detecting speaker change because the theoretical boundary region is still experiencing some homogeneity.
In the following subsections, two hypotheses are proposed to solve this problem.

2.1.2 Bidirectional segmentation

Due to the shifted variable size window used in the GLR-BIC algorithm, processing from "left to right"may detect different points of change than processing from "right to left", and therefore, there is a chance that a missed boundary in the first direction can be detected in the other direction and vice versa. Figures 2.2, 2.3 and 2.4 illustrate the three possible corrections: S1 (respectively S2) corresponds to the set of boundaries provided by the "left to right "segmentation (respectively "right to left") and S1 $\bigcup$ S2 is the resulting union. Those corrections can be divided into two types: perfect and partial corrections. Practically, we have noticed that partial corrections outnumber the perfect ones.

2.1.3 Penalty coefficient decreasing technique

In Equation 1.12, we notice that when the penalty λ decreases, ΔBIC increases and consequently, it is possible that ΔBIC becomes positive and an additional point of change is detected in this case. However, the diminution of λ in an unsupervised manner can be harmful to the system performance because it may introduce many false alarms. That is why we must be sure that the region under investigation is unstable i.e. it contains an interaction zone. In section 2.3, a framework is proposed to handle the detection of the unstable segments.

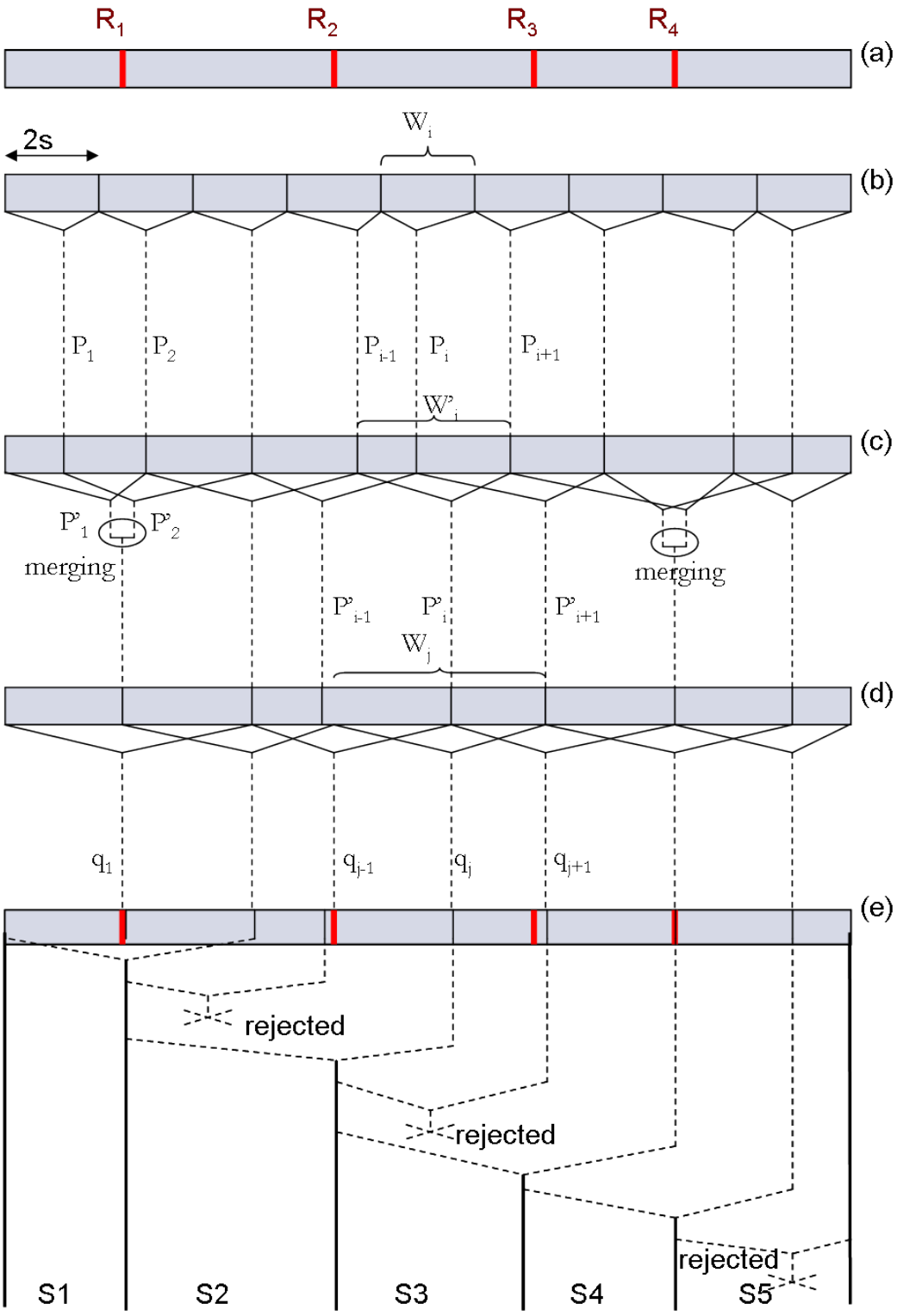


Figure 2.1: GLR-BIC segmentation.

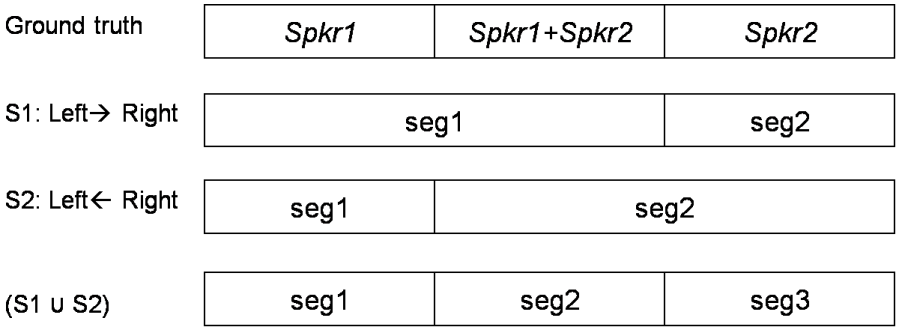


Figure 2.2: Correction due to S1 and S2 (perfect correction).

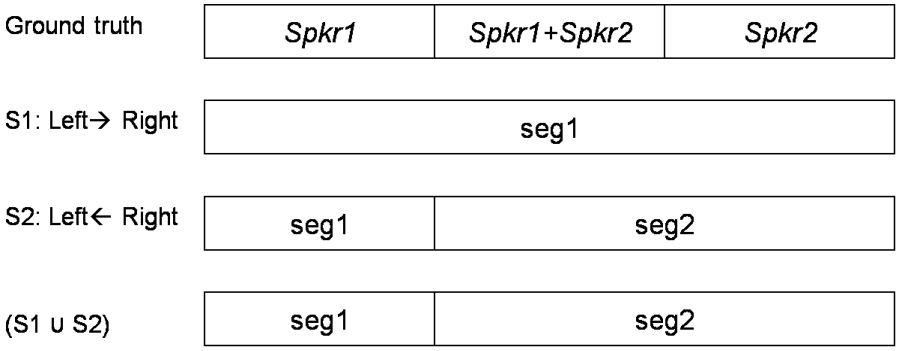


Figure 2.3: Correction due to S1 (partial correction).

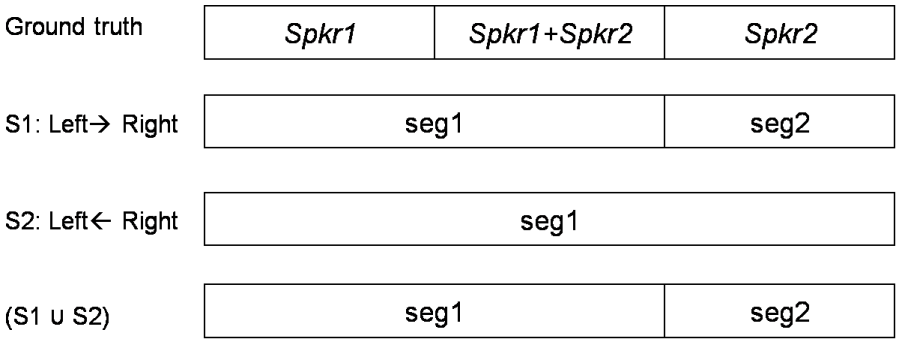


Figure 2.4: Correction due to S2 (partial correction).

2.1.4 Other applications of the method

Because this segmentation method is based on an acoustic homogeneity criterion, it has the ability of segmenting the audio stream into different sources: different types of music, different speakers and silence. This advantage is used to help speech/non-speech separation as well as music/non-music (cf. 2.3).

Moreover, at a more general level, this proposed segmentation method was also tested on other modalities: as for shot boundary detection using visual features (cf. chapter 5), for programs boundary detection using audiovisual features (cf. Appendix A).

2.2 Proposed clustering

In the following subsections two unsupervised clustering techniques are proposed. The first one is a vector space based technique and the second one is a BIC based technique. Moreover, a supervised clustering technique based on the cross likelihood ratio (CLR) is investigated and then improved.

2.2.1 Improved EVSM clustering

This clustering method is based on the work of Tsai and al. [TCCW05] that uses Eigen Vector Space Model (EVSM) with a hierarchical bottom-up clustering. Figure 2.5 presents the different steps: from all the segments $S_1, S_2, ..., S_N$ a universal Gaussian Mixture Model (GMM) Λ is created. This GMM is then adapted on each segment S_i to obtain the GMM Λ_i. From each Λ_i, a super-vector V_i is created by concatenating the mean vectors of each gaussian distribution of that Λ_i. Then, PCA (Principal Component Analysis) is applied to obtain for each vector V_i, a vector W_i with a lower dimension. Then, the cosine formula computes the similarity between each couple of vectors (W_i, W_j). The stopping criterion is based on a threshold comparison: if the cosine is higher than this threshold, the two segments are grouped.

A stronger merging criterion. Our contribution consists in choosing a stronger merging criterion based both on the previous similarity measure and on prosodic information. The pitch $F0$ feature is estimated every 10ms on voiced regions with the ESPS signal processing software which utilizes the normalized cross correlation function and dynamic. Then, a difference (called $\Delta F0$) between the averages of the $F0$ values of each couple of segments is computed. We have to notice that, whatever the chosen softwares, some pure music segments will be considered erroneously as voiced regions of the signal; but they will never be grouped with speakers segments because the cosine similarity will separate them. The new merging criterion becomes: two clusters correspond to the same speaker if 1) the similarity (cosine formula) is higher than a

threshold $th1$, and 2) $\Delta F0$ is lower than a threshold $th2$. Those thresholds were tuned on the training set of ESTER-1 as well as the number of Gaussians that was optimally fixed to 128.

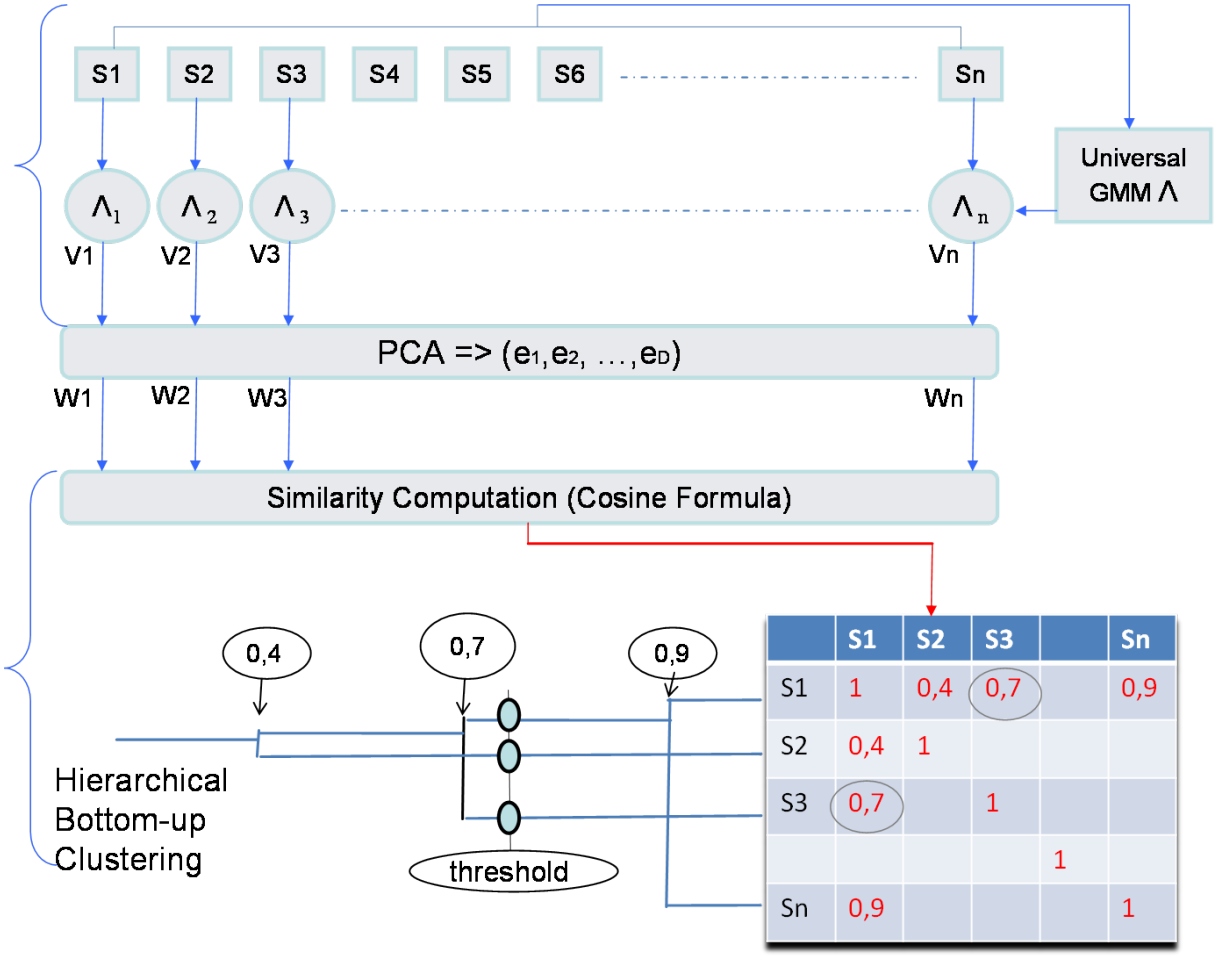


Figure 2.5: EVSM-based hierarchical clustering.

Results obtained on ESTER-1 [EKSAO07] show that the EVSM-based method is very competitive to the state-of-the-art speaker diarization systems dedicated for broadcast news. However, in some cases where the duration of the segment is small, the corresponding GMM is not well modeled. This weakness encourages us to use the BIC clustering in order to deal with the conversational data.

2.2.2 BIC clustering

The BIC clustering was previously described in section 1.4.1. But in this case, X_1 and X_2 denote the clusters under investigation and X the resulting cluster.

But for some kinds of recording data as meetings, there is high interaction between speakers:

Table 2.1 shows that the average length of speaker turns is relatively lower than the one of broadcast news, and the regions where many people talk simultaneously are more numerous.

Table 2.1: Comparison between broadcast news (ESTER-1) and meetings (EPAC-ESTER) corpora.

	TEST ESTER-1	TEST EPAC-ESTER
Average length of speaker turns	20.22 sec	8.33 sec
Time ratio of multi-speakers turns	0.21 %	5.26 %

The two factors mentioned above decrease the segments purity, and so introduce a risk of cumulative errors in the clustering process. It is obvious that homogeneous segments with long duration are more confident and provide better clustering. To deal with this problem, two contributions were proposed:

- **Local-global clustering.** In the standard hierarchical clustering, the initial clusters correspond to segments, and as described above for meeting data, those segments have relatively small duration. Due to the iterative structure of the clustering, it is very probable that the comparison is done between clusters of very different sizes. In this case, the BIC-based inter-cluster similarity is not precise as explained in [HN07], and may introduce cumulative errors in the clustering process. Our solution to cure this weakness is to do a local clustering every N consecutive segments (practically $N = 20$) before processing the global one. The reason behind this proposition is to build a first level of confident clusters with balanced sizes.

- **Similarity matrix and clusters updating.** At the end of a clustering process, each segment is theoretically assigned to the cluster providing the highest BIC similarity. However, due to the hierarchical bottom up manner, there are some segments that do not respect this hypothesis. To correct these errors and therefore enhance the clusters purity, we propose to compute the similarity matrix between segments S_i and clusters C_j and then reclassify segments regarding this matrix. For example, in Fig.2.6, the similarity matrix shows that the segment S_8 will be assigned to the cluster C_3 ($-\Delta BIC = 0.7$) instead of C_1 ($-\Delta BIC = 0.1$) as in the previous clustering.

-ΔBIC	S_1	..	S_8	S_9	...	S_{14}	S_{15}	...	S_{20}
C_1	0.5	...	0.1	-0.3	...	-0.6	-0.2	...	-0.7
C_2	0.2	...	-0.4	0.8	...	0.5	0.3	...	-0.1
C_3	-0.6	...	0.7	-0.1	...	0.2	0.9	...	0.3

Figure 2.6: Similarity matrix between segments and clusters.

2.2.3 CLR Post-Clustering

The CLR clustering is already described in 1.4.3. This clustering was historically used with a fixed threshold ($Thr_{CLR} = 1.5$) and without additional constraint like in [BZMG06]. In the following subsections, we propose some improvements by using an adaptive threshold that depends on the recording file duration and by reducing the background effect and adding the BIC measure constraint.

A. Adaptative thresholding. During our work to find the optimum threshold, we studied the duration time of the recording file: we have noticed that the range of the optimum thresholds is generally higher for the recordings of short duration as seen in Fig.2.2.

Table 2.2: Optimal ranges for two recording files of different duration time.

	the optimal margin of the Thr_{CLR}
File 1 (60 minutes)	[1.00; 1.55]
File 2 (20 minutes)	[1.45; 4.00]

In [EKMS08], the dependency between the threshold and the file duration is modeled by a linear equation:

$$Thr_{CLR} = aL + b \tag{2.1}$$

where L is the duration time (in minutes) of the recording audio file. a and b two parameters that were tuned on 30 files of the training data of ESTER-1 ($a = -0.013$ and $b = 2.22$).

But during the ESTER-2 competition, we used a more complex dependency curve. Fig.2.7 shows the curve tuned on the development set of ESTER-2. It illustrates the threshold value in respect to the duration time.

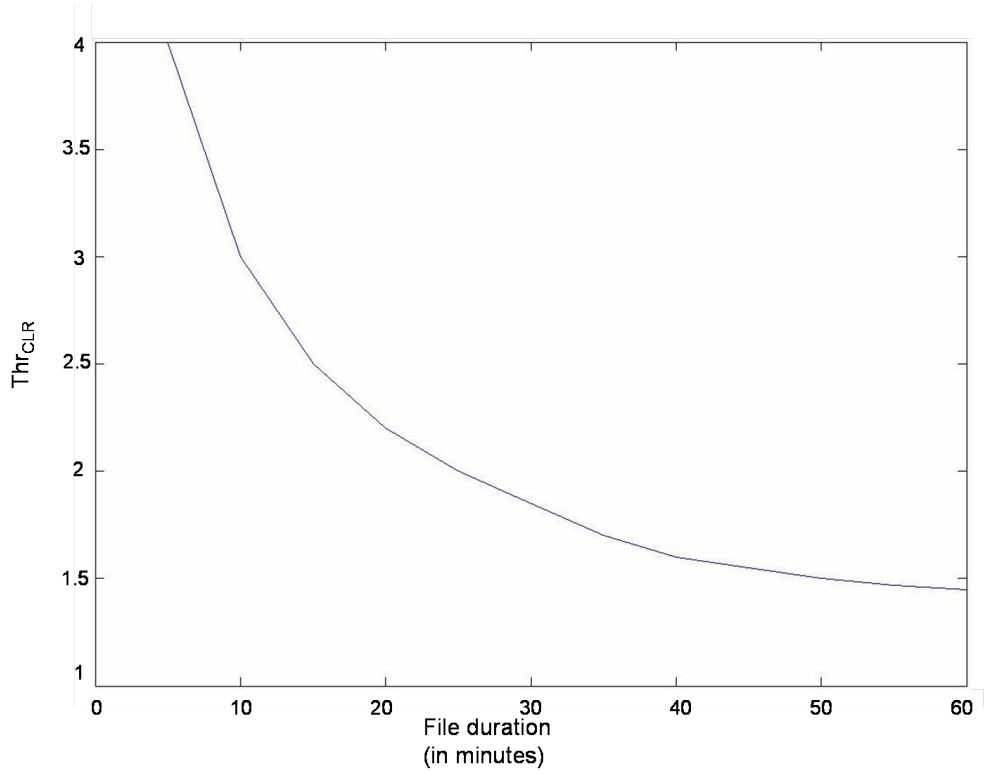


Figure 2.7: The threshold Thr_{CLR} in respect to the duration time.

B. CLR+BIC fixed thresholding. Also, we observed that during the CLR clustering process, there are some errors due to the fact that 2 different speakers with the same background may be merged because their CLR distance value is high. That is why we propose to reduce the background effect by eliminating regions where the background is dominant and then use the BIC distance as an additional constraint between those clusters.

First, the reduction of the background effect is achieved using the software (ESPS) that computes the F0 Feature. In fact, this software provides a F0 value if this value is between 60 and 400 Hz, otherwise it is set to zero. In another sense, if there is noise or background music, the F0 value is probably equal to zero.

After reducing the background effect, the CLR clustering can be applied. In the ESTER competition, the CLR clustering was applied with a fixed threshold.

Second, the BIC distance is computed another time at this level. Here, the BIC is used only to validate the merging between the clusters candidate or to reject it. In fact, although the CLR distance between 2 clusters is good enough (less than a fixed threshold) to merge those clusters, the BIC distance between them might be very high. In this case, the merging is rejected. Results described later in section 3 show the profits of those improvements.

2.3 System architecture

After reviewing the strengths and the weaknesses of each essential component of a speaker diarization, we propose our iterative system (cf. Fig.2.8). It can be summed up by the following algorithm:

1. Parameters extraction where the MFCCs, the 4Hz modulation energy, the number of segments, their duration, and the log-likelihoods of speech, non-speech, music and non-music GMMs are computed.

2. First Bidirectional GLR/BIC segmentation using a penalty coefficient $\lambda = \lambda_1$ (practically $\lambda_1 = 1$).

3. Speech/non-speech separation by using the 4Hz modulation Energy (ME) and speech and non-speech GMM scores for each segment.

4. Music/non-music separation by using the number and the duration of segments as well as the music and non-music GMM scores.

5. Local BIC clustering applied every N consecutive segments.

6. Global BIC clustering based on clusters provided from previous step.

7. Computation of the similarity matrix between segments S_i (i=1 to N_s) and clusters C_j (j=1 to N_c) where N_s is the number of segments and N_c is the number of clusters.

8. Updating clusters by assigning each segment S_i to $arg\max_{C_i}(-\Delta BIC(S_i, S_j))$ when j varies from 1 to N_c.

9. Splitting unstable segments using the bidirectional GLR/BIC segmentation with $\lambda = \lambda_2$, $\lambda_2 < \lambda_1$ (practically $\lambda_2 = 0.8$) as explained in subsection 2.1.3. Unstable segments are segments for which $-\Delta BIC(S_i, S_j)) < 0$ i.e. the similarity between segment S_i and its corresponding cluster is low.

10. Stop loop if no more splitting can be done. Otherwise, do a speech/non-speech separation and a music/non-music separation and go back to step 7 and so on.

11. Final CLR clustering in order to group clusters corresponding to the same speaker but under different backgrounds.

We notice that the number of segments N_s and the number of clusters N_c are dynamic: N_s can decrease at the end of step 3 and increase at the end of step 9. However N_c can only decrease at the end of step 8 due to the segments re-assignment.

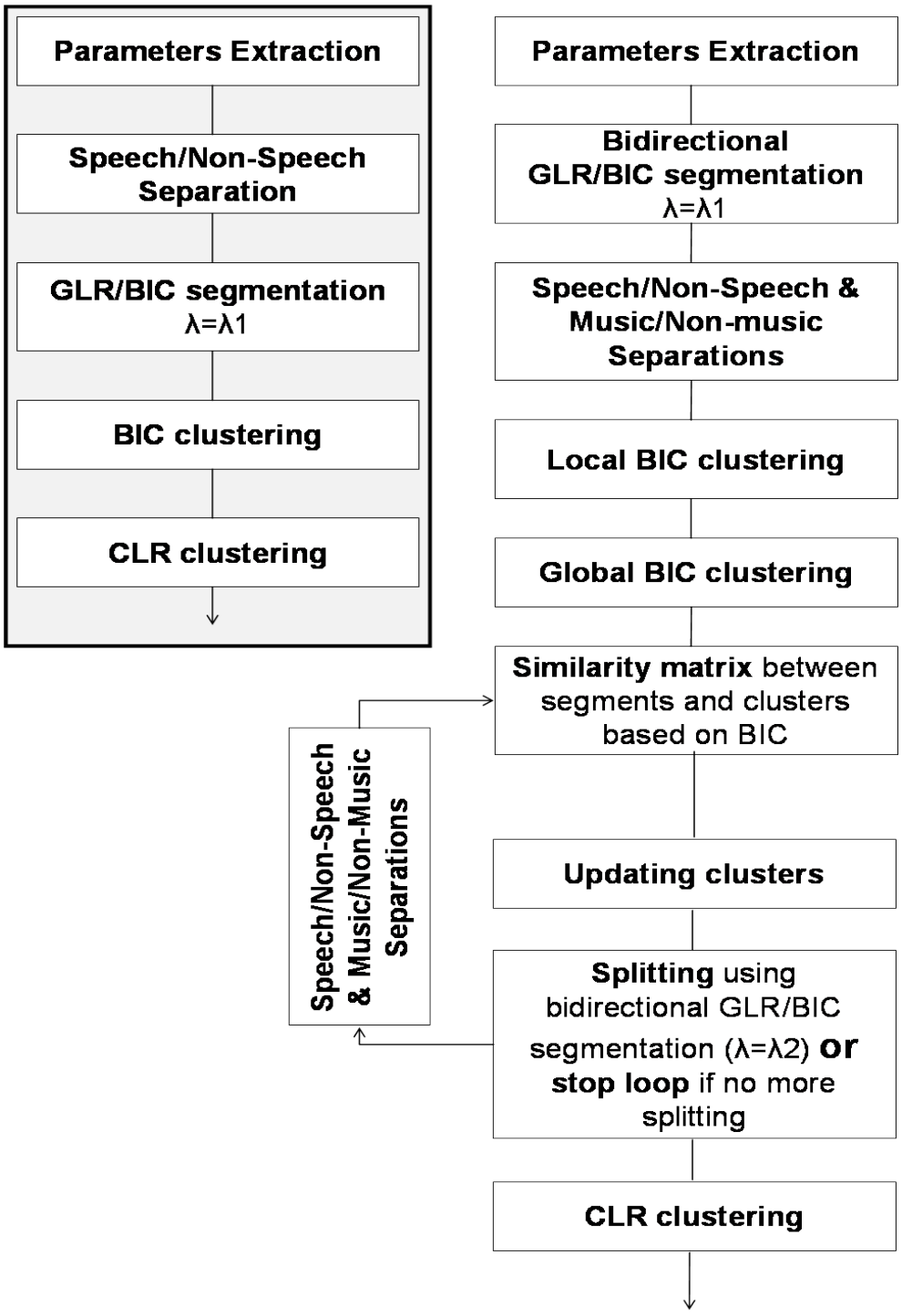


Figure 2.8: Standard vs Proposed Speaker Diarization systems.

Chapter 3

Experiments and Results

Contents

3.1 Evaluation of the speaker segmentation 47

3.1.1 Experiments and Results 48

3.2 Evaluation of the acoustic events detection 52

3.2.1 Experiments and results 53

3.3 Evaluation of the speaker clustering 55

3.4 Evaluation of the speaker diarization system 56

3.4.1 Experiments and results 56

Experiments are done in order to see the impact of our contribution. Four types of evaluation are described in the following subsections:

3.1 Evaluation of the speaker segmentation

This evaluation is done by computing the recall, the precision and the F-measure. In general, the number of false positives (FP), false negatives (FN), true positives (TP), and true negatives (TN) are considered with respect to positive (P) and negative (N) instances manually annotated in the ground truth. Then, the precision ($Prec$) and the recall (Rec) are defined as:

$$Prec = \frac{TP}{(TP + FP)} \tag{3.1}$$

and

$$Rec = \frac{TP}{P} \tag{3.2}$$

Prec decreases when the number of *FP* increases. *Rec* decreases as the number of *FN* increases. We observe in many occasions that *Prec* decreases when *Rec* increases. That is why the F-measure F is defined as:

$$F = \frac{2}{\frac{1}{Prec} + \frac{1}{Rec}} \tag{3.3}$$

A new metric approach based on maximum overlap between extracted segments was proposed in [JBPKQ07] for the ARGOS[16] evaluation campaign. This metric was initially used for "Shot Boundary Detection" in the context of video indexing. In our work, we use it for speaker segmentation because it takes into account the homogeneity of the temporal instances (segments) unlike other methods that favor transition detection between instances.

This "ARGOS" metric considers a "reference segmentation" and a "system segmentation". From both reference and system segmentations, a "maximum intersection" segmentation is extracted: it must completely be present in a reference segment and in a system segment as seen in Fig.3.1. Then, the precision, the recall and the F-measure F are defined by:

$$Prec = \frac{|Intersection|}{|System|} \tag{3.4}$$

$$Rec = \frac{|Intersection|}{|Reference|} \tag{3.5}$$

$$F = \frac{2.|Intersection|}{|Reference| + |System|} \tag{3.6}$$

where $|.|$ denotes the sum of the lengths of the segments.

3.1.1 Experiments and Results

Three different methods were tested to show the improvements we made throughout our work: 1) the existing method [CV03] that uses a shift variable size window with BIC criterion, 2) the proposed GLR-BIC segmentation method and 3) the overall iterative speaker diarization system

[16] www.irit.fr/argos

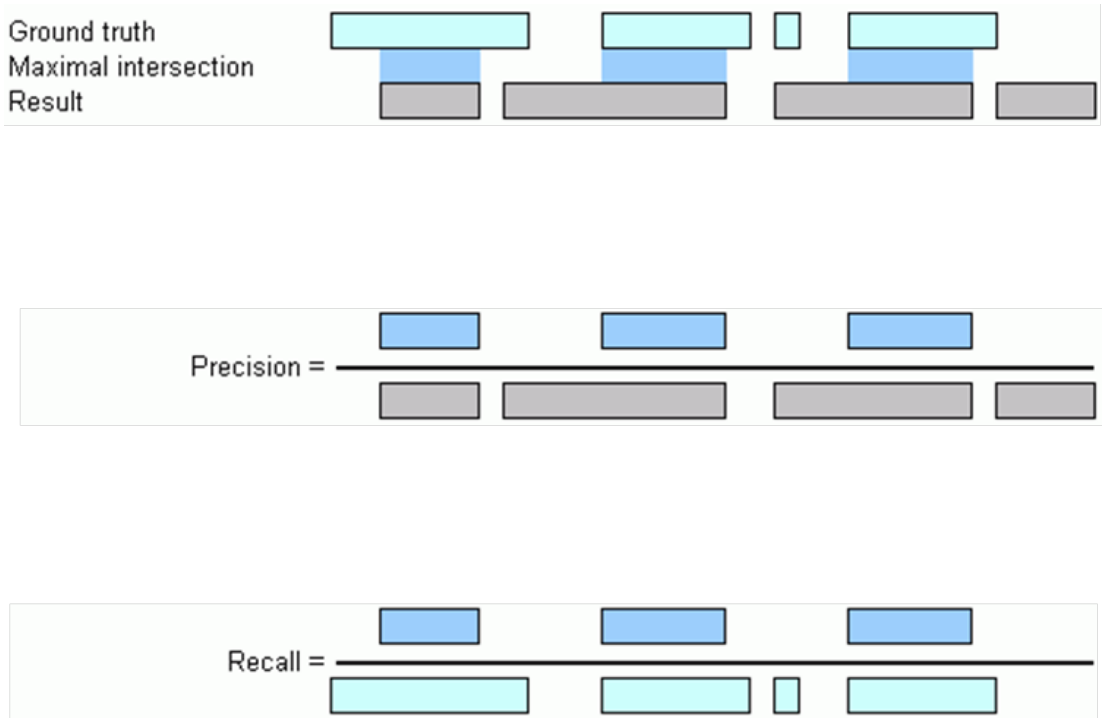


Figure 3.1: The Argos precision and recall measures in respect to the ground truth segmentation and the system segmentation.

where speaker segmentation and speaker clustering help each other. The experiments were done on both broadcast news and broadcast meetings.

Broadcast news In this evaluation, the corpus used was the ESTER-2 development set. This set contains 20 audio recording files from 4 different radio stations.

First, table 3.1 shows the results of the existing BIC method in terms of Recall, Precision and F-measure for the 4 different sources. It can be seen that the best segmentation is obtained on RFI radio station with a F-measure of about 85% and the worse one is obtained on France Inter radio station eventhough the two radio sources seem of the same difficulty on listening.

Table 3.2 our proposed GLR-BIC segmentation method, we can denote an absolute improvement of 12.46 % (the mean average F-measure raises from 72.91% to 85.37%).

When using the proposed iterative segmentation-clustering process, we achieve an additional relative improvement of 3.15% that corresponds to an absolute improvement of 2.73%. Another thing to notice is the rise of scores on Inter radio with an overall absolute improvement of

Table 3.1: Results of BIC segmentation on the 4 different radio stations of the ESTER-2 development set.

	Recall (%)	Precision (%)	F-measure (%)
Africa	72.92	72.26	72.61
RFI	86.47	83.56	84.99
France Inter	66.24	65.24	65.74
TVME	81.02	71.29	75.84
mean weighted average	**74.25**	**71.62**	**72.91**

Table 3.2: Results of GLR-BIC segmentation on the 4 different radio stations of the ESTER-2 development set.

	Recall (%)	Precision (%)	F-measure (%)
Africa	84.98	85.47	85.22
RFI	94.6	92.71	93.65
France Inter	82.04	83.13	82.58
TVME	85.74	84.27	85.00
mean weighted average	**85.36**	**85.37**	**85.37**

21.43% (from 65.74% to 87.14%). It shows that the segmentation process is more stable and less dependent from the penalty coefficient unlike the BIC segmentation where 9 parameters must be tuned.

Table 3.3: Results of the iterative process on the 4 different radio stations of the ESTER-2 development set.

	Recall (%)	Precision (%)	F-measure (%)
Africa	88.44	88.63	88.53
RFI	95.03	92.80	93.90
France Inter	86.55	87.73	87.14
TVME	85.48	85.83	85.65
mean wieghted average	**87.95**	**88.26**	**88.10**

Moreover, Fig.3.2 and Fig.3.3 show respectively the precision and the recall of the three methods on the 20 audio files. The GLR-BIC and iterative system are almost better than the BIC segmentation on all files. However, the iterative process is slightly better than the GLR-BIC segmentation on only 12 files.

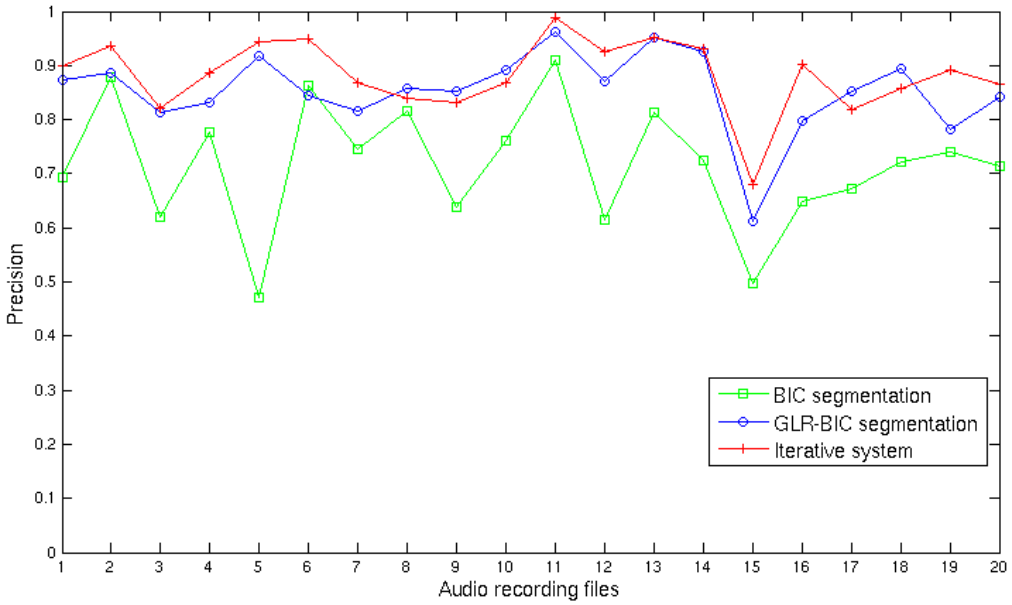


Figure 3.2: Comparison between the precision of BIC, GLR-BIC and Iterative system segmentation methods on the 20 files of ESTER-2 development set.

Moreover, the three above methods were tested on the EPAC meeting corpus. Table 3.4 shows that the BIC segmentation provides a low F-measure of 51.26%. Results are improved by 6.93% when GLR-BIC segmentation is used. An additional gain of 10.83% is obtained with the iterative segmentation and clustering process.

Table 3.4: Results of the 3 methods on EPAC broadcast meetings.

	Recall (%)	Precision (%)	F-measure (%)
BIC segmentation	50.44	52.11	51.26
GLR-BIC segmentation	55.73	60.87	58.19
Iterative System	**66.13**	**72.18**	**69.02**

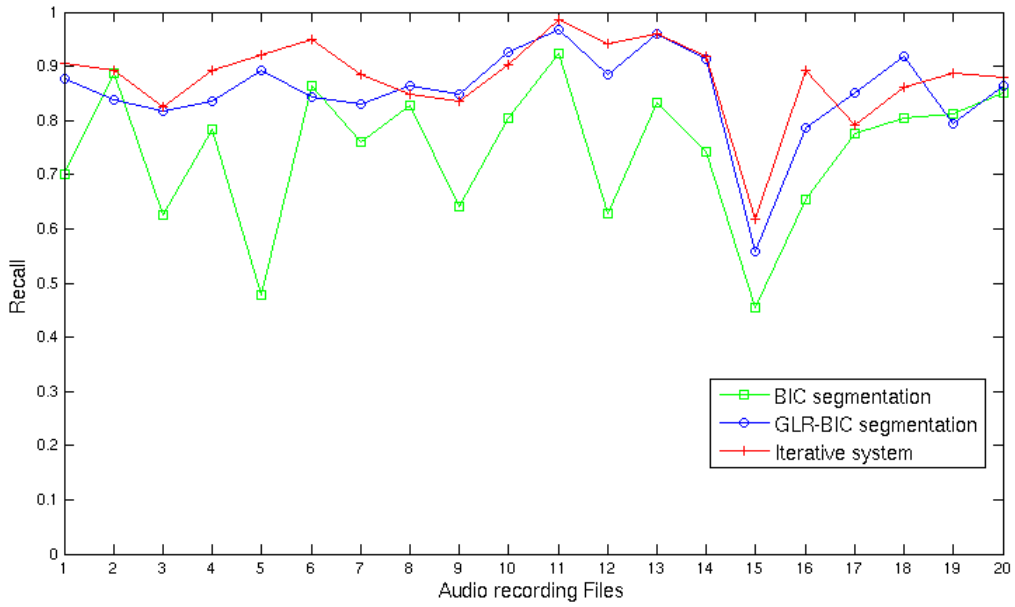


Figure 3.3: Comparison between the recall of BIC, GLR-BIC and Iterative system segmentation methods on the 20 files of ESTER-2 development set.

