INSTITUTO SUPERIOR TÉCNICO
Universidade Técnica de Lisboa

Automatic Audiovisual Summarization for Generic Content

Design, Implementation and Evaluation

NUNO FREDERICO COSTA FERREIRA DE MATOS

Dissertação para obtenção do Grau de Mestre em

ENGENHARIA DE REDES DE COMUNICAÇÕES

Júri

Presidente: Prof. Luís Eduardo Teixeira Rodrigues

Orientador: Prof. Fernando Manuel Bernardo Pereira

Vogal: Prof. Joaquim Armando Pires Jorge

Setembro de 2007

Acknowledgments

My first words go to Inês, my Mother and my Father, the most special figures in my life and to whom I deeply thank for the endless patience and infinite support.

I would also like to thank Prof. Fernando Pereira, for his guidance, devotion and dedication to this work. His immense knowledge, the ability to motivate the ones that work with him and his remarkable good humor, even when things were not going so well, characterize him as a unique reference and the cornerstone of this work. I truly believe that I could not have had a better advisor.

A word to João Ascenso, from the IT Image Group, for his availability and help on technical issues when I needed the most.

I also want to dedicate a very special word to all my close group of friends, namely to Guido Varatojo, Pedro Guerra e Pedro Xavier, who give me the honor to be my friends.

Last but not least, a profound word of appreciation to all my family in the name of my grandmother, for her strength and singularity.

A deep “Obrigado” to all of the above mentioned as well as to everyone close to me, which, in their own way, contributed for the development of this Thesis.

Abstract

Nowadays, with the explosion of multimedia content availability, the capability to be selective regarding its consumption increases in importance. Audiovisual content is no longer brought to us only by the television, being available in many other ways, like Personal Video Recorders and Video On Demand systems, on each one's Personal Computer and, obviously, on the Internet. The exponential growth of websites like *YouTube* shows that people give great relevance to audiovisual content in these days. Moreover, common people can easily produce, store, distribute and view those contents as no specialized skills are required to do so. To look for a specific content in any of these systems can be a long, painful task. To manually summarize videos for the user's own purposes, as showing a summary of his/her vacations to friends and family also takes much time and it is a complex task. As people's time is getting more precious and scarce every day, an application capable of saving the time spent in these tasks by automatically summarizing audiovisual generic content looks like something very useful and promising.

Motivated by these situation and factors, this report describes the motivations for the development of this solution, its architecture as well as the entire process for summarization designed and implemented in the course of this work.

To evaluate the quality of the created summaries, a user evaluation study was also conducted with encouraging results, showing that the developed application is able, with relative success, to summarize audiovisual generic content.

The major novelties of this work regards the affective approach to the problem by trying to model the viewers' excitement, or arousal, while watching a video and the deep user evaluation study conducted to evaluate the system's performance, performed in a unusual scale for an automatic summarization solution.

Keywords: Automatic audiovisual summarization; Generic content; Arousal; Motion intensity; Shot cut density; Sound energy.

Resumo

Hoje em dia, com a explosão da quantidade de conteúdos audiovisuais disponíveis, a escolha do que consumir revela-se cada vez mais crítica e importante. Os conteúdos audiovisuais já não chegam até nós apenas através da televisão, estando agora também disponíveis através de outros sistemas, como sistemas do tipo *Personal Video Recorder* ou *Video On Demand*, através do computador de cada um e, claro, através da Internet. O crescimento exponencial de websites como o *YouTube* mostra que as pessoas atribuem cada vez mais importância aos conteúdos audiovisuais nos dias que correm. Para além deste facto, a produção, armazenamento e distribuição de conteúdos audiovisuais, assim como o seu visionamento, está cada vez mais banalizada já que não são precisos conhecimentos específicos, nomeadamente técnicos, para o fazer. Procurar um conteúdo específico em qualquer dos sistemas acima referidos pode ser uma tarefa longa e 'dolorosa'. Sumariar vídeos manualmente para os propósitos do utilizador, como por exemplo, mostrar um sumário das férias a amigos e família, demora normalmente muito tempo e pode ser uma tarefa de elevada complexidade. Como o tempo de cada um é cada vez mais um recurso precioso e escasso, uma aplicação capaz de poupar tempo em cada uma dessas tarefas, sumariando automaticamente conteúdos audiovisuais genéricos, surge como muito útil e prometedora.

Motivado por estes factores, este relatório descreve a solução desenvolvida para sumariação audiovisual automática para conteúdos genéricos, as suas motivações, a sua arquitectura assim como todo o processo para sumariação desenhado e implementado durante este trabalho.

Para avaliar a qualidade dos sumários criados, foi também desenvolvido um estudo de avaliação subjectiva do desempenho, usando um número significativo de sujeitos, tendo-se obtido resultados encorajadores, mostrando que a aplicação desenvolvida é capaz de, com relativo sucesso, sumariar conteúdos audiovisuais genéricos.

As principais inovações deste trabalho estão relacionadas com a abordagem afectiva ao problema, através da tentativa de modelar a excitabilidade/estímulo dos espectadores enquanto vêem o vídeo e com o estudo de avaliação subjectiva levado a cabo para avaliar o desempenho do sistema, que foi feito numa escala pouco usual para sistemas de sumariação automática.

Palavras-chave: Sumariação audiovisual automática; Conteúdos genéricos; Excitabilidade/Estímulo; Intensidade de movimento; Densidade de cortes de cena; Energia de som.

Table of Contents

Index of Figures

Index of Tables

List of Acronyms

DTD Document Type Definition
GOP Group of Pictures
GUI Graphic User Interface
IDE Internal Development Environment
MBR Minimum Bounding Rectangle
MPEG Moving Pictures Experts Group
MVF Motion Vector Field
NTSC National Television System Committee
PAL Phase Alternating Line
PCM Pulse-Code Modulation
PDA Personal Digital Assistant
PVR Personal Video Recorder
RIFF Resource Interchange File Format
VoD Video-on-Demand
XML eXtended Markup Language

Chapter 1

Context and Objectives

This chapter intends to present the scope and objectives of this project, after providing its motivation and context. Finally, the structure of this report is presented.

1.1 Motivation

With the recent explosion of multimedia content availability, the selective consumption of audiovisual content has been increasing in importance. Audiovisual content is no longer brought to us only by the television, as happened for many decades, but it is also available to the world from an endless number of systems, notably the personal computer, the Internet, Personal Video Recorders (PVR), Video on Demand (VoD) systems, mobile networks, among others. All these systems store or stream audiovisual content in a multimedia format, often rather big in size (this means in number of bits) and duration, and usually there are massive collections of contents available to the users located in any part of the world.

One of the distinctive features of audiovisual content usage is that, in these days, it is not required having any specialized skills at all to produce, process, store, distribute or view this type of content. Common people can make videos from their vacations or special moments with their personal camcorders and store them in their personal computers, or use them on a daily basis on VoD and PVR systems in their homes, and frequently use the Internet to make their own videos available to the world or browse for videos of their interest. One screaming example of this later case is the fantastic boom of *YouTube* [1], in recent years, with millions of new video additions per day, from people all over the world. Figure 1 which was taken from Alexa's website [2], an Internet traffic analyzer website, shows the growth of *YouTube* in the past three years, compared with other popular websites such as the online encyclopedia *Wikipedia* [3] or the well-known MSN's Web portal [4]. *YouTube* represents today almost 3% of all daily page views

across the Internet, arising from near 0% at the beginning of 2006. This illustrates that the time spent by people consuming audiovisual contents is increasing at a very fast pace.

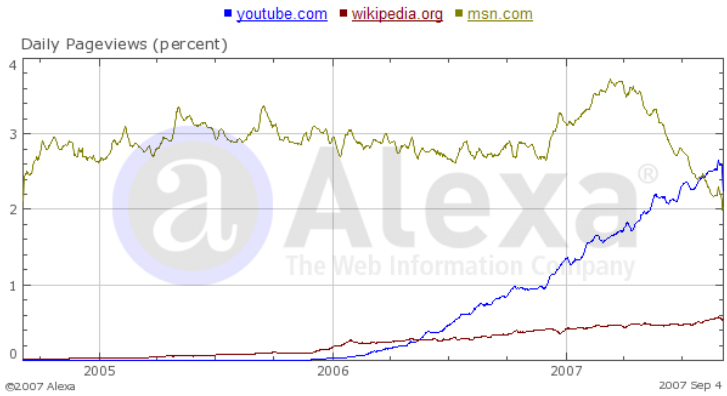

Figure 1 – YouTube's growth in the past 3 years in relation to msn.com and Wikipedia [2].

However, the huge amount of available data is also a problem since everybody has a limited viewing time and thus it is not only important to find quickly the audiovisual material one is looking for but it is sometimes also important to filter from that material the more relevant or exciting parts, especially if the content is long and has many less relevant parts. In this context, imagine, for example, an application capable of allowing someone to produce a video summary of his/her vacations to family and friends, or to automatically produce trailers or teasers of movies for a VoD system, or even to extract the highlights of a football match or a Formula 1 race to include them as reports in a TV news show. These examples present some critical usages of automatic audiovisual summarization, justifying the development of tools capable of successfully automatically identifying and filtering the most exciting moments from a content asset and include them in a summary.

Also manual browsing in looking for a specific audiovisual content is, in the majority of these systems, a common task performed by its users. Searching for a specific birthday video of someone in special, from a personal collection, stored in the personal computer, or choosing a movie in a VoD system, are not rare tasks performed by people in its every day life. The main problem of this kind of tasks is the great amount of time and effort that has to be spent by the user to be successful in his/her search. Audiovisual contents take time to be available to and consumed by the user and, in the majority of the cases, the desired video is found after browsing/viewing many contents which proved to be of low interest. Automatically summarizing audiovisual content may allow the user to spend much less time in browsing tasks, as summaries are smaller files, in size and duration, and therefore take less time to become available and to be consumed by the user who can infer the relevance of the entire content, deciding afterwards if it suits his/her wishes.

All this motivates automatic audiovisual summarization. The utility of automatic summarization in the above mentioned systems and situations proves to be rather wide and powerful, motivating the work designed, implemented and evaluated in this Thesis which objectives are explained next.

1.2 Objectives

Currently, many approaches to the automatic audiovisual summarization problem have been studied and proposed; major approach regard the type of content addressed, notably specific content versus generic content. The utility of solutions addressing only specific content, for example football or basketball matches, is obviously more limited regarding applications dealing with more generic content even if at the cost of some summarization performance. Therefore, this Thesis targets the development of an automatic summarization solution for generic audiovisual content.

The main objectives of this Thesis are to design, implement and evaluate an application capable of, based on some input audiovisual content, of any kind, producing summaries with the most interesting and exciting events occurred in that content in a simple and intuitive way to the user.

To do so for generic content, the design of the summarization application was based on modeling the user excitement, in this Thesis named as *arousal,* felt by the viewer along the content, including in the summary the segments which provoke more excitement in the viewer. This arousal modeling approach allows any content to be summarized regardless of its kind, origin, etc. assuring, in this manner, the genericity of the approach and thus the utility for a wide range of applications.

1.3 Report structure

This Thesis is organized in seven chapters, including this first one introducing the Thesis, and the seventh one referring to the conclusions and future work.

Chapter 2 presents a detailed review on the audiovisual summarization problem as well as the main technologies and systems already developed to address the problem. First, it presents a classification tree for audiovisual summarization solutions, structuring the problem and reviewing after in detail four different solutions which have been considered more representative and relevant.

Chapter 3 introduces the solution developed in this Thesis, by presenting its architecture and a functional description of each of its modules.

Chapter 4 presents an in-depth description of all architecture's modules, notably the processing algorithms implemented, in order to allow the reader to get a complete understanding of the entire process proposed for the automatic summarization.

Chapter 5 is dedicated to the application developed, presenting an implementation's overview with a brief description of the application's implementation class diagram, the software frameworks and libraries used as well as a detailed Installation guide and application's Graphic User Interface guide. This chapter intends to provide sufficient information for a proper and easy usage of the summarization application by any user.

Chapter 6 presents the conditions, methodology, and results of the subjective tests carried out to evaluate the proposed summarization solution.

Finally, Chapter 7 is dedicated to the conclusions and eventual future work.

Chapter 2

Reviewing Technologies for Audiovisual Summarization

This chapter has the objective to provide an overview on technologies for audiovisual summarization. With that purpose in mind, it is first proposed a way to organize the various types of approaches to solve the summarization problem and after some specific, most relevant, solutions are briefly reviewed.

2.1 Structuring the audiovisual summarization problem

As for the majority of technical problems, the various ways to address the audiovisual summarization problem can be clustered and classified depending on the approach, concepts and tools used. Figure 2 shows the proposed structuring and classification tree for solutions addressing the audiovisual summarization problem.

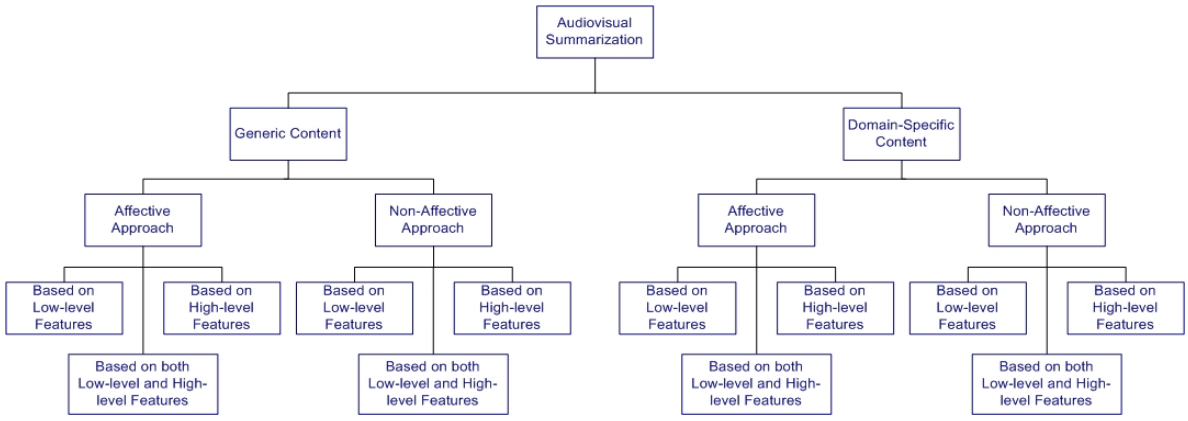

Figure 2 – Family tree of solutions for the audiovisual summarization problem.

As shown on Figure 2 the dimensions adopted to organize and classify the technologies and solutions for audiovisual summarization are:

1. Generic versus specific content solutions
2. Affective based versus non-affective based solutions
3. Low-level features based versus high-level features based solutions

The meaning behind these dimensions is explained in the following. There are other ways to organize the same technologies, very likely as good as the one proposed here; what is important here is to get one good broad view of this technical field with this view presented in a structured, organized way and not just as a simple list of solutions. As can be seen in the following, the three dimensions presented above can be fully combined creating a variety of technical approaches as described in the following,

1) Generic content approach to audiovisual summarization

The main difference between the two main families of solutions relies on the target type of audiovisual content. The generic approach is designed to have the ability to produce summaries from any type of video content while the specific content approach addresses some specific type of content, e.g. football broadcasts or news programs.

1.1) Affective based summarization solutions

In the context of generic content solutions, it is important to distinguish between affective and non-affective solutions. Audiovisual summarization or highlighting is one of many possible applications of affective video content analysis. These solutions are characterized by the usage, as filtering criteria for the presence of a video segment in the summary, of a certain amount of feelings or emotions provoked on the viewer or a certain amount of attention that the viewer is expected to dedicate to that part of the video. Therefore, affective audiovisual summarization is performed after a process of affect or attention modeling. This can be done based on low-level or high-level audiovisual features or even on a hybrid solution, which is the most common approach as both low-level and high-level information are important to model human reactions to video content.

Concluding, affective summarization can be described as a solution to audiovisual summarization that uses expected human reactions to the input video to decide which segments should and should not be included in the final summary.

1.1.1) Low-level features based solutions

Low-level features based summarization solutions are those which are based on low-level information automatically extracted from the audiovisual segments. The main difference among the various low-level features based models is what is done with the low-level information extracted. In affective summarization solutions, low-level information is typically used to model affection or attention. This type of solution is commonly chosen to explore affective summarization and can be based on metadata standards or not (this means the extracted features may correspond to certain descriptors in a metadata standard or not, e.g. MPEG-7). An example of this type of approach is the system developed by A. Hanjalic [5][6][7], described in the next section which models the level of excitement that a video produces in the human viewer based only on low-level features as sound energy or motion vectors extracted from the video, producing afterwards the desired final summary.

Standard based approaches, in low-level or high-level features based solutions, are meant to provide some degree of interoperability between systems whilst non-standard based approaches are typically used in proprietary solutions with more specific or local objectives.

1.1.2) High-level features based solutions

In a generic content context, it is difficult to build an effective audiovisual summarization solution based only on high-level features as they are mainly used to describe events in a domain-specific content context. In an affective generic content summarization approach, high-level features are usually used to build models to aid affect modeling as, for example, face detection models or speech pitch inference models.

1.1.3) Combined low-level and high-level features based solutions

This type of hybrid approaches are those that theoretically can achieve better results following an affective generic content summarization solution as they combine the potentialities and benefits of both low-level and high-level features in modeling human reactions to video. To effectively model human reactions or, in a more abstract way, affect, low-level and high-level features based models are built, with low-level features based models normally performing the core role of the solution. High-level features based models tend to have the mission to improve the efficiency of the modeling, allowing best final results in affect modeling and, therefore, in the audiovisual summary.

An example of a combined approach is described in the next section [8]. This solution uses low-level features as sound energy, motion vectors, color and texture as well as auxiliary high-level features based models such as a face detection model and a camera motion model. This last model produces information on how much do camera movements influence the viewers' attention to a video to precisely model the expected human attention to a video.

1.2) Non-affective based summarization solutions

Concerning generic content approaches, non-affective summarization solutions are developed in fewer number that affective summarization solutions as it is more complicated to generate summaries for any type of audiovisual content without modeling the expected human reaction when viewing the audiovisual content. Therefore, this type of approach may adopt a rather diversified position, e.g. using only low-level information, only high-level information or both in a combined manner from the video to detect which segments should be present in the summarized version.

1.2.1) Low-level features based solutions

The main difference between this approach and the analogous one described above for affective summarization is that here the extracted low-level features are not used to model any kind of human emotion or feelings when reacting to a video. What is typically done in this case is to include in the summary all segments that have, for example, a sound energy value or a density of cuts higher than a pre-determined threshold value; the other segments are left behind. As in affective solutions, low-level features can be standard based or not standard based for the same sort of reasons.

1.2.2) High-level features based solutions

As for affective summarization, a non-affective generic content solution based only on high-level features is not usual. For the same reasons, high-level features based models can hardly decide which segments should or

should not be included in the final audiovisual summary all by themselves in a generic content context. In contrast, in a domain-specific content context, high-level features can have a more important role, as will be shown in the following.

1.2.3) Combined low-level and high-level features based solutions

In a non-affective generic content context, combined solutions are also those that can achieve better final results as they produce summaries based not only on low-level but also on high-level features. These approaches are similar to those described for affective summarization with the difference that now both low-level and high-level features are not used to model any kind of human reaction but are used to directly decide which segments will be present in the summary.

2) Domain-specific content approach to audiovisual summarization

Domain-specific content approaches differ from generic approaches as they generate summaries from a specific type of video input. The main objective of domain-specific content approaches is to model events to afterwards include the most important ones in the summary. Most of the solutions studied have the final goal of creating summaries from a specific type of sport event such as American football, football, basketball, etc. or a specific genre of movie. Domain-specific content approaches can also be divided into affective and non-affective based solutions.

2.1) Affective based summarization solutions

The affective approach is also possible in domain-specific content solutions and some systems were developed but in smaller numbers than non-affective-based solutions. Affective-based solutions are developed in a similar manner as for generic content approaches. The main difference is that in domain-specific content approaches, it is not only the viewers' excitement or attention that is modeled but also which events correspond to which type of viewers' excitement. For example, in a football domain-specific approach, different types of user's excitement can correspond to a goal, a free-kick or a harsh tackle among other kinds of events. Affective-based summarization solutions in domain-specific content approaches can present better results than in generic content solutions but tend to neglect all the capabilities of affective summarization, as it's potentialities are far more exploited when it is applied to generic content videos. One of the main appeals of affective summarization is its potential ability to create summaries from any type of input audiovisual content.

2.1.1) Low-level features based solutions

Low-level features based approaches in an affective domain-specific content context and the ones described in affective generic content context differ mainly in one aspect. This distinguishing factor is that in domain-specific content approaches low-level features are used to model expected human reactions with the objective of modeling domain-specific events, such as an ace or a match point in a tennis match, for example. A. Hanjalic describes in [9] a solution based on an affective approach to produce summaries for football content; this solution is a simpler version of the solution proposed by the same author and referred before to produce summaries for generic content [5][6][7].

2.1.2) High-level features based solutions

In this case, the difference between a high-level features based approach and a low-level features based approach is in the type of features used, as both approaches use features to model the expected human reactions and, afterwards, to model events, according to the reactions obtained. High-level features based approaches can

use, for example, a slow-motion detection model or a shot classification model to decide which segments are or are not interesting for the viewers.

2.1.3) Combined low-level and high-level features based solutions

Low-level and high-level features based approaches in the affective domain-specific content context do not bring substantial differences regarding the equivalent solutions addressing generic content. Normally, the solutions are simpler as they know the type of input video; as for generic content, these solutions use both low-level and high-level features to model the expected human reactions to the video. As the low-level and high-level based solutions described immediately before, the human reactions modeling is done to link verified human reactions to the domain-specific events, e.g. a peak in the viewer's excitement while watching a football match may be linked with a goal.

2.2) Non-affective based summarization solutions

Contrary to the generic content approach, non-affective-based solutions are more common to address domain-specific content and can be based on low-level features, high-level features or both.

2.2.1) Low-level features based solutions

Low-level features based models in non-affective domain-specific content solutions are characterized by the extraction of low-level information from the video as motion or sound energy to model events. After event modeling, the more important events are chosen to be included in the summary.

2.2.2) High-level features based solutions

High-level features as slow-motion detection or shot classification are also used in many systems to model events. In a football match, for instance, a goal can be modeled as a close-up shot followed by a view of the crowd and a slow-motion replay. As for low-level features based solutions, after modeling the events, the more important ones are included in the summary.

2.2.3) Combined low-level and high-level features based solutions

A combined approach here is not much different from the two approaches described before. These solutions gather both low-level and high-level features to model domain-specific events. Using the same example, a football goal, it can be modeled as a peak of sound energy, the crowd roar, followed by a high density of shot cuts and, for instance, a slow-motion replay segment. In this manner, the solution uses both low-level and high-level features from a video to model events and, afterwards, to produce effective audiovisual summaries. Of course, sometimes the edge between low-level and high-level features is not that sharp but there is clearly an evolution regarding the single usage of rather simple low-level features.

2.2 Most relevant audiovisual summarization systems

In order to make this review on audiovisual summarization technologies more complete and useful for the reader, four systems will be reviewed next. Each system shows a different approach on addressing the audiovisual summarization problem; they have been selected due to their conceptual richness and capability in addressing the problem at hand in this Thesis: generic audiovisual content summarization. They will be presented from the simpler domain-specific content solution to the, conceptually more challenging and complex, generic content solutions.

2.2.1 A domain-specific football video summarization system using MPEG-7 metadata

The first system described here was developed by James, Echigo, Teraguchi and Satoh [10] and proposes a framework for generating personal video summaries based on metadata, this means based on extracted features. In the proposed classification tree presented before, this system fits under the domain-specific content, non-affective branch, since it performs audiovisual summarization based on high-level features, in this case standard high-level descriptors according to the MPEG-7 standard.

2.2.1.1 Objectives

The objective of this system is to effectively summarize football matches in a personalized way using available high-level metadata. This objective is constrained by three main requirements highlighted by the proposers of this system: limited resources available, summaries based on personal preferences, and availability of semantic metadata.

Multimedia users normally have limited resources in terms of the devices they have access to and, even more important, in terms of the available time to consume the content. A viewing device, for instance, can be a small PDA with limited computational and bandwidth resources or a desktop connected to high speed bandwidth networks and with considerable computational resources. Personal preferences also play a major role as the users' needs for personalized information, this means to get what they want and the way they want, increases every day. In this case, it is assumed there is semantic metadata available which becomes a core characteristic of this system. The MPEG-7 standard greatly facilitates the management of metadata as it can contribute for the explosion of metadata availability by providing easy interoperability. As semantic information has great importance, a large part of the MPEG-7 descriptors used in this system have semantic capabilities.

2.2.1.2 Approach and architecture

The main objective of many audiovisual summarization approaches is to create an overview of the entire video contents. This system differs in three ways:

1. The goal is to select only the most relevant sections of content.
2. The proposers aim to create individual, instead of general, summaries depending on personal preferences.
3. The system makes use of detailed, available event metadata.

This system uses the high-level semantic features in the available metadata and uses supervised learning to create personalized digests this means personalized summaries. Figure 3 shows the system architecture.

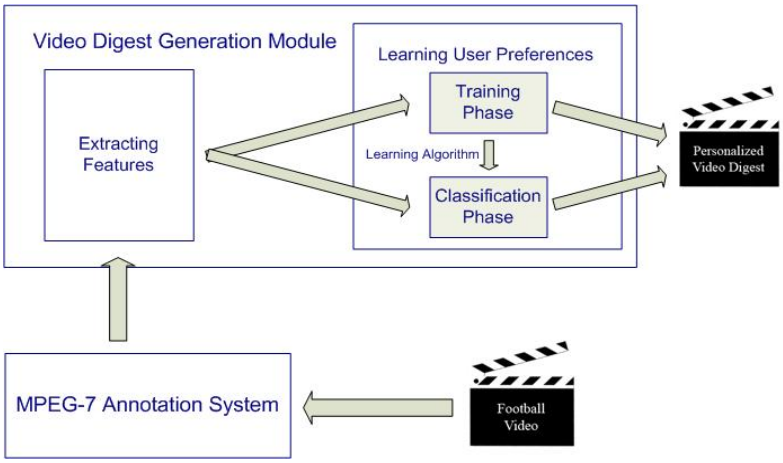

Figure 3 – System architecture.

A short walkthrough of the system is presented next:

1. **Extracting features** - An *activity window* is defined by a time interval and features that semantically represent the importance of the events are extracted in the *activity window*.
2. **Training phase** - Then, a user labels the highlights in a set of training videos, generating a set of relevant and non-relevant segments. This phase is not realized for every created summary and it is only done with the purpose of constructing a consistent supervised learning algorithm for each user.
3. **Classification phase** - Given new metadata from a video, in this case a football video, segments can be classified as relevant or non-relevant by a classifier that is constructed by a supervised learning algorithm aided by the set of training videos already labeled by the user.
4. **Creation of summary** - The relevant game segments are included in the summary.

2.2.1.3 Main tools

The main modules in this system are the MPEG-7 Annotation System and the Video Digest Generation Module, as can be seen in Figure 3. These two modules will be briefly described in the following:

- **MPEG-7 annotation system** - The MPEG-7 Annotation System is used by an operator to manually input metadata for a certain video. The system contains a set of football-related event labels, such as "goal", "free-kick", "yellow card" and "shoot block"; each annotation consist of an event label and a time stamp. The objective of this module is to maximize the annotations' utility while minimizing the operator's load of work. This manual process is clearly not the ideal one since it requires time and effort but this situation is still the most common and thus the summarization system here presented takes benefit of this.
- **Video digest generation module** - The metadata is then used to create the personalized video digests. The Video Digest Generation Module is divided in two main sub-modules:
  - **Extracting features** - The Feature Extraction sub-module performs the extraction of high-level semantic features from the available MPEG-7 metadata. Considering a user profile as a list of event-importance pairs, a video S can be described as:

    $S=\{(label_1,timestamp_1,importance\text{-}weight_1),(label_2,timestamp_2,importance\text{-}weight_2),\ldots,(label_n,timestamp_n,importance\text{-}weight_n)\}$

    In this context, each video sample corresponds to an event-importance pair with a label, a timestamp and an importance-weight based on the user profile. A discrete-time system can be used to map the available metadata taken as input to an event-importance metadata output. This highlight extraction approach defines a window over a time interval as shown in Figure 4:

    $W_1=\{(goal,13:12:05,10),(corner\text{-}kick,13:11:50,9),\ldots,(goal,44:05:04,10)\}$.

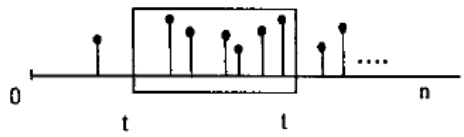

Figure 4 – Event-importance metadata signal [10].

Within each window, features that are related to important domain-specific semantic information are extracted. Those features are:

- Number of events inside the window: m

- Number of location events: $\sum_{i=0}^{i<m} Le_i$ where $Le_i = 1$ if e_i is a location event and $Le_i = 0$ otherwise
- Number of match interruptions: $\sum_{i=0}^{i<m} Ie_i$ where $Ie_i = 1$ if e_i is a location event and $Ie_i = 0$ otherwise
- Number of defensive events: $\sum_{i=0}^{i<m} De_i$ where $De_i = 1$ if e_i is a location event and $De_i = 0$ otherwise
- Number of offensive events: $\sum_{i=0}^{i<m} Oe_i$ where $Oe_i = 1$ if e_i is a location event and $Oe_i = 0$ otherwise
- Play time: $\sum_{i=0}^{i<m} t(e_i)$ where $t(e_i)$ is the function that returns the time duration for which the ball is in play for the event e_i.

- **Learning user preferences** – This sub-module has two main phases: a training phase and a classification phase described in the following:
  - **Training phase -** In the training phase, a user watches a set of videos and tags the events he/she is interested with a "yes"; untagged events are marked as negative segments. A training set is then generated, by extracting the features related to the user-interesting metadata signals. In the training set, the window examples containing events which the user marked as interesting are marked as "positive examples" while the remaining are marked as "negative examples". As mentioned above, this phase does not exist in every summary creation process. The training set is used by a supervised learning algorithm to make a classifier capable to mark highlight preferences for a user or a group of users.
  - **Classification phase** - In the classification phase, metadata from a new video is input into the system. The same features that were extracted in the training phase are extracted from the new metadata signal and then the leaning algorithm is applied. The classifier's output is a set of segments, each of them classified as positive, and therefore a highlight, or negative. Only positive segments are included in the personalized video digest.

2.2.1.4 Performance

This system was tested with six different football matches. While one of the system proposers collected the matches and used the MPEG-7 annotation system to perform their annotations, another of the proposers watched the six football matches and personally selected the highlights. Table 1 shows some characteristics of the tested videos, notably their duration, number of events detected and average number of events per minute.

GAME	TIME (mins)	No. events	Avg. / min.
FUKUOKAvs.ICHIHARA	101	1,800	18
JAPAN vs. JAMAICA	97	2,041	21
JAPAN vs. PARAGUAY	101	1,798	18
JAPANvs.YUGOSLAVIA	100	1,820	18
JAPAN vs. CROATIA	93	1,581	17
ITALYvs.FRANCE	96	1,890	19
TOTAL	588	10,930	18

Table 1 – Data summary for the six football games tested [10].

The features mentioned in the previous section were extracted from each of the video's metadata using overlapping windows and three training sets were created from the resultant database. The training sets are presented in Table 2: the columns show the percentage of positive (containing interesting events) windows which are referred as Training Positive and the percentage of Training Negative that relate to the opposite case; the last column presents the number of window samples of each training set.

Set	Description	Training Pos.	Training Neg.	Samples
A	All highlights	50%	50%	3156
B	67% of the data	50%	50%	1986
C	All examples	14%	86%	10917

Table 2 – Description of the training sets [10].

The goal of the tests was to compare the selection of highlights made by humans to highlights automatically selected using the system proposed with the various supervised learning algorithms from [11] using different window sizes. The window size depends on the viewing device; the viewing devices, in this case, were Personal Digital Assistants (PDAs) and cellular phones. A window size of 16 seconds was used as it was found to be suitable for generating short video summaries with enough detail for those devices.

The metrics used to evaluate the summarization performance were the so-called Precision and Recall. Precision is the ratio of the number of positive window samples obtained to the total number of positive and negative window samples obtained. Recall is the ratio of the number of positive window samples obtained to the total number of positive window samples in the test sets. Precision is commonly referred as the purity of retrieval whilst Recall is referred as the completeness of retrieval.

Three experiments were made. The first one with the aim of understanding how a human would select highlights only from given metadata and the second and third experiments to measure the system's performance.

The first experiment was then made to evaluate how well a human would do on selecting highlights only from the available metadata. This experiment was done on Set C of the training sets. The results were 89% Precision and 16% Recall as it is shown in the last row of Table 3. This experiment intended to show the difficulty of selecting highlights from given metadata, even to humans.

The second experiment was performed using supervised learning algorithms with 16 seconds windows and cross-validation [12]. Cross-validation, is the statistical practice of partitioning a sample of data into subsets such that the analysis is initially performed on a single subset, while the other subset(s) are retained for subsequent use in confirming and validating the initial analysis. The initial subset of data is called the training set; the other subset(s) are called validation or testing sets. In this case 90% of the data was used as training sets, distributed equally between positive and negative training examples.

On Set B the results were 80% precision and 85% recall. The complete set of results is shown on Table 3.

Set	A	A	B	B	C	C
Method	Prec.	Recall	Prec.	Recall	Prec.	Recall
1-Nearest N.	78%	69%	72%	80%	54%	36%
3-Nearest N.	80%	77%	80%	80%	60%	35%
Neural N.	79%	83%	**80%**	**85%**	63%	40%
Human	-	-	-	-	89%	16%

Table 3 – 10-fold cross-validation performance results on training sets using 16 seconds windows and various learning algorithms [10].

In the third experiments were used 4 games (67%) of the set on training and the remaining 2 games (33%) on testing and was used a Neural Network [13] classifier. The results were 44% Precision and 75% Recall as it is shown in Table 4. This set of results when compared with the one obtained using cross-validation shows that larger training sets present better results. The use of a supervised learning algorithm also improves performance although the algorithm depends on the viewing device. For this case, where PDAs and mobile phones were used, a Neural Network proved to be suitable as it provides faster classification speed mainly when compared with a Nearest Neighbor approach.

Training Set	B	B
Algorithm	Precision	Recall
1-Nearest Neighbor	19%	87%
5-Nearest Neighbor	42%	74%
NN	44%	75%

Table 4 – Independent test set results [10].

2.2.1.5 Summary

This approach is an example of a domain-specific, non-affective approach based on the extraction of high-level features from available event metadata; these features are after used in a supervised learning context. After a training phase where a user selects his/her personal highlights from a set of training videos, features are extracted from the training set. Those features are used by a classifier that chooses, using the event metadata from a new video, which segments will be included in the digest according to the user's preferences. The system's overall performance, when tested with PDAs and mobile phones, is satisfactory, mainly with larger training sets, as Table 3 shows with a top result of 80% for Precision and 85% for Recall.

2.2.2 An automatic football video analysis and summarization system

Following the approach of describing systems from the simpler to the more complex ones, the second system reviewed in this chapter is a domain-specific content audiovisual summarization solution developed by Ekin, Tekalp, Mehrotra [14]. This system proposes an effective manner of summarizing football matches based on both low-level and high-level features; therefore, in the proposed classification tree, fits under the domain-specific, non-affective branch and under the non-affective approaches is labeled as a hybrid solution. Its hybrid approach, considering both high-level and low-level features constitutes a major difference in relation to the previous described system that only relies on high-level features.

2.2.2.1 Objectives

The objective of the system is to achieve an efficient, effective and fully automatic framework for analysis and summarization of football videos using object-based and cinematic features. Cinematic features are extracted from usual video production and composition rules as shot types or replays which are typically slow motion segments. Object-based features are, as expected, based on objects. Objects can be described by their spatial (e.g. color, texture, shape) and spatio-temporal features (object motion and interactions). Object-based features allow a high-level domain analysis but, on the other hand, bring to the table a much more computationally costly summarization process for real-time implementation. Cinematic features offer a good tradeoff between final results and computational cost.

The system uses both low-level and high-level football videos processing algorithms. Under low-level algorithms, we find dominant color region detection, robust shot boundary detection and shot classification algorithms. High-level goal detection, referee detection and penalty-box detection algorithms were also implemented. Applying these algorithms, three types of summaries can be produced:

i) all slow-motion segments in a match;

ii) all goals in a match, and

iii) slow-motion segments classified according to object-based features.

The last type of summaries can contain higher-level semantics, as referee or penalty-box detection, whilst the first two types use only cinematic features in its process, with the main objective of a faster process.

The system aims to be computationally efficient as there is no need to use object-based features when cinematic features are sufficient to detect main events, such as goals in football. It also aims to be effective, allowing an increase of accuracy by employing object-based features when needed, at the expense, of course, of more computational complexity.

2.2.2.2 Approach and architecture

The proposers of this system present a framework for automatic and real-time football video analysis and summarization using cinematic and object-based features. The flowchart of the developed framework is shown in Figure 5.

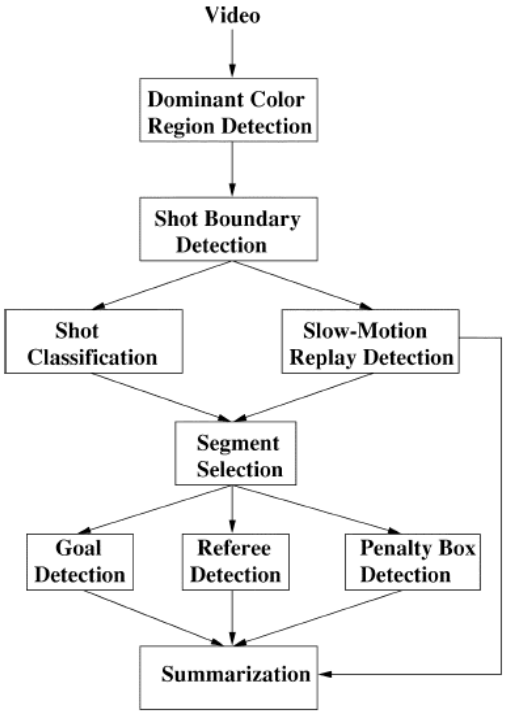

Figure 5 – System architecture [14].

A short walkthrough of the system is presented next:

1. **Low-level analysis for cinematic features extraction algorithms** – The first two steps regard the detection of dominant color regions and shot boundaries. Dominant color region detection and shot boundary detection algorithms are proposed that aim to be robust to variations in the dominant color. The grass field color can vary from stadium to stadium and between different times of the day. These color variations are automatically captured by the dominant color region detection algorithm. Dominant color variation along a game, due to lighting or shadows variations are also predicted by automatic adaptation to local statistics. The following steps are shot classification and slow-motion replay detection, both regarding cinematic features. The two proposed algorithms aim to provide robustness to variations in cinematic features, which can occur by different cinematic styles, resulting from different producers and production crews.

2. **Football event and object detection algorithms** – Three high-level algorithms are proposed to detect in football broadcasts: i) goals; ii) referee; and iii) penalty boxes. Goal detection is based only on cinematic features that result from broadcasting common rules employed by producers after goal events to provide a better experience for the viewer. Referee detection is made by distinguishing referee's jersey color from the rest of match participants. As for penalty-box detection, a three-parallel-line rule is used that identifies the penalty-box in a football field.
3. **Summarization and presentation process –** At last, an effective and efficient framework for summarization, combining all the previous algorithms is proposed. As mentioned before, this solution is efficient because it doesn't compute object-based features when cinematic features are sufficient and it is effective because it can use object-based features when more accuracy in the summary is required.

All referred algorithms are described in detail in the next section.

2.2.2.3 Main tools

The algorithms described in this section will be included in three categories, for better understanding. Robust dominant color region detection, shot boundary detection, shot classification and slow-motion replay detection will be placed under *Low-level analysis for cinematic features extraction algorithms* while goal, referee and penalty-box detection algorithms will be described in *Football event and object detection algorithms;* finally, summarization process tools are described in *Summarization and adaptation of parameters*.

- **Low-level analysis for cinematic features extraction algorithms** – The first algorithm to be explained is the robust dominant color region detection algorithm since the remaining algorithms need an efficient detection of the football field in each frame (made using the dominant color).
    - **Robust dominant color region detection** – The tone of green of a football field grass constitutes the dominant color, varying from stadium to stadium and according to lightning and weather conditions. Consequently, the proposers do not assume any specific value for the field color. The only assumption made is the existence of a unique dominant color, in the Hue-Saturation-Intensity (HSI) space. The statistics of this dominant color are acquired by the system at the beginning of the process and updated through time in order to adapt to the above mentioned variations. The dominant color is identified by the mean value of each color component. Color components are computed around their histogram peaks. This process includes determining the peak index for each histogram, which can be obtained from one or more frames. Then, an interval satisfying the conditions below is defined. The conditions define the minimum and maximum indexes as the smallest and largest indexes to the left and right of the peak with a predefined number of pixels. K is a constant and intends to establish a minimum number of pixels. The proposers chose a percentage of 20% of the peak count for the minimum number of pixels and, therefore, K = 0.2.

$$H[i_{\min}] \geq K * H[i_{peak}] \quad (1)$$

$$H[i_{\min} - 1] < K * H[i_{peak}] \quad (2)$$

$$H[i_{\max}] \geq K * H[i_{peak}] \quad (3)$$

$$H[i_{\max} + 1] < K * H[i_{peak}] \quad (4)$$

$$i_{\min} \leq i_{peak} \quad (5)$$

$$i_{\max} \geq i_{peak} \quad (6)$$

The mean color is then calculated in the detected interval, for each color component as shown in the equation below. Q_{size} is the quantization size and is used to convert the index to a color value.

$$ColorMean = \frac{\sum_{i=i\min}^{i\max} H[i] * i}{\sum_{i=i\min}^{i\max} H[i]} * Q_{size} \quad (7)$$

The field color detection in each frame is made by finding the difference of each pixel to the mean color using the robust cylindric metric [15]. As the algorithm works in the HSI space, achromaticity, i.e. the lack of hue and saturation, has to be taken into account. If saturation and intensity means are in the achromatic region, only intensity is computed, using (8) the top equation in the group below for achromatic pixels; otherwise, the two first equations are used for chromatic pixels in each frame.

$$d_{\text{int}ensity}(j) = \left| I_j - \bar{I} \right| \quad (8)$$

$$d_{chroma}(j) = \sqrt{(S_j)^2 + (\bar{S})^2 - 2S_j S\cos(\Theta(j))} \quad (9)$$

$$d_{cylindrical}(j) = \sqrt{(d_{\text{int}ensity}(j))^2 + (d_{chroma}(j))^2} \quad (10)$$

$$\Theta(j) = \begin{cases} \Delta(j) & if \leq 180^{\circ} \\ 360^{\circ} - \Delta(j) & otherwise \end{cases} \quad (11)$$

$$\Delta(j) = \left| \overline{Hue} - Hue_j \right| \quad (12)$$

Hue, S and I refer, obviously to Hue, Saturation and Intensity while j refers to the j_{th} pixel and Δ to the dominant color value. θ is defined in the fourth equation presented in the figure. Concluding the field region is defined as the pixels having:

$$d_{cylindrical} < T_{color} \quad (13)$$

T_{color} is a predefined threshold value and can be adjusted from video to video.

- **Shot boundary detection –** In the algorithm developed by the proposers for shot boundary detection, the first new feature considered was the absolute difference between two frames in their ratios of dominant, or grass, colored pixels to total number of pixels, which was represented by the variable G_d. G_d between i^{th} and k^{th} frames computation is given by:

$$G_d(i,k) = | G_i - G_{i-k} | \quad (14)$$

G_i is the ratio of the grass colored pixel in the i^{th} frame. The second feature is the difference in color histogram similarity, presented as H_d, computed by:

$$H_d(i,k) = | HI(i,k) - HI(i-k,k) | \quad (15)$$

HI(i, k) stands for histogram intersection and represents the similarity between two histograms in the i^{th} and $(i\text{-}k)^{th}$ frames. HI(i, k) is computed as below:

$$HI(i,k) = \frac{1}{N}\sum_{m-1}^{N}\sum_{j=0}^{B_m-1}\min\left(H_i^m[j], H_{i-k}^m[j]\right) \quad (16)$$

N is the number of color components, three in this case, R,G and B; B_m is the number of bins in the histogram of the m^{th} color component; and H_i^m represents the normalized histogram of the i^{th} frame for the same color component.

Different values of k are used to detect hard cuts or gradual transitions. Hard cuts are detected with k = 1, while a range of k values with k > 1 is used to detect gradual transitions. An upper bound of 5 was determined for k. A shot boundary is detected by comparing G_d and H_d with a set of thresholds. Thresholds for H_d are adaptive. When a sports video shot corresponds to a close-up or to a out of the field view, the number of field color pixels will be very low and, therefore, shot properties will be similar to a generic video shot. In this case, the proposers defend the use of only H_d with a high threshold. When the field is visible, both G_d and H_d are used, and H_d is used with a lower threshold. In this manner, four thresholds were defined by the proposers: T_h^{low}, T_h^{high}, T_g and $T_{lowgrass}$. As their names imply, the first two represent the low and high thresholds for H_d, while T_g is the threshold for G_d. $T_{lowgrass}$ is an estimate of the low grass ratio and determines when the conditions vary from a field view to a close-up view. When grass colored pixel ratio in the i^{th} frame is lower than $T_{lowgrass,}$ H_d is compared with T_h^{high}; otherwise it is compared with T_h^{low}.

- **Shot classification –** Football shots can be classified, according to the proposers, into three different classes:
    1. Long Shot – A long shot represents the global view of the field, as shown in Figure 6 (a) and (b). A long shot aims to accurately localize events on the field.
    2. In-Field Medium Shot – A medium shot is usually a shot where a whole human body is visible such as Figure 6 (c) and (d) illustrate. A group of consecutive medium shots normally implies a game-break in football, while a single isolated medium shot usually is a part of a game play.
    3. Close-Up and Out of Field Shot – A close-up shot shows a person above his/her waist-line and, generally, indicates a break in the game. Figure 6 (e) shows an example of a close-up shot. Out of field shots are exemplified in Figure 6 (f) and also often indicate a game break. The audience, the coaches or other shots are included in out of field shots.

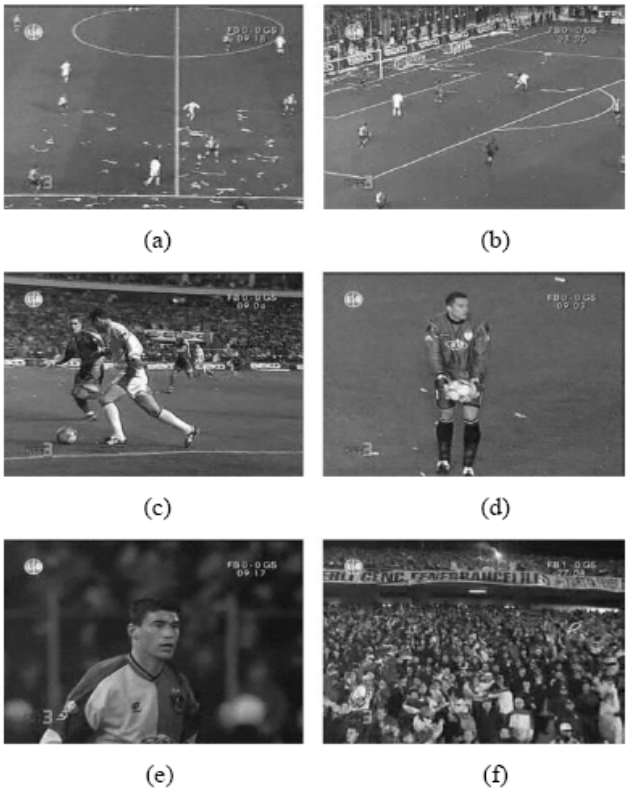

Figure 6 – View types in football: (a),(b) long views; (c),(d) in-field medium views; (e) close-up view; and (f) out of field view [14].

Classification of each shot into one of the three classes is done based on spatial features. The proposed algorithm classifies each frame in a shot into a class and assigns the shot class to the label of the majority of frames. To decide as which shot class a frame should be labeled, the frame grass colored pixel ratio, G, is computed. When a frame has a low G value, it is labeled as a close-up or out of field shot. For frames with a high G value, a cinematographic algorithm is used. Regions are defined using the Golden Section spatial decomposition rule [16][17], which suggests dividing up the screen in a 3:5:3 proportion in both directions, and positioning the main subjects on the intersection points of these lines. The rule was revised by the proposers for football videos and instead of dividing the whole frame, only the grass region box was divided. The grass region box is defined as the minimum bounding rectangle, or as scaled version of it, of grass colored pixels. An example of Golden Section spatial decomposition is shown in Figure 7.

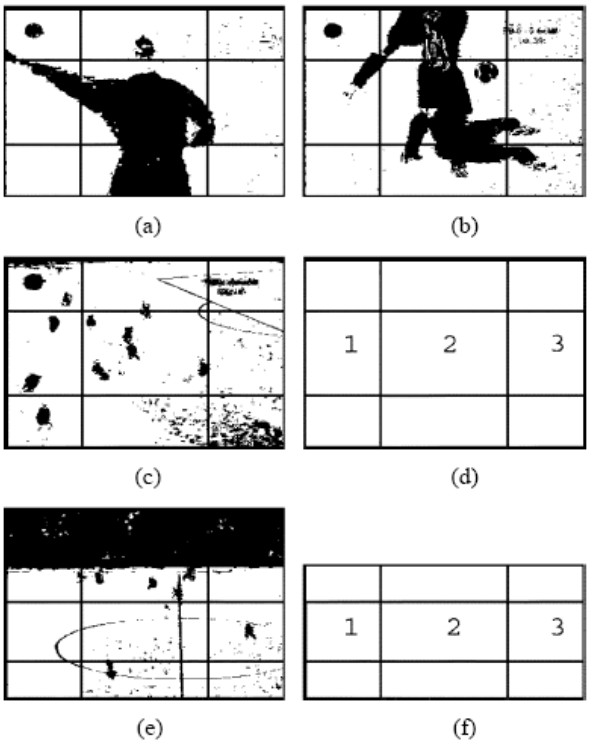

Figure 7 – Examples of Golden Section spatial composition in (a), (b) medium and (c-e) long views, the resulting grass region boxes and the region are shown in (d) and (f) for (a-c) and (e), respectively. [14]

A flowchart of the algorithm is shown in Figure 8.

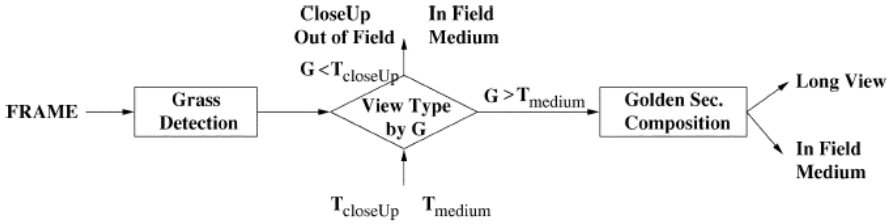

Figure 8 – Flowchart of the shot type classification algorithm [14].

First of all, G and two thresholds, $T_{closeUp}$ and T_{medium}, are used to determine the frame view label. The thresholds are initialized as 0.1 and 0.4, respectively, at the start of the system, and are updated to the minimum of the grass colored pixel ratio, G, histogram as the system collects more data. When $G > T_{medium,}$ golden section composition is applied to decide if the frame view is a long view or an in field medium view.

- **Slow-motion replay detection –** In this algorithm, as the only objective of the proposers was to determine if a shot consists in a slow-motion segment or not, they used the zero crossing measure proposed in [18]. Zero crossing measure evaluates the amplitude of the fluctuations in the frame differences – D(t) values – within a window of length L.

- **Football event and object detection algorithms** – Detection of certain objects events and objects in a football match allows more semantically rich summaries. As goals are the most important event in football, a goal detector algorithm was developed which is the first to be explained here. Controversial calls by the referee, such

as a red card or a penalty call, involving close-up shots of referees and plays inside the penalty box are also relevant for summarization. Therefore, referee and penalty-box detection algorithms were also developed.

- **Goal detection –** According to the proposers, a goal in a football broadcast is generally followed by a special pattern of cinematic features. In football, a goal leads to a game break and, during the break, the broadcast producers usually show one or more close-up views of the scorer and goal-keeper as well as shots of the audience followed by one or two slow-motion replays for a better visual experience. After the close-up views and slow-motion replays, usually a long shot represents the restart of the game. According to these premises, the authors defined a cinematic template to represent a goal event in football broadcasts:
    1. Duration of the break: a game break provoked by a goal event should last more than 30 seconds and less than 120 seconds.
    2. The occurrence of, at least, one close-up/out of field shot: It could be either a close-up of a player or an out of field view of the audience.
    3. The existence of at least one slow-motion replay shot: A goal event is always replayed at least once.
    4. The relative position of the replay shot: The slow-motion replay(s) always follow the close-up/out of field shots.
- **Referee detection –** Referees in football matches wear different colors from the rest of the actors on the field. Therefore, the proposers decided to use their dominant color region detection algorithm described before to detect referee regions in medium or out of field/close-up shots. They assume a single referee in this type of shots. Considering this, horizontal and vertical projections of the feature pixels are used to detect the referee region, named by the proposers as Minimum Bounding Rectangle (MBR_{ref}). The peak of both vertical and horizontal projections and the spread around the peaks are used to compute the rectangle parameters surrounding the MBR_{ref}. MBR_{ref} coordinates are defined to be the first projection coordinates at both sides of the peak index without enough pixels. The decision of the existence or not of a referee in a frame is based in the following shape descriptors:
    1. The ratio of MBR_{ref} area to the frame area: a low value indicates that the current frame does not contain a referee.
    2. MBR_{ref} aspect ratio – width/height: frames with MBR_{ref} aspect ratio values outside (0.2, 1.8) are discarded**.**
    3. Feature pixel ratio in the MBR_{ref}: This feature approximates the compactness of the MBR_{ref} ; higher referee pixel ratios are favored.
    4. The ratio of the number of feature pixels in the MBR_{ref} to that of the outside: It measures the correctness of the single referee assumption; if the ratio is low, it means that this assumption is not verified and the frame is discarded.
- **Penalty-box detection –** The proposers decided to reduce the problem of penalty-box detection to the search of three parallel lines in a frame as the penalty-box is easily identified by the parallels goal-line and horizontal goal-area and the penalty-area lines, as shown on Figure 9.

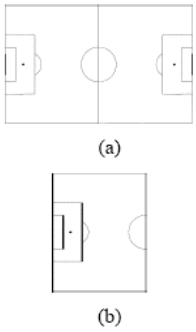

Figure 9 – (a) Football field model and (b) three highlighted parallel lines around the goal area [14].

To detect the three lines, the grass detection result described before in dominant color region detection algorithm section is used. To limit the operating region to the field pixels, a mask image from the grass colored pixels is computed, shown in Figure 10 (b). The mask is obtained by computing a scaled version of the grass MBR, illustrated on the same figure, and afterwards by including all field region that have enough pixels inside the computed rectangle. To detect line pixels, an edge response defined as the pixel response to the 3×3 Laplacian mask defined below is used:

$$h = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

The three parallel lines are then detected by using a Hough transform that employs size, distance and parallelism constraints. The detected three lines of the penalty-box shown in Figure 10 (a) are shown in Figure 10 (f).

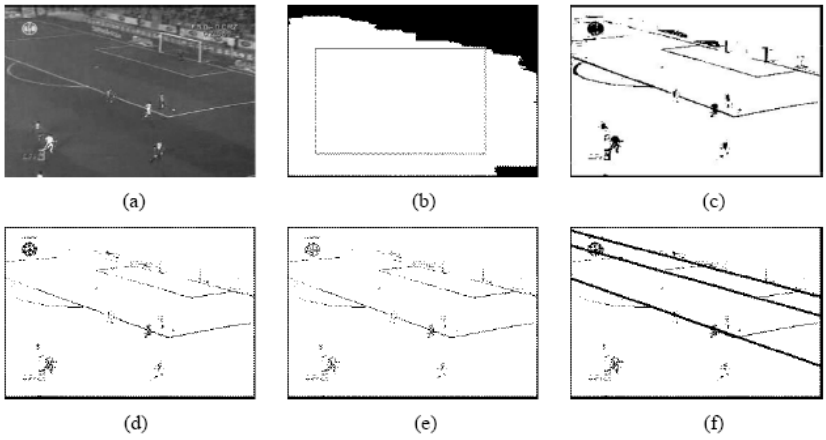

Figure 10 – Penalty box detection, (a) input frame (b) field mask (c) grass/non-grass image in the field region (d) the pixels in (c) with high gradient (e) image after thinning and (f) the three detected lines [14].

- **Summarization and presentation** – Three types of summaries are proposed: i) all slow-motion segments in a game; ii) all goal events in a game; and iii) the extension of the two with object-based features. The first two types are based on cinematic features and generated in real-time, while the third also uses the referee and penalty-box detection results.
    1. Slow-motion summaries consist of slow-motion replay shots and are generated by using the shot boundary, shot class and slow-motion replay frames. Slow-motion summaries may also consist of all shots in a pre-

defined time window defined by the user around each replay or they can include the closest long shot before each replay in the summary as the closest shot usually corresponds to action in normal motion.

2. Goal summaries consist, of course, of goal events. As described before, goals are detected in a cinematic template and, therefore, goal summaries consist of shots in the above explained template.
3. Object-based summaries are based on determining if a slow-motion shot involves the referee and/or the penalty-box. For this, the segments of interest are selected. The segments of interest include close-up shots around the corresponding slow-motion replay for referee detection, and one or more long shots before the replay for penalty-box detection. Object-based summaries include those shots with the detected object, in addition to the slow-motion replay(s).

2.2.2.4 Performance

Results for all algorithms described in the above section were exhaustively presented by the proposers. As this work's focus is video summarization, only high-level analysis and summarization results will be described here. All algorithms were tested on a data set of more than 800 minutes of football video. 17 MPEG-1 clips, 16 of which with 352×240 spatial resolution at 30 fps, and one (*Spain1* sequence from MPEG-7 set) in 352×288 spatial resolution at 25 fps. Figure 11 shows the name and length of all clips in the database.

Names and the length (min:sec) of the clips
Korea1(54:53), Gant1(48:22), Gant2(46:59), Spain1(14:58), Mlt1(49:57), Mlt2(49:09), Ts2(48:09), Den1(47:27), Den2(47:51), Rize1(47:36), Rize2(49:15), DBak1(46:25), DBak2(48:25), Goz1(47:30), Goz2(48:56), Ant1(45:43), Gs1(49:05)

Figure 11 – Names and lengths of the clips in the database [14].

Goal events are detected in 15 of the 17 video sequences in the database. Each video sequence is processed to detect shot boundaries, shot types and replays. Every time a replay is found, the goal detector algorithm uses the cinematic template features to try to find goal events. The goal detector's algorithm results for each video sequence and for the entire set are shown in Table 5. The algorithm achieves 90.0% Recall and 45.8% Precision rates, which are considered satisfactory by the proposers as they feel that Recall is far more important than Precision in this algorithm as the user can always fast-forward non-goal events or can even find them interesting to watch.

Sequence	Correct	False	Miss	Sequence	Correct	False	Miss
Gant1	2	2	0	Spain1	2	0	0
Gant2	3	3	0	Korea1	2	2	1
Mlt1	2	3	0	Den1	2	2	0
Mlt2	1	1	1	Den2	1	3	0
DBak1	1	1	0	Goz1	1	2	0
DBak2	2	1	1	Goz2	2	2	0
Rize1	2	6	0	Ts2	3	0	0
Rize2	1	4	0	**TOTAL**	**27**	**32**	**3**

Table 5 – Distribution of goal detection results for each video sequence [14].

In the main tools section, it was stated that the existence of referee or penalty-box in a summary segment that also contains a slow-motion replay typically implies the occurrence of possible interesting events. Considering this, the user can search for summaries using object-based features. Recall and Confidence metrics were used to evaluate referee and penalty-box detection algorithms' performance. Recall rate illustrates the accuracy of the algorithm while the

Confidence is defined as the ratio between the number of events with that object to the total of such events in the database. This value indicates the applicability of the object-based feature to search for a certain event. For instance, a penalty kick event has high confidence values for both objects - referee and penalty box - as both are usually visible when a penalty kick occurs. The Recall and Confidence results for referee and penalty-box detection algorithms are shown in Table 6 and Table 7.

	Yellow/Red Cards	Penalties	Free-Kicks
Total	19	3	8
Referee Appears	19	3	5
Detected	16	3	5
Recall(%)	84.2	100	100
Confidence(%)	100	100	62.5

Table 6 – Statistics on the appearance of the referee for specific semantic events [14].

	Shots/Saves	Penalties	Free-Kicks
Total	50	3	8
Penalty Box Appears	49	3	8
Detected	41	3	8
Recall(%)	83.7	100	100
Confidence(%)	98.0	100	100

Table 7 – Statistics on the appearance of penalty-box for specific semantic events [14].

In both tables above, the first row is the total number of specific events in the summaries, the second row shows the number of events where the referee and/or the penalty-box are visible, and the third row presents the number of detected events.

Concluding, on the average, 12,78% of a game is included in the summaries of all slow-motion segments and 4,68% is included in the summaries of only goal events, including all detected non-goal events. These averages correspond to, approximately, 12 and 5 minutes of 90 minutes football match.

2.2.2.5 Summary

This work presents an approach to the video summarization problem, in the domain-specific content branch as it mainly serves to summarize football matches. In the domain-specific area, it achieves its objectives in a hybrid manner since the proposers use both low-level and high-level features with good results. Low-level features are used in the low-level analysis for cinematic features extraction algorithms while high-level features are used in the detection of goal, referee and penalty-box events. This hybrid approach constitutes the main difference for the rest of the systems here described justifying its presence in this chapter for a wider review of audiovisual summarization technologies.

2.2.3 Generic content summaries based on affect

The third system to be described has been developed by Hanjalic and Xu [5][6][7] and focuses on the semantic summarization of multimedia content, mainly based on the extraction of moods from pictures and sounds. In the context of the classification tree proposed in the previous section, this system fits as a generic content solution exploring

affective summarization based on low-level features. This system will assume great relevance in the following as it will become the main reference for the system developed in the context of this Thesis.

2.2.3.1 Objectives

The objective of this system is to summarize generic video content based on the emotions and feelings evoked on a user by an audio or video clip, or even some speech content. Those emotions and feelings are referred as the affective content of an audiovisual signal.

Affective content analysis can be of great utility in many multimedia applications, notably content indexing and retrieval applications. This user centric approach can lead to a major breakthrough in VoD and PVR systems, as it can identify the prevailing mood of a film or even the sad, happy, exciting or dull parts of a movie.

This approach is also relevant for the problem at hand, audiovisual summarization, as it brings the possibility of extracting the "most interesting" parts of a video or several video clips from an affective perspective and concatenating them in a "summary trailer". This also means that one single algorithm, capable of extracting the more exciting parts of a generic video clip, can do the same work of an entire group of algorithms, each of them developed to analyze a particular type of content or event; for example, in a football content context, the proposed algorithm shall be able to (simultaneously) detect goals, free kicks or a violent tackle since these are events that provoke strong emotive reactions.

The advantages of such generic content approach to audiovisual summarization are the simplification of the algorithmic structures of content analysis tools and its independence regarding the type of events. Only the excitement provoked by the segments constituting the event matters for the decision of including them in the summary or not.

2.2.3.2 Approach and architecture

As mentioned above, this system makes summaries based on affective properties. Affect has three main dimensions used to extract the affective value of a video: Arousal (A), Valence (V) and Control or Dominance (C):

- Valence is typically considered the "sign" of emotion, going from "pleasant" if the emotion is positive to "unpleasant" if the emotion is negative.
- Arousal indicates the "intensity" of the emotion and may distinguish feelings like peacefulness, excitement, alert or calmness.
- Control is useful only to distinguish feelings with similar arousal and valence but it has been recognized that it doesn't have a significant role in the extraction of the affective content.

Therefore, affect modeling is mainly based on the extraction of arousal and valence; in the following, temporal curves for each one of these two dimensions will be constructed using low-level video features. These time curves are known as arousal time curve and valence time curve. The arousal time curve illustrates the temporal positions of the "most exciting" video segments and, in certain video genres, for instance sports, it is sufficient to characterize its affective content. The valence time curve distinguishes the "positive" and the "negative" video segments and can be important to personalize video retrieval, as the user can choose, for example, to remove all the "negative" segments of a certain video.

Valence modeling has great relevance in VoD and PVR systems as it enhances the ability of user personalization in such systems; its ability to determine the "sign" of the feelings, turns out to be important to determine the mood of a certain movie and, therefore, to automatically classify the movie to its specific genre. This movie classification gives the user the possibility to choose films according to its current mood or to create a user profile with some preferred movie genres. Figure 12 illustrates both arousal and valence time curves.

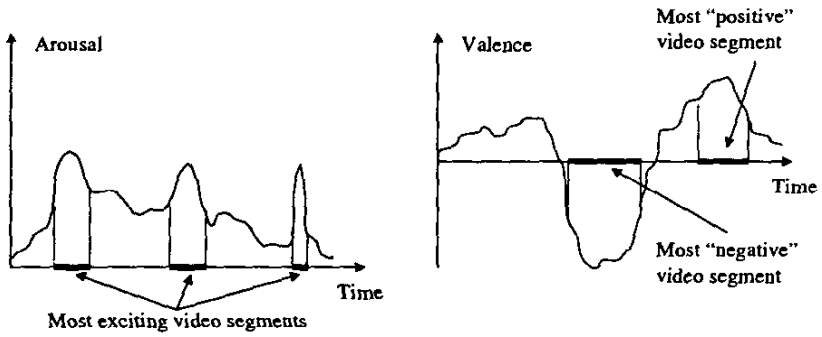

Figure 12 – Illustration of arousal and valence time curves [5].

To achieve a complete representation of the affective content of a video, the arousal time curve is drawn against the valence time curve, obtaining the so called "affect curve", which can be very relevant to find the prevailing mood of a video clip, by just finding in the 2D space the zone where the curve lies for most of the time. This can be utilized to classify videos into different affective genres. Figure 13 shows an example of an affect curve.

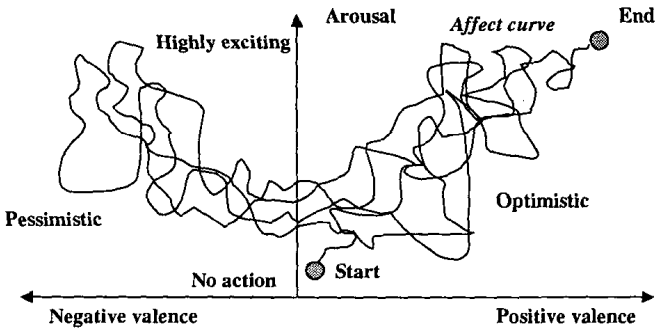

Figure 13 – Illustration of an affect curve [5].