3.2 Evaluation of the acoustic events detection

The evaluations of the speech detection and the music detection are made by computing the error rate and the F-measure F for both tasks.

- The error rate for speech activity detection is defined by:

$$ErrorRate = \frac{sum(False_{Neg}) + sum(False_{Alarm})}{sum(T(Speech)) + sum(T(nonSpeech))} \tag{3.7}$$

where

$False_{Neg}$ is the false negative time i.e. the time when the speech was missed.

$False_{Alarm}$ is the false alarm time i.e. the time when the speech was erroneously detected.

$T(Speech)$ is the time when the speech was present

$T(nonSpeech)$ is the time when the speech was not absent.

- The F-measure F is defined in the same way as for Equ.3.3. But in this case the recall and the precision are given by:

$$Rec = \frac{sum(TP)}{sum(T(c))} \tag{3.8}$$

$$Prec = \frac{sum(TP)}{sum(TP) + sum(FA)} \tag{3.9}$$

where TP is the true positive time i.e. the time when the speech is correctly detected.

3.2.1 Experiments and results

The experiments are done on the test corpus of ESTER-2. Those results are the official results published at the competition where each team could submit many runs.

For speech/non-speech detection, were submitted 2 meaningful runs. The first run processes on the output of the GLR-BIC segmentation by computing the likelihood of the speech and the non-speech GMMs of each segment and by computing the mean average of the 4 Hz energy modulation on each segment. Table 3.5 shows an error rate of 1.8% and a F-measure of 99.03%.

Table 3.5: Results of the 1st run for IRIT speech detection at ESTER-2 competition.

	Error Rate (%)	Recall (%)	Precision (%)	F-measure (%)
africa	04.18	96.75	98.92	97.82
France Inter	00.90	99.56	99.44	99.50
RFI	00.97	99.88	99.13	99.50
TVME	02.25	98.62	98.89	98.75
mean average	**01.80**	**98.87**	**99.20**	**99.03**

The second run is the output of the iterative segmentation and clustering scheme described in section 2.3. The new results are shown in table 3.6. The error rate becomes 1.31% (best system: 1.08%) and the F-measure becomes 99.29% (best system: 99.42%).

Table 3.6: Results of the 2nd run for IRIT speech detection at ESTER-2 competition.

	Error Rate (%)	Recall (%)	Precision (%)	F-measure (%)
africa	02.05	98.58	99.31	98.94
France Inter	00.85	99.61	99.45	99.53
RFI	00.65	99.92	99.42	99.67
TVME	02.47	98.57	98.70	98.64
mean average	**01.31**	**99.28**	**99.30**	**99.29**

For music/non-music detection, three meaningful runs were submitted: the first system processes the output of the GLR-BIC segmentation by computing the likelihood of the music and non-music models and by computing the mean average of the number of sub-segments and their durations on each segment. Table 3.7 shows an error rate of 9.6% and a F-measure of 25.79%.

Table 3.7: Results of the 1st run for IRIT music detection at ESTER-2 competition.

	Error Rate (%)	Recall (%)	Precision (%)	F-measure (%)
africa	10.82	03.82	88.05	07.32
France Inter	11.30	16.78	96.66	28.59
RFI	07.17	09.24	99.37	16.90
TVME	05.20	32.75	92.71	48.40
mean average	**09.60**	**14.91**	**95.41**	**25.79**

The second one uses the same features as than the first one. The only difference is the iterative scheme. The new error rate becomes 6.42% and the F-measure is absolutely improved of 44.01% as seen in table 3.8.

The last submitted system uses additional features proposed in [Lac09]. The mean average error rate becomes 5.51% (best system: 5.25 %) with a F-measure equal to 69.8% (best system: 78.85%) as seen in table 3.9.

Table 3.8: Results of the 2nd run for IRIT music detection at ESTER-2 competition.

	Error Rate (%)	Recall (%)	Precision (%)	F-measure (%)
africa	07.44	36.09	93.33	52.05
France Inter	06.50	52.39	98.79	68.47
RFI	06.40	19.46	97.36	32.44
TVME	04.80	40.76	88.65	55.84
mean average	**06.42**	**44.16**	**96.69**	**60.63**

Table 3.9: Results of the 2nd run for IRIT music detection at ESTER-2 competition.

	Error Rate (%)	Recall (%)	Precision (%)	F-measure (%)
africa	06.63	43.60	93.88	59.54
inter	05.17	70.10	89.27	78.53
RFI	05.93	25.66	97.14	40.60
TVME	04.63	43.03	89.18	58.05
mean average	**05.51**	**56.87**	**90.34**	**69.80**

3.3 Evaluation of the speaker clustering

The speaker clustering is evaluated by computing the errors where speaker turns of the automatic system do not match the expected speaker in the ground truth. In other words, the evaluation is made by computing the speaker time attributed to the wrong speaker (called speaker error time).

$$SpkrErr = \frac{\sum_{AllSegs}(dur(seg) * (min(N_{Ref}(seg), N_{Sys}(seg)) - N_{Correct}(seg)))}{\sum_{AllSegs}(dur(seg).N_{Ref}(seg))} \tag{3.10}$$

where the speech data file is divided into contiguous segments at all speaker change points and where, for each segment *seg*:

dur(*seg*)=the duration of *seg*,

$N_{Ref}(seg)$= the number of reference speakers speaking in *seg*,

$N_{Seg}(seg)$= the number of system speakers speaking in *seg*,

$N_{Correct}(seg)$= the number of reference speakers speaking in *seg* for whom their matching (mapped) system speaker are also speaking in *seg*. The results of speaker clustering will be detailed in next section.

3.4 Evaluation of the speaker diarization system

The diarization error rate (DER) is the sum of three error rates: the speaker error rate (cf. section 3.3), the missed detection rate and the false alarm rate.

$$DER = SpkrErr + Miss + False \tag{3.11}$$

where the missed detection rate is given by:

$$Miss = \frac{\sum_{AllSegs}(dur(seg) * (N_{Ref}(seg) - N_{Sys}(seg)))}{\sum_{AllSegs}(dur(seg).N_{Ref}(seg))} \tag{3.12}$$

and the false alarm rate is given by:

$$False = \frac{\sum_{AllSegs}(dur(seg) * (N_{Sys}(seg) - N_{Ref}(seg)))}{\sum_{AllSegs}(dur(seg).N_{Ref}(seg))} \tag{3.13}$$

3.4.1 Experiments and results

In this section, we describe the evolution of our system by detailing the results after every step. First, the baseline system uses the GLR-BIC segmentation with a first speech/non-speech detection and then the local-global clustering. Table 3.10 shows a missed detection rate of 1.4%, a false alarm rate of 0.7% and a speaker error rate of 15.3%. Thus, the overall DER is 17.42%.

Table 3.10: Results of the baseline speaker diarization system at ESTER-2 competition.

	Miss	False	SpkrErr	DER
africa	01.40	00.70	12.30	14.47
France Inter	01.80	00.60	19.80	22.24
RFI	00.00	00.60	05.30	05.88
TVME	01.30	01.40	16.90	19.60
mean average	**01.40**	**00.70**	**15.30**	**17.42**

Second, the iterative process provides an absolute SpkrErr improvement of 0.9% and the overall DER becomes 16.40% (cf. Fig. 3.11).

Table 3.11: Baseline + iterative process) results at ESTER-2 competition.

	Miss	False	SpkrErr	DER
Africa	01.50	00.60	12.40	14.48
France Inter	02.00	00.50	17.40	19.90
RFI	00.10	00.40	05.30	05.78
TVME	01.30	01.10	17.80	20.19
mean average	**01.50**	**00.60**	**14.40**	**16.40**

Third, the CLR clustering with an adaptive threshold is applied. This provides an additional improvement of 2.40% as seen in table 3.12.

Table 3.12: Baseline + iterative process + adaptative CLR results at ESTER-2 competition.

	Miss	False	SpkrErr	DER
Africa	01.40	00.60	12.80	14.84
France Inter	01.90	00.50	12.00	14.40
RFI	00.10	00.40	05.00	05.51
TVME	01.30	01.30	18.50	21.10
mean average	**01.40**	**00.60**	**12.00**	**14.01**

Instead of doing an adaptative threshold, we use fixed thresholds on both CLR and BIC similarity matrices in a last system. Table 3.13 shows an overall DER of 11.01% (best system: 10.80%). Recently, a better implementation of our system is done and the actual score is 9.85%.

Finally, Table 3.14 shows the performance of our improved system on the EPAC meeting corpus compared to the baseline standard system (a gain of about 8%). Table 3.15 shows the scores when excluding zones when two or more speakers are talking at the same time.

Table 3.13: IRIT-4 speaker diarization system results at ESTER-2 competition.

	Miss	False	SpkrErr	DER
Africa	2.10	00.60	05.60	08.32
France Inter	01.90	00.50	10.80	13.22
RFI	00.10	00.40	02.50	03.02
TVME	01.30	01.10	14.50	17.00
mean average	**01.50**	**00.60**	**08.90**	**11.01**

Table 3.14: IRIT speaker diarization system results vs Standard diarization system for EPAC-ESTER.

	Miss	False	SpkrErr	DER
Standard System	9.7	0.6	14.5	24.77
Improved System	**8.9**	**0.1**	**7.6**	**16.72**

Table 3.15: IRIT speaker diarization system results vs Standard diarization system at EPAC-ESTER exluding multi-speaker turns.

	Miss	False	SpkrErr	DER
Standard System	3.9	0.7	15.0	19.55
Improved System	**3.1**	**0.2**	**8.4**	**11.66**

Conclusion

After reviewing the state-of-the-art of speaker diarization systems and methods existing for speaker segmentation and speaker clustering, we proposed an original method for, not only speaker segmentation, but also for audio event segmentation. This method is based on GLR and BIC criteria. It was applied as a pre-processing step for speech/non-speech detection and music/non-music detection as well as for speaker segmentation in the speech part. Improvements of the baseline GLR-BIC segmentation were proposed to deal with the problem of short segments and multi-speaker boundaries detection by proposing coupled bi-directional segmentation and penalty coefficient decreasing techniques.

Then, we proposed improvement for EVSM speaker clustering methods by using additional pitch constraint. Because this clustering uses GMMs that need relatively good amount of data to be well modeled, we drop out this method and we use the BIC clustering to deal with meeting data. The state-of-the-art BIC clustering was improved by proposing a local-and-global clustering and by applying an update of clusters after computing the segments-to-clusters similarity matrix. In addition to the BIC clustering, we used the CLR post-clustering that deal with the background variation. Instead of applying the ordinary fixed threshold that highly depends on the type of the data, we propose both the adaptative thresholding technique that takes into account the file size, and the fixed thresholding method with additional BIC distance constraints.

Part II

Visual people indexing

Introduction

Most works in video indexing are sequentially analyzing a video document to produce a video index by attaching content-based labels to that document. Some of these processes aim to extract from the video data the temporal location of some more or less semantic features (e.g. car, bridge, dog, person, or colors, shapes, velocity).

Indexing video data is essential for providing content based access. It has typically been viewed either from a manual annotation perspective or from an image sequence processing perspective. The indexing effort is directly proportional to the granularity of video access. As applications need finer grain access to video, automation of the indexing process becomes essential.

More particularly, *visual people indexing* aims to annotate video documents according to people occurring in those documents using only visual (image) information. It is one of the important techniques for accessing video data effectively and for perceiving the user interacting in a Human Computer Interaction (HCI) context. It becomes important since it enables many applications of such "intelligent fast-forwards" where the video document is browsed for example by the shots containing a particular actor from the hundreds of short video sequences available for that document.

Practically, unsupervised visual people indexing must pass through different steps: people detection, people tracking and people clustering. Fig. 1 illustrates the general architecture of a visual people indexing system:

Hypotheses

In order to make the people indexing techniques as workable and portable as possible, many hypotheses were taken.

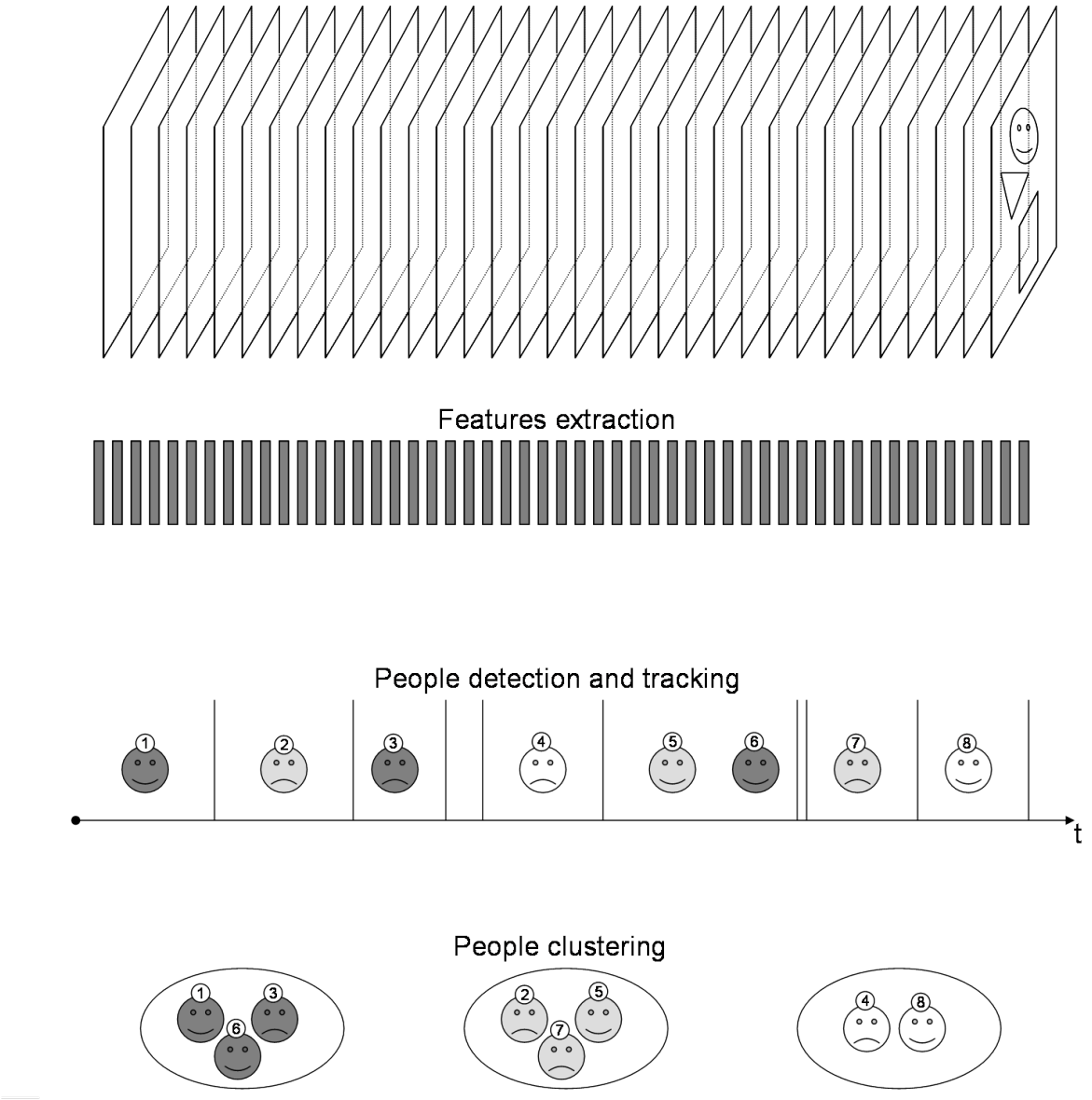


Figure 1: General steps for unsupervised visual people indexing.

- The video document may contain frames in black and white.
- A frame may contain more than one person.
- The number of people appearing in the video document is unknown.
- There is no *a priori* knowledge about the identity of the people appearing in the video.
- a person is considered as appearing in a shot if at least his frontal or profile face appears theoretically for more than 200 milliseconds (corresponding to 5 consecutive frames if

the frequency is 25 images/second) in the video document. In addition the minimum dimensions box in which the face is detected is supposed equal to 20x20.

- a person may change clothing during the video document. Although this problem seems very rare to occur in some video data as broadcast news and talk show programs, it is very common in TV series and movies.

- two different people may wear the same costume. This case is very recurrent in team sports like soccer and basketball.

- lightning conditions may change along the video.

Applications

Video people indexing can be present in many kinds of products used by both professionals and amateur consumers. Those applications are listed in the following items.

- **Automated authoring of Web content.** Media organizations and TV broadcasting companies have shown considerable interest in representing their information on the Web because the number of people who obtained their news on the Internet is growing at an astonishing rate. One of the main issues people are searching for is the presence of celebrities. It will be important to be able to access directly to the shots where those celebrities appear.

- **Searching and browsing large video TV archives.** Another professional application of people-based video indexing is in organizing and indexing large volumes of video data to facilitate efficient and effective use of these resources for internal use. Major news agencies and TV broadcasters own large archives of video that have been accumulated over many years. Traditionally, the indexing information used to organize these large archives has been limited to titles, dates, and human-generated synopses. Intelligent video segmentation and sampling techniques can reduce the visual contents of the video program to a small number of static images. Higher-level analysis is then used to extract information relevant to the presence of humans in the video such as anchors, politicians, etc. in TV broadcast news.

- **Searching and browsing movies.** This kind of application aims to provide an automatic cast listing in movies as well as the utterances where the corresponding persons appear.

- **Searching and browsing large video internet database.** It is a more ambitious application of the people indexing task. It can be used by both professionals and amateurs. Here, the major differences from the above applications are the relative bad resolution of the video data and the real time constraints.

- **Automatic visual surveillance** in dynamic scenes (both in indoor and outdoor environment) by monitoring human activities. It has two major components: detecting people and tracking them in sequence of video images. The goal of such a system understanding high-level events and complex actions such as detection of walking, running, dancing. For example, we invite you to review the work done in our team on human shape analysis [FJ06].

This part is organized as follows: in chapter 4, we review the existing visual features, and the state-of-the-art works for people detection, tracking, clustering and recognition. In chapter 5, we describe our proposed people indexing system starting from the shot boundaries detection, passing through the people detection and tracking and finally describing the people clustering algorithm using both face and clothing information. Experiments and results are detailed in chapter 6.

Chapter 4

State-of-the-art

Contents

4.1 Low-level visual features 67

4.2 People detection 72

4.2.1 Face detection 72

4.2.2 Upper-body detection 74

4.3 People tracking 75

4.3.1 Existing methods for people tracking 75

4.3.2 Face tracking 76

4.3.3 Clothing tracking 77

4.4 People clustering 77

4.4.1 Drawback of people clustering methods 77

4.4.2 The use of hair descriptors 79

4.4.3 The use of SIFT features 80

In this chapter, the main existing techniques related to the task of visual people indexing are reviewed. Initially, the visual features that have been found useful for people indexing are described. Then, a brief look on people detection, tracking and clustering techniques is done.

4.1 Low-level visual features

The only basic information an image is carrying are the R, G and B values of each pixel within this image. Other color spaces like YCbCr, YUV and HSV were defined either for encoding

purposes or for highlighting special characteristics of the pixel, but in all cases, we can convert from one space to another. Using that information as well as the position of each pixel, many low-level features were historically extracted in order to characterize that image. In video scenarios, the time can be added to that information to help extracting features related to movement and action in sequences.

The "low-level features" contain usually color, shape, texture and motion information. The most interesting ones are listed below, from the very simple to the very complicated.

- **Average color.** This is the simplest feature that can be extracted from an image. It is a triplet that is equal to the mean averages of the R, G and B values of all the pixels within the image or sub-image/blob (e.g. an image can be divided into 4 parts).

- **Average luminance.** The average luminance value is often computed on basic units used in video encoding (e.g. blocks of 8x8 pixels) of the image. It is equal to the mean average of the luminance values of all the pixels within the blob. The luminance of a pixel is given by:

$$L_p = 0.2126R_p + 0.7152G_p + 0.0722B_p \tag{4.1}$$

  and the average luminance:

$$L_{average} = \sum_{p=1}^{N} \frac{L_p}{N} \tag{4.2}$$

  where N is the number of pixels within the block.

- **Dominant color.** This feature is used in MPEG-7 standard. A set of dominant colors in an image provides a compact description that is easy to index. The feature descriptor consists of representative colors, their percentages in the image, their spatial coherency, and their color variances. In [LJH03], authors proposed an efficient and fast method based on JPEG standard and some statistical parameters of DCT coefficients to extract dominant color feature from the compressed bit stream. In [HJC06], authors used an easier way to compute the first dominant color of an image. This method will be improved in our work and used for extracting dominant color of the clothing (cf. section 5.6.2).

- **Normalized r and b colors for skin.** According to skin color theory, under certain lighting conditions, a skin color distribution can be modeled by a 2D-Gaussian model [VSA03]. In order to reduce the lighting effects of the human skin, the original RGB color images

are converted to the chromatic color images. If we suppose that $X(R,G,B)$ and $X^{'}(r,g,b)$ are pixels in the original color image and the chromatic color image respectively, r, b, and g are expressed by:

$$\begin{cases} r = \frac{R}{R+B+G} \\ b = \frac{B}{R+B+G} \\ g = \frac{G}{R+B+G} \end{cases}$$

As $r + b + g = 1$, the g component is omitted. The 2D-Gaussian distribution is then expressed by:

$$p(x) = \frac{1}{2\pi \left|\Sigma\right|^{\frac{1}{2}}} \exp[-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)] \tag{4.3}$$

where μ is the mean vector and Σ is the covariance matrix for r and b. Those values are generally estimated on a large training dataset of hand-segmented images. The resulting skin color model was used to localize and detect possible faces in images [HVB06] and to detect naked images [LKCC07].

- **Gray-scale and Color histograms.** They represent the distribution of gray-scale levels or colors in an image. They are obtained by counting the number of pixels of each intensity value for each channel if the histogram is mono-dimensional. Some works use a 2D or 3D histograms to take into account the correlation between color channels.

- **Color moments.** Those features were introduced in [SO95]. For each color channel i (i may be R, G or B), the first three color moments are computed as follow:

$$E_i = \frac{1}{N} \sum_{j=1}^{N} p_{ij} \tag{4.4}$$

$$\sigma_i = (\frac{1}{N} \sum_{j=1}^{N} (p_{ij} - E_i)^2)^{\frac{1}{2}} \tag{4.5}$$

$$s_i = (\frac{1}{N} \sum_{j=1}^{N} (p_{ij} - E_i)^3)^{\frac{1}{3}} \tag{4.6}$$

where p_{ij} is the *j-th* pixel of the *i-th* channel and N is the total number of pixels of that image.

- **Texture features.** Texture is a measure of the intensity variation of a surface which quantifies the appearance characteristics of an object such as smoothness, roughness, waviness, lay, flaws, etc. There are various texture descriptors such as "gray-level co-occurrence matrices" [HDS73], Law's texture measures [Law80] and Gabor wavelets [MM96].

  In [MM96], authors define the Gabor wavelet transform of a given image I(x,y) by:

$$W_{mn}(x,y) = \int I(x_1,y_1) g^*_{mn}(x-x_1, y-y_1) dx_1 dy_1 \tag{4.7}$$

  where

$$g_{mn}(x,y) = a^{-m} g(x',y') \tag{4.8}$$

  with $a > 1$, $m, n = integer$ and

$$x' = a^{-m}(x\cos\theta + y\sin\theta) \tag{4.9}$$

$$y' = a^{-m}(-x\sin\theta + y\cos\theta) \tag{4.10}$$

  where $\theta = n\pi/k$ and g^*_{mn} indicates the complex conjugate of g_{mn}.

  The mean μ_{mn} and standard deviation σ_{mn} of the magnitude of the transform coefficients are used to represent the region for classification and retrieval purposes:

$$\mu_{mn} = \int\int |W_{mn}(xy)|\, dxdy \tag{4.11}$$

$$\sigma_{mn} = \sqrt{\int\int (|W_{mn}(x,y)| - \mu_{mn})^2 dxdy} \tag{4.12}$$

  A feature vector is then reconstructed using those μ_{mn} and σ_{mn}.

$$F = [\mu_{00}, \sigma_{00}, \mu_{01}, \sigma_{02}, ..., \mu_{NM}, \sigma_{NM}] \tag{4.13}$$

  where N and M are respectively equal to 3 and 5 in [MM96].

- **Interest points.** Interest points are used in the problems of image segmentation, object detection, recognition and tracking. They have an expressive texture in their respective localities. A good interest point is a point that is invariant to lighting and camera viewpoint. Many detectors were proposed to detect the interest points such as:

- Moravec's point of interest operator [Mor79] that is based on intensity variation computation,
- Harris point of interest detector [HS88] that computes the first order image derivatives to highlight the directional intensity variation and the second moment matrix that encodes this variation,
- KLT detector [TK91] that uses, in addition to the second moment variation, the eigenvalues of this matrix to compute an interest point confidence,
- SIFT feature descriptor introduced by Lowe [Low04] and that will be detailed later in section 4.4.3.

- **Video features.** In addition to the image features described above many video features were also used in the literature. In [HSH07], authors defined four low-level video features that are used to classify films into three broad categories: action, dramas, and thriller films.

  - *Average shot length feature.* It represents the tempo of a scene. In a film, the director can control the speed at which the attention of the audience is directed by varying the tempo of the scene. A shot is defined as a sequence of frames taken by a single camera continuously along the time. It generally does not show any major change in the color content.
  - *Color variance.* It can be intuitively seen that comedy films tend to have a large variety of bright colors, whereas horror films generally adopt only darker hues. Thus, color variance may be a good discriminator for some kind of applications as the films classifications. As a consequence, it affects negatively algorithms for people indexing.
  - *Lighting key.* Generation of film makers has exploited luminance to induce emotions. Therefore, a correlation exists between the lighting and the gender of a film.
  - *Motion content.* This feature represents the amount of activity in a film. For example, action films would have higher values for such a measure contrarily to dramatic or romantic movies for example. To find the disturbance in the scene, a structure tensor was used in [HSH07] to provide an orientation angle θ. This angle will be constant for all pixels if there is no motion in the shot. If there is a global motion as camera

translation, the values of θ will be equal or similar. However, in the case of local motion, pixels will have different orientations, thus, different values of θ.

Wang et al. [WDV+03] listed additional video features, used especially in the compressed-domain (MPEG technologies). They group them into 3 main categories: spatial visual features, motion features and video coding features.

The above listed features are mostly used in many of video indexing methods: detection, tracking and recognition. Additional visual features can be reviewed in Appendix B.

4.2 People detection

People detection in images is a challenging problem that was widely explored in the literature. It consists in identifying and locating humans in an image regardless of their position, scale and illumination. Many methods that aim to detect people were proposed in the literature, they are often based on full-body detection, partial body detection (upper-body and lower-body) or face detection. The full-body detection is generally used in Video Surveillance Systems ([IF01], [VJS03], [ARS08]) for detecting and tracking people in indoor or outdoor scenes where real-time constraints are mandatory. As we are interested in TV videos, the following subsections will focus on detecting face and upper-body parts as they are more occuring in that kind of data.

4.2.1 Face detection

Face detection is often used as a salient cue to detect people in images or videos. It is also strongly used to search for other body parts. Moreover, it is the first step of any fully automatic system that analyzes the information contained in faces (e.g. identity, gender, expression, age, race and pose). Many conditions make the face detection task difficult like the pose (frontal, 45 degree, profile and upside down), presence or absence of structural components (beards, moustaches and glasses), facial expression, occlusions, orientation (in-plane rotation) and imaging conditions (lighting, camera characteristics and resolution).

From a practical point of view, the face detection paradigm is: given an arbitrary image, the goal is to determine whether or not there are any faces in the image, and in a positive case, return the location and size of each face.

Historically, many approaches were proposed to detect faces in images and sequence of images. They can be classified into four categories: knowledge-based detection [YH94], Feature-based detection [LBP95], Template-based [LTEA95] and Appearance-based detection ([RBK98], [VJ04]). Most of them carry out the task by extracting certain properties (e.g. local features) of a set of training images acquired at a fixed pose (e.g., upright frontal pose) in an offline setting. To reduce the effects of illumination change, these images are processed with histogram equalization [RBK98] or standardization (i.e., zero mean unit variance) [VJ04]. Based on the extracted properties, the face detection system scan through the entire tested image at every possible location and scale in order to locate faces.

The most recent and interesting approaches for detecting faces are the appearance-based approaches. They generally use Neural Network [RBK98], Principal Component Analysis (PCA), Factor Analysis, Support Vector Machine (SVM), Mixture of PCA, Mixture of factor analyzers, Distribution-based method, Naïve Bayes classifier, Hidden Markov model (HMM), Sparse network of winnows (SNoW), Kullback relative information and AdaBoost technique [VJ04]. For more details about those techniques, please refer to [hYKMA02], [hY09] and [HL01b].

In our work, we choose to use the AdaBoost method to detect frontal faces thanks to the OpenCV toolbox[17]. This method contains three major phases: a rectangular feature extraction, a classifier training using boosting techniques and a multi-scale detection algorithm [VJ04].

- **Feature extraction.** Those features are reminiscent from the Haar basis functions which have been used in [POP98a] for object detection. But authors used three kinds of features as seen in Fig.4.1: 1) the value of *two-rectangle feature* that is the difference between the sum of the pixels within two rectangular regions, 2) a *three-rectangle feature* that computes the sum within two outside rectangles subtracted from the sum in a center rectangle, 3) a *four-rectangle feature* computes the difference between diagonal pairs of rectangles.

- **Learning classifiers**. Given a feature set and a training set of positive and negative images, authors used a variant of AdaBoost technique [FS97] in order to select the features and to train the classifier by boosting the classification performance of a simple learning algorithm: the AdaBoost algorithm combines a collection of weak classification functions to form a stronger classifier.

[17] OpenCV is a free computer vision library originally developed by Intel. It focuses mainly on real-time image processing. Webpage: http://opencvlibrary.sourceforge.net/

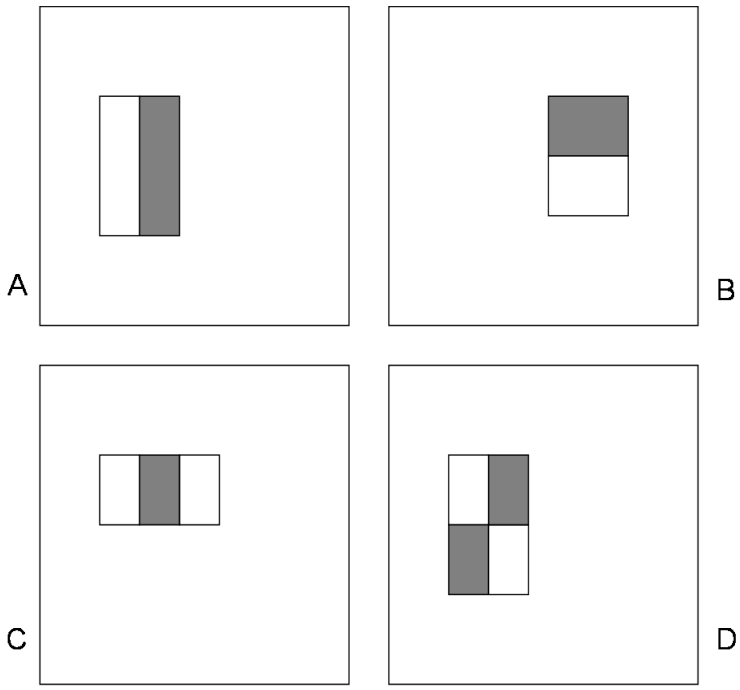


Figure 4.1: Examples of rectangle features: (A) and (B) show *two-rectangle feature*, (C) shows a *three-rectangle feature*, (D) shows a *four-rectangle feature* [VJ01].

- **Multi-scale detection algorithm**. Viola and Jones proposed an algorithm for constructing a cascade of classifiers which achieves good performance while radically reducing computation time. The algorithm detailed in [VJ04], aims to reject many of the negative sub-windows using very simple classifiers while detecting almost all positive instances, and then using more complex classifiers to achieve low false positive rates.

4.2.2 Upper-body detection

Although successful frontal face detectors are available, faces may not be clearly visible in some TV shows and movies. To cope with this situation, we can use an upper-body detector.

As for face detection, the AdaBoost algorithm explained above can also be used to detect upper-body [MOB06]. Another technique was proposed by Ferrari et al. in [FMJZ08]: it uses the Histograms of Oriented Gradients (HOG) [DT05] to detect the upper-body. This detector is designed to detect the region between the top of the head and the upper half of the torso. This near-frontal detector works well for viewpoints up to 30 degrees away from straight frontal, and also detects back views.

4.3 People tracking

People tracking is a key task in many promising computer vision applications, such as smart video surveillance (prosecution, intelligence gathering, crime prevention, traffic statistics), people indexing in video (movies, news), motion-based human recognition (person identification based on gait), synthesis (games, movies), driving assistance systems, or biomechanics (spot diseases).

However, basic difficulties should be expected. The people tracking system should deal with: non-rigid or articulated nature of targets, partial targets occlusion, scene illumination changes, small targets size, noise in images, and real-time constraints especially for visual surveillance systems.

The following section presents a brief survey of non-rigid object (such as person) tracking methods.

4.3.1 Existing methods for people tracking

Numerous approaches for non-rigid object (such as human) tracking have been proposed in the literature. They generally differ on the way the object is represented and the image features are selected, and on the algorithm the tracking is processed.

Object representation.

The object shape representations commonly employed for tracking are:

- *a single point* that is generally the centroid of the object or a set of points of interest within the object,

- *primitive geometric shapes* such as rectangle [FJ06], ellipse [CRM03],etc,

- *object silhouette and contour* where the silhouette is defined as the region inside the contour [YS04],

- *articulated shape models* where objects are composed of body parts that are held together with joints,

- *skeletal models* where skeleton is obtained by applying medial axis transform to the object silhouette.

Feature selection for tracking.

Feature selection is a crucial step in tracking. The features that were investigated for people tracking are edges, optical flow, texture and color. The last one is the most widely used feature for tracking. Mostly those features are chosen depending on the application domain even though some methods were proposed to handle the automatic feature selection like the filter methods that use a general criterion (for example: the features should be uncorrelated) and the wrapper methods that select the features based on their usefulness.

The tracking.

Object tracking aims to generate the trajectory of an object over time by locating its position in every frame of the video. The tasks of detecting the object and tracking it can either be performed separately or jointly [ARS08]. Historically, the object tracking methods can be divided into three categories depending on the chosen object representation:

1. *Point tracking* methods that are either deterministic such as the GOA tracker [VRB01] or statistical such as the Kalman filter [BC86] and the particle filter [Kit87].

2. *Kernel tracking* methods where the shape and the appearance of the object are taken into consideration in order to track it. These methods use either the *Template and density based appearance models* like Mean-shift [CRM03] or the *Multi-view appearance models* like the SVM tracker [Avi01].

3. *Silhouette tracking* methods that provide an accurate shape description for objects. These methods are based either on contour evolution like the variational methods [BSR00] or matching shapes like the hough transform [SA04].

After reviewing methods for people tracking, we will focus on tracking faces and clothes because unlike video surveillance, scenes in movies, TV talk-shows, TV games and TV News often contain people whose upper-bodies are the only visible parts.

4.3.2 Face tracking

Tracking, which is a crucial part of most face processing systems, is essentially motion estimation. However, general motion estimation has fundamental limitations such as the aperture problem. In face recognition systems, it is necessary to track each face over the video sequence in order to

extract the appropriate information. Tracking is also necessary for 3D model-based recognition systems where the 3D model is estimated from the input video [RcG05], but the tracking is computationally intensive in this case. Many applications may be derived from face tracking such as video surveillance, biometrics, face modeling and video communications and multimedia systems (MPEG technology).

Historically, many methods were proposed for face tracking. They can be divided into three categories: (1) head tracking where the entire face is tracked as a single rigid entity (such in [ASHP93]); (2) facial features tracking (such in [TW93]) where features like eyes, ears, mouth, nostrils, eyebrow, lips, and nose are limited by the anatomy of the head that is supposed here as a non-rigid object influenced with motion due to speech or facial expressions; (3) complete tracking which involves tracking both the head and the facial features (such in [SKK08]). Besides, many of those methods are able to handle challenging situations such as facial deformations, changes of lighting, partial occlusions, pose variation and facial resolution.

4.3.3 Clothing tracking

The clothing is also used to help tracking people in videos. Even though researchers do not give it a special attention in many of their publications, it remains one of the most important cues for people tracking since a good amount of color information related to it are used by the system trackers like in [LAMA05].

In [HE09], authors used cloth tracking in order to re-texture it for real-time virtual clothing applications. A more sophisticated work for tracking clothed people can be found in [RKP+07] where authors used it for motion capture.

4.4 People clustering

4.4.1 Drawback of people clustering methods

The issue of visual persons clustering is relatively new. Some researchers generally view it as a recognition problem such in [AZ05] and [BLGT06] or a classification problem such in [ESZ06] and [PL08]. In both cases, a set of face exemplars is generally used. This task can be used for indexing purpose of the video data ([CMM03]) as well as for organizing consumers album photos [CLY09] since the basic technique remains the same for sequence images or still images. In [FZ02], authors introduced a distance metric for clustering and classification which is invariant

to affine transformation including priors. They apply it for face clustering in order to produce an automatic cast listing in movies.

Arandjelovic and Zisserman develop in [AZ05] a recognition method based on a cascade of processing steps that normalize the effects of the changing environment: they first suppressed the background surrounding the face, enabling the maximum area of the face to be retained. Then, they added a pose refinement step to optimize the registration (using facial features like eyes and mouth detected using SVM) between the test image and face exemplar. They used a distance to a subspace to allow for partial occlusion and expression change. This method was tested and evaluated on two episodes of "Buffy the Vampire Slayer".

In [ESZ06], both visual and textual information are combined. Visual information relies on face and clothing. Each unlabelled face track is represented as a set of face and clothing descriptors f, c . Exemplar sets λ_i have the same representation but are associated with a particular name obtained by aligning subtitles and transcripts. For a given track F, the quasi-likelihood that the face corresponds to a particular name λ_i is defined by:

$$p(F/\lambda_i) = \frac{1}{Z} e^{\left\{-\frac{d_f(F,\lambda_i)^2}{2\sigma_f^2}\right\}} e^{\left\{-\frac{d_c(F,\lambda_i)^2}{2\sigma_c^2}\right\}} \tag{4.14}$$

where

$$d_f(F,\lambda_i) = \min_{f_j \in F} \min_{f_k \in \lambda_i} \|f_j - f_k\| \tag{4.15}$$

and the clothing distance $d_c(F,\lambda_i)$ is similarly defined.

In [CMM03], after detecting faces using an iterative algorithm that gives a confidence measure for the presence or absence of faces within video shots, authors process the clustering of those faces using a PCA-based dissimilarity measure in conjunction with spatio-temporal correlation. Experiments were done on a broadcast news test corpus.

In [HWS08], a new method for multi-view face clustering in video sequence is proposed: first, a "pose clustering" is done followed by a clustering for different individuals within each "pose group". The eyes are detected using Gabor filters. Their location is then used to perform the "pose clustering". Finally, images of the similar pose are clustered using Principal Component Analysis and Local Binary Pattern and kmeans algorithms. Experiments were done only on a database containing only 8 persons. Each person has a short sequence that includes 7 poses: $\pm 60°, \pm 45°, \pm 30°, 0°$.

Chu and al. proposed in [CLY09] a face clustering method in consumer photos by matching images using local features. They represented matching situations using visual sentences. Then, visual language models are constructed to describe the dependency of image patches on faces.

4.4.2 The use of hair descriptors

The hair was studied and analyzed in [YD06]. In this paper, authors proposed an automatic hair detection algorithm that can be summarized in four steps:

- **Face detection** where they employ the algorithm of Viola-Jones [VJ04],
- **Eye detection** using the same cascade of boosted classifiers in order to train eye detectors to localize eyes within the face region. Face and eye allow normalizing face sizes so hair representations can be compared.
- **Skin color modeling** based on the automatic selection of three regions: two are below the eyes and one at the forehead.
- **Head hair color modeling** by assuming that the hair is present at one or more of three principal locations adjacent to facial skin: the right, middle and left sides of the upper face.

After hair detection, authors define a list of hair characteristics that are used to recognize and index people:

- **the hair color** by assuming that it is lambertian,
- **the hair-split location** that appears at either a darker shade of the hair region or as revealed skin within the hair region,
- **the hair volume** that might be very important to differentiate people,
- **the hair length** which is defined as the largest vertical distance between the hair boundary point and the vertical coordinate of the ear,
- **the surface area covered by hair** which is computed for the top of the head i.e. the region above the eye-level,
- **the hair symmetry** which is defined as the ratio of the volumes of hair in the left and right sides,

- **the inner and outer hairlines** that also can be helpful,
- **the hair texture** where authors used the Gabor wavelets to compute the distance between two subjects.

Authors show some successful results when taking each characteristic alone and then when doing the fusion between all characteristics.

We should notice that this work cannot be applied in our case because:

- the face resolutions are quite different 1600x1200 and 768x576 in the paper and from 120x120 to 20x20 in our case,
- the subject may have no frontal appearance in our case unlike in the paper where the faces are all in frontal symmetrical view,
- the authors do not give special attention on variations in hair appearances, lightning conditions, etc.

Due to all these reasons, the tasks of hair detection and hair description in our case are more difficult than the one implemented in [YD06].

4.4.3 The use of SIFT features

The Scale Invariant Feature Transform (SIFT) was introduced by Lowe in 2004 [Low04]. These features are invariant to image scale and rotation, and are shown to provide robust matching across a substantial range of affine distortion, change in illumination, addition of noise and change in 3D view-point. The major stages of extracting the SIFT features are:

1. **Scale-space extrema detection.** This first stage identifies potential points of interest (called keypoints in the SIFT framework) that are invariant to scale and orientation. It is implemented efficiently by using a difference-of-Gaussian (DoG) function applied on different scales of the image. Then, each sample point is compared to its eight neighbors in the current image, nine neighbors in the scale above and nine neighbors in the scale below. It is selected only if it is larger than all of these neighbors or smaller than all of them.
2. **Accurate keypoint localization.** At each location of a candidate keypoint, a detailed model is fit to determine location and scale. This model allows points to be rejected based

on measures of their stability i.e. when having low contrast, being sensitive to noise or being poorly localized along an edge.

3. **Orientation assignment.** To determine the keypoint orientation, a gradient orientation histogram is computed in the neighborhood of the keypoint. Peaks in the histogram correspond to dominant orientations. A separate keypoint is created for the direction corresponding to the histogram maximum. Then another keypoint is created for every other direction within 80% of the maximum value. Therefore, for location with multiple peaks of similar magnitude, there will be multiple keypoints created at the same location and scale but different orientation. All the properties of the keypoints are then measured relative to the keypoint orientation, this provides invariance to rotation.

4. **Keypoint descriptor.** Once a keypoint orientation has been selected, the feature descriptor is computed as a set of orientation histograms on 4x4 pixel neighborhoods. The orientation information come from the Gaussian image closest in scale to the scale of the keypoint. Each histogram contains 8 bins. This leads to a SIFT feature vector with 4x4x8=128 elements. This vector is normalized to enhance invariance to change in illumination.

Large numbers of features can be extracted from typical images with this algorithm. A typical image size 500x500 pixels will give rise to about 2000 stable features. Furthermore, those features are highly distinctive, which allows only few features to be correctly matched with high probability against a large database of features, providing a basis for object and scene recognition. That explains why those features were recently used for face recognition.

Fig.4.2 illustrates an image on which the SIFT features were extracted using the code implemented by Lowe [18]: for each keypoint, is associated a vector that contains the scale and the orientation information.

The SIFT features was successfully applied for object recognition. Many strategies were proposed in order to match the SIFT features computed in the test image with the SIFT features of the template image.

In [Low04], the best candidate match for each keypoint is found by identifying its nearest neighbor in the template image (or more generally in the database of keypoints from training

[18] http://www.cs.ubc.ca/ lowe/keypoints.

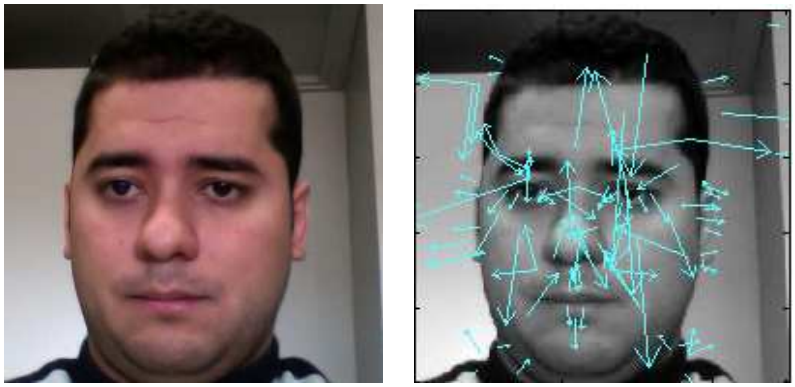


Figure 4.2: A face image and its corresponding sift features.

images). The nearest neighbor that is defined as the keypoint with the minimum Euclidean distance for the invariant descriptor vector is computed using the Best-Bin-First (BBF) algorithm that returns the closest neighbor with high probability: bins in feature space are searched in the order of the their closest distance from the query location.

Moreover, many features that generally correspond to the background clutter are discarded because they do not have any correct match in the training images. An efficient way to get rid of those features is by computing the ratio of distances to the closest neighbor and the second closest neighbor in the feature space. Lowe has chosen a threshold of 0.8: if the distance ratio is greater than this threshold, matches are rejected. Experimentally, this eliminates about 90% of the false matches and discards less than 5% of the correct matching.

In order to maximize the performance of object recognition, Lowe found that reliable recognition is possible with as few as 3 features. In order to reduce the outliers, he used the Hough transform that allow clustering features in pose space. The Hough transform identifies clusters of features with a consistent interpretation by using each feature to vote for all object poses that are consistent with that feature.

In [ANP07], authors introduced a method to create a dissimilarity matrix using the number of matching between each couple of faces (A_i, A_j). The dissimilarity distance is defined by:

$$DR(i,j) = DR(j,i) = 100(1 - \frac{M_{ij}}{min(K_i, K_j)}) \tag{4.16}$$

where M_{ij} is the maximum number of keypoint matches found between A_i and A_j, and K_i, K_j are the numbers of keypoints found in A_i and A_j respectively. Experiments were carried only on the feature length movie "Two weeks Notice". One thing that was not mentioned is that

the processing is done on all detected faces even within the same face track that corresponds to a sequence of consecutive images that contain the same face (generally a shot). That is time consuming due to the explosive amount of time needed to compute the SIFT features of every face.

Three more focused and simplest methods for recognizing faces were proposed in [BLGT06]:

- **Minimum pair distance.** It consists in computing the distance between all pairs of keypoint descriptors in the test image (I_{test}) and the template image (I_{temp}), and use as matching score the minimum distance.

$$MPD(I_{test}, I_{temp}) = \min_{i,j}(d(F(k_j^{I_{test}}), F(k_i^{I_{temp}})) \tag{4.17}$$

  where the sets of features for test and template images are respectively:

$$K(I_{test}) = \left\{k_1^{I_{test}}, k_2^{I_{test}}, ..., k_{M_1}^{I_{test}}\right\}$$

  and

$$K(I_{temp}) = \left\{k_1^{I_{temp}}, k_2^{I_{temp}}, ..., k_{M_2}^{I_{temp}}\right\}$$

- **Matching eyes and mouth.** The most discriminate part of face information is located around the eyes and the mouth [NPAA97]. Bicego et al. used this fact to consider only SIFT features belonging to this image areas. So in this case, the eyes and mouth regions must be found. Then they compute the average distance as follows:

$$D^{EM}(I_{test}, I_{temp}) = \frac{1}{2}MPD(I_{test}^{eyes}, I_{temp}^{eyes}) + \frac{1}{2}MPD(I_{test}^{mouth}, I_{temp}^{mouth}) \tag{4.18}$$

- **Matching on a regular grid.** This is the best among the three methods proposed in [BLGT06] since it takes into consideration the location of the features (by comparison with the first method), and also, features located on the right eye could not be matched any more with features located on the left one (by comparison to the second method). The matching between two images is performed by computing the average distance between all pairs of corresponding sub-images of dimensions 1/4 and 1/2 of width and height.

$$D^{RG}(I_{test}, I_{temp}) = \frac{1}{N}\sum_{n=1}^{N} MPD(I_{test}^{n}, I_{temp}^{n}) \tag{4.19}$$

Chapter 5

Proposed Face-and-clothing based people indexing

Contents

5.1 System Architecture 86

5.2 Shot Boundary Detection 86

5.3 Face based detection 87

5.4 Clothing extraction 88

5.5 People tracking 88

5.5.1 Face-based people tracking 89

5.5.2 Clothing-based people tracking 93

5.6 Proposed methods for people clustering 94

5.6.1 Face-based clustering 94

5.6.2 Clothing based clustering 99

5.6.3 Hierarchical bottom-up clustering 103

After reviewing the most interesting works for people indexing, we detail in this chapter our person indexing system using only visual information: we present our proposed methods for shot boundary detection, face detection, our clothing extraction, our forward/backward people tracking based on face tracking and clothing tracking. Finally, we describe our most interesting contribution that was done for people clustering using both face an clothing cues.

5.1 System Architecture

Fig.5.1 illustrates the general architecture of our people indexing system. After extracting low-level information from every frame in the video sequence, shot boundary detection is processed in order to split the sequence into homogeneous chunks. Then, a face-based people detection is done on each shot. Tracking of every person is processed within this shot thanks to the skin of the face and the color histogram of the clothing in order to improve the people detection results. Finally, a people clustering method is applied by combining distances on local features and skin color model of the face, 3D color histogram and texture of the cloth.

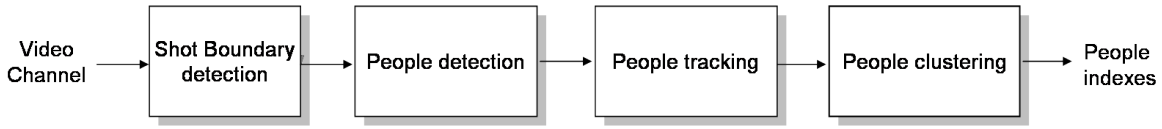


Figure 5.1: General architecture of the people indexing system.

In our team, this work was introduced by Jaffre [Jaf05] who developed a system that first detects people and then uses clothing information to track those people. Our contribution is a mature and complete solution of that problem by adding different components (as shot boundary detection, people tracking and hierarchical people clustering), different descriptors (as SIFT, skin color and clothing texture), different clustering criteria (simple linkage, average linkage and complete linkage). All those allow us to reduce false alarms and missed detections. One clear improvement is that our old system confuses between different people that wear the same clothing, our new system will not make this confusion because it uses the face descriptor that is related to the person identity.

5.2 Shot Boundary Detection

Historically, the first studied video segmentation task is the "shot boundaries detection" which aims to break the massive volume of video into smaller chunks. Shots are concatenated by editing effects such as hard cuts, fades, dissolves and wipes. A reliable shot detection algorithm should identify such short breaks.

Because it is an important preprocessing task for video analysis, quite a lot of approaches were proposed in the literature [TDV00], [LGZ+07]. See for example the report of TRECVid [SOD09] for a review and a comparison of the state-of-the-art systems.

In this work, we tried to apply our GLR-BIC method that was used for audio segmentation and that aims to split the audio stream into homogeneous zones by assuming that a shot is a homogeneous sequence that contains more or less identical images. In order to fit the model selection problem, a feature vector is extracted as follows: each image provided every $40ms$ (25 images/second) is divided into 4 equivalent parts. The mean values of R, G and B colors are computed in each part. Therefore, the feature vector of dimension d is composed of those values ($d = 3*4$). Then, the GLR-BIC algorithm is applied as explained in the section 2.1 of chapter 2.

In order to eliminate some false alarm detections (improvements of about 5 to 10%), a final step of histograms comparison is applied on the detected boundaries using the Manhattan (or City-Block) distance:

$$d_{Manhattan} = \sum_{i=0}^{255} |h_{1,i} - h_{2,i}| \tag{5.1}$$

where h_1 and h_2 are the histograms of two compared 8-bits images. Tab.5.1 shows the results of the proposed system compared to the average system and the best system at the ARGOS French competition using both ARGOS and TRECVid evaluation metrics.

Table 5.1: Comparison between the proposed system, the average system and the best system at the ARGOS competition using both ARGOS and TRECVid metrics.

	Proposed system	Average system	Best system
ARGOS F-measure	93.3%	87%	94%
TRECVid F-measure	91%	88.9%	89.9%

5.3 Face based detection

In this work, we used the face detector proposed by Viola and Jones [VJ04] due to its high accuracy and speed. This method is implemented in OpenCV toolbox. We use it to detect frontal faces in still images as seen in Fig.5.2. Then, a trivial improvement was done in [Jaf05] on sequence of images by taking into account that a face must be present in at least 5 consecutive frames in order to be visible. Accordingly, many false alarm detections are removed.



Figure 5.2: Multi-face frontal detection on a still image.

5.4 Clothing extraction

Once the face is detected and located, the second goal is to extract the most interesting clothing part in order to use it as a discriminate descriptor for recognition on next stages.

For frontal faces, the clothing of the upper-body since in TV broadcast (movies, debates, news, etc.) usually the face and the upper-body are generally the most appearing parts of the body. Thus, the clothing is extracted as seen in Fig.5.3: the width of the clothing is considered as equal to 2.3 times the width of the face, and its height equal to 2.6 times the height of the face.

5.5 People tracking

After localizing frontal and profiles faces and their corresponding clothing locations, a tracking step is needed to follow the face in order to detect people in regions where the face detector

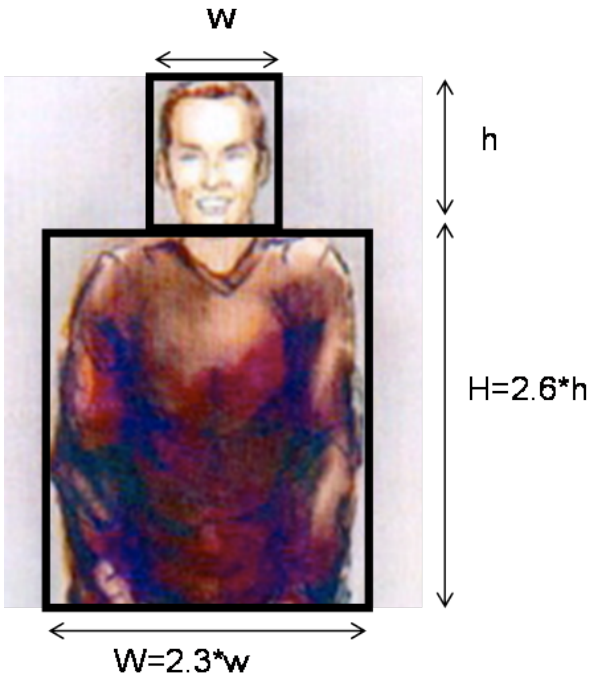


Figure 5.3: Extraction of clothing using frontal faces.

fails. We proposed two tracking systems: the first is based on face tracking and the second is based on clothing tracking.

5.5.1 Face-based people tracking

As the size of the face is relatively small, we consider the face as an single non-rigid entity with no need to track the face features (eyes, lips, etc.). Based on skin tracking, two tracking processes are done using face: a backward tracking and a forward tracking.

Fig.5.4 resumes the extraction and the modeling of the skin color of the face: since the face is detected and localized, the purpose is to extract the skin part of that face and to model it in order to help tracking people in cases where the face detectors fail. This can be summarized in 2 steps:

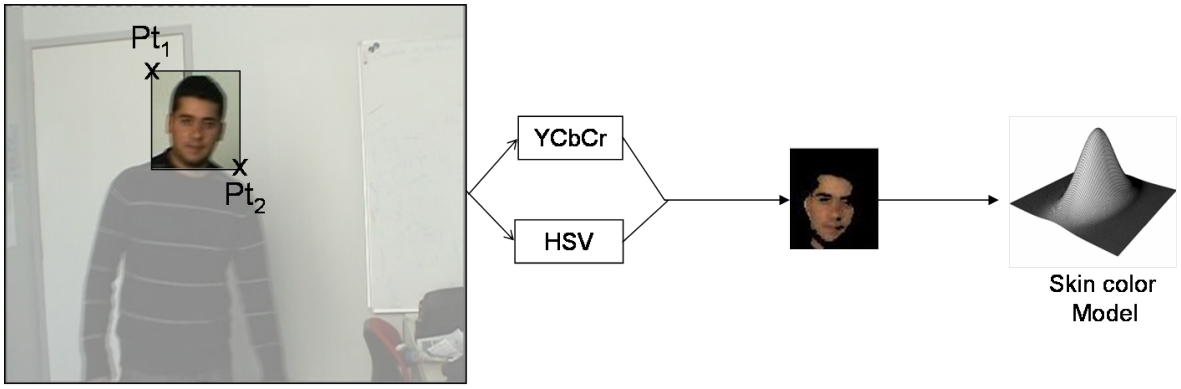


Figure 5.4: Extracting and modeling the color of the skin within the face.

1. **Skin extraction**. The RGB image is converted to YCrCb and HSV spaces. Then, thresholding is applied on the C_r, C_b components that are coded on 1 byte, and the hue H is normalized between 0 and 1, using the following expressions:

$$\left.\begin{array}{l} 135 \leq C_r \leq 170 \\ 130 \leq C_b \leq 200 \\ 0.01 \leq H \leq 0.1 \end{array}\right\} \tag{5.2}$$

   Those thresholds were chosen in order to allow detecting from the very light to the very dark skin. Fig.5.5 shows some examples of skin detection within the face region.

2. **Skin modeling.** Once the skin is extracted, the corresponding normalized r and b are computed as explained in section 4.1. Then, they are used to train a 2D Gaussian model. This model is used to help forward and backward tracking. It has been shown [VSA03] the rgb normalized space is better than RGB, YCrCb and HSV spaces since it handles the lighting variation.

The backward-forward tracking. For each detected face, two points are computed as illustrated in Fig.5.4. The first point Pt_1 denotes the top-left corner of the rectangle in which the face is detected and the second one Pt_2 denotes the bottom-right corner. Supposing that a shot contains n frames and that the face is only detected in a sequence of frames $F_s, ..., F_e$, the purpose is to see if that face is also present in $F_{s-1}, F_{s-2}, ..., F_1$ on the left side, and present in $F_{e+1}, F_{e+2}, ..., F_n$ on the right side as seen in Fig.5.6.

The proposed algorithm is an iterative process and can be divided into 4 steps:

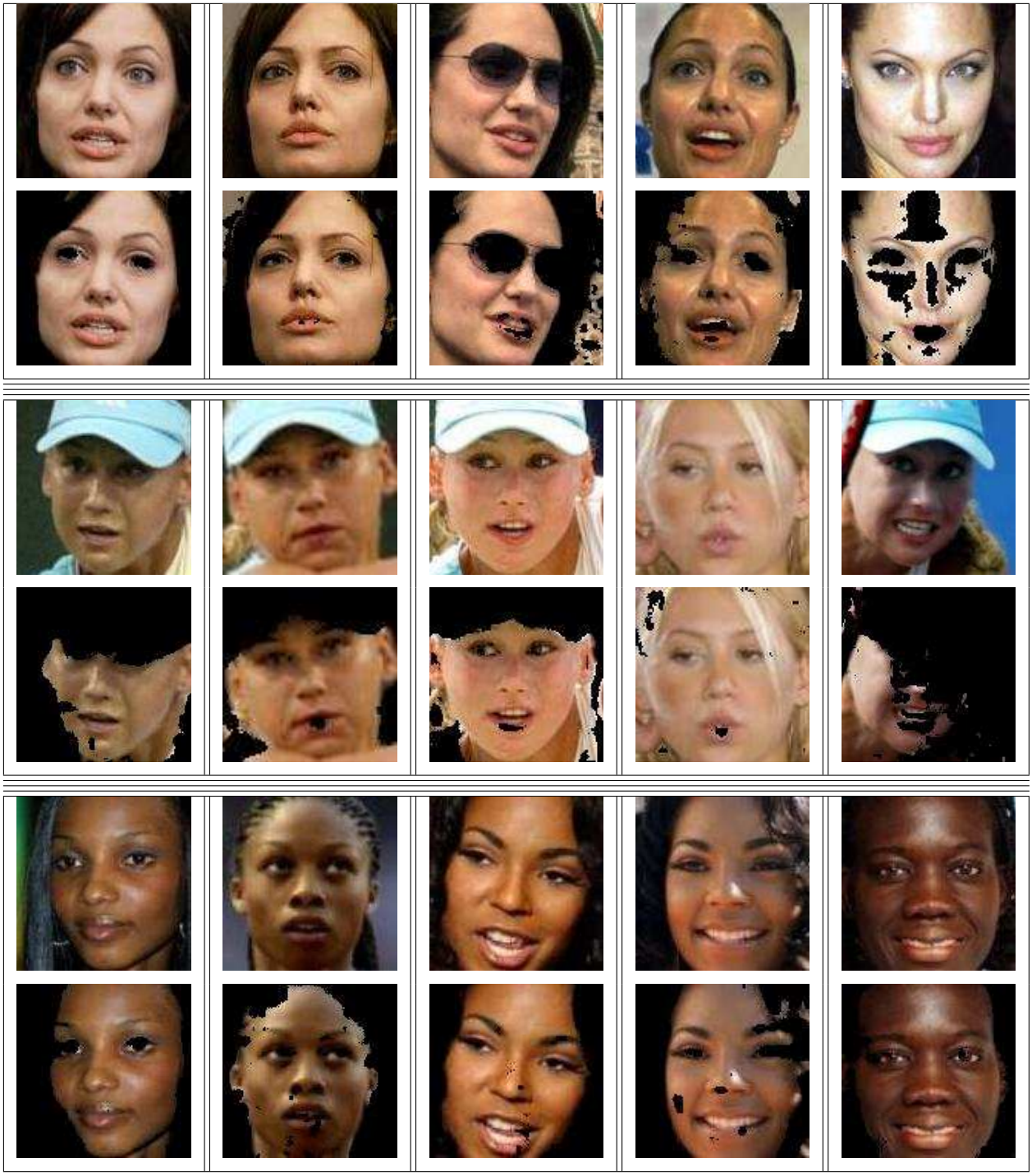


Figure 5.5: Examples of skin color extraction within the face: For each face image, the corresponding extracted skin part image appears below it.

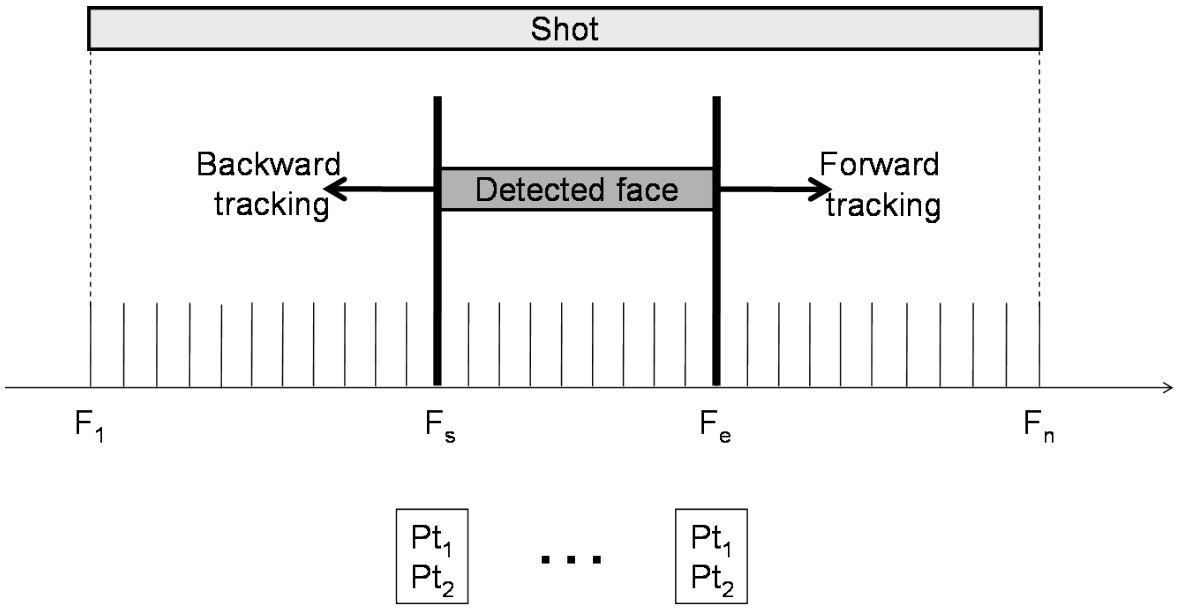


Figure 5.6: The backward-forward tracking scheme.

1. For the backward (*respectively* forward) tracking, two points are estimated in the frame F_{s-1} (*respectively* F_{e+1}) as follows:

$$\begin{aligned} Pt'_1 &= Pt_1 - \alpha(Pt_2 - Pt_1) \\ Pt'_2 &= Pt_2 + \alpha(Pt_2 - Pt_1) \end{aligned} \tag{5.3}$$

   where Pt_1 and Pt_2 are the corners of the face box obtained on the starting frame F_s (*respectively* F_e) and α a fixed coefficient (for example $\alpha = 0.1$).

   Pt'_1 and Pt'_2 delimit the estimated box in which the candidate face may be present.

2. each pixel (i,j) within the box is evaluated using the probability function:

$$p(x) = \frac{1}{2\pi\,|\Sigma|^{\frac{1}{2}}} \exp[-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)] \tag{5.4}$$

   where $x = (i, j)$, and the mean μ and covariance Σ are adapted on the skin color of the frame F_s (*respectively* F_e).

3. Because the face is considered as a single entity, pixels are processed using the dilation and erosion morphological filters.

4. if the Ratio of the Skin Part (RSP) within the box is higher than a certain threshold Thr (e.g. $Thr = 0.6$ means that the skin is detected in more than 60% of the face region),

the face is considered as present and the points Pt_1 and Pt_2 are updated according to the proper box in the candidate image, the 2D Gaussian model is also updated with the new data and the process is repeated for frame F_{s-2} (*respectively* F_{e+2}) starting from step 1. If RSP is lower than Thr or if the boundaries of the shot are reached, the iterative process is stopped.

We find a method to compute the threshold Thr according to the coefficient α by considering some assumptions: we suppose that theoretically, that the skin part occupy an area A_s of more than 0.7 of the face box. The ratio of the skin part can be computed as follows:

$$RSP = \frac{A_s}{xy} \tag{5.5}$$

If we assume that there is no change in the face aperture and movement, the RSP in the new estimated box that depends from α will be:

$$RSP' = \frac{A_s}{(1+2\alpha)x(1+2\alpha)y} = \frac{A_s}{(1+2\alpha)^2xy} = \frac{RSP}{(1+2\alpha)^2} \tag{5.6}$$

Therefore, the adapted threshold Thr will be equal to $\frac{0.7}{(1+2\alpha)^2}$ (e.g. $Thr = 0.4861$ for $\alpha = 0.7$).

5.5.2 Clothing-based people tracking

In some cases, the face tracking fails because the face may be occluded or the skin color model is not well estimated. In this case, we can use clothing feature to enable the tracking.

After extracting the clothing (cf. section 5.4), the image is converted to HSV system, and then a 3D or 2D histogram (in some works, the V component is not considered in orded to reduce the lighting effect) is computed on the clothing zones as seen in Fig.5.7.

The backward-forward tracking of the clothing is iteratively done using histograms comparison between the clothing box of F_s (respectively F_e) and the estimated clothing box of F_{s-1} (respectively F_{e+1}) computed using equations 5.3. The distance used for comparison is the Bhattacharyya distance:

$$D_{Bhattacharyya} = -\ln\left[\sum_i \sum_i \sum_k h_1(i,j,k)(h_2(i,j,k)\right] \tag{5.7}$$

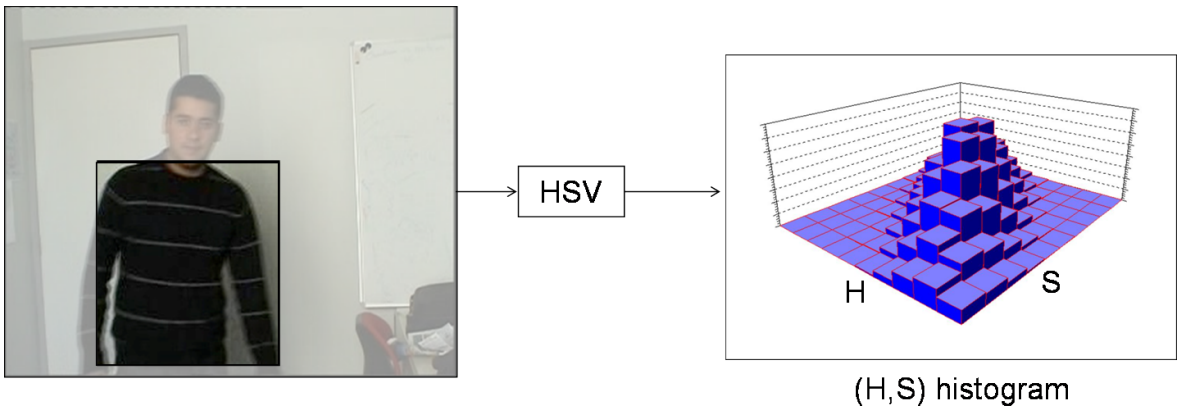


Figure 5.7: 2D histogram for the H and S components computed on the clothing zone.

Finally, the new value of Pt_1 and Pt_2 are updated by computing the optimal values that correspond to the following expression:

$$\underset{(Pt_1^{'},Pt_2^{'})}{\arg\min}[D(cloth_s, cloth_{s-1}(Pt_1^{'}, Pt_2^{'})] \tag{5.8}$$

where $cloth_s$ corresponds to the clothing extracted in frame F_s and $cloth_{s-1}$ corresponds to the estimated box delimited by $Pt_1^{'}$ and $Pt_2^{'}$ ($Pt_1^{'}$ and $Pt_2^{'}$ are chosen in the neighbors of the old Pt_1 and Pt_2).

5.6 Proposed methods for people clustering

The visual clustering consists in grouping all sequences of images that contain the same appearing person. It is obvious that the most reliable descriptor for doing that is the face. But also, other visual features can be helpful like the clothing and the background. In the following sections, we will present our proposed methods for face-based clustering using SIFT features, for clothing based clustering and then the hierarchical bottom-up algorithm.

5.6.1 Face-based clustering

Face is a very important discriminate high level feature: the skin color, the hair, the geometrical layout, the ears, the eyes, the mouth, the nose are all descriptors that might be used to recognize people. But the variation in illumination between the video documents or even within the document, the different kinds of face scales we may have, the pose variation, the partial occlusions, etc. are constraints that make the task of face-based clustering difficult.

In this work, we decide to study three face descriptors: the skin color that was analyzed in section 5.5.1, the hair and the SIFT descriptors. Moreover, instead of processing the whole sequence of faces which is time consuming, we decide to work only on *key-faces*: for every sequence of frames, we choose one face that must be the most representative containing the maximum amount of information.

5.6.1.1 Choice of the key-face

We define a list of criteria that the *key-face* must respect:

1. The area $(w * h)$ of the face box must be as large as possible (cf. Fig. 5.8 (a)), In our experiments, we found that the use of $[\min(w, h)]^2$ is slightly better than $w * h$.

2. The ratio of skin part (RSP) within the face box must be as high as possible (cf. Fig.5.8 (b)).
$$RSP = \frac{number_of_skin_pixels}{total_number_of_pixels} \tag{5.9}$$

3. The width (w) to height (h) ratio must be as close as possible to 3/4 which is used in many face recognition databases [SH94] [BHaPHK96] (cf. Fig.5.8 (c)).

4. The face must be as frontal and vertically aligned as possible (cf. Fig.5.8 (d)). An indication of the face orientation is given by the image moment mu_{30} computed on the normalized gray-scale image I_g:
$$mu_{30} = \sum_x \sum_y (x - x_0)^3 I_g(x, y) \tag{5.10}$$
where (x, y) the coordinates of a pixel within the image and x_0 the mean of x horizontal values. The face is frontal and symmetric if mu_{30} is close to 0.

One good way to model the choice of the *keyface* K is to use the following expression:
$$K = \arg\max_k \left(\frac{RSP_k * [\min(w_k, h_k)]^2}{\left(1 + \left|\frac{w_k}{h_k} - \frac{3}{4}\right|\right) * (1 + |mu_{30})|} \right) \tag{5.11}$$
where $k \in [1, ..., N_k]$, N_k is the number of frames within the sequence.

Results detailed in section 6 show that the choice of the keyface using the above formula is widely better than arbitrary selecting the face of the middle frame within the sequence.

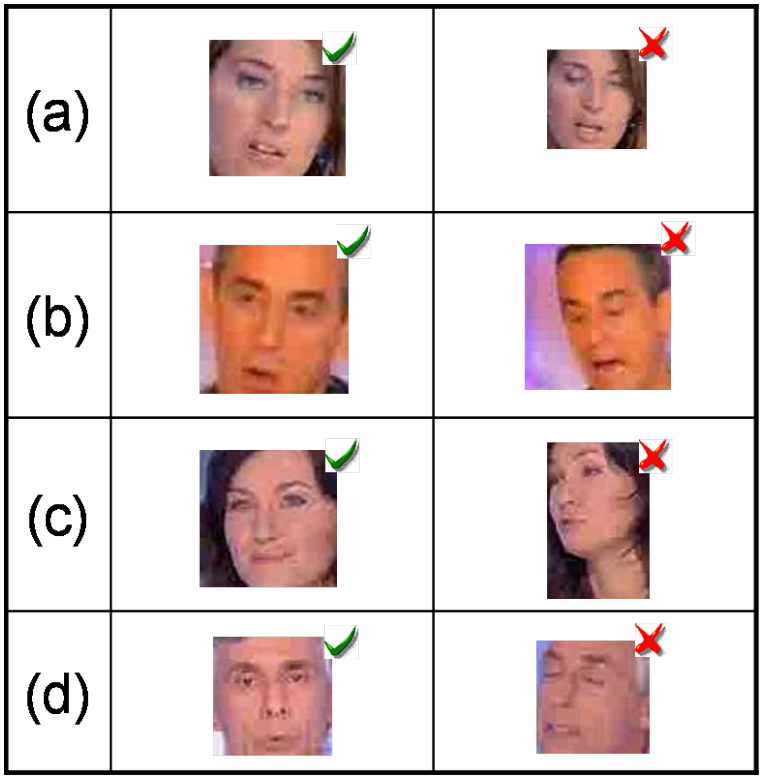


Figure 5.8: Choice of the key-face.

5.6.1.2 SIFT matching

As described previously in section 4.4.3, the SIFT features are invariant to scale, rotation, illumination and noise. Nowadays, they are used as a baseline features for object recognition in most successful systems like the Columbia University system [CHJ+08] presented in TRECVid 2008 evaluation competition where 20 High level different concepts were extracted. Systems performing this kind of task need huge training on positive and negative data. But in the case of clustering algorithm, there are two main differences:

1. The system is an unsupervised system where no initial training is allowed.

2. The challenging problem here is not to match between test and template images to see if there is a face in the tested image (as the above object recognition systems do), but the issue here is to verify if the two faces are assigned to the same person.

In order to process the matching between SIFT features, we were inspired by the works of [Low04] and [BLGT06] that were described in section 4.4.3.

In [Low04], the matching between keypoints in the first keyface and keypoints in the second keyface is done by computing the ratio of distances of the keypoint to its closest and its second closest neighbor. If this ratio is under a fixed threshold (0.8), the matching between this keypoint and its closest neighbor is considered to be correct. Fig.5.9 and Fig.5.10 show examples of correct and false matching between faces. First, we can notice that if the faces correspond to the same person, the number of matches is greater than case where the two faces are different. Second, we can notice that the matching between faces of the same person works even if there are illumination, scale and head pose changes.



Figure 5.9: Example of good matches under some variation in lighting, orientation and scale.

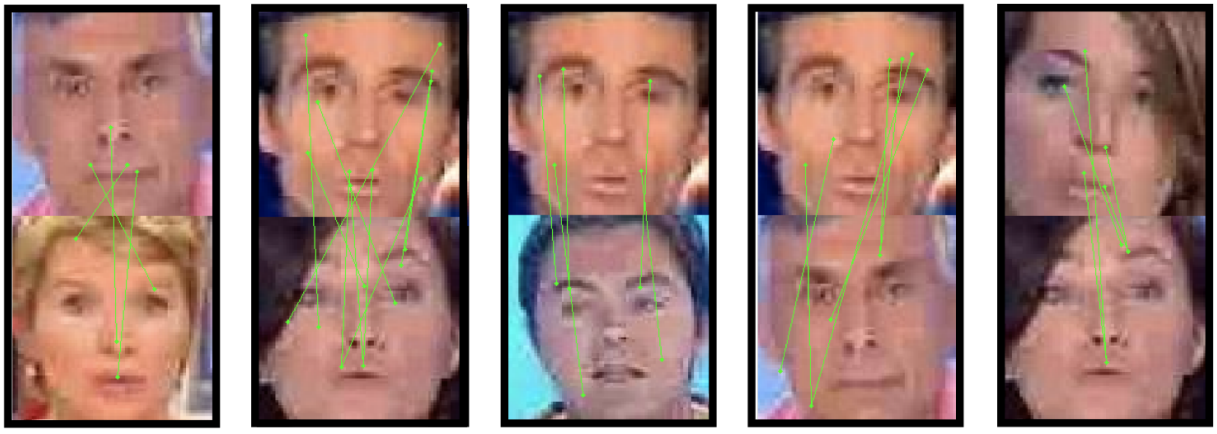


Figure 5.10: Example of bad matches.

The number of matches gives an idea about the fact that the two faces are identical or not, but it is not efficient because the number of extracted keypoints may vary from an image to

another: it is more probable to have more matches between images that both provide great numbers of keypoints than matching between images if at least one of them has few number of keypoints.

Moreover, there is some similarity between faces even though they do not correspond to the same person. There may be matching between features around the eyes, the mouse of those different faces. That is why in this case, we should find an additional criteria like the *minimum pair distance* in order to evaluate this matching.

In [BLGT06], the issue is to recognize/authenticate faces. The distance is computed between all pairs of keypoints and only the *minimum pair distance* (MPD) is thresholded to verify if the two faces correspond to the same person. In order to improve their system, authors used information about the eyes and the mouth to have location information, and then they divided the images into sub-images and do the matching between corresponding sub-images. There are many difficulties in this method of matching using a regular grid:

1. computing the position of the eyes and the mouth is another challenging task especially on low resolution of faces like in our framework,

2. there might be no extracted keypoints in a sub-image. That will distort the average *minimum pair distance.*

3. in some cases, two pairs of matched keypoints taken from the same pair of sub-images (those pairs correspond to the *minimum pair distance* and the *second minimum pair distance* for that pair of sub-images) may be more distinctive than taking only one pair from each sub-image.