For this audiovisual summarization system, the proposers considered only the arousal dimension as it is the main affective dimension relevant for highlight extraction. Using only arousal modeling, a highlight time curve can be constructed as will be shown in the next section.

To summarize, this system performs the following processes as shown in the architecture presented in Figure 14:

1. **Extracting features** – The system extracts the low-level features needed to model arousal from the input audiovisual signal.
2. **Arousal modeling** – Based on the low-level features extracted, arousal is modeled and an affect curve is constructed.
3. **Extracting highlights** – Using the affect curve, highlights will be extracted and included in the summary.

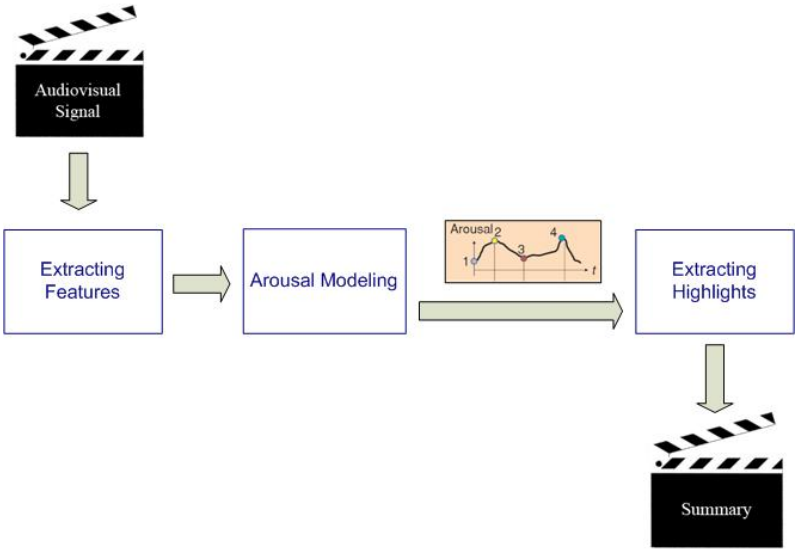

Figure 14 – High-level architecture.

2.2.3.3 Main tools

As referred before, the three main modules of this system are the Extracting features, the Arousal modeling, and the Extracting highlights modules described in the following:

- **Extracting features** - To obtain the required highlight time curve, it is first needed to model the arousal which is done based on some adequate low-level features. The features extraction process must be straightforward; the main decision here regards which features to extract in order effective affect modeling may after be performed. Much research has been done on this issue in past years. Results show that motion in an audiovisual signal has a high impact on affective responses; an increase of motion intensity on a screen has direct influence on arousal. Regarding the audio features, the loudness or sound energy and the speech rate are also related to arousal. Another feature that strongly influences arousal is the shot length; this feature is typically used by content authors/producers to change the arousal intensity that is provoked in the audience. A movie director normally chooses shorter shot lengths for film sequences with strong action development to create stressed moments for the viewer. Longer shots are often used after action sequences to relax the viewer. In a sports event, for example football, a goal or a goal chance are typically characterized by an increase of shot changes in a try to cover everything that is happening in the field and in the stands.
- **Arousal modeling** – The arousal modeling is done by an integrated function of the three low-level video features referred before:
    1. Overall motion activity measured from consecutive frames;
    2. Sound energy; and
    3. Density of shot cuts.

    Each of the three low-level features has to be convoluted with a Kaiser window after its direct extraction. The Kaiser window formula to be applied to each of the low-level features is:

$$\tilde{f}(k) = f(k) * K(l, \beta) \quad (17)$$

In the above formula, f(k) represents the result of the feature extraction process and K(l,β) the Kaiser window function, with l as the length of the Kaiser window and β is an arbitrary real number that determines the shape of

the window. The larger the value of β the narrower the window becomes with β=0 corresponding to a rectangular window.

The need for the Kaiser window convolution is related with the fact that the direct extraction of low-level characteristics may result in highly varying curves, full of high peaks followed by periods of no activity at all or vice-versa. The viewer cannot increase and decrease its level of excitement as abruptly as the direct extraction low-level features curves may show. The Kaiser window convolution turns the abrupt changes into gradual changes in the low-level features curves, making them more likely to represent viewer's arousal changes.

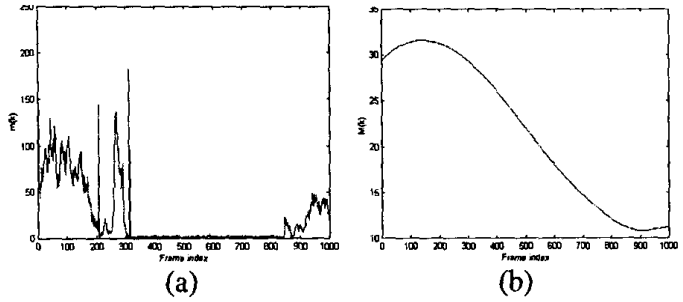

Figure 15 – (a) Raw motion activity directly computed; and (b) Motion after convolution with a Kaiser window [5]

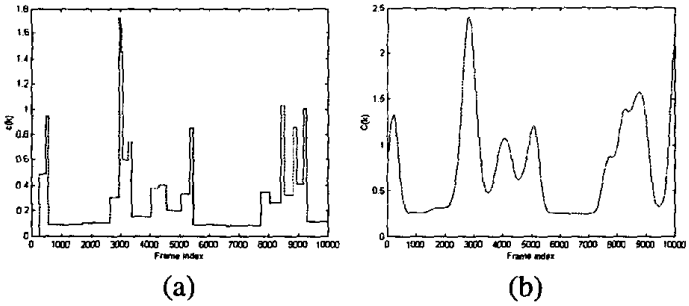

Figure 16 – (a) Density of cuts directly computed and (b) Density of cuts after convolution with a Kaiser window [5].

- **Extracting highlights** – After arousal modeling, there are two ways of extracting highlights:
    1. **Based on highlight duration** - The most straightforward way uses duration as the input control parameter; the arousal time curve is thresholded to that same duration and the selected video segments are concatenated to create the video summary.
    2. **Based on highlight strength** - The other approach extracts highlights selectively, based on a highlight strength *M* that represents the richness and power of the highlight. *M* corresponds to the number of feature time curves that achieve a specific arousal threshold during the video segment. The *M* parameter can be entered by the user using an intuitive scale ranging from "the most interesting moments only" to "everything worth watching". Only the segments with strength of, at least, *M* can be part of the highlight extraction process. The selectiveness comes from filtering the arousal time curve by the *M* parameter. The higher the value of *M*, the stricter the filtering of the arousal time curve becomes.

2.2.3.4 Performance

As affective video content representation and modeling has many different target applications, the proposers of this system [5] did not provide in detail many results for the highlight extraction model. However, they exhaustively present results for the arousal and valence modeling. In the scope of this report, only arousal results will be presented as they are the relevant ones for audiovisual summarization.

The system has been tested mainly with football TV broadcasts. Figure 17 shows some results related to a typical football television broadcast. The length of the Kaiser window used to smooth the highlight time curve was set to 1500 and its shape was set to 5. Figure 17 (a) shows the low-level features time curves for this video sequence. The highlights time curve was computed based on these features using three values for M: M = 1, 2 or 3. Figure 17 (b) shows the highlights time curve this means the segments selected for the summary for M = 3 as well as the original arousal time curve resulting from the arousal modeling process.

Comparing Figure 17 (a) with Figure 17 (b), it can be seen that the highlights temporal curve has notorious peaks at segments corresponding to goals. The horizontal line in Figure 17 (b) uses a threshold corresponding to a 50s summary duration, including the two goals and the preceding and succeeding moments. Moving the thresholding line vertically with the same highlight strength only the length of the extracted segments will change but not the composition of the extracted segments. With M = 3, the summary will always consist of the two goals and the game play related to them. Figure 17 (c)-(d) reflect the same procedure but now with different highlight strength, M = 2 and M = 1, respectively. With the reduction of the highlight strength, more segments will be extracted from the video content. Figure 17 (c) shows the arousal and highlights curves including a free kick and a goal chance besides the goals while in Figure 17 (d) the arousal and highlights curves for these four segments are also included but with more content related to each of the segments since no other enough relevant events could be found in the original video sequence.

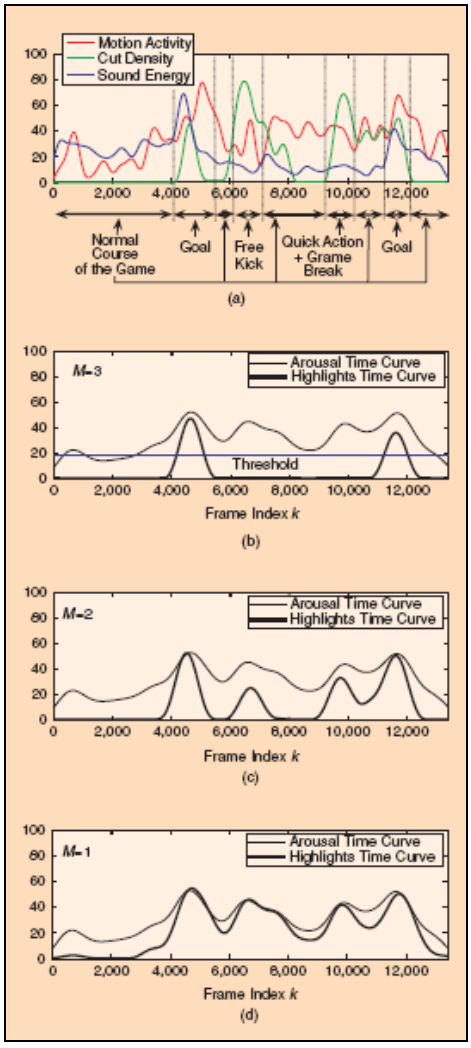

Figure 17 – (a) Arousal features time curves and (b)-(d) arousal and highlights time curves obtained from a football match with different values of M [5].

2.2.3.5 Summary

This system is a generic content solution for audiovisual summarization. This fact makes it a more powerful approach to the problem than the system previously presented as it produces summaries for any kind of input video. In the scope of this report, it reveals another very relevant dimension which is the exploitation of affect in summarization. The generic and affective aspects of this solution come through the concept of "arousal": the search for highlights is done by tracing the video segments where the arousal experienced by the viewer is expected to be high, instead of modeling each potential highlight event individually. For this set of reasons, this system has become a major reference for this Thesis.

2.2.4 Generic content summaries based on a user attention model

The last system presented in this chapter addressing the video summarization problem was developed by Ma, Hua, Lu and Zheng [8] and provides a generic framework for user attention modeling and its application to the problem at hand this means audiovisual summarization. Regarding the classification tree proposed before, this system fits as a generic content, affective summarization solution based on both low-level and high-level features.

2.2.4.1 Objectives

The final goal of this system is to summarize video segments effectively. This means to extract the main information from a video excluding all redundant or unimportant data. The authors believe that low-level approaches, though used abundantly, are insufficient to produce effective audiovisual summaries as they are not consistent with human perception. Therefore, they aim to build an effective alternative solution to audiovisual summarization by exploring the human perception mechanisms. To achieve the established objective, the authors developed a solution divided in three main parts:

- Generic user attention model framework;
- Set of visual and aural attention modeling methods; and
- Solution to video summarization based on the defined attention model.

The focus of this approach on the notion of attention comes from its utility to solve the video summarization issue. Attention means the human ability or power to concentrate mentally on an object or person. With the increase of computational power along the years, the time spent to model attention has decreased significantly.

2.2.4.2 Approach and architecture

This system is divided into two different models: user attention model, and video summarization model:

- **User attention model** - Regarding content analysis, the viewers' type of attention paid to audiovisual content can be classified into stimulus-driven, semantic-driven and goal-driven. The stimulus-driven attention results from the direct stimulation of human receptors, such as drums in ears or retina in eyes, while semantic-driven attention means that specific prior knowledge is needed to fully understand the object; finally, goal-driven attention mainly regards camera motions (panning, zooming, etc.) and aims to guide the user's attention according to the producers' intentions. Figure 18 presents the architecture of the user attention model.

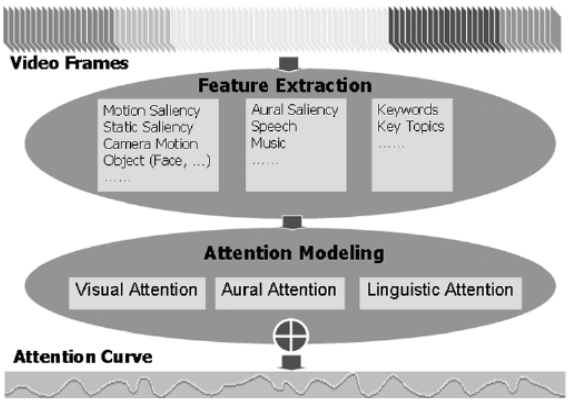

Figure 18 – User attention model architecture [8].

The user attention model is composed by three-sensory channels: Visual, Aural; and Linguistic. The main focus here goes to the visual and aural models as the linguistic model does not play a major role in audiovisual summarization.

Each of these basic channels is then decomposed into its primary elements. The primary elements for visual attention are motion, appearance and objects, while for aural attention are speech, music and salience. After the decomposition, a set of modeling methods is used to obtain visual and aural attention curves separately and then fusion schemes are required to generate a final attention curve. The attention curve reflects each frame's importance along a temporal axis. It is used to summarize videos in two different ways: static summarization which corresponds to a set of images or key-frames extracted from the video, and dynamic summarization which corresponds to a set of audiovisual segments selected from the original video according to their importance.

- **Video summarization model** - Regarding video summarization, which is an important application of user attention modeling, a solution was developed that can show the robustness of the user attention framework; it includes processes for dynamic and static summarization. The video summarization architecture is shown in Figure 19.

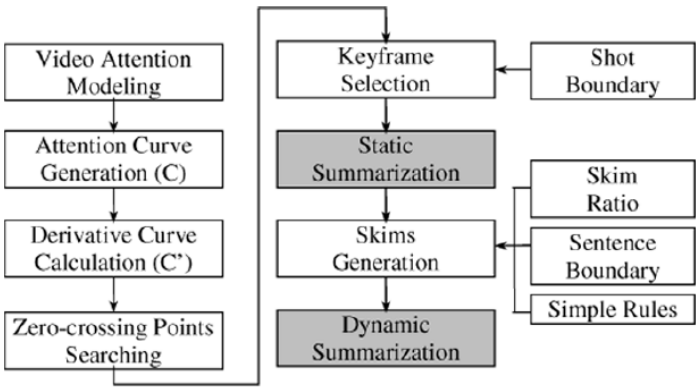

Figure 19 – Video summarization architecture [8].

A short walkthrough of the video summarization model may be seen as:

1. **Attention curve generation –** Visual and aural models are computed and then fused in Video attention modeling, generating an overall attention curve. The key-frames and video skims are extracted from around the attention curve crests that indicate the video segments more likely to attract the viewers' attention.

2. **Key frame selection –** To determinate the exact positions of the attention curve crests, a derivative curve is calculated with the zero-crossing points from positive to negative on the derivative curve corresponding to the peaks of wave crests on the attention curve.
3. **Static summarization –** Consists in extracting only key-frames from the input video and, therefore, in generating static summaries. This can be achieved by ranking the attention values obtained for each key-frame. The maximum attention value in all key-frames is taken as a reference of shot importance. If only one key-frame is required for each shot, the key-frame selected is, naturally, the one with maximum value but if the key-frames required are less that the number of shots in a video, the key-frames with lower attention values will be discarded.
4. **Dynamic summarization –** Is also possible in this model and consists on creating audiovisual segments as summaries. With a skimming ratio, i.e. the length of the summary related to the original video, as input, skim segments are selected around the key-frames for each shot. The consistency and fluidness of the skim segments is guaranteed by not truncating speech sentences, through an adaptive approach to pause detection.

The main focus of this system goes to visual and aural attention modeling and, therefore, a more detailed description of these modeling methods will be given in the next section. The video summarization model will be evaluated in the section after the next targeting performance evaluation.

2.2.4.3 Main tools

This section will describe the visual and aural attention models and also the fusion schemes proposed to create a final attention curve from the attention curves obtained from the elementary modeling processes.

- **Visual Attention modeling** - Visual attention modeling is defined mainly by motion and appearance, e.g. color, shape, texture, two major attributes of video. These two attributes are the basis for the motion and static attention models and characterize the stimulus-driven attention. To achieve goal-driven attention, a face attention model based on a face detection system was also developed. In addition, a camera attention model, aiming to reflect the producers' intentions to guide the viewers' attention through camera movements was also defined.
  - **Motion attention model -** Motion attention can be estimated based on a Motion Vector Field (MVF) which can be obtained from the MPEG coded stream, directly or by block-based estimation. The proposers make the analogy of treating the MVF as retina in eyes and, therefore, motion vectors as the perceptual responses of the optic nerves. In this system a MVF gives information regarding to:
    - Motion intensity,
    - Motion spatial coherence and
    - Motion temporal coherence.

    When motion vectors are extracted, information regarding these three features produces an intensity, spatial coherence and temporal coherence maps that are shown in Figure 20 (c)-(e). Figure 20 (b) represents the saliency map, which indicates the spatio-temporal distribution of motion attention; this is obtained from the fusion of the outputs of the three above mentioned features.

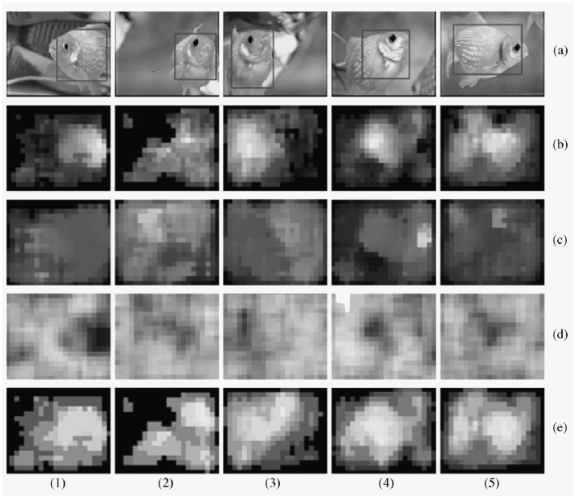

Figure 20 – Motion attention estimation results [8].

These three features measure the motion in a video sequence and each of them is calculated for the macroblocks corresponding to the given MVF.

- **Static attention model -** The static attention model was developed to detect static scenes in a video sequence, which can be of great relevance to human attention and cannot be detected by the motion attention model. The main step of static attention model is the generation of a contrast-based saliency map. As in the motion attention model, the saliency map is very important; in this system, it is created only from appearance. The main feature behind appearance is contrast. The perception of an object depends greatly on the way it can be distinguished from its environment and this distinctiveness is achieved from contrast. The contrast-based saliency map gives simultaneously information about texture, color and approximate shape. Figure 21 shows some examples of static attention estimation results and of saliency maps in Figure 21 1(b), 2(b), 3(b) and 4(b). A method called Fussy Growing proposed in [19] is used to extract attended areas from the saliency maps. Attended areas are areas that do not belong to the video frame background and therefore can be of interest to the viewer. These areas are marked as highlighted boxes in Figure 21 1(a), 2(a), 3(a) and 4(a) and correspond to the bright areas in the saliency maps. Based on the number of attended areas as well as their brightness, area and position, a model for the static attention corresponding to each image is computed.

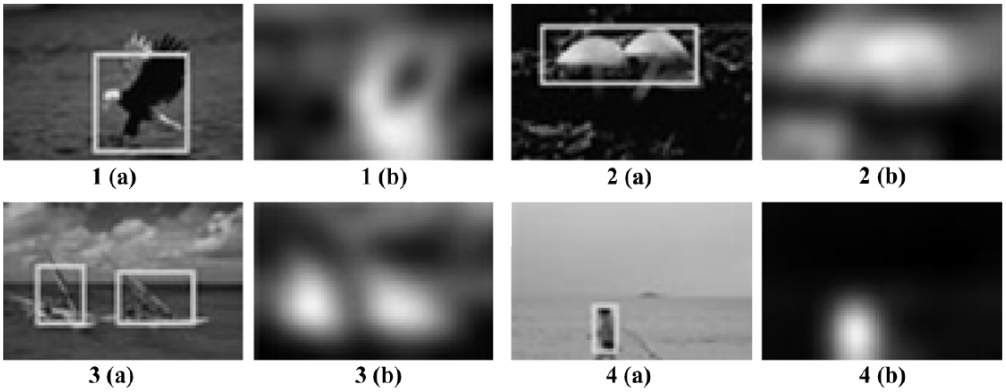

Figure 21 – Static attention estimation results [8].

- **Face attention model -** The face attention model is a semantic attention model and is defined because faces in a video sequence usually attract users' attention. For this, a face detection module is used [20]; the face

attention model is computed based on production rules establishing that the position and size of the faces reflect the importance of protagonists.

- **Camera motion model -** A camera motion model that is a guided attention model is also defined; its inclusion in this system results from the importance that camera motion has in human attention. Producers use camera motion to give more or less importance to some video content and, consequently, guide the viewers' attention. For this, a fast camera motion analysis module is used [21]. The goal of this model is to create an attention curve based only on camera motion. The model is defined as a magnifier, i.e. it is multiplied with the sum of the other visual attention models; its value is in the range [0-2]. If the camera motion tends to emphasize some video content, the camera motion attention value is approximately 2; if it tends to neglect some video content, than the value is near 0; finally, if the camera motion does not intend to guide the viewer's attention, its value is 1.

- **Aural attention modeling -** As for visual information, aural information is very important to attract viewers' attention. Regarding audio, the user attention model includes an aural saliency model for stimulus-driven attention, and speech and music models for goal-driven attention.
  - **Aural saliency model -** The aural saliency model is based on sound energy; it assumes that the viewer pays attention to a sound in two possible scenarios:
    - An absolute loud sound measured by the average energy or
    - The increase or decrease of the loudness measured by the energy peak.

    A sliding window is used to compute the aural saliency along a temporal axis. As in the camera motion model, the aural saliency model works as a magnifier in the aural attention model.
  - **Speech and music attention models -** Viewers usually pay more attention to speech and music than to special sound effects as they normally have some specific semantic meaning. The saliency of speech and music is computed by the ratio of the sub-segments with speech/music and other sounds sub-segments along an audio segment. As in the aural saliency model, a sliding window is used to compute both the speech and music attention models.

2.2.4.4 Performance

The performance evaluation provided by the authors of this system relies on the described video summarization model and considers two distinct scenarios:

- Static summarization evaluation
- Dynamic summarization evaluation

Twenty viewers were invited to evaluate summaries created based on the proposed user attention model. The experiment is composed by three parts corresponding to single key-frame and multiple key-frames summaries, both related to static summarization and video skimming related to dynamic summarization. The viewers were asked to evaluate video skimming first to guarantee the total lack of knowledge of the video content when evaluating. The evaluated videos cover a wide range of genres and lengths to assess the robustness of the solution and, consequently, of the user attention model. The sum of the videos' length is about 70 min and each one varies from 6 to 30 min. The shot is the basic unit evaluated and so shot boundaries are presented too. 742 shots were segmented. The list of evaluated videos and some of their features, e.g. genre, duration and number of shots, are detailed in Table 8.

No.	Video	Genre	Shot	Time
I	Animals.mpg	Documentary	83	9:14
II	Bahamas.mpg	Sightseeing	53	5:39
III	Basketball.mpg	Sport game	75	10:02
IV	Nbcnews.mpg	TV news	488	30:10
V	TheBoy.mpg	Home video	43	12:35
Total	——	——	**742**	**67:40**

Table 8 – List of test videos [8].

Figure 22 shows the user attention model results for video 1, "animals.mpg". The vertical lines crossing all curves define the shot boundaries corresponding to each of the 31 shots shown in Figure 22 - I. There is a single key-frame for each of the 31 shots as shown. Figure 22 - II represents the attention curves of all models already described. The skimming ratio, i.e. the length of the summary related to the original video, is 30% and is previously chosen.

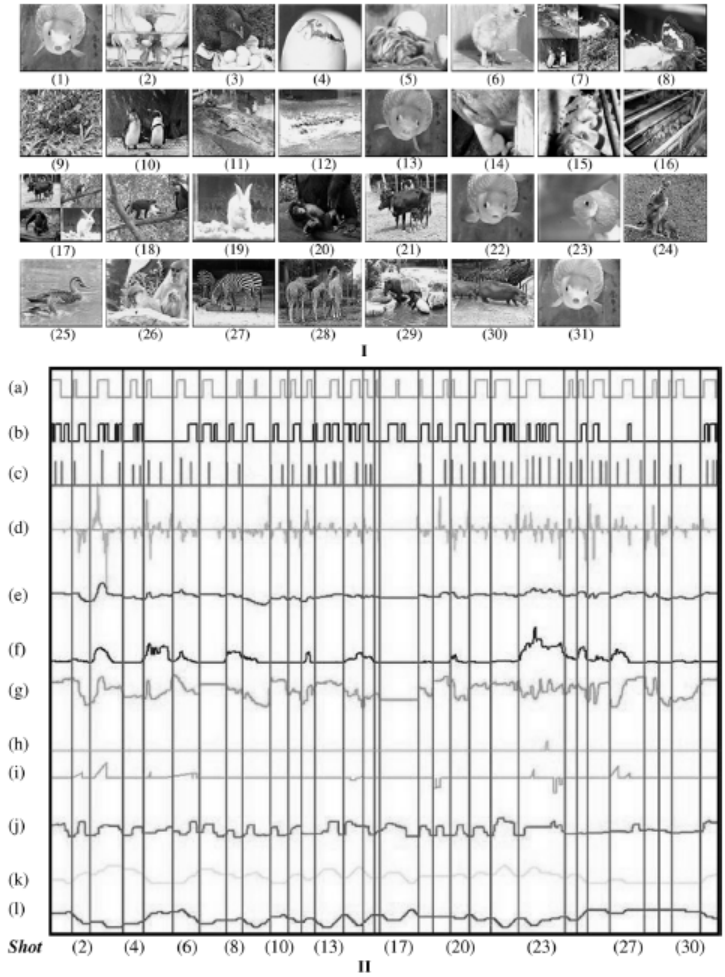

Figure 22 – Attention curves. I) First 31 shots of animals.mpg with one key-frame per shot; II) Corresponding attention curves: (a) skims curve; (b) sentence boundary; (c) key-frames (zero-crossing curves); (d) derivative curve; (e) final attention curve; (f) motion attention curve; (g) static attention curve; (h) face attention curve; (i) camera motion curve; (j) aural saliency curve; (l) music attention curve [8].

- **Static summarization evaluation**

  Static summarization is divided into two categories: Single key-frame solution and Multiple key-frame solution.

Single key-frame is, usually, used as a representative key-frame of an entire shot. Therefore, the main issue to assess is if the single key-frame extracted is informative enough or not whilst the main concern for multiple key-frames is if they are insufficient or redundant. Consequently, the viewers can label the single key-frame and the multiple key-frames results shot by shot. The single key-frames are labeled as good, neutral or bad and the multiple key-frames are labeled as good, too much or too few. In order to obtain a quantitative measurement, the single key-frame evaluation done by viewers was quantified into scores 100, 50 or 0, corresponding to good, neutral or bad and then an average score was calculated. Table 9 shows an average score of near 87 for the single key-frame solution. For the multiple key-frames solution, the quantitative assessment is done using the ratio of "goods" over all the shots in the video sequence. In this case, and as shown in Table 10 the average score is close to 75%.

No.	100 (Good)	50 (Neutral)	0 (Bad)	Avg.
I	63.55	17.11	2.34	86.87
II	39.81	12.71	0.48	87.10
III	53.98	18.07	2.95	84.02
IV	412.10	58.80	17.10	90.47
V	32.57	8.82	1.61	86.00
Avg.	——	——	——	**86.89**

Table 9 – Evaluation of the single key-frame static summarization solution [8].

No.	Good	Too Much	Too Few	G/All
I	62.28	9.62	11.10	0.7504
II	39.70	3.25	10.05	0.7491
III	52.56	7.43	15.01	0.7008
IV	403.42	35.67	48.91	0.8267
V	31.02	7.95	4.03	0.7214
Avg.	——	——	——	**0.7497**

Table 10 – Evaluation of the multiple key-frames static summarization solution [8].

- **Dynamic summarization evaluation**

Summaries or skimming videos should be, in principle, as short and informative as possible. However, it is difficult to achieve both goals at the same time. Therefore, the target for one of these objectives was fixed in order the performance for the other to be evaluated. Skimming ratios were then fixed in 15% and 30%. The tested videos were evaluated using two criteria: informativeness and enjoyability as it is also important to assess how enjoyable the skimmed video is (especially for certain types of content like sports and movies). The subjects were required to assign scores in the 0-100 range to the skimmed video in both criteria; the subjects were also required to view the original video before providing their assessment. The results are shown in Table 11.

No.	Informativeness			Enjoyablity		
	15%	30%	100	15%	30%	100%
I	63.69	75.54	98.50	63.81	72.98	95.40
	64.66	76.69	100	66.89	76.50	100
II	62.72	70.88	90.05	62.87	72.16	88.40
	69.65	78.71	100	71.12	81.63	100
III	56.73	70.12	97.06	55.02	66.97	93.05
	58.45	72.24	100	59.13	71.97	100
IV	62.51	72.98	98.83	62.12	72.44	96.60
	63.25	73.84	100	64.31	74.99	100
V	60.01	71.89	96.02	61.11	71.04	89.50
	62.50	74.87	100	68.28	79.37	100
Avg.	**63.70**	**75.27**	——	**65.94**	**76.89**	——
Drop	**36.30**	**24.73**	——	**34.06**	**23.11**	——

Table 11 – Evaluation of dynamic video summarization [8].

While the non-highlighted rows in Table 11 include the average scores from the twenty subjects', the grey highlighted rows include the normalized average scores. The average score presented on the bottom-line is obtained from the normalized scores and the drop value represents the average loss of informativeness and enjoyability for each case regarding the full video. The drop value is calculated by subtracting the average score to 100% and so should be as small as possible.

The informativeness and enjoyability increase with the skimming ratio but even for a low skimming ratio like 15%, the results are rather satisfactory (63.70 and 65.94 for informativeness and enjoyability, respectively).

2.2.4.5 Summary

This system is an example of an interesting alternative to the third system described since both are generic content, affective-based solutions. It aggregates both low-level and high-level features in the construction of its user attention model and presents a complete solution to address the audiovisual summarization problem as it includes a video summarization model able to produce static or dynamic summaries, i.e. summaries based only on key-frames or also audiovisual segments. The solution differs from the one previously described in its use of high-level features to provide a different approach to affective summarization.

2.3 Final remarks

The main goal of this chapter was to classify and present the wide range of technologies and systems available to address the audiovisual summarization problem. This should provide a bridge between the introduction and the developed summarization solution, described in the next chapters.

Regarding the classification of the existing solutions, the main distinguishing factor selected was the genericity of the approach, this means if the system is able to deal with generic content or is domain-specific. In both generic and domain-specific approaches, another main distinguishing factor is the exploitation or nor of the notion of affect for the selection of the segments to be include in the summary. This division turns out to be a very important one as the solution to be developed in the context of this Thesis fits in the generic content, affective branch to solve the video summarization problem. The choice between low-level and high-level features or even a hybrid type of solution for both affective and non-affective solutions is also important but less relevant that the main conceptual choices referred

before. It is important to state that other divisions (classification dimensions) could have been adopted but this is the one that better suits the scope of this Thesis mainly because the target here is generic content, affective summarization.

After, four summarization systems have been described in detail, each of them presenting a different approach to solve the same problem. The third described system [5][6][7] has a central role for this work since it is the one that will be taken as reference for the implementation made. The other three systems are important to give the reader the notion of the great variety of approaches available to address the video summarization problem.

Following the reviewing made in this chapter, the next chapter will present the architecture of the solution developed for generic content, affective audiovisual summarization.

Chapter 3

Architecture and Functional Description

Chapter 3 intends to present the reader with a first perspective of the developed summarization solution. To achieve this goal, this chapter will present the system architecture as well as a functional description of each of the architecture's modules. Chapter 4 will present an in-depth description of the processing solutions found to implement each module. Under the classification tree proposed in Chapter 2, the solution developed in this Thesis should be classified as an affective generic content approach based on low-level features.

3.1. System architecture

As referred before, the reference system for the technical solution adopted for this work was developed by A. Hanjalic from the Technical University of Delft and presented in [5][6][7]. Therefore, there are many similarities between the architecture proposed by Hanjalic and the system architecture developed and implemented in the context of this Thesis. In the summarization system presented in this Thesis, the excitement or arousal felt by the viewer of a certain audiovisual content is modeled in order to present him/her with a summary of the selected audiovisual content. This arousal modeling relies on low-level features extracted from the audiovisual content. A major difference between Hanjalic's solution and the system developed in this Thesis relies on the information necessary to present the summary to the user. While Hanjalic uses only the arousal curve resulting from arousal modeling to decide which segments of the original audiovisual content should be included in the final summary, here an MPEG-7 compliant hierarchical summary description is created; this allows all the segments of the audiovisual content to be labeled and included in the final summary to be created later based on that description and the user needs. This hierarchical summary description allows the user to create and view many summaries of the same audiovisual content, e.g. with different lengths, without repeating the entire summarization process. Moreover, as it is created in an MPEG-7 compliant format, this also allows

other MPEG-7 players to present a summary to its users this mean it increases interoperability. From the hierarchical summary description, MPEG-1 summaries can be made by the user at the end of the process.