Our algorithm consists in combining the strong ideas of both [Low04] and [BLGT06] papers. First, we consider two *keyfaces* $K1$ and $K2$ with the respective set of extracted SIFT features:

$$\begin{cases} F1 = f_1^{K1}, f_2^{K1}, ..., f_L^{K1} \\ F2 = f_1^{K2}, f_2^{K2}, ..., f_M^{K2} \end{cases}$$

After applying the Lowe's matching in terms of ratio of distances to the first and second closest keypoints in the feature space, a new set of pairs of matched keypoints is provided:

$$P = \{p_1, p_2, ..., p_Q\} \tag{5.12}$$

where p_i is a pair of features $(f_{i_1}^{K1}, f_{i_2}^{K2})$ and $Q \leq \min(L, M)$.

Second, we compute the distance D_{p_i} for each pair of keypoints. Those keypoints are then sorted ascending (i.e. from the minimum to the maximum distance). After that, only the first N pairs are selected to compute their average distance value that we call the "Average of the N-Minimum Pair Distances" ANMPD:

$$D_{sift} = ANMPD = \frac{1}{N}\sum_{i=1}^{N} D_{p_i} \tag{5.13}$$

This average distance is used as a merging criterion in the hierarchical bottom-up clustering (cf. section 5.6.3).

Experiments show that the best value of N is 5 (cf. Tab.6.4 in chapter 6). An example of good matching between faces under different conditions using SIFT is shown in Fig.5.11.



Figure 5.11: Example of 13 faces of the same person that were correctly matched using ANMPD distance: we can notice different facial expressions, lightning conditions, glasses and occlusions. This example is taken from the AR database [MB98].

5.6.2 Clothing based clustering

Since within video documents like debates, TV games, movies and series, a character is wearing the same clothing during all the document or on at least a short period of time (especially for movies), clustering using clothing information of the person is a significant solution. In our work, we investigate three clothing descriptors: the dominant color, the 3D histograms and the texture.

Histograms Comparison The comparison of the 3D histograms of the clothing box is done using the bhattacharyya distance that was previously expressed in equation 5.7. This distance is used as a merging criterion in the clustering process. However this distance can be influenced with some noise due to the background clutter or the foreground occlusions like the examples

shown in 5.12. To eliminate this noise we decide to extract the dominant color and then apply the histograms comparison on dominant colors.



Figure 5.12: Two people with two different costume box: the noise is due to the background and to the foreground objects like hands and characters.

Dominant Color The extraction of the dominant color we applied is inspired from the work of [HJC06]. The main difference is that our method considers that the dominant color is distributed on a margin of colors in the RGB or HSV space unlike the method used in [HJC06] where the extracted dominant color is a unique triplet of (R,G,B) or (H,S,V) values. In our work, we decide to use the HSV space since it gives slightly better results than RGB (about 1 to 2% improvements).

We consider the costume box presented in the image (a) of Fig.5.13.

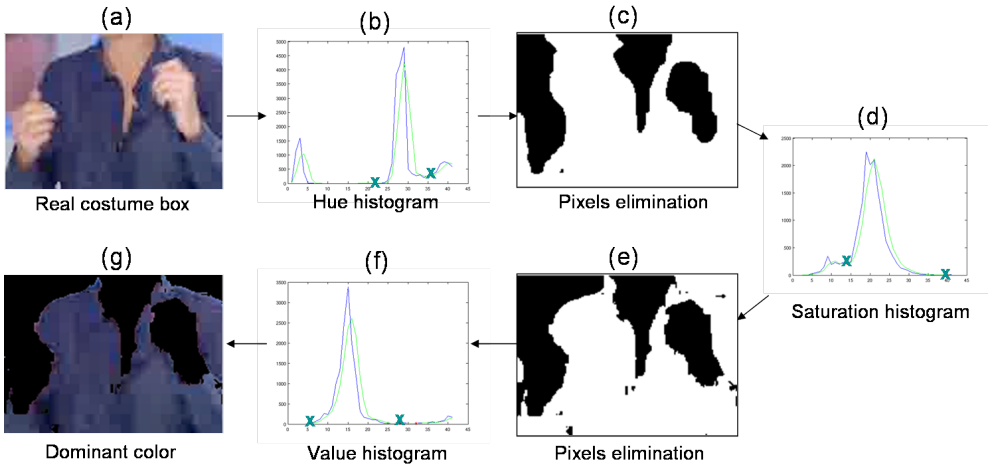


Figure 5.13: Extraction of the dominant color.

Five successive steps are done in order to extract the dominant color:

1. In the HSV space, we plot the *Hue* histogram as seen in figure 5.13.b, then a smoothing process is done in order to eliminate local minima. The maximum value is found on the histogram and its two minimum adjacent neighbors are selected. The most represented hue is located in the margin delimited by those two minima.

2. We return back to the image and we exclude all pixels where the Hue value does not correspond to the selected margin. In Fig.5.13.c, the eliminated pixels are represented in black while the pixels left are illustrated in white.

3. On the pixels left, the *Saturation* histogram is computed. Then, the most represented saturation is selected like in step 1. (cf. Fig.5.13.d).

4. Again, the pixels that do not correspond to the saturation margin are eliminated as illustrated in Fig.5.13.e.

5. The same process of searching for the most representative value is done (Fig.5.13.f) and the corresponding pixels are selected.

Finally, as seen in Fig.5.13.g, the dominant color is extracted from the image box while the black color corresponds to the eliminated pixels. More examples are shown in Fig.5.14 where the clothing and its dominant color are shown.

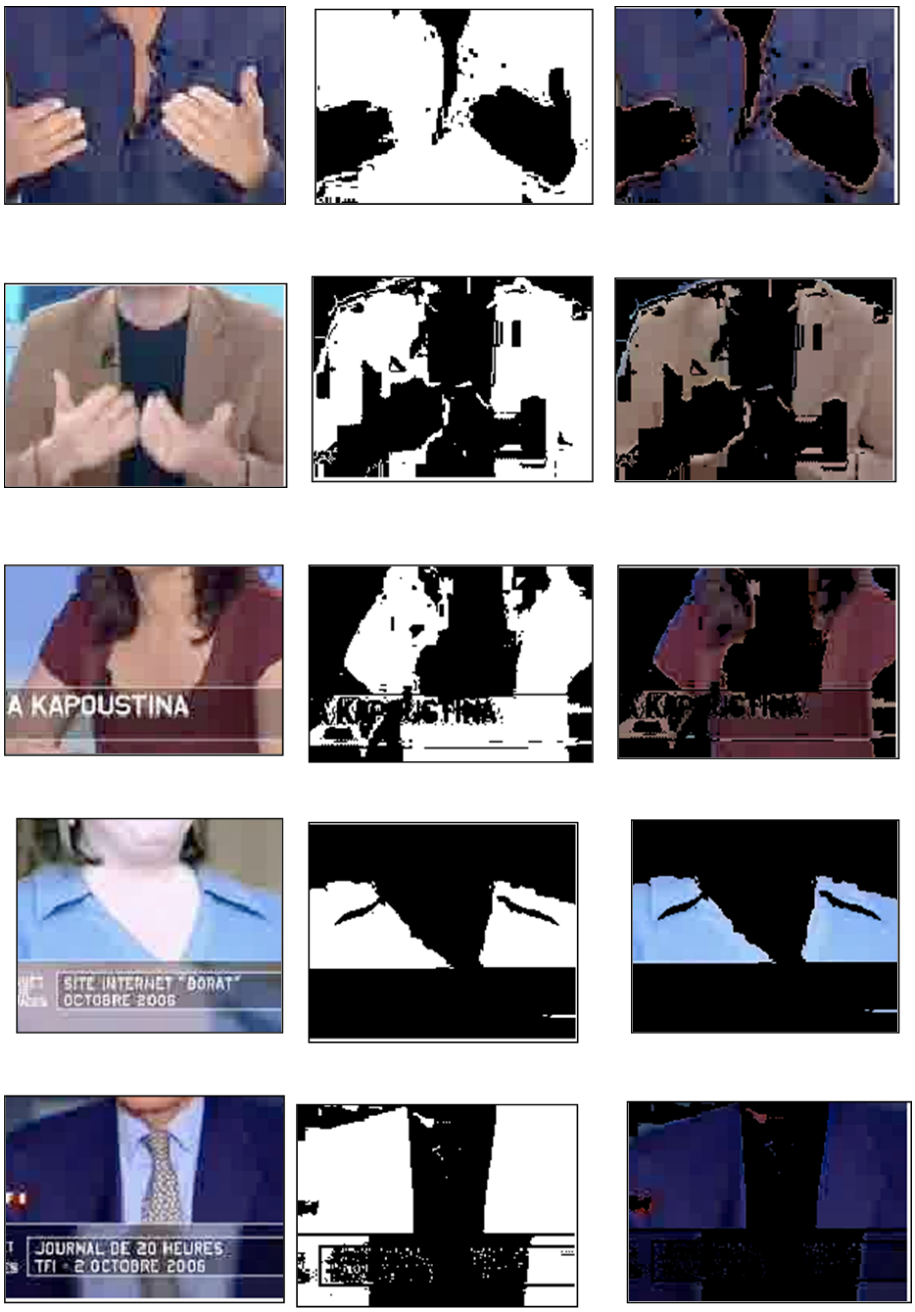


Figure 5.14: Examples of dominant color areas extracted.

Texture In this work, we use the Gabor texture feature vector [MM96] that was previously introduced in section 4.1. In order to compute the distance between the textures of two different clothes i and j, we compute the normalized distance in the feature space between the corresponding feature vectors F^i and F^j.

$$\begin{cases} F^i = [f_1^i, f_2^i, ..., f_Q^i] \\ F^j = [f_1^j, f_2^j, ..., f_Q^j] \end{cases} \tag{5.14}$$

The distance is defined by:

$$D(i,j) = \sum_q \left| \frac{f_q^i - f_q^j}{\alpha(f_q)} \right| \tag{5.15}$$

where $\alpha(f_q)$ is the standard deviation of the q^{th} coefficient of the feature vector all over the database.

5.6.3 Hierarchical bottom-up clustering

After listing the different kinds of face and costume features that can be used to help clustering tracks that correspond to the same person, the issue here is to find an efficient way to combine all this information in order to perform the most accurate clustering. It is obvious that tracks that verify all the merging criteria listed above are favored to be merged. But in some cases where illumination, background clutter and clothing may change, some of the above criteria will not be verified. In this case, we give more confidence to some special descriptors. That is why we decide to do a 3-levels hierarchical clustering:

- **First-level hierarchical clustering.** This step is illustrated in figure 5.15. After extracting face and clothing features, distance matrices D_1 (SIFT), D_2 (Skin), D_3 (Histogram) and D_4 (Texture) are reconstructed by computing the appropriate distance between every pair of tracks in terms of the corresponding feature. Then we define a similarity matrix that combines all the above matrices. Every element of that matrix is computed using the following expression:

$$S(i,j) = \prod_{a=1}^{A} \max(Thr_a - D_a(i,j), 0) \tag{5.16}$$

  where $S(i,j)$ denotes the similarity between the i^{th} track T_i and the j^{th} track T_j where i and j varies from 1 to $N1$ which is the number of tracks. $S(i,j)$ may be even positive if there is good matching or equal to 0 if at least one of the descriptor disagrees the matching. $D_a(i,j)$ is the distance between T_i and T_j in terms of the a^{th} descriptor. Thr_a is the threshold that corresponds to the a^{th} descriptor. It is tuned by processing the clustering method using only this descriptor (cf. table 6.6). In this study, $A = 4$ since there are only 4 descriptors.

  Then, the clustering is done between tracks/clusters that are similar in terms of the resulting similarity matrix. It is done in a hierarchical bottom-up manner, i.e. starting

from the most similar tracks/clusters, using the complete linkage property. After each merging between two tracks T_i and T_j, the matrices are updated by eliminating the i^{th} and j^{th} rows and the i^{th} and j^{th} columns and by inserting only a row and a column at the I^{th} position where $I = min(i, j)$ and their elements are computed as follows:

At the position k of the I^{th} row (or column),

- the distance based on the SIFT features of the face uses the single linkage:

$$D_{sift}(I, k \notin \{i, j\}) = min(D_{sift}(i, k), D_{sift}(j, k)) \tag{5.17}$$

- the distance based on the skin color of the face uses the average linkage:

$$D_{skin}(I, k \notin \{i, j\}) = \frac{n_i D_{skin}(i, k) + n_j D_{skin}(j, k)}{n_i + n_j} \tag{5.18}$$

where n_i and n_j are the number of skin pixels of the i^{th} and j^{th} tracks/clusters.

- the distance based on the color histogram of the clothing uses the full linkage:

$$D_{hist}(I, k \notin \{i, j\}) = D_{bhattacharyya}(H_I, H_k) \tag{5.19}$$

where

$$H_I = \frac{n_i H_i + n_j H_j}{n_i + n_j}$$

- the distance based on the texture of the clothing uses the average linkage:

$$D_{texture}(I, k \notin \{i, j\}) = \frac{D_{texture}(i, k) + D_{texture}(j, k)}{2} \tag{5.20}$$

The appropriate linkage type for each descriptor is chosen according to the nature and the behaviour of this descriptor.

Consequently, the updated similarity matrix is computed using equation (5.16). The clustering is repeated until the stopping criterion is verified i.e. when all similarities are equal to 0.

At the end of the clustering, a new set of clusters (N_2 clusters with $N_2 < N_1$) is obtained with their corresponding distance matrices as seen in figure 5.15.

- **Second-level hierarchical clustering.** After a first clustering where the merging confidence is very high, a second clustering is done in terms of the **clothing similarity**. In this case, two sufficient conditions should be verified:

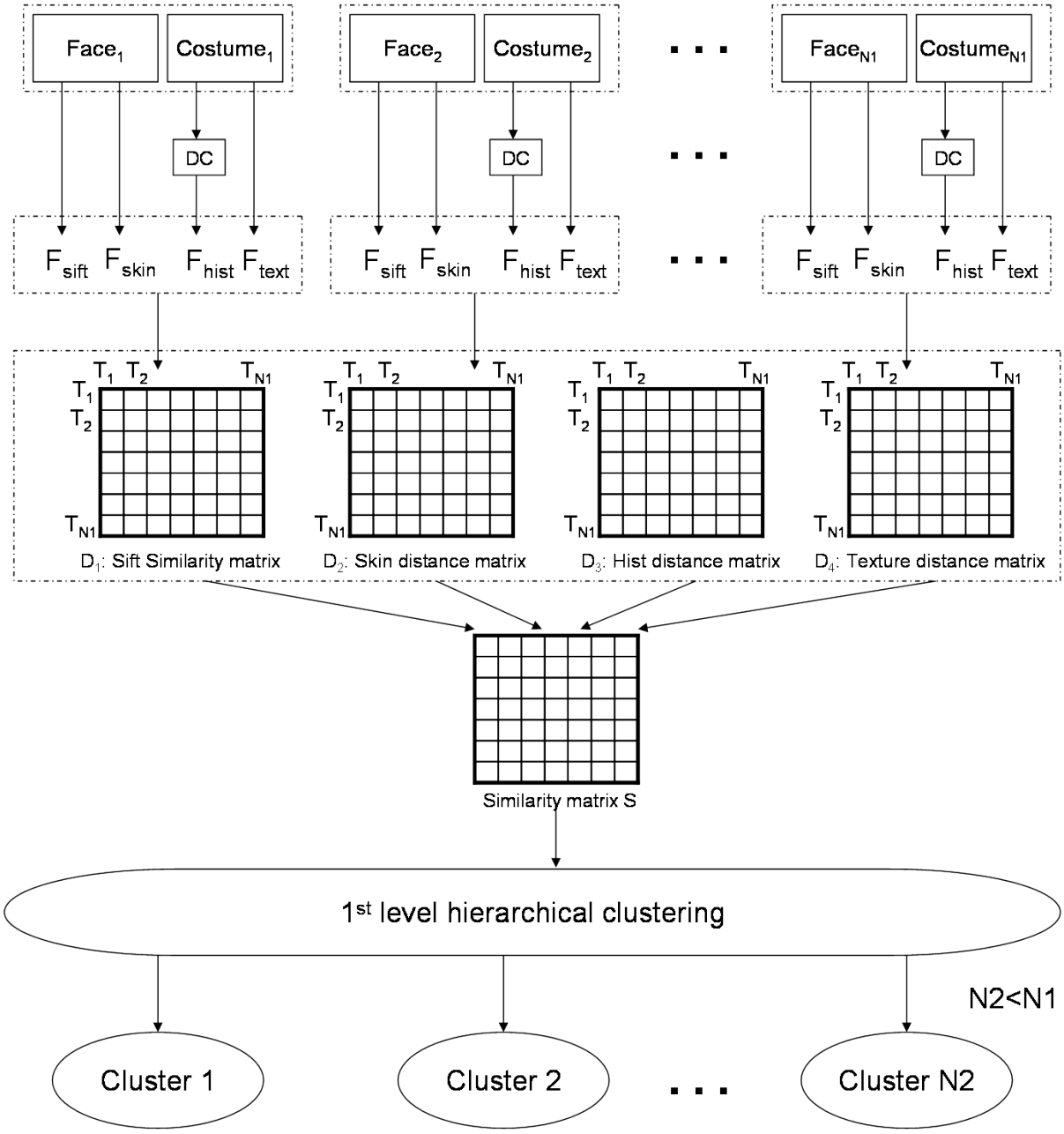


Figure 5.15: First-level hierarchical clustering.

- at least one among the two clothing descriptors is working: the second descriptor may fail if there are partial occlusions (the texture descriptor fails!) or lightning variations (the color histogram comparison fails!);

- at least one among the two face descriptors is working: it is taken into account in order to prevent merging between two people that are wearing the same clothing.

The above constraints are expressed by the following formula:

$$S(i,j) = \max(S_{13}(i,j), S_{14}(i,j), S_{23}(i,j), S_{24}(i,j)) \tag{5.21}$$

where

$$S_{ab}(i,j) = \min(D_a(i,j) - Thr_a, 0).\min(D_b(i,j) - Thr_b, 0) \tag{5.22}$$

S_{13} is the similarity based on the SIFT features of the face and the histogram of the clothing, S_{14} is the similarity based on the SIFT features of the face and the texture of the clothing, S_{23} is the similarity based on the skin color of the face and the histogram of the clothing, and S_{24} is the similarity based on the skin color of the face and the texture of the clothing.

After each merging between two clusters, the matrices are updated as above. The clustering is repeated until the stopping criterion is reached, i.e. all similarities are equal to 0.

- **Third-level hierarchical clustering.** When the illumination varies or the clothing of the person changes, color-based features and texture features are subject to change. In this case, the only confident features that will remain useful are the SIFT features on faces. That is why a final clustering step must be done according only to SIFT features. This clustering is repeated until the stopping criterion is verified i.e. all similarities are higher than Thr_1.

Chapter 6

Experiments and Results

Contents

6.1 Evaluation tool 107

6.2 Corpora 108

6.2.1 Development corpus 108

6.2.2 Test corpus 108

6.3 Experiments on the development set 110

6.4 Results on the test set 112

6.1 Evaluation tool

After building our face-and-clothing people indexing system, the next step aims to evaluate it. At the beginning, we must mention that many components of that system can be evaluated each one apart: the shot boundaries detector (that was tested in section 5.2), the people detector, the people tracker and the people clustering algorithm. But since our main contribution lies mostly on the clustering method, and since the rules for manual annotation lead to many ambiguities in how to consider the presence of a person in a video, we decide in this chapter to evaluate only the clustering part.

In order to mesure the performance of that clustering, we are inspired by the work done in the speech processing community to evaluate speaker diarization systems. The tool we use is defined by the speech group of NIST[19].

[19] http://www.itl.nist.gov/iaui/894.01/

Thus, the people clustering task is evaluated according to the errors that occur when person turns detected by the automatic system do not match the expected person turn in the ground-truth. It means that the error is measured by computing the overall person time that is attributed to the wrong person.

$$Err = \frac{\sum_{Allseqs} (dur(seq) * (min(N_R(seq), N_S(seq)) - N_C(seq)))}{\sum_{Allseqs} (dur(seq).N_R(seq))} \tag{6.1}$$

where for each sequence *seq*:

- dur(*seq*)=the duration of *seq*,
- $N_R(seq)$= the number of people appearing in *seq* according to the **reference**,
- $N_S(seq)$= the number of people appearing in *seq* according to the **system**,
- $N_C(seq)$= the number of **correct** matching, i.e. the number of people appearing in *seq* for whom their matching (mapped) system people are also appearing in *seq*.

6.2 Corpora

Since there is no training step needed in this work but only a step of fixing parameters and tuning thresholds, we divide our data into 2 sets: a development corpus and a test corpus.

6.2.1 Development corpus

The development corpus contains 520 tracks of a *talk show* program of about 40 minutes length where many reports and movie scenes occur. The annotation time for that document took about 12 hours. This is due to the fact that more than one person may appear in the same shot. The total number of the people appearing in this video is equal to 25: 4 of them appear with two different clothing and 3 others have the same clothing appearance. The resolution of the images is 320x240.

6.2.2 Test corpus

The test corpus was chosen in order to cover all the possible types of video data (news, debates and movies):

- Table 6.1 describes the broadcast news corpus: two files of American news: "19980104_ABC" (ABC), "19980202_CNN" (CNN) chosen from TRECVid 2003 corpus, two files of French news: "JT_F2_20h_13_05_03" and "JT_F2_20h_18_05_03" (France 2) chosen from the Argos 2007 corpus, one file of Lebanese news: "20041117_200000_LBC_LBCNEWS_ARB" (LBC) chosen from TRECVid 2005 corpus and one file of Chinese news: "CCTV_2009" (CCTV) that was personally recorded. The total duration of those files is 4 hours, 5 minutes and 22 seconds. We annotate semi-automatically those files in terms of people appearing in every image (we annotate only faces that were detected by our face detector). The clustering annotation period took about 50 hours of work.

Table 6.1: News corpus.

	description	main talking language	duration length	number of people
ABC news	American	english	1708 sec	79
CNN news	American	english	1779 sec	55
France2 news (1)	French	french	2496 sec	145
France2 news (2)	French	french	2231 sec	117
LBC news	Lebanese	arabic	3500 sec	156
CCTV news	Chinese	french	3008 sec	74

- Table 6.2 describes the broadcast debates corpus. All are french programs: two files of the program "Le Grand Journal", one file of the program "C'est notre affaire" and one file of the program "C'est dans l'air". Their total duration time is 3 hours, 30 minutes and 36 seconds. The manual annotation took about 70 hours of work.

- Table 6.3 describes the movies corpus. This corpus contains excerpts from the following movies: "Les Choristes", "Amelie", "Virgins Suicide" and "Asterix Obelix". The total duration of the annotated part is 3 hours 4 minutes and 42 seconds. The annotation period is estimated to 100 hours of work due to the difficulty of this task in the chosen movies: many persons with the similar clothing, different lightning conditions, variation in the face size, pose variation, etc.

Table 6.2: Debates corpus.

	Channel	main talking language	duration length	number of people
Le Grand Journal (1)	Canal+	french	3599 sec	74
Le Grand Journal (2)	Canal+	french	3524 sec	167
C'est notre affaire	France 5	french	1602 sec	21
C'est dans l'air	France 5	french	3911 sec	49

Table 6.3: Movies corpus.

	description	main talking language	duration length	number of people
Les Choristes	French	french	3600 sec	139
Amelie	French	french	2645 sec	50
Virgins Suicide	American	english	2191 sec	93
Asterix Obelix	French	french	2646 sec	96

6.3 Experiments on the development set

Six experiments are processed on the development set. Their goal is not only to validate the proposed measures and methods, but also to tune the parameters and the thresholds of the system.

The first experiment is done in order to choose the best value of N for the proposed ANMPD method for SIFT (cf. section 5.6.1.2). Tab.6.4 reports the minimum clustering error rate (CER) obtained for the different values of N. It shows that the CER decreases and then increases with a minimum value at N=5. In next experiments, we fixed the value of N to 5.

Table 6.4: Clustering Error Rate for different N values used in Equation 6.

N	1	2	3	4	**5**	6	7	8
CER(%)	55.1	49.1	32.8	31.2	**28.4**	30.2	33	35.1

The second experiment aims to study the impact of the keyface selection. Results show that the arbitrary choice of the middle face gives a CER equal to 43.7%. However, the proposed method for selecting keyfaces gives a CER equal to 28.4%.

The third experiment aims to compare the ANMPD with Lowe's matching and Minimum pair distance matching. Results in Tab.6.5 show that our proposed method outperforms the Lowe's matching by an absolute gain of 7.5%, the MPD method by 26.7% and the MPD on regular grid by 3%.

Table 6.5: Comparison between different sift matching techniques: Lowe's matching, MPD matching, MPD matching on regular grid and the proposed ANMPD matching.

	Lowe's matching	MPD	MPD on regular grid	proposed ANMPD
CER(%)	35.9	33.2	31.4	28.4

The fourth experiment aims to compare the clustering using each descriptor alone. Tab.6.6 shows that the descriptor that gives best results is the 3D-Histogram of the clothing with a $CER = 16.8\%$. The second good results are provided by SIFT matching with $CER = 28.4\%$. The two other descriptors are consecutively the clothing texture and the skin color of the face. The corresponding stopping criteria for each descriptor are also reported. These thresholds are used in equation 5.16 and 5.22 to compute similarity matrices for the hierarchical clustering.

Table 6.6: Minimum clustering error rate for each visual descriptor: 3D-Histogram of the clothing, texture of the clothing, skin color of the face, and sift features of the face. The thresholds that correspond to the stopping criterion are also reported.

	SIFT	Skin	Hist	Texture	Fusion
CER (%)	28.4	56.6	16.8	55.5	13.0
Stopping criterion	$Thr1 = 0.41$	$Thr2 = 3.2$	$Thr3 = 3.3$	$Thr4 = 0.126$	-

The fifth experiment is done to report the behavior of the proposed fusion method compared to the four descriptors and at different levels of the clustering process. Fig.6.1 shows that the proposed clustering is better than almost all descriptors each one taken alone. For example:

- when the number of clusters is equal to 400, the proposed clustering outperforms the best one (skin color descriptor) by an absolute gain of 1.7%.

- when the number of clusters is equal to 250, the CER of the proposed method is equal to 44.8% however the best of the four descriptors was the SIFT with CER equal to 46.4%.

- when the number of clusters is equal to 25, the CER of the proposed method is equal to 14.5% however the best of the descriptor was the 3D-histogram of the clothing with CER equal 42.3%.

- the best CER value is 13%. It is obtained for a number of clusters equal to 50.

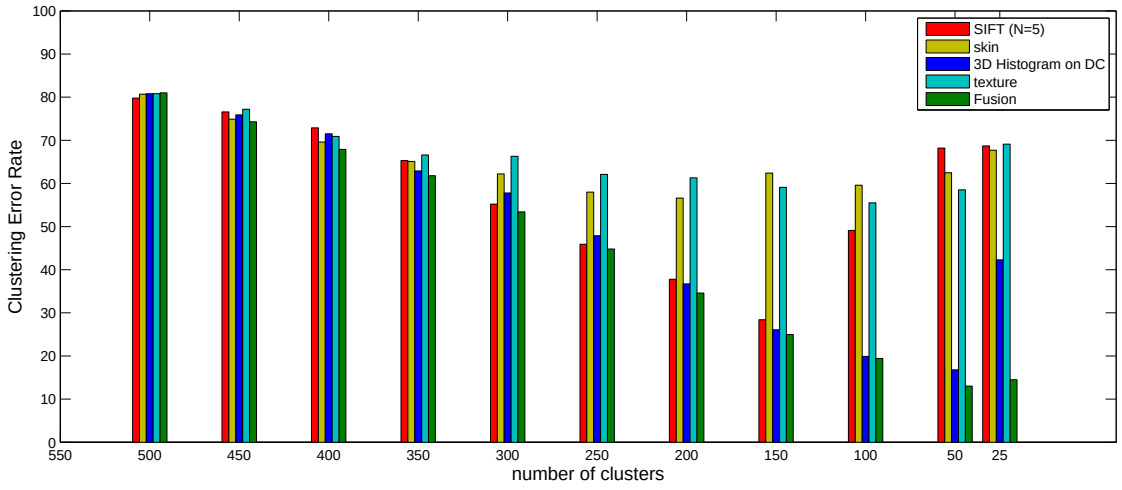


Figure 6.1: Comparison between the four features and the proposed clustering method.

The sixth experiment is done in order to evaluate the impact of using the dominant color. Fig.6.2 shows that at the beginning of the clustering process (number of cluster higher than 200), no real comparison can be made. However, when the number of clusters approaches the real number of people, the impact of using the dominant color is highest: when the number of clusters is equal to 50, the absolute gain is 34.9%.

Fig.6.3 and Fig.6.4 illustrate the clusters obtained at the end of the hierarchical clustering. Each of these clusters corresponds effectively to only one person under different lightning, pose and scale conditions.

6.4 Results on the test set

In the following experiments, we use the same thresholds as the one fixed for the development set. Tables Tab.6.7, Tab.6.8 and Tab.6.9 show the CER of the different existing methods for SIFT

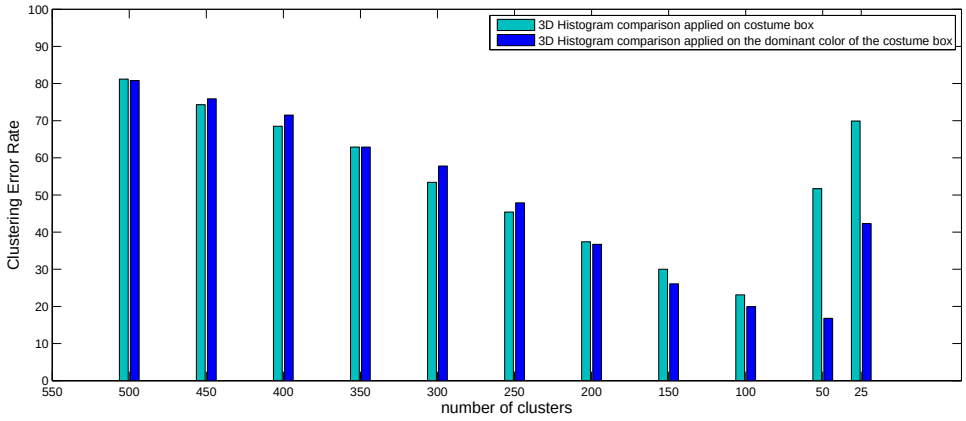


Figure 6.2: Comparison between applying the Histogram comparison directly on the costume box and applying it on the dominant color area.

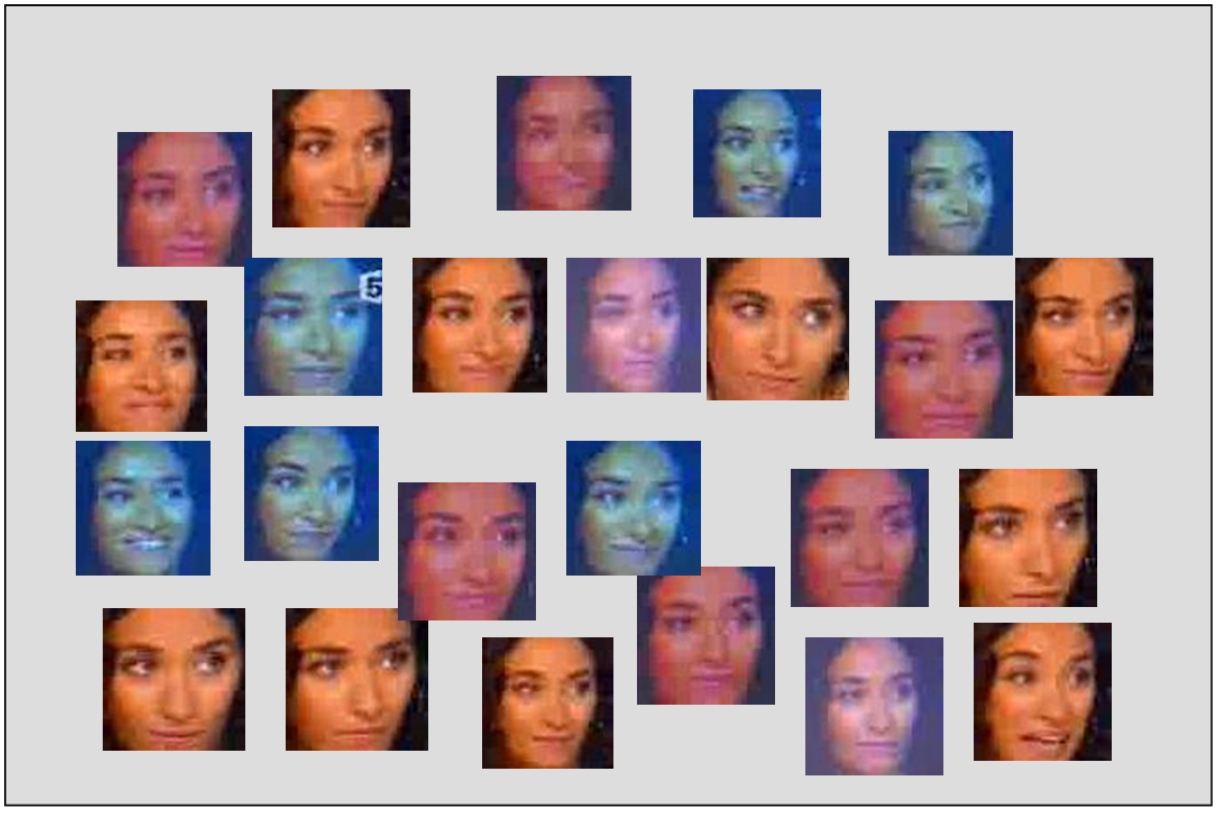


Figure 6.3: Example 1 of cluster delivered at the end of the clustering process of "arret sur image" TV debates.

compared to our proposed method based on ANMPD. For news, the absolute improvements of our method is 12.26% comparing to Lowe's matching, 10.41% comparing to MPD matching

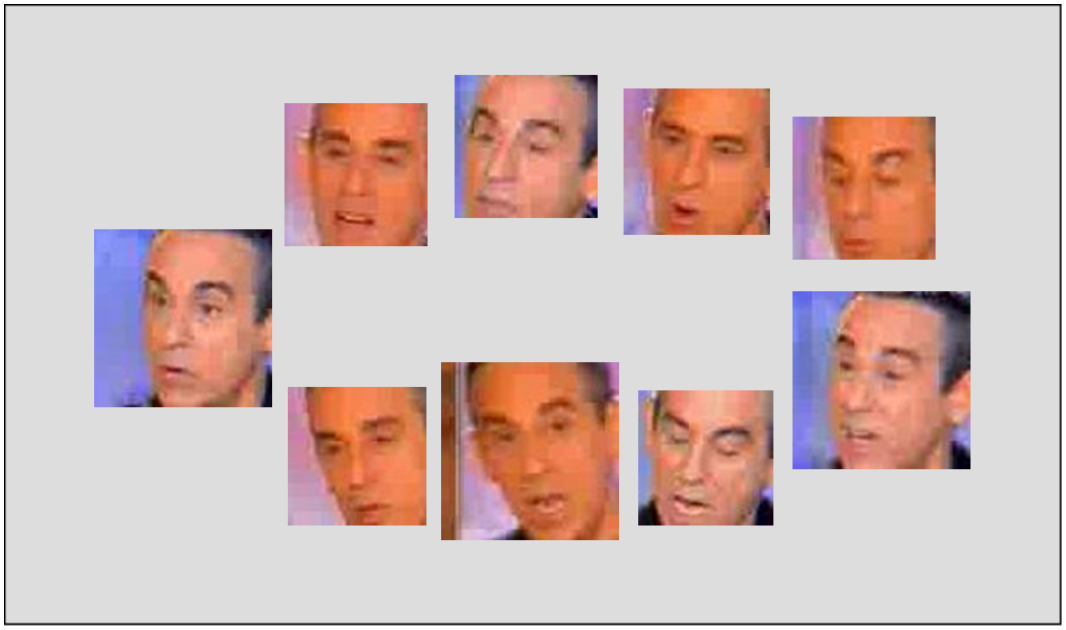


Figure 6.4: Example 1 of cluster delivered at the end of the clustering process of "arret sur image" TV debates.

and 10.07% comparing to MPD matching on regular grid. For debates, the proposed ANMPD method outperforms the old methods by 17%. For movies, the improvement is over 14%. The error is computed in terms of the weighted average CER because it takes into account the time of detected faces in each file.

Table 6.7: Comparison between different SIFT matching techniques on broadcast news: Lowe's matching, MPD matching, MPD matching on regular grid and the proposed ANMPD matching.

	Lowe's matching	MPD	MPD on regular grid	proposed ANMPD
ABC news	10.1	5.86	7.72	**5.22**
CNN news	31.66	31.36	31.66	**26.76**
France2 news (1)	18.39	16.31	11.54	**4.26**
France2 news (2)	25.57	23.94	25.32	**4.85**
LBC news	26.63	25.00	25.18	**15.82**
CCTV news	14.10	12.15	11.86	**5.27**
Weighted average CER (%)	22.22	20.37	20.03	**9.96**

Table 6.8: Comparison between different SIFT matching techniques on debates: Lowe's matching, MPD matching, MPD matching on regular grid and the proposed ANMPD matching.

	Lowe's matching	MPD	MPD on regular grid	proposed ANMPD
Le Grand Journal (1)	58.4	49.8	50.7	**28.9**
Le Grand Journal (2)	44.89	44.18	40.61	**27.72**
C'est notre affaire	65.98	61.51	62.32	**54.91**
C'est dans l'air	33.43	40.14	39.54	**23.28**
Weighted average CER (%)	43.09	44.33	43.27	**27.51**

Table 6.9: Comparison between different SIFT matching techniques on movies: Lowe's matching, MPD matching, MPD matching on regular grid and the proposed ANMPD matching.

	Lowe's matching	MPD	MPD on regular grid	proposed ANMPD
Les Choristes	61.70	59.34	59.59	**44.48**
Amelie	72.04	69.59	69.64	**62.33**
Asterix Obelix	63.92	63.77	63.75	**43.79**
Virgin Suicide	48.16	44.21	42.43	**31.69**
Weighted average CER (%)	61.44	59.30	59.07	**44.92**

Tables Tab. 6.10, Tab. 6.11 and Tab. 6.12 describe the behavior of the clustering in terms of each descriptor and each video file. Unlike in the development set, we can deduce that the most interesting descriptor is the SIFT. Then, come the color histogram descriptor, the skin color descriptor and the texture descriptor. Moreover, the fusion of those descriptors gives better results:

- For news, table 6.10 shows that the most confident descriptor is the SIFT descriptor with a weighted average CER of 9.96%. The influence of other descriptors is not very relevant because the weighted average CER of the fusion system is 9.10%.

  Figures Fig. 6.5, Fig. 6.6, Fig. 6.7, Fig. 6.8, Fig. 6.9 and Fig. 6.10 show that the intra-variation within the cluster is relatively low.

Table 6.10: Results of the different descriptors and the fusion clustering on Broadcast news: SIFT features of the face, skin color of the face, 3D-Histogram of the clothing and texture of the clothing.

	SIFT	Skin	Hist	Texture	Fusion
ABC news	**5.22**	45.05	15.5	18.72	**5.18**
CNN news	26.76	**19.59**	42.32	38.44	**18.35**
France2 news (1)	**4.26**	24.48	21.04	24.58	**3.94**
France2 news (2)	**4.85**	37.92	8.73	15.41	**5.22**
LBC news	**15.82**	23.80	28.59	32.10	**15.56**
CCTV news	**5.27**	34.65	19.22	16.16	**5.27**
Weighted average CER (%)	9.96	30.05	21.67	24.22	**9.10**



Figure 6.5: Example of a cluster delivered at the end of the clustering process on the ABC news.