The system architecture is presented in Figure 23: it shows three core modules, its sub-modules, as well as the inputs and outputs for each sub-module. The three core modules are:

- **Low-level features extraction**
- **Arousal modeling**
- **Hierarchical summary description creation**

The MPEG Summaries Creation module is not considered a core module as it is not fundamental to the summarization process. The main output of this system is the hierarchical summary description from which, if it is the user's wish, can be created several MPEG-1 summaries.

The system's architecture will be introduced and described in detail in the next section.

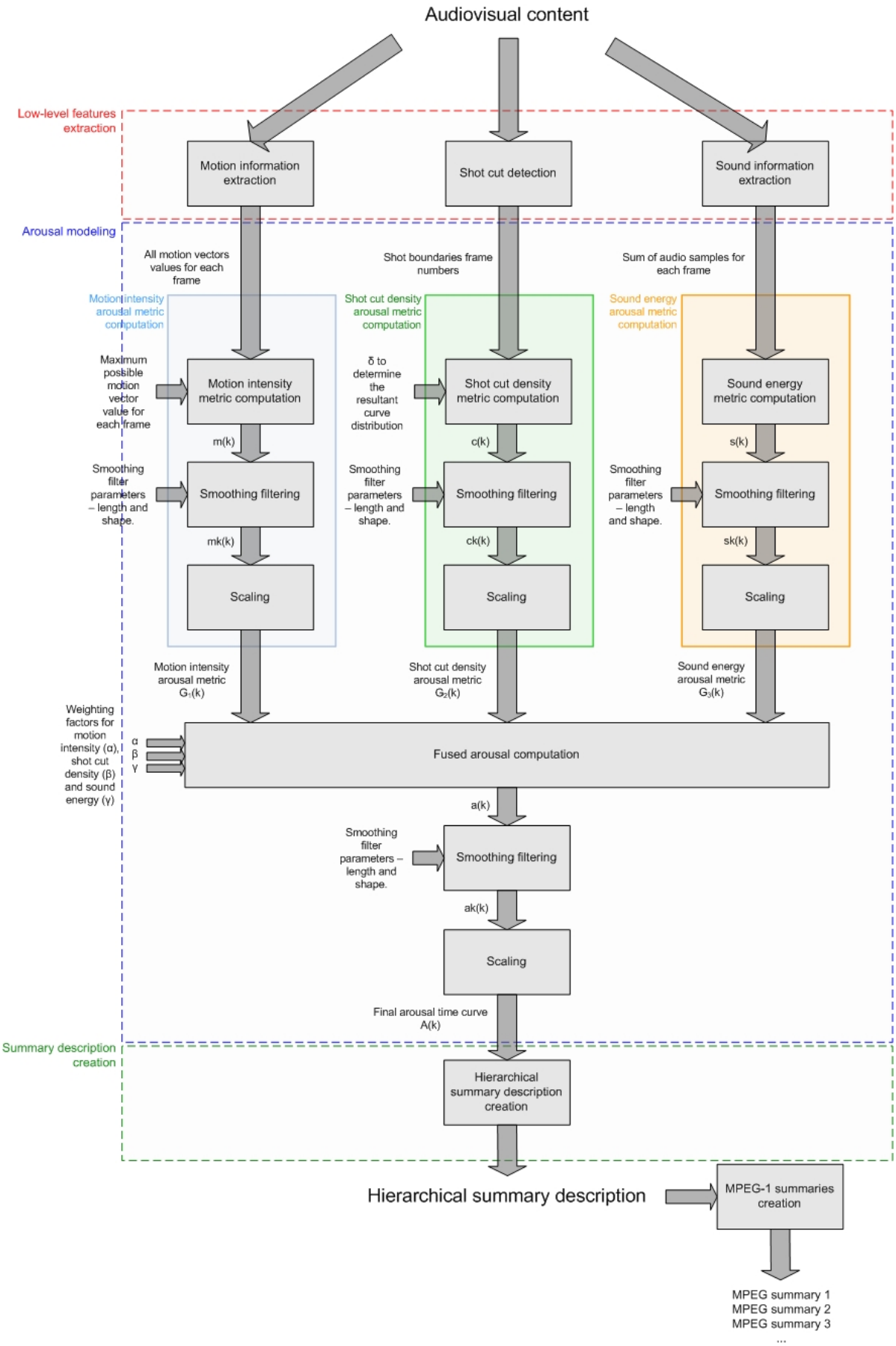

Figure 23 – Architecture of the developed solution.

A brief walkthrough of the architecture is presented next:

1. **Low-level features extraction –** The first step in the summarization process is the extraction of low-level features, necessary to model the arousal for the input audiovisual content, in MPEG-1 (coded) format, to be summarized. This module has a fundamental role as it provides the necessary information about the audiovisual content to effectively model the arousal. Each of the feature extraction processes – one per feature – is done independently, and it is possible to produce summaries based only on one or more of the three selected features. The outputs of this module will serve as input for the arousal modeling module.
2. **Arousal modeling –** The information obtained by extracting the low-level features from the audiovisual content is used to model its arousal. As in the extraction module, arousal is modeled independently for each feature, producing an arousal curve for each of them. A smoothing filter is applied after in the process, for each of the features, with the objective to transform the (sometimes) abrupt arousal changes resulting from the feature extraction in smoother arousal changes more likely to express the viewer's feeling when watching a video. After smoothing filtering, scaling is applied to scale the resulting curve to percentage curve, which is fundamental to permit the comparison between all arousal curves. When all features' arousals are modeled, a fusion function is applied to integrate them into one single final arousal curve. The arousal curve illustrates the arousal evolution along the audiovisual content duration. In the fusion process, different weights can be assigned to the various features, producing a different arousal final curve and, therefore, different final summaries.
3. **Hierarchical summary description creation –** After arousal modeling, a hierarchical summary description is created, resulting in the final output of the system. According to the frame arousal, frames will be grouped together in segments and those arousal segments will be labeled as "Top Highlights", "Key Points", "Extended Summary" or "Remaining Content" and included in the summary description under those labels. The arousal labels go from the most exciting - "Top Highlights" - to the less exciting - "Remaining Content". The user can view the summary for an audiovisual content by choosing through the first three labels (as the "Remaining Content" represents all audiovisual content) or by entering the length of the desired summary. If the user decides to input the summary length, the summary will include segments from "Top Highlights" to "Remaining Content", respectively, until the desired length is reached.

The next section will describe each module and its function in more detail.

3.2. Functional description by module

In this section, the function of the three core modules in the system will be explained, beginning with the low-level features extraction, as it is the basis of the entire summarization process.

3.2.1 Low-level features extraction

As the solution proposed fits under affective generic content based on low-level features approaches, the choice of the low-level features to be used to model the arousal is very likely one of the most important in the development of this system. For this reason, before describing the low-level features extraction module, the choice of the low-level features to be used in the summarization process will be motivated and explained.

3.2.1.1 Choice of low-level features

For the system developed, three features of great importance for any audiovisual content have been selected:

1. **Motion intensity –** Motion is very likely the feature from the three selected with more research associated, in recent years, in the video processing field. Motion intensity appears to be deeply related to the individual

affective response to audiovisual content, being able to provoke strong reactions in the viewers. An increase of object motion, as well as camera motion, typically implies an increase in the arousal. Wang and Cheong [22], based on the work of Detenber *et al* [23][24], state that "Motion plays a central role in the cinema owing to the intimate correlation between the degree of mental excitement and the perception of motion in screen. This correlation (…) seems to result from the natural association of fast motion with danger and excitement, as well as new activity or information". This direct correlation between motion intensity and mental excitement, or arousal, clearly justifies the choice of motion intensity as one of the three low-level features to take part in the summarization process.

2. **Density of shot cuts –** Research has also proven [7][22] that the density of shot cuts, or shot lengths, is deeply connected with the arousal experienced by the audiovisual content viewer. Shot lengths, or its patterning, are used many times by movie directors to inflict a desired pace of action. Normally, a higher density of shot cuts, or consequently shorter shot lengths, means action and stressful segments, while longer shot lengths are useful to provoke more relaxing and calmer moments for the viewer. A change in shot lengths during a video is likely to cause significant changes in the viewer's arousal, similarly to motion intensity. As a result, the density of shot cuts is also useful to model the arousal sensed while watching audiovisual content and, consequently, was also chosen to take part in the summarization process. Quoting Wang and Cheong in [22] "As each shot conveys an event, the director can heighten arousal and intensify a scene by increasing the event density via rapid shot changes (…). To the viewer, rapid shot changes capturing the main action from different angles certainly convey the dynamic and breathtaking excitement far more effectively than a long duration shot".
3. **Sound energy –** The third feature chosen for arousal modeling was sound energy. As motion intensity, the loudness or energy of the audio signal has a direct influence on the emotions that viewers may experience while watching a video. An increase of the sound energy in determined segments of the audiovisual content leads to a boost of the audience's arousal. In a football match, for example, when a goal event occurs or when a rough tackle takes place, normally the commentator shouts or starts to talk louder and the audience cheers or boos. In an action movie, gunfire or explosions are related to action sequences and, therefore, segments where the arousal experienced significantly increases. Like in these examples, in the majority of audiovisual content, sound energy takes a major role on the viewer's interest and, frequently, the segments with higher sound energy are coincident with the segments that seem to impose higher arousal values to the audience.

After motivating the selection of three low-level features, the function of the low-level features extraction module will be described in detail.

3.2.1.2 Low-level features extraction function

The name of this module clearly indicates its function. The low-level extraction module has the objective of providing enough information, based on the low-level features, to model the arousal and, consequently, to produce a summary of the input audiovisual content. Low-level features extraction is the first step to achieve the desired summary and, therefore, arises as the core of the proposed approach to the audiovisual summarization problem. Producing an analogy, low-level extraction can be viewed as a brute force process of gathering data, while arousal modeling is the 'brain' in the process for selective summarization.

Three feature extraction tracks were introduced in the proposed system architecture, one for each low-level feature, with the capability of delivering the information needed for arousal modeling. To model arousal from motion intensity, all motion vectors for each of the video frames (with the exclusion of I frames which do not have motion vectors) are

required; to model arousal from the density of shot cuts the frame indexes which represent shot boundaries are needed; and, to model sound energy, all audio samples from the audiovisual content are necessary. The feature extraction sub-modules have thus the following functions:

- **Motion information extraction –** Motion information extraction sub-module intends to extracts the necessary motion vectors from the MPEG-1 coded stream.
- **Shot cut detection –** Shot cut detection sub-module aims for the detection of all shot boundary frame indexes in the audiovisual content.
- **Sound information extraction –** Sound information extraction sub-module has the function of obtaining all audio samples from the audiovisual content.

3.2.2 Arousal modeling

After extracting the low-level features, the following step in the summarization process is arousal modeling. Arousal modeling is done by computing the arousal from each low-level feature independently, followed by the arousal computation resulting from the weights assigned to each feature. Although the architecture presented in Figure 23 shows three sub-modules for the arousal metrics computation, one per feature, the description of its function will be made together below since their function is similar. The Smoothing filtering and Scaling sub-modules functions will be explained next and, finally, an explanation of the Fused arousal computation sub-module will be given.

3.2.2.1 Arousal metrics computation

Arousal metrics computation for each feature is the starting point of the arousal modeling process. It has the main objective of transforming the raw information extracted by the previous modules into more useful information for modeling arousal. This is done by defining three arousal metrics, which will receive as inputs the corresponding low-level information and will be after convoluted with the smoothing filter to create three arousal curves, one per feature. In this manner, arousal metrics computation alone can be seen as the process of providing the basis for arousal modeling for each feature independently based on the low-level features extracted before. Chapter 4 will present and explain in detail each arousal metric used. Figure 24 shows some example charts illustrating the temporal evolution for each of the adopted arousal metrics: motion intensity, shot detection density and sound energy.

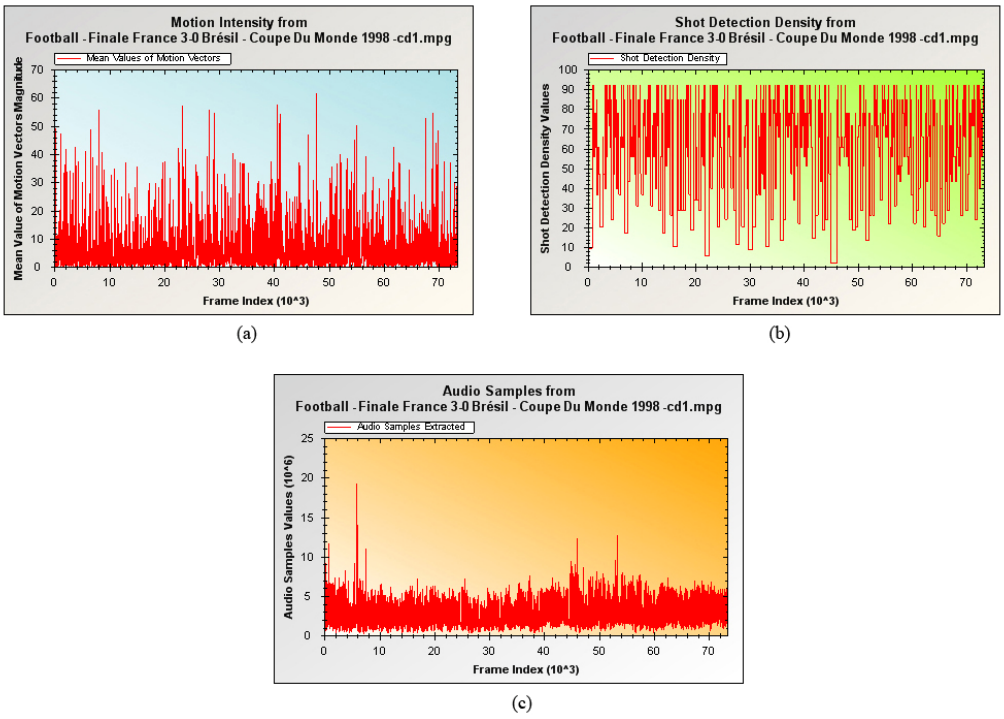

Figure 24 – Example of (a) motion intensity time curve, (b) shot detection density time curve, and (c) sound energy time curve for a football sequence.

3.2.2.2 Smoothing filtering

For the purpose of this work, the smoothing filter modules turn out to be essential as they have the function of smoothing the curves resulting from the arousal metrics computation processes. With the smoothing filter, the summarization process is able to better model the viewer's reaction to the audiovisual content. This happens mainly because, as the arousal curves become smoother, the gradual increase and decrease of the arousal values are much similar to the arousal experienced by the audience than the eventual abrupt changes directly resulting from the arousal metrics computation. Figure 25 shows an example of the motion intensity arousal curve before and after applying the smoothing filter for a football sequence.

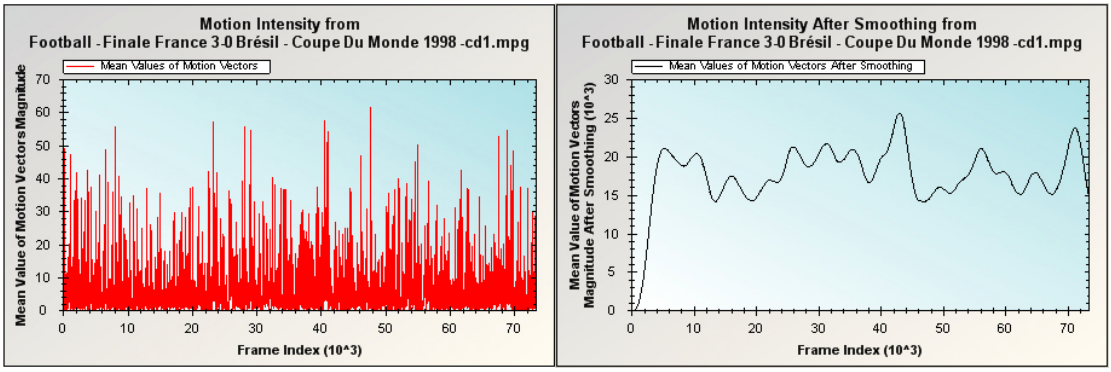

Figure 25 – (a) Example of motion intensity arousal curve before and (b) after applying the smoothing filter for a football sequence.

3.2.2.3 Scaling

The Scaling module function is also quite simple. Mainly, Scaling aims to scale the curves obtained after the smoothing filtering to percentage values. This is important because arousal time curves for each feature must be **comparable** and **combinable** in order to create a coherent final arousal time curve. In this manner, after scaling, the arousal curves for each feature are in percentage values which allow to compare and combine the three features in terms of arousal, targeting the creation of the final arousal time curve capable of representing, in an accurate way, the arousal experienced by the viewer. In the Sound energy arousal metric computation, the Scaling module has also a second function which is to scale the arousal time curve in relation to its peak, as sound energy can be dependent on the volume level at which the audio track was recorded. This factor could lead to sound arousal curves that could not be compared from video to video, if scaling did not take place. This process will be explained in detail in the corresponding section in Chapter 4. Figure 26 shows the motion intensity arousal curve before and after scaling. Note that the chart in the right is in percentage values while the chart on the left is not, having a top value of mean value of motion values after smoothing and before scaling of near **25000**.

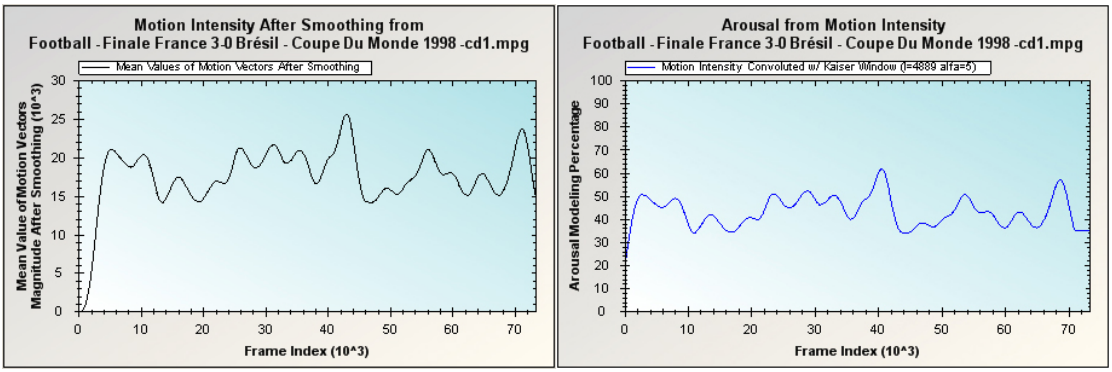

Figure 26 – (a) Example of motion intensity arousal curve before and (b) after scaling for a football sequence.

The time curves originated after scaling represent the arousal resulting for each low-level feature – in the arousal metrics computation – or the final arousal curve when applied in the Fused arousal computation.

3.2.2.4 Fused arousal computation

This sub-module, which has the final arousal curve as output, will determine the choice of the segments to include in the summary description. Despite its major importance in the process, its function is quite simple. Overall or fused arousal computation intends to integrate all individual arousal curves resulting from the previous sub-modules, by applying user defined weights for the combination of the features. The default weights are 1/3 for every feature as none of them seems to deserve, *a priori*, to be considered more important than the others. However, some types of content may deserve an adjustment of these values, for instance, sport content normally require a lower weight for shot cut density as the shot cut density significantly increases **after** the event occurrence, with slow-motion replays and several shots of the event's intervenient and the audience. The user can assign different feature weights if he/she wishes to see how a final summary would change if a feature becomes more relevant than the remaining ones.

Resuming, this sub-module creates a final arousal curve by fusing the arousal curves of each feature, resulting from the smoothing filter and scaling sub-modules. After final arousal computation, the smoothing filter and scaling sub-modules are, once again, applied, for the obvious purpose of smoothing and scaling the fused arousal for the same reasons explained before.

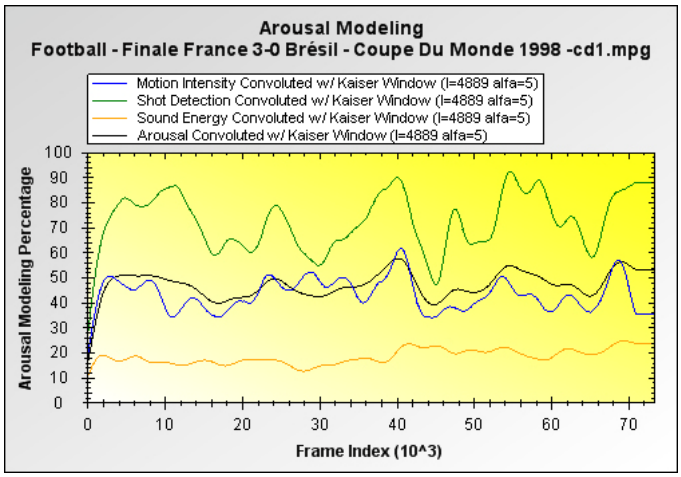

Figure 27 – Examples of fused and individual feature arousal curves.

3.2.3 *Hierarchical summary description*

To finalize the summarization process, a hierarchical summary description is created, as the main output of the summarization system developed. This module receives as input the fused arousal curve resulting from the arousal modeling process and has the function of deciding which segments should be included in the various possible types of summary. This labeling is fundamental as a user can, after the hierarchical summary description creation, view as many different summaries as he/she wishes, with different lengths or even of different types (meaning different levels of arousal). The user can, therefore, decide to watch two types of summaries:

- **By length –** The user inputs the summary length and a summary is presented with the desired duration. The choice of the segments to include is done by parsing the summary description from the segments labeled as most important – "Top Highlights" – to the segments labeled as least important – "Remaining Content" –, including them, or part of them, until the required length is reached.
- **By arousal level –** The user can decide to watch from three types of summaries. He/she may choose to watch only the "Top Highlights" of an audiovisual content, its "Key Points" or an "Extended Summary". The segments labeled as "Remaining Content" are used only when a summary By Length is chosen, as a "Remaining Content" summary would be synonym of playing the entire audiovisual content.

It is also important to create this summary in a format that provides interoperability between systems. In this way, the main output of this proposed solution can serve other systems and its users.

3.2.4 *MPEG-1 summaries creation*

The MPEG-1 summaries creation module provides a group of complementary outputs to the hierarchical summary description created in the process for summarization. It has the only purpose of, using the hierarchical summary description, creating MPEG-1 files with the summaries desired by the user. Note that, from the same hierarchical summary description an infinite number of MPEG-1 files can be created as the user can choose to produce summaries by time or by arousal level as explained in the before section.

Chapter 4 will provide an in-depth description of the algorithms used for all modules and sub-modules presented in this chapter.

Chapter 4

Processing for Summarization

This chapter will offer the reader a detailed description of all algorithms implemented in the context of the summarization application developed. The description will be made by appearance order in the system architecture, i.e. from the low-level features extraction module to the hierarchical summary description creation module. The system was implemented in C# language using Microsoft Visual Studio 2005; this choice will be motivated in Chapter 5.

4.1 Low-level features extraction

As explained and justified in Chapter 3, motion intensity, shot cut density and sound energy were the low-level features chosen to achieve the objectives defined, this means modeling the user arousal. In this sense, this section intends to motivate and detail the algorithms used in the low-level features extraction module. In common, the three algorithms only have the output format, this means an eXtended Markup Language (XML) file containing all the information needed by each feature's arousal modeling module. This format option is justified by the wide usage of XML in these days. From now on, motion will be treated as the first feature, shot cut localizations pr shot lengths as the second feature and sound energy as the third feature, despite the fact that all have the same importance in the system. Therefore, the first algorithm to be explained will be the one used to extract motion information. Bear in mind that the audiovisual content is assumed to be coded in MPEG-1 format, as the motion information and sound energy extraction algorithms were designed to deal with material coded with this standard.

4.1.1 Motion information extraction

As stated in Chapter 3, in order to model arousal using motion information it is necessary to extract each frame's motion vector values since the input audiovisual material is already MPEG-1 coded and thus motion vectors are available in the coded stream. To provide these values in a structured manner to the arousal modeling module, a XML

file containing all motion vectors values associated to their corresponding frame index is created as output of this sub-module. Because the motion information is always the same for the same codec content, when a user repeats the motion information extraction process for the same audiovisual content, he/she can start the process from scratch or he/she can use the motion information XML file created already in previous extraction processes reducing the processing time. Figure 28 shows the architecture of the motion information extraction sub-module.

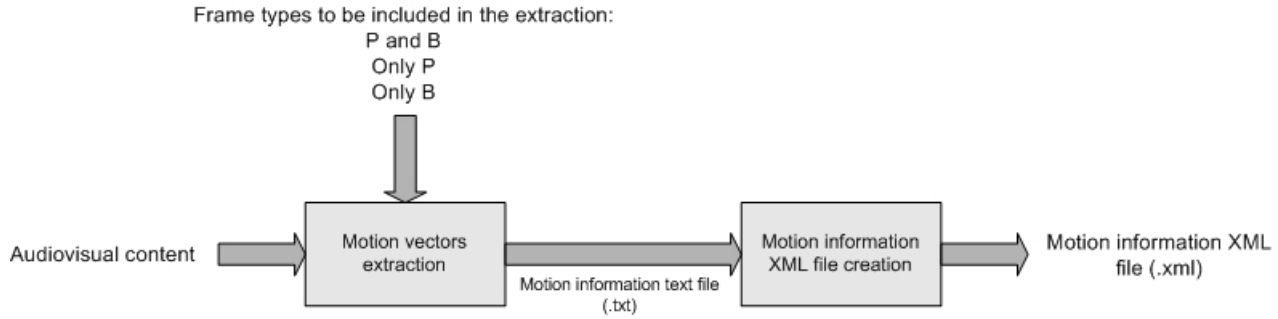

Figure 28 – Motion information extraction architecture.

Each of the architecture sub-modules will be described and explained in the following sections.

4.1.1.1. Motion vectors extraction

To extract each frame's motion vectors values from the audiovisual coded content, a MPEG-1 decoder developed by Ascenso and Hidalgo [25], capable of providing the motion vector values by extracting them directly from the MPEG bitstream, was adapted. The adaptation consisted in the creation of a cycle that, in each iteration, parses a frame extracting its motion vector values and writing them to a textual file, along with the frame type and the frame number. This algorithm, despite being quite simple, proved to be computationally expensive in the low-level extraction processes. For this reason, the software was also amended to allow the user to select the type of frames for which motion information should be gathered – P, B, or P and B - with the aim to fasten up the process, even if at the cost of a less precise result.

MPEG-1 coded frames are organized in Groups of Pictures (GOPs) and can be of three types: I, P and B. I frames do not use temporal prediction and, therefore, do not have motion vector values. P frames use forward prediction, i.e. perform motion estimation based on a precedent frame. B frames use forward and/or backward prediction which means that B frames do motion estimation based on precedent and/or future frame. In this manner, P frames use precedent I or other P frames to estimate motion while B frames use precedent and/or future I or P frames to compute their motion vectors. The GOP size – number of frames in a GOP - is normally designated as N and can vary during the video stream: the GOP size typically corresponds to the number of frames between two I frames, including one of them. Figure 29 shows a typical MPEG GOP: as shown in Figure 29, typically, in each GOP, B frames are present in higher number than P frames as they exploit the audiovisual content temporal redundancy better because of their forward and/or backward prediction ability, being more useful for MPEG coding.

Figure 29 – MPEG Typical Group of Pictures (GOP).

The user can then choose to include in the motion information gathering process, all P and B frames which are the only ones that have motion vectors, or only P frames or only B frames. This choice has direct influence on the algorithm's performance as P frames, in addition to existing in fewer numbers than B frames, have much less motion

vectors to extract as they only use forward prediction while B frames can use forward and/or backward prediction. In this way, if a user chooses to include only P frames, the motion information extraction process will perform much faster but the ending result will be less complete and precise. If the choice falls only over B frames, the result will be more precise but the process will take more time. Consequently, the option that will present the most precise results will be the one that includes all P and B frames at the cost of a much slower process.

As the MPEG-1 decoder was written in C++ language, a dll library was built in order to be included in the developed application project. The output of this sub-module is the text file written during the cycle's computational life and consists on all frame indexes together with their motion vector values. This text file will be passed to the motion information XML file creation sub-module as shown in the architecture in Figure 23. After the XML file creation, the temporary text file is deleted in order to save disk space.

4.1.1.2. Motion information XML file creation

As referred before, in order to store the information needed by the motion intensity arousal metric computation module to model arousal, a XML file is created from the text file resulting from the motion vectors extraction process. This XML file creation process is straightforward. It receives as input the text file and creates a structured XML file with all motion vectors values associated to their frame index and respective frame type. Figure 30 shows the Document Type Definition (DTD) of the created Motion information XML files. The DTD is one of several XML Schema languages which specify the syntax of a XML file.

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT video (filename)>
<!ELEMENT filename (motionextraction)>
<!ELEMENT motionextraction (lastFrame, frame+)>
<!ELEMENT lastFrame (index)>
<!ELEMENT index (#PCDATA)>
<!ELEMENT frame (type, index, mvs)>
<!ELEMENT type (#PCDATA)>
<!ELEMENT mvs (#PCDATA)>
<!ATTLIST filename
  name CDATA #REQUIRED
>
```

Figure 30 – DTD of Motion information XML files.

As Figure 30 shows, Motion information XML file has the following structure:

- ***video* element –** The root element of the XML file; contains only one child element, the *filename* element.
- ***filename* element –** Contains as attribute the path for the audiovisual content original file. Has also one child element, the *motionextraction* element.
- ***motionextraction* element –** Has several child elements, one related to the last frame – *lastFrame* element – and one for each P or B frame, depending on the user's choice of parameters – *frame* elements.
- ***lastFrame element*** – Contains the index of the last frame that will be important in the arousal metrics computation.
- ***frame element* –** Include the frame's type (P or B), its index and all its motion vectors.

This XML is useful to pass the feature information to the arousal modeling module in a structured manner and it is also useful as a user can skip the motion vectors extraction process if a XML file referring to the same audiovisual content has been already created. However, the user has to repeat the process, if he/she wants to change the input parameters in order to obtain different arousal values. If he/she wants a faster summarization process, the use of the XML file, if available, is (more than) advisable as the entire motion extraction process is skipped. Figure 31 shows part of a Motion information XML file which is the output to the Motion information extraction sub-module.