- For debates, table 6.11 also shows that the SIFT descriptor is by far the best descriptor for clustering with a weighted average CER of 27.51%. However, the use of other descriptors give an additional improvement of 11.78%.

  Fig.6.11, Fig.6.12, Fig.6.13, Fig.6.14, Fig.6.15 and Fig.6.16 show that the resulting clusters contain faces that are more heterogeneous than faces in news. However they have generally the same lightning and the same scale.



Figure 6.6: Example of a cluster delivered at the end of the clustering process on the CNN news.



Figure 6.7: Example 1 of a cluster delivered at the end of the clustering process on France 2 news.

- For movies, the weighted average CER of the proposed clustering is 43.72% as seen in Tab.6.12. It is clearly higher than errors obtained on news and debates.

  As seen in Fig.6.18, 6.19, Fig.6.20, Fig.6.21, the high CER can be explained by the fact that there are high variation on many levels: lightening variation, face orientation, face size, intra-clothing variation (i.e. people are changing clothing), inter-clothing similarity

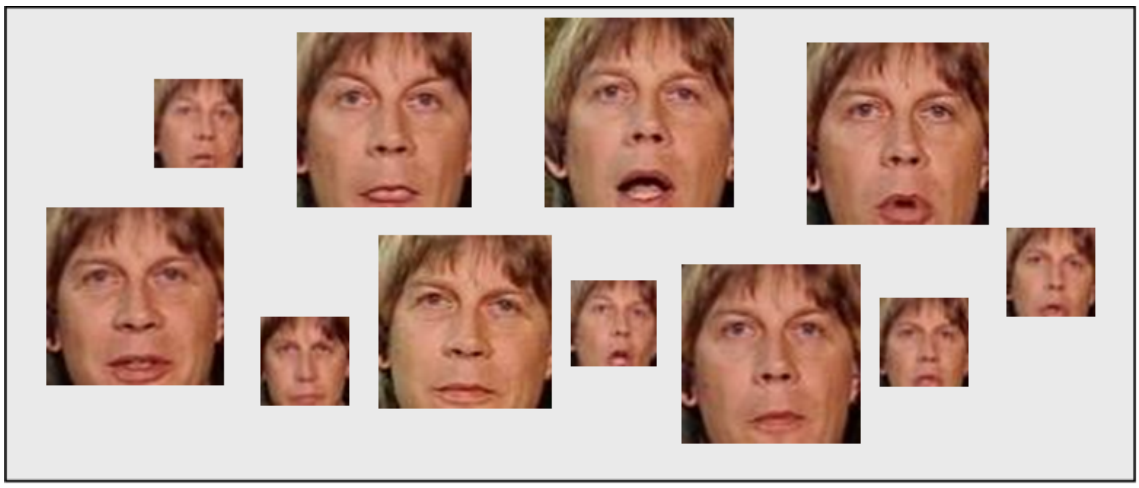


Figure 6.8: Example 2 of a cluster delivered at the end of the clustering process on the France 2 news.

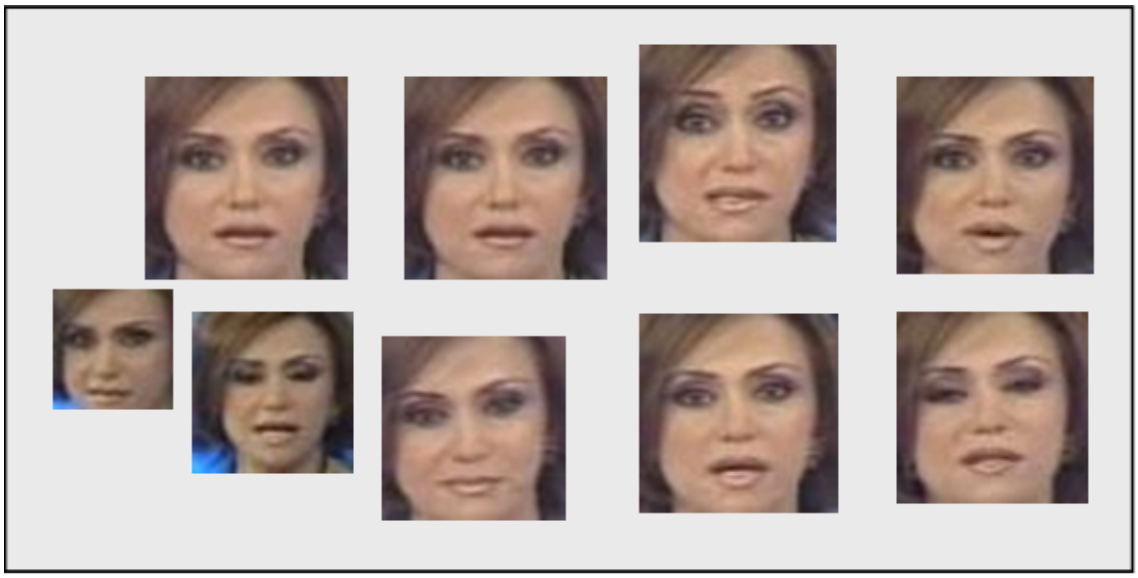


Figure 6.9: Example of a cluster delivered at the end of the clustering process on the LBC news.

(i.e. people are wearing similar clothing), etc. Even if our method can handle all those kinds of variations, the hierarchical clustering remains difficult. In part 3, we will use the audio information to correct this remaining weakness.

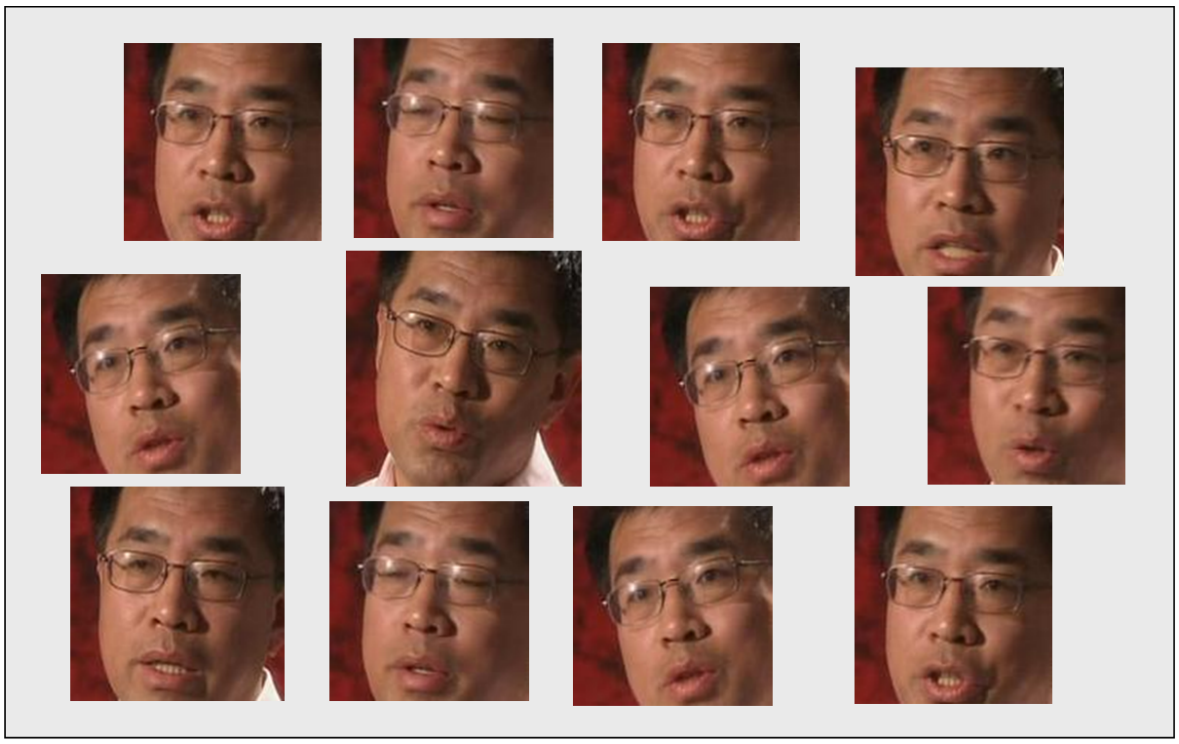


Figure 6.10: Example of a cluster delivered at the end of the clustering process on the CCTV news.

Table 6.11: Results of the different descriptors and the fusion clustering on Broadcast news: 3D-Histogram of the clothing, texture of the clothing, skin color of the face, and sift features of the face.

	SIFT	Skin	Hist	Texture	Fusion
Le Grand Journal (1)	**28.90**	33.82	41.98	45.43	**14.6**
Le Grand Journal (2)	**27.72**	34.92	50.46	54.04	**20.60**
C'est notre affaire	54.91	**40.75**	42.66	69.21	**49.75**
C'est dans l'air	**23.28**	72.23	70.65	78.56	**8.92**
Weighted average CER (%)	27.51	53.16	58.23	65.34	**15.73**



Figure 6.11: Example 1 on "le Grand Journal".

Table 6.12: Results of the different descriptors and the fusion clustering on movies: 3D-Histogram of the clothing, texture of the clothing, skin color of the face, and sift features of the face.

	SIFT	Skin	Hist	Texture	Fusion
Les Choristes	**44.48**	56.44	53.54%	55.44	**43.38**
Amelie	62.33	64.03	**52.50**	60.83	**60.37**
Asterix Obelix	48.33	48.33	59.49	**45.70**	**42.83**
Virgins Suicide	42.47	42.47	**39.73**	50.17	**30.61**
Weighted average CER (%)	44.92	53.20	52.22	53.06	**43.72**

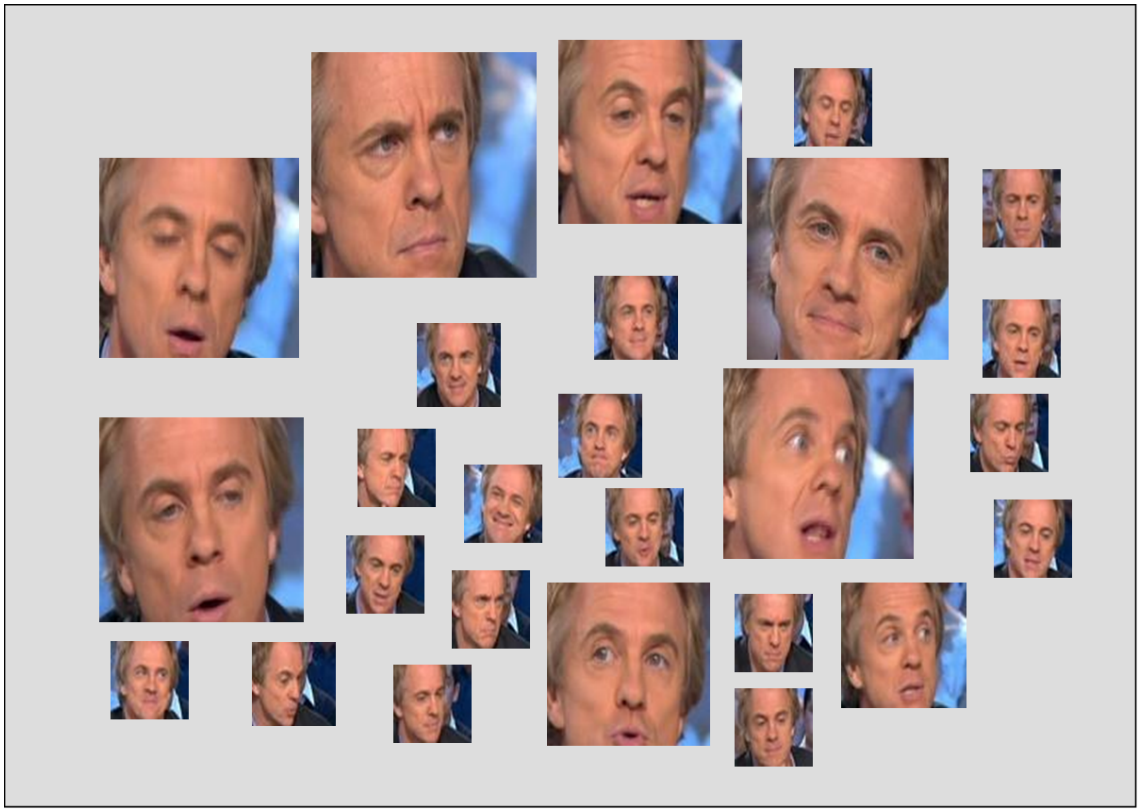


Figure 6.12: Example 2 on “Le Grand Journal”.



Figure 6.13: Example 3 on “Le Grand Journal”.



Figure 6.14: Example 4 on “Le Grand Journal”.



Figure 6.15: Example 5 on “Le Grand Journal”.

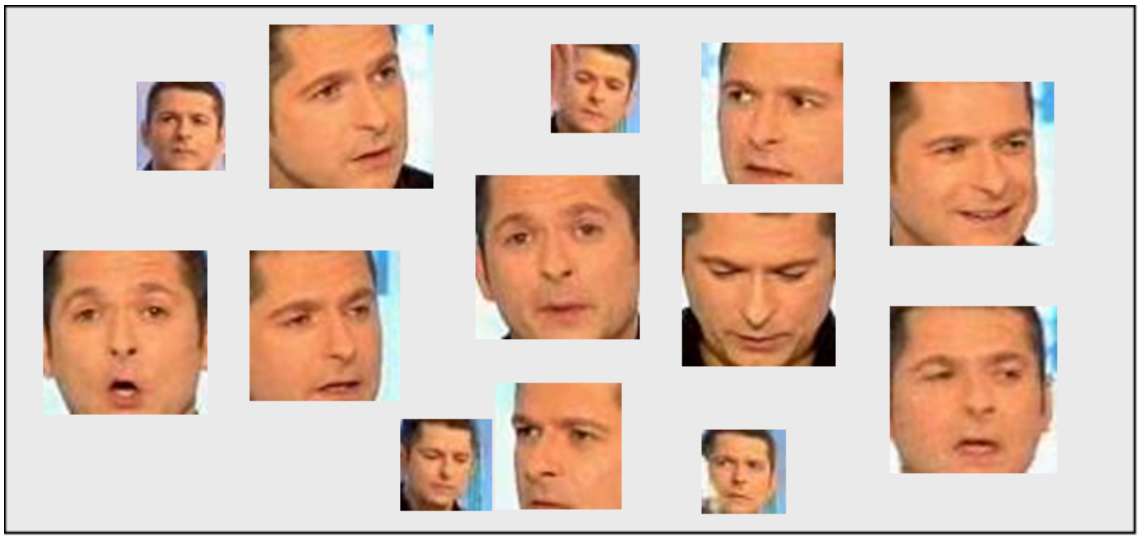


Figure 6.16: Example on "C'est dans l'air".

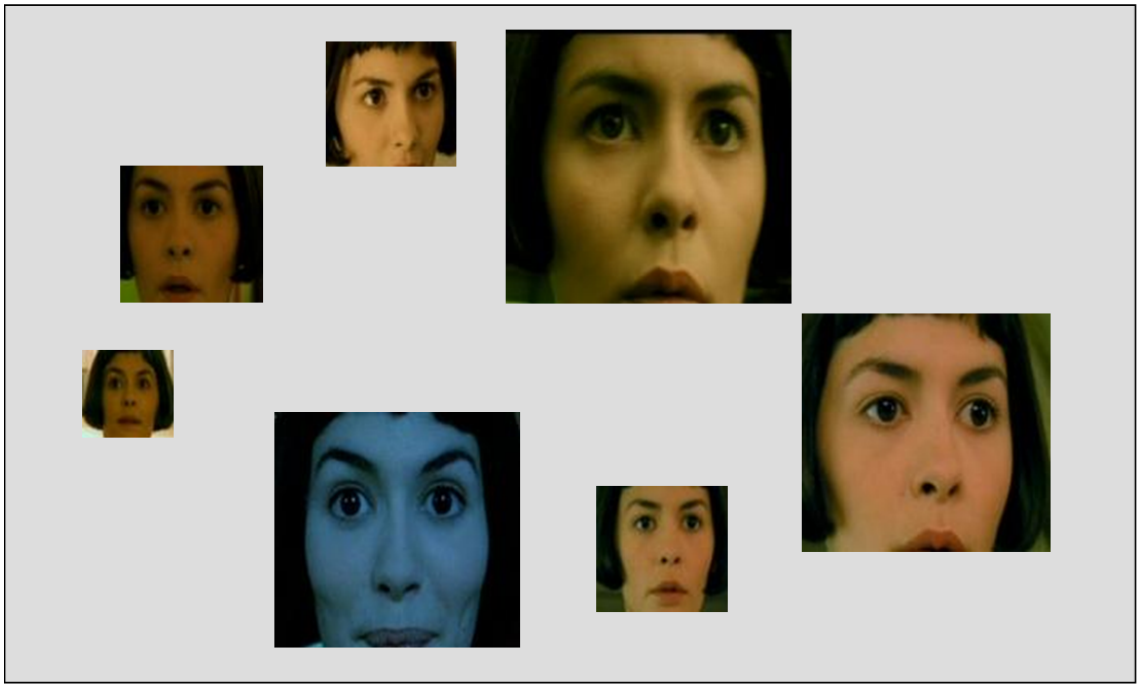


Figure 6.17: Example of the movie "Amelie".



Figure 6.18: Example 1 of the movie “Asterix et Obelix”.



Figure 6.19: Example 2 of the movie “Asterix et Obelix”.



Figure 6.20: Example 1 of the movie “Virgins suicide”.



Figure 6.21: Example 2 of the movie “Virgins suicide”.

Conclusion

After reviewing the existing visual features, and the state-of-the-art works for people detection, tracking, clustering and recognition, we present our proposed people indexing system starting from the shot boundaries detection, passing through the people detection and tracking and finally focusing on the people clustering algorithm using both face and clothing information.

For this clustering, we investigate different descriptors especially the SIFT features within the face box and the histogram color of the clothing part. An adequate matching method that outperforms the state-of-art techniques was proposed for the SIFT features. The Bhattacharyya distance was used for computing similarities between clothing parts. Then a similarity measure that combines all the face and clothing descriptors was defined. The clustering is finally processed in a hierarchical bottom-up manner.

Experiments were done on broadcast news, debates and movies that were manually annotated. The results show the impact of each technique/descriptor on the whole system as well as the efficiency and the robustness of that system.

Part III

Audiovisual fusion

Introduction

Data fusion is a formal framework in which are expressed the means and tools for the alliance of data originating from different sources. It consists in using techniques that combine data from multiple sources and gather that information in order to achieve inferences, which will be more efficient and potentially more accurate than if they were achieved by means of a single source. The word "data" in data fusion is taken in a broad sense. It may be replaced by information fusion.

In [HL01a], authors define "Data fusion" as "a process dealing with the association, correlation and combination of data and information from single and multiple sources to achieve refined position and identity estimates, and complete and timely assessments of situations and threats, and their significance." The process is characterized by continuous refinements of its estimates and assessments, and the evaluation of the need for additional sources, or modification of the process itself, to achieve improved results.

The challenge we are facing in this chapter is the fusion issue of audiovisual features. By its nature, a multimedia document contains a set of information more or less synchronized like images, sound and sometimes textual information. In previous chapters, we review state-of-the-art and proposed techniques for handling each of the audio and video media separately both extracted from an audiovisual stream. In this chapter, we are interested by techniques that allow taking into account the set of available media in order to represent and analyze a whole multimedia document. Then, more particularly, we will give special care to the problem of associating voices from the audio channel to characters from the video channel in an audiovisual document. We will then use this association to improve results of "only-audio" speaker diarization system seen in chapter 1, and "only-video" character indexing system seen in chapter 2. Hence, this association allows building an audiovisual model for each person appearing and talking along the database without *a priori* knowledge and that can be dynamically updated.

Several applications may emerge from associating voices to their corresponding faces in video sequences. Other than applications seen in part 1 (cf. section I) for an audio speaker diarization system and in part 2 (cf. section II) for character indexing, the audiovisual association can be used in many tasks such as:

- **Audiovisual automatic speech recognition (ASR).** The multimodal modelization find probably its beginning in the speech recognition. It aims to find what people are saying using both sound and lips movement. Interested readers are invited to review [PNLM04] and [LP07].

  Having information about speech turns of a specific person within a document is very helpful to recognize what that person is saying especially in challenging audiovisual conditions because the speech and visual models can be adapted on that specific person in order to improve speech recognition.

- **Audiovisual speaker recognition.** This task aims to retrieve in the video database a famous person previously defined [CRPN03a], [CRPN03b]. It is generally used to find politicians or anchors in TV video archives. This must respond to a query such as: "find sequences where president Sarkozy is appearing and talking". In most of audiovisual speaker recognition systems, positive and negative video sequences containing the person to search are given in order to train the model. Then, processing on the whole database is done, shot by shot (or sub-shots), to find whether the tested tracks fit the model or not.

  The use of our system of **voice-face** association as a preprocessing step will improve the performance of the audiovisual speaker recognition since a very precise analysis is done on the document level and clusters containing all occurrences that belong to the same person are collected within each video document. That will ensure more accuracy in the matching confidence.

- **Similarity between documents.** Two documents can be considered as similar if we find that many people occur in both of them. It is for example the case of TV broadcast debates, News and series where anchors, politicians and actors will potentially re-appear in a topic of the same gender. This will be helpful to classify programs in an unsupervised manner (cf. Appendix A).

- **Rushes and video understanding.** Having information on people that act (talk and appear) in the video, on their time-life and their number can be very helpful to understand and to find a comprehensive summarization of that video. Reader can check [BFP10], [RSJU07] and [BLSO08].

This part is organized as follows: chapter 7 reviews the fusion architectures, the mathematical aggregation operators and the existing audiovisual works. Chapter 8 presents our voice-to-face association method and our audio-visual indexing system that improves iteratively the speaker indexing system described in part 1, the face-and-clothing based people indexing presented in part 2 and also the voice-to-face association. Chapter 9 shows some experimental results.

Chapter 7

State-of-the-art

Contents

7.1 Fusion architectures 135

7.2 Mathematical aggregation operators 137

7.3 Existing works in audiovisual fusion 138

7.3.1 Audiovisual scene segmentation 139

7.3.2 Audiovisual video structuring 139

7.3.3 Audiovisual music video segmentation 140

7.3.4 Spatio-temporal detection of talking person 141

7.3.5 Audiovisual speaker recognition 141

7.3.6 Audiovisual synchronization 142

7.3.7 Audiovisual speaker diarization 143

7.3.8 Major casts list 145

This chapter reviews the fusion architectures and the mathematical aggregation operators. It also presents some of the existing works in audiovisual fusion.

7.1 Fusion architectures

There are three main fusion architectures: low-level, intermediate-level and high-level fusions;

- **Low-level fusion** combines several sources of raw data to produce new raw data. The expectation is that fused data is more informative and synthetic than the original inputs. But here two major problems are generally faced: 1) the non-balanced dimensionalities of

the data produced from the different sources and 2) the choice of the "relevant information" from those sources. In the audiovisual context, we can imagine fusion of the audio wave and the video signal raw that seems very unrealistic.

- **Intermediate-level fusion**, also called *feature level fusion*, combines various features. Those features may come from several raw data sources (for example: the MFCCs extracted from the audio wave channel and the SIFT features from the video channel) or from the same raw data (for example: MFCC and LPCC from the audio channel or SIFT and color histograms from the video channel). In the latter case, the objective is to find relevant features among available features that might come from several feature extraction methods. The purpose is to ensure the use of a limited number of relevant features.

- **High-level fusion**, also called *decision fusion* or *late fusion* combines decisions coming from several experts or different systems. By extension, it is called decision fusion even if the experts return confidence scores and not a decision. To distinguish both cases, we call them "hard" fusion (decision) and "soft" fusion (confidence scores). Methods of decision fusion include voting methods, statistical methods, fuzzy logic based methods, etc.

  One good example of "decision fusion" is the method we implement in part 2 to combine information coming from different visual descriptors (SIFT features, histograms, skin color and texture features) in order to find the most efficient way to cluster tracks that correspond to the same appearing character under different lightening, occlusions and clothing changes conditions.

The above categorization does not encompass all possible fusion paradigms, as input and output of the fusion process may present different levels of processing. Dasarathy [Das94] expands on the ideas of low and high level fusion by putting forward an I/O-based characterization. The five I/O modes are shown in Fig. 7.1:

1. the first fusion takes data as input and delivers data on output (similar to "low level fusion" in the above categorization),

2. the second fusion takes data as input and delivers features on output,

3. the third fusion takes features as input and delivers features on output,

4. the fourth fusion takes features as input and delivers measures (e.g. distance, similarity, likelihood) on output in order to make decision,

5. the fifth fusion combines decisions taken as input to deliver a decision on output (similar to "high-level fusion" in the above categorization).

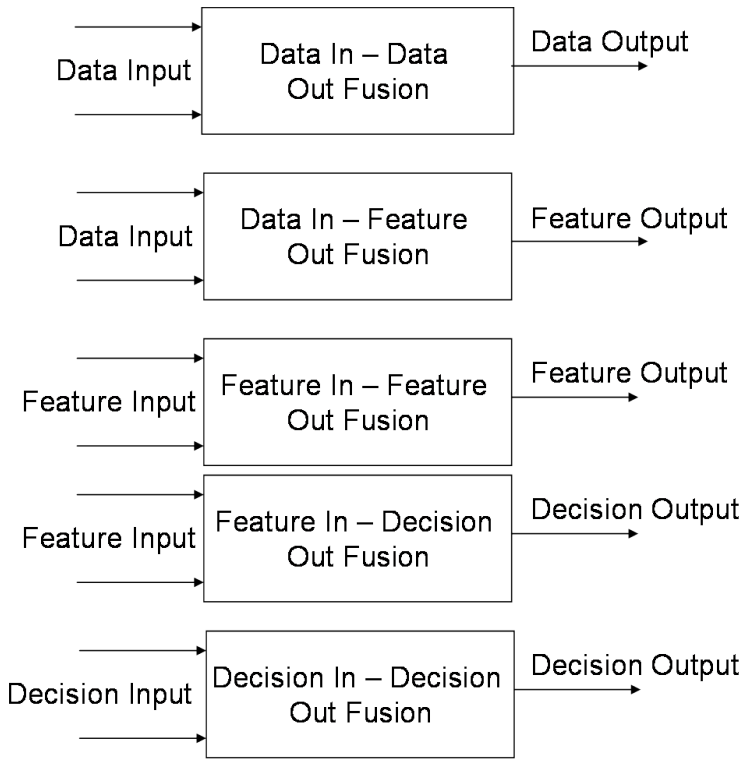


Figure 7.1: Different types of fusion defined by Dasarthy [Das94].

7.2 Mathematical aggregation operators

The aggregation operators are mathematical objects that aim to reduce a set of numbers into a unique representative number. That is why those objects are used to resolve the problem of data fusion. Detyniecki [Det00] listed those mathematical operators: the arithmetic mean, the weighted mean, the median, the quasi-arithmetic mean, the symmetric sum, the ordered weighted averaging operators, the Choquet-Sugeno discrete Fuzzy integrals, the bayesian fusion approach, the possibilistic fusion approaches, the T-norms, the T-conorms, the compensatory operators and the uninorms.

We are especially interested in the T-norms and the T-conorms since they generalize respectively the conjunctive "AND" operator and the disjunctive "OR" operator.

The concept of a triangular norm (t-norm) and its dual (t-conorm) was introduced in [Men42] and [BS60] in order to generalize the triangular inequality of a metric. A t-norm is a function $T : [0,1] \times [0,1] -> [0,1]$ that is commutative, monotone (increasing), associative, and having 1 as a neutral element. Formally, a t-conorm is also a function $T : [0,1] \times [0,1] -> [0,1]$ that is commutative, monotone (increasing), associative, and having 0 as a neutral element.

The following table 7.1 lists the more common t-norms and their dual t-conorms. Their definitions are given for two elements but they can be simply generalized to n elements since these operators are associative. We should notice that x and $y \in [0,1]$ in those examples.

Table 7.1: Common t-norms and their dual t-conorms [Det00].

	T-norm	T-conorm
Min-Max	$min(x,y)$	$max(x,y)$
Probabilistic	$x.y$	$x+y-x.y$
Lukasiewicz	$max(x+y-1,0)$	$min(x+y,1)$

For more details about the properties of these operators, please refer to [KMP00].

7.3 Existing works in audiovisual fusion

The goal of the multimodal description is to use the different representations of a document in order to extract reliable information about its content. The difficulty of this task is due to two main factors. On the first hand, the data to model are often heterogeneous (color histograms, SIFT features, presence of the face, size of the face, etc...) and correspond to different levels of granularity. On the other hand, there is the problem of streams synchronization due to the fact that the extractions of low-level features are not generally done on the same timestamps.

Because manual annotation is time consuming and sometimes inconsistent, many research efforts have been involved to automate the procedure of video indexing. Even though most of the existing works were especially focusing on one type of modality, there are some exceptions.

7.3.1 Audiovisual scene segmentation

Sundaram and Chang [SC00b] model a scene as a semantically consistent chunk of audio-visual data. Central to the segmentation framework is the idea of a finite-memory model. The audio and video data are separately segmented into scenes, using data in the memory. The audio segmentation uses the correlations amongst the envelopes of audio features. The video segmentation uses the correlations amongst shot keyframes. The fusion of the resulting segments is done using a nearest neighbor algorithm that is further refined using a time-alignment distribution derived from the ground truth.

Saraceno and Leonardi [SL98] considered segmenting a video into the following basic scene types: dialogues, stories, actions, and generic. This is accomplished by first dividing a video into audio and visual shots independently, and then grouping video shots using Learning Vector Quantization approach, so that audio and visual characteristics within each group follow some predefined patterns.

Boreczky [BW98] used a hidden Markov model (HMM) framework for video segmentation using both audio and image features: Video is segmented into regions defined by shots, shot boundaries, and camera movement within shots. Features for segmentation include an image-based distance between adjacent video frames, an audio distance (GLR) based on the acoustic difference in intervals just before and after the frames, and an estimate of motion between the two frames.

Lienhart et al. [LPE99] proposed a method to segment a video into scenes with similar audio characteristics and approaches combining multiple modalities in video content scenes with similar settings, and dialogues. The scheme considers audio features, color features, orientation features, and face information.

In [KKP07], an enhanced set of eigen-audioframes is created that is related to an audio signal subspace, where audio background changes are easily discovered, then an additional process is used to detect audio scene change candidates in this subspace. Visual information is used to align audio scene change indications with neighboring video shot changes.

7.3.2 Audiovisual video structuring

The video structure parsing relies on the analysis of the temporal interleaving of video sequences, with respect to a priori information about video and audio content and editing rules.

In [KGG+03], authors use the Hidden Markov Models (HMMs) to merge audio and visual cues. This structuring method is applied on tennis videos in order to identify typical tennis scenes. In their work, the basic temporal unit used is the video shot. Visual features are used to characterize the type of shot view. Audio features describe the audio events within the video shot.

Another type of video structuring can be found in [HJC06] where authors are focusing on computing the similarities between videos using audiovisual production invariants (APIs). Those APIs are characterized by invariant segments obtained on a set of low-level features.

Readers are also invited to review our additional work done for unsupervised TV program boundaries detection based on audiovisual features [EKSJ08].

7.3.3 Audiovisual music video segmentation

In [GER07], a study on the correlation of automatic audio and visual segmentation of music videos is done. A two-level structuring of the music and the video is achieved separately. Note onsets are detected from the audio signal, along with section changes. The visual signal is segmented to detect changes in motion activity, as well as shot boundaries. Based on this two-level segmentation of both streams, four audio-visual measures are computed using either Pearson's correlation or mutual information. Assuming that $a(m)$ and $b(m)$ two sequences of independent realizations of random variables A and B:

- Pearson's correlation is defined as:

$$\rho(A,B) = \frac{\mathrm{E}\left[(A - \mathrm{E}\left[A\right])(B - \mathrm{E}(B))\right]}{\sqrt{\mathrm{E}\left[(A - \mathrm{E}\left[A\right])^2\right]\mathrm{E}\left[(B - \mathrm{E}\left[B\right])^2\right]}} \tag{7.1}$$

  where E is the statistical expectation.

- the mutual information for the discrete case is defined as:

$$I(A,B) = \sum_a \sum_b P(A,B) \log \frac{P(A,B)}{P(A)P(B)} \tag{7.2}$$

Thus four audiovisual correlation measures are deduced:

$$C_{onsets/shots} = \rho(d_o, d_s)$$

$$C_{sections/shots} = \rho(d_c, d_s)$$

$$C_{onsets/motion} = I(d_o, d_m)$$

$$C_{sections/motion} = \rho(d_c, d_m)$$

where d_o denotes the note onset detection function, d_c the detrended section change detection function, d_m the motion activity changes function, and d_s the shot boundary detection function.

7.3.4 Spatio-temporal detection of talking person

In [CD00], a method of detecting a talking person (both spatially and temporally) using video and audio data from a single microphone is described. The audiovisual correlation is learned using a time-delayed neural networks, which is then used to perform a spatio-temporal search for a speaking person.

In [BJA02], authors presented a self-calibrated algorithm for audio-visual tracking using two microphones and a camera. This algorithm uses a parameterized statistical model which combines simple models of video and audio. Those models are estimated using the EM algorithm.

In [MJV+07], authors proposed a method for learning fundamental multimodal patterns by defining a model of multimodal signals based on their sparse decomposition over a dictionary of multimodal structures. This algorithm is applied to audiovisual sequences and is tested on audiovisual speaker localization.

7.3.5 Audiovisual speaker recognition

In [FLL+03] and [NLFL03] authors investigate the use of coupled hidden Markov models (CHMM) for the task of audio-visual text dependent speaker identification. The use of CHMM is justified by the capacity of this model to describe the natural audio and visual state asynchrony as well as their conditional dependency over time. Their system determines the identity of the user from a temporal sequence of audio and visual observations obtained from the acoustic speech and the shape of the mouth. The multimodal observation sequences are then modeled using a set of CHMMs, one for each phoneme-viseme pair and for each person in the database.

In [LNK03], [LNK04a], [LNK04b], an adaptive speaker identification system which employs both audio and video cues is proposed for movie content analysis. First, a likelihood-based approach is applied for speaker identification using pure speech data, and then face detection/recognition and mouth tracking are applied for talking face recognition using pure visual data. These two information cues are then effectively integrated under probabilistic framework

for achieving more robust results. Moreover, an update of the speaker acoustic model is done by adapting on the fly to their incoming speech data.

Fisher and Darrell [FD04] proposed a speaker association method with signal-level (low-level) audiovisual fusion technique. A probabilistic multimodal generation model is used to derive an information theoretic measure of cross-modal correspondence. By comparing the mutual information between different pairs of signals, authors automatically identified which person is speaking a given utterance and discount errant motion or audio from other utterances or non-speech events.

In [ATD05], authors proposed a face recognition system using a combined audiovisual approach. First, audio and video information are used independently to obtain confidence values that indicate the likelihood that a specific person occurs in a video shot. Then, a post-classifier is applied to fuse audio and visual confidence values. The advantage of a post-classifier approach choice is that it is possible to combine confidence values from different experts, even if their output falls in different ranges.

A method for multimodal person authentication is presented in [Pal08]. This method uses speech, face and visual speech modalities. First, the motion information is used to localize the face region. This face region is then processed in YC_rC_b color space to determine the eyes location. Facial and visual speech features are extracted using multi-scale morphological erosion and dilation operations. Acoustic features used are the Weighted Linear Prediction Cepstral Coefficients (WLPCC). Auto-associative neural network (AANN) models are used to capture the distribution of the extracted acoustic, facial and visual speech features. The evidences from those modalities are combined using a weighting rule.

7.3.6 Audiovisual synchronization

Bredin and Chollet [BC07] overview transformations that can be applied on audiovisual spaces with the aim of improving subsequent measure of correspondence between audio and visual clues:

1. Principal component analysis (PCA) is a well-known linear transformation which is optimal for keeping the subspace that has the largest variance. It was used in [CMD97] for speaker identification problem where an assessment of feature fusion-based on audiovisual feature vector concatenation was done. Also, it was used in our additional work to deal with

the issue of unsupervised program boundaries detection. Interested readers can review Appendix A.

2. Independent component analysis (ICA) was originally introduced to deal with the problem of source separation. It was applied in [SC03] to find an association between the audio and visual note in an audiovisual recording of a piano session.

3. Canonical correlation analysis (CANCOR) is a multivariate statistical analysis allowing to jointly transform the audio and visual spaces while maximizing their correlation in the resulting transformed audio and visual feature spaces [SC00a].

4. Co-inertia analysis (CoIA) is similar to CANCOR. The main difference is that CANCOR relies on the maximization of the correlation between audio and visual features, and CoIA is based on the maximization of the covariance. CoIA was used in [EB05] for speaker independent "liveness" verification method for audiovisual identification system.

In their paper [BC07], authors used the audiovisual synchronization to deal with the issue of impersonation scenarios in an identity verification system. Impersonation means that an impostor can use the voice recordings or the picture of the face that belong to the authorized person.

Kumar et al. [KNM+09] studied the problem of detecting audiovisual synchronization in video segments containing a speaker in frontal head pose. They proposed a time-evolution bimodal linear prediction model for AV features to capture the linear dependence between them, and then derived an analytical approach to capture the notion of synchronization between them. Finally, they use CANCOR to reduce the AV features dimensionality.

7.3.7 Audiovisual speaker diarization

In [TP98], authors proposed a method that aims to temporally index the video sequence according to the actual speaker. Audio analysis leads to the extraction of a speaker identity label versus time diagram. Visual analysis includes scene cut detection, face detection, mouth region extraction and tracking, and talking face detection. A combination of the labeling time diagrams obtained by audio and visual processing is achieved using simple decision rules. Boundaries of a face shot ensure the existence of a person. Mouth movement detection in this shot implies that this person speaks. Non-face shot durations cannot be used for speaker detection, since interchangeability between speakers cannot be detected by the visual information. Consequently,

if the speaker-dependent indexing achieved by the audio processing module, a refinement is performed in face shots with talking faces. This refinement involves estimation of speaker S_i presence likelihoods in every face shot F_k:

$$P(S_i|F_k) = \frac{M_i}{L_k} \tag{7.3}$$

where M_i is the number of the speech frames assigned to speaker S_i and L_k is the total number of speech frames in face shot F_k. The speaker that exhibits the maximum presence likelihood is the winner.

Liao and Syu [LS08] proposed an actor-based video segmentation system using visual and audio information in E-learning by assuming some segmentation rules. A classroom scene can be classified as following three types: teacher-blackboard, teacher-students and students.

In [HF08], authors implement a real-time and online audiovisual diarization system for group meetings. Rather than labeling the speaker regions with numbers as traditional speaker diarization systems do, they are associated with video segments of the corresponding participant. Audio features used are the well-known MFCC features. However, for visual module, authors estimate the visual activity of each person by computing the residual coding bit-rate (in MPEG-4 format) that was found [YR08] to be the most discriminative for speech-visual activity association. Finally, the distance between video (v_i) and audio (a_j) streams is quantified using the pair-wise correlation formula:

$$\rho(v_i, a_j) = \frac{\sum_{t=0}^{T} v_i(t).a_j(t)}{\sum_{t=0}^{T} v_i(t) \sum_{t=0}^{T} a_j(t)}, \forall\{i, j\} \tag{7.4}$$

where T is the total length of the meeting and t indexes the feature value at each frame which rate was 5 frames/second.

In [TPC09], authors proposed a system for detecting the active speaker in cluttered and reverberant environments where more than one person speaks and moves. It uses audiovisual information from multiple acoustic and video sensors that feed separate audio and video tracking modules. The audio module tracker is based on a particle filter. The video module tracker is based on a variation of Stauffer's adaptive background algorithm [SG00] with spatio-temporal adaptation of the learning parameters and a Kalman tracker in a feedback configuration. Finally, the association between audio and video is done by selecting the minimum Euclidean distance between the active speaker location provided by the audio tracker and every location of every person in the room provided by the video tracker.