```
- <video>
 - <filename name="Cocas BMW.mpg ">
  - <motionextraction >
   - <lastFrame >
      <index>774</index>
    </lastFrame >
   - <frame>
      <type >I</type >
      <index>0</index>
      <mvs>0</mvs>
    </frame>
   - <frame>
      <type >P</type >
      <index>3</index>
      <mvs>0 0 -15 0 0 0 0 0 52 0 32 0 0 0  -16 0 -15 0 0 0 16 0 0 0  -54 9 2 0 -4 5 -15 15 0 0 0 0 0 0 0 0 0 0 0 0 0 0 3 0 31   -6 19 0 32 -1 16 -1 16 -32 0 -7 17 -7 17 -
        15 1 0 16 1 31 0 0 0 0  -16 16 22 32 -15 17 -1 19 -13 13 -5 0 0 0 0 0 0 12 0 8 0 16 0 0 0 0 0 0 0 32 0 0 26 32   -15 13 -15 15 28 20 8 0  -1 15 0 0 -16 15 -15
        13 0 12 -15 13 -21 13 -32 -32 3 16 0 0 0 0 0 5 0  -32 0 -32 0 31 -1 17 0 -32 0 0 -64 -64 -1 11 0 0 2 0 16 -32 32 32 -1 13 -4 30 -16 16 -30 48 16 16 -16 16
        -50 15 0 0 0 0 0 0 0 37 0 16 2 0 0 0   -7 15 -16 -64 19 -15 3 31 -46 -16 13 0 6 -16 2 -1 16 0 26 48 0 6 -47 48 32 32 16 -16 -32 43 -45 33 -46 -21 0 0 0 0 0 -
        53 0 38 0 16 0 48  -14 -48 1 15 -37 0 -48 -32 -46 -70 16 -16 -16 0 0 0 -16 -1 15 -32 -16 -16 12 0 0 0 -28 16 -25 15 -22 11 -21 7 -33 12 -41 13 0 3 0 48 0
        0 -5 0 33 -15 -13 14 0 0 0 0 16  -1 -15 3 35 6 0 9  -15 0 -48 -32 0 0 66 -20 0 0 -21 11 -31 13 -30 12 -33 12 32 -9 -34 12 0 6 0 1 0 -17 6 -16 -25 11 -58 -16
        0 0 -15 15 33 30 0 0  -15 17 0 -15 0 0 1 5 47  -10 -31 13 0 0 -31 13 -30 11 -33 13 0 0 4 3 0 0 0  -15 0 -32 0 0 -10 13 32 -19 16 -60 0 39 0 0 0 0 0  -3 -29 -5
        -1 1 1 3 1 4 5 0  -19 8 0 0 -25 11 79 1 0 0 0 0 0 0 0 16 0 9 0   -16 0 8 -21 5 -13 11 -15 13 0 0 0 0 0 0  -15 13 0 0 -3 2 -3 2 0 0 0 0 30 -17 0 0 0 0 0 0 0  -3 0 0 0
        0 0 -15 0 14 0 31 0 0 0 0 0 0 0 0 4   -25 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0    -1 0 -32 0 0 -16 13 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 21 65 0 0
        0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 23 0 0 0 0 0 0 0 0 0 36 21 0 0 0 0 0 0 0 0 0 0 0 0 0     -4 -19 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0   -9 0 21 0 0 0 0 0 0 0 0 0 0
        0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 11 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0
        -20 0 0 16 23 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0   -27 13 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  -5 0 -15 0 0 0 0 0 0 0 0 0 0 0  -23 15 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
        0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  -21 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  -15 0 -15 0 -31 0 -47 0 -19 0 -1 0 -15 0 -11 0 9 0 -1 0 11 0 -10 0 -7 0 0 -1</mvs>
    </frame>
   - <frame>
      <type >B</type >
      <index>1</index>
      <mvs>0 0 0 0 -11 0 9 0 0 0 0 0 29 0 0 0 0 0 0 0 0 5 0 35 0 0 0 0 0 0   -15 0 0 0 0 0 0 0 0 0 0 0 0 1 1 3 0 0  -16 16 0 0 0 0 0 0 0  -47 0 0 0 0 3 0 0  -9 7 0 0 5 5 0 0 1 5 0 0 1
        5 0 0 0 4 0 0 1 6 0 0 1 5 5 7 0 1 5 7 0 4 5 7 0 4 1   -13 6 21 0 -12 0 6 0 -12 48 48 18 32 -16 0 0 -16 -1 3 5 -11 0 5 31 -18 0 5 17 -16 -16 -16 17 -16 0 0 0 0 0
        0 0 0 0 0 46 16 -21 32 46 16 -21 32 11 -11 -7 7 11 -9 0 5 11 -9 -1 9 -19 -23 -1 9 5 -1 -6 6 -16 -17 0 5 -16 -17 0 6 0 -9 0 5 0 -9 0 16 0 -9 0 16 0 -9 0 16 0
        -16 0 0 0 0 0 16 0 0 0 16  -16 -16 0 16 4 -11 -1 7 8 -8 0 6 2 -32 16 -16 2 -32 0 6 2 -32 0 6 -14 -7 0 6 16 -16 0 6 16 -16 0 6 14 -11 -15 5 14 -11 -3 5 14 -11
        -5 5 14 -11 5 6 -37 -43 1 6 -47 -64 7 5 -2 -11 0 4 -6 -7 0 5 -6 -7 0 5 0 -16 0 16 5 -33 0 0 0 0 0 0  -8 15 0 0 -16 -15 0 0 -16 -15 0 0 -16 -15 3 5 4 32 0 0 0 0
        0 0 0 0 0 0 0 16  -15 0 -8 -1 5 16 -8 -1 5 16 -8 45 2 12 -8 1 5 12 -8 9 4 17 -8 9 4 15 -7 9 4 -49 13 0 0 0 0 0 0  -7 -15 -11 5 -7 -15 0 1 -7 -15 0 -16 -7 -15 0
        0 0 0 0 0 0 -16 0 0 0 -21 -32 16 -32 -16 -48 -25 -23 16 -30 -10 -23 16 -14 32 -23 16 -14 32 2 -32 -24 32 0 -16 -2 6 0 -16 40 49 0 -16 -16 0 0 -16 46 -16
        0 -16 0 0 -16 16 0 0 45 32 0 0 14  -26 0 0 -47 -16 0 0 16 -7 0 0 0 0 1 7 1  -7 -12 6 -13 -10 0 4 0 -13 0 -16 0 -11 0 -16 0 16 4 33 0 -16 4 33 -14 -48 4 33 -31
        -59 -16 0 20 0 -30 -36 -62 -48 0 0 0 0 0 0 16  -49 0 0 0 -16 0 0 3 -16 -42 -17 3 -16 -64 -32 3 -16 -64 -32 30 -16 -64 -32 17 16 -64 -32 46 -42 -6 0 46 -42
        -23 15 32 -56 -7 5 15 -7 -9 5 15 -7 -11 5 15 -7 -10 4 15 -7 0 -1 0 -5 0 -1 0 -1 0 -1 0 -11 -1 0 43 16 -1 0 1 -1 -78 -21 1 -1 -13 -1 -55 7 1 10 -55 7 0 10 -
        55 7 17 16 -55 7 -35 -1 -55 7 -42 -3 -26 -15 0 0 0 0 0 0 1 5 20  -48 1 5 16 -45 4 -48 16 -45 6 -24 -7 5 6 -24 -9 5 6 -24 -9 5 6 -24 -11 5 6 -24 -11 5 26 -8 -
        12 6 26 -8 0 1 0 -2 0 -17 0 -2 0 -17 20 16 16 1  -16 16 -50 17 -55 16 1 -25 -58 -7 -5 -1 15 1 -3 3 1 -18 0 0 0 0 11  -10 47 10 -59 -17 47 10 -59 -17 8 -8 -8
        -6 -48 -24 0 0 0 0  -11 6 21 -8 -9 6 19 -8 -9 5 17 -7 -9 5 17 -7 -9 5 21 -7 -9 5 27 -8 -7 5 19 -5 0 0 19 -5 0 0 19 -5 0 0 0 0 0 1 0 0  -1 -9 0 0 18 37 -16 -3 18
        37 -15 -16 18 37 3 -32 -1 3 0 15 0 0 0 0  -5 5 0 0 1 -15 0 0 -7 3 -9 -12 -1 2 1 0 15 -6 16 -10 -7 5 16 -10 -9 5 21 -7 -7 4 21 -5 -9 5 21 -5 -20 5 -32 17 67 -
        10 34 13 -7 5 16 -5 31 -11 -18 -16 -5 -2 3 -13 0 17 3 -13 0 15 0 0 0 15 0  -16 6 -3 0 -16 -4 5 0 -16 34 -63 20 -48 3 -77 -6 -52 -6 -51 -2 -47 -6 7 -28 -9 9
```


Figure 31 – Example of part of a motion information XML file.

4.1.2 Shot cut detection

For the same reasons stated for the first feature, the shot cuts extraction module is divided in two sub-modules: a shot cut detection algorithm based on luminance and saturation histogram difference to detect shot cuts, and a shot cuts information XML file creation sub-module with the purpose of structuring the information delivered to the next modules. Figure 32 shows the shot cut detection architecture.

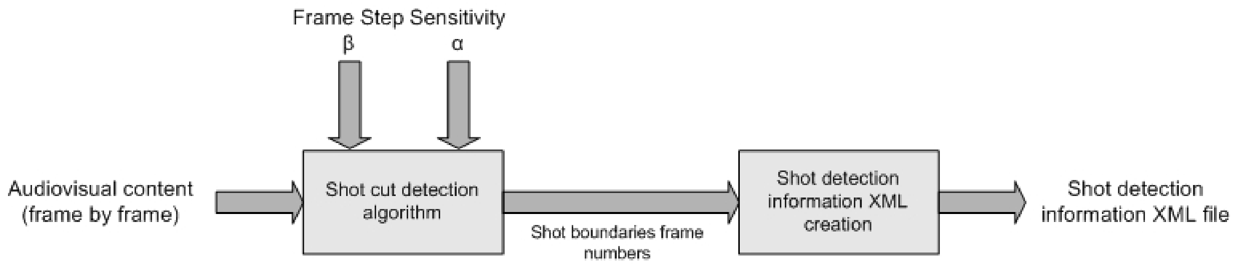

Figure 32 – Shot cut detection architecture.

4.1.2.1. Shot cut detection algorithm

There are many algorithms available in the literature to detect shot cuts, both in the compressed and uncompressed domains [26][27][28][29] . For the system developed, a method using luminance and saturation histogram differences to detect the shot cuts has been adopted with some similarities with the method proposed on [26]. The basic approach is to use the difference between luminance and saturation histograms of consecutive frames to detect the shot boundaries. If the difference of luminance histograms added to the difference of saturation histograms is superior to a determined threshold, a shot boundary is assumed. Together, these two features seem to provide a good indication of overall changes as, when one fails to reflect these changes, the other can compensate that failure. DirectShow [30] was used to extract each frame from the audiovisual content via its *SampleGrabber* method. DirectShow is a multimedia framework and API developed by Microsoft for software developers to perform various operations with media files. It was of great

use in the implementation of the application developed as it allowed to playback MPEG-1 files and took part in other low-level features extraction algorithms, namely in the sound information extraction module as will be described later.

Beside, of course, the video frames themselves, the algorithm needs to use two parameters:

- **Shot cut detection process sensitivity, α –** The sensitivity, α, determines the value of the threshold to be compared with the overall histogram difference. The α's range is [0-1] and represents the percentage of pixels that can have different values of luminance and saturation without a shot cut being assumed. The threshold formula is:

$$t = width * height * 2 * \alpha \quad (18)$$

  In (18), width and height refer to the audiovisual content's width and height, i.e. the number of pixels in the vertical and horizontal dimensions. The multiplication by 2 is applied as two histograms are computed in the process.

- **Frame step, β –** The choice of β is given to the user, in order to permit, for example, a faster process at the cost, of course, of some precision. The frame step is available to the user, mainly because of performance. A frame step of 1 implies that the histograms of all consecutive frames are compared and brings the best results; however, it implies a slower process while a higher frame step makes the process faster but will present less precise results as entire shots will have been leaped in the process.

Shot cut detection is, in this way, a full customizable process as the user can change α and β in order to obtain the best possible results for his/her intentions. Described the inputs, a short walkthrough of the algorithm is presented:

1. **Luminance and saturation histogram calculation –** Starting with the second frame**,** the first step of the algorithm is to compute the luminance and saturation histograms. Histograms are bin-wise, i.e. they represent the number of pixels common to each luminance and saturation value. For example, if 27 pixels have the luminance value of 137, the histogram's value for 137 would be 27. The same is applied, of course, for saturation and for all other values, other than 137. Therefore, for each pixel of each frame, and based on the values of Red, Green and Blue – R, G, B, the luminance, Y, and saturation, σ, are computed as indicated in the following:

$$Y = 0.3R + 0.59G + 0.11B \quad (19)$$

$$\sigma = \sqrt{\frac{(R-\mu)^2 + (G-\mu)^2 + (B-\mu)^2}{3}} \quad (20)$$

   In (20), μ represents the R, G, B brightness which is defined by the mean of R, G and B. After calculating the values of luminance and saturation for each pixel, the histogram indexes corresponding to the bins where the two values fit in each of the two histograms are incremented. When all frames are processed, the luminance and saturation histograms calculation is concluded.

2. **Frame by frame difference between histograms calculation -** The calculation for each frame of the difference between histograms is quite straightforward. After computing the histograms for a frame, they are compared with the histograms calculated for the previous frame: the total difference between histograms is computed as the sum of the differences for each bin of each histogram. A chart showing the total difference for the luminance plus saturation histograms along time, for a short advertising movie, is shown in Figure 33. In Figure 33, the peaks of the chart correspond to the shot cut boundaries.

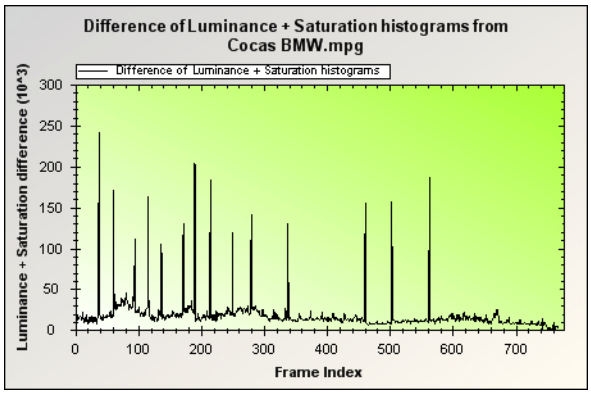

Figure 33 – Difference of luminance plus saturation histograms for a short advertising movie.

3. **Decision about frame's condition –** The decision if a frame should be considered as a shot boundary or not is done by simply comparing the value calculated in step 2. with the threshold defined at the beginning of the process. If the difference value is superior to the threshold, the frame in question is considered as a shot boundary; otherwise, it is considered to belong to the same shot as the previous frame. The frame indexes corresponding to frames considered to be shot boundaries are inserted in an *ArrayList* in order to create after the XML file describing the shot detection information.
4. **Iterating the process –** After processing one frame, the process continues with the next frame and returning to step 2 above.

The algorithm above is rather simple but effective for the purposes of the system developed. Since the intensity of shot cuts will be determined later, there are here no constraints imposed on the time distance between consecutive detected shot cuts.

4.1.2.2. Shot detection information XML file creation

The output of the shot cut detection algorithm sub-module, as for the motion information extraction sub-module, is a XML file that intends to provide, in a structured manner, the shot cut information extracted by the luminance and saturation histograms difference algorithm to the arousal modeling module. In addition, and again as for the motion information extraction, the user is able to skip the shot cut detection process if it was previously done and thus the corresponding XML file is already available. If the user wants to run the process with different parameters, he/she should start the process from scratch; otherwise, he/she can choose to use the XML file resulting from a previous process.

The DTD containing the syntax of the shot detection information XML file is presented in Figure 34.

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT video (filename)>
<!ELEMENT filename (hardcutdetection)>
<!ELEMENT hardcutdetection (shot+)>
<!ELEMENT shot (index, frame, position, type)>
<!ELEMENT index (#PCDATA)>
<!ELEMENT frame (#PCDATA)>
<!ELEMENT position (#PCDATA)>
<!ELEMENT type (#PCDATA)>
<!ATTLIST filename
  name CDATA #REQUIRED
>
```

Figure 34 – DTD of Shot detection information XML file.

The DTD presented in Figure 34 for the shot detection information XML file is not very different from the DTD used for the motion information XML file. Its structure is described next:

- ***video* element –** As in motion, it forms the root element of the XML file and contains only one child element, the *filename* element.
- ***filename* element –** Contains as attribute the path for the audiovisual content original file. Has also one child element, the *hardcutdetection* element.
- ***hardcutdetection* element –** Has several *shot* child elements.
- ***shot element*** – Contains the cut number – *index* element; the frame index boundary – *frame* element; its position in seconds – *position* element and the type of cut – *type* element. In this algorithm, only hard cuts are detected and, therefore, the *type* element has always the "hardcut" value

Figure 35 shows an example with part of a shot detection information XML file which contains the elements described above: the cut number – *index* element –, the frame index boundary – *frame* element – as well as its position in seconds – *position* element –. It also includes information related to the type of the cut – *type* element –, although the selected algorithm only detects hard cuts.

```
- <video>
  - <filename name="Cocas BMW.mpg ">
    - <hardcutdetection >
      - <shot >
          <index>0</index>
          <frame>0</frame>
          <position >5,5E-06</position >
          <type >hardcut</type >
        </shot >
      - <shot >
          <index>1</index>
          <frame>50</frame>
          <position >1,9999945 </position >
          <type >hardcut</type >
        </shot >
      - <shot >
          <index>2</index>
          <frame>75</frame>
          <position >2,9999945 </position >
          <type >hardcut</type >
        </shot >
```

Figure 35 – Example of part of a shot detection information XML file.

4.1.3 Sound information extraction

The extraction of the third feature, sound energy, is the fastest from the computational point of view; its architecture is presented in Figure 36.

Figure 36 – Sound information extraction architecture.

The audiovisual content, in MPEG-1 format, is decoded and written in a WAV file where audio is not compressed anymore. Next audio samples will be read from the WAV file and the sum of the squares of the values for each frame as well as the frames' index will be stored in a sound information XML file, as done for the other two features.

4.1.3.1. MPEG to WAV conversion

MPEG-1 to WAV conversion is the first step in the sound information extraction process. The obvious purpose of this sub-module is to convert the input MPEG-1 coded file with the audiovisual content into an uncompressed domain WAV file. A WAV file consists of three chunks of information; a Resource Interchange File Format (RIFF) chunk that identifies the file as a WAV file, the FORMAT chunk that identifies the file parameters, such as the sample rate, the number of channels, the number of bytes per seconds or per sample, etc. and also the DATA chunk containing the actual audio samples.

The MPEG to WAV conversion algorithm used is an adaptation of C++ to C# programming language of Khan's work on [31]. The architecture of this sub-module is presented in Figure 37.

Figure 37 – MPEG to WAV conversion sub-module architecture.

Khan used the DirectShow framework from Microsoft in his work. MPEG to WAV conversion sub-modules correspond to DirectShow filters. This set of filters will be briefly described next, providing also a short walkthrough of the MPEG to WAV conversion sub-module:

1. **MPEG-1 systems splitter –** The MPEG-1 content splitter has the purpose of splitting the MPEG-1 Systems stream (including audio, video, multiplexing and synchronization data) into its component audio and video sub-streams. In this manner, the sub-module receives the MPEG-1 file as input and splits it into the audio and video sub-streams. Here, only the audio stream is relevant since the audio samples are needed. So, the output of this sub-module is the audio sub-stream that will be passed to MPEG-1 Audio decoder.
2. **MPEG-1 audio decoder –** The MPEG-1 Audio decoder filter decodes MPEG-1 Audio layer I and layer II to Pulse-code Modulation (PCM) samples. It receives the audio stream as input and decodes it to PCM in order to write the audio samples to a WAV file. In implementation terms, the writing of the WAV file to the disk is performed by a DirectShow WAV Dest filter that must be previously installed in the running machine as the Windows Operative Systems does not install WAV Dest filter by default.

4.1.3.2. Sound energy information XML file creation

The audio samples present in the previously created WAV file are extracted next in order to be created an associated XML file with that information. To achieve this goal, the WaveUtility C# classes, found in [32], were used. These classes permitted to obtain all information needed from the WAV file regarding the audio samples, such as the number of bits per sample, the number of samples per second or the number of audio channels – 1 for mono, 2 for stereo, in addition, of course, to the audio samples themselves.

Therefore, to successfully gather the audio samples corresponding to each video frame, the first thing to compute is the number of audio samples corresponding to the time of a video frame. This is calculated according to the formula:

$$SamplesPer\,VideoFrame = \frac{SamplesPer\,Second}{VideoFrame\,Rate} * NumberOfAu\,dioChannel\,s \qquad (21)$$

The number of samples per second and the number of audio channels were extracted from the WAV file bitstream while the video frame rate was extracted from the MPEG-1 bitstream. After calculating the number of audio samples corresponding to each frame, a simple loop was implemented to gather the audio samples for each frame, and compute the sound energy here defined as the sum of the squares of the samples. This sound energy is after stored in the sound information XML file, together with the associated frame index. This XML file contains the sound energy and not all the audio samples because this is the only information which will be needed after to model arousal based on sound; this option saves disk space as a XML file with all the audio samples would become much bigger.

Resuming, as for the other feature extraction processes, a final XML file is created to provide the sound related information obtained to the arousal modeling process, in a structured manner; this also allows the user to skip the sound energy extraction process, if this file is already available, obtaining results in a faster way. The sound information XML file DTD is presented in Figure 38.

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT video (filename)>
<!ELEMENT filename (audioextraction)>
<!ELEMENT audioextraction ((lastFrame, frame+))>
<!ELEMENT lastFrame (index)>
<!ELEMENT index (#PCDATA)>
<!ELEMENT frame (index, sumAudioSamples)>
<!ELEMENT sumAudioSamples (#PCDATA)>
<!ATTLIST filename
  name CDATA #REQUIRED
>
```

Figure 38 – DTD of Sound information XML file.

The structure of the Sound information XML presented on the DTD is similar to the ones described before for the other two features. The structure is, therefore, similar:

- ***video* element –** As before, represents the root element of the XML file and has, as it's only child element the *filename* element.
- ***filename* element –** Has one only attribute which is the path for the audiovisual content original file. Has also one child element, the *audioextraction* element.
- ***audioextraction* element –** Similar to motion, has many child elements, one is related to the last frame – *lastFrame* element – and all the others are related to each video frame – *frame* elements.
- ***lastFrame element*** – Indicates the index of the last frame that will be important in the sound energy arousal metrics computation.
- ***frame element* –** Include the frame's type (P or B) and the sum of the squares of all its audio samples.

Figure 39 shows an example of part of a sound information XML file, following the DTD presented above, for a short advertising movie.

```
- <video>
  - <filename name="Cocas BMW.mpg ">
    - <audioextraction >
      - <lastFrame >
          <index>776</index>
        </lastFrame >
      - <frame>
          <index>0</index>
          <sumAudioSamples>8046</sumAudioSamples>
        </frame>
      - <frame>
          <index>1</index>
          <sumAudioSamples>1615073958 </sumAudioSamples>
        </frame>
      - <frame>
          <index>2</index>
          <sumAudioSamples>27563893722 </sumAudioSamples>
        </frame>
```

Figure 39 – Example of part of a sound information XML file for a short advertising movie.

4.2 Arousal modeling

After extracting the selected low-level information in the low-level features extraction sub-module and storing it in XML files, the next stage in the process for summarization is arousal modeling. Arousal modeling is done in two main steps: Arousal metrics computation and Fused arousal computation. The final arousal curve A(k) can be seen as the fusion of the $G_i(k)$ functions which represent the arousal resulting from feature *i* information along the video frames. The computation of the $G_i(k)$ functions will be made in the Arousal metrics computation module while A(k) will be generated in the Fused arousal computation module. These processes will be explained in the following sections.

4.2.1 *Arousal metrics computation*

As referred before, Arousal metrics computation is the immediate stage after low-level information extraction. Each feature's arousal metric is computed independently, originating an individual arousal function, $G_i(k)$, as output. After

the individual arousal metrics are computed for each feature, a final fused arousal curve will be generated. In the process of determining an arousal curve for each feature, to be given to the fused arousal process, three main steps have to be taken: computing the associated metric, smoothing that same metric, and scaling the smoothed curve.

4.2.2.1. Motion intensity arousal metric computation

The goal of the motion intensity arousal metric computation module is, as the name says, to compute a metric expressing the arousal from motion intensity for each video frame *k*.

- Motion intensity metric computation

The first step in building a metric capable of representing the arousal felt by an audience in terms of motion is to identify what increases or decreases viewer arousal regarding motion. As discussed in Section 3.2.1.1, when justifying the low-level features choice, a video sequence with fast motion is typically associated with a high degree of excitement experienced by the viewer. Therefore, when building a metric aiming to represent arousal from motion information, the aim is to measure the motion intensity for each video frame *k*. Having all motion vectors values for each frame stored in the motion information XML file, a possible way to represent the motion intensity for each video frame *k* may be to calculate the average motion vector magnitude for each frame and dividing it by the maximum possible motion vector magnitude for that frame, obtaining therefore, for each video frame *k*, the motion intensity in relation to its maximum, possible in percentage.

In this context, it is proposed here to adopt as motion intensity metric m(k), the function [7]:

$$m(k) = \frac{100}{\left|\overrightarrow{v_{k\ \max}}\right|} * \frac{\sum_{i=1}^{TotalMV} \vec{v}_i(k)}{TotalMV}\% \qquad (22)$$

In (22), and as proposed above, the motion intensity for each frame, m(k) corresponds to the average magnitude of all motion vectors values extracted, regardless of its direction, horizontal or vertical – the main factor is magnitude – normalized by the maximum possible magnitude of a motion vector for that frame, $|v_{kmax}|$. Motion vector values are read from the motion information XML file created in the motion information extraction process and $|v_{kmax}|$ value varies from frame to frame as frames estimate their motion based on different frame anchors (at difference time distances). As explained for the motion information extraction module, frames are organized in GOPs, and motion estimation depends on the frame coding type, e.g. P or B. Therefore, when computing m(k) for a P frame, $|v_{kmax}|$ will be 64 – the maximum possible magnitude value for a motion vector with ½ pixel resolution – times the number of frame periods between the frame in question and its anchor. For example, if the P frame anchor is two frames away, $|v_{kmax}|$ will be 64×2 = 128. When computing m(k) for a B frame, the same type of reasoning is applied, but this time it is important to keep in mind that B frames have two anchors: one for its forward prediction and another for its backward prediction. In this context, two averages will be computed: one for each type of prediction, with its own $|v_{kmax}|$ and motion vectors, and then the average of the averages is calculated, originating the m(k) value.

- Smoothing filtering

The smoothing filter is applied to all arousal curves – independent and fused - produced in the summarization process. The chosen smoothing filter was the so-called Kaiser window [33] as it proved to produce the desired smoothing results in modeling arousal [5][6][7]. The Kaiser window is a window function w_k that is defined by:

$$w_k = \begin{cases} \dfrac{I_0(\pi\alpha)\sqrt{1-\left(\dfrac{2k}{N}-1\right)^2}}{I_0(\pi\alpha)} & if\ 0 \le k \le N \\ 0 & otherwise \end{cases} \qquad (23)$$

I_0 is the zeroth order modified Bessel function [34] of the first kind, Bessel functions of the first kind are solution's of Bessel's differential equation which are finite at the origin and diverge as x approaches zero. N is an integer and represents the length of the window and α is a real number that determines the shape of the window. An example of a Kaiser window with N=100 and several values of α is shown in Figure 40.

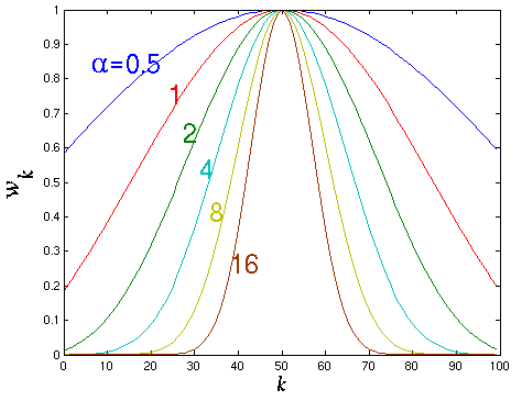

Figure 40 – Example of a Kaiser window function for N = 100 and α = 0.5, 1, 2, 4, 8 and 16 [33].

By default, the peak of the Kaiser window function is at the center of the window, when k = N/2, decaying exponentially to the windows edges. As Figure 40 shows, the larger the value of α, the narrower the window becomes, with α=0 corresponding to a rectangular window.

For the purpose of this work, the Kaiser window accurately fulfills the function of smoothing the curves resulting from the arousal metrics and fused arousal computational processes. After those computational processes take place, its resulting curves are convoluted with the Kaiser window, allowing the summarization process to better model the viewer's reaction to the audiovisual content. This happens mainly because, as the curves become smoother, the gradual increase and decrease of the arousal values are much similar to the arousal experienced by the audience than abrupt changes directly resulting from the low-level features extraction processes. In this work, the values of N are, typically, the total duration of the video divided by 15 and α=5 as these were the values that proved, after intensive testing, to be more suitable for the desired purposes.

In the motion intensity arousal metric computation case, after computing m(k), the curve needs to be smoothed and to do so, a mathematical convolution with the Kaiser window described above is applied (24), originating a function mk(k) with N and α taking the values indicated [7]:

$$mk(k) = m(k) * K(N,\alpha) \qquad (24)$$

Note that the mathematical convolution of any metric with the Kaiser window will produce a curve in a different scale range and, therefore, scaling is needed, precisely to scale back the curve for percentage values.

- Scaling

After smoothing the m(k) metric by convolution with the Kaiser window, the values of the smoothed curve are not in percentage values, as said above. As explained in Section 3.2.2.3, in order for the arousal curves resulting from the three features to be comparable and, consequently, to create a coherent final arousal curve, the values should be in

percentage and, therefore, scaling arises as fundamental. The scaling of the convoluted metric back to percentage values represents the final step on modeling the motion intensity arousal, leading to $G_1(k)$, computed as [7]:

$$G_1(k) = \frac{\max(m(k))}{\max(mk(k))} * mk(k)\% \quad (25)$$

Resuming, (25) represents the motion intensity metric convolved with the Kaiser window – mk(k) – scaled back to percentage values as expresses the ratio between the max of the m(k) original function with the max of the convoluted function. Figure 41 shows, on the left, the motion intensity curve without scaling, with a maximum value of **25000** and, on the right, $G_1(k)$ – the arousal curve from motion intensity, after scaling to percentage values, computed for a football sequence.

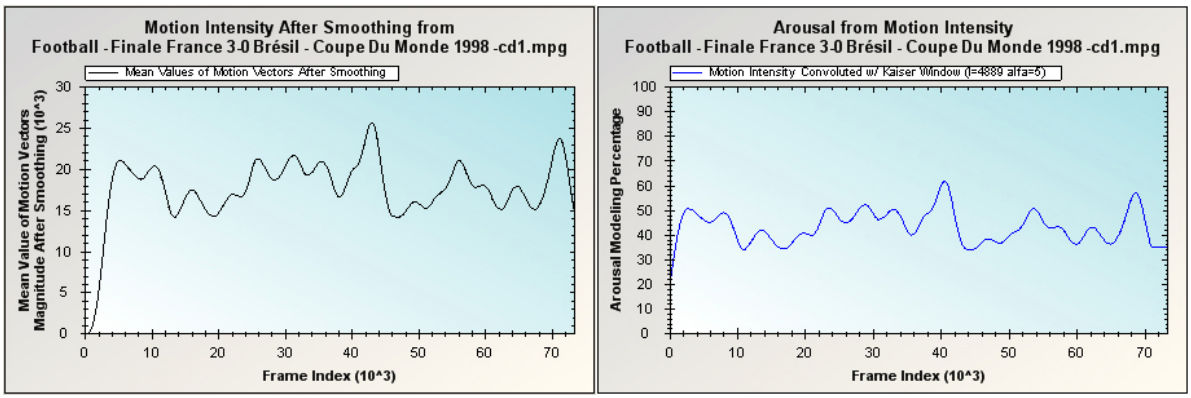

Figure 41 – Example of motion intensity (a) before and (b) after scaling the motion information arousal curve, $G_1(k)$ for a football sequence.

4.2.2.2. Shot cut density arousal metric computation

The shot cut density arousal metric computation process is similar to the motion intensity process. At first, a function c(k) intended to relate the viewer's arousal with shot duration is computed and after the convolution with the Kaiser window and the scaling to percentage values are applied.

- Shot cut density metric computation

The same type of reasoning made for motion information has to be made now for shot cut detection. Here, the goal is to relate the shots duration with the arousal experienced by the audience. As justified in Section 3.2.1.1., shot duration is deeply related to the type of excitement provoked on the audiovisual content's viewer. Shorter consecutive shots are normally related with moments of fast action while long shots often mean calmer and more relaxing segments in the audiovisual content. With this in mind, the objective of the defined shot cut density arousal metric to be proposed is to compute coherent values with each shot's duration. In this manner, the metric should result in higher values for shorter shots and lower values for longer shots. The metric adopted [7] to fulfill these requirements, c(k), was:

$$c(k) = 100 * e^{\left(\frac{1-(n(k)-p(k))}{\delta}\right)}\% \quad (26)$$

In (26), n(k) and p(k) represent, respectively, the frame index of the next and previous shot boundaries in relation to the current frame *k*. These index values are retrieved from the XML file created as output of the shot cut detection process. The difference between 1 and the shot duration (n(k) – p(k)) in e^x implies using e^x only for x<0. e^x has value 1 for x=0 and tends to 0 when x tend to -∞; consequently, the smaller the difference between n(k) and p(k), and therefore the duration of the shot, the closer the outcome value is to 1. The opposite, i.e. higher differences between n(k) and p(k), results in lower c(k) values for longer shots. As e^x tends to 0 when x tend to -∞ it is advisable to constrain the x values

to an adequate range. This is the function of the constant δ which will determine the shape of the curve. A high value of δ will result in a curve where all values are too close of 100% as all x values are close to 0, while a δ value too small will result in a curve always near 0% as shot durations are normally near the hundreds or thousands of frames and, therefore, the e^x value will be much smaller. A good proven value for δ is around 300 as the resulting curve shows fluctuations adequate for an arousal modeling process.

- Smoothing filtering

The smoothing filtering is applied in the exact same manner as for the motion intensity process. To smooth the c(k) metric, the same mathematical convolution of c(k) with the Kaiser window is performed to generate ck(k); the N and α values are the same. Again, the smoothing filtering metric is represented by the equation [7]:

$$ck(k) = c(k) * K(N, \alpha) \quad (27)$$

As in motion the resulting curve is in a different scale range and, therefore, needs to be scaled back to percentage values. Thus, scaling is applied next.

- Scaling

Scaling for the shot cut density feature is performed for the same reasons and in the same manner as in the motion intensity arousal metric computation. In this way, $G_2(k)$, the result of the scaling module in the context of shot cut density arousal metric computation, representing the shot cut density arousal, is computed as [7]:

$$G_2(k) = \frac{\max(c(k))}{\max(ck(k))} * ck(k)\% \quad (28)$$

Figure 42 shows $G_2(k)$ for the same football sequence used for Figure 41.

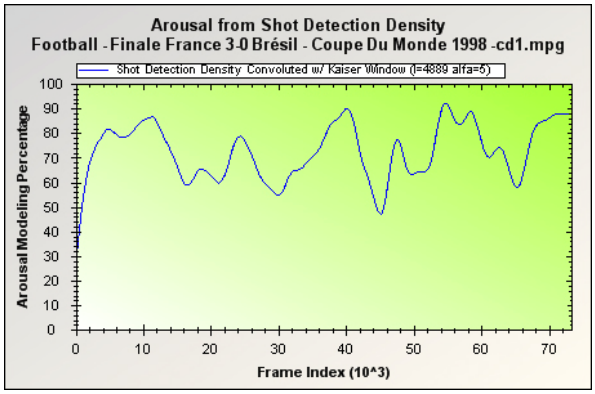

Figure 42 – Example of shot cut density arousal curve, $G_2(k)$, for a football sequence.

4.2.2.3. Sound energy arousal metric computation

The sound energy arousal metric computation process presents some differences when compared with the two previously described arousal computation processes. This happens mainly because sound energy turns out to be dependent on the audio recording level. This fact has to be taken into consideration since otherwise non comparable sound energy arousal curves will result for the same content depending on the audio recording level. This need will be addressed in Scaling sub-module. First, the sound energy metric will be described and next the Smoothing filtering.

- Sound energy metric computation

As for the other low-level features metric definitions, the goal here is to relate the feature information with the degree of excitement felt by the audience. For the sound, this relation may be quite simple; the louder the sound of an audiovisual segment, the higher is the arousal experienced by the viewer. Explosions or gunfire in action films, cheers

of the audience in sport broadcasts, screams in horror films are all segments with high values of sound energy and all these segments are capable of provoking high arousal experiences in the viewer. On the other hand, silent segments are usually associated to calming moments for the viewer.

Therefore, the aim is to use a metric capable of providing higher values for higher sound energy values and lower values for lower sound energy values. To do so, s(k), representing for each frame *k* the sum of the squares of the audio samples is computed [7].

$$s(k) = \sum_{i=1}^{TotalSamples} \left(AudioSample_i(k) \right)^2 \quad (29)$$

The equation in (29) represents the sound energy metric and, although presented here, it is in practice not computed in this sub-module as the XML file created as output of the sound information extraction sub-module already contains this data for the reasons explained in Section 4.1.3.2. In this context, at this stage, the s(k) function is created by simply reading the sound information XML file. After defining s(k), the Smoothing filtering and Scaling have to be applied.

- Smoothing filtering

Smoothing filtering is applied in the same manner, and for the same reasons, as described before, convoluting s(k) with the Kaiser window, with N and α having the same values as before:

$$sk(k) = s(k) * K(N, \alpha) \quad (30)$$

As before, the mathematical convolution produces a curve in a different scale range. In this manner, scaling is necessary and is applied next.

- Scaling

Scaling in sound energy is done by two scaling functions with different purposes. The first has the same function as the scaling functions used in the other low-level features arousal metrics computations, i.e. to scale back the curve resulting from smoothing filtering back to percentage values. As sound energy does not have a maximum value by default the smoothed sound energy curve must be scaled according to its own peak. In this manner, sound energy curves from different contents can be compared. Therefore, the first scaling function is performed by [7]:

$$sk_n(k) = \frac{sk(k)}{\max(sk(k))} \quad (31)$$

If no other scaling function was applied, a sound energy arousal curve with a 100% value on the highest sound energy value would result, which is not desirable because, as for motion intensity and shot cut density, the purpose of the sound energy arousal curve is to model the arousal experienced by the viewer while watching an audiovisual content and not to produce a curve relating the sound energy values to its maximum value. Therefore, a second scaling function is necessary to transform the first scaled function into a curve capable of represent the level or arousal related to sound energy felt by the viewer along the content. The solution found was to compute the mean of $sk_n(k)$, i.e. the mean of the sound energy values related to its peak. This will give more precise information about the arousal related to sound energy experienced by the viewer along the video. A high mean value shows that the sound energy of most of the audiovisual content is close to the peak value and, therefore, the arousal curve should be flatten and scaled down, as the audiovisual content is very constant in terms of sound energy. If the mean value is low, then the sound energy peak is considerably higher than the rest of the content meaning that the variations are more significant and, therefore, the curve should be able to highlight its peaks as they represent important and relevant changes in terms of arousal. The formula below represents the mean of the sound energy values referred to its peak [7]:

$$\overline{sk_n} = \frac{1}{K}\sum_{k=1}^{K} sk_n(k) \quad (32)$$

Finally, to create the final sound energy arousal curve satisfying the requirements above and using the scaling functions described above, $G_3(k)$ [7] is computed as:

$$G_3(k) = 100 * sk_n(k) * (1 - \overline{sk_n})\% \quad (33)$$

Figure 43 represents two sound energy arousal curves, without and with scaling, for a football sequence. The right chart represents the final sound energy arousal curve, $G_3(k)$, with scaling while the left chart represents a sound energy arousal curve without scaling. As the audiovisual content is a football sequence, the sound energy values are pretty constant along the content only with some changes in goals and other significant events. In terms of sound energy arousal, this should result in a more flatten curve with some increase when that events occur. The right chart represents that desired curve while the left chart shows a curve that can only relate the sound energy values with its peak, assuming that the moment with higher sound energy has to be bombastic for the viewer, which is not true.

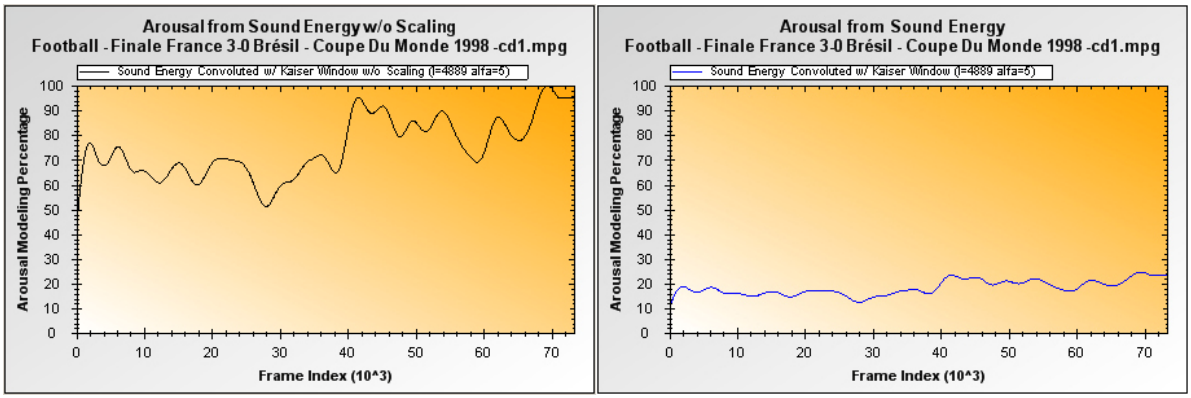

Figure 43 – Example of sound energy arousal curve (a) without and (b) with scaling for a football sequence.

4.2.2 *Fused arousal computation*

The last sub-module before creating the summary description is Fused arousal computation. Fused arousal computation aims to integrate all arousal curves resulting from the low-level features arousal metrics computation processes. As none of the features proved to deserve, with some exceptions for some types of content, having more weight than the others in the fusion process, all three features will be considered, by default, in equal manner, i.e. having 1/3 of the weight in the creation of the final arousal curve. However, the user can change the weights assigned to each feature if he/she wishes to see how the final summary evolves (differently).

4.2.3.1. Fused arousal metric computation

To create a final fused arousal curve, the arousal curves computed before for each low-level feature have to be integrated and combined. This constitutes the main objective of fused arousal metric computation process and, therefore, the main challenge when defining a metric to achieve this goal. As the maximum of each $G_i(k)$ can be located on different frame indexes, a weighted average of the various feature metrics seems to be an appropriate fusion function as it proves to be faithful enough to the variations of each individual $G_i(k)$ function. Than the Fused arousal metric, a(k), is computed as [7]:

$$a(k) = \sum_{i=1}^{3} w_i G_i(k) \quad (34)$$

In (34), w_i refers to the weight assigned to each feature and its sum must be 1. By default, and as explained above, the weight is equal for the three features this means 1/3. If so whishes, the user may change the weights of each feature in order to obtain different final summaries.

4.2.3.2. Smoothing filtering

Although the curves to be included in the fusion process are already smoothed, the Smoothing filtering is again applied after computing the fused arousal metric. The need for the Smoothing filtering, even when each G_i (k) is already smoothed, is related with the merging the neighboring maxima of each G_i (k) function. The Smoothing filtering is applied in the same way as for the individual metrics this means using a mathematical convolution of a(k) with the Kaiser window, with the same values for N and α as before [7]:

$$ak(k) = a(k) * K(N, \alpha) \quad (35)$$

As in all arousal metrics computation processes described before, smoothing filtering changes the scale range of the produced curve and consequently scaling is needed.

4.2.3.3. Scaling

Finally, to obtain the final arousal curve, A(k), for the audiovisual content, scaling has to be applied. Scaling is done for the same reasons as it is performed for the individual features. This means to scale back the arousal values to percentage values and it is also performed in the exact same manner [7]:

$$A(k) = \frac{\max(a(k))}{\max(ak(k))} * ak(k) \quad (36)$$

Figure 44 shows the final arousal curve, A(k), in black, resulting from the Fused arousal computation as well as the three individual arousal curves, one for each feature.

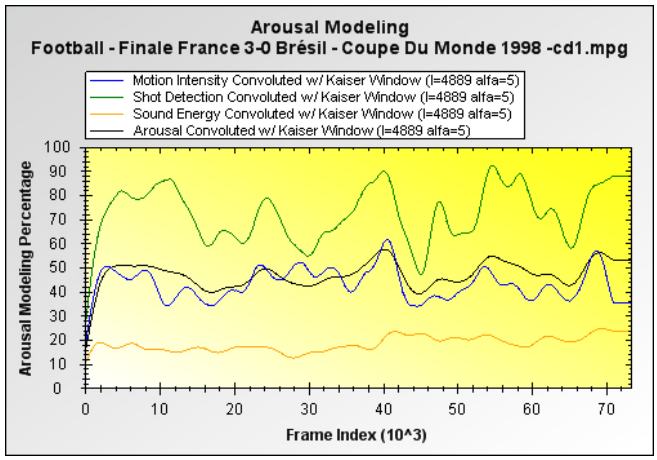

Figure 44 – Example of the final arousal curve – A(k) in black – and the individual arousal curves – $G_1(k)$, $G_2(k)$ and $G_3(k)$.

4.3 Hierarchical summary description creation

The main output of the entire summarization process, derived from the fused arousal curve resulting from previous modules, is a XML file, MPEG-7 compliant, which contains a hierarchical summary description of the audiovisual content. A hierarchical summary description provides a mean to represent the audiovisual content in segments labeled according to their importance. The most important embody the top of the hierarchy and, as one goes down, less important segments will be included in the summary.

The hierarchical summary description has two main motivations that were described in Chapter 3: To allow the user to view and create many different summaries fulfilling different needs, e.g. different types or with different lengths without having to repeat the entire process for summarization, and to create a summarization output capable of allowing interoperability with other systems. To achieve the first requirement, the obvious choice was to create a XML description capable of hierarchically representing the audiovisual content in terms of its summarization relevance. To fulfill the second requirement, the solution was to create the XML file with a standard format; in this case, the MPEG-7 standard was chosen since it defines precisely a description tool with this purpose.

The process to create the MPEG-7 compliant hierarchical summary description is not very complex. First, the choice for describing the summary in a MPEG-7 compliant manner will be justified, and next the labeling of the segments according to its arousal level and the creation of the XML description processes will be explained.

4.2.1 *The choice for the MPEG-7 standard*

MPEG-7 [35][36] is an ISO/IEC standard developed by Moving Pictures Experts Group (MPEG), formally named as "Multimedia Content Description Interface". MPEG-7 is a standard that aims to provide a set of standardized tools to describe multimedia content. As all standards, it intends to perform interoperability between applications and, therefore, the description tools standardized in MPEG-7 aim to support a range of application as wide as possible.

MPEG-7 offers various types of audiovisual *description tools*, which represent the metadata elements and their structure and relationships – named *descriptors* and *description schemes,* respectively. The description tools have the purpose of creating descriptions, i.e. by instantiating some description schemes with their associated descriptors, which can constitute the basis for applications that need efficient access, filtering, retrieval, summarization, etc, of multimedia content.

Two of the many description schemes defined in the MPEG-7 standard regard summarization capabilities: the *SequentialSummary* and *HierarchicalSummary* description schemes [35]. The SequentialSummary description scheme is used to specify summaries of variable length aiming to support sequential navigation. The HierarchicalSummary description scheme is used to specify summaries of variable length but intending to support both sequential and hierarchical navigation. The description schemes are represented in XML format and, therefore, the HierarchicalSummary description scheme emerged as the perfect solution to fulfill the objectives defined for the developed summarization application: allowing the creation of a summary description capable of providing as many different summaries as the user may need and also providing the desired interoperability regarding the output of the application. In this context, the XML file, MPEG-7 compliant, containing the hierarchical summary description created as output of this application can be used by other MPEG-7 players to provide summaries to its own users.

To finalize, the DTD structure of the MPEG-7 HierarchicalSummary description scheme and, consequently, of the output XML file is shown in Figure 45.

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT Mpeg7 ((Description))>
<!ATTLIST Mpeg7
  xmlns CDATA #FIXED "urn:mpeg:mpeg7:schema:2001"
  xmlns:mpeg7 CDATA #FIXED "urn:mpeg:mpeg7:schema:2001"
  xmlns:xsi CDATA #FIXED "http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation CDATA #FIXED "urn:mpeg:mpeg7:schema:2001 Mpeg7-2001.xsd"
>
<!ELEMENT Description ((Summarization))>
<!ATTLIST Description
  xsi:type CDATA #FIXED "SummaryDescriptionType"
>
<!ELEMENT Summarization ((Summary))>
<!ELEMENT Summary ((SourceLocator, SummarySegmentGroup))>
<!ATTLIST Summary
  components CDATA #FIXED "keyVideoClips"
  hierarchy CDATA #FIXED "dependent"
  xsi:type CDATA #FIXED "HierarchicalSummaryType"
>
<!ELEMENT SourceLocator ((MediaUri))>
<!ELEMENT MediaUri (#PCDATA)>
<!ELEMENT SummarySegmentGroup ((Name, SummarySegment+, SummarySegmentGroup?))>
<!ELEMENT Name (#PCDATA)>
<!ELEMENT SummarySegment ((KeyAudioVisualClip))>
<!ATTLIST SummarySegment
  order ID #REQUIRED
>
<!ELEMENT KeyAudioVisualClip ((MediaTime))>
<!ELEMENT MediaTime ((MediaTimePoint, MediaDuration))>
<!ELEMENT MediaTimePoint (#PCDATA)>
<!ELEMENT MediaDuration (#PCDATA)>
<!ATTLIST SummarySegmentGroup
  level (0 | 1 | 2 | 3) #REQUIRED
>
```