Name-It [SNK99] is a project aiming at automatically associating faces detected from video frames and names extracted from closed captions for news video. One novelty of the system is that it does not rely on any pre-stored face templates for selected names. However, the system associates faces and names by integrating face sequence extraction and similarity evaluation, name extraction, and video caption recognition based on their temporal correlation. Besides the difficulties in detecting faces and names, the association of them also poses a challenge since multiple faces may appear in one frame and multiple names may be mentioned in one closed caption sentence.

7.3.8 Major casts list

The work the most similar to our topic is the work done by Liu and Wang [LW01], [LW07] for detecting the major casts in video. Major casts, for example the anchor persons or reporters in news programs and principal characters in movies play an important role, and their occurrences provide good indices for organizing and presenting video content. The users may easily digest the main scheme of a video by skimming through clips associated with major casts.

In a certain sense, the goal is similar to that in [SNK99]. The difference is that Liu and Wang associate sound and face to each cast, instead of name and face. In their work, they assume that the majority of speech that accompanies the appearances of each character is from the same person. For example, they cannot handle the case where a person appears in silence or is mostly accompanied with other person's speech.

Authors found a way to associate speakers to faces using correlation matrix where each coefficient $C(i,j)$ is computed as follows:

$$C(i,j) = \sum_{m=1}^{L_i} \sum_{n=1}^{l_j} OL(s_m^i, f_n^j).FS(f_n^j) \tag{7.5}$$

where $OL(s_m^i, f_n^j)$ is the overlapping time of speaker sub-segment s_m^i and face sub-track f_n^j, and $FS(f_n^j)$ is the face size of f_n^j. L_i is the number of segments corresponding to the speaker i, and l_j the number of tracks corresponding to the face j.

This definition considers not only the temporal overlapping among speaker segments and face tracks, but also takes into account the importance of face size. The consideration of face size is helpful when more than one face show up during a speech segment, where the face with bigger size is more likely to be the real speaker.

Moreover, authors introduced an integrated Speaker-Face clustering. In fact, instead of clustering the speaker segments and the face tracks independently, performing such clustering jointly will help to improve the performance. They define an augmented distance matrices based on not only the distance among speaker segments (respectively face tracks), but also distances among corresponding face tracks (respectively speaker segments).

$$D_S^{'}(i,j) = \lambda_f \frac{\sum_{1 \leq m,n \leq N} C(i,m)C(j,n)D_F(m,n) + T_f\varepsilon}{\sum_{1 \leq m,n \leq N} C(i,m)C(j,n) + \varepsilon} + D_S(i,j) \tag{7.6}$$

with $l \leq i,j \leq M$

$$D_F^{'}(i,j) = \lambda_s \frac{\sum_{1 \leq m,n \leq M} C(m,i)C(n,j)D_S(m,n) + T_s\varepsilon}{\sum_{1 \leq m,n \leq M} C(m,i)C(n,j) + \varepsilon} + D_F(i,j) \tag{7.7}$$

with $l \leq i,j \leq N$

where $D_S(i,j)$ (respectively $D_F(i,j)$) is the initial distance between speaker segments (respectively face tracks) i and j, $C(i,j)$ is the correlation coefficient between i and j (cf. equation 7.5), N (respectively M) is the total number of speakers (respectively the total number of faces), T_s and T_f are two thresholds that are used in speaker segments/face tracks independent clustering, λ_s and λ_f are ratios that determine the weighting of distance effect from different modalities and ε is a small constant to prevent division by zero.

For more details about the integrated clustering procedure, please refer to the papers [LW01] and [LW07].

Chapter 8

Proposed audiovisual fusion methods

Contents

8.1 Association between audio and video indexes 147

8.1.1 Automatic matching using weighted co-occurrence matrix 148

8.1.2 The use of the face size 152

8.1.3 Lips activity detector 154

8.2 Audiovisual system for people indexing 157

In this chapter, we describe our contribution that aims to associate the audio and the video indexes. Then, we propose an automatic iterative audiovisual system that enables improvements of not only both audio indexing system and video indexing system, but also their association.

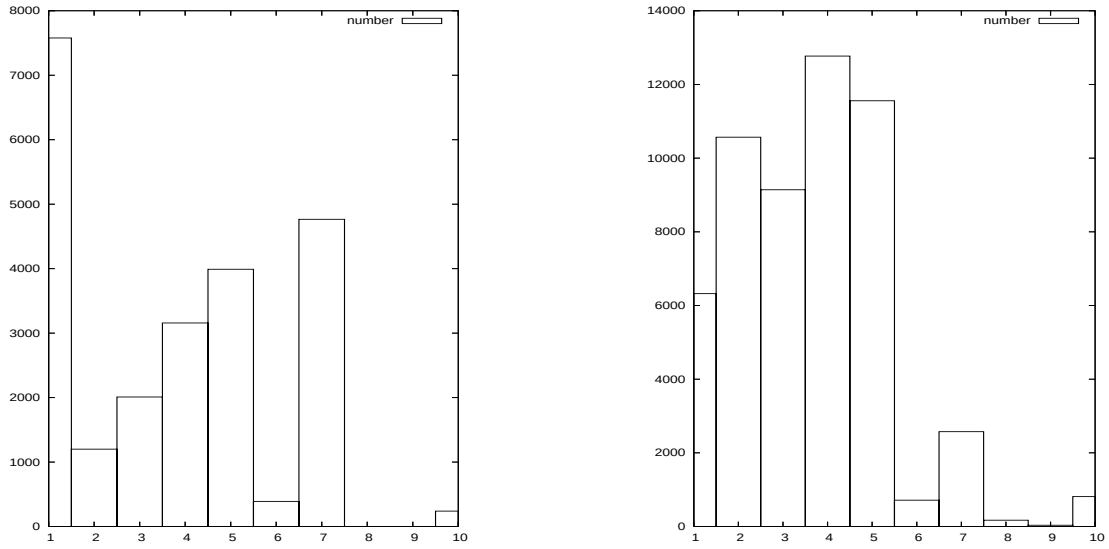
8.1 Association between audio and video indexes

As seen in previous chapter, existing audiovisual people association methods like in [LW07] consider that both visual and speech features are simultaneously relevant in video subsequences and assume that the current voice corresponds to a face in the frame.

In real sequences, this hypothesis is often violated. It is very common to find sequences where appearing people do not talk during many frames or many shots. Moreover, it is also possible that the current voice belongs to a person whose face is not in the current frame.

Figure 8.1 presents the appearance durations (in terms of number of frames) of the ten main persons in a TV talk show, for both audio and video streams. We can see that these probability distributions are quite different for ground-truth audio and video indexes. For example, person

1 is the most talking cast with almost 7600 frames, however he appears during only 6200 frames. On the other side, Person 4 is the most appearing cast with almost 12500 frames, however he talks only along 3200 frames. If the previous assumptions were verified, the number of occurrences of each person would be similar in the two indexes.



audio index video index

Figure 8.1: Number of frames for each character appearance, on a TV talk show.

In this work, we propose to compute co-occurrences between audio and video indexes, i.e. we match up the voices with the faces. This approach is suitable to handle the cases where the usual assumptions are not verified.

The scale of audio and video indexes are different: an audio frame is typically extracted every $10ms$. However, a video frame is generally extracted every $40ms$ (25 images/sec). Thus, a direct comparison of the two indexes is not possible. That is why we use a common scale for audio and video indexes in order to be able to directly associate them.

8.1.1 Automatic matching using weighted co-occurrence matrix

The use of co-occurrence matrix was firstly introduced in [Jaf05] and applied in our work presented in [EKJPS07]. Jaffré et al. proposed a framework to automatically realize the association between voices and their corresponding appearing characters characterized only by their clothing,

using a statistical analysis of audio and video indexes. In this section, we describe this method and then, we present our own contribution in paragraphs 8.1.2 and 8.1.3.

First to validate our matching method, we make the assumption that the two streams are perfectly segmented, i.e. they are not over-segmented and correspond to the ground truth. So, each person has only one voice and exactly one visual entity (face). Moreover, each voice is associated to exactly one face, and conversely. However, in section 8.2, we will show a "real" framework to deal with over-segmentation.

8.1.1.1 Index intersection

First, we compute a matrix which represents the intersection between audio and video indexes. We use the following notations:

- n_a is the number of different voices in the audio index,
- n_v is the number of different appearing persons in the video index,
- $\{A_i\}_{i=1...n_a}$ is the set of voices of all persons,
- $\{V_j\}_{j=1...n_v}$ is the set of visual features of all persons.

To compute this intersection matrix, we go through the two indexes, frame by frame. For each frame, if the voice A_i is heard and the visual person V_j is present, the number of occurrences m_{ij} of the pair (A_i, V_j) is incremented. Then, we obtain the following matrix:

$$m = \begin{array}{c} \\ A_1 \\ A_2 \\ \vdots \\ A_{n_a} \end{array} \begin{array}{c} \begin{array}{cccc} V_1 & V_2 & \dots & V_{n_v} \end{array} \\ \begin{pmatrix} m_{11} & m_{12} & \dots & m_{1n_v} \\ m_{21} & m_{22} & \dots & m_{2n_v} \\ \vdots & \vdots & \vdots & \vdots \\ m_{n_a1} & m_{n_a2} & \dots & m_{n_an_v} \end{pmatrix} \end{array} \tag{8.1}$$

In this matrix, the value m_{ij} means that in all the frames where the voice A_i is heard, the visual person V_j appears m_{ij} times. Conversely, in all the frames where the person V_j is present, the voice A_i is heard m_{ij} times.

An intuitive idea would be to sort this matrix by rows (or by columns). However, this solution is often wrong, because it makes the assumption that while a voice is heard, its corresponding face is the most present in the frames (sorting by rows). Sorting by columns would mean that for each face its corresponding voice is the most heard while the feature appears.

In real TV shows, this assumption is often wrong as seen in Fig. 8.1. For instance, in French TV games, the person who speaks the most is usually the presenter. In this case, his voice can be the most heard even when the players appear on screen. Thus, even if this intersection matrix is interesting for the fusion of audio and video, it cannot be directly used. A post-processing is required: it will be presented in the next section.

8.1.1.2 Index fusion

With some special contents, like TV talk shows and TV games, the matrix m can be directly read if we have some prior information about the persons. For example, in a TV talk show, if a person is assumed to be the presenter, his voice is the most heard when he appears on screen (which is wrong for a guest). Conversely, if the person is assumed to be a guest, his face is the most seen while he is speaking (which is wrong for a presenter).

With real data, when there is no training stage, we cannot have prior information about this "class" of persons, which makes direct reading of the matrix m impossible because we cannot determine for each person if the matrix must be sorted by rows or by columns. Thus, one way to bypass the problem is to read m both by rows and by columns, and to keep the most significant information.

This fusion is carried out by computing two new matrices, m_a and m_v, where the frame numbers are replaced with percentage by rows and by columns:

$$m_a = \begin{array}{c} \\ A_1 \\ A_2 \\ \vdots \\ A_{n_a} \end{array} \begin{array}{c} \begin{array}{cccc} V_1 & V_2 & \dots & V_{n_v} \end{array} \\ \begin{pmatrix} f^a_{11} & f^a_{12} & \dots & f^a_{1n_v} \\ \boxed{\begin{matrix} f^a_{21} & f^a_{22} & \dots & f^a_{2n_v} \end{matrix}} \\ \dots & \dots & \dots & \dots \\ f^a_{n_a1} & f^a_{n_a2} & \dots & f^a_{n_an_v} \end{pmatrix} \end{array} 100\,\% \tag{8.2}$$

$$m_v = \begin{array}{c} \\ A_1 \\ A_2 \\ \vdots \\ A_{n_a} \end{array} \begin{array}{c} \begin{array}{cccc} V_1 & V_2 & \dots & V_{n_v} \end{array} \\ \begin{pmatrix} f^v_{11} & \boxed{f^v_{12}} & \dots & f^v_{1n_v} \\ f^v_{21} & \boxed{f^v_{22}} & \dots & f^v_{2n_v} \\ \dots & \boxed{\dots} & \dots & \dots \\ f^v_{n_a1} & \boxed{f^v_{n_a2}} & \dots & f^v_{n_an_v} \end{pmatrix} \\ 100\,\% \end{array} \tag{8.3}$$

Matrix m_a gives the probability density of each voice A_i, whereas m_v gives the one of each visual feature V_j. From these matrices, we define the fusion matrix F, by computing, for each pair (i, j), a fusion between f_{ij}^a and f_{ij}^v with a fusion operator of conjonction nature ("AND") like the minimum or the probabilistic operator. In this work, we use the probabilistic operator (product operator).

If we note $C(A_i, V_j)$ the fusion coefficient between A_i and V_j, the expression of matrix F is given by:

$$F = \begin{pmatrix} C(A_1, V_1) & \dots & C(A_1, V_{n_v}) \\ C(A_2, V_1) & \dots & C(A_2, V_{n_v}) \\ \dots & \dots & \dots \\ C(A_{n_a}, V_1) & \dots & C(A_{n_a}, V_{n_v}) \end{pmatrix} \tag{8.4}$$

where $C(A_i, V_j) = f_{ij}^a \times f_{ij}^v$.

This matrix F can be directly used to realize the association: we look for the maximum value in this matrix (line i, column j). Thus, the voice i is associated with the costume j. Then, we delete this row and this column, and we repeat this search, to obtain another association, until having an empty matrix. At the end of this process, we obtain a list of all the persons like in the example illustrated in Fig.8.2: person P1 is characterized by the voice A1 and the face V1, person P2 is characterized by the voice A2 and the face V3, P3 is characterized by the voice A3 and there is no face associated, P4 is characterized by the face V2 and there is no voice associated, etc. Three main types of errors occur using the above assumptions for the

P1 → (A1,V1)
P2 → (A2,V3)
P3 → (A3,Ø)
P4 → (Ø,V2)
… → …

Figure 8.2: A list containing the output of the association process.

audiovisual association:

- if two or more persons are appearing almost at the same time (i.e. in the same frames), the choice is not significant. This corresponds in the matrix to the following scenario: in the same row, we obtain two or more similar frequencies. To solve this problem we will use information on the face size (cf. section 8.1.2).

- if a person appears almost with the same voice heard but this person is not talking, we should not allow the association between face and voice as explained later in section 8.1.3. Thus, there will be three types of results: talking faces, only faces and only voices.

- if there are over-segmentation errors in the audio processing (respectively video processing), this will propagate along the association. One way to correct this is to use the mutual audiovisual information to help the mono-media clustering process. This will be developed in section 8.2.

8.1.2 The use of the face size

One case that may occur in some scenarios such as TV shows is that two or more people appear almost in the same frames. In this case an additional cue that takes into account the face size of each person is very helpful prior to decide which person is talking: as said in [LW07], the person with a relatively big face is more likely to be the real speaker. A typical example that occurs in debates is shown in figure 8.3 where the talking person and as well as the people from the audience appear in the same shot: it is clear that the size of the talking face is greater than those of that appear in the background.



Figure 8.3: Talking person appearing with the audience in a TV debate.

Unlike in [LW07] where authors used the real value of the face size as a factor that is multiplied by a corresponding distance, we use a weight value w_k (between 0 and 1) that is

derived from the size of the face f_k. This weight takes into account the other faces that appear in the same image:

$$w_i = \frac{size(f_k)}{\sum_{l=1}^{N} size(f_l)} \tag{8.5}$$

where N is the total number of faces within the image.

Figure 8.4 shows an example of three faces appearing in the same image as well as their corresponding weights.

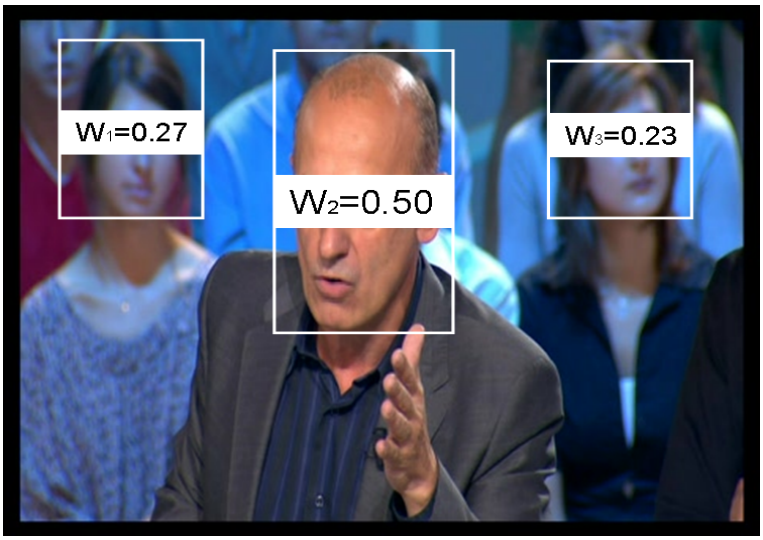


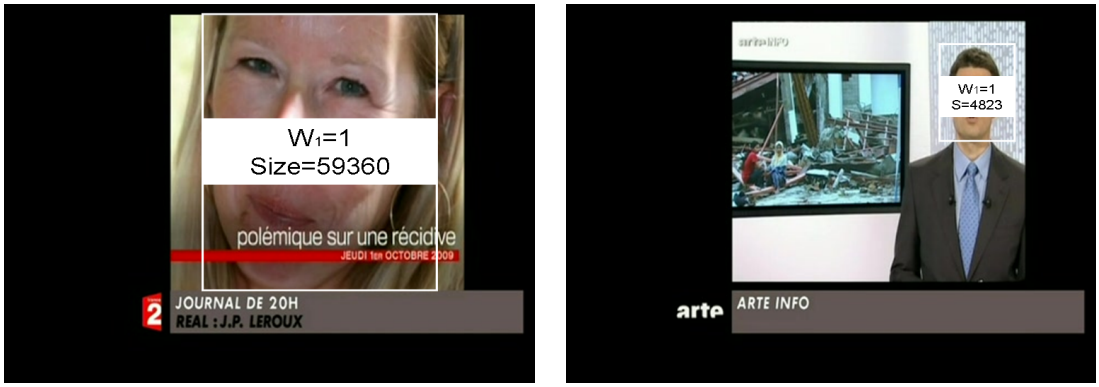
Figure 8.4: Three faces detected with their corresponding weights.

The choice of weight value rather than size value is justified by the fact that there should be no priorities in the clustering process between two faces that have different sizes and where each of them appears alone in its track as seen in Fig.8.5: despite the difference in face sizes, the person appearing in (b) is talking unlike the one appearing in (a).

Thus, the average weight $W_{A_iV_j}$ on video frames that contain commonly audio turns of A_i and video tracks of V_j is:

$$W_{A_iV_j} = \frac{\sum_{k=1}^{N_{A_iV_j}} w_k}{N_{A_iV_j}} \tag{8.6}$$

where $N_{A_iV_j}$ is the number of all images that correspond to the common time between A_i and V_j.



(a) (b)

Figure 8.5: Two faces with different sizes but similar weights.

If we consider shots, $W_{A_iV_j}$ becomes:

$$W_{A_iV_j} = \frac{\sum_{s=1}^{N_{AllShots}} \sum_{k}^{N_s} w_k}{\sum_{s=1}^{N_{AllShots}} N_s} \tag{8.7}$$

where $N_{AllShots}$ is the total number of shots that contain A_i and V_j, and N_s is the number of frames within the shot s. Instead of computing the weights on each frame which is time consuming, we assume that the weights are constant for every shot. Thus, weights are computed only on keyframes, and the expression of $W_{A_iV_j}$ can be simplified:

$$W_{A_iV_j} = \frac{\sum_{s=1}^{N_{AllShots}} N_s w_k}{\sum_{s=1}^{N_{AllShots}} N_s} \tag{8.8}$$

Then the weighted cooccurrence matrix F' becomes:

$$F' = \begin{pmatrix} W_{11}.C(A_1,V_1) & \dots & W_{1n_v}.C(A_1,V_{n_v}) \\ W_{21}.C(A_2,V_1) & \dots & W_{2n_v}.C(A_2,V_{n_v}) \\ \dots & \dots & \dots \\ W_{n_a1}.C(A_{n_a},V_1) & \dots & W_{n_an_v}.C(A_{n_a},V_{n_v}) \end{pmatrix} \tag{8.9}$$

8.1.3 Lips activity detector

An additional constraint may be added to deal with the case where a person appears when another one is talking. In this case, and in order to eliminate any confusion, it will be better

to find an indicator of the lips activity. Even though many works were done in the literature to detect the activity of the lips, they were generally specified for faces with high resolution to deal with the problem of audiovisual speech recognition. In this work, and given the range of face sizes in our data, we suggest an easiest way to estimate the lips activity:

- first, we select the mouth region using some geometrical constraint related to the size of the face box by assuming that the face is frontal. Fig.8.6 illustrates those constraints: the mouth is located in the middle-bottom of the face box [PPJ06].

- second, and in order to quantify lips activity, we proceed by pairs of frames to obtain a global result. Considering two successive frames f_1 and f_2 containing the face of a same person, and after mouth localization in the frame f_1 represented by regions M_1, we build a searching zone around M_1 in frame f_2 and we move a window M_2 of the same size of M_1 in this zone. The matching and the value representing the difference between M_1 and M_2 pixels are both obtained by the Minimal Mean Square Error (MMSE) on the luminance channel of the HSV color space.

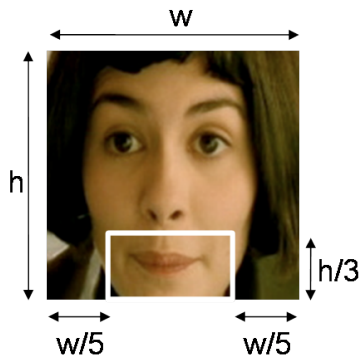


Figure 8.6: Mouth localization using geometrical constraint.

Fig.8.7 shows a curve of the lips activity for about 1900 consecutive frames: silence regions are characterized by low values, however, lips and motions are characterized by higher values. Since head motions are generally related to talking expression, we take the assumption that somebody who is moving his lips or head is typically talking. Thus, low activity values correspond surely to non-talking faces.

Moreover, to be more precise, instead of computing the lips activity on the whole frames where the face appear, we compute it on talking faces i.e. common time between A_i and V_j.

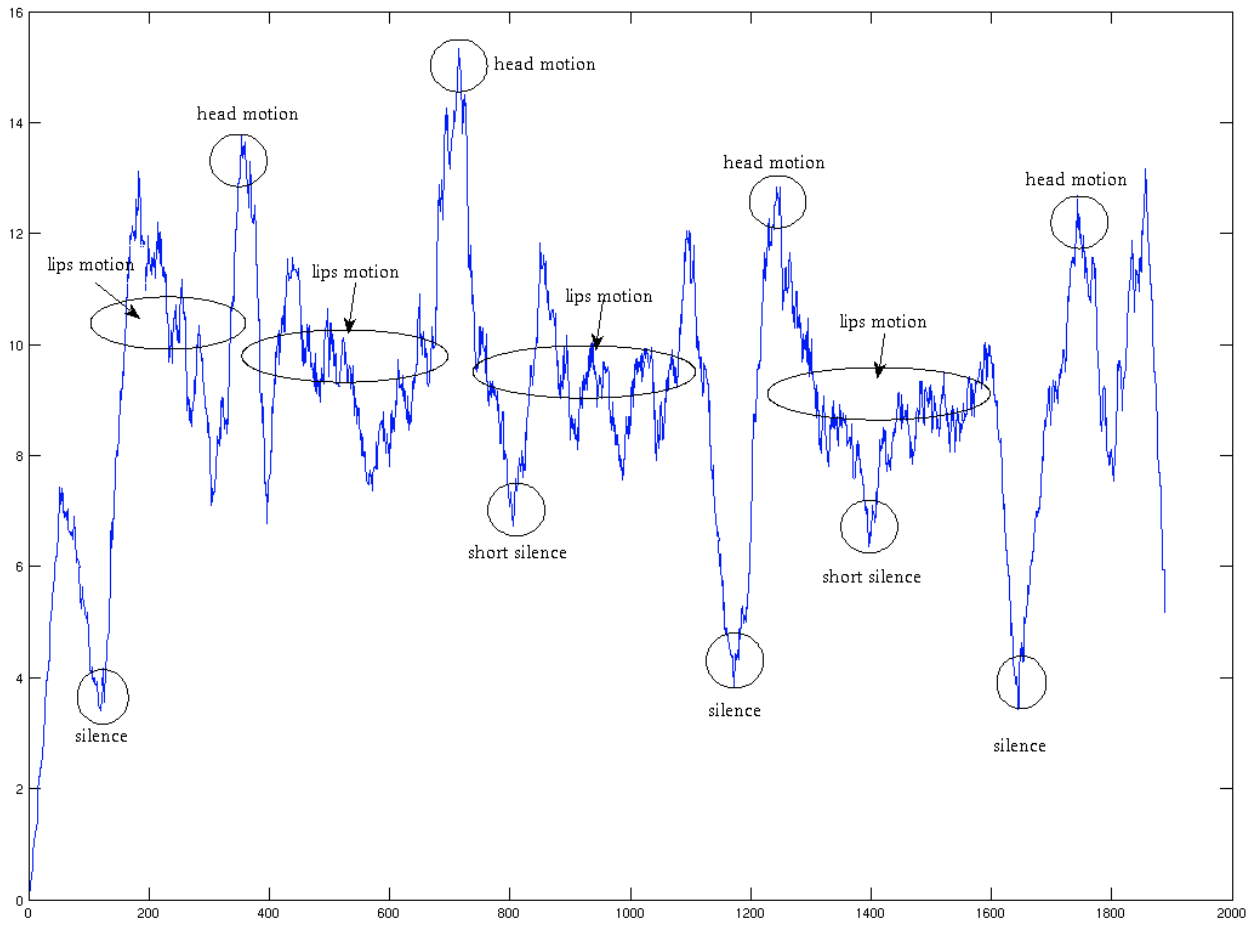


Figure 8.7: Lips activity curve.

First, the lips activity is estimated on every sequence of images as follows:

$$la_s = \frac{\sum_{k=1}^{N_w-1} mse(k, k+1)}{N_w - 1} \tag{8.10}$$

Then, the lips activity for each couple (A_i,V_j) is expressed by:

$$la_{A_iV_j} = \frac{\sum_{s=1}^{N_{AllShots}} N_s.la_s}{\sum_{s=1}^{N_{AllShots}} N_s} \tag{8.11}$$

If la_c is higher than a fixed threshold Thr_{la}, we assume that the corresponding person is talking. Practically, the threshold Thr_la is chosen very low in order to allow the association between faces and speech even if the person is not talking for a long time.

Thus, the elements of the matrix F' are multiplied by coefficients $\delta_{A_iV_j}$:

$$\delta_{A_iV_j} = \begin{cases} 0 & if \quad la_c < thr_{la} \\ 1 & if \quad la_c \geq thr_{la} \end{cases} \tag{8.12}$$

and the new matrix F'' is equal to:

$$F'' = \begin{pmatrix} \delta_{11}.W_{11}.C(A_1,V_1) & \dots & \delta_{1n_v}.W_{1n_v}.C(A_1,V_{n_v}) \\ \delta_{21}.W_{21}.C(A_2,V_1) & \dots & \delta_{2n_v}.W_{2n_v}.C(A_2,V_{n_v}) \\ \dots & \dots & \dots \\ \delta_{n_a1}.W_{n_a1}.C(A_{n_a},V_1) & \dots & \delta_{n_an_v}.W_{n_an_v}.C(A_{n_a},V_{n_v}) \end{pmatrix} \tag{8.13}$$

Consequently, the association between the face v and the voice a is prohibited if $\delta_{av} = 0$.

We notice in the above matrix that δ is a binary value ($\delta \in \{0,1\}$). However W belongs to an interval ($W \in [1,0]$). This can be justified by the facts that:

- if a person is shown to not having labial activity, it is sure he is not talking. Thus the decision is binary ($\delta \in \{0,1\}$). However, if the size of the person is greater than others, we can not make a binary decision but we can combine it to other information to build stronger decision.

- from the optimization point of view, it will be helpful to discard some of the non-necessary computation by focusing on faces where the lips activity is significant. Moreover, as said previously, we may choose tolerated threshold to allow low lips activity rates.

8.2 Audiovisual system for people indexing

At the end of both audio clustering (cf. part 1) and video clustering (cf. part 2), a list of audio (respectively video) clusters as well as the similarity measures for each couple of clusters are computed. We have studied in section 8.1 the association between each audio cluster and video cluster by computing the co-occurrence matrix.

Since the confidence of the bottom-up clustering process decreases gradually as it approaches to the top of the clustering hierarchy, the use of additional information in later stages such as the co-occurrence matrix will help improving the clustering performance.

A good way to implement our proposal is to apply the algorithm illustrated in figure 8.8:

1. A first step of confident audio clustering and video clustering is applied using "strong" thresholds that insure high clusters purity but possibly some additional clusters. The n_a audio clusters, the n_v video clusters, as well as the similarity matrices S_a and S_v computed for each couple of clusters, are retained.

2. Using audio clusters and video clusters, compute the co-occurrence matrix m of $n_a \times n_v$ dimension. Then, deduce the matrices m_a and m_v as explained in section 8.1.

3. Using matrix m_a defined in 8.2, update matrix S_a as follows: for each couple of clusters A_i and A_j, compute $\alpha(A_i, A_j)$:

$$\alpha(A_i, A_j) = \sum_{v=1}^{n_v} m_a(A_i, V_v).m_a(A_j, V_v) \tag{8.14}$$

If

$$S_a^{'}(A_i, A_j) = \tau_1.S_a(A_i, A_j) + \tau_2.\alpha(A_i, A_j) \tag{8.15}$$

Then, merge the couple of audio clusters that correspond to the maximum similarity, only if that maximum is higher than the threshold Thr_a. Experimentally, τ_1, τ_2 and Thr_a were fixed respectively to $\frac{1}{2}$, 2 and $\frac{1}{2}$.

$$(A_I, A_J) = \underset{(A_i, A_j)}{\arg\max}(S_a^{'}(A_i, A_j)) \quad if \quad \max(S_a^{'}(A_I, A_J)) > Thr_a \tag{8.16}$$

Identically, using matrix m_v defined in 8.3, update matrix S_v as follows: for each couple of clusters V_k and V_l, compute $\beta(V_k, V_l)$:

$$\beta(V_k, V_l) = \sum_{a=1}^{n_a} m_v(A_a, V_k).m_v(A_a, V_l) \tag{8.17}$$

and then,

$$S_v^{'}(V_k, V_l) = \rho_1.S_v(V_k, V_l) + \rho_2.\beta(V_k, V_l) \tag{8.18}$$

Then, merge the couple of video clusters that correspond to the maximum similarity, only if that maximum is higher than the threshold Thr_v. Experimentally, ρ_1, ρ_2 and Thr_v were fixed respectively to $\frac{1}{2}$, 2 and $\frac{1}{2}$.

$$(V_K, V_L) = \underset{(V_k, V_l)}{\arg\max}(S_v^{'}(V_k, V_l)) \quad if \quad \max(S_v^{'}(V_K, V_L)) > Thr_v \tag{8.19}$$

After each merging, the number of clusters decreases by 1, thus the matrices S_a, S_v, m, m_a and m_v are updated at the end of each iteration.

4. When the stopping criteria for both audio and video clustering are verified, and the final co-occurrence matrix m is provided, we compute the weighted co-occurrence matrix F'' in terms of the face size and the lips activity detection, as explained in previous section. Using this matrix, we can deduce the voice and/or the face of each person. Three types of clusters emerge: 1) voice-only clusters where people talk in the video but do not appear. 2) face-only clusters where people appear in the video but do not talk. 3) voice-and-face clusters where people appear and talk in the video.

Table 8.1 shows an example illustrating the output of the system: row 1 means that person P1 is talking (voice=1) and appearing (face=1) during 12.35 sec; row 2 means that person P2 is only talking in the interval [12.350 , 23.475]; row 3 means that nobody is appearing or talking in the interval [24.474 , 28.325]; rows 4 and 5 mean that P1 is talking (and not appearing) and P2 is appearing (and not talking) in the interval [28.325 , 31.050], etc.

Figure 8.9 illustrates another possible output where persons are listed with samples of their voice and face if they exist: P3 corresponds to a non-appearing person, P4 corresponds to a non-talking person. Moreover, readers are invited to check the format of XML output file automatically generated described in Appendix C.

Table 8.1: Output 1: index file.

row	start time (sec)	end time (sec)	Identity	voice	face
1	0.000	12.350	P1	1	1
2	12.350	23.475	P1	1	0
3	23.475	28.325	nobody	0	0
4	28.325	31.050	P1	1	0
5	28.325	31.050	P2	0	1
6	31.050	50.450	P2	1	1
7	50.450	54.275	nobody	0	0
8	54.275	93.525	P3	1	1
...	...	...	...	...	...

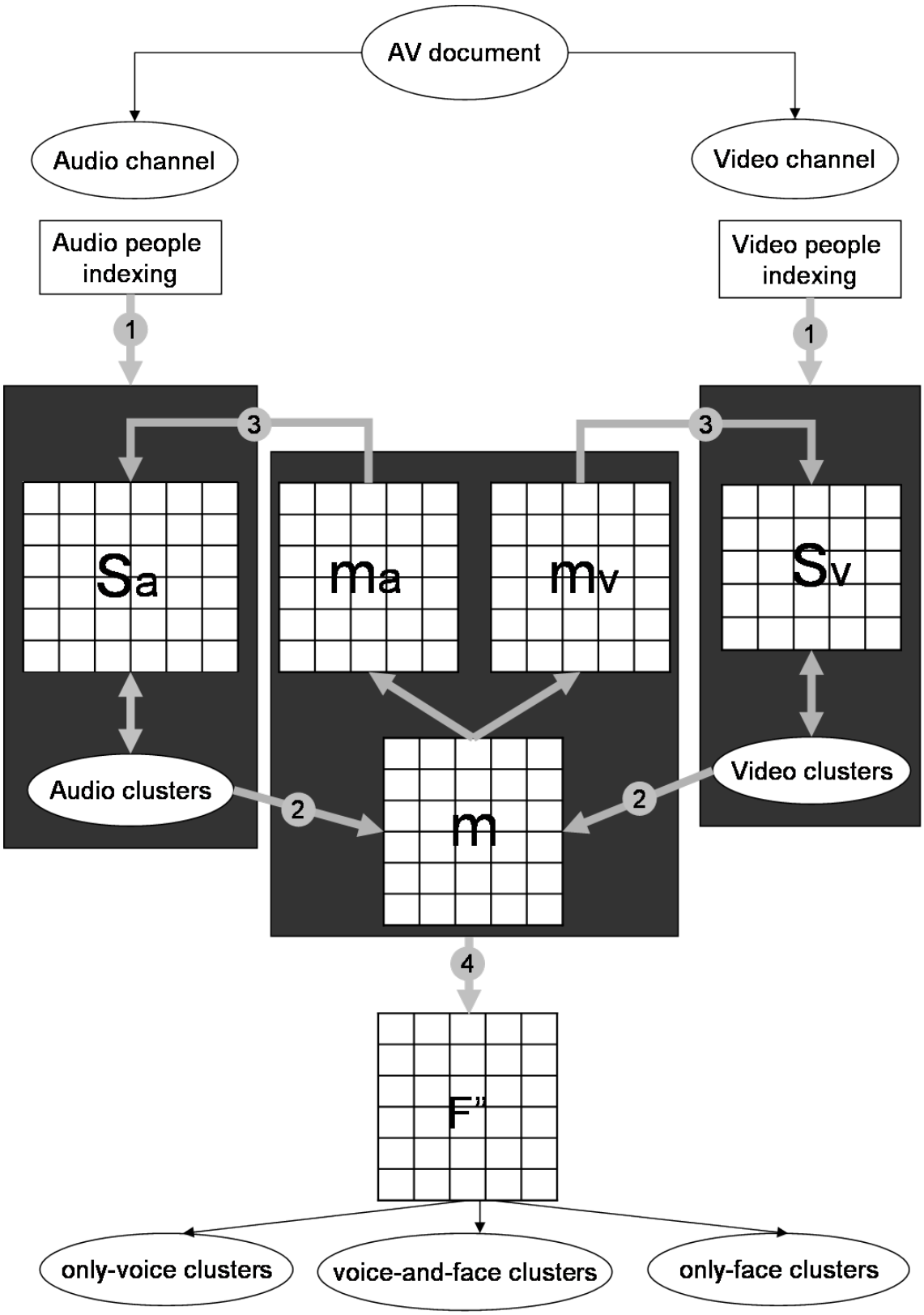
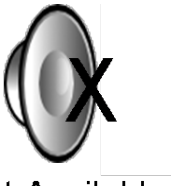


Figure 8.8: Architecture of the audio-visual people indexing system.

Figure 8.9: Output 2: list of persons with the most representative voices and faces.

P1		
P2		
P3		
P4		
...	...	...

Chapter 9

Experiments and Results

Contents

9.1 Database 164

9.2 Results of the speaker diarization 164

9.3 Results of the video people diarization 165

9.4 Results of the audiovisual association 166

9.4.1 The baseline system 167

9.4.2 The use of the face size 169

9.4.3 The use of the lips activity rate 170

9.4.4 The combined system 171

9.5 Analysis of the errors 173

In this chapter, we test our proposed methods. First, we detail the characteristics of our database. Second, we evaluate the impact of using video information on the audio speaker diarization output. Then we evaluate the impact of using audio information on the video person clustering output. Moreover, we evaluate the performance of the audiovisual baseline association using the cooccurrence matrix. Then, we evaluate the gain of using the face size and the gain of using the lips activity. Finally we test the performance of the combined association system that uses the cooccurrence matrix, the face size and the lips activity rate.

9.1 Database

The audiovisual database is the same as the one used in part II to evaluate the visual indexing system. Beside the annotation in terms of appearing persons that was already done, we annotate the audio channel in terms of different speakers talking in each video file, as well as the talking faces (audiovisual persons), non-talking faces (only-face persons), non-appearing persons (only-voice persons).

Table 9.1 shows the characteristics of the 14 files used in our test. They can be divided into three sub-sets: news, debates and movies. For each file, are reported the file duration, the speech time, the number of speakers in the ground-truth (Ref. spkrs), the time when faces appear (Faces time), the number of appearing faces in the ground-truth (Ref. faces). We can deduce that the total number of speakers is 568 and the total number of faces is 1315.

9.2 Results of the speaker diarization

In this section, we mesure the performance of the diarization system before and after using the video information. To do this, we use the diarization error rate (DER) that was introduced in part I (cf. section 3.4).

Table 9.2 shows that the overall weighted DER decreases from 25.35% to 19.64% when applying our audiovisual association. For news, the gain is about 2.83 % (from 18.68% to 15.85%). For debates, the improvement is very important (from 25.96% to 14.89%). This can be explained by the fact that the clustering while using the audio information is more difficult than for news, however, the use of video information corrects this problem because the face clustering is very good in these scenarios (cf. table 6.11). For movies, there is a slight gain of 1.11% (from 40.81% to 39.70%). It can be explained by the fact that both audio and video error rates are high.

Table 9.2 also reveals that the benefit of the fusion method is shown in 11 over 14 files: we notice that despite the loss of 2.5% on “CCTV” program, the gain is over 20% on “C’est dans l’air” talk show.

Table 9.1: Description of the audiovisual database.

	Time (s)	Speech time (s)	Ref. spkrs	Faces time (s)	Ref. faces
ABC news	1708	1362	54	556	79
CNN news	1779	1496	41	629	55
France2 news (1)	2479	2125	86	1142	145
France2 news (2)	2231	1925	50	1454	117
LBC news	3500	2348	46	1773	156
CCTV news	3008	1771	34	925	74
News	**14705**	**11027**	**311**	**6479**	**626**
Le Grand Journal (1)	3599	2661	40	987	74
Le Grand Journal (2)	3524	1648	45	1512	167
C'est notre affaire	1602	1495	19	3514	21
C'est dans l'air	3911	3847	25	2677	49
Debates	**12636**	**9651**	**129**	**8690**	**311**
Les Choristes	3600	796	21	1566	139
Amelie	2645	1067	35	635	50
Asterix Obelix	2191	1668	33	902	93
Virgins Suicide	2646	840	39	705	96
Movies	**11082**	**4371**	**128**	**3808**	**378**
Overall	**38423**	**25049**	**568**	**18977**	**1315**

9.3 Results of the video people diarization

In this section, we evaluate our system in regards to the clustering of faces. To do this, we use the clustering error rate (CER) that was introduced in part II (cf. section 6.1).

Table 9.3 shows the CER before and after using the audio information. The overall CER decreases from 19.75% to 17.22%. For news, the gain is 1.46% (from 9.10% to 7.64%). For debates, the gain is 3.32% with a final CER of 12.41%. For movies, the gain is 3.23% but the CER is still high (40.49%). Although the high CER, figures 9.1 and 9.2 show some cases where the clustering remains good although the high variation in almost all visual descriptors.

Table 9.2: Results of the "Only-Audio" and "Audiovisual" processings.

	Only-audio processing		**Audiovisual processing**	
	Sys. spkrs.	DER (%)	Sys. spkrs.	DER (%)
ABC news	49	23.3	43	16.5
CNN news	42	11.5	41	12.2
France2 news (1)	73	18.6	61	16.2
France2 news (2)	69	27.7	56	21.3
LBC news	60	14.3	50	10.1
CCTV news	56	17.3	53	19.7
News	**349**	**18.68**	**301**	**15.85**
Le Grand Journal (1)	49	14.1	43	9.0
Le Grand Journal (2)	110	16.3	99	12.5
C'est notre affaire	26	48.1	25	43.4
C'est dans l'air	40	29.7	29	8.9
Debates	**225**	**25.96**	**196**	**14.89**
Les Choristes	71	40.1	63	41.5
Amelie	81	36.6	75	34.7
Asterix Obelix	41	43.4	36	42.2
Virgins Suicide	51	41.7	46	38.4
Movies	**244**	**40.81**	**220**	**39.70**
Overall	**818**	**25.35**	**717**	**19.64**

9.4 Results of the audiovisual association

In this section, we test the efficiency of our proposed audiovisual association. To do this, we compute the precision and the recall of detecting "talking faces", "non-talking faces", and "non-appearing" persons (only voice): for each, the number of false positives (FP), false negatives (FN), true positives (TP), and true negatives (TN) are computed with respect to positive (P) and negative (N) people manually annotated in the ground truth.

First, we evaluate the baseline system where only cooccurrence matrix is used (cf. section 8.1.1). Then, we test the impact of using "the face size" (cf. section 8.1.2), the "lips

Table 9.3: Results of the "Only-video" and "Audiovisual" processings.

	Only-video processing		**Audiovisual processing**	
	Sys. faces	CER (%)	Sys. faces	CER (%)
ABC news	99	5.18	94	4.60
CNN news	82	18.35	78	14.24
France2 news (1)	176	3.94	171	3.33
France2 news (2)	145	5.22	141	3.90
LBC news	233	15.56	212	12.87
CCTV news	74	5.27	73	5.25
News	**809**	**9.10**	**769**	**7.64**
Le Grand Journal (1)	293	14.60	280	10.20
Le Grand Journal (2)	278	20.60	265	15.07
C'est notre affaire	50	49.75	43	35.82
C'est dans l'air	91	8.92	90	8.66
Debates	**712**	**15.73**	**678**	**12.41**
Les Choristes	370	43.38	360	41.82
Amelie	135	60.37	109	50.12
Asterix Obelix	216	42.83	205	41.82
Virgins Suicide	165	30.61	157	27.36
Movies	**886**	**43.72**	**831**	**40.49**
Overall	**2407**	**19.75**	**2278**	**17.22**

activity" (cf. section 8.1.3). Finally, we evaluate the overall system where all those components are added.

9.4.1 The baseline system

Table 9.4 shows the results obtained for each file and category as well as the weighted overall scores. We notice that the talking faces are detected with a precision of 80% despite the low recall score (32%). On the other hand, the non-talking faces are detected with a precision of 65% and a recall of 92%. Besides, the non-appearing persons are detected with a precision of



Figure 9.1: Example 1 of the movie "Les Choristes".



Figure 9.2: Example 2 of the movie "Les Choristes".

43% and a recall of 55%. Another thing to notice is that, as expected, the results obtained for news are clearly better than for debates and movies.

Table 9.4: Results of the audiovisual association: detection of the talking faces, non-talking faces and only-voices.

	Talking faces			**Non-talking faces**			**Only voices**		
	Num.	Prec.	Rec.	Num.	Prec.	Rec.	Num.	Prec.	Rec.
ABC news	21	0.86	0.56	70	0.80	0.95	10	0.10	0.25
CNN news	16	0.88	0.58	57	0.82	0.96	6	0.17	0.20
France2 news (1)	29	0.90	0.56	124	0.84	0.97	17	0.53	0.82
France2 news (2)	23	0.87	0.54	115	0.85	0.97	19	0.53	0.77
LBC news	22	0.86	0.49	167	0.88	0.98	25	0.52	0.69
CCTV news	21	0.81	0.85	32	0.90	0.89	5	0.40	0.33
News	**132**	**0.87**	**0.58**	**565**	**0.86**	**0.96**	**82**	**0.44**	**0.45**
Le Grand Journal (1)	20	0.55	0.34	175	0.88	0.94	22	0.68	0.65
Le Grand Journal (2)	30	0.43	0.25	137	0.72	0.85	24	0.54	0.50
C'est notre affaire	12	0.92	0.38	29	0.38	0.92	3	0.33	0.50
C'est dans l'air	16	1.0	0.44	46	0.55	1.0	3	0.33	1.0
Debates	**78**	**0.65**	**0.34**	**387**	**0.75**	**0.91**	**52**	**0.57**	**0.58**
Les Choristes	11	1.0	0.12	120	0.32	1.0	19	0.21	0.80
Amelie	14	0.64	0.19	65	0.40	0.54	14	0.21	0.75
Asterix Obelix	12	1.0	0.13	89	0.10	1.0	16	0.56	0.82
Virgins Suicide	15	0.87	0.16	80	0.12	1.0	11	0.18	0.67
Movies	**52**	**0.90**	**0.15**	**354**	**0.23**	**0.77**	**60**	**0.30**	**0.72**
Overall	**262**	**0.80**	**0.32**	**1306**	**0.65**	**0.92**	**194**	**0.43**	**0.55**

9.4.2 The use of the face size

In this section, we test the efficiency of adding the weight of the face size information as explained in section 8.1.2. Table 9.5 shows slight improvements: we can notice a gain of about 4% on the precision (from 0.80 to 0.84) and the recall (from 0.32 to 0.36) of detecting the talking faces compared to the baseline system (cf. table 9.4).

Table 9.5: Results of the audiovisual association improved by the face size: detection of the talking faces, non-talking faces and only-voices.

	Talking faces			**Non-talking faces**			**Only voices**		
	Num.	Prec.	Rec.	Num.	Prec.	Rec.	Num.	Prec.	Rec.
ABC news	19	0.95	0.56	72	0.81	0.98	12	0.17	0.50
CNN news	16	0.88	0.58	57	0.82	0.96	6	0.17	0.20
France2 news (1)	29	0.90	0.56	124	0.84	0.97	17	0.53	0.82
France2 news (2)	23	0.87	0.54	115	0.85	0.97	19	0.53	0.77
LBC news	25	0.88	0.56	164	0.90	0.98	22	0.68	0.79
CCTV news	21	0.81	0.85	32	0.90	0.89	5	0.40	0.33
News	**133**	**0.88**	**0.59**	**564**	**0.86**	**0.96**	**81**	**0.48**	**0.67**
Le Grand Journal (1)	23	0.61	0.47	173	0.91	0.95	20	0.60	0.52
Le Grand Journal (2)	33	0.55	0.35	135	0.75	0.87	22	0.59	0.50
C'est notre affaire	13	1.0	0.45	28	0.43	1.0	2	0.50	1.0
C'est dans l'air	16	1.0	0.44	46	0.55	1.0	3	0.33	1.0
Debates	**85**	**0.72**	**0.41**	**382**	**0.77**	**0.93**	**47**	**0.57**	**0.52**
Les Choristes	11	1.0	0.12	120	0.32	1.0	19	0.21	0.80
Amelie	17	0.71	0.25	62	0.42	0.54	11	0.27	0.50
Asterix Obelix	14	1.0	0.15	88	0.10	1.0	15	0.50	0.73
Virgins Suicide	18	1.0	0.26	78	0.13	1.0	9	0.33	1.0
Movies	**60**	**0.92**	**0.19**	**348**	**0.24**	**0.77**	**54**	**0.32**	**0.72**
Overall	**278**	**0.84**	**0.36**	**1294**	**0.67**	**0.93**	**182**	**0.46**	**0.62**

9.4.3 The use of the lips activity rate

In this section, we test the efficiency of adding, to the baseline cooccurrence matrix, the lips activity information as explained in section 8.1.3. Table 9.6 shows slight improvements: for example, we can notice a gain of 3% on the precision (from 0.65 to 0.68) and the recall (from 0.92 to 0.95) of detecting non-talking faces compared to the baseline system (cf. table 9.4).

Table 9.6: Results of the audiovisual association using face size: detection of the talking faces, non-talking faces and only-voices.

	Talking faces			**Non-talking faces**			**Only voices**		
	Num.	Prec.	Rec.	Num.	Prec.	Rec.	Num.	Prec.	Rec.
ABC news	24	0.92	0.69	67	0.85	0.97	7	0.14	0.25
CNN news	15	0.87	0.54	58	0.81	0.96	7	0.29	0.40
France2 news (1)	29	0.93	0.59	124	0.85	0.98	17	0.59	1.0
France2 news (2)	25	0.84	0.58	113	0.87	0.96	17	0.53	0.82
LBC news	25	0.84	0.58	164	0.91	0.97	22	0.59	0.68
CCTV news	22	0.73	0.80	32	1.0	0.84	5	0.40	0.33
News	**140**	**0.86**	**0.61**	**558**	**0.88**	**0.97**	**72**	**0.49**	**0.69**
Le Grand Journal (1)	21	0.71	0.47	175	0.90	0.96	22	0.59	0.57
Le Grand Journal (2)	34	0.53	0.35	134	0.75	0.86	21	0.52	0.73
C'est notre affaire	13	0.92	0.41	28	0.39	0.92	2	0.50	0.50
C'est dans l'air	17	1.0	0.46	45	0.56	1.0	2	0.50	1.0
Debates	**85**	**0.73**	**0.39**	**382**	**0.77**	**0.93**	**47**	**0.55**	**0.50**
Les Choristes	11	1.0	0.12	120	0.32	1.0	19	0.16	0.67
Amelie	17	0.71	0.27	63	0.63	0.89	12	0.25	0.50
Asterix Obelix	12	0.92	0.12	89	0.10	0.90	16	0.63	0.91
Virgins Suicide	15	1.0	0.18	80	0.15	1.0	11	0.18	0.67
Movies	**55**	**0.89**	**0.16**	**352**	**0.28**	**0.91**	**58**	**0.31**	**0.72**
Overall	**280**	**0.83**	**0.35**	**1292**	**0.68**	**0.95**	**177**	**0.45**	**0.62**

9.4.4 The combined system

In this section, we test the audiovisual association system when adding all above components to the baseline system as explained in equation 8.13. Results are reported in table 9.7. This table shows a gain of 10% on the precision of detecting talking faces (from 0.80 to 0.90) compared to the baseline system, and a gain of 14% on the recall (from 0.32 to 0.46). Similarly there is a gain of 7% (respectively 4%) on the precision (respectively recall) of detecting non-talking

faces, and a gain of 35% (respectively 15%) on the precision (respectively recall) of detecting the non-appearing persons.

Table 9.7: Results of the audiovisual association using face size: detection of the talking faces, non-talking faces and only-voices.

	Talking faces			**Non-talking faces**			**Only voices**		
	Num.	Prec.	Rec.	Num.	Prec.	Rec.	Num.	Prec.	Rec.
ABC news	28	0.96	0.84	63	0.92	0.98	3	1.0	0.75
CNN news	20	0.85	0.74	53	0.89	0.94	2	1.0	0.40
France2 news (1)	36	0.97	0.76	117	0.91	0.99	10	0.90	0.82
France2 news (2)	31	0.94	0.78	107	0.93	0.98	11	0.91	0.77
LBC news	32	0.88	0.72	157	0.93	0.97	15	1.0	0.79
CCTV news	23	0.87	1.0	30	1.0	0.91	3	1.0	0.50
News	**170**	**0.92**	**0.80**	**527**	**0.92**	**0.97**	**44**	**0.95**	**0.72**
Le Grand Journal (1)	26	0.73	0.53	169	0.92	0.96	16	0.87	0.61
Le Grand Journal (2)	39	0.72	0.54	128	0.81	0.91	15	0.87	0.50
C'est notre affaire	14	0.93	0.45	27	0.41	0.92	1	1.0	0.50
C'est dans l'air	18	1.0	0.50	44	0.59	1.0	1	1.0	1.0
Debates	**97**	**0.80**	**0.49**	**368**	**0.80**	**0.93**	**33**	**0.88**	**0.56**
Les Choristes	15	1.0	0.16	116	0.34	1.0	15	0.33	1.0
Amelie	24	0.92	0.71	55	0.84	0.96	4	1.0	0.67
Asterix Obelix	14	1.0	0.15	87	0.10	1.0	14	0.79	1.0
Virgins Suicide	16	1.0	0.19	79	0.15	1.0	10	0.30	1.0
Movies	**69**	**0.97**	**0.23**	**337**	**0.32**	**0.98**	**43**	**0.54**	**0.92**
Overall	**331**	**0.90**	**0.46**	**1232**	**0.72**	**0.96**	**120**	**0.78**	**0.70**

Table 9.8 summerizes the overall improvements: the total precision (respectively recall) for each system adds up the precisions (respectively recalls) of the talking faces, non-talking faces and non-appearing voices obtained for that system. This table shows that the overall gain is about 11% on the precision (from 65% to 76%) and about 8% on the recall (from 67% to 75%).

Table 9.8: Comparison between the baseline audiovisual association system and the improved systems.

	S1: Baseline	**S2: S1 + Face size**	**S3: S1 + Lips activity**	**S4: S1 + Face size + Lips activity**
Precision	65 %	67%	71%	**76**%
Recall	67%	69%	68%	**75**%

9.5 Analysis of the errors

After combining all audio and video components, different types of errors still remains. From the audio point of view, we have found that:

- On broadcast news, the errors are especially due to the confusion between people that have the same background noise (e.g. interviewees in demonstrations, etc.). Sometimes, errors are due to the dissimularity between turns of a reporter that is either talking in the studio or in noisy places.

- On debates, the errors are especially due to the high interaction rate between people (overlapping voices).

- On movies, the errors are due to the high variations in the background (music, indoor, outdoor, croud, etc.), the little turns of speech, and the high interaction rate between actors.

From the video point of view, we have found that:

- On broadcast news, the errors are especially due to the similarity between faces that have little sizes, similar lightening and for whom the clothes look similar.

- On debates, the errors are especially due to the reports that are shown during the program (such in “C’est notre affaire” video file).

- On movies, the errors are due to the huge variation in the lightning conditions, the pose and the size of the face. Moreover, there are the similarity in clothes (especially in Asterix Obelix) between different actors, and variation in clothes for the same actor.

Conclusion

In this part, we review the fusion architectures, and we describe some of the existing approaches in audiovisual fusion. Then, we present our proposed method for audiovisual association using cooccurrence matrix as well as the enhancements that can be added by using additional constraints such as the size of the face and the lips activity rate. Moreover, we describe a framework that improves simultaneously the audio indexing output, the video indexing output, and the audiovisual association.

Experiments are done on a database that contains news, debates and movies. Results show the efficiency of the association method, and confirm the gain that video information (respectively audio information) can bring to the audio indexing (respectively the video indexing).

General Conclusions and Perspectives

In this work, we propose methods for unsupervised video indexing based on audio, video and audiovisual characterization of persons. Some of those methods are generic because they may be applied on other modalities or other kind of features. Those contributions are on different levels (low, intermediate and high level) of the system architecture, and on different modalities. Those skills enable us to build an efficient and robust overall system.

More particularly, in the **audio domain**, we propose a robust and portable audio indexing that has many strong points:

- It splits the audio stream into homogeneous regions using our proposed bidirectional GLR/BIC algorithm. Each of these regions corresponds to one audio source (one speaker, noise, music, etc.).

- After discarding non-speech part, a first local-global hierarchical bottom-up clustering step is done using BIC criterion.

- An iterative process is done to correct simultaneously the segmentation boundaries and the clustering purity, and discard the retrieved non-speech segments.

This system is tested on radio broadcast debates and evaluated in ESTER2 competition on radio broadcast news. Results show the efficiency and the portability of the proposed system.

In the **video domain**, we propose methods for visual persons indexing that have many strong points:

- It splits the video stream into shots using the same GLR/BIC method proposed for audio segmentation.

- It detects and tracks persons based on face and clothing.

- It clusters the detected persons using SIFT features extracted within the face box, skin color of the face, 3D color histogram and texture of the clothing part. New similarity distances were proposed to ensure a robust clustering criterion. The clustering is processed in a hierarchical bottom-up manner.

Experiments are done on broadcast news, debates and movies that were manually annotated. The results show the impact of each technique/descriptor on the whole video indexing system as well as the robustness and the portability of that system.

From the audiovisual fusion point of view, we propose methods for audiovisual fusion system that has many strong points:

- an association between faces and voices is done using cooccurrence matrix and some additional priors like the face size and the lips activity.

- an iterative process enhances the only-audio speaker diarization system (respectively the only-video persons indexing system) with the help of video (respectively the help of audio).

Results obtained on broadcast news, debates and movies show the efficiency of the association method, and confirm the correlation between audio and video information, and the gain ensured by using both media.

Perspectives

Because of the diversity and the architecture of the whole audiovisual system, many perspectives result from this work.

From the **architectural, speediness and optimization point of view**, many enhancements can be done:

- A parallel processing can be done on audio channel and video channel at early stages (before the fusion process) in order to speed up the processing computation.

- Optimizations on some proposed algorithms and updating functions can be done (GLR/BIC, tracking, sampling, etc.) as well as the evaluation of those optimizations. This will enable real time applications.

In the **audio domain**, we highlight some new directions that aim to improve speaker diarization system:

- The detection of multi-sources/multi-speakers segments will be very helpful to locate interaction zones in an audio recording. Many hints can be revealed from our "in deep" observations of both features and processes behaviors, and can be useful to detect those zones and identify the speakers interaction in those zones. On the first hand, we have found that the values of the 4Hz modulation energy on interaction zones are lower than on clean scenarios. This may be useful to locate regions of simultaneous speakers. On the second hand, we may identify the speakers talking in these zones by computing the maximum likelihood between every frame within each zone and the automatic speaker clusters.

- Additional works can also be done to combine different clustering outputs, or to choose the best output between two of more different speaker clustering algorithms. Theoretically, the best of two clustering processes for the same number of clusters (N_c) is the one that maximizes the intra-cluster similarities and that minimizes the inter-clusters similarities.

In the **video domain**, many components of the video indexing system can be improved:

- The person detection component can be improved by detecting, not only frontal faces, but also profile faces, upper-body and full-body.

- The forward-backward tracking component can be improved by using particle or Kalman filters.

- The clustering process can also be enhanced by adding other descriptors like hair, by detecting special characteristics (glasses, mustaches, etc.).

In the **audiovisual fusion**, future works will focus on the **dynamic audiovisual model** of each person:

At the end of our association process, we have been able to classify persons into "talking faces", "non-talking faces" and "non-appearing persons". This classification will enable us to define, within each document (intra-document), audiovisual models for talking persons, visual models for non-talking person and voice model for non-appearing persons. Those models may be Gaussian mixture models (GMMs), Eigen vector space models (EVSMs), etc.

A second step will aim to define similarity measures in order to cluster the persons in the whole database (inter-document clustering). This measure must handle the huge variations that may appear on many levels: voice, background noise, lightening, mustaches, glasses, beard, clothing, etc.

This should contribute, in addition to our work in database structuring, to define a **dynamic unsupervised audiovisual identity** for each person within not only the document, but the whole database, and will help to better index, organize, classify, and browse the documents in regards to the persons that interact within them.

The last perspective, but not the least, should consider the way to present this work to the targeted users in exciting tools, because they may encounter difficulties in accepting new ways of doing things that involves changes in mindset. In all times, **users** are the ultimate judges of the usefulness of any technology in meeting their needs!

Appendix A

Application of the GLR-BIC segmentation for Program Boundaries Detection

We consider here that hypothesizes made for shot detection can be extended to program segmentation. It means that a selected set of features during a program behave in an homogeneous manner so that their values distribution can be modeled by a Gaussian law, and that features of two consecutive programs follow two rather different Gaussian laws. The last hypothesis is that a segment is of a minimal duration (in order to fix the size of the window used at the beginning of the algorithm, and to determine when fusion of boundaries must be operated). In our work, the goal is to check if typical video and audio features could validate the above hypothesizes.

A.1 Program boundaries detection using visual features

Each TV program has a certain number of visual characteristics that makes this program different from the others. For example, the luminance, the dominant colors, the activity rate in a soap episode are different from those observed on a TV game or a TV News program. As input for the system, a vector of features is originally provided as follows: every k seconds where k denotes the approximate value of the most frequent shot duration in seconds (experimentally k=8) for the tested content set, a frame is extracted and then, the three corresponding 2^m-dimension color histograms (R, G and B) are computed and their $3 * 2^m$ ($m = 8$ if the images are 8-bits images)

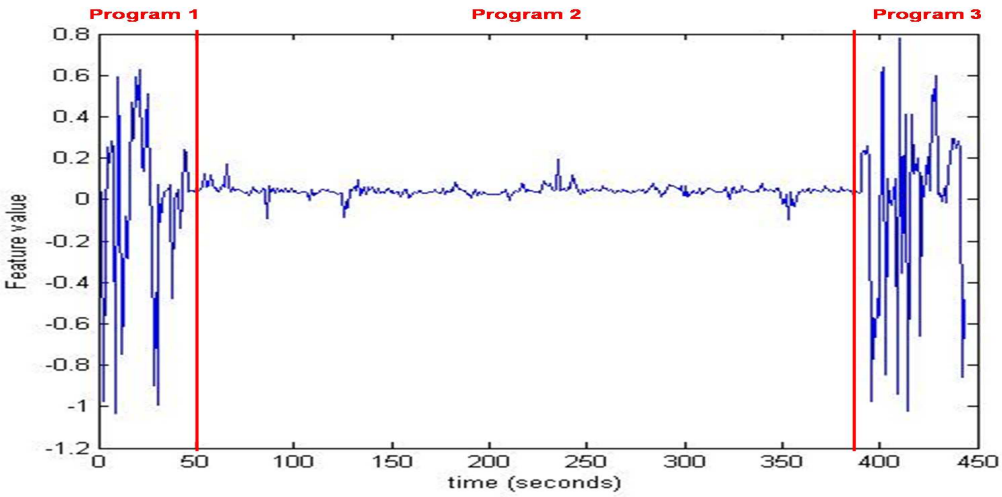


Figure A.1: Variation of Feature F1 for 3 consecutive programs.

values are concatenated in a vector. Furthermore, the Singular Value Decomposition (SVD) is applied in order to reduce the vectors dimension. Experimentally, an inertia ratio higher than 95% is reached with a vector dimension reduced to 12. Finally, the segmentation method explained above is applied on the sequence of these 12-dimension feature vectors. Results in table A.1 show a precision of about 78% on 5 days of television (120 hours). The major errors appear when there are commercial breaks: it may be typically explained because in this type of programs, in addition to their short duration, the homogeneity hypothesis is not still verified.

The variation and the distribution of the first "video" feature (after SVD) on 3 consecutive programs are given on figures A.1 and A.2. Figures A.3 and A.4 show the same phenomena for the third "video" feature obtained after SVD. We can verify that both variation and repartition are different for the three programs.

A.2 Program boundaries detection using acoustic features.

In this sub-section, we evaluate the ability of our segmentation method to detect program boundaries using only audio features. The input feature vectors are provided as follows: the first p Mel Frequency Cepstrum Coefficients ($p = 16$) are extracted every 10 ms using a sliding window of 20 ms. Those coefficients are then normalized and quantified between 0 and $D - 1$ ($D = 48$). Every k seconds ($k = 8$), histogram vectors are computed for each MFCC coefficient and concatenated to build a super-vector of dimension $p * D$. Then, the SVD is applied in order

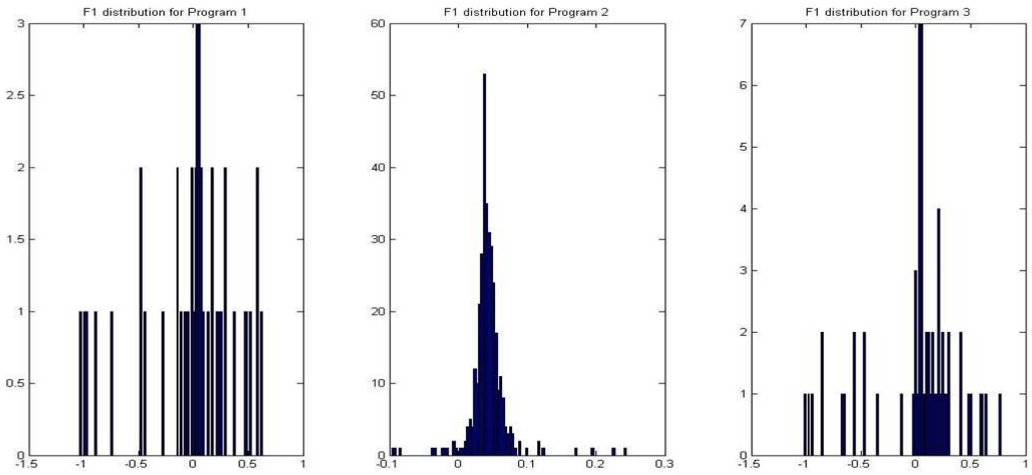


Figure A.2: Distribution of Feature F1 for 3 consecutive programs.

to reduce the dimension of those vectors. Practically, an inertia ratio of about 90% is obtained for a resulting vector dimension of 40. Finally the segmentation is applied. Table A.1 shows that scores are lower with the acoustic features (75%) than with the visual features.

A.3 Program boundaries detection using audiovisual features.

In order to exploit the complementary information brought by the two different modalities, the previous audio and video features are simultaneously used. Because we took the same temporal sampling to produce feature vectors ($k = 8$) with the same dimensions value ($3 * 2^m = p * D$) reduced by the same processing (SVD, histograms) for the above two methods, it is very easy to combine them using two kinds of fusion: fusion at the decision level and fusion at the feature level.

At the decision level, the fusion was done by computing:

$$\Delta BIC_{AV} = \Delta BIC_A + \Delta BIC_V \tag{A.1}$$

where $\Delta BIC_{AV} > 0$ corresponds ideally to a change between two TV programs.

At the feature level, the early fusion aims to concatenate the visual vector of features ($dimension = 3 * 2^m$) and the acoustic vector of features ($dimension = p * D$). SVD is then applied: a resulting vector of dimension 60 is obtained for an inertia ratio of about 90%. Finally,

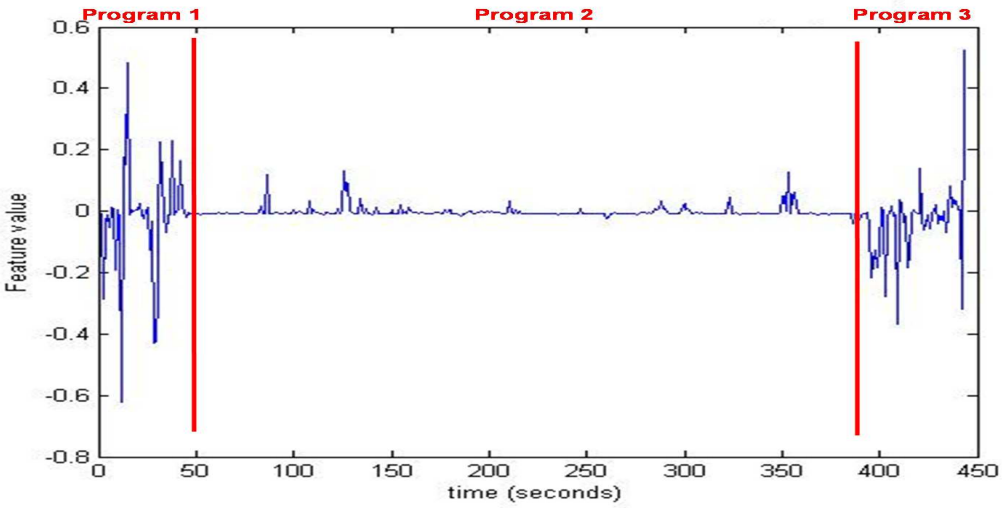


Figure A.3: Variation of Feature F3 for the same 3 consecutive programs.

the segmentation is processed to detect the frames of change. Experimentally, the early fusion at the features level gives (80.7%) better results than the fusion at the decision level (78.5%).

A.4 Experiments

Tests were carried out on 120 hours of TV videos recorded continuously from a general French TV channel during 5 days (including various kinds of programs such as news, weather forecast, talk-shows, movies, sports and sitcoms) with a rate of 25 frames/second. The size of the programs is very variable: from few minutes for weather forecast to 3 hours for a film.

For the segmentation step, we had to define the length of the fixed size window W and the penalty coefficient which depends on W and the dimension of the feature vectors (12 for video features, 40 for audio features and 60 for audio/video features). We chose a window size of 4 minutes (corresponding to 30 vectors) as the hypothesis on the minimal duration of a program. The penalty coefficient λ was tuned to 5 for the Video system, to 1.2 for the Audio system and 1 for the AV system.

To evaluate those systems, the ARGOS F-measure metric, described above, was used: it highlights the ability of the segmentation tool to gather units belonging to a same segment.

Results in Table A.1 show that the visual system is better (78%) than the audio one (75%). With audio features, the majority of errors appear especially when there are commercial breaks. This might be explained typically because this type of program does not follow the homogeneity

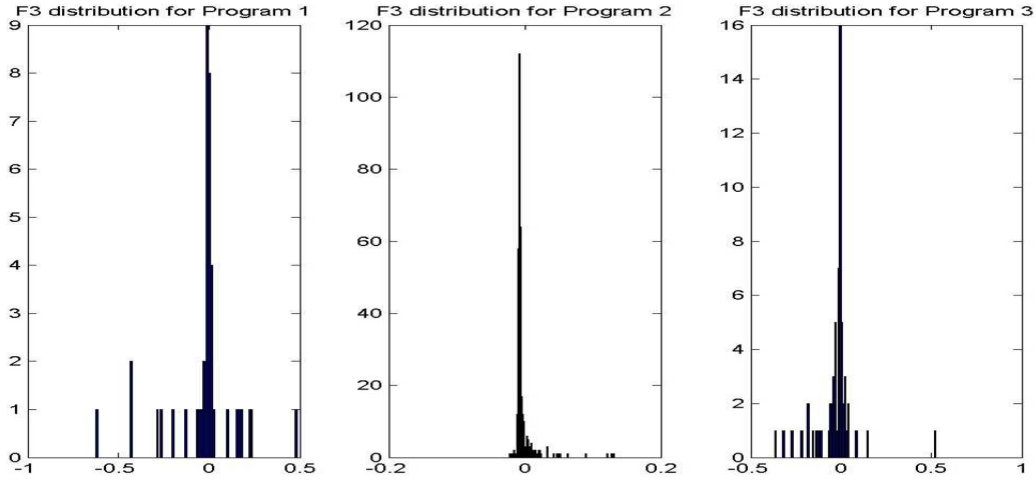


Figure A.4: Distribution of Feature F3 for the same 3 consecutive programs.

Table A.1: Results of the program boundaries detection (120 hours of test).

	Visual sytem	Audio system	**AV system**
$F - measure$	78.04%	75.16%	**80.72%**

hypothesis. We can see that the two modalities audio and visual bring complementary information because the results are better than those obtained with only one modality.

Many improvements can be done while taking into account some knowledge already identified in the state of the art. For example, on French TV, commercials are separated by a sequence of monochrome images (white, blue or black). As this kind of effect can be easily detected, improvements of about 9% ($F - measure = 87.34\%$) can easily be reached while gathering advertisements in a single program.

Comparison of the above results with those obtained by the state-of-the-art systems is a difficult task because corpora, units and metrics are different for each experience and cannot be shared. To our knowledge, there is no international campaign addressing this topic. In this case, the evaluation we provide here should be considered as a basic reference which can be used later to evaluate improvements of this method, or to compare with other future approaches.

As our system is almost knowledge-free, it can process any kinds of TV content without any prior training phase. In this way, it can be seen as a useful pre-processing step in the context of video indexing for example.

As part of the project ANR EPAC[20], the program boundaries detection was applied on 1700 hours of TV and Radio contents: the processing took less than 16 hours that is lower than (recording duration * 10^{-2}) with a non optimized version written in Matlab on a classical PC architecture.

[20]http://www.epac.univ-lemans.fr/

Appendix B

Additional visual features

- **HMMD Color Histograms.** The HMMD (Hue, Min, Max, Diff) [MrOVY01] color space is used in MPEG-7 standard. It is derived from the HSV and RGB spaces. The hue component is the same as in the HSV space, and max and min denote the maximum and the minimum among the R, G, and B values. The diff component is defined as the difference between max and min.

- **Contrast.** It is *the difference in visual properties that makes an object distinguishable from other objects and the background*[21]. There are different manners to compute the contrast of an image like the Weber contrast, the Michelson contrast and the RMS contrast. For more details about the definition, the measurements and the evaluation of the contrast, please refer to [Pel90].

- **Local edge features.** Object boundaries usually generate strong changes in images intensities. Edge detection is used to identify these changes in image segmentation task. The most popular edge detection approach is the Canny edge detector [Can86]. In [Per92], authors used steerable filters to extract local image edge features. Steerable filters [FY91] can provide any orientations information because they are excellent for the detailed analysis of boundaries. Those features can be used to detect frontal faces in images [SS04].

- **Histogram of edge directions.** Low-level shape-based features can be constructed from the edges in the images. A histogram of edge directions is translation invariant and

[21] http://en.wikipedia.org/wiki/Contrast_(vision)

it captures the general shape information in the image. Because the feature is global, it is robust to partial occlusion and local disturbance in the image.

- **Fourier and wavelet features.** The Fourier and wavelet transforms [BC99] are powerful tools for pattern recognition. One of their important properties is that a shift in the time domain causes no change in the magnitude spectrum. This can be used to extract invariant features in pattern recognition.

- **Illumination invariant color histograms.** Some undesirable limitations on the use of color features in content-based applications are due to the variation of the scene illumination conditions. In [OB02], a set of illumination-invariant descriptors is defined in order to achieve some robustness to variation in lighting conditions. These histograms are computed using invariant moments of the distribution in the RGB space.

- **Haar-like features.** These features owe their name to their intuitive similarity with Haar wavelets. They were introduced by Papageorgiou et al. in [POP98b]. The features set considers rectangular regions of the image and sums up the pixels in this region. Those features were used for object recognition ([POP98b], [VJ01]) and more particularly for face detection [VJ04].

- **Optical flow.** It is a dense field of displacement vectors which defines the translation of each pixel in a region. Optical flow is commonly used as a feature in motion-based segmentation and tracking applications. It is computed using the brightness constraint of consecutive images [HS92].

Appendix C

Output XML format of the audiovisual fusion

In order to deliver a comprehensive and portable output index file, we decide to use the XML format and we define three important elements as seen in figure C.1:

- the **Audio-Visual-People** element that contains the list of different persons that have audiovisual identities. For each person, we define: 1) the **Audio** element that contains the different turns where that person is talking, 2) the **Video** element that contains the different turns where that person is appearing.

- the **Only-Audio-People** element that contains the list of different persons that have only audio identities. For each person, we define the **Audio** element as above.

- the **Only-Video-People** element that contains the list of different persons that have only video identities. For each person, we define the **Video** element as above.

In the example shown in figure C.1, we incorporate additional information that may be useful for future use: Pitch, the confidence coefficient POF (Product of frequencies), the keyface (it corresponds to the maximum similarity between the face and the associated face cluster), the keyvoice (it corresponds to the maximum similarity between the voice segment and the associated speaker cluster), etc.

```
– <Document video_filename="19980104_ABC" run_version="IRIT-1"
  version_date="2010-02-27" author="Elie El-Khoury">
    <Description category="news" language="english"/>
  – <Audio-Visual-People>
    – <Person id="pers1" name="unknown" type="female" faceId="1" keyFace="face_1.jpg"
      voiceId="1" keyVoice="voice_1.wav">
        <Fusion POF="0.65" commonTime="19.040 sec"/>
      – <Audio time="20.010 sec" Pitch="177 Hz">
          <Turn startTime="8.630" endTime="18.700"/>
          <Turn startTime="362.290" endTime="372.230"/>
        </Audio>
      – <Video time="235.760 sec">
        – <Clothes>
            <Cloth id="cloth_1_1" image="cloth_1_1.jpg"/>
            <Cloth id="cloth_1_2" image="cloth_1_2.jpg"/>
          </Clothes>
        – <Turns>
            <Turn startTime="0.000" endTime="8.440"/>
            <Turn startTime="9.600" endTime="19.400"/>
            <Turn startTime="361.720" endTime="373.040"/>
          </Turns>
        </Video>
      </Person>
    + <Person id="pers2" name="unknown" type="male" faceId="3" keyFace="face_3.jpg"
      voiceId="2" keyVoice="voice_2.wav"></Person>
    </Audio-Visual-People>
  – <Only-Audio-People>
    – <Person id="pers3" name="unknown" type="female" voiceId="3"
      keyVoice="voice_3.wav">
      – <Audio time="51.350 sec" Pitch="190 Hz">
          <Turn startTime="181.630" endTime="212.980"/>
          <Turn startTime="214.290" endTime="334.290"/>
        </Audio>
      </Person>
    + <Person id="pers4" name="unknown" type="male" voiceId="4" keyVoice="voice_4.wav">
      </Person>
    </Only-Audio-People>
  – <Only-Video-People>
    – <Person id="pers5" name="unknown" type="male" faceId="2" keyFace="face_2.jpg">
      + <Video time="52.200 sec"></Video>
      </Person>
    </Only-Video-People>
  </Document>
```

Figure C.1: The output file in the XML format.

Appendix D

Publications

1. Elie El Khoury, Christine Senac, Philippe Joly. Unsupervised segmentation methods of TV contents. *International Journal of Digital Multimedia Broadcasting*, accepted with minor revision, to appear, april 2010.

2. Elie El Khoury, Christine Senac, Philippe Joly. Face-and-Clothing Based People Clustering in Video Content. *ACM Multimedia*, Philadelphia, Pennsylvania, ACM, march 2010.

3. Hervé Bredin, Lionel Koenig, Hélène Lachambre, Elie El Khoury. IRIT @ TRECVid HLF 2009 Audio to the Rescue. *TREC Video Retrieval Workshop (TRECVID 2009)*, Gaithersburg, MD, National Institute of Standards and Technology (NIST), november 2009.

4. Elie El Khoury, Gaël Jaffré, Julien Pinquier, Christine Senac. People indexing using audio and video segmentations. *International Workshop on Electronics, Control, Modelling, Measurement and Signals (ECMS 2009)*, Mondragon, Spain, 2009.