Figure 45 – DTD of MPEG-7's HierarchicalSummary description scheme.

From the DTD Structure presented above, the main elements to highlight are the Summary, SummarySegmentGroup and SummarySegment elements, described next:

- ***SummarySegment* element –** There's only one Summary element that represents the summary and has as child element the SummarySegmentGroup representing the top of the hierarchy.
- ***SummarySegmentGroup* element –** The SummarySegmentGroup element represents a level of the hierarchy and has as child elements several SummarySegment elements that are related to the segments from the audiovisual content belonging to that hierarchy level and the SummarySegmentGroup representing the immediate lower hierarchy level. The process is iterative which means the description may include as many SummarySegmentGroup elements as levels defined for the hierarchy.

4.2.2 *Labeling the audiovisual segments*

The first step in creating the output XML, MPEG-7 compliant, hierarchical summary description is to decide which audiovisual segments should belong to which hierarchical level. First of all, four hierarchical levels were created, aiming to represent three different types of summaries in terms of additional content. The bottom of the hierarchy does not represent a type of summary as it would present a summary with the entire audiovisual content. It is important as a user can decide to watch a summary be length and therefore, segments belonging to the bottom of the hierarchy can be included. The choice of four hierarchical levels is deeply related with the choice for three types of summaries. The existence of three types of summaries aims to provide to the user a sufficiently wide range of summaries capable of representing what is important in the original content.

The hierarchical levels are in a top-down approach:

- **Top Highlights, level 0** – Intends to represent the most exciting moments of the audiovisual content, i.e. the segments that will provoke more arousal on the viewer and, consequently, with higher arousal values on the Fused arousal curve. Thus, the segments with the top 10% arousal values are labeled as "Top Highlights". 10% value was chosen as it represents the minimum percentage able to successfully transmit the relevance of the content.

- **Key Points, level 1** – The second level of the description hierarchy aims to provide some context for the "Top Highlights" segments. A "Key Points" summary would be presented to the user including the "Top Highlights" segments as well as the segments labeled as "Key Points", i.e. it would include the first two levels of the description hierarchy. Thus, the frame segments in the top 10-25% arousal values are labeled as "Key Points". 25% value was selected because it is a middle value able to produce a relatively short summary capable of providing interesting segments without being too extensive.
- **Extended Summary, level 2** – This level regards a summary with longer duration with the goal of offering to the user a wider view of the audiovisual content. An "Extended Summary" will present to the user 50% of the total audiovisual content, excluding the dullest and least interesting segments of the audiovisual content. The choice for half of the content is justified by the assumption of being the most extensive summary to fit in the concept of summary itself.
- **Remaining Content, level 3** – The remaining segments are labeled as "Remaining Content" and should contain the less (arousal) relevant parts of the audiovisual content.

The process of labeling the frames is quite simple and relies on a function capable of retrieving the top x% frames with maximum arousal values from the final arousal curve. Therefore, first the 10% of frame indexes with top arousal segments are retrieved and grouped together. Those segments are labeled as **Top Highlights, level 0**. The second step is related to **Key Points** summaries. To retrieve the desired top 25% frame indexes with top arousal values, 15% additional frame indexes are retrieved from the final arousal curve; these indexes are afterwards grouped in segments and labeled as **Key Points, level 1**. The same process is done for **Extended Summary, level 2**. More 25% frame indexes are retrieved, grouped in segments, and labeled as **Extended Summary, level 2**. Finally, the remaining 50% frame indexes are also grouped together, forming the segments labeled as **Remaining Content, level 3**.

4.2.3 *Creating the XML hierarchical summary description*

After labeling all segments, the creation of the summary description XML file is quite straightforward as it only has to follow the DTD structure of the MPEG-7 hierarchical summary description scheme presented in Figure 45. An example of the created XML file, resulting in the output of this system is shown in Figure 46.

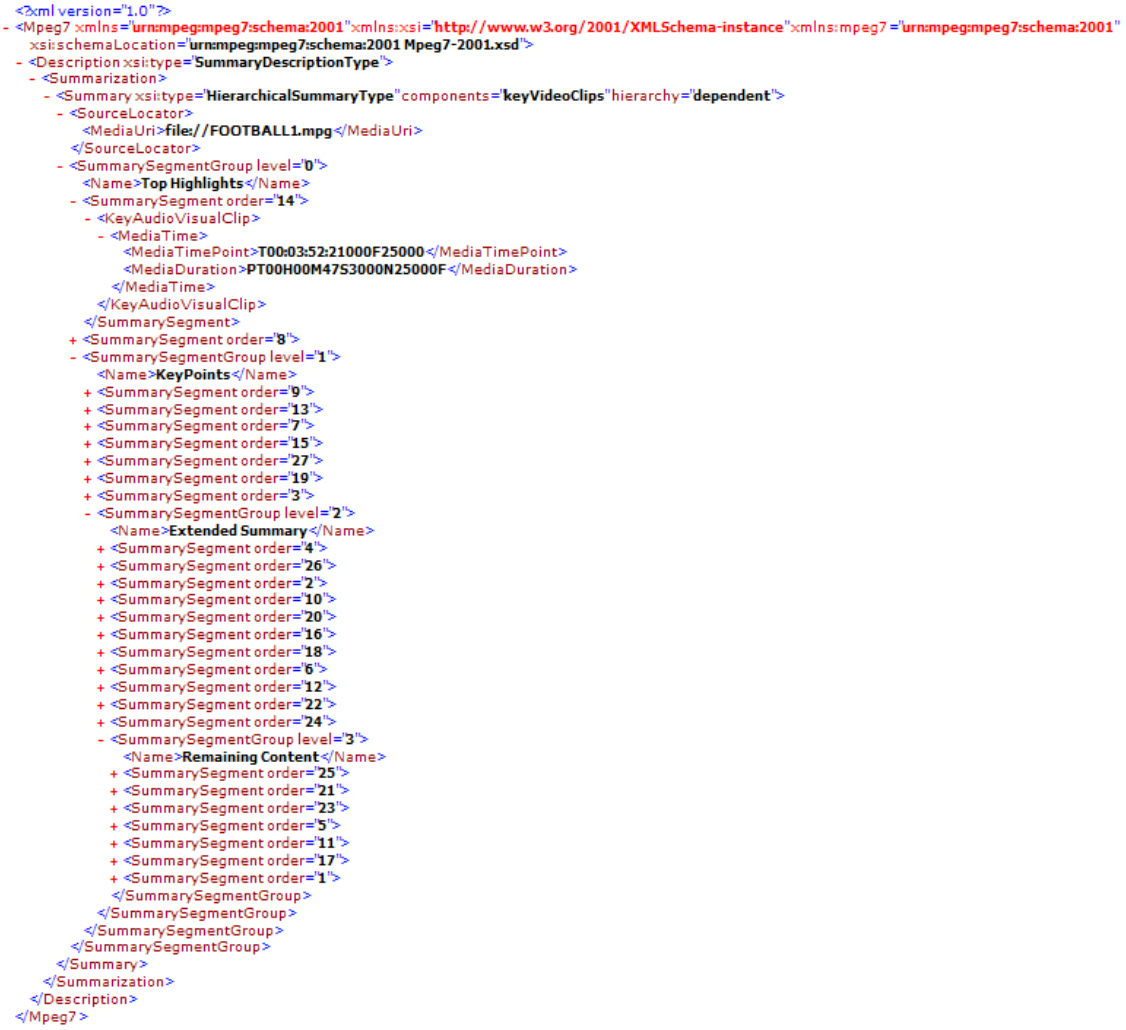

Figure 46 – Example of a MPEG-7 compliant hierarchical summary description for a football sequence.

4.4 MPEG-1 summaries creation

As explained in Chapter 3, the MPEG-1 summaries creation module exists only to give the option to the user of exporting a desired summary, created from the MPEG-7 compliant hierarchical summary description, to a MPEG-1 file. This can be considered an additional feature as it is not crucial to the summarization process. Still, it represent a useful tool to provide to the user as, in this way, he/she can save the summaries produced in MPEG-1 files and use them for their own purposes.

The MPEG-1 summaries creation process was implemented integrating the MPGTX project [37] in the developed system. MPGTX is a command line MPEG audio/video/system toolbox with the ability to slice and join MPEG-1 files. This is precisely what is needed to create MPEG-1 summaries as, from the hierarchical summary description and off the user's choice of parameters, a summary is presented, formed by a group of segments from the audiovisual content.

In this manner, to create the MPEG-1 files with the desired summary, it is provided to the MPGTX application, as its arguments, the segments constituting the summary. The application will create, by slicing from the audiovisual content the segments and in the end, by joining them, a MPEG-1 file representing the desired summary.

From one MPEG-7 compliant hierarchical summary description can be created an infinite number of MPEG-1 files representing summaries as the user can choose to create summaries by length and from the three types available.

This chapter intended to offer the reader an in-depth description of the entire process for summarization, presenting all modules and algorithms developed to achieve the system's purposes. The chapter should be able to provide an understanding of the input audiovisual content evolution until the MPEG-7 hierarchical summary description and if, it is the user's wish, the MPEG-1 summary files are created.

Chapter 5 will describe the application in detail, justifying the choice of the programming language, presenting the application's high level architecture and the frameworks and libraries used and finally describing the application's Graphic User Interface (GUI).

Chapter 5

Describing the Summarization Application

Chapter 5 intends to provide the reader with a description of the developed summarization application. First, the choice of the programming language as well as other options, e.g. regarding frameworks and libraries used and the application's structure will be motivated. Next, a detailed description of the application will be given by means of an installation guide and a user's guide. Finally, the user's guide will provide a complete description of the application's GUI.

5.1 Implementation overview

Before describing the application, this section aims to motivate the choice of C# as the programming language and to briefly described the frameworks and libraries used in the development of the summarization application.

5.1.1 Programming language selection

As the developed application was intended to be a Microsoft Windows application, the use of Microsoft Visual Studio 2005 [38][39] was consensual. Microsoft Visual Studio is a software development product for computer programmers, providing a development environment allowing programmers to create applications, web sites or web applications to run on platforms supported by Microsoft's .NET framework [40][41]. Microsoft's .NET framework provides a large number of pre-coded solutions to common program requirements, managing the execution of applications written specifically for this framework.

After choosing the Internal Development Environment (IDE), the following decision was related to the programming language. Using Microsoft's Visual Studio 2005, two languages arose as the main candidates: C++ and C#. Although many multimedia applications have been developed in both languages in recent years, the choice was C#. The main motivation for this choice was the developer's experience on this programming language in contrast with an

absolute lack of experience on C++. As Visual Studio allows the integration of different types of programming languages, the problem regarding the reduced number of available multimedia C# libraries was minimized, as multimedia C++ libraries could and were integrated in the development of this solution.

5.1.2 Frameworks and libraries

Almost all frameworks and libraries used in the development of this system were introduced and explained in Chapter 4, in their respective sections. Even so, the frameworks and libraries included in the solution, without any adaptation, will be briefly described next:

1. **DirectShow –** DirectShow [31] (already described in section 4.1.2) is a multimedia framework and API developed by Microsoft for software developers to perform various operations with media files. In this system, DirectShow allowed to playback and to perform all kinds of common playback operations to MPEG files, such as "Stop", "Pause", "Play", "Step One Frame", "Increase/Decrease Rate", and "Mute/Unmute". In addition, it was of great use in two of the three low-level features extraction processes, namely in Shot cut detection and in Sound information extraction, as explained in the corresponding sections in Chapter 4.
2. **ZedGraph –** ZedGraph library [42] was fundamental in the development of this work as it allowed to construct all charts included in the solutions. ZedGraph library is constituted by a set of C# classes allowing the creating of 2D line and bar charts, being highly flexible to the developer with almost every aspect of the chart being able to be modified. All time curves created during the summarization process are built using the ZedGraph library. The charts exemplifying time curves included in this Thesis were also built with the ZedGraph library.

The remaining development references used in this system, namely Ascenso and Hidalgo's [25], Khan's [31] and John's [32] works, notably at the low-level features extraction level, were adapted to fit this application and, therefore, are only referred in this section as they were already presented in sections 4.1.1 and 4.1.3.

5.1.3 Application's structure

This section aims to provide to the reader, in a very high level manner, the structure of the implementation adopted for the summarization application. To do so, it is shown in Figure 47 a high-level class diagram of the application.

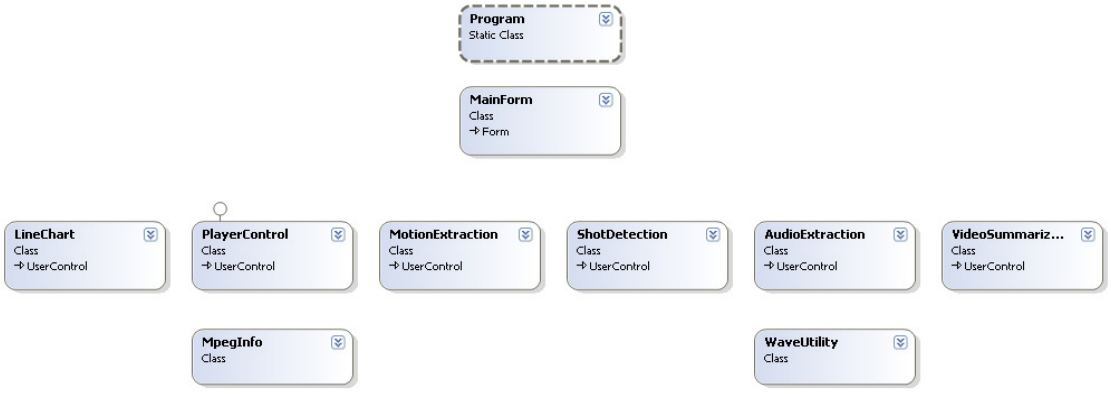

Figure 47 – Application's high-level class diagram.

The class diagram in Figure 47 shows the class diagram hierarchy, notably the MainForm class that constitutes the Windows Form presented to the user. Several UserControl classes were also implemented, corresponding to the second level of the hierarchy.

UserControl classes have its own User Interface and exist to divide and structure the implementation architecture according to its function; they also have the benefit of being able to be reused. There are six UserControl classes:

1. **LineChart –** Implements the charts used in the application.
2. **PlayerControl –** Implements the audiovisual content player.
3. **MotionExtraction, ShotDetection and AudioExtraction –** Implement the low-level information extraction processes.
4. **VideoSummarization –** Implements the Fused arousal metric computation.

The six UserControl classes implement the User Interface for each of these processes as well as the corresponding low-level source code. Each UserControl class uses auxiliary Classes in its process, as exemplified in Figure 47 by MpegInfo and WaveUtility classes used in the PlayerControl and AudioExtraction UserControl classes.

5.2 Installation guide

The application's installation process is straightforward and quite simple. The system has two requirements:

1. **Microsoft's .NET Framework 2.0 –** Available for download at [43]. Its installation is straightforward.
2. **WAV Dest DirectShow filter –** WAV Dest DirectShow filter has to be previously registered in the running machine for the application to work properly as it is used for the sound information extraction. It is not installed together with the Windows Operative Systems and, therefore, is not usually registered. WAV Dest DirectShow filter is available in the application package and the process for its registering is explained next.

Resuming, after installing Microsoft's .NET Framework 2.0, the installation of the application is done in three steps:

1. **Decompress the application's package –** The first step to install the application developed in this Thesis is to decompress the content of the application's package previously provided to a folder of choice. This can be done with any common archive manager software such as Winzip [44] or Winrar [45].
2. **WAV Dest DirectShow filter register –** The second step regards the registration of the WAV Dest DirectShow filter which is also rather simple and can be done following these steps:
    1. After decompressing the application's package, go to the PrototypeAffSum\PrototypeAffSum\Library folder located in the folder of choice (created when decompressing the application package).
    2. Then copy the WAVDEST.ax and msvcr71d.dll files located on the Library folder to the WINDOWS/System32 folder.
    3. Next, open the start menu and click on the *Run* command; type *cmd* and click *OK*.
    4. Now, the command line should be open. Go to the WINDOWS\System32 folder by typing: *cd C:\WINDOWS\system32*. Finally, in C:\WINDOWS\system32, call the dllregistry server by typing: regsvr32 WAVDEST.AX as Figure 48 illustrates. At this stage, the 'RegSvr32 DllRegisterServer success' message shown in Figure 48 will appear.

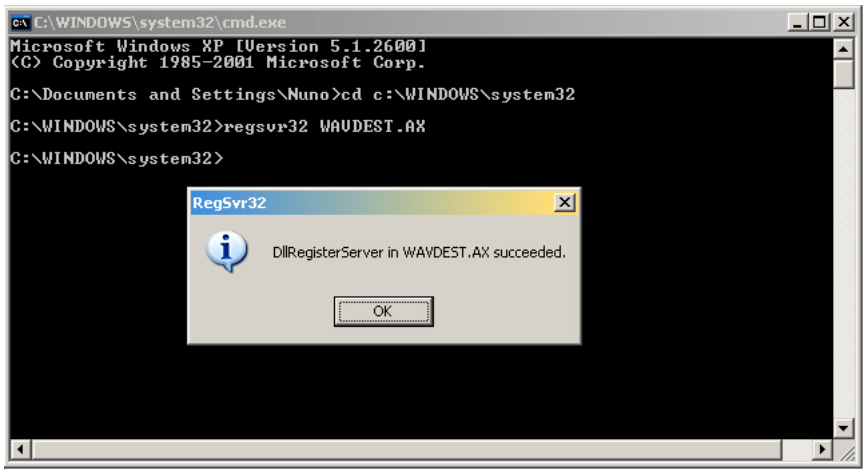

Figure 48 – Registering WAVDEST.AX – WAV Dest DirectShow filter.

3. **Running the application** – The application should now be fully installed. To run the application, go to PrototypeAffSum\PrototypeAffSum\bin\release and run PrototypeAffSum.exe.

5.3 GUI description

This section aims to provide a detailed description of the application's GUI.

The application in composed by a single Windows Form, which is divided in 5 main areas, as shown in Figure 49. Figure 49 represents the state of the application after extracting motion information. The 5 areas will be described next. In the charts and tab colors, **light blue** is related to **motion information**, **green** to **shot cut information** and **orange** to **sound information**. The 5 areas highlighted in Figure 49 have the following main functions:

1. **Player –** Intended to play the audiovisual content.
2. **Charts/Summary player tab control –** Area destined to present the arousal charts from each feature to the user and also to play the final summary.
3. **Main tab control –** Area meant for presenting information resulting from the low-level features extraction to the user and also to parameterize the fused arousal computation process as well as the summary creation and viewing processes.
4. **Side menu –** Has three buttons which serve as start for the respective low-level features extraction processes.
5. **Side menu options tab –** Area reserved for the option related to each low-level features extraction process.

The top menu strip and top tool strip shown on Figure 49 provide the basic functionalities of the application. They are briefly described next, starting from top menu strip:

- **File→Open Video** – Allows the user to open a video to be summarized. It constitutes the first step needed for the summarization process. Enables the Side Menu.
- **File→Exit** – Allows the user to exit the application.
- **Help→Help –** Shows the user the help menu.
- **Help→About –** Shows the user a window containing information about the application and the authors.

The top tool strip only provides two buttons:

- **Open Video** – Does the same as top menu strip's File->Open Video
- **Help –** Does the same as top menu strip's Help->Help.

Figure 49 is finally presented next.

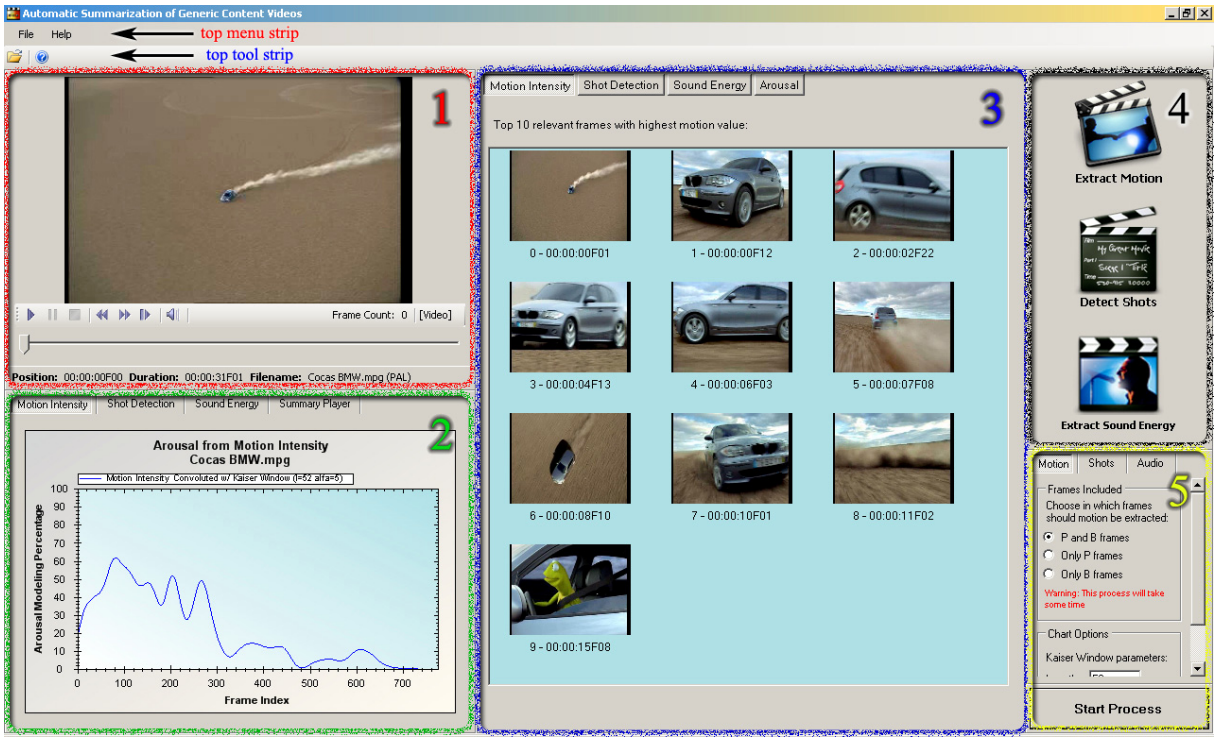

Figure 49 – Application's GUI.

5.3.1 Player

The player area is naturally destined to play the MPEG-1 files that will be submitted to the summarization process. Figure 50 shows the player in more detail. Each of player's tools will be described next. To open a file click on **File→Open Video** in the top menu strip or click on the **Open Video** button in the top tool strip both located above the player.

Figure 50 – Player controls.

1. **Panel** – The video panel is where the audiovisual content is presented.
2. **Play button** – The play button plays the audiovisual content.
3. **Pause button** – The pause button pauses the audiovisual content.
4. **Stop button** – The stop button stops the audiovisual content, and brings the player time position back to instant zero.
5. **Decrease rate button** – The decrease rate button decreases the player frame rate of the audiovisual content by ¼.
6. **Increase rate button** – The increase rate button increases the player frame rate of the audiovisual content by ¼.

7. **Step one frame button** – The step one frame button puts the audiovisual content's position one frame ahead.
8. **Mute/Unmute button** – The mute/unmute button toggles between muted and unmuted audiovisual content.
9. **Frame counter** – The frame counter indicates the current frame number.
10. **Track bar** – The track bar indicates the relation of the audiovisual content's current position regarding the total duration. The user can scroll the track bar to put the audiovisual content in the desired position.
11. **Position label** – The position label indicates the current position of the audiovisual content in time scale.
12. **Duration label** – The duration label indicates the total duration of the audiovisual content.
13. **Filename label** – The filename label shows the name of the file and its format – Phase Alternating Line (PAL) or Nation Television System Committee (NTSC) – inside parenthesis.

5.3.2 Charts/Summary player tab control

The Charts/Summary player tab control is formed by four tabs, highlighted in Figure 51; the first three relate to the arousal charts for each low-level feature, while the fourth is composed by a Player, similar to the one described above, for playing any created summary.

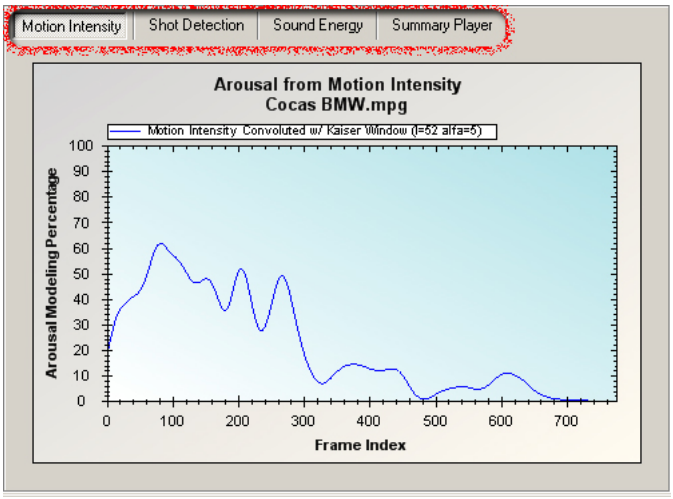

Figure 51 – Charts/Summary player tab control.

- Low-level features arousal charts tabs

The first three tabs of the Charts/Summary player tab control present to the user the arousal charts resulting from the arousal metrics computation for each feature. After extracting the low-level information for each feature, the tab control automatically switches to the corresponding tab, showing to the user the arousal curve resulting from the arousal metric computation for that feature. Figure 52 shows an example of each of the three tabs.

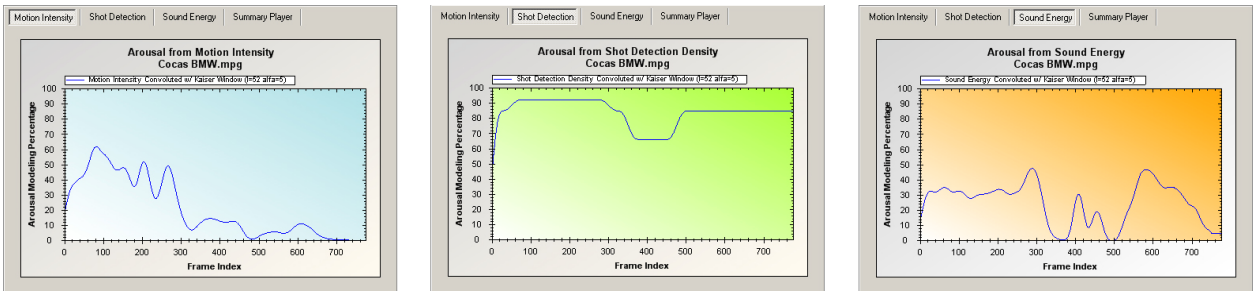

Figure 52 – Low-level features arousal charts tabs.

The user is presented with a set of chart options for every chart presented by the application. By right-clicking on the chart, the option menu as illustrated in Figure 53 will appear. The options - Copy, Save image as, Page setup, Print, Show point values, Un-zoom, Undo all zoom/pan and Set scale to default - are self-explanatory.

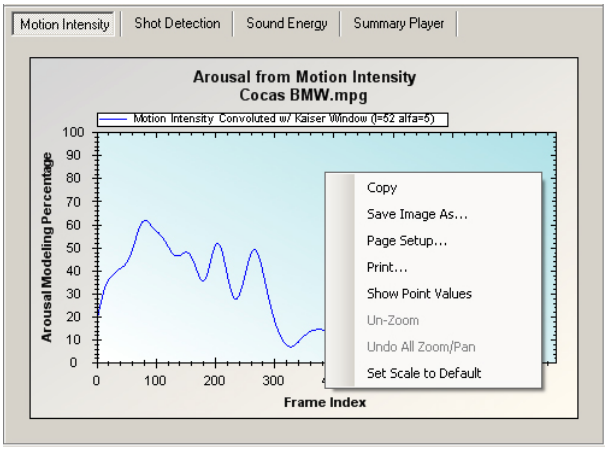

Figure 53 – Chart related options.

- Summary player tab

The summary player tab is equal to the player area presented in Section 5.3.3.1. Summary player tab will present the user the summary created after the user clicks the **Play Summary** button on Main tab control's arousal tab.

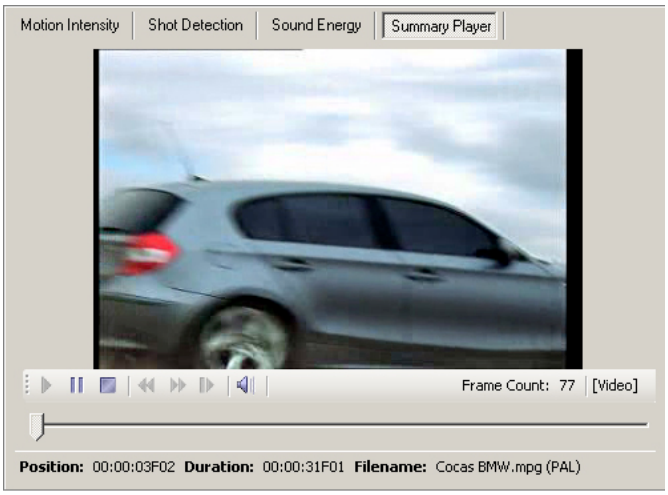

Figure 54 – Summary player tab.

5.3.3 Main tab control

The Main tab control, as the Charts/Summary player tab control has four tabs, highlighted in Figure 55. The first three relate to each feature and its low-level information extraction process while the last tab relates to the final arousal computation.

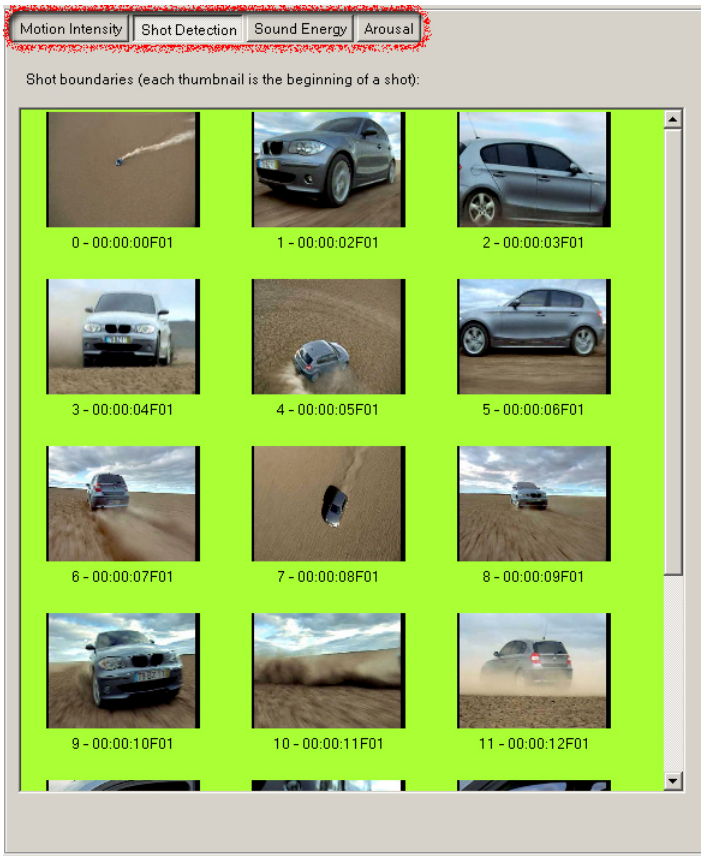

Figure 55 – Main tab control.

- Low-level features related tabs

The low-level features related tabs are switched and filled after each low level extraction processes. The motion and sound information main tabs are similar, representing only the thumbnails corresponding to the 10 frames with highest motion/sound energy values. The motion and sound main tabs are represented in Figure 56.

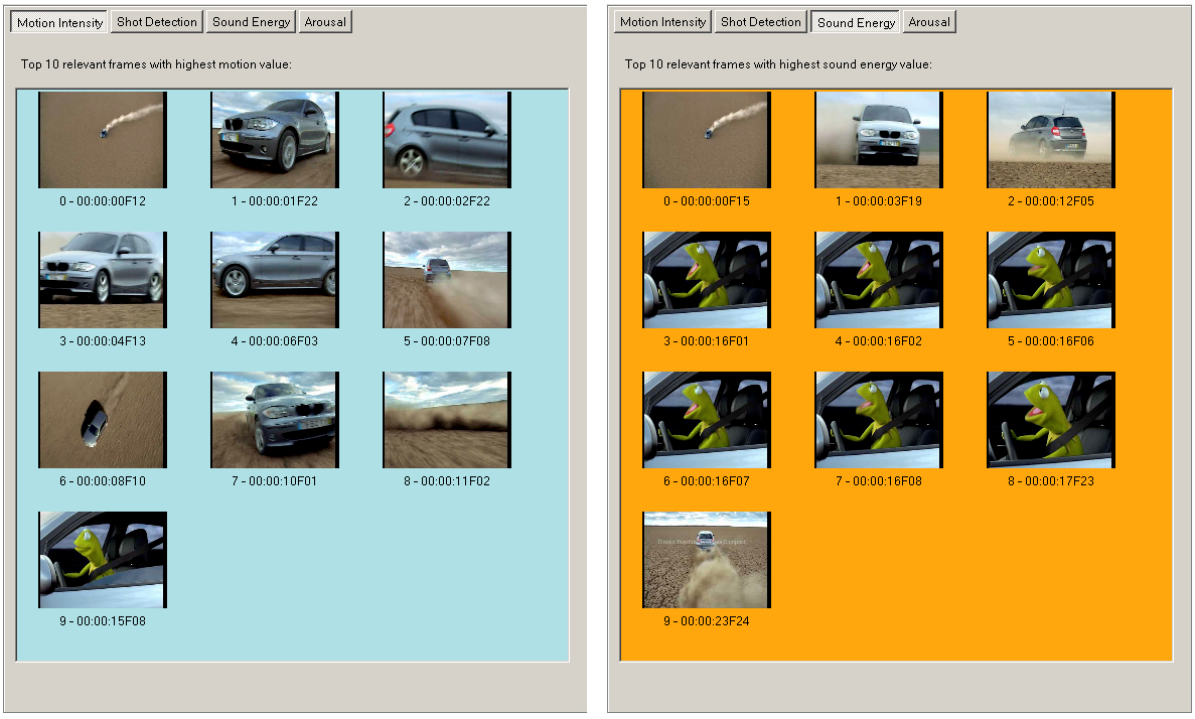

Figure 56 – Motion and sound information main tabs.

The shot detection tab is a bit different as all shot boundaries are represented in the tab and the user can **play the shot** correspondent to a frame boundary if he/she double-clicks on the respective frame thumbnail. Figure 57 shows the shot detection main tab.

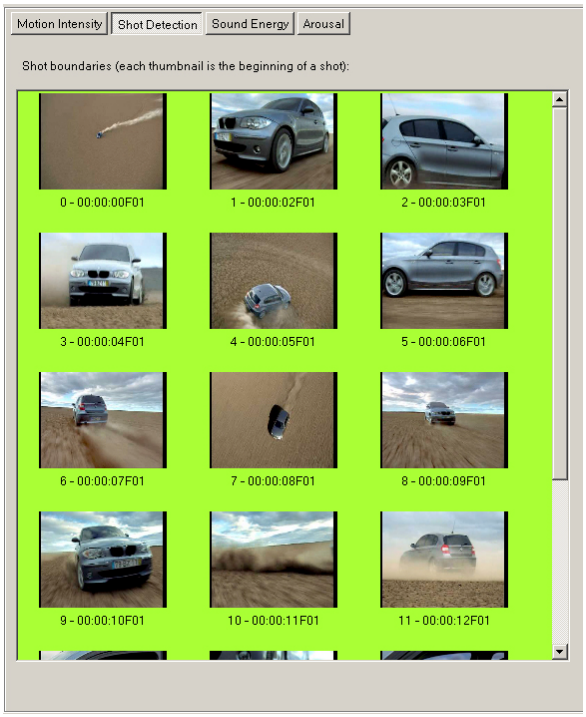

Figure 57 – Shot detection main tab.

- Arousal tab

The final Arousal tab in the Main tab control is where all options corresponding to the fused arousal metric computation and the play of the summary are presented to the user. Figure 58 shows the Arousal tab; the tab's show in Figure 58 will be explained next. The Arousal tab is only enabled after at least one low-level feature has been extracted. At first only the Arousal chart and the Save summary options panel are enabled.

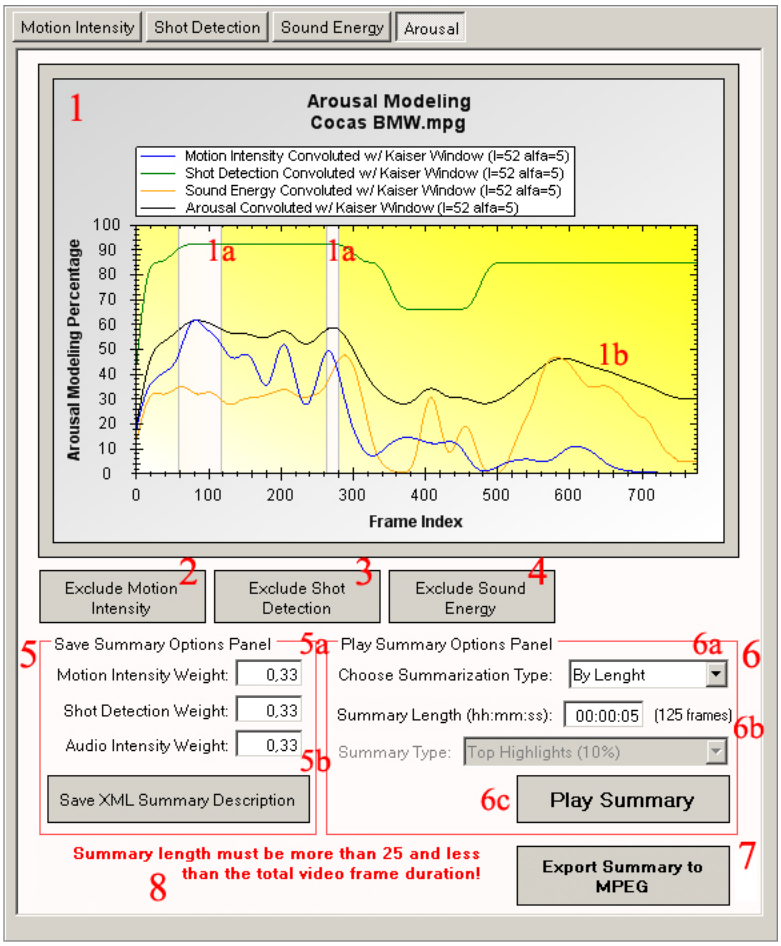

Figure 58 – Arousal tab.

1. **Arousal chart** – The arousal chart presents to the user the three arousal curves, with the color corresponding to the feature, resulting from the arousal metric computation and a black curve corresponding to fused arousal metric.
    a. **Summary segments bars** – The highlighted bars shown in the chart are the segments that can be shown in the Summary player tab, in the Charts/Summary Player Tab Control. The bars appear after the user clicks the 6c, Play summary button.
    b. **Arousal curves** – Each of the curves is represented by the respective low-level feature color. The blue curve is related to motion intensity, the green to shot detection and the orange to sound energy. The black curve is the final arousal curve.
2. **Exclude/Include motion intensity button** – This button excludes/includes the motion intensity feature from the fused arousal metric computation and updates the arousal chart.
3. **Exclude/Include shot detection button** – This button excludes/includes the shot information feature from the fused arousal metric computation and updates the arousal chart.
4. **Exclude/Include sound energy button** – This button excludes/includes the sound energy feature from the fused arousal metric computation and updates the arousal chart.
5. **Save summary options panel**
    a. **Feature's weights combo-boxes** – These boxes allow the user to assign different weights to each feature for the fused arousal computation, and thus for the creation of the XML summary description.

b. **Save XML summary description button** – This button will ask the user the name of the XML summary description he/she wants to save and creates the XML summary description, with the desired name, with the weights defined in 5a and updating the arousal chart, with the arousal curve representing the weights changes. Enables the Play summary options panel.

6. **Play summary options panel**
   a. **Type of summary combo-box** – This box allows the user to choose from two different types of summaries: **By Length** and **By Type**.
   b. **Type of summary options** – These two combo-boxes are enabled depending on which type of summary is chosen. If the type of summary is **By Length**, the user can insert the desired length; if it is **By Type**, the user can choose from three different summary types, corresponding to the levels on the XML summary description, i.e. **Top Highlights**, **Key Points** and **Extended Summary**. Remaining Content level is ruled out as it corresponds to playing the entire content.
   c. **Play summary button** – Will ask the user to select the XML summary description he/she wants to use and will draw the summary segment bars in the arousal chart and plays the summary in the Summary player tab, in the Charts/Summary player tab control according to the user's choice of 6a and 6b. Enables the Export summary to MPEG button.
7. **Export summary to MPEG** – Exports the summary currently being played in Summary player tab, in the Charts/Summary player tab control to an MPEG-1 file, giving the user the choice to choose its name.
8. **Warning label** – The warning label appears when some information has to be given to the user.

5.3.4 Side menu

The Side menu has 3 buttons: **Extract Motion, Detect Shots and Extract Sound Energy**. Each button will enable the corresponding Side menu options tab. Figure 59 shows the side menu.

Figure 59 – Side menu.

5.3.5 Side menu options tab control

The Side menu options tab control area has 3 tabs, one for each low-level feature. Each tab is enabled by the Side menu buttons. Figure 60 shows the three tabs present in the Side menu options tab control.

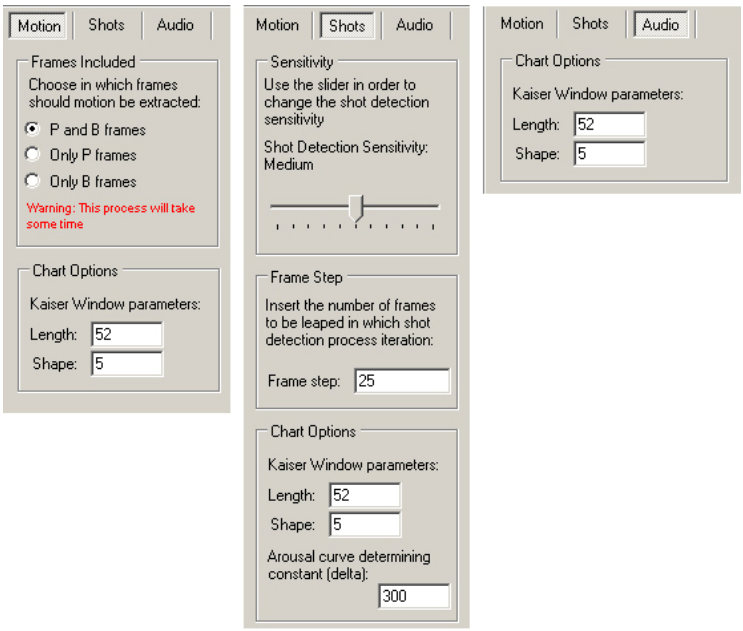

Figure 60 – Side menu options tabs.

Each tab allows the user to control the inputs for each low-level information extraction processes; they are coherent with the module architectures presented in Chapter 4. The motion tab allows the user to choose which type of frames he/she wishes to include in the summary; the shots tab gives the user the chance of deciding the frame step and the sensitivity to be used in the shot detection process; finally, the audio information extraction process has no inputs besides the content itself. Besides the inputs of the low-level information extraction processes, the user can in each tab, parameterize the chart options, with the Kaiser window's length and shape and, in Shot detection, also with the constant δ presented in Section 4.2.2.2.

The **Start Process** button starts the corresponding low-level information extraction process.

Chapter 5 introduced the application, presenting an overview about its implementation with a brief description of the frameworks and libraries used as well as its high-level architecture.

In order to a proper use of the application, an installation guide was also provided, in addition to a detailed description of the GUI aiming to explain all the options given to the user in the process for summarization.

Chapter 6 will present the results obtained when evaluating the system developed in this Thesis.

Chapter 6

Performance Evaluation

Since the development of any multimedia application is not finished without a serious and meaningful evaluation of its performance regarding the user objectives, this chapter aims to present and analyze the results obtained by a subjective evaluation study which was carried out to evaluate the developed application's performance.

Considering the type of system developed, it is believed that the adequate performance evaluation methodology should follow a subjective and not objective approach. The reviewing of the relevant literature confirmed that there are no objective evaluation methodologies available for the problem at hand. Thus, it was decided to design an adequate evaluation methodology and conduct a user evaluation study to assess how the developed application performs in view of the initially defined objectives. This chapter will present the test objectives, its methodology, the subjective scores, and an analysis of the obtained results.

6.1 Test objectives

This test intended to evaluate the overall summarization performance of the developed application and, implicitly, of the implemented algorithms. With this purpose in mind, a subjective evaluation study was designed with two main objectives:

- To evaluate how good is the summarization user experience provided by the created summaries, according to each summary type, i.e. if a "Top Highlights" summary captures only the indispensable segments of the content; if a "Key Points" summary is able to additionally provide some context to a "Top Highlights" summary, capturing only interesting segments without being too extensive; and, if an "Extended Summary" is able to exclude the most boring and least interesting segments of the content.

- To evaluate if any important segments are excluded from any of the created summaries for the various types; of course, the "Top Highlights" summary is typically the most critical since it is the one that has to be more selective due to its shorter duration.

The next section presents the test methodology designed to achieve these two objectives.

6.2 Test methodology

In order the test to be performed is credible, it has to be well defined enough to be reproducible by other experts in order the results and conclusions are statistically similar.

1. **Test Questions** – The test questions defined for this test to address the objectives above are:
   - **Question 1** – The summary viewed satisfies its type definition, i.e. contains the top most relevant/exciting 10%, 25% and 50% of the original content?

     a) Not at all b) Badly c) Reasonably d) Mostly e) Totally
   - **Question 2** – Any relevant segments were ruled out of the viewed summary for each summary type, i.e. Top Highlights, Key Points and Extended Summary?

     a) All b) Many c) Some d) Few e) None

2. **Sequence of Testing –** A group of 13 volunteers were asked to view the original audiovisual content and to give their subjective assessment to the questions above, for each of the summaries created, following the sequence of steps defined next:
   - **a.** Open and visualize the original content, starting from "Basketball" content.
   - **b.** For the original content at hand, visualize 1 only time its three possible summaries, i.e. Top Highlights, Key Points and Extended Summary.
   - **c.** Answer to questions 1 and 2, marking with a cross (X), in the evaluation tables, the desired classification mark, for each one of the three summaries just visualized, i.e. Top Highlights, Key Points e Extended Summary
   - **d.** Go back to point 1. for all lasting contents until tables 1 and 2 are completely filled.

3. **Test Material** – The test set was constituted by 6 audiovisual pieces. Based on these 6 pieces, 18 summaries were produced and exported for MPEG-1 files, using the developed summarization application. For each audiovisual piece, three summaries were produced: i) "Top Highlights" summary; ii) "Key Points" summary; and iii) "Extended Summary". The set of 6 pieces was divided into two classes, with 3 pieces per class:
   - **Sports content –** Pieces containing sports broadcasts, with two clips from football matches and one clip from a basketball match.
   - **Entertainment content –** Pieces containing clips from TV series containing action events: one from "Lost", another from "Prison Break" and, finally, another from "Heroes".

As it is intended the user to see the original content, as well as three types of summary created for each content, the original contents were clipped in order to reduce the test's duration in a way that the results can be considered meaningful but without exhausting the test subject. The contents duration as well as their resolution and format are presented next in Table 12.

	Content duration	Content resolution	NTSC/PAL
Sport content			
BASKETBALL.mpg	10:02	320×240	NTSC
FOOTBALL1.mpg	13:27	418×288	PAL
FOOTBALL2.mpg	10:10	418×288	PAL
Entertainment content			
ACTION1.mpg	14:14	624×352	PAL
ACTION2.mpg	12:48	624×352	PAL
ACTION3.mpg	14:38	624×352	PAL

Table 12 – Test material characteristics.

4. **Application control** – The summaries for each of the contents were produced using the following parameters for the low-level information extraction algorithms:

	Motion information extraction	Shot cut information	Sound information extraction
Sports content			
BASKETBALL.mpg	P and B frames extracted	Frame step: 10 Sensitivity: 0.5	No input parameters
FOOTBALL1.mpg	P and B frames extracted	Frame step: 10 Sensitivity: 0.5	No input parameters
FOOTBALL2.mpg	P and B frames extracted	Frame step: 10 Sensitivity: 0.6	No input parameters
Entertainment content			
ACTION1.mpg	P and B frames extracted	Frame step: 10 Sensitivity: 0.6	No input parameters
ACTION2.mpg	P and B frames extracted	Frame step: 10 Sensitivity: 0.7	No input parameters
ACTION3.mpg	P and B frames extracted	Frame step: 10 Sensitivity: 0.5	No input parameters

Table 13 – Algorithms parameters.

The shot cut detection process is the only one with different parameters for all contents. The choice for a frame step of 10 instead of 1 is related to the detection of transitions. In this way, the shot cut detection process is able to detect some fade transitions instead of only detecting hard cuts. Soft transitions, mainly in TV series, occur very frequently and, therefore, its detection is rather relevant. The sensitivity of the process has to be adjusted according to the type of content as different content types may have been recorded with different light conditions; this is an important aspect to consider when setting the sensitivity of the shot cut detection process. For motion processing, all the motion vector values were always extracted from all P and B frames. As explained before, sound energy has no input control parameters.

The Kaiser window parameters for all contents, and for all curves, are equal, with a length corresponding to the duration of the video divided per 15 and a shape of 5 as these were the values that proved, after intensive testing, to be more suitable for the desired purposes, i.e. to smooth the curves obtained from the low-level features metrics.

Regarding the weights used for the fused arousal metric computation, the values of **0,5 for sound energy**, **0,35 for motion activity** and **0,15 for shot cut density** were used for sports content. Shot cut density proved to have less relevance in this kind of content, mainly because the segments with higher shot cut density occur **after** the relevant event occurs, with slow-motion replays and different shots from the audience or the event intervenient.

For the entertainment contents, the weights used were the same for all three features, i.e. 0,33 for sound energy, 0,33 for motion activity, and 0,33 for shot cut density, as none of the features proved to deserve more importance than the others.

The next section presents the obtained results and their analysis.

6.3 Results and analysis

This section will present the obtained results in this user evaluation study and their in-depth analysis. First, an informal analysis of the results will be given, for each content, with the help of the final arousal charts; next, a more precise analysis of the results obtained for each question will be made, after the presentation of the results.

- **Informal analysis of summarization results**

To start an informal analysis of the obtained summarization results, the final arousal charts, for each content, are presented, divided into sport and entertainment content. Figure 61 shows the final arousal charts for the sports content; in the charts, the black curve corresponds to the fused arousal determining the final summary.

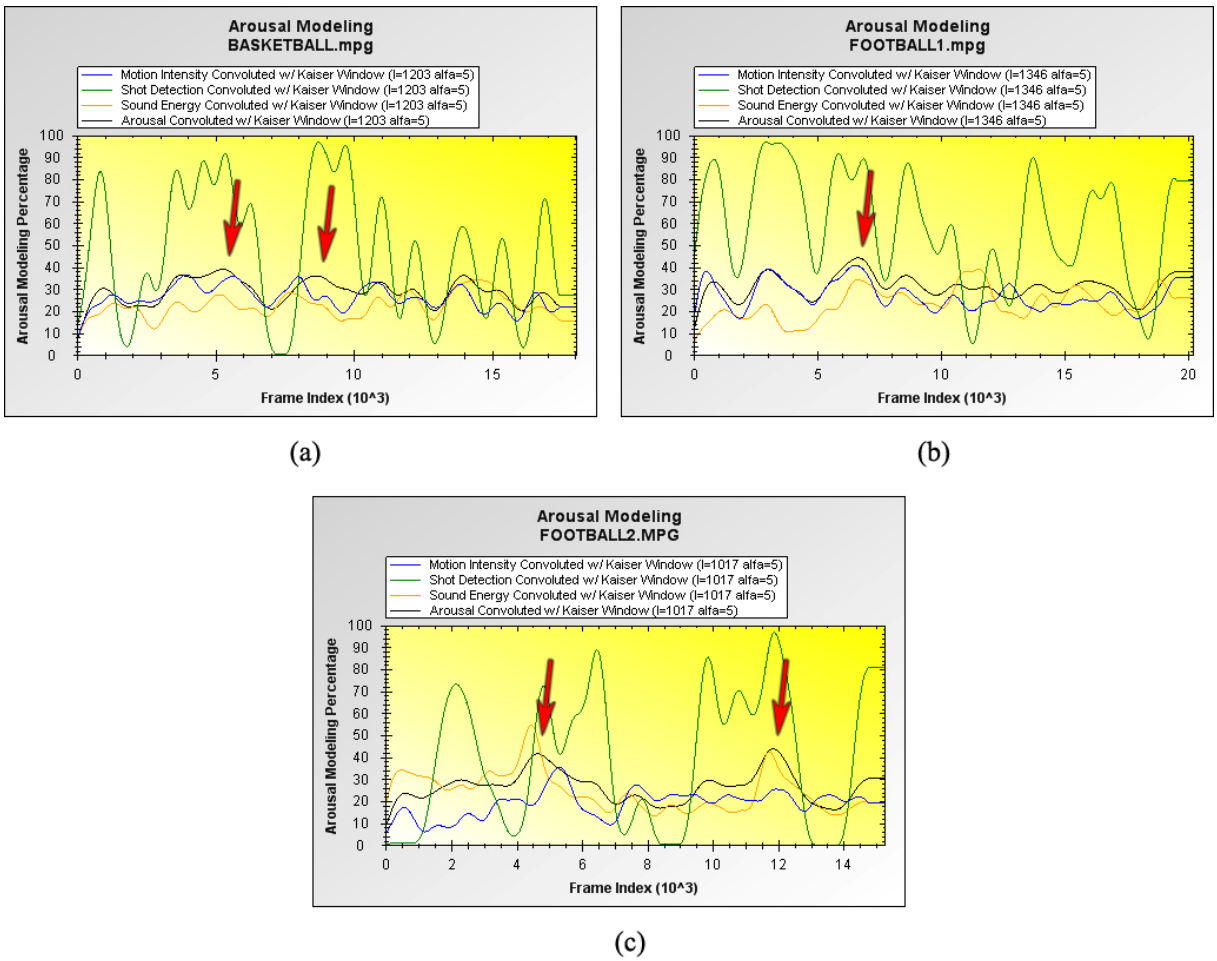

Figure 61 – Sports content final arousal charts: (a) BASKETBALL.mpg*; (b) FOOTBALL1.mpg; and (c) FOOTBALL2.mpg.*

1. **BASKETBALL –** Starting from the Basketball piece, it can be concluded that its arousal curve is quite constant with some higher values around frames 5000 and 8500 which correspond to segments with faster plays in the match, with attacks and counter attacks from both teams. Those segments constitute the most exciting events in the content. From frame 14500 ahead, a timeout event takes place with shots of each

team's technical staff talking with their players, forming a less interesting event. It was important to verify that the "TopHighlights" summary, which is the shorter and aims to capture the top most 10% relevant/exciting events, did, in fact, capture the plays with higher rhythm occurring around frames 5000 and 8500. The "KeyPoints" summary increased the duration of the segments including those frames, and providing them with some additional context. This summary also includes a segment around frame 14000, representing a doubtful decision by the referee and the following dissatisfaction by one of the players. Regarding the "ExtendedSummary" summary, the most relevant aspect is the fact that it excluded the timeout event, probably the dullest event in the whole piece of content.