5. Elie El Khoury, Christine Senac, Julien Pinquier. Improved Speaker Diarization System for Meetings. *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2009),* Taipei, Taiwan, IEEE, p. 4241-4244, 2009.

6. Shih-Fu Chang, Junfeng He, Yu-Gang Jiang, Elie El Khoury, Chong-Wah Ngo, Akira Yanagawa, Eric Zavesky. Columbia University/VIREO-CityU/IRIT TRECVID2008 High-Level Feature Extraction and Interactive Video Search. *TREC Video Retrieval Workshop (TRECVID 2008)*, NIST in Gaithersburg, MD, National Institute of Standards and Technology (NIST), 2008.

7. Elie El Khoury, Christine Senac, Philippe Joly. Unsupervised TV Program Boundaries Detection Based on Audiovisual Features. *International Conference on Visual Information Engineering (VIE 2008)*, Xi'an China, IET (The Institution of Engineering and Technology, 2008.

8. Elie El Khoury, Gaël Jaffré, Julien Pinquier, Christine Senac. Association of Audio and Video Segmentations for Automatic Person Indexing. *International Workshop on Content-Based Multimedia Indexing (CBMI 2007)*, Bordeaux, France, IEEE, p. 287-294, 2007.

9. Elie El Khoury, Christine Senac, Régine André-Obrecht. Speaker Diarization: Towards a more Robust and Portable System. *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2007)*, Honolulu, Hawaii, USA, IEEE, p. 489-492, 2007.

10. Elie El Khoury, Sylvain Meigner, Christine Senac. Segmentation et regroupement en locuteurs pour la parole conversationnelle. *Journées d'Etudes sur la Parole (JEP 2008)*, Avignon, France, Association Francophone de la Communication Parlée (AFCP), p. 345-348, 2008.

Bibliography

[ALM02] J. Ajmera, H. Bourlard I. Lapidot, and I. Mccowan. Unknown-multiple speaker clustering using hmm. In *In Proceedings of ICSLP-2002*, pages 573–576, 2002.

[ANP07] P. Antonopoulos, N. Nikolaidis, and I. Pitas. Hierarchical face clustering using sift image features. In *Computational Intelligence in Image and Signal Processing, 2007. CIISP 2007. IEEE Symposium on*, pages 325–329, April 2007.

[AO88] R. André-Obrecht. A new statistical approach for automatic speech segmentation. *Transaction on Audio, Speech, and Signal Processing, IEEE*, 36(1):29–40, 1988.

[ARS08] M. Andriluka, S. Roth, and B. Schiele. People-tracking-by-detection and people-detection-by-tracking. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8, 2008.

[ASHP93] A. Azarbayejani, T. Starner, B. Horowitz, and A. Pentland. Visually controlled graphics. *IEEE Trans. Pattern Anal. Mach. Intell.*, 15(6):602–605, 1993.

[ATD05] A. Albiol, L. Torres, and E.J. Delp. Fully automatic face recognition system using a combined audio-visual approach. *Vision, Image and Signal Processing, IEE Proceedings* -, 152(3):318–326, June 2005.

[Avi01] Shai Avidan. Support vector tracking. In *IEEE Trans. on Pattern Analysis and Machine Intelligence*, pages 184–191, 2001.

[AZ05] Ognjen Arandjelovic and Andrew Zisserman. Automatic face recognition for film character retrieval in feature-length films. In *CVPR '05: Proceedings of*

the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, volume 1, pages 860–867. IEEE Computer Society, 2005.

[BC86] T J Broida and R Chellappa. Estimation of object motion parameters from noisy images. *IEEE Trans. Pattern Anal. Mach. Intell.*, 8(1):90–99, 1986.

[BC99] T. D. Bui and G. Chen. Invariant fourier-wavelet descriptor for pattern recognition. *Pattern Recognition*, 32:1083–1088, 1999.

[BC07] H. Bredin and G. Chollet. Audiovisual speech synchrony measure: application to biometrics. *EURASIP J. Appl. Signal Process.*, 2007(1):179–179, 2007.

[BFP10] Benjamin Bigot, Isabelle Ferrané, and Julien Pinquier. Exploiting speaker segmentations for automatic role detection. An application to broadcast news documents (regular paper). In *International Workshop on Content-Based Multimedia Indexing (CBMI), Grenoble, France, 23/06/2010-25/06/2010*, juin 2010.

[BHaPHK96] Peter N. Belhumeur, João P. Hespanha, Jo ao P. Hespanha, and David J. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19:711–720, 1996.

[BJA02] M.J. Beal, N. Jojic, and H. Attias. A self-calibrating algorithm for speaker tracking based on audio-visual statistical models. In *Acoustics, Speech, and Signal Processing, 2002. Proceedings. (ICASSP '02). IEEE International Conference on*, volume 2, pages 1997–2000, 2002.

[BLGT06] M. Bicego, A. Lagorio, E. Grosso, and M. Tistarelli. On the use of sift features for face authentication. In *Computer Vision and Pattern Recognition Workshop, 2006. CVPRW '06.*, pages 35–41, June 2006.

[BLSO08] Liang Bai, Songyang Lao, Alan F. Smeaton, and Noel E. O'Connor. Automatic summarization of rushes video using bipartite graphs. In *SAMT '08: Proceedings of the 3rd International Conference on Semantic and Digital Media Technologies*, pages 3–14, Berlin, Heidelberg, 2008. Springer-Verlag.

[BS60] A. Sklar B. Schweizer. Statistical metric spaces. *Pacific J. Math.*, pages 314–334, 1960.

[BSR00] Marcelo Bertalmio, Guillermo Sapiro, and Gregory Randall. Morphing active contours. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(7), 2000.

[BW98] J.S. Boreczky and L.D. Wilcox. A hidden markov model framework for video segmentation using audio and image features. In *Acoustics, Speech and Signal Processing, 1998. Proceedings of the 1998 IEEE International Conference on*, volume 6, pages 3741–3744 vol.6, May 1998.

[BZMG06] C. Barras, X. Zhu, S. Meignier, and J.L. Gauvain. Multi-stage speaker diarization of broadcast news. *IEEE Transactions on Audio, Speech and Language Processing*, 14:1505–1512, 2006.

[Can86] J Canny. A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 8(6):679–698, 1986.

[CD00] R. Cutler and L. Davis. Look who's talking: speaker detection using video and audio correlation. In *Multimedia and Expo, 2000. ICME 2000. 2000 IEEE International Conference on*, volume 3, pages 1589–1592 vol.3, 2000.

[Cet00] Mauro Cettolo. Segmentation, classification and clustering of an italian broadcast news corpus. In *In Proc. of RIAO*, pages 372–381, 2000.

[CG98] Scott S. Chen and P. S. Gopalakrishnan. Clustering via the bayesian information criterion with applications in speech recognition. volume 2, pages 645–648, 1998.

[CHJ+08] Shih-Fu Chang, Junfeng He, Yu-Gang Jiang, Elie El Khoury, Chong-Wah Ngo, Akira Yanagawa, and Eric Zavesky. Columbia University/VIREO-CityU/IRIT TRECVID2008 High-Level Feature Extraction and Interactive Video Search. In *TREC Video Retrieval Workshop (TRECVID), NIST in Gaithersburg, MD*. NIST, 2008.

[CLY09] Wei-Ta Chu, Ya-Lin Lee, and Jen-Yu Yu. Visual language model for face clustering in consumer photos. In *MM '09: Proceedings of the seventeen ACM*

international conference on Multimedia, pages 625–628, New York, NY, USA, 2009. ACM.

[CMD97] C.C. Chibelushi, J.S.D. Mason, and N. Deravi. Integrated person identification using voice and facial features. In *Image Processing for Security Applications (Digest No.: 1997/074), IEE Colloquium on*, pages 4/1–4/5, Mar 1997.

[CMM03] Csaba Czirjek, Sean Marlow, and Noel Murphy. Face detection and clustering for video indexing applications. In *in Proceedings of Advanced Concepts for Intelligent Vision Systems*, pages 2–5, 2003.

[CRM03] Dorin Comaniciu, Visvanathan Ramesh, and Peter Meer. Kernel-based object tracking. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25(5):564–575, 2003.

[CRPN03a] U.V. Chaudhari, G.N. Ramaswamy, G. Potamianos, and C. Neti. Audio-visual speaker recognition using time-varying stream. In *Proc. ICASSP, Hong Kong*, pages 712–715, 2003.

[CRPN03b] U.V. Chaudhari, G.N. Ramaswamy, G. Potamianos, and C. Neti. Information fusion and decision cascading for audio-visual speaker recognition based on time-varying stream reliability prediction. *Multimedia and Expo, IEEE International Conference on*, 3:9–12, 2003.

[CV03] Mauro Cettolo and Michele Vescovi. Efficient audio segmentation algorithms based on the bic. In *Proc. of the International Conference on Acoustics, Speech, and Signal Processing*, pages 537–540, 2003.

[Das94] Belur V. Dasarathy. *Decision Fusion.* IEEE Computer Society Press, Los Alamitos, CA, USA, 1994.

[Det00] Marcin Detyniecki. Mathematical aggregation operators and their application to video querying, 2000.

[DT05] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *In CVPR*, pages 886–893, 2005.

[DW00] P. Delacourt and C. J. Wellekens. Distbic: a speaker-based segmentation for audio data indexing. *Speech Commun.*, 32(1-2):111–126, 2000.

[EB05] N. Eveno and L. Besacier. Co-inertia analysis for "liveness" test in audio-visual biometrics. In *Image and Signal Processing and Analysis, 2005. ISPA 2005. Proceedings of the 4th International Symposium on*, pages 257–261, Sept. 2005.

[EKJPS07] E. El Khoury, G. Jaffré, J. Pinquier, and C. Senac. Association of Audio and Video Segmentations for Automatic Person Indexing. In *International Workshop on Content-Based Multimedia Indexing (poster session) (CBMI), Bordeaux, France, 25/06/07-27/06/07*, pages 287–294, http://www.ieee.org/, 2007. IEEE. http://cbmi07.labri.fr/ 5th-CBMI ISBN 1-4244-1010-X.

[EKMS08] Elie El-Khoury, Sylvain Meigner, and Christine Senac. Segmentation et regroupement en locuteurs pour la parole conversationnelle. In *Journées d'étude sur la parole, JEP'08*, 2008.

[EKSAO07] Elie El Khoury, Christine Senac, and Régine André-Obrecht. Speaker Diarization: Towards a more Robust and Portable System. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Honolulu, Hawaii, USA, 15/04/2007-20/04/2007*, pages 489–492, http://www.ieee.org/, 2007. IEEE.

[EKSJ08] Elie El Khoury, Christine Senac, and Philippe Joly. Unsupervised TV Program Boundaries Detection Based on Audiovisual Features. In *International Conference on Visual Information Engineering (VIE), Xi'an China, 29/07/2008-01/08/2008*, page (electronic medium), http://www.theiet.org/publishing/, 2008. IET. http://vie08.qmul.net/index.php.

[ESZ06] M. Everingham, J. Sivic, and A. Zisserman. Hello! my name is... buffy – automatic naming of characters in tv video. In *Proceedings of the British Machine Vision Conference, BMVC06*, page III:899, 2006.

[FD04] III Fisher, J.W. and T. Darrell. Speaker association with signal-level audiovisual fusion. *Multimedia, IEEE Transactions on*, 6(3):406–413, June 2004.

[FE08] Corinne Fredouille and Nicholas Evans. The lia rt'07 speaker diarization system. pages 520–532, 2008.

[FJ06] Thomas Foures and Philippe Joly. Scalability in human shape analysis. In *IEEE International Conference on Multimedia & Expo (ICME) (ICME), Toronto - Ontario - Canada, 09/07/06-12/07/06*, pages 2109–2112, http://www.ieee.org/, 2006. IEEE. ISBN 1-4244-0367-7.

[FLL+03] Tieyan Fu, Xiao Xing Liu, Lu Hong Liang, Xiaobo Pi, and A.V. Nefian. Audio-visual speaker identification using coupled hidden markov models. In *Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on*, volume 3, pages III–29–32 vol.2, Sept. 2003.

[FMJZ08] V. Ferrari, M. Marin-Jimenez, and A. Zisserman. Progressive search space reduction for human pose estimation. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8, 2008.

[FS97] Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting,. *Journal of Computer and System Sciences*, 55(1):119 – 139, 1997.

[FY91] William T. Freeman and Edward H. Adelson Y. The design and use of steerable filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13:891–906, 1991.

[FZ02] Andrew W. Fitzgibbon and Andrew Zisserman. On affine invariant clustering and automatic cast listing in movies. In *ECCV '02: Proceedings of the 7th European Conference on Computer Vision-Part III*, pages 304–320, London, UK, 2002. Springer-Verlag.

[GER07] O. Gillet, S. Essid, and G. Richard. On the correlation of automatic audio and visual segmentations of music videos. *Circuits and Systems for Video Technology, IEEE Transactions on*, 17(3):347–355, March 2007.

[GGM+05] S. Galliano, E. Geofrois, D. Mosterfa, J.F. Bonastre, and G. Gravier. The ester phase ii evaluation campaign for the rich transcription of the french broadcast news. In *European Conference on Speech Communication and Technology*, pages 1149–1152, 2005.

[GL94] J.L. Gauvain and C.H. Lee. Maximum a posteriori estimation for multivariate gaussian mixture observations of markov chains. *IEEE Transactions on Speech and Audio Processing*, pages 291–298, 1994.

[GSR91] H. Gish, M.H. Siu, and R. Rohlicek. Segregation of speakers for speech recognition and speaker identification. In *ICASSP '91: Proceedings of the Acoustics, Speech, and Signal Processing, 1991. ICASSP-91., 1991 International Conference*, pages 873–876, Washington, DC, USA, 1991. IEEE Computer Society.

[HDS73] R. M. Haralick, Dinstein, and K. Shanmugam. Textural features for image classification. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-3:610–621, November 1973.

[HE09] Anna Hilsmann and Peter Eisert. Tracking and retexturing cloth for real-time virtual clothing applications. In *MIRAGE '09: Proceedings of the 4th International Conference on Computer Vision/Computer Graphics CollaborationTechniques*, pages 94–105, Berlin, Heidelberg, 2009. Springer-Verlag.

[HF08] H. Hung and G. Friedland. Towards audio-visual on-line diarization of participants in group meetings. In *Workshop on Multi-camera and Multi-modal Sensor Fusion Algorithms and Applications - M2SFA2 2008*, Marseille France, 2008. Andrea Cavallaro and Hamid Aghajan.

[HJC06] Siba Haidar, Philippe Joly, and Bilal Chebaro. Mining for video production invariants to measure style similarity. *International Journal of Intelligent Systems (IJIS)*, 21(7):747–763, july 2006.

[HL01a] D.L. Hall and J. Linas. *Handbook of Multisensor Data Fusion*. CRC Press, May 2001.

[HL01b] E. Hjelmas and B.K. Low. Face detection: A survey. *Computer Vision and Image Understanding*, 83:236–274, 2001.

[HMVP08] Jing Huang, Etienne Marcheret, Karthik Visweswariah, and Gerasimos Potamianos. The ibm rt07 evaluation systems for speaker diarization on lecture meetings. pages 497–508, 2008.

[HN07] Kyu J. Han and Shrikanth S. Narayanan. A robust stopping criterion for agglomerative hierarchical clustering in a speaker diarization system. In *Interspeech*, pages 1853–1856, 2007.

[HS88] C. Harris and M. Stephens. A combined corner and edge detection. In *Proceedings of The Fourth Alvey Vision Conference*, pages 147–151, 1988.

[HS92] Berthold K. P. Horn and Brian G. Schunck. Determining optical flow. pages 389–407, 1992.

[HSH07] Hui-Yu Huang, Weir-Sheng Shih, and Wen-Hsing Hsu. A film classifier based on low-level visual features. pages 465–468, Oct. 2007.

[HVB06] Rudra N. Hota, Vijendran Venkoparao, and Saad Bedros. Face detection by using skin color model based on one class classifier. In *ICIT '06: Proceedings of the 9th International Conference on Information Technology*, pages 15–16, Washington, DC, USA, 2006. IEEE Computer Society.

[HWS08] Panpan Huang, Yunhong Wang, and Ming Shao. A new method for multi-view face clustering in video sequence. *Data Mining Workshops, International Conference on*, 0:869–873, 2008.

[hY09] Ming hsuan Yang. *Encyclopedia of Biometrics*, chapter : Face Detection. Springer, July 2009.

[hYKMA02] Ming hsuan Yang, David J. Kriegman, Senior Member, and Narendra Ahuja. Detecting faces in images: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24:34–58, 2002.

[IF01] S. Ioffe and D.A. Forsyth. Human tracking with mixtures of trees. In *ICCV01*, pages 690–695, 2001.

[Jaf05] Gaël Jaffré. *Indexation de la vidéo par le costume* . Thèse de doctorat, Université Paul Sabatier, Toulouse, France, novembre 2005.

[JBPKQ07] Philippe Joly, Jenny Benois-Pineau, Ewa Kijak, and Georges Quénot. The ARGOS campaign: Evaluation of Video Analysis Tools. *Signal Processing: Image Communication*, 22(7-8):705–717, 2007.

[KGG+03] E. Kijak, G. Gravier, P. Gros, L. Oisel, and F. Bimbot. Hmm based structuring of tennis videos using visual and audio cues. volume 3, pages III – 309–12 vol.3, july 2003.

[Kit87] Genshiro Kitagawa. Non-gaussian state-space modeling of nonstationary time series. *Journal of the American Statistical Association*, 82(400):1032–1041, 1987.

[KKP07] M. Kyperountas, C. Kotropoulos, and I. Pitas. Enhanced eigen-audioframes for audiovisual scene change detection. *Multimedia, IEEE Transactions on*, 9(4):785–797, June 2007.

[KL51] S. Kullback and R. A. Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, 1951.

[KMP00] Erich Peter Klement, Radko Mesiar, and Endre Pap. *Triangular Norms*. Springer, 1 edition, 2000.

[KNM+09] K. Kumar, J. Navratil, E. Marcheret, V. Libal, G. Ramaswamy, and G. Potamianos. Audio-visual speech synchronization detection using a bimodal linear prediction model. In *Computer Vision and Pattern Recognition Workshops, 2009. CVPR Workshops 2009. IEEE Computer Society Conference on*, pages 53–59, June 2009.

[Lac09] H. Lachambre. *Caractérisation de l'environnement musical dans les documents audiovisuels*. Thèse de doctorat, Université de Toulouse, Toulouse, France, decembre 2009.

[LAMA05] Charay Lerdsudwichai, Mohamed Abdel-Mottaleb, and A-Nasser Ansari. Tracking multiple people with recovery from partial and total occlusion. *Pattern Recognition*, 38(7):1059 – 1070, 2005.

[LAOP09] H. Lachambre, R. André-Obrecht, and J. Pinquier. Singing Voice Detection in Monophonic and Polyphonic Contexts (regular paper). In *European*

Signal and Image Processing Conference (EUSIPCO), Glasgow, Scotland, UK, 24/08/09-28/08/09, pages 1344–1348, http://www.eurasip.org/, 2009. EURASIP. http://www.eusipco2009.org/.

[Lap03] I. Lapidot. Som as likelihood estimator for speaker clustering. In *Eurospeech-2003*, pages 3001–3004, 2003.

[Law80] k. Laws. *Textured image segmentation.* PhD thesis, University of California, 1980.

[LBP95] T. K. Leung, M. C. Burl, and P. Perona. Finding faces in cluttered scenes using random labeled graph matching. In *ICCV '95: Proceedings of the Fifth International Conference on Computer Vision*, page 637, Washington, DC, USA, 1995. IEEE Computer Society.

[LGZ+07] Zhu Liu, David Gibbon, Eric Zavesky, Behzad Shahraray, and Patrick Haffner. A Fast, Comprehensive Shot Boundary Determination System. In *IEEE International Conference on Multimedia and Expo*, pages 1487–1490, Beijing, China, July 2007.

[LJH03] P. Li, J. Jiang, and P.N. Hashimah. Dominant color extraction in dct domain. *International Conference on Visual Information Engineering*, pages 258–261, July 2003.

[LKCC07] Jiann-Shu Lee, Yung-Ming Kuo, Pau-Choo Chung, and E-Liang Chen. Naked image detection based on adaptive and extensible skin color model. *Pattern Recogn.*, 40(8):2261–2270, 2007.

[LNK03] Y. Li, S.S. Narayanan, and C.C.J. Kuo. *Movie Content Analysis, Indexing and Skimming Via Multimodal Information*, page Chapter 5. 2003.

[LNK04a] Ying Li, S. Narayanan, and C.C.J. Kuo. Content-based movie analysis and indexing based on audiovisual cues. *Circuits and Systems for Video Technology, IEEE Transactions on*, 14(8):1073–1085, Aug. 2004.

[LNK04b] Ying Li, Shrikanth S. Narayanan, and C. C. Jay Kuo. Adaptive speaker identification with audiovisual cues for movie content analysis. *Pattern Recognition Letters*, 25(7):777 – 791, 2004. Video Computing.

[Low04] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60:91–110, 2004.

[LP07] Jong-Seok Lee and Cheol Hoon Park. Temporal filtering of visual speech for audio-visual speech recognition in acoustically and visually challenging environments. In *ICMI '07: Proceedings of the 9th international conference on Multimodal interfaces*, pages 220–227, New York, NY, USA, 2007. ACM.

[LPE99] Rainer Lienhart, Silvia Pfeiffer, and Wolfgang Effelsberg. Scene determination based on video and audio features. *Multimedia Computing and Systems, International Conference on*, 1:9685, 1999.

[LS08] Yi-Chun Liao and Ming-Hong Syu. An actor-based video segmentation system using visual and audio information in e-learning. In *Intelligent Systems Design and Applications, 2008. ISDA '08. Eighth International Conference on*, volume 3, pages 575–580, Nov. 2008.

[LTEA95] A. Lanitis, C. J. Taylor, T. Ecootes, and T. Ahmed. Automatic interpretation of human faces and hand gestures using flexible models. In *In International Workshop on Automatic Face- and Gesture-Recognition*, pages 98–103, 1995.

[LW01] Zhu Liu and Yao Wang. Major cast detection in video using both audio and visual information. *Acoustics, Speech, and Signal Processing, IEEE International Conference on*, 3:1413–1416, 2001.

[LW07] Zhu Liu and Yao Wang. Major cast detection in video using both speaker and face information. *IEEE Transactions on Multimedia*, 9(1):89–101, 2007.

[MB98] A.M Martinez and R. Benavente. The ar face database. Technical report, CVC Technical Report, 1998.

[MBI01] S. Meignier, J-F Bonastre, and S. Igounet. E-hmm approach for learning and adapting sound models for speaker indexing. In *in Proc. Odyssey Speaker and Language Recognition Workshop*, pages 175–180, 2001.

[MC98] C. Montacie and M.J. Caraty. A silence/noise/music/speech splitting algorithm. In *International Coference on Spoken Language Processing*, pages 1579–1582, 1998.

[Men42] K. Menger. Statistical metrics. *Natural Academy Science*, pages 535–537, 1942.

[Mer76] P. Mermelstein. Distance measures for speech recognition: Psychological and instrumental. In C. H. Chen, editor, *Pattern Recognition and Artificial Intelligence*, pages 374–388. Academic Press, New York, 1976.

[MJV+07] G.. Monaci, P.. Jost, P.. Vandergheynst, B.. Mailhe, S.. Lesage, and R. Gribonval. Learning multimodal dictionaries. *Image Processing, IEEE Transactions on*, 16(9):2272–2283, Sept. 2007.

[MM96] B. S. Manjunath and W.Y. Ma. Texture features for browsing and retrieval of image data. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI - Special issue on Digital Libraries)*, 18(8):837–42, Aug 1996.

[MOB06] Antonio Micilotta, Eng-Jon Ong, and Richard Bowden. Real-time upper body detection and 3D pose estimation in monoscopic images. In *ECCV 2006: Proceedings of the European Conference on Computer Vision*, pages 139–150, 2006.

[Mor79] Hans Moravec. Visual mapping by a robot rover. In *Proceedings of the 6th International Joint Conference on Artificial Intelligence*, pages 599–601, 1979.

[MrOVY01] B. S. Manjunath, Jens rainer Ohm, Vinod V. Vasudevan, and Akio Yamada. Color and texture descriptors. *IEEE Transactions on Circuits and Systems for Video Technology*, 11:703–715, 2001.

[NA99] M. Nishida and Y. Ariki. Speaker indexing for news articles, debates and drama in broadcasted tv programs. In *ICMCS '99: Proceedings of the IEEE International Conference on Multimedia Computing and Systems Volume II-Volume 2*, page 466, Washington, DC, USA, 1999. IEEE Computer Society.

[NLFL03] A.V. Nefian, L.H. Liang, T.Y. Fu, and X.X. Liu. A bayesian approach to audio-visual speaker identification. pages 761–769, 2003.

[NPAA97] Frederick K. D. Nahm, Amelie Perret, David G. Amaral, and Thomas D. Albright. How do monkeys look at faces? *J. Cognitive Neuroscience*, 9(5):611–623, 1997.

[OB02] R.J. OCallaghan and D.R. Bull. Improved illumination-invariant descriptors for robust colour object recognition. volume 4, pages IV–3393–IV–3396 vol.4, 2002.

[Pal08] Palanivel. Multimodal person authentication using speech, face and visual speech. *Computer Vision and Image Understanding*, 109(1):44 – 55, 2008.

[Pel90] Eli Peli. Contrast in complex images. *Journal of the Optical Society of America*, 7:2032–2040, 1990.

[Per92] P. Perona. Steerable-scalable kernels for edge detection and junction analysis. pages 3–18, 1992.

[PL08] Jiang Peng and Qin Xiao Lin. Automatic classification video for person indexing. *Image and Signal Processing, Congress on*, 2:475–479, 2008.

[PNLM04] Gerasimos Potamianos, Chalapathy Neti, Juergen Luettin, and Iain Matthews. Audio-visual automatic speech recognition: An overview. In *Issues in Visual and Audio-visual Speech Processing*. MIT Press, 2004.

[POP98a] C. P. Papageorgiou, M. Oren, and T. Poggio. 1998, a general framework for object detection. *Computer Vision, 1998. Sixth International Conference on*, pages 555–562, 1998.

[POP98b] C. P. Papageorgiou, M. Oren, and T. Poggio. A general framework for object detection. In *Computer Vision, 1998. Sixth International Conference on*, pages 555–562, 1998.

[PPJ06] Jeremy Philippeau, Julien Pinquier, and Philippe Joly. Intervenant Classification in an Audiovisual Document. In *International Conference on Signal Processing and Multimedia Applications (poster session) (SIGMAP), Setùbal, Portugal, 07/08/06-10/08/06*, pages 185–188, http://www.insticc.net/, 2006. INSTICC Press. http://www.sigmap.org/SIGMAP2006, ISBN: 972-8865-64-3.

[PPJC08] Jeremy Philippeau, Julien Pinquier, Philippe Joly, and Jean Carrive. Dynamic organization of audiovisual database using a user-defined similarity measure

based on low-level features. In *IEEE International Conference on Image Processing (ICIP), San Diego, California, U.S.A., 12/10/08-15/10/08*, pages 33–36, http://www.ieee.org/, 2008. IEEE. http://www.icip08.org/.

[PRAO03] J. Pinquier, J.L. Rouas, and R. André-Obrecht. A fusion study in speech/music classification. In *International conference on Acoustics, Speech and Signal Processing*, volume II, pages 17–20. IEEE, April 2003.

[RBK98] Henry A. Rowley, Shumeet Baluja, and Takeo Kanade. Neural network-based face detection. *IEEE Transactions On Pattern Analysis and Machine intelligence*, 20:23–38, 1998.

[RcG05] Amit K. Roy-chowdhury and Himanshu Gupta. 3d face modeling from monocular video sequences. In *In Face Processing: Advanced Modeling and Methods*. Academic Press, 2005.

[RFPAO05] J-L Rouas, J. Farinas, F. Pellegrino, and R. André-Obrecht. Rhythmic unit extraction and modelling for automatic language identification . *Speech Communication*, 47(4):436–456, 2005.

[Ris89] J. Rissanen. *Stochastic Complexity in Statistical Inquiry Theory*. World Scientific Publishing Co., Inc., River Edge, NJ, USA, 1989.

[RKP+07] B. Rosenhahn, U. Kersting, K. Powell, T. Brox, and H. P. Seidel. Tracking clothed people. In *Human Motion - Understanding, Modeling, Capture, and Animation*. Springer, 2007.

[Roy97] Deb K. Roy. Speaker indexing using neural network clustering of vowel spectra. *International Journal of Speech Technology*, 1:2–143, 1997.

[RSC+98] D. A. Reynolds, E. Singer, B. A. Carlson, G. C. OÕLeary, J. J. McLaughlin, and M. A. Zissman. Blind clustering of speech utterances based on speaker and language characteristics. In *International Conference on Spoken Language Processing*, 1998.

[RSJU07] Reede Ren, Punitha Puttu Swamy, Joemon M. Jose, and Jana Urban. Attention-based video summarisation in rushes collection. In *TVS '07:*

Proceedings of the international workshop on TRECVID video summarization, pages 89–93, New York, NY, USA, 2007. ACM.

[SA04] Koichi Sato and J. K. Aggarwal. Temporal spatio-velocity transform and its application to tracking and interaction. *Comput. Vis. Image Underst.*, 96(2):100–128, 2004.

[SC00a] Malcolm Slaney and Michele Covell. Facesync: A linear operator for measuring synchronization of video facial images and audio tracks. In *NIPS*, pages 814–820, 2000.

[SC00b] H. Sundaram and Shih-Fu Chang. Video scene segmentation using video and audio features. In *Multimedia and Expo, 2000. ICME 2000. 2000 IEEE International Conference on*, volume 2, pages 1145–1148 vol.2, 2000.

[SC03] P. Smaragdis and M. Casey. Audio/visual independent components. In *Proceedings of the 4th International Symposium on Independent Component Analysis and Blind Signal Separation*, pages 709–714, April 2003.

[SFA01] P. Sivakumaran, J. Fortuna, and A.M. Ariyaeeinia. On the use of the bayesian information criterion in multiple speaker detection. In *he 7th European Conference on Speech Communication and Technology (Eurospeech'01)*, pages 795–798, 2001.

[SG00] Chris Stauffer and W. Eric L. Grimson. Learning patterns of activity using real-time tracking. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(8):747–757, 2000.

[SH94] F.S. Samaria and A. Harter. Parameterisation of a stochastic model for human face identification. In *WACV94*, pages 138–142, 1994.

[SJRS97] M.A. Siegler, U. Jain, B. Raj, and R.M. Stern. Automatic segmentation, classification and clustering of broadcast news audio. In *Proc. DARPA Speech Recognition Workshop*, pages 97–99. Morgan Kaufmann, 1997.

[SKK08] J.W. Sung, T. Kanade, and D.J. Kim. Pose robust face tracking by combining active appearance models and cylinder head models. 80(2):xx–yy, November 2008.

[SL98] C. Saraceno and R. Leonardi. Identification of story units in audio-visual sequences by joint audio and video processing. In *Image Processing, 1998. ICIP 98. Proceedings. 1998 International Conference on*, volume 1, pages 363–367 vol.1, Oct 1998.

[SNK99] S. Satoh, Y. Nakamura, and T. Kanade. Name-it: Naming and detecting faces in news videos. *IEEE MultiMedia*, 6(1):22–35, 1999.

[SO95] M. Stricker and M. Orengo. *Similarity of Color Images*, volume 2, pages 381–392. 1995.

[SOD09] Alan F. Smeaton, Paul Over, and Aiden R. Doherty. Video shot boundary detection: Seven years of trecvid activity. *Computer Vision and Image Understanding*, 2009.

[SS04] Y. SUZUKI and T. SHIBATA. An edge-based face detection algorithm robust against illumination, focus, and scale variations. In *EUSIPCO 2004: XII. European Signal Processing Conference*, pages 2279–2282, 2004.

[SSGALMBC06] Sancho Salcedo-Sanz, Ascensión Gallardo-Antolín, José M. Leiva-Murillo, and Carlos Bousoño-Calzón. Offline speaker segmentation using genetic algorithms and mutual information. *IEEE Trans. Evolutionary Computation*, 10(2):175–186, 2006.

[TCCW05] W. H. Tsai, S. S. Cheng, Y. H. Chao, and H. M. Wang. Clustering speech utterances by speaker using eigenvoice-motivated vector space model. In *Proc. of the International Conference on Acoustics, Speech, and Signal Processing*, pages 725–728, 2005.

[TDV00] Ba Tu Truong, Chitra Dorai, and Svetha Venkatesh. New enhancements to cut, fade, and dissolve detection processes in video segmentation. In *MULTIMEDIA '00: Proceedings of the eighth ACM international conference on Multimedia*, pages 219–227, New York, NY, USA, 2000. ACM.

[TG99] A. Tritschler and R. Gopinath. Improved speaker segmentation and segments clustering using the bayesian information criterion. volume 2, pages 679–682, 1999.

[TK91] Carlo Tomasi and Takeo Kanade. Detection and tracking of point features. Technical report, Carnegie Mellon University, April 1991.

[TMN07] A. Temko, D. Macho, and C. Nadeu. Enhanced svm training for robust speech activity detection. In *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, volume 4, pages IV–1025–IV–1028, April 2007.

[TP98] S. Tsekeridou and I. Pitas. Speaker dependent video indexing based on audio-visual interaction. In *Image Processing, 1998. ICIP 98. Proceedings. 1998 International Conference on*, volume 1, pages 358–362 vol.1, Oct 1998.

[TP99] S. Tsekeridou and I. Pitas. Audio-visual content analysis for content-based video indexing. In *in Proc. of 1999 IEEE Int. Conf. on Multimedia Computing and Systems*, pages 667–672. IEEE, IEEE, 1999.

[TPC09] F. Talantzis, A. Pnevmatikakis, and A. G. Constantinides. Audio−visual active speaker tracking in cluttered indoors environments. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 39(1):7–15, Feb. 2009.

[TW93] D. Terzopoulos and K. Waters. Analysis and synthesis of facial image sequences using physical and anatomical models. *IEEE Trans. Pattern Anal. Mach. Intell.*, 15(6):569–579, 1993.

[TW07] W. H. Tsai and H. M. Wang. Speaker clustering based on minimum rand index. In *Proc. of the International Conference on Acoustics, Speech, and Signal Processing*, pages 485–488, 2007.

[VCR03] M. Vescovi, M. Cettolo, and R. Rizzi. A dp algorithm for speaker change detection. In *Proceedings of the 8th European Conference on Speech Communication and Technology*, pages 2997–3000, 2003.

[VJ01] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features, 2001.

[VJ04] Paul Viola and Michael Jones. Robust real-time face detection. *International Journal of Computer Vision*, 57:137–154, 2004.

[VJS03] Paul Viola, Michael J. Jones, and Daniel Snow. Detecting pedestrians using patterns of motion and appearance. In *ICCV '03: Proceedings of the Ninth IEEE International Conference on Computer Vision*, page 734, Washington, DC, USA, 2003. IEEE Computer Society.

[VRB01] C. J. Veenman, M. J. T. Reinders, and E. Backer. Resolving motion correspondence for densely moving points. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23:54–72, 2001.

[VSA03] Vladimir Vezhnevets, Vassili Sazonov, and Alla Andreeva. A survey on pixel-based skin color detection techniques. In *in Proc. Graphicon-2003*, pages 85–92, 2003.

[WDV+03] Hualu Wang, Ajay Divakaran, Anthony Vetro, Shih-Fu Chang, and Huifang Sun. Survey of compressed-domain features used in audio-visual indexing and analysis. *J. Visual Communication and Image Representation*, 14(2):150–183, 2003.

[YD06] Yaser Yacoob and Larry S. Davis. Detection and analysis of hair. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(7):1164–1169, 2006.

[YH94] Gaungzheng Yang and Thomas S. Huang. Human face detection in a complex background. *Pattern Recognition*, 27:53–63, 1994.

[YR08] Chuohao Yeo and Kannan Ramchandran. Compressed domain video processing of meetings for activity estimation in dominance classification and slide transition detection. Technical Report UCB/EECS-2008-79, EECS Department, University of California, Berkeley, Jun 2008.

[YS04] Alper Yilmaz and Mubarak Shah. Contour-based object tracking with occlusion handling in video acquired using mobile cameras. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26:1531–1536, 2004.

[ZBLG08] X. Zhu, C. Barras, L. Lamel, and J-L. Gauvain. Multi-stage speaker diarization for conference and lecture meetings. pages 533–542, 2008.

[ZBMG05] X. Zhu, C. Barras, S. Meignier, and J.L. Gauvain. Combining speaker identification and bic for speaker diarization. In *Interspeech*, pages 2441–2444, 2005.

[ZH05] B. Zhou and J.H.L. Hansen. Efficient audio stream segmentation via the combined $t2$ statistic and the bayesian information criterion. *IEEE Trans. Speech Audio Processing*, 13:467–474, 2005.

[ZN05] J. Zdansky and J. Nouza. Detection of acoustic change-points in audio records via global bic maximization and dynamic programming. In *INTERSPEECH-2005*, pages 669–672, 2005.

Résumé

Cette thèse consiste à proposer une méthode de caractérisation non-supervisée des intervenants dans les documents audiovisuels, en exploitant des données liées à leur apparence physique et à leur voix. De manière générale, les méthodes d'identification automatique, que ce soit en vidéo ou en audio, nécessitent une quantité importante de connaissances a priori sur le contenu. Dans ce travail, le but est d'étudier les deux modes de façon corrélée et d'exploiter leur propriété respective de manière collaborative et robuste, afin de produire un résultat fiable aussi indépendant que possible de toute connaissance a priori.

Plus particulièrement, nous avons étudié les caractéristiques du flux audio et nous avons proposé plusieurs méthodes pour la segmentation et le regroupement en locuteurs que nous avons évalué dans le cadre d'une campagne d'évaluation.

Ensuite, nous avons mené une étude approfondie sur les descripteurs visuels (visage, costume) qui nous ont servi à proposer de nouvelles approches pour la détection, le suivi et le regroupement des personnes.

Enfin, le travail s'est focalisé sur la fusion des données audio et vidéo en proposant une approche basée sur le calcul d'une matrice de cooccurrence qui nous a permis d'établir une association entre l'index audio et l'index vidéo et d'effectuer leur correction. Nous pouvons ainsi produire un modèle audiovisuel dynamique des intervenants.

Mots-clés: Diarization, Fusion audiovisuel, Segmentation en locuteurs, Regroupement en locuteurs, Détection des visages, Regroupement des visages, Extraction du costume, GLR-BIC segmentation.

Abstract

This thesis consists to propose a method for an unsupervised characterization of persons within audiovisual documents, by exploring the data related for their physical appearance and their voice. From a general manner, the automatic recognition methods, either in video or audio, need a huge amount of a priori knowledge about their content. In this work, the goal is to study

the two modes in a correlated way and to explore their properties in a collaborative and robust way, in order to produce a reliable result as independent as possible from any a priori knowledge.

More particularly, we have studied the characteristics of the audio stream and we have proposed many methods for speaker segmentation and clustering and that we have evaluated in a french competition.

Then, we have carried a deep study on visual descriptors (face, clothing) that helped us to propose novel approches for detecting, tracking, and clustering of people within the document.

Finally, the work was focused on the audiovisual fusion by proposing a method based on computing the cooccurrence matrix that allowed us to establish an association between audio and video indexes, and to correct them. That will enable us to produce a dynamic audiovisual model for each speaker.

Keywords: Diarization, Audiovisual fusion, Speaker segmentation, Speaker clustering, Face detection, Face clustering, Clothing extraction, GLR-BIC segmentation.