2. **FOOTBALL1 –** The first Football content, "Football1" had one goal event, constituting the most important relevant/exciting moment of the content, occurring around frame 6000. This event is included in all summaries, namely in the "Top Highlights" summary, which is the shortest. Other two plays were included in the "TopHighlights" summary, with fast pace and of some interest, near frames 4000 and 19500 near the end; unfortunately, a corner kick around frame 1000, probably more exciting, was left out. In fact, only the "ExtendedSummary" includes the corner kick as the "KeyPoints" summary oddly also ruled out that event, choosing to include instead, the same moments as the "TopHighlights" summary but with more context and also more one or two relatively exciting plays. This exclusion was provoked by sound energy, being certainly related with the corner kick being in favor of the away team, not originating a significant increase of the audience's noise.
3. **FOOTBALL2 –** The second Football piece of content, "Football2", includes two goal events which constitute the "TopHighlights" summary. They occur around frames 5000 and 12000 and correspond to the highest peaks of the arousal curve. As before, the "KeyPoints" summary provides context to the "TopHighlights" summary, including the same interesting moments but wrapped in longer segments; it also includes a rough tackle near the end of the content. To register, the fact that in all the summaries, even in the "Extended Summary" summary, a free kick event was left out near frame 6500. Although it was not a dangerous free kick and therefore, an event of great importance, it could have been included, mainly, in the "Extended Summary". It was not mainly because of motion intensity as the free kick sequence was almost entirely filmed from a long shot of the field. Before the free kick all players were standing waiting for the free kick to be taken and after the free kick there was no increase in the motion intensity as the free kick had no important consequences. Therefore, the entire segment was low on motion intensity, bringing the arousal curve down.

An informal analysis of the entertainment content is presented in the following. Figure 62 shows the final arousal curves for each piece of entertainment content; as before, the black curve represents the final arousal.

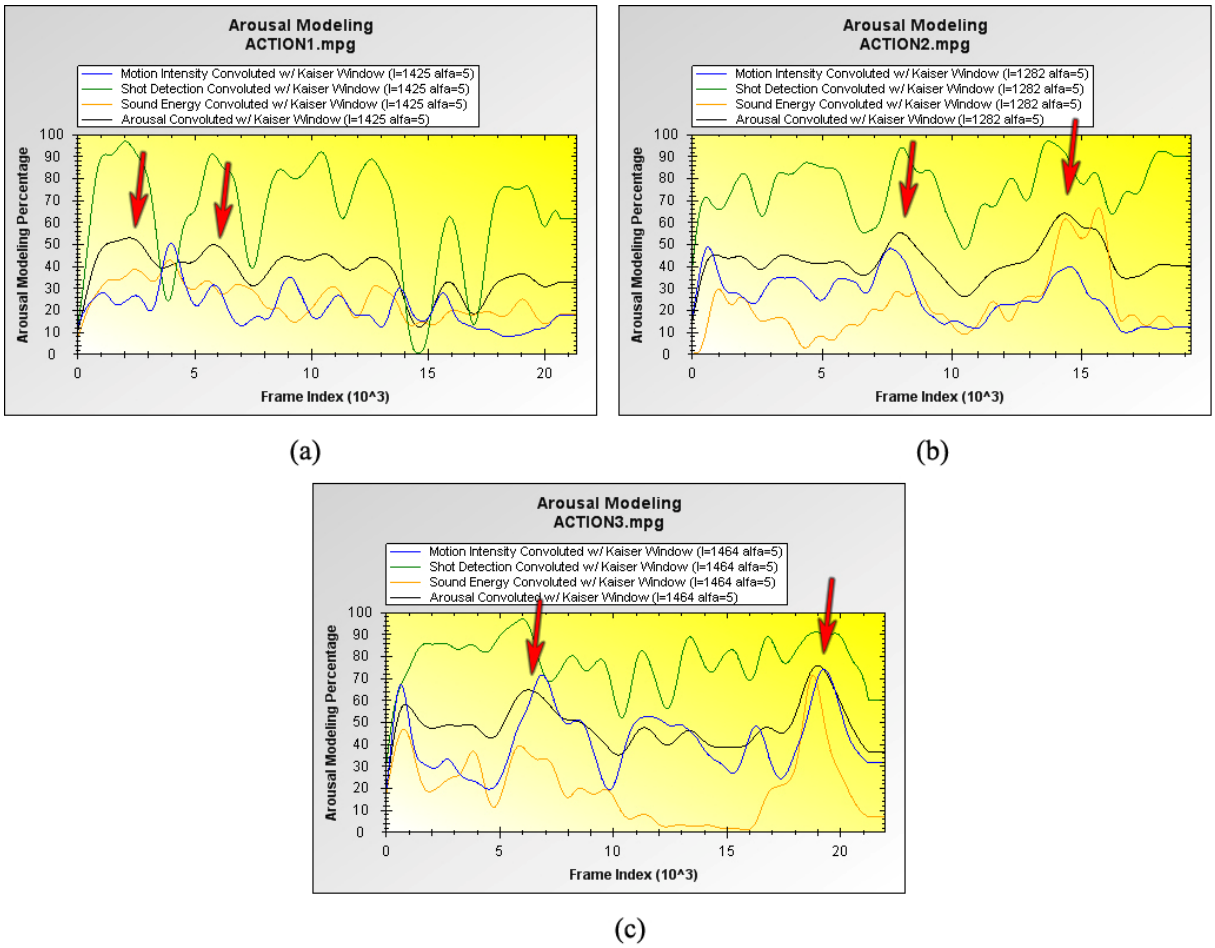

Figure 62 – Entertainment content final arousal charts: (a) ACTION1.mpg; (b) ACTION2.mpg; and (c) ACTION3.mpg.

4. **ACTION1 –** From all chosen contents, this revealed to be the one with less segments of interest; it just includes a fighting scene, right at the start, near frame 2000, and a scene with high rhythm (a pursuit), near frame 6000. Those segments are the ones rightly included in the "TopHighlights" summary. The rest of the content is quite constant in terms of excitement until around frame 14000 when two entire dialog scenes between two characters take place, originating a break in the arousal curve. No segments are chosen from frame 14000 ahead for none of the summaries.
5. **ACTION2 –** The second action content has one significantly most exciting moment, around frame 14000 which is a high suspense scene. The "TopHighlights" summary includes only that segment. Other relevant/exciting events occur, namely, a police apprehension around frame 7500 which is included in the "KeyPoints" summary. The remaining of the content is quite constant in excitement, registering only a significant break in the arousal curve near frame 11000 when a dialog scene occurs. Successfully, that scene was left out even for the longer summary, "ExtendedSummary".
6. **ACTION3 –** The third action content has two clearly more exciting moments, one starting around frame 5500, representing a gunfire scene, and another, near frame 18500, related to a police pursuit. Those two segments are present in the "TopHighlights" summary. The "Key Points" summary provides some context for these segments, increasing their duration. From frame 9000 to frame 18000, the content is calmer and that is reflected in the summaries as no segment in this zone was included in none of the summaries.

Following this informal evaluation, for the reader to get acquainted with the test material and the arousal charts, the results obtained with the user evaluation study will be described in the following.

Table 14 and Table 15 contain the total average results as well as the average results for each summary type, obtained for each question of this user evaluation study, based on the answers provided by the 13 volunteer subjects involved.

- **Results and Analysis for Question 1**

Question 1 *The summary viewed satisfies its type definition, i.e. contains the top most relevant/exciting 10%, 25% and 50% of the original content?*					
	a) Not at all	b) Badly	c) Reasonably	d) Mostly	e) Totally
Sports content					
TopHighlights	0,00%	0,00%	23,08%	51,28%	25,64%
KeyPoints	0,00%	0,00%	7,69%	48,72%	43,59%
ExtendedSummary	0,00%	0,00%	7,69%	38,46%	53,85%
Entertainment content					
TopHighlights	0,00%	17,95%	28,21%	28,21%	25,64%
KeyPoints	0,00%	2,56%	7,69%	58,97%	30,77%
ExtendedSummary	0,00%	2,56%	0,00%	23,08%	74,36%
TopHighlights Average	0,00%	8,97%	25,64%	39,74%	25,64%
KeyPoints Average	0,00%	1,28%	7,69%	53,85%	37,18%
ExtendedSummary Average	0,00%	1,28%	3,85%	30,77%	64,10%
Sport content Average	0,00%	0,00%	12,82%	46,15%	41,03%
Entertainment content Average	0.00%	7,69%	11,97%	36,75%	43,59%
Total Average	0,00%	3,85%	12,39%	41,45%	42,31%

Table 14 – Evaluation results for Question 1.

Table 14 presents the results for question 1. Regarding question 1, the average results show that 41,45% and 42,31% of the inquired subjects considered that the viewed summaries 'Mostly' or 'Totally' satisfied its summary type definition, this means 10% for the TopHighlights" summary and so on. None of the inquired subjects considered that the summary did not satisfy at all its type definition and only 3,85% considered that it satisfied 'Badly'.

Analyzing the results by summary type, all summaries had positive results, with the results of the 'Mostly' and 'Totally' scores adding always to more than 65%, which was the poorest result, achieved for the "TopHighlights" type summary. The "KeyPoints" and "ExtendedSummary" type summaries presented even better results, with the 'Mostly' and 'Totally' scores adding to the order of 90%.

In terms of content, the results were slightly more satisfactory, on average, for the sports content than for the entertainment content. It is believed that this is related with the fact that exciting moments in sport broadcasts are easier to identify, this means more obvious from the semantic point of view, than in TV series and so do not depend so much on the viewer's interpretation.

- **Results and Analysis for Question 2**

Question 2 *Any relevant segments were ruled out of the viewed summary, for each summary type, i.e. Top Highlights, Key Points and Extended Summary?*					
	a) All	b) Many	c) Some	d) Few	e) None
Sports content					
TopHighlights	0,00%	10,26%	30,77%	41,03%	17,95%
KeyPoints	0,00%	0,00%	15,38%	56,41%	28,21%
ExtendedSummary	0,00%	5,13%	7,69%	41,03%	46,15%
Entertainment content					
TopHighlights	0,00%	17,95%	43,59%	23,08%	15,38%
KeyPoints	0,00%	2,56%	17,95%	46,15%	33,33%
ExtendedSummary	0,00%	2,56%	7,69%	33,33%	56,41%
TopHighlights Average	0,00%	14,10%	37,18%	32,05%	16,67%
KeyPoints Average	0,00%	1,28%	16,67%	51,28%	30,77%
ExtendedSummary Average	0,00%	3,85%	7,69%	33,33%	56,41%
Sport content Average	0,00%	5,13%	17,95%	46,15%	30,77%
Entertainment content Average	0,00%	7,69%	23,08%	34,19%	35,04%
Total Average	0,00%	6,41%	20,51%	40,17%	32,91%

Table 15 – Evaluation results for Question 2.

The results gathered for question 2 are presented in Table 15. Regarding question 2, on average, 40,17% and 32,91% of the subjects considered that only 'Few' and 'None' of the relevant segments were excluded from the viewed summaries for the various summary types; the two top scores add to a total of 73% meaning that the summaries rarely missed what they should include.

Performing an analysis by summary type, the "TopHighlights" summary type is the one that shows the poorer results, as it is the smaller one in duration and, therefore, the one with higher probability to exclude any relevant segment. Even so, it presents an added "TopHighlights" average result for the 'Few' and 'None' scores of near 50% (significantly better results for sports content) with the majority of the subjects considering that 'Some' segments were excluded from the summary and only 14,10% considering than 'Many' relevant segments were excluded. The "KeyPoints" and "ExtendedSummary" summaries performed quite well in question 2, as in question 1, with combined results of 'Few' and 'None' near 82% and 90%, respectively.

Question 2 presented a greater discrepancy of results between the sports content and entertainment content, with the sports summaries achieving better results than the entertainment content summaries. Once more, it is believed this is related with the fact that it is harder to distinguish points of interest common to all users in entertainment content because of the increased subjectivity regarding the identification of the exciting moments. For sports content, for example a football match, a goal is considered an event of top importance, and therefore, it is supposed to be included in the summaries by the great majority of users. For entertainment content, the situation tends to be slightly different as, for example, an emotive scene, is more probable not to reveal the same amount of importance for all users, as context, even unconsciously, takes part in the user's choice of segments that should be included in the summaries. Therefore, for entertainment content, the "quality" of the summary is more permeable to each user's interpretation than in sports content where the main events, as goals in football, are normally and unambiguously identified by every user.

This chapter presented the performance evaluation of the developed system which was obtained by conducting an user evaluation study aiming to evaluate: i) how good was the summarization experience provided by the created summaries, according to each one's type; and ii) if any relevant segments were excluded from any of the summaries presented to the user. The main purposes of this chapter were thus to describe the test objectives, its methodology and a deep analysis of the obtained evaluation results.

.

Chapter 7

Conclusions and Future Work

Chapter 7 finalizes this report by presenting to the reader a brief summary of the solution developed to address the automatic summarization problem as well as some conclusions followed by some prospects for future work.

7.1 Summary and conclusions

Chapter 1 introduced the problem addressed in this Thesis, highlighting that the recent boom of audiovisual content availability as well as its common use in people's every day life, justifies the development of a system capable of automatically summarize any kind of audiovisual content. Chapter 2 structured the problem at hand, automatic audiovisual summarization, by dividing it into two main technical areas: solutions aiming to summarize specific content, as a specific sport broadcast event, and solutions intended to summarize generic content, i.e. any kind of audiovisual content. As the solution to be developed intended to have the widest possible applicability, the generic content approach was chosen for this Thesis.

Chapter 3 presented the architecture of the developed solution as well as a functional description of each of its modules. It also motivated the decision to adopt an affective approach to achieve the goal of successfully summarizing any kind of audiovisual content. In this context, 'affective' regards the viewer's interest and involvement along the content. To do so, an arousal modeling solution was developed, largely based on the work of A. Hanjalic from the Technical University of Delft [5][6][7]. The arousal modeling solution intended, as the name suggests, modeling the arousal experienced by the viewer during the content viewing, to extract the segments considered most exciting for the final summary. To wide the solution's applicability range, it was decided to create as main output of the system, an MPEG-7 compliant hierarchical summary description, which should allow some interoperability with other developed metadata systems, namely other MPEG-7 players that from the created summary description can also present an

audiovisual summary to its users. In addition to the summary description, the user can also, if it is his/her so wishes, to export summaries to MPEG-1 compliant streams, using them after for his/her own purposes.

Chapter 4 described in detail the processing algorithms developed to perform the automatic summarization. It was made clear that three main steps have to be taken in order to produce the final summary. The first step has to extract low-level features, in this case three low-level features, to serve as the basis for the arousal modeling. Those features were chosen based on their influence on the viewer's reactions; the selected features were motion, shot cuts density and sound energy. The second step was related to the arousal metrics computation with the goal to produce three arousal curves, one for each feature, based on the information collected in the first step. A final arousal curve, fusing those three curves, in a weighted manner, and representing the arousal experienced by the viewer along the content is, finally, created. The final step was the creation of the MPEG-7 compliant hierarchical summary description from the final arousal curve produced. The hierarchical summary description identifies segments of the original content, labeled accordingly to its arousal level (in four levels), allowing the user to create three types of summaries: a summary including the "Top Highlights" of the content, its "Key Points", or an "Extended Summary". The user can also choose to create a summary with a specified, desired length, obtaining a summary with the most exciting moments with the specified time duration.

Chapter 5 intended to offer all the information needed by the user to make a proper usage of the developed application, explaining briefly its high-level software structure, providing a detailed installation guide, and brining an in-depth explanation of the application's GUI.

Chapter 6 presented the user evaluation study conducted to evaluate the solution's performance, describing the test's objectives, its methodology and, obviously, a deep analysis of the obtained results.

Resuming, the main objective of this work was to develop an application capable of summarizing any kind of audiovisual content by applying an arousal modeling process, producing a final arousal curve aiming to represent the viewer's excitement during the content. From that curve, a hierarchical summary description of the content, organizing its segments according to their arousal level is created. The summaries produced intend to represent the most exciting/relevant events on the content and, of course, to not exclude any relevant events.

Based on the results obtained in Chapter 6, it is possible to conclude that, for the majority of the cases, the application developed allows reaching good summarization results, being able to produce summaries including the most exciting/relevant segments from the original content. Even so, it was also possible to conclude that the performance results differ significantly from content type to content type. More 'objective' content types, as sports, where 'a goal is a goal', may present better results as its more exciting events are more easily recognizable than for more subjective content types, as entertainment. Entertainment results were more permeable to the user's interpretation of the story and its relevant segments. Users have different opinions about the most exciting segments more often in entertainment content than in sports. About the summary types, the choice for three types of summaries appeared to be right one as the obtained results in the user evaluation study revealed considerable different marks, in average, given by the users for each summary regarding each question, meaning that significantly different summaries were constructed. The results and the feedback given by the test subjects were taken as promising, indicating that automatic audiovisual summarization tools can be already useful in today's multimedia world. Despite the positive results, some improvements can be made in the developed application. These improvements were left for future work and are discussed in the next section.

7.2 Future work

Despite the encouraging results obtained in the user evaluation study, the solution developed still leaves room for improvement.

- **Shot detection algorithm revision** – The application was developed with a modular design, meaning that for example the low-level extraction algorithms can be replaced or complemented without having to change the application's global architecture. The motion and sound information algorithms seem to present the expected results but the shot cut detection algorithm can be improved as it presents some false positives and negatives for the majority of contents. Although the default values for sensibility and frame step normally present satisfactory results, to achieve the best results possible, the process for shot detection has, sometimes, to be done by several tries, trying to find the sensibility and frame step that better suits the content in question. As eventual future work, a revision of the shot detection algorithm implemented could, therefore, be performed.
- **Low-level algorithms performance analysis and research for alternatives** – Besides shot detection, and aiming to achieve better computational performance in the low-level information extraction processes, other motion and sound extraction algorithms can also be studied in the future.
- **Additional features** – In terms of the developed application, some additional features can be eventually added in the future, namely a summarization wizard able to guide the user step by step through the entire process for summarization and possibly a more complete summary player, capable of giving the user the ability to leap from segment to segment inside the summary, perhaps by clicking in the respective segment in the chart. These are only two of many examples of features that in the future can be added to the developed application.
- **More sophisticated fusion** – Despite presenting good results and fulfilling its function in a satisfactory manner, the fused arousal metric can be reviewed or even other fusion metrics can be studied in order to enhance the system's performance. A more sophisticated fusion process of the low-level features information can, for instance, present arousal curves more adjusted to the content type and therefore capable of delivering better summaries to the final user.
- **Performance evaluation** – Regarding performance evaluation more complete tests can be done in the future, mainly with more types and longer pieces of content to evaluate how the system's performs under those different circumstances. Besides sport and entertainment content, the "quality" of the summaries can be evaluated for other sorts of contents, since news clips to documentaries or music video clips. A more complete set of questions can also be placed to the users, in order to evaluate, in more detail each of the created summaries. Finally, the test can be done in a larger scale, with a higher number of subjects participating, collecting, in this manner, much more samples and, therefore, more accurate results.

In conclusion, there is still a lot of research to do in the field of automatic audiovisual summarization, specially if the semantic value of the content is to be taken into account … automatically modeling semantics still is, and very likely will be for a long time, a difficult task …

Appendix A

User Evaluation Study Instructions

This annex includes the instructions sheet given to the test subjects while performing the test. It defines first the objective of the test and defined after the sequence of steps to be performed by the test subjects.

Audiovisual summarization application evaluation test

1. Test objective

This test is destined to evaluate the final product of a project developed with the goal of automatically producing audiovisual summaries from generic content.

The test will take, approximately 2 hours and consists on the visualization of 6 videos and their respective summaries (3 per content). After visualizing each summary, the evaluator should score the "quality" of the created summary, accordingly with 2 questions formulated below.
The videos are split in 2 classes, one with sport content and other with entertainment content.

2. Useful definitions for this test

There are 3 types of summaries to be evaluated for each audiovisual content. Their definitions are:

1. Top Highlights – Summary intended to provide to the user only the segments considered indispensable in the content; corresponds to the top 10% most relevant/exciting segments of the audiovisual content.

2. Key Points – Summary longer than the previous that pretends to include, not only the most relevant segments but also some contextualization; corresponds to the top 25% most relevant/exciting segments of the audiovisual content.

3. Extended Summary – The longest summary which pretends to give to the user an extended perspective of the audiovisual content, excluding the dullest segments; corresponds to the top 50% most relevant/exciting segments of the audiovisual content.

3. Test methodology

Using the provided web page, go through the next steps with attention and concentration:

3.1 SPORT CONTENT

For each sport sequence, do:

1. Open and visualize the original content, by clicking in the respective link, starting form "Basketball" content.

2. For the original content at hand, visualize 1 only time its three possible summaries, i.e.:

2.1 – Top Highlights

2.2 – Key Points

2.3 – Extended Summary

3. Answer to questions 1 and 2, marking with a cross (X), in tables 1 and 2, the desired classification mark, for each one of the three summaries just visualized, i.e. Top Highlights, Key Points e Extended Summary.

4. Go back to point 1. for the 2 lasting contents until tables 1 and 2 are completely filled for the 3 sport contents, i.e. Basketball, Football1 e Football2.

Question 1 – The summary viewed satisfies its type definition, i.e. contains the top most relevant/exciting 10%, 25% and 50% of the original content?

Table 1
Question 1 - Sport content evaluation

	a) Not at all	**b) Badly**	**c) Reasonably**	**d) Mostly**	**e) Totally**
Vídeo 1 – Basketball					
BASKETBALL – "TopHighlights" Summary					
BASKETBALL – "KeyPoints" Summary					
BASKETBALL – "ExtendedSummary" Summary					
Video 2 – Football1					
FOOTBALL1 – "TopHighlights" Summary					
FOOTBALL1 – "Key Points" Summary					
FOOTBALL1 – "Extended Summary" Summary					
Video 3 – Football2					
FOOTBALL2 – "TopHighlights" Summary					
FOOTBALL2 – "Key Points" Summary					
FOOTBALL2 – "Extended Summary" Summary					

Question 2 – Any relevant segments were ruled out of the viewed summary for each summary type, i.e. Top Highlights, Key Points and Extended Summary?

Table 2
Question 2 – Sport content evaluation

	a) All	**b) Many**	**c) Some**	**d) Few**	**e) None**
Vídeo 1 – Basketball					
BASKETBALL – "TopHighlights" Summary					
BASKETBALL – "KeyPoints" Summary					
BASKETBALL – "ExtendedSummary" Summary					
Video 2 – Football1					
FOOTBALL1 – "TopHighlights" Summary					
FOOTBALL1 – "Key Points" Summary					
FOOTBALL1 – "Extended Summary" Summary					
Video 3 – Football2					
FOOTBALL2 – "TopHighlights" Summary					
FOOTBALL2 – "Key Points" Summary					
FOOTBALL2 – "Extended Summary" Summary					

3.2 ENTERTAINMENT CONTENT

For each entertainment sequence, do:

1. Open and visualize the original content, by clicking in the respective link, starting form "Action1" content.

2. For the original content at hand, visualize 1 only time its three possible summaries, i.e.:

2.1 – Top Highlights

2.2 – Key Points

2.3 – Extended Summary

3. Answer to questions 1 and 2, marking with a cross (X), in tables 3 and 4, the desired classification mark, for each one of the three summaries just visualized, i.e. Top Highlights, Key Points e Extended Summary.

4. Go back to point 1. for the 2 lasting contents until tables 1 and 2 are completely filled for the 3 sport contents, i.e. Action1, Action2 e Action3.

Question 1 – The summary viewed satisfies its type definition, i.e. contains the top most relevant/exciting 10%, 25% and 50% of the original content?

Table 1
Question 1 - Entertainment content evaluation

	a) Not at all	b) Badly	c) Reasonably	d) Mostly	e) Totally
Vídeo 1 – Action1					
ACTION1 – "TopHighlights" Summary					
ACTION1 – "KeyPoints" Summary					
ACTION1 – "ExtendedSummary" Summary					
Video 2 – Action2					
ACTION2 – "TopHighlights" Summary					
ACTION2 – "Key Points" Summary					
ACTION2 – "Extended Summary" Summary					
Video 3 – Action3					
ACTION3 – "TopHighlights" Summary					
ACTION3 – "Key Points" Summary					
ACTION3 – "Extended Summary" Summary					

Question 2 – Any relevant segments were ruled out of the viewed summary for each summary type, i.e. Top Highlights, Key Points and Extended Summary?

Table 2
Question 2 – Entertainment content evaluation

	a) All	b) Many	c) Some	d) Few	e) None
Vídeo 1 – Action1					
ACTION1 – "TopHighlights" Summary					
ACTION1 – "KeyPoints" Summary					
ACTION1 – "ExtendedSummary" Summary					
Video 2 – Action2					
ACTION2 – "TopHighlights" Summary					
ACTION2 – "Key Points" Summary					
ACTION2 – "Extended Summary" Summary					
Video 3 – Action3					
ACTION3 – "TopHighlights" Summary					
ACTION3 – "Key Points" Summary					
ACTION3 – "Extended Summary" Summary					

Thank you for your cooperation,
Nuno Matos

References

1. YouTube: http://www.youtube.com
2. Alexa – The Web Information Company: http://www.alexa.com
3. Wikipedia: http://en.wikipedia.org
4. MSN Web Portal: http://www.msn.com
5. A. Hanjalic and L. Q. Xu, "User-oriented video content analysis", IEEE Workshop on Content-based Access to Image and Video Libraries (CBVAIL '01), pp. 50-57, Kauai, HW, USA, Dec. 2001.
6. A. Hanjalic, "Extracting moods from pictures and sounds", IEEE Signal Processing Magazine, vol. 23, nº 2, pp. 90-100, Mar. 2006.
7. A. Hanjalic and L. Q. Xu, "Affective video content representation and modeling", IEEE Transactions on Multimedia, vol. 7, nº 1, pp 143-154, Feb. 2005.
8. Y. F. Ma, X. S. Hua, L. Lu and H. J. Zhang, "A generic framework of user attention model and its application in video summarization", IEEE Transactions on Multimedia, vol. 7, nº 5, pp. 907-918, Oct. 2005.
9. A. Hanjalic, "Adaptive extraction of highlights from a sport video based on excitement modeling", IEEE Transactions on Multimedia, vol. 7, nº. 6, pp. 1114-1122, Dec. 2005.
10. A. Jaimes, T. Echigo, M. Teraguchin and F. Satoh, "Learning personalized video highlights from detailed MPEG-7 metadata", in Proc. IEEE ICIP, vol. 1, pp. 133-136, Rochester, New York, USA, Sep. 2002.
11. I. H. Witten and E. Frank, "Data mining: practical machine learning tools and techniques with java implementations", Morgan Kaufman, New York, 1999.
12. Cross-validation: http://en.wikipedia.org/wiki/Cross-validation
13. Neural Networks: http://en.wikipedia.org/wiki/Neural_network
14. A. Ekin, A. Murat Tekalp and R. Mehrotra, "Automatic soccer video analysis and summarization", IEEE Transactions on Image Processing, vol. 12, nº 7, pp. 796-807, Jul. 2003.
15. K.N. Plataniotis and A. N. Venetsanopoulos, "Color image processing and applications", Berlin, Germany: Springer-Verlag, pp. 25-32 and 260-275, 2000.
16. G. Millerson, "The technique of television production", 12th Ed. New York: Focal, March 1990.

17. A.M. Ferman and A.M. Tekalp, “A fuzzy framework for unsupervised video content characterization and shot classification”, J. Electron Imag., vol.10, no.4, pp. 917-929, Oct. 2001.
18. H. Pan, P. van Beek and M.I. Sezan, “Detection of slow-motion replay segments in sports video for highlights generation”, in Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing (ICASSP), pp. 1649-1652, Salt Lake City, Utah, USA, May 2001.
19. X. S. Hua, S. Li, “Contrast-Based image attention analysis by using fuzzy growing”, in Proc. 11th ACM Int. Conf. Multimedia, pp. 374-381, Berkeley, CA, USA, Nov. 2003.
20. S. Z. Li et al, “Statistical learning of multi-view face detection”, in Proc. of European Conf. Computer Vision, vol. 4, pp-67-81, Copenhagen, Denmark, May – Jun. 2002.
21. D. J. Lan, Y. F. Ma and H. J. Zheng, “A novel motion-based representation for video mining”, in Proc. IEEE Int. Conf. Multimedia and Expo, vol. 3, pp 469-472, Baltimore, MD, USA, Jul. 2003.
22. H. L. Wang and L. F. Cheong, “Affective understanding in film”, IEEE Transactions on Circuits and Systems for Video Technology, vol. 16, nº 16, pp. 689-704, Jun. 2006.
23. B. H. Detenber, R. F. Simons, and G. G. Bennett, “Roll ’em!: The effects of picture motion on emotional responses”, J. Broadcasting and Electron. Media, vol. 21, pp. 112–126, 1997.
24. R. Simons, B. H. Detenber, T.M. Roedema, and J. E. Reiss, “Attention to television: Alpha power and its relationship to image motion and emotional content”, Media Psychol., vol. 5, pp. 283–301, 2003.
25. MPEG MDC decoder: http://iielab-secs.secs.oakland.edu/demossoftware/MDC.html
26. Shot Detection in Video Sequences: http://users.isr.ist.utl.pt/~jsm/teaching/piv/shotdetection/shotdetection.htm
27. J. Yuan, H. Wang, L. Xiao, W. Zheng, J. Li, F. Lin and B. Zhang, “A formal study of shot boundary detection”, IEEE Transactions on Circuits and Systems for Video Technology, vol. 17, nº 2, pp. 168-187, Feb. 2007
28. J. Zheng, F. Zou and M. Shi, “An efficient algorithm for video shot boundary detection”, in Proc. International Symposium on Intelligent Multimedia, Video and Speech Processing, pp. 266-269, Hong-Kong, China, Oct. 2004.
29. W. J. Heng and K. N. Ngan, “Integrated shot boundary detection using object-based technique”, in Proc. International Conference on Image Processing, vol.3, pp. 289-293, Kobe, Japan, Oct. 1999.
30. DirectShow: http://msdn2.microsoft.com/en-us/library/ms783323.aspx
31. MPEG to WAV conversion: http://www.codeproject.com/directx/Mpeg2WavConversion.asp
32. WaveUtility classes: http://www.codeproject.com/csharp/steganodotnet15.asp
33. Kaiser window: http://en.wikipedia.org/wiki/Kaiser_window
34. Bessel function: http://en.wikipedia.org/wiki/Bessel_function
35. B. S. Manjunath, P. Salembier and T. Sikora, “Introduction to MPEG-7: Multimedia Content Description Interface”, Wiley, 2002.
36. MPEG-7 overview: http://www.chiariglione.org/mpeg/standards/mpeg-7/mpeg-7.htm
37. MPGTX project: http://mpgtx.sourceforge.net/
38. Microsoft Visual Studio: http://msdn2.microsoft.com/en-gb/vstudio/default.aspx
39. Microsoft Visual Studio: http://en.wikipedia.org/wiki/Microsoft_Visual_Studio
40. Microsoft .NET Framework: http://www.microsoft.com/net/
41. Microsoft .NET Framework: http://en.wikipedia.org/wiki/.NET_Framework
42. ZedGraph Library: http://zedgraph.org/wiki/index.php?title=Main_Page

43. Microsoft's .NET Framework 2.0 download page: http://www.microsoft.com/downloads/info.aspx?na=90&p=&SrcDisplayLang=en&SrcCategoryId=&SrcFamilyId=0856eacb-4362-4b0d-8edd-aab15c5e04f5&u=http%3a%2f%2fdownload.microsoft.com%2fdownload%2f5%2f6%2f7%2f567758a3-759e-473e-bf8f-52154438565a%2fdotnetfx.exe
44. Winzip: http://www.winzip.com/index.htm
45. Winrar: http://www.rarlab.com